VYTAUTAS MAGNUS UNIVERSITY

INSTITUTE OF MATHEMATICS AND INFORMATICS

Svajonė BEKEŠIENĖ

PARALLEL COMPUTATION SYSTEM FOR MATHEMATICAL MODELING OF ELECTRONIC STRUCTURE OF EXPLOSIVE MATERIALS

Doctoral Dissertation

Physical Sciences (P 000)
Informatics (09 P)
Informatics, System Theory (P 175)

Vilnius 2008

Dissertation has been prepared in the period from 2002 to 2007 at the Institute of Mathematics and Informatics.

Scientific Supervisor:
Doc. Dr. Vytautas KLEIZA (Kaunas University of Technology, Institute of Mathematics and Informatics, physical science, informatics – 09 P).

VYTAUTO DIDŽIOJO UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Svajonė BEKEŠIENĖ

LYGIAGREČIŲJŲ SKAIČIAVIMŲ SISTEMA SPROGSTAMŲJŲ MEDŽIAGŲ STRUKTŪRAI TIRTI

Daktaro disertacija

Fiziniai mokslai (P 000)
Informatika (09 P)
Informatika, sistemų teorija (P 175)

Vilnius 2008

Disertacija rengta 2002-2007 metais Matematikos ir Informatikos institute.

Mokslinis vadovas:
doc. dr. Vytautas KLEIZA (Kauno technologijos universitetas, Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09 P).

ABSTRACT

Intensive development of science and the dissemination of human ideas into all new areas pose new, as a rule more difficult, tasks. One of the serious human problems is terrorism. Actually, it became urgent to prevent and stop the terror acts. Nowadays it became important to have such technical instrumentality that would enable immediately, safely and desirably in remote way detect and identify dangerous chemical materials. To solve this problem, it is important to include the power of scientists in each state of science. Scientists of computational discipline simulate by supercomputers phenomena that are too complex to be reliably predicted by theory and too dangerous or expensive to be reproduced in the laboratory.

The objective of this research work for quantum chemical investigations was to design, creation and adaptation of PC cluster consisted of commodity hardware.

The organization of the dissertation comprises five parts.

The first part is Introduction. It presents the main information on the research, such as the statement of the problem and actuality, research question and objectives, motivation of research, methodology used in the research, research issues and results, its scientific novelty, practical importance and approbation, in which the main results of the research were introduced.

The second part describes the theory of computational quantum mechanical methods, used for investigating molecular electronic structure and dynamical properties of many atomic structures of explosives.

The third part proposes the cluster system environments involved in this study, and the PC cluster built in a quite functional compound of 16 PC's with 32 CPU (processors) and 100 Mbps Fast Ethernet network. The self-made cluster system environment was tested and evaluated by NAS NPB benchmarks. The analysis of the results obtained is presented here.

The fourth part proposed the SCore PC cluster environment adaptation for the investigation of explosive molecules. It also describes the GAMESS computer code implementation and development in PC cluster. Then the results of tests, used for study of quantum chemical investigations and the selected NAS NPB benchmarks are presented.

The fifth part highlights the investigation results that evaluated the build SCore PC cluster quality in practice. The investigation results corroborate that the such type PC cluster allows us to solve urgent quantum chemical problems at necessary level and to achieve quantitative results comparable with the experimental ones.

ACKNOWLEDGMENT

I wish to thank all whom have helped support and contribute to this effort. Firstly, I thank my supervisor Assoc. Prof. Dr. Vytautas Kleiza for directing this dissertation. I am grateful to him for his understanding, help and valuable advices, also for his time, his patience and the opportunity for being one of his graduate students.

I was given very valuable and insightful advice from Professor Yutaka Ishikawa, the head of the Parallel Distributed System Software Laboratory of RWCP (Japan). I appreciate his help in every way especially his quick and complete e-mail responses and kindly guidance in the first tentative steps towards new technology. My interest was kindled by his passion for study.

I would like to express my gratitude to Prof. Gintautas Dzemyda, the director of the Institute of Mathematics and Informatics, for valuable criticism, comments and discussions.

I would also like to thank Professor Mifodijus Sapagovas for his invaluable advice and warm support.

Thanks to Dr. Jonas Jankauskas, Assoc. Prof. Dr. Albertas Pincevičius and Prof. Evaldas Maldutis for their encouragement throughout this process and allowing me to mix this research work with work at the Mathematical Modeling laboratory at the Department of Applied Sciences, Gen. J. Zemaitis Military Academy of Lithuania.

I thank each of my instructors by this research work for making complex ideas simple and memorable.

Thanks to Danutė Rimeisienė and all personnel of the Institute of Mathematics and Informatics for help, inducement and understanding.

I would like to thank all my friends, who morally supported me during the time that I have been studying.

Last, and most important, I thank my family for their support and sacrifice, for giving me lively example of unconditional love.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Concept
HF	Hartree-Fock method
MCSCF	Multiconfigurational self consisted field
DFT	Density Functional Theory
MP2	Moler pleser perturbation second order
STO-3G	Basis set consists of Slater -type orbitals
6-31G	Basis sets in which atomic orbital is represented by six Gaussians, while the valence orbitals are splitted into parts that are described three and one Gaussians.
6-311G	Basis sets in which atomic orbital is represented by six Gaussians, while the valence orbitals are splitted into parts that are described three, one and one Gaussians. The polarized *-p* and *-d* functions are included.
MO	Molecular Orbital
AO	Atom Orbital
NO_2	Nitrate oxides
IR	Infra red
TNT	2,4,6-trinitrotoluene
TNP	2,4,6-trinitrophenol
PC	Personal computer
CPU	Central Process Unit
SMP	Symmetric Multi-processing
RAM	Randomize Area Memory
DSM	Distributed Shared Memory
SAN	System Area Networks
MPI	Message Passing Interface
PVM	Parallel Virtual Machine
EHA	Ethernet Hardware Address
MAC	Media Access Control address
NIC	Network Interface Card.
PM	Parallel Memory
OS	Operating System
MBCF	Memory Based Communication Facilities
SHMEM	Shared Memory
LINPACK	Linear Package
GAMESS	General Atomic and Molecular Electronic Structure System
NAS NPB	Numerical Aerospace Simulation Network Parallel Benchmarks
Gflops	1Gflops=10^9 floating-point operations per second
Mflops	1Mflops=10^6 floating-point operations per second
MIMD	Multiple Instruction Multiple Data
MPP	Massively Parallel Processors
RWCP	Real World Computing Partnership

INTRODUCTION

1.1. Statement of the Problem and its Relevance

Intensive development of science and the dissemination of human ideas into all new areas side-by-side with the solution of the current problems constantly invokes a flow of questions and poses new, as a rule more difficult, tasks. During the first super computers coming it seemed, that the increase of their speed in 100 times will allow to solve the majority of problems, however Gflops (1Gflops=10^9 floating-point operations per second) the productivity of modern super computers today is obviously insufficient for many scientists. Seism reconnaissance and weather forecast, modeling of chemical compounds, research of a virtual reality are not a complete list of areas of science, in which researchers take every opportunity to speed up the performance of programs.

Fast computers stimulated a fast growth of new ways of doing science. Scientists of computational discipline simulate by supercomputers phenomena that are too complex to be reliably predicted by theory and too dangerous or expensive to be reproduced in the laboratory. Success in computational science caused a demand for supercomputing resources that have risen sharply over the past ten years.

One of the serious human problems is terrorism. Actually, it became urgent to prevent and stop the terror acts. To solve this problem, it is important to include the power of scientists in each state of science. The order to realize these tasks successfully it is important to have, consummate and develop absolutely new and progressive methods transportation and distribution control methods for various chemical materials. Nowadays it became important to have such technical instrumentality that would enable immediately, safely and desirably in remote way detect and identify dangerous chemical materials.

Molecules having nitro groups are known as explosives. If we want to found the existent of these materials there is way in chemical methods (how these materials react with test reagents). However, sometimes when the amounts of explosive materials are small, chemical methods are no effective. In this way, we can try the physical spectroscopic methods. This possibility is evidential, because the molecules of materials have only them typical spectrum, to be precise, the parts of spectrum. Some substances of spectroscopic type, such as Raman spectroscopic, let us register very small amounts (some of molecules) of materials. Problems are related with these investigations, because materials are not in vacuum and spectra of molecules are mixing and distorting, so that the possibility to identify the materials is very vague. Notwithstanding these problems, it is known that spectra of molecules and his pats are an inherent only in them.

2,4,6-trinitrotoluene (TNT) is better known by its initials. TNT is an important explosive, since it can very quickly change from a solid into hot expanding gases. TNT has 3 NO_2 groups (from nitric acid) to toluene. TNT is explosive because it contains the elements carbon, oxygen and nitrogen, which means that when the material is burning it produces highly stable substances (CO, CO_2 and N_2) with strong bonds, thus releasing a great deal of energy. TNT manufacturing involves stepwise nitration of toluene in a three-stage batch process or continuous process producing mono-, di-, and finally trinitrotoluene, respectively.

The 2,4,6-trinitrophenol (known as picric acid) and nitro phenols are a class of materials that are classified as secondary explosives and are the ones most frequently used in making ammunitions. Only few of their molecular structures were examined experimentally. Hence, it is necessary to find the geometries of some energetic materials calculated by *ab initio* methods.

The spectra of such multi atom molecules include a wide range area of IR spectra. To detect such molecules, we have to classify IR spectra, assigning by oscillation modes and self-frequencies. Theoretical research is concerned with computation and investigation of multidimensional potential energy surface of molecules by means of accurate quantum mechanical computational methods and, on the other hand, with solving of the multidimensional vibration Schrödinger equation of the molecule.

The basis for investigating the explosive molecular electronic structure or geometrical data is quantum mechanics, however, it is impossible to achieve quantitative results comparable with experimental without power of supercomputer or parallel computer cluster. That is why these tasks was solved on the local area PC cluster with the SCore cluster system software. The results obtained show that the cluster is a good tool for such tasks.

1.2. Research Issues and Objectives

The aim of the dissertation was to create a personal computer cluster based on commodity hardware running Linux OS and free – license software. The created computational facilities were applied to investigate the electronic structure and vibrational spectra of explosive molecules by means of *ab initio* quantum mechanical computational methods. One of methods of investigations was a theoretical research of the stability and spectra of explosives. Unfortunately, the experiment cannot show to which piece of molecule this or that line of spectrum belonged. That is why the present theoretical investigations have become like the standards. Another method of investigations was to realize parallel computations on a PC cluster. To determine a high accuracy of the results obtained by cluster, some of them were evaluated with theoretical and experimental public results by other authors published in science papers.

The objectives of the dissertation are as follows:

- To create a Beowulf type cluster for analyzing and evaluating the environment and to determine the performance.
- To build a personal computer cluster with the SCore parallel system software, to analyze and evaluate such environment for realizing the created cluster for parallel calculations.
- To collect, analyze, and compare two environments with intention to propose an efficient parallel environment using the new parallel technology.
- To implement and adapt the GAMESS computer code for parallel calculations by the cluster after collecting, analyzing and comparing existing possibilities.
- To develop GAMESS computer code possibilities by the PC cluster with MPI-2 standard.

- To specialize the created computational facilities for investigating the electronic structure and vibrational spectra of explosive molecules (TNT, TNP) by means of *ab initio* quantum mechanical computational methods.

1.3. Motivation

Topicality of the dissertation is determined by practical needs. In the battle against terrorism, the characterization of explosive materials and their products of fragmentation by determining the spectroscopic signatures play the main role. On the other hand, parallel computers have evolved from experimental contraptions in laboratories to become the everyday tools of computational scientists who need the ultimate in computer resources in order to solve their problems. In other words, the two broad classical branches of theoretical science and experimental science have been joined by computational science, also the clusters based on PC's running Linux OS have become the cheapest "supercomputers" in the academic and commercial field. This type of cluster was created and computational facilities were applied to investigate electronic structure and vibrational spectra of explosive molecules by means of non-empirical quantum mechanical computational methods. One of the methods of investigations is a theoretical research of the stability and spectra of explosives. Incidentally, the experiment cannot show to which piece of molecule this or that line of spectrum belongs, therefore the current the theoretical investigations have become as the standards.

This dissertation is very important from scientific point of view. Research on the evaluation of PC cluster abilities were analyzed by theoretical calculation of the signatures of explosive materials, such as TNT and TNP. The computational results are compared with the experimental ones, obtained by other authors using the spectroscopic technique.

1.4. Research Methodology

The research methodology used in the dissertation includes the following methods: qualitative comparative analysis of different parallel environments, evaluation of the complexity and efficiency of parallel calculations in different environments, methods for testing theory, statistical methods.

- The *methods of qualitative comparative analysis* were used to compare the created cluster with the other existing clusters, after registration in the http://clusters.top500.org/db. In addition, this method was used to analyze the Beowulf and SCore parallel environments.
- The *methodology to evaluate* functionality characteristics of internal quality is used by combining the methods of non-statistical sampling and that to testing theory for Beowulf and SCore environments. The basis of these investigations was NAS NPB tests.

- To make a *detailed analysis* of computational capacity. To explore two environments of the cluster: that of Beowulf and Score type. The standard NAS NPB tests, used in the assessment of supercomputers and cluster capacity, should make the basis for these investigations.

- To make a *comparative analysis* with a new to maximally well-organized the cluster build for quantum mechanical calculations and to implement the GAMESS program for ab initio calculations in its environment.
- In odder to evaluate the efficiency of parallel calculations and GAMESS program in the cluster, to make a *comparative analysis* of pilot calculations of explosive molecules using TCP/IP sockets and MPI-1 both in the parallel Beowulf environment and in the parallel Score environment by realizing the MPI-1-PM technology.
- Using the SCore software trunking technology in the created PC cluster to make a *comparative analysis* of calculation efficiency by varying the number of network interface cards (NIC).
- To apply the MPI-2 library in the study of efficiency of the GAMESS package operation principles in the cluster.
- With a new to evaluate the GAMESS package calculations in the SCore environment after exploring explosive molecules in the cluster, to compare the obtained calculation results with that of other authors, obtained after calculating by the known supercomputers, as well as with the experimental results.

1.5. Research Findings and Results

The main findings and results of the dissertation are as follows:

- A new parallel system – PC cluster has been created and realized for computational facilities by *ab initio* quantum mechanical computational methods.
- The GAMESS computer code abilities have been developed and expanded by means of MPI-2 standard. All that minimized the use of recourses on PC cluster for parallel calculations.
- Has been determined a high accuracy by the well-organized PC cluster, the results of investigations of explosive molecules, have been evaluated and compared with theoretical and experimental public results of other authors published in scientific articles.
- The electronic structure and vibrational spectra of explosive molecules have been established by means of *ab initio* quantum mechanical computational methods.

1.6. Scientific Novelty

This dissertation is the first work, in which an exhaustive quality of characteristics of PC cluster use for the investigating the electronic structure and vibrational spectra of explosive molecules. The precise knowledge of vibrational spectra would allow detecting small amounts of these materials by means of spectroscopic methods. On the other hand, a highly sophisticated quantum chemical computation was used as tests of the SCore type PC cluster performance.

Besides, this dissertation proposes a new approach to evaluate the PC cluster quality in use. It is based on the construction of a quality model. This approach can be considered as more objective than the other known approaches to evaluate quality of realized parallel calculations. Because, first it allows evaluation of the quality in use heterogenic CP cluster quality goals of a particular cluster system software, second, it proposes the evaluation of theoretical results, get after investigations of materials by ab initio quantum chemical methods using General Atomic and Molecular Electronic Structure System quantum chemistry computer code, with experimental. The obtained scientific results are given below.

- A new parallel environment has been proposed and realized for PC clusters, that can be used for investigation the structures of materials. GAMESS computer code implemented and adapted in this environment enables to investigate the electronic structure and vibrational spectra can to compare the results with the experimental ones.
- The General Atomic and Molecular Electronic Structure System quantum chemistry computer code has been accomplished, implemented and improved correctly under the specific environment of SCore PC cluster.
- The GAMESS computer code has been modified by MPI-2 communication libraries.
- Have been obtained the results of explosives: the electronic structure and the geometrical structure (bond length, angles, and dihedral angles) by using the parallel environment with the SCore cluster system software. The results achieved allow a more detailed interpretation of experimental vibrational spectra of the investigated molecules and a more accurate investigation of termochemical reaction dynamics.
- The parallel environment proposed can be applied in modeling other scientific problems.

1.7. Practical Importance

The first purpose of this dissertation is to propose the personal computer (PC) cluster built at the Department of Applied Sciences of the Lithuanian Military Academy with the SCore cluster system software, which ensures the maximal communication performance of cluster systems. The information of technical characteristics and software used in the PC cluster could be applied as a guide in the creation of a new own cluster, because it provides valuable information on the conceptual structure of specification SCore clusters and on the parallel environment features, on which their internal quality and the quality in use depends. In additions, the proposed PC cluster with the SCore cluster system software, internal quality and the quality in use, as well Trunking technology evaluation procedures could be used to compare the clusters of different specification.

The second task solved in the dissertation can be used to efficient the General Atomic and Molecular Electronic Structure System (GAMESS) quantum chemistry computer code by PC clusters. To this end, the MPI-2 library as well as Trunking technology were used.

The third achievement of the dissertation is the created and adapted computational facilities that were applied to investigate the electronic structure and vibration spectra of explosive molecules by means of *ab initio* quantum mechanical computational methods. The results obtained are relevant in the determination and classification of vibrational spectra of explosive molecules.

1.8. Approbation and Publications

The mine results of the dissertation were presented and discussed during the following in Lithuania and abroad hold national and international conferences:

- The 34th Lithuania National Physic Conference, 2001, Vilnius, Lithuania, Vilnius Pedagogical University.
- The International Scientific Seminar "Lygiagrečios skaičiavimo sistemos ir jų programinė įranga", 2001, Vilnius, Lithuania, Lithuania Military Academy.
- The 15th International Scientific Seminar "Лазеры и оптическая нелинейность", 2002, Minsk, Belarus.
- The 9th International Conference "Laser applications in life science", 2002, Vilnius, Lithuania, Vilnius University.
- XLIII- Conference of Lithuanian Mathematicians, 2002, Vilnius, Lithuania, Lithuania Military Academy.
- The 6th World Congress of Theoretically Oriented Chemists WATOC 02, 2002, Lugano, Switzerland.
- The International Conference „Baltic-Norwegian Defense Research seminar", 2003, Norvegija.
- The 35th Lithuania National Physic Conference, 2003, Vilnius, Lithuania.
- XLIV Conference of Lithuanian Mathematicians, 2003, Vilnius, Lithuania, Vilnius Pedagogical University.
- Seminar „AK klasterio tobulinimo ir panaudojimo perspektyvos", 2003, Vilnius, Lithuania, Lithuania Military Academy.
- Seminar „Lygiagrečiųjų skaičiavimų architektūros optimizavimas sprogstamųjų medžiagų savybėms tirti", 2003, Vilnius, Lithuania, Mathematics and Informatics Institute.
- Seminar „Nuotolinės teršalų ir sprogstamųjų medžiagų detekcijos lazeriniais metodais tyrimų analizė", 2003, Vilnius, Lithuania, Lithuania Military Academy.
- XLV Conference of Lithuanian Mathematicians, 2004, Kaunas, Lithuania, Lithuanian University of Agriculture.

- The 4th Scientifical Practical Conference "Informacinės technologijos 2005: aktualijos ir perspektyvos", 2005, Alytus, Lithuania , Alytus College.
- XLVI Conference of Lithuanian Mathematicians, 2005, Vilnius, Lithuania, Vilnius University.
- Conference "Sprogstamųjų medžiagų ir šaudmenų, naudojamų poligonuose karinių pratybų metu, sprogimo produktų poveikio aplinkai vertinimas", 2005, Vilnius, Lithuania, Lithuania Military Academy.

The main results of the dissertation were published in **7** papers. The list of papers is presented in the section "The List of Publications by the Author".

1.9. Synopsis

The dissertation consists of five sections.

The first section is Introduction. It presents the statement of the problem and its topicality, research issues and objectives, motivation of the research, research methodology used, research findings and results, its scientific novelty, practical importance and approbation as well as the main results of the research.

The second section describes the background and needs of parallel processing and introduces the SCore type personal computer (PC) cluster built in the Department of Applied Sciences of Lithuanian Military Academy with SCore cluster software. This section presents an overview of the Real World Computing (RWC) project in the framework of this thesis has done. A particular attention was paid to the SCore Cluster System software developed in the RWC project, because it is the software used in this research work. The SCore Cluster System software has been installed and used as a practical parallel processing environment. This dissertation has been written as a part of development of self-made clusters from commodity hardware. This one of the goals of this thesis is to realize the self made cluster with the SCore cluster system software in the commodity network and to demonstrate its effectiveness in practical use, as well as to demonstrate that the performance of a cluster system is equal to that of a commercially available parallel computer.

The Numerical Aerospace Simulation (NAS) Parallel Benchmarks discussed here include effective algorithms for computational and communicational performance, which were used to evaluate the PC cluster. The tasks solved were used to compare the capabilities of our cluster with other clusters and supercomputers. The cluster is registered in the list of top500 cluster [1].

The third section is analytical, and it describes the research design. The object of analysis is the known approaches for evaluate in the quality of Beowulf and SCore type cluster environments. First, a comparative analysis has shown that, using the SCore cluster system software communication facility PM in Fast Ethernet network the performance of application programs can be comparable with the dedicated cluster network communication. Because of evaluation by the NAS parallel benchmarks, it is obvious that practical high-performance cluster systems can be built using a commodity Fast Ethernet network. Second, the comparative analysis described to show the effectiveness of created computational facilities, applied to investigate the electronic structure and vibrational spectra of explosive molecules by *ab initio* quantum mechanical computational methods. The approaches for evaluating quality of GAMESS quantum chemistry computer code implementations in Beowulf and SCore type cluster environments are studied in detail and compared using the proposed methodology of comparative analysis. It allows preparing valuable recommendations how to use the GAMESS computer code in PC clusters with the SCore cluster system software.

The fourth section presents the background and needs of computational quantum mechanical methods that are widely used for investigations. The computational quantum chemistry is a technique that helps us to resolve, predict, and study new concepts, compounds, reactions and mechanisms. This method is very helpful in compounds that require exceptional worry in their use, e.g., explosives, for reducing the risk to personnel testing and maintenance costs in service. Molecular modeling is the basis of computational quantum chemistry, concerned on predicting the behavior of individual molecules within a chemical system. The molecular modeling let us to obtain molecular characteristics. This section describes the *ab initio* quantum chemistry methods used for these theoretical investigations performed by means of ab initio quantum mechanical calculations using the GAMESS computer code.

The basis of these investigations is quantum mechanics; however, in order to achieve quantitative results comparable with the experimental ones we need supercomputer power or a cluster of parallel computers. In this way, the tasks have been solved during the last decade of the 20th century.

The fifth section is analytical as well. The object of analysis here is the known methods for evaluating the quality of calculation results in a new parallel environment and experimental characterization of explosive molecules. Unfortunately, the experiment cannot determine to which piece of molecule this or that line of spectrum belonged. Therefore, the current theoretical investigations have become as the standards. The molecular modeling by means of the SCore type PC cluster made it possible to obtain molecular characteristics such as: heats of formation, bond and reaction energies, molecular energies and structure, energies and structures of transition states, charge distribution in molecules, vibrational frequencies (IR and Raman spectra), and electronic transitions. The analysis and evaluation of these results by experiments lead for the conclusion on that accurate *ab initio* quantum chemical computations (molecular electronic structure and vibrational spectra investigations) are essential in the development of spectroscopic methods for detection of small amounts of materials. Also the clusters based on PC's running Linux can become the cheapest "supercomputers" in the academic and commercial field.

The dissertation ends with the main conclusions of this research work. The list of references is provided; the summary in Lithuanian and Appendix are added as well.

2. MATHEMATICAL MODELING OF EXPLOSIVE MATERIALS

The precise knowledge of vibrational spectra would allow detecting small amounts of explosive materials by means of spectroscopic methods. The purpose of our work was to investigate how strong calculated vibrational frequencies, infrared (IR) and Raman radiation intensities depend from the *ab initio* method used. We did not find in literature or in available databases the results of analogous theoretical investigations for given molecules.

Clusters based on PCs running Linux have become the cheapest "supercomputers" in the academic and commercial field. Created computational facilities can be applied for mathematical modeling of any problem. In our way SCore PC cluster was applied to investigate electronic structure and vibration spectra of explosive materials by means of no empirical quantum mechanical computation methods.

The accurate no empirical quantum chemical computations (molecular electronic structure and vibration spectra investigations) are essential in spectroscopy. The basis of these investigations is quantum mechanics, but in order to achieve quantitative results comparable with experimental we need supercomputer's power or parallel computers cluster. This is the way how such tasks are solved nowadays.

2.1. Methods of Investigations

All molecular modeling techniques can be classified under three general categories:

- *Ab initio* electronic structure calculations,
- Semi empirical methods,
- Molecular mechanics.

Ab initio or 'first principles' electronic structure methods are based upon quantum mechanics and therefore provide the most accurate and consistent predictions for chemical systems. This name is given to computations which are derived directly from theoretical principles, with no inclusion of experimental data. The approximations made are usually mathematical approximations, such as using a simpler functional form for a function or getting an approximate solution to a differential equation [66].

Semi empirical calculation methods are set up with the same general structure as a Hartree-Fock (HF) calculation. Within this framework, certain pieces of information, such as two electron integrals, are approximated or completely omitted. In order to correct for the errors introduced by omitting part of the calculation, the method is parameterized, by curve fitting in a few parameters or numbers, in order to give the best possible agreement with experimental data. The good side of this calculation method is that they are much faster than the ab initio calculations. The bad side of semi empirical calculations is that the results can be erratic. If the molecule being computed is similar to molecules in the data base used to parameterize the method, then the results may be very good. If the molecule being computed is significantly different from anything in the parameterization set, the answers may be very poor. Semi empirical calculations have been very successful in the description of organic chemistry, where there are only a few elements used extensively and the molecules are of moderate size. However, semi-empirical methods have been devised specifically for the description of inorganic chemistry [66].

If a molecule is too big to effectively use a semi empirical treatment, it is still possible to model its behavior by avoiding quantum mechanics totally. The methods referred to as molecular mechanics set up a simple algebraic expression for the total energy of a compound, with no necessity to compute a wave function or total electron density. The energy expression consists of simple classical equations, such as the harmonic oscillator equation in order to describe the energy associated with bond stretching, bending, rotation, and intermolecular forces, such as van der Waals interactions and hydrogen bonding. All of the constants in these equations must be obtained from experimental data or an ab initio calculation.

In a molecular mechanics method, the data base of compounds used to parameterize the method (a set of parameters and functions is called a force field) is crucial to its success. Semi empirical method may be parameterized against a set of organic molecules; whereas a molecular mechanics method may be parameterized against a specific class of molecules, such as proteins. Such a force field would only be expected to have any relevance to describing other proteins.

The good side of molecular mechanics is that it allows the modeling of enormous molecules, such as proteins and segments of DNA, making it the primary tool of computational biochemists. The bad side of molecular mechanics is that there are many chemical properties that are not even defined within the method, such as electronic excited states. In order to work with extremely large and complicated systems, often molecular mechanics software packages have the most powerful and easiest to use graphical interfaces. Because of this, mechanics is sometimes used because it is easy, but not necessarily a good way to describe a system [66].

2.2. Quantum Chemical Computations

Theoretical investigations were performed by means of *ab initial* quantum mechanical calculations using GAMESS computer code [2]. The basis of all such calculations is solving of multidimensional time independent Schrödinger equation.

$$\widehat{H}\,\Psi_i = E_i\,\Psi_i\,, \tag{2.1}$$

here $\widehat{H}$ - Hamilton operator of investigated quantum particle system; Ψ_i - wave function of the system; E_i - allowable energy value. Hamiltonian operator and wave function of the system depend on the coordinates of all particles (electrons and nucleus).

Equation (2.1) is second order differential equation depending from $3(N+M)$ variables.

Hamiltonian operator of the molecule with N electrons and M nuclease in no relativistic quantum mechanics has the form:

$$\widehat{H} = \widehat{T}_N + \widehat{T}_e + \hat{V}_N + \hat{V}_e + \hat{V}_{Ne} . \qquad (2.2)$$

Here $\widehat{T}_N$ is nuclei kinetic energy operator (M_ν - the ν -th nuclear mass of the)

$$\widehat{T}_N = -\sum_{\nu=1}^{M} \frac{1}{2M_\nu} \nabla_\nu^2 ; \qquad (2.3)$$

$\widehat{T}_e$ - operator of kinetic energy of electrons

$$\widehat{T}_e = -\frac{1}{2} \sum_{i=1}^{N} \nabla_i^2 ; \qquad (2.4)$$

$\hat{V}_N$ - operator of nuclear repulsion potential energy

$$\hat{V}_N = \sum_{\nu>\mu}^{M} \frac{Z_\nu Z_\mu}{R_{\nu\mu}} ; \qquad (2.5)$$

$\hat{V}_e$ - operator of potential energy of interaction between electrons

$$\hat{V}_e = \sum_{i>j}^{N} \frac{1}{r_{ij}} ; \qquad (2.6)$$

$\hat{V}_{Ne}$ - operator of electrostatic interaction between nuclei and electrons

$$\hat{V}_{Ne} = -\sum_{i=1}^{N} \sum_{\nu=1}^{M} \frac{Z_\nu}{r_{1\nu}} . \qquad (2.7)$$

Here we use atomic units system (the mass of electron, radius of Bohr orbit, Plank constantan, the charge of electron).

In one particle approximation the molecular wave function for closed shell electronic system (with odd number of electrons) is presented by Slater determinant

$$\Psi = \frac{1}{\sqrt{N!}} \det \left| \psi_1 \overline{\psi}_1 \psi_2 \overline{\psi}_2 \ldots \psi_{N/2} \overline{\psi}_{N/2} \right| , \qquad (2.8)$$

where N is the total number of electrons;

$$\psi_i = \phi_i \alpha \; ; \quad \overline{\psi}_i = \phi_i \beta \qquad (2.9)$$

ψ_i and $\overline{\psi}_i$ are one electron wave functions depending from i -th electron coordinates: φ_i - spatial part of electron wave function; α and β - spin orbitals.

When using non relativistic Hamilton operator for electronic part of the problem

$$\hat{H}_e = \sum_{p=1}^{N} h_p + \sum_{p>q}^{N} g_{pq} \,. \tag{2.10}$$

One of the first the Born-Openheimer approximation is used for simplifying the (2.1) equation. In this approximation molecular nuclei and electrons movement are treated separately. It is assumed that light electrons move in stationary electrostatic field of nuclei. The nuclei move in averaged field of all electrons.

Kinetic energy of nuclei $\hat{T}_N$ in equation (2.1) is much smaller than other members of equation. So we regard that $M_\nu \to \infty$. Then

$$\hat{H} = \hat{T}_N + \hat{H}_e + V_N \quad \to \quad \hat{H}_e + V_N \,, \tag{2.11}$$

here $\hat{H}_e$ - molecular electronic Hamiltonian operator:

$$\hat{H}_e = \hat{T}_e + V_e + V_{Ne} \,. \tag{2.12}$$

Commutator

$$\left[\hat{H}_e, \vec{R}\right] = 0 \,. \tag{2.13}$$

That means that electronic Hamiltonian energy values and wave functions can be achieved for any nuclear coordinates $\vec{R}$.

Electronic energies and wave functions will depend from nuclear coordinates as parameters

$$(\hat{H}_e + V_N)\Phi_m(\vec{R}) = E_m(\vec{R})\Phi_m(\vec{R}) \,. \tag{2.14}$$

(2.14) is molecular electronic Schrödinger equation.

2.2.1. Independent Electrons Models

It is impossible to solve (2.15) equation for systems with hundreds and thousands electrons. Another simplification is used in formation of wave function – one electron model. It is assumed that wave function of the system can be presented as multiplication of one-electron functions, which depend from one electron, coordinates only:

$$\Phi_0 = N \chi_{j_1}(\vec{r}_1) \chi_{j_2}(\vec{r}_2) \ldots \chi_{j_N}(\vec{r}_N) \,, \tag{2.15}$$

here N - normalization factor; $\vec{r}_j$ - j - the electron spatial coordinates; $\chi_j(\vec{r}_j)$ - one electron function.

One-electron functions (atomic orbitals, molecular orbitals) are chosen on the background of physical or chemical considerations

Molecular one electron Hamilton operator may be written:

$$\hat{H}_e = \sum_{p=1}^{N} h_p + \sum_{p>q}^{N} g_{pq} \,, \tag{2.16}$$

Here h_p - one electron part:

$$h_p = \frac{1}{2}\nabla_p^2 - \sum_{\mu=1}^{M} \frac{Z_\mu}{r_{p\mu}} \,, \tag{2.17}$$

g_{pq} - two electron part:

$$g_{pq} = \frac{1}{r_{pq}} \,. \tag{2.18}$$

The mean of physical independent electron principle background is that the efficient Hamiltonian of N electron system form can be written

$$\hat{H}_{eff} = \sum_{p=1} (h_p + u_p) \tag{2.19}$$

where u_p - potentials of one electrons. The sum of N one electron Hamiltonian is the form of N electron Hamiltonian. The u_p describes inter electron interaction and nuclear charge screening.

The full Hamiltonian can be written

$$\hat{H} = \hat{H}_{eff} + \left\{ \sum_{p>q} g_{pq} - \sum_{p} u_p \right\} \tag{2.20}$$

The difference of sum written in the brackets must be small if the orbital model is the good approximation.

2.2.2. Hartree-Fock Method

The closed electron system energy (system with even electron number) when Hartree-Fock wave function is used equals (after integration over all spatial coordinates and summation over all spin coordinates):

$$E = 2\sum_{i} \langle \Phi_i | h | \Phi_i \rangle + \sum_{i>j} \left\{ 2\langle \Phi_i \Phi_j | g | \Phi_i \Phi_j \rangle - \langle \Phi_i \Phi_j | g | \Phi_j \Phi_i \rangle \right\} \tag{2.21}$$

In self consistent field spatial parts of molecular orbitals are optimized in order to achieve „the best" one determinant wave function with respect of variation principle – we search the minimum of full energy for the ground electronic state

$$E_0 \le E = \langle \varphi^* | \hat{H} | \varphi \rangle \tag{2.22}$$

or excited electronic state

$$E_1 \leq E = \left\langle \varphi^* \middle| \hat{H} \middle| \varphi \right\rangle \tag{2.23}$$

with additional requirement of orthogonality of functions φ and φ_0, and orthonormalization requirement $\left\langle \phi_i \middle| \phi_j \right\rangle = \delta_{ij}$.

Usually orbitals are expressed as linear combination of known basic functions

$$\phi_i = \sum_k \chi_k T_{ki} \tag{2.24}$$

or in matrix form

$$\mathbf{\Phi} = \chi \mathbf{T} \tag{2.25}$$

here $\mathbf{T}$ - $(m \times n)$ matrix; m is the number of basis functions.

The matrix elements of one electron operator (2.17) are equal:

$$\left\langle \Phi^* \middle| \sum_p h_p \middle| \Phi \right\rangle = \sum_m E_m , \tag{2.26}$$

here

$$E_m = \left\langle \varphi_m^*(\mu) \middle| h_\mu \middle| \varphi_m(\mu) \right\rangle . \tag{2.27}$$

The interaction between the electrons is described by the thermo

$$\left\langle \Phi^* \middle| \sum_{p<q} \frac{1}{r_{pq}} \middle| \Phi \right\rangle = \sum_{m<n} J_{mn} - \sum_{m<n}^{\uparrow\uparrow} K_{mn} . \tag{2.28}$$

Matrix element

$$J_{mn} = \int \varphi_m^*(\mu) \varphi_n^*(\nu) \frac{1}{r_{\mu\nu}} \varphi_m(\mu) \varphi_n(\nu) d\bar{r}_\mu d\bar{r}_\nu . \tag{2.29}$$

is called Coulomb integral.

