Bench4Q: A QoS-Oriented E-Commerce Benchmark Tool

ISCAS

2010-9-25

Contents

- TPC-W
- Bench4Q
- What we have done for OW2 contest

Background

- E-commerce
  - a very popular business model
- Benchmark
  - a typical and impartial way to evaluate the performance of e-commerce system
  - E.g. TPC-W

TPC-W

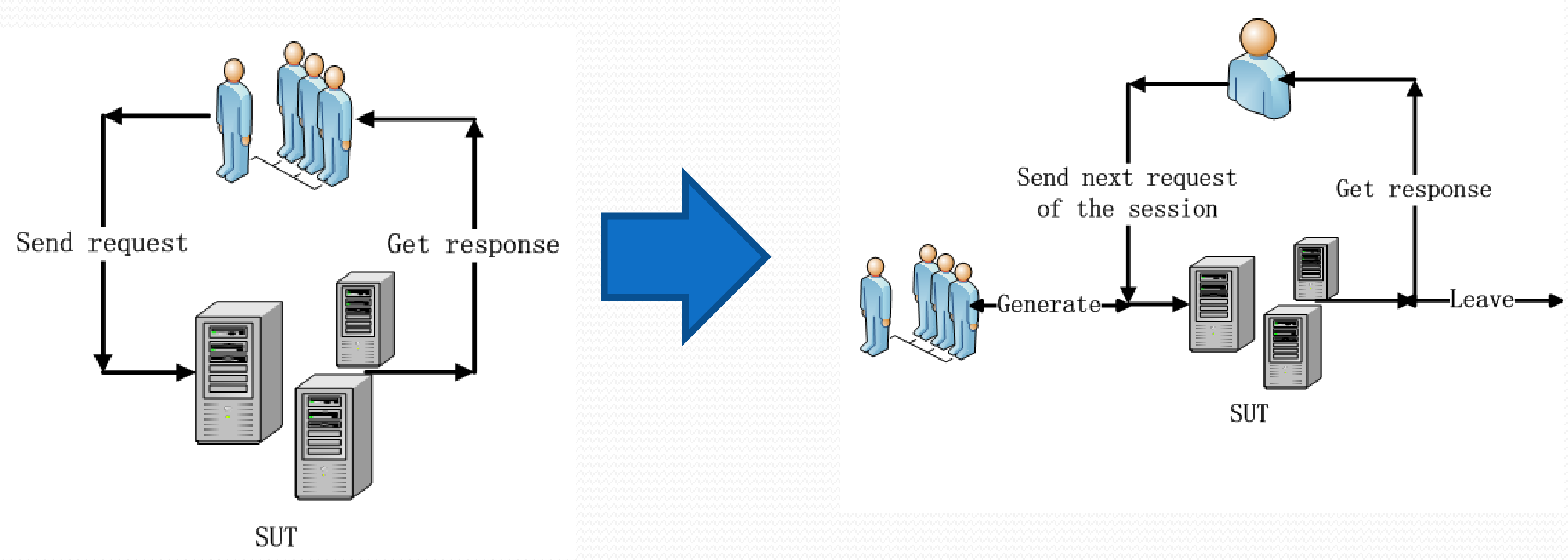
- An on-line book store
- Emulate Browser
  - 14 kinds of web interaction
  - 3 kinds of mix
  - Think time
- Performance metric
  - WIPS web interactions processed per second
  - WIRT web interaction response time

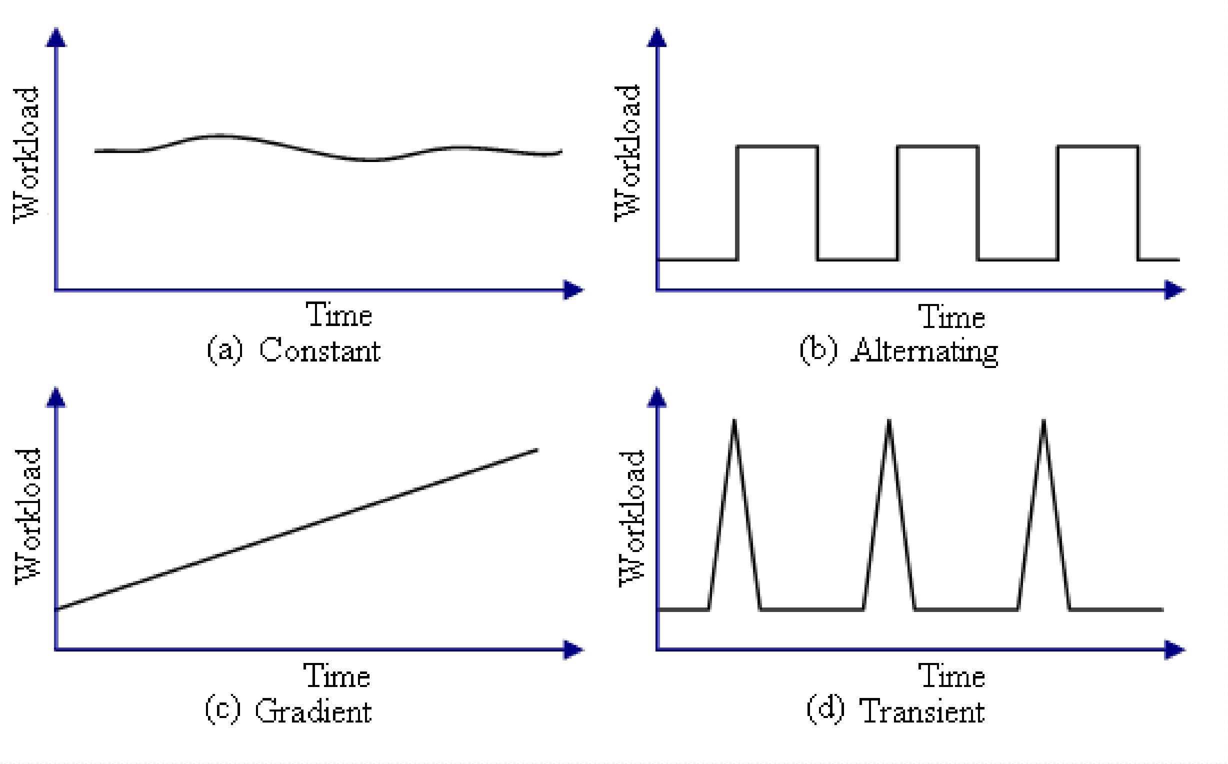
Web Interaction	Browsing mix	Shopping mix	Ordering mix
Browse-related	**95.00**	**80.00**	**50.00**
Home	29.00	16.00	9.12
New products	11.00	5.00	0.46
Best sellers	11.00	5.00	0.46
Product detail	21.00	17.00	12.35
Search request	12.00	20.00	14.53
Search result	11.00	17.00	13.08
Order-related	**5.00**	**20.00**	**50.00**
Shopping cart	2.00	11.60	13.53
Registration	0.82	3.00	12.86
Buy request	0.75	2.60	12.73
Buy confirm	0.69	1.20	10.18
Order inquiry	0.30	0.75	0.25
Order display	0.25	0.66	0.22
Admin request	0.10	0.10	0.12
Admin confirm	0.09	0.09	0.11

