Characterization of PlanetLab Nodes

Table of Contents

Methodology

We tested the performance of various nodes by measuring goodput using the following configuration:

- Nodes were ran in single node configuration
- Test client was the standalone closed loop test client (2019), running on an AWS c5.large instance located in Oregon
- Each node had the same test conducted, namely, running the test client with 256 clients for 60 seconds

Our overall methodology was:

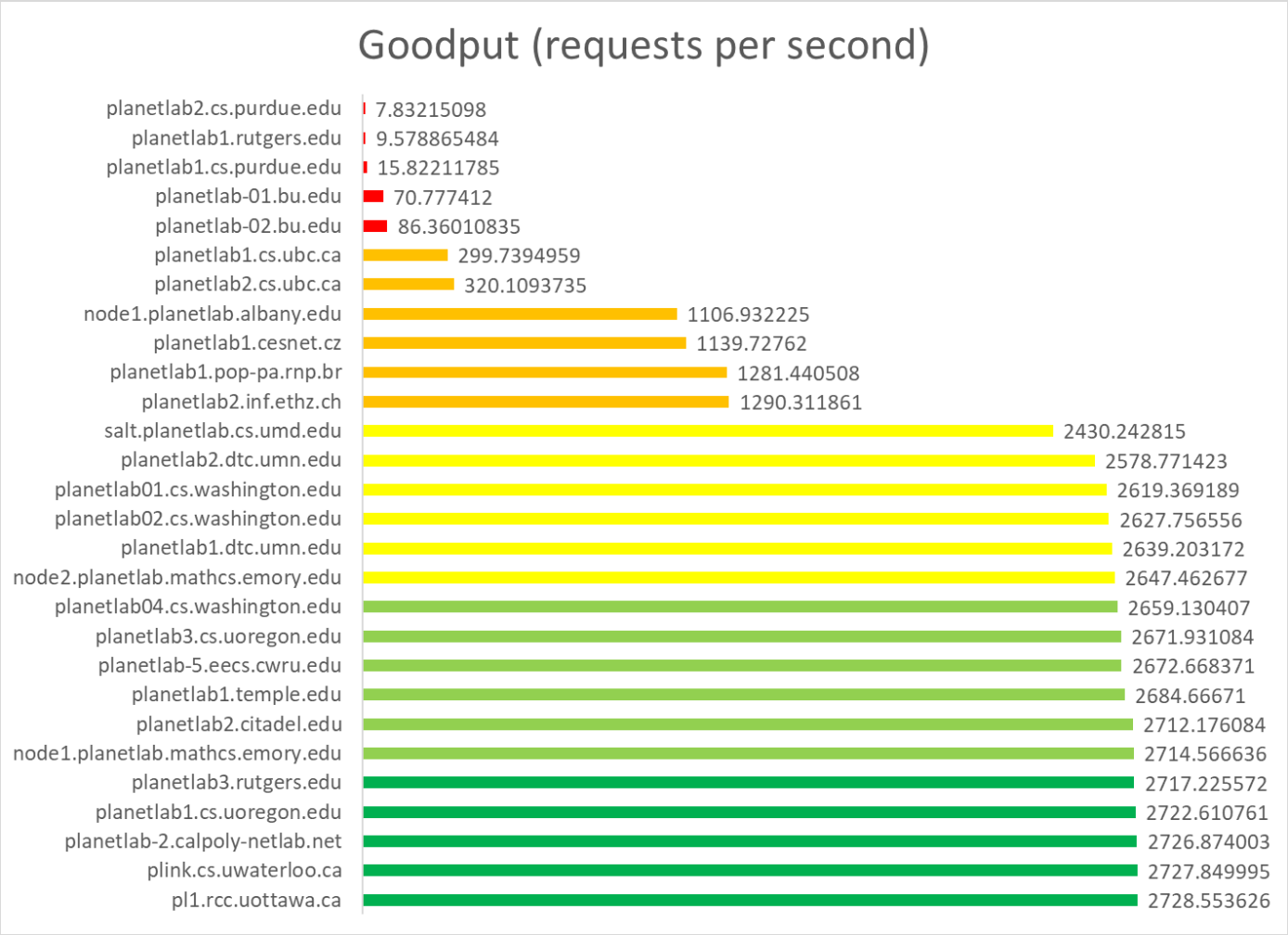
1. Adding Nodes
    a. Add nodes from PlanetLab to our Slice using monitoring service
    b. Wait for approximately 60 minutes for SSH keys to propagate and eliminate nodes that still don't allow SSH. (this removed about 120/175 nodes)
    c. Set up Java for all remaining nodes and eliminate any where it could not be installed or it took too long. (this removed about 20/55 nodes)
    d. Set up KVStore by uploading A8.jar. (we had about 35 reliable nodes here)
    e. Start KVStore by selecting the nodes and pressing the "Start App" button.
    f. Run the characterization tests.

2. Gathering Results:
    a. Python script to ping nodes (latency), run tests while collecting data from the running server (CPU/MEM % from ps aux - calculated min, max, average) and the test itself (goodput)
    b. Monitoring service to grab relevant information on each node (Disk Free (MB), Disk Used (MB), Avg Load Over 1m, Avg Load Over 5m, Memory Free (MB), Memory Used (MB)) to detect anomalies

Results

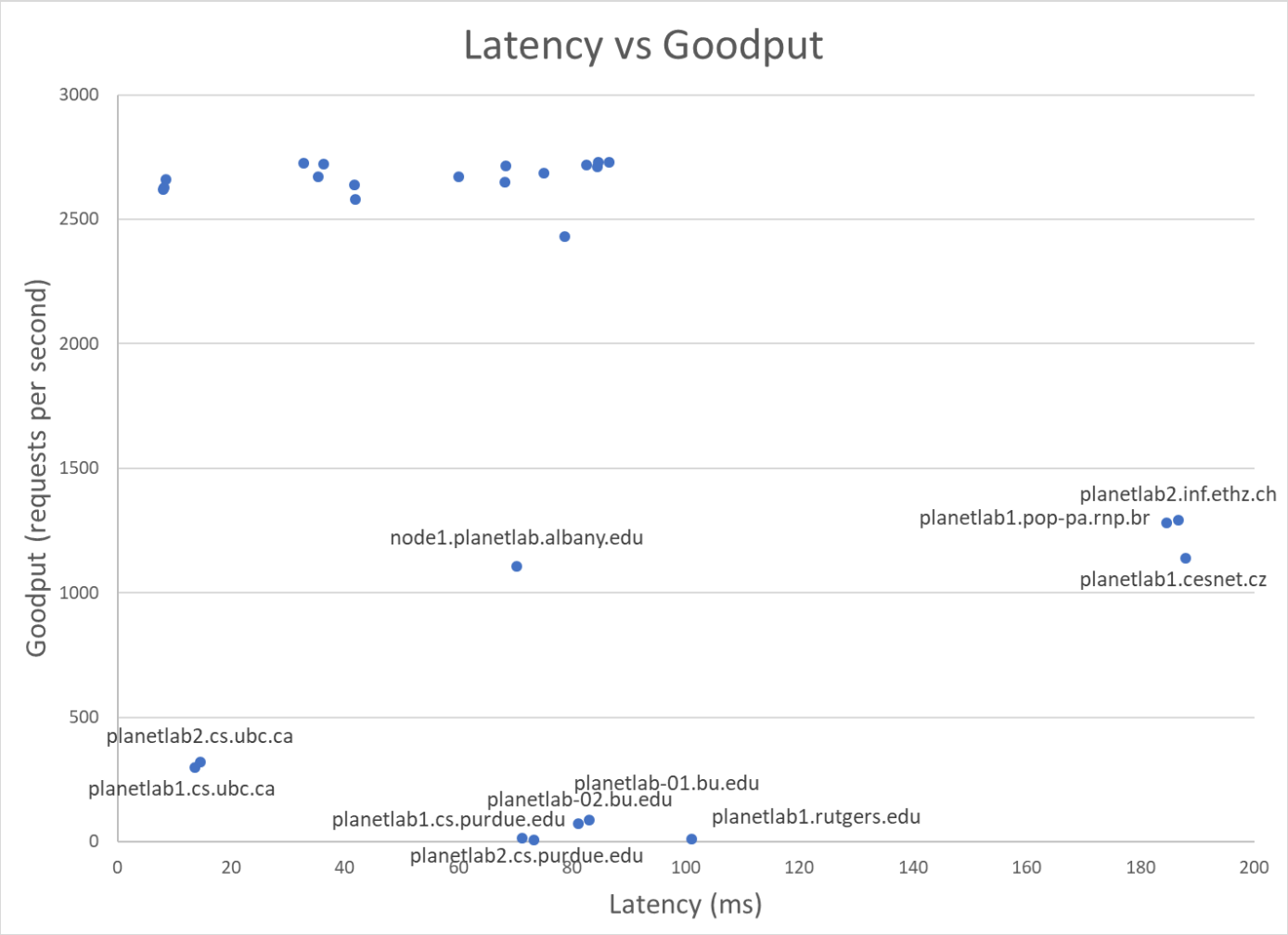
The performance of our server varied significantly between nodes, as evident from the following bar chart:

Figure #1: Average Goodput for selected nodes



Note:
Despite the large variation in performance, of all the metrics we measured during testing, only a few differed significantly between nodes.

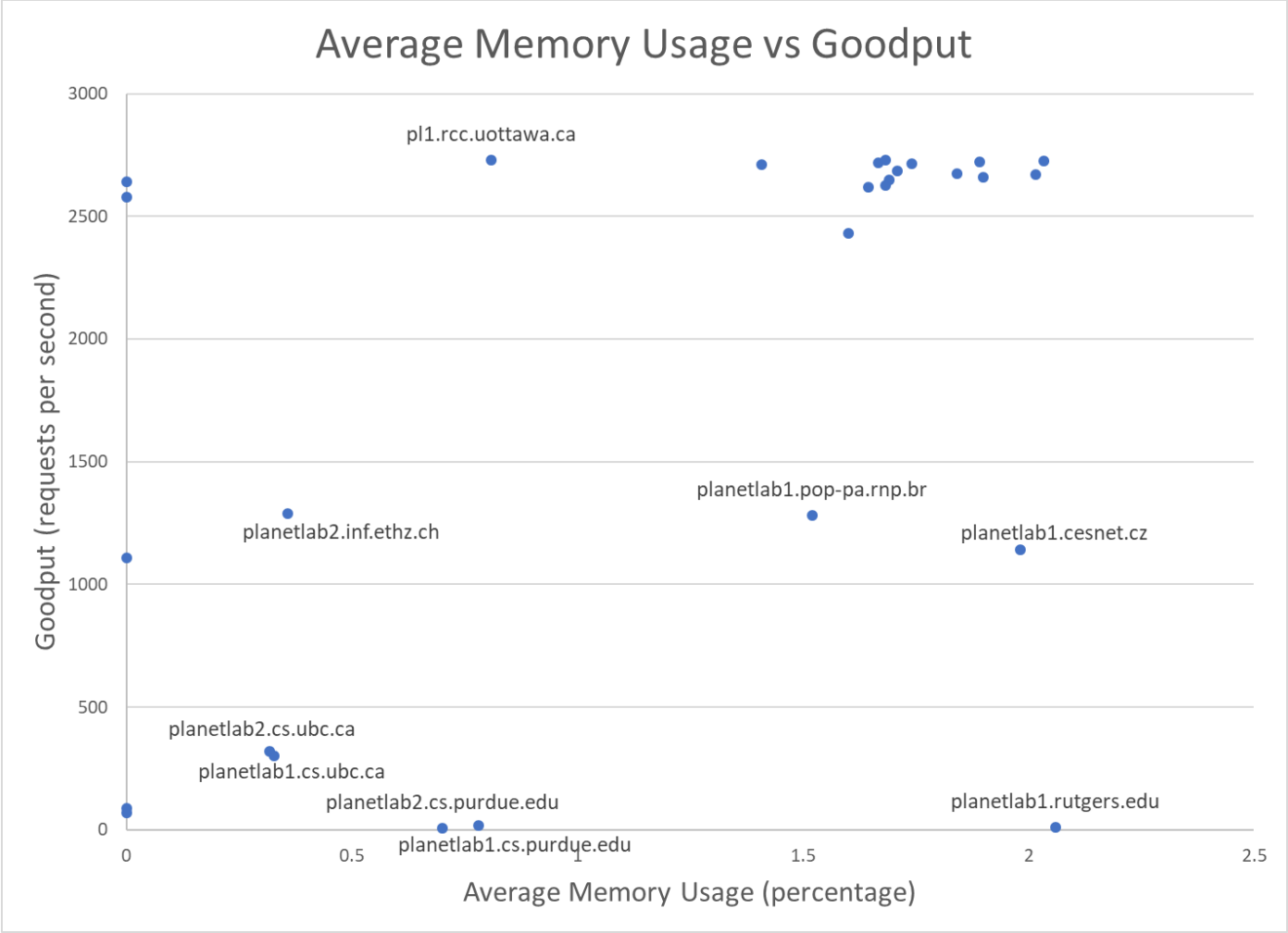
Figure #2: Latency vs Goodput for selected nodes



Observation:

There is no strong correlation between latency and goodput. However, we can see that in general, the well performing nodes generally have relatively low latency (< 100ms), and three nodes with the highest latency (~180ms) all have relatively weak performance. However, this observation is not always true as there are quite a few nodes that have low latency but very weak performance (worse than the three high latency nodes).

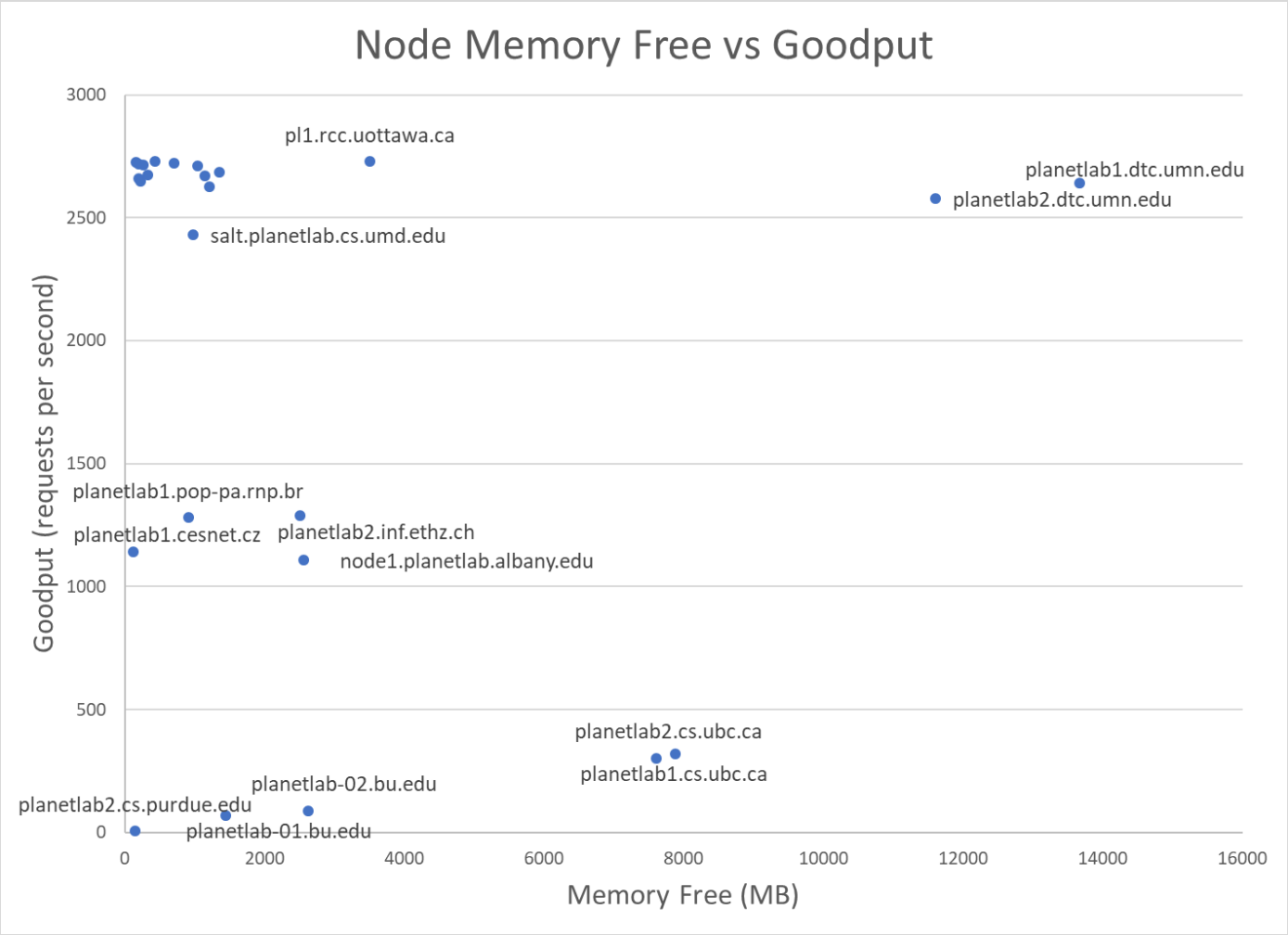
Figure #3: Average Memory Usage vs Goodput for selected nodes



Observation:

There is some correlation between average memory usage and goodput like what we hypothesized. This makes sense because the nodes which have high goodput will use more memory, but this does not really help us determine if nodes are good or not.

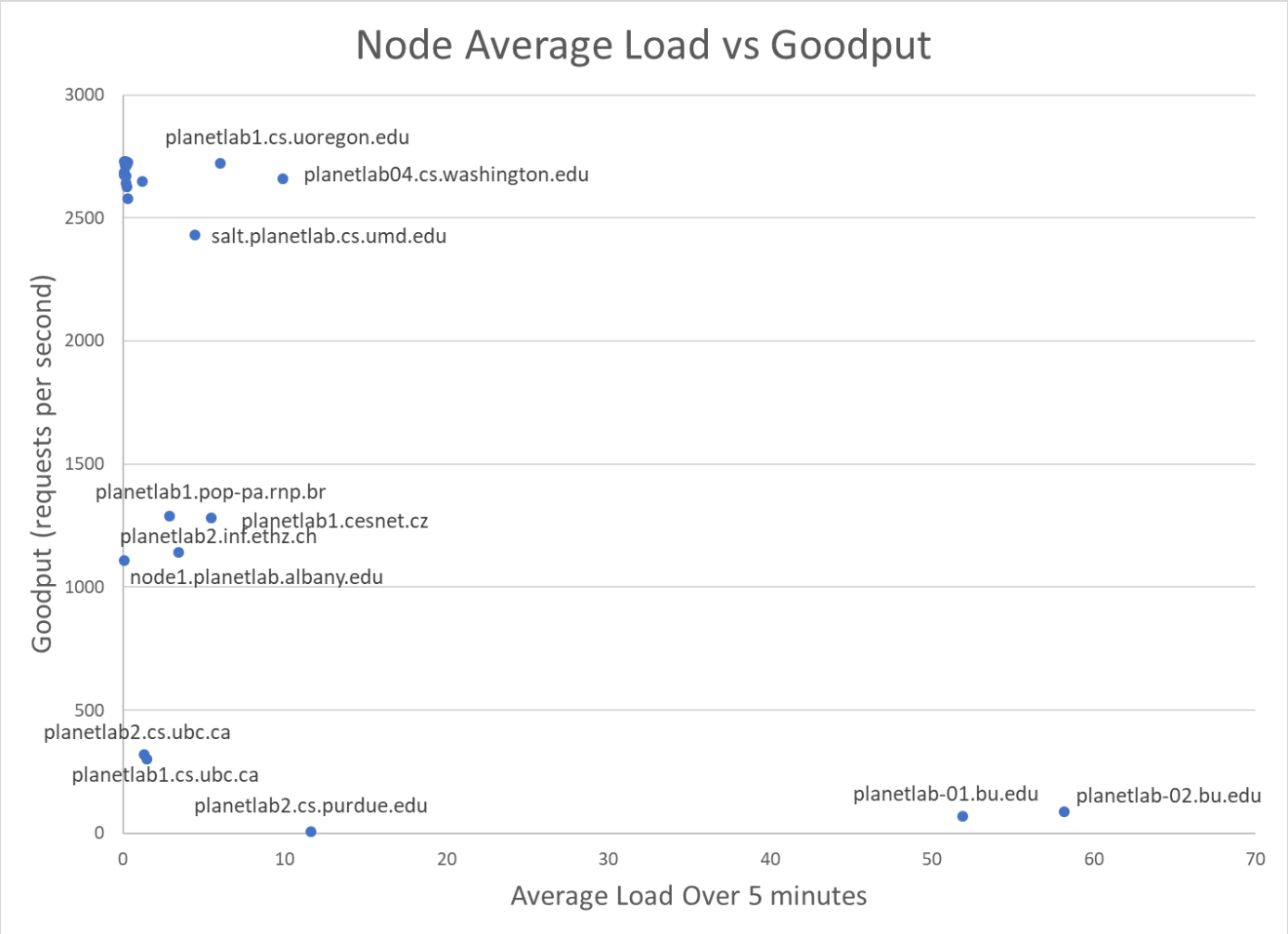
Figure #4: Free Memory vs Goodput for selected nodes



Observation:

There is no strong correlation between memory free on the node and goodput, contrary to what we anticipated. There may be a positive relationship as the two nodes with the most free memory had good performance, but the nodes that have 3rd and 4th highest memory free do not have good performance. We anticipated that low memory free would lead to low performance, but this does not seem to be the case.

Figure #5: Average Load vs Goodput for selected nodes



Observation:

It is hard to see a trend in this particular metric, since the vast majority nodes have low average load (< 10%). However, we do have two nodes with > 50% average load, and they both performed very poorly. So perhaps high load does impact the performance.

Conclusion:

Regarding the observed performance for our selected set of nodes, it does not seem to be strongly dependent on any one of the decided metrics. Although this is only true to a certain extent as the individual performance for some nodes may be bounded by the memory capacity, average load, and latency. Thus, our test suite is unable to allow us to immediately determine which nodes have the best performance when subjected to loads coming from a test client running in Oregon. However, it does allow us to make educated guesses on which nodes will have the worst performances and allow us to eliminate them from our set. For example, picking nodes with the lowest latency should give us a good chance of having good performance. Additionally, it appears some nodes perform consistently poor without any apparent cause. This means that throughout the assignments, we can keep a blacklist of poor performing nodes and avoid those. From our tests, the nodes consistently performing the worst are:

- Planetlab1.cs.purdue.edu
- Planetlab2.cs.purdue.edu
- Planetlab1.rutgers.edu
- Planetlab-01.bu.edu
- Planetlab-02.bu.edu
- Planetlab1.cs.ubc.ca
- Planetlab2.cs.ubc.ca

As a result of our testing, we believe the best way to determine which nodes to use would be to run quick tests on all nodes and then remove the nodes with the worst performance. To pick based on metrics is not as reliable but selecting nodes with the lowest average load and lowest latency seems to work well for our assignments so far.