THE UNIVERSITY OF CHICAGO

ARGONNE NATIONAL LABORATORY

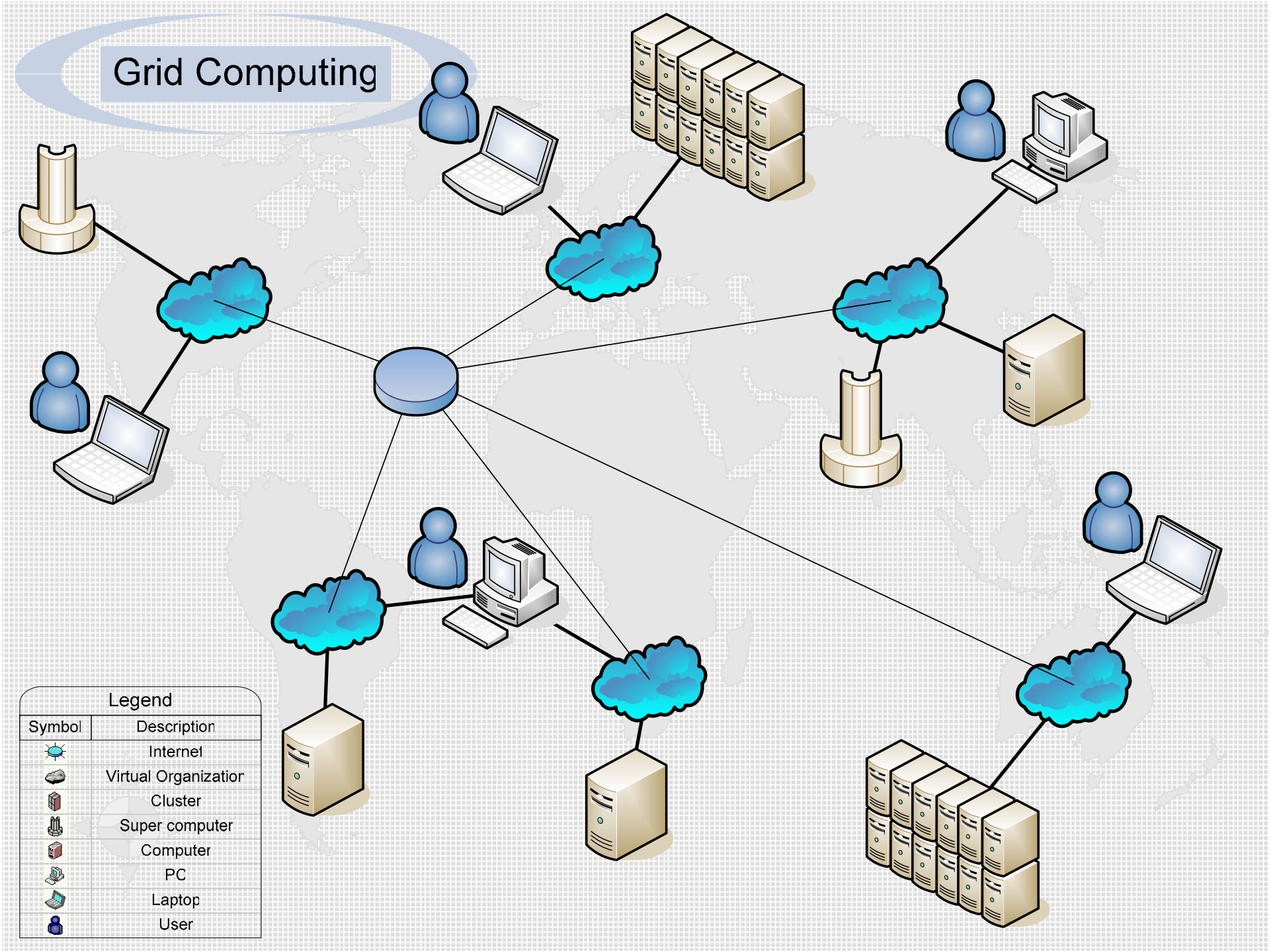
A Performance Study of the Globus Toolkit® and Grid Services via DiPerF, an automated DIstributed PERformance testing Framework

Ioan Raicu

Distributed Systems Laboratory
Computer Science Department
University of Chicago

Adviser:
Ian Foster
Committee Members:
Ian Foster, Rick Stevens, John Reppy

MS Thesis Defense
May 11th, 2005


Grid Computing
Legend
Symbol
Description
Internet
Virtual Organization
Cluster
Super computer
Computer
PC
Laptop
User

Grid Computing & the Globus Toolkit®

- The Globus Toolkit® (GT®) is the "*de facto standard*" for grid computing
- Grid Computing's focus:
  - **large-scale resource sharing:** direct access to computers, software, data
  - innovative applications
  - high-performance orientation
- The 'Grid problem':
  - **Definition:** flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources, namely virtual organizations
  - **Challenges:** Authentication, Authorization, resource access, resource discovery
- Globus Toolkit® Components
  - **GRAM:** Job Management
  - **MDS:** Monitoring and Discovery System
  - **GridFTP:** File Transfer
  - **Others:** RLS, RFT, CAS, OGSA-DAI, GTCP

Motivation & Goals

- **Part 1: Testing the performance of the Globus Toolkit®**
  - The Globus Toolkit® is the “de facto standard” for grid computing
  - Performance of GT® in a WAN & LAN is essential
    - expected performance from the GT® in a realistic deployment in a distributed and heterogeneous environment
  - Performance of grid services in a WAN
    - complex interactions between network connectivity and service performance.
- **Part 2: Developing DiPerF**
  - Performance testing is an ‘everyday’ task, **HOWEVER** testing harnesses are often built from scratch for a particular service
  - DiPerF can be used to test the scalability and performance limits of a service
  - controlled LAN-based tests are not enough
  - Wide-area, heterogeneous deployment provided by the PlanetLab and/or Grid3 testbed
  - DiPerF can provide accurate estimation of the service performance as experienced by both LAN and WAN clients

Obstacles in Performing Distributed Measurements

- *Accuracy*
  - synchronizing the time across an entire system that might have large communication latencies
- *Flexibility*
  - in heterogeneity normally found in WAN environments and the need to access large number of resources
- *Scalability*
  - the coordination of large amounts of resources
- *Performance*
  - the need to process large number of transactions per second

My Thesis in a "Nutshell": Part 1 - Performance of GT®

- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2

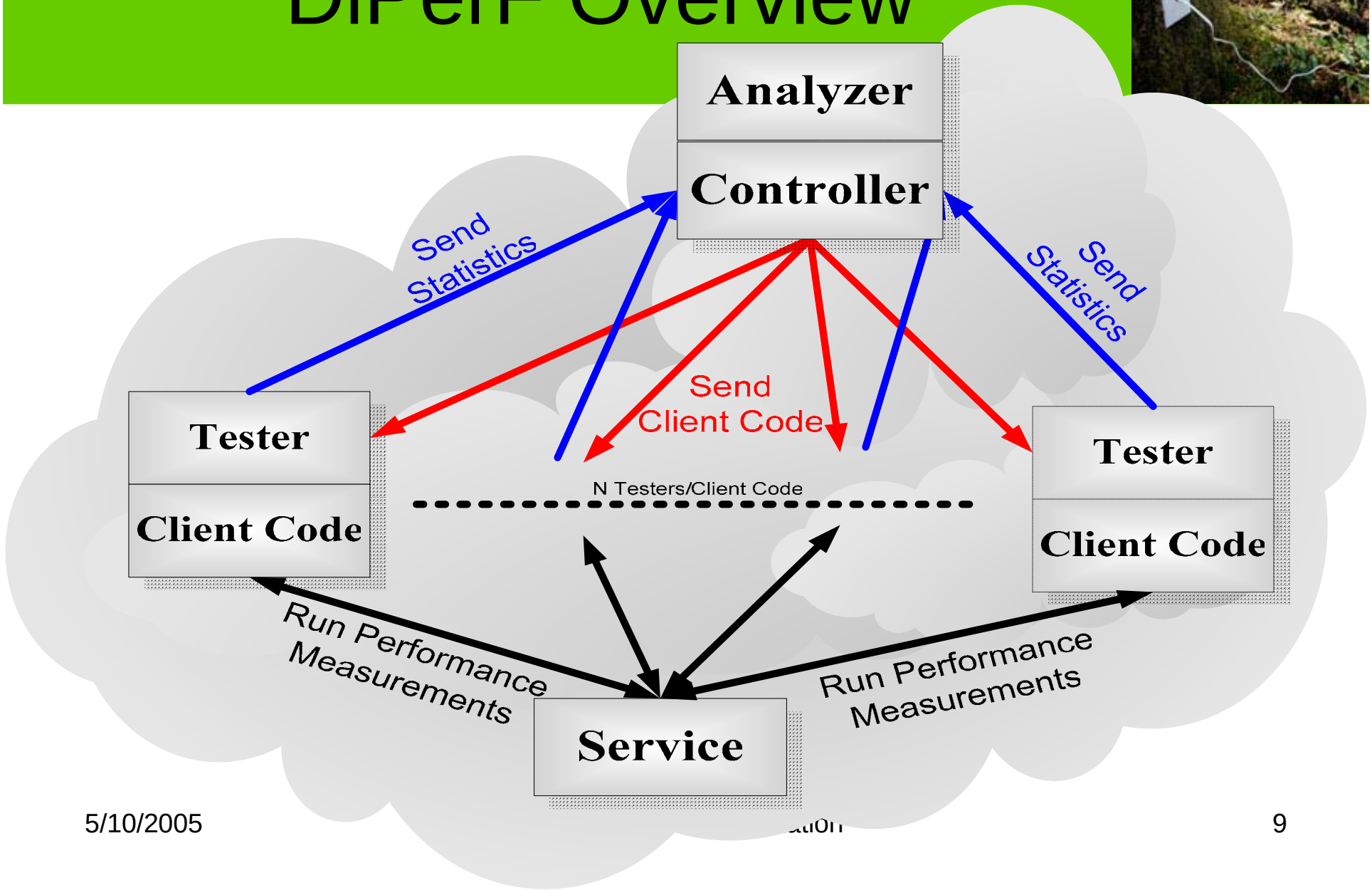
My Thesis in a "Nutshell": Part 2 - DiPerF

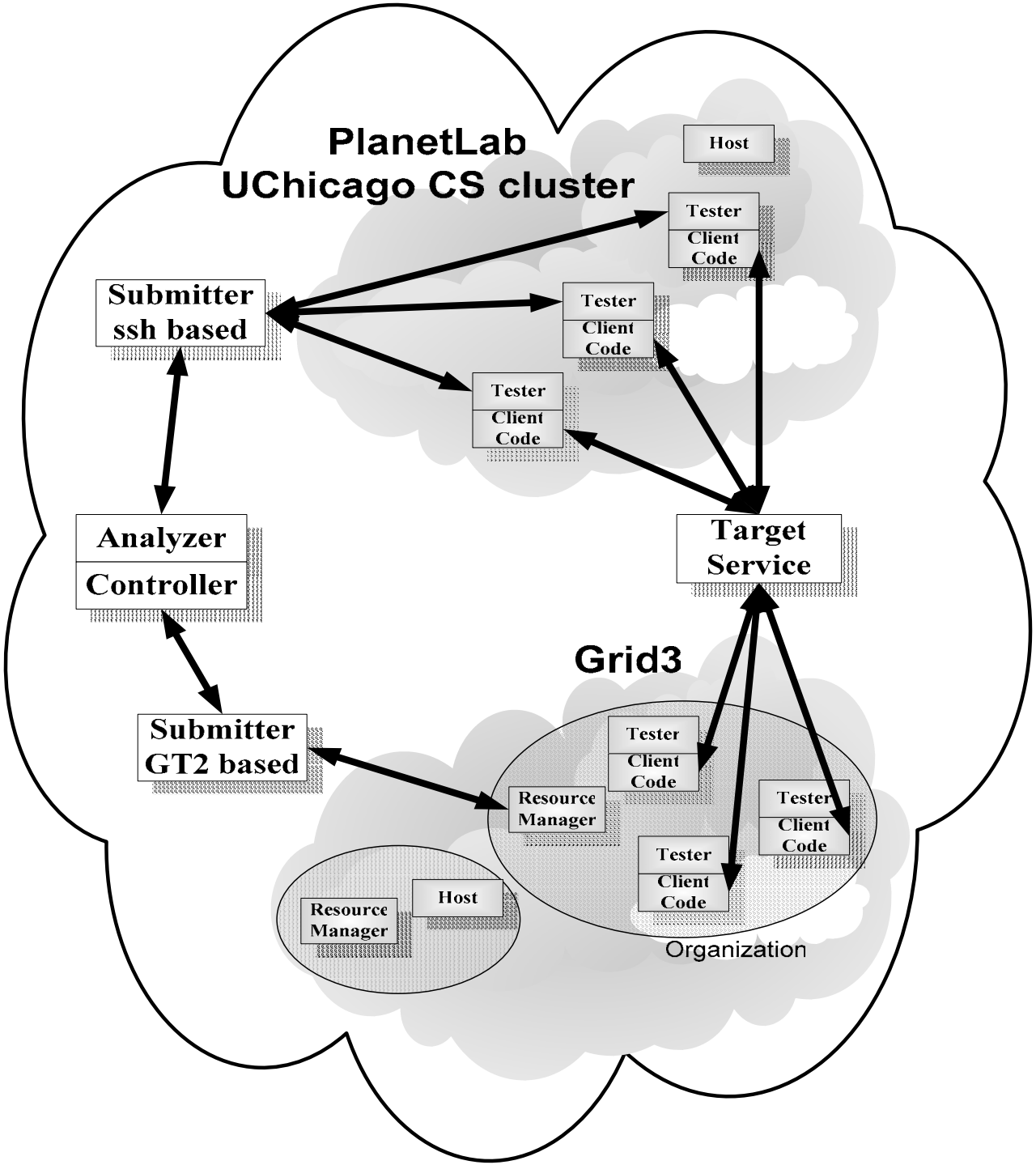
- **Goals: *simplify and automate large scale service performance evaluation***
- DiPerF Features
  - coordinates a pool of machines that test a single or distributed target service
  - collects and aggregates performance metrics from the client point of view
  - generates performance statistics
- DiPerF Implementation
  - modularized tool written in C/C++/perl
  - Uses off-the-shelf tools and protocols: Ssh-based tools (i.e. scp, rsync), telnet, TCP/UDP/IP
  - tested over various testbeds: PlanetLab, Grid3, Computer Science Cluster at the University of Chicago
- DiPerF Performance:
  - 10,000+ clients & 100,000+ transactions per second & validation study

DiPerF Components

- Controller
  - Receives the address of the service and a client code
  - Distributes the client code across all machines in the pool
  - Gathers and stores performance statistics
- Tester
  - Receives client code
  - Runs the code and produce performance statistics
  - Sends back to “controller” raw statistic metrics
- Analyzer
  - Aggregates and summarizes performance statistics