Bench4Q vs TPC-W

- Bench4Q implements and extends TPC-W benchmark to a QoS-oriented benchmark
  - Open load simulation
  - QoS-aware users
  - More business metrics

Open mode


Send request
Get response
SUT
Send next request
of the session
Get response
Generate
Leave
SUT



(a) Constant

(b) Alternating

(c) Gradient

(d) Transient

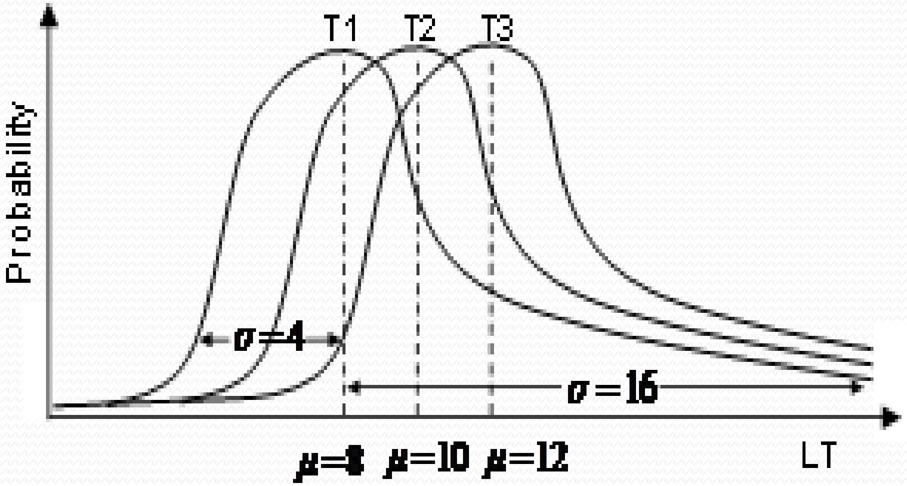
Latency Tolerance

- Latency Tolerance measures the time a customer will wait for a response before change his behavior.

- E.g. for some important interactions, customers usually would like to wait longer than some unimportant interactions.

Latency tolerance and tenacity

$$f(x,\mu,\sigma_1,\sigma_2)=\begin{cases}\dfrac{1}{\sqrt{2\pi}\,\sigma_1}e^{-\frac{(x-\mu)^2}{2\sigma_1^2}}, & -\infty < x \le \mu \\ \dfrac{1}{\sqrt{2\pi}\,\sigma_2}e^{-\frac{(x-\mu)^2}{2\sigma_2^2}}, & \mu < x < \infty\end{cases}$$



Cherkasova, L., Fu, Y., Tang, W., Vahdat, A., 2003. Measuring and characterizing end-to-end internet service performance. ACM Transactions on Internet Technology, 3(4), pp. 347-391.

Definition of tenacity for online behavior

μ

Class	μ	Interactions
Important	12	Buy Request, Buy Confirm, Admin Request, Admin Confirm
Less important	10	Shopping cart, Registration
Unimportant	**8**	Others

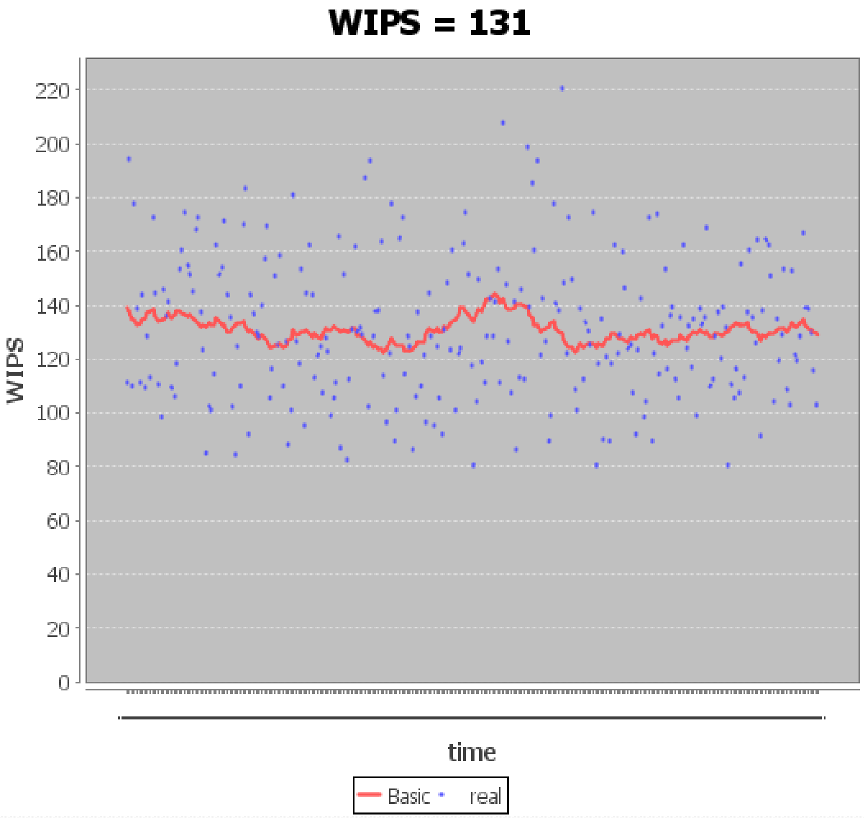
Business metrics

- Metrics of TPC-W
  - WIPS
  - WIRT
- Is that enough to describe performances?
- let's do an experiment!

