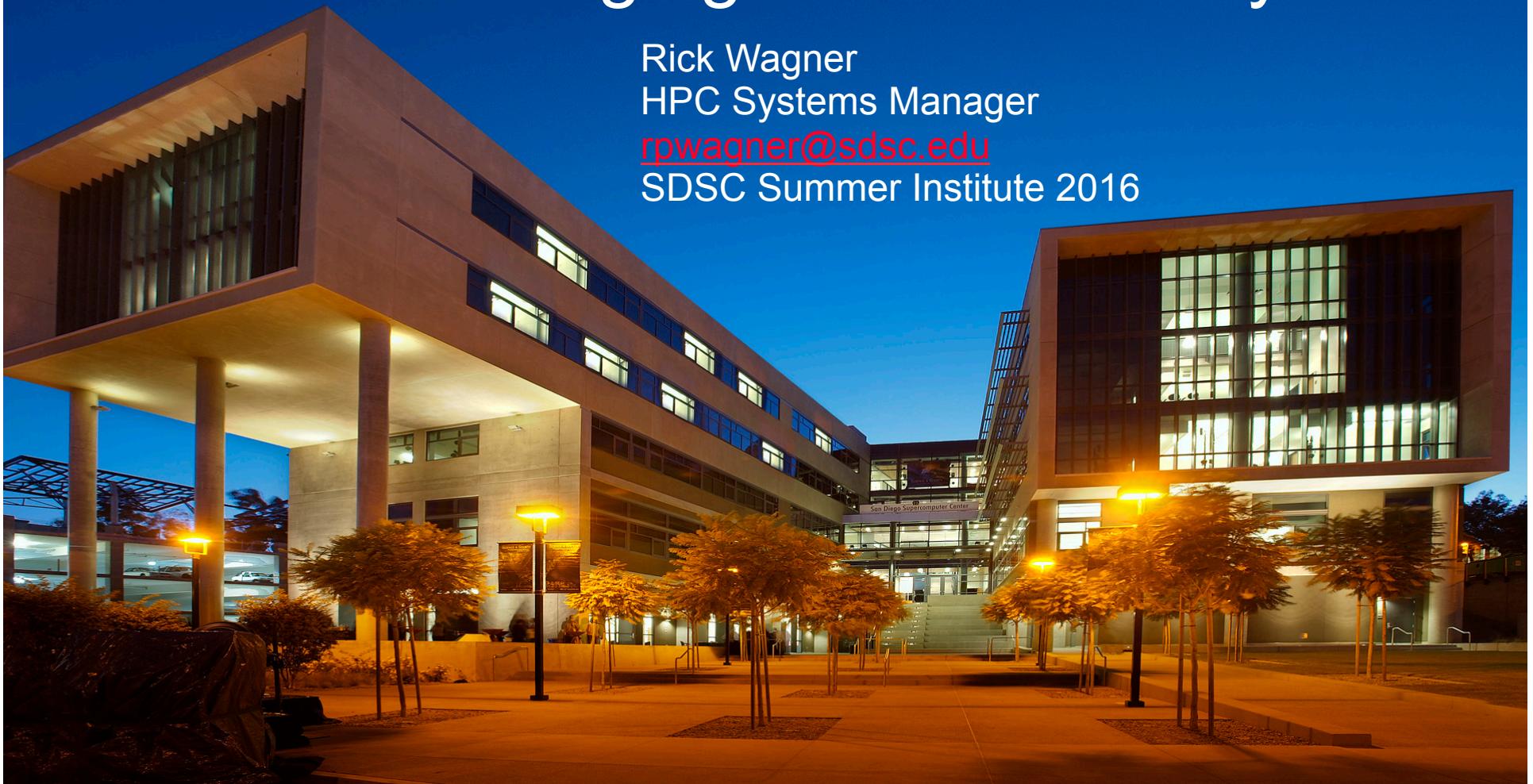


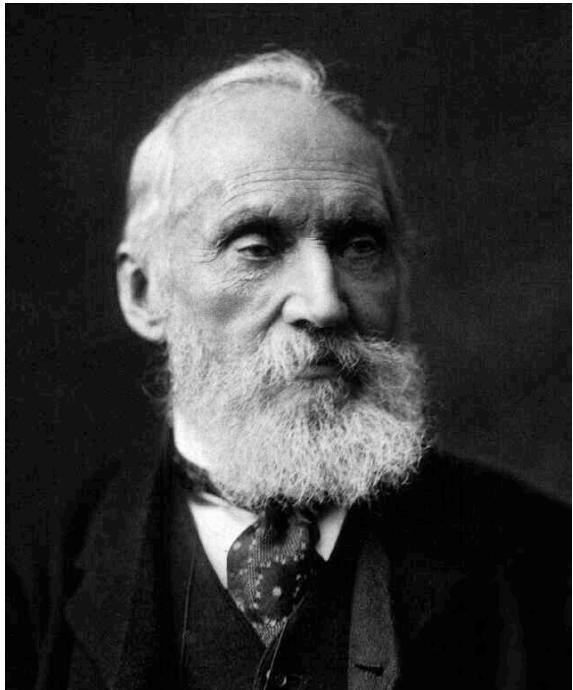
Managing Data on File Systems

Rick Wagner
HPC Systems Manager

rpwagner@sdsc.edu

SDSC Summer Institute 2016





This image has been identified as being free of known restrictions under copyright law, including all related and neighboring rights.

"I have no satisfaction in formulas unless I feel their numerical magnitude."

Lord Kelvin

Pretty relevant. Lord Kelvin helped define the concept of bandwidth.

Thomson, W. (1854) "On the theory of the electric telegraph" *Math. and Phys. Papers*, 2, p.61

Example I: an IOP

How long does it take to get or put a small amount of data?

- **What is “small”?**
~4k
- **From where?**
 - Case I: Network storage
 - Case II: Local SSD
 - Case III: RAM

Dominant term:
Latency (Δt)

Case I:
 $\Delta t = 5 \text{ ms}$

Case II:
 $\Delta t = 100 \text{ us}$

Case III:
 $\Delta t = 50 \text{ ns}$

Example II: Moving a GB

How fast can I move a GB?

- **From where?**
 - Case I: Network storage
 - Case II: Local SSD
 - Case III: RAM
- **Dominant term:**
Bandwidth
 $\Delta d/s$ or $GB/\Delta t$

Case I:

$$\Delta d = 1\text{-}5 \text{ GB}$$

$$\Delta t = 200 \text{ ms} - 1 \text{ s}$$

Case II:

$$\Delta d = 500 \text{ MB}$$

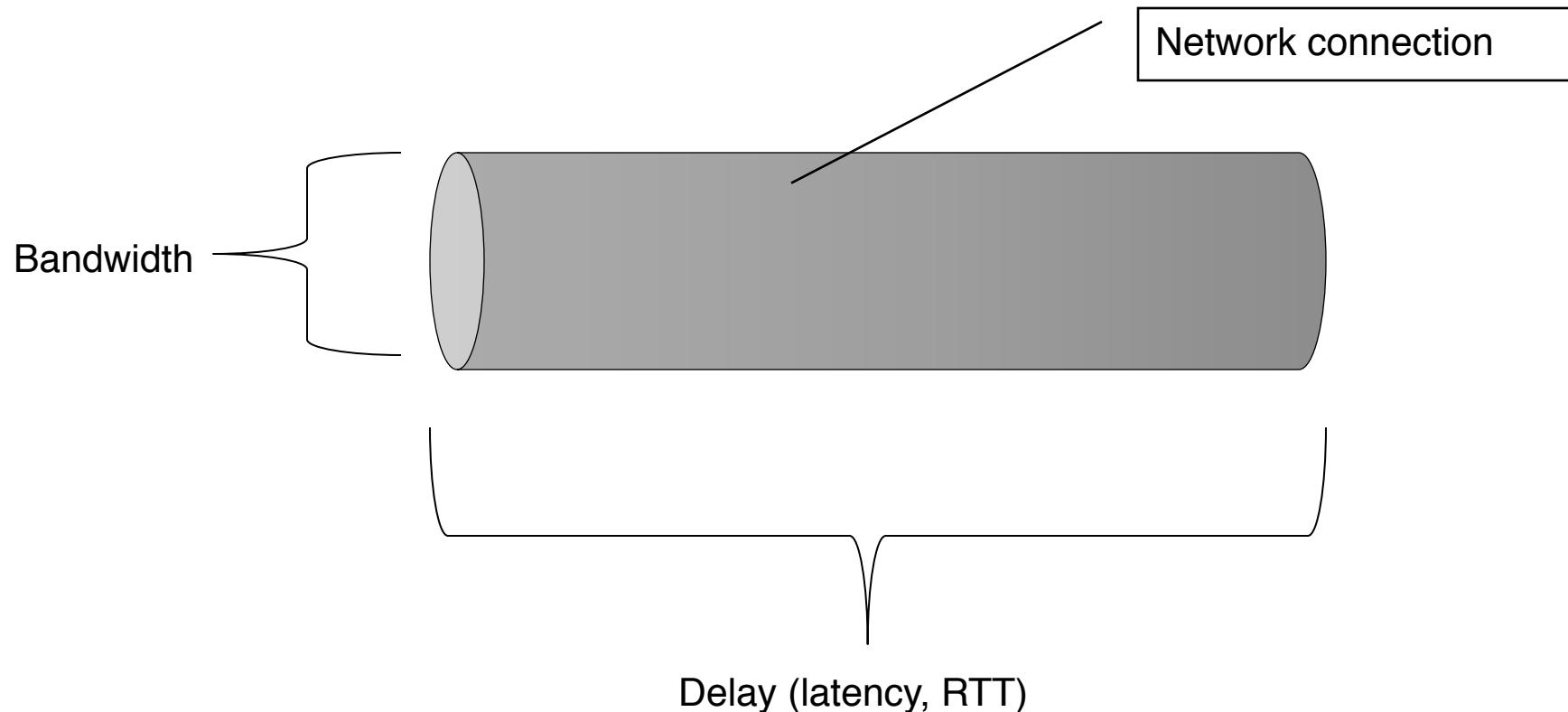
$$\Delta t = 2 \text{ s}$$

Case III:

$$\Delta d = 50 \text{ GB}$$

$$\Delta t = 20 \text{ ms}$$

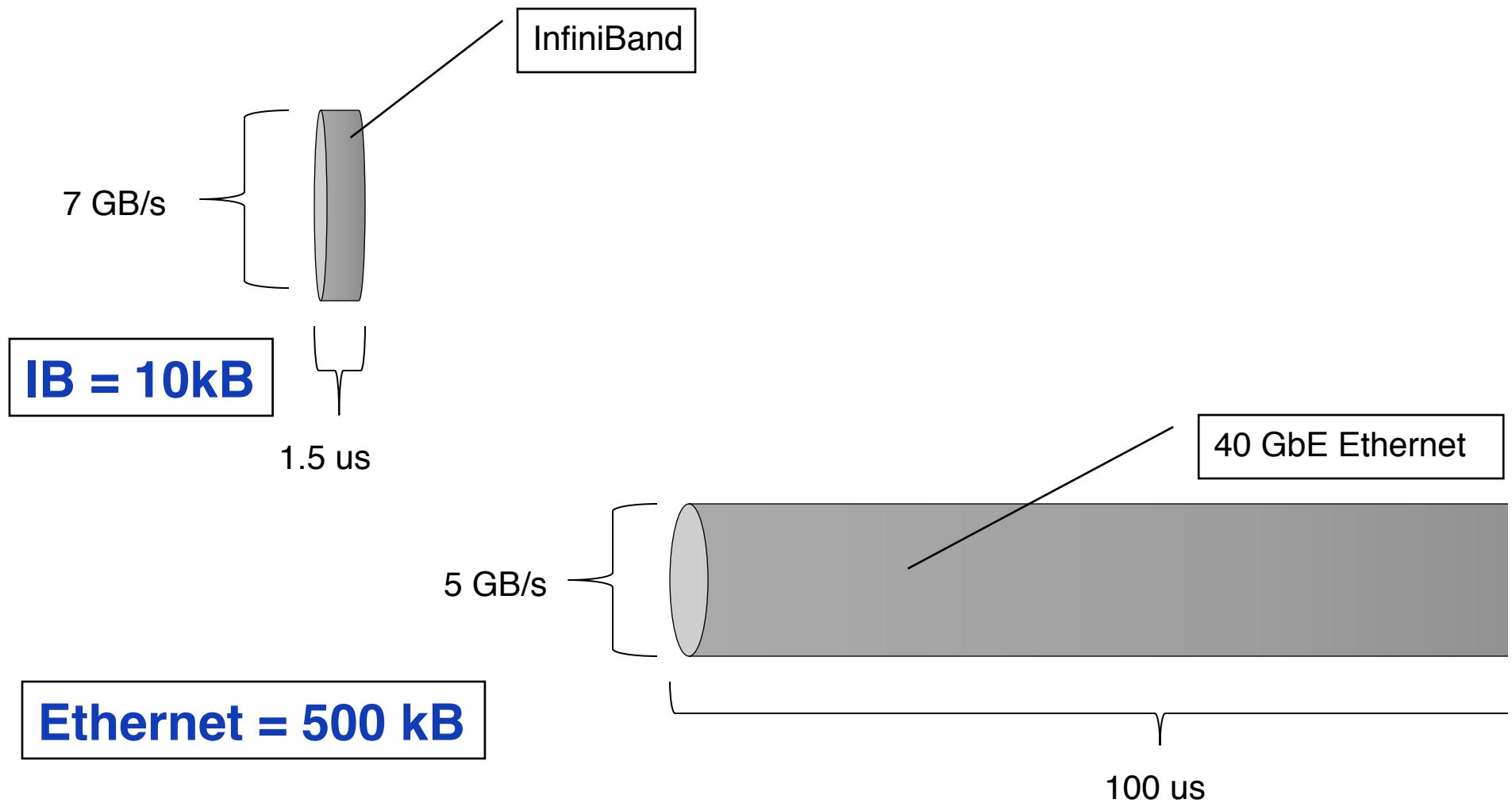
Fancy Term: Bandwidth Delay Product



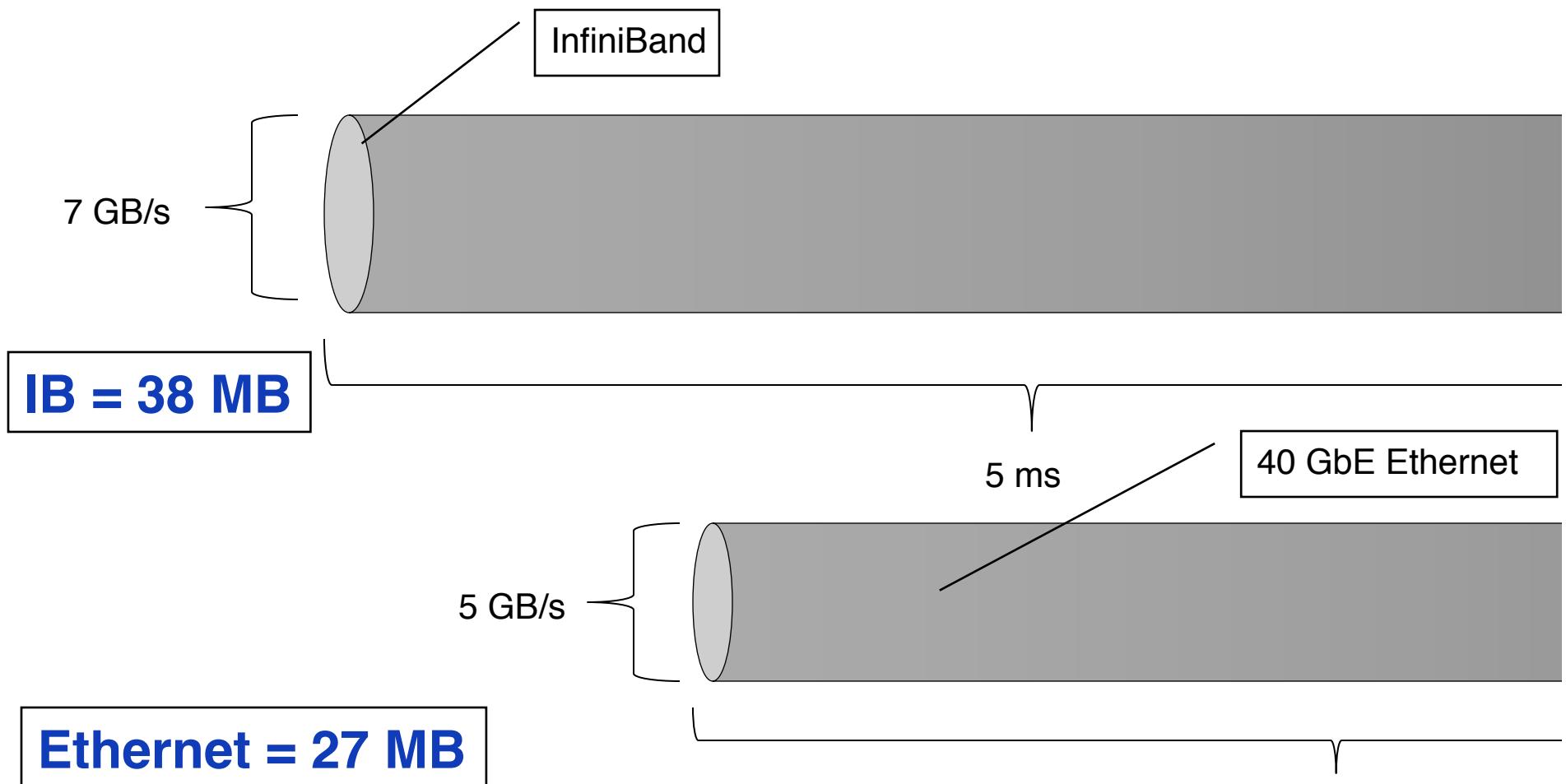
BDP = Bandwidth x RTT

How much data fits in the pipe.

Fancy Term: Bandwidth Delay Product



BDP w/Drive Access



The Lesson

If you ask for data
over the network,
ask for a lot!

Corollary

If you don't need a
lot of data, don't
use the network!



Dealing with Data: Choosing a Good Storage Technology for Your Application



Application Focus

Storage choices should be driven by application need, not just what's available.

But, applications need to adapt as they scale.

Writing a few small files to an NFS server is fine... writing 1000's simultaneously will wipe out the server.

If you use binary files, don't invent your own format. Consider HDF5.

Storage Technologies

File Systems

ext4

ext4

NFS

Lustre

GPFS

FUSE

Devices

memory

block

Services

HTTP(S)

MySQL

CouchDB

memcached

Storage Technologies

File Systems

ext4

ext4

NFS

Lustre

GPFS

FUSE

Devices

memory
block

**Each has its own
performance characteristics**

**Not all are available
everywhere**

Services

HTTP(S)
MySQL

CouchDB
memcached

File Systems

Classic access, POSIX, Windows

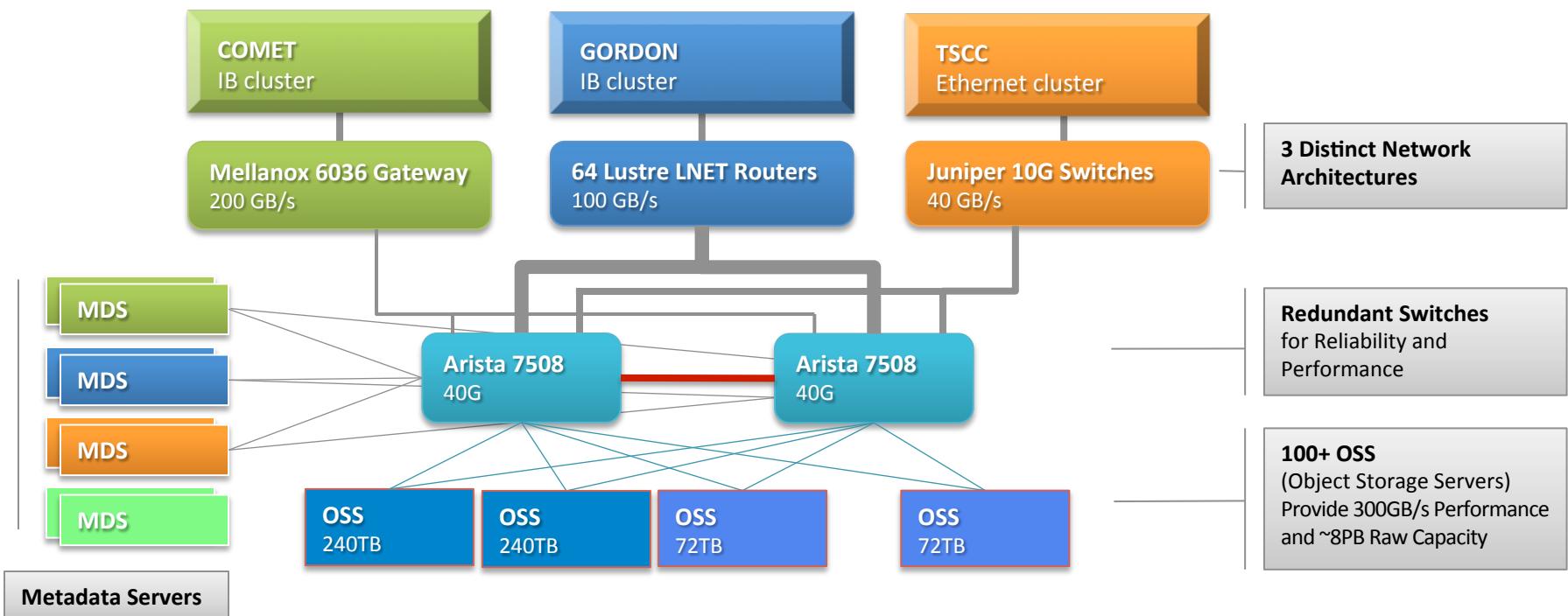
Most relevant:

- Local
- Remote
 - NFS, CIFS
 - Parallel (Lustre, GPFS)

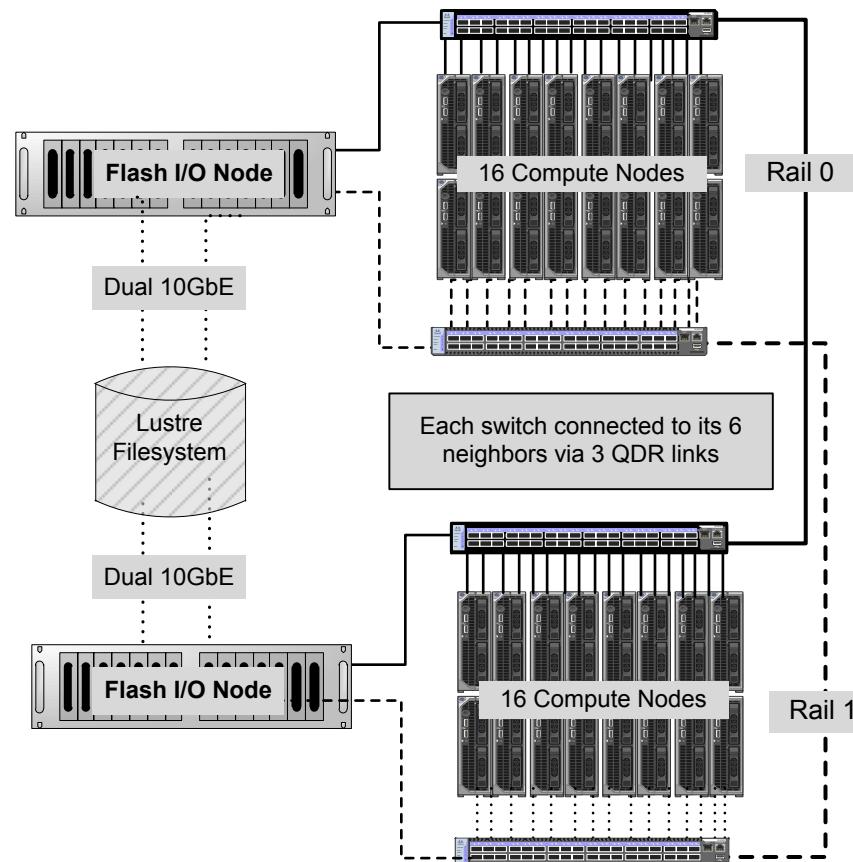
Local file systems are good for small and temporary files
(low latency, modest bandwidth)

Network file systems very convenient for sharing data
between systems
(high latency, high bandwidth)

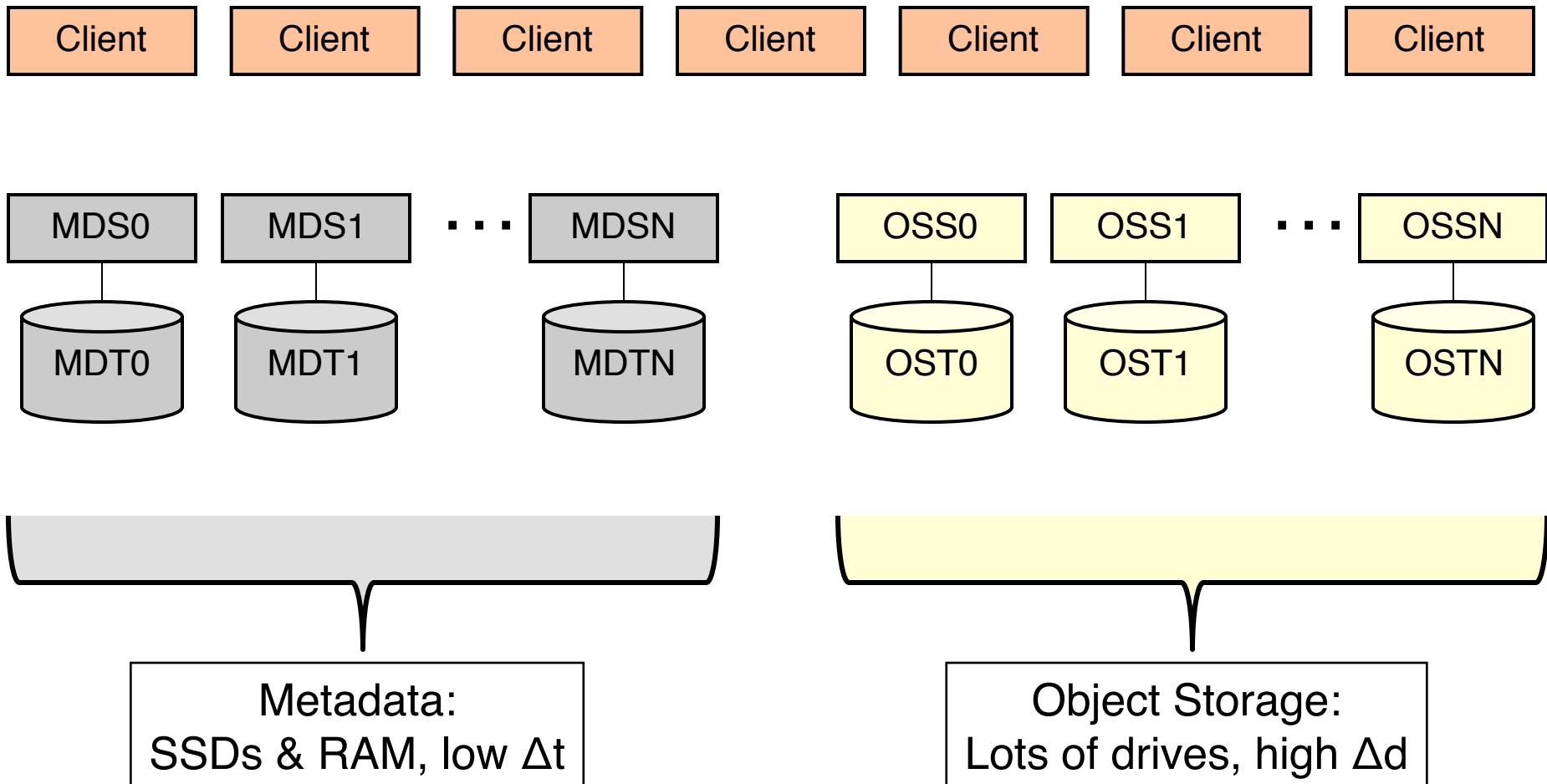
Parallel File Systems



Parallel File Systems



Parallel File Systems



A Cautionary Tale

<http://www.youtube.com/watch?v=gDfLXAtRJfY&feature=youtu.be>

Devices

Raw block device (`/dev/sdb`) or RAM FS (`/dev/shm`)

Useful in specific cases, like fast scratch

Can be very good for small I/O

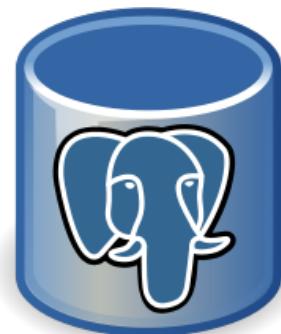
Services



Things accessed programmatically



Frequents the last thought for HPC applications: A MISTAKE



Databases
Cloud storage (Amazon S3)
Document storage (MongoDB, CouchDB)



Know What You Need

<http://www.youtube.com/watch?v=F4OIDszDA9E>

Order of Magnitude Guide

Storage	file/directory	file sizes	BW	IOPs
Local HDD	1000s	GB	100 MB/s	100
Local SSD	1000s	GB	500 MB/s	10000
RAM FS	10000s	GB	GB/s	10000
NFS	100s	GB	100 MB/s	100
Lustre/GPFS	100s	TB	100 GB/s	1000
Cloud	Infinite	TB	10 GB/s	0
DB	N/A	N/A	N/A	10000

Choosing

My application needs to:

Write a checkpoint dump from memory from a large parallel simulation.

I should consider:

A parallel file system and a binary file format like HDF5.

Choosing

My application needs to:

Run analysis on remote systems and return the results to a web portal for users.

I should consider:

Cloud storage for results and input, and local scratch space for the job.

Choosing

My application needs to:

Randomly access many small files, or read and write small blocks from large files.

I should consider:

A database, RAM FS, or local scratch space.

Homework: Many Boxes Make a Sad Panda

<http://www.youtube.com/watch?v=N2zK3sAtr-4>

Database logos courtesy of [RRZEicons](#)
<http://commons.wikimedia.org/>



Another Topic: Moving Data

Example from an industrial project with Janssen Pharmaceuticals on large-scale genomic analysis. See Glenn Lockwood's webinar:

http://www.sdsc.edu/Events/ipp_webinars/index.html

http://www.sdsc.edu/Events/ipp_webinars/large_scale_genomics.pdf



Step #3: Loading Data

- **Input: raw reads from 438 full human genomes**
 - 50 TB of compressed, encrypted data from Janssen
 - 4,230 files (paired-end .fastq.gz)

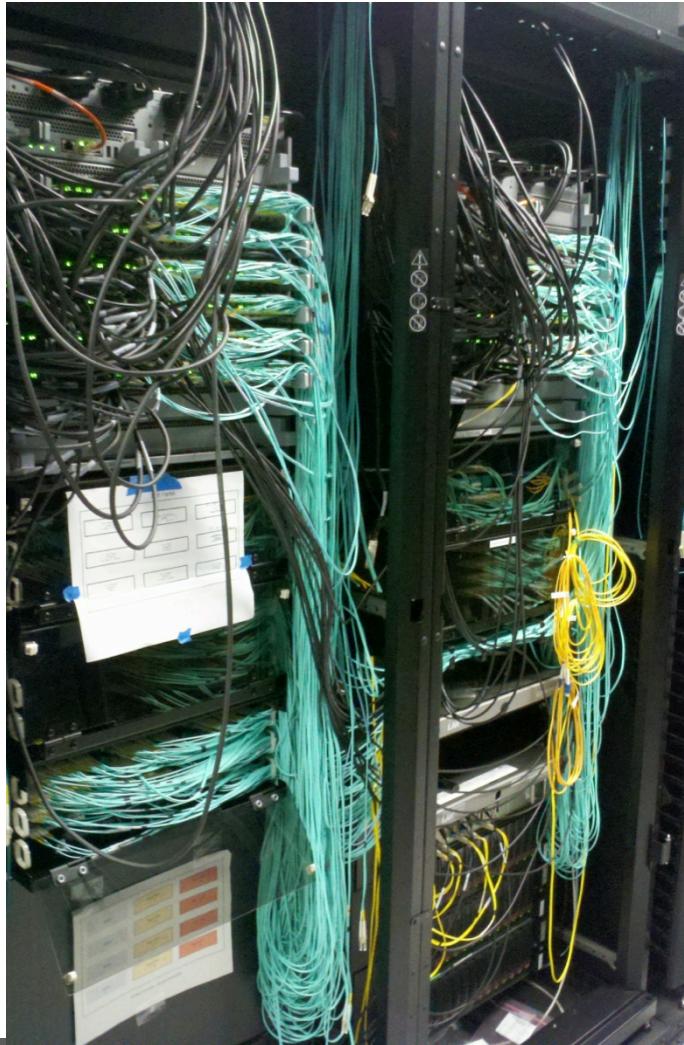


How do you get this data into a supercomputer?

They don't exactly have USB ports

Pictured: 2 of 12 Data Oasis racks; each blue light is 2x4 TB disks

Loading Data via Network



External

- 60 Gbit/s connectivity to outside world
- 100 Gbit/s connection installed and being tested

Internal

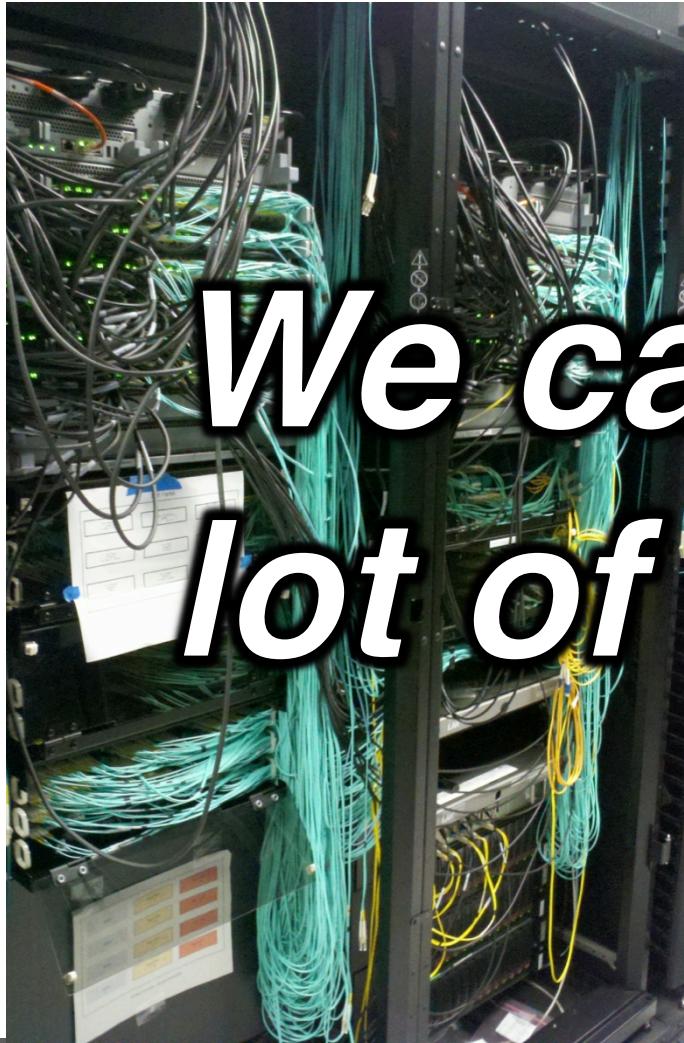
- 60 Tbit/s switching capacity
- 100 Gbit/s from Data Oasis to edge coming in summer

Gordon

- 20 Gbit/s from IO nodes to Data Oasis
- 40 Gbit/s dedicated storage fabric (IB)

Pictured: 2x Arista 7508 switches, core of the HPC network

Loading Data via Network



**We can move a
lot of data**

External

- 60 Gbit/s connectivity to outside world
- 100 Gbit/s connection installed and being tested

nat
Tbit/s switching capacity

- 100 Gbit/s from Data Oasis to edge coming in
- 20 Gbit/s from IO nodes to Data Oasis
- 40 Gbit/s dedicated storage fabric (IB)

Pictured: 2x Arista 7508 switches, core of the HPC network

Loading Data via "Other Means"

- Most labs and offices are not wired like SDSC
 - Janssen's 50 TB of data behind a DS3 link (45 Mbit/sec)
 - Highest bandwidth approach...



*Left: 18 x 4 TB HDDs from Janssen
Right: BMW 328i, capable of > 1 Tbit/sec*

Step #3: Loading Data

- **Use the network if possible:**
 - .edu, federal labs (Internet2, ESnet, etc)
 - Cloud (e.g., Amazon S3)
 - 100 MB/sec to `us-west-2` disk-to-disk
 - 50 MB/sec to `us-east-1` disk-to-disk
 - SDSC Cloud Services (> 1.6 GB/sec)
- **Sneakernet is high bandwidth, high latency**
 - Sneakernet is error-prone and tedious
 - Neither scalable nor future-proof
 - For Janssen, it was more time-effective to transfer results at 50 MB/s than via USB drives and a car trunk