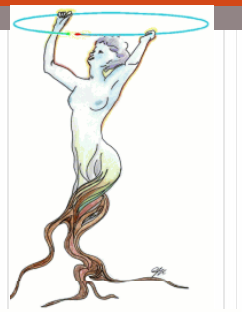
PROOF Benchmark on Different Hardware Configurations

Mengmeng Chen, Annabelle Leung,
Bruce Mellado, Neng Xu and Sau Lan Wu

University of Wisconsin-Madison

PROOFWORKSHOP 2007

Special thanks to Catalin Cirstoiu (ApMon), Gerardo Ganis, Jan Iwaszkiewicz and Fons Rademakers (PROOF TEAM), Sergey Panitkin (BNL)



The Purpose of This Benchmark

- To find out the detailed hardware usage of PROOF jobs.
- To find out what is the bottle neck of processing speed.
- To see the performance on the multi-core system.
- To see the performance on different disk configurations and RAID configurations.
- To see what else also affects the PROOF performances. (the file size, file content)

Hardware/Software Configuration

- Hardware:
  - Intel 8 core, 2.66GHz, 16GB DDR2 memory.
  - With single disk.
  - With 8 disks without RAID
  - With RAID 5(8 disks)
  - With RAID 5(24 disks)
- Benchmark files:
  - Benchmark files.(900MB)
- ATLAS format files:
  - EV0 files(50MB)
- ROOT version
  - URL: http://root.cern.ch/svn/root/branches/dev/proof
  - Repository UUID: 27541ba8-7e3a-0410-8455-c3a389f83636
  - Revision: 21025

The Data Processing settings

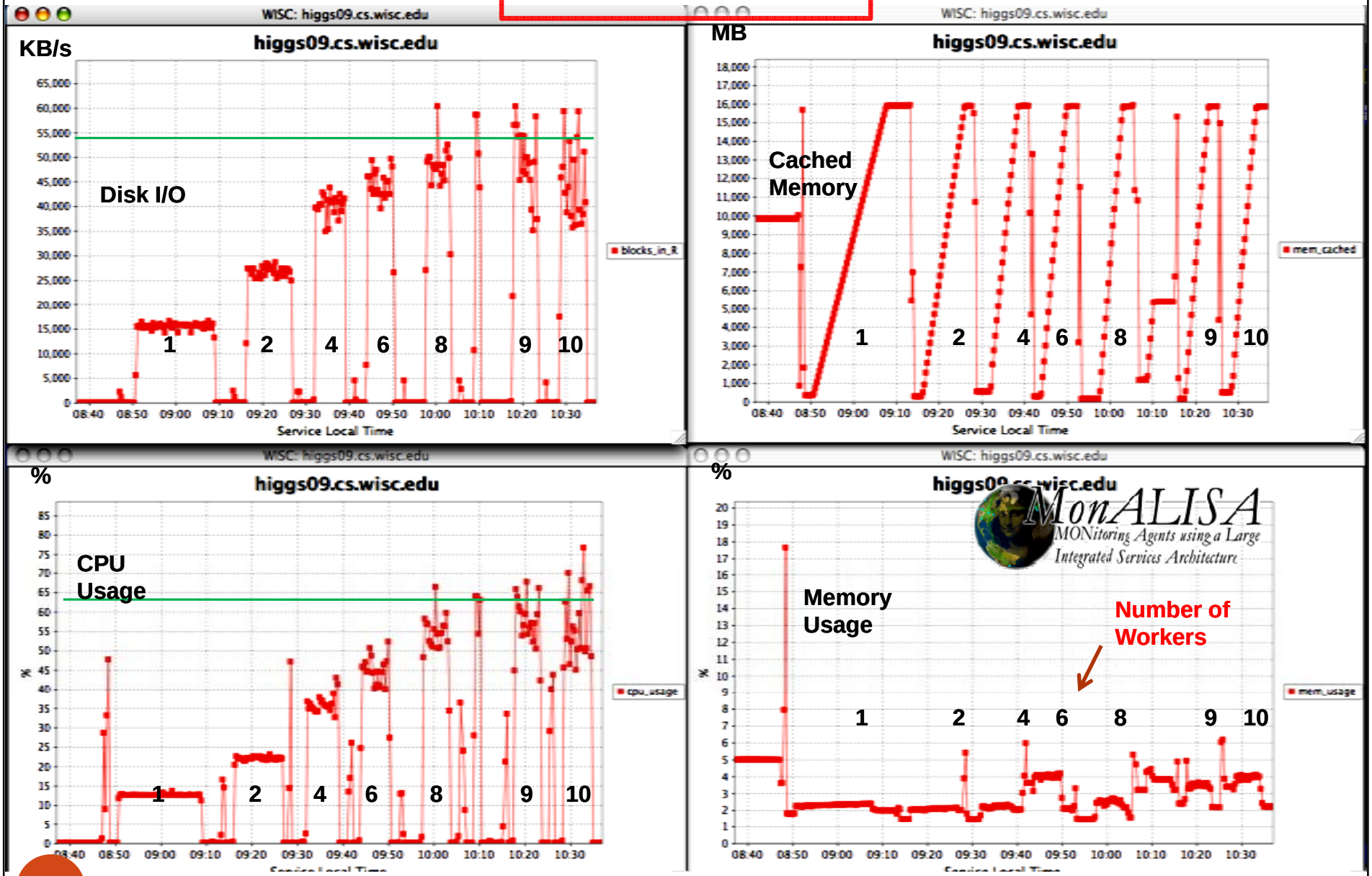
- Benchmark files:
  - Details can be found in $ROOTSYS/test/ProofBench/README
  - With EventTree_ProcOpt.C (Read 25% of the branches)
  - With EventTree_Pro.C (Read all the branches)
- ATLAS format files:
  - With EV0.C (Read only few branch)
- Memory Refresh:
  - After each PROOF job, the Linux kernel stores the data in the physical memory. When we process the same data again, the PROOF will read from memory instead of disk. In order to see the real disk I/O in the benchmark, we have to clean up the memory after each test.

What can we see from the results?