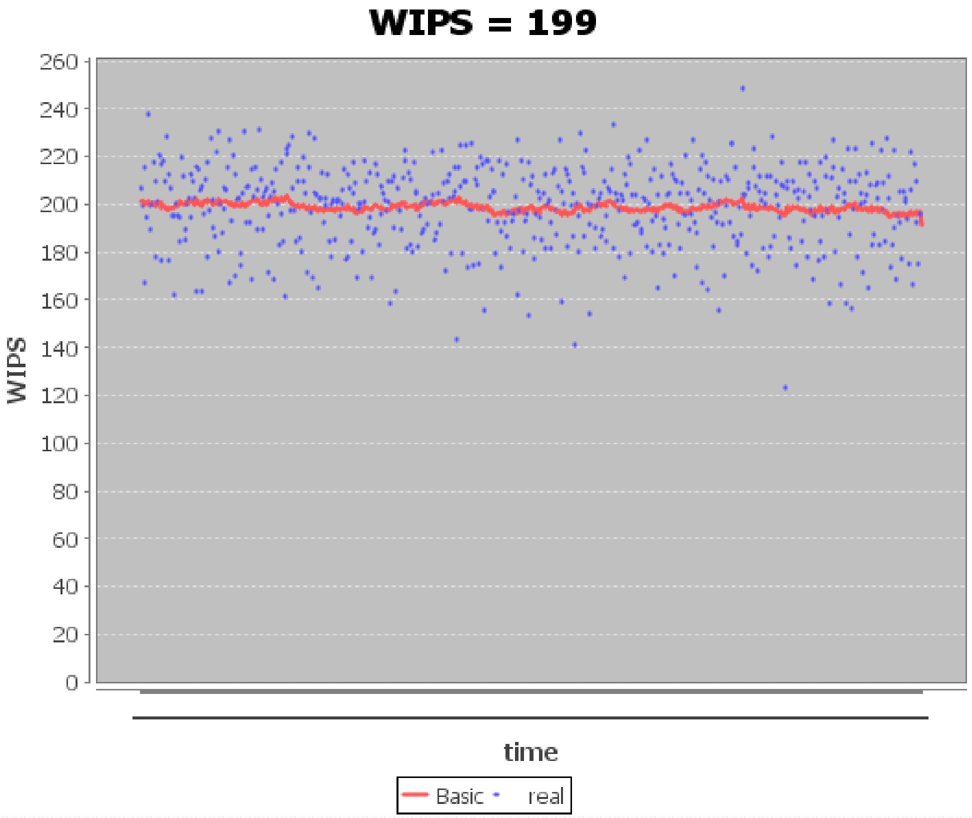
Preliminaries

- In the same conditions, including hardware and OS, WIPS can be improved by optimizing parameters below in Tomcat.
  - sessionTimeout
  - connectionTimeout
  - acceptCount
- Environment
  - DBMSs: IBM DB2 V95
  - OSs: Microsoft Windows 2003 server
  - HTTP Servers: Tomcat 6.x
  - CPU : P4 2.8G * 2
  - Memory : 1G
  - EB : 500 and no think time

Normal VS Optimized



A: normal

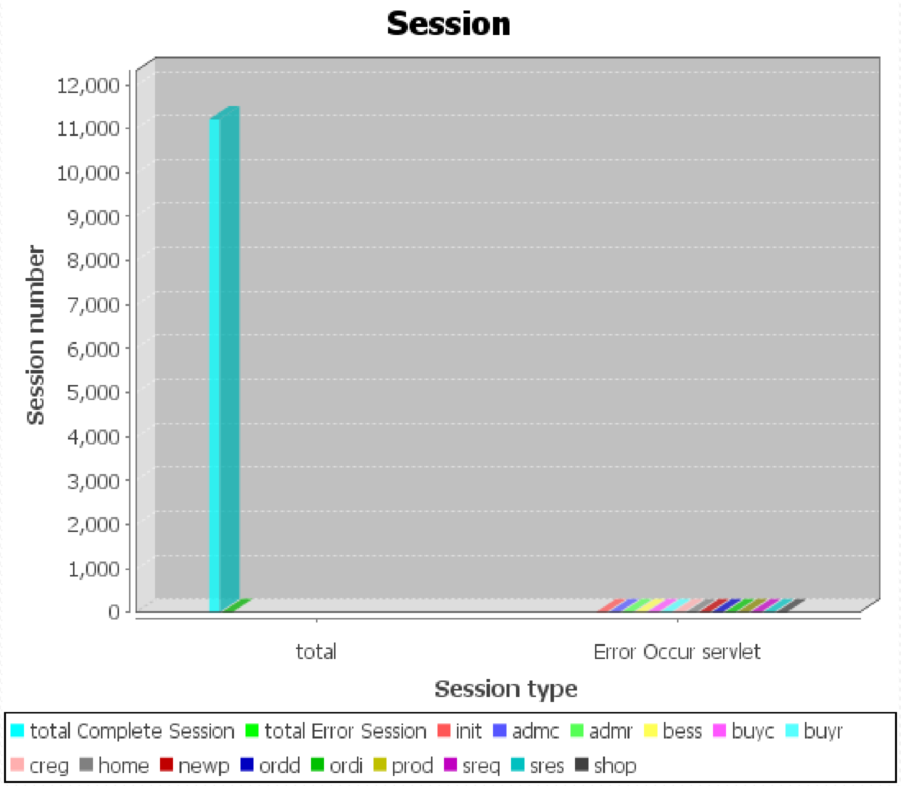


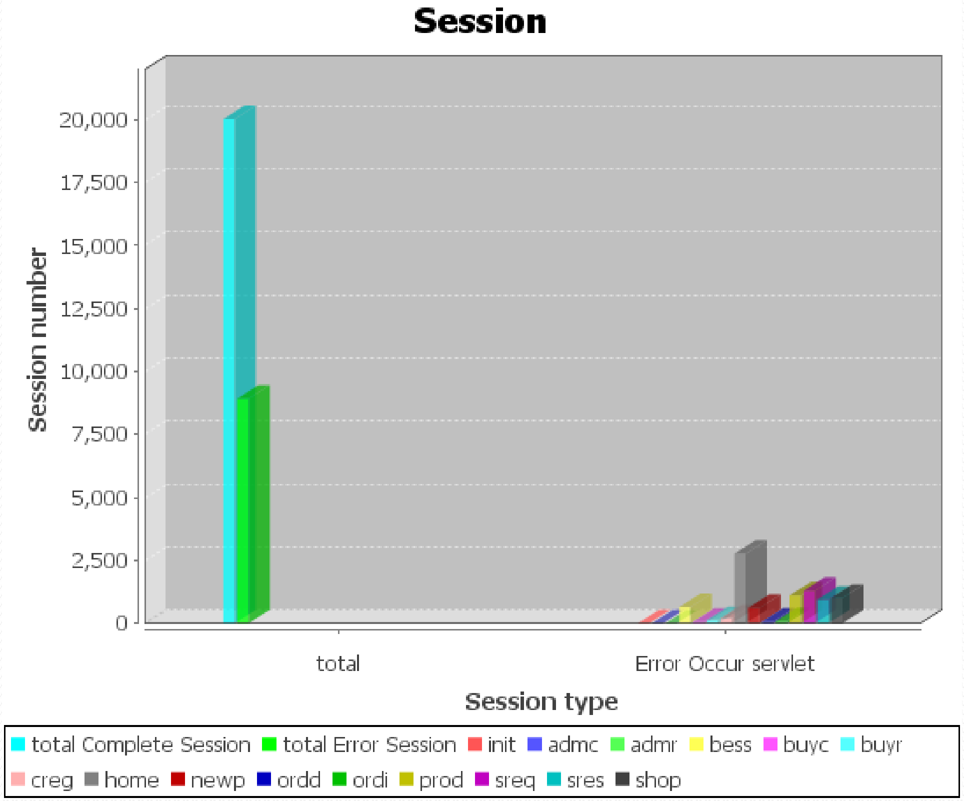
B: unrealistic optimized

Illusion

- Obviously, B is better than A.
- Vendor could improve the performance by modifying the parameters when the other conditions are the same.
- And users may choose B because of high performance.

Seeing is believing?