Matrix element

$$K_{mn} = \int \varphi_m^*(\mu) \varphi_n^*(\nu) \frac{1}{r_{\mu\nu}} \varphi_n(\mu) \varphi_m(\nu) d\bar{r}_\mu d\bar{r}_\nu . \tag{2.30}$$

is exchange integral. It may be written in short

$$J_{mn} \equiv \langle mn | mn \rangle ; \quad K_{mn} \equiv \langle mn | nm \rangle . \tag{2.31}$$

The full energy of the molecule equals:

$$E=\sum_{m}E_m+\sum_{m<n}J_{mn}-\sum_{m<n}^{\uparrow\uparrow}K_{mn}=\sum_{m}E_m+\sum_{m<n}\langle mn|mn\rangle-\sum_{m<n}^{\uparrow\uparrow}\langle mn|nm\rangle, \quad (2.32)$$

here $\uparrow\uparrow$ means summation according the spin orbitals with parallel spins only.

One electron operators $\hat{J}_n$ and $\hat{K}_n$ are introduced:

$$\hat{J}_n\varphi_m(\mu)=\int\frac{|\varphi_n(\nu)|^2}{r_{\mu\nu}}d\vec{r}_\nu\varphi_m(\mu); \quad (2.33)$$

$$\hat{K}_n\varphi_m(\mu)=\int\frac{\varphi_n^*(\nu)\varphi_m(\nu)}{r_{\mu\nu}}d\vec{r}_\nu\varphi_m(\mu). \quad (2.34)$$

Then:

$$\langle mn|mn\rangle=\langle m|\hat{J}_n|m\rangle=\langle n|\hat{J}_m|n\rangle; \quad (2.35)$$

$$\langle mn|nm\rangle=\langle m|\hat{K}_n|m\rangle=\langle n|\hat{K}_m|n\rangle. \quad (2.36)$$

On electron orbitals should be orthonormalized:

$$\langle m|n\rangle\equiv\int\varphi_m^*\varphi_n d\vec{r}=\delta_{mn}. \quad (2.37)$$

The energy (2.32) variation takes the form:

$$\delta\left[E-\sum_{m,n}\varepsilon_{mn}\langle m|n\rangle\right]=0, \quad (2.38)$$

here ε_{mn} - Lagrange multipliers.

Hatree-Fock equations for determination of one-electron functions look

$$\left[\hat{h}+\sum_{n}(2\hat{J}_n-\hat{K}_n)\right]\varphi_m=\sum_{n}\varepsilon_{mn}\varphi_n. \quad (2.39)$$

After introduction of Hatree-Fock opertor:

$$\hat{F}=\hat{h}+\sum_{n}(2\hat{J}_n-\hat{K}_n), \quad (2.40)$$

$$\hat{F}\varphi_m=\varepsilon_m\varphi_m, \quad m=1,2,\ldots. \quad (2.41)$$

In matrix form

$$\widehat{F}\varphi = \varphi\varepsilon \,, \tag{2.42}$$

Hartree-Fock equations are solved by iterational self-consistent field method.

2.2.3. SCF Vibrational Equations

Hamiltonian of many atomic molecule in normal coordinates Q for molecular rotational quantum number J=0 have the form

$$\begin{aligned}\hat{H} &= \frac{1}{2}\sum_{\alpha\beta}\mu_{\alpha\beta}\pi_{\alpha}\pi_{\beta} + \frac{1}{2}\sum_{k=1}^{N}P_k^2 - \frac{\hbar^2}{8}\sum_{\alpha}\mu_{\alpha\alpha} + V(Q)\\ &= T_{cor} + T_{vib} + T_{rot} + V(Q)\end{aligned} \tag{2.43}$$

here N – number of normal modes , α and β – components of Cartesian x,y, z coordinates, $\mu_{\alpha\beta}$ - components of inverse effective inertia tensor, π_{α} - Cartessian component of vibrational angular momentum. It have the form

$$\pi_{\alpha} = \sum_{i>j}\xi_{ij}^{(\alpha)}(Q_iP_j - Q_jP_i) \tag{2.44}$$

Vibrational part of hamiltonian have the form

$$\hat{H}_{vib} = \sum_{k=1}^{N}P_k^2 / 2\mu + V(Q_1,...,Q_N)\,, \tag{2.45}$$

Here Q_i - normal coordinate and P_i – corresponding angular momentum, μ_i – reduced mass of many particle system, N –total number of vibrational modes.

$$V(Q_1,...,Q_N) = \frac{1}{2}\sum_{k=1}^{N}k_iQ_i^2 + W_{\alpha nh}(Q_1,...,Q_N)\,, \tag{2.46}$$

Here k_i – force constant of the i-th normal mode, $W_{\alpha nh}$ – unharmonic part of potential describing coupling of normal modes.

Usually $W_{\alpha nh}$ is presented as higher order (3-rd, 4-th, . . .) terms in Tailor expansion.When we consider vibrations of small amplitudes higher order than quadratic terms in potential energy expansion (2.46) may be neglected. Then vibrational hamiltonian of molecule is the sum of hamiltonians of harmonic oscilators

$w_i=(k_i/\mu_i)^{1/2}$. $W_{\alpha nh}$ member can't be neglected when high amplitude vibrations or excited vibrational states are investigated. This can be accomplished by means of self-consistent field method. Initial molecule's wave function in harmonic approximation ($W_{\alpha nh}=0$) may be written in the form

$$\Psi(Q_1,...,Q_N)=\prod_{I=1}^{N}\Psi_I(Q_i), \tag{2.47}$$

here $\Psi_i(Q_i)$ describes i-th state of Q_i-th normal mode.

Application of variation principal

$$\delta\left\{\langle\Psi\,|\,\hat{H}\,|\,\Psi\rangle/\langle\Psi\,|\,\Psi\rangle\right\}=0 \tag{2.48}$$

leads to the system of equations

$$(h_{ef}(i)-\varepsilon_i)\Psi_i(Q_i)=0\,,\quad i=1,...,N\,, \tag{2.49}$$

which is usually solved by iteration method.

Effective Hamiltonian has the form

$$\begin{aligned} h_{ef}(i)&=\mathrm{P}_i^2/2\mu+V_i(Q_i)+\left\langle\prod_{j\neq i}^{N}\Psi_j(Q_j)\,|\,W_{\alpha\pi h}(Q_1,...,Q_N)\,|\prod_{j\neq i}^{N}\Psi_j(Q_j)\right\rangle \\ &=\mathrm{P}_i^2/2\mu+V_i(Q_i)+V_{SC}(Q_i). \end{aligned} \tag{2.50}$$

The i-th mode self consistent (SC) potential $V_{SC}(Q_i)$ is derived from molecule's full potential after averaging according to all normal modes. In order to find wave function $\Psi_i(Q_i)$ it is necessary to solve the system of N equations by iteration method full vibrational energy of molecule equal:

$$E=\sum\varepsilon_i^{SC}-(N-1)\left\langle\prod_{j\neq i}^{N}\Psi_j^{SC}(Q_j)\,|\,V_{SC}(Q_1,...,Q_N)\,|\prod_{j\neq i}^{N}\Psi_j^{SC}(Q_j)\right\rangle, \tag{2.51}$$

here ε_i^{SC} and Ψ_j^{SC} - solutions of self consistent equations.

We have to solve equations by self consistent field methods in two steps:

1) to solve one dimensional differential equations of 2-nd order (2.49),

2) computation of (N-1)-dimension integrals in order to find effective potential ($V_{SC}(Q_i)$). So, we have changed one dimensional differential equation to N one dimensional equations. This enables us to use parallel computations.

Necessary computer processor's time have linear dependence form N. When solving the same equation by means of variational method and expanding wave function in the basis of harmonic oscillators CPU time grows faster. Self consistent field solving parallel algorithm of one dimensional Schrödinger equation is describe in [2].

Another algorithm for firs principles calculation of vibration spectroscopy of polyatomic molecules was proposed [72], which combines electronic ab initio codes with the vibrational self-consistent field (VSCF) method, and with a perturbation-theoretic extension of VSCF. The integrated method directly uses points on the potential energy surface, computed from the electronic ab initio code, in the VSCF part. No fitting of an analytic potential function is involved. A key element in the approach is the approximation that only interactions between pairs of normal modes are important, while interactions of triples or more can be neglected. Unfortunately at this time only 10-15 atom molecules can be treated by this method because of very computer time consuming potential energy surface computations.

2.2.4. Multi-Configurational Self-Consistent Field Method (MCSCF)

The MCSCF method is a nonlinear approximate model for the computation of eigenfunctions and eigenvalues of the N-body Schrödinger Hamiltonian. For a molecule containing M point wise nuclei of positive charges z_1, ⋯ , z_M and located at $\bar{x}_1,\ldots,\bar{x}_M \in R^3$ (we work within the usual Born-Oppenheimer approximation), and N no relativistic quantum electrons, the N-body Hamiltonian written in atomic units reads

$$H_N = \sum_{i=1}^{N}\left(-\frac{1}{2}\Delta_{xi} + V(x_i)\right) + \sum_{1\le i<j\le N}\frac{1}{|x_i - x_j|}. \tag{2.52}$$

It acts on normalized electronic wave functions Ψ (x_1, ..., x_N) $\in L^2_\alpha\left(\left(R^3\right)^N\right)$, $\int_{R^{3N}}\Psi^2 = 1$. The subscript a indicates that, due to the fermionic nature of the electrons, one solely considers wave functions Ψ which are ant symmetric under permutations of
variables:

$$\Psi = (x_1,\ldots,x_i,\ldots,x_j,\ldots,x_N) = -\Psi(x_1,\ldots,x_j,\ldots,x_i,\ldots,x_N). \tag{2.53}$$

Finally, V is the electrostatic potential generated by the nuclei

$$V(x) = -\sum_{m=1}^{M} \frac{z_m}{\left|x - \bar{x}_m\right|}. \tag{2.54}$$

In what follows, we denote by $Z = \sum_{m=1}^{M} z_m$ the total nuclear charge. For the sake of simplicity, we do not take the spin into account in the first part of this paper, but the following arguments can be straightforwardly adapted to the case of spin-dependent wave functions.

When Z > N – 1, it has been proved by Zhislin and Zhislin-Sigalov, that the spectrum $\sigma(H_N)$ of H_N has the form:

$$\sigma(H_N) = \{\lambda_0 \le \lambda_1 \le \cdots \le \lambda_k \le \cdots\} \cup \left[\sum, \infty\right), \tag{2.55}$$

where the $\{\lambda_k\}k \ge 0$ are eigenvalues of finite multiplicity which are below and converge to the bottom of the essential spectrum $\sum$ as $k \to \infty$. A ground state is a normalized eigenfunction associated with the first eigenvalue λ_0, whereas excited states are obtained for $\lambda_k > \lambda_0$. A ground or excited state Ψ_k is a solution of the time-independent Schrödinger equation H$_N$ $\Psi_k = \lambda_k \Psi_k$. The ($\lambda_k$)'s can be obtained by the usual Rayleigh-Ritz formula

$$\lambda_k = \inf_{\substack{V \subset D(H_N) \\ \dim(V) = k+1}} \sup_{\substack{\Psi \in V, \\ \int \Psi^2 = 1}} \left\langle \Psi \mid N \mid \Psi \right\rangle, \tag{2.56}$$

where D (H$_N$) is the domain of H$_N$ and <· ·> denotes the usual scalar product of $L^2(R^{3N})$.

The MCSCF method is based on the following remark:

$$L_\alpha^2((R^3)^N) = \overset{N}{\underset{n=1}{\Lambda}} L^2(R^3), \tag{2.57}$$

an equality which can be xplicated in the following way. Consider an orthonormal basis $(\Psi_i)1 \le i$ of $L^2(R^3)$, $\int_{R^3} \Psi_i \Psi_j = \delta_{ij}$. Then, one obtains an orthonormal basis of $L_\alpha^2((R^3)^N)$ by considering the antisymmetrized products $(\Psi_{i_1} \Lambda \cdots \Lambda \Psi_{i_N})_{1 \le i_1 < \cdots < i_N}$ where $\Psi_{i_1} \Lambda \cdots \Lambda \Psi_{i_N}$ denotes the so-called Slater determinant of the Ψ_{i_k} 's:

$$(\Psi_{i_1} \Lambda \cdots \Lambda \Psi_{i_N})(x_1, \ldots, x_N) = \frac{1}{\sqrt{N!}} \det(\Psi_{i_k}(x_l))_{k,l}. \tag{2.58}$$

In other words every antisymmetric wave function Ψ is antinfinite linear combination of such Slater determinants:

$$\Psi = \sum_{1 \le i_1 < \cdots < i_N} c_{i_1} \cdots_{i_N} \Psi_{i_1} \Lambda \cdots \Lambda \Psi_{i_N}, \tag{2.59}$$

the sum being convergent in $L_{\alpha}^{2}((R^{3})^{N})$ .Remark that $\int_{R^{3N}} \Psi^{2} = 1$ is then equivalent to the condition

$$\sum\nolimits_{i_{1<\cdots<iN}} (c_{i_{1\ldots iN}})^{2} = 1 . \qquad (2.60)$$

In the MCSCF approximation, one restricts the number of Slater determinants appearing in (2.59), by restricting the number of occupied orbitals [68]. Moreover, to obtain a suitable approximation, the orbitals will not be fixed but rather optimized.

Let K ≥N be the number of considered orbitals, denoted as φ_1, ..., φ_K $\in L^{2}(R^{3})^{K}$ and which still must satisfy the orthonormal condition $\int \varphi_i \varphi_j = \delta_{ij}$. Both the coefficients $(c_{i_1},\ldots,_{i_N})_{1\le i_k \le K}$ and the K orbitals $(\varphi_k)_{k=1}^{K}$ are considered as being variational parameters, and the form of the N-body wavefunction Ψ is now

$$\Psi = \sum_{1\le i_1<\cdots<i_N\le K} c_{i_1}\ \cdots_{i_N}\ \varphi_{i_1}\Lambda\cdots\Lambda\varphi_{i_N} . \qquad (2.61)$$

When there is no ambiguity, we shall use the following convenient notation

$$\Psi = \sum_{I=\{i_1<\cdots<i_N\}} c_I \Phi_I \qquad (2.62)$$

where by definition $\Phi_I = \varphi_{i_1}\Lambda\cdots\Lambda\varphi_{i_N}$ when I = { i1 < ⋯ < iN } ⊂ { 1, ..., K } .

Of course in practice not all the $\begin{pmatrix} K \\ N \end{pmatrix}$ possible Slater determinants in (2.61) are considered during an MCSCF computation, and only some of them are selected. For the sake of clarity, we shall only consider in this preceding the "full" MCSCF method (2.61) in which all orbitals are active.

When the quadratic form $\Psi \mapsto \langle \Psi \mid H_N \mid \Psi \rangle$ is restricted to states of the form (2.61), one obtains an energy depending on $(c_{i_1},\ldots,_{i_N})_{1\le i_k\le K}$ and $(\varphi_i,\ldots,\varphi_K) \in L^{2}(R^{3})^{K}$ which is nonlinear. This is due to the fact that functions of the form (2.75) do not form a vector subspace of $L_{\alpha}^{2}((R^{3})^{N})$. In the following, we denote by $c = (c_{i_1},\ldots,_{i_N})_{1\le i_k\le K}$ the collection of the cI's (lexicographical ordering), and by $\Phi = (\varphi_1,\ldots,\varphi_K) \in (L^{2}(R^{3}))^{K}$ the collection of the orbitals. Due to the normalization constraint on the wave function Ψ, our variable (c,Φ) belongs to the following manifold

$$M_N^K = \left\{ (c,\Phi) \in R^{\begin{pmatrix} K \\ N \end{pmatrix}} \times (L^{2}(R^{3}))^{K}, \sum_I (c_I)^2 = 1, \int_{R^3} \varphi_i \varphi_j = \delta_{ij} \right\}. \qquad (2.63)$$

The MCSCF energy is then defined as $\varepsilon_N^K(c,\Phi) = \langle \Psi \mid H_N \mid \Psi \rangle$ where Ψ is given by (2.61).
However, it is important to realize that the two variables c and Φ play different roles. The energy ε_N^K is still quadratic with respect to c:

$$\varepsilon_N^K(c,\Phi) = \sum_{I,J} c_I c_J \langle \Phi_I \mid H_N \mid \Phi_J \rangle = \sum_{I,J} c_I c_J (H_\Phi)_{IJ}$$

$$\varepsilon_N^K(c,\Phi) = \sum_{I,J} c_I c_J \langle \Phi_I \mid H_N \mid \Phi_J \rangle = \sum_{I,J} c_I c_J (H_\Phi)_{IJ} \qquad (2.64)$$

where $(H_\Phi)_{IJ} = \langle \Phi_I \mid H_N \mid \Phi_J \rangle$ and $\Psi = \sum_I c_I \Phi_I$. The Hamiltonian matrix H_Φ is the $\begin{pmatrix} K \\ N \end{pmatrix} \times \begin{pmatrix} K \\ N \end{pmatrix}$ matrix of the quadratic form associated with H$_N$, when it is restricted to the $\begin{pmatrix} K \\ N \end{pmatrix}$ -dimensional space V$_\Phi$ = Span(Φ$_I$) spanned by the Slater determinants that can be constructed with the orbitals $\Phi = (\varphi_1, \ldots, \varphi_K)$. But, as mentioned above, the energy ε_N^K is not quadratic with respect to the φ_i's and takes the following general form:

$$| \ \varepsilon_N^K(c,\Phi) = \sum_{i,j} \frac{\gamma_{ij}}{2} \int_{R^3} (\nabla\varphi_i \cdot \nabla\varphi_j + V\varphi_i\varphi_j) + \sum_{i,j,k,l} W_{ijkl} \iint_{R^6} \frac{\varphi_i(x)\varphi_j(x)\varphi_k(y)\varphi_l(y)}{|x-y|} dxdy \qquad (2.65)$$

where γ_{ij} and W$_{ijkl}$ only depend on c. The first sum in (2.79) is quadratic whereas the last term is quadratic. Thus, the MCSCF equations (i.e. the equations satisfied by a stationary point of ε_N^K on the manifold M_N^K consist in an eigenvalue equation for c coupled with a system of K nonlinear elliptic partial differential equations for
(φ$_1$, ..., φ$_K$), of the general form

$$\begin{cases} n_i(c)\left(-\dfrac{\Delta}{2} + V\right)\varphi_i + \displaystyle\sum_j W_{i,j}^{(c,\Phi)} \varphi_j = \sum_j \lambda_{ij} \varphi_j \\ H_\Phi c = \beta c. \end{cases} \qquad (2.66)$$

The operators $W_{i,j}^{(c,\Phi)}$ depend on (c,Φ) in a complicated manner.
Mathematically, it is not at all obvious to prove the existence of solutions to the MCSCF equations (2.66) when the problem is posed in infinite dimension. There is a possible lack of compactness at infinity: for any sequence (c$_n$, Φ$_n$) of approximate solutions of (2.66) it is possible that some electrons "escape to infinity" as $n \to \infty$ [69].
This phenomenon will typically occur when N » Z, if the nuclei are not able to bind the N electrons.

Of course, in practice the computation is carried out in finite dimension: the space L^2(R^3) is replaced by a finite dimensional space V , usually spanned by a chosen finite set of atomic orbitals. Then the possible lack of compactness mentioned above never occurs. Notice however that the problems that will be raised below concerning the existence and the computation of excited states (specific solutions of (2.66) which approximate the true eigenfunctions of H_N) appear in finite dimension also. They will not be related to the mathematical problem of lack of compactness at infinity which only occurs in infinite dimension.

The definition of the MCSCF ground state energy is clear: it suffices to minimize the energy ε_N^K on M_N^K , i.e. to minimize the N-body energy over MCSCF functions of the type (2.49):

$$\lambda_0^K = \inf_{(c,\Phi)\in M_N^K} \varepsilon_N^K(c,\Phi). \tag{2.67}$$

It can be proved that $\lambda_0^K > \lambda_0$ and that $\lim_{K\to\infty} \lambda_0^K = \lambda_0$ which justifies the MCSCF approach for ground states. Recall that K is the number of orbitals appearing in the model.

Numerically, a minimizer of (2.81) is usually computed by a Newton-like algorithm, sometimes improved by a trust-region method, see e.g. [67] and the references in [70]. For the Hartree-Fock model K = N, efficient numerical methods based on combinations of fixed-point and optimization strategies are available [68]. Unfortunately, such algorithms are specifically designed for solving the Hartree-Fock problem and seem to be difficult to adapt to the more general MCSCF setting.

2.3. Wave Functions

In quantum chemical calculations, wave functions of systems are present as linear combinations of elementary functions, which are called basis orbitals [91]. Mathematical aspect that is adapted for description of atoms and molecules is given below.

Any function may be composed as a linear combination of elementary functions:

$$f(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + c_3\varphi_3(x) + \ldots \tag{2.68}$$

where x is a variable.

Using the composition of Eq. (2.68), the completeness of the basis set is to be described.

Let us assume that the variable x is referred to the interval [a,b]. The functions f(x) investigated here as well as functions φ1(x), φ2(x)... are defined in this interval. Then the function f(x) may be approximated as follows:

$$f(x) \approx f_n(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + \ldots c_n\varphi_n \tag{2.69}$$

where a finite number of basis orbitals is included.

The composition coefficients are chosen so that the integral $D = \int_a^b | f(x) - f_n(x) |^2 \, dx$ is of the minimal value so that the composition is the most accurate. The approximation is the most accurate when ortonormalized functions are applied. In this case, the composition coefficients are equal to [20]:

$$c_i = < \varphi_i(x) | f(x) > \quad (2.70)$$

When D→0 for every function f(x) defined in the interval [a, b], the set of functions $\varphi_i(x)$ is complete. Then the function f(x) takes the form:

$$f(x) = \sum_{i=1}^{\infty} c_i \varphi_i(x) \quad (2.71)$$

In the case when the function is two variables, and the set $\{\varphi_i(x_i)\}$* is complete for x_1 in the interval [a,b], as well as $\{\varphi_j^{'}(x_2)\}$ is complete for the x_2 in the interval [a′, b′], the function of the system is as follows:

$$f(x_1, x_2) = \sum_{i,j} c_{i,j} \varphi_i(x_1) \varphi_j(x_2) \quad (2.72)$$

The above mentioned bilinear combination is valid if the coefficients c_i are the following:

$$c_i = c_{i1}\varphi_1^{'}(x_2) + c_{i2}\varphi_2^{'} + ... \quad (2.73)$$

The similar bilinear combination for the many variable function is constructed in the same way and the above mentioned functions take the form of [20]:

$$\phi(r_1, r_2, r_3, ...r_n) = \sum C_{i,j,k,...p} \varphi_i(r_1) \varphi_j(r_2) ... \varphi_p(r_n)$$

Subscripts I, j,...p designate the orbitals from the complete set of orbitals φ_1, $\varphi_{2.}$, ... φ_p.

The above-described mathematical approaches of wave functions are valid for any quantum system of many identical particles. However, new features were observed.

When investigating many electron systems, additional conditions are be important for the wave functions, i. e., the wave functions of the system should be antisymmetrized with respect to permutations of coordinates. The ant symmetric functions are of the form:

$$\phi(r_1, r_2, ...r_n) = \sum_{i,j,k,...,p} C_{i,j,k,...p} \Phi_{i,j,k,...p}(r_1, r_2, ...r_n)$$

where

$$\Phi_{i,j,k,...p}(r_1, r_2, ...r_n) = M_{i,j,k,...p} \sum_p \varepsilon_p \widehat{P} \varphi_A(r_1) \varphi_B(r_2) ... \varphi_X(r_n) \quad (2.74)$$

and $\hat{P}$ is a permutation operator, ε_p is equal to 1 or –1 that is depended on $\hat{P}$ parity, M_x is a normalization constant.

The sum shown in Eq. (2.73) may be written in the form:

$$\Phi_{i,j,k,...p}(r_1, r_2, ...r_n) = M_{i,j,k,...p} \det | \varphi_A(r_1) \varphi_B(r_2) ... \varphi_X(r_n) \quad (2.75)$$

For convenience, only diagonal elements of the determinant $\Phi_{I,j,k,...p}(r_1, r_2, ... r_n)$ are indicated.

In the quantum chemical calculations, linear combinations of atomic functions are used for approximations of molecular orbitals. The use of these combinations implies that the identity of an atom is not destroyed, but only perturbed slightly. These combinations are applied, because finding one-electron wave functions for molecules is a more complicated problem than that for atoms. The nuclear potential in an atom is of a spherically symmetrical form and, therefore, it is possible to separate radial, angular and spin parts of the wave function. On the other hand, molecules are many – center systems and such a separating of variables is impossible. Therefore, practically it is very difficult to calculate an integral of three variables. To make the above-mentioned solution easily, molecular orbitals are expressed as linear combinations of atomic functions. The above – mentioned approximation is called the linear combination of atomic orbitals (LCAO). Such an approximation is based on two assumptions. First, it is assumed that the potential of an electron localized near to a nucleus in molecule is completely described by the potential of this nucleus, and the potentials of other atoms exert is a relatively small influence, therefore, it may be neglected. Second, the sets of molecular orbitals must continuously transit to the atomic orbital sets. This approximation describes the quantum states of molecules fairly well. However, the accuracy deficiency appears due to incompleteness of the sets of basis functions when applying the above – mentioned method.

The wave functions $\phi(r_1, r_2,... r_n)$ contain much more detailed information than it is actually needed in any practical application. It would be a very great computational simplification if calculating of this extra insignificant information could be avoided. Now a more general way of description of quantum states will be considered.

2.4. Basis Sets

Different basis sets were used during these investigations: from 6-31 G* (with polarization *d* functions on second period elements; all in all 250 basis functions) to 6-311 G** (with two polarization *d* functions on second period elements). Also were used *f* function and diffuse *s* and *p* functions on second period atoms and diffuse functions on hydrogen atoms (all in all 770 basis functions). The basis functions used are necessary in order to achieve reliable results in such investigations During multiconfigurational self consistent field (MCSCF) calculations performed for trinitrotoluene we left unfrozen 10 highest occupied MO and 10 lowest unoccupied MO from Hartree-Fock Slater determinant. All configurations generated from unfrozen MOs with spin projection zero were included in MCSCF procedure. We had more than 65000 configurations. All unfrozen MOs are orbitales of benzene ring and NO_2 groups.

Basis sets are a variety of mathematical functions used to resolve a differential equation. In quantum chemical calculations, this term is applied to Gaussian curves that represent atomic orbitals, which are optimized to reproduce the chemical properties of a system. Basis sets vary in size and in their description of the electrons in different atomic orbitals. As long as the basis sets get larger they include more of basis functions which improve the calculations but involve a high computational cost.
Also, it is possible adding polarize and diffuse functions. There are two general categories of basis sets [92].

The two primary and competing criteria for selecting a basis set are the accuracy and size, i. e., a given set should be suitably to describe the system of interest, and the computations should be performable. In our calculation we have used STO-3G [93], 6-31G, 6-311G, 6-311G** [94] basis sets. These are discussed below.

2.4.1 Minimal Basis Sets

A minimal basis set describes only the most basic aspects of the orbitals. The essential idea of the minimal basis set is that it selects one basis function for every atomic orbital that is required to describe the free atom. All basis sets equations that use Slater Type Orbitals (STO) in the form STO-*k*G are considered to be "minimal" basis sets. STO are more exact, but take much longer to calculate than Gaussian Type Orbitals (GTOs) and you can get a good approximation of the STO adding several GTOs to the calculations. STO-*k*G means that you are modeling a STO calculation, with "*k*" number of GTOs. For example, the STO-3G basis set consists of Slater – type orbitals (STO), each of them approximated by three Gaussian functions. For molecules containing no atoms heavier than fluorine, the atomic functions take the form [93]:

$$\varphi_{1s}(\varepsilon_1, r) = \left(\frac{\varepsilon_1^{3}}{\pi}^{1/2} e^{-\varepsilon_1 r} \right)$$

$$\varphi_{2s}(\varepsilon_2, r) = \left(\frac{\varepsilon_2^{5}}{3\pi})^{1/2} r e^{-\varepsilon_2 r} \right)$$

$$\varphi_{2p}(\varepsilon_2, r) = \left(\frac{\varepsilon_2^{5}}{\pi}^{1/2} r e^{-\varepsilon_2 r} \cos\varphi \right) \tag{2.76}$$

These combinations are obtained for Slater – type orbitals with ε equals to 1 and then uniformly scaled. Additionally, each of these orbitals is replaced by atomic orbitals that are the sum of Gaussian type orbitals. Thus, the Slater type orbitals are represented:

$$\varphi_{1s}(1, r) = \sum_{k}^{K} d_{1s,k} g_{1s}(\alpha_{1k}, r)$$

$$\varphi_{2s}(1, r) = \sum_{k}^{K} d_{2s,k} g_{1s}(\alpha_{1k}, r)$$

$$\varphi_{2p}(1, r) = \sum_{k}^{K} d_{2p,k} g_{2p}(\alpha_{2k}, r) \tag{2.78}$$

where α, and d are constants, and

$$g_{1s}(\alpha, r) = \left(\frac{2\alpha}{\pi} \right)^{3/4} e^{-\alpha r^2}, \; g_{2p}(\alpha, r) = \left(\frac{128\alpha^5}{\pi^3} \right) r e^{-\alpha r^2} \cos\varphi \tag{2.79}$$

are Gaussian type orbitals. In the STO-3G basis set, K is equal to 3. As it is shown in Eq. (2.78), a 2s exponential function is replaced by a linear combination of 1s Gaussian functions what is the deficiency of the basis set.

2.4.2. Split Valence Basis Sets

In split valence basis sets, additional basis functions (one contracted Gaussian plus some primitive Gaussians) are allocated to each valence atomic orbital. The resultant linear combination allows the atomic orbitals to adjust independently for a given molecular environment. The number of functions assigned to valence orbitals characterizes Split valence basis sets.

Figure 2.0 shows a replace of each minimal basis set orbital by two orbitals. In each molecular orbital, both orbitals of the set appear and they will mix in the ratio that gives the lowest energy. The combination of a large orbital and a small orbital is essentially equivalent to an orbital of intermediate size.

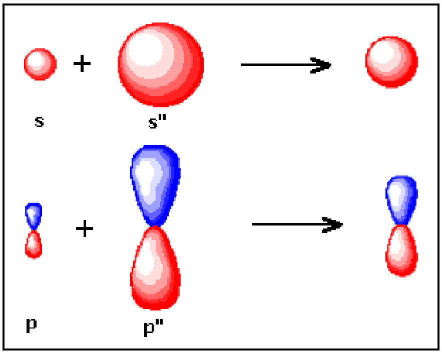


Figure 2.0. Effect of the split valence basis set

The resulting orbital is of a size that best fits the molecular environment since it is obtained from minimizing the energy. For example, consider the 6-31G* for H and C. For H, there is only one valence electron, and it is represented by two orbitals, one constructed of 3 primitives and the other with 1 primitive. With C, the s-orbital is a core orbital, and is represented by 6 Gaussian primitives. The sp-orbital, on the other hand, is a valence orbital, and is represented by two orbitals, one with 3 Gaussians and the other with 1.

It is also quite common to use split valence basis sets where the valence orbitals are split into say three, rather than two, functions. For example, let us consider the 6-311G* basis set. The 6 represents the 6 Gaussian primitives used to calculate the s-shell, the 3 represents the number of GTOs for one of the sp-shells and each 1 represents the number of GTOs for the other two sp-shells.

2.4.3. Extended Basis Sets

Extended basis sets describe the orbitals in great detail. This basis set can have a variety of different polarization functions and diffuse functions added. Non-uniform displacement of charge away from the atomic nuclei can be taken in account using the polarization functions in order to improve the chemical bonding description. For best approximation we have to take in mind that the orbitals share properties of “*s*” and “*p*” or of “*p*” and “*d*” orbitals.

Therefore, they do not have individual orbital characteristics. As long as the atoms get close, their charge distribution cause a polarization effect, whose positive charge shift to one side and the negative charge move to the other side, producing the atomic orbitals distortion.

In Figure 2.1, we can see the effect of the polarization functions, in which the "*s*" orbitals are affected for the "*p*" orbitals and the "p" orbitals are affected for the "*d*" orbitals. The asterisk (*) is the symbol used to add the polarization functions to the basis sets. One * at the end of the basis set denote that "*p*" orbitals were polarized getting properties of "*d*" orbitals. The use of two ** indicate the polarization in "*s*" orbital with properties of "*p*" orbitals, in addition of "*p*" polarized orbitals.

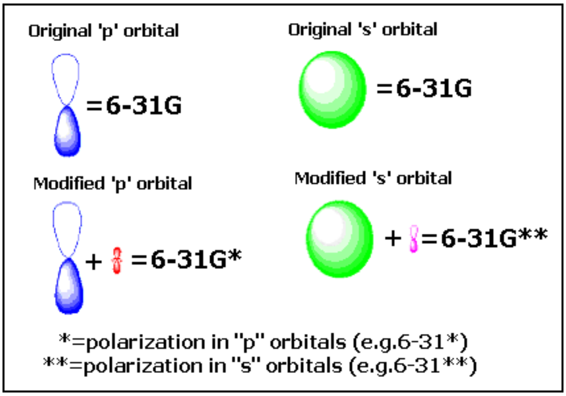


Figure 2.1. Effects of polarization functions added to the basis sets.

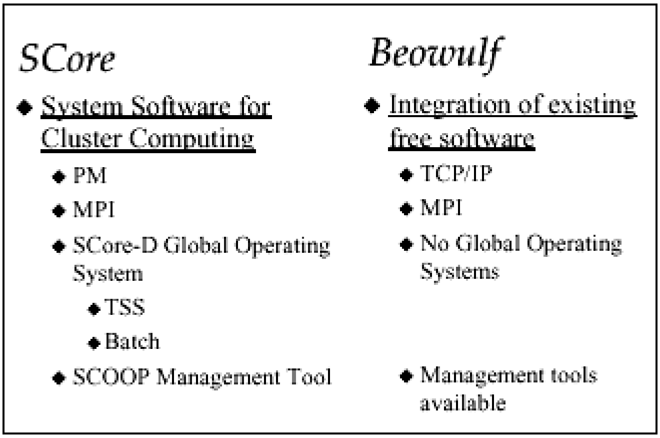
For the other side, diffuse functions are necessary to evaluate the species which significant electronic density was far of the nuclear center, so that the outermost weakly electrons have to be taken into account. The plus symbol (+) is used to add Gaussian diffuse functions of s and p type to the basis sets. One + indicate that at all atoms, except H atoms, diffuse functions are included. Using two plus symbols (++), add these functions to all atoms including the H atoms [64].

2.5. Conclusions of this Chapter

Computational quantum chemistry is a technique that helps us to resolve, predict, and study new concepts, compounds, reactions and mechanisms. This method is very useful with compounds that require exceptional worry in their use, e.g. explosives, reducing risk to personnel testing and maintenance costs in service. Molecular modeling is a ground of computational quantum chemistry, focus on predict the behavior of individual molecules within a chemical system. The molecular modeling let us to obtain molecular characteristics such as: heats of formation, bond and reaction energies, molecular energies and structure, energies and structures of transition states, charge distribution in molecules, vibrational frequencies (IR and Raman spectra), and electronic transitions.

