GridFTP Scalability and Performance Results

Ioan Raicu – iraicu@cs.uchicago.edu
Catalin Dumitrescu - catalind@cs.uchicago.edu

1.0 Introduction

We are evaluating the GridFTP server located at ISI.

Table 1: Host specifications

Machine Name	ned-6.isi.edu
Machine Type	x86
OS	Linux
OS Release	2.6.8.1-web100
Number of Processors	2
CPU Speed	1126 MHz
Memory Total	1.5 GB
Swap Total	2 GB
Network Link	1 Gb/s Ethernet
Network MTU	1500 B
GridFTP Server Version	0.13 (gcc32dbg, 1103191677-1) Development Release **

The metrics collected (client view) by DiPerF are:

- **Service response time** or time to serve a request, that is, 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; each request involved the transfer of a 10MB file from the client hard disk to the server's memory (/dev/null)
- **Service throughput**: aggregate MB/s of data transferred from the client view
- **load**: number of concurrent service requests

The metrics collected (server view) by Ganglia are, with the *italicized* words being the metric names collected from Ganglia:

- **Number of Processes**: *proc_total* – (number of processes running at start of the experiment)
- **CPU Utilization**: *cpu_user* + *cpu_system*
- **Memory Used**: *mem_total* + *swap_total* – *mem_free* – *mem_cache* – *mem_buffers* – *mem_share* – *swap_free*
- **Network throughput**: *bytes_in* (converted to MB/s); we were only interested in the inbound network traffic since we were performing uploads from clients to the server

We ran our experiments on about 100 client machines distributed over the PlanetLab testbed throughout the world. Some of the later experiments also included about 30 machines from the CS cluster at University of Chicago (UofC). We ran the GridFTP server at ISI on a machine with the specs outlined in Table 1; the DiPerF framework ran on an AMD K7 2.16GHz with 1GB RAM and 100Mb/s network connection located at UofC. The machines in PlanetLab are generally connected by 10Mbps Ethernet while the machines at UofC are generally connected by 100Mbps Ethernet.

For each set of tests, the caption below the figure will address the particular configuration of the controller which yielded the respective results. We also had the testers synchronize their time every five minutes against our time server running at UofC.

In the figures below, each series of points representing a particular metric and is also approximated using a moving average over a 60 point interval, where each graphs consists of anywhere from 1,000s to 100,000s of data points. There is an appendix with the same graphs found in Figures 1 through 6 that are higher resolution.

2.0 GridFTP Results

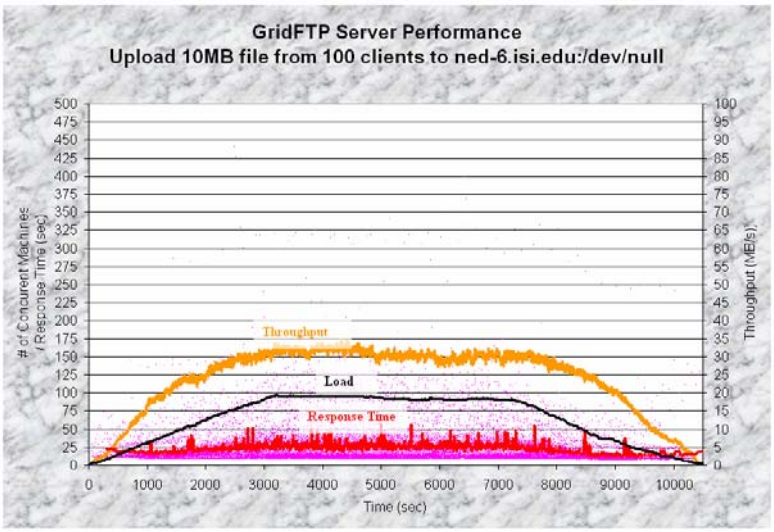


Figure 1: GridFTP server performance with 100 clients running on 100 physical nodes in PlanetLab; tunable parameters: utilized 100 concurrent clients, starts a new client every 30 seconds, each client runs for 7200 seconds; 243.4 GB of data transferred over 24,925 file transfers; left axis – load, response time; right axis - throughput

The results from Figure 1 are the initial tests that we did which we already sent you the results of earlier. For these experiments, Ganglia had died very early on in the experiment, and therefore we did not have Ganglia's server view for this test.

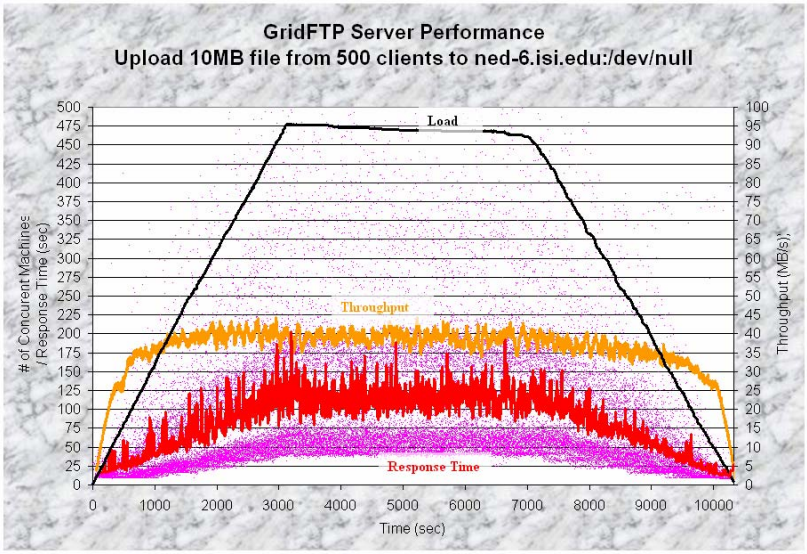


Figure 2: GridFTP server performance with 500 clients running on 100 physical nodes in PlanetLab; tunable parameters: utilized 500 concurrent clients, starts a new client every 6 seconds, each client runs for 7200 seconds; 363.3 GB of data transferred over 37,201 file transfers; left axis – load, response time; right axis - throughput

The interesting thing about the results from Figure 2 are that running multiple clients on the same host improved the aggregate throughput from around 30 MB/s to almost 40 MB/s. For these experiments, Ganglia was still down, and therefore we did not have Ganglia's server view for this test either.

