

CS 5463 Cloud and Big Data
Assignment 1: Benchmarking MPI Cluster in the Cloud
SPRING 2017

Team Memebers

1. **Sharif Mohammad**
2. **Saeef Ahmad**
3. **Tuan Dinh**
4. **Akash Sudhakaran Lissy**

Contributions from team members

1. **Sharif Mohammad** – Coded the python program named ‘assign1.py’ to automate the experiments, and graph plotting for Q. 1 and Q. 2.
2. **Saeef Ahmad** – Troubleshooting and Worked on Q2 to evaluate the performance of two HPC benchmarks: HPL(the Linpack benchmark) and PTRANS (Parallel matrix transpose).
3. **Tuan Dinh** – Connectivity checking and Setting up the cluster and coding assign1.py to automate the experiments, and graph plotting for Q. 1 and Q. 2 and Analysis of performance by -byslot and –bynode scheduling policies. He wrote a bash file as execute.sh to run all the command line executions.
4. **Akash Sudhakaran Lissy** – Worked on Q1 to analyze the performance by two different scheduling policies -byslot and –bynode for each cluster size and Plotted graphs to compare the performance. He also worked on report writing.

Q1) This exercise is to measure the performance of a ray tracing engine, executing parallel tasks in a MPI cluster. Report the “Ray Tracing Time” for various cluster sizes (1, 2, 3, 4 VMs). For each cluster size, use two default slots per node, and run processes to fill up all the default slots.