3. THE PERSONAL COMPUTER CLUSTER

Computational quantum mechanical methods are widely used for investigation of molecular electronic structure, dynamical properties of many atomic structures, chemical reactions and mechanisms. Computational quantum chemistry is the ground of molecular modeling, on prediction the behavior of individual molecules within a chemical system. The basis of molecular modeling investigations is quantum mechanics, but in order to achieve quantitative results comparable with experimental we need supercomputer's power or parallel computers cluster. Such way is very useful with compounds that require exceptionally care in their handling, such as explosives, decreasing risk to persons testing and maintenance costs in service. Created computational facilities were applied to investigate vibrational spectra of trinitrotoluene and 2,4,6-trinitrophenol molecules by means of non-empirical quantum mechanical computational methods. The precise knowledge of vibrational spectra would allow detecting small amounts of these materials by means of spectroscopic methods. The purpose of our work was to investigate how strong calculated vibrational frequencies, infrared (IR) and Raman radiation intensities depend from the *ab initio* method used. We did not find in literature or in available databases the all results of analogous theoretical investigations for given molecules.



SCore

- <u>System Software for Cluster Computing</u>
  - PM
  - MPI
  - SCore-D Global Operating System
    - TSS
    - Batch
  - SCOOP Management Tool

Beowulf

- <u>Integration of existing free software</u>
  - TCP/IP
  - MPI
  - No Global Operating Systems
  - Management tools available

Figure 3.0. The differences of two parallel environments used in research work.

As was sad, we need the supercomputer's power or cluster systems of personal computers (PC), which have become widely accepted in high performance computing (HPC). Because, the clusters can be constructed and used in different ways, was chosen some cluster related projects that may be of general interest. This includes a combine of Linux-specific and generic cluster references. In our interest at first time was the Beowulf type cluster, because such clusters are scalable performance clusters based on commodity hardware, on a private system network, with open source software (Linux OS) infrastructure [20]. The designer can improve performance proportionally with added machines. The commodity hardware can be any of a number of mass-market, stand-alone compute nodes as simple as two networked computers each running Linux and sharing a file system or as complex as 1024 nodes with a high-speed, low-latency network.

At first time in applied sciences department in Lithuanian Military Academy (LMA) we created the Beowulf cluster completely using commodity hardware of teaching class and software using standard technology such as Fast Ethernet, and IDE. Beowulf programs are usually written using languages such as C and FORTRAN. They use message passing to achieve parallel computations, but to achieve higher performance on the Beowulf clusters we must used the specialized hardware, that is very expensive. It was the reason why we looking for another cluster system environment, which will let us to achieve higher performance with our possibilities and hardware, we had (Figure 3.0). Detailed Beowulf cluster performance is described in the chapter 4 in this thesis.

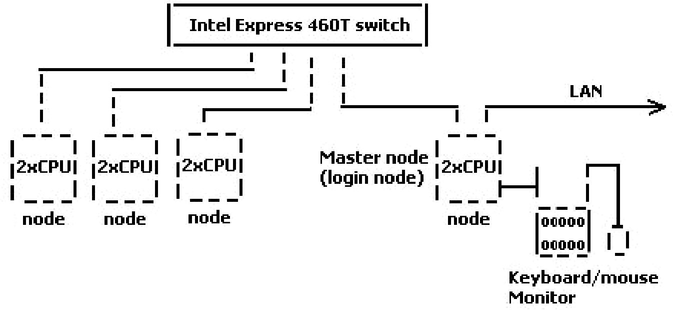


Figure 3.1. The communication map of the PC cluster TAURAS.

So at the end in Lithuanian Military Academy was self made local area PC cluster TAURAS for parallel calculations. Created computational facilities were applied to investigate explosive molecules and during investigations, the General Atomic and Molecular Electronic Structure System (GAMESS) quantum chemistry computer code was used [2].

One of important features of cluster TAURAS was that we used the heterogeneous commodity hardware of teaching class. All PC computers on our cluster connected as is shown in Figure 3.1, by local area network, so PC cluster TAURAS is made like distributed memory Multiple Instruction Multiple Data (MIMD) type parallel computer cluster.

As was explaining above the cluster TAURAS made from commodity hardware and for HPC was used the cluster system software, which let us to obtain the features: high scalability, high-performance communication, a parallel programming environment, job management and checkpoint.

Parallel computers can be efficiently used if the software exploits the hardware architecture. Therefore, the other important feature of this cluster is the software we used. For cluster TAURAS to achieve higher performance computing was used cluster system software, which was developed by the Real World Computing Partnership (RWCP), and now is released by a PC Cluster consortium [3]. SCore cluster system software was built on top of a Linux commodity operating system without kernel modifications but with the addition of a driver to the kernel, that let us to up the cluster's TAURAS performance.

3.1. Structure of Made PC Cluster

The structure of cluster essentially consists of three steps:

- Determination of the requirements of the overall system,
- Definition and purchase of the hardware,
- Installation of the software.

From it results also the arrangement of the following chapter, which contains reason of realizations and design decisions, in addition, aims at a rough outlining of the installation.

We decided to run license-free Linux Red Hat 7.3 OS on each PC, because, its source code is available and there are no license fees. The open source code is very important, as the operating may have bugs that need to be fixed or may need modifications to tune performance for parallel computation or a parallel operating system. With the open source code, we can overcome these problems.

We decided to use Fast Ethernet network, which is a commodity hardware product. Network performance plays a very important role in parallel computing and network characteristics affect how we program for higher performance.

The PC world is changing so fast, taking one year to develop a PC cluster would put us behind, and the cost-effectiveness of the PC cluster would be lost. Therefore, we decided to use conventional hardware.

Scalability is also important. A rack-mounted PC cluster should be easily expanded merely by connecting additional PC cluster.

For us was important that the PC cluster should contain a monitoring facility to diagnose PCs and should boot-up by itself, so that the PC cluster will behave as a complete parallel machine. With this configuration, we can compare the cost of the PC cluster with other parallel machines.

In addition, it is necessary that, the hardware should be easy to maintain. Since our PC cluster will also be used as a test bench, we expected that some hardware components would need to be replaced. We should configure the cluster so that replacement is easier than that of pile of PCs.

3.1.1. Requirements of the System

The hardware of our cluster consists of three substantial components, which are dimensioned accordingly, the requirements:

Cluster node: the cluster nodes are the actual workers, on whom a parallel setting of tasks is distributed. These PC's have one high arithmetic performance furnish, in order to decide the task posed in shorter time. Thus, not only a fast processor is responsible for the rapid decision of the problem posed, but also the entire configuration around the processor:

- High-quality and fast memory (RAM), in order not to in-brake the processor
- a fast Motherboard with a large bus width and

- Suitable plug-in cards, e.g. network maps, are the components system, which are coordinated, in order to prevent negative consequences.

Master node: Master one calls the computer, from which the processes are distributed on the nodes. There the data are central stored and distributed if necessary on all nodes. It is the gate of the cluster system to the external world, with which the users connect themselves and their parallel programs can start.

Network: Most network hardware is interfaced via Linux OS kernel driver, typically supporting TCP/UDP communication. Used hardware and software are important for all sources of latency. In most cases, the network latency is just a few microseconds, the much larger numbers reflect layers of inefficient hardware and software interfaces. For parallel calculations is important both the bandwidth and latency. The connection of the individual nodes among themselves and the connection of the nodes to the master play a central role in the structure of the cluster. If communication of the nodes is too slow within the computer federation, the arithmetic performance of fast nodes cannot be higher. They must wait on other nodes constantly. The fast computed results more belong to network.

For administering the cluster, we have “login node” with a keyboard, monitor and mouse. Others nodes can be “headless” (no keyboard, mouse, or monitor) but in this way there is all machines like “login node”, because cluster TAURAS is made on the basis of teaching class.

Figure 3.1 shows the communication map of PC cluster TAURAS and Table 3.0 shows the specification of the cluster. One of important features of cluster TAURAS is that it is made like heterogeneous cluster.

Table 3.0. Specifications of personal computer cluster TAURAS [1].

Integrator	Self-made
Number of nodes	16
Processors total	32
Total peak performance (GFlops)	30,18
Total memory (GB)	12
Total disk (GB)	320
Interconnect Technology	Fast Ethernet
Operating System	Linux
OS Extension used	SCore
Communication software	SCore
Main application area	Ballistics, Materials Science, Quantum Mechanics of Molecules, Chemistry

We used the commodity hardware of teaching class. It was personal computers: 2 of them dual AMD 1.2 GHz processors, 8 - dual 733 MHz Pentium III processors, 5 of them have dual 800 MHz Pentium III processors and the rest 1 node have dual 450 MHz Pentium II Celeron processor; two of them have 1024 MB RAM others 768 MB RAM, all have 20 GB disk drives. Machines have been installed with the standard Red Hat 7.3- Linux OS.

All PCs are assumed to boot from their own hard drive and have a fast Ethernet network (100 Mb/s) connection to a switch controlling the private cluster network. The suggested range of addresses for a private network is from 192.168.0.0 to 192.168.254.0. Nobody is able to connect directly to a compute node from outside this network anyway. This keeps normal traffic from interfering with inter-node communication and vice versa.

Total peak performance of the cluster is 30.96 Gflops; total memory-12 GB; total disk space - 320 GB. Our SCore type PC cluster's parameters was comparable with other ones located in academic areas and by name TAURAS was registered in the top500 clusters list [1].

The cluster differs from the network of workstations in security, application software, administration, and booting and files systems. Application software uses underlying message passing systems like Message Passing Interface (MPI) [4] and Parallel Virtual Machine (PVM). There are many ways to express parallelism, but message passing is the more effective and more modern, so in cluster TAURAS we used the MPI.

MPI is a message passing library specification, which is designed to run faster with a large multiprocessor. MPI has many point-to-point and collective communication options. Moreover, it has the ability to specify a logical communication topology. There are a lot of MPI implementations, such as MPICH [5], and MPI-LAM and so on [6]. These environments can be used not only on MPP's but also on PCs or workstations connected by a LAN.

3.2. The RWC Project

Since 1992, the Real World Computing Partnership (RWCP) has been actively promoting the Real World Computing (RWC) Project, a 10-year plan, under the guidance of the Ministry of International Trade and Industry (MITI), to create an innovative system of information processing technologies capable of handling various types of information in the real world.

One of the main research fields of the RWC project is Distributed Computing Technology. In this field, the Parallel and Distributed System Software Laboratory has been researching a cluster system software called the SCore Cluster System Software" to realize a seamless parallel and distributed system"(seamless system) since 1995. The seamless system will become a next-generation parallel and distributed system for the LAN of the future with a transmission throughput ranging from several Gbps to 10 Gbps. A seamless system is a distributed system that enables users to obtain the optimal computing power to meet users' needs from computers on a network in a distributed environment using a high-speed LAN. The seamless system must also support different CPU architectures and different types of networks.

3.3. SCore Cluster System Software Architecture

SCore is a cluster distribution whose development was started in 1992 by Real World Computing Partnership (RCWP). In 2002, RCWP was closed, and SCore was transferred to PC Cluster Consortium. System development stopped at that point, and no information on further development can be found. That is the main disadvantage of the SCore cluster distribution. This thesis has done as a part of the SCore cluster system software implementation on commodity hardware. The high performance-computing environment on PC cluster TAURAS cannot be realized without the SCore cluster system software on top of Linux OS [20, 55].

Cluster system consists of a number of nodes which are connected by a network, so when a parallel program is executed, some software is needed to provide an environment which supports start-up function of parallel processes on the cluster nodes. Moreover, when a cluster system is used by multiple users, some resource management, such as global scheduler, which provides automatic dispatching to the unused nodes, time sharing scheduler and so on, should be provided.

SCore cluster system software is not only software, which provides the start-up function of parallel processes, but also software that provides a multi-user environment, job management and the other sophisticated features, such as checkpoint-restart for long-period jobs. In a practical cluster system environment, users only specify the number of nodes and a program, then, cluster system software allocates the number of nodes and executes the user program automatically. As shown in the Figure 3.2., the SCore Cluster System Software consists of the following components:

- **PM II** - a low-level communication library for cluster computing. The PM communication facility is realized using a user-level communication [43, 49]. PM provides a virtual network mechanism called PM channel in order to realize a multi-user environment on the SCore cluster system software. The channel provides reliable data gram communication, instead of connection-oriented communication such as that of TCP/IP. Each process for the same parallel application uses a PM channel exclusively, and communicates using the same number of PM channels. A communication to another destination node is done using the node number. PM context is also introduced, which stores the status of the PM channel, and enables context switching on the SCore-D gang scheduler. PM realizes not only message passing but also a remote memory transfer (Zero-Copy communication), to provide high bandwidth communication. In remote memory transfer, a message is transferred using the DMA facility of the Fast Ethernet network without any memory copy operation by the host processor. Since the DMA facility accesses the physical memory address space, the user virtual memory must be pinned down to a physical memory location before the message is transferred. If each message transfer involves pin-down and release of kernel primitives, the message transfer bandwidth will decrease since those primitives are quite expensive. The pin-down cache technique, which reuses the pinned-down area to decrease the number of, calls to pinned-down and release primitives has been proposed and implemented with a Zero-Copy message transfer. The PM II API (Application Program Interface) is carefully designed so that many types of networks and shared memory are accessed in a uniform way for cluster computing. PM II drivers for Myrinet, Ethernet, UDP, and SHMEM have been implemented.

SCore-D - a user-level global operating system to utilize cluster resources, such as processors, networks. Various PM network devices are managed and utilized by SCore-D. SMP clusters, heterogeneous clusters as well as homogeneous clusters are supported. SCore-D provides a flexible parallel job-scheduling scheme. User parallel jobs are time-shared and space-shared simultaneously and effectively. SCore-D, and any user programs can be check pointed by only specifying a checkpoint interval time provide preemptive consistent check pointing.

The SCore-D global operating system is implemented as a set of daemon processes on top of a UNIX operating system, written in MPC++ without any kernel modification. To effectively utilize processor resources and to realize an interactive programming environment, parallel processes are multiplexed in the processors' space and time domains simultaneously under SCore-D. Parallel processes are gang-scheduled when multiplexed in the time domain. It has been proven that the SCore-D gang scheduler overhead is less than 4 % of the total application execution time.

To realize gang scheduling under a communication layer, which accesses the network hardware directly, the network hardware status and messages insight on the network must be saved and restored when switching to another parallel process. This mechanism is called network preemption. By co-designing PM and SCore-D, the network preemption technique has been developed.

- **SCASH** - a software for DSM (Distributed Shared Memory) system using PM II [57]. It employs the Lazy Release Consistency model with both update and invalidates protocols. SCASH is software distributed shared memory system using PM library, and memory management functions, such as memory protection, supported by an operating system kernel. It is implemented as a user level runtime library. SCASH is a page-based distributed shared memory system where the consistency of shared memory is maintained on a per-page basis. SCASH is based on the Release Consistency (RC) memory model with the multiple writer protocol.

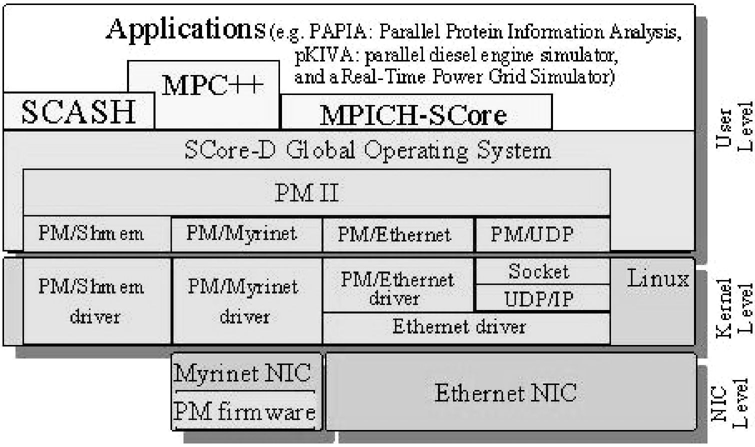


Figure 3.2. The scheme of SCore version 5 system software.

- **MPICH-SCore,** the former MPICH-PM/CLUMP, is based on the MPICH MPI library using PM II for inter process communication. MPICH-SCore is a high-performance MPI on workstation, PC, and SMP clusters. MPICH-PM is an MPI library developed for PM. When a message passing library such as MPI is implemented on top of a lower level communication facility that supports the Zero-Copy message transfer primitive, the message passing library must handle the pinned-down memory area which is a restricted quantity resource under a paging memory system. Deadlock can be caused by allocation of pinned-down memory by multiple simultaneous requests for sending and receiving without some form of control. MPICH-PM has overcome this issue and achieved a good performance.

MPICH-PM/CLUMP, the successor of MPICH-PM, supports a cluster of multiprocessors, called CLUMP. Using MPICH-PM/CLUMP, The MPI legacy programs run on CLUMP without any modifications. For example, suppose we have a CLUMP consisting of 16 nodes, each of which contains dual processors.

MPICH-PM/CLUMP provides an MPI application with 32 processors or 32 processes. The communication between two processes on different nodes is realized by the PM communication facility using an Ethernet network. Message transfer between two processes on one node is handled by the Direct Memory Copy technique which copies between the two processes directly using the PM kernel-level driver.

- **MPC++** - a multi-threaded C++ using the C++ template feature. It provides synchronous/asynchronous remote function invocation facilities, synchronization structures, global pointer, and other such functions. MPC++ also supports a meta-level architecture to achieve an extensible and modifiable programming language system. The architecture makes it possible to incorporate new optimizers into the compiler. Library designers are able to provide an efficient specific to their class/template library in the library header file. The library user may use such a high performance library by including the header file. The MPC++ meta-level architecture is not included in this release. SCore-D is written in MPC++.

3.3.5. Network Trunking on PM/Ethernet

Network with PM/Ethernet is the method for minimizing information exchange cost. There are significant design choices: where should a reliable protocol be processed, in the host CPU or in the NIC CPU; and where should the data structures for managing of the send and receive buffers and triggering be allocated, in the host or in the NIC memory (Figure 3.3).

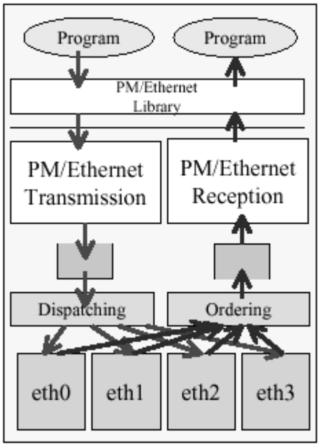


Figure 3.3. PM Truncing using Ethernet Network.

Since those structures are shared resources, the cost between host and NIC information exchange is crucial in performance.

The Network Trunking technique solves the problems of the Channel Bonding technique by rebuilding the Ethernet frame using the PM/Ethernet mechanism to all of the destination nodes. Features of the Network Trunking technique on PM/Ethernet are described as follows:

1. Higher network communication bandwidth can be providing by multiple network usage.
2. Implementation:
   - Using Ethernet MAC addresses of other PM/Ethernet units.
   - On message transmission: selects a NIC, and replaces Ethernet MAC addresses and submits the message to the NIC's driver.
   - On message reception: introduces a receive message queue for message ordering.

3.3.6. Features of SCore Software

SCore software supports the following features (Table 3.1), which are very important for parallel calculations.

Table 3.1. The features differences of two parallel environments used in research work

Features	**SCore Cluster**	**Beowulf Cluster**
Multi – User Support	**Support**	-
Multiple Network Support	**Support**	-
High Availavly	**Support**	-

- **Single System Image**

Using SCore, users are not aware whether or not a system is a cluster of single/multiprocessor computers or a cluster of clusters. A parallel application and an ordinary UNIX command may run by just specifying a computer node group of such a cluster. A UNIX command runs in the SIMD execution style.

- **Multiple Network Support.**

The PMvII high performance communication library is a dedicated communication library for cluster computing using many types of networks. PMvII allows a program to communicate on different types of networks. PMvII drivers for Myrinet, Ethernet, UDP, RHiNET, SCI and the Shmem shared memory interface have been implemented.

- **Seamless Programming Environment.**

The *Hmake* command enables users to compile a program on heterogeneous computers. It generates binaries for different underlying types of execution environments such as Intel Pentium, Itanium and Compaq processors. Using the MPC++ Multi-Threaded Template Library, a program runs on such a heterogeneous processor environment.

- **Heterogeneous Programming Language.**

Programs written in the MPC++ Multi-Threaded Template Library (MTTL) may not only run on homogeneous computer environment but also run on heterogeneous computer environment without modifying the code.

- **Multiple Programming Paradigms.**

Unlike other cluster software, SCore not only supports the message passing paradigm, but also supports the shared memory parallel programming paradigm and the multi-threaded parallel programming paradigm.

Parallel programming support:

- **Real-time process activity monitor.**

SCore-D allows us to watch parallel process activity in real-time using the Real-Time Load Monitor. Each bar of the Real-Time Load Monitor represents the processor utilization on each node.

- **Deadlock detection.**

Since SCore-D knows the global status of a parallel process, i.e., each process status and communication buffer status, it can detect whether or not the parallel process is deadlocked.

- **Automatic debugger attachment.**

In most cluster systems, when a process dies, there is no chance to invoke a debugger interactively. SCore attaches the gdb(1) debugger to the target parallel process when an exception signal is detected.

Fault Tolerance:

– **Preemptive checkpoint.**

To enable a checkpoint function, the SCore checkpoint facility does not require any additional API and does not assume any parallel programming languages. The user parallel process image is stored to a local hard disk for checkpoint by a user specified interval. Moreover, the user's process image on the local hard disk is stored redundantly, so, when one of the checkpointing data disk is broken, the user's process image is able to re-produce using the data of the other nodes.

– **Parallel process migration.**

Using the checkpoint function, a parallel process may migrate to another group of computers in SCore.

– **Flexible Job Scheduling.**

To utilize processor resources and to enable an interactive programming environment, the SCore-D global operating system multiplexes parallel processes in processors' space and time domains simultaneously.

– *Gang scheduling*. Parallel processes are gang-scheduled when multiplexed in the time domain.

– *Batch scheduling*. Batch scheduling is implemented by setting an infinite value for the scheduling slice time.

SCore cluster software system is introduced to show that a compact and well maintainable self-made from commodity hardware PC Cluster TAURAS using this cluster system software can be comparable with known commercial clusters. The NAS parallel benchmark results, which are introduced below, show that practical high-performance cluster systems can be built using commodity hardware and demonstrate good results such as PC Cluster TAURAS.

3.4. Cluster Installation Procedure

Installation of clusters is extremely difficult and it requires knowledge on computer architecture, OS and certain cluster technologies. Furthermore, integration of the cluster technologies described is a complex procedure. Except for integration, a special problem is presented by enhancements of the versions of certain systems. Problem with enhancements is interdependency of components.

Both problems are solved by using cluster distributions – integration and enhancements. Additionally, cluster distributions usually have large communities of users whom they can turn to for help in case of errors in cluster distribution operations.

3.4.1. Platform and Architecture Requirements

The minimum list of basic parts that are necessary before you can start building your SCore Cluster System and optional parts must be described.

Basic Parts - Processors and Chipsets

The Intel Pentiums (Pro, II, III, 4) and Compaq Alphas have been tested in Real World Computing Partnership, or are known from SCore user's information in Table 3.2. Configuration of CPU can be Single or SMP.

Memory

Random-access memory (RAM) will be more than 64MB. Memory size depends on you applications. The SCore system software uses just a few Mbytes of RAM.

Table 3.2. The basic parts-processors and chipsets.

Processors	**Chipsets**
Intel Pentiums(Pro, II, III, 4)	IBM compatible PCs using 440FX,440BX,450NX 4-way Pentium II Xeon 450 2-way Pentium III 500MHz 2-way Pentium III 800 MHz
AMD Duron, Athlon (XP)	AMD 760 chipset 2-way AMD 760MP chipset 2-way AMD 760MPX chipset
AMD Opertron, Athlon 64	AMD 8131 chipset AMD 8111 chipset
Compaq Alpha 21164	21172 21174 chipset
Compaq Alpha 21264	Compaq 21272(Tsunami) chipset on XP1000 2-way Compaq 21272(Tsunami) chipset on DS20E
Intel Itanium2	2-way HP zx1 chipset on zx6000

Disk space

It will be need the 3 GB or more disk space for each compute host, 4 GB or more disk space for the server host (Table 3.3).

Table 3.3. HDD usage for SCore cluster system installation.

	Server host	**Compute host**
System	4 GB	2 GB
User	~	~
ChkPnt	0 B	1GB or more
Total	4 GB or more	3 GB or more

Ethernet Card and HUB/Switch

It is strongly recommended using an Ethernet switch or switching HUB instead of a HUB.

The following network cards have been supported on PM/Ethernet:

- Intel Pro 1000
- Broadcom Tigon3

Network Information is very important for cluster, so before installation it is preparing such information: server IP address, compute host IP addresses net mask, default gateway IP address, domain name, NIS domain name.

The other important information:

• Host names should be FQDN (Fully Qualified Domain Name).

• The host name and IP address database must be consistent in DNS, NIS, and */etc/hosts* file.

VGA

The SCore system does not require a VGA card in compute hosts, but requires the card in the server host. Note that most BIOS require a VGA card and a keyboard.

Floppy and CD-ROM

A floppy drives on the all compute hosts is required for use by the Easy Installation Tool. A floppy and CD-ROM drive on the server host are required for use by the Easy Installation Tool.

3.4.2. SCore Software Installation

After preparation of hardware for cluster there is other steps, you must to do. All lists are taken below.

1. To install the SCore software on your cluster:

- Obtain the gziped ISO 9660 images (505 MB) which includes SCore i386 binariy rpm and sources and minimum Linux system rpm files you can at page http://www.pccluster.org/score/dist/pub/score-5.2.0/score-5.2.0-redhat7.3.i386.iso.gz
- Issue the following command:
- # gunzip score-5.2.0-redhat7.3.i386.iso.gz
- Write the image into a CD-R

- o Prepare an SCore server host in which Red Hat 7.3 has been already installed with **NO FIREWALL** setting.
- o Mount the CD-R on the CD-ROM drive of the SCore server host
- o Then issue the following commands in the SCore server host:
- o # mount /mnt/cdrom
- o # cd /mnt/cdrom
- o # ./Install
- o See the /opt/score/doc/ directory

2. To obtain the SCore i386 binary rpm for Red Hat 7.2 or Red Hat 7.3, please go here: http://www.pccluster.org/score/dist/pub/score-5.2.0/source/.
3. To obtain the SCore source, please go here: http://www.pccluster.org/score/dist/pub/score-5.2.0/source/.
4. You may see the SCore 5.2 online documentation here: http://www.pccluster.org/score/dist/score-5.2.0/html/index.html.en

It is possible to install system from SCore source in two ways - manually or by Easy Installation tool.

3.4.2.1. Manually installation

For this type cluster installation, first step will be to install all nodes and master node with Linux OS Red Hat 7.2 or Linux OS Red Hat 7.3 and then install the SCore i386 binary rpm.

3.4.2.2. Easy Installation Tool

If you decided to install cluster by Easy Installation tool you must install Linux OS Red hat 7.2 or Linux OS Red Hat 7.3 only on the master node. On the other nodes OS will be installed automatically.

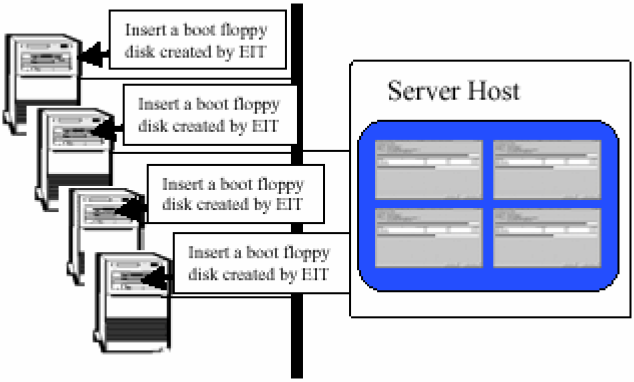

Insert a boot floppy disk created by EIT
Insert a boot floppy disk created by EIT
Insert a boot floppy disk created by EIT
Insert a boot floppy disk created by EIT
Server Host

Figure 3.4. The installation by EIT uses graphical guide on master node.

This installation type uses graphical guide on master node (Figure 3.4), so you can to keep the track of all processes of installation performance. At the same time it is possible to install for nodes. So to install cluster by Easy Installation tool you will need for installation and configuration a less time.

Overview of Installation by EIT

1. EIT is based on the Red Hat anaconda tool.
2. Network setting in the server host
    - Kick start files for anaconda are created
3. Boot floppy disk is created
    - The boot floppy disk does not include host information.
4. Compute host installation:
    4.1. Booting using the floppy disk
    4.2. IP address is obtained by the EITD running in the server host
    - IP address is assigned.
    4.3. File systems in the server host are mounted so that the kick start file is read.
    4.4. Network installation
    4.5. SCore setting (post processing).
    4.6. Reboot.
5. Server daemons settings in the server host

For building PC cluster of 16 compute hosts, it will be need 17 computers (one for cluster server), but it was possible install software for the cluster server onto one of the compute hosts. For PC cluster TAURAS for the server was used one of the computer hosts.

3.4.3. Setting of Test Tasks

After installation it is important to test the following servers, which are running to maintain the system for parallel calculations:

- **Cluster Database Server.** All the cluster information required by the SCore Cluster System Software is maintained by the *scoreboard* server.
- **Compute Host Lock Server.** The Compute Host Lock Server (*msgbserv*) provides a locking facility that prevents other users from using cluster hosts. The *msgb* command allows cluster users to browse the current status of a cluster. Users of parallel application programs are assumed to lock hosts via the *msgbserv* server. This locking management is done automatically by user programs when they are linked with the *SCore* runtime library.
- **System Information Server.** The *scbcast* server obtains system information generated by the *SCore-D* operating system and distributes it to clients. *scbcast* must be invoked before starting the *SCore-D* operating system and clients.

Basic system tests:

- SCOUT Test Procedure

Try the following test procedure on your server host to verify your cluster operation. If you have installed the system by your hand, please make sure that the SCBDSERV shell environment is set. If not, please login again. If you still do not see the variable, please make sure that a profile file for your login shell has been created under the /etc/profile.d, which is described in the server host settings section.

- PM Test Procedure

This test procedure assumes that the X window system is already running on the server.

- SCore Test Procedure

Try the following test procedure on your server host to verify your cluster operation. If you have installed the system by your hand, please make sure that the SCBDSERV shell environment is set. If not, please login again. If you still do not see the variable, please make sure that a profile file for your login shell has been created under the */etc/profile.d*, which is described in the server host settings section.

- Demonstrations

The following X Window demonstrations require the host where *scrun* is invoked to be allowed to make connections to the X server where the program output will be viewed.

If you wish to test more example programs you will find them located under the */opt/score/example* directory.

If all tests pass correctly it means that PC cluster with a cluster server used for the following purposes: a file server for SCore cluster administration the Cluster Database Server, *scoreboard*, and the Compute Host Lock Server, *msgbserv* are running on this computer; software development compilation, running cluster maintenance daemons.

3.5. NAS Parallel Benchmarks

The Numerical Aerospace Simulation (NAS) Parallel Benchmarks are a set of 8 programs designed to evaluate the performance of parallel supercomputers [8]. The NPB was developed by the Numerical Aerospace Simulation Systems Division of the NASA Ames Research Center in order to provide the Nation's aerospace research and development community a high-performance, operational computing system capable of simulating and entire aerospace vehicle system within a computing time of one to several hours.

NPB 2.3 is a set of eight benchmarks based on Fortran 77 (with a few common extensions that are also a part of Fortran 90) and the MPI message-passing standard. They are intended to run with little or no tuning, approximating the performance of a portable parallel program on a distributed memory computer. The benchmarks are not intended to test only MPI, but to measure the overall system performance.

Table 3.4. Different problem size of NPB 2.3 benchmarks.

Benchmark code	Class W	Class A	Class B	Class C
CG	7000	14000	75000	150000
FT	128^2x32	256^2x128	512x256^2	512^3
LU	32^3	64^3	102^3	162^3
EP	2^{17}	2^{28}	2^{30}	2^{32}
BT	33^3	64^3	102^3	162^3

The programs of the NPB suite are based on computational fluid dynamics (CFD) applications. The NPB programs are written using MPI, so, the NPB programs can be executed on any system, which supports MPI. The NPB programs have several classes, which depend on data size, called class S, W, A, B, C, and D. Class S uses the smallest data size and class D the largest. All of the NPB programs, except IS, execute floating-point calculations, and the communication patterns of each benchmark program are different from the others.

CG, FT and LU runs on a power-of-two number of processors, whereas SP and BT require a square number of processors. FT and CG are kernel benchmarks and the rest are application benchmarks. To appropriately test different sizes of supercomputers, NPB 2.3 contains four different classes of problem sizes (Table 3.4). The classes are W (workstation), A, B, and C.

From the original set of eight benchmarks was selected five to measure the performance of the cluster TAURAS. They are divided into two groups depending on their utilization of CPU, memory and network: kernel benchmarks and application benchmarks. The kernel benchmarks are intended to put pressure on the Linux kernel with its implementation of the TCP/IP stack. The application benchmarks concentrate more on CPU and memory utilization.

Conjugate Gradient (CG) CG is used to compute an approximation to the smallest value of a large, sparse, symmetric positive definite matrix. It represents typical unstructured grid computations with its test of irregular long distance communication using unstructured matrix vector multiplication.

Table 3.5. Class "A" NAS Parallel Benchmarks and Dominant Message characteristics for 16 nodes programs.

Name	**Note**	**Message characteristics on Class A, 16 node**		
		Dominant Sizes of Messages	**Number of Messages**	**Execution Time**
CG	Conjugate Gradient	16-32 Kbytes	1248 / PE	14.11 sec.
EP	Embarrassingly Parallel	8-16 Kbytes	12 / PE	48.93sec.
FT	Fourier Transform	512-1024 Kbytes	128 / PE	28.63 sec.
LU	LU Decomposition	512-1024 Kbytes	46000 / PE	113.32 sec.
BT	Block Tridiagonal Solver	16-128 Kbytes	4800 / PE	200.17 sec.

3-D FFT PDE (FT) FT contains the computational kernel of a three-dimensional FFT-based spectral method. It performs 1-D FFTs in the x and y dimensions on a distributed 3-D array, which is done entirely within each processor, and then continues with an array transposition, which requires an all-to-all communication. The final FFT is then performed.

LU solver (LU) LU simulates a CFD application, which uses successive over relaxation (SSOR) to solve a block lower-block upper triangular system of equations, derived from an unfactored implicit finite-difference discretization of the Navier-Stokes equations in three dimensions.

Block tridiagonal solver (BT) BT originates from the same problem as SP, but instead of solving scalar systems, it solves block-triangular systems of 5 x 5 blocks.

Table 3.5 shows some of the NPB programs and their dominant message characteristics. Class "A" 16 node programs are used to show examples of message communication patterns: approximate message sizes; number of messages and execution time on a 16 node cluster with a Pentium III 733MHz processor and Fast Ethernet - Network.

The data in the Table 3.5 shows that the CG and FT programs are responsive to communication band-width and latency. The EP, LU and BT programs are not responsive to communication performance when compared with CG and FT. The FT program use All-to-All collective communication. The dominant message size of the FT Class A program is in the 512-1024 Kbytes range on a 16 node cluster. The CG program uses neighbor-to-neighbor communication and sends a number of messages continuously. The dominant message size of CG Class A is in the 16-32 Kbytes range. The LU program also uses neighbor-to-neighbor communication and sends messages periodically.

Application benchmark: LU

NAS NPB test LU is a simulated computational fluid dynamics application, which uses symmetric successive over-relaxation (SSOR) to solve a block lower triangular-block upper triangular system of equations resulting from an unfactored implicit finite-difference discretization of the Navier-Stokes equations in three dimensions.

A 2-D partitioning of the grid onto processors occurs by halving the grid repeatedly in the first two dimensions, alternately *x* and then y, until all power-of-two processors are assigned, resulting in vertical pencil-like grid partitions on the individual processors. The ordering of point-based operations constituting the SSOR procedure proceeds on diagonals, which progressively sweep from one corner on a given *z* plane to the opposite corner of the same *z* plane, thereupon proceeding to the next *z* plane.

