



SAPIENZA
UNIVERSITÀ DI ROMA

Department of Computer, Control, and Management Engineering

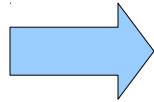
Providing Transaction Class-Based QoS in in-Memory Data Grids Via Machine Learning

Pierangelo Di Sanzo, Francesco Maria Molfese, Diego Rughetti and Bruno Ciciani

DIAG, Sapienza University of Rome

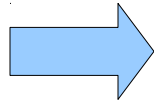
- Cloud computing – Infrastructure as a Service (IAAS)

elastic architectures



users can rapidly change the amount of used (virtual) computing resources

“pay-as-you-go”
pricing model



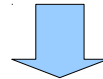
users pay on the basis of the actual resource usage
(e.g. usage time, storage size).

reducing infrastructure cost:

- acquire the minimum amount of resources you need (e.g. for meeting Service Level Agreements with you customers)
- dynamically regulate the amount of acquired resource on the basis of the current workload

Context

- in-memory transactional data grids (IMDGs)
 - distributed in-memory cache (data distributed across multiple servers)
 - no-SQL data model (e.g. key-value)
 - high horizontal scalability
 - easy/fast run-time re-sizing (in terms of number of cache servers)
 - distributed transaction execution support



ideal candidate for building elastic cloud data platforms

Available IMDGs releases:

- Oracle Coherence
- IBM eXtreme Scale
- JBoss Infinispan
- VMware Gemfire
- Terracotta Enterprise Suite
- Gigaspaces XAP Elastic
- Caching Edition
- Hazelcast,
-

IMDGs and QoS

Typical **IMDGs** configuration parameters:

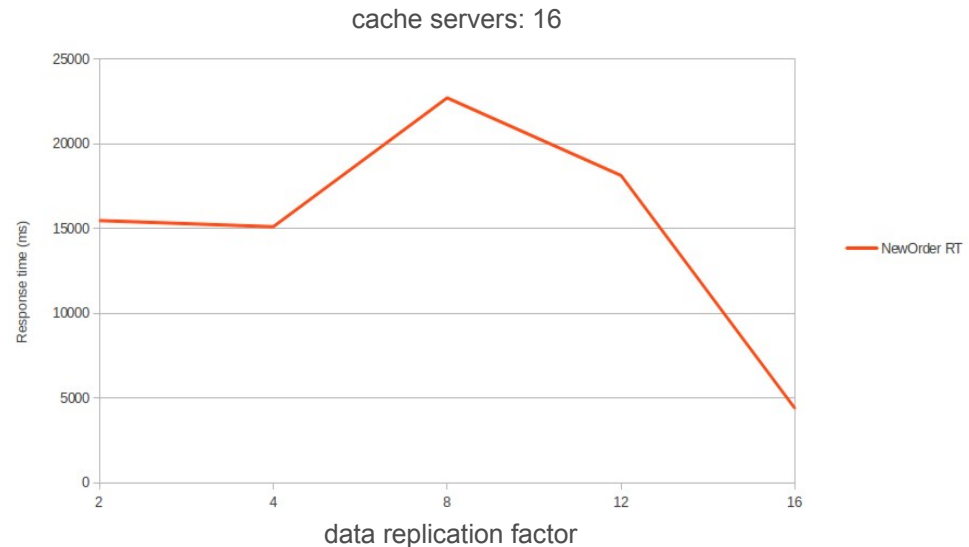
- data platform size (number of cache servers)
- data object replication factor (number of replicas of data objects)
- concurrency level (number of concurrent threads for executing transactions)
-

IMDGs workload profile parameters:

- transaction arrival rate
- transaction mix

QoS in transactional applications:

- transaction response times/throughput

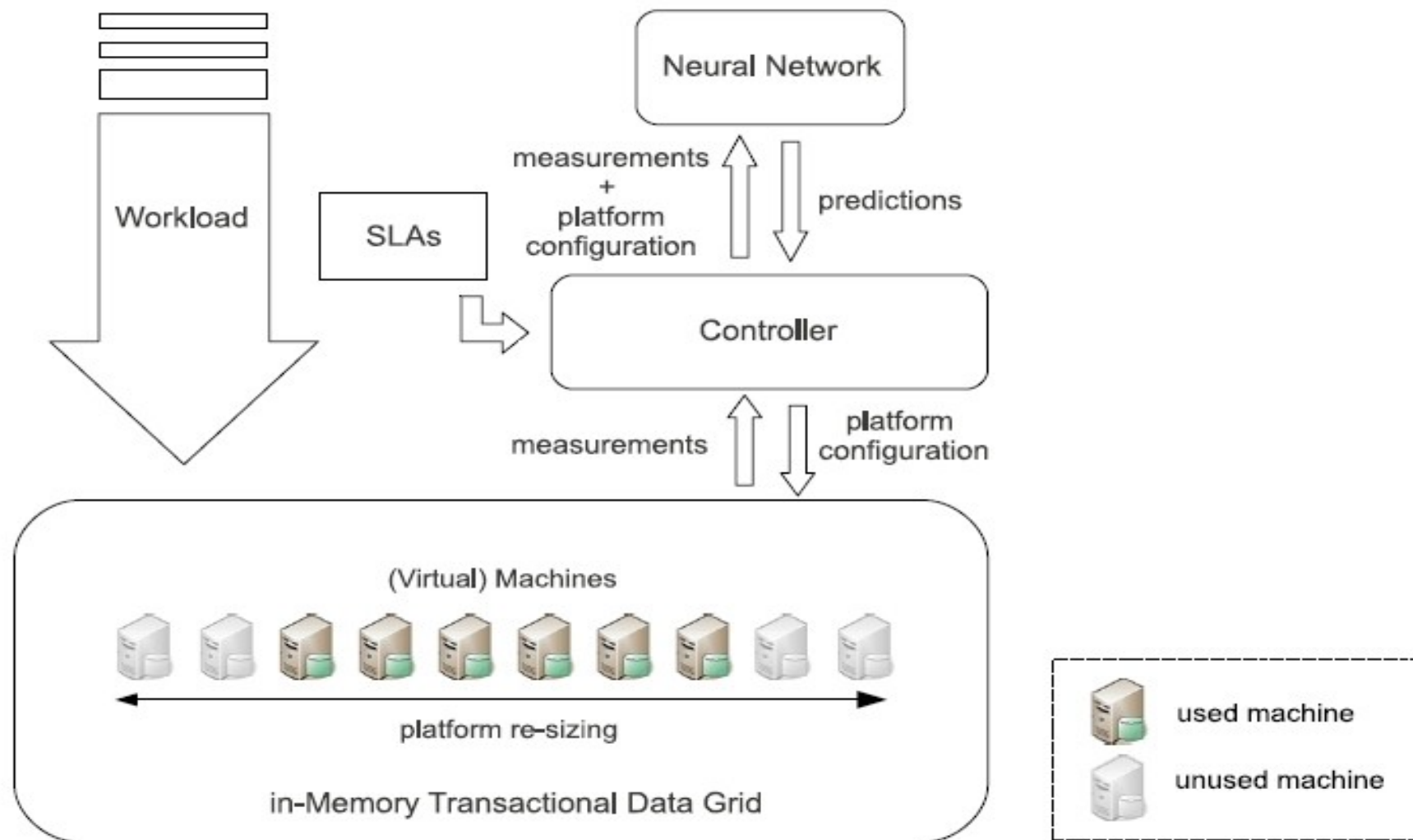


non-linear dependences

Automatic re- sizing/configuration of in-memory transactional data grids deployed on top of elastic cloud infrastructures

Goal: meeting transaction class-based Service Level Agreements (SLAs) using the minimal amount of resources (virtual machines)

Architecture



Performance Prediction Scheme

A neural network-based performance prediction scheme provides estimated **response times** of all transaction classes depending on:

- the current workload profile of the application, identified by

λ (overall transaction arrival rate)

f_i (fraction of class i transactions)

- a given IMDG configuration, identified by

n (number of active cache servers)

g (data object replication factor)

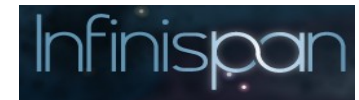
k (number of concurrent threads)

Neural network function: $(r_1, \dots, r_c) = f(n, g, k, \lambda, f_1, \dots, f_c)$

The neural network is trained in-advance.

