

100% [d]Caching at Illinois

Sarah Williams (Indiana University)
David Lesny (University of Illinois)

USAtlas Distributed Computing Workshop at the
University of Arizona 2013-12-11

Caching at a Distributed T2

- What do we already cache at a local site level?
 - CVMFS (software), pCache (production inputs and analysis non-ROOT inputs), FRONTIER (conditions metadata)
- What could still benefit from caching?
 - Analysis ROOT files. It is not efficient to put these into pCache on individual worker nodes disks because often only a small portion of the file is used. Reading them off the storage server is better, but because analysis jobs often do many smaller reads they are adversely affected by latency. The magnitude of this effect depends on the job.
 - Production inputs and Analysis non-ROOT files could benefit from being cached at the local site, as well as at on the worker node.

Illinois Campus Cluster

Storage Infrastructure

- (1) DDN SFA12K40D-56IB Couplet with 5 enclosures
- 60 3TB 7,200 rpm 2.5 SATA drives expandable to 600 disk drives
- (4) Dell PowerEdge R720 GPFS-NDS Nodes each configured with:
 - (2) Intel E5-2665 2.4GHz 8 Core Processors, 115W
 - 256 GB RAM via (16) 16GB 1333MT/s RDIMMs
 - Mellanox ConnectX-3 Dual Port FDR IB HCA
 - Intel X520 DP 10Gb DA/SFP+ Server Adapter
 - (4) 300GB 15K RPM 6Gb/s SAS drives

Network Infrastructure

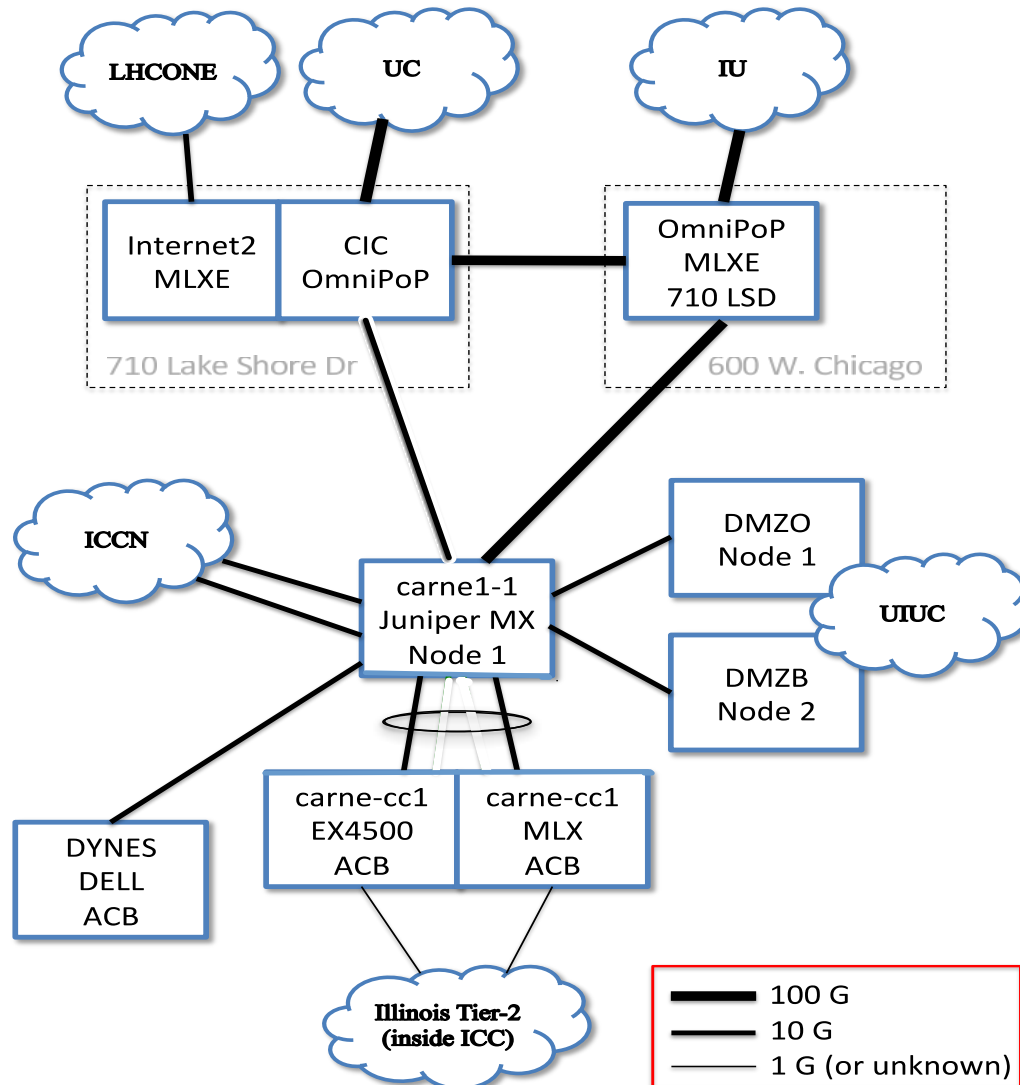
- High Speed InfiniBand Cluster Interconnect
 - Mellanox MSX6518-NR FDR InfiniBand (384 port capable)
- Management and IPMI Control Networks
 - (2) Dell PowerConnect 8064F 48 Port 10GigE switches
 - (41) Dell PowerConnect 5548 49 Port 1GigE switches
 - (2) Dell PowerConnect 2848 48 Port 1 GigE switches

dCache Local Site Headnode

- mwt2-s1 is a Dell R710 with 96GB memory and a dual port mellanox interface.
- It has an Infiniband connection to the storage infrastructure, and a 10Gb connection to the worker nodes and to the other MWT2 sites.

Source: <https://campuscluster.illinois.edu/>

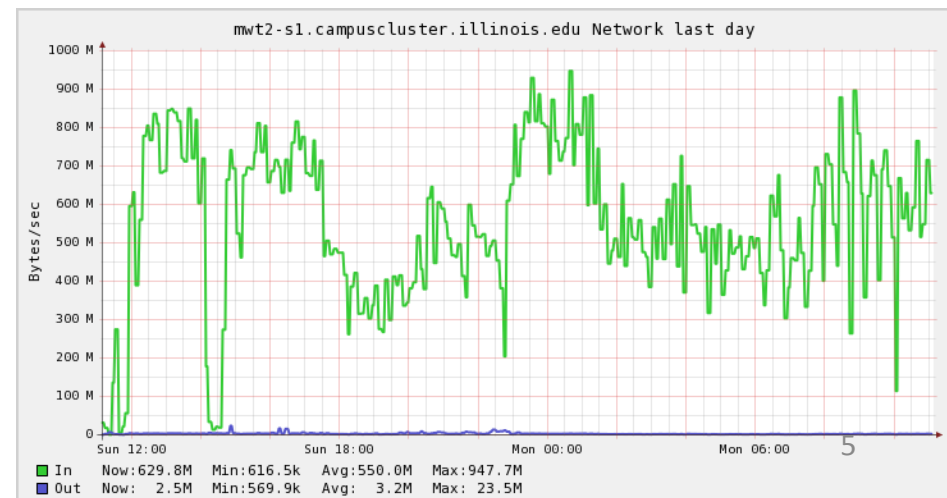
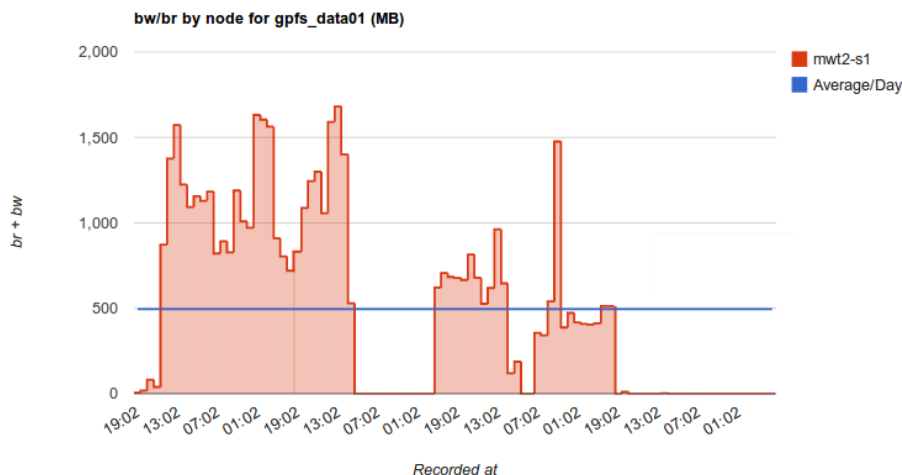
Illinois Campus Cluster



Pre-staging Data

- To prep this machine for actual production, we filled it with recently accessed data from the other data servers
- Accomplished with dCache 'permanent migration feature'
 - *migration copy -tmode=cached -target=pgroup -accessed=0..300 -refresh=300 -cpuCost<0.3 -permanent mwt2ICCPools*
 - Could be a long-term strategy, to stage data to the local site before jobs request it.
- Through trial-and-error, found 12 active migration (two per pool) gave best performance, writing at a consistent 6.25Gbps. We saw peaks of almost line speed (10Gbps). Transferring large files was efficient at using the bandwidth than small files.

Bandwidth of reads and writes to GPFS, in MB/s Network bandwidth to mwt2-s1, in MB/s

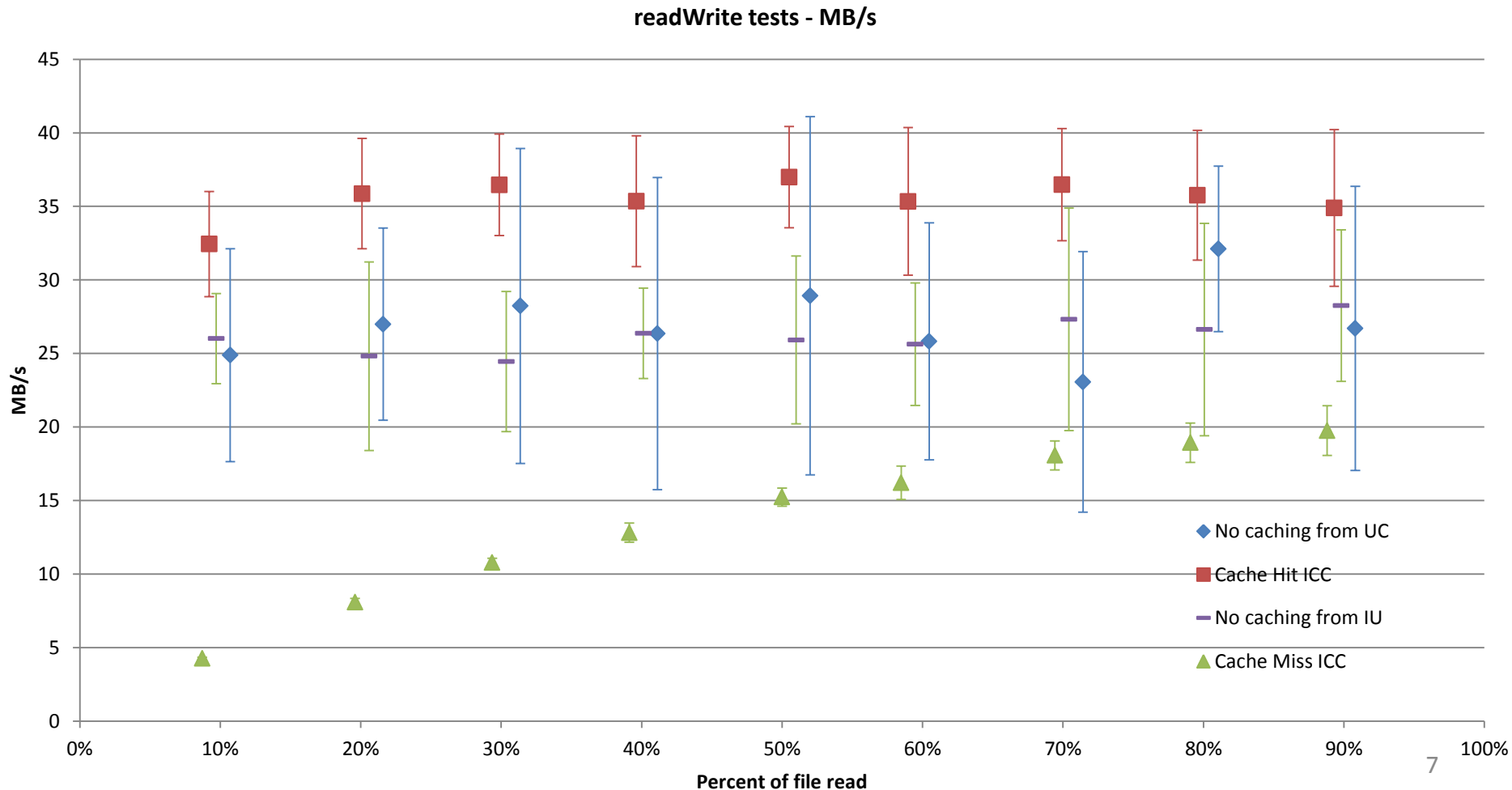


Performance Testing

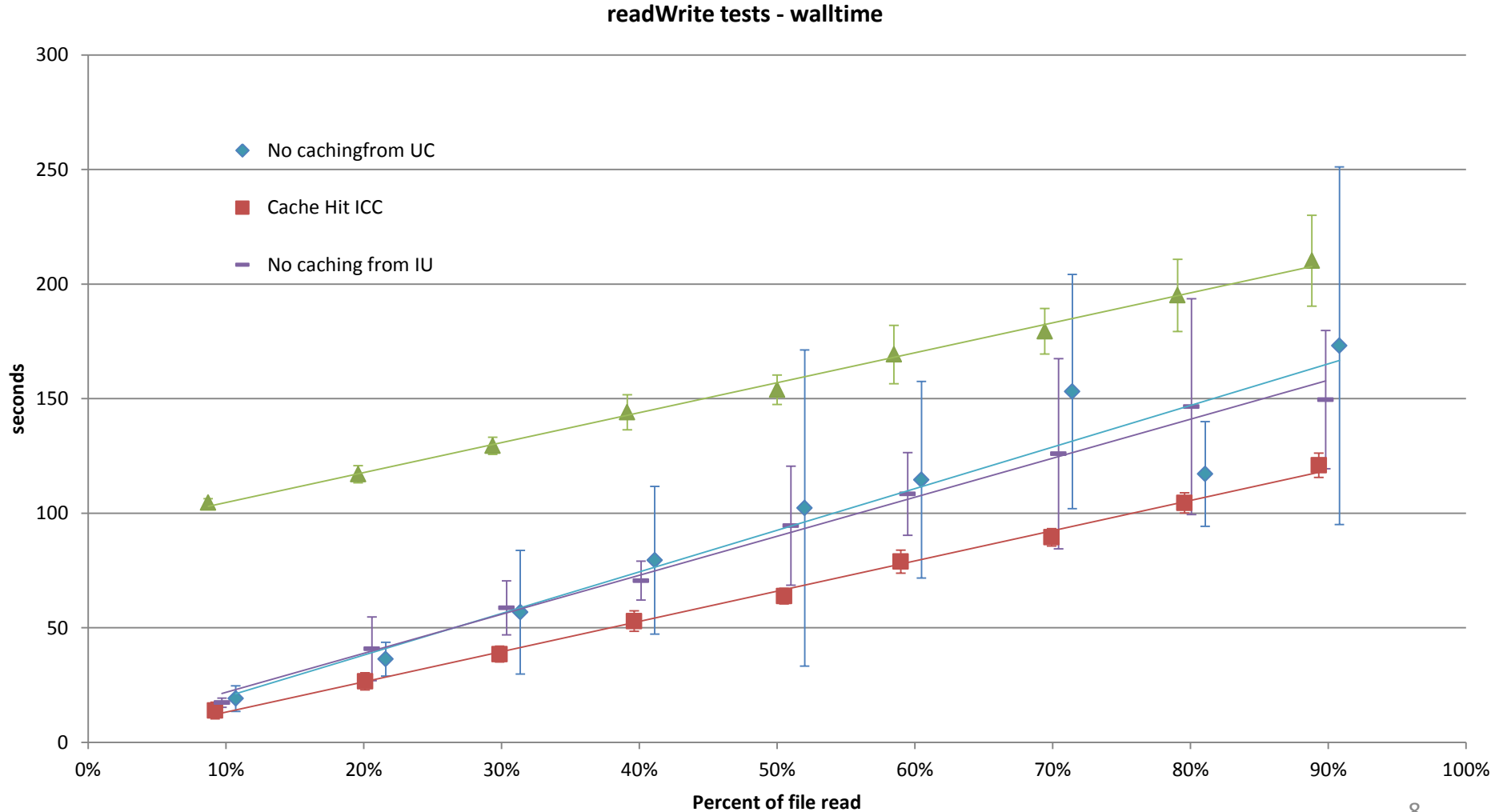
- To model an analysis job, we used readDirect from the FAX End-user tutorial code base:
 - <https://github.com/ivukotic/Tutorial/>
- readDirect opens a ROOT file and reads the specified percentage of events from it.
- We cannot directly control the bytes read, only number of events read. Instead we ran a series of tests to find the number of events read which would correspond to 10-90% of bytes read.
- The test file is NTUP_SMWZ.01120689._000089.root.1, a fairly typical user input file of size 4.6 GB.
- The tests read 10%-90% of the file, in 10% increments

Performance Testing

Cache misses are always worse than direct access. Caching in this case only makes sense if files are re-used.



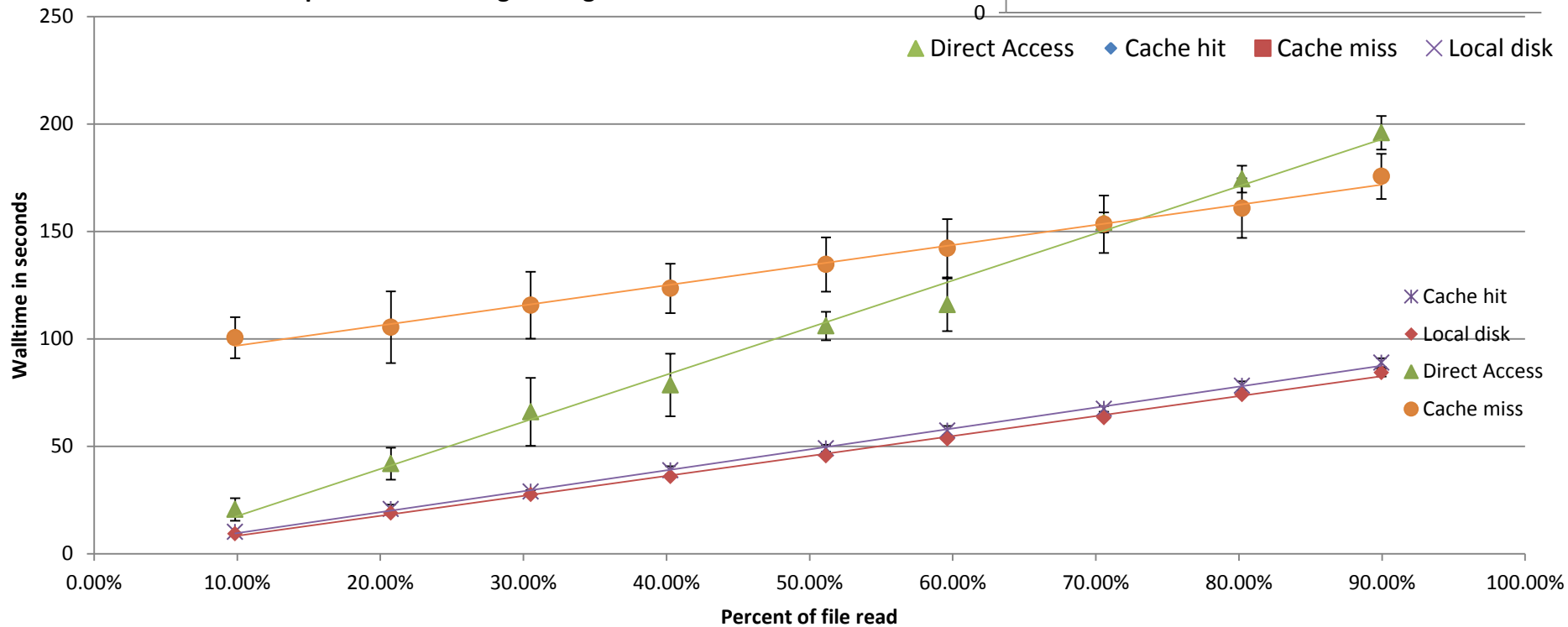
Performance Testing



From IU in August: Direct comparison of caching strategies

No caching is the optimum strategy when less than 75% of the file is read. When more than 75% is read, caching becomes optimal.

Comparison of Caching Strategies on X5660



Comparison of Download Speed of Caching Strategies, on X5660