DiPerF Overview

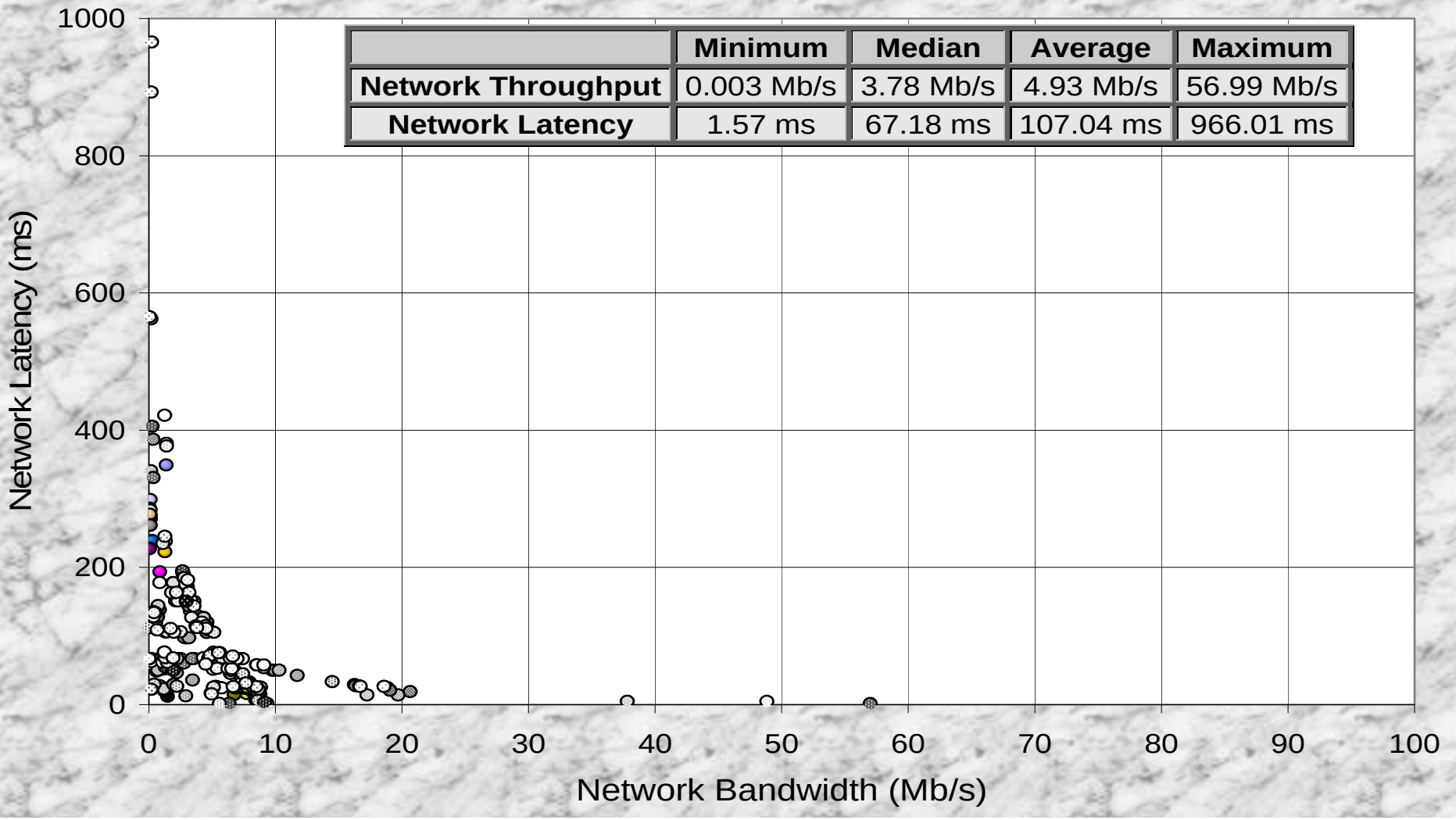



PlanetLab
UChicago CS cluster
Host
Tester
Client Code
Submitter ssh based
Tester
Client Code
Tester
Client Code
Analyzer
Controller
Target Service
Grid3
Submitter GT2 based
Tester
Client Code
Resource Manager
Tester
Client Code
Tester
Client Code
Resource Manager
Host
Organization

PlanetLab Testbed Characteristics

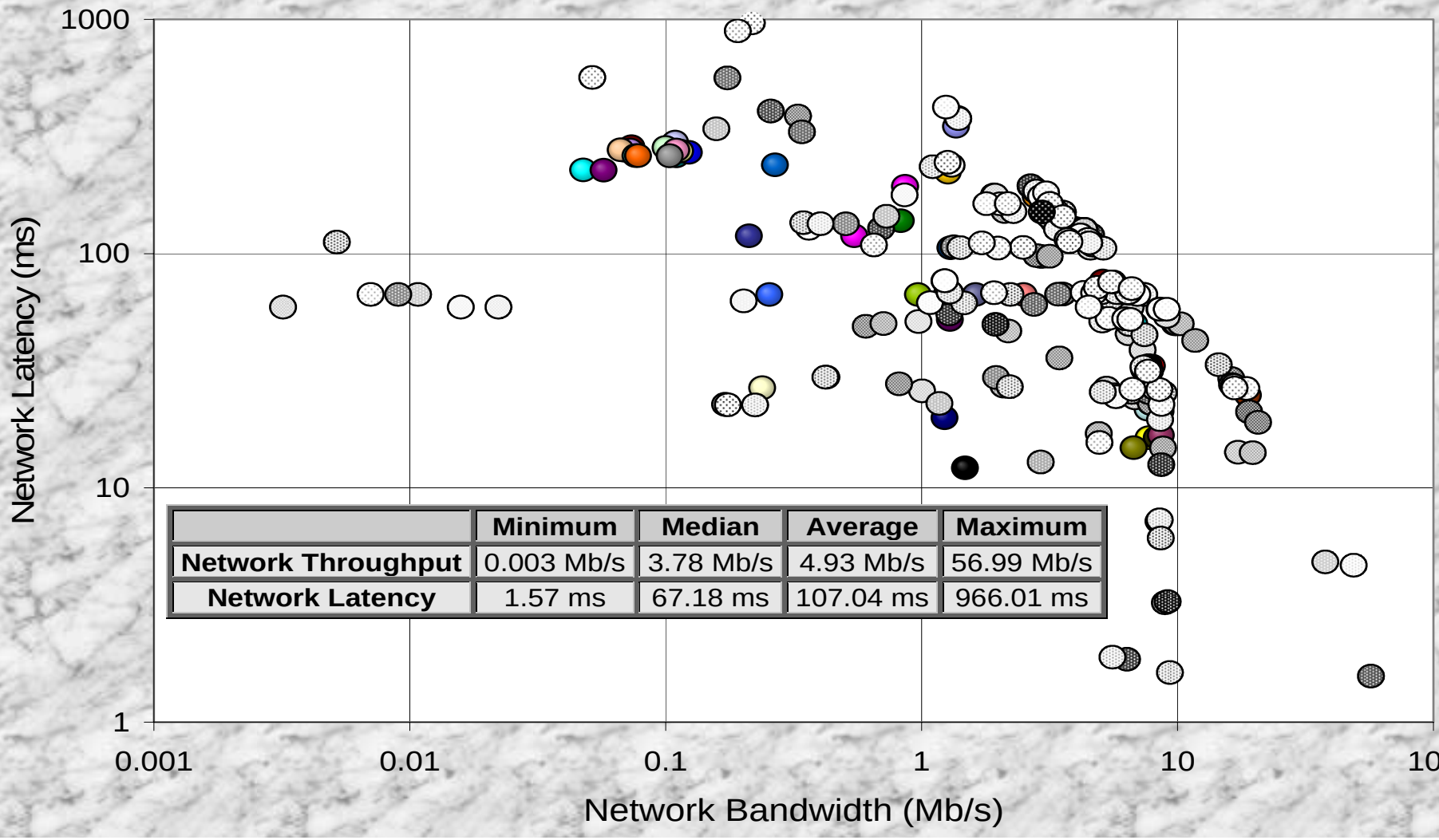
PlanetLab Network Performance from 268 nodes to UChicago

	Minimum	Median	Average	Maximum
Network Throughput	0.003 Mb/s	3.78 Mb/s	4.93 Mb/s	56.99 Mb/s
Network Latency	1.57 ms	67.18 ms	107.04 ms	966.01 ms



PlanetLab Testbed Characteristics

PlanetLab Network Performance from 268 nodes to UChicago

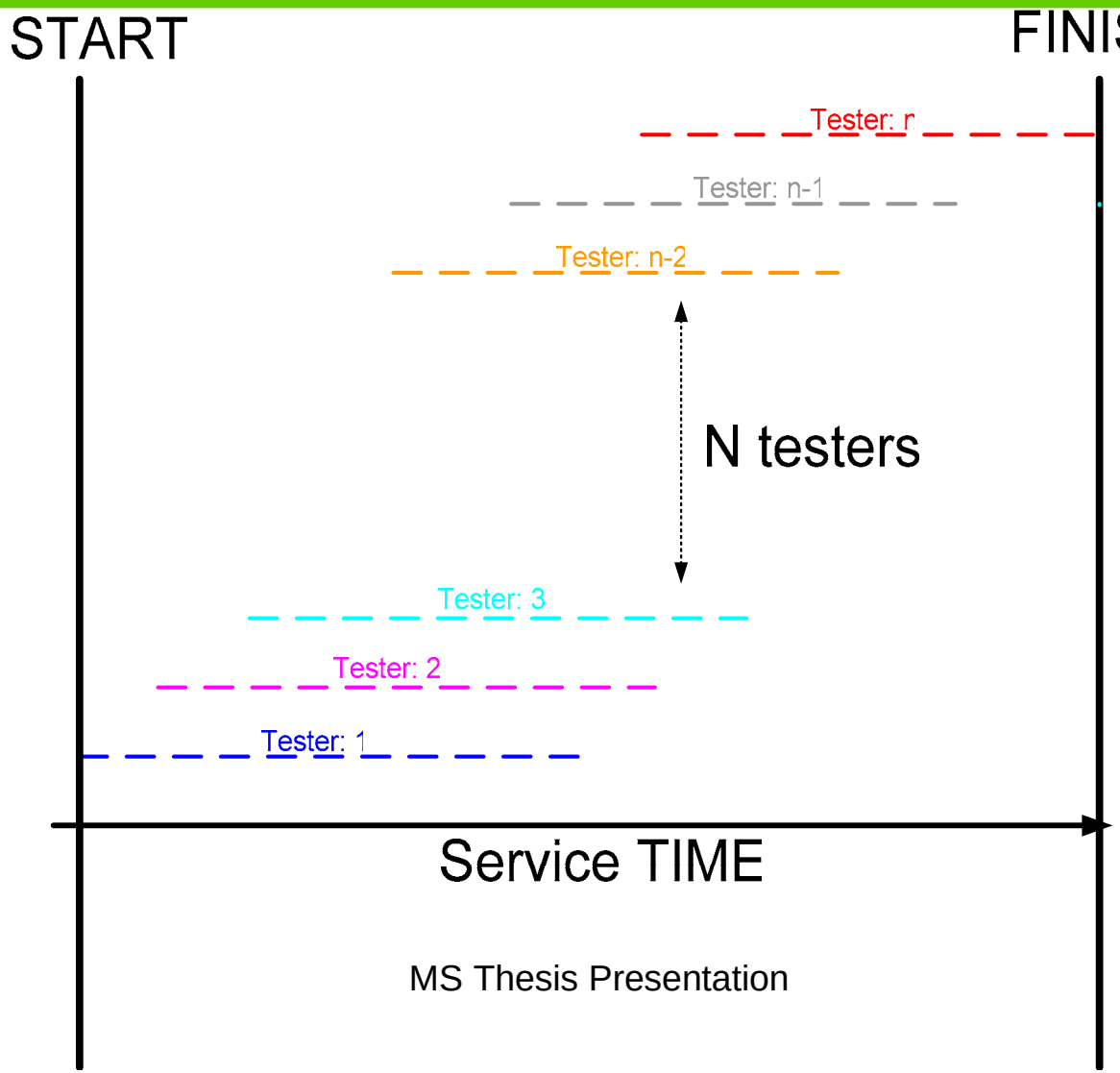


	Minimum	Median	Average	Maximum
Network Throughput	0.003 Mb/s	3.78 Mb/s	4.93 Mb/s	56.99 Mb/s
Network Latency	1.57 ms	67.18 ms	107.04 ms	966.01 ms

Time Synchronization

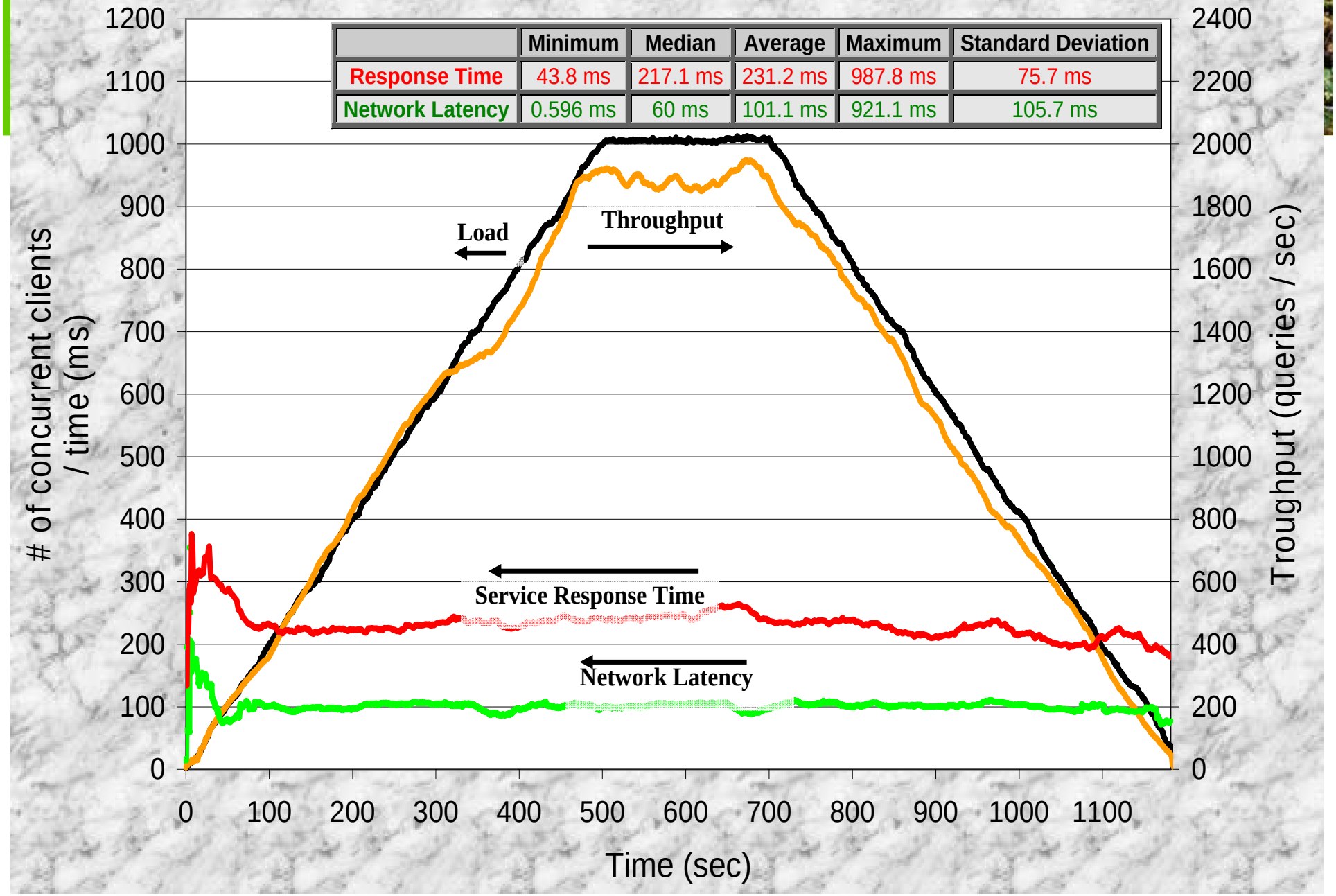
- Distributed approach:
  - Tester uses Network Time Protocol (NTP) to synchronize time
  - Not deployed or configured properly everywhere
- Centralized approach:
  - Controller uses time translation to synchronize time
  - Could introduce some time synchronization inaccuracies due to non-symetrical network links and the RTT variance