Experimental Setting

- Future Grid IaaS Cloud
 - cluster with 20 machines (Intel Xeon X5570 @2.93 GHz quad-core processor, 2 GB RAM, Centos 5.7 operating system)
- Jboss Infinispan (open source IMDG by Red Hat)
- Implementation of TPC-C Benchmark
 - online product store
 - users execute orders and payments
 - five transaction classes:
 - *entering new orders (New-Order transaction)*
 - *recording payments (Payment transaction)*
 - *checking the status of orders (Order-Status transaction)*
 - *delivering orders (Delivery transaction)*
 - *checking the level of product stock (Stock-Level transaction)*
- Neural network implementation
 - *FANN open-source libraries, version 2.2.0 (acyclic feed-forward full connected neural network)*



TPC BENCHMARK™ C

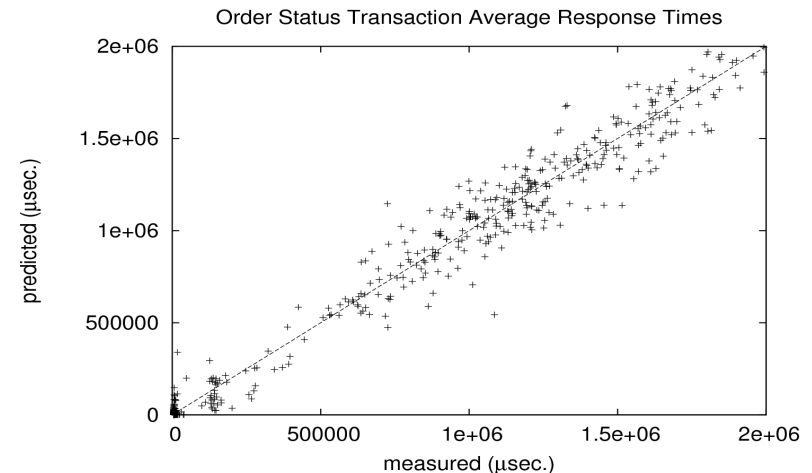
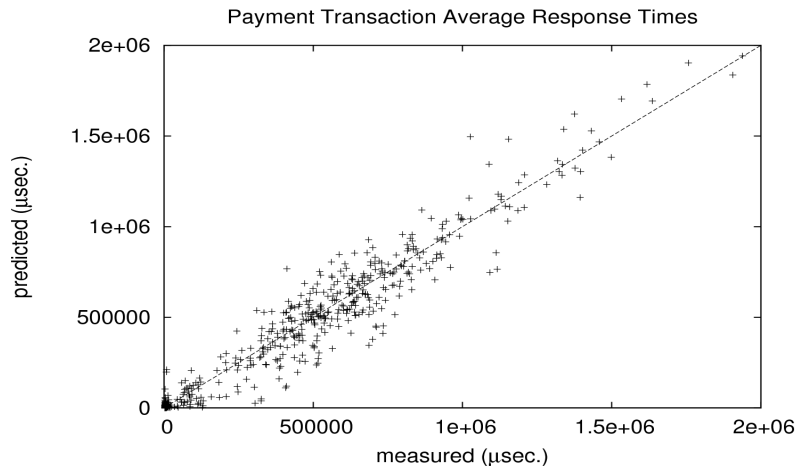
FANN
Fast Artificial Neural Network Library