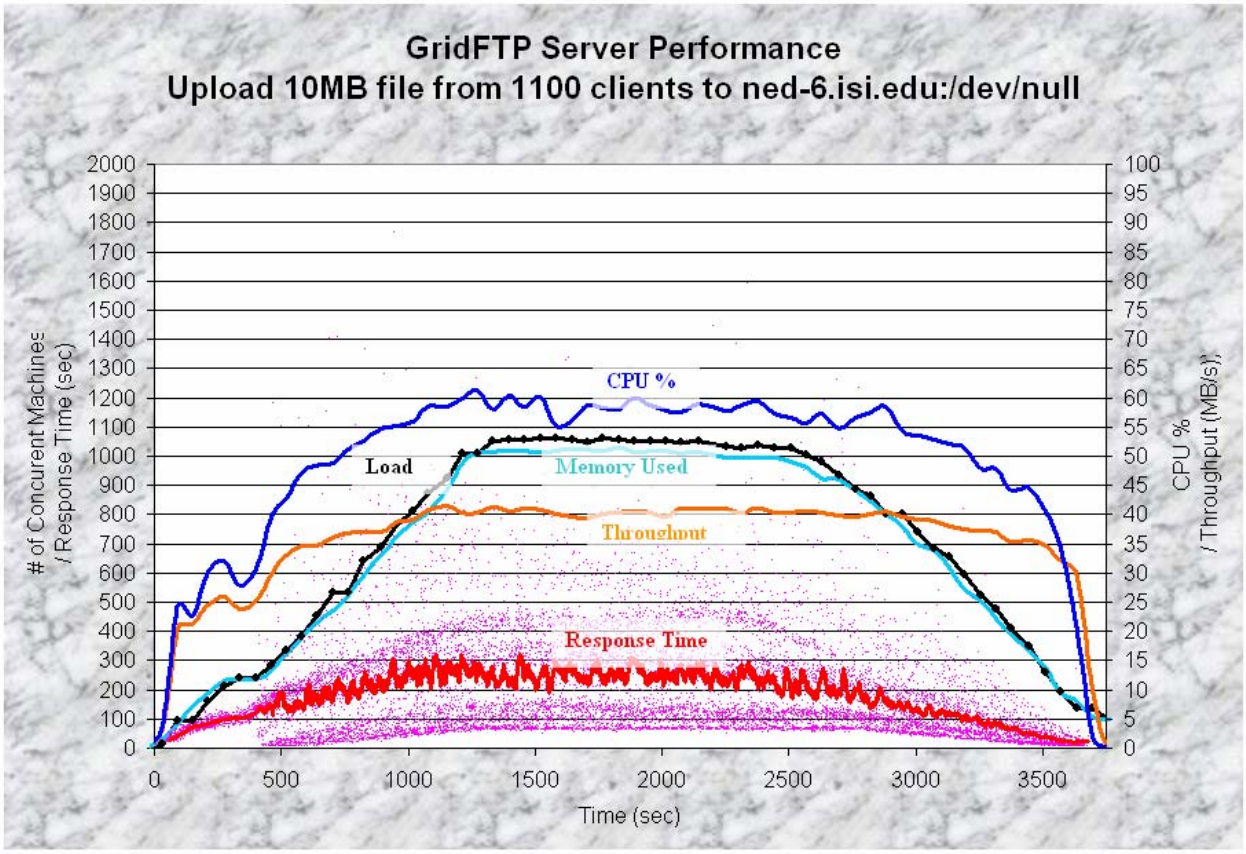


Figure 3: GridFTP server performance with 1100 clients running on 100 physical nodes in PlanetLab and 30 physical nodes in the CS cluster at UofC; tunable parameters: utilized 1100 concurrent clients, starts a new client every 1 second, each client runs for 2400 seconds; 131.1 GB of data transferred over 13,425 file transfers; left axis – load, response time, memory; right axis – throughput, CPU %

Here we added a few more physical nodes located in the CS cluster at UofC and essentially doubled the number of clients. We notice that the peak aggregate throughput remains the same, around 40 MB/s, despite the fact that independent tests of the PlanetLab testbed yielded around 40 MB/s and similar tests of just the CS cluster testbed yielded around 25 MB/s. We conclude that since the testbeds could achieve an aggregate throughput of 65 MB/s, the limitation that yielded around 40 MB/s was either the server or the network connection to ISI. The server seemed to have ample CPU resources available, which makes it likely that the connection into ISI was the bottleneck. This can only be verified by performing some internal LAN tests to make sure that the server could indeed serve the same scale of clients with a higher aggregate throughput than we observed. When comparing some of the metrics from Ganglia and those of DiPerF, it is interesting to note that the memory used follows pretty tight the number of concurrent clients. In fact, we computed that the server requires about 0.94 MB of memory for each new client it has to maintain state for; we found this to be true for all the tests we performed within +/- 1%. Another interesting observation is that the CPU utilization closely mimics the achieved throughput, and not necessarily the number of concurrent clients.

