Multi-Leveled Hierarchical Control to Optimize Workload of a Service-Oriented Platform

Maksim Khegai, Tatiana Kharchenko, Dmitrii Zubok and Alexandr Maiatin
ITMO University, St.Petersburg, Russia

ITMO UNIVERSITY

Table of Contents

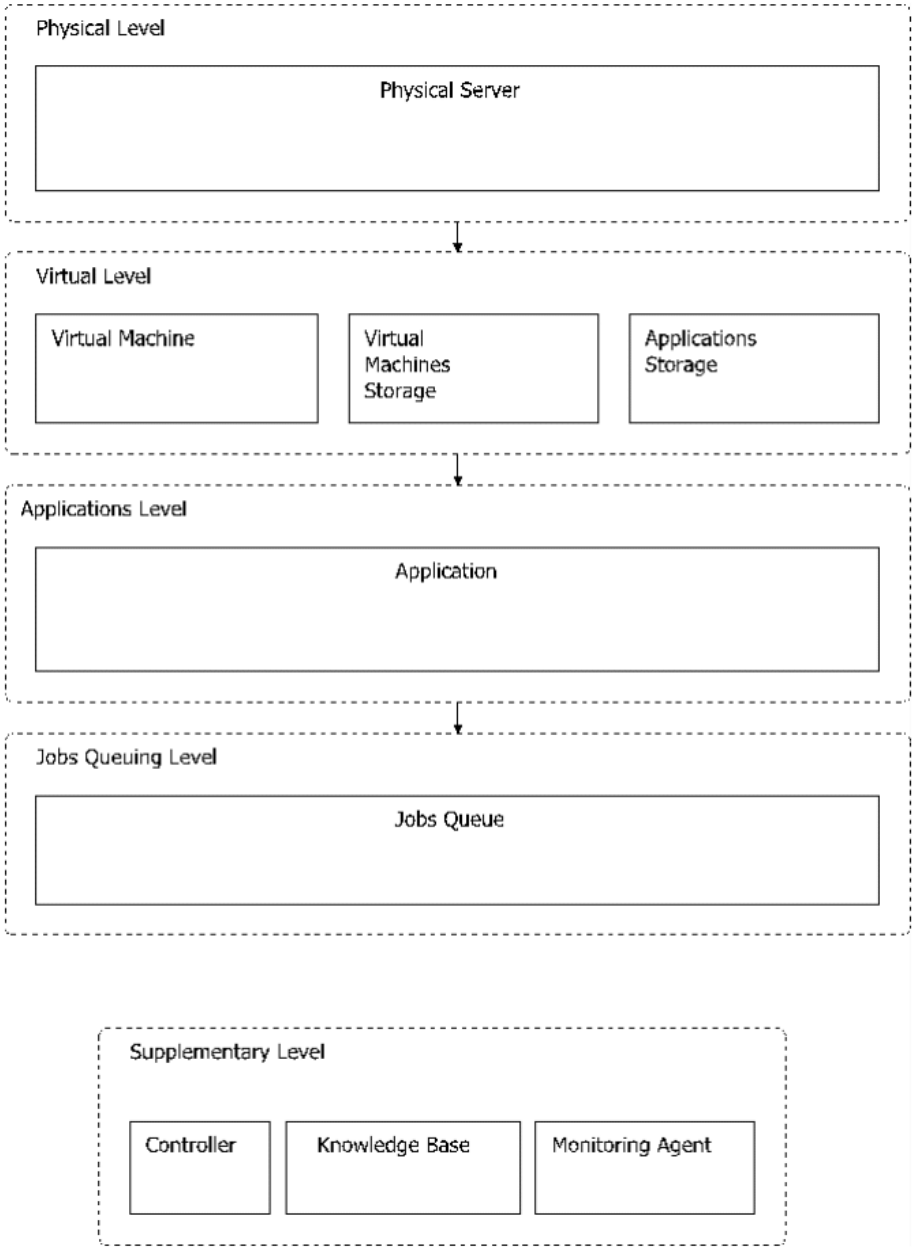
- Introduction and motivation
- Basic architecture of the system
- Architecture of the system
- The algorithm
- Experiments results
- Conclusion

Introduction and motivation

- A performance control was and will be a very important part of every computing system
- Modern cloud systems utilize complex system architecture and as a result have a lot of parameters for performance control
- Simultaneous use of several optimization techniques is not widely researched