Metric Aggregation

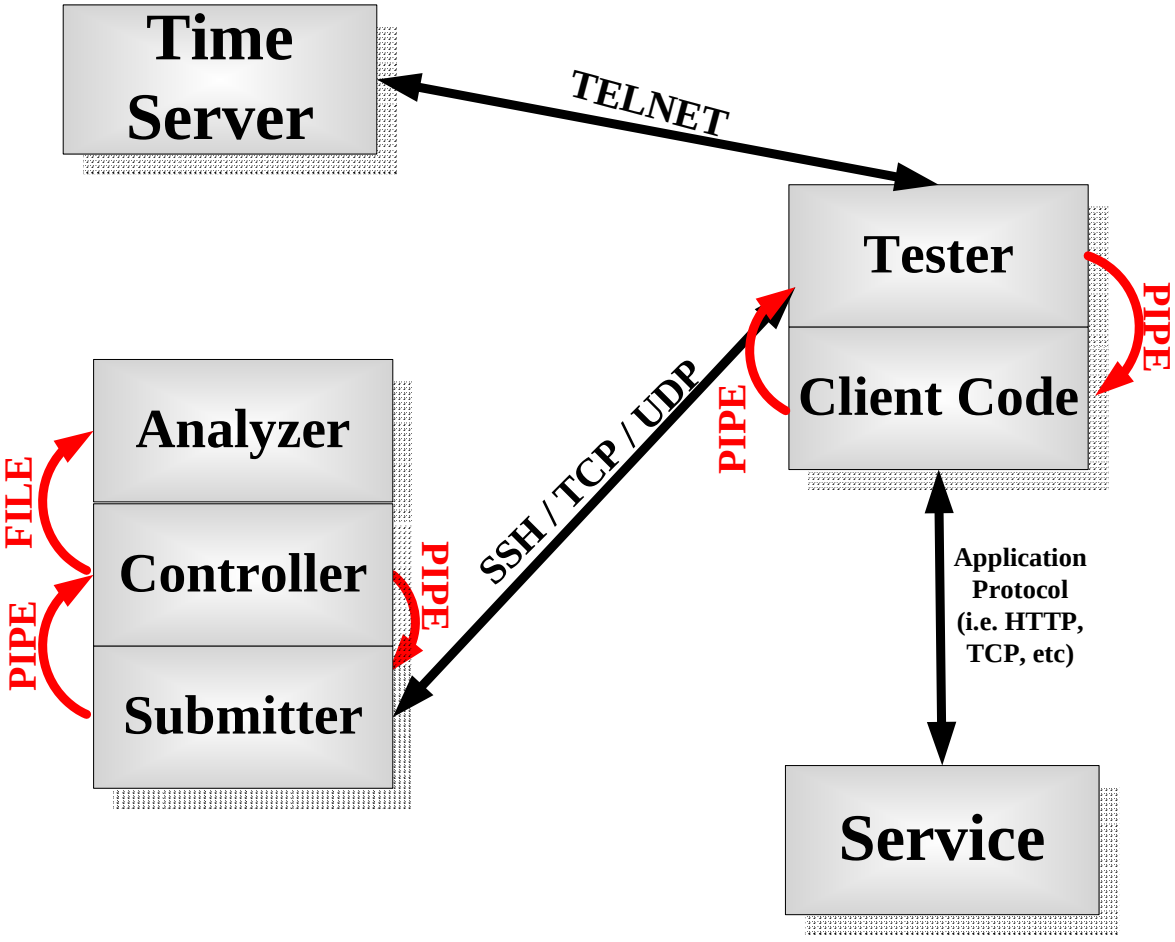


Time Server Performance

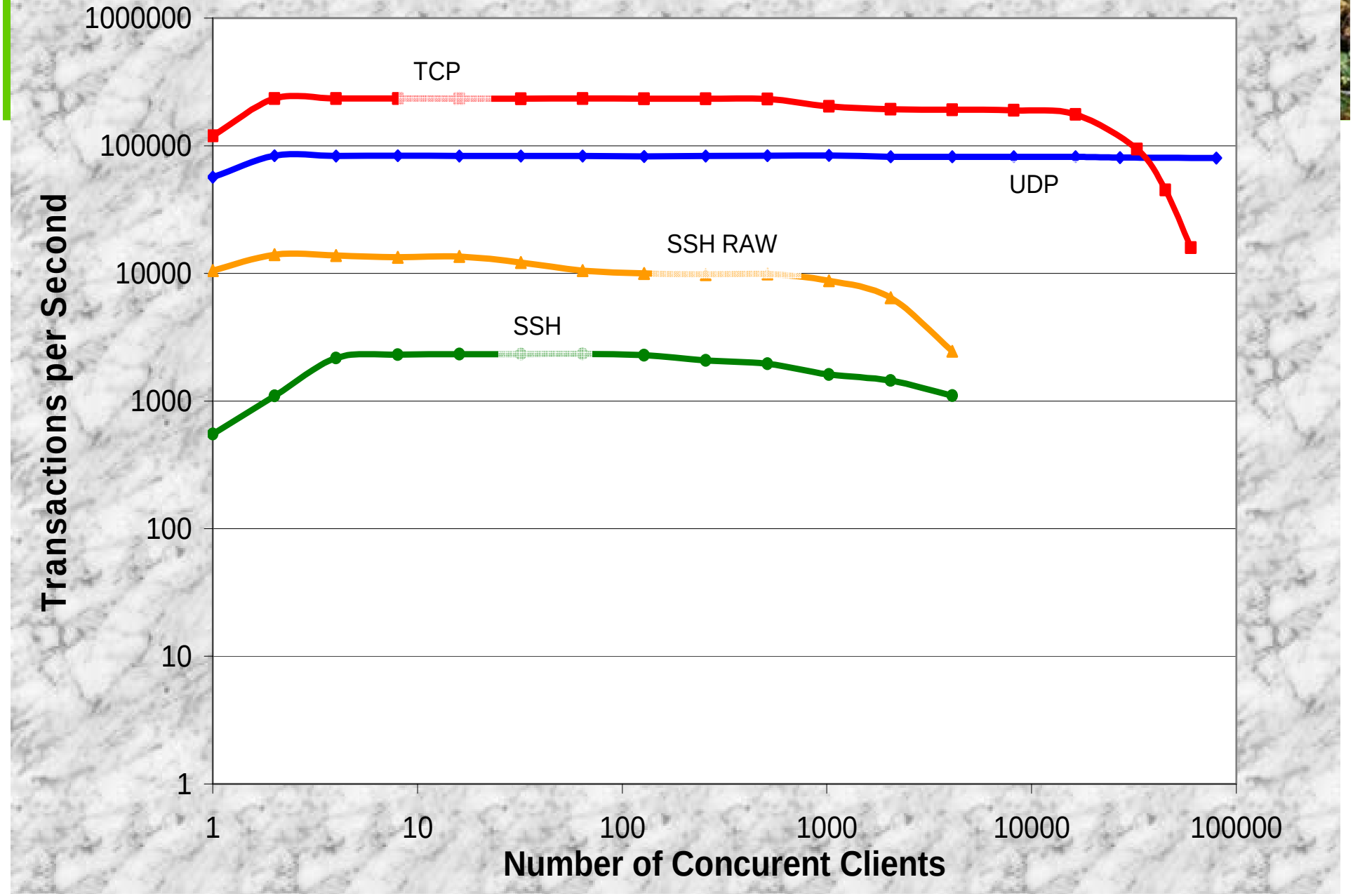
	Minimum	Median	Average	Maximum	Standard Deviation
Response Time	43.8 ms	217.1 ms	231.2 ms	987.8 ms	75.7 ms
Network Latency	0.596 ms	60 ms	101.1 ms	921.1 ms	105.7 ms



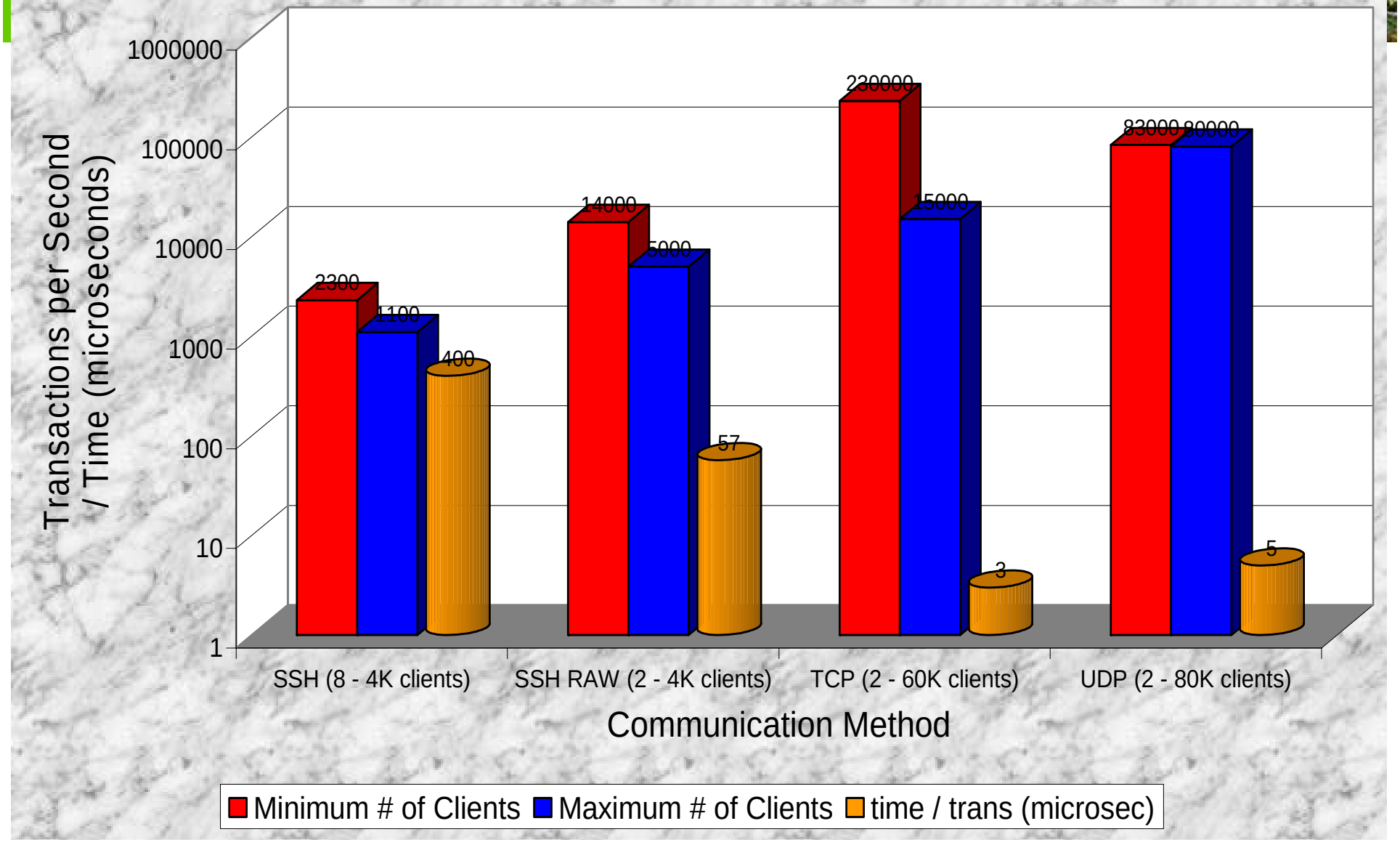
Communication Overview



Summary of Communication Performance and Scalability



Summary of Communication Performance and Scalability



Analyzer: Performance Metrics

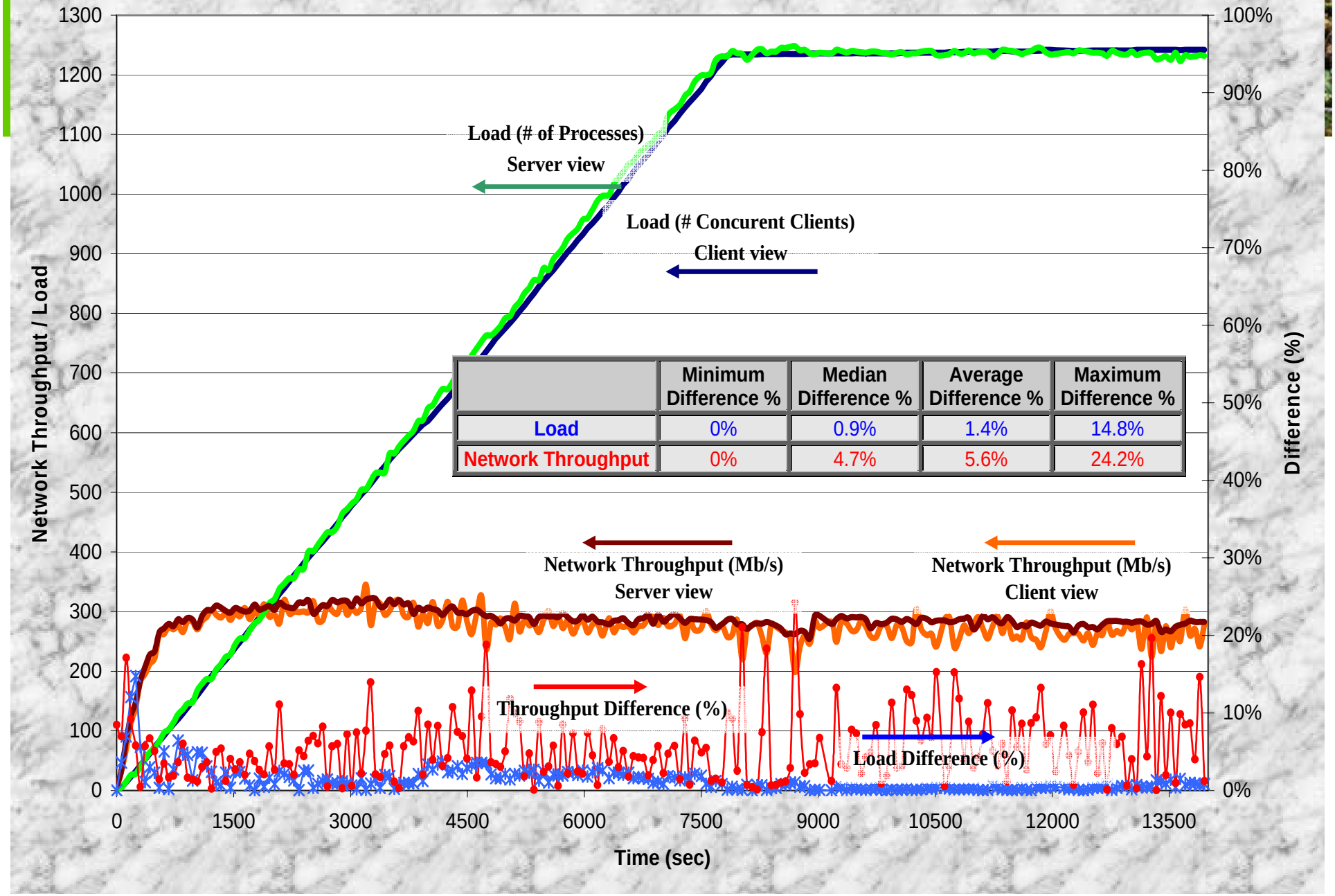
- ***service response time:***
  - the time from when a client issues a request to when the request is completed minus the network latency and minus the execution time of the client code
- ***service throughput*:**
  - number of jobs completed successfully by the service averaged over a short time interval
- ***offered load*:**
  - number of concurrent service requests (per second)
- ***jobs completed / failed* (per client):**
  - The number of jobs completed successfully and the number of failed jobs
- ***service utilization* (per client):**
  - ratio between the number of requests served for a client and the total number of requests served by the service during the time the client was active
- ***service fairness* (per client):**
  - ratio between the number of jobs completed and service utilization
- **network latency to the service:**
  - time taken for a minimum sized packet to traverse the network from the client to the service
- **time synchronization error:**
  - real time difference between client and service measured as a function of network latency variance
- **client measured metrics:**
  - Any performance metric that the client measures and communicates with the tester