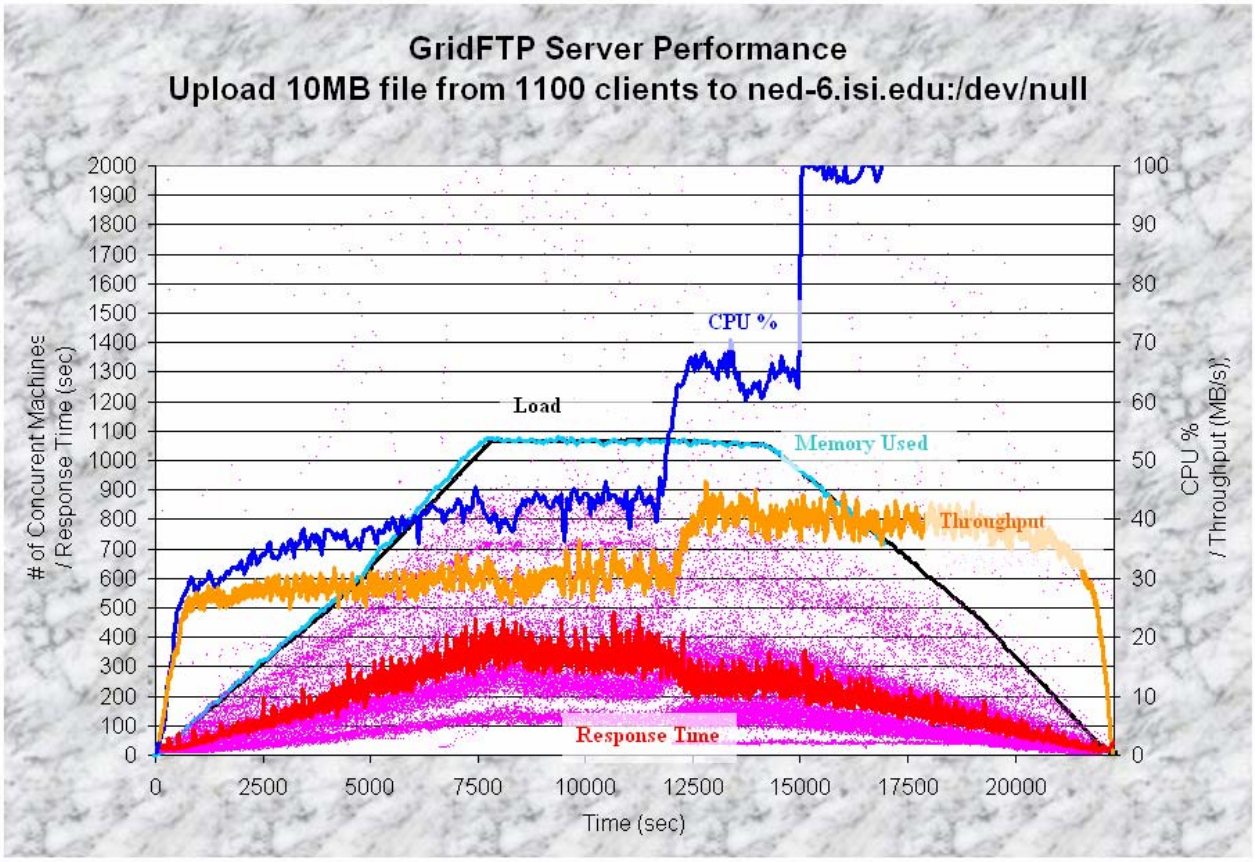


Figure 4: GridFTP server performance with 1100 clients running on 100 physical nodes in PlanetLab; tunable parameters: utilized 1100 concurrent clients, starts a new client every 6 seconds, each client runs for 14400 seconds; 916.58 GB of data transferred over 93,858 file transfers; left axis – load, response time, memory; right axis – throughput, CPU %

Figure 4 is really interesting because it showed the behavior of the server under a changing network condition. Here is a little background on PlanetLab and its policies (which we found out after we ran most of these experiments). Once a slice (which could run multiple clients) sends 16GB in a day on a particular node, the slice is then limited to 1.5Mbps for the rest of the day on that node. Think of this 16GB as a very big token bucket: you are permitted to burst (up to the physical line rate) until you send 16GB, then the slice gets rate limited. Therefore, the results from Figure 4 represent exactly this behavior. First of all, we had been running many tests before this one, so we probably managed to get the better connected hosts to trigger their 16GB cap, and limit themselves to only 1.5 Mb/s. In the middle of the tests, apparently there were some nodes that reset their counters, and all of a sudden had lots more bandwidth to use, and hence we see the increase from 30 to 40 MB/s and a CPU utilization from 45% to 65%. After another hour of the experiment, the CPU got pegged at 100% utilization for no apparent reason. In order to put things into perspective in terms of the amounts of data we transmitted in this experiment, we transferred about 916 GB of data over almost 94,000 files transfers.