Communication of partition boundary data occurs after completion of computation on all diagonals that contact an adjacent partition. It results in a relatively large number of small- to medium-sized communications. The results of the LU benchmark are plotted in Figure 3.5. The LU benchmark shows that the cluster TAURAS handle well the scaling of computationally intensive processes with many small- to medium-sized communications. The decrease in execution time is consistent as processors are added. The Figure 3.5 also shows the results of other clusters, so it is possible to compare the cluster TAURAS performance [8].

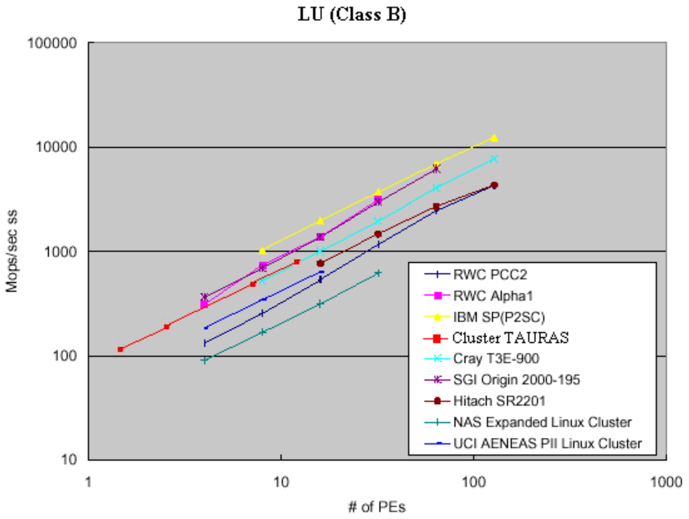


Figure 3.5. NAS NPB calculation results for solving LU test class B on Cluster TAURAS. Results for other supercomputers are from [9].

Kernel Benchmark: FT

The FT benchmark (Figure 3.6) solves a partial differential equation (PDE) using forward and reverse 3-D Fast Fourier Transforms (FFT's).

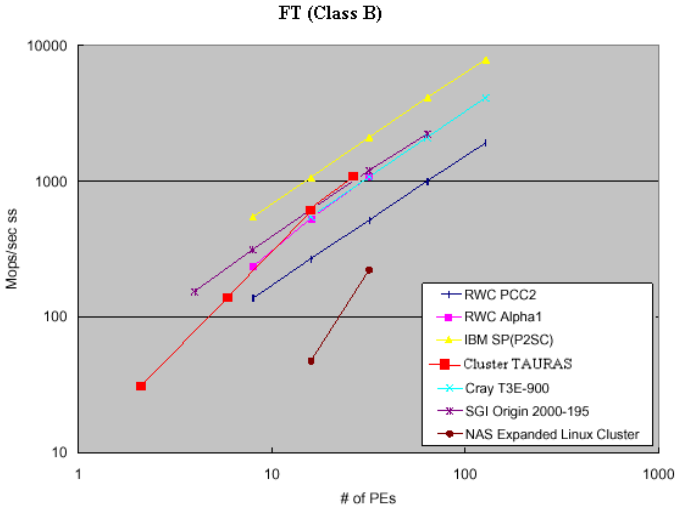


Figure 3.6. *NAS NPB calculation results for solving FT test class B on Cluster TAURAS. Results for other supercomputers are from [9].*

FT tests both CPU and message passing, as there are a number of array operations which require that each process share its portion of the array with the other processes. The implementation of the 3-D FFT PDE benchmark follows a standard scheme.

The 3-D array of data is distributed according to *z* planes of the array with one or more planes stored in each processor. The forward 3-D FFT is then performed as multiple 1-D FFT's in each dimension, first in the *x* and *y* dimensions, which can be done entirely within a single processor, with no inter processor communication. An array transposition is then performed, which amounts to an all-to-all exchange, wherein each processor must send parts of its data to every other processor. The final set of 1-D FFT's is then performed. A conventional Stockham-transpose-Stockham scheme is used for the 1-D complex FFT's. This procedure is reversed for inverse 3-D FFT's. The communication between processes is medium grained with larger message sizes for smaller number of processes. Figure 3.6 shows the results of cluster TAURAS of this benchmark.

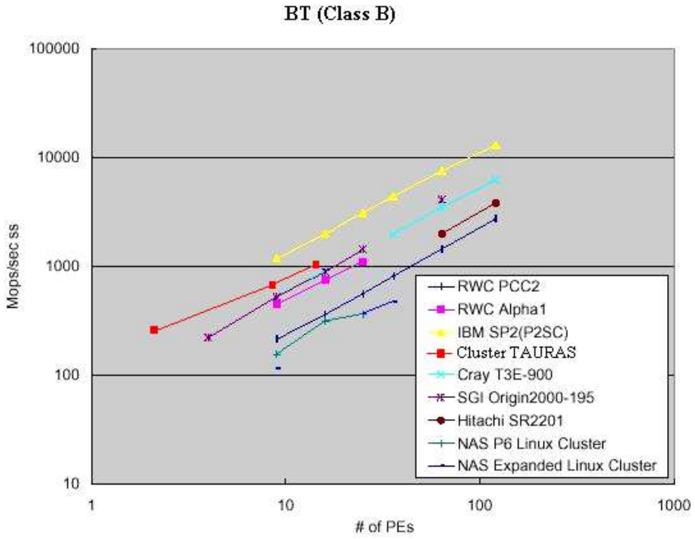


Figure 3.7. *NAS NPB calculation results for solving BT test class B on Cluster TAURAS. Results for other supercomputers are from [9].*

Application Benchmark: BT

The BT algorithm solves three sets of uncoupled systems of equations, first in the *x*, then in the *y* , and finally in the *z* direction. These systems are block tridiagonal with 5x5 blocks in the BT code.

The BT code requires a square number of processors. This code have been written so that if a given parallel platform only permits a power-of-two number of processors to be assigned to a job, then unneeded processors are deemed inactive and are ignored during computation, but are counted when determining Mflop/s rates (Figure 3.7).

Kernel Benchmark: EP

The NPB implementation of the EP (Embarrassingly Parallel) benchmark has its origins in the NAS provided reference code from 1991. The current implementation can run on any number of processors. Each processor independently generates pseudorandom numbers (PN's) in batches of 2^{17} and uses these to compute pairs of normally distributed numbers [11].

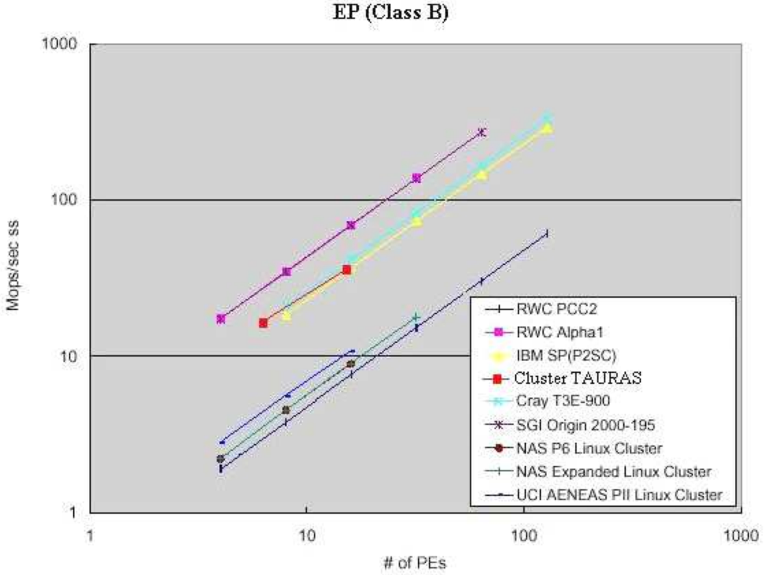


Figure 3.8. NAS NPB calculation results for solving EP test class B on Cluster TAURAS. Results for other supercomputers are from [9].

The rounded norms of these pairs are tallied before the next batch is created and processed. No communication is needed until the very end when the tallies of all processors are combined Performance of the code which is reported as the number of PNs produced per second is governed by the PN generator see below. The calculation also contains a significant number of logarithm and square root operations. This benchmark tests the pure combined computational power of each system. Separate sections of the uniform pseudorandom numbers can be independently computed on separate processors. The only requirement for communication is the initial distribution of the processes and the combination of the 10 sums from various processors at the end. Figure 3.8 shows Cluster TAURAS computational power.

Kernel Benchmark: CG

The CG benchmark uses the inverse power method to find an estimate of the largest value of a symmetric positive definite sparse matrix with a random pattern of nonzero [61].

With computation spread out by the large number of zero values and randomly placed non-zero values, CG is a good measurement of random communications between processes. The random communication will vary in size and number of messages passed between processes. The results from CG are shown in Figure 3.9.

The results of some NPB Class "B" tests are based on the histograms in the Figure 3.5-Figure 3.9, respectively. The message sizes and number of messages are different based on the number of nodes and the class.

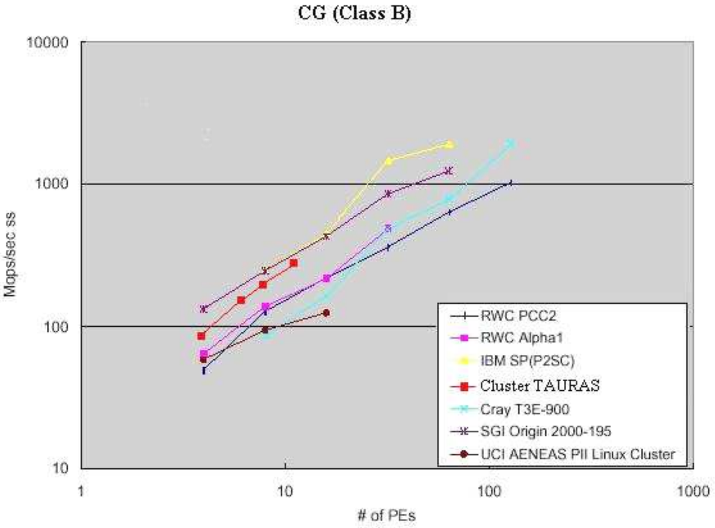


Figure 3.9. NAS NPB calculation results for solving CG test class B on Cluster TAURAS. Results for other supercomputers are from [9].

The NPB results on several commercial parallel computers are published on the NPB web page (http://www.nas.nasa.gov/Software/NPB/). The NPB results have been widely measured on a number of parallel machines, and published in various papers. Therefore, it is possible to compare performance with that of others.

3.6. Conclusions of this Chapter

First, there was discussed the increasing need for powerful computing. As technology prices dropped, new alternatives to expensive supercomputers emerged. Among the methods explored, two architectures of parallel computing which provide substantial computing power are Distributed Shared-Memory (DSM) machines, and clusters with System Area Networks (SAN's). These systems provide reliable, powerful computing and allow for some degree of scalability. However, the communication between processes can be a bottleneck for both kinds of systems [1].

Clusters consist of PCs connected together by a high-performance network [2]. This arrangement allows inexpensive cots nodes to be combined to provide substantial computing power at a lower cost than traditional supercomputers. Each computing node has its own processor, memory, and storage. These nodes usually communicate through the high-performance network by various means of message passing. The need for powerful computing has led to the development the clusters system software for administering. Such as SCore cluster system software, which was used for PC cluster TAURAS.

This section shows the strengths and weaknesses associated primarily with the PC cluster TAURAS architecture of the SCore system clusters. By analyzing the performance of the communication and computational aspects of the system and comparing the NAS NPB 2.3 benchmark results with the known architectures clusters.

The PC world is changing so fast, taking one year to develop a PC cluster would put behind, and the cost effectiveness of the PC cluster would be lost. Using commodity hardware, we can use a large software resource and to devote energies to the software research.

4. INVESTIGATION OF PC CLUSTER PRODUCTIVITY

PC Cluster TAURAS productivity was realized for investigations of explosive materials such as TNT molecule by means of no empirical quantum mechanical computation methods. The accurate no empirical quantum chemical computations (molecular electronic structure and vibration spectra investigations) are essential in spectroscopy. The basis of these investigations is quantum mechanics, but in order to achieve quantitative results comparable with experimental we need supercomputer's power or parallel computers cluster. Such tasks are solved nowadays in this way.

On the other hand the highly sophisticated quantum chemical computations have been becomes the routine tests of supercomputers performance. The tasks solved were used to compare the possibilities of SCore cluster system software with Beowulf [2]. During investigations the GAMESS quantum chemistry computer code was used [3].This packed was integrated and adapted in cluster TAURAS environment. All calculation results are presented in this chapter.

4.1. Quantum Chemistry Package GAMESS

Quantum chemistry is the discipline of computing the properties of atoms and molecules using quantum mechanics. Quantum chemists today are involved in both developing and applying these methods. Solving the equations governing the properties of molecules is extremely complex, and it cannot be done exactly. A hierarchy of approximate methods has therefore been developed that allow quantum chemists to select the method most appropriate for a given problem. Which methods are practicable depends on the computational expense of applying each method to a particular problem. The least computationally expensive methods also tend to be the least accurate methods. Such methods require an amount of computer time that formally grows like the fourth power of the size of the molecule being studied. That is, doubling the size of the molecule will require sixteen times as much computing power. The most expensive methods require processing time that grows exponentially in the size of the molecule. Between these extremes are a variety of methods whose computational expense grows like the fifth, sixth, or higher power of the size of the molecule.

Clearly, the expense of quantum chemistry calculations is significant. Two approaches are being used to deal with this cost. The first is the ongoing effort to reformulate existing methods and develop new methods that greatly reduce the scaling of the computational cost with respect to the size of the molecule. The second approach is to apply more computing power to the problem, ideally in addition to using reformulated methods. The motivation for developing quantum chemistry programs is to use clusters and massively parallel machines.

As was explained above, the theoretical investigations of explosive molecules were performed by means of *ab initio* quantum mechanical calculations using GAMESS computer code [2].

Wide ranges of quantum chemical computations are possible using GAMESS:

1. Calculates RHF, UHF, ROHF, GVB, or MCSCF self-consistent field molecular wave functions.
2. Calculates CI or MP2 corrections to the energy of these SCF functions.
3. Calculates semi-empirical MNDO, AM1, or PM3 RHF, UHF, or ROHF wave functions.
4. Calculates analytic energy gradients for all SCF wave functions, plus closed shell MP2 or CI.
5. Optimizes molecular geometries using the energy gradient, in terms of Cartesian or internal cords.
6. Searches for potential energy surface saddle points.
7. Computes the energy Hessian, and thus normal modes, vibrational frequencies, and IR intensities.
8. Traces the intrinsic reaction path from a saddle point to reactants or products.
9. Traces gradient extremely curves, which may lead from one stationary point such as a minimum to another, which might be a saddle point.
10. Follows the dynamic reaction coordinate, a classical mechanics trajectory on the potential energy surface.
11. Computes radiative transition probabilities.
12. Evaluates spin-orbit coupled wave functions.
13. Applies finite electric fields, extracting the molecule's linear polarizability, and first and second order hyperpolarizabilities.
14. Evaluates analytic frequency dependent non-linear optical polarizability properties, for RHF functions.
15. Obtains localized orbitals by the Foster-Boys, Edmiston-Ruedenberg, or Pipek-Mezey methods, with optional SCF or MP2 energy analysis of the LMOs.
16. Calculates the following molecular properties:
    a) dipole, quadruple, and octupole moments
    b) electrostatic potential
    c) electric field and electric field gradients
    d) electron density and spin density
    e) Mulliken and Lowdin population analysis
    f.) viral theorem and energy components
    g) Stone's distributed multiple analyses
17. Models solvent effects by:
    a) effective fragment potentials (EFP),
    b) polarizable continuum model (PCM),
    c) self-consistent reaction field (SCRF).

4.1.1. Distributed Data Parallel code on MPP

For parallel running of GAMESS all code needed to implement the Distributed Data Interface (DDI) is provided with the GAMESS source code distribution, the program compiles and links ready for parallel execution on all machine types.

GAMESS parallel code is based on the Distributed Data Interface (DDI). DDI relies on low-level message transports: SHMEM, TCP/IP sockets, or MPI-1.

In order to isolate the quantum chemistry application GAMESS from any particular messaging library, a new application-programming interface (API) has been designed. Every call in this API begins with the letters DDI, making it easy to locate all parallel constructs in the application code. The code implementing the DDI calls is collected in small interface files, which translate these application level calls to the appropriate lower level routines, which accomplish the needed tasks.

DDI includes the traditional global operations and point to point messages that one would expect. Routines DDI_CREATE and DDI_DESTROY allocate and deal locate distributed data structures. At present DDI supports only doubly subscripted FORTRAN arrays, with complete columns stored on any particular node, in keeping with the normal FORTRAN storage convention. The workhorse distributed memory access routines are DDI_PUT, DDI_GET, and DDI_ACC, which deal with "patches" of memory within the large distributed array. Note that when a patch falls across the memory of two or more nodes, it is treated as separate sub patches. In general, these three routines cloak details about which node's memory actually stores the patch of data. However, since access to the portion of the distributed data which happens to be stored locally should be much more efficient, DDI_DISTRIB reports which columns are available locally, so that algorithms can maximize their use of locally stored data.

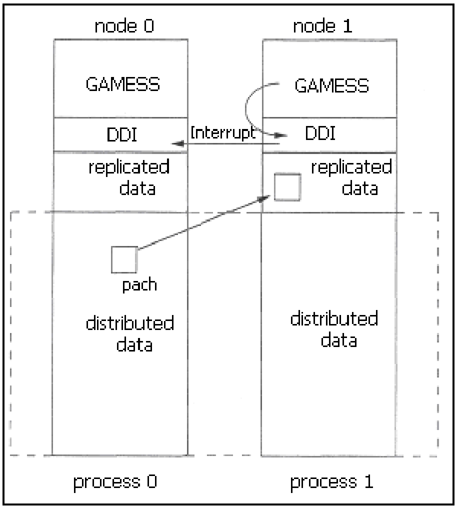


Figure 4.0. DDIT3E file provides direct call translation from DDI to Cray's T3E SHMEM.

On the Cray T3E system, DDI has been implemented over the system SHMEM library by writing a special file named DDIT3E. This file provides a nearly direct call translation from DDI to Cray's SHMEM, a full function one-sided messaging library, which is relatively easy to use. Near linear speedups of the MP2 program in GAMESS to 128 nodes and beyond on the T3E have been demonstrated elsewhere.

The single program image model shows Figure 4.0. In this way GAMESS runs on the T3E. In addition, Figure 4.0 illustrates a DDI_GET operation to bring a patch of distributed memory into local memory for computation. It is instructive to consider how this happens, as it illustrates the programming difficulties one encounters in implementing this kind of functionality on different hardware. When the GAMESS application running on node 1 decides it needs a particular patch, it calls DDI_GET to obtain it. Within the DDI subroutine, a decision is made about which node actually owns that patch, meaning that the application layer need not know such details. If it happens that this patch belongs to node 0, the computations on node 0 must be interrupted long enough to obtain the patch. The actual transfer of the patch is a conventional point to point message. (The hardware design on the T3E includes a Block Transfer Bus to handle memory requests, rather than interrupting the main processor.) This is the origin of the terminology "active" or "one-sided", since the procedure is driven by the requirements of node 1. In addition to interrupt handling, one must also deal with the issue of memory locking, since if two nodes were to decide to accumulate (DDI_ACC) to the same patch, the one which starts first must finish before the second is allowed to accumulate its values. This is accomplished by placing a lock upon the entire portion of distributed memory on the node to which data is being accumulated, rather than attempting to lock specific sub regions.

4.1.2. Distributed Data Parallel Code on PC Clusters

On a cluster parallel computer, a powerful active message-passing library such as SHMEM is not likely to exist. As just mentioned, the use of distributed data requires the capability to interrupt computations on remote nodes shortly to access their memory, and to guarantee exclusive access to that memory during the operations. A two-process model has been adopted to solve both problems, as shown in Figure 4.1. The GAMESS application is executing and owns any replicated storage by one process, termed the "compute process". A second process, termed the "data server", owns the portion of the node's memory, which is dedicated to the distributed data structures. This second process runs in a loop in the DDI service routine, sleeping until a data access request arrives. This scheme means the compute process is unaware of any interruption, since the operating system mediates giving the data server a time slice when a DDI service request message arrives. Memory locking is taken care of by having each data server handle only one DDI request at a time, until it is completed. Thus, the two process model nicely finesses the need to program for these two complex system level issues.

Note that each process, whether a "compute process" or a "data server" is actually a copy of the GAMESS program. Each program decides which role it is to play based on whether its process ID is in the first or second half of the total. Ordinary operations such as global sum or broadcast involve only the first half of the processes. This is easily implemented, for example, by use of communication groups in MPI-1.

At first glance, it may seem wasteful to have the GAMESS application object code and local variables duplicated in the data servers, but since this code is never executed, it actually represents just a few wasted Mbytes in the system swap partition. It is simpler not to deal with a separate, lightweight data server main program.

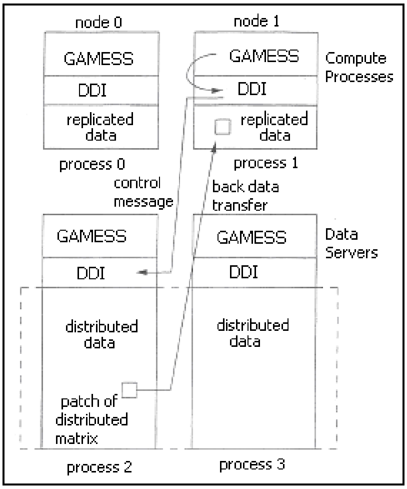


Figure.4.1. Each DDI distributed memory access involves two messages.

As shown in Figure 4.1, each DDI distributed memory access involves two messages. The first is a "control message" of length 24 bytes specifying the type of operation and what patch is being touched. Following this short message, a "bulk data message" follows in which the entire patch is transferred. While the programmer is presumably trying to write algorithms in which the size of the patches is reasonably big, the presence of an equal number of short control messages means that low communication latency is as important as high bandwidth for DDI to function well.

The scheme depicted in Figure 4.1 requires only ordinary point-to-point messages, and thus can be run over a variety of messaging libraries, including MPI-1 and TCP/IP sockets. MPI-1 is used as the low level-messaging layer on IBM SP systems, since a significant fraction of the SP install base at the time this is written either has a switch that does not support LAPI or has not been upgraded to recent IBM software releases. The use of two processes on a SP node is not without penalty, as older systems are rendered unable to use the special low latency MPI-1 mode, which IBM terms "user space". As the older SP systems begin to disappear, a single program image with LAPI as the "active message" support layer will be implemented, analogous to that used on the Cray T3E. Since SGI now includes MPI-1 as part of the Origin operating system, the same FORTRAN translation layer from DDI application calls to MPI-1 point-to-point messages will be used on this machine.

On cluster systems, MPI-1 may or may not be available, depending on the vendor and whether a license for this has been purchased. However, since the existence of TCP, IP sockets is assured, low level code to deal directly with socket messaging has been written to implement the scheme shown in Figure 4.1 on low cost clusters.

Since sockets are considerably removed from the typical programming knowledge of computational chemists, a summary of the system calls needed may be of interest [32]. First, one needs a program to initiate processes on all CPUs; this "kickoff" program is called DDIKICK. This is a C program, which proceeds by system calls to fork to create duplicate of it self. Each newly generated child process then replaces itself with a new program, either direct execution of a GAMESS process on the local node by the *execvp* call, or by invoking *rsh* to generate a GAMESS process on a remote node. While *rsh* and its associated *rhosts* authentication scheme present some security concerns, this remote process generation mechanism is ubiquitous in UNIX. After all GAMESS compute processes and data servers have been generated for p CPUs, DDIKICK opens a socket to each of its 2p children, and facilitates establish direct socket connections between them. Each compute process has a total of 2p sockets open, one to every data server, one to every other compute process, and one to the kickoff program.

Data servers have just *p + 1* socket, one to every compute process and one to the kickoff program. There is, of course, no need for a socket connecting pairs of data servers. Inter-child socket establishment begins by a call to socket by both processes, then one side begins the connection by a call to connect and the other side answers by the calls bind, listen, and accept. Fortunately, once the socket connections are established, data is transferred very simply, by send and *recv* calls. All operations, even the simple barrier synchronization, must be built up from these two calls. Signal and kill are used to terminate all other process in the event one exits abnormally.

4.2. Parallel GAMESS on SCore PC Cluster

In this research work, the parallel GAMESS compute code is continuously developing to take into account advances in HPC hardware and software as shown in Figure 4.2. The local DDI implementations mean that the essential parallel libraries fully support all necessary communication operations.

The no local implementations of DDI imply the essential parallel libraries are deficient in some method and that additional explicit programming is required to reach full DDI functionality. Any implementation must support all necessary communication operations require by DDI. The “virtual shared-memory” is the DDI programming model, where a segment of the physical memory available to each processor is designated for the storage of distributed data.

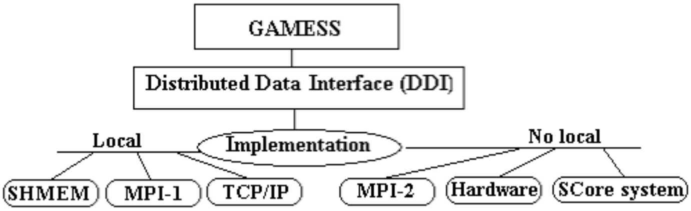


Figure. 4.2. The local implementation of Distributed Data Interface (DDI) is provided with the GAMESS source code distribution and no local implementation require explicit programming or special models to achieve full functionality.

Therefore, there are two types of distributed-memory: local and remote. Local distributed-memory is defined as the memory a given process uses to store its segment of the distributed data, while remote distributed-memory is the memory reserved by all the remaining parallel processes for their segments of the distributed data. Every process in a parallel job is allowed to access or to modify any element in the distributed memory segment regardless of its physical location. The access to local distributed-memory is faster than access to remote distributed-memory.

The goal of this thesis is programming strategy of DDI. The cluster architecture (hardware) and the parallel libraries used in the implementation of DDI completely depend to the performance consequence for accessing distributed-memory (local or remote). To efficient parallel calculations on TAURAS we used SCore cluster system software to minimize the communication latency.

The PC cluster TAURAS is the distributed memory MIMD cluster and the GAMESS program generates two processes on each CPU a “compute process" and a "data server", because the MPI-1 standard unable to administer the global memory. There are two main types of MIMD clusters: distributed memory, where communication is done by message passing interface (MPI); and shared memory, where communication is achieved using common memory locations.

4.2.1. The Beowulf cluster performance

GAMESS compute processes on Beowulf cluster systems are implemented by use of communication groups in MPI-1 messaging libraries as shown in Figure 4.2. This is local DDI implementation and the essential parallel libraries fully support all necessary communication operations. We use personal computers as Beowulf type cluster to evaluate speedup against the number of CPUs during trinitrotoluene molecule electronic structure determination in HF approximation. For testing calculations, only first nine nodes with CPUs faster than 700 MHz have been used.

Table 4.0. Dependence of CPU loads from the single processor performance.

Node	0	1	2	3	4	5	6	7	8
CPU MHz	733	800	1200	1200	800	733	733	733	800
CPU time, min	3132	2839	1537	1538	2848	3073	3073	3085	2845

Testing ab initio calculations of geometry optimization of trinitrotoluene molecule in Hartree-Fock approximation in 6-31G* basis were performed with different numbers of CPUs using program package GAMESS [2]. In Table 4.0, we give the load of different processors during that task.

The results of calculations show that 1.2 GHz AMD processor’s performance is about two times greater than 733 MHz Pentium III, and the ratio in MHz is only 1.6. GAMESS computer code evaluates all computers performance not only processor. Evaluating peak performance of clusters the AMD processors performance (nodes 2, 3) are usually evaluated as their MHz multiplied by 2 in order to achieve peak performance comparable with Intel CPUs. It is clear that CPU’s load will be equal and cluster’s performance is greater when nodes characteristics are the same. During our test (Table 4.1) AMD processors where idle about half the time.
Another is a LINPACK test of solving of linear equations system [63].

Table 4.1. Dependence of performance of different size clusters from the order of solved linear equations systems (in Gflops) on cluster TAURAS and others [63].

Linear equation system order	**8 nodes (LMA) 100 Mbs Network- 1 NIC per node**	**8 Duals Intel PIII 550 Mhz (512 Mb) – Myrinet**	**4 Duals (1.2 GHz AMD CPUs)**	**4 AMD Athlon K7 500 Mhz (256 Mb) - (2x) 100 Mbs Switched - 2 NICs per node (channel bonding)**	**4 Duals Intel PIII 733 MHz (512 Mb) – Ethernet**
2000	0.62	1.76	1.51	1.28	0.79
5000	1.49	2.32	2.54	1.73	1.03
8000	1.87	2.51	3.01	1.89	1.11
10000	1.98	2.58	3.13	1.95	1.06
15000	1.91	2.72			

As shown in table 4.1 the order of linear equations system was changed from 2000 to 15000. 512 MB of RAM per node was not enough to solve system of 20000 equations without memory swapping. When linear equations orders greater than 8000 the reduction of the performance (column 2) may be explained by slower network performance. On other hand node, performance plays the same role (columns 4 and 6)

Cluster speed up is important to know to analyze parallel calculations efficiency on PC cluster. The scientific computing: performance=1/time:

$$speedup = \frac{time(1\,processor)}{time(n\,processors)}.$$

The formula above means that the speedup equals the time of the task with one CPU divided by the time necessary to solve the same task with n CPUs. The task solved on PC cluster was single point trinitrotoluene molecule electronic structure determination in HF approximation using 6-31G* basis set, the results are shown in Figure 4.3.

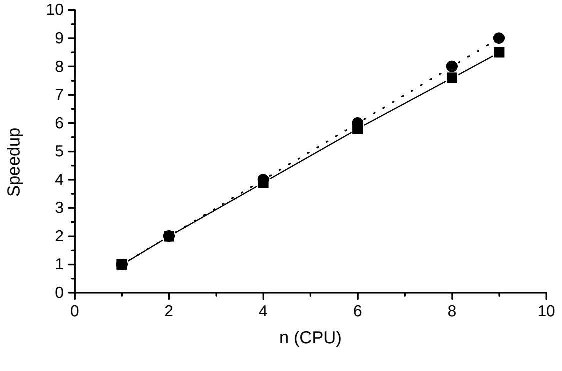


Figure 4.3. Speedup against the number of CPU's (n) during trinitrotoluene molecule electronic structure determination in HF approximation (solid line, solid squares; 6-31G basis). Dashed line (solid circles) is pure linear dependence between n and speedup.*

The dependence is almost linear like in similar clusters [8]. Of course, strait physical expansion of the cluster slows down the performance of the network communications speed.

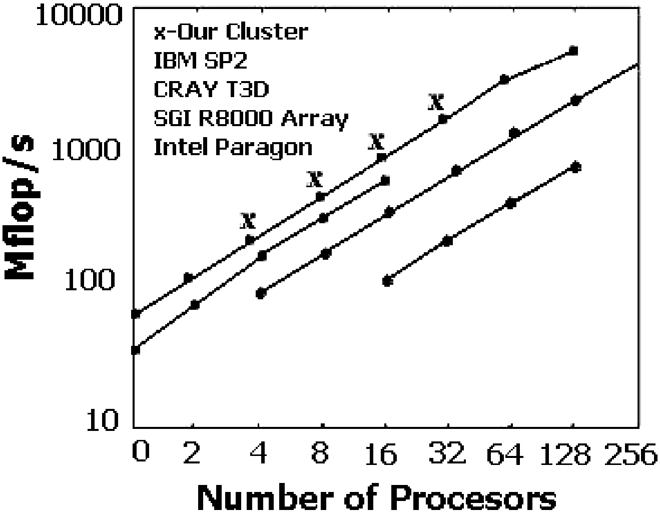


Figure 4.4. High performance LINPACK benchmark, solving of linear equation system n = 25 000. Results for other supercomputers are from [63].

The performance of cluster TAURAS compared with other supercomputers was investigated in high performance LINPACK benchmark calculations by solving the system of linear equations of order 25 000 (Figure 4.4). This benchmark measures the floating-point rate of execution for solving a linear system of equations of order n.
The operation count for LINPACK test is one order higher than the number of data involved in communication. This test is useful for testing the peak performance rate of the parallel system.

Total CPU time necessary for solving of the task on cluster TAURAS was 2080.77 sec. and real performance of the cluster achieved – 5,007 Gflops.
The can see that performance of our cluster is higher than IBM RS/6000 SP2 performance (Figure 4.4). The price of the cluster is tenths of times lower than the price of supercomputer IBM RS/6000 SP2.

So the usage of local hard drives and interchange of information between of the nodes was highly reduced.

Calculations with IBM RS/6000 SP2 supercomputer with 4 nodes and 512 MB RAM show that analogous tasks in LMA cluster are solved about 25 times faster. The wall time may be even hundreds of times smaller, because our cluster is used as "personal" computer and SP2 are multi-user supercomputer.

4.2.2. The SCore Cluster System Performance

The GAMESS DDI data server model was well used on Beowulf architecture clusters constructed from single CPU PCs, and the pairs of compute and data server processes could be naturally constrained to run on each PC node [8]. DDI supports distributed memory programming, providing between GAMESS and the underlying communication libraries and hardware.

In previous section 3.21 the research work shows, that strait physical expansion of the cluster slows down the performance of the network communications speed. It was the reasons for cluster TAURAS to choose the SCore cluster system software with a high performance communication facility called PM (Figure 4.5).

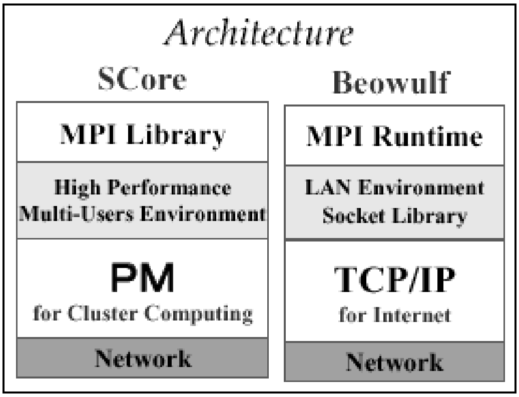


Figure 4.5. The main differences of SCore and Beowulf architectures.

SCore cluster system software for parallel calculations used own communication facility, that is no local implementation of GAMESS computer code. We have recompiled GAMESS using SCore cluster system implementation of MPI-1. Computations were performed in one electron Hartree-Fock (HF) approximation and with account of electron correlation.

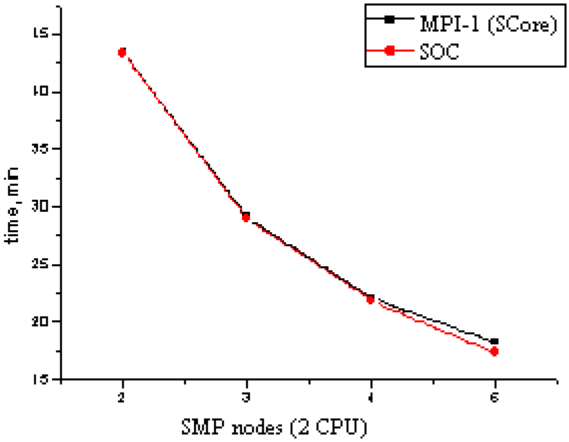


Figure 4.6. The computations time dependence on number of nodes on the cluster. The Trinitrotoluene molecule computations of single point in HF approximation are shown (black line – MPI-1 in SCore environment; red line by TCI/IP sockets in Beowulf environment).