Parameter setting

- $n^{max} = 20$
- $m^{max} = 1 \text{ GB}$
- $k^{max} = 16$
- f_1 (New-Order transaction) between 0.2 and 0.6
- f_2 (Payment transaction) between 0.2 and 0.6
- f_3 (Delivery transaction) between 0.02 and 0.2
- f_4 (Order-Status) between 0.02 and 0.2
- f_5 (Stock-Level transaction) between 0.02 and 0.2
- total size of data objects (excluding replicas) = 1,066GB

Neural network training

- 200 samples gathered while randomly varying workload profile and IMDG configuration (parameters n, m, k, f_1, \dots, f_5)
- average transaction response time prediction error calculated with respect 600 samples:
 - *New-Order transaction* 20%,
 - *Payment transaction* 12%
 - *Order-Status transaction* 13%
 - *Delivery transaction* 19%
 - *Stock-Level transaction* 11%



Test Case 1

initial configuration

n	2
g	1
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	535MB

SLAs

r_1^{max}	0.5 sec
r_2^{max}	0.5 sec
r_3^{max}	0.5 sec
r_4^{max}	0.5 sec
r_5^{max}	0.5 sec



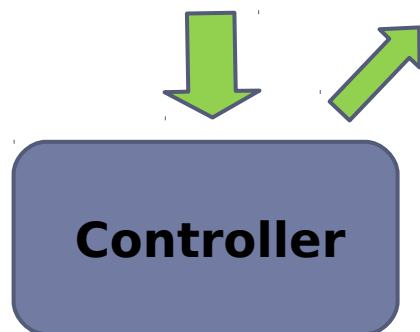
Controller

Test Case 1

initial configuration	
n	2
g	1
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	535MB

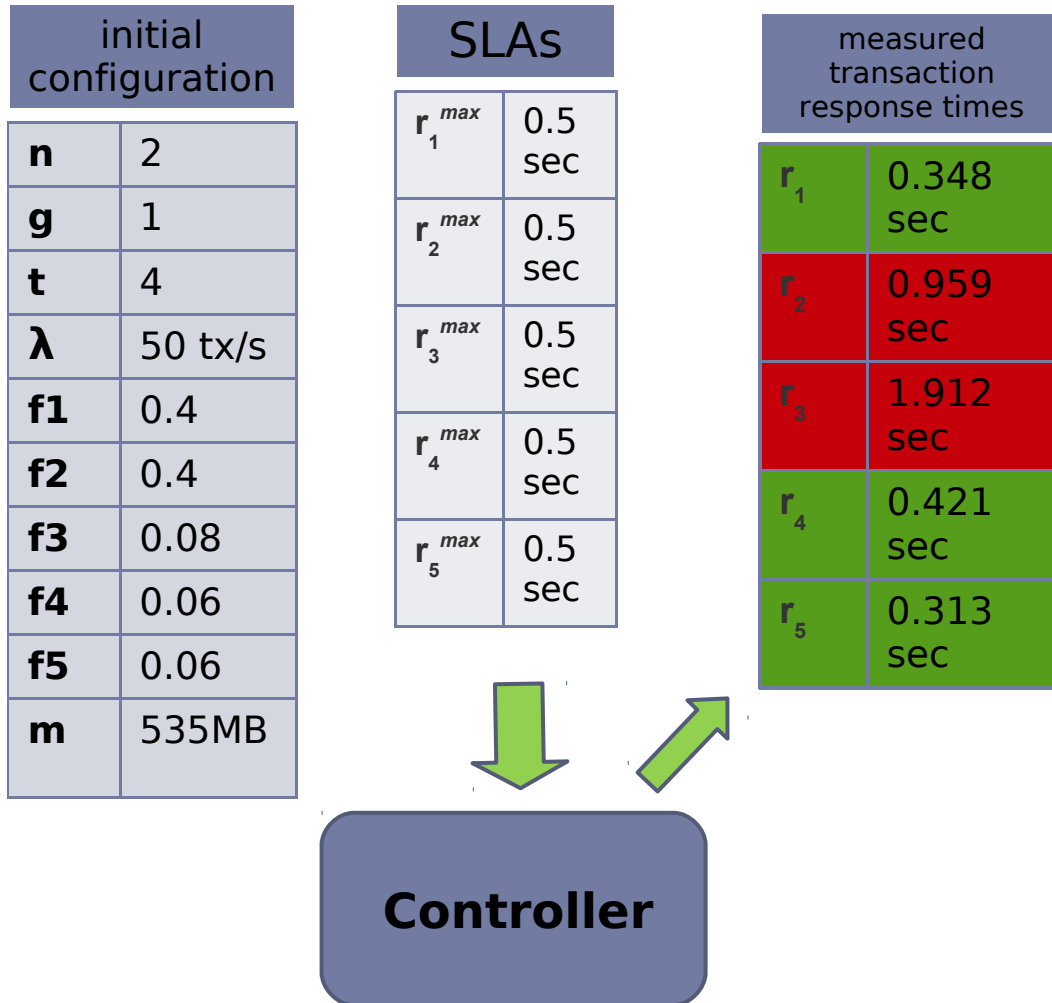
SLAs	
r_1^{max}	0.5 sec
r_2^{max}	0.5 sec
r_3^{max}	0.5 sec
r_4^{max}	0.5 sec
r_5^{max}	0.5 sec