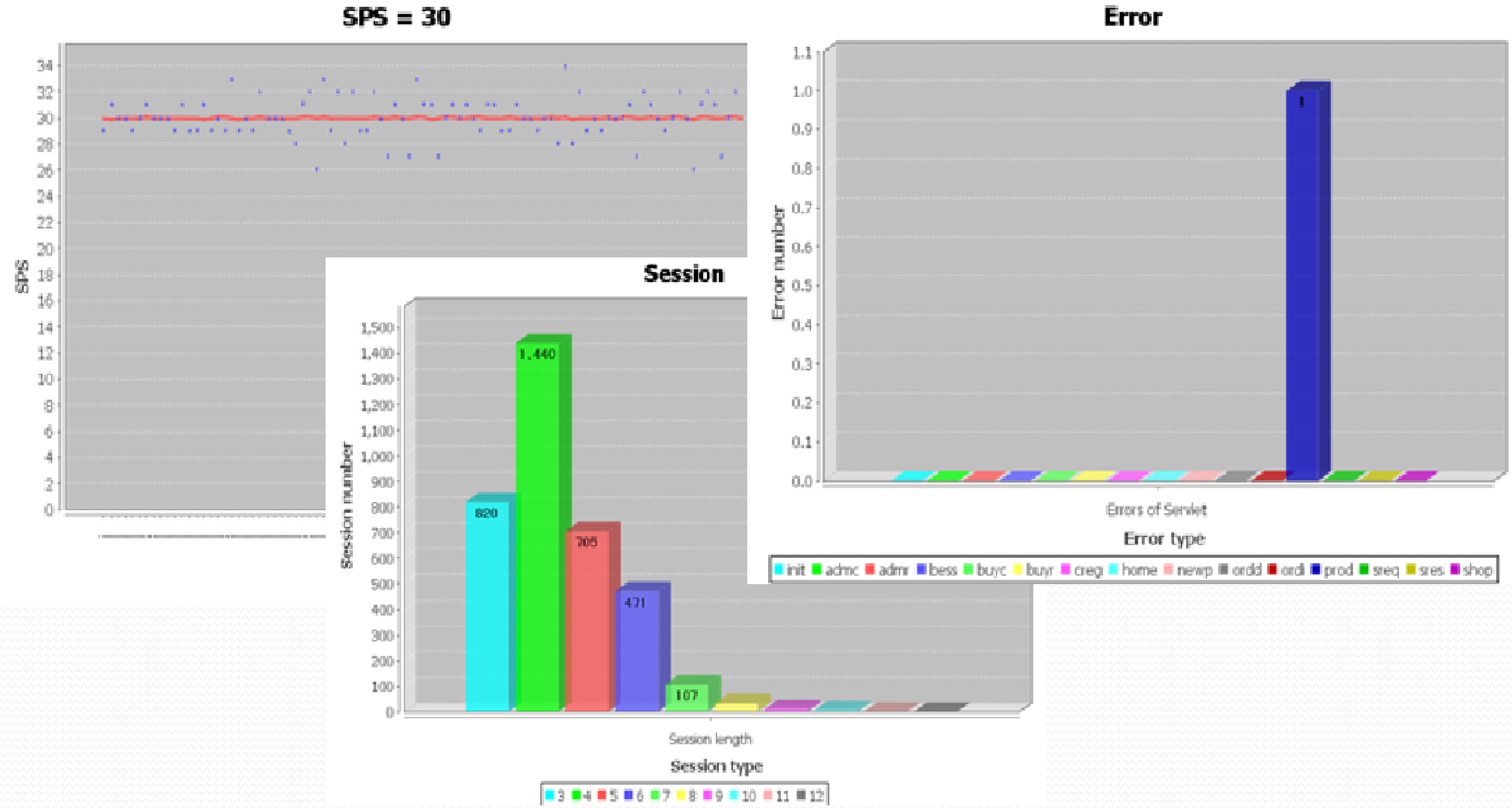
A: normal

B: unrealistic optimized

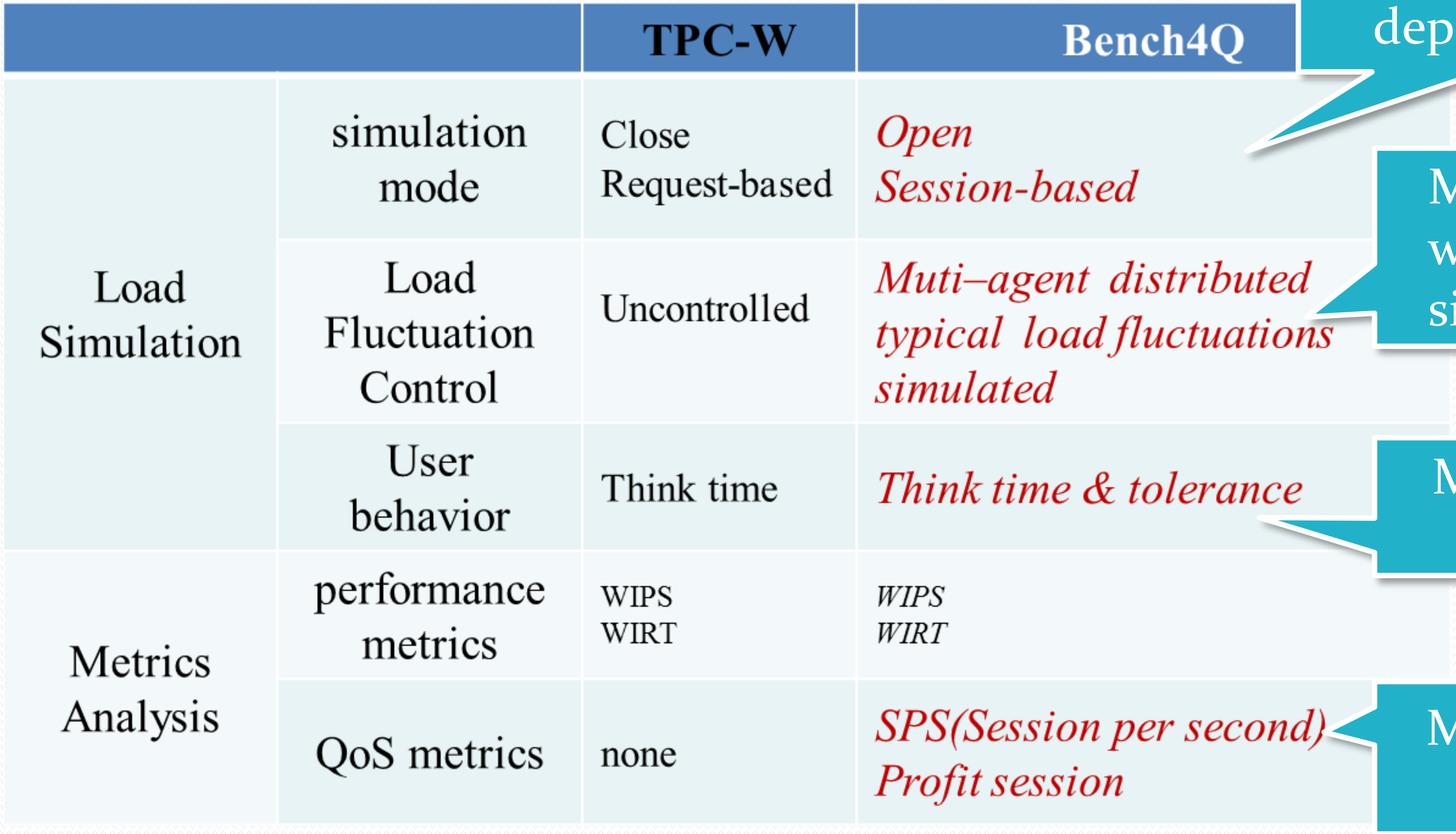
Real phenomenon

- It's surprised that A, may be, is better than B, because the completed sessions in A almost 11000, but which in B just 10000.