Basic architecture of the system: Levels

- Physical level
- Virtual level
- Applications level
- Jobs queuing level
- Supplementary level


Physical Level
Physical Server
Virtual Level
Virtual Machine
Virtual Machines Storage
Applications Storage
Applications Level
Application
Jobs Queuing Level
Jobs Queue
Supplementary Level
Controller
Knowledge Base
Monitoring Agent

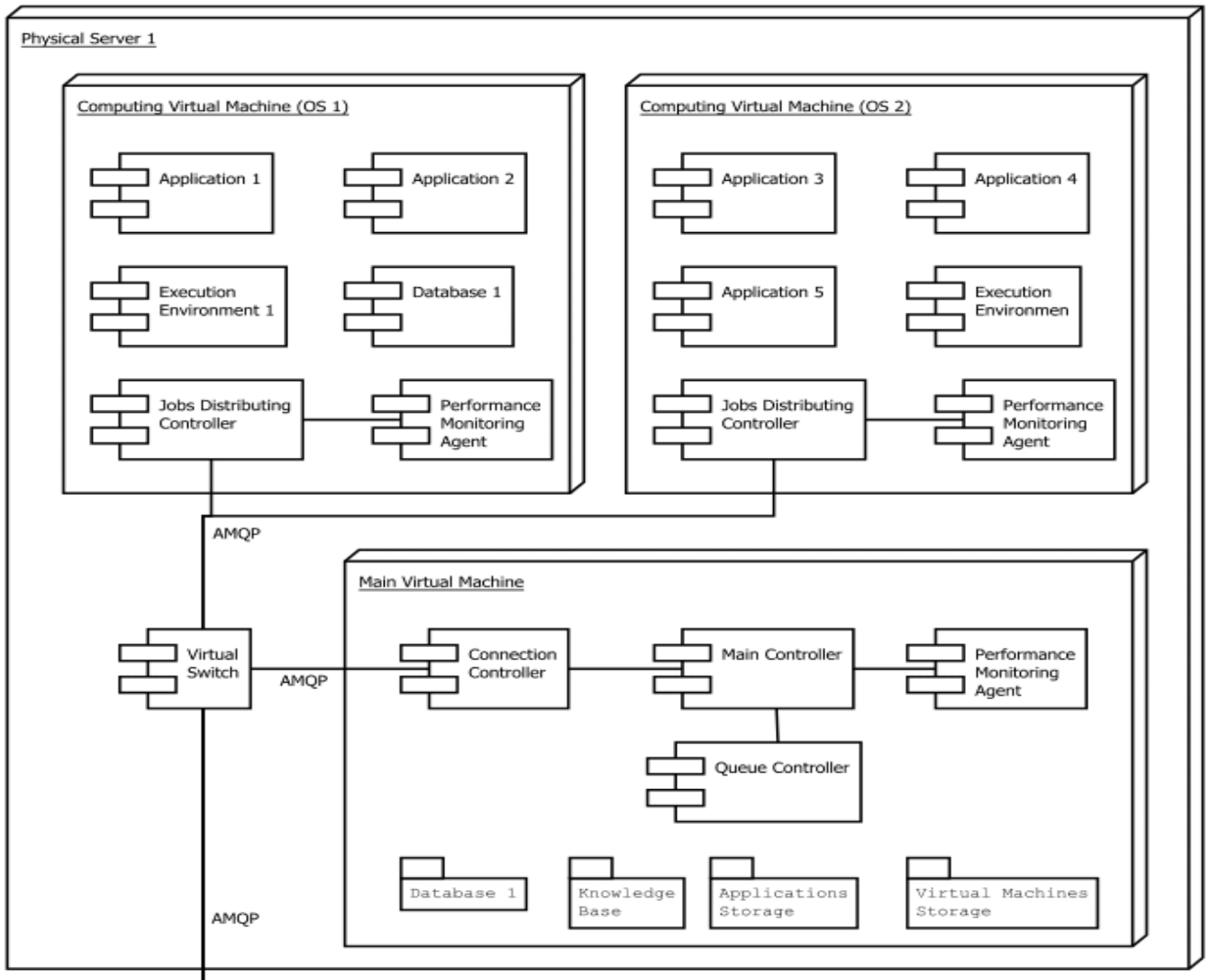
Basic architecture of the system: Controller