measured transaction response times	
r_1	0.348 sec
r_2	0.959 sec
r_3	1.912 sec
r_4	0.421 sec
r_5	0.313 sec



Controller

Test Case 1



Test Case 1

initial configuration

n	2
g	1
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	535MB

SLAs

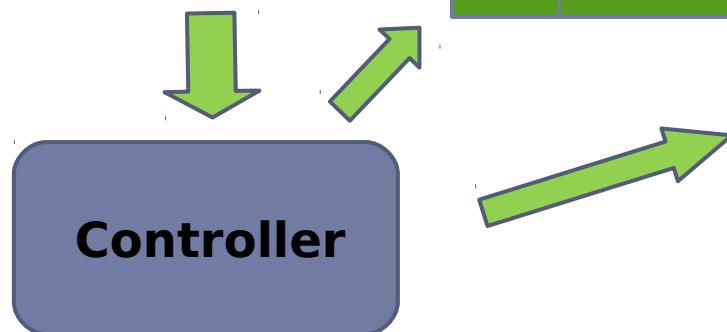
r_1^{max}	0.5 sec
r_2^{max}	0.5 sec
r_3^{max}	0.5 sec
r_4^{max}	0.5 sec
r_5^{max}	0.5 sec

measured transaction response times

r_1	0.348 sec
r_2	0.959 sec
r_3	1.912 sec
r_4	0.421 sec
r_5	0.313 sec

selected configuration

n	7
g	6
t	4



Test Case 1

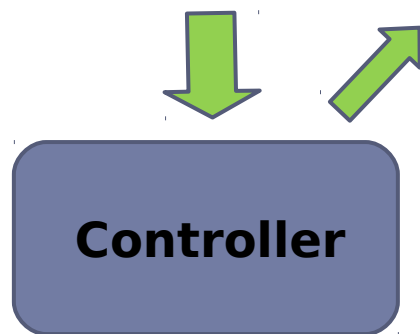
initial configuration	
n	2
g	1
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	535MB

SLAs	
r_1^{max}	0.5 sec
r_2^{max}	0.5 sec
r_3^{max}	0.5 sec
r_4^{max}	0.5 sec
r_5^{max}	0.5 sec

measured transaction response times	
r_1	0.348 sec
r_2	0.959 sec
r_3	1.912 sec
r_4	0.421 sec
r_5	0.313 sec

selected configuration	
n	7
g	6
t	4

measured transaction response times	
r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec



m=913MB

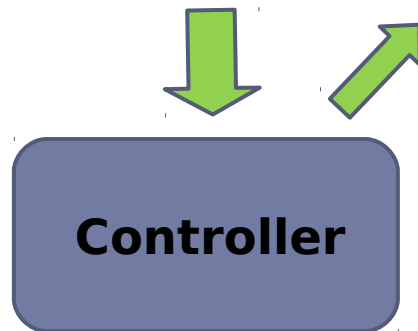
Test Case 1

initial configuration	
n	2
g	1
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	535MB