QoS metrics


SPS = 30
SPS
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
Session
Session number
1,500
1,400
1,300
1,200
1,100
1,000
900
800
700
600
500
400
300
200
100
0
820
1,440
705
471
107
Session length
Session type
3
4
5
6
7
8
9
10
11
12
Error
Error number
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
Errors of Servlet
Error type
init
admc
admr
bess
buyc
buyr
creg
home
newp
ordd
ordi
prod
sreq
sres
shop

Bench4Q vs TPC-W



		TPC-W	Bench4Q
Load Simulation	simulation mode	Close Request-based	*Open* *Session-based*
	Load Fluctuation Control	Uncontrolled	*Muti–agent distributed typical load fluctuations simulated*
	User behavior	Think time	*Think time & tolerance*
Metrics Analysis	performance metrics	WIPS WIRT	*WIPS* *WIRT*
	QoS metrics	none	*SPS(Session per second)* *Profit session*

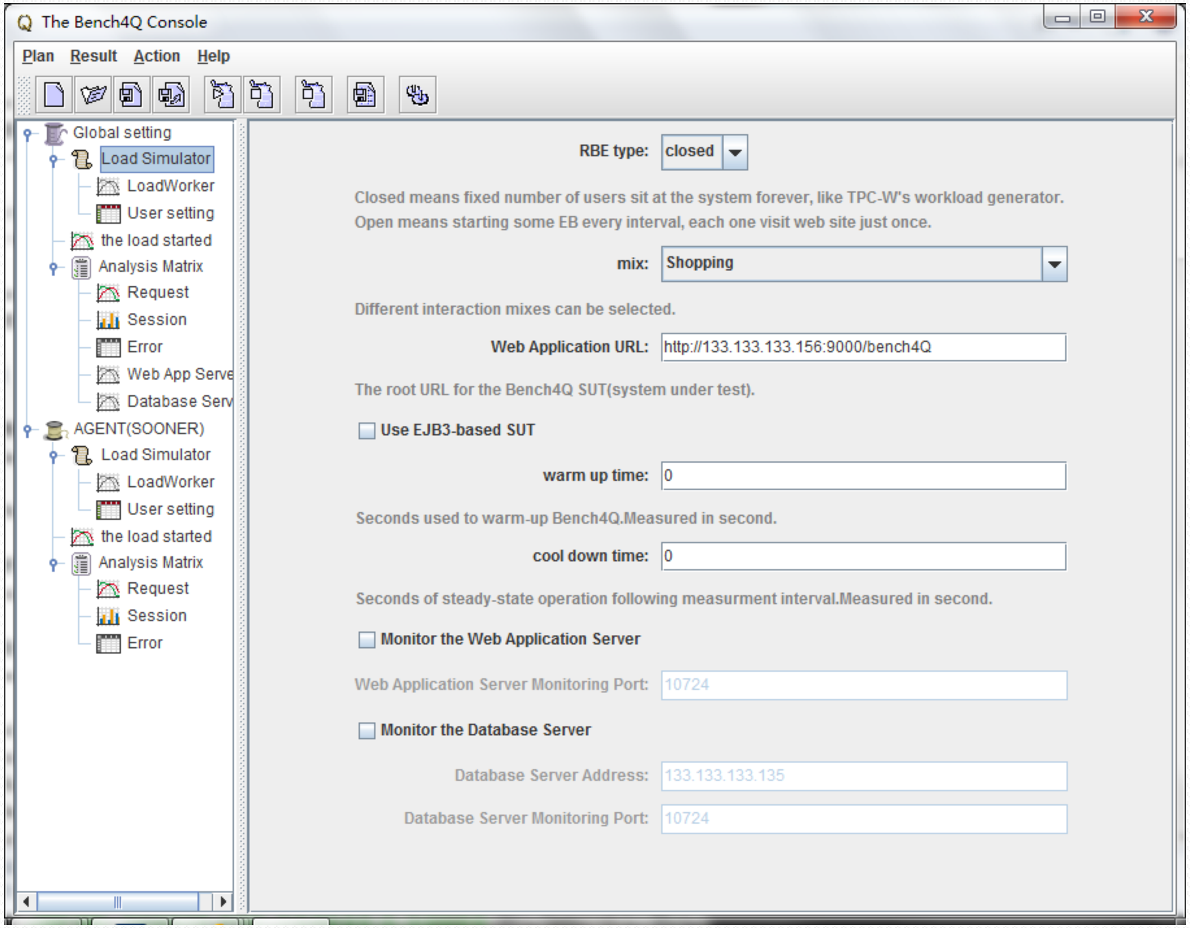
Break session dependency

More flexible workload simulation

More realistic workload

More effective metrics

Bench4Q tool



RBE type: closed

Closed means fixed number of users sit at the system forever, like TPC-W's workload generator.
Open means starting some EB every interval, each one visit web site just once.

mix: Shopping

Different interaction mixes can be selected.

Web Application URL: http://133.133.133.156:9000/bench4Q

The root URL for the Bench4Q SUT(system under test).

□ Use EJB3-based SUT

warm up time: 0

Seconds used to warm-up Bench4Q.Measured in second.

cool down time: 0

Seconds of steady-state operation following measurment interval.Measured in second.