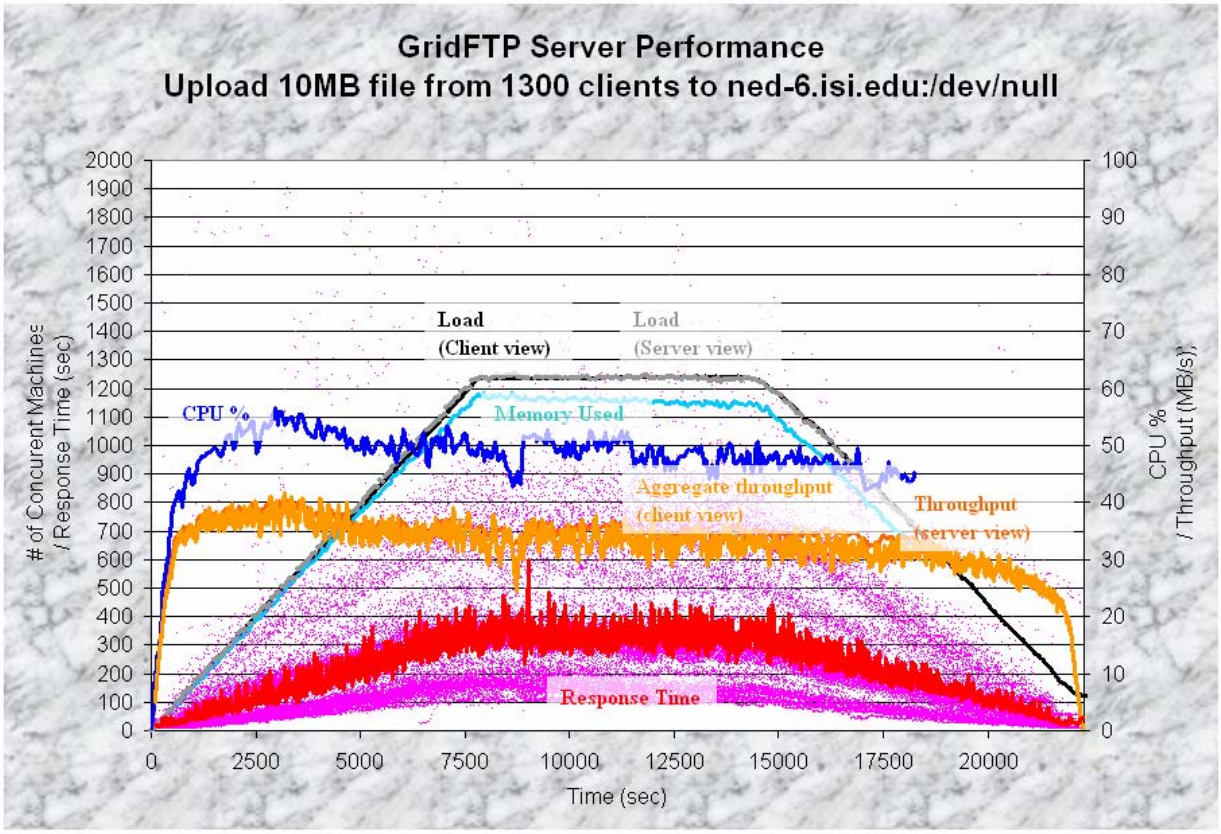


Figure 5: GridFTP server performance with 1300 clients running on 100 physical nodes in PlanetLab; tunable parameters: utilized 1300 concurrent clients, starts a new client every 6 seconds, each client runs for 14400 seconds; 767 GB of data transferred over 78,541 file transfers; left axis – load, response time, memory; right axis – throughput, CPU %

Figure 5 is really interesting because it tries to depict both the client and server view simultaneously in order to validate the results from DiPerF. The two metrics that line up nearly perfect are: 1) load (number of concurrent clients [black] vs. number of processes [gray]) and 2) throughput (aggregate client side [light orange] vs. server side [dark orange]). If it is hard to discern between the client and server view, that is because they light up almost perfectly most of the time, and it is only in certain small places where they diverge enough to see that the data contains two separate metrics. Notice how the achieved throughput is steadily decreasing, as we are probably hitting the 16GB per day limits on certain nodes, and hence we get a lower aggregate throughput by the end of the experiment.

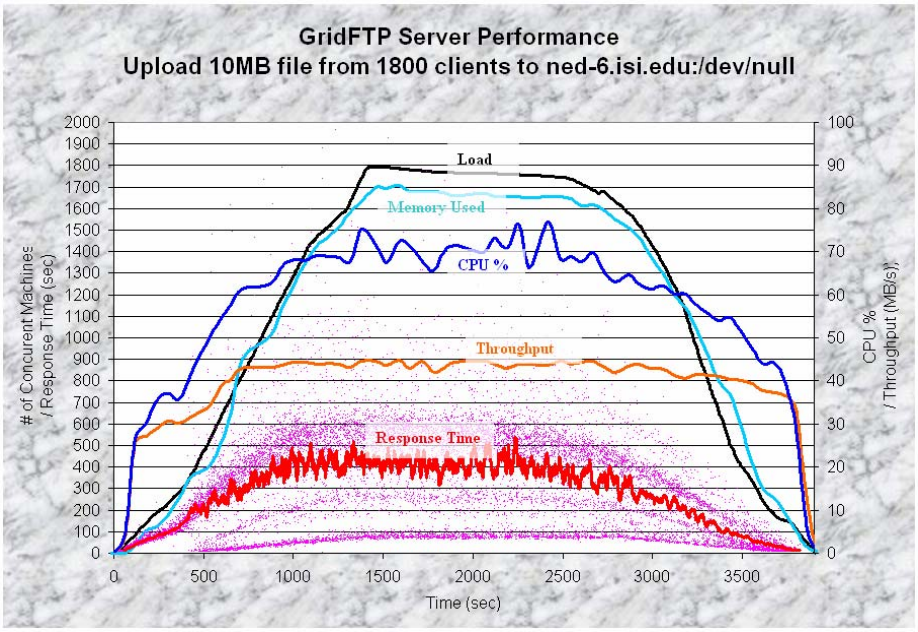
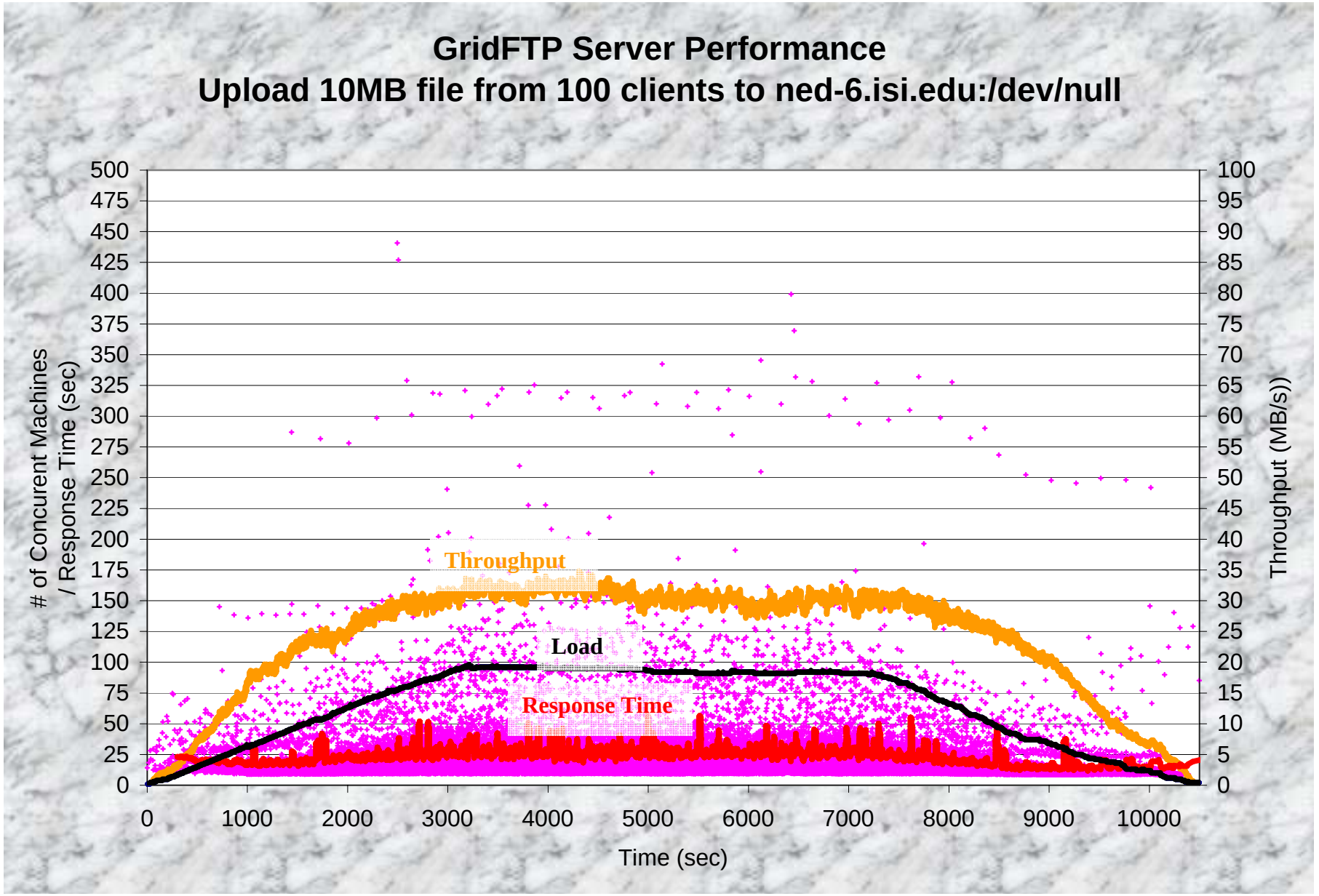


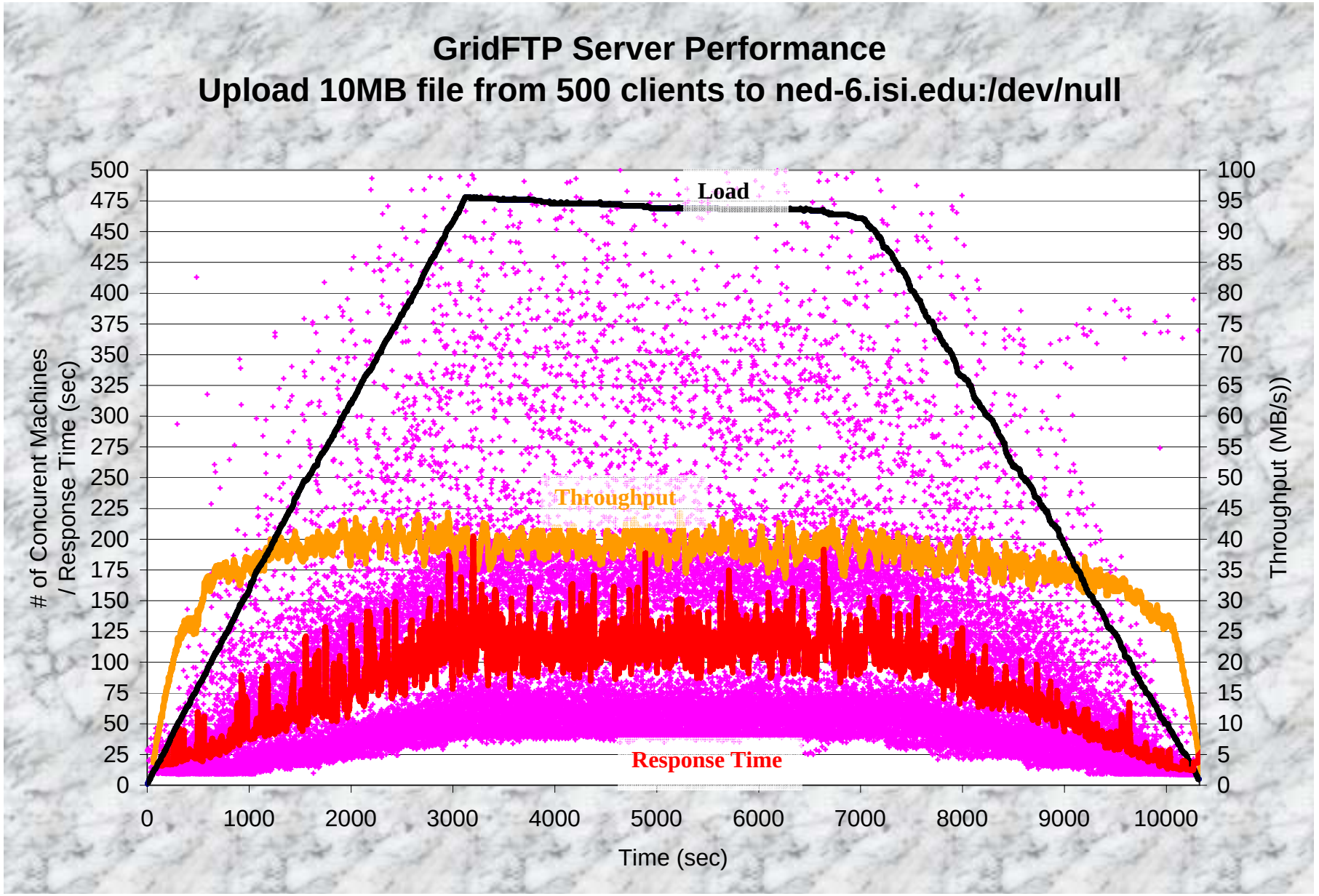
Figure 6: GridFTP server performance with 1800 clients running on 100 physical nodes in PlanetLab and 30 physical nodes in the CS cluster at UofC; tunable parameters: utilized 1800 concurrent clients, starts a new client every 1 second, each client runs for 2400 seconds; 150.7 GB of data transferred over 15,428 file transfers; left axis – load, response time, memory; right axis – throughput, CPU %