- How much resource PROOF jobs need:
  - CPU
  - Memory
  - Disk I/O
- How does PROOF job use those resources:
  - How to use a multi-core system?
  - How much data does it load to the memory?
  - How fast does it load to the memory?
  - Where does the data go after processing? (Cached memory)

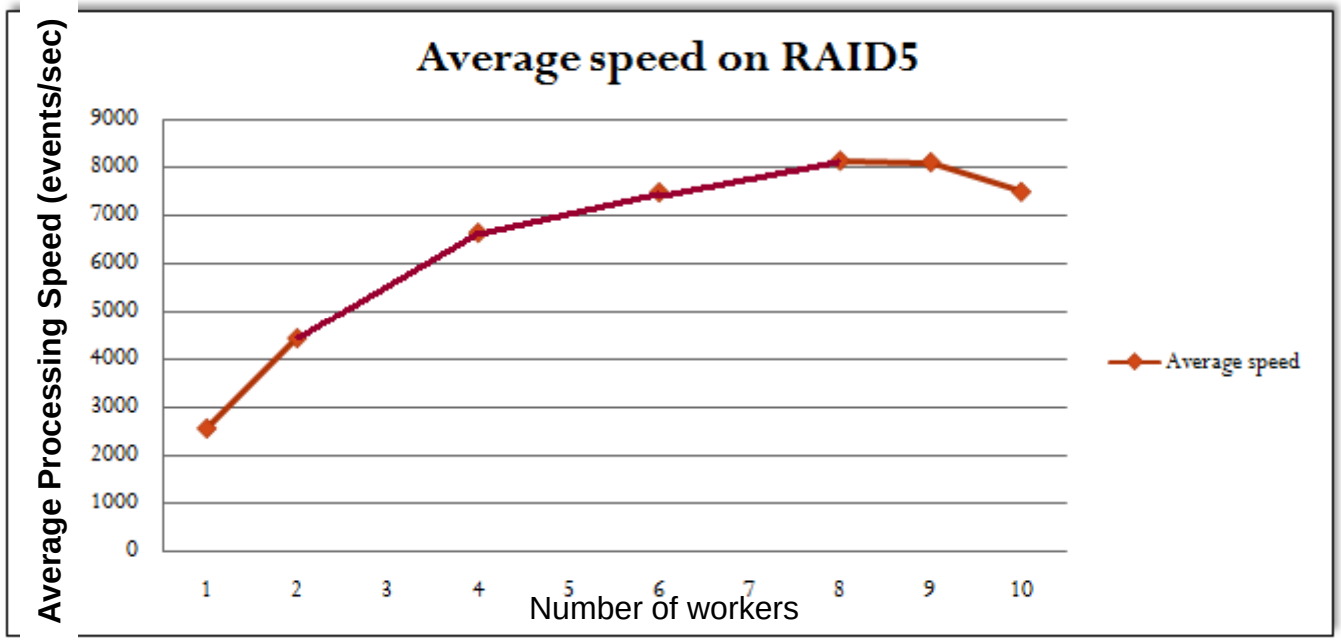
The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

8 disks on RAID 5.



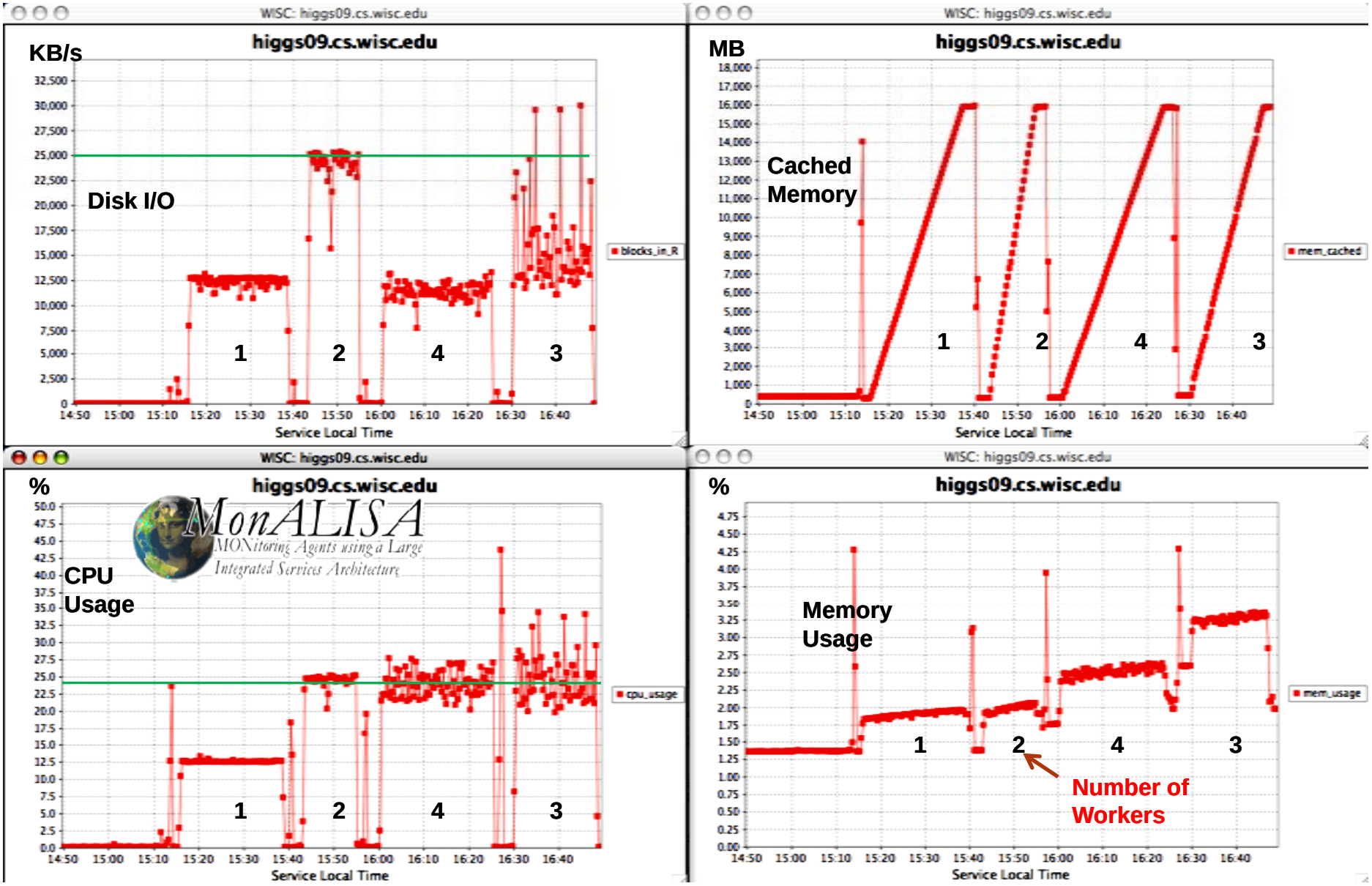
Benchmark files, big size, read all the data

Performance Rate(with RAID 5)



- The maximum disk I/O can reach ~55MB/sec
- Performance keeps increasing until the num. of workers is equal to the num. of CPUs
- CPU usage is limited by the raid disk throughput.
- Scalability is limited and affected by the Raid disk throughput after number of workers reaches 6

