14th International Workshop DAMSS 2023
November 30 – December 2, 2023, Druskininkai

VILNIUS TECH
Vilnius Gediminas Technical University

Performance-based SLO Recovery for Containerized Applications

Olesia Pozdniakova, Dalius Mažeika
Vilnius Gediminas Technical University, Saulėtekio al. 11 Vilnius, Lithuania

Summary. Developing cloud-ready applications needs to be scaled dynamically. Most of Auto-scaling algorithms aim to meet Service Level Objective (SLO) requirements by monitoring the actual value of Service Level Indicator. However, non of existing autoscalers do not provide SLO recovery mechanisms in case of violations. A threshold-based autoscaling approach named as Service Level Agreement (SLA) Adaptive Auto-scaler (SAA) was proposed to restores SLO and fulfi SLA requirements. Effectiveness of the SAA algorithm was evaluated to recover SLA under five different conditions and load scenarios. Obtained results were compared with a similar dynamic thresholds-based autoscaler named as Dynamic Multi-level Auto-scaling Rules for Containerized Applications (DMAR). Results showed that SAA utilizes a similar amount of conteiners DMAR algorithm while providing better conformance with SLA in most of the load scenarios.

INTRODUCTION

Fulfiling quality of service requirements defined in SLAs is a well-known challenge for the applications running on a clouds. Achievement of SLA requirements strongly depends on autoscaling algorithm, as well as Service Level Indicators (SLIs) and metrics that the algorithm utilizes to make an autoscaling decision. Adjustement of autoscaling principle using the actual metric value, such as the response time, CPU load, or network traffic is not sufficient to achieve the required SLO. In addition to SLI, the autoscaling mechanism must also take into consideration the SLO state during the SLA-defined service monitoring window in order to detect SLO violations and to restore the SLO to its required status.

DESCRIPTION OF AUTOSCALERS

SLA Adaptive Autoscaler (proposed)

1. Uses dynamic upscale and downscale CPU thresholds to trigger autoscaling of the containers
2. Dynamic CPU threshold value is:
   - ❑ selected based on load velocity
     - Highest velocity -> lowest threshold
     - Lowest velocity -> highest threshold
     - No change -> normal operation
   - ❑ adjusted based on SLA status
     - SLO violated -> lower CPU thresholds
     - SLO above target -> higher CPU thresholds
     - SLO on target -> no change

Dynamic Multi-level Autoscaler

1. Uses predefined target CPU and Response Time thresholds to trigger upscaling action
2. Calculates number of containers to achieve desired CPU threshold based on:
   - current average CPU resource utilization
   - current number of containers
   - increase in the rate of throughput
3. Uses dynamic downscale threshold calculated based on:
   - current number of containers
   - current average CPU resource utilization