SLAs	
r_1^{max}	0.5 sec
r_2^{max}	0.5 sec
r_3^{max}	0.5 sec
r_4^{max}	0.5 sec
r_5^{max}	0.5 sec

measured transaction response times	
r_1	0.348 sec
r_2	0.959 sec
r_3	1.912 sec
r_4	0.421 sec
r_5	0.313 sec

- for configurations with $g=n$ (full replication) lower transaction response times were achieved, but the constraint $m < m^{max}$ was violated
- for configurations with $n < 7$ SLAs were violated



selected configuration	
n	7
g	6
t	4

measured transaction response times	
r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec

m=913MB

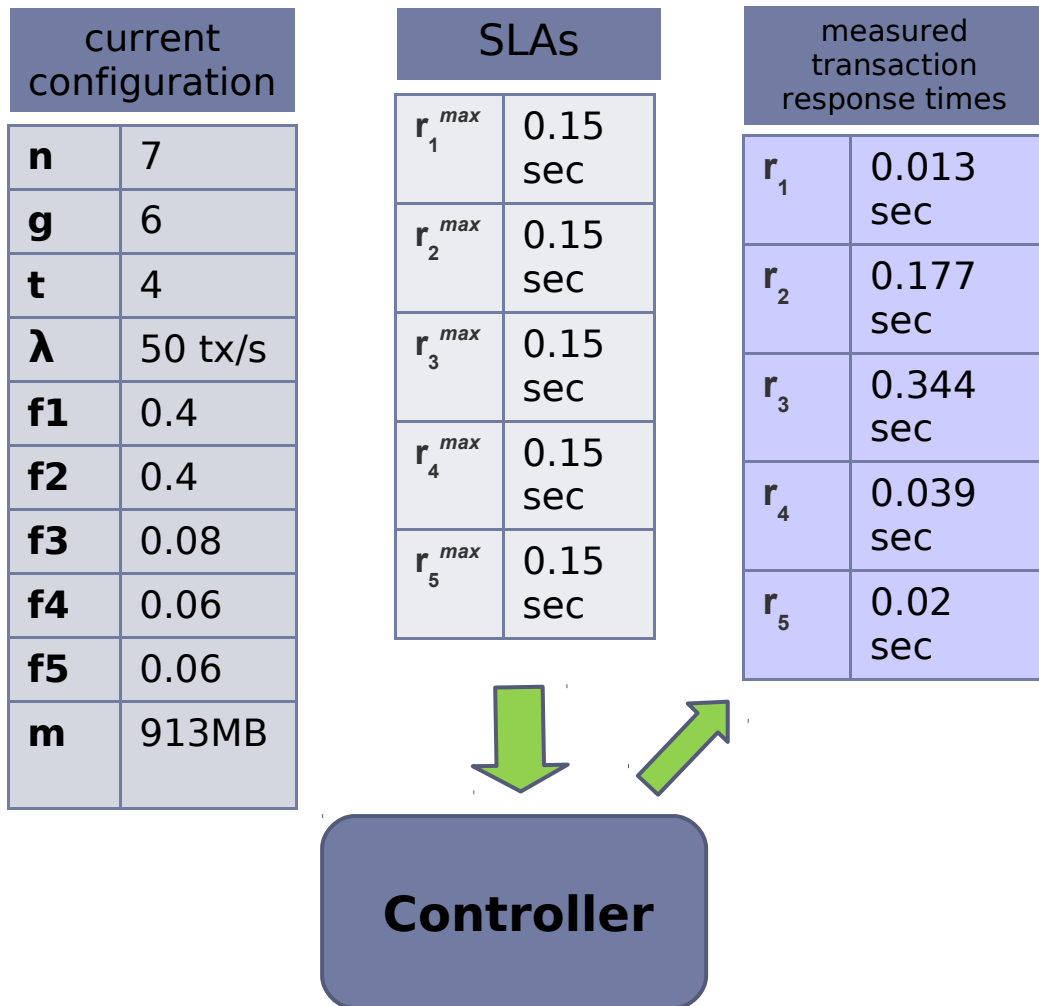
Test Case 2

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec



Test Case 2

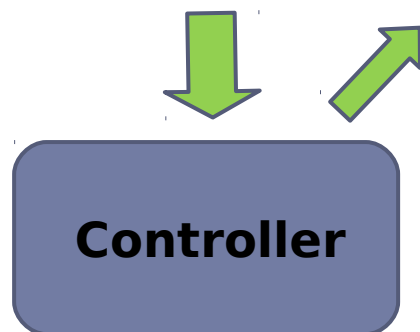


Test Case 2