Analyzer

- Design
  - 4000+ lines of C++ code
  - Performance metrics:
    - 8 generic metrics
    - Client specific metrics
  - Supported features:
    - Analyze just parts of data
    - Verify data files
    - Time Quanta
- Performance

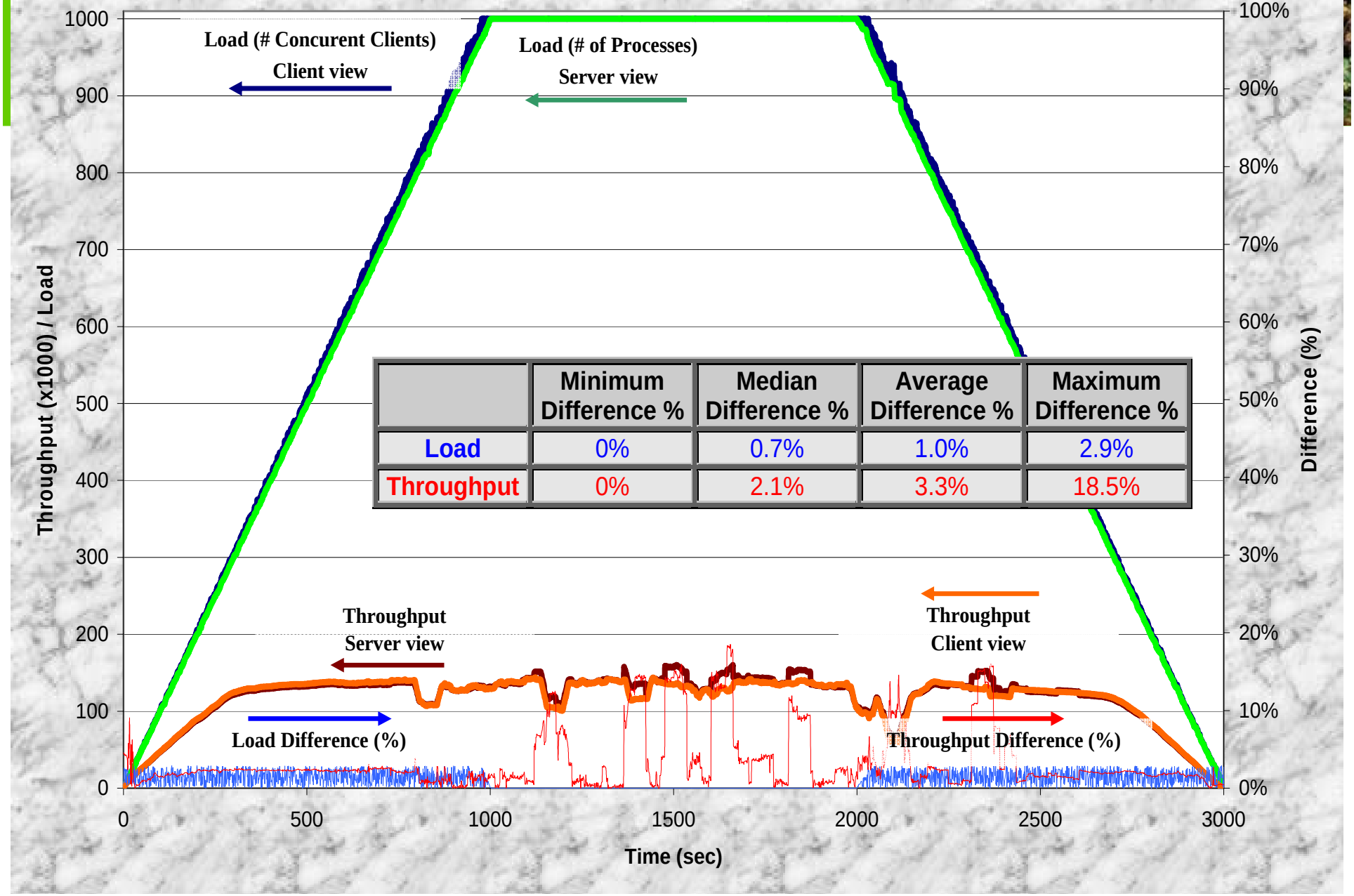
# of Mach	Test Length	# of Trans	Memory Footprint (MB)	Time Quanta	Execution Steps	Time (sec)	Time / Trans (ms)	Trans / sec
8	168	100	0.0	1 sec	Verify Load Throughput Resp. time	1.4	14.0	71.4
40	1000	2900	0.5	1 sec		1.6	0.6	1812.5
200	11000	45000	26.5	1 sec		5.1	0.1	8840.9
1000	255	125000	6.5	1 sec		33.0	0.2	3787.9
1000	3000	2000000	91.6	1 sec		951.7	1.0	2101.6
1700	6500	671000	145.7	1 sec		166.5	0.3	4030.0
3600	12000	354000	504.5	1 sec		359.0	0.5	986.1

DiPerF Validation: GridFTP

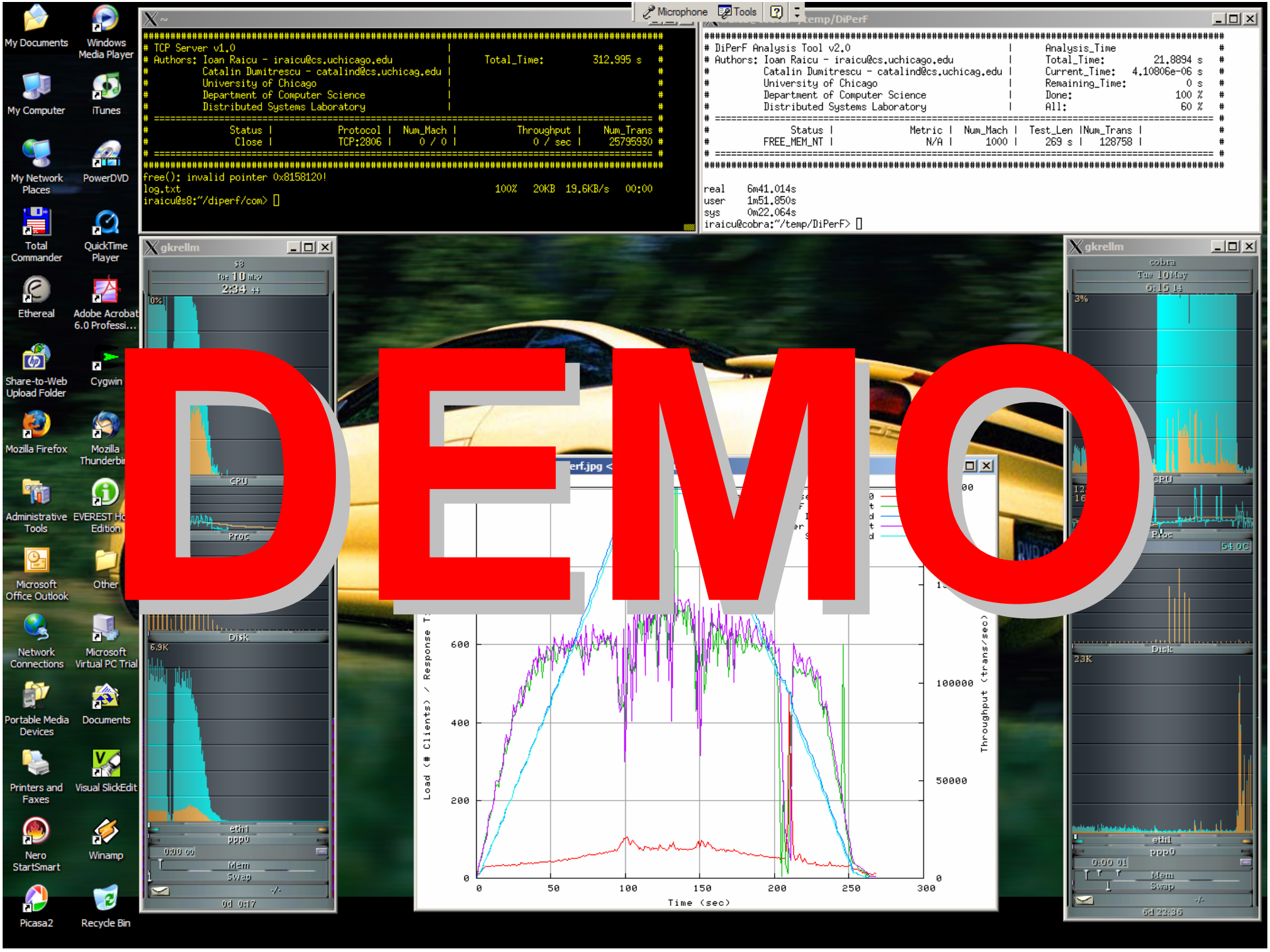


	Minimum Difference %	Median Difference %	Average Difference %	Maximum Difference %
Load	0%	0.9%	1.4%	14.8%
Network Throughput	0%	4.7%	5.6%	24.2%

DiPerF Validation: TCP Server



	Minimum Difference %	Median Difference %	Average Difference %	Maximum Difference %
Load	0%	0.7%	1.0%	2.9%
Throughput	0%	2.1%	3.3%	18.5%

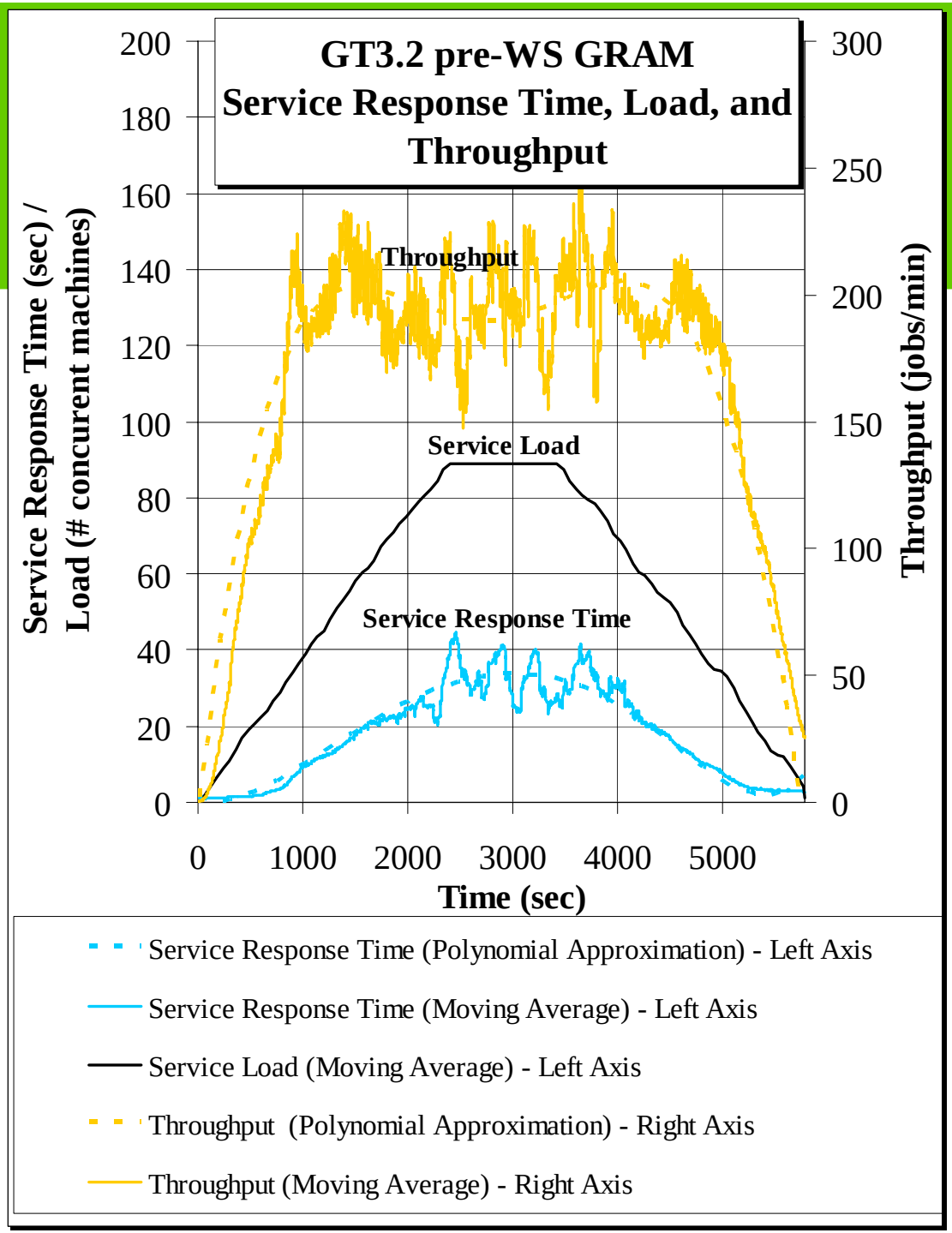

DEMO
TCP Server v1.0
Authors: Ioan Raicu - iraicu@cs.uchicago.edu
Catalin Dumitrescu - catalind@cs.uchicag.edu
University of Chicago
Department of Computer Science
Distributed Systems Laboratory
Total_Time: 312.995 s
Status | Protocol | Num_Mach | Throughput | Num_Trans
Close | TCP:2806 | 0 / 0 | 0 / sec | 25795930
free(): invalid pointer 0x8158120!
log.txt 100% 20KB 19.6KB/s 00:00
iraicu@s8:~/diperf/com>
DiPerF Analysis Tool v2.0
Authors: Ioan Raicu - iraicu@cs.uchicago.edu
Catalin Dumitrescu - catalind@cs.uchicag.edu
University of Chicago
Department of Computer Science
Distributed Systems Laboratory
Analysis_Time
Total_Time: 21.8894 s
Current_Time: 4.10806e-06 s
Remaining_Time: 0 s
Done: 100 %
All: 60 %
Status | Metric | Num_Mach | Test_Len | Num_Trans
FREE_MEM_NT | N/A | 1000 | 269 s | 128758
real 6m41.014s
user 1m51.850s
sys 0m22.064s
iraicu@cobra:~/temp/DiPerF>
Load (# Clients) / Response T
Throughput (trans/sec)
Time (sec)
gkrellm
My Documents
Windows Media Player
My Computer
iTunes
My Network Places
PowerDVD
Total Commander
QuickTime Player
Ethereal
Adobe Acrobat 6.0 Professi...
Share-to-Web Upload Folder
Cygwin
Mozilla Firefox
Mozilla Thunderbi
Administrative Tools
EVEREST Ho Edition
Microsoft Office Outlook
Other
Network Connections
Microsoft Virtual PC Trial
Portable Media Devices
Documents
Printers and Faxes
Visual SlickEdit
Nero StartSmart
Winamp
Picasa2
Recycle Bin

