Performance Testing of DDN WOS Boxes

Shaun de Witt, Roger Downing

Future of Big Data Workshop

June 27th 2013

Topics

- Bit of Background on WOS
- Testing with IRODS
- Testing with compute cluster

WOS Boxes

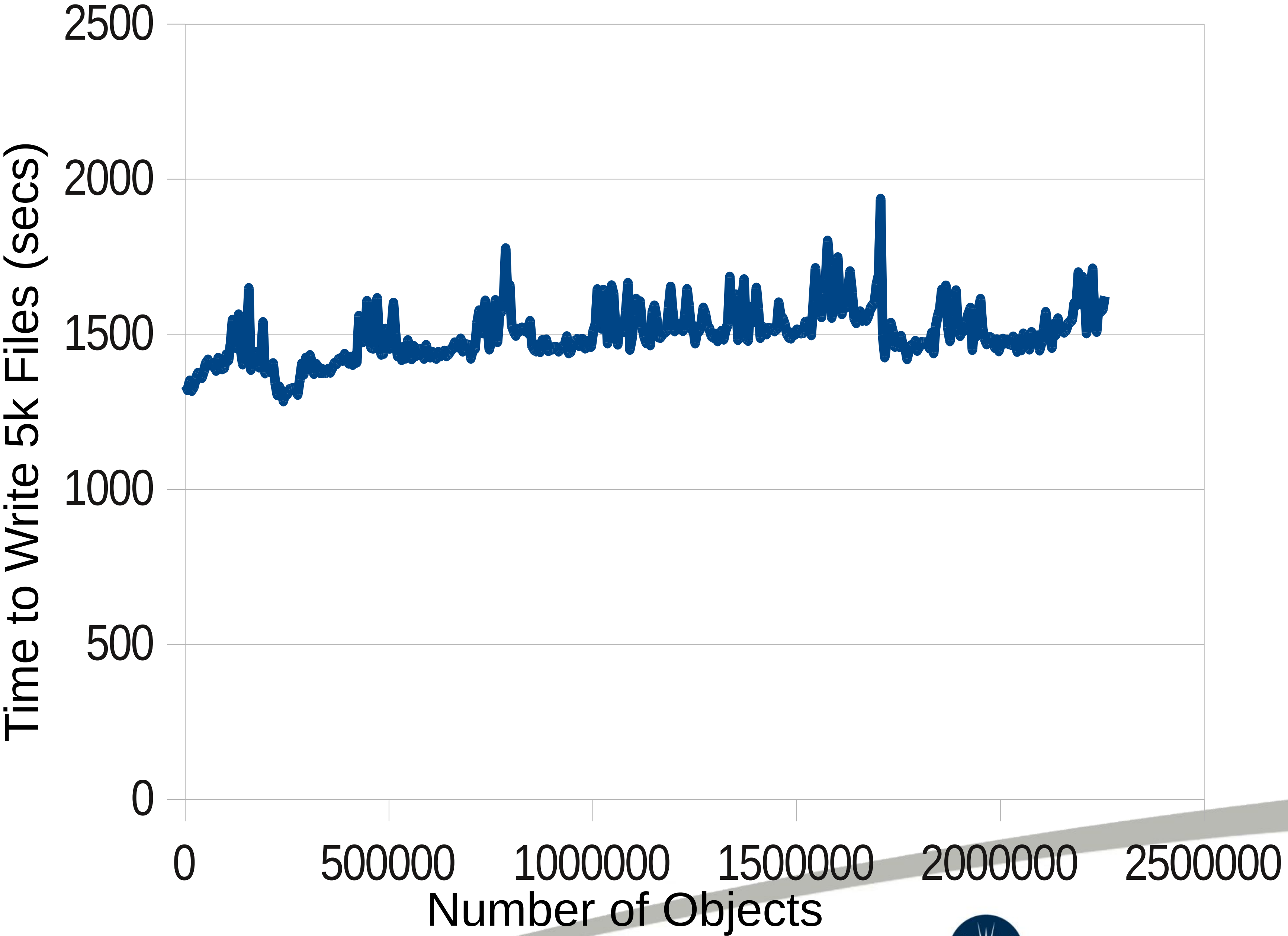
- WOS in a nutshell (and why STFC are interested)
  - DataDirect Networks Web Object Scaler
  - Policy based replication between boxes at remote sites
  - Data Access through standard protocols
  - Fast Data Access Times
  - Easy integration with GPFS and IRODS
- Loan DDN WOS Boxes Provided by OCF
  - STFC are very grateful for this opportunity to test hardware

Use With IRODS

- STFC runs IRODS as part of EUDAT Project
- Reasons for Possible use of WOS within EUDAT
  - High Speed Online Storage
  - Replicated data between STFC sites at RAL and DL
- IRODS Setup
  - Used IRODS 3.2
  - RHEL 6.2 VM with single core @ 2.4 GHz and 1GB RAM
    - Definitely sub-optimal!
  - Data loaded to cache and then sync'd to WOS

IRODS Testing (1)

- Tested object storage rate using scripts developed as part of EUDAT
- Sequentially add small files in groups of 5k
  - Looking for evidence of performance degradation with capacity or number of inodes used
- Test ran for ~ 1 week
  - 2.26 Mobjects inserted



Science & Technology
Facilities Council

Prelim Analysis

- Quite flat upload times
  - If anything rates increase as number of entries increases!
- Long period humps associated with Oracle Automated Optimisation
- Sort period humps associated with Oracle backups
- Observed rate on writing files (single client) ~5/sec
  - Almost certainly due to bottlenecks associated with VM

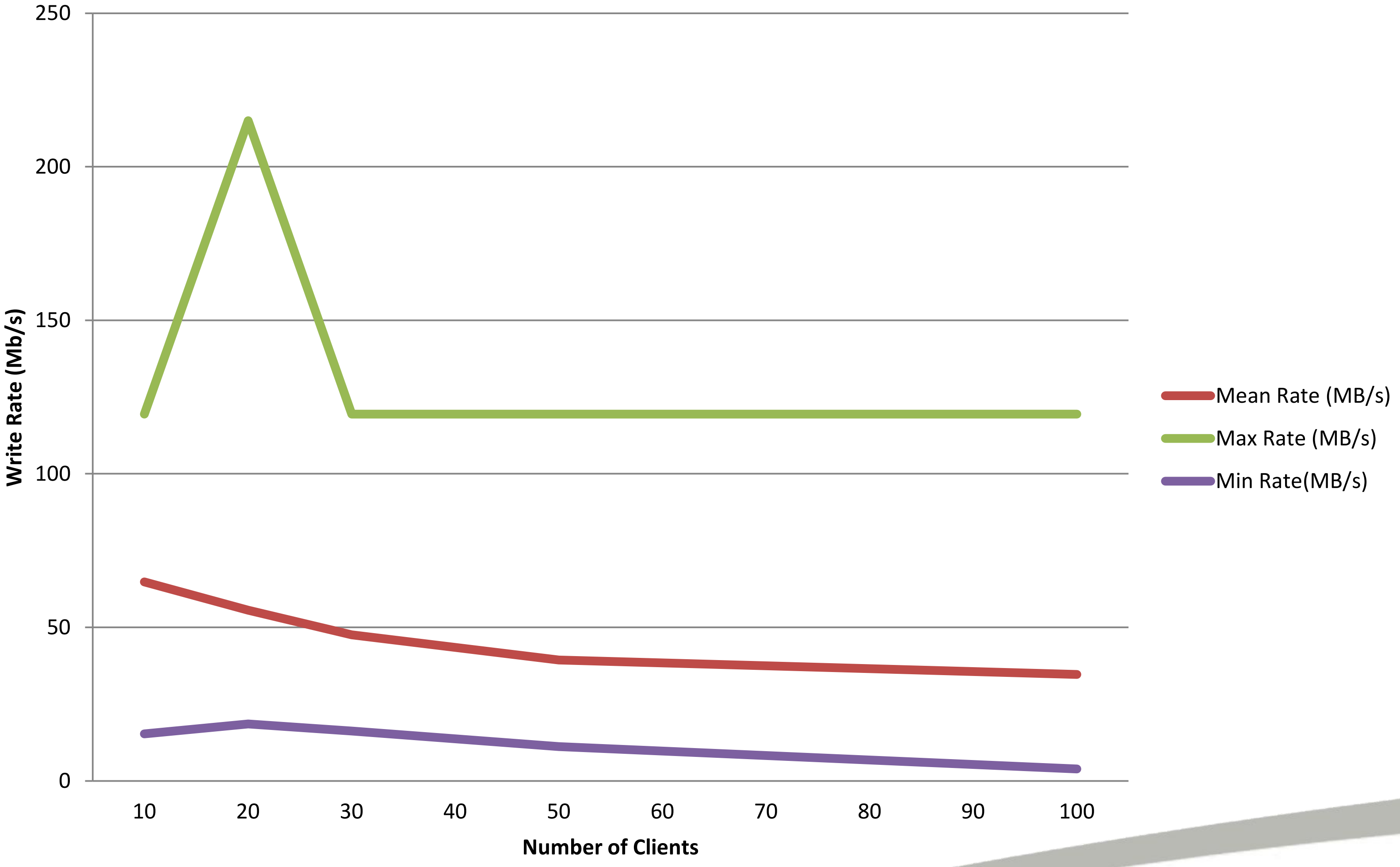
Other IRODS Tests

- Tried to perform massively parallel tests of WOS through IRODS but frustrated again by VM limitations
  - Stressed the VM more than WOS!
- Tested WOS directly using WOS native C/C++ API
  - Parallel writes of 1KB file with >30 threads completed
  - Object creation rate observed in WOS ~10 objects/sec

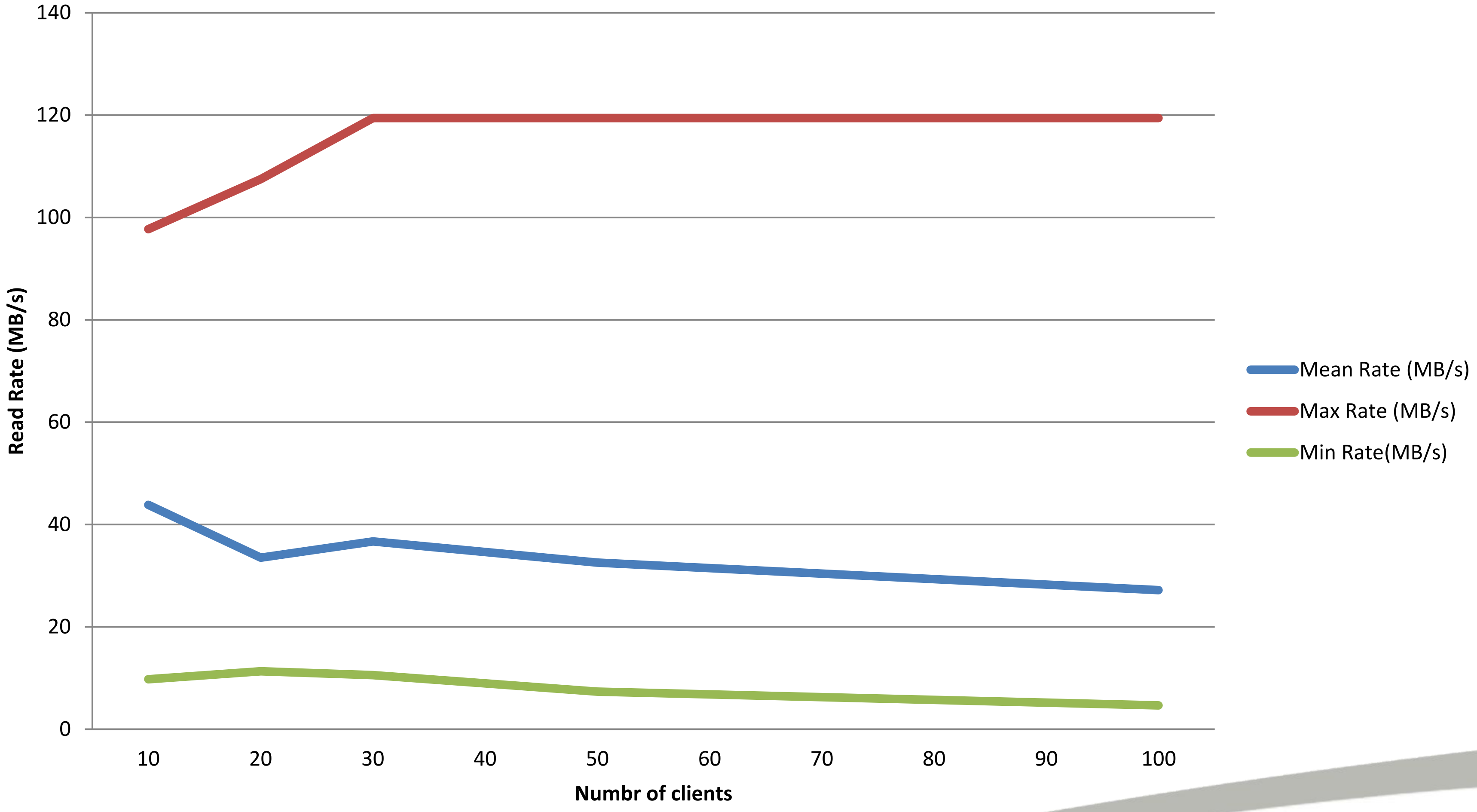
Scale out Tests

- Run on production compute cluster
- Attached to WOS via 1GB link
- Caveats of results
  - Typically multiple job run on a node (1 job/core)
  - Scheduling done via batch queue system
- Jobs did simple read/write/delete of 1GB files directly into WOS using *wget* (also tested using curl but no significant difference)
- Number of jobs up to 100

Write Rates With Differing Numbers of Clients



Read Rates with Differing Numbers of Clients

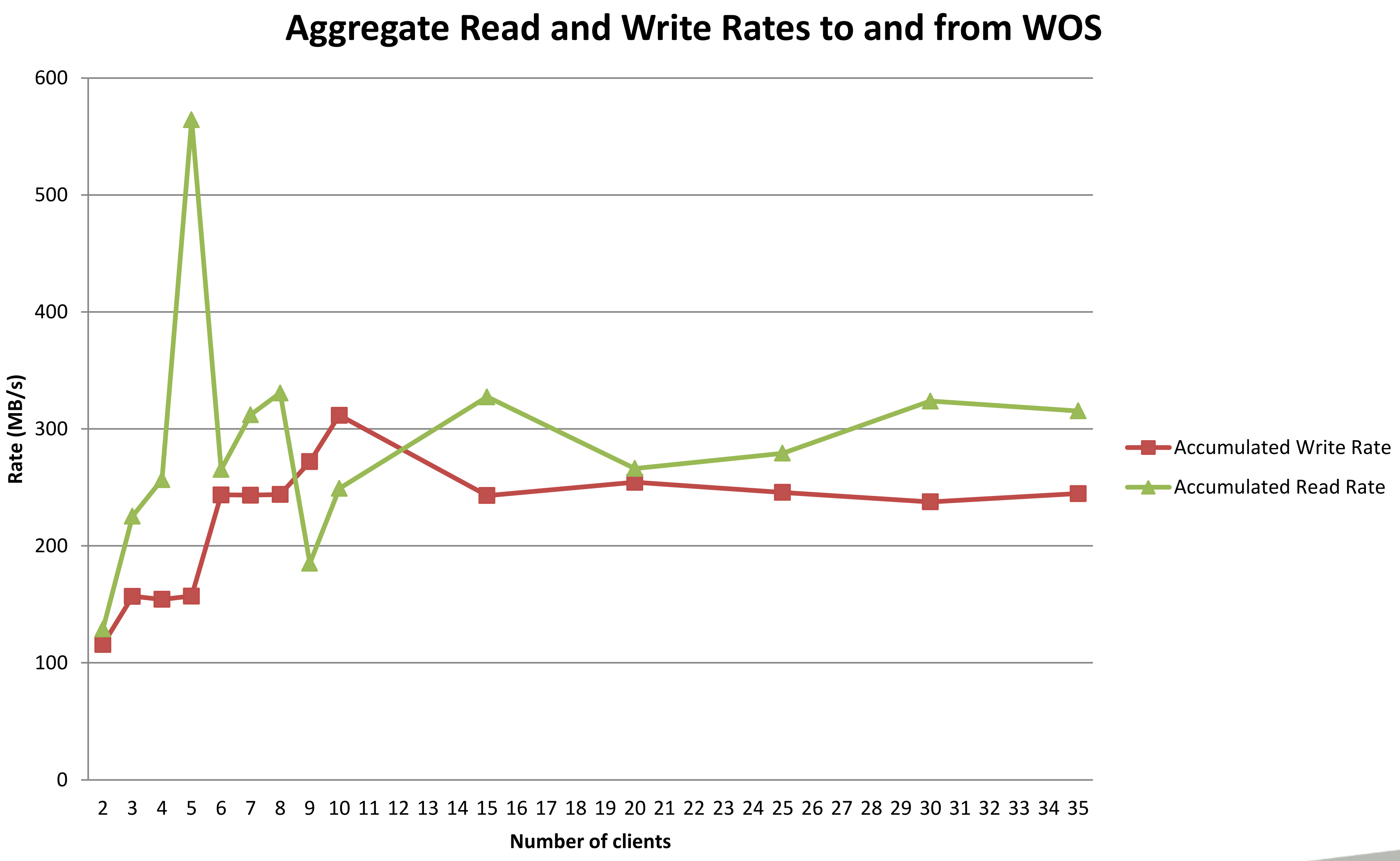


Preliminary Analysis

- Max results just show line speed into box
  - Not a big surprise!
- Mean and Min rates show
  - Pretty good read/write balance
  - Performance does drop off but at a lower rate than expected
    - -146 KB/sec per client for reads
    - -300 KB/sec per client for write
    - BUT remember in these tests not all clients may be concurrently readin/writing

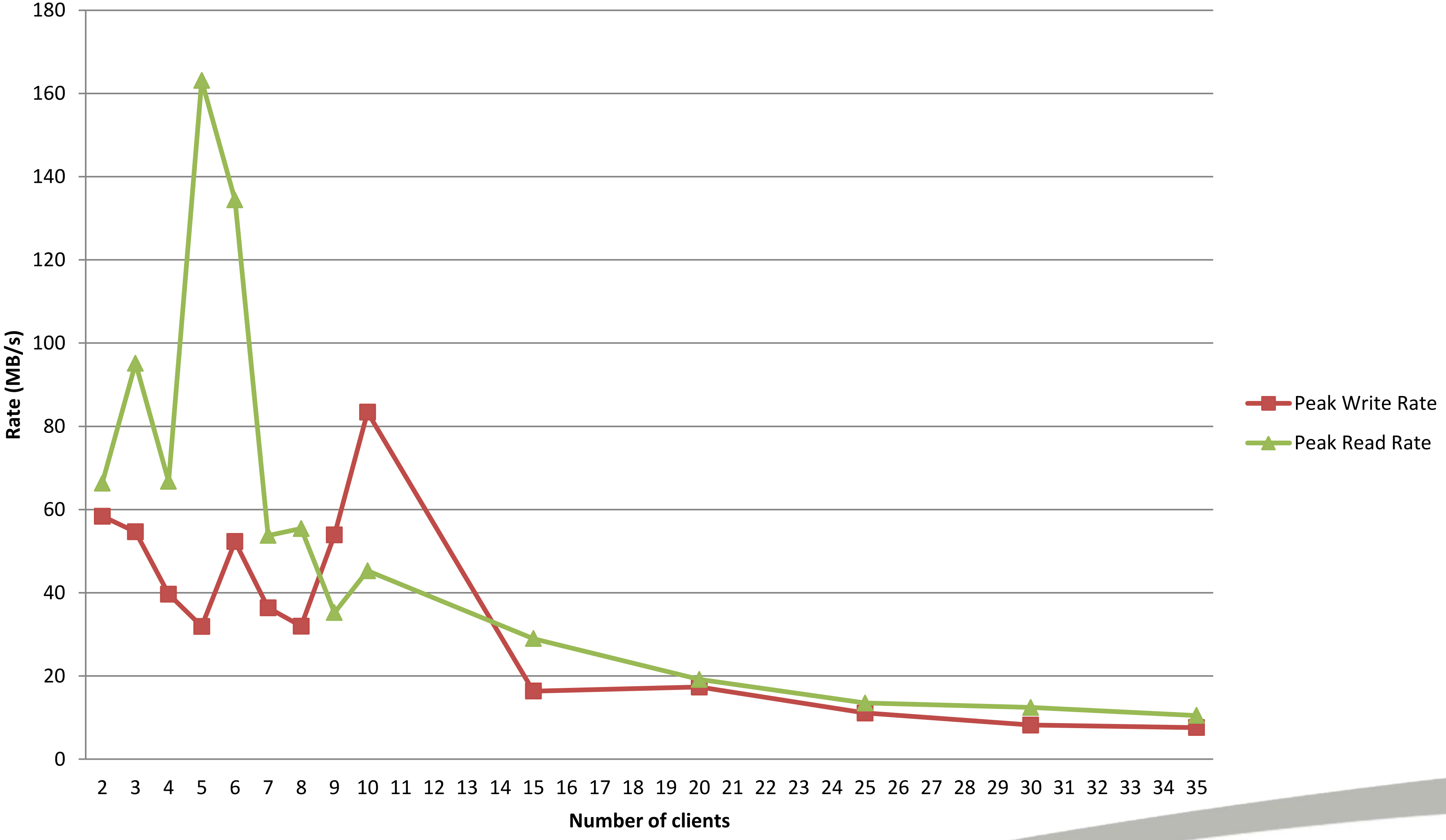
Modified Test

- Reconfigured tests so that all transfers happen concurrently
  - Initial test showed severe performance limitation
    - Connection exhaustion at 10 concurrent connects
  - But rapid response from DDN to provide firmware update



Science & Technology
Facilities Council

Peak Read and Write Rates to and from WOS



Preliminary Analysis

- Peak rates for read and write fall off fairly predictably with number of concurrent connections
- What is very god is the aggregate rate being delivered is almost constant with number of clients
  - To limit shown
- Would have been interested to show performance under combined read/write conditions
  - But is this really what WOS was designed for?

Summary

- WOS box lives up to its claims!
  - Highly scalable data ingest and delivery
  - Replication policies are easy to set up and work well
    - May be possible to set up an academic WOS network if we can find a suitable project
  - Predictable performance under ‘real life’ operation
  - Use of common language APIs and support for web based reading/writing through restful interface provides good integration potential
  - IRODS driver works out of the box
    - GPFS support not tested

Thanks

- Thanks to:
  - Georgina Ellis (OCF) for helping arrange the loan
  - Glenn Wright (DDN) for support during testing
  - Edlin Browne (DDN) for coming with this idea and helping arrange the loan
    - And paying for dinner tonight!
  - And to all of you!
- Questions?
  - I guess some of which can be answered by DDN!