Lustre at Petascale: Experiences in Troubleshooting and Upgrading

CUG 2012

Stuttgart, Germany

Matthew Ezell – ORNL, HPC Systems Administrator

Rick Mohr – NICS, HPC Systems Administrator

John Wynkoop – NICS, HPC Systems Administrator

Ryan Braby – NICS, HPC Operations Group Lead

National Institute for Computational Sciences

University of Tennessee

- NICS runs three production machines
  - Keeneland (HP GPU Cluster)
  - Nautilus (SGI UltraViolet)
  - Kraken (Cray XT5)

Kraken XT5

Current Kraken Configuration	
Cabinets	100
Interconnect	SeaStar2 3D Torus
Peak Speed	1173 Teraflops
Compute processor type	AMD 2.6 GHz Istanbul-6
Compute cores	112,896
Compute nodes	9,408
Memory per node	16 GB (1.33 GB/core)
Total memory	147 TB

Overview

- Troubleshooting Lustre Issues at NICS
- Upgrading Kraken to CLE 3.1 / Lustre 1.8.4
- Future Plans

Kraken Lustre Hardware

- 6 Couplets of DDN S2A9900-10 Shelves

Kraken Lustre Hardware

- Currently internal
  - Although there are plans in the works to change this
- Peak of 36GB/sec, demonstrated performance over 30GB/sec with IOR

Lustre Credits

- Lustre uses a “credit system” as a flow control mechanism between peers
- “credits” controls how many LNet messages can be sent concurrently over a given network interface (NI)
- “peer credits” controls how many LNet messages can be sent concurrently to a single peer
- The Cray CLE install script, by default, sets the number of client credits to 2048

Lustre Credits

- Kraken's OSS server were frequently becoming overloaded and unresponsive
- Unfortunately, NICS staff were unable to find specific recommendations providing a formula to calculate an appropriate number
- The number of credits was slowly reduced to 192 on each compute node
  - The aggregate performance was not degraded but the maximum load on the OSS servers declined
  - Single-node performance was slightly limited due to this change.

Small I/O

- Kraken's DDN Controllers are optimized for large, streaming IO operations (namely 1MB)
- Some jobs read and write many small requests, causing a very high load on the OSS servers
- How do you tell which jobs are performing "poor I/O"?

Small I/O

- Lustre keeps a ring-buffer request history containing NID and opcode
- Use *apstat* info to correlate this to jobs

```
kraken# ./lustre_requests
Job      User       Cores  Age    Count
1850782  userA      3072   00:06  85522
1849593  userB      600    09:10  39986
1850042  userC      2628   11:57  22386
1849819  userD      132    05:59  12368
1849929  userD      132    --     9994
1849722  userD      132    05:16  6855
1848293  userE      2160   00:52  6835
1850787  userF      120    --     6481
1849936  userD      132    02:12  5796
1850779  userG      24     00:11  5088
```

Small I/O

- **How can you tell if a job is just doing a lot of I/O compared to a lot of "bad" I/O?**

```
kraken# cat extents_stats
snapshot_time: 1325878779.789272
                read     |       write
 extents   calls % cum% | calls    %   cum%
 0K - 4K:    34   20   20 |  1758   98     98
 4K - 8K:     0    0   20 |     0    0     98
8K - 16K:   135   79  100 |    32    1    100
```

MDS Lock Exhaustion

- The MDS must keep track of granted locks
- Compute nodes keep a LRU of locls
- Kraken's compute nodes cached 1200 locks (100 locks per core) with lru max age set to 9000000 seconds
  - Although ALPS "flushes" Lustre after each aprun
- The MDS was OOMing because there were too many locks outstanding
- LRU set to 300 to avoid the issue

OST Allocation Method

- Lustre has two methods to choose where to place stripes of files
  - Quality of Service (QOS) attempts to even OST utilization
  - Round-Robin (RR) tries to spread out allocations, maximizing bandwidth
- The method currently in use depends on *qos_threshold_rr* and the difference in minimum and maximum OST utilization

OST Allocation Method

Table I
IOR POSIX FILE-PER-PROCESS (CLE 2.2, 300 NODES, 1 STRIPE)

Test	Max Write (MB/sec)	Max Read (MB/sec)
QOS 1	9760	9465
QOS 2	9437	8981
RR 1	29880	18970
RR 2	29987	20486

Table II
IOR POSIX FILE-PER-PROCESS (CLE 2.2, 300 NODES, 4 STRIPES)

Test	Max Write (MB/sec)	Max Read (MB/sec)
QOS 1	7797	11930
QOS 2	8444	12666
RR 1	9969	16886
RR 2	12653	16590

Poorly Striped Files

- Users can easily fill up an OST
  - Usually from someone running “tar” with default striping
  - Typically pushes us into QOS allocator
- Use “lfs df” to determine which OSTs are full
- Use “lfs quota” to determine which user is causing the problem
- Use “lfs find” to determine which file(s)
- Re-stripe the file

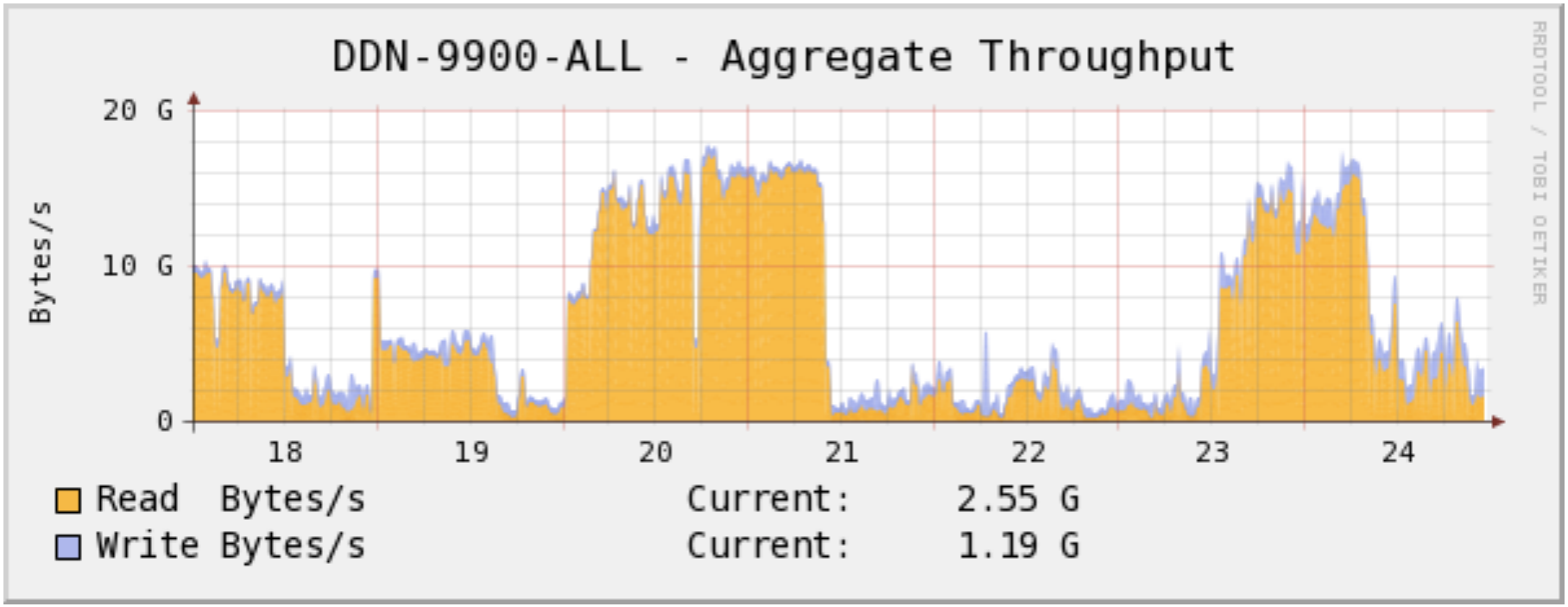
Purging

- "Users will fill up any file system you give them"
- Files not accessed in 30 days are eligible for purging
- Currently use scripts based on "lfs find"
- Looking into taking advantage of "ne2scan" output

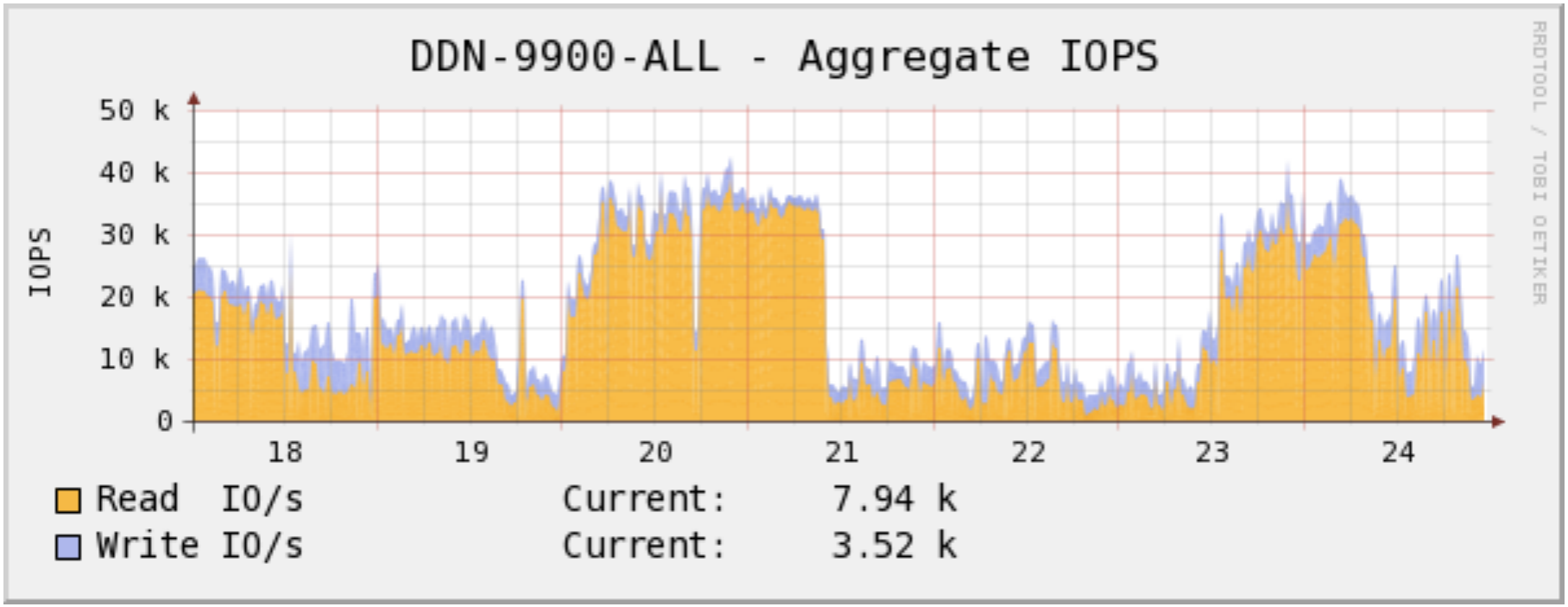
Monitoring

- Simple scripts integrated into Nagios
- Performance monitoring by Cacti

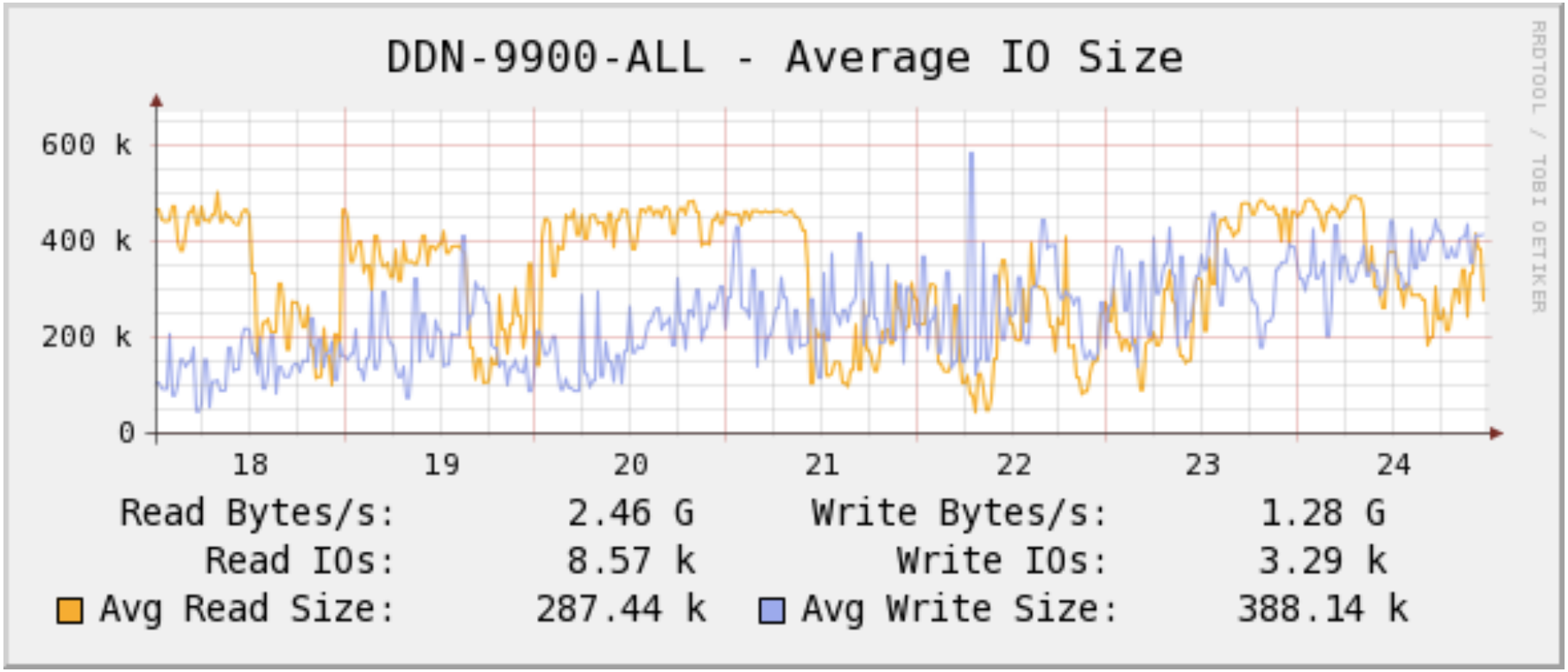
Aggregate Throughput



Aggregate IOPS



Average IO Size



Comparing CLE2.2 to CLE 3.1

- **CLE 2.2:**
  - **SLES 10**
  - **Lustre 1.6.5**
  - **Past end-of-life**
- **CLE 3.1**
  - **SLES 11**
  - **Lustre 1.8.4**
  - **End-of-life**

A full re-install is required to migrate

Athena Test Installation

- 48 Cabinet XT4 decommissioned to users in 2011
- 2 cabinets left powered on for a test system
  - 24 compute blades
  - 24 service blades
- Installed CLE 3.1 while preserving the Lustre 1.6 file system
- It “just worked”