Graphs

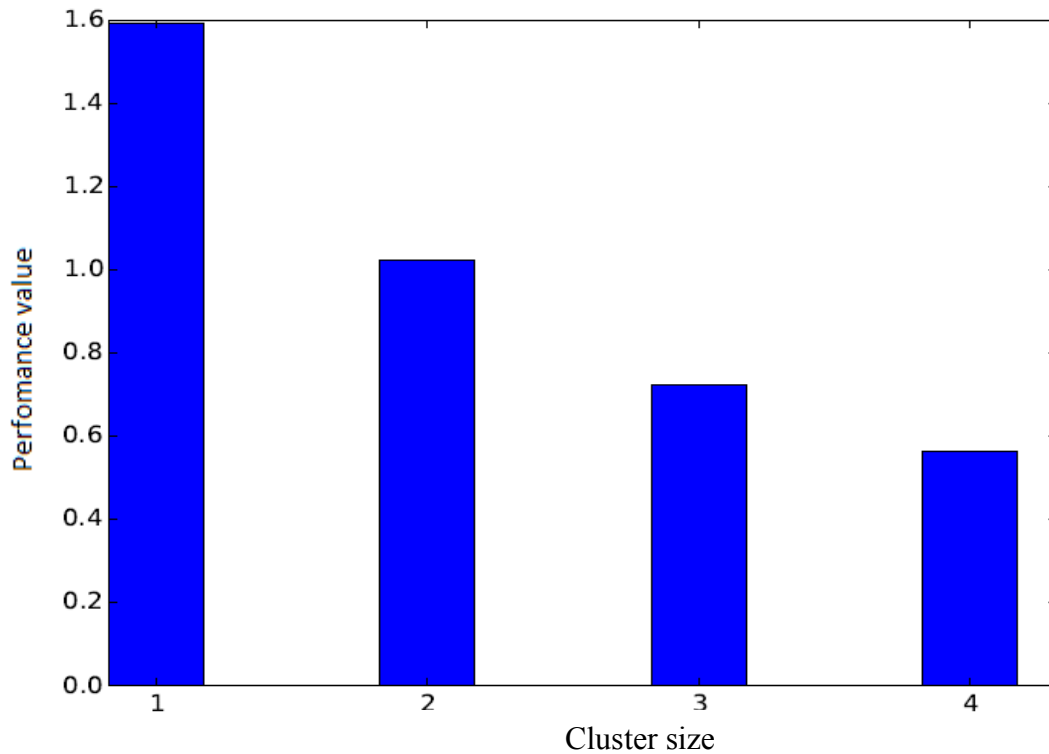


Figure 1. Tachyon Benchmark By Slot

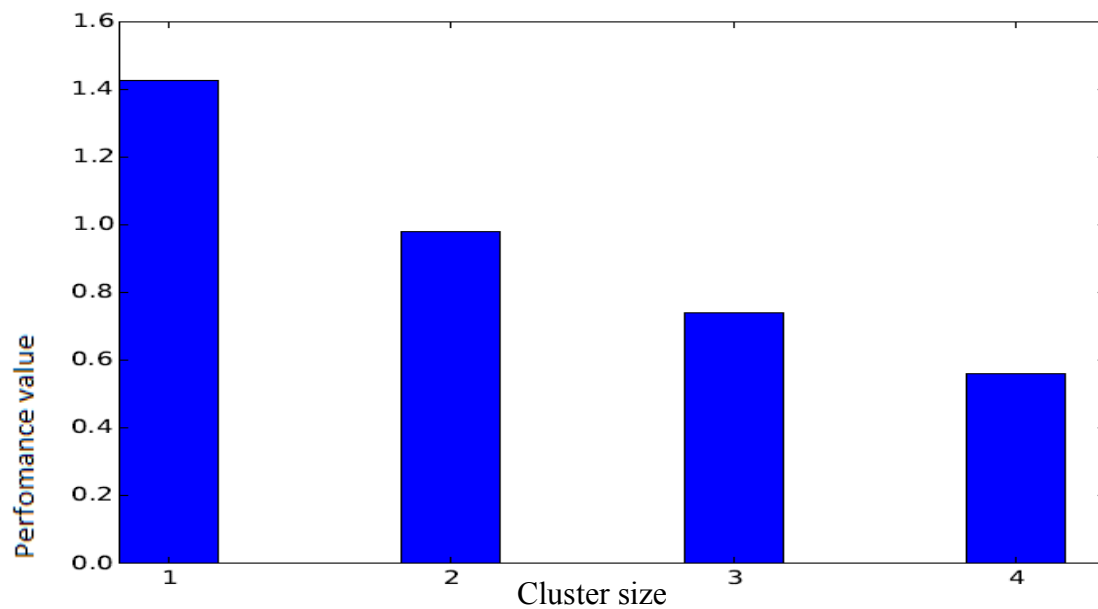


Figure 2. Tachyon Benchmark by Node

Output file

Tachyon byslot

Cluster 1: 1.6107

Cluster 2: 0.8378

Cluster 3: 0.7134

Cluster 4: 0.5643

Tachyon bynode

Cluster 1: 1.3700

Cluster 2: 0.9659

Cluster 3: 0.7325

Cluster 4: 0.5565

Put them together, we have

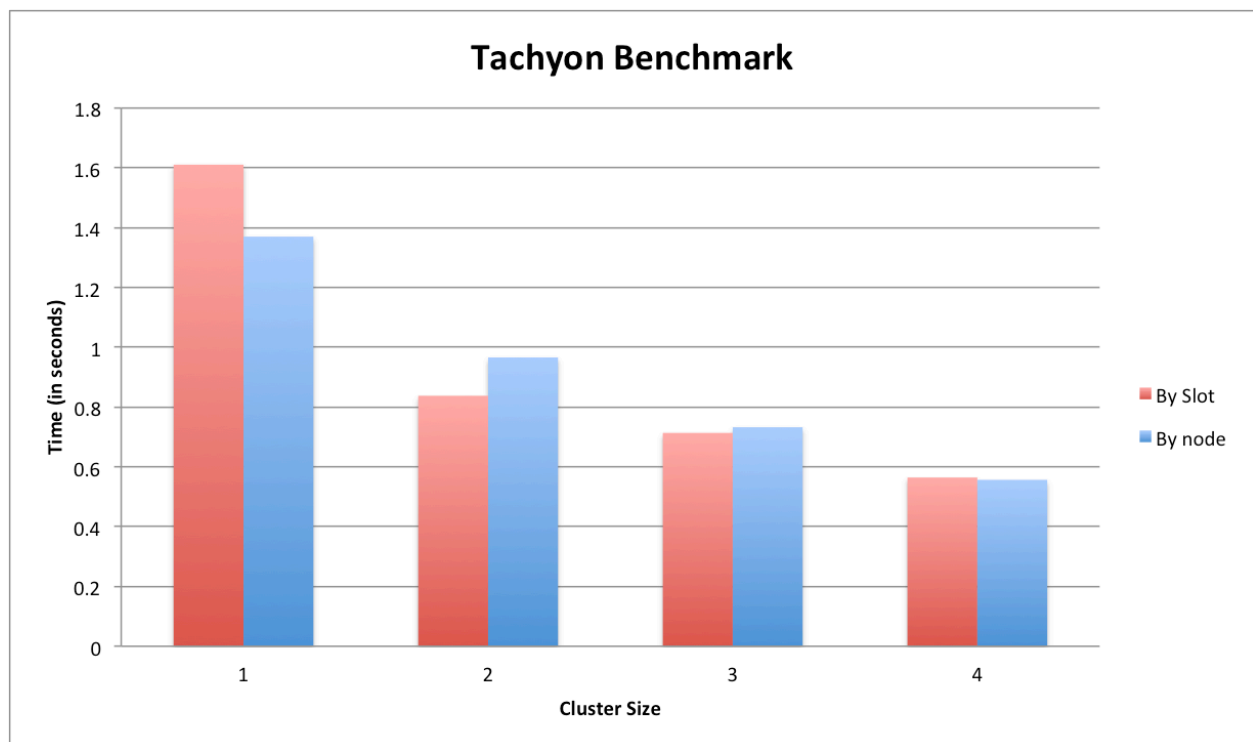


Figure 3. Tachyon Benchmark for both schedules

Considering the performance for two different scheduling policies -byslot and –bynode for each cluster size, Bynode scheduling policies benchmark is not significantly different than Byslot scheduling policies. The reason is my group using 4 host files which have one, two, three, and four IP address for each VM. Since the default slots are two and the processes were scheduled to run to fill all defaults slot (2,4,6,8 processes for each cluster size respectively), both of the schedules can maximize utilizing all of nodes.

Q2. Evaluate the performance of two HPCC (High Performance Computing Challenge) benchmarks: HPL(the Linpack benchmark which measures the floating point rate of execution for solving a system of linear equations) and PTRANS (Parallel matrix transpose).

HPCC-byslot

Cluster 1: PTRANS_GB: 0.72136 HPL_Gflops: 2.488

Cluster 2: PTRANS_GB: 0.388115 HPL_Gflops: 2.463

Cluster 3: PTRANS_GB: 0.180132 HPL_Gflops: 2.925

Cluster 4: PTRANS_GB: 0.122479 HPL_Gflops: 2.325

HPCC-bynode

Cluster 1: PTRANS_GB: 0.788665 HPL_Gflops: 2.97

Cluster 2: PTRANS_GB: 0.417782 HPL_Gflops: 2.823

Cluster 3: PTRANS_GB: 0.179384 HPL_Gflops: 3.276

Cluster 4: PTRANS_GB: 0.140594 HPL_Gflops: 4.793

Graphs:

PTRANS bySlot

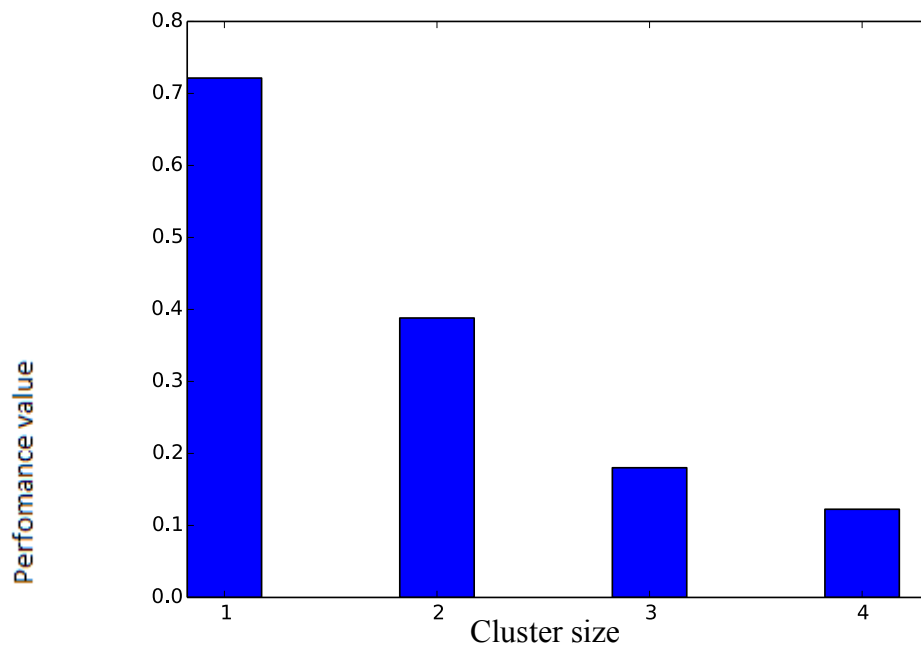


Figure 4. PTRANS Benchmark by Slot

PTRANS bynode

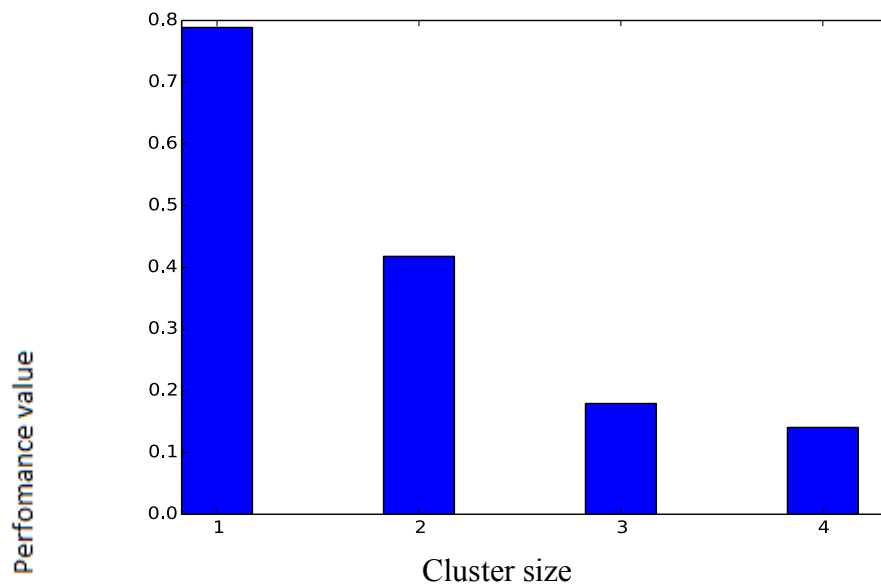


Figure 5.PTRANS Benchmark by Node

Put them together

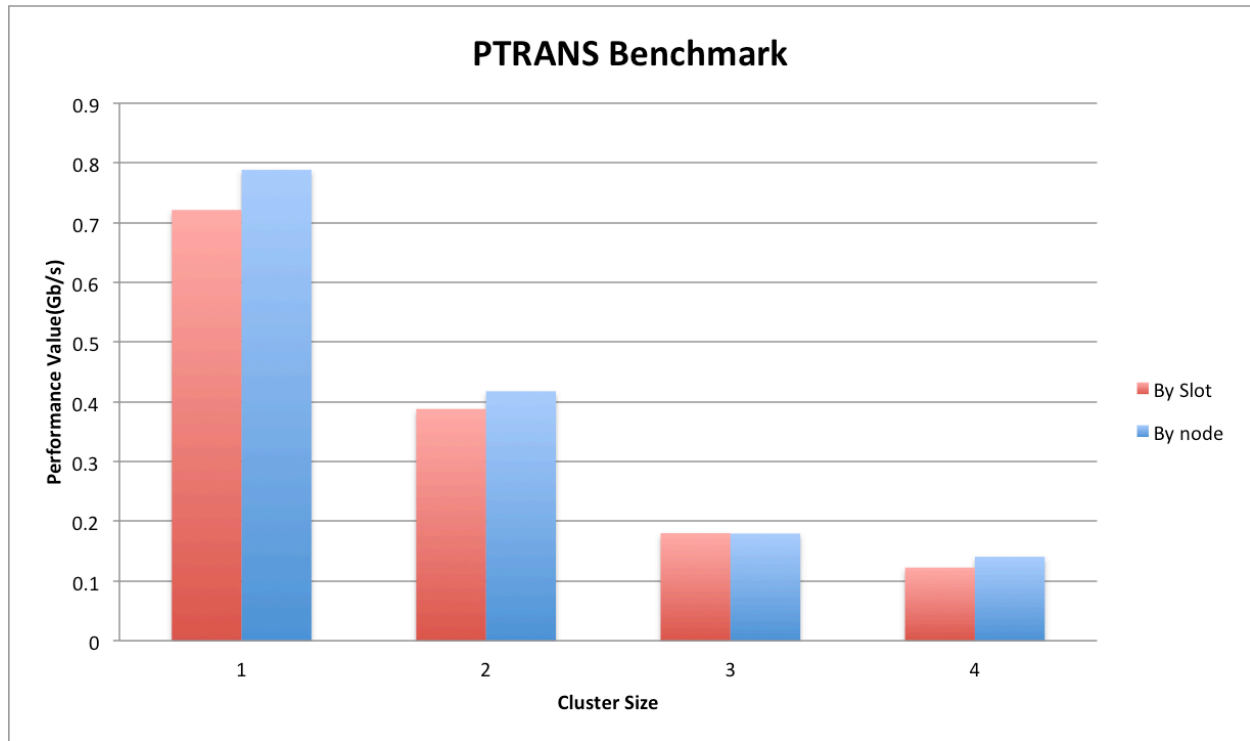


Figure 6. PTRANS Benchmark

In the PTRANS benchmark , there is no significant different between two ways of scheduling. The reason is similar to the above settings, the amount of scheduled tasks are 2,4,6,8 for each cluster sizes 1,2,3,4. Since the default slots are two for each nodes, so every nodes are busy running tasks and bring the best performance regardless the scheduling policy.