The dependence of solution CPU time from the number of nodes in our cluster was achieved using sockets libraries like in Beowulf type cluster and with MPI-1 where we used SCore system. As an example we used earlier investigated trinitrotoluene molecule [10].

The AO basis 6-31G was used. Figure 4.6 shows almost the same speed up using sockets libraries and SCore. It can be explained taking into account the fact that in HF approximation network is not much loaded separate nodes just evaluate blocs of multicenter integrals.

GAMESS can compute wide range of quantum chemistry calculations of HF wave functions. We used two different implementations direct and conventional in our investigations. The direct method recomputed integrals for each iteration. This method requires enormous main memory and CPU resources, so achieving a very good CPU utilization, as have shown in Figure 4.7, where CPU utilization from the number of nodes in cluster TAURAS was achieved using sockets libraries like in Beowulf type cluster and with MPI-1 on SCore cluster system.

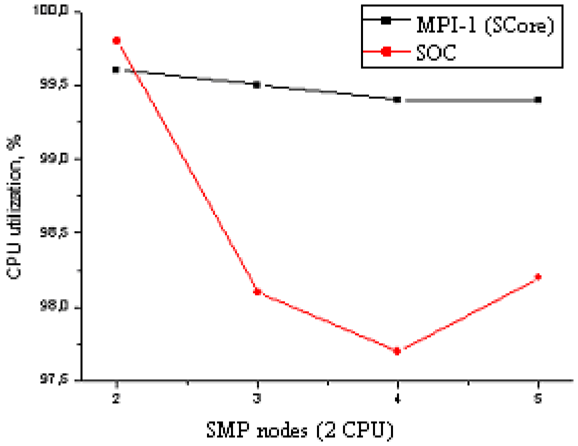


Figure 4.7. The TAURAS cluster CPU utilization dependence by computations on number of nodes. Trinitrotoluene molecule computations of single point in HF approximation are shown (black line – MPI-1 in SCore environment; red line by TCI/IP sockets in Beowulf environment).

We used a second order perturbation theory (MP2) to assess performance on cluster TAURAS. The computational cost of this type of calculation, as implemented in GAMESS, scales as O(N5), where N is the number of atomic basis functions. The memory requirements for the distributed arrays scale as O (N4). The MP2 method is widely used in quantum chemistry to provide accurate information on the energetic, kinetics and infrared spectra of molecules.

The distributed-memory MP2 gradient [67, 68] was chosen as a benchmark because the algorithm performs a significant number of local and remote distributed data operations, thus stressing the underlying DDI implementation.

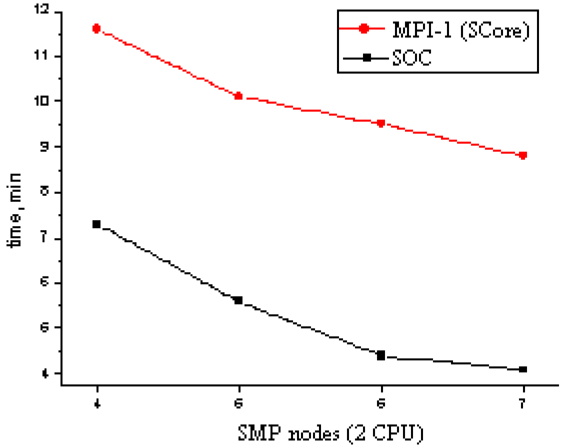


Figure 4.8. The computations time dependence on number of nodes on the cluster. The Trinitrotoluene molecule computations of single point in HF+MP2 approximation are shown (black line – MPI-1 in SCore environment; red line by TCI/IP sockets in Beowulf environment).

In the case of HF+MP2 all many electron integrals should be transformed from AO to MO basis. In that case, we expect more load of the network and we expect better results from SCore computations. Figure 4.8 shows that in that case sockets library preferable against SCore in speedup evaluation and even in absolute times evaluations.

It can be explain by the fact that in parallel GAMESS runs two processes, which are generated, for each computer in the cluster. It is all correct in sockets case: on each CPU we have one service process and one computing process.

As SCore regards all processes equally it happens that SCore runs more than one computational process on one CPU and this slows computation regardless of faster network communications in SCore cluster system, but with this clustering software we had the possibility to expand the communication productivity with network Trunking technology.

4.3. Solution of Realization of Parallel Calculations

Efficiency and well adapted calculations for express jobs on PC cluster TAURAS was the main task of these investigations. The general goal of the parallel realization development was to learn to use the SCore cluster system on TAURAS for implementation of GAMESS computer code and to solve efficiency problems in which lack of computing speed or power are the major bottlenecks.

4.3.1. Compilers and Cluster Productivity

Cluster TAURAS created by connecting PC's via a commodity high-speed network system using open source clustering software. Such type system clusters have several processors working together, but executing potentially different programs to solve the problem.

If the processors are to cooperate in executing a programming task, then they must be able to communicate with each other. The communications depend to some factors and are very important for parallel calculations one of them compiler used.The program compiled with free f2c+gcc compiler in free Linux operating system runs about 9-10 times slower than the same program compiled with effective commercial translators in commercial Windows NT system. It means that computational productivity highly depends on operating system and compilers used.

Table 4.2. NPB LU class A, B and C results get with free and commercial PGI translators.

LU class	**CPU 1x2 Time/sec**		**CPU 2x2 Time/sec**		**CPU 2x4 Time/sec**		**CPU 2x8 Time/sec**	
	GNU	**PGI**	**GNU**	**PGI**	**GNU**	**PGI**	**GNU**	**PGI**
A (64x64x64)	989	834	507	440	258	221	113	100
B(102x102x102)	3726	3624	2054	1922	1098	979	588	533
C(164x164x164)	-	-	8622	7697	4236	3950	2851	2214

The cluster TAURAS is made on top of open source of Linux Red Heat 7.3 OS and SCore cluster system. Typically, SCore system built to use C, C++ and FORTRAN77 GNU compilers: mpicc, mpic++ and mpif77, and as backend compilers: gcc, g++, g77. On the other hand SCore can be re-installed to use optional commercial compiler: Intel, Fujitsu, Portland Group, Compaq or Absoft ProFortran.

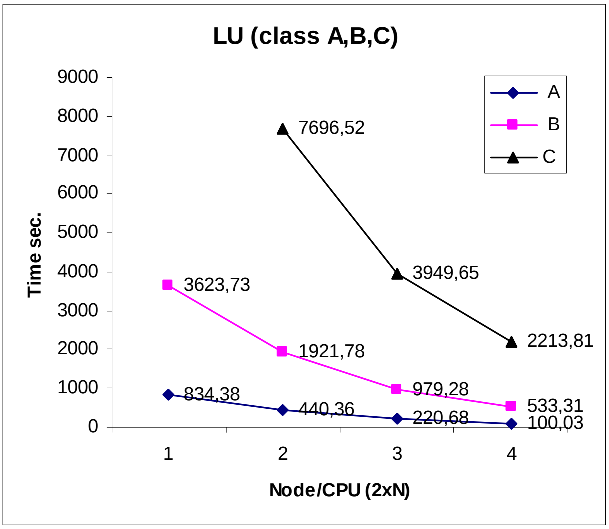


Figure 4.9. NAS NPB calculation results for solving LU tests of class A, B and C on Cluster TAURAS with a high performance parallel compiler of Portland Group (PGI).

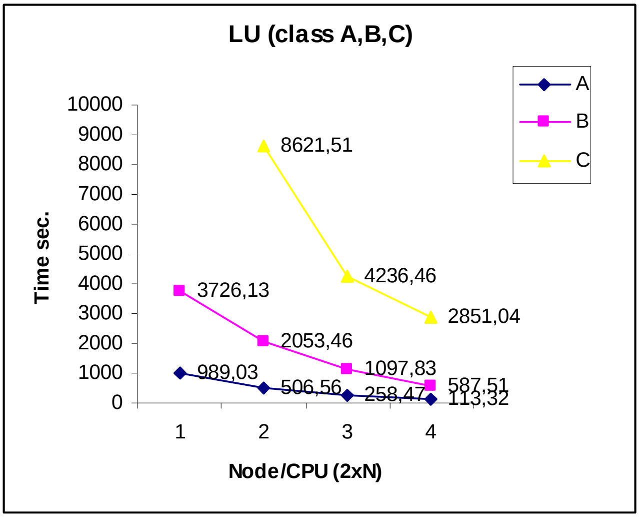


Figure 4.10. NAS NPB calculation results of solving LU tests of class A, B and C on Cluster TAURAS with free compiler (GNU).

The high performance parallel compiler of Portland Group (PGI) was fixed. So the TAURAS performance was evaluated with PGI and with free GNU compilers. The first objective is to determine which one performs best on our cluster.

To do this was performed the NAS NPB LU tests class A, B and C size problems using both compilers, and then calculate an average performance difference. Get results on cluster TAURAS are shown below by diagrams in Figures 4.9-4.10 and in Table 4.2.

As we can see in diagram, solving the NAS NPB LU test of class A, B and C when is used the commercial Portland Group compiler, calculation shows better results than was used the free GNU compiler of Linux OS. The Table 4.2 also confirms that the code produced by commercial PGI compiler is greater about 1,3 time to the code produced by the free GNU compiler. The results by PGI translator are better.

The other bottleneck of clusters is network they used. As usually, commodity network hardware is not designed for parallel processing. Typically, latency is very high and bandwidth relatively low compared to symmetric multi-processors (SMP) and attached processors. For example, SMP latency is generally no more than a few microseconds, but is commonly hundreds or thousands of microseconds for a cluster.

4.3.2. Network Software Interface

Network interconnects may be the one component in a high-performance cluster configuration that most significantly affects the system's ability to achieve required performance levels. High-performance interconnects can also become the most expensive part of the cluster configuration [52]. This makes it important to understand the communications requirements of the applications.

We decided to use Fast Ethernet Network, which is a commodity hardware product [53]. Network performance plays a very important role in parallel computing and network characteristics affect how we program for higher performance.

Using Fast Ethernet Network and PM, we achieved in user-to-user, own-way latency, of 7.2 microseconds and a bandwidth of 117.6 MByte/s on PCs. That latency and the bandwidth performance are far better to that of commodity Ethernet.

4.3.3. Network Trunking on PC Cluster

As was wrote in chapter 2 in order to achieve the possibility of high communication bandwidth by combining multiple NIC's into one big channel the SCore cluster system software propose the network Trunking technique. This technique was used on cluster TAURAS and the communication performance of this technique was evaluated.

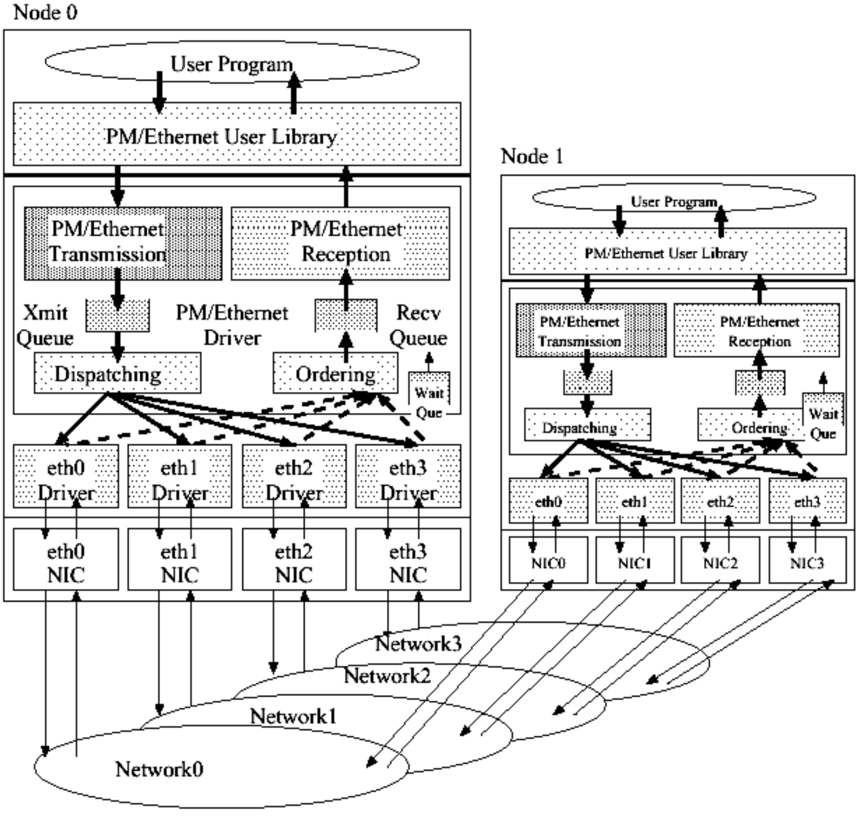


Figure 4.11. The communication by the network Trunking architecture.

The network Trunking facility realized using the PM/Ethernet mechanism on the Linux OS. PM/Ethernet driver: the network Trunking part is separated from the other parts, such as the sending and receiving processes, which are not related to the network Trunking part to minimize the need for additional code for the network Trunking facility.

Figure 4.11 shows the internal configuration of the Network Trunking facility. The network Trunking facility is realized using a part of the lower portion of dispatching and ordering from the PM/Ethernet driver described in Figure 4.11, and these parts control multiple Ethernet device drivers.

Dispatching: This part has a table consisting of the MAC address of every node used by the NIC's (eth0, eth1, eth2, eth3). After having decided on a transfer NIC according to the message from the sending queue, the Dispatching part replaces the Ethernet header, and calls the sending function of the Ethernet device driver.

For communication among many nodes, the round-robin method was adopted in order to utilize all of the NIC's equally.

Ordering: sequence number manages ordering of message. A message is stored in the reception queue when the sequence number matches during reception and inserted in the internal wait queue when the sequence number does not match.

The ordering part always checks whether the sequence number matches or not during reception of a message. When the sequence number of a message matches, the message is moved to the receive queue from the wait queue.

When the message has already been received in the wait queue, the message is discarded. Normal device driver structure (net device structure) is used in controlling multiple Ethernet device drivers in order to maintain the independence of existing Ethernet device drivers.

The network Trunking facility is implemented without any modification to existing Ethernet device drivers. So, there is no need to use the same NIC hardware.

The main negative aspect of cluster TAURAS is a reduced communication capability between processors. Therefore, it is important to test the performance of the PC clusters to define computational and communication powers [32, 33].

4.3.4. Evaluation of network Trunking

The NPB tests show effect of higher PM level communication performance of the Network Trunking technique on cluster TAURAS.

Table 4.3. Evaluation environment for network Trunking.

Hardware	Pentium III 733MHz (2xCPU) AMD 1200MHz (2xCPU)
NIC	Intel EEP-RO100 3Com 3C905B
Host OS	Red Hat 7.3 Linux
Device Driver	
for EEP-RO100	Eepro100c:v1.08 Donald Becker
for 3Com 3C905B	3c59x.c:v0.99H Donald Becker

The specification of evaluation environment shows the Table 4.3. For NPB tests was used the Intel EEP-RO100 and 3Com 3C905B network interface cards, because they are the NICs currently was used. The device driver for all Ethernet NICs was written by the same developer, and the control code of each device driver is similar to that of the others, except for the initialization of the device.

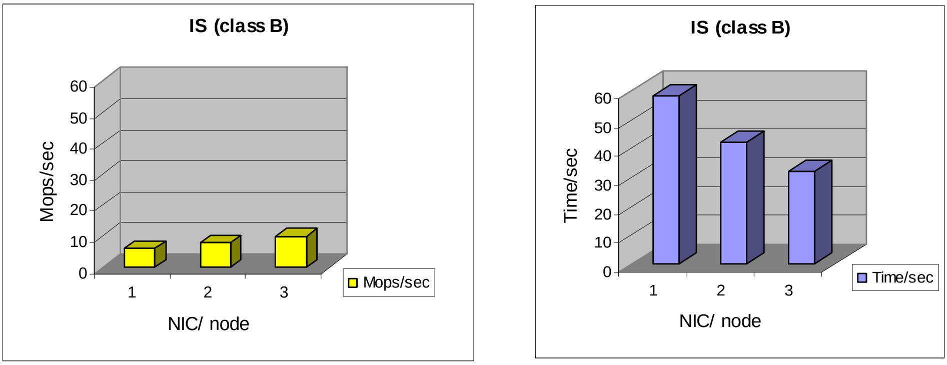
Figures 4.12, 4.13 and 4.14 show the PM level communication bandwidth achieved by changing the number of NICs for the EEPRO100 and 3C905B.

NPB Benchmarks on PC Cluster

In this work, was used NPB 2.3 [8]. NPB 2:3 is intended to be run, with little or no tuning, to approximate the performance a typical user can expect to obtain for a portable parallel program. The benchmarks, which are resulting from computational fluid dynamics (CFD), consist of five parallel kernels (EP, MG, CG, FT, and IS) and three simulated applications (LU, BT, and SP). The following is a brief overview of each of the benchmarks, and a detailed description can be found in [8, 62].

Integer Sort (IS) performs an integer sorting operation that is important in particle method codes. This benchmark tests both integer computation speed and communication performance. This problem is unique in that floating-point arithmetic is not involved. Significant data communication, however, is required.

Figure 4.12 displays the relative performance reported by each benchmark normalized with respect to the performance obtained with that benchmark using SCore PM driver. NPB reports the performance in millions of operations per second (Figure 4.12. (a)). For each benchmark in Figure 4.12, the result with SCore PM driver is set as base and relative performance using other (SCore PM Trunking) is plotted. As can be seen, the best performance with IS was obtained with the SCore PM Trunking driver. The SCore PM drivers yielded 55% of the performance obtained using SCore PM Trunking 3xNIC.



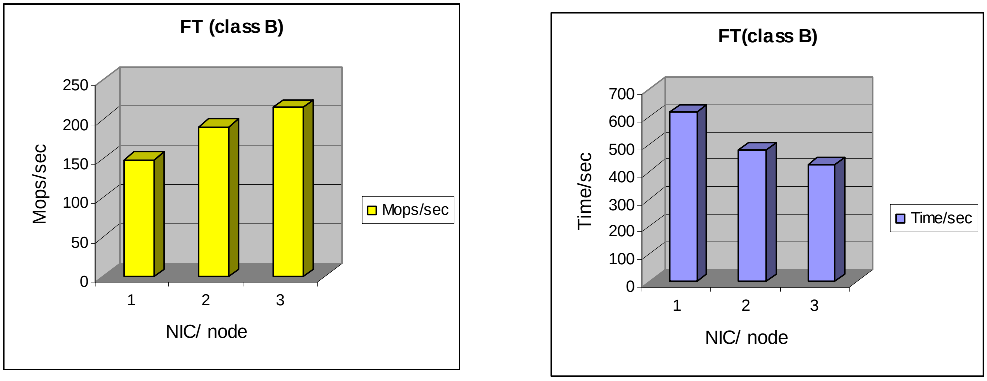
(a) (b)

Figure.4.12. The influence of Trunking technology: (a) for reached performance; (b) for computation time.

At each iteration about Mbytes messages are exchanged by collective all to all communication functions. Because the amount of computation is rather small the performance of collective operations dominates the performance of IS more and more as the number of processes is increased. Figure 13 (b) shows the execution time of IS with the different number of NIC's. The problem size is 20 ^{20}and the number of iterations is 10. The same test was used for each benchmark and was reached the 1,8 time speedup result.

Fast Fourier Transform (FT) benchmark solves a partial differential equation (PDE) using forward and inverse FFTs. 3D FFTs (512 x 256 x 256 grid size) are a key part of a number of computational fluid dynamics (CFD) applications and require considerable communications for operations such as array transposition. FT requires intensive float-point operations. It also is a good test of long-distance communication performance.

With FT, using the SCore PM Trunking 3xNIC driver resulted in 35% better performance than using the SCore PM driver (Figure 4.13. (a)).



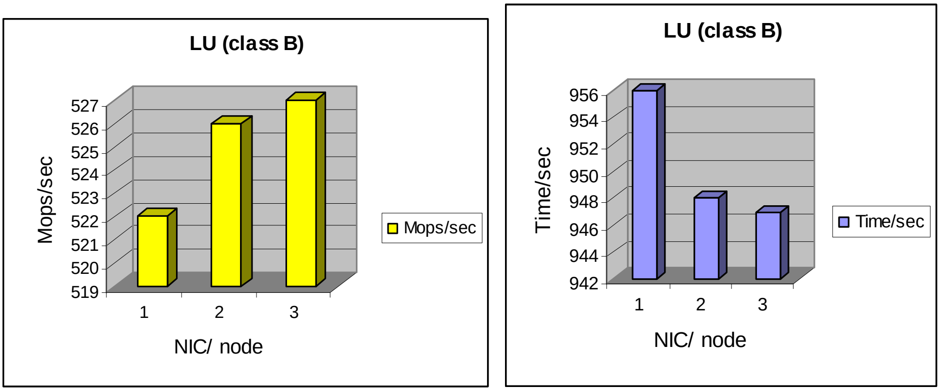
(a) (b)

Figure 4.13. The influence of Trunking technology: (a) for reached performance; (b) for computation time.

This result may seem surprising at first glance if we only look at the point-to-point latency and throughput tests with these drivers. However, it is quite expected given the fact that FT is a highly computation intensive benchmark. Using the SCore PM Trunking for communication frees up CPU re-sources that can be utilized to achieve better performance.

Figure 4.13.(b) shows the execution time of FT with the different number of NIC's. The problem size is 512 x 256^2 and the number of iterations is 20. The same test was used for each benchmark and was reached the 1,8 time speedup result.

The Lower-Upper (LU) diagonal benchmark employs a symmetric successive over-relaxation (SSOR) numerical scheme to solve a regular sparse, block 5x5 lower and upper triangular matrix system. The LU decomposition (Figure 4.14) benchmark solves a finite difference discretization of the 3-D compressible Navier-Stokes equations through a block-lower-triangular block-upper-triangular approximate factorization of the original difference scheme. The LU factored form is cast as a relaxation, and is solved by the symmetric successive over-relaxation (SSOR) numerical scheme. The grid size of class B is (102 x 102 x 102). Moreover, the communication is fine-grained; therefore, LU gives a good indication of the cluster interconnect latency. In this case however, for similar reasons to FT, SCore PM Trunking provided better results and exhibited lower latency.



(a) (b)

Figure 4.14. The influence of Trunking technology: (a) for reached performance; (b) for computation time.

With LU, using the SCore PM Trunking 3xNIC resulted in 65% better performance than using the SCore PM driver. Communication of partition boundary data occurs after completion of computation on all diagonals that contact an adjacent partition. This constitutes a diagonal pipelining method. It results in a relatively large number of small point –to- point communication operations of 5 words each (40 byte).

As shown in Figures 4.12-4.14 the SCore software Trunking technology helps to reach good performance.

HPL benchmark

HPL solves a system of linear equations of order n:

$$\mathbf{A}X = F, \quad \mathbf{A} \in \mathbb{R}^{n\times n}, \quad X, F \in \mathbb{R}^{n}$$

by computing LU factorization with row partial pivoting. This test stresses the floating-point performance of a parallel system. However, a system's communication bandwidth also significantly influences the overall LINPACK performance.

To understand better the impact of different network technology and SCore cluster system Trunking possibilities on general performance of cluster, was tested two SMP node (2xCPU) of cluster.

The HPL performance results (Figure 4.15) on a two SMP node (2xCPU) running with different technology of network interface cads (Fast Ethernet NIC's and Gigabit Ethernet NIC's) with different number of NIC's.

In each case we have conducted a series of HPL runs from small problem size to large. As it is known for LINPACK test, the larger size of the problem specified for execution, the better is the performance of the system in all cases. The results show that Trunking technology by running the HPL benchmark with 2xNIC Fast Ethernet reached the best result. Such a good performance can be explained by the fact that the HPL benchmark has a big influence of communication.

Again, the results show that the Gigabit Ethernet with 1x NIC_Gbytes/s can reach the better result than the 2x NIC_Gbytes/s. These bad results of HPL benchmark on Trunking technology on Gigabit Ethernet network can be explain. For Trunking ethnology it is need for each NIC in the network own switch for parallel computations, but the cost of this hardware was the problem to get one more. So the negative effects of this test on Gigabit Ethernet are possibility to solve and get the good results.

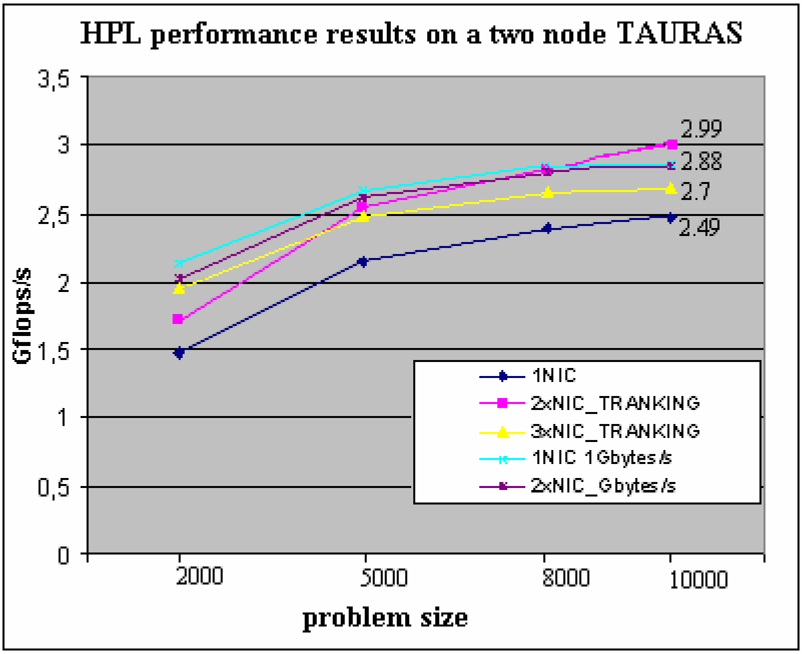


Figure 4.15. The HPL performance results on a two SMP node (2xCPU) running with different technology of network interface cads (Fast Ethernet NIC's and Gigabit Ethernet NIC's) with different number of NIC's.

Top performance with Fast Ethernet 2xNIC's with Trunking technology was around 1,2 time better than with 1xNIC. Reaching 94 % of theoretical peak performance of two nodes is estimated as a very good result for using Trunking technology.

4.3.5. MPI-2 on PC Cluster

The clusters have great potential, but that potential may be very difficult to achieve for most applications. In our way, we have GAMESS using for parallel execution, Distributed Data Interface over SHMEM, MPI-1, or TCP/IP sockets. To realize and achieve good performance of parallel calculations on cluster TAURAS we used the MPI-2 standard libraries and communication features of SCore system [65].

With an idea to realize the performance of cluster TAURAS we changed the GAMESS parallelization code in odder to have only one computation process on one CPU. Was implemented the MPI-2 standard libraries to administer the global memory, because the DDI don't relies such standard. These routines extended the communication mechanism, which will let to eliminate two processes for one CPU for GAMESS parallel code on clusters TAURAS.

MPI-2 communication libraries provides one-sided communication routines to support the Remote Memory Access (RMA) programming model similar functionality as use a powerful active message passing library such as SHMEM which exist on Cray T3E system. Since SHMEM library is only available on Cray T3E system we chose the MPI-2 extensions for our study to have more potential for portability to other systems. The use of distributed data requires the capability to interrupt computations on remote nodes for a moment to access their memory, and to guarantee exclusive access to that memory during the operations this influence the latency of traffic.

Most of our modifications are made to the distributed data interface (DDI) portion of the code. DDI is the message passing layer on which the application part of GAMESS is constructed, and is designed to provide a universally applicable model of processing very large arrays whose patches are distributed among CPU nodes. GAMESS developers set one of their goals as to make it run on any type of UNIX computer, from a cluster built from symmetric multi-processor (SMP) enclosures to high-end massively parallel computers. Actually almost all types of UNIX machines have complete access over every functionality of the previous (second) generation of DDI regardless of the type of the communication model: TCP/IP sockets, MPI, LAPI (of IBM), or SHMEM.

At the release of the latest (third) generation of DDI, a concept of dynamic grouping of processors was newly incorporated. Under the scheme, processors can be grouped into subsets working on completely independent quantum mechanical tasks. The hierarchical parallelization is extended DDI model embodies and is termed “generalized DDI (GDDI)”. Its implementation has, however, been excluded in isolated codes for SHMEM machines, namely, ddio3k.src and *ddishmem.src* , and consequently users of Cray X1/T3E, Compaq SC, and SGI Origin/Altix have never benefited from the significant improvement in DDI. We guess that this situation might not be for technical reasons, but being due to a limited number of contributors whose recent efforts have naturally focused on tuning to symmetric multi-processor (SMP) clusters, which are the most commonplace parallel computer system today.

There were the following reasons to modify the DDI code: the analyses of realized calculations on cluster TAURAS with SCore cluster system would need to use FMO; the improvements in DDI would be increased scalability based on use of one-sided communication routines such as those of SHMEM and MPI-2. In modifications, all SHMEM operations have been successfully merged into the standard source files.

The MPI-2 RMA mechanism in order overcomes any potential problems with the corresponding data structure on various processors residing at different memory locations. This is done by referencing a "window" that implies the base address, protection against out-of-bound accesses, and even address scaling. Efficient implementation is aided by the fact that RMA processing may be delayed until the next MPI_Win_fence. In summary, the RMA mechanism may be a strange cross between distributed shared memory and message passing. A very clean interface potentially generates very efficient communication. RMA is initiated with a collective library call where each process specifies an area of memory that is made accessible to remote processes. This shared memory buffer is used for the exchange of data. A call to a one-sided communication routine needs to be issued only by one process and does not require a matching call by sender or receiver respectively. MPI-2 provides point-to-point and barrier synchronization operations and it is the user's responsibility to ensure memory coherence. The MPI-2 extensions that we used in our study are:

- MPI_Win_create: A collective routine for setting up a shared memory buffer.
- MPI_Get, MPI_Put: Routines for transferring data to and from a shared memory buffer.
- MPI_Win_fence: A routine for performing collective synchronization.

MPI-2 extensions also include routines for point-to-point synchronization; however, they are not available on the hardware platforms that we used for our study.

4.4. Necessary Resources for Parallel Computations

Efficient use of GAMESS requires an understanding the critical issues. There is the difference between two types of memory (replicated MWORDS and distributed MEMDDI) in the GAMESS. The selected proper number of CPUs belonged to understanding each type of the computation scales degree.

Table 4.4. Necessary resources find out for computations on PC cluster TAURAS.

Basis set	**Number of basis functions**	**Approximation**	**Necessary RAM (MB)**	**Necessary Nodes with 2GB RAM**
TNT investigation				
6-311G (3d,1f,3p)	716	**HF + MP2**	7 320	**4**
6-311G (3d,1f,3p) +diffsp,+diffs	785	**HF + MP2**	8 776	**5**
6-311G (3d,1f,3p) +diffsp,+diffs	**785**	**HF**		
TZV (3d,1f,3p)	732	**HF + MP2**	7648	**5**
TZV (3d,1f,3p) +diffsp,+diffs	801	**HF + MP2**	9128	**6**
Mathematical Modeling calculations				
	903	**HF + MP2**	113 608	**66**
	1290	**HF + MP2**	492 496	**224**

Memory allocations and check jobs are important. However, closed and open shell MP2 calculations, MCSCF wave functions, and their analytic Hessian or MCQPDT energy correction do use distributed memory when run in parallel.

It is important to know how to obtain the correct value for MEMDDI, and how to compute how many CPUs are needed to do the run. Check runs (EXETYP=CHECK) need to run quickly, and the fastest turn around always comes on one CPU only. Runs which do not currently exploit MEMDDI distributed storage will formally allocate their MWORDS needs, and feel out their storage needs while skipping almost all of the real work. However, MEMDDI is typically a large amount of memory, and this is unlikely to be available on a single CPU. The solution is that the check job will not actually allocate the MEMDDI storage, instead it just remembers what you gave as input and checks to see if this will be adequate.

The check row $CONTRL „EXETYP=CHECK"ordered that will be cheked only recourses need for future calculations. In addition, this type checks the get data. It is very useful for the new investigations start, predicts the possible mistakes and impossible tasks of the PC cluster. The recourse needs straightly depends on used methods for calculations and sets of basis functions.

Therefore, the check row helps to plan the recourses for investigations. In this type the Table 4.4 was done.

The mathematical modeling calculations on PC cluster TAURAS are impossible at this time. This depends on cluster recourses that are inadequate at this time.

4.5. Results of Calculations

To performe the PC cluster TAURAS performance was done the investigations of 2,4,6-trinitrotoluene molecule. The family of basis sets: 6-31G*(1d), 6-311G**(1d), 6-311G**(2d), 6-311G**(2d+1f), 6-311G**(3d+1f) and 6-311G**(3d+1f+3p) vas used.

Table 4.5. The calculations of trinitrotoluene molecule geometry optimization in SCore environment. Computations were performed in Hartree-Fock approximation. The cluster of 9 CPU's was used.

Basis sets	**Numb. of basis functions**	**Computational time**	
		min	**h**
6-31 G*(1d)	250	146.9	2.45
6-311 G**(1d)	319	247.4	4.12
6-311 G**(2d)	415	555.0	9.25
6-311 G**(2d+1f)	575	1511.7	25.20
6-311 G**(3d+1f)	671	5349.8	89.16
6-311 G**(3d+1f+3p)	716	6773.7	112.9

The investigations was done from 6-31 G* (with polarization d functions on second period elements; all in all 250 basis functions) to 6-311 G** (with two polarization d functions on second period elements). Additionally was used one f function and diffuse s and p functions on second period atoms and diffuse functions on hydrogen atoms (all in all 770 basis functions).

The sets of basis functions were used to get a full optimization of 2,4,6-trinitrotoluene molecule with a C_1 symmetry and to check the cluster possibilities. The time of investigations of 2,4,6-trinitrotoluene molecule are shown in the Table 4.5. The increasing of the execution time is very fast, from 146.9 min for 250 numbers of basis functions to 6773,7 min for 716 numbers of basis functions.

The claculations have done by PC cluster show that with the increasing number of basis functions incresing and calculation time, because the quality investigations needs a lot of computation resurses.

Achieved results show that even with such comparatively small, cluster TAURAS we can perform all necessary computations at Hartree-Fock level and some necessary computations at MCSCF. It can be stated that created PC's cluster allows solving of modern quantum chemical problems at necessary level.

4.6. Conclusions of this Chapter

This section seeks to show the strengths and weaknesses associated primarily with the PC cluster TAURAS architecture of the SCore system clusters. By analyzing the performance of the communication and computational aspects of the system and comparing the results with the known architectures clusters, these strengths and weaknesses are explored.

The existing communication facilities on a commodity network with a cluster system generally use the TCP/IP protocol, which cannot ensure the maximum communication performance of network hardware. SMP communication bandwidth is often more than 100 MBytes/second; although the fastest network hardware such as Gigabit Ethernet offers comparable speed, the most commonly used networks are between 10 and 1000 times slower.

The computing speed on cluster TAURAS in this research work was made efficient by used a high performance communication facility called PM, which provides a virtual network mechanism called PM channel [54]. The channel provides reliable data gram communication, instead of connection-oriented communication such as that of TCP/IP. It means that latency for the application performance is short and the time of calculations are minimized.

The MPI -1-PM library in the parallel Score environment allows us to efficiently realize calculations in the TAURAS cluster. This proposition is corroborated by the fact that when testing 25000 order XHPL, the results of the clusters were as follows: the general CPU time is 2080,77 sec. and the capacity achieved was 5,007 Gflops.

In this section was given an introduction to the GAMESS computer code including how to run it on different environments, by analyzing the performance of the communication and computational aspects of the system and comparing the results that shows the strengths and weaknesses of used environments. This study compared two type clusters Beowulf and SCore built on the same architecture.