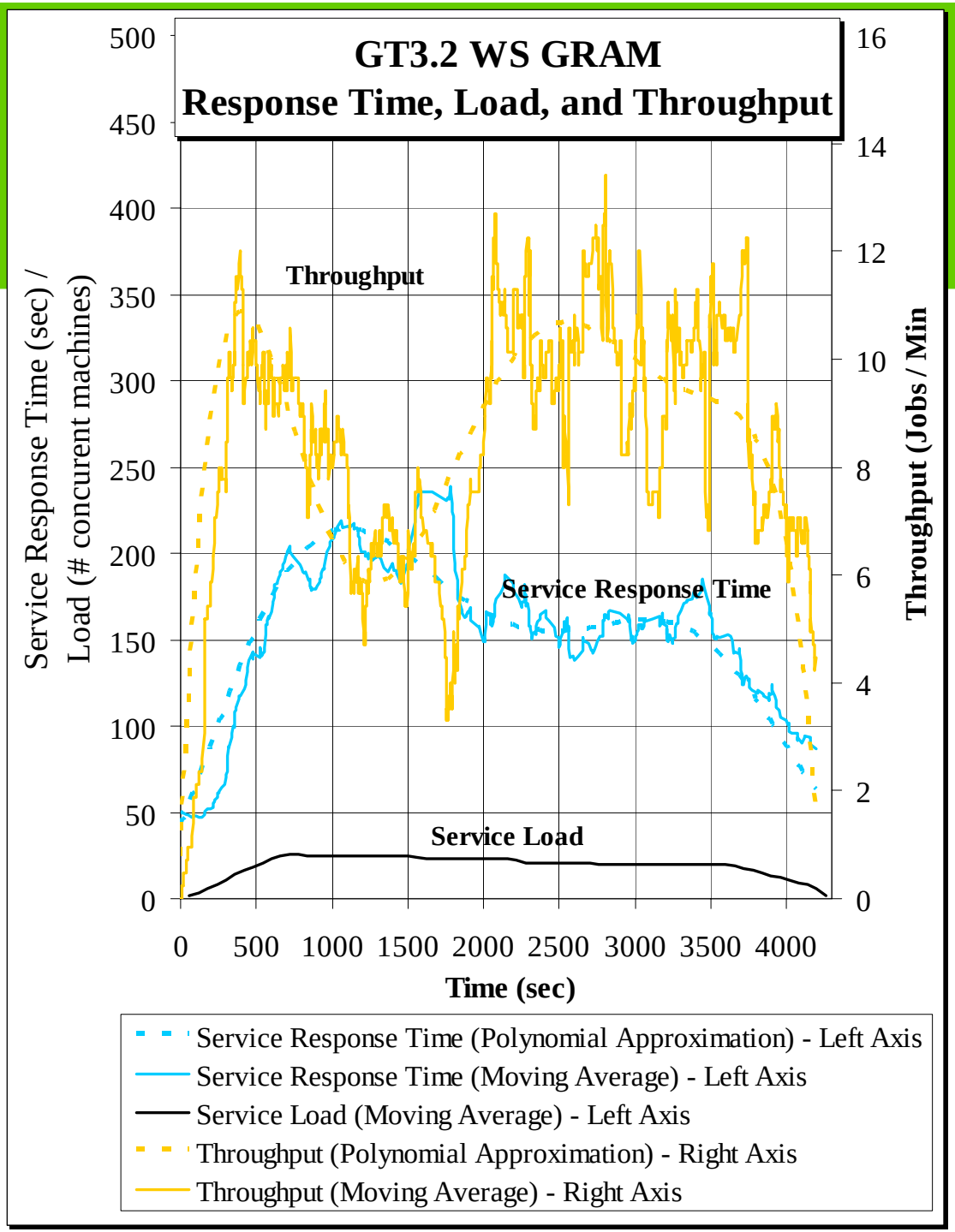
Performance of GT®

- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2

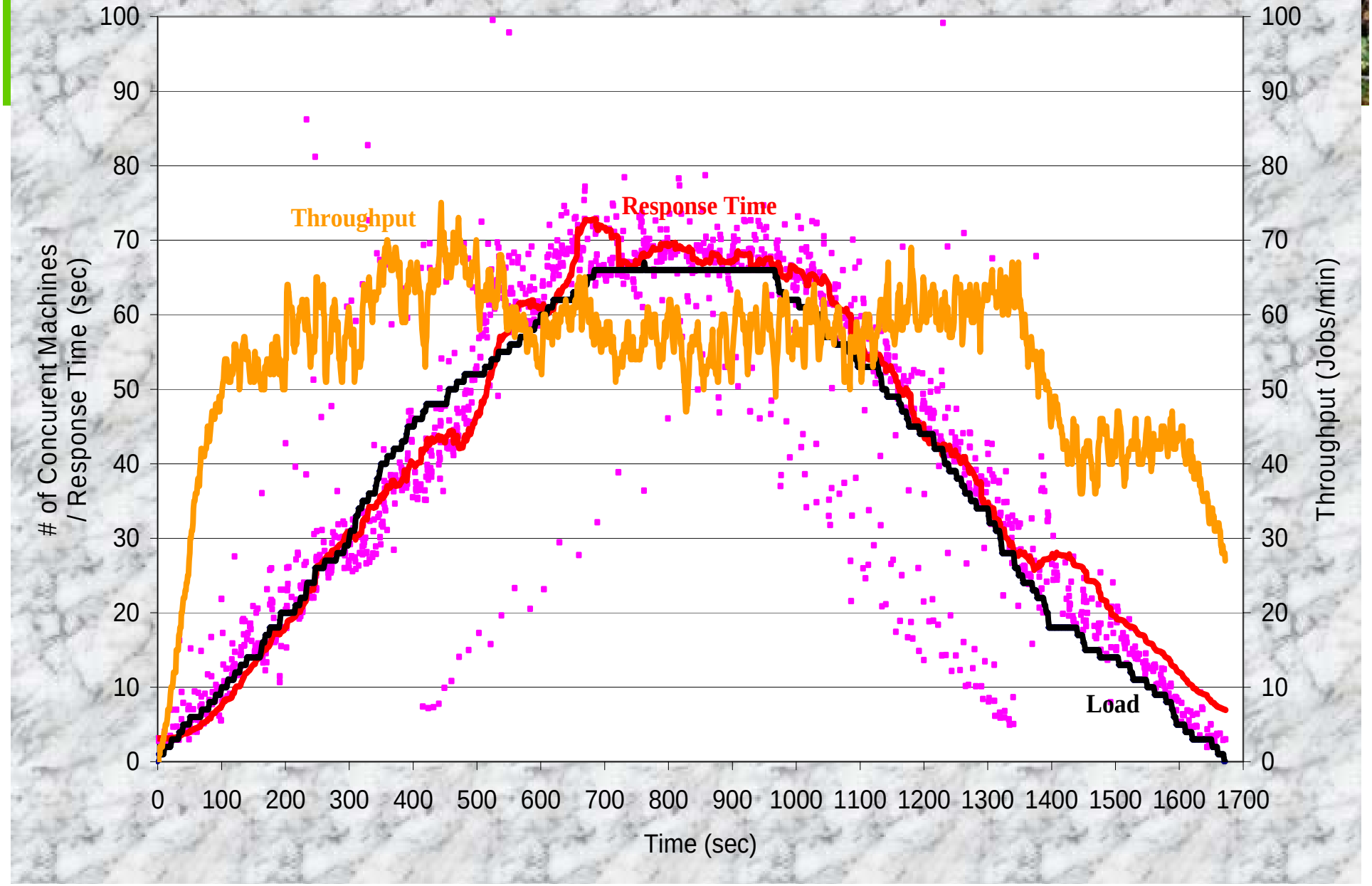
Job Submission: GRAM

- GT3.2 GRAM
  - Job submission via Globus Gatekeeper 2.4.3 using Globus Toolkit 3.2 (C version)
  - Job submission using Globus Toolkit 3.2 (Java version)
- GT3.9.4 GRAM
  - Job submission using Globus Toolkit 3.9.4 and a pre-WS GRAM client (C) and pre-WS GRAM Service (C)
  - Job submission using Globus Toolkit 3.9.4 and a WS GRAM client (C) and WS GRAM Service (Java)

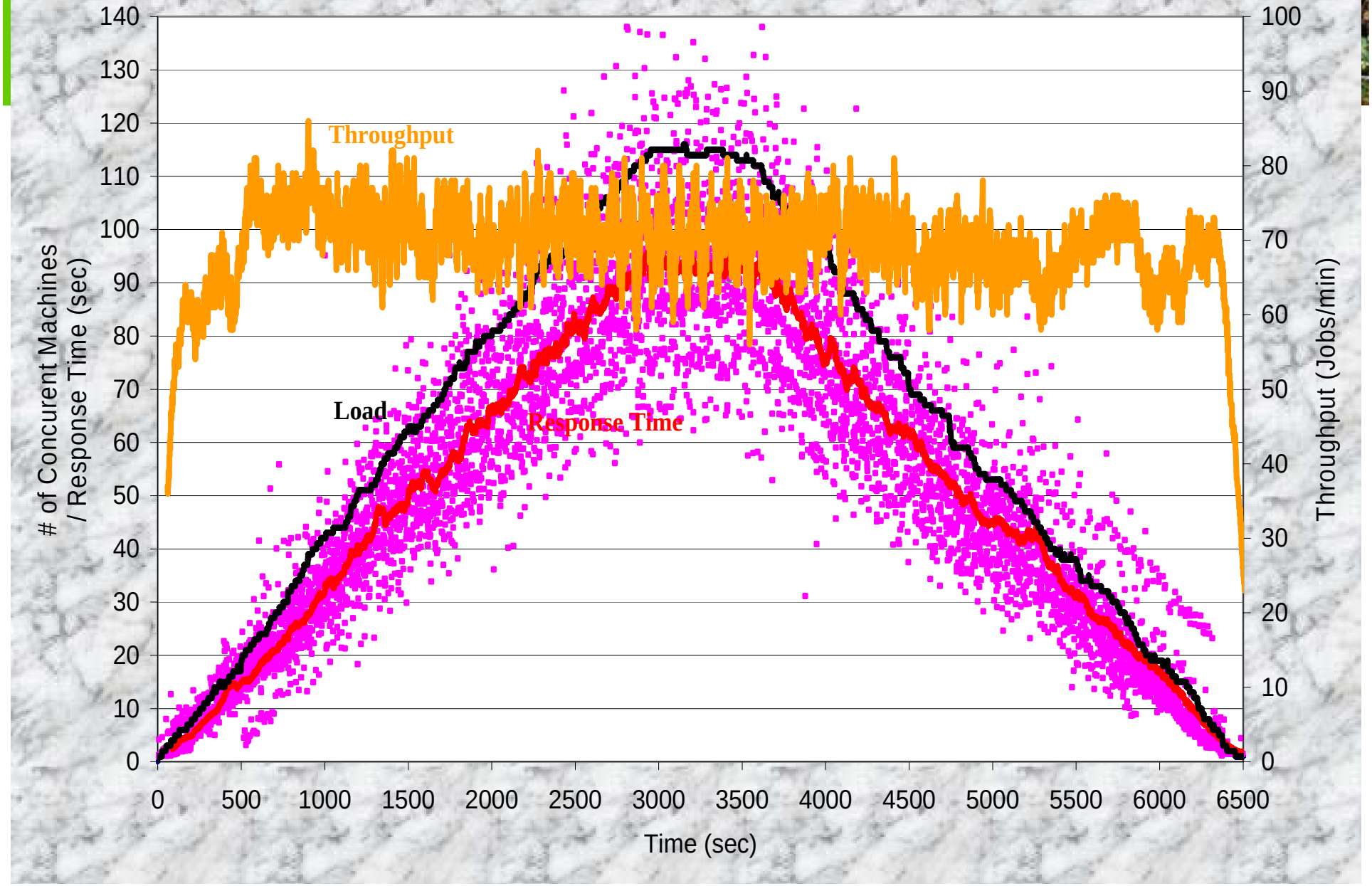

GT3.2 pre-WS GRAM
Service Response Time, Load, and Throughput
Throughput
Service Load
Service Response Time
Service Response Time (sec) / Load (# concurent machines)
Throughput (jobs/min)
Time (sec)
0
20
40
60
80
100
120
140
160
180
200
0
50
100
150
200
250
300
0
1000
2000
3000
4000
5000
Service Response Time (Polynomial Approximation) - Left Axis
Service Response Time (Moving Average) - Left Axis
Service Load (Moving Average) - Left Axis
Throughput (Polynomial Approximation) - Right Axis
Throughput (Moving Average) - Right Axis


GT3.2 WS GRAM
Response Time, Load, and Throughput
Throughput
Service Response Time
Service Load
Service Response Time (sec) / Load (# concurent machines)
Throughput (Jobs / Min
Time (sec)
0
50
100
150
200
250
300
350
400
450
500
0
2
4
6
8
10
12
14
16
0
500
1000
1500
2000
2500
3000
3500
4000
Service Response Time (Polynomial Approximation) - Left Axis
Service Response Time (Moving Average) - Left Axis
Service Load (Moving Average) - Left Axis
Throughput (Polynomial Approximation) - Right Axis
Throughput (Moving Average) - Right Axis

GT3.9.4 WS GRAM Client (C) and WS GRAM Service (JAVA)



GT3.9.4 Pre-WS GRAM Client (C) and Pre-WS GRAM Service (C)

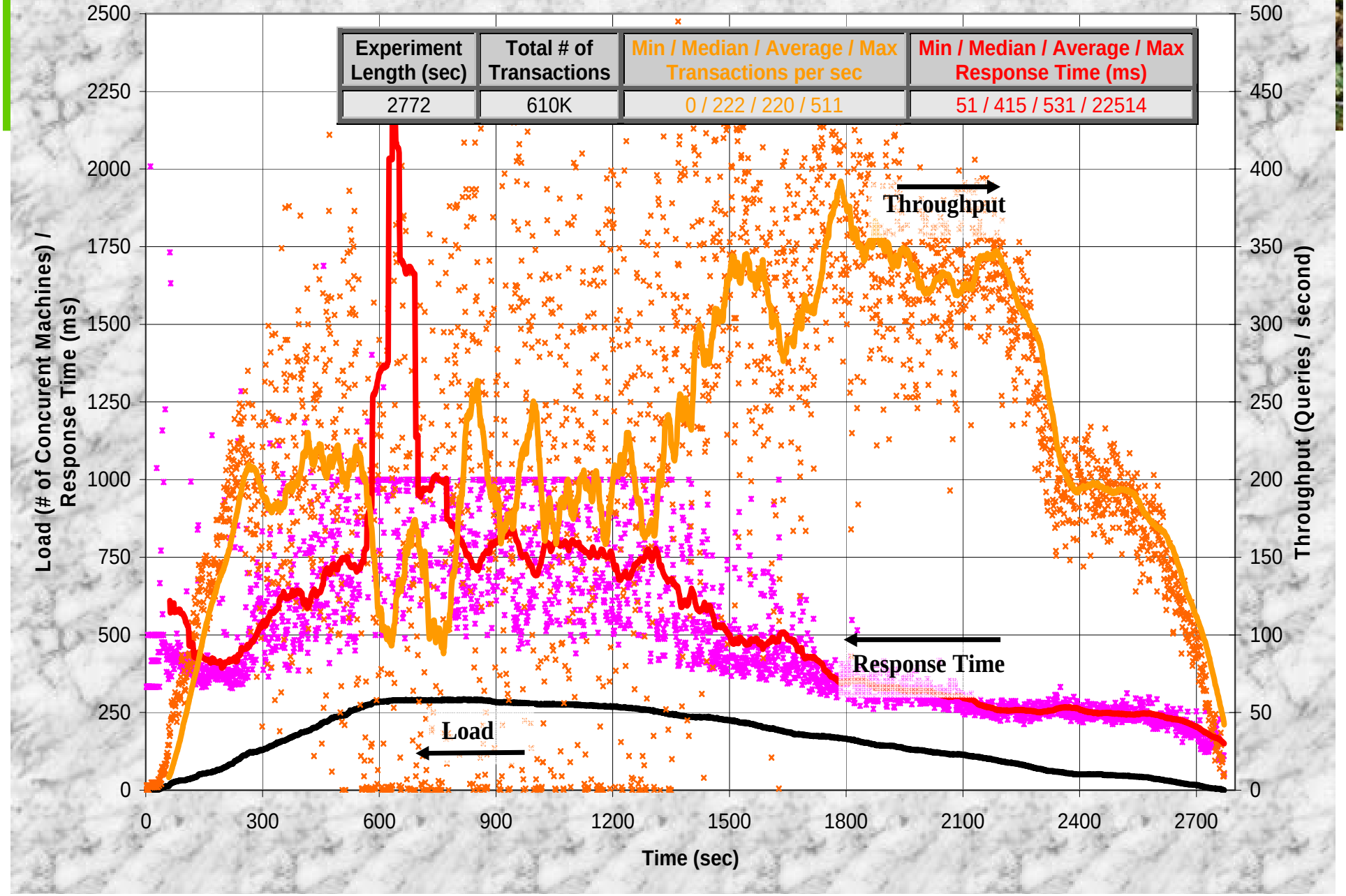


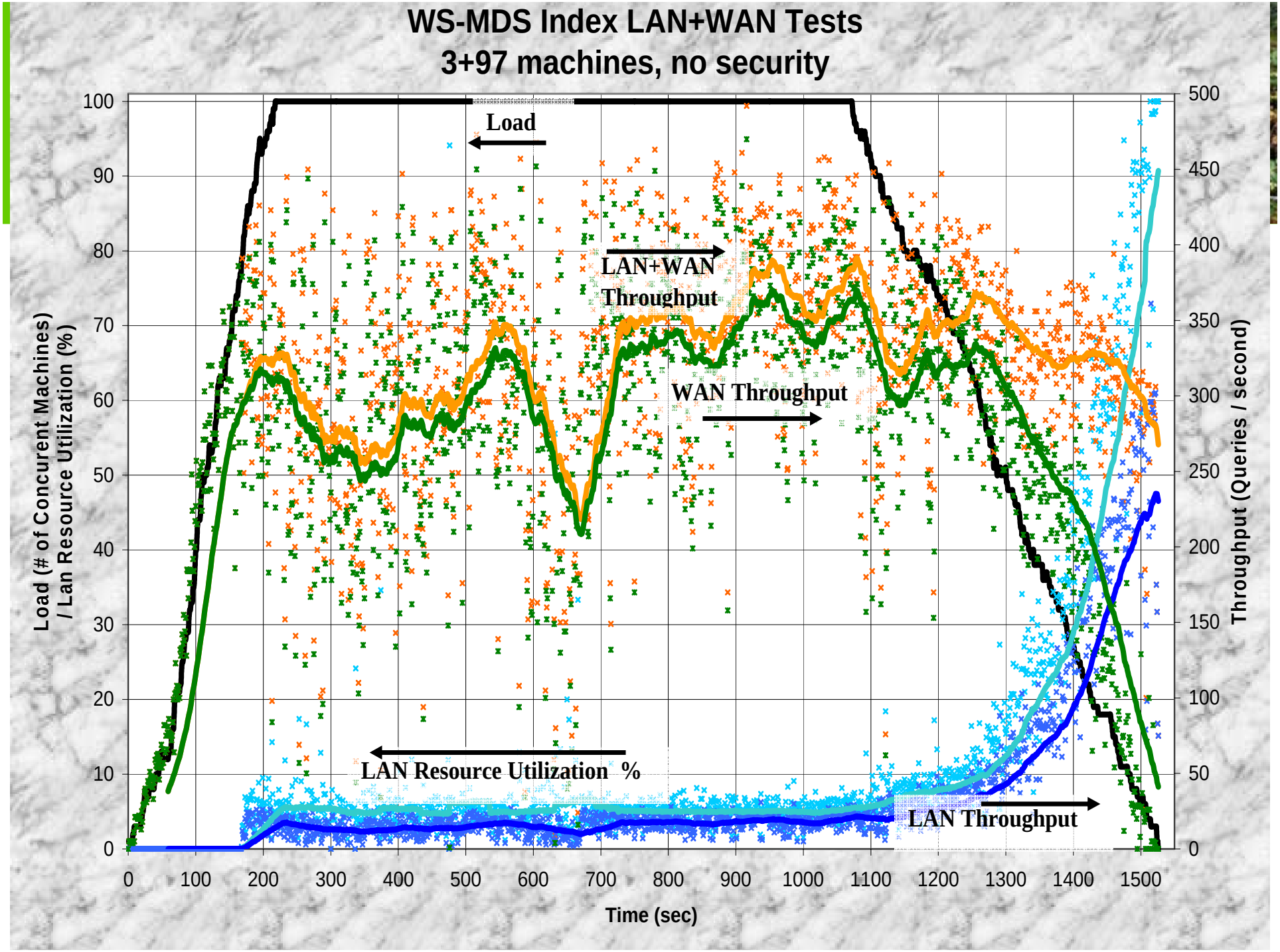
Performance of GT®

- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2

WS-MDS Index WAN Tests: 288 machines, no security

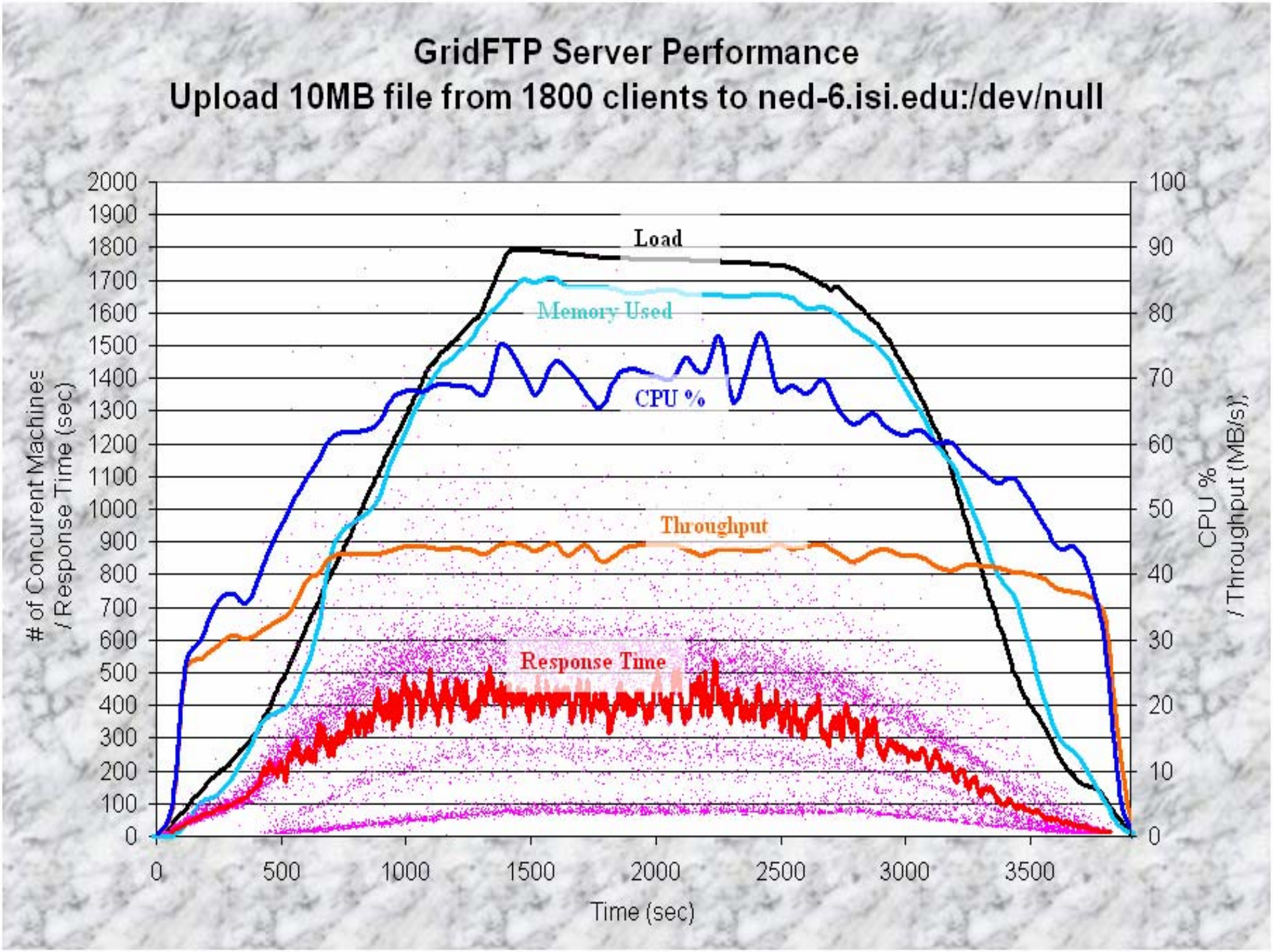
Experiment Length (sec)	Total # of Transactions	Min / Median / Average / Max Transactions per sec	Min / Median / Average / Max Response Time (ms)
2772	610K	0 / 222 / 220 / 511	51 / 415 / 531 / 22514

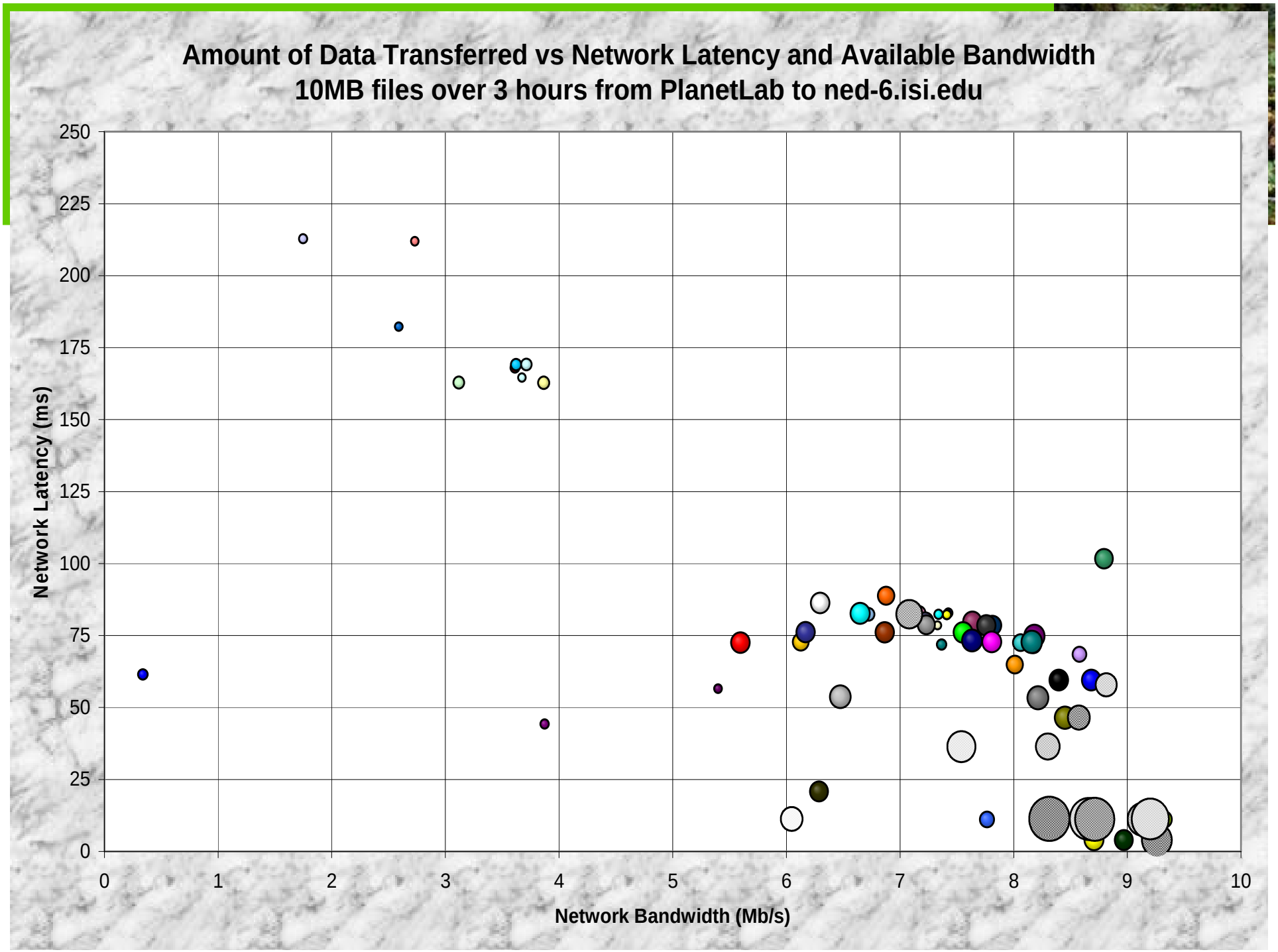


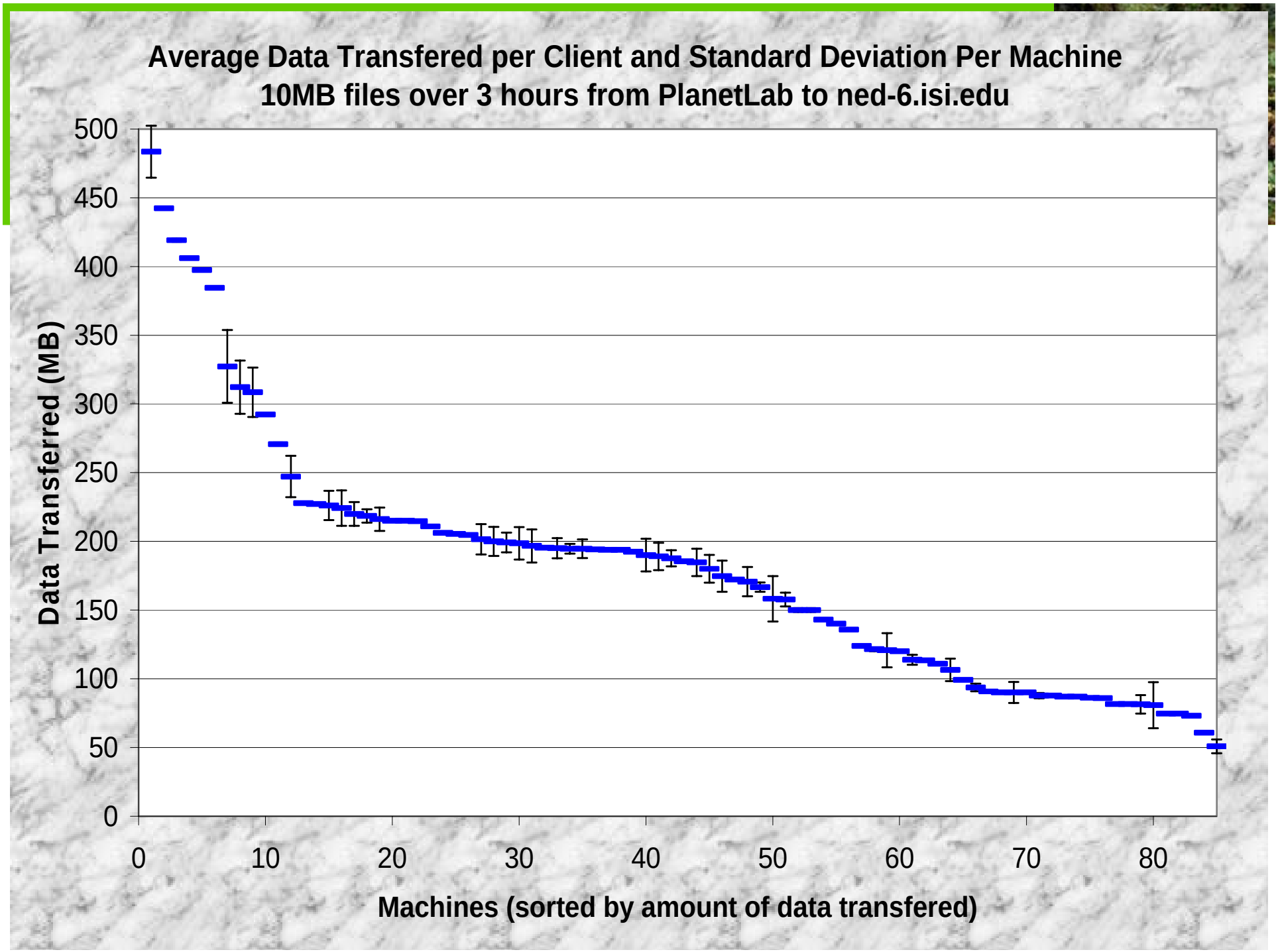

WS-MDS Index LAN+WAN Tests
3+97 machines, no security
Load
LAN+WAN
Throughput
WAN Throughput
LAN Resource Utilization %
LAN Throughput
Load (# of Concurent Machines)
/ Lan Resource Utilization (%)
Throughput (Queries / second)
Time (sec)
0
10
20
30
40
50
60
70
80
90
100
0
50
100
150
200
250
300
350
400
450
500
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500

Performance of GT®

- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2


GridFTP Server Performance
Upload 10MB file from 1800 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
CPU % / Throughput (MB/s)
Time (sec)
Load
Memory Used
CPU %
Throughput
Response Time
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
0
10
20
30
40
50
60
70
80
90
100
0
500
1000
1500
2000
2500
3000
3500

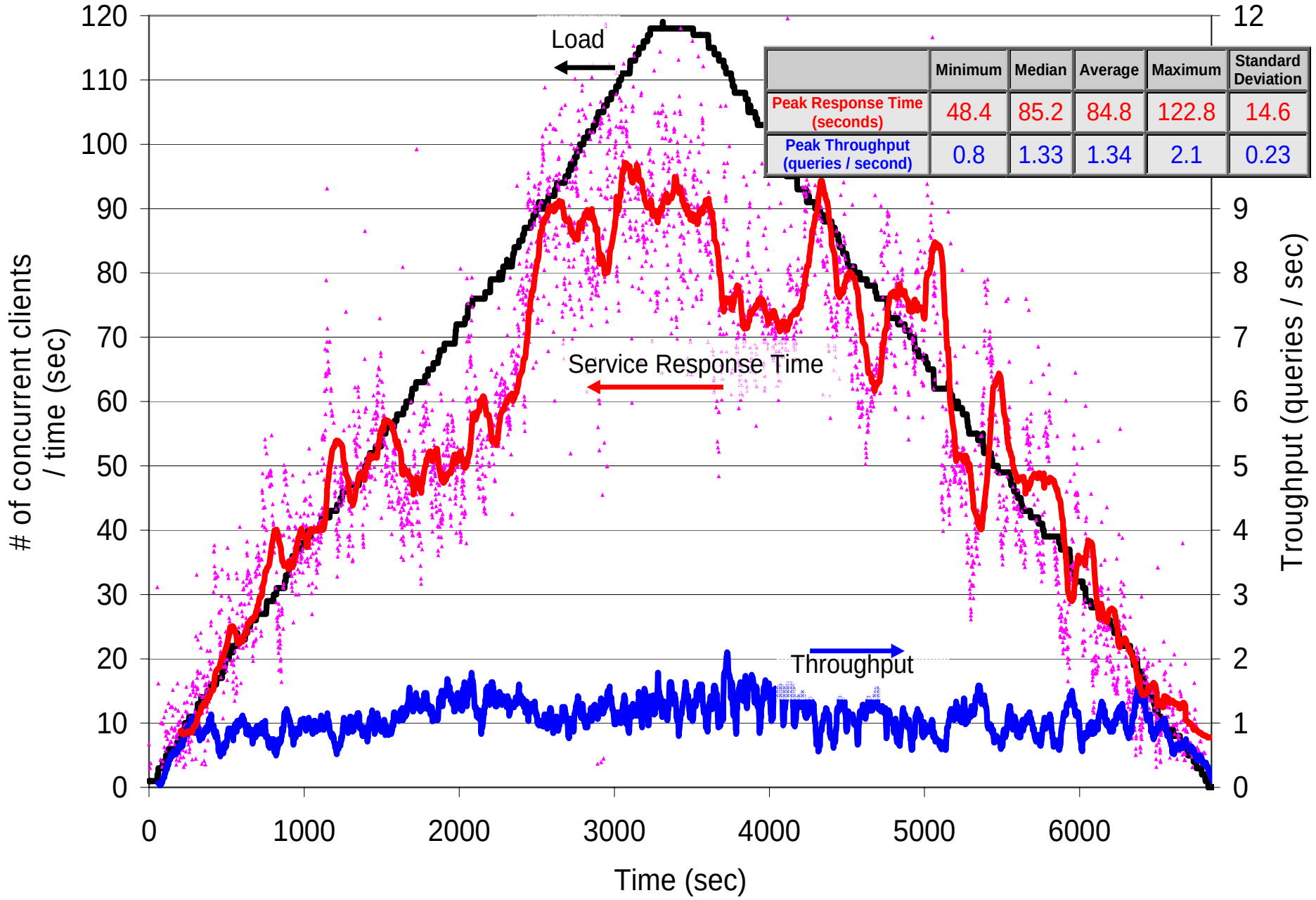

Amount of Data Transferred vs Network Latency and Available Bandwidth
10MB files over 3 hours from PlanetLab to ned-6.isi.edu
Network Latency (ms)
250
225
200
175
150
125
100
75
50
25
0
0
1
2
3
4
5
6
7
8
9
10
Network Bandwidth (Mb/s)


Average Data Transfered per Client and Standard Deviation Per Machine
10MB files over 3 hours from PlanetLab to ned-6.isi.edu
Data Transferred (MB)
500
450
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
Machines (sorted by amount of data transfered)

Performance of GT®

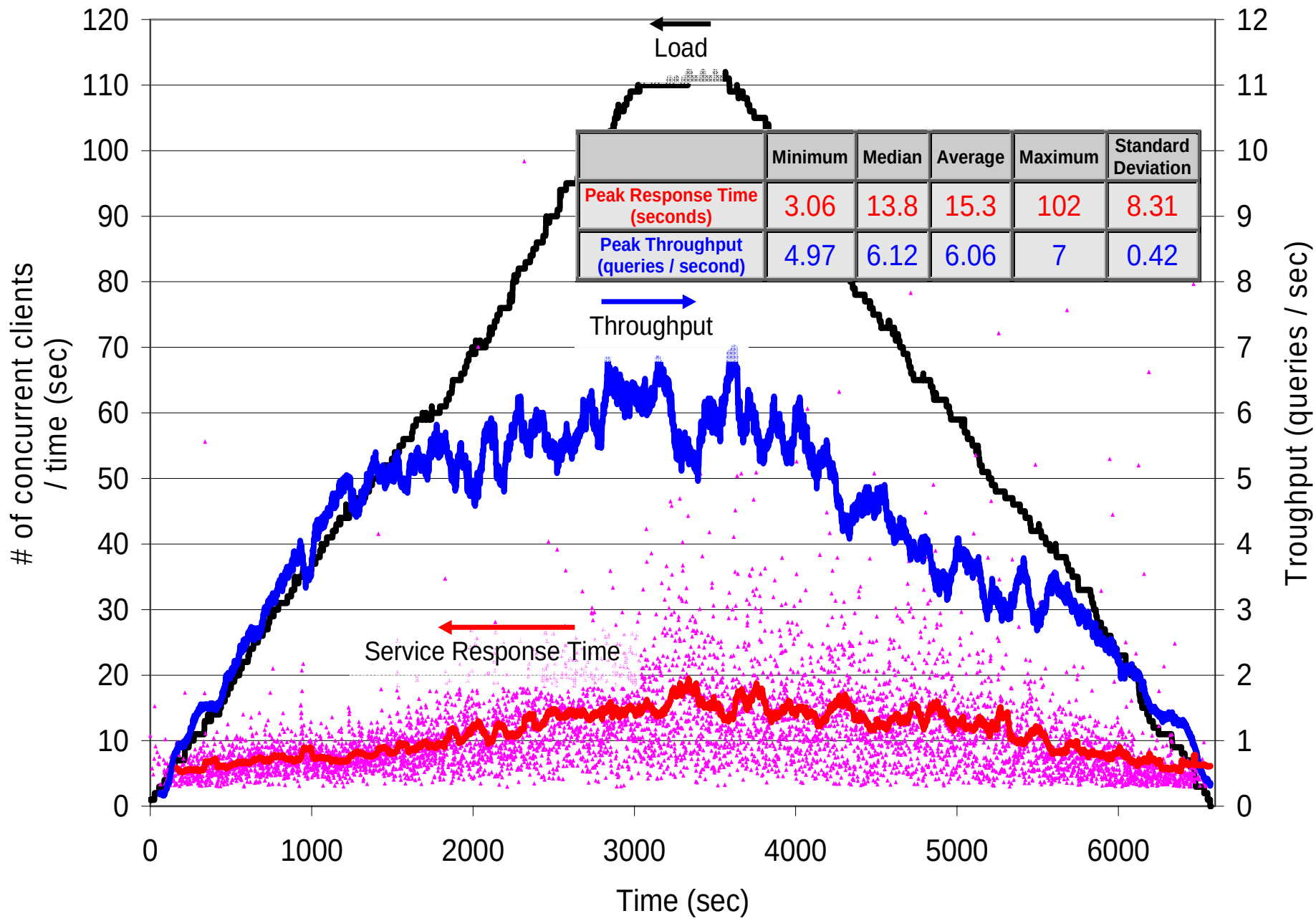
- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2

DI-GRUBER GT4: 1DP/120CL



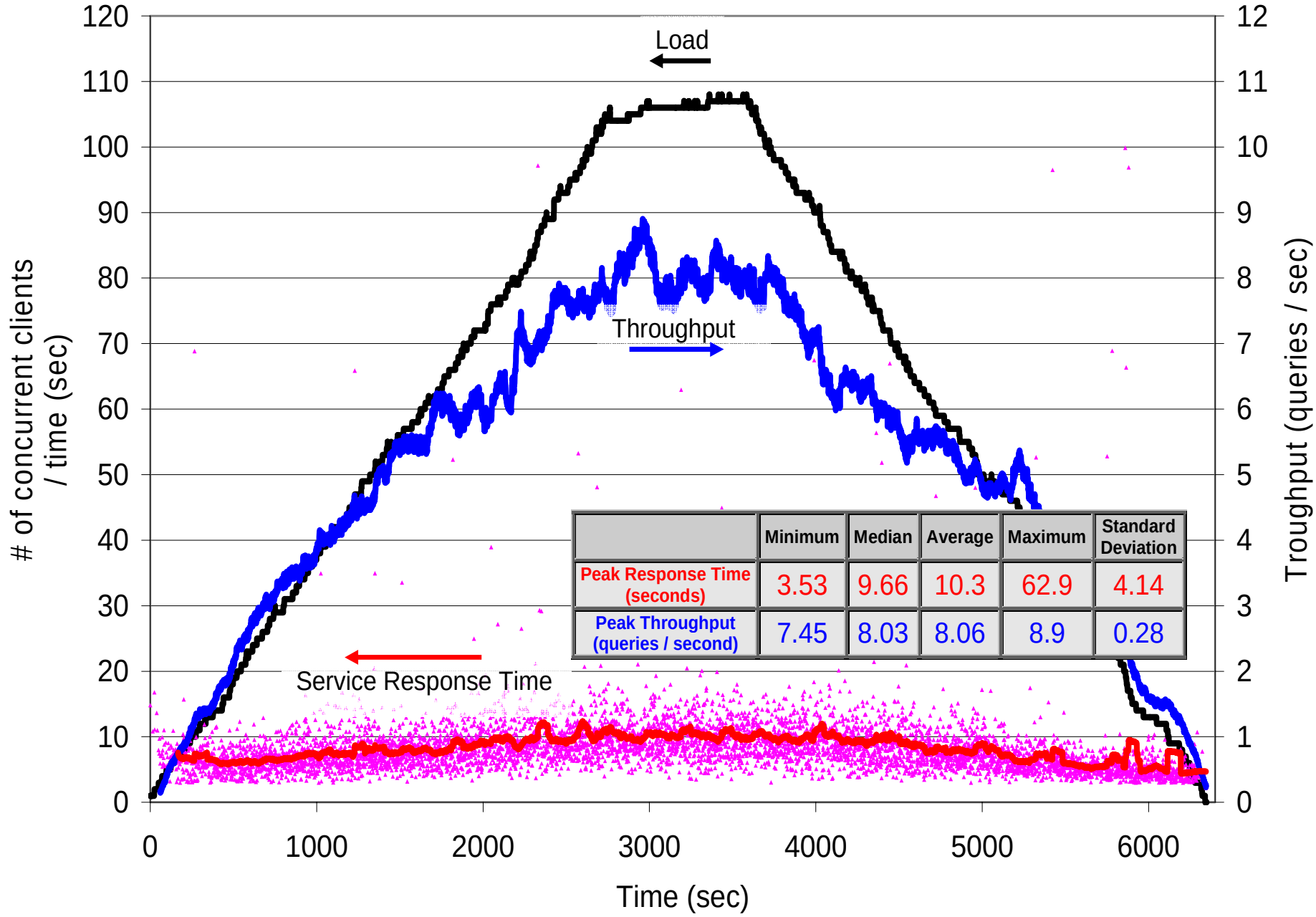
	Minimum	Median	Average	Maximum	Standard Deviation
Peak Response Time (seconds)	48.4	85.2	84.8	122.8	14.6
Peak Throughput (queries / second)	0.8	1.33	1.34	2.1	0.23

DI-GRUBER GT3: 3DP/120CL



	Minimum	Median	Average	Maximum	Standard Deviation
Peak Response Time (seconds)	3.06	13.8	15.3	102	8.31
Peak Throughput (queries / second)	4.97	6.12	6.06	7	0.42

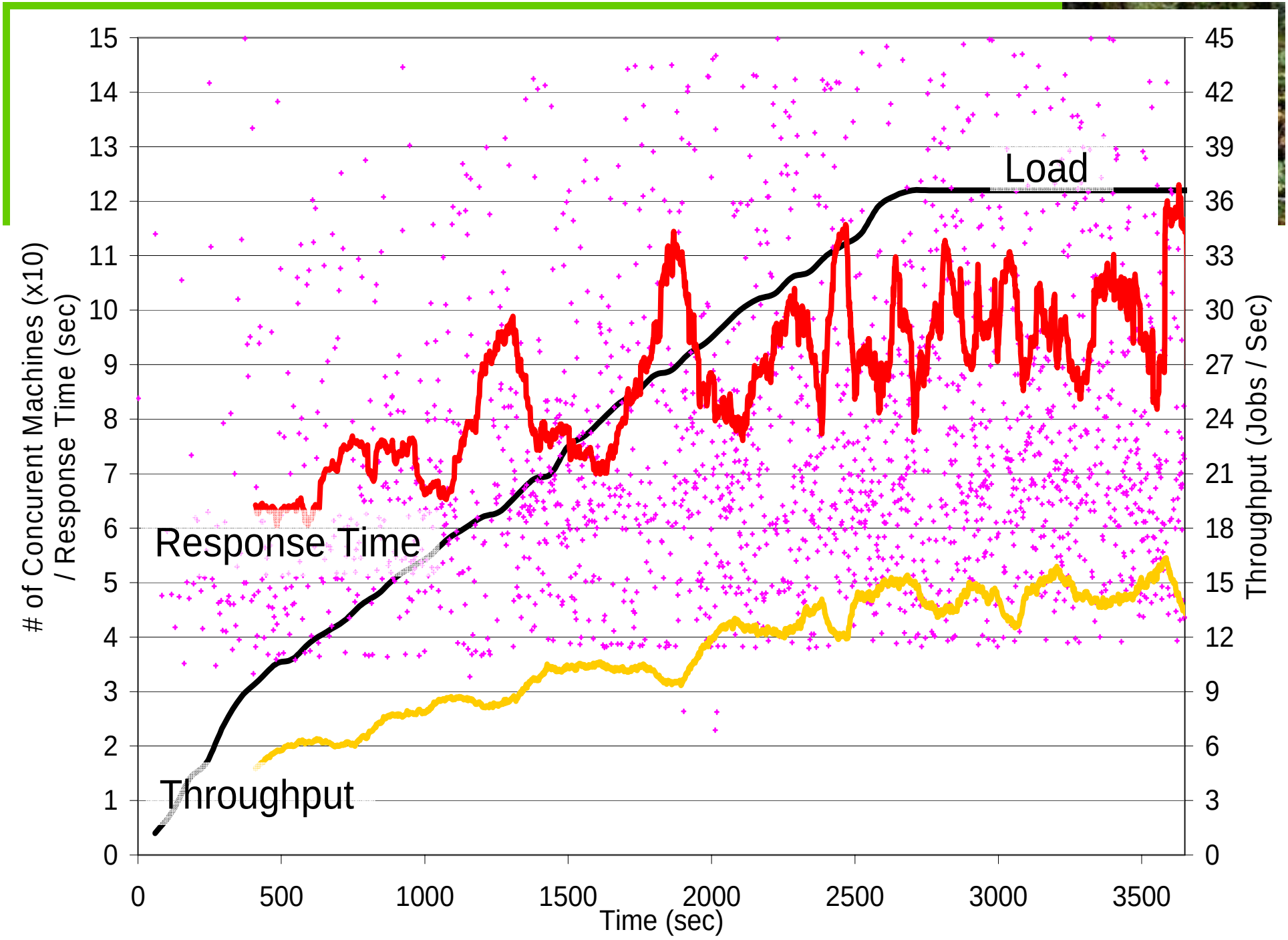
DI-GRUBER GT3: 10DP/120CL



	Minimum	Median	Average	Maximum	Standard Deviation
Peak Response Time (seconds)	3.53	9.66	10.3	62.9	4.14
Peak Throughput (queries / second)	7.45	8.03	8.06	8.9	0.28

Performance of GT®

- Job submission: pre-WS GRAM and WS-GRAM included with GT® 3.2 and 3.9.4
- Information services: the scalability and performance of the WS-MDS Index bundled with GT® 3.9.5
- A file transfer protocol: the scalability and fairness of the GridFTP server included with the GT® 3.9.5
- Grid Services:
  - DI-GRUBER, a distributed usage SLA-based broker based on the GT® 3.2 and 3.9.5
  - Instance creation and message passing performance in the GT® 3.2


Load
Response Time
Throughput
# of Concurent Machines (x10)
/ Response Time (sec)
Throughput (Jobs / Sec)
Time (sec)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
0
500
1000
1500
2000
2500
3000
3500

Contributions: Performance Testing of GT

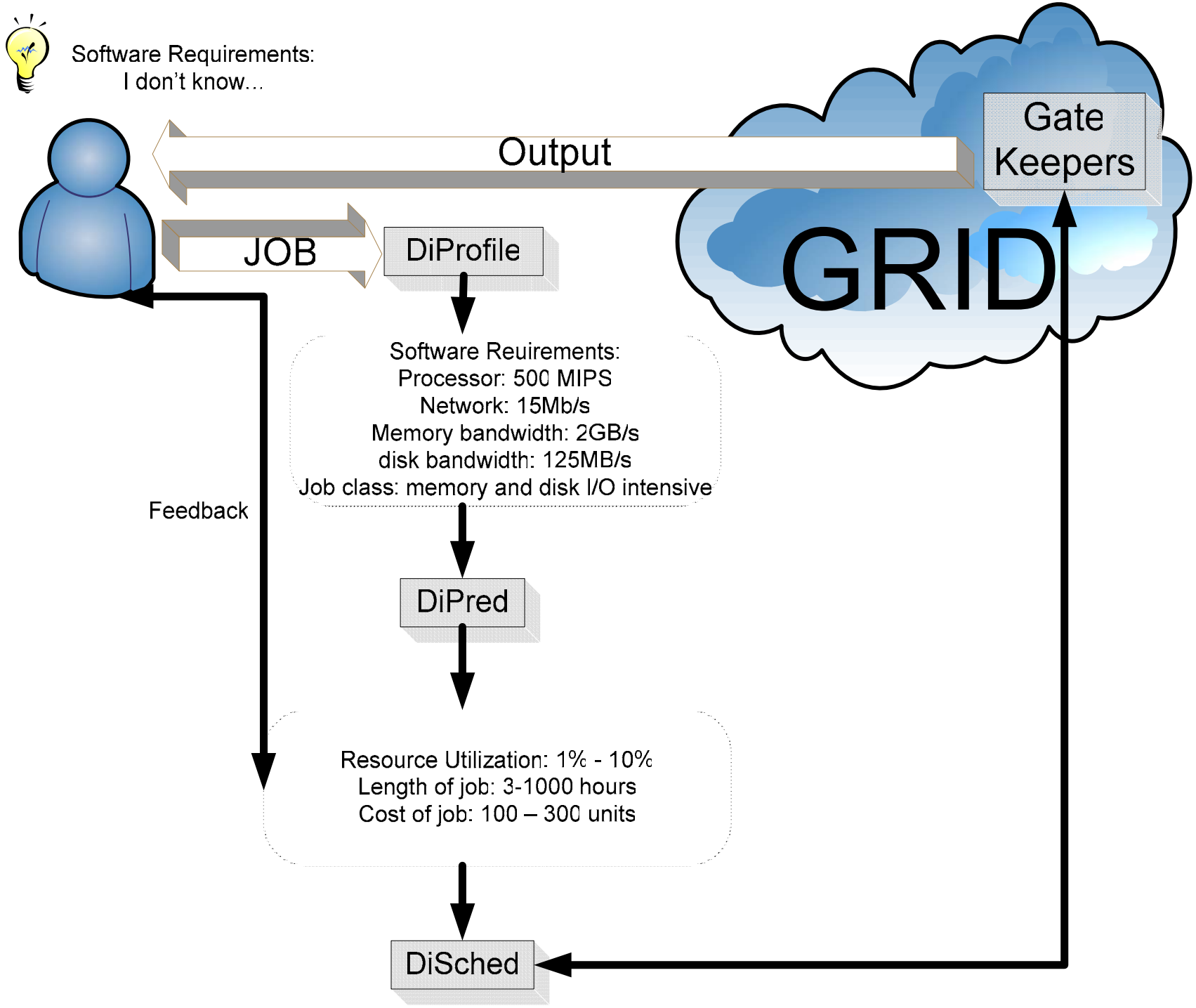
- Quantified the performance gain or loss among different versions or implementations
- Discovered upper limits on scalability and performance
- Gave users a tool for better resource planning
- Gave developers feedback

Contributions: DiPerF

- Allows large scale testing of grid services, web services, and network services to be done in both LAN and WAN environments
  - Service capacity
  - Service scalability
  - Resource distribution among clients
  - Accurate client views of service performance
  - How network latency or geographical distribution affects client/service performance
  - Allows the collection of the appropriate metrics to build analytical models
- DiPerF has been automated to the extent that once configured, the framework will automatically do the following steps:
  - check what machines or resources are available for testing
  - deploy the client code on the available machines
  - perform time synchronization
  - run the client code in a controlled and predetermined fashion
  - collect performance metrics from all the clients
  - stop and clean up the client code from the remote resources
  - aggregate the performance metrics at a central location
  - summarize the results
  - generates graphs depicting the aggregate performance of the clients and tested service

Future Work

- Analytical Models
  - Large data sets…
  - AI and Machine Learning techniques:
    - Neural networks, decision trees, support vector machines, regression, statistical time series, wavelets, polynomial approximations, etc…
- Resource Management
  - Job Profiling
  - Co-scheduling
  - Predictive Scheduling


Software Requirements:
I don't know...
Output
Gate Keepers
GRID
JOB
DiProfile
Software Reuirements:
Processor: 500 MIPS
Network: 15Mb/s
Memory bandwidth: 2GB/s
disk bandwidth: 125MB/s
Job class: memory and disk I/O intensive
Feedback
DiPred
Resource Utilization: 1% - 10%
Length of job: 3-1000 hours
Cost of job: 100 – 300 units
DiSched

Related Publications & Tech Reports

Published / Technical Reports

- C. Dumitrescu, **I. Raicu**, M. Ripeanu, I. Foster. "*DiPerF: an automated DIstributed PERformance testing Framework*", 5th International IEEE/ACM Workshop in Grid Computing, 2004, Pittsburg, PA.
- **I. Raicu**. "*Decreasing End-to-End Job Execution Times by Increasing Resource Utilization using Predictive Scheduling in the Grid*", Technical Report, Grid Computing Seminar, Department of Computer Science, University of Chicago, March 2005.
- C. Dumitrescu, I. Foster, **I. Raicu**. "*A Scalability and Performance Evaluation of a distributed Usage SLA-based Broker in Large Grid Environments*", GriPhyN/iVDGL Technical Report, March 2005.

Under Review

- C. Dumitrescu, **I. Raicu**, I. Foster. "*DI-GRUBER: A Distributed Approach for Grid Resource Brokering*", submitted for review to IEEE/ACM SC 2005.
- B. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, **I. Raicu**, I. Foster. "*Zebra: The Globus Striped GridFTP Framework and Server*", submitted for review to IEEE/ACM SC 2005.
- C. Dumitrescu, **I. Raicu**, I. Foster. "*Performance Measurements in Running Workloads over a Grid*", submitted for review to IEEE/ACM SC 2005.

Work in Progress

- **I. Raicu**, C. Dumitrescu, I. Foster. "*A Performance Analysis of the Globus Toolkit®'s Job Submission, GRAM*", will submit to IEEE/ACM Grid 2005.
- **I. Raicu**, C. Dumitrescu, I. Foster. "*A Performance Evaluation of WS-MDS in the Globus Toolkit®*" , will submit to IEEE/ACM Grid 2005.
- **I. Raicu**, C. Dumitrescu, I. Foster. "*A Performance Study of the Globus Toolkit®*", will submit to a journal.
- C. Dumitrescu, **I. Raicu**, M. Ripeanu, I. Foster. "*Extending a distributed usage SLA resource broker with overlay networks to support Large Dynamic Grid Environments*", will submit to ICSOC 2005.

Other Publications

Network Protocols:

- S. Zeadally, R. Wasseem, **I. Raicu**, "*Comparison of End-System IPv6 Protocol Stacks*", IEE Proceedings Communications, Special issue on Internet Protocols, Technology and Applications (VoIP), June 2004.
- Sherali Zeadally, **Ioan Raicu**. "*Evaluating IPV6 on Windows and Solaris*", IEEE Internet Computing, May-June 2003.
- **I. Raicu**, S. Zeadally. "*Impact of IPv6 on End-User Applications*", IEEE International Conference on Telecommunications 2003, ICT'2003, Feb 2003, Tahiti Papeete, French Polynesia.
- **I. Raicu**, S. Zeadally. "*Evaluating IPv4 to IPv6 Transition Mechanisms*" , IEEE ICT'2003, Feb 2003, Tahiti Papeete, French Polynesia.
- **I. Raicu**. "*An Empirical Analysis of Internet Protocol version 6 (IPv6)*", Wayne State University, Computer Science Department, MS Thesis, May 2002, Detroit, Michigan.

Wireless Sensor Networks:

- **I. Raicu**, L. Schwiebert, S. Fowler, S.K.S. Gupta. "*Local Load Balancing for Globally Efficient Routing in Wireless Sensor Networks*", Intrnl. Journal of Distributed Sensor Network, 2005.
- **I. Raicu**, L. Schwiebert, S. Fowler, S.K.S. Gupta. "*e3D: An Energy-Efficient Routing Algorithm for Wireless Sensor Networks*", IEEE ISSNIP 2004 (The Intrnl. Conference on Intelligent Sensors, Sensor Networks and Information Processing), Melbourne, Australia, December 2004.
- **I. Raicu**. "*Efficient Even Distribution of Power Consumption in Wireless Sensor Networks*", ISCA 18th Intrnl. Conf. on Computers and Their Applications, CATA 2003, March 2003, Honolulu, Hawaii, USA.
- **I. Raicu**, O. Richter, L. Schwiebert, S. Zeadally. "*Using Wireless Sensor Networks to Narrow the Gap between Low-Level Information and Context-Awareness*", CATA 2002, San Francisco, CA, April, 2002.

Questions?

- More info on thesis:
  - http://people.cs.uchicago.edu/~iraicu/research/uchicago/ms_thesis/
- More info on DiPerF:
  - http://diperf.cs.uchicago.edu/
- Questions?

THE UNIVERSITY OF CHICAGO

ARGONNE NATIONAL LABORATORY