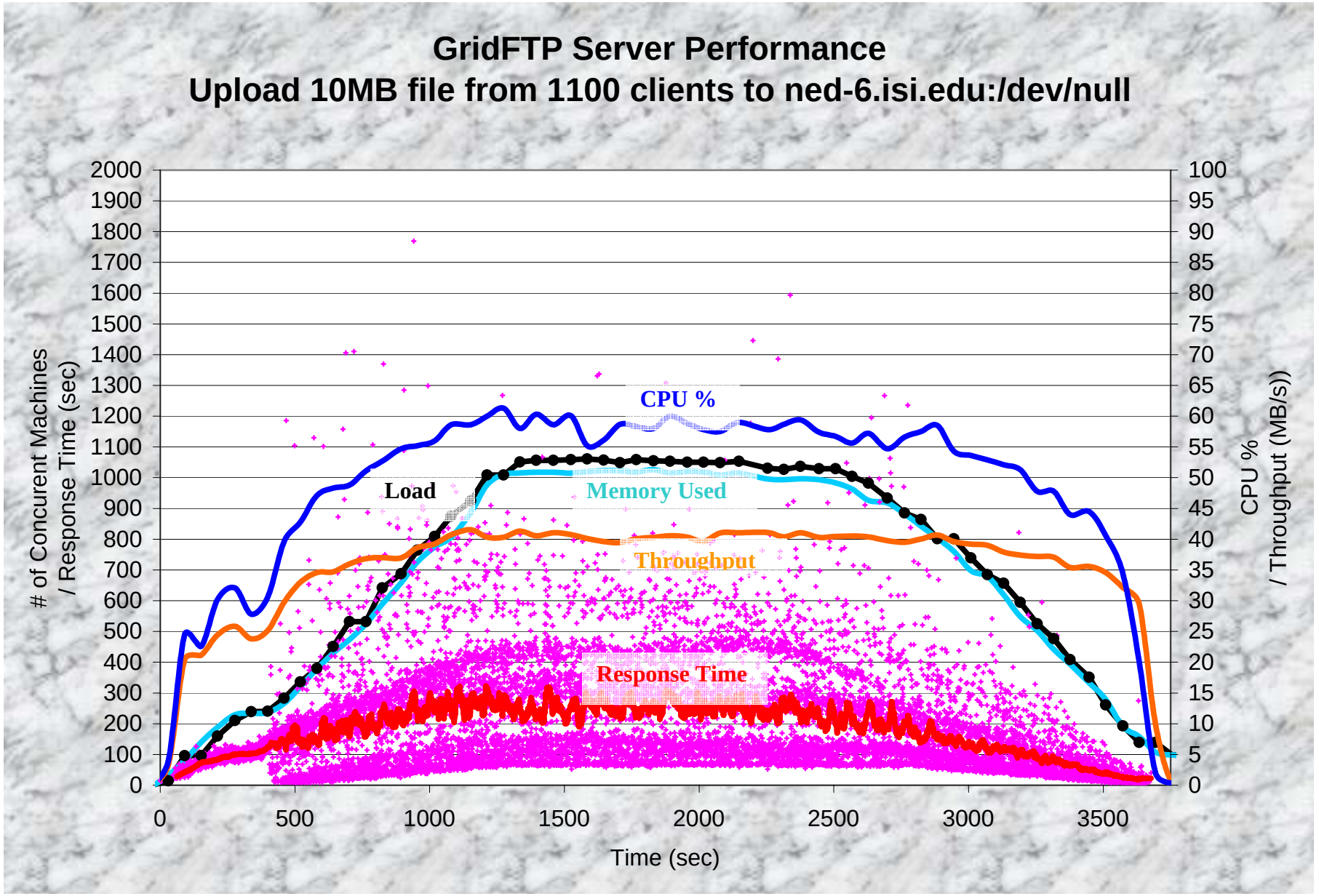
Figure 6 is probably the most impressive due to the scale of the experiment. We coordinated 1800 clients over 130 physical nodes distributed around the world. It is very interesting to see that the throughput reached around 45 MB/s (~360 Mb/s) and stayed consistent with good throughout despite the fact that the server ran out of physical memory and started swapping memory out. Note that the CPU utilization is getting high, but with a 75% utilization and another 1.5GB of swap left, the server seems as if it could handle additional clients. From a memory point of view, we believe that it would take about 5000 concurrent clients to leave the server without enough memory to handle new incoming clients. From a CPU point of view, we are not sure how many more clients it could support since as long as the throughput does not increase, it is likely that the CPU will not get utilized significantly more. Another issue at this scale of tests is the fact that most OSes have hard limits set in regards to file descriptors and number of processes that are allowed to run at any given point in time. With 1800 clients, we seemed to have perhaps saturate the network link into the server, but I do not believe we were able to saturate the server's raw (CPU, memory, etc…) resources. Unfortunately, PlanetLab experienced some major problems today from a bug in the CPU share initialization which appeared to have introduced a divide by zero error; this bug limited our testbed to only a small fraction of the machines we normally have access to, and therefore we were not able to run more tests with more clients to see what happens all the way up to where the server runs out of memory, swap, and CPU resources. If we can and PlanetLab get back into a normal state, we will try to run one more test with 5000 clients before the Monday deadline of HPDC.