The capabilities of GAMESS algorithms proposed have been realized in the transition from the standard capabilities by means of parallel calculations in the parallel Beowulf environment by TCP/IP sockets and MPI-1 to the new technology in the process of parallel calculations by MPI-1-MP in the parallel SCore environment. The well known structure of the GAMESS program start for parallel calculations in retained by means of DDI. At present the new technology has been tested in the Linux Red Hat

7.3 operational system, using gcc, g++, g77 and commercial Portland Group compilers and MPI realizations.

The performance of the SCore PM library was evaluated on a cluster connected with a Fast Ethernet network. By executing the NAS Parallel Benchmarks, it was proved that the SCore PM Trunking software technology with multiple NIC's is superior in communication performance to the SCore PM with alone NIC. To performance with Fast Ethernet 2xNIC's with Trunking technology was around 1,2 time better than with 1xNIC. Reaching 94 % of theoretical peak performance of two nodes is estimated as a very good result for using Trunking technology. These results give corroborative evidence to claim: the efficiency of communication mechanism depends on the clusters performance.

The networks are realized to allow for good throughput with low latency for efficient passing of varying message sizes. As more computing power is needed, more PCs may be added. SCore PM and Fast Ethernet are two types of technologies used for cluster TAURAS.

MPI-1-PM also makes efficient use of capabilities of commercial compilers. NPB tests testified an increase in efficiency by ~1,5 times, using a commercial PGI compiler.

Analysis of applications of the cluster TAURAS has shown that applying the SCore software, it is possible to achieve the set targets and good results. The technical facilities of the cluster were quite sufficient for the realization of *ab initio* selected quantum chemical parallel methods. Extension of the parallel calculations by using the Trunking technology efficiency corresponded to characteristically predictions.

MPI-2 library are used to maintain synchronized distributed information. Thus, multiple memory banks are physically distributed in the system, but through MPI-2 functions, the memory appears to be uniformly shared. This allows a good performance but issues of bandwidth and latency become more important as processors are added. DSM computers are based on a number of processors controlling their own memory space, but also allowing other processors to read and write to their memory. The GAMESS code capability has been increased on the basis of the MPI-2 library, by adapting a parallel GAMESS version, operating with one DDI process, to PC clusters, thus seeking a more efficient usage of cluster resources running Linux OS, making complex calculations. The GAMESS version of parallel shared memory was adapted for investigating explosive molecules.

Clusters of personal computers built by using the available resources and freely disseminated software are most popular at present. The MP library is used most frequently for parallel comparison in the clusters. It has been established that the existing technologies and means of parallel comparison, applied in parallel calculations, do not allow us to achieve desirable results.

5. RESULTS & DISCUSSIONS

5.1. Theoretical Characterization of Explosives

The analysis of explosives is of main significance in several analytical fields such as forensic identification of explosive residues in bombing investigations, detection of explosives hidden in airline baggage or mail, determination of purity assessment of explosives in storage, and environmental analysis of traces of explosives in seawater and industrial environments. Simultaneously, with the increase of analytical fields in which explosives are involved and the development of new types of explosives, the techniques used for the analysis of explosives have become more sophisticated.

Molecules having nitro groups are usually known as explosives. The thermal decomposition mechanisms and resultant products are important for any explosive, since they play a major role in predicting the feasibility of large-scale synthesis, the long-term stability for the purpose of storage and the sensitivity to various stimuli.

2,4,6-trinitrotoluene is better known by its initials TNT, has been available as an explosive since around 1870. It is the most widely used explosive because of its compatibility with other explosives. TNT is an important explosive, since it can very quickly change from a solid into hot expanding gases. 2, 4, 6-trinitrotoluene has 3 NO_2 groups (from nitric acid) to toluene and is explosive because it contains the elements carbon, oxygen and nitrogen, which means that when the material burns it produces highly stable substances (CO, CO_2 and N_2) with strong bonds, so releasing a great deal of energy. TNT manufacturing involves stepwise nitration of toluene in a three-stage batch process or continuous process producing mono-, di-, and finally trinitrotoluene, respectively.

Also we study the 2,4,6-trinitrophenol known as picric acid. Nitro phenols are a class of materials witch are classified as secondary explosives and is the one most frequently used in making ammunitions. Only few of them molecular structures were examined experimentally. Hence it is necessary to find the geometric and electronic structure of the materials to know their properties that help us to create tools for explosive detection and more safe harmless.

The basis of investigation of explosive molecular electronic structure and geometrical data is quantum mechanics, but in order to achieve quantitative results comparable with experimental we need supercomputer's power or parallel computers cluster. The way in which such tasks are solved nowadays is this. All calculations were made on the local area PC cluster TAURAS with Score cluster system software. Get results show that the cluster is good device for such tasks.

The number of NO_2 groups influences the electronic structure and vibration spectra of explosive molecule, so one of the goals of this thesis was the investigation of explosive molecules positional isomer spectral properties.

5.2. Quantum Chemical Investigations

Theoretical investigations of 2,4,6-trinitrotoluene and 2,4,6-trinitrophenol molecules were performed applying of *ab initio* quantum mechanical methods using the GAMESS computer code integrated under the PC cluster TAURAS [1]. Was used Hartree-Fock approximation with 6-31G* basis set (polarization *d* functions on second period elements; all in all 250 basis functions). Also 6-311 G** basis set with two polarization *d* functions on second period elements, and plus one *f* function and diffuse *s* and *p* functions on second period atoms, and additionally diffuse functions on hydrogen atoms (all in all 770 basis functions).

The basis functions used are necessary to achieve reliable results in such investigations. During multiconfigurational self consistent field (MCSCF) calculations performed for trinitrotoluene we left unfrozen 10 highest occupied MO's and 10 lowest unoccupied MO's from the Hartree-Fock Slater determinant. All configurations generated from unfrozen MO's with spin projection zero were included in the MCSCF procedure. In that case, we had more than 65000 configurations. All unfrozen MO's are π orbitals of the benzene ring and NO_2 groups.

5.2.1. 2,4,6-trinitrophenol

2,4,6-trinitrophenol is the molecule which have NO_2 groups and was investigated on cluster TAURAS in order to know more about the effect on the aromatic nitro compounds and in order to get the results comparable with experimental date or with results calculated on known supercomputers.

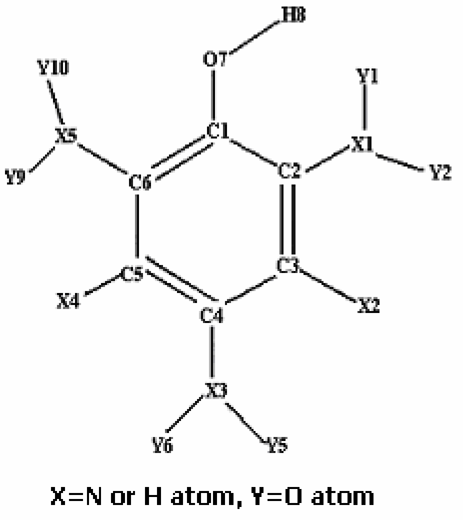


Figure 5.0. The molecular structure of 2,4,6-trinitrophenol. ***X*** *is the* ***N*** *atoms or* ***H*** *atoms,* ***Y*** *is the* ***O*** *atoms.*

Calculated bond lengths, bond angles, dihedral angles, dipole moment and energy by HF/6-31* method are presented below, in Table 5.0, Table 5.1 and Table 5.2. In Figure 5.0 are shown the molecular structure of 2,4,6-trinitrophenol molecule. The geometrical values of this molecule were optimized. All get results are compared with the experimental data and calculated results of P. C. Chen [77].

With the available experimental geometrical data of 2,4,6-trinitrophenol, a comparison between the theoretical calculations and the experimental values can be made. The C- C bonds are dominated by the resonance and inductive effects. The bond lengths calculated by the HF/6-31G* method are close to the experimental values.

Table 5.0 . 2,4,6-trinitrophenol molecule equilibrium bond lengths (Å) calculated in the HF approximation in the 6-31G basis set on cluster TAURAS and by P. C. Chen, also experimental values[77;97].*

Bound	HF/6-31G* (normal) [77]	HF/6-31G* (transition state) [77]	Exp. [97]	HF/6-31G*
C1-C2	1.410	1.416	1.412	1.413
C2-C3	1.382	1.382	1.373	1.378
C3-C4	1.372	1.373	1.367	1.370
C4-C5	1.386	1.382	1.376	1.379
C5-C6	1.371	1.377	1.355	1.374
C1-C6	1.408	1.415	1.407	1.412
C1-O7	1.301	1.298	1.312	1.294
O7-H8	0.958	0.968	-	0.952
C2-X1	1.450	1.451	1.455	1.455
X1-Y1	1.206	1.206	1.227	1.200
X1-Y2	1.184	1.184	1.201	1.177
C3-X2	1.070	1.070	-	1.070
C4-X3	1.451	1.451	1.455	1.455
X3-Y5	1.192	1.192	1.214	1.186
X3-Y6	1.192	1.191	1.217	1.185
C5-X4	1.071	1.069	-	1.069
C6-X5	1.458	1.462	1.477	1.466
X5-Y9	1.195	1.196	1.183	1.189
X5-Y10	1.187	1.185	1.202	1.179
Y1-H8	1.789	1.761	1.680	1.750

Table 5.1. 2,4,6-trinitrophenol molecule equilibrium bond angles (°) calculated in the HF approximation in the 6-31G basis set on cluster TAURAS and by P. C. Chen, also experimental values[77;97].*

Bond angles	HF/6-31G* (normal) [77]	HF/6-31G* (transition state) [77]	Exp. [97]	HF/6-31G*
C1-C2-C3	122.1	122.5	123.2	122.6
C2-C3-C4	119.2	119.1	117.7	119.0
C3-C4-C5	121.1	121.0	122.1	122.1
C4-C5-C6	119.2	119.9	119.3	119.9
C1-C6-C5	122.5	122.7	122.4	121.7
C2-C1-C6	115.9	115.8	115.2	115.7
C2-C1-O7	125.0	123.0	124.5	123.5
C6-C1-O7	119.1	120.5	120.3	120.8
C1-O7-H8	110.8	110.6	-	110.3
C1-C2-X1	121.0	121.1	119.4	121.0
C2-X1-Y1	117.9	118.1	119.0	118.1
C2-X1-Y2	118.1	119.1	118.7	119.0
C2-C3-X2	120.0	120.1	-	120.1
C3-C4-X3	119.5	119.4	118.9	119.4
C4-X3-Y5	117.3	117.2	118.2	117.2
C4-X3-Y6	117.2	117.2	118.0	117.3
C4-C5-X4	120.8	120.6	-	120.6
C5-C6-X5	116.7	118.5	117.1	116.1
C6-X5-Y9	116.2	116.3	118.9	116.3
C6-X5-Y10	118.0	118.8	120.2	118.6

For the C- NO_2 bonds, the HF/6.3l G* calculation tends to give shorted bonds than the experimental data.

It is as expected that the C2- C1- C6 bond angle, which is attached by the hydroxyl group is smaller than 120° and the C- C- C bond which is attached by the nitro group is larger than 120°. Both on cluster TAURAS and P.C. Chen calculations on the C- C-C bond angles give very close values similar to the X-ray data. By contrast, deviation of the C- O- H bond angle is more sensitive than that of other bond angles.

The nitro group, linked to the hydroxyl group, of 2,4,6-trinitrophenol is found to be coplanar with the phenyl ring both experimentally and theoretically. For 2,4,6-trinitrophenol , the calculated twisted angles of the nitro group, which is attached to the opposite side of the hydroxyl group, are different from the experimental values. Similar results were also observed by P.C. Chen calculation [77].

Table 5.2. 2,4,6-trinitrophenol molecule dihedral angles (º), dipole moments (Debye), energies (hartree) calculated in the HF approximation in the 6-31G basis set on cluster TAURAS and by P. C. Chen, also experimental values[77;97].*

Dihedral angles	HF/6-31G* (normal) [77]	HF/6-31G* (transition state) [77]	Exp. [97]	HF/6-31G*
O7-C1- C6- C2	-176.5	180.0	-	180.0
H8-O7-C1-C2	0.7	0.0	-	0.0
X1-C2-C1-C3	0.3	180.0	-	180.0
Y1-X1-C2-C1	1.7	0.0	7.4	0.0
X2-C3-C2-C1	180.0	180.0	-	180.0
X3-C4-C3-C2	-179.7	180.0	-	180.0
Y5-X3-C4-C3	0.4	0.0	0.6	0.0
X4-C5-C4-C3	180.0	180.0	-	180.0
X5-C6-C5-C1	177.8	180.0	-	180.0
Y9-X5-C6-C5	34.4	0.0	17.2	0.0
Dipole moment	1.84	2.02	-	2.02
Energy	**-915.94982**	**-915.94861**	**-**	**-916.08196**

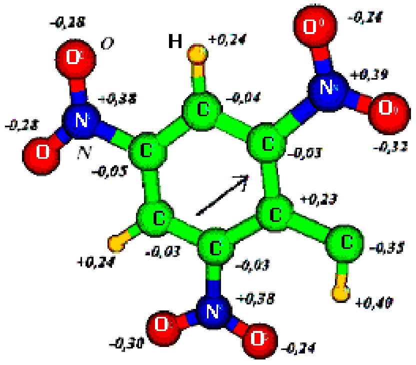


Figure 5.1. Charges of the atoms in the 2,4,6-trinitrophenol molecule (in a. u.) calculated in the Hartree-Fock approximation in the 6-31G basis set. Arrow shows approximate direction of the dipole moment calculated in the Hatrtee-Fock approximation (DHF = 2.021 D).*

The suspect is that this may be an effect of the bonding between the O atom of the nitro group and the O atom of the hydroxyl group. However, when closely examining this non-bond O-O distance, the calculated value is found to be quite similar to the experimental one. Figure 5.1 shows significant change of electronic charges of atoms and dipole momentum for 2,4,6-trinitrophenol molecule. The results in Figure 5.1 where achieved by the cluster [1].

In order to get the results comparable with experimental date, calculation were arrange the 2,4,6-trinitrophenol molecular structure to a planar molecule. Through geometrical optimization and second derivative examination, a negative vibrational frequency can be seen, which means that this planar molecule is a transition state structure. Tables 5.1-5.3 lists the transition state structure of 2,4,6-

trinitrophenol. There is a significant difference in all dihedral angles between the transition state structure and the "normal" structure. No matter which calculation method is used, the results show that hydrogen bounds are similar to those obtained from experimental data.

5.2.2. Optimization of TNT Molecular Structure

Trinitrotoluene ($C_7H_5N_3O_6$), can exist as different isomers such as: 2,3,4-, 2,4,5-, 2,4,6-and 2,5,6-TNT. The isomer that is used in the explosive industry is the symmetrical isomer 2,4,6-trinitrotoluene. Figure 5.2 show the 2,4,6-trinitrotoluene – the most common nitro aromatic compound used to charge landmines and some of its physical constants. TNT is very stable, no hygroscopic, relative insensitive to affect friction, shock, and electrostatic energy.

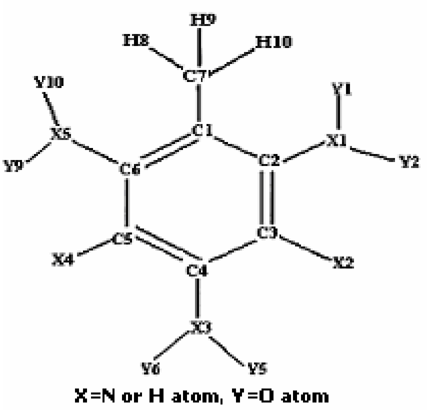
 H9 H8 H10 C7 Y10 Y1 X5 C1 X1 Y9 C6 C2 Y2 C5 C3 X4 C4 X2 X3 Y6 Y5 X=N or H atom, Y=O atom	Molecular weight: 227.13
	Density : 1.65 (g/mL)
	Melting point: 80.65 °C
	Boiling point: 210 °C (10 torr)
	Flash point: 240 °C
	Ignition point: 295 °C
	Energy of formation: -184.8 kJ/kg
	Enthalpy of formation:-261.5 kJ/kg

Figure 5.2. The molecular structure of 2,4,6- trinitrotoluene and characteristics.

In this investigations the geometry of the 2,4,6-trinitrotoluene was optimized in order to determine the lowest energy conformation and the optimized symmetry. The energy of a molecular system varies with changes in its structure and it is specified by its potential energy surface. A potential energy surface is a mathematical relationship linking molecular structure and the resultant energy. Geometry optimizations usually attempt to achieve minimum potential energy surface, in that way-predicting equilibrium structures of the molecular systems. To optimize the geometry we use the GAMESS computer code [2].

In this investigations the geometry of the 2,4,6-trinitrotoluene was optimized in order to determine the lowest energy conformation and the optimized symmetry. The energy of a molecular system varies with changes in its structure and it is specified by its potential energy surface. A potential energy surface is a mathematical relationship linking molecular structure and the resultant energy. Geometry optimizations usually attempt to achieve minimum potential energy surface, in that way-predicting equilibrium structures of the molecular systems. To optimize the geometry we use the GAMESS computer code [2].

Table 5.4 presents the optimized energies of 2,4,6-trinitrotoluene at different basis sets, in hartrees. Among the several basis sets used for these calculations, the 6-311G** basis set provides the lower energy conformation. The most stable geometry obtained for the 2,4,6-trinitrotoluene is show in Figure 5.4. This molecule has C_1 symmetry and consists of a phenyl ring with a methyl group and three nitro groups at positions 2, 4 and 6. Figure 5.3 shows significant change of electronic charges of atoms and dipole momentum for trinitrotoluene molecule. The results in Figure 5.3 where achieved by the cluster [1]. Figure 5.3 shows significant influence of electron correlation and necessity of MCSCF method for investigated molecule multiconfigurational properties.

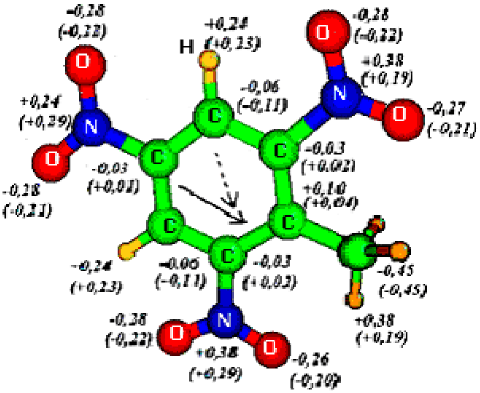


Figure.5 3. Charges of the atoms in the trinitrotoluene molecule (in a. u.) calculated in the Hartree-Fock approximation in the 6-31G basis set and by the MCSCF method (in parentheses). Arrows show approximate direction of the dipole moment calculated in the Hartree-Fock approximation (D_w = 1.759 D) and by the MCSCF method (D_{MCSCP} - 1.444 D, dashed arrow).*

The MOLDEN version 4.6 was used as visualization method. All quantum mechanical calculations were performed using the Hartree-Fock approximation. The family of basis sets: 6-31G*(1d), 6-311G**(1d), 6-311G**(2d), 6-311G**(2d+1f), 6-311G**(3d+1f) and 6-311G**(3d+1f+3p) vas used. The investigations was done from 6-31 G* (with polarization d functions on second period elements; all in all 250 basis functions) to 6-311 G** (with two polarization d functions on second period elements). Additionally was used one f function and diffuse s and p functions on second period atoms and diffuse functions on hydrogen atoms (all in all 770 basis functions). The sets of basis functions were used to get a full optimization of 2,4,6-trinitrotoluene molecule with a C_1 symmetry.

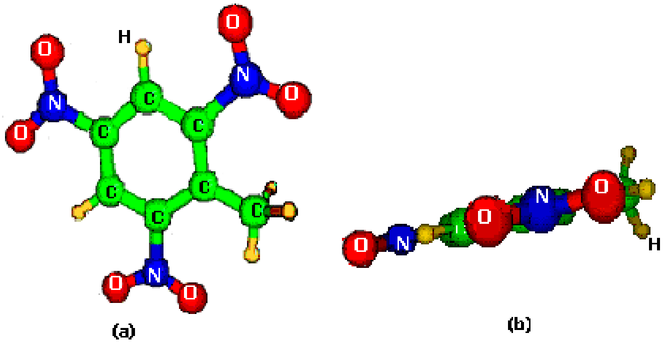


Figure 5.4. The view of optimized 2,4,6-trinitrotoluene molecular structure: (a) top and (b) side.

The basis functions used are necessary in order to achieve reliable results in such investigations During multiconfigurational selfconsistent field (MCSCF) calculations performed for trinitrotoluene was left unfrozen 10 highest occupied MO and 10 lowest unoccupied MO from HF Slater determinant [68, 71]. All configurations generated from unfrozen MOs with spin projection zero were included in MCSCF procedure. Was used more than 65000 configurations.

Table 5.3. Geometry optimization results of trinitrotoluene molecule: basis sets and number of basis functions, total energy and dipole moment. Computations were performed in Hartree-Fock approximation. The cluster of 9 CPU's was used.

Basis set	**Numb. of basis functions**	**$E_{tot.HF}$, a.u.**	**Dipole moment, D**	**Computational time, min (h)**
6-31 G* (1d)	250	-880.117	1.759	146.9 (2.45)
6-311 G** (1d)	319	-880.326	1.785	247.4 (4.12)
6-311 G** (2d)	415	-880.365	1.702	555.0 (9.25)
6-311 G** (2d+1f)	575	-880.406	1.713	1511.7 (25.20)
6-311 G** (3d+1f)	671	-880.426	1.727	5349.8 (89.16)
6-311 G** (3d+1f+3p)	716	-880.432	1.716	6773.7 (112.9)

Full geometry optimization was performed for trinitrotoluene molecule in Hartree-Fock and in MCSCF approximations. During investigation the C_1 symmetry point group of the molecule was assumed. Figure 5.4. show significant influence of electron correlation and necessity of MCSCF method for investigation of optical properties of investigated molecule. multiconfigurational properties.

Full geometry optimization was performed for the trinitrotoluene molecule in the Hartree-Fock and in MCSCF approximations. During investigation the C_1 symmetry point group of the molecule was assumed.

The results of trinitrotoluene geometry optimization are presented in Table 5.3: basis sets and number of basis functions, total energy and dipole moment. Even with such comparatively small cluster TAURAS we can perform all necessary computations at Hartree-Fock level and some necessary computations at MCSCF. It can be stated that created PC's cluster allows solving of modern quantum chemical problems at necessary level.

The calculated structural parameters (bond lengths, angles, and dihedral angles) of the 2,4,6-trinitrotoluene are listed in Tables 5.4, 5.5 and 5.6, respectively. The labeling of the atoms in these tables is consistent with the labeling with the labeling model shown in Figure 5.3. The parameters were obtained using the Hartree-Fock method increasing the basis sets from 6-31G*(1d) to 6-31G**(3d+1f). Those tables also provide the parameters obtained from Chen and coworkers using the Hartree-Fock with the basis sets 6-31G* level.

The parameters obtained with the HF/6-31G*(1d) basis set are compared with the values obtained by Chen and coworkers using HF/6-31G*. The HF method used by P. C. Chen et al. (1997) underestimates the bond lengths of the 2,4,6-trinitrotoluene optimized compare structure to the DFT level of theory. The C-C, C-N, and N-O average bond lengths calculated by HF method has a lengths differences of -0.0085, -0.0170 and -0.0367 Å, respectively. However no significant differences were observed between the bond angles and dihedral angles (Table 5.4 and 5.6).

Tables 5.4. B*ond lengths of the 2,4,6-trinitrotoluene*

Bond(Å)	**Basis sets**				
	6-31G* (1d)	**6-311 G** (1d)**	**6-311 G** (2d)**	**6-311 G** (2d+1f)**	**6-311G** (3d+1f)**
C1-C2	1.413	1.409	1.412	1.409	1.408
C2-C3	1.382	1.385	1.380	1.385	1.377
C3-C4	1.373	1.370	1.372	1.370	1.369
C4-C5	1.371	1.371	1.370	1.372	1.366
C5-C6	1.386	1.380	1.385	1.380	1.382
C1-C6	1.410	1.412	1.409	1.412	1.406
C1-C7	1.518	1.518	1.518	1.518	1.515
C7-H8	1.067	1.066	1.066	1.066	1.065
C7-H9	1.080	1.080	1.080	1.080	1.080

Table 5.4. (continued).

C7-H10	1.080	1.080	1.080	1.080	1.080
C2-X1	1.474	1.489	1.482	1.482	1.480
X1-Y1	1.190	1.183	1.183	1.183	1.180
X1-Y2	1.193	1.185	1.186	1.186	1.183
C3-X2	1.068	1.068	1.068	1.068	1.066
C4-X3	1.456	1.463	1.463	1.463	1.461
X3-Y5	1.191	1.184	1.185	1.185	1.182
X3-Y6	1.191	1.185	1.184	1.184	1.182
C5-X4	1.068	1.068	1.068	1.068	1.066
C6-X5	1.481	1.482	1.489	1.489	1.487
X5-Y9	1.192	1.186	1.185	1.185	1.182
X5-Y10	1.190	1.183	1.183	1.183	1.180
Y1-H8	3.68622	3.51721	3.68959	3.68959	3.68715
Y1-H9	2.37672	2.01836	2.38180	2.38180	2.37886
Y10-H10	2.37672	3.51721	2.38180	2.38180	2.37886
Energy	**-808.117**	**-808.326**	**-808.366**	**-808.407**	**-808.432**
Dipole moment	**1.702897**	**1.701599**	**1.701575**	**1.710988**	**1.716012**

Table 5.5.Calculated angles (º) of the 2,4,6-trinitrotoluene on cluster TAURAS.

Bond angles	**Basis sets**				
	6-31G* (1d)	**6-311 G** (1d)**	**6-311 G** (2d)**	**6-311 G** (2d+1f)**	**6-311G** (3d+1f)**
C1-C2-C3	123.8	123.1	123.8	123.8	123.8
C2-C3-C4	118.7	119.3	118.7	118.7	118.7
C3-C4-C5	121.1	121.1	121.1	121.1	121.1
C4-C5-C6	119.4	118.7	119.3	119.3	119.3
C1-C6-C5	123.0	123.8	123.1	123.1	123.1
C2-C1-C6	114.1	114.0	114.0	114.0	114.0
C2-C1-C7	121.3	124.6	121.4	121.4	121.4
C6-C1-C7	124.6	121.4	124.6	124.6	124.5

Table 5.5. (continued).

Bond angles	**Basis sets**				
	6-31G* (1d)	**6-311 G** (1d)**	**6-311 G** (2d)**	**6-311 G** (2d+1f)**	**6-311G** (3d+1f)**
C1-C7-H8	111.9	109.5	112.0	112.0	111.8
H8-C7-H9	109.6	106.9	109.5	109.5	109.7
H8-C7-H10	109.6	109.5	109.5	109.5	109.7
C1-C2-X1	122.8	124.4	122.9	122.9	122.9
C2-X1-Y1	119.2	119.9	119.0	119.0	119.1
C2-X1-Y2	116.7	116.2	116.6	116.6	116.6
C2-C3-X2	120.4	120.4	120.5	120.5	120.4
C3-C4-X3	119.5	119.4	119.5	119.5	119.5
C4-X3-Y5	117.2	117.1	117.1	117.0	117.1
C4-X3-Y6	117.2	117.0	117.1	117.1	117.2
C4-C5-X4	120.3	120.8	120.3	120.3	120.4
C5-C6-X5	112.6	113.4	112.6	112.6	112.5
C6-X5-Y9	116.3	116.6	116.2	116.2	116.2
C6-X5-Y10	120.1	119.0	119.9	119.9	120.0

Table 5.6. Calculated dihedral angles (°) of the 2,4,6-trinitrotoluene on cluster TAURAS.

Dihedral angles	**Basis sets**				
	6-31G* (1d)	**6-311 G** (1d)**	**6-31 G** (2d)**	**6-31 G** (2d+1f)**	**6-31G** (3d+1f)**
C2-C1-C7-C6	180.0	180.0	180.0	180.0	180.0
C2-C1-C7-H8	180.0	121.6	180.0	180.0	180.0
X1-C2-C1-C3	180.0	180.0	180.0	180.0	180.0
Y1-X1-C2-C1	0.0	0.0	0.0	0.0	0.0
X2-C3-C2-C4	180.0	180.0	180.0	180.0	180.0
X3-C4-C3-C5	180.0	180.0	180.0	180.0	180.0
Y5-X3-C4-C3	0.0	0.0	0.0	0.0	0.0
X4-C5-C4-C6	180.0	180.0	180.0	180.0	180.0
X5-C6-C5-C1	180.0	180.0	180.0	180.0	180.0
Y9-X5-C6-C5	0.0	0.0	0.0	0.0	0.0

The geometrical conformation of the trinitrotoluene explosive molecule helps to understand the thermal decomposition mechanisms.

5.3. Quantum chemical investigations of isomers

To check the possibilities specific of the cluster TAURAS we performed quantum mechanical investigation of the explosive molecules. As was explain above there was used the specific environment for calculations, so it is important to confirm that molecular modeling calculations can be doing correct on the cluster TAURAS and get results can be comparable with calculations on known supercomputers.

The calculations were performed at the HF level using a 6-31G* basis set (with polarization d functions on second period elements). These studies can provide valuable information in the field of aromatic nitro compounds. The obtained performances were compared with other published results [76].

5.3.1. TNP(2), TNP(3), TNP(4),TNP(5), TNP(6)

There are some positional nitrophenol isomers, which are: 2-dinitrophenol, 3-nitrophenol, 4-nitrophenol, 5-nitrophenol and 6-nitlrophenol. Among them, only few of their molecular structures were examined experimentally. Previous calculations indicated that the geometries of some energetic materials calculated by ab initio methods are similar to that of the experimental data. From this, the ab initio approach can be applied to examine other explosives.

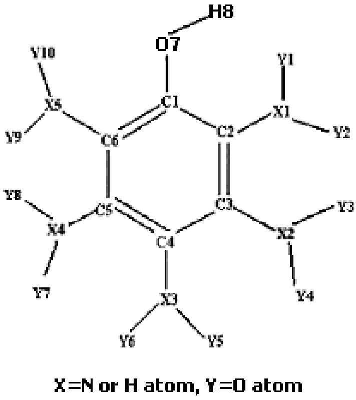


Figure 5.5. The molecular structures of nitrophenols.

Their geometries and molecular structures are presented in Figure 5.5 and Figure 5.6. The bound lengths, angles, and dihedral angles of nitrophenols was calculated by cluster. The calculated bound lengths presented in Table 5.7.

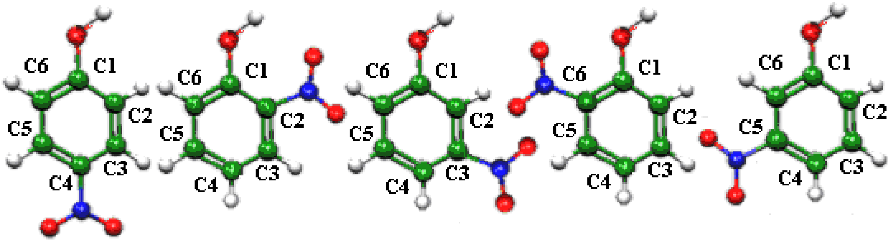


Figure 5.6. The NO$_2$ group positions on studied nitrophenol *molecules.*

Table 5.7.Calculated bound lengths (Å) of nitrophenols on cluster TAURAS.

Bond	**2-TNP**	**3-TNP**	**4-TNP**	**5-TNP**	**6-TNP**
C1-C2	1.380	1.380	1.380	1.380	1.380
C2-C3	1.380	1.380	1.380	1.380	1.380
C3-C4	1.380	1.380	1.380	1.380	1.380
C4-C5	1.380	1.380	1.380	1.380	1.380
C5-C6	1.380	1.380	1.380	1.380	1.380
C1-C6	1.380	1.380	1.380	1.380	1.380
C1-O7	1.290	1.290	1.290	1.290	1.290
O7-H8	1.090	1.090	1.090	1.090	1.090
C2-X1	1.470	1.090	1.090	1.090	1.090
X1-Y1	1.380	-	-	-	-
X1-Y2	1.380	-	-	-	-
C3-X2	1.090	1.470	1.090	1.090	1.090
X2-Y3	-	1.380	-	-	-
X2-Y4	-	1.380	-	-	-
C4-X3	1.090	1.090	1.380	1.090	1.090
X3-Y5	-	-	1.380	-	-
X3-Y6	-	-	1.380	-	-
C5-X4	1.090	1.090	1.090	1.470	1.090
X4-Y7	-	-	-	1.380	-
X4-Y8	-	-	-	1.380	-
C6-X5	1.090	1.090	1.090	1.090	1.470
X5-Y9	-	-	-	-	1.380
X5-Y10	-	-	-	-	1.380
Y1-H8	1.57838	-	-	-	-
Y10-H8	-	-	-	-	3.43668
Dipole moment	3.852150	3.762023	5.348594	6.018802	6.236732
Energy	-509.1057	-509.1006	-509.1042	-509.0998	-509.0908

The performed theoretical analysis of the molecular structures of nitrophenols let to evaluate the stability of these molecules. The lowest energy was obtained for 2-nitrophenol.

5.3.2. DNT(4,6), DNT(2,4), DNT(2,6), DNT(4,5), NT(4), NT(3), NT(2)

Dinitrotoluenes (DNT) are intermediates in the manufacture of TNT and are used as ingredients in mining explosives and in some smokeless powder. Was performed a theoretical analysis of the molecular structures of dinitrotoluene isomers, that includes 2,4-dinitrotoluene, 2,6-dinitrotoluene, 4,5-dinitrotoluene and 4,6-dinitrotoluene.

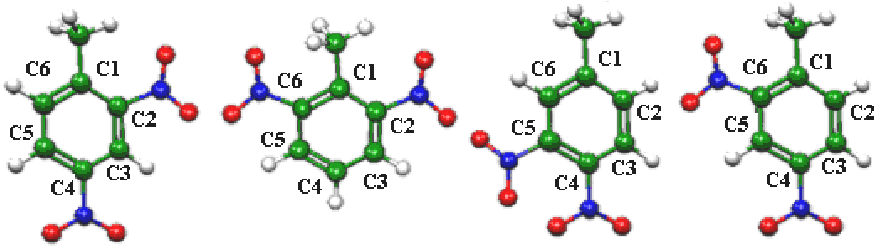


Figure 5.7. The NO_2 group positions on studied dinitrotoluene molecules.

The most stable molecule has the lowest HF energy. Energies of the dinitrotoluene isomers calculated by HF/6-31G* method coincides with results presented in [76]. According to our results of the investigation, the 2,4-dinitrotoluene is more stable than 4,5-dinitrotoluene.

Their geometries and molecular structures are presented in Figure 5.7 and Figure 5.8. The calculated bound lengths, angles, and dihedral angles of dinitrotoluenes are presented in Table 5.8, Table 5.9, Table 5.10. The geometrical structure (bond length, angles, and dihedral angles) had been compared and good coincidences were found. Both P. C. Chen and we obtain that the isomers deformations of the phenyl ring were affected by the methyl and nitro groups [76].

The most important results of these investigations indicate that more stable molecules are if the nitro and methyl groups of dinitrotoluene are separated from each other. These results confirm also predictions of P. C. Chen [76].

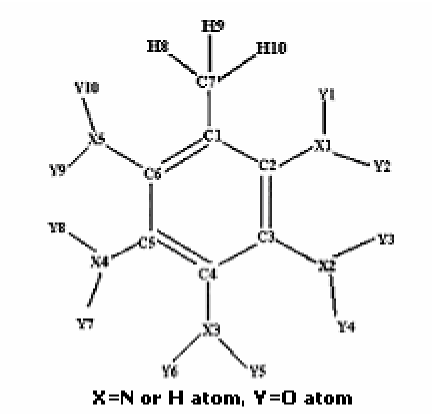


Figure 5.8. The molecular structures of dinitrotoluenes.