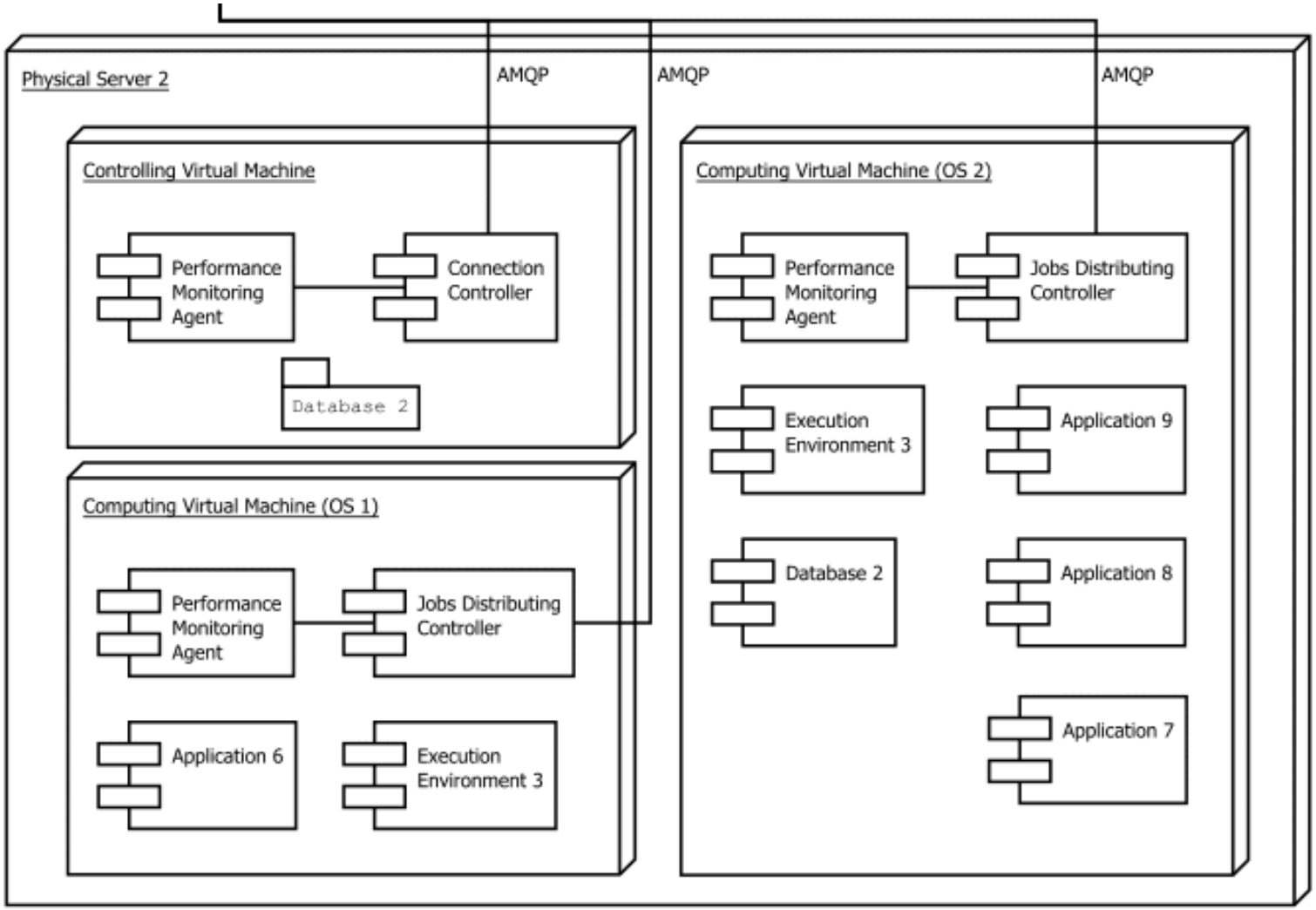
- Receive a job
- Decide on which path a job must take
- Send data to application
- Receive data from application
- Send a result to user

Basic architecture of the system: Optimization

- Monitor performance and available resources
- Search for a path
- Store knowledge about performance for future use