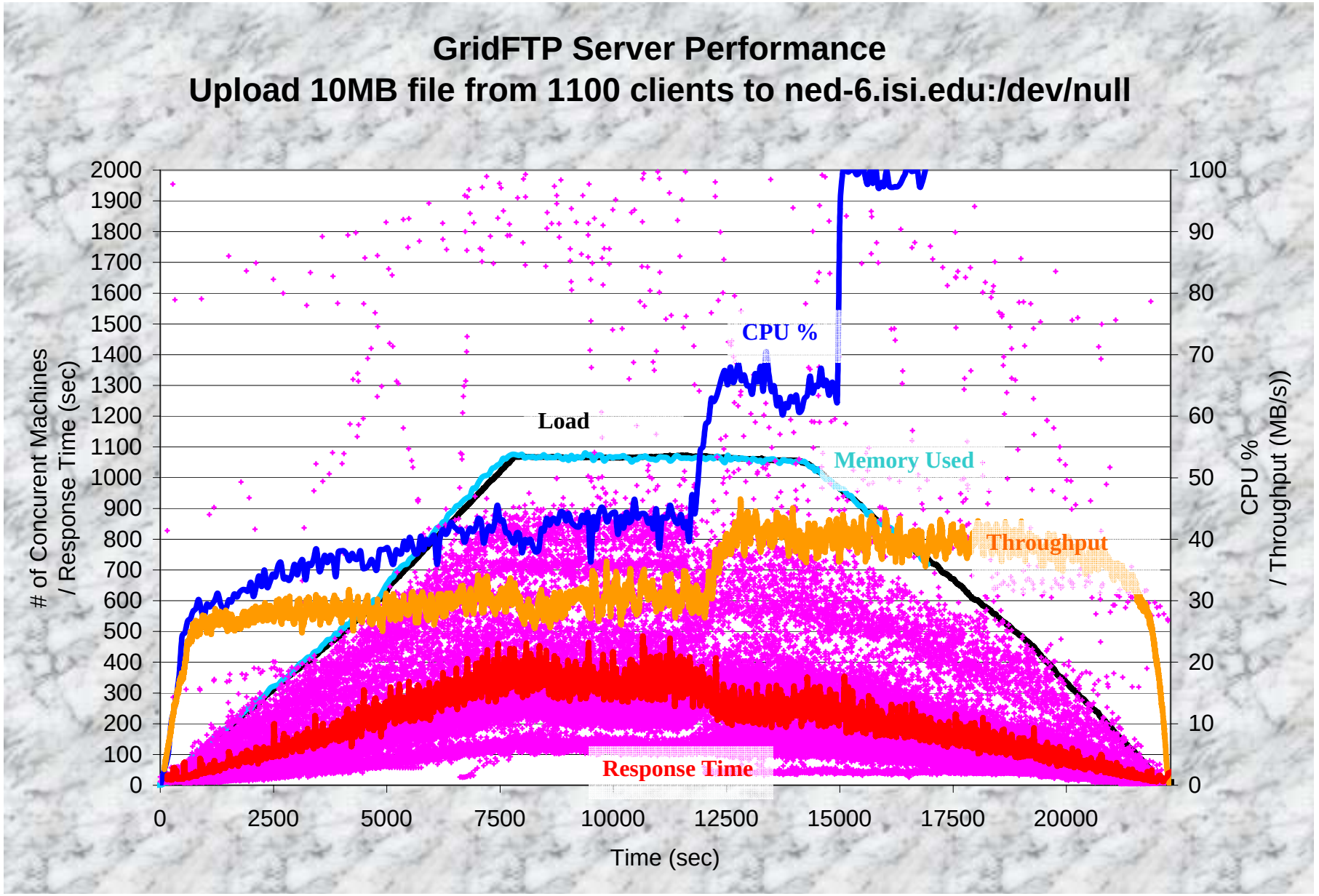
3.0 Appendix – High Resolution Graphs

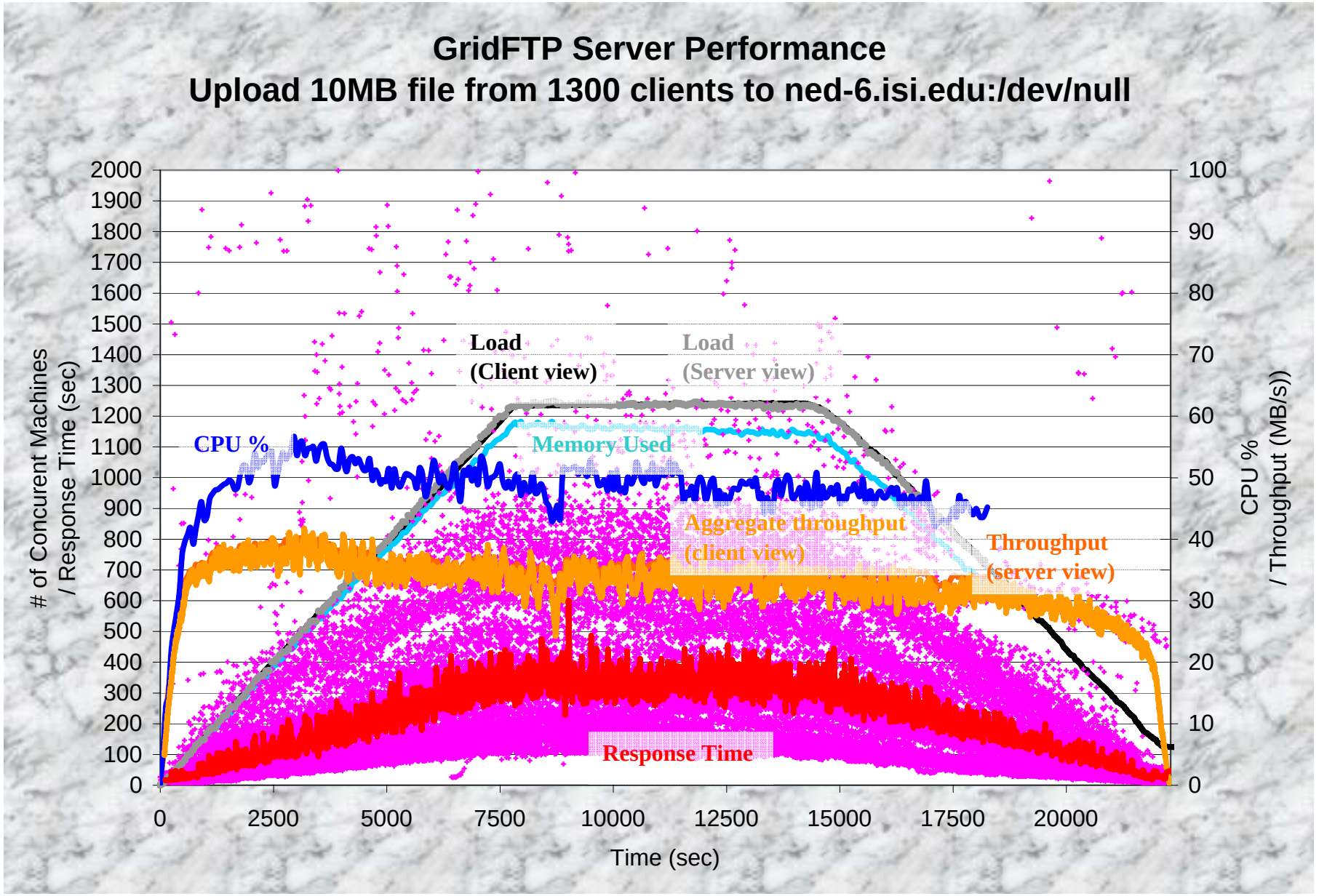
The next few pages contain high resolution graphs from Figures 1 through 6.

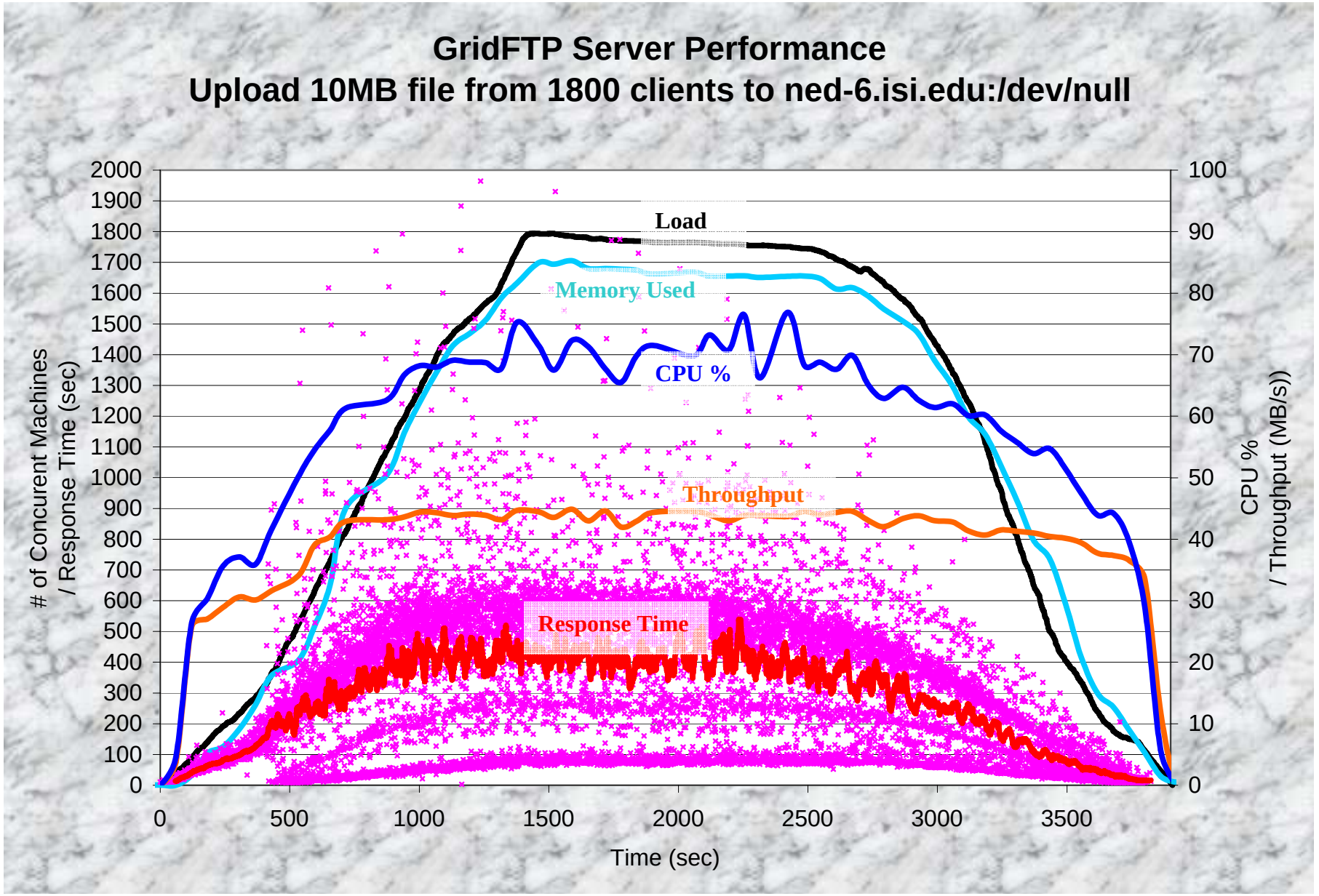

GridFTP Server Performance
Upload 10MB file from 100 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
Throughput (MB/s))
Time (sec)
Throughput
Load
Response Time
0
25
50
75
100
125
150
175
200
225
250
275
300
325
350
375
400
425
450
475
500
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


GridFTP Server Performance
Upload 10MB file from 500 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
Throughput (MB/s))
Time (sec)
Load
Throughput
Response Time
0
25
50
75
100
125
150
175
200
225
250
275
300
325
350
375
400
425
450
475
500
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


GridFTP Server Performance
Upload 10MB file from 1100 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
CPU % / Throughput (MB/s))
Time (sec)
CPU %
Load
Memory Used
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
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
0
500
1000
1500
2000
2500
3000
3500


GridFTP Server Performance
Upload 10MB file from 1100 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
CPU % / Throughput (MB/s))
Time (sec)
CPU %
Load
Memory Used
Throughput
Response Time
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
10
20
30
40
50
60
70
80
90
2500
5000
7500
10000
12500
15000
17500
20000


GridFTP Server Performance
Upload 10MB file from 1300 clients to ned-6.isi.edu:/dev/null
# of Concurent Machines / Response Time (sec)
CPU % / Throughput (MB/s))
Time (sec)
Load (Client view)
Load (Server view)
CPU %
Memory Used
Aggregate throughput (client view)
Throughput (server view)
Response Time
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
10
20
30
40
50
60
70
80
90
2500
5000
7500
10000
12500
15000
17500
20000


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