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec



Test Case 2

current configuration

n	7
g	6
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	913MB

SLAs

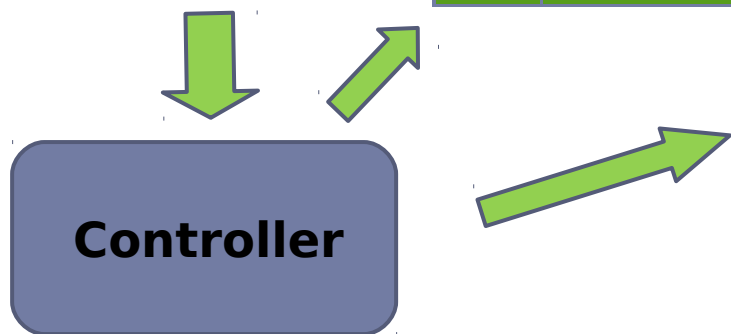
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times

r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec

selected configuration

n	16
g	15
t	6

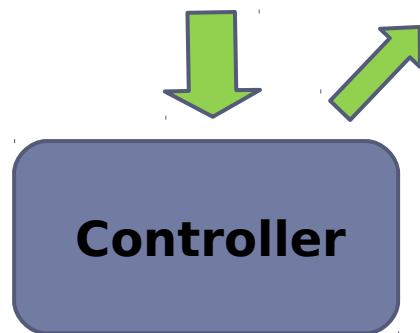


Test Case 2

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec



selected configuration	
n	16
g	15
t	6

measured transaction response times	
r_1	0.012 sec
r_2	0.090 sec
r_3	0.126 sec
r_4	0.016 sec
r_5	0.02 sec

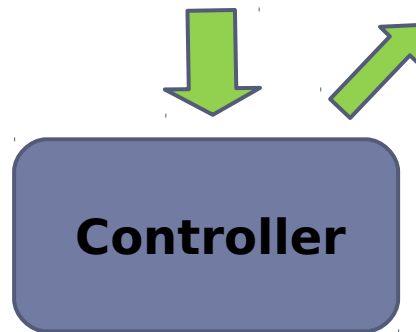
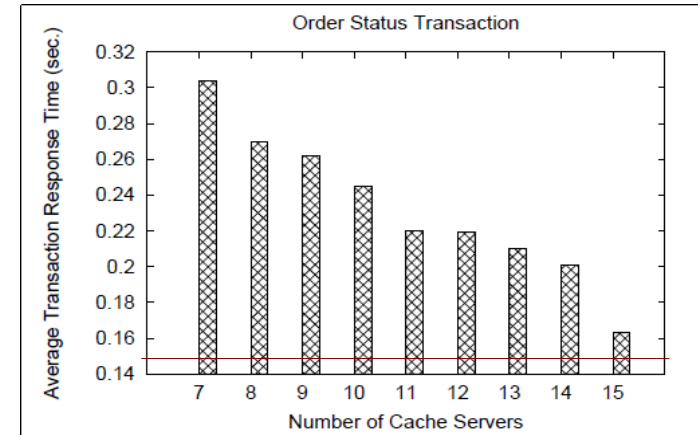
m=998MB