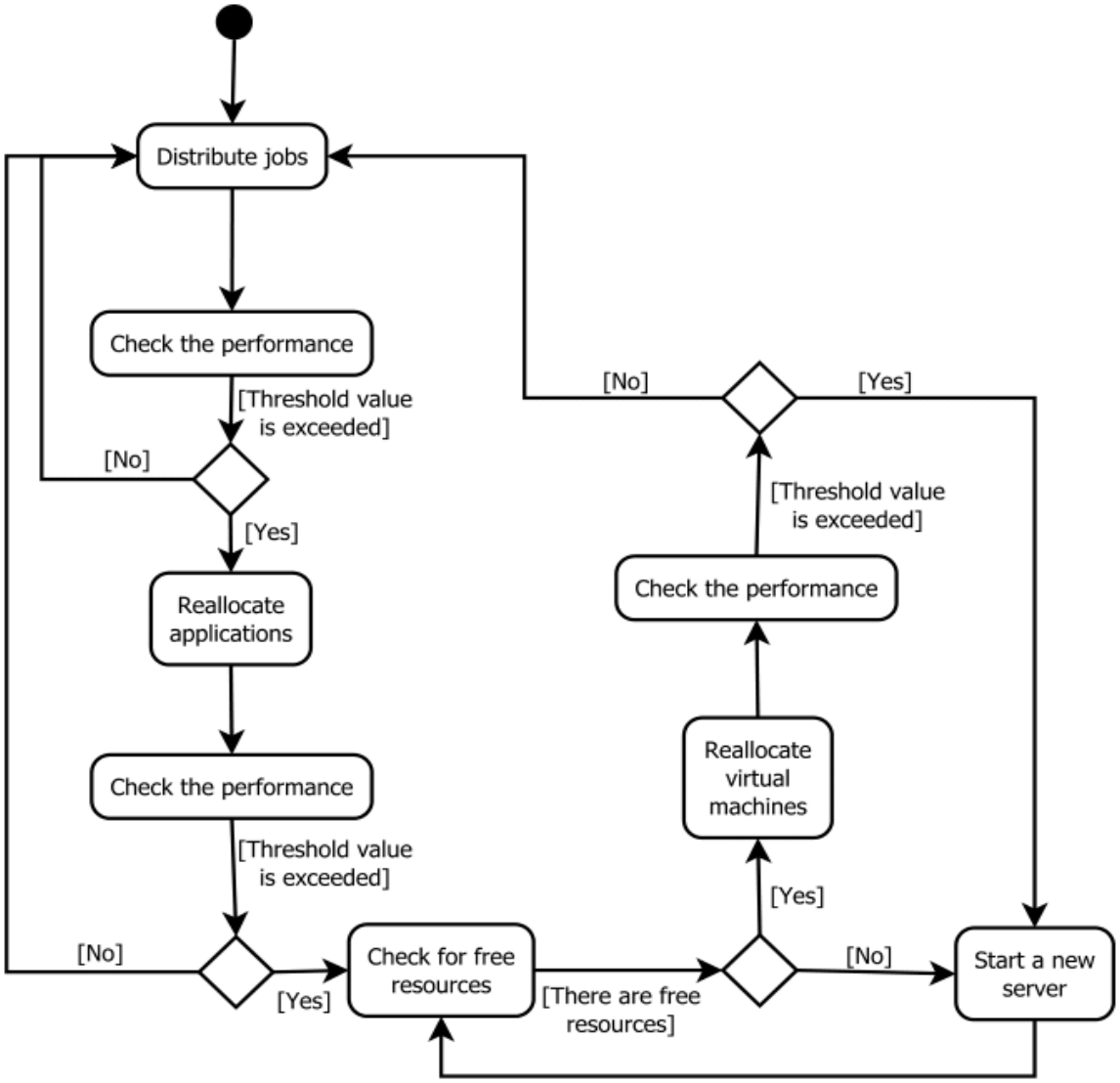
Architecture of the system



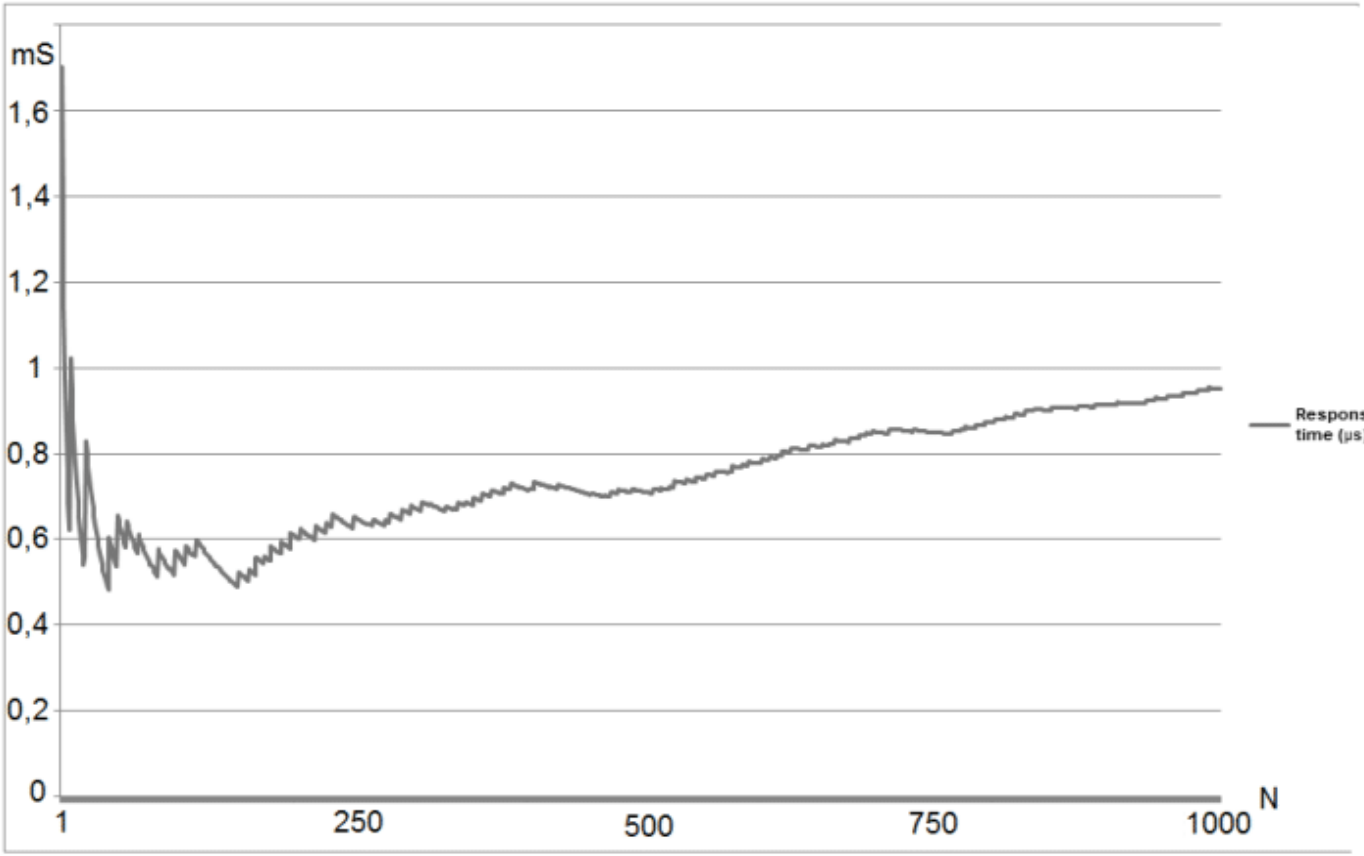

Physical Server 2
AMQP
AMQP
AMQP
Controlling Virtual Machine
Performance Monitoring Agent
Connection Controller
Database 2
Computing Virtual Machine (OS 1)
Performance Monitoring Agent
Jobs Distributing Controller
Application 6
Execution Environment 3
Computing Virtual Machine (OS 2)
Performance Monitoring Agent
Jobs Distributing Controller
Execution Environment 3
Application 9
Database 2
Application 8
Application 7

The algorithm

- Jobs queuing step
- Applications redistributing step
  - Determine application with lowest performance
  - Determine virtual machine with highest performance and maximum free resources
  - Check if there are enough resources
  - Reallocate application
- Migration and creation of virtual machines
- Starting or stopping physical servers

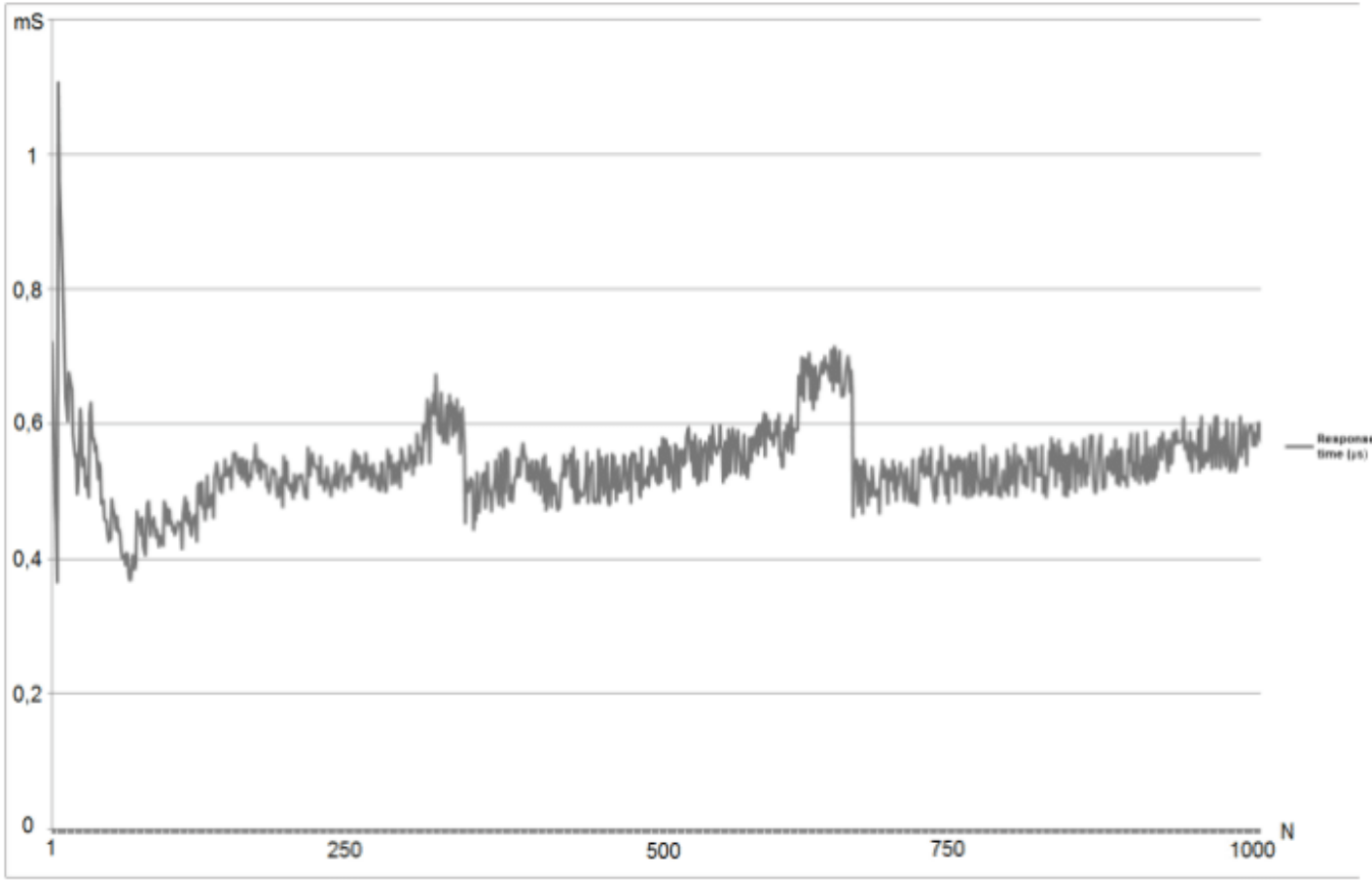

Distribute jobs
Check the performance
[Threshold value is exceeded]
[No]
[Yes]
Reallocate applications
Check the performance
[Threshold value is exceeded]
[No]
[Yes]
Check for free resources
[There are free resources]
[No]
[Yes]
Reallocate virtual machines
Check the performance
[Threshold value is exceeded]
[No]
[Yes]
Start a new server

Experiments results: First step only



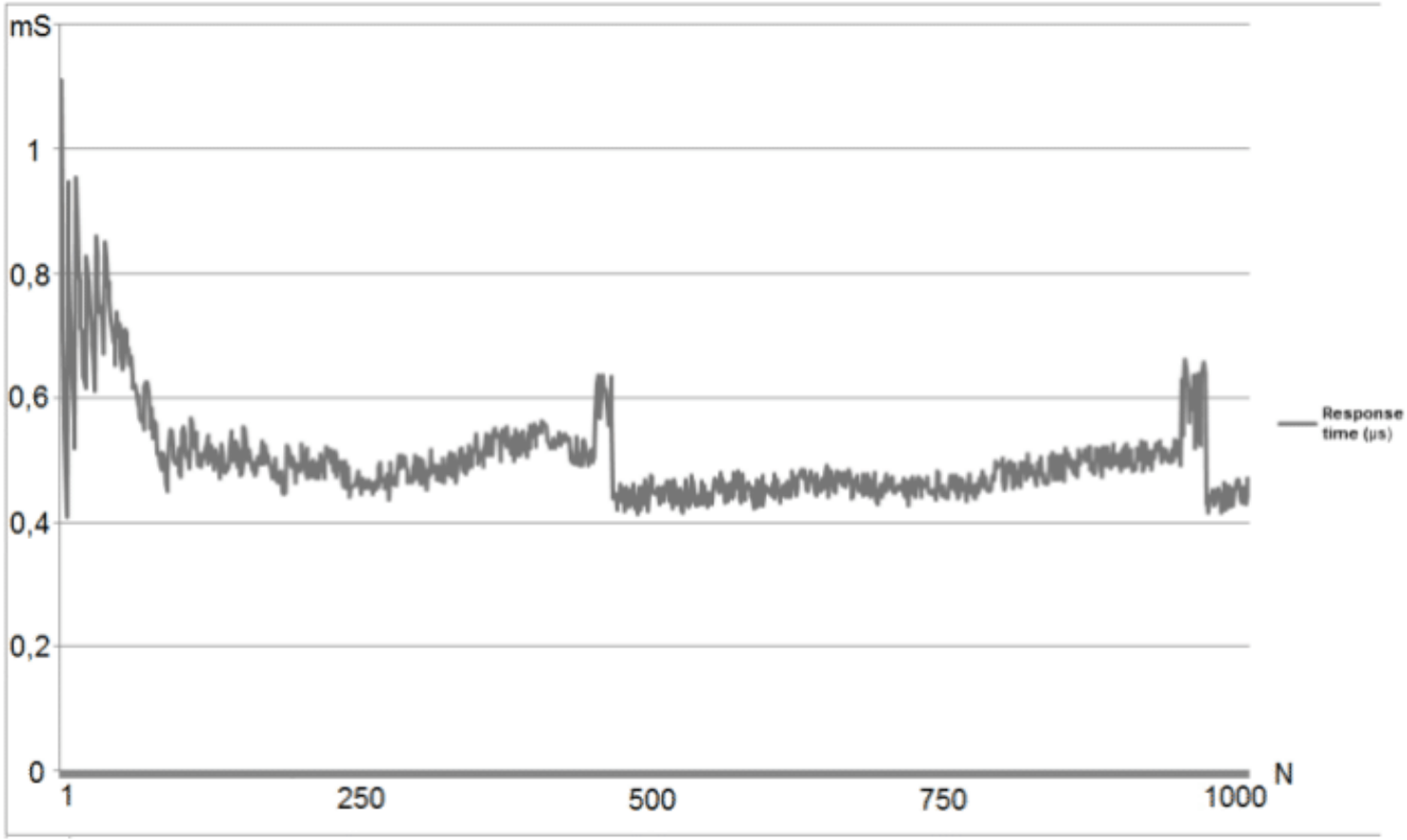
ITMO UNIVERSITY

Experiments results: Second step only



ITMO UNIVERSITY

Experiments results:
First and Second steps together



ITMO UNIVERSITY

Conclusion

- The performance of the system was improved by using two optimization methods.
- Use of a multi-leveled optimization shows its potential.
- Future work lies in implementing two last steps.

Thank you for your attention

ITMO UNIVERSITY

Maksim Khegai, St. Petersburg, Russia, E-mail: MaxHegai@rambler.ru