□ Monitor the Web Application Server

Web Application Server Monitoring Port: 10724

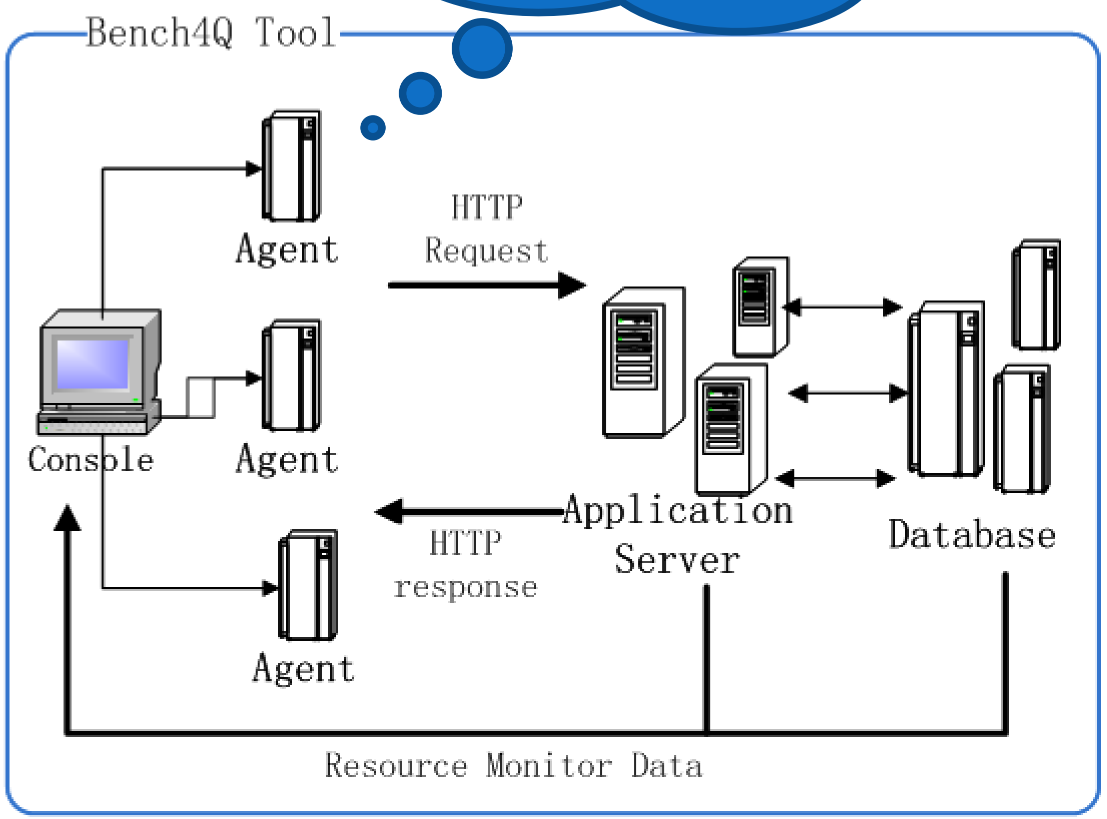
□ Monitor the Database Server

Database Server Address: 133.133.133.135

Database Server Monitoring Port: 10724

Architecture of Tool

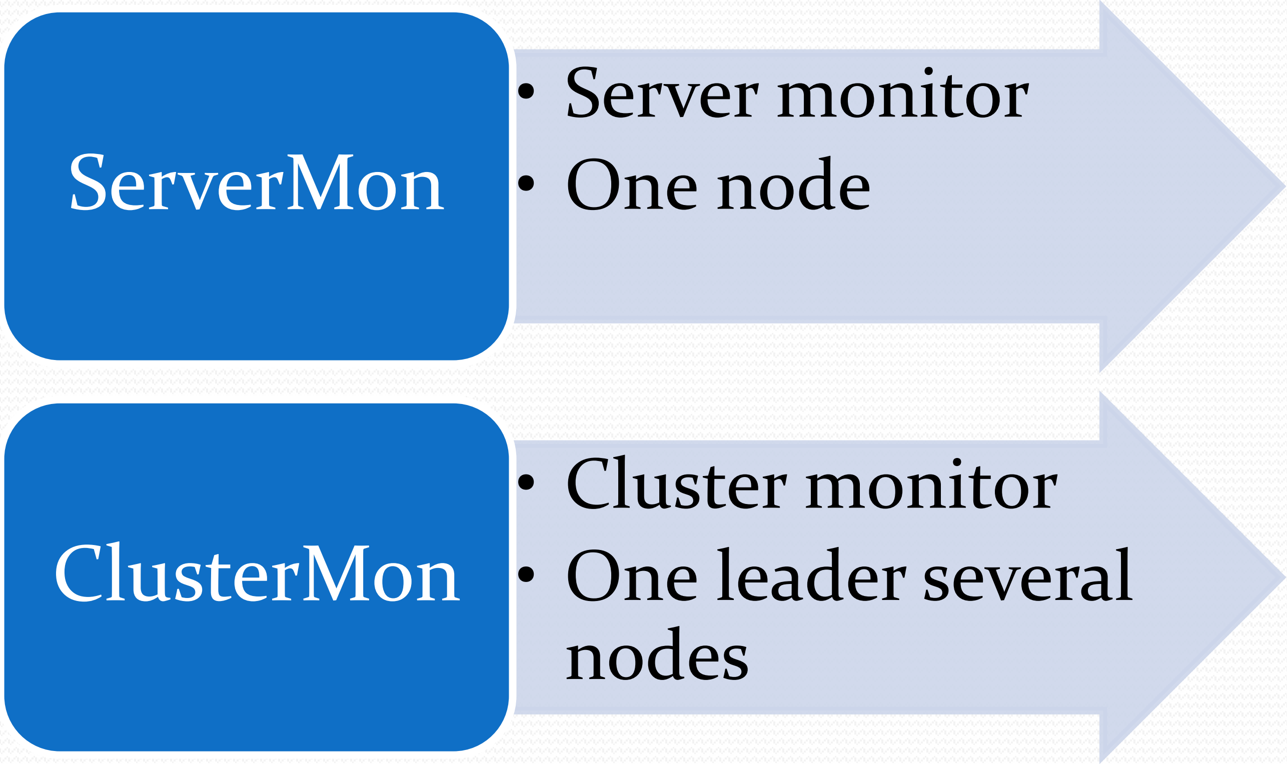
large scale workload



What we have done for OW2 contest

- Monitor the resource of servers
  - CPU usage
  - Free memory
  - Disk write & read
  - Network sent & received
- Workload recording
- Result comparison

Resource Monitor



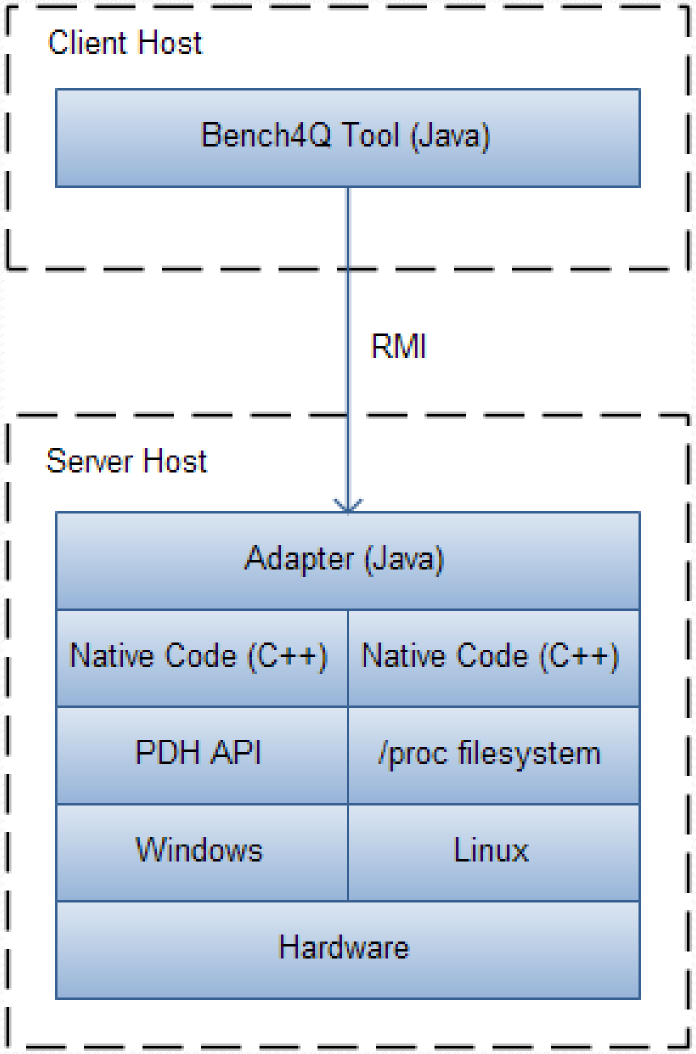
ServerMon

- Server monitor
- One node

ClusterMon

- Cluster monitor
- One leader several nodes

ServerMon



Client Host

Bench4Q Tool (Java)