Early Kraken Test Shots

- Wanted to test the OS before moving there in production
- The Lustre file system is internal, so it gets upgraded also!
  - Not sure we are ready for that
  - Worried about incompatibilities
- Solution: don't mount Lustre

IB SRP and scsidev

- CLE 2.2 had “scsidev” to provide persistent device names
- This is now deprecated
- Cray created a udev rule and script called “scsidev-emulation”
- It doesn’t work for new devices
- Solution: re-trigger udev later in the boot

MDS Hardware Incompatibility

- Originally used a DDN EF2915
  - RAID6, not the best for metadata
  - Device is approximately 3.5TB
- Ran into one small problem:

```
READ CAPACITY(16) failed
Result: hostbyte=0x07 driverbyte=0x00
Use 0xffffffff as device size
4294967296 512-byte hardware sectors: (2.19 TB/2.00 TiB)
```

MDS Hardware Incompatibility

- DDN provided new hardware
  - DDN EF3015
  - RAID10, much better for metadata
- Had to do a block-level ‘dd’ to transfer the data

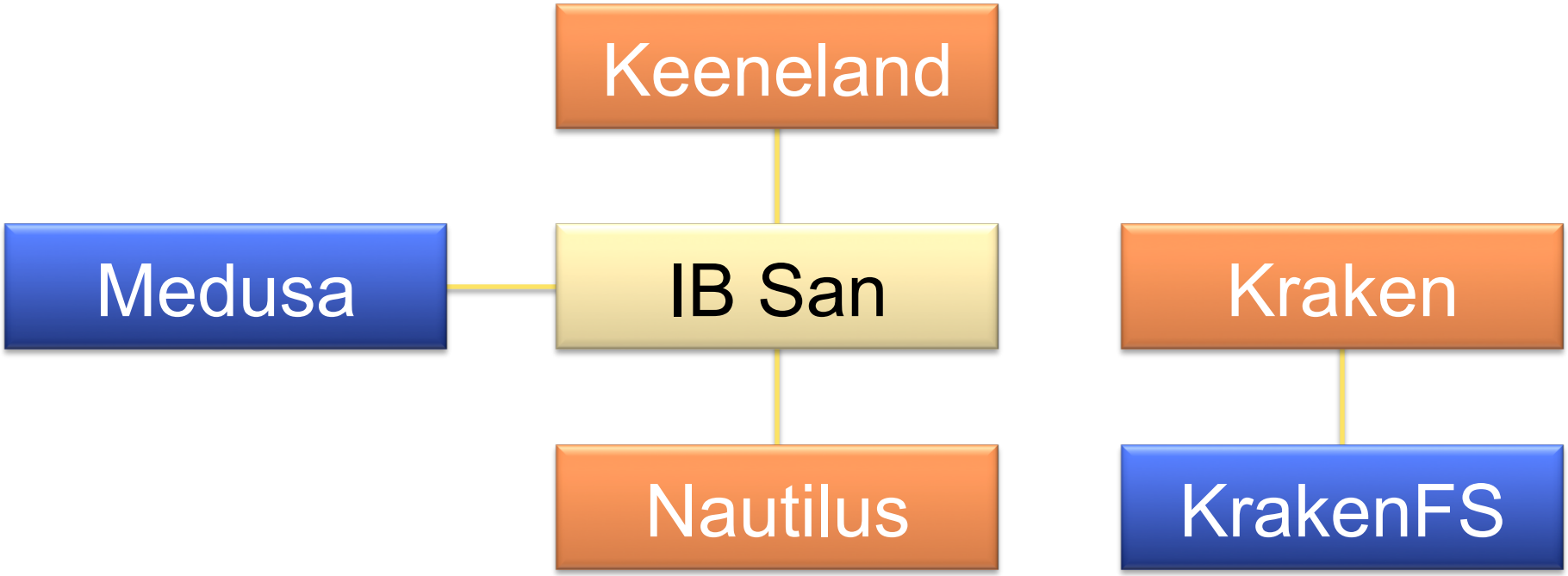
Production with CLE 3.1

- Seems to be working as expected

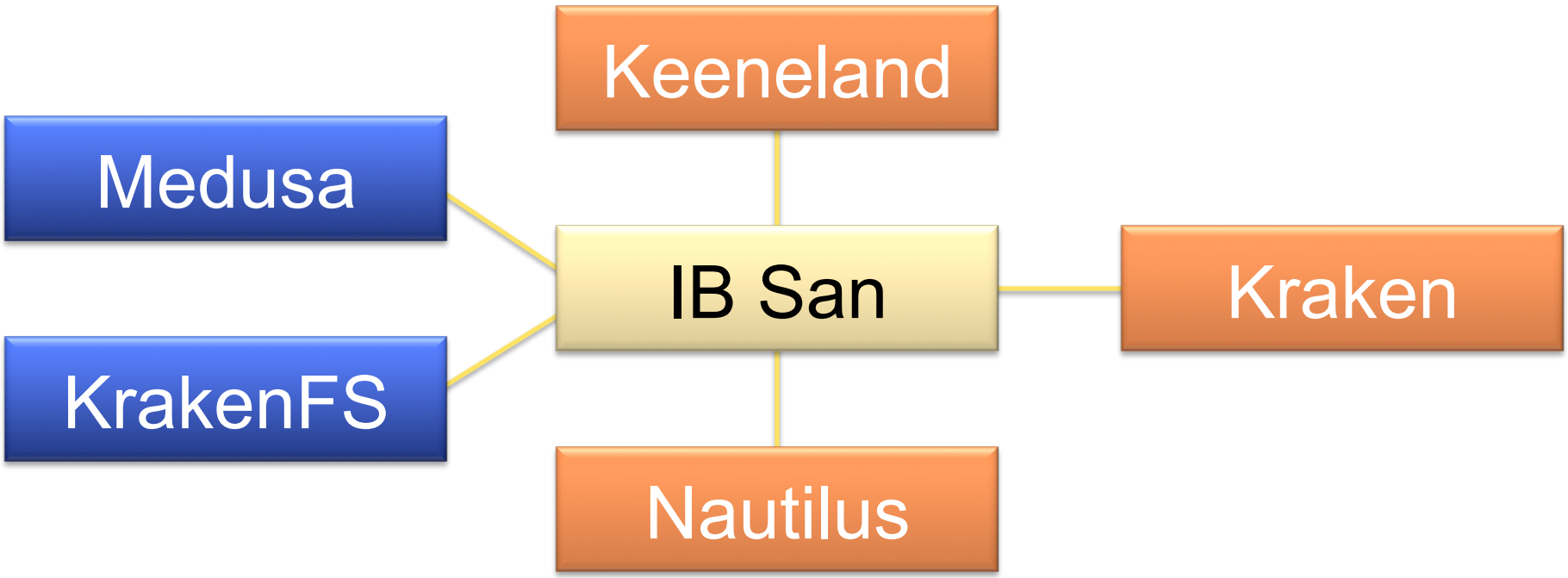
Path Forward

- Want to mount the site-wide Medusa file system on Kraken
- Want to externalize Kraken's file system
- Chicken and the egg problem

Current Topology



Future Topology



New Lustre Version

Bug ID	779592
Summary	CLE3.1 Interoperability with External Lustre 2.1 Servers

Created	12/15/2011 9:36:00 AM	**Product**	CLE
Status	RESOLVED	**Component**	Lustre
Resolution	WONTFIX	**Version**	3.1.UP03
Severity	urgent	**Hardware**	XT5
Keywords		**OS**	Service Node (SeaStar)
Class	Software	**Fixed In**	
Change IDs		**Clones**	

Questions?

Feel free to contact me at ezellma@ornl.gov