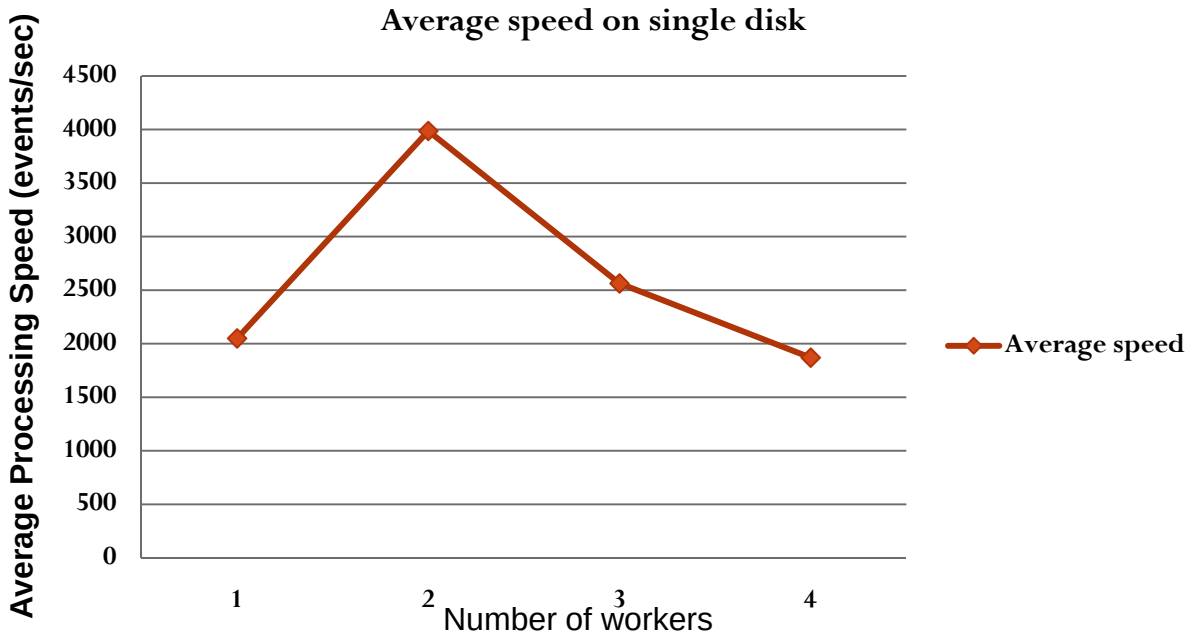
Benchmark files, big size, read all the data



The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

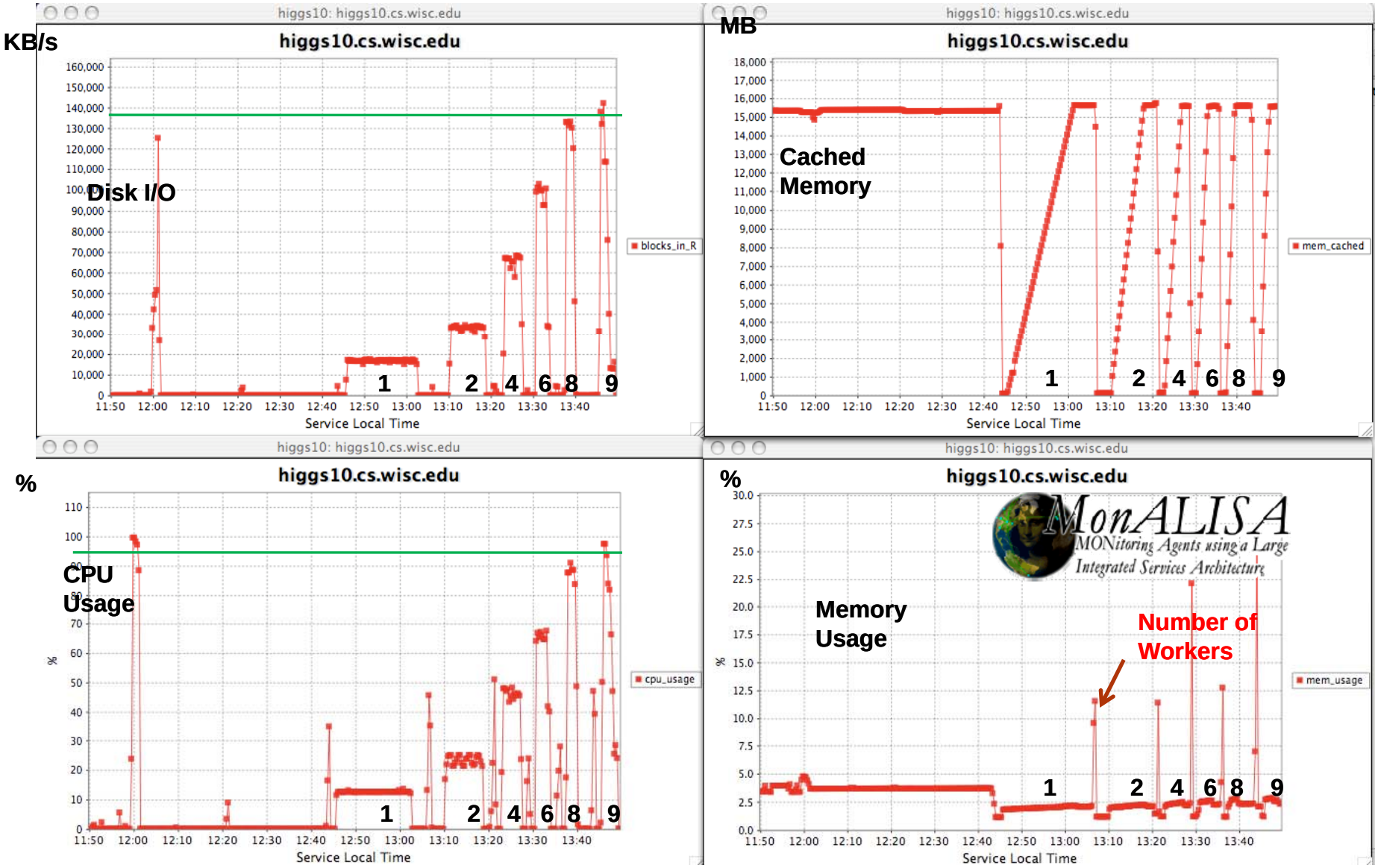
Single Disk.

Performance Rate (with 1 disk)



- The maximum disk I/O can reach ~25MB/sec
- CPUs are only ~25% used.
- 2 workers provide the best performance. For single disk, two proof workers will best utilize the throughput of the disk. Therefore, 2 cores vs. 1 disk is an ideal solution.