HPL byslot

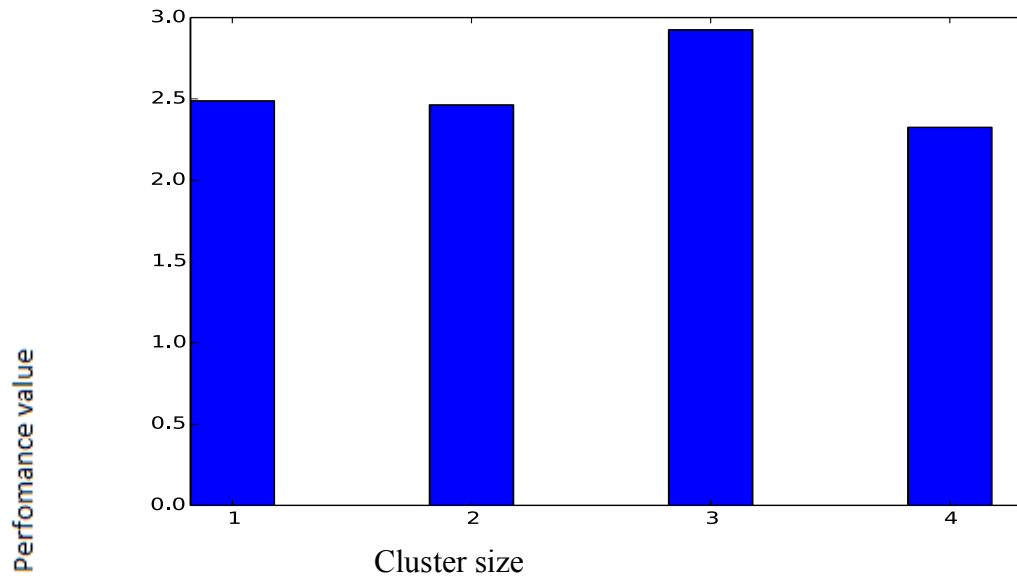


Figure 7.HPL Benchmark by slot

HPL bynode

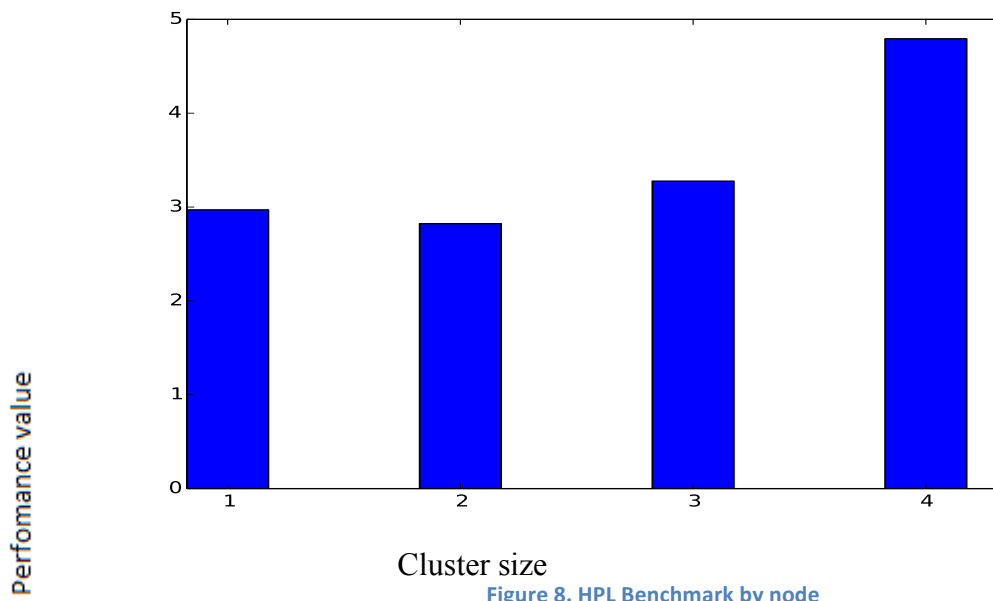


Figure 8. HPL Benchmark by node

Put them together:

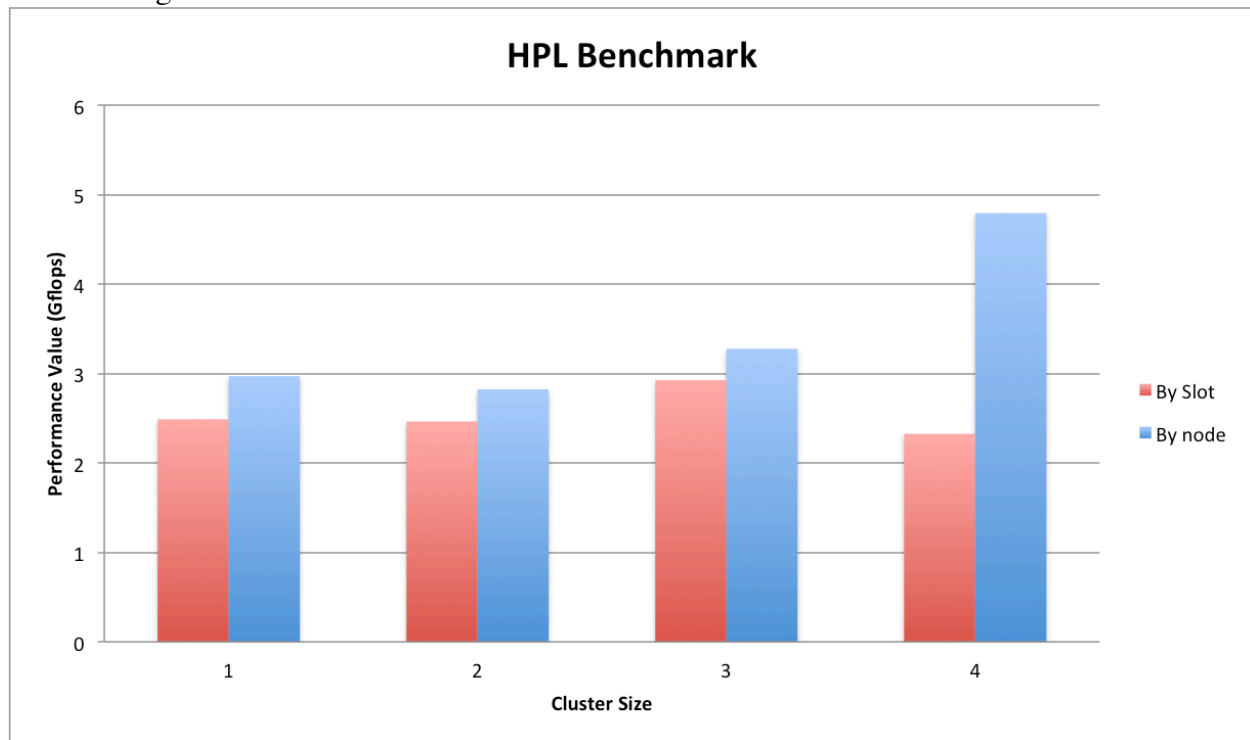


Figure 9.HPL Benchmark

This time, scheduling by node shows more efficient than scheduling by slot. The reason might be the amount of big problem to solve and all the problems are distributed evenly for every node. Scheduling by node is definitely good choice for calculating HPL.

By the analysis of performance of two HPCC (High Performance Computing Challenge) benchmarks: HPL(the Linpack benchmark which measures the floating point rate of executionfor solving a system of linear equations) and PTRANS (Parallel matrix transpose) , we could conclude the following points.

1. With respect to increase in number of nodes the performance of the system is increased.
2. With respect to the performance analysis done , bynode scheduling policy is more efficient over byslot scheduling policy when dealing with big amount of problems.