The geometrical conformation of the dinitrotoluene explosives molecules help to understand the thermal decomposition mechanisms.

Table 5.8.Calculated bound lengths (Å) of dinitrotoluenes on cluster TAURAS and by P. C. Chen [76].

Bond	2,4-DNT	2,4-DNT [76]	2,6-DNT	2,6-DNT [76]
C1-C2	1.3899	1.3982	1.3970	1.3964
C2-C3	1.3843	1.3831	1.3841	1.3838
C3-C4	1.3758	1.3758	1.3790	1.3793
C4-C5	1.3833	1.3827	1.3790	1.3793
C5-C6	1.3789	1.3793	1.3840	1.3839
C1-C6	1.3937	1.3948	1.3960	1.3963
C1-C7	1.5133	1.5135	1.5191	1.5189
C7-H8	1.0800	1.0810	1.0816	1.0815
C7-H9	1.0801	1.0811	1.0768	1.0770
C7-H10	1.0802	1.0812	1.0769	1.0770
C2-N10	1.4600	1.4611	1.4621	1.4624
N10-O11	1.1899	1.1932	1.2920	1.2921
N10-O12	1.1930	1.1926	1.1940	1.1937
C3-H7	1.0689	1.0691	1.0719	1.0716
C4-N13	1.4601	1.4556	1.0728	1.0730
N13-O14	1.1933	1.1921	-	-
N13-O15	1.1918	1.1933	-	-
C5-H8	1.0701	1.0712	1.0715	1.0716
C6-H9	1.0733	1.0735	1.4622	1.4624
N20-O21	-	-	1.1939	1.1937
N20-O22	-	-	1.1920	1.1921

Table 5.9.Calculated angles (°) of dinitrotoluenes on cluster TAURAS and by P. C. Chen [76].

Bond angles	2,4-DNT	2,4-DNT [76]	2,6-DNT	2,6-DNT [76]
C1-C2-C3	122.509	122.8	123.677	124.0
C2-C3-C4	118.614	118.4	120.212	119.8
C3-C4-C5	121.420	121.4	118.722	118.9
C4-C5-C6	118.706	118.8	120.760	119.8

Table 5.9. (continued).

Bond angles	2,4-DNT	2,4-DNT [76]	2,6-DNT	2,6-DNT [76]
C1-C6-C5	122.602	122.4	122.988	124.0
C2-C1-C6	116.149	116.2	113.641	113.6
C2-C1-C7	126.094	125.4	121.564	123.1
C6-C1-C7	117.757	118.4	124.795	123.3
C1-C7-H8	109.193	111.0	111.768	109.6
H8-C7-H9	107.127	107.4	109.512	108.5
H8-C7-H10	109.007	109.3	109.512	108.5
C1-C2-X1	122.093	121.7	122.607	120.9
C2-X1-Y1	118.098	118.0	119.366	118.0
C2-X1-Y2	117.413	117.3	116.864	117.0
C2-C3-X2	120.518	120.7	118.837	119.1
C3-X2-Y3	-	-	-	-
C3-X2-Y4	-	-	-	-
C3-C4-X3	119.029	119.1	120.692	120.6
C4-X3-Y5	117.541	117.6	-	-
C4-X3-Y6	117.185	117.3	-	-
C4-C5-X4	120.108	120.1	120.526	121.1
C5-X4-Y7	-	-	-	-
C5-X4-Y8	-	-	-	-
C5-C6-X5	118.824	118.9	112.859	115.1
C6-X5-Y9	-	-	116.409	117.0
C6-X5-Y10	-	-	120.258	118.0

Table 5.10.Calculated dihedral angles (°) of dinitrotoluenes on cluster TAURAS and by P. C. Chen [76].

Dihedral angles	2,4-DNT	2,4-DNT[76]	2,6-DNT	2,6-DNT[76]
C2-C1-C7-C6	180.0	180.8	180.0	175.1
C2-C1-C7-H8	59.7	48.4	58.5	92.7
X1-C2-C1-C3	180.0	180.6	180.0	178.0
Y1-X1-C2-C1	0.0	22.0	0.0	-35.0

Table 5.10. (Continued).

Dihedral angles	2,4-DNT	2,4-DNT[76]	2,6-DNT	2,6-DNT[76]
X2-C3-C2-C4	-	-	-	-
Y3-X2-C3-C2	-	-	-	-
X3-C4-C3-C5	180.0	180.3	-	-
Y5-X3-C4-C3	0.0	1.1	-	-
X4-C5-C4-C6	-	-	-	-
Y7-X4-C5-C4	-	-	-	-
X5-C6-C5-C1	-	-	180.0	178.1
Y9-X5-C6-C5	-	-	0.0	36.7

It has long been recognized that Hartree-Fock theory usually gives bond lengths, which are too short, and the description of multiple bonds tends to be problematic due to the neglect of electron correlation. The MP2 approach, conversely, frequently overestimate bond distances.

5.3.3. Estimated Energies of TNT Isomers

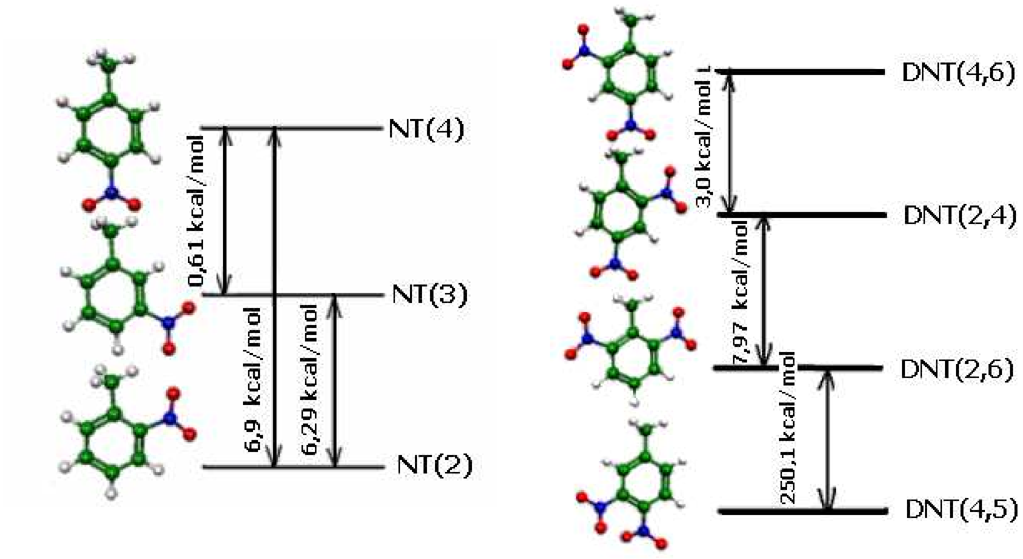


Figure 5.9 .The stability of: (a) NT molecule calculated; (b) of DNT molecule calculated. The results by HF approximation in the 6-311G(3d.1f.3p) basis set.

Table 5.11. The DNT energies calculated on cluster TAURAS and others from [76].

NO_2 positions	Energies calculated on cluster TAURAS	Energies calculated on the National centre for High-Performance Computing [76]
2,4	-676.90214	-676.67410
2,6	-676.88390	-676.66442
4,5	-676.87332	--------
4,6	-676.91083	--------

The energies of: 2,4-dinitrotoluene, 2,6-dinitrotoluene, 4,5-dinitrotoluene and 4,6-dinitrotoluene, conformers were calculated to determine which one is the most stable (Figure5.9). The get results indicate that 4,6-dinitrotoluene was the most stable isomer (Table 5.11). Also for NT investigations, the 4-NT was the most stable isomer.

Therefore, the calculations confirm the above theoretical prediction that the NO_2 groups influence the explosive molecule energy. The results allow us foresee that cluster TAURAS allows solving of modern quantum chemical problems at necessary Hartree-Fock level.

5.4. Spectra Computations on PC Cluster

The vibrational spectra are fingerprints for molecular species identification, so for detection explosives from spectral characteristics can be used Infrared (IR) and Raman spectroscopy methods. Raman spectroscopy is the study of a small portion of light scattered by a molecule, the light undergoing an exchange of energy with that molecule.

Infrared spectroscopy is a technique based on the vibrations of the atoms in molecules. An infrared spectrum is commonly obtained by passing infrared radiation through a sample and determining what fraction of the incident radiation is absorbed at a particular energy. The energy at which any peak in an absorption spectrum appears corresponds to the frequency of a vibration of a part of a sample molecule. IR absorption information is generally presented in the form of a spectrum with wavelength or wave number as the x-axis and absorption intensity as the y-axis.

Vibrational frequencies depend on interactions between atoms and groups of atoms. They tell us about intra- and intermolecular interactions. Vibrational modes vary with the nature of atoms in the bond and chemical environment. They give us information about chemical bonding and the chemical environment. Many bond types exhibit characteristic (“fingerprint”) vibrational frequencies that allow us to establish the presence of certain functional groups, or products in a catalytic reaction.

In order to obtain the spectroscopic signature of the explosive molecules we performed frequency calculations for the different basis sets. The theoretical frequencies for 2,4,6-TNT and 2,4,6-TNP was obtained. This work present the vibrational frequencies of the most stable geometry observed. These frequencies have to be real, or positive to consider that the geometry reach a minimum.

5.4.1. Vibrational Spectra Computations of 2,4,6-TNT

Trinitrotoluene molecule vibrational spectra computations were performed using computer code GAMESS [2]. For potential energy surface determination in Hartree-Fock (HF) approximation atomic orbital (AO) basis 6-31G* (total 250 basis functions) was used. Number of necessary to compute many center integrals was more than 10^9.

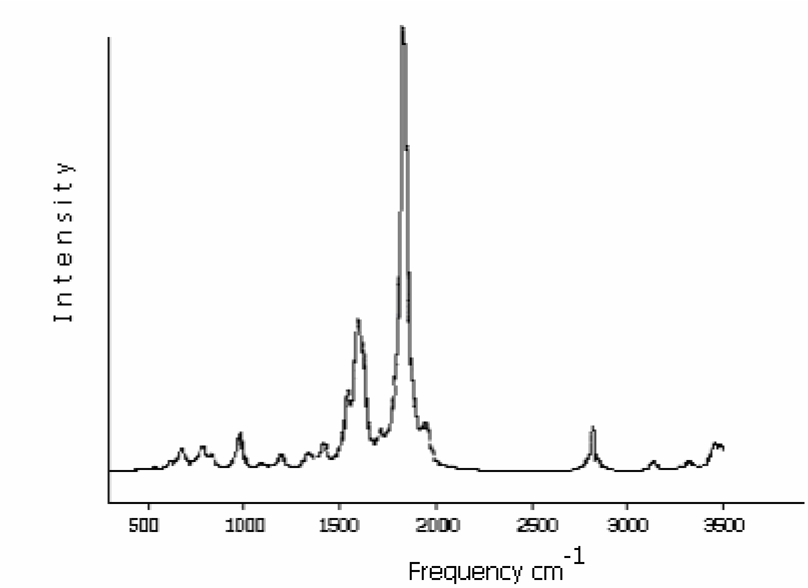


Figure.5.10. Infrared spectrum of neat 2,4,6-TNT in the range of 300–3500 cm^{-1}. IR vibrations calculated in the Hartree-Fock approximation in the 6-G311 (1d) basis.

The calculated by cluster TAURAS neat infrared spectrum corresponding to 2,4,6-TNT was visualized by MOLDEN 4.6 (Figure 5.10). The bands that allow identifying the neat TNT were obtained in the range of 350-3350 cm $^{-1}$. The vibrational signatures observed in the region of 3269-3496 cm $^{-1}$ can be assigned to asymmetric and symmetric C-H stretch vibrations, respectively, belong to the alkyl CH_3 group and the aromatic ring. Other peaks of high intensity are 1885 cm $^{-1}$ (NO_2 symmetric stretching vibration), in the region of 1623-1639cm $^{-1}$ (Valence angle deformations in NO_2, CH_3 groups and benzene ring), 1306cm $^{-1}$ (benzene ring in plane deformations).The assignments of the bands are summarized in the Table 5.13.

Harmonic vibrational spectra of molecules (vibrational frequencies, IR and Raman intensities) were calculated in the HF approximation. The frequencies with greatest IR and Raman intensities only are summarized in the Tables 5.12, 5.13, and 5.14 .

Though the bond length changes in the trinitrotoluene molecule were not large in HF and MCSCF methods, the change of vibrational frequencies and vibrational forms was significant (the changed molecular force field by the MSCSF method was the main cause). Most of the frequencies increased when the MCSCF method was used, all very low frequencies with out-of-benzene ring vibrational forms disappeared.

Table 5.12. Trinitrotoluene molecule harmonic vibrational frequencies, IR and Raman intensities and forms of vibrations calculated in the Hartree-Fock approximation in the 6-31 G basis. Only the most significant intensities are shown.*

Frequency cm -1	IR intensity, $D^2/(a.u.A^2)$	Raman intensity, A4/a.u.	Vibrational form
1306		16	Benzene ring in plane deformations
1623		37	Valence angle deformations in NO2, CH3 groups and benzene ring
1627	11	7	Valence angle deformations in NO2, CH3 groups and benzene ring
1639		40	Valence angle deformations in NO2, CH3 groups and benzene ring
1788		71	Benzene ring deformations
1885	14		NO2 group stretching vibrations
3269		137	CH3 valence angle deformations
3332		74	CH3 group stretching vibrations
3467		36	CH3 group stretching vibrations
3470		29	CH bond stretching vibrations
3496		29	CH group stretching vibrations

Table 5.13. Trinitrotoluene molecule harmonic vibrational frequencies, IR and Raman intensities and forms of vibrations calculated in the MCSCF approximation in the 6-31G basis. Only the most significant intensities are shown.*

Frequency cm -1	IR intensity, D2/(a.m.v.A2)	Raman intensity, A/a.m.v.	Vibrational form
849		145	Benzene ring deformations in the molecule plane
853		123	Benzene ring deformations in the molecule plane
1231		86	NO, CN bond stretching and benzene ring in the molecule plane deformations
1287		85	NO, C-CH3 bond stretching and benzene ring in the molecule plane deformations
1478		286	NO, CN bond stretching and benzene ring in the molecule plane deformations
1520		156	Benzene ring deformations
1561	1.4	47	NO, CN bond stretching and benzene ring in the molecule plane deformations

Table 5.13. (Continued).

Frequency cm -1	IR intensity, D2/(a.m.v.A2)	Raman intensity, A/a.m.v.	Vibrational form
1594	1.4	90	O, CN bond stretching and benzene ring in the molecule plane deformations
1708		158	Benzene ring in the molecule plane deformations
1789	2.7	82	NO bond stretching vibrations
1860	1.4	65	NO_2 group stretching vibrations
1903	3.3	81	NO_2 group stretching vibrations
2211		184	CH and CH_3 valence angle deformations
2889	1.5	233	CH and CH_3 valence angle deformations
2957		122	NO group stretching vibrations
2999		40	NO group stretching vibrations
3931		233	CH_3 group stretching vibrations
3504		25	CH bound stretching vibrations
3661		68	CH group stretching vibrations

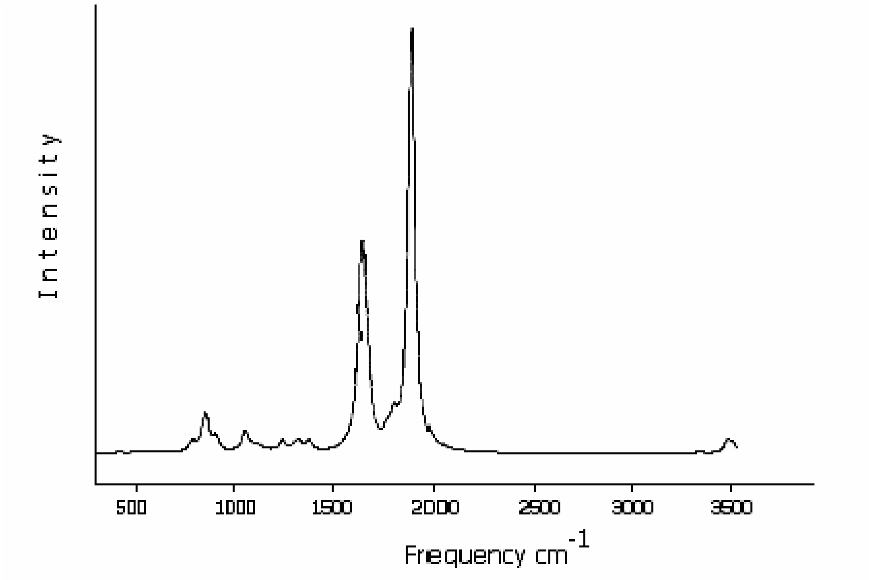


Figure.5.11. Infrared spectrum of neat 2,4,6-TNT in the range of 300–3500 cm^{-1}. IR vibrations calculated in the MCSCF approximation in the 6-31G basis.*

Very few frequencies retained the same form in HF and MCSCF approximations: 3496 cm^{-1} (HF) and 3661 cm^{-1} (MCSCF), 3470 cm^{-1} (HF) and 3504 cm^{-1} (MCSCF), 3467 cm^{-1} (HF) and 3931 cm^{-1} (MCSCF) —all CH bonds stretching vibrations; 3269 cm^{-1} (HF) and 2957 cm^{-1} (MCSCF), 3332 cm^{-1} (HF) and 2999 cm^{-1} (MCSCF) — CH_3 group CH bonds stretching vibrations; 1885 cm^{-1} (HF) and 1860 cm^{-1} (MCSCF) —NO_2 group stretching vibrations; 1788 cm^{-1} (HF) and 1520 cm^{-1} (MCSCF) — benzene ring deformations.

Zero point energy for trinitrotoluene changes significantly: from 32572 cm^{-1}/molecules in the HF approximation to 44090 cm^{-1}/molecules in the MCSCH

5.4.2. Vibrational Spectra Computations of 2,4,6-TNP

No empirical calculation of harmonic vibrational spectra in HF approximation took about 40 min of clusters CPU time. In Table 5.15 we present vibrational frequencies of trinitrotoluene molecule. Computation of points on potential surface (for VSCF) even at HF level was very computer time consuming. We could compute only 8 points along each normal mode and interaction between modes was neglected. It is quite rood approximation. It should be mentioned, that SCF calculation of vibrational spectra took very small amount of clusters facilities in comparison with potential energy surface calculation.

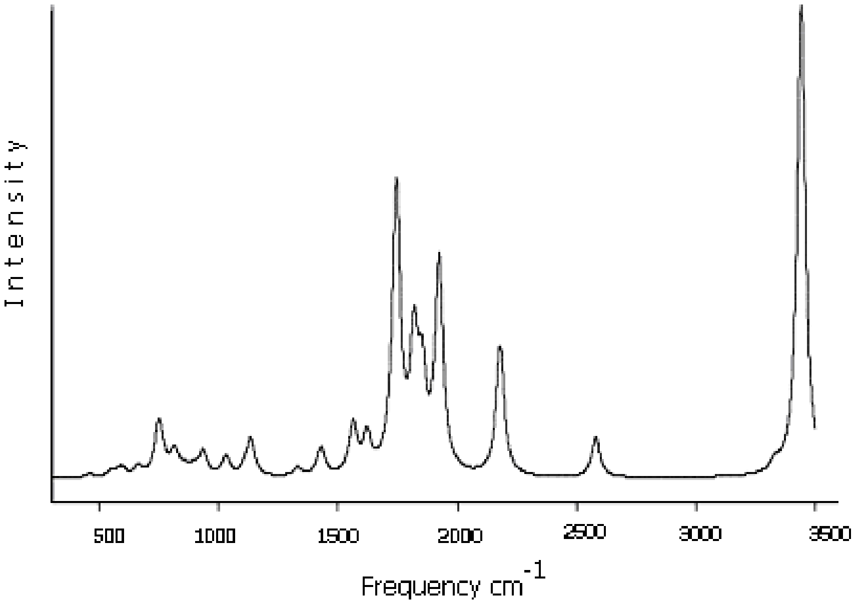


Figure 5.12. Infrared spectrum of neat 2,4,6-TNP in the range of 300-3500 cm $^{-1}$. IR vibrations calculated in the HF approximation in the 6-31G basis.*

Table 5.14. 2,4,6-trinitrophenol molecule harmonic vibrational frequencies, IR and Raman intensities and forms of vibrations calculated in the HF approximation in the 6-31G basis.*

Frequency cm -1	**IR intensity, $D^2/(a.m.v.A^2)$**	**Raman intensity, A/a.m.v.**	**Vibrational form**
1474		142	Benzene ring and NO_2 group deformations
1605	9	63	Benzene ring and NO_2 group deformations
1651	2	33	Benzene ring and NO_2 group deformations
1709		39	Benzene ring deformations
1747	14	120	NO bond stretching vibration
1826	12	24	NO bond stretching vibration
2029	21	14	NO bond stretching vibration
2851		337	CH valence angle deformations
2962		85	CH valence angle deformations
3445		1260	CH bond stretching vibrations
3736		29	CH bond stretching vibrations

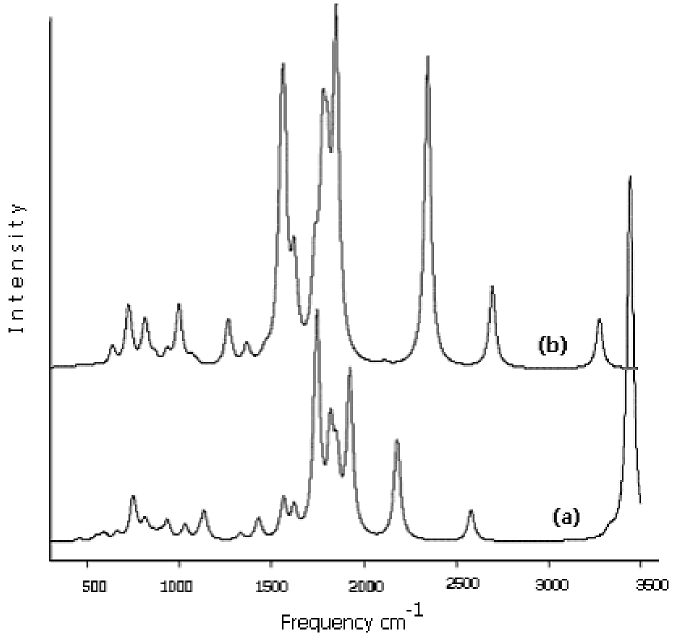


Figure 5.13. Infrared spectrum of neat 2,4,6-TNP in the range of 300-3500 cm $^{-1}$. IR vibrations calculated in the HF approximation:(a) in the 6-31G basis; (b) in the 6-311G* basis.*

Table 5.15. 2,4,6-Trinitrophenol molecule unharmonic vibrational frequencies calculated by vibrational self consistent field method. Unharmonicity of potential energy surface along normal modes only was regarded. Potential surface was calculated in HF approximation in 6-31G basis. Only most intensive modes are presented.*

Harmonic		Diagonal SCF
Frequency, cm^{-1}	IR intensity, $D**2/(a.m.u. A**2)$	Frequency, cm^{-1}
3348.31	0.63834	3123.05
2583.51	2.52890	2016.19
2180.10	8.35142	1483.23
1923.09	12.92500	1615.28
1852.33	5.38627	1819.30
1819.39	7.70223	1734.63
1743.60	17.68548	1700.75
1619.05	2.28316	1481.65
1560.38	2.94315	1558.27
1428.95	1.82790	1464.54
1322.81	0.51712	1361.71
1132.08	0.00918	1136.76
1128.34	2.21725	1204.47
1109.83	0.41348	1111.80
1026.44	1.29588	1056.13
932.30	1.51777	1052.87
894.50	0.54349	1047.29
854.63	0.51690	843.75
814.89	1.26546	801.87
804.38	0.31846	790.72
768.82	0.34285	1020.11
754.07	0.00987	741.97

Infrared spectrum of 2,4,6-trinitrophenol investigations after increasing basis set shows that few frequencies preserved the same form in Hartree-Fock approximation in the 6-311G* basis and in the 6-31G*basis. The two spectra are compare in Figure 5.13 : 1562 cm^{-1} (a) and 1583 cm^{-1} (b) – benzene ring and NO_2 group deformations; 1817cm^{-1} (a) and 1743 cm^{-1} (b), 1922 cm^{-1} (a) and 1864 cm^{-1} (b) –all NO bond stretching vibrations.

This means that quantitative investigation results with increasing the calculations can help us better understand for whom belongs: the benzene ring and NO_2 group deformations, the frequencies of NO group and CH bonds stretching vibrations. This information is important for detection.

5.4.3. The Frequencies of NO_2 Group

Vibrational frequencies and IR intensities were obtained, at several basis sets, for the TNT and for conformers of TNT. Was obtained the IR vibrational spectrum of DNT. In addition, IR spectra were calculated for the NT.

Table 5.16. NO_2 group harmonic vibrational frequencies and IR intensities calculated in HF approximation.

Molecule	**Approximation**	**Frequency, cm^{-1}**	**IR intens., $D^2/(a.m.v.A^2)$**	**Form**
TNT	6-311G (1d,1f,1p) HF	1601	3,0	NO_2 sim. (2,6-DNT); CH_3 def.
		1610	11,3	NO_2 sim. (2,6-DNT)
		1619	0,3	NO_2 sim. (2,6-DNT); CH_3 def.
		1634	5,4	NO_2 sim. (4-NT)
		1736	1,0	NO2 asim. (4-NT); CH def., CH3 def., ring def.
		1771	2,2	NO_2 asim. (2,6-DNT); CH_3 and ring def.
		1819	0,0	NO_2 asim. (2,4,6)
		1857	14,8	NO_2 asim. (2,4,6); ring def.
		1859	19.9	NO_2 asim. (2,6-DNT); ring def.
TNT	6-311G (3d,1f,3p) HF	1611	4,7	NO_2 sim. (2,4); CH_3 def.
		1621	7,0	NO_2 sim. (2,4,6); CH_3 def.
		1648	2,9	NO_2 sim. (4,6-DNT); ring and CH_3 def.
		1711	0,6	NO_2 sim. (4-NT); ring and CH_3 def.
		1819	0,9	NO_2 asim. (2,4,6)
		1830 1845	13,8 18,3	NO_2 asim. (2.4-DNT) NO_2 asim. (2,4,6)
DNT(2,4)	6-311G (3d,1f,3p) HF	1368		NO_2 sim.
		1381		NO_2 sim.
		1581		NO_2 asim.
		1592		NO_2 asim.
NT(4)	6-311G (3d,1f,3p) HF+MP2	1433	0,6	NO_2 sim.; CH_3 def.
		1773	2,2	NO_2 asim.

5.5. Experimental Analysis

A standard solution of the explosive (TNT) was prepared from a solid compound obtained from Chem Service, Inc. All standards were prepared in methanol (HPLC grade CH_3OH, 99.9 %) solutions with a nominal concentration of 2,000 parts per million (ppm) and 4,000 ppm. Stainless steel cylinders of dimensions 3 in. diam. x 1 in. height were used to contain and analyze the mixtures of explosive and soil particles in the Raman microscopy technique experiments. These cylinders were constructed at Añasco Precision Mechanical Shop, Añasco, PR. For FTIR plastic recipient were used to contain the mixture until analyzing. A sample holder of the FTIR instrument was used to place the pellet with the mixture of explosive and soil particles.

5.5.1. Instruments Used for Vibrational Analysis

A Renishaw Raman Microspectrometer RM2000 system was employed for the vibrational spectroscopy measurements. The system was equipped with a Leica microscope and two lasers: Spectra Physics Millennia laser and Renishaw, high power diode laser. The microscope contained objectives with magnifications of 5x, 10x, 20x and 50x. The Millennia laser had a monochromatic excitation source generated by a green diode laser of 532 nm with and a variable output power up to1 W. The second laser was a Renishaw High Power 785 nm diode laser with a maximum power of 200 mW. The spectra were obtained in the 100-3800 cm $^{-1}$ range using 2 scans and 20 seconds of integration time. A multi channel Charge Coupled Device (CCD) detector and GRAMS software were used for data acquisition.

A Bruker Vector-22 spectrometer equipped with a DTGS infrared detector was used to carry out the analysis samples. The KBr pressed disc technique (3.0 mg of sample and 50.0 mg of KBr) was used for FTIR analysis. The sample was placed in the instrument on a sample holder and spectra were measured at a resolution of 4 cm $^{-1}$ and 32 scans. The data was obtained using Opus V3.1 (Bruker Optics) Software [81].

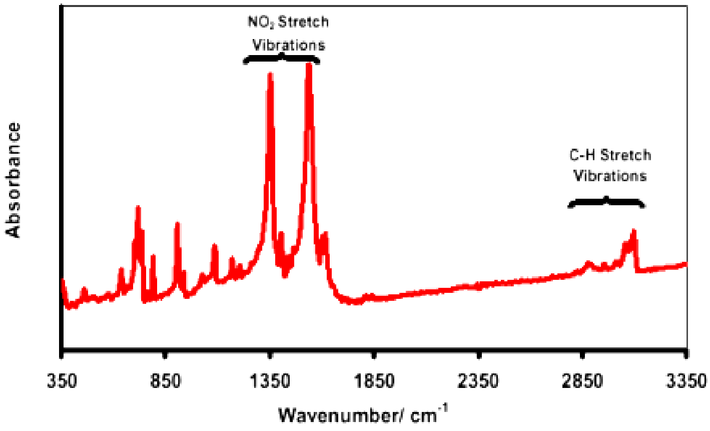


Figure 5.14. The experimental infrared spectrum corresponding to neat 2,4,6-trinitrotoluene in the range of 350-3350 cm $^{-1}$[81].

5.5.2. Spectra of TNT from IR Technique

The experimental infrared spectrum corresponding to neat 2,4,6-trinitrotoluene are shown in Figure 5.14 [81]. The bands that allow identifying the neat TNT were obtained in the range of 350-3350 cm^{-1} .

The vibrational signatures (Figure 5.15) observed in the region of 3000-3100 cm^{-1} can be analyzed and assigned to asymmetric and symmetric C-H stretch vibrations, respectively, belong to the alkyl CH_3 group and the aromatic ring. Other peaks of high intensity are 1355 cm^{-1} (NO_2 symmetric stretching vibration), 1540 cm^{-1} (asymmetric NO_2 stretching), 1025 cm^{-1} (CH_3 deformation), 1085 cm^{-1} (ring C-H in plane bend), at 909 cm^{-1} (methyl rock, C-N stretching band), at 794 cm^{-1} (ring in plane bend, C-CH_3 stretch) and at 720 cm^{-1} (C-N-O bend). The assignments of the bands are summarized in the Table 4 [32-33].

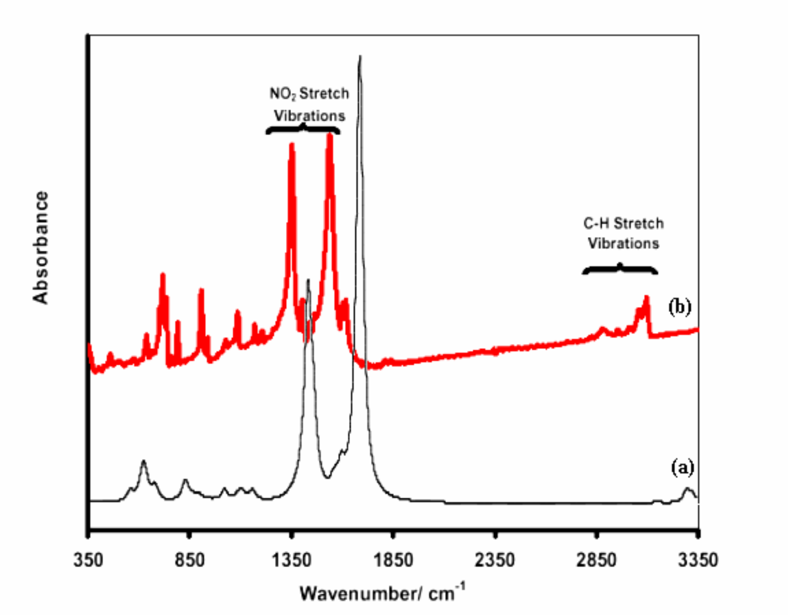


Figure 5.15. The infrared spectrum corresponding to neat 2,4,6-trinitrotoluene in the range of 350-3350 cm $^{-1}$: (a) calculated in the MCSCF approximation in the 6-31G basis; (b) experimental [81].*

5.5.3. Spectra of TNT from Raman Technique

Raman Microscopy was used to identify the spectroscopic signatures of TNT in soil particles. In this step, the particles of soil used were sand and montmorillonite clay. The mixtures of TNT–soil were studied under different environmental conditions (pH, humidity, UV light exposure, temperature, concentration of explosive in the soil, and aging effect).

Figure 5.16 present the spectra of neat TNT. The peaks that permit identify compound can be observed. The major strong bands that allow identifying the TNT were observed in 3016 and 2955 cm $^{-1}$. These bands can be assigned to aromatic C-H stretch vibration and symmetric belong to the alkyl CH3 group, respectively.

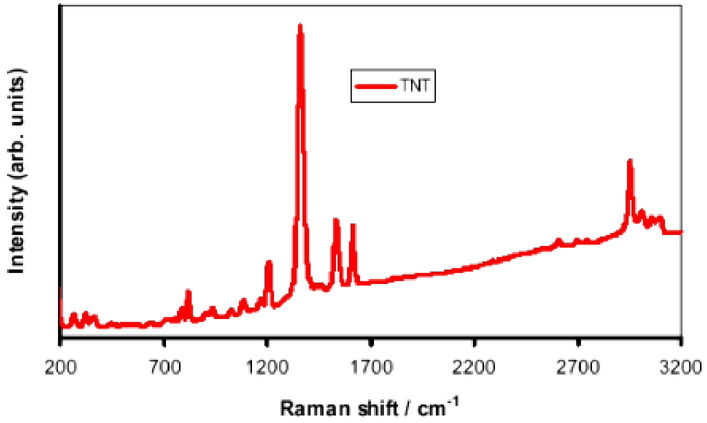


Figure 5.16. The experimental Raman spectrum corresponding to neat 2,4,6-trinitrotoluene in the range of 200-3200 cm $^{-1}$[81].

Table 5.17. Raman frequency assignment of TNT explosive.

Experimental vibrational Frequencies cm $^{-1}$	Assignments by theoretical investigations
326	2,4,6 C-N in plane torsion
366	Methyl group deformation
792	C-CH_3 stretch, 2,4,6-NO_2 scissors
822	2,4,6-NO_2 scissors
914	C-H (ring) out of plane bend
941	C-H (ring) out of plane bend
1091	C-H (ring) in plane bend
1175	Symmetric methyl C-C-H bend
1210	C-H (ring) in plane bend, ring breathing
1361	NO2 symmetric, C-N stretching
1538	NO2 asymmetric stretching
1616	2,6 NO2 asymmetric stretching
2956	C-H asymmetric stretching
3016	Aromatic C-H stretching
3065	Aromatic C-H stretching
3102	Aromatic C-H stretching

The principal peaks of high intensity are 1365cm $^{-1}$ (NO_2 symmetric stretching vibration), 1535 cm^{-1} (asymmetric NO_2 stretching), 1617 cm -1 (2,6-NO_2 asymmetric stretching vibration), 1210 cm $^{-1}$ (C-H (ring) in plane bend, ring breathing). Bands very weak too are observed at 822 cm $^{-1}$ (nitro-group scissoring mode), and 792 cm $^{-1}$ (C-CH_3 stretch, 2,4,6-NO_2 scissoring). Assignments of the bands are summarized in Table 5.17.

