Determining health of Lustre filesystems at scale

Jason Hill
HPC Operations Group
ORNL

OLCF
OAK RIDGE LEADERSHIP COMPUTING FACILITY

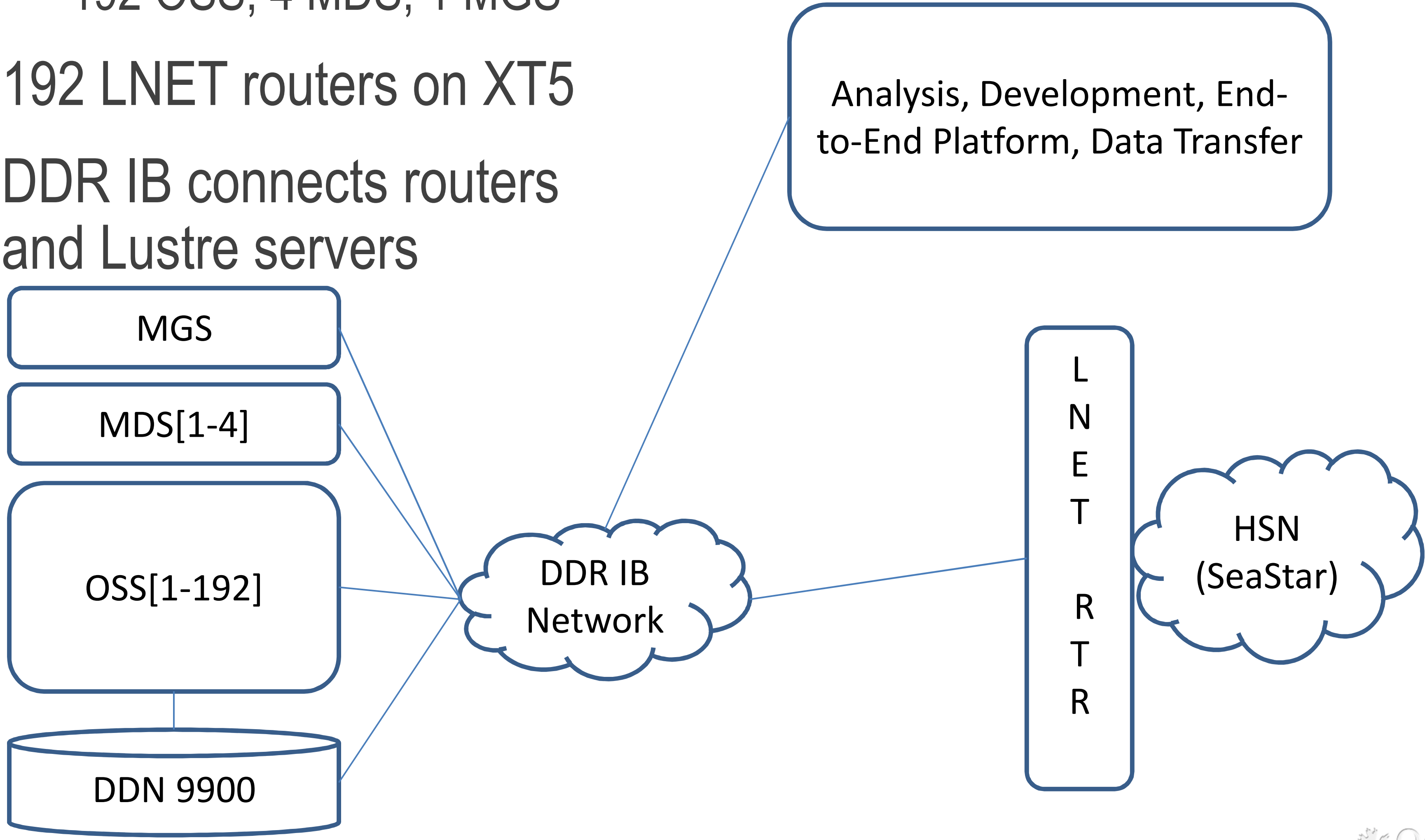
Cray User's Group
2011, Fairbanks, AK
05-25-2011

Overview

- Overview of architectures
- Lustre health and importance
- Storage monitoring
- Server monitoring
- Lustre monitoring
- