DiPerF: automated DIstributed PERformance testing Framework

Ioan Raicu, Catalin Dumitrescu,
Matei Ripeanu, Ian Foster

Distributed Systems Laboratory
Computer Science Department
University of Chicago

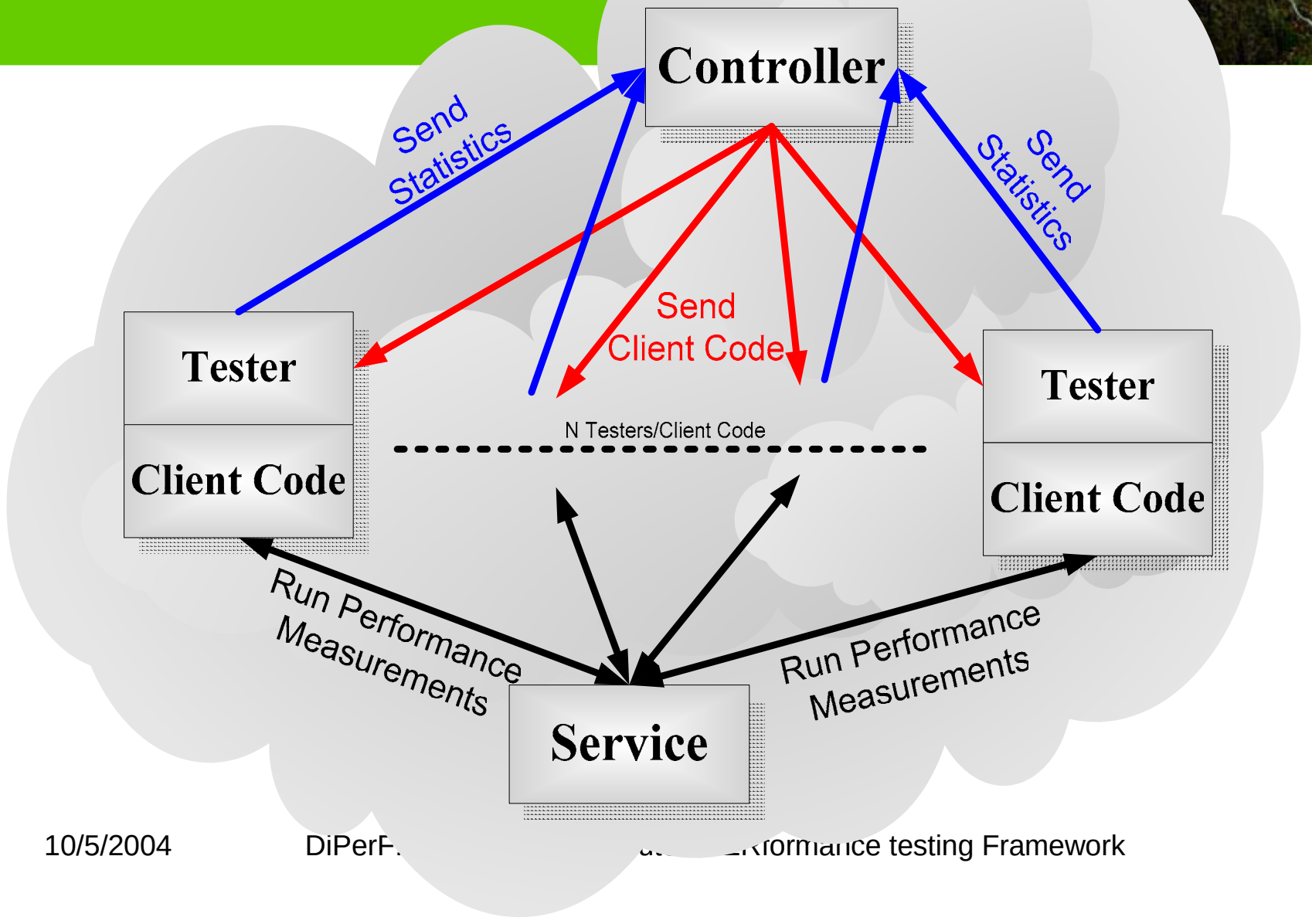
Introduction

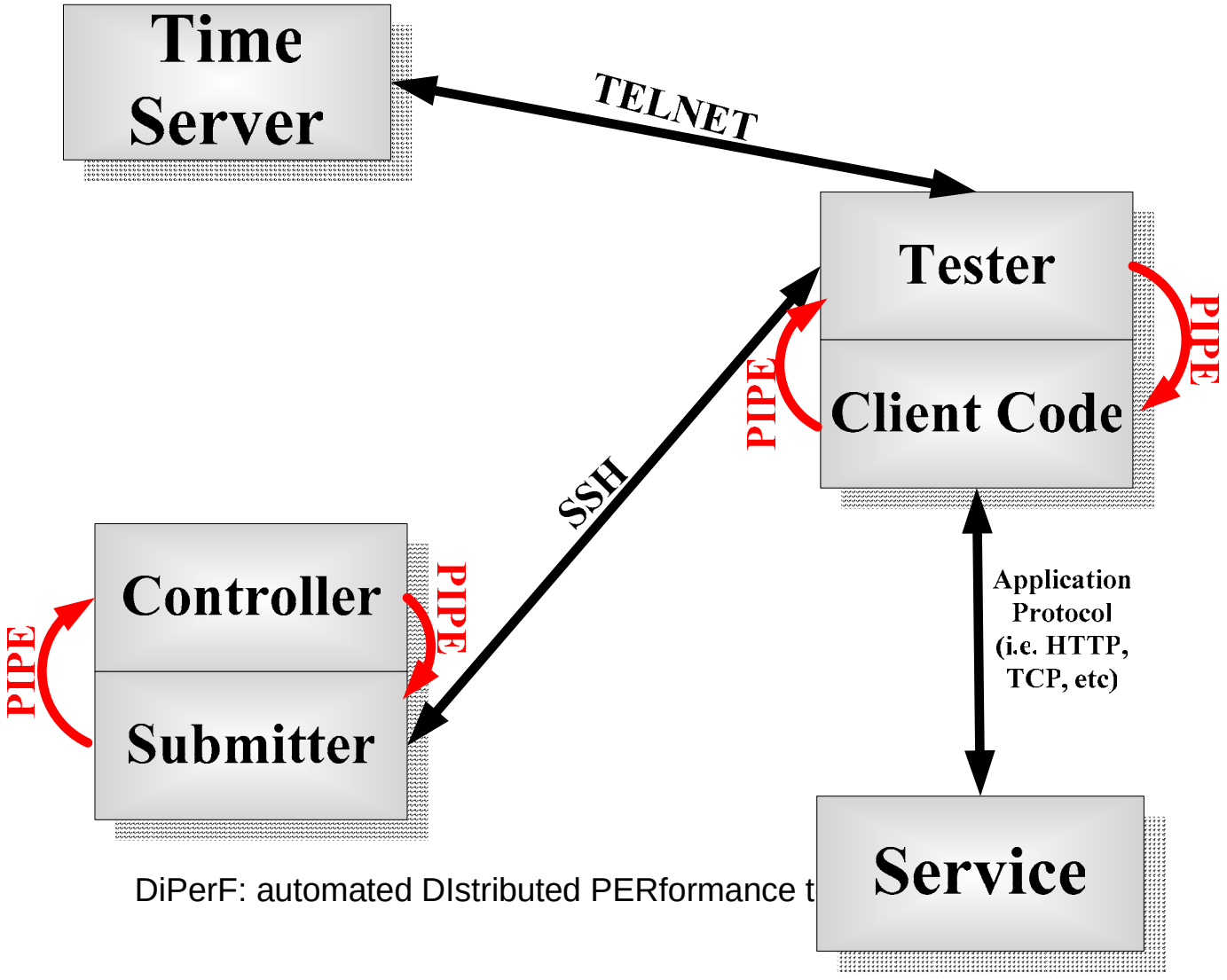
- Goals
  - Simplify and automate distributed performance testing
    - grid services
    - web services
    - network services
  - Define a comprehensive list of performance metrics
  - Produce accurate client views of service performance
  - Create analytical models of service performance
- Framework implementation
  - Grid3
  - PlanetLab
  - NFS style cluster (UChicago CS Cluster)

Framework

- Coordinates a distributed pool of machines
  - Tested with 1600 clients
  - Could scale even further with a lighter weight communication protocol (i.e. TCP, UDP)
- Controller
  - Receives the address of the service and a client code
  - Distributes the client code across all machines in the pool
  - Gathers, aggregates, and summarizes performance statistics
- Tester
  - Receives client code
  - Runs the code and produce performance statistics
  - Sends back to “controller” statistic report

Architecture Overview

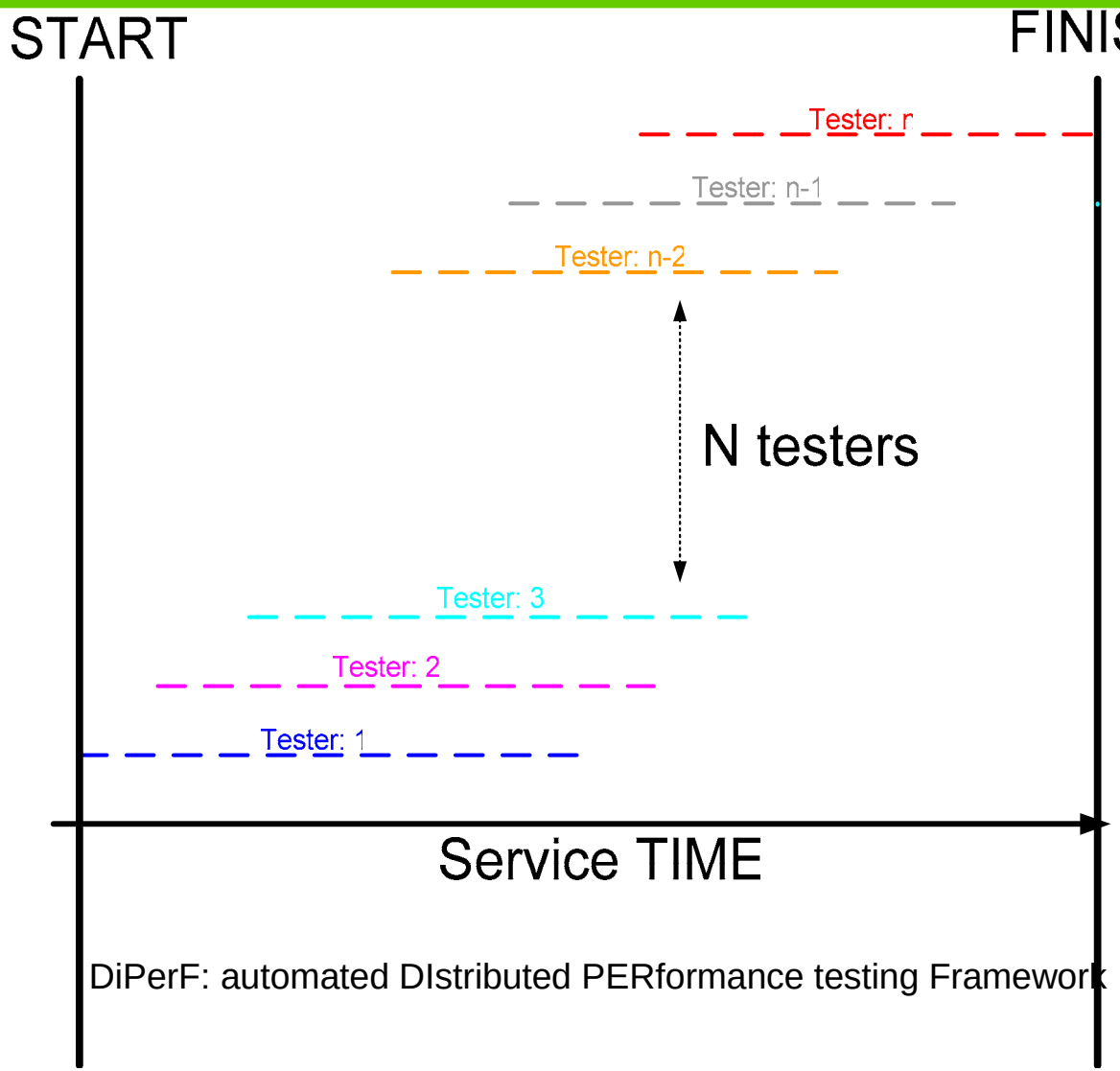



Time Server
TELNET
Tester
Client Code
PIPE
PIPE
SSH
Controller
Submitter
PIPE
PIPE
Application Protocol (i.e. HTTP, TCP, etc)
Service

Time Synchronization

- Time synchronization needed at the testers for data aggregation at controller?
  - Distributed approach:
    - Tester uses Network Time Protocol (NTP) to synchronize time
  - 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

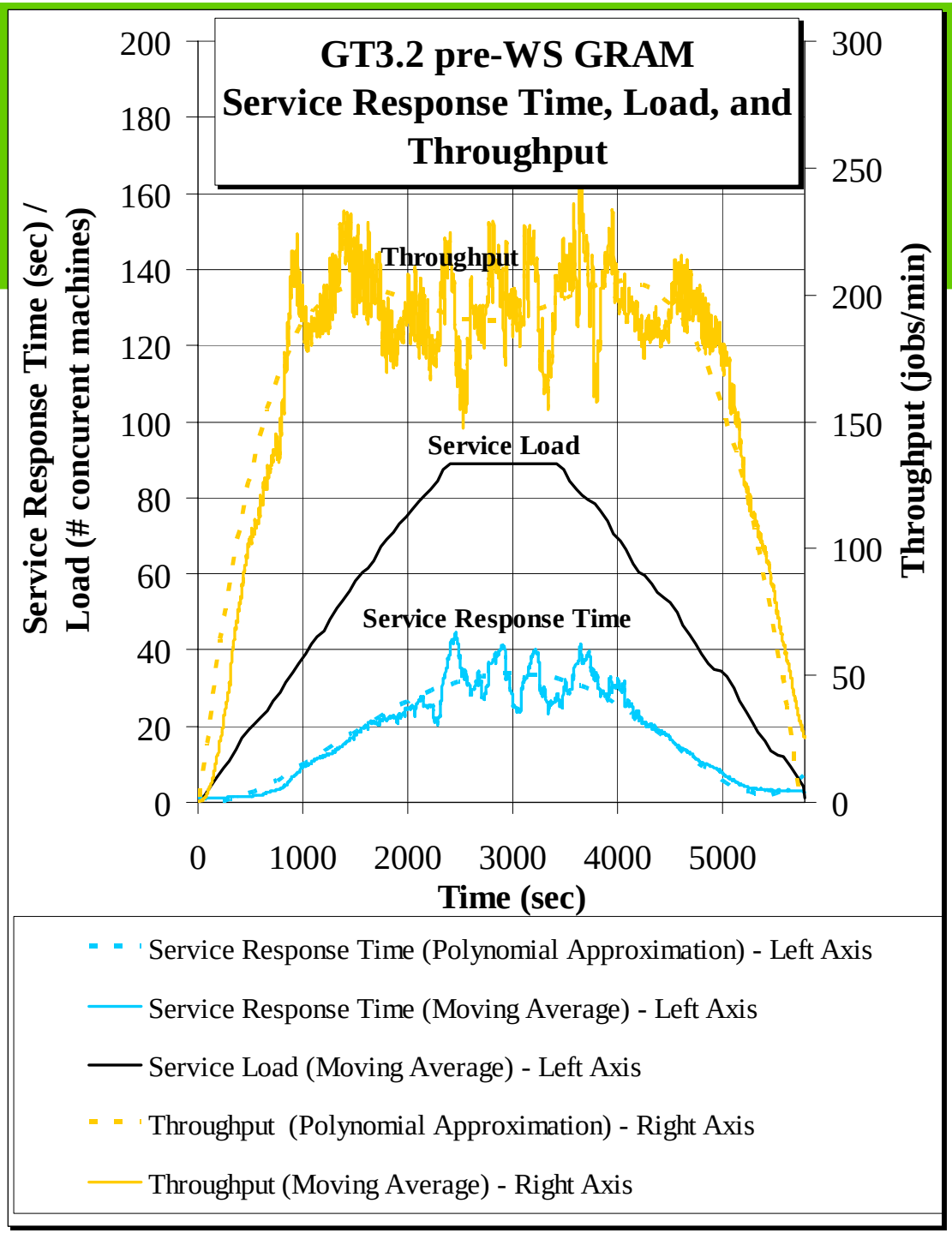


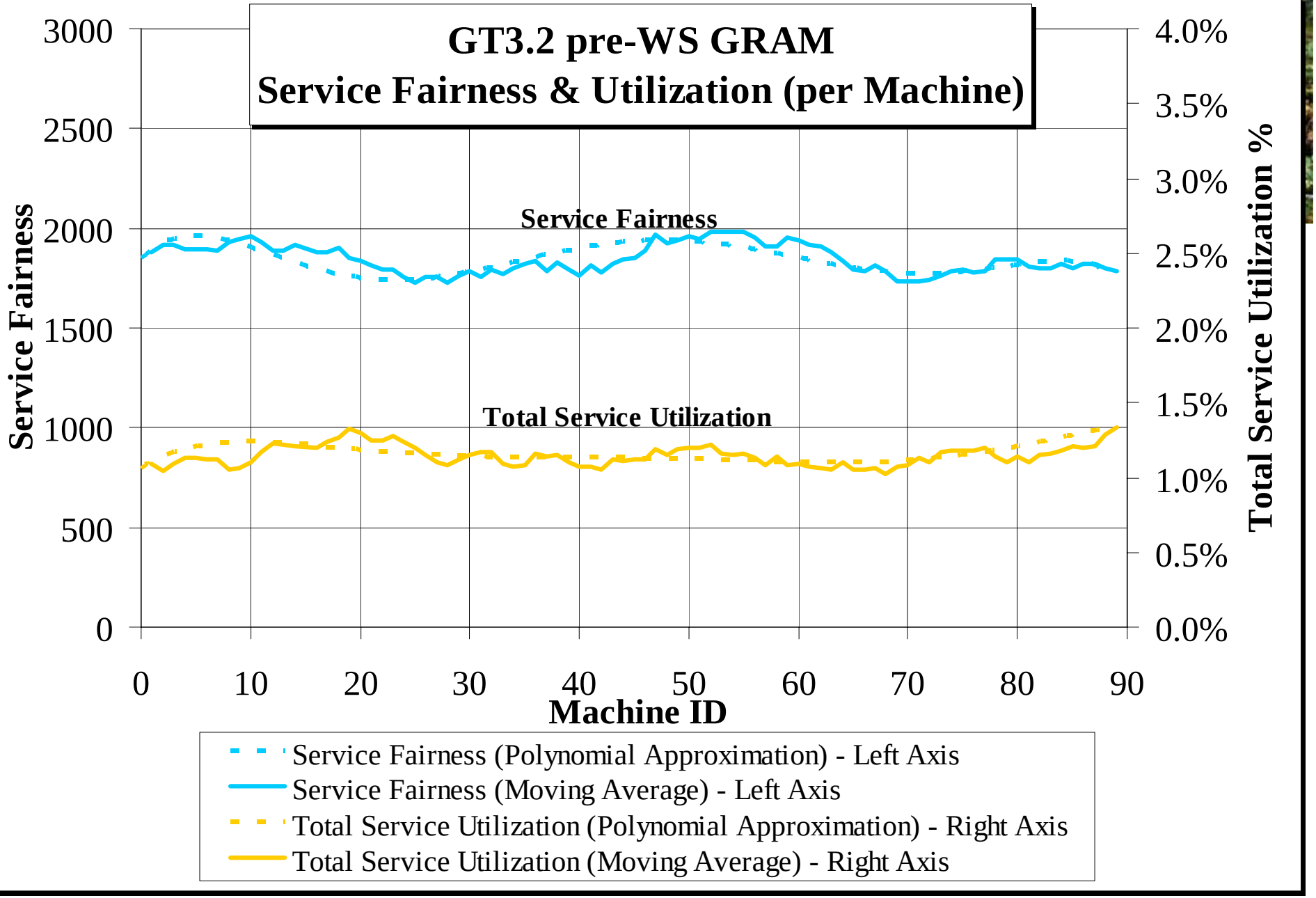
Performance Metrics

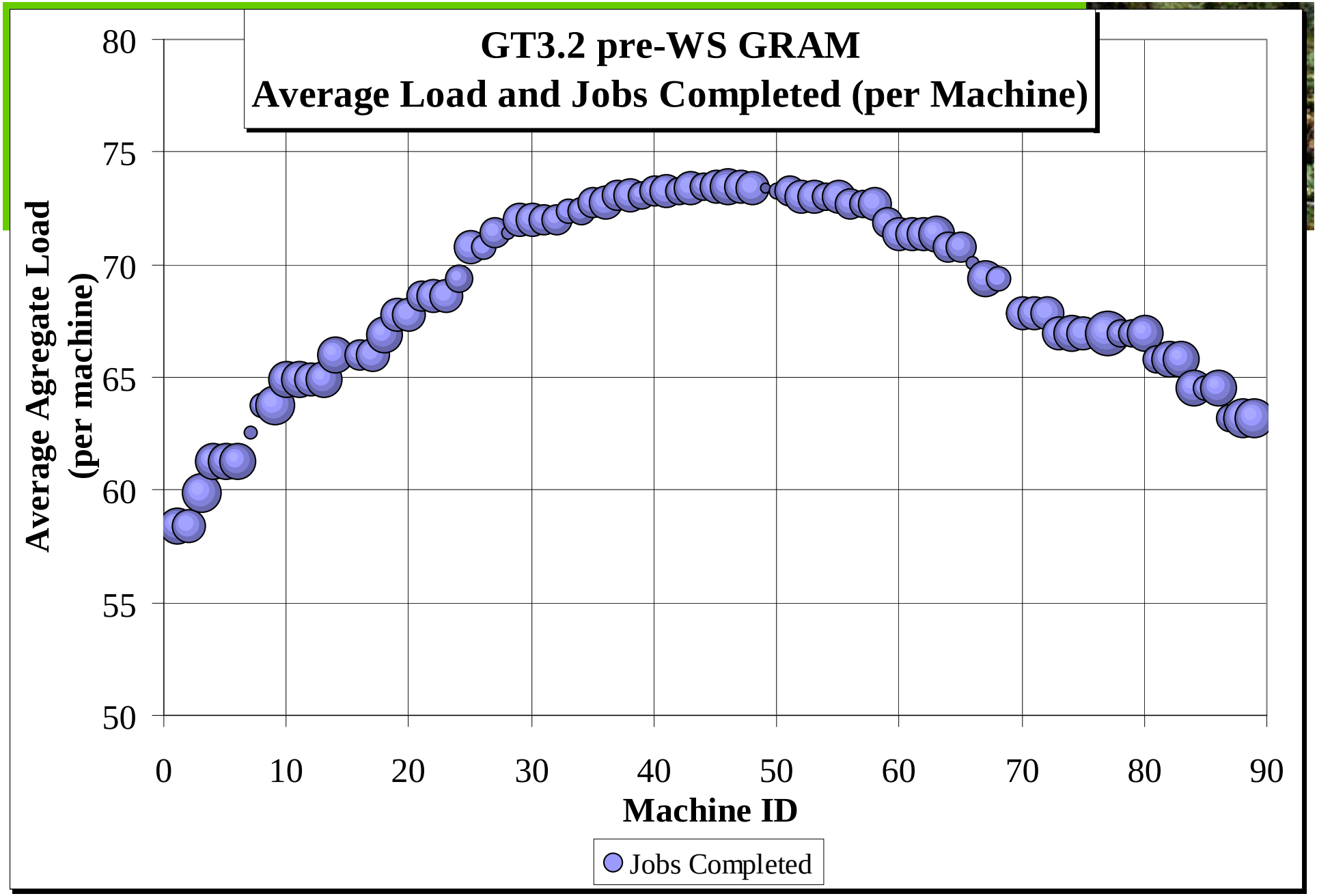
- ***service response time:***
  - the time from when a client issues a request to when the request is completed minus the network latency and minus the execution time of the client code
- ***service throughput*:**
  - number of jobs completed successfully by the service averaged over a short time interval
- ***offered load*:**
  - number of concurrent service requests (per second)
- ***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

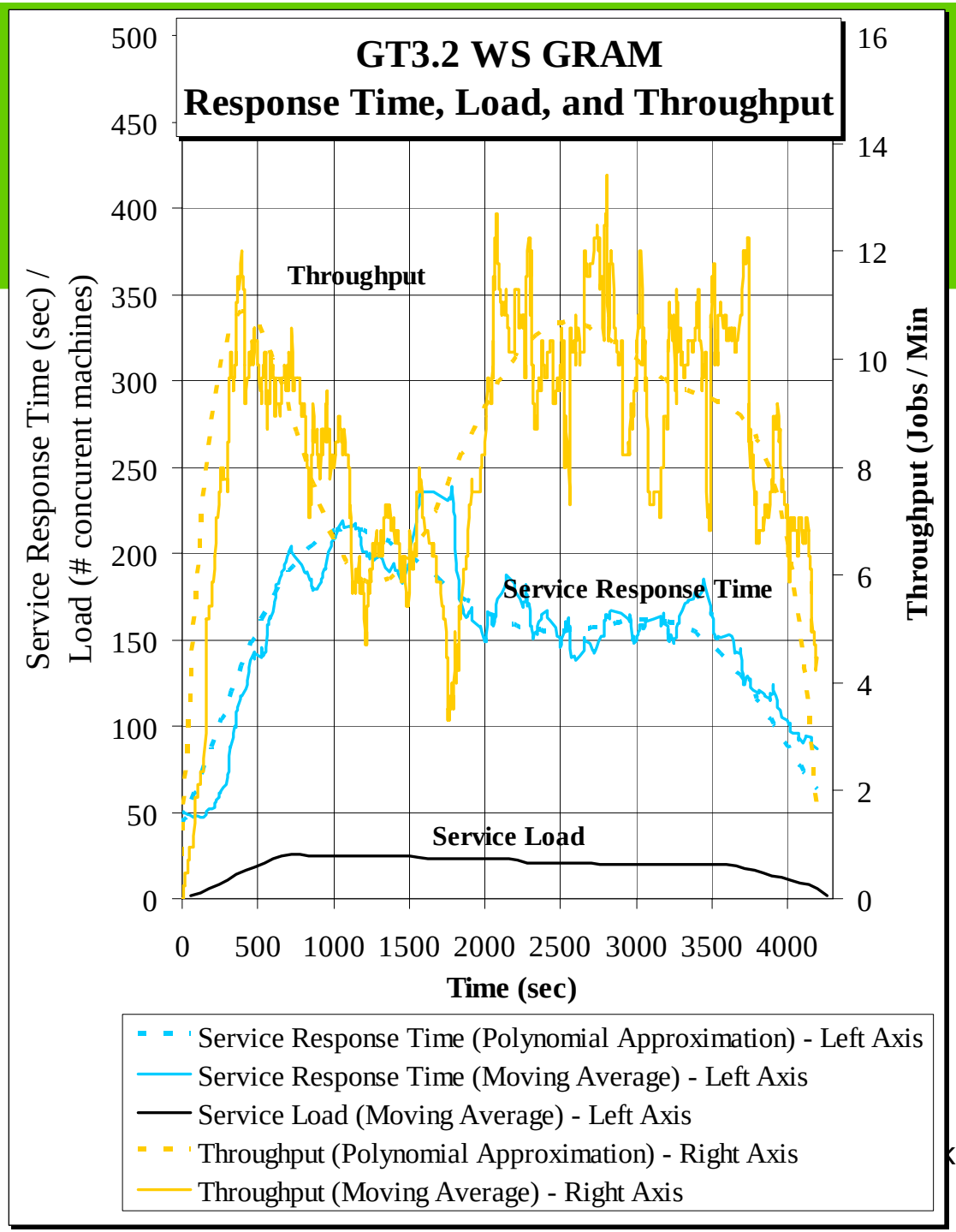
Services Tested

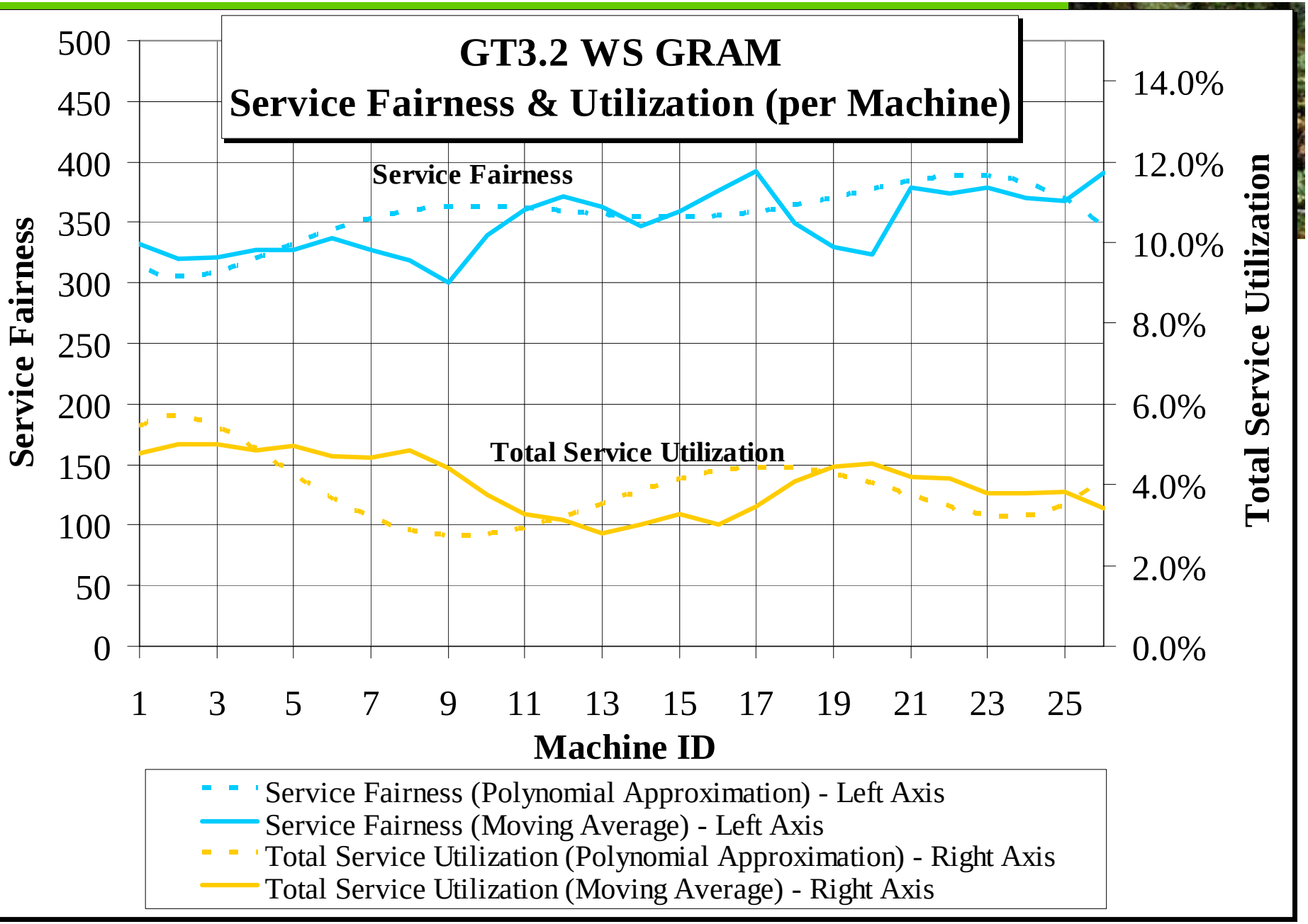
- GT3.2 pre-WS GRAM
  - job submission via Globus Gatekeeper 2.4.3 using Globus Toolkit 3.2 (C version)
  - a gatekeeper listens for job requests on a specific machine
  - performs mutual authentication by confirming the user's identity, and proving its identity to the user
  - starts a job manager process as the local user corresponding to authenticated remote user
  - the job manager invokes the appropriate local site resource manager for job execution and maintains a HTTPS channel for information exchange with the remote user
- GT3.2 WS GRAM
  - job submission using Globus Toolkit 3.2 (Java version)
  - a client submits a *createService* request which is received by the Virtual Host Environment Redirector
  - attempt to forward the *createService* call to a User Hosting Environment (UHE) where mutual authentication / authorization can take place
  - if the UHE is not created, the Launch UHE module is invoked
  - WS GRAM then creates a new Managed Job Service (MJS)
  - MJS submits the job into a back-end scheduling system
- HTTP
  - client used "httperf" to retrieve a file over HTTP on an Apache HTTP server
- MonaLisa
  - monitoring grid webservice