5.6. Conclusions of this Chapter

The PC Cluster TAURAS with the SCore cluster system software on top of Linux OS was used to present the investigations of explosive molecules. Created computational facilities were applied to investigate electronic structure and vibration spectra by means of no empirical quantum mechanical computation methods.

The General Atomic and Molecular Electronic Structure System (GAMESS) quantum chemistry computer code was implemented and improved correctly under the specific environment of cluster TAURAS. From the mathematical point, these investigations are nonlinear mathematical modeling problems. They are usually solved by means of computational linear algebra methods. So the parallel computers cluster was used in order to achieve quantitative results comparable with experimental. Cluster TAURAS parameters are comparable with other ones located in academic areas. Cluster is registered in the top500 cluster list [1].

Electronic structure calculations provide useful estimates of the energetic properties of chemical systems, including preferred molecular structures, spectroscopic features and probable reaction paths. This review of some current electronic structure calculations has concentrated on discussion of *ab initio* techniques, as these are the most accurate and, in principle, ,universal methods.

Calculations of positional isomers of 2,4,6-trinitrotoluene molecule at Hartree-Fock level show the influence of NO_2 groups to electronic structure of TNT compounds. When there are more than two constituents, effect plays an important role if the substituted groups are placed close together. This effect was observed in dinitrotoluenes. Results of this study provide valuably information in the field of aromatic nitro compounds.

Created PCs cluster allowed solving of modern quantum chemical problems at necessary level. Trinitrotoluene molecule calculations by MCSCF method showed significant change of vibrational spectrum characteristics from those calculated at Hartree-Fock level. Achieved results allow more detail interpretation of experimental vibrational spectra of investigated molecules and more accurate investigation of thermo chemical reactions dynamics.

Rapid advances in computer technology are making computationally expensive *ab initio* methods increasingly more practical for use with realistic chemical systems. In particular, cheaper methods such as density functional calculations and, layered models are continually being refined, and show promise of providing consistent and accurate chemical predictions for most complex systems.

Energetic materials, which often contain nitro groups that require sophisticated treatments of electronic structure, fall into this category. This dissertation described the application of electronic structure methods to estimates of the chemical properties of energetic materials.

The dissertation reviews the various techniques mostly used for experimental analysis of explosives. Also reviews and evaluates recent developments in methods and techniques for the analysis of explosives from the point of view of both experimental use of instrumentation and quality of the data obtained. Comparison of the performances obtained by different environments, self-made PC cluster from commodity hardware and supercomputer from National centre for High-Performance Computing, confirms that created PCs cluster allows solving of modern quantum chemical problems at necessary level.

CONCLUSIONS

After creating the cluster of personal computers, implementing SCore software and adapting the GAMESS calculation program, integrated in this environment, for real investigations of explosive molecules, the following scientific and practice conclusions are drawn:

1. The cluster of PC’s proposed in this work, was created by implementing freely disseminated Linux OS and SCore software that allows a considerable reduction of data exchange expenditure in the network thus increasing its efficiency and effectiveness. Testing of different environments and analysis of the results obtained were very helpful in ascertaining the merits and demerits of SCore .This enabled us to decrease drawbacks of parallel data transferring models that occur are present in the Beowulf type clusters.
2. Having implemented the *ab initio* calculation program GAMESS of molecule electronic structures, the SCore type PC cluster was adapted for research of quantum chemistry molecules.
3. The SCore type PC cluster self-made for parallel calculations has been proposed and adapted in order to investigate molecular properties of materials by means of the GAMESS code. The theoretical research results obtained and their comparison with experimental data and with results of other authors have illustrated:
   - the results obtained in this work can be widely used in future for investigating molecular properties of materials as well as building SCore type clusters.
   - the work can be of interest for those specialists who have been improving the GAMESS code since 1961, increasing its capabilities for various types of supercomputer and cluster architectures.
   - the *ab initio* results of explosive molecule research can be employed in creating devices for remote detection of pollutants.

BIBLIOGRAPHY

1. Basis of PC clusters. Available at: http://clusters.top500.org/db.

2. Alex A. Granovsky. Available at:
http://www.classic.chem.msu.su/gran/gamess/index.html.

3. PC Cluster consortium. Available at : http://www.pccluster.org/.

4. The Message Passing Interface (MPI) standard. Available at: http://www.mpi-forum.org.

5. MPICH-A Portable Implementation of MPI. Available at:
http://www-unix.mcs.anl.gov/mpi/mpich/.

6. LAM / MPI Parallel Computing. Available at: http://www.mpi.nd.edu/lam/.

7. W. Gropp, E. Lusk, A. Skjellum. Using MPI. The MIT Press. 1999. P. 371.

8. The NAS Parallel Benchmarks (NPB). Available at:
http://www.nas.nasa.gov/Software/NPB/ .

9. Results for supercomputers available at: http://www.pccluster.org/.

10.The Beowulf Project. Available at: http://www.beowulf.org/.

11. Berkeley VIA Project. Available at:
http://www.millennium.berkeley.edu/via.php3.

12. The Computational Plant project. Available at: http://www.cs.sandia.gov/cplant.

13. Dolphin Interconnect. Available at: http://208.179.47.35/index.html.

14. The GM API. Available at: http://www.myri.com/GM/doc/gm toc.html.

15. Packet Engines Hamachi Performance. Available at:
http://www.nscl.msu.edu/~kasten/perf/hamachi/.

16. Chronology of Personal Computers. Available at:
http://www.islandnet.com/~kpolsson/comphist/.

17. Intel Museum, Processor Hall of Fame. Available at:
http://www.intel.com/intel/museum/25anniv/hof/hof main.htm

18. High Performance Virtual Machines. Available at:
http://www-csag.ucsd.edu/projects/hpvm.html.

19. Architecture and Implementation of Memory Channel2. Available at:
http://research.compaq.com/wrl/decarchives/dtj/dtjp03/dtjp03hm.htm.

20. M-VIA: A High Performance Modular VIA for Linux. Available at:
http://www.nersc.gov/research/FTG/via.

21. The Public Netperf. Available at: http://www.netperf.org/.

22. The Berkeley Network of Workstations (NOW) project. Available at:
http://now.cs.berkeley.edu/.

23. John K. Ousterhout, Andrew R. Cherenson, Fred Douglis, Michael N. Nelson, and Brent B. Welch. The sprite network operating system. In IEEE Computer, volume 21, pp 23-36, 1988.

24. Francis O'Carroll, Hiroshi Tezuka, Atsushi Hori, and Yutaka Ishikawa. The Design and Implementation of Zero Copy MPI Using Commodity Hardware with a High Performance Network. In ICS'98, pp 243 - 250, July 1998.

25. PCI technical briefs. Available at: http://www.intel.com/product/tech-briefs/pcibus.htm.

26. D. A. Patterson and D. R. Ditzel. The case for the reduced instruction set computer. In Computer Architecture News, volume 8, pp 25-33, October 1980.

27. PCI Industrial Computer Manufacturers Group. Available at: http://www.picmg.org/.

28. PM 2.1 API. Available at: http://pdswww.rwcp.or.jp/dist/score/html/reference/man/man3/PM.html.

29. Parallel Virtual Machine. Available at: http://www.epm.ornl.gov/pvm/pvm home.html.

30. Richard F. Rashid and George G. Robertson. Accent: A communication oriented network operating system kernel. In the Eighth ACM Symposium on Operating System Principles, volume Operating System Review 15, pp 64-75, December 1981.

31. Overview of Recent Supercomputers. Available at: http://www.top500.org/ORSC/.

32. K.-J. Andersson, D. Aronsson and P. Karlsson. An evaluation of the system performance of a Beowulf cluster. Technical Report 2001-4, LIU, Sweeden, 2001.

33. Gigabit Ethernet and Low-Cost Supercomputing. Available at: http://www.scl.ameslab.gov/Publications/Gigabit/tr5126.html.

34. EEE 802.3 CSMA/CD (ETHERNET). Available at: http://grouper.ieee.org/groups/802/3/.

35. Peripheral Component Interconnect(PCI) Special Interest Group (PCI SIG). Available at: http://www.pcisig.com/.

36. Isa Bus Overview. Available at: http://quatech.salcomm.com/Application Objects/FAQs/comm-over-isa.htm.

37. PCI BUS OVERVIEW. Available at: http://quatech.salcomm.com/Application Objects/FAQs/comm-overpci.htm.

38. The Standard Performance Evaluation Corp (SPEC). Available at: http://www.spec.org/.

39. W. Richard Stevens. In UNIX Network Programming, Volume 1: Networking APIs - Sockets and XTI. Prentice Hall.

40. T. von Eicken, D. E. Culler, S. C. Goldstein and K. E. Schauser. Active messages: a Mechanism for Integrated Communication and Computation. In Proc. of the 19th ISCA, pp 256-266, May 1992.

41. T. von Eicken, V. Avula, A. Basu and V. Buch. Low-Latency Communication over ATM Networks using Active Messages. In Proceedings of Hot Interconnects II, 1994 Palo Alto, August 1994.

42. T. Sterling, D. Savarese, D. J. Becker, B. Fryxell, K. Olson. Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation. In Proceedings of the Fourth IEEE Symposim on High Performance Distributed Computing (HPDC-95), August 1995.

43. Hiroshi Tezuka, Atsushi Hori, Yutaka Ishikawa, and Mitsuhisa Sato. PM: An Operating System Coordinated High Performance Communication Library.

44. In Peter Sloot Bob Hertzberger, editor, High-Performance Computing and Networking, volume 1225 of Lecture Notes in Computer Science, pp 708-717. Springer-Verlag, April 1997.

45. Tosiyuki Takahashi, Yutaka Ishikawa, Mitsuhisa Sato, and Akinori Yonezawa. Class specific optimization environment using compile-time metalevel architecture. In ISCOPE'97.

46. Marvin Theimer, Keith A. Lantz, and David R. Cheriton. Preemptable remote execution facilities for the v-system. In the Tenth ACM Symposium on Operating System Principles, volume Operating System Review 19, pp 2-12,December 1985.

47. Hiroshi Tezuka, Francis O'Carroll, Atsushi Hori, and Yutaka Ishikawa. Pindown Cache: A Virtual Memory Management Technique for Zero-copy Communication. In IPPS/SPDP'98, pp 308-314. IEEE, April 1998.

48. Toshiyuki Takahashi, Francis O'Carroll, Hiroshi Tezuka, Atsushi Hori, Shinji Sumimoto, Hiroshi Harada Yutaka Ishikawa, and Pete H. Beckman. Implementation and Evaluation of MPI on an SMP Cluster. In Parallel and Distributed Processing -IPPS/SPDP'99 Workshops, volume 1586 of Lecture Notes in Computer Science, pp 1178-1192. Springer-Verlag, April 1999.

49. Toshiyuki Takahashi, Shinji Sumimoto, Atsushi Hori, Hiroshi Harada, and Yutaka Ishikawa. PM2: A High Performance Communication Middleware for Heterogeneous Network Environments. In Supercomputing 2000, IEEE and ACM SIGARCH, November, 2000, (Published by CD-ROM). November 2000.

50. The Specification for the Virtual Interface Architecture. Available at: http://www.viarch.org/.

51. Shinji Sumimoto, Hiroshi Tezuka, Atsushi Hori, Hiroshi Harada, Toshiyuki Takahashi, and Yutaka Ishikawa: GigaE PM: a High Performance Communication Facility using a Gigabit Ethernet, New Generation Computing, Springer-Verlag, Vol. 18, pp 177-186, January, 2000.

52. Shinji Sumimoto, Hiroshi Tezuka, Atsushi Hori, Hiroshi Harada, Toshiyuki Takahashi, and Yutaka Ishikawa: The Design and Evaluation of High Performance Communication using a Gigabit Ethernet, International Conference on Supercomputing`99(ICS'99), ACM SIGARCH, pp 243 - 250, June, 1999.

53. Shinji Sumimoto, Hiroshi Tezuka, Atsushi Hori, Hiroshi Harada, Toshiyuki Takahashi, and Yutaka Ishikawa: High Performance Communication using a Commodity Network for Cluster Systems, in the Ninth International Symposium on High Performance Distributed Computing (HPDC-9), IEEE, pp 139-146, August, 2000.

54. Toshiyuki Takahashi, Shinji Sumimoto, Atsushi Hori, Hiroshi Harada, and Yutaka Ishikawa: PM2: A High Performance Communication Middleware for Heterogeneous Network Environments, Supercomputing 2000, IEEE and ACM SIGARCH, Novemver, 2000, (Published by CD-ROM).

55. Yutaka Ishikawa, Hiroshi Tezuka, Atsushi Hori, Shinji Sumimoto, Toshiyuki Takahashi, F. O'Carroll, and Hiroshi Harada: RWC PC Cluster II and SCore Cluster System Software - High Performance Linux Cluster, In Proceedings of the 5th Annual Linux Expo, pp 55 - 62, 1999.

56. Toshiyuki Takahashi, Francis O'Carroll, Hiroshi Tezuka, Atsushi Hori, Shinji Sumimoto, Hiroshi Harada, Yutaka Ishikawa, and Pete H. Beckman: Implementation and Evaluation of MPI on an SMP Cluster, IPPS'99 2nd Workshop on Personal Computer Based Networks of Workstations, pp 1178 - 1192, 1999.

57. Hiroshi Harada, Hiroshi Tezuka, Atsushi Hori, Shinji Sumimoto, Toshiyuki Takahashi, and Yutaka Ishikawa: SCASH:Software DSM using High performance network on commodity hardware and software, ACM Eighth Workshop on Scalable Shared- memory Multiprocessors, pp 26-27, 1999.

58. Yasushi Fujiimoto, Yuan Bin, Hisao Taoka, Hiroshi Tezuka, Shinji Sumimoto and Yutaka Ishikawa: Real-time Power System Simulator on a PC Cluster, Intl. conf. on Power Systems Transients, pp 671-676, June, 1999.

59. Yutaka Ishikawa, Atsushi Hori, Hiroshi Tezuka, Shinji Sumimoto, Toshiyuki Takahashi, and Hiroshi Harada: Parallel C++ Programming System on Cluster of Heterogeneous Computers, IPPS'99 Heterogeneous Computing Workshop '99, pp 73-82, April, 1999.

60. Hiroshi Harada, Yutaka Ishikawa, Hiroshi Tezuka, Atsushi Hori, Shinji Sumimoto, and Toshiyuki Takahashi: Dynamic Home Node Reallocation on Software Distributed Shared Memory, HPC Asia '2000, pp 158-163, May 2000.

61. D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, S. Weeratunga, “The NAS Parallel Benchmarks,” RNR Technical Report RNR-94-007, http://www.nas.nasa.gov/Software/NPB/ Specs/RNR-94-007/node12.html, March 1994.

62. Udaya Ranawake. Performance Comparison of the Three Subclusters of the HIVE.http://webserv.gsfc.nasa.gov/neumann/hive comp/bmarks/bmarks.html.

63. High Performance Computing Linpack Benchmark (HPL) HPL 1.0. Available at:

www.netlib.org/benchmark/hpl, September 27, 2000.

64. H. Dorsett, A. White, "Overview of Molecular Modelling and Ab initio Molecular Orbital Methods Suitable for Use with Energetic Materials". Available at: http://www.dsto.defence.gov.au/publications/2271/DSTO-GD-0253.pdf

65. Message Passing Interface Forum. MPI-2: Extensions to the message-passing interface. http://www.mpi-forum.org/docs/mpi-20.ps, Jul 1997.

66. D.C. Young, Computational Chemistry: A practical guide for applying techniques to real-world problems, Wiley-Interscience, John Wiley & Sons, Inc., N.Y., 2001.

67. E. Canc`es, M. Defranceschi, W. Kutzelnigg, C. Le Bris and Y. Maday, Computational quantum chemistry: a primer, Handbook of Numerical Analysis vol. X, Elsevier, Amsterdam, pp 3–270, 2003

68. E. Canc`es, H. Galicher & M. Lewin. Computing electronic structures: a new multiconfigu-ration approach for excited states. J. Comput. Phys. 212, pp 73–98, 2006.

69. R. Eade and M. Robb, Direct minimization in MCSCF theory. The quasi-Newton method, Chem. Phys. Lett. 83, no. 2, pp 362–368, 1981.

70. G. Friesecke, The multiconfiguration equations for atoms and molecules: charge quantization and existence of solutions, Arch. Rat. Mech. Anal. 169, pp 35–71, 2003.

71. M. Lewin. Solutions of the multiconfiguration equations in quantum chemistry. Arch. Ration.Mech. Anal. 171, no. 1, pp 83–114, 2004.

72. M. McCourt and J. McIver Jr. On the SCF calculation of excited states: singlet states in the two-electron problem, J. Comput. Chem. 8, no. 4, pp 454–458, 1987.

73. R. Shepard, The multiconfiguration self-consistent field method. Ab initio methods in quantum chemistry - II, Adv. Chem. Phys. 69, pp 63–200, 1987.

74. H.-J. Werner, Matrix-formulated direct multiconfiguration self-consistent field and multi-configuration reference Configuration-Interaction methods. Ab initio methods in quantum chemistry - II, Adv. Chem. Phys. 69, pp 1–62, 1987

75. H.-J. Werner and P. Knowles, A second order multiconfiguration SCF procedure with optimum convergence, J. Chem. Phys. 82, no. 11, pp 5053–5063, 1985.

76. P. C. Chen, W. Lo, K.H. Hu, "Ab initio studies of the molecular structures of dinitrotoluenes", Journal of Molecular Structure (Theochem), 389, pp 91-96, 1997.

77. P. C. Chen, S.C.Tzeng,"Theoretical study on the molecular structures of dinitrophenols and trinitrophenols", Journal of Molecular Structure (Theochem), 467, pp 243-257, 1999.

78. P. C. Chen, W. Lo,” Molecular orbital studies of the isomers of 2,4,6-trinitrotoluene and some of its thermal decomposition products”, Journal of Molecular Structure (Theochem), 397, pp 21-32, 1997.

79. D.L. Bish, R. B. Von Dreele, “Rietveld refinement of non-hydrogen atomic positions in kaolinite”, *Clays and Clay Mine*r., 37, 1989, 289-296.

80. E. Balan, A. M. Saitta, F. Mauri and G. Calas, “First-principles modeling of the infrared spectrum of kaolinite”, *American Mineralogis*t, 86, 2001, 1321-1330.

81. Vibrational signatures of 2,4,6 - trinitrotoluene (tnt) in soil particles. Available at: http://grad.uprm.edu/tesis/herrerasandoval.pdf.

82. Prasad, R., Photoacoustic Spectra and modes of Vibration of TNT and RDX at CO2 Laser Wavelengths, *Spectrochim. Act*a, Part A, 58: 3093 – 3102, 2002.

83. Clarkson, J.A., Theoretical Study of the Structure and Vibrations of 2,4,6-trinitrotoluene, *J. Molec. Structur*e, 648: 203 -214, 2003.

84. E.N. Duesler, J.H. EngeImann, D.Y. Curtin, I.C. Paul, Cryst.Strut. Comm. 7, p 449, 1978.

85. J.R Holden, C. Dickinson, C.M. Bock, J. Phys. Chem. 76 (1972) 3597.

86. W.R. Carper, L.P. Davis, J. Phys. Chem. 86 (1982) 459.

87. A. Crockett, T. Jenkins, H. Craig, and W. Sisk, Overview of on-Site Analytical methods for Explosives in Soil, US Army Corps of Engineers, Cold Regions Research & Engineering Laboratory, Special report, pp 98-4, 1998.

88. W. J. Hehre, R. Ditchfield, J. A. Pople, J. Chem. Phys. V. 56, p. 2257 (1972).

89. M.W.Schmidt, et al, J. Comput. Chem. V. 14, p. 1347 (1993).

90. M.W.Schmidt, M. S. Gordon, Ann. Rev. Phys. Chem. V. 49, p. 233 (1998).

91. B. Jezionki, W. Kolos. *Molecular Interaction*s. Eds. Wiley, New York, 3, 1982.

92. H. dorsett and A. White, Overview of molecular modelling and ab initio molecular orbital methods suitable for use with energetic materials. DSTO, pp1-35, 2000.

93. P. Hobza, J. Sponer, T. Reschel, *J. Comput. Che*m., 11, 1995, 1315.

94. S. F. Boys, F. Bernardi, “The calculation of small molecular interactions by the differences of separate total energies. Some procedures with reduced errors”, *Mol Phy*s, 19, pp 553-561,1970.

95. Madejova´, J. and Komadel, P., Baseline Studies of the Clay Minerals Society SourceClays: Infrared Methods, *Clays and Clay Mineral*s, 49: pp 410 – 432, 2001.

96. Madejova´ J. Review. FT –IR techniques in Clay Minerals Studies. Vibrational Spectroscopy, 31: pp 1 – 10, 2003.

97. E.N.Duesler, J.H. Engelmann, D.Y. Curtin, I.C. Paul, Cryst. Strukt. Comm. 7, pp 449-461, 1978.

LIST OF PUBLICATIONS BY THE AUTHOR

Articles on which this dissertation is based:

1. Cicėnas S., Rakauskas R.J., **Vošterienė S.**, Šulskus J. Theoretical Investigation of the Vibrational Spectra of Trinitrotoluene And 2,4,6-Trinitrophenol Molekules // Lietuvos fizikos žurnalas. ISSN 1392-1932. **2001**, Vol. 41, No. 3, p. 221-225.
2. Šulskus J., Rakauskas R. J., **Vošterienė S.** Parallel Algorithms for Solving of Multidimensional Vibrational Shrödinger Equation // *Lithuanian Mathematic al Journal*. ISSN 0132-2818. **2002**, 42 (spec.no.), p.345-350.
3. Šulskus J., Rakauskas R. J, **Vošterienė S.** PC Cluster Possibilities in Mathematical Modeling in Quantum Mechanical Molecular Computations // Nonlinear analysis: Modelling and Control. ISSN 1392-5113. **2002**, Vol.7, No.2, p. 113-121.
4. R. J. Rakauskas, J. Šulskus, **S .Vošterienė.** Simulation of Vibrational Spectra Peculiarities of Trinitrotoluene Molecule Using Parallel Calculations // Материалы XV международного семинара "Лазеры и оптическая нелинейность" 6-8 июня **2002** г., Минск, Беларусь. с. 162-168.
5. Pankevičius E., Šulskus J., Rakauskas R. J., **Vošterienė S.** Investigation of PC Cluster Productivity in Quantum Mechanical Molecular Computations // *Lithuanian Mathematical Journal*. ISSN 0132-2818. **2003**, 43 (spec.no.), p.190-193.
6. **Vošterienė S.**, Šulskus J. GAMESS Calculations in Parallel Environment // *Lithuanian Mathematical Journal*. ISSN 0132-2818. **2005**, 45 (spec.no.), p.190-193.
7. **S. Bekešienė**, S. Sėrikovienė. Quantum Chemical Calculations by Parallel Computer from Commodity PC Components // Nonlinear Analysis: Modelling and Control. ISSN 1392-5113. **2007**, Vol. 12, No. 4, p. 461-468.

SANTRAUKA

Viena rimčiausių žmonijos problemų – įvairios terorizmo apraiškos. Aktualia tampa teroro aktų prevencija ir užkarda, kurią sprendžiant būtina sutelkti įvairių krypčių mokslininkus. Norint sėkmingai spręsti šiuos uždavinius, būtina tobulinti esamas ir kurti iš esmės naujas pažangias sprogstamųjų medžiagų aptikimo priemones. Iškyla poreikis turėti tokias technines priemones, kurios leistų greitai, selektyviai ir pageidautina nuotoliniu būdu aptikti ir identifikuoti pavojingas nuodingąsias medžiagas.

Sprogstamųjų medžiagų buvimą, lokalizaciją, identifikavimą galime realizuoti cheminės analizės metodais (specifiškos reakcijos su testinėmis medžiagomis). Tačiau, kai medžiagos kiekiai maži, medžiagos lokalizacija nežinoma – cheminiai metodai yra nereprezentatyvūs. Praktikoje tokiais atvejais dažniausiai taikomi fizikiniai spektroskopijos metodai, pagrįsti tuo, kad medžiagos molekulės pasižymi tik joms būdingais spektrais, o tiksliau, spektro dalimis. Kai kurios spektroskopijos rūšys, tokia kaip Ramano spektroskopija, leidžia pagal žinomus spektrus registruoti labai mažus medžiagų kiekius (iki nedaugelio molekulių). Tačiau ir šiuo būdu galimybė iš karto detektuoti medžiagą realiai esančią aplinkoje, o ne vakuume, yra labai sudėtinga, dėl aplinkoje esančių kitų medžiagų, kurių spektrai persiklodami iškraipo informaciją. Žinant, kad spektrai yra unikalūs, dažnai taikoma idealizacija, ir medžiagos detektuojamos neatsižvelgiant į jų superpoziciją su fonine aplinka.

Daugiatomių molekulių spektrai apima plačią infraraudonojo (IR) spektro sritį. Norint tokias molekules detektuoti, reikia atlikti IR spektrų klasifikavimą, nustatant svyravimų modas ir savuosius dažnius. Tai sudėtingas uždavinys. Todėl ab initio kvantiniai-cheminiai skaičiavimai naudojami nustatant molekulių elektroninės struktūros ypatumus: molekulių geometriją, jų svyravimų modas, dažnius, šuolių intensyvumus, izotopinių spektrų slinktis, savaiminio terminio skilimo reakcijų kanalus, aktyvacijos energiją ir rotacines konstantas.

Sprogstamųjų medžiagų, tokių kaip trinitrotoluenas (TNT), izomerai yra daugiatomiai azoto junginiai, sudarantys nemažai darinių su viena ar dviem NO_2 grupėmis, įvairiai susijungusiomis su benzolo žiedu. Šių darinių, nors visi jie TNT izomerai, spektrai skiriasi, todėl būtina ištirti visus junginius. Tarp jų trinitrotolueną arba trotilą, jo skilimo produktus dinitrotolueną (DNT) ir nitrotolueną (MNT) bei heksahydro-1.3.5-trinitro-1.3.5-triaziną (RDX).

Ta pati medžiaga gali turėti įvairių darinių, kurių spektrai yra skirtingi. Norint detektuoti visus junginius atsirandančius dėl reakcijos (sprogimo), būtina ištirti jų stabilumą ir spektrus. Eksperimentas negali parodyti, su kuriomis molekulės dalimis susijusi ta ar kita spektro linija, tad taikomas kitas būdas – nagrinėjama teoriškai. Šiuo metu kvantinės chemijos tyrimai yra tapę standartu, o siekiant teorinio tyrimo rezultatus gauti kaip lyginamuosius skaičiavimus su eksperimento rezultatais, naudojami superkompiuteriai arba asmeninių kompiuterių klasteriai.

Šiuo tikslu Lietuvos karo akademijoje ir buvo iš asmeninių kompiuterių sukurtas SCore tipo asmeninių kompiutrių klasteris, realizuotas sprogstamųjų medžiagų molekulių tyrimui. Klasteryje, įdiegus ir modifikavus kompiuterinį paketą GAMESS, atliktas išsamus sprogstamųjų medžiagų molekulių skaitmeninis modeliavimas, padėjęs realizuoti specifinę SCore tipo klasterių aplinką kvantcheminiams tyrimams.

Atlikus klasteryje sudėtingus ir daug išteklių reikalaujančius skaičiavimus, molekulių geometriniai parametrai surasti Hartrio ir Foko artinyje panaudojus 6-311G Gauso atominių orbitalių bazę su poliarizacinėmis funkcijomis (2d,2p,3d,1f,3p). Siekiant patikslinti gautus skaičiavimus, atsižvelgiant į elektronų koreliaciją, molekulių geometriniai parametrai optimizuoti trikdžių metodu MP2 artinyje, t. y. buvo ieškoma tokių geometrinių parametrų, kuriems esant molekulės pilnoji energija mažiausia, nes tada sistema stabiliausia.

Klasteryje atlikti skaičiavimai sprogstamųjų medžiagų detektavimui nustatantys spektrų charakteringąsias sritis. Harmoniniame artinyje apskaičiuoti tiriamų molekulių vibracinių šuolių dažniai ir santykiniai IR spektro intensyvumai. Teoriškai nustatyta NO_2 grupių įtaka molekulių junginių elektroninei sandarai ir IR spektrams.

APPENDIX

A. Methods of Quantum Chemistry

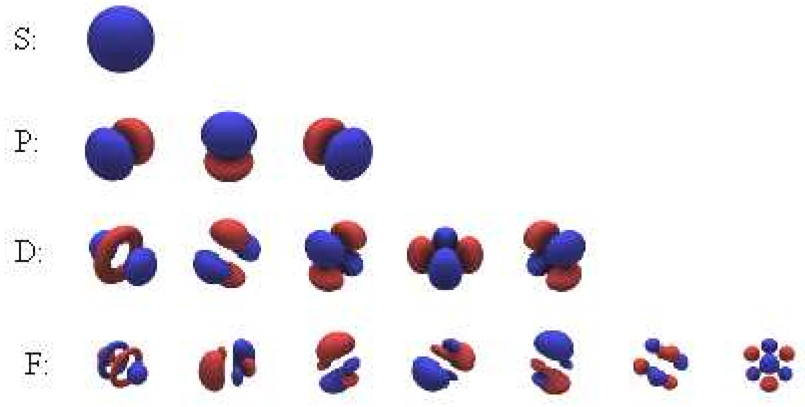
One of the most basic quantum chemistry methods is Hartree-Fock theory (HF). In this method each electron sees the average effect of all the other electrons. The resulting equations are solved iteratively until the entire solution is self-consistent. HF is not very accurate, but it serves as the foundation for better approximations, known as correlated methods. In correlated methods, electrons do not feel merely the average influence of other electrons, they instead respond to the dynamic motion of other electrons, and, hence, the electrons' motions are correlated. Møller-Plesset perturbation theory, coupled cluster theory, and configuration interaction theory are all examples of correlated methods.

The HF approximation for certain molecules is so poor that it cannot serve as an acceptable starting point for correlated methods. For these molecules it is not possible to separate the self-consistent HF solution from the inclusion of electron correlation, and more expensive and difficult to use multi-reference methods must be used.

An entirely different quantum chemistry method is Density Functional Theory (DFT). This method is based on the Hohenberg-Kohn theorem, which, in effect, states that the exact electron density can be determined by finding the density that minimizes a density functional. The exact form of the functional is unknown. In the Kohn-Sham approach to DFT, equations similar to the Hartree-Fock equations, but with an extra term, are solved. DFT generally gives good answers with less processing time than other methods, but it does not offer extremely high accuracy, and it is not possible to vary the DFT method in a systematic way to check for convergence of the solution.

B. What is a basis set?

When employing classical mechanics for modeling, the location of all the particles is known. This is not the case, however, when using quantum mechanics. Because of the large mass of the nuclei it is usually a good approximation to pinpoint the nuclear locations, but the electrons are smeared out over space and must be described by a wave function. This wave function is typically constructed of orbitals, each of which describes a single electron. Each orbital is described by a linear combination of basis functions. The form of the basis functions are motivated by the exact quantum mechanical descriptions of simple atomic systems. These basis functions naturally fall into shells labeled S, P, D, F, and so on. In one real-valued representation, they look like the following:



Below are listed some well-known basis sets to illustrate the available range. In all, there are hundreds of basis sets. Also given below are the number of basis functions on the first, second and third row elements of the periodic table. The more basis functions there are, the better the description of the orbitals, and the more costly the calculation.

Basis Name	Description
STO-3G	Minimum sized basis. Used as initial guess for better calculations.
3-21G*	Better than STO-3G, but still small.
6-311G**	Usually acceptable for HF and DFT.
cc-pTZV	Designed for correlated methods such as MP2 and CCSD(T).
aug-cc-pTZV	Supplements cc-pVTZ for molecules with loosely bound electrons.
aug-cc-pCV5Z	Very high quality basis including functions for describing effects involving low energy electrons.

C. Parallelization of First Principle Calculations

Hartree-Fock Theory

The most computationally intensive piece of the construction of the Fock matrix is the calculation of the two electron integrals. The density matrix, D and the Fock matrix, F, are symmetric and for any (i,j,k,l) the following integrals are equivalent:

(ij|kl) = (ji|kl) = (ij|lk) = (ji|lk) = (kl|ij) = (kl|ji) = (lk|ij) = (lk|ji)

Hence, once (ij|kl) has been computed then the elements Fij, Fik, Fil, Fjk, Fjl and Fkl can be updated with the product of this integral and the appropriate element of the density matrix. Thus, rather than having to compute N4 integrals, only ~N4/8 integrals need to be calculated.

Screening is often considered to reduce the number of integrals requiring calculation. Simply this means that integrals whose size is so small that they are negligible are eliminated. This can reduce the number of integrals from O(N4) to O(N2) in some cases.

The fact that each integral may be computed separately means that integral evaluation can be parallelized.

The replicated data method is the same principle as that described for molecular dynamics.

There are several examples of the use of the replicated data scheme for SCF methods, however one of the most well known is by Cooper et al. (1991) who parallelized GAMESS-UK (Guest et al., 1992) on a transputer based system.

An alternative method to replicated data schemes work is that of distributed data algorithms. Burkhart et al. (Burkhart et al., 1993) distributed the integral evaluation between processors and in the process accumulated the Fock matrix on one fast processor with a large amount of memory. One problem with this algorithm was the serial accumulation of the Fock matrix on the one processor which limited the speed up to the ratio of:

$$\frac{\textit{time taken to compute integrals}}{\begin{array}{c}\textit{time required to send them to the master processor}\\ \textit{and add them to the Fock matrix.}\end{array}}$$

The model Burkhandt et al. considered was a farm model; one specific processor (the master processor) generates the jobs and distributes them to the other (server) processors. In this situation the number of data sets must significantly exceed the number of processors. To optimize the efficiency, the authors considered two different communication options and two different methods of updating the Fock matrix.

Communication options:

1) The master processor distributes the jobs to all the server processors and then receives the results from all the server processors. The master processor then schedule new jobs to the now idle server processes (called global communication management).

2) Each server processor decides whether to process a given job or to send it to another server (local communication management).

Fock matrix generation:

1) All the calculated integrals are returned to the master processor where the Fock matrix is calculated using the integrals and the density matrix (sequential Fock matrix update).

2) Each server receives the density matrix and builds its own partial Fock matrix (distributed Fock matrix update).

The problem with using a sequential Fock matrix update was the Fock matrix determination created a bottleneck for the communications required. They achieved better success with this technique when local communication management was utilized rather than global communication management however they concluded that for more than sixteen processors using a distributed Fock matrix was most effective.

There are a number of different methods however of mapping the computational tasks to the processors:

1) If computational task (ijk) is mapped onto the same processor as data task i, then the amount of communication is reduced by one third. This is because computational task (i,j,k) communicates with data tasks i (which is now local), j and k. The number of integrals within a computational task and the amount of computation per integral can vary and hence different amounts of computation may be allocated to different processors.

2) Computations can be allocated randomly or cyclically i.e. a probabilistic mapping.

3) A task-scheduling algorithm. A simple centralized scheduler can be used to allocate tasks to idle processors.

4) A hybrid scheme. e.g. tasks are allocated randomly to a set of processors, whereby a master/scheduler is used.

This algorithm has higher communication costs than the replicated data algorithm however; it is very scalable unlike the replicated data algorithm. The distribution of tasks is complex, load balancing problems can occur and the overall scaling was, although good, not as good as desired.

Svajonė Bekešienė

PARALLEL COMPUTATION SYSTEM FOR MATHEMATICAL MODELING OF ELECTRONIC STRUCTURE OF EXPLOSIVE MATERIALS

Doctoral Dissertation

Išleido ir spausdino leidykla spaustuvė “Mokslo aidai”,
A. Goštauto g. 12, LT-01108 Vilnius
Užsakymo Nr. 1708. Tiražas 20 egz.