RMI

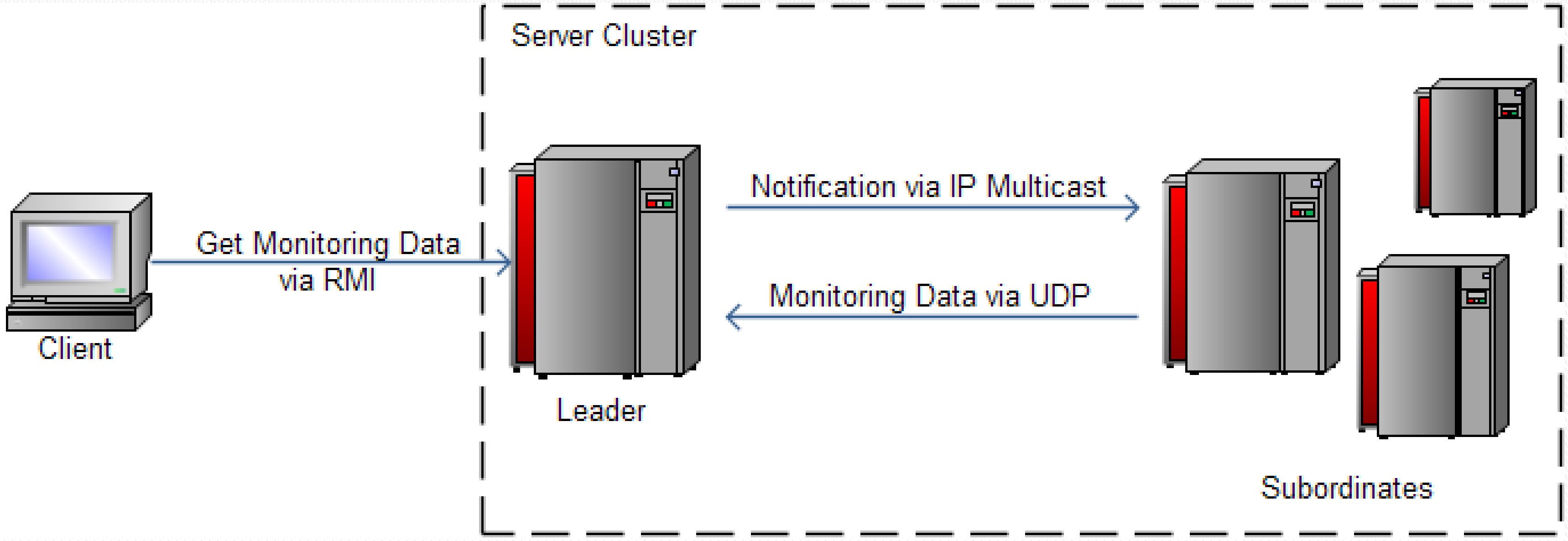
Server Host

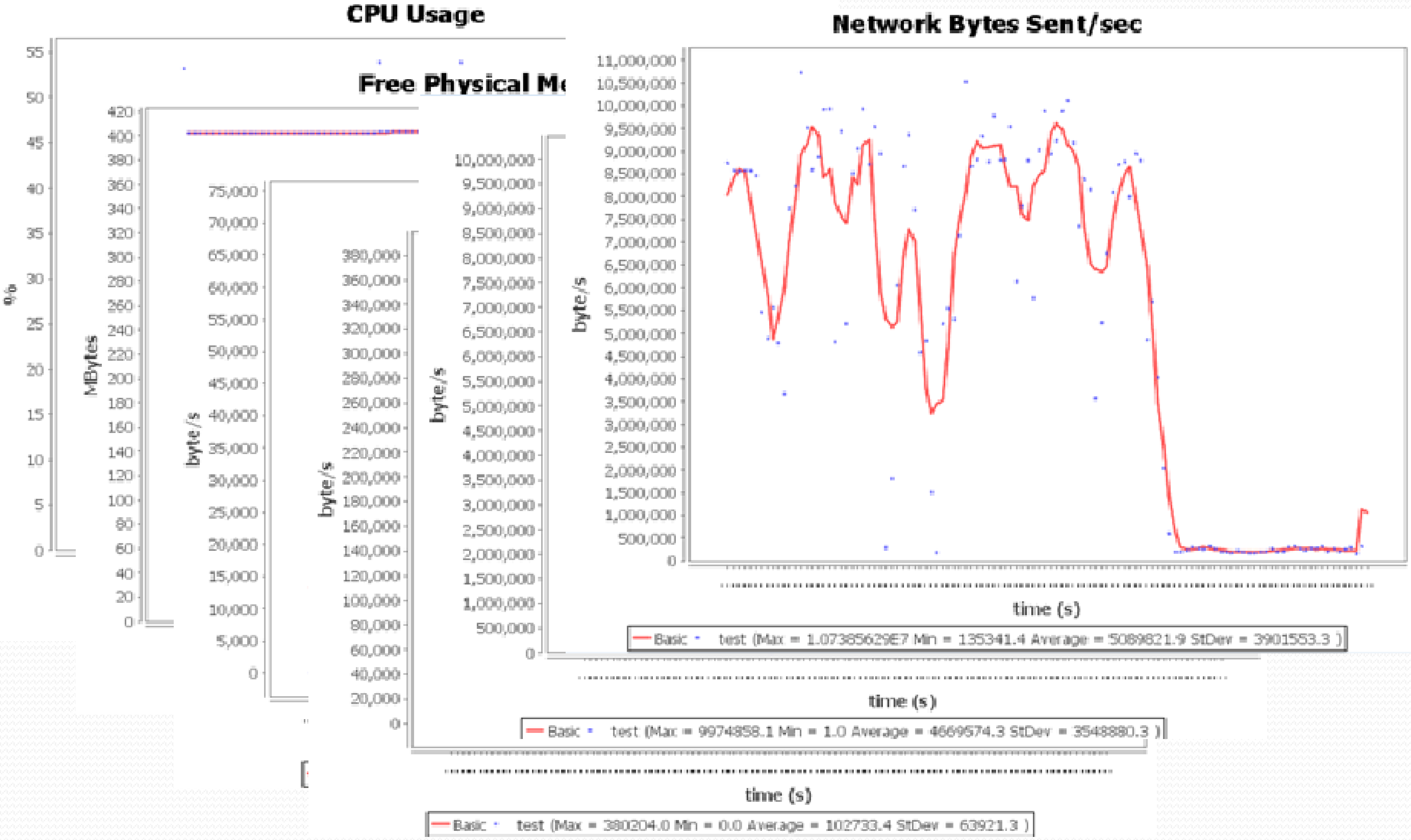
Adapter (Java)