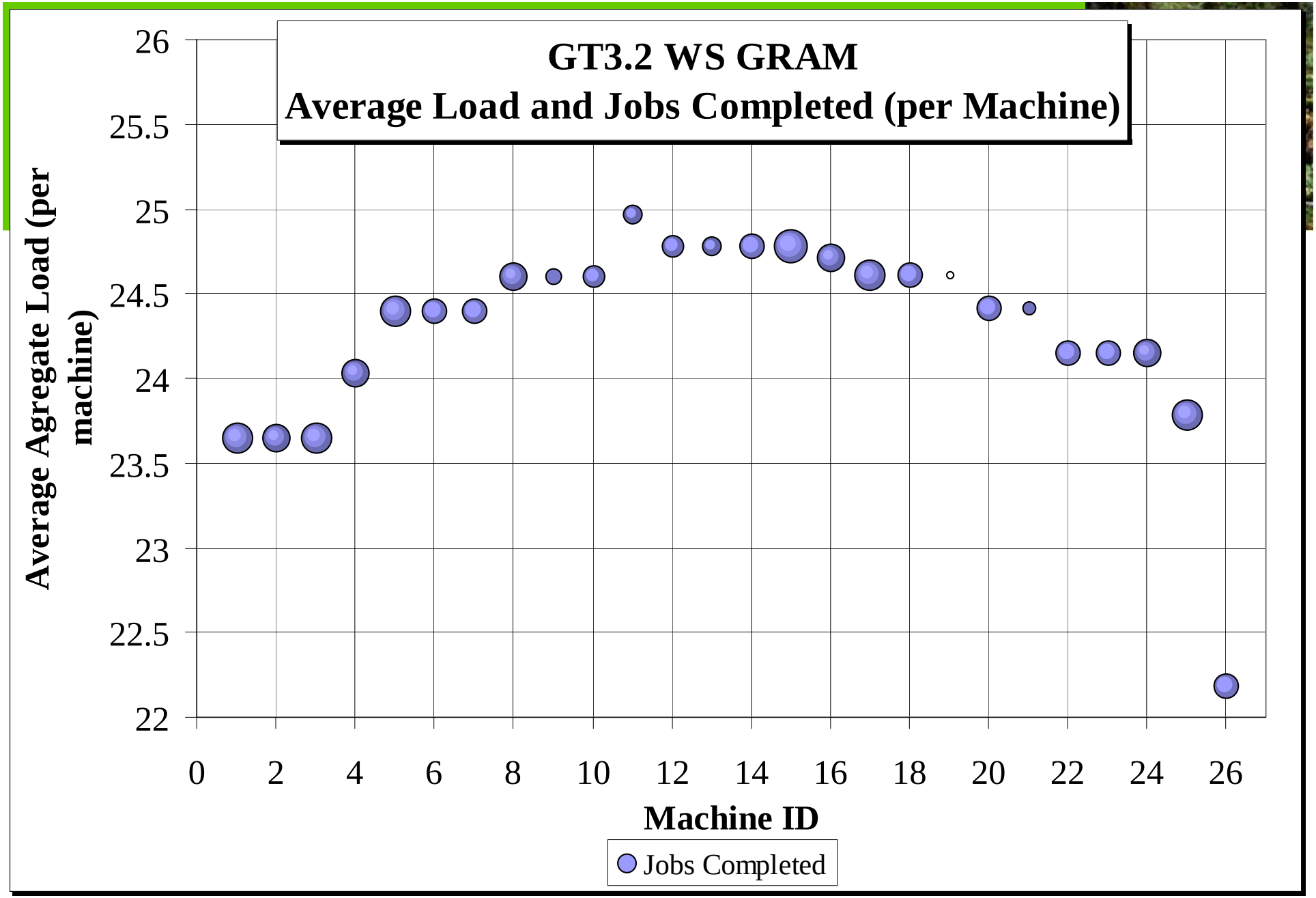

GT3.2 pre-WS GRAM
Service Response Time, Load, and Throughput
Throughput
Service Load
Service Response Time
Service Response Time (sec) / Load (# concurent machines)
Throughput (jobs/min)
Time (sec)
0
20
40
60
80
100
120
140
160
180
200
0
50
100
150
200
250
300
0
1000
2000
3000
4000
5000
Service Response 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 pre-WS GRAM
Service Fairness & Utilization (per Machine)
Service Fairness
Total Service Utilization
Service Fairness
Total Service Utilization %
Machine ID
3000
2500
2000
1500
1000
500
0
4.0%
3.5%
3.0%
2.5%
2.0%
1.5%
1.0%
0.5%
0.0%
0
10
20
30
40
50
60
70
80
90
Service Fairness (Polynomial Approximation) - Left Axis
Service Fairness (Moving Average) - Left Axis
Total Service Utilization (Polynomial Approximation) - Right Axis
Total Service Utilization (Moving Average) - Right Axis


GT3.2 pre-WS GRAM
Average Load and Jobs Completed (per Machine)
Average Agregate Load (per machine)
80
75
70
65
60
55
50
0
10
20
30
40
50
60
70
80
90
Machine ID
Jobs Completed


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


GT3.2 WS GRAM
Service Fairness & Utilization (per Machine)
Service Fairness
Total Service Utilization
Service Fairness
Total Service Utilization
Machine ID
500
450
400
350
300
250
200
150
100
50
0
14.0%
12.0%
10.0%
8.0%
6.0%
4.0%
2.0%
0.0%
1
3
5
7
9
11
13
15
17
19
21
23
25
Service Fairness (Polynomial Approximation) - Left Axis
Service Fairness (Moving Average) - Left Axis
Total Service Utilization (Polynomial Approximation) - Right Axis
Total Service Utilization (Moving Average) - Right Axis


GT3.2 WS GRAM
Average Load and Jobs Completed (per Machine)
Average Agregate Load (per machine)
26
25.5
25
24.5
24
23.5
23
22.5
22
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Machine ID
Jobs Completed

Analytical Model

- Model performance characteristics
  - used to estimate a service's performance based on the service load
    - Throughput
    - Service response time
- Modeling choices
  - Neural networks
  - Decision trees
  - Support vector machines
  - Regression
  - Statistical time series
  - Wavelets
  - Polynomial approximations

Contributions

- Service capacity
- Service scalability
- Resource distribution among clients
- Accurate client views of service performance
- How network latency or geographical distribution affects client/service performance
- Analytical models

Future Work

- Test more services
  - Job submission in GT3.9/GT4
  - HTTP/HTTPS
  - WS GRAM in a LAN vs. WAN
  - Perform testbed characterization
- Testing DiPerF Scalability
- Validate analytical models
- Test predictive power of analytical models

References

- Presentation Slides
  - http://people.cs.uchicago.edu/~iraicu/research/docs/diperf_presentation_v2.pdf
- Paper – Accepted for publication to Grid2004
  - Catalin Dumitrescu, Ioan Raicu, Matei Ripeanu, Ian Foster. "DiPerF: an automated DIstributed PERformance testing Framework." IEEE/ACM GRID2004, Pittsburgh, PA, November 2004
  - http://people.cs.uchicago.edu/~iraicu/research/publications/GRID2004_DiPerF_v28.pdf

References

- Catalin Dumitrescu, Ioan Raicu, Matei Ripeanu, Ian Foster. "DiPerF: an automated DIstributed PERformance testing Framework." IEEE/ACM GRID2004, Pittsburgh, PA, November 2004.
- L. Peterson, T. Anderson, D. Culler, T. Roscoe, "A Blueprint for Introducing Disruptive Technology into the Internet", The First ACM Workshop on Hot Topics in Networking (HotNets), October 2002.
- A. Bavier et al., "Operating System Support for Planetary-Scale Services", Proceedings of the First Symposium on Network Systems Design and Implementation (NSDI), March 2004.
- Grid2003 Team, "The Grid2003 Production Grid: Principles and Practice", 13th IEEE Intl. Symposium on High Performance Distributed Computing (HPDC-13) 2004.
- The Globus Alliance, www.globus.org.
- Foster I., Kesselman C., Tuecke S., "The Anatomy of the Grid", International Supercomputing Applications, 2001.
- I. Foster, C. Kesselman, J. Nick, S. Tuecke. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
- The Globus Alliance, "WS GRAM: Developer's Guide", http://www-unix.globus.org/toolkit/docs/3.2/gram/ws.
- X.Zhang, J. Freschl, J. M. Schopf, "A Performance Study of Monitoring and Information Services for Distributed Systems", Proceedings of HPDC-12, June 2003.
- The Globus Alliance, "GT3 GRAM Tests Pages", http://www-unix.globus.org/ogsa/tests/gram.
- R. Wolski, "Dynamically Forecasting Network Performance Using the Network Weather Service", Journal of Cluster Computing, Volume 1, pp. 119-132, Jan. 1998.
- R. Wolski, N. Spring, J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing," Future Generation Computing Systems, 1999.
- Charles Robert Simpson Jr., George F. Riley. "NETI@home: A Distributed Approach to Collecting End-to-End Network Performance Measurements." PAM 2004.
- C. Lee, R. Wolski, I. Foster, C. Kesselman, J. Stepanek. "A Network Performance Tool for Grid Environments," Supercomputing '99, 1999.
- V. Paxson, J. Mahdavi, A. Adams, and M. Mathis. "An architecture for large-scale internet measurement." IEEE Communications, 36(8):48–54, August 1998.
- D. Gunter, B. Tierney, C. E. Tull, V. Virmani, On-Demand Grid Application Tuning and Debugging with the NetLogger Activation Service, 4th International Workshop on Grid Computing, Grid2003, November 2003.
- Ch. Steigner and J. Wilke, "Isolating Performance Bottlenecks in Network Applications", in Proceedings of the International IPSI-2003 Conference, Sveti Stefan, Montenegro, October 4-11, 2003.
- G. Tsouloupas, M. Dikaiakos. "GridBench: A Tool for Benchmarking Grids," 4th International Workshop on Grid Computing, Grid2003, Phoenix, Arizona, November 2003.
- P. Barford ME Crovella. Measuring Web performance in the wide area. Performance Evaluation Review, Special Issue on Network Traffic Measurement and Workload Characterization, August 1999.
- G. Banga and P. Druschel. Measuring the capacity of a Web server under realistic loads. World Wide Web Journal (Special Issue on World Wide Web Characterization and Performance Evaluation), 1999.
- N. Minar, "A Survey of the NTP protocol", MIT Media Lab, December 1999, http://xenia.media.mit.edu/~nelson/research/ntp-survey99
- K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke, "A Resource Management Architecture for Metacomputing Systems", IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, pg. 62-82, 1998