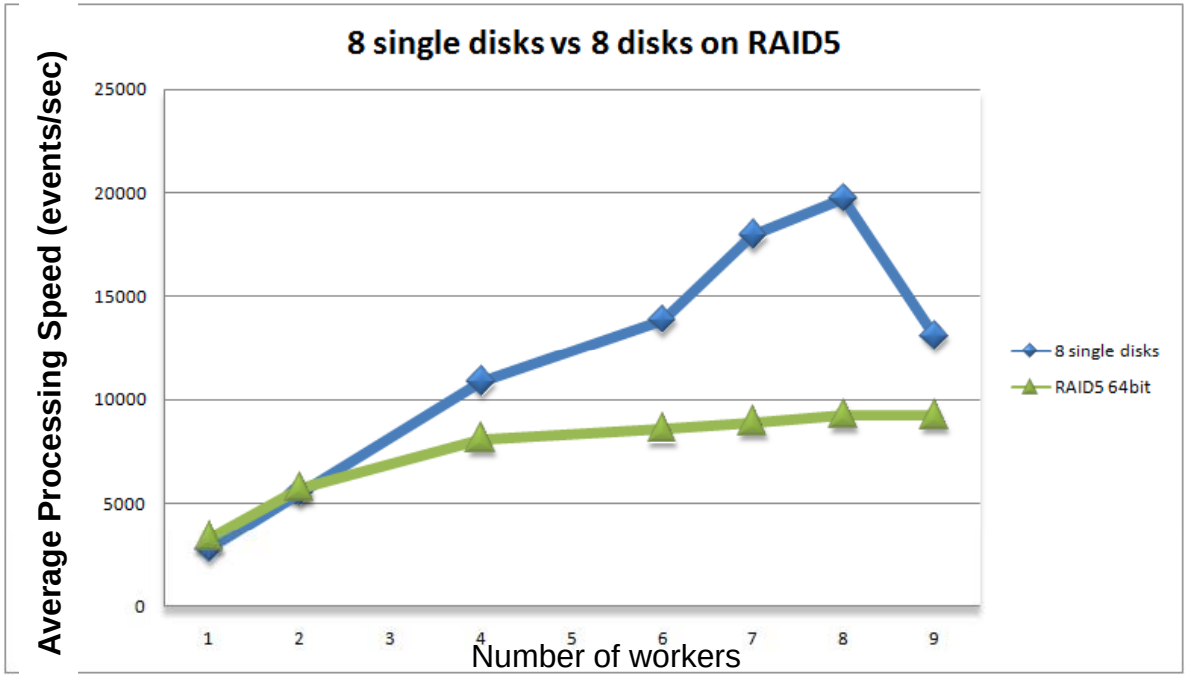
Benchmark files, big size, read all the data



The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

8 single disks on RAID Controller

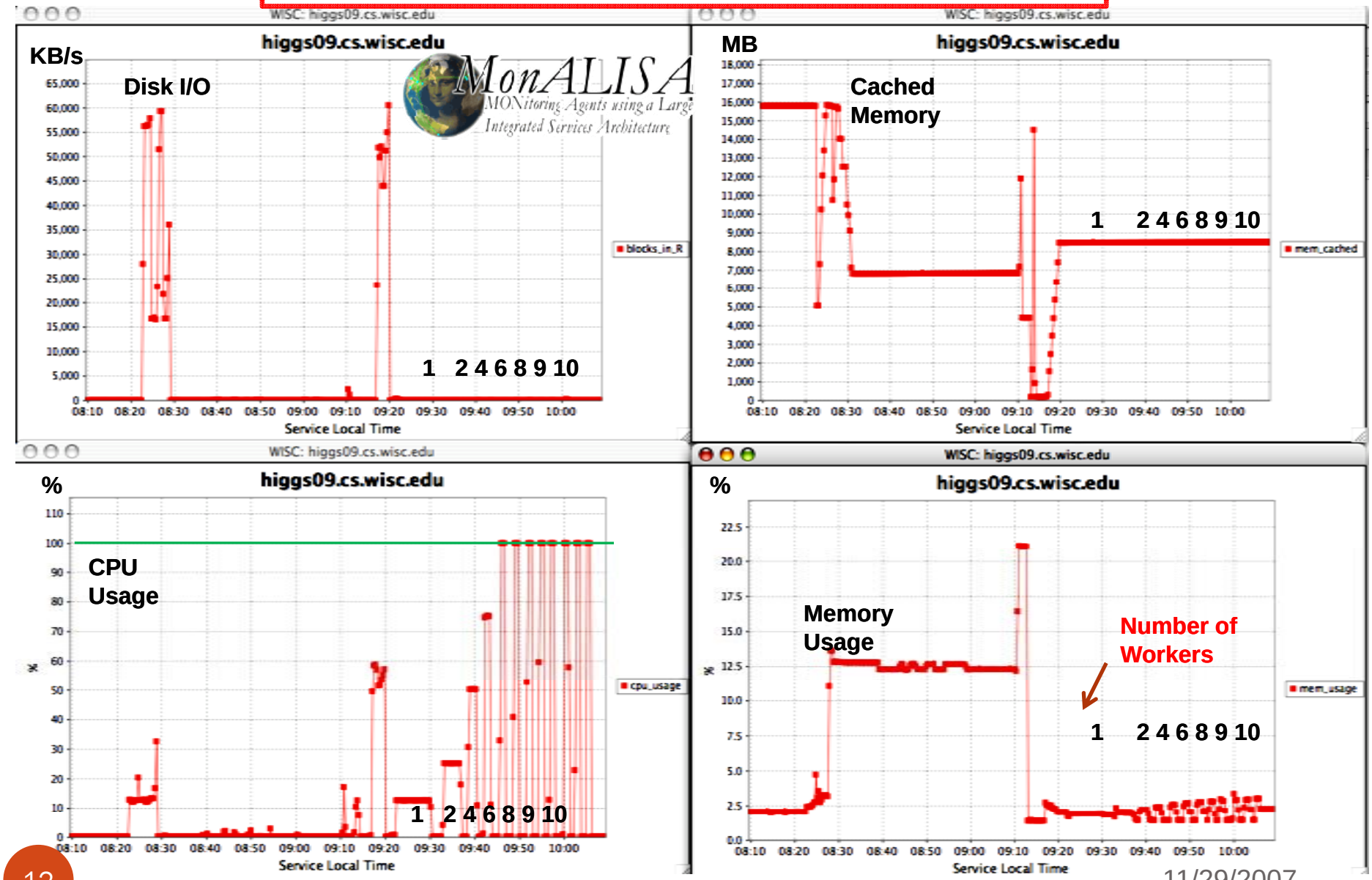
Performance Rate (RAID vs. non-RAID)



- Using single disks can provide significant increase of Disk throughput and performance.
- Trade-off between data protection and performance has to be carefully checked.
- 8 single disk could sustain the scalability before running out of 8 CPUs.
- More concrete conclusion could be acquired in the same test on the machines with more cores and disks(ie. 16CPU,24 disks)

The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory.

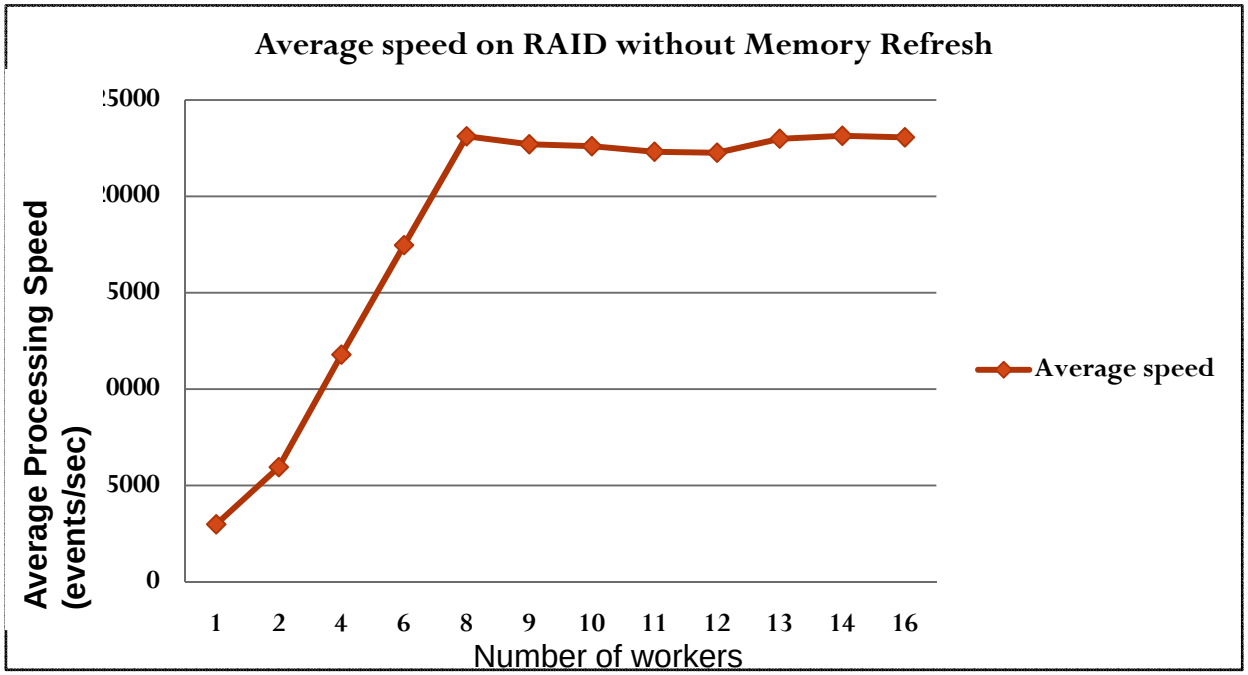
8 disks on RAID 5. Without Memory Refresh



11/29/2007

Benchmark files, big size, read all the data

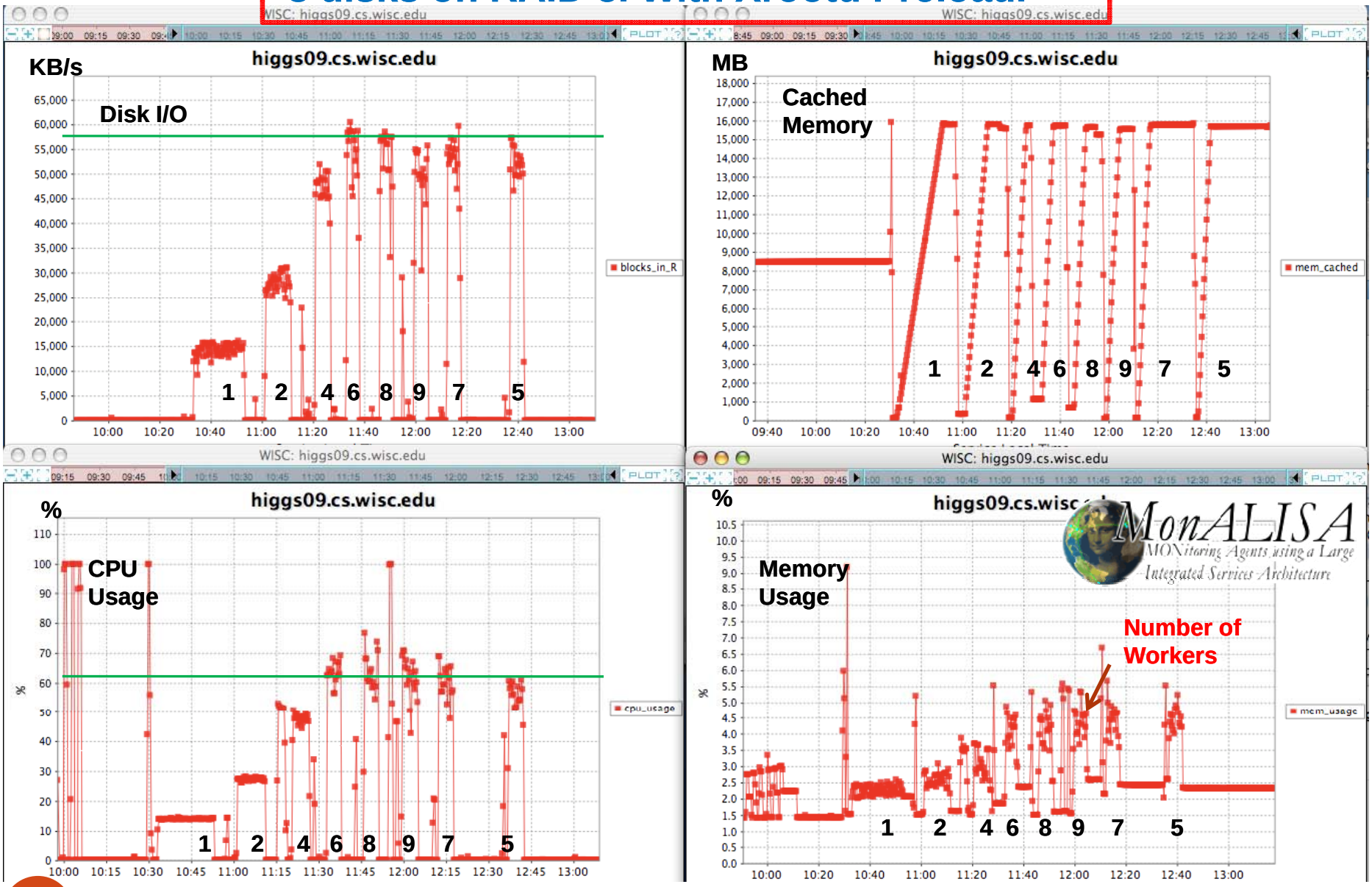
Performance Rate (read from memory)



- When we didn't refresh the memory, there was no disk I/O at all.
- CPUs usage reached 100% and it becomes the bottleneck.
- CPU becomes the only bottleneck for analysis speed and scalability.

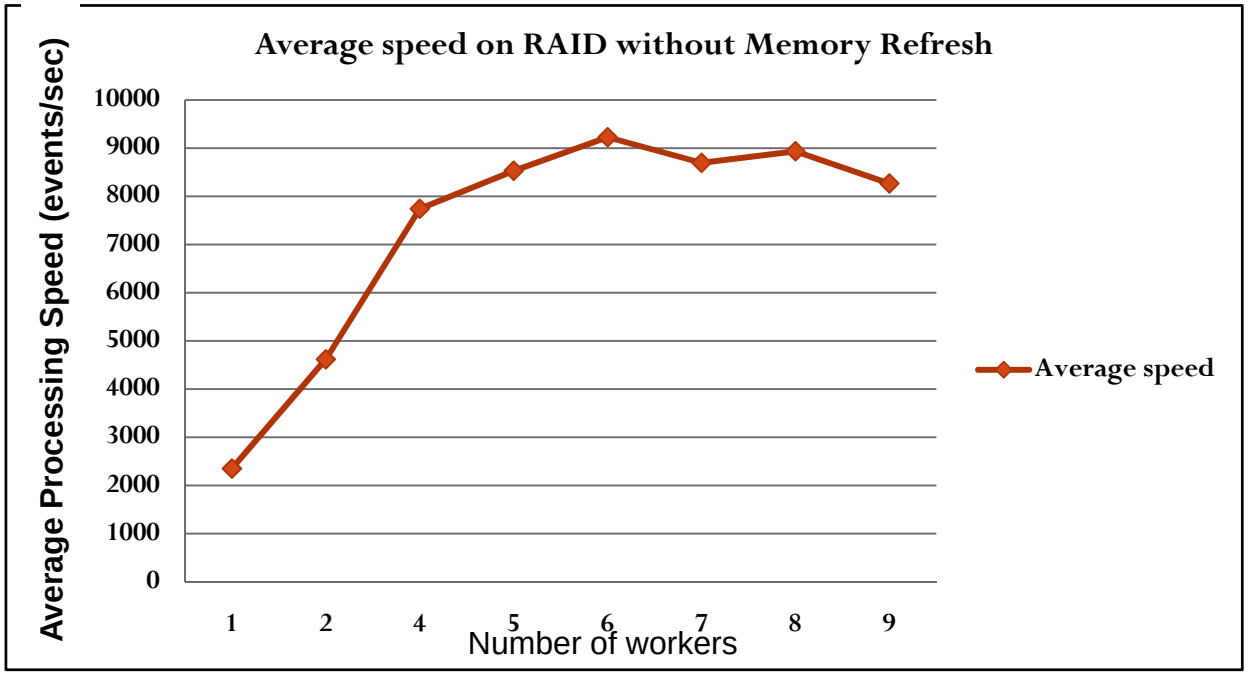
The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

8 disks on RAID 5. With Xrootd Preload.



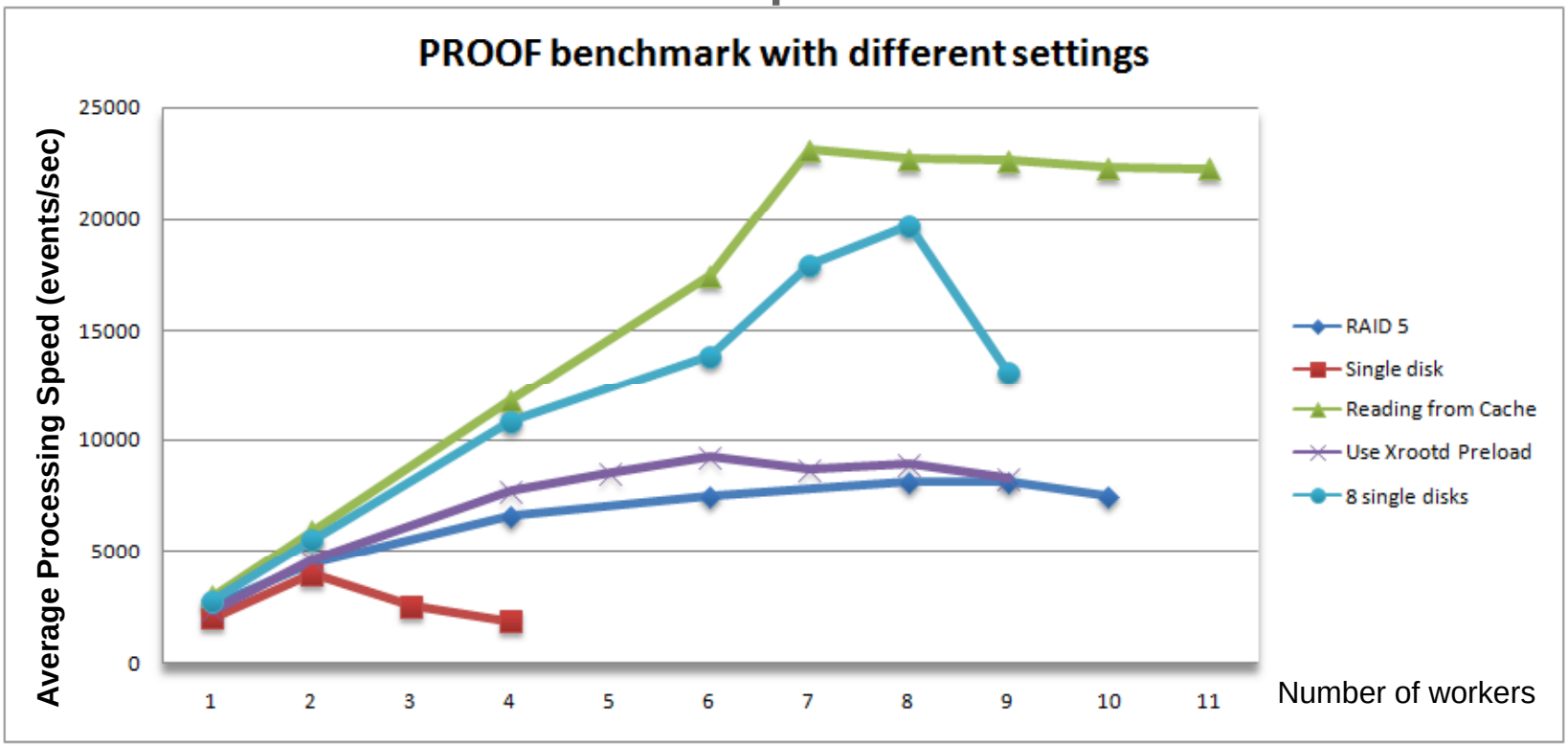
Benchmark files, big size, read all the data

Performance Rate(with Xrootd preload)



- Xrootd preloading doesn't change disk throughput much. (see Page 7)
- Xrootd preloading helps to increase the top speed by ~12.5%.
- When we use Xrootd preload, disk I/O reaches ~60MB/sec
- CPUs usage reached 60%.
- The best performance is achieved when the number of workers is less than the number of CPUs.(6 workers provides the best performance.)

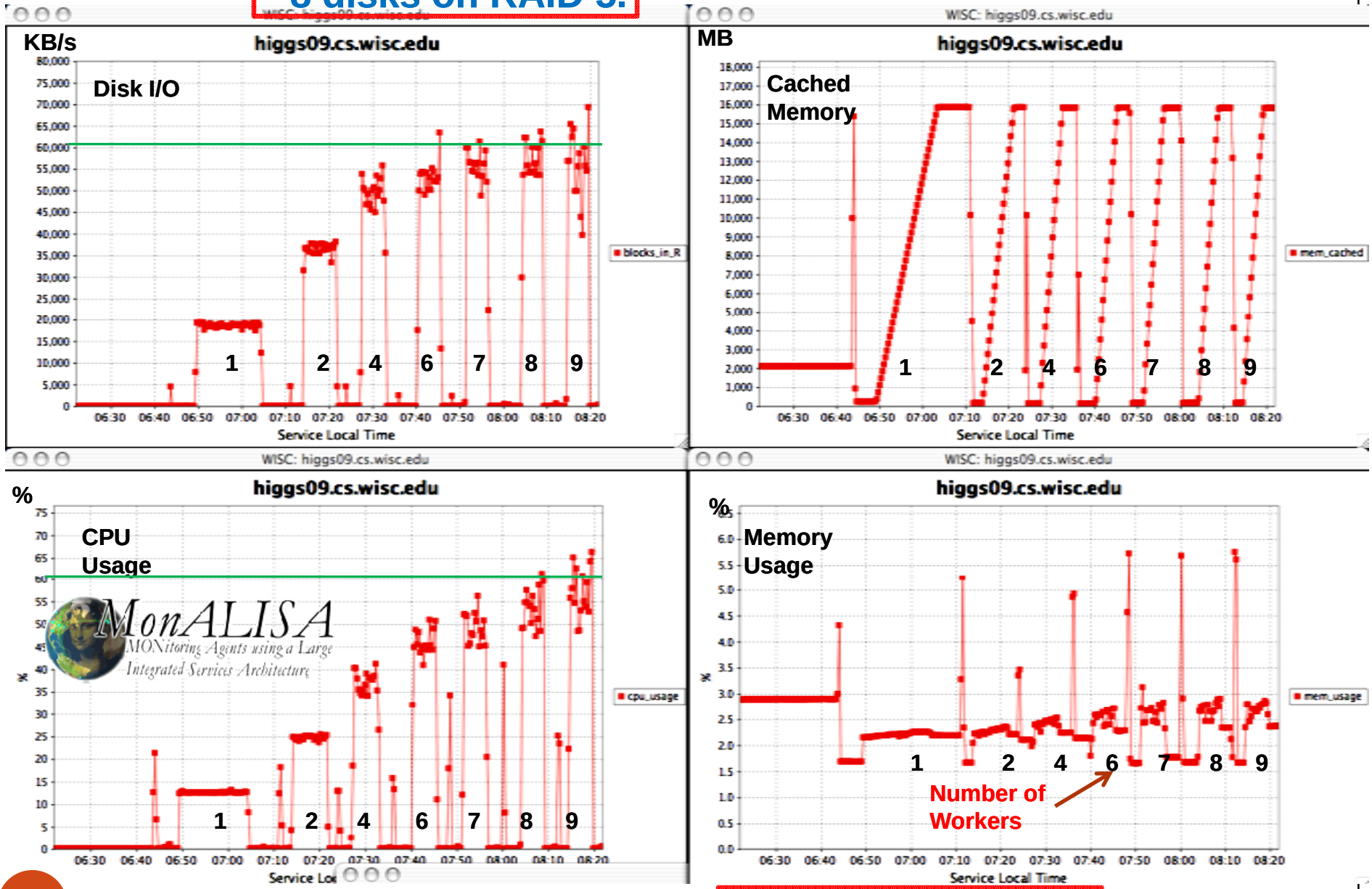
An overview of the performance rate



- For I/O bound job, disk throughput will be the bottleneck for most of the hardware configurations; reading data from memory will provide the best scalability and performance.
- Trade-off between the data protection and the working performance has to be made.
- Using Xrootd Preload function will help to increase the analysis speed.
- 2 cores vs. 1 disk seems to be a reasonable hardware ratio without using Raid technology.

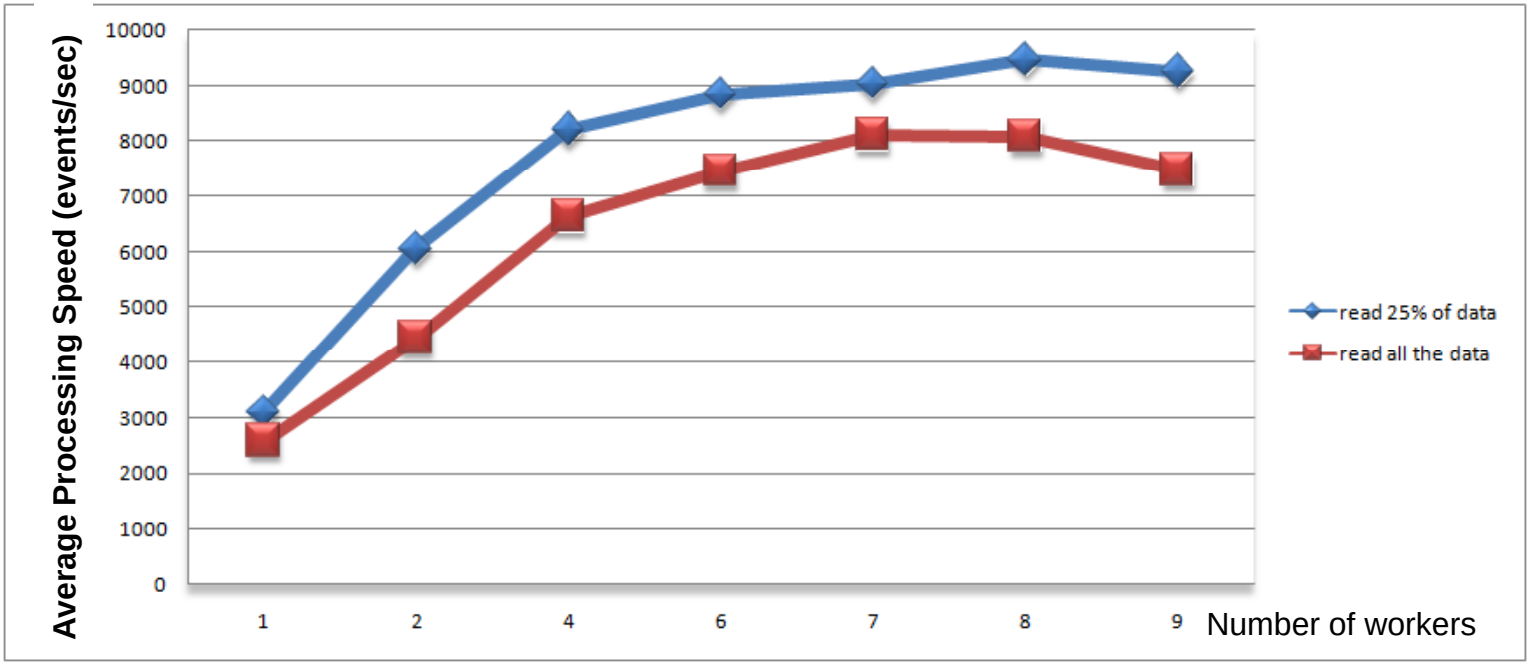
The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

8 disks on RAID 5.



Benchmark files, big size, read 25% of the data

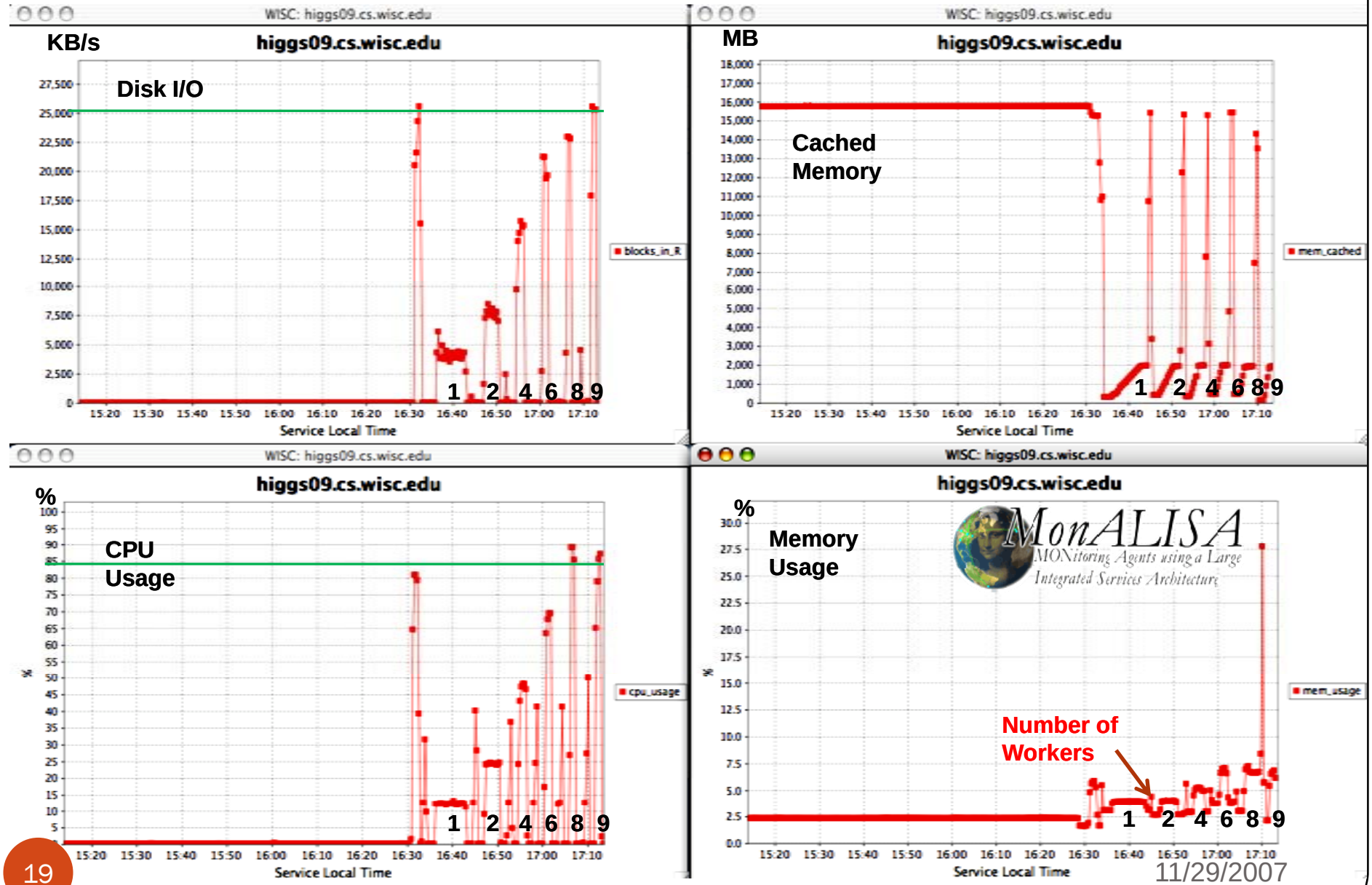
Performance Rate



- For I/O-bound-like case(benchmark analysis) , the less I/O intensive job will have a better performance.

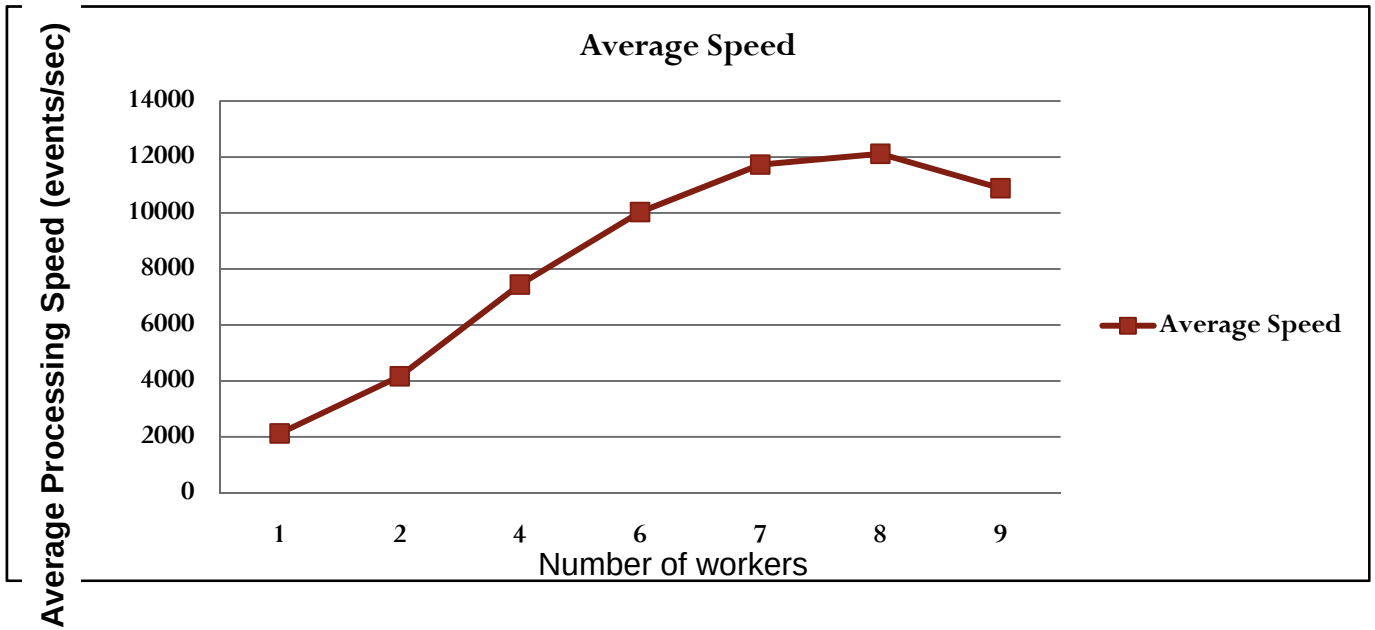
The jobs were running on a machine with Intel 8 core, 2.66GHz, 16GB DDR2 memory,

8 disks on RAID 5.



EV0 files, read one branch

Performance Rate(with ATLAS data)



- Disk I/O only reaches ~25MB/sec even when we were using RAID.
- CPUs usage reached ~85%.
- 8 workers provide the best performance.

Summary

- We think PROOF is a viable technology for Tier3s . Various performance tests have been done in the purpose of understanding the usage of the computing resource of Proof and seeking the optimal configurations of different computing resources.
- Number of issues have come to our attention :
  - The balance between CPU processing power and the disk throughput.
  - The configuration of Xrootd file servers. (Xrootd preload)
  - The trade-off between performance and data protection.
  - Different types of jobs. (I/O intensive or CPU intensive)
  - Hardware/software configurations of the data-serving storage.
- More issues need to be understood in designing a larger scale proof cluster for multi-users.(users priority, proof job scheduling, network traffic, data distribution, and etc.)