Use of Lustre in a WAN environment DRAFT

LARGE DATA

Joint Capabilities Technology Demonstration

USSTRATCOM - NGA - NRL - DISA - INSCOM

James B Hofmann
Naval Research Lab
(202)-404-3132
jhofmann@cmf.nrl.navy.mil

David McMillen
System Fabric Works

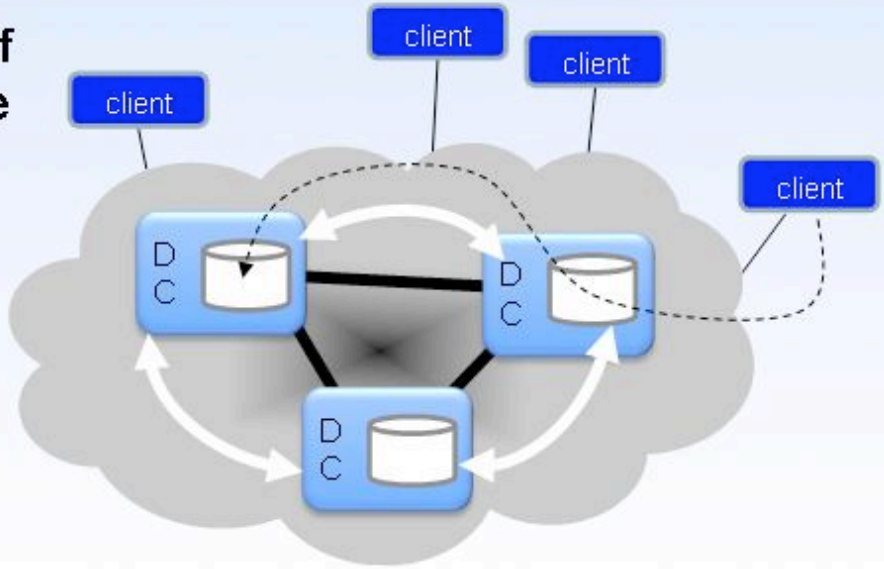
LD JCTD Concept of Operations

Large Data JCTD

Why the LD JCTD works

The LD JCTD demonstrated the use of RDMA and a clustered Global File System over long distances to create a globally accessible storage and compute cloud

- **Data available to clients anywhere in the world**
- **Remote clients may disconnect at will**
- **Centralized apps available to clients**



RDMA interconnect

+

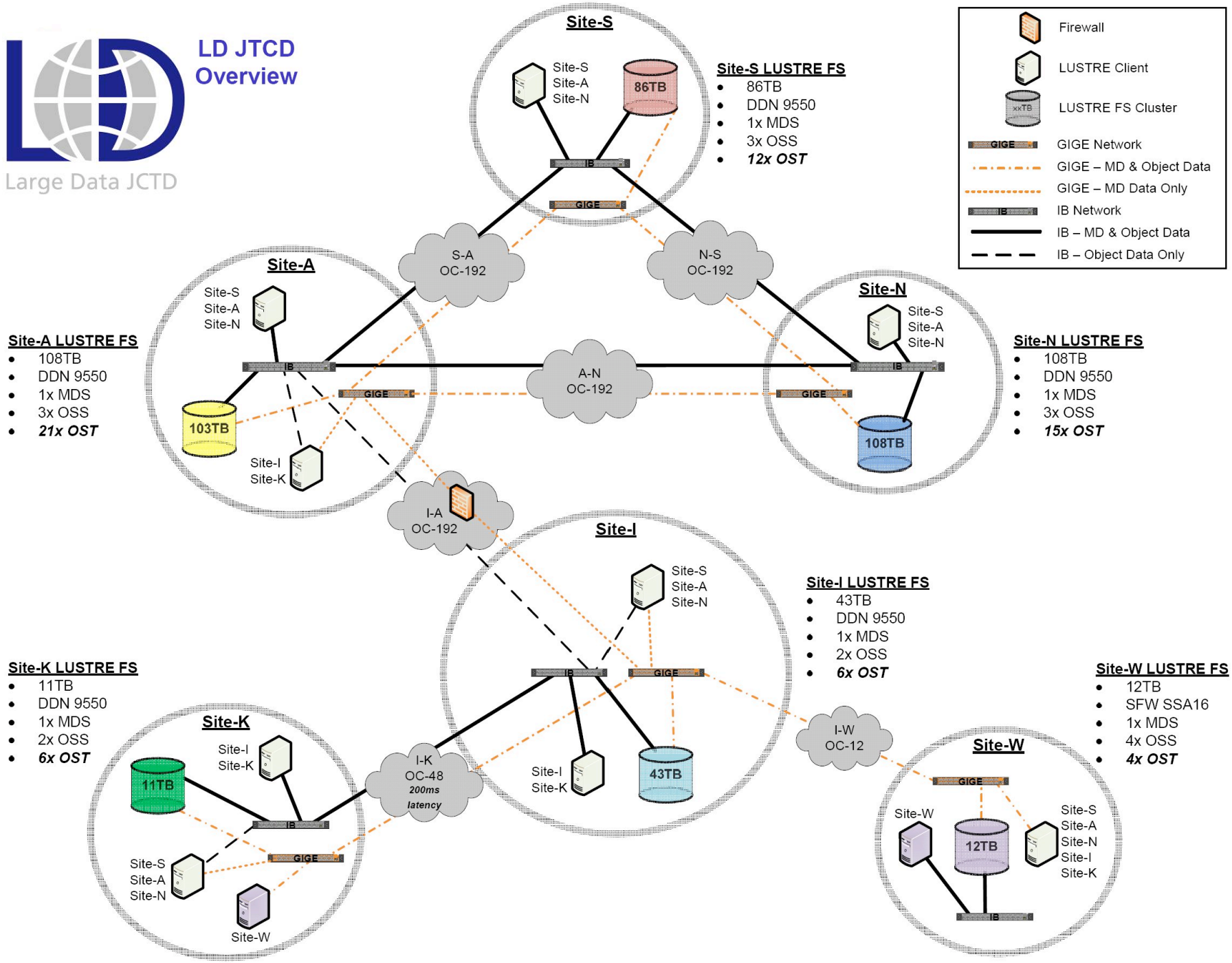
Global File System

1. ***LD JCTD used RDMA and parallel filesystems to build five scalable, cost effective data centers (DCs).***
2. ***LD JCTD extended RDMA over a high b/w WAN to virtualize the data centers.***
3. ***Clients connect to the virtual data center via cost effective, low bandwidth (1 Gb/s) IP networks (SIPRNet, JWICS)***

LD JCTD used existing, standards-based COTS technology and components to demonstrate a cloud computing infrastructure, w/ zero impact on the underlying DISN

LD JTCD Overview

Large Data JTCD



Lustre LD JCTD Summary

- Status
  - Lustre is running well and the principal file system
    - Some file systems have filled up
    - Now able to find problems (3-4 at present) that are due to heavier usage
  - Within CONUS, Lustre performance is excellent
  - Intercontinental speeds are faster than ftp
    - With system tuning, we have achieved very good write performance with Lustre US to USFK via OC-48 IB/Sonet
  - Using Lustre on regular basis in unclass and classified
    - Updating Lustre to 1.6.7 now on all systems
    - Most of the applications use Lustre via web apps
  - More work needed on documentation, performance analysis, and tuning

Large Data JCTD

Unclassified Testbed

SLES10SP2/OFED 1.4/Lustre 1.6.7+

options ko2iblnd peer_credits=128 credits=256 concurrent_sends=256 ntx=512 fmr_pool_size=512
fmr_flush_trigger=384 fmr_cache=1

OBSIDIAN STRATEGICS

0 - 210ms delay

Infiniband @ 10Gbps (SDR)

2 x Ethernet @ 1Gbps

SiteX

SiteY

SiteZ

-XE-H

-XE-0

-XE-1

-XE-2

-XE-3

-XE-4

-XE-5

-XE-6

1 x MDS/
1.8T MDT

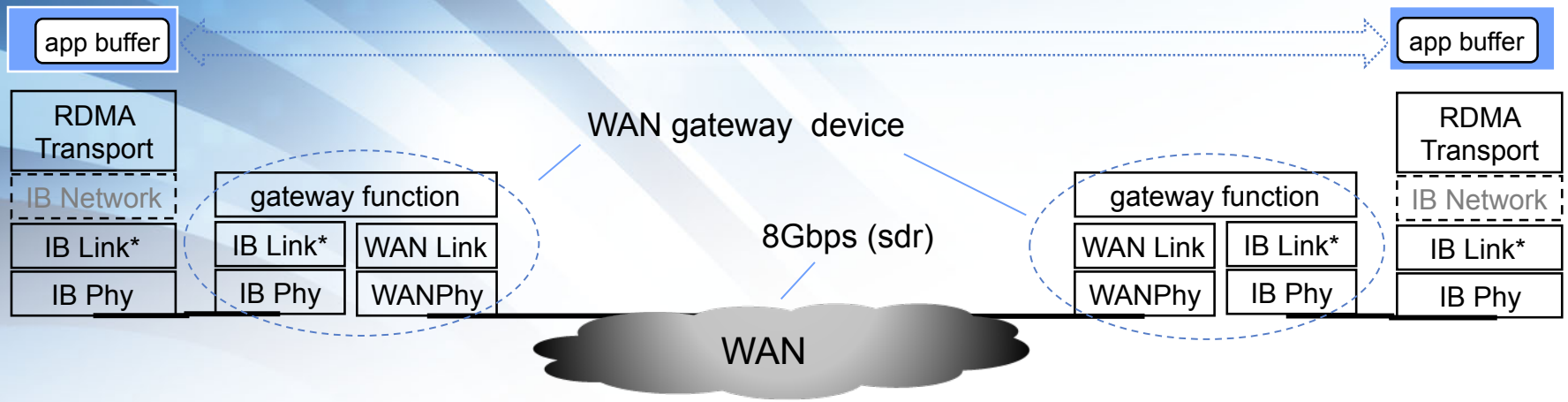
4 x OSS/
7.2T OST

-WS-1

-WS-2

LD WAN Gateway

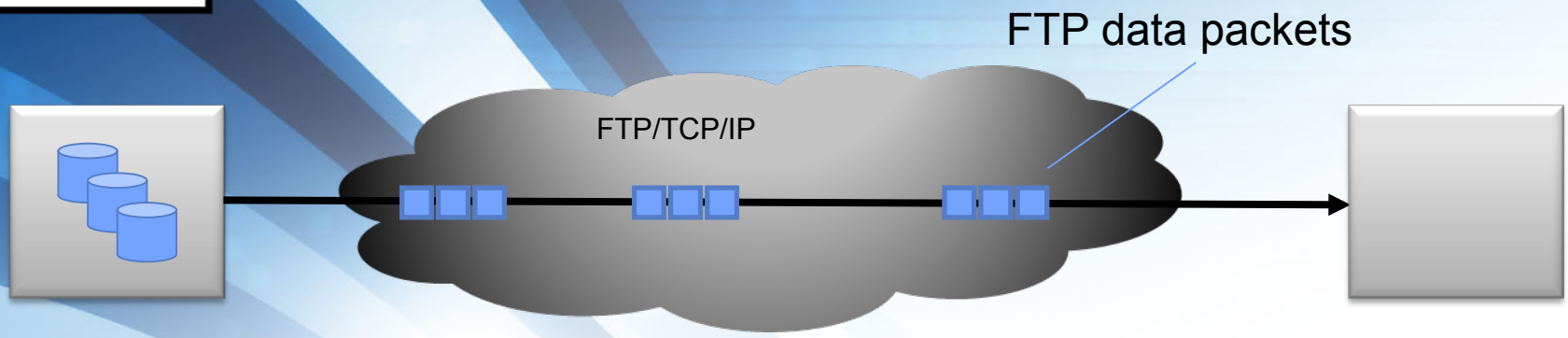
- WAN Gateway extends the qualities of the local, flow-controlled InfiniBand network across a long haul WAN
- Buffer-to-buffer data transfer over the WAN



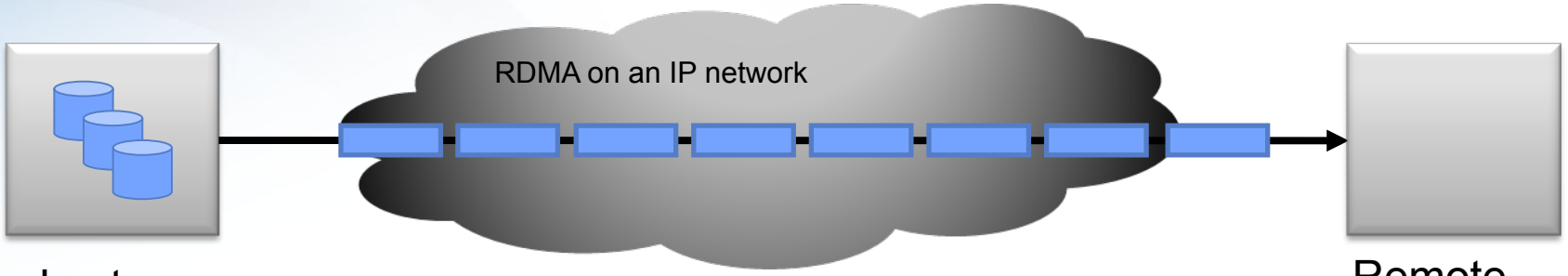
- IB link layer flow control is extended with WAN flow control
- Current implementation is a single subnet, no e-2-e network layer is used
- Gateway device inserts an IP header for gateway ←→ gateway routing purposes

  - NRL-developed concept commercialized by Obsidian, Bay Micro
  - RDMA/IB/WAN supports high performance Lustre over the WAN
  - Lossless delivery

An efficient WAN

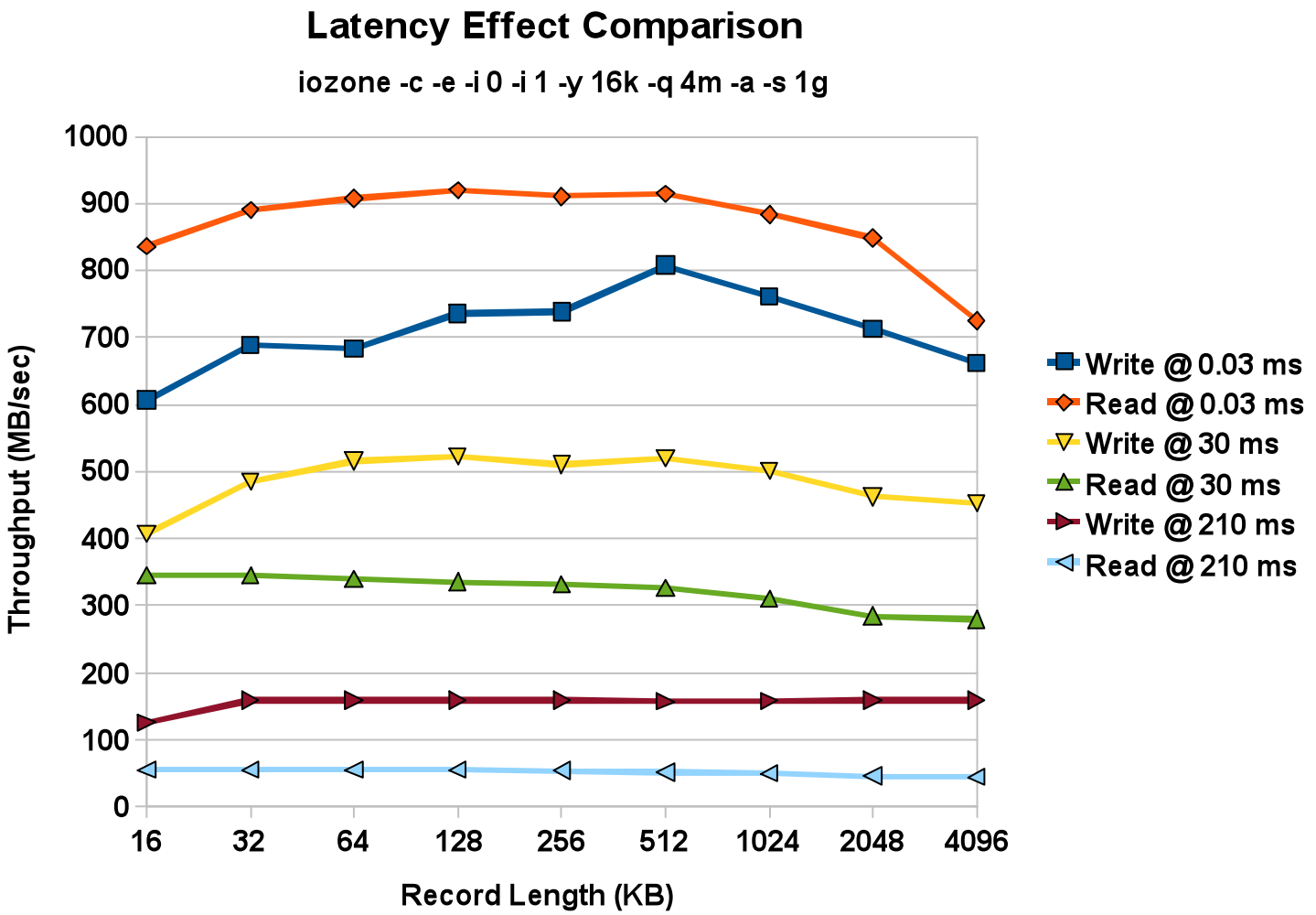


FTP/TCP - TCP is a ‘windowing protocol’.
The ‘window size’ severely limits the number of packets in flight and reduces network efficiency, especially over long distance.

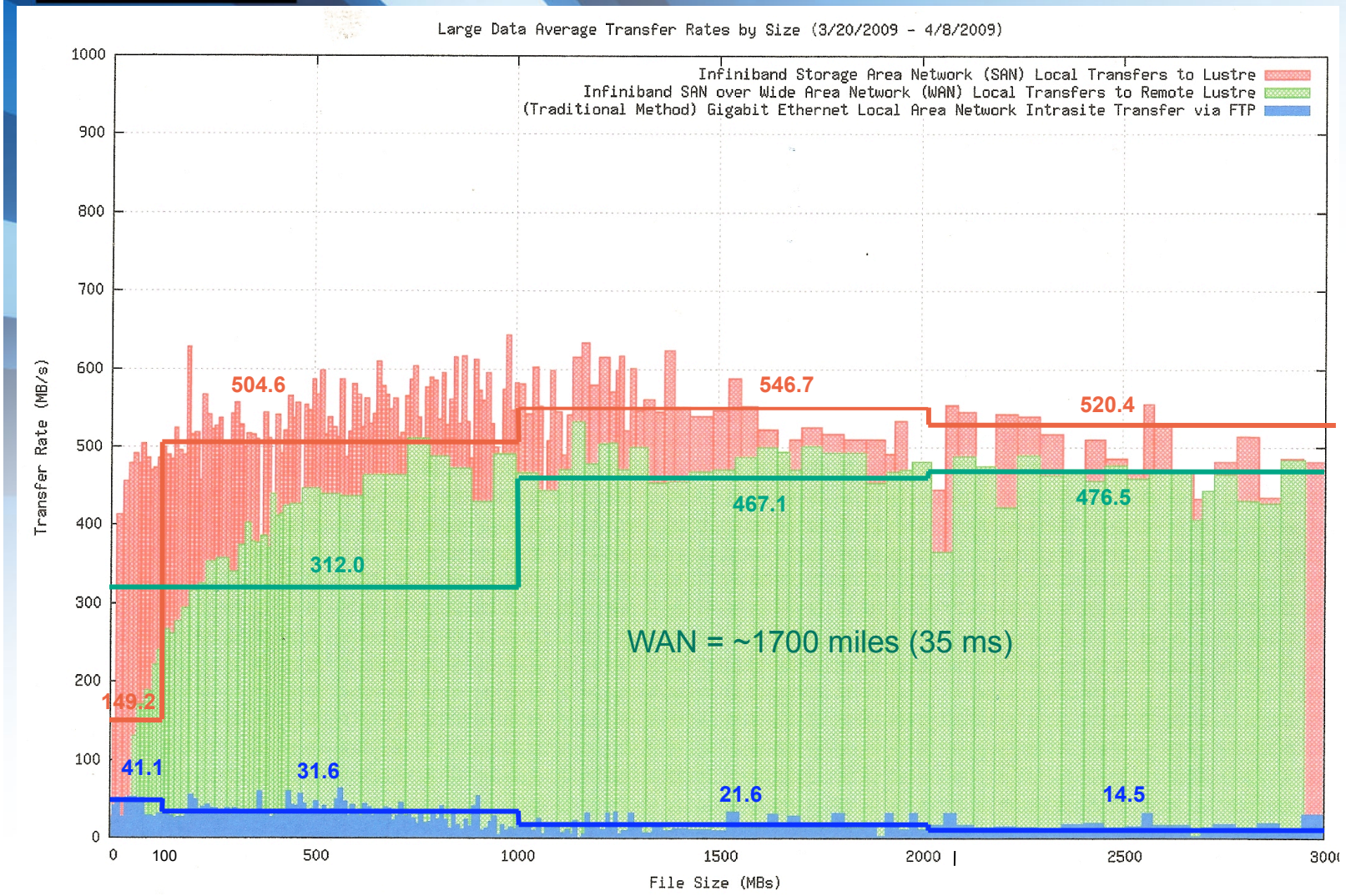


RDMA protocol is capable of keeping the pipe continuously full.

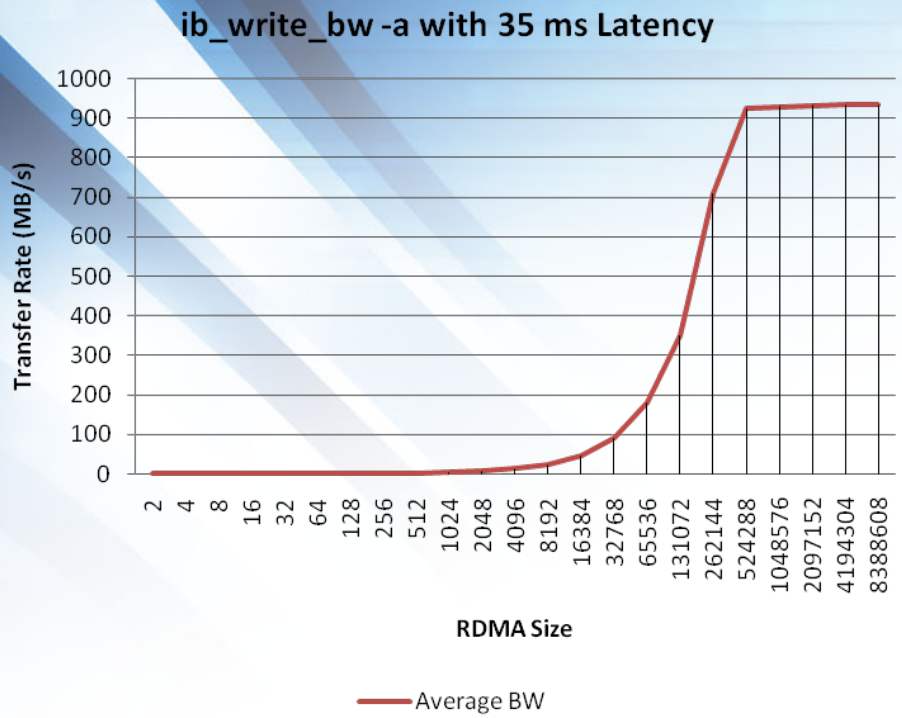
Unclassified Testbed



LD Average Data Transfer Rates

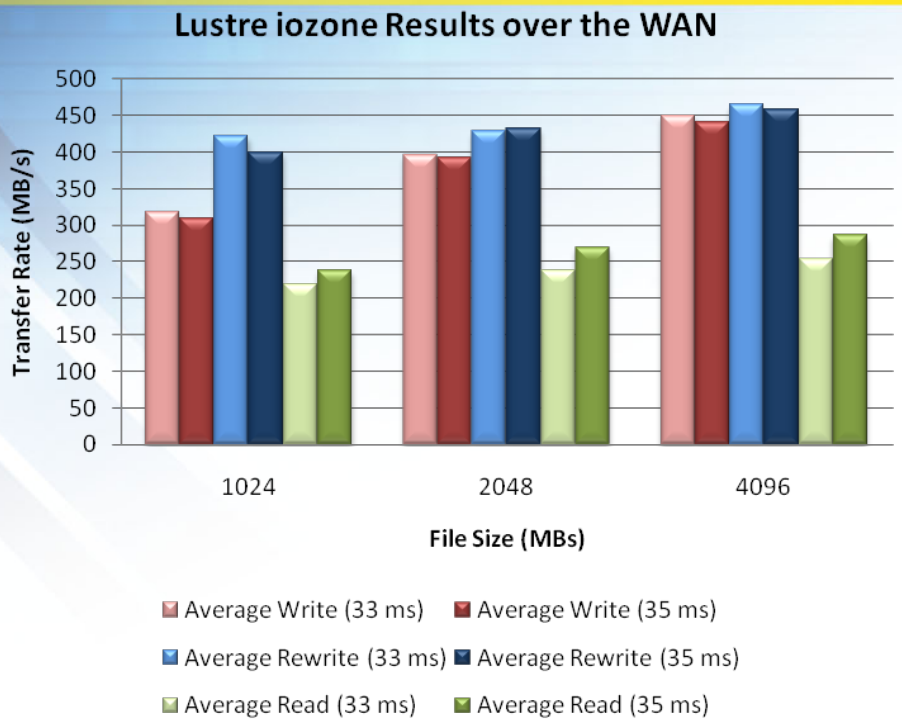


10 Gbps WAN Performance



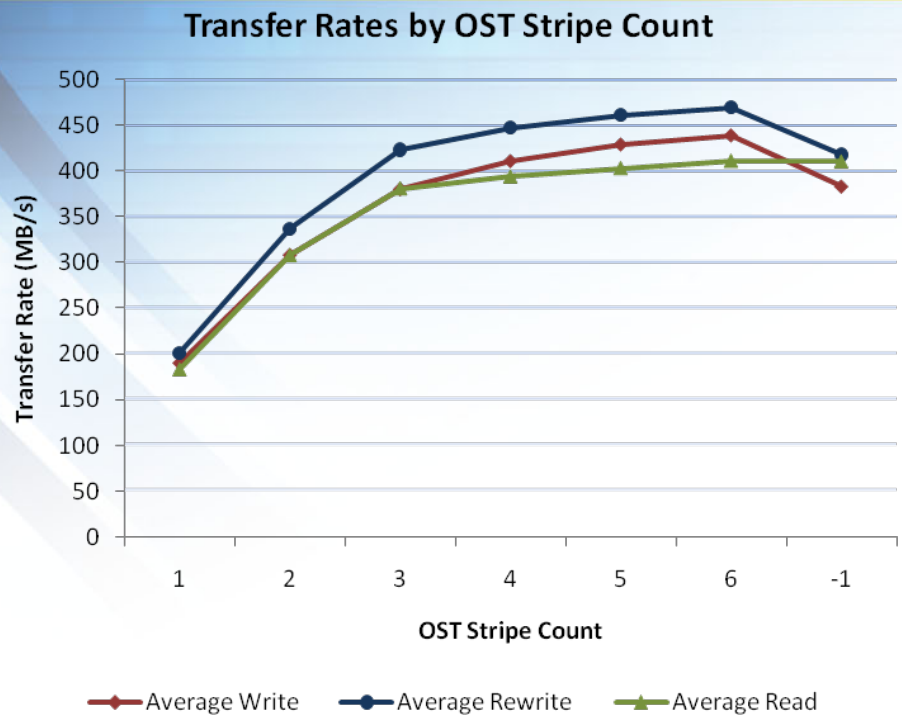
- OC-192 (~10 Gbps – where 8 Gbps or 1024 MBps is IB traffic)
- "Ib_write_bw –a" average bandwidth at 35 ms latency
- Gives best case performance measurement over WAN

10 G WAN Lustre Performance



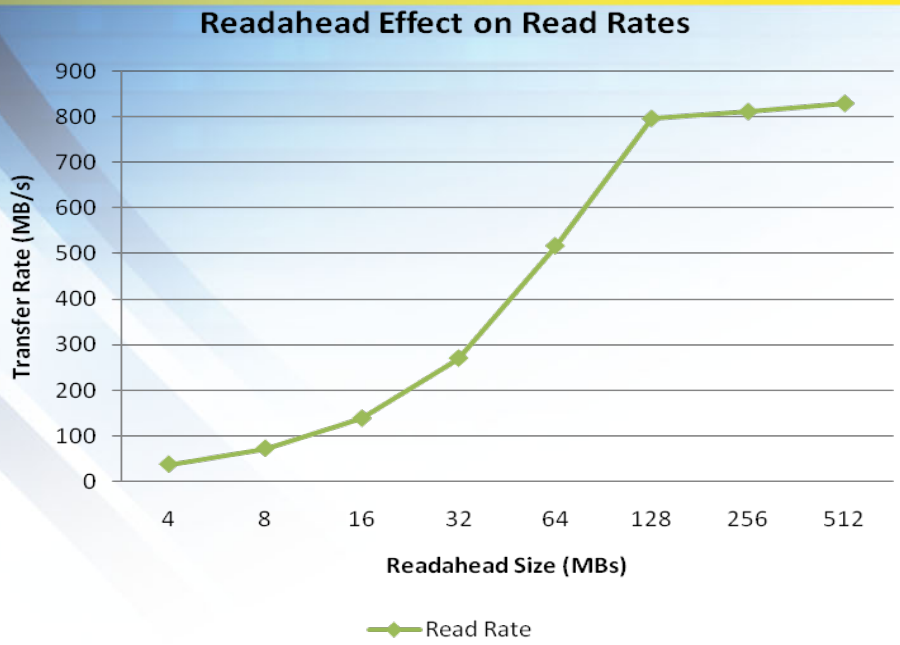
- OC-192 (~10 Gbps – 8 Gbps IB)
- Two different paths to the same Lustre file system
  - 33 ms over Net NX5010
  - 35 ms over Obsidian Longbow XR
- Directory striping: stripe_size=262144, stripe_count=-1, and stripe_offset=-1
  - max_rpcs_in_flight=8
- 12 OSTs at the remote side, 3 OSS

10 G WAN Lustre Performance



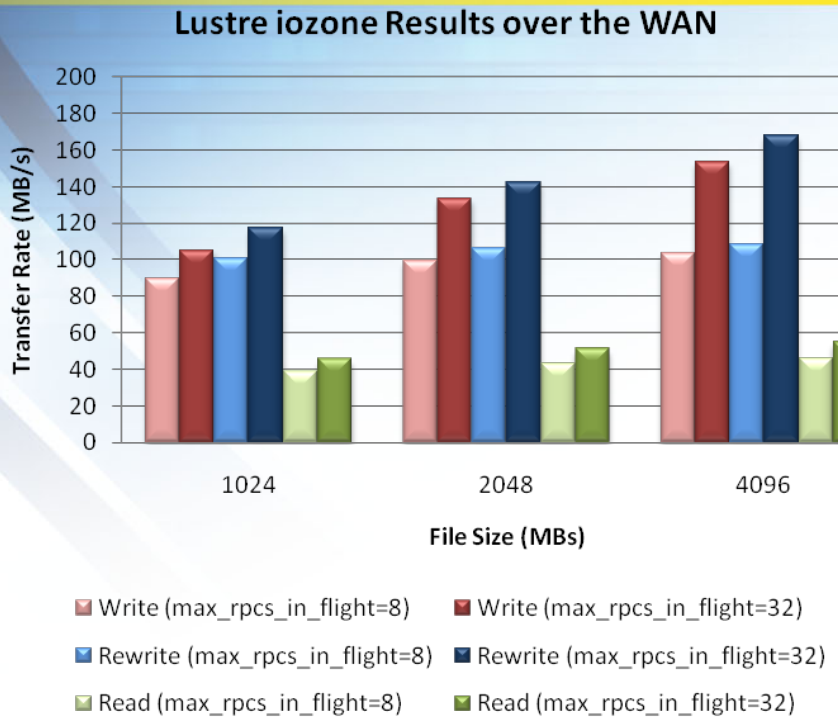
- Results are from several iozone tests run in a single day
- Stripe size was varied from 128 KB to 4 MB
- max_read_ahead_mb parameter was bumped up to 64 MB
- Maximum of 12 OSTs (1 case)

10 G WAN Lustre Performance



- Data is from an application which reads raw pixel data and writes to display
  - Data is read from the remote file system (35 ms latency)
  - Uses readv, posix_fadvise (sequential), and posix_memalign() to read optimal size blocks into an optimally aligned buffer
- File had stripe_size=1 MB, stripe_count=6,
- max_rpcs_in_flight=24
  - Had to be tuned to for the larger max_read_ahead_mb sizes to benefit

2.5 Gbps WAN Performance



- OC-48 (~2.5 Gbps) with application's CBR limiting IB bandwidth to 200 MB/s
- 203 ms latency over Net NX5010
- Directory striping: stripe_size=262144, stripe_count=-1, and stripe_offset=-1
    - max_rpcs_in_flight=8 and max_rpcs_in_flight=32
- 6 OSTs at the remote side, 2 OSS

Desired Capabilities

- Windows native client -- pCIFS is painful and still slow
- Replication/Failover with Multi-Master Distributed MDS
  - Lustre OST Pools is expected to be a start. We need policy-based notions of locality. Without this kind of feature is Lustre useful in a COOP (redundancy/failover) scenario?
    - Possible for each site with a separate lustre filesystem and some sort of rsync/merge when the filesystems could see each other.
- Metadata speedups for Cross WAN traffic
- MapReduce – is this a square peg in a round hole?
- Mac OSX client – is anyone else interested in this for workstations and servers?

NRL Recent Contributions to Lustre

- Identified and assisted in testing Infiniband/RDMA client over long haul networks (Bugzilla 14358)
  - More work is required but performance is much improved
- Identified bug in strideahead/readahead code that is only visible on current windows PCIFS client.
  - Reportedly fixed and then reappears in last release
- Kerberos collaboration.