RESULTS OF THE EXPERIMENTS

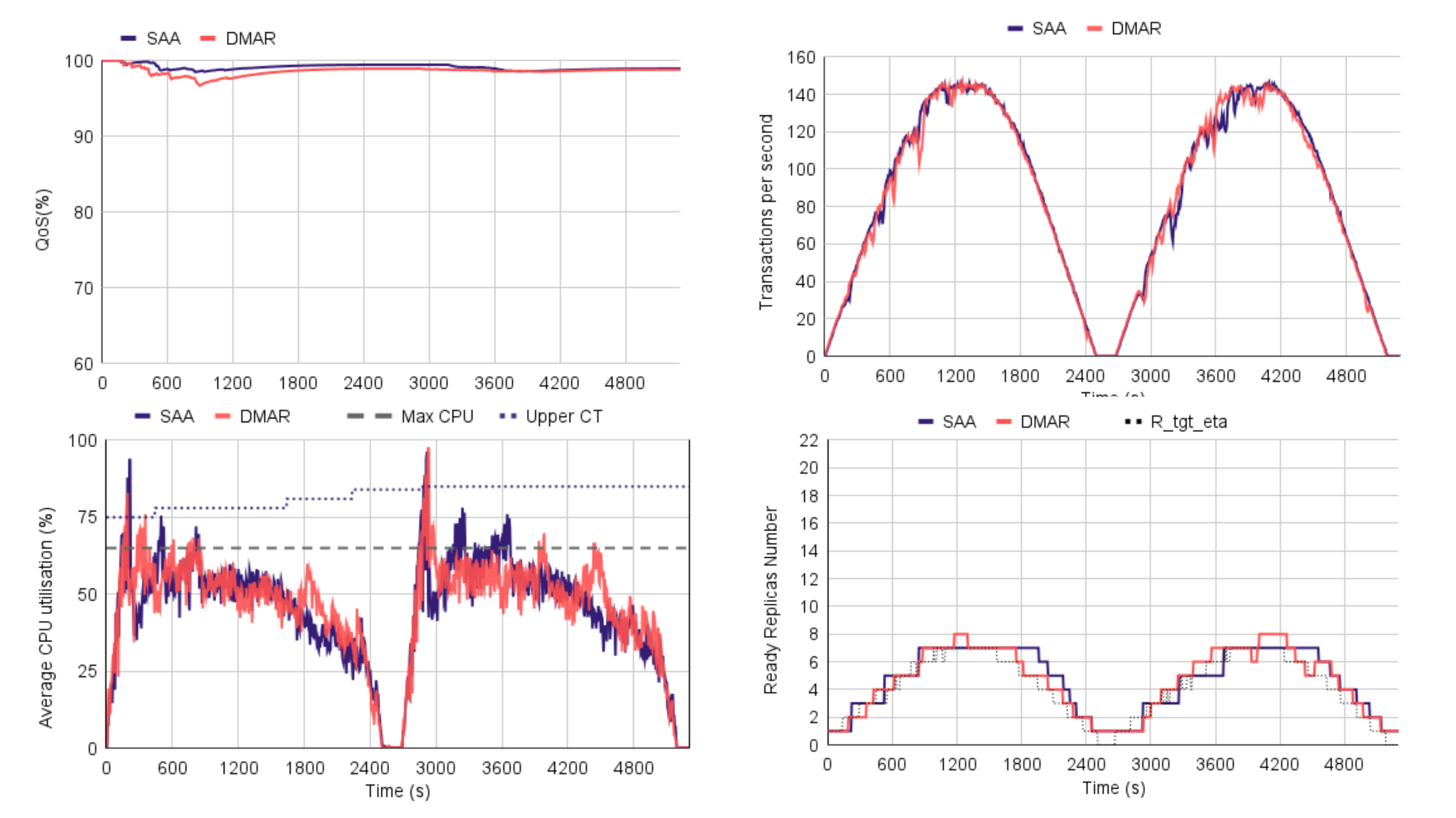

Fig. 1 Results obtained under slow changing RPS load

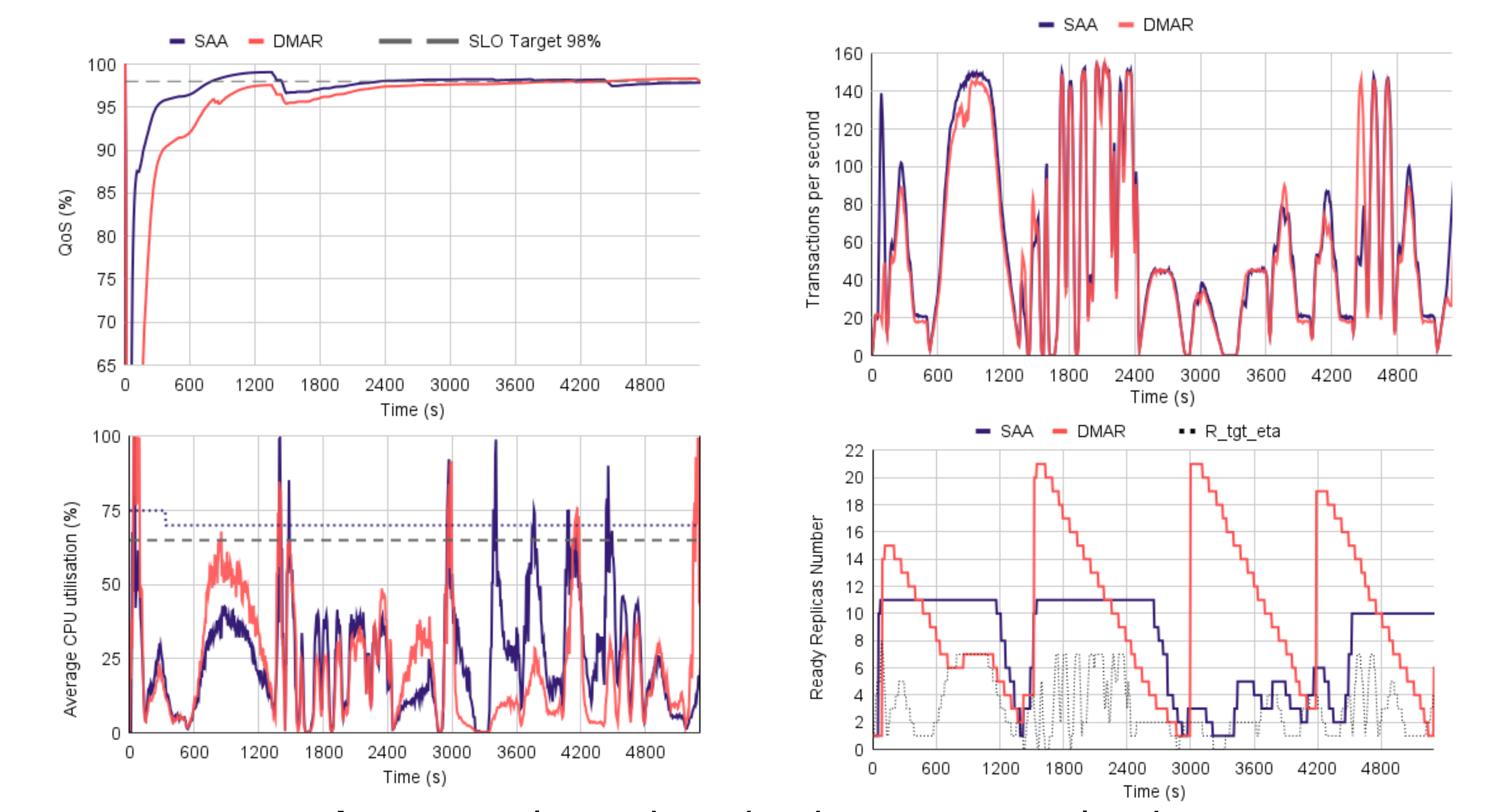

Fig. 2 Results under mixed pattern RPS load

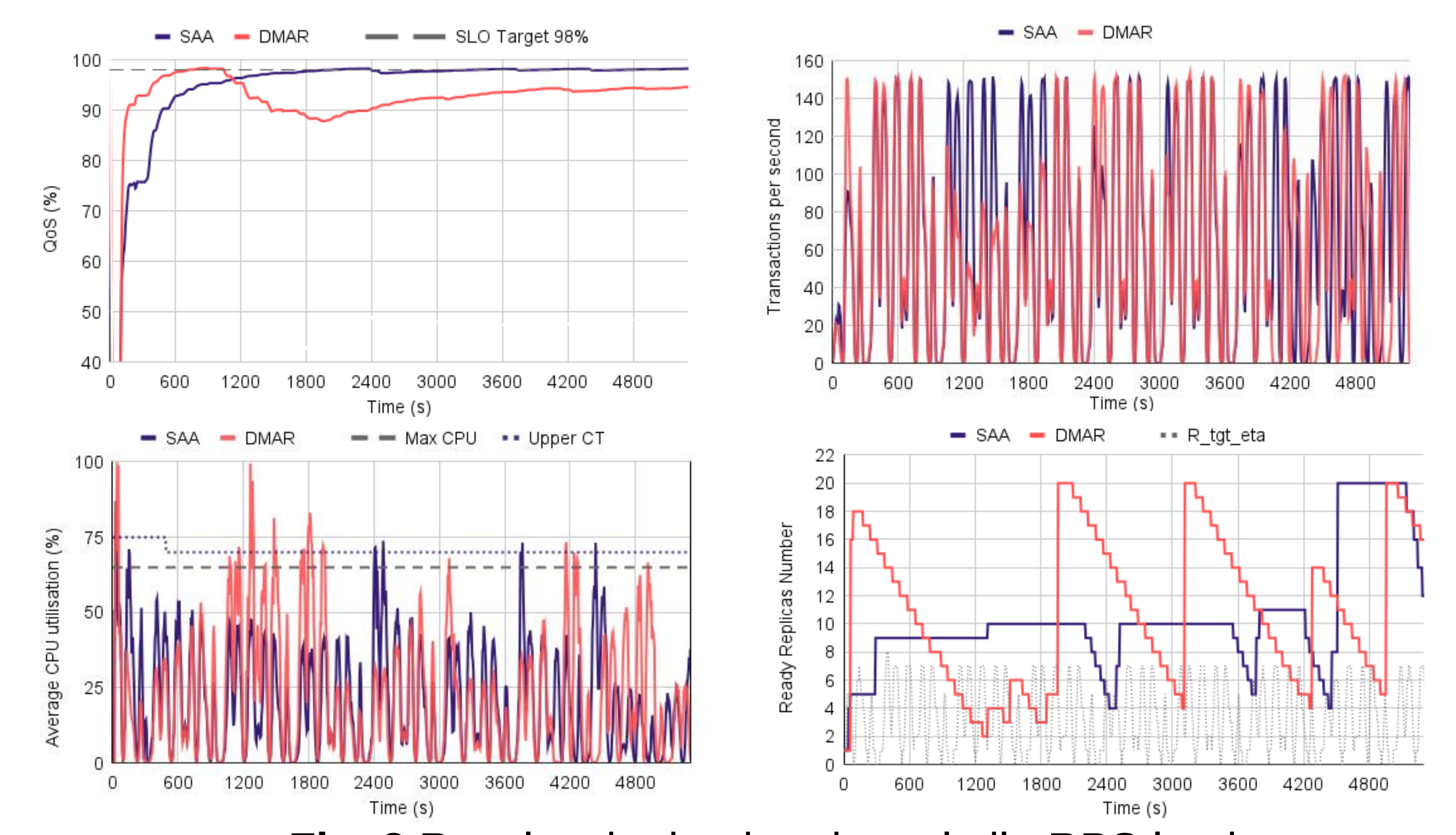

Fig. 3 Results obtained under volatile RPS load

Table 1. Summary of the experimental study

Load type	Autoscaler	Required Pods	DMAR Pods	SAA Pods
Slow	DMAR	3722	4049 (109%)	4115 (110%)
Volatile	SAA	2973	9849 (333%)	9373 (313%)
Mixed	SAA	2734	9006 (333%)	6994 (253%)

REFERENCES

1. Pozdniakova O., Cholomskis A., Mažeika D. Self-adaptive autoscaling algorithm for SLA-sensitive applications running on the Kubernetes clusters. Cluster. Comput., 2023.
2. Taherizadeh S., Stankovski V. Dynamic multi-level auto-scaling rules for containerized applications. Computer Journal, Vol. 62(2), p. 174–197 (2019).