Test Case 2

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.4
f2	0.4
f3	0.08
f4	0.06
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.013 sec
r_2	0.177 sec
r_3	0.344 sec
r_4	0.039 sec
r_5	0.02 sec



selected configuration	
n	16
g	15
t	6

measured transaction response times	
r_1	0.012 sec
r_2	0.090 sec
r_3	0.126 sec
r_4	0.016 sec
r_5	0.02 sec

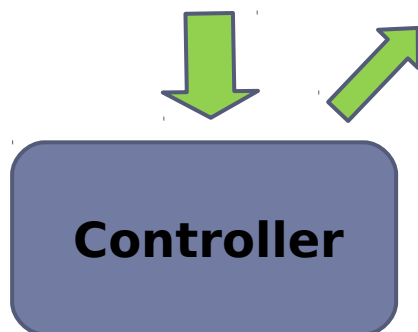
m=998MB

Test Case 3

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.2
f2	0.4
f3	0.18
f4	0.16
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.016 sec
r_2	0.091 sec
r_3	0.152 sec
r_4	0.018 sec
r_5	0.017 sec

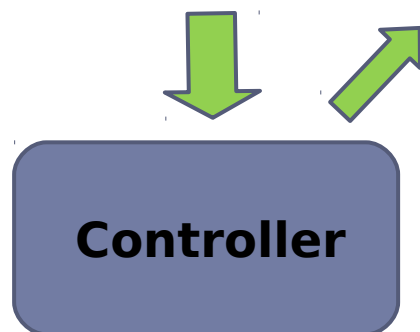


Test Case 3

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.2
f2	0.4
f3	0.18
f4	0.16
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.016 sec
r_2	0.091 sec
r_3	0.152 sec
r_4	0.018 sec
r_5	0.017 sec



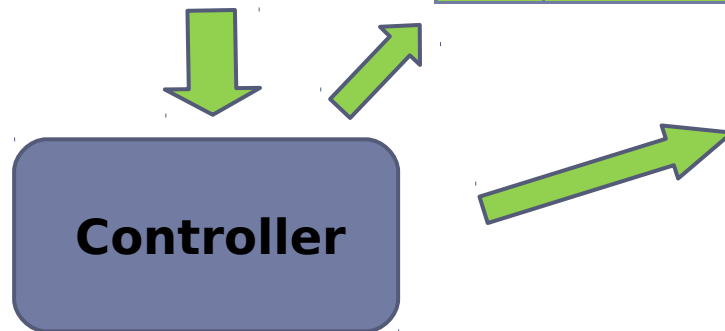
Test Case 3

current configuration	
n	7
g	6
t	4
λ	50 tx/s
f1	0.2
f2	0.4
f3	0.18
f4	0.16
f5	0.06
m	913MB

SLAs	
r_1^{max}	0.15 sec
r_2^{max}	0.15 sec
r_3^{max}	0.15 sec
r_4^{max}	0.15 sec
r_5^{max}	0.15 sec

measured transaction response times	
r_1	0.016 sec
r_2	0.091 sec
r_3	0.152 sec
r_4	0.018 sec
r_5	0.017 sec

- for configurations with $g=n$ (full replication) the constraint $m < m^{max}$ was violated
- for all configurations with $g < n-1$ higher transaction response times were achieved



selected configuration

X

NOT FOUND
no solutions
for the
optimization
problem

Two directions

- other workload profile parameters may change
(e.g. transaction read- and write-set, data access distribution, ...)
 - more input parameters for the neural network

- performing in-advance training of neural networks may be difficult/not possible:
 - incremental training
 - on line neural network accuracy evaluation for triggering new training phases