Native Code (C++)	Native Code (C++)
PDH API	/proc filesystem
Windows	Linux

Hardware

ClusterMon

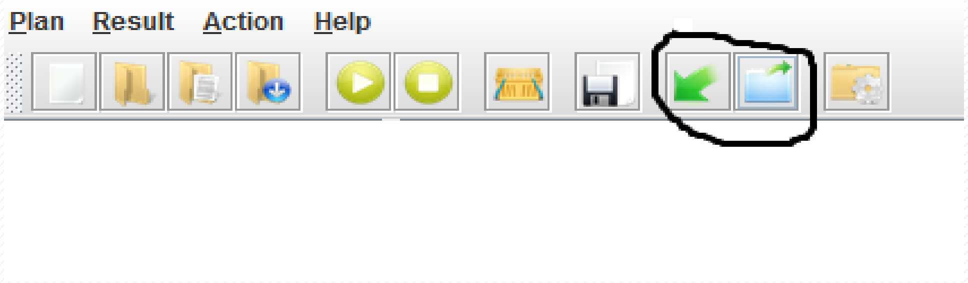

Server Cluster
Get Monitoring Data via RMI
Client
Leader
Notification via IP Multicast
Monitoring Data via UDP
Subordinates


CPU Usage
%
55
50
45
40
35
30
25
20
15
10
5
0
Free Physical Me
MBytes
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
byte/s
75,000
70,000
65,000
60,000
55,000
50,000
45,000
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
byte/s
380,000
360,000
340,000
320,000
300,000
280,000
260,000
240,000
220,000
200,000
180,000
160,000
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
time (s)
Basic · test (Max = 380204.0 Min = 0.0 Average = 102733.4 StDev = 63921.3)
byte/s
10,000,000
9,500,000
9,000,000
8,500,000
8,000,000
7,500,000
7,000,000
6,500,000
6,000,000
5,500,000
5,000,000
4,500,000
4,000,000
3,500,000
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
time (s)
Basic · test (Max = 9974858.1 Min = 1.0 Average = 4669574.3 StDev = 3548880.3)
Network Bytes Sent/sec
byte/s
11,000,000
10,500,000
10,000,000
9,500,000
9,000,000
8,500,000
8,000,000
7,500,000
7,000,000
6,500,000
6,000,000
5,500,000
5,000,000
4,500,000
4,000,000
3,500,000
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
time (s)
Basic · test (Max = 1.07385629E7 Min = 135341.4 Average = 5089821.9 StDev = 3901553.3)

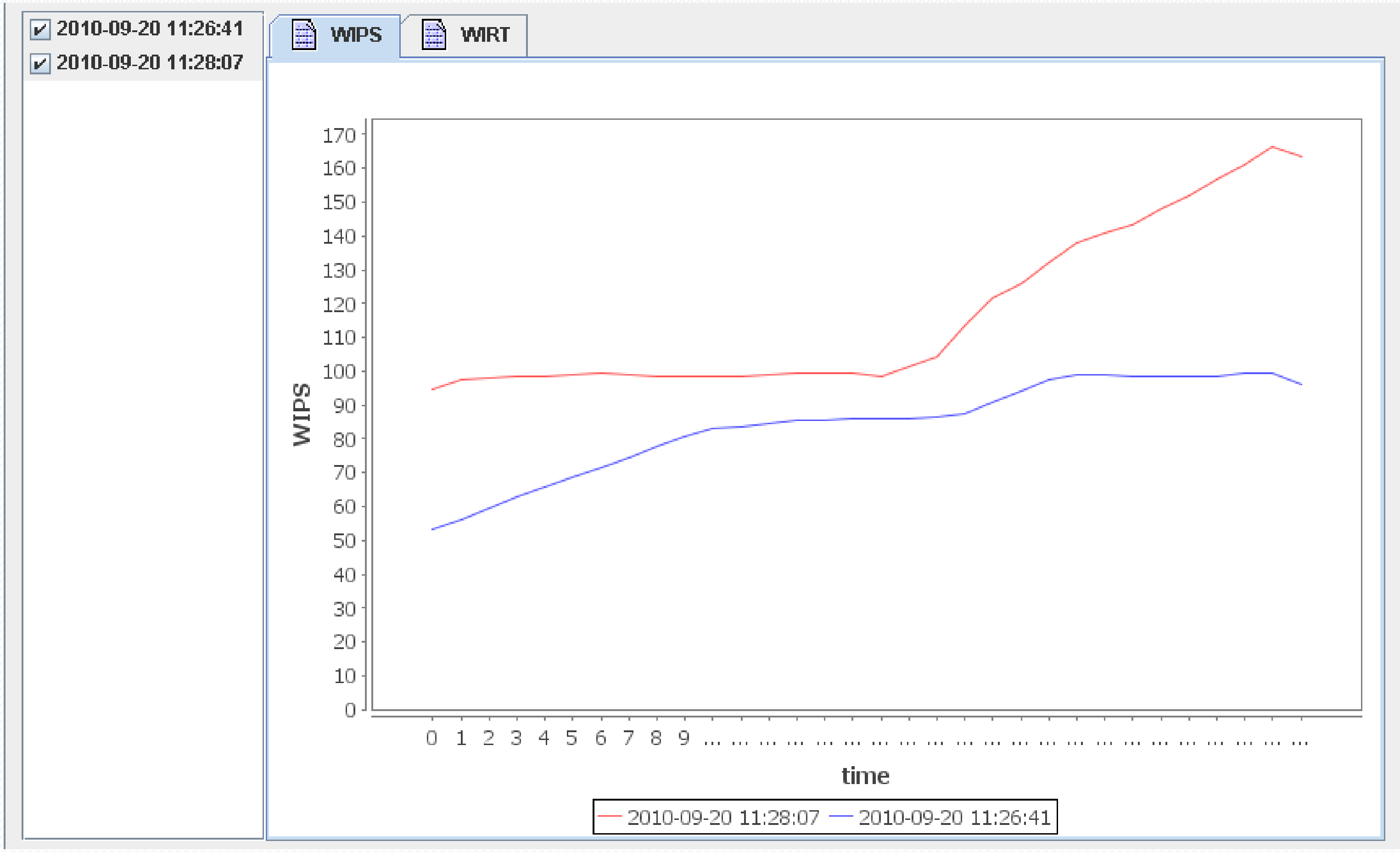
Workload recording

- Workload are generated randomly
- Compare the performance of different servers

Replay the workload



Result comparison



☑ 2010-09-20 11:26:41
☑ 2010-09-20 11:28:07

Welcome

http://forge.ow2.org/projects/jaspte

http://www.trustie.com/projects/project/show/Bench4Q

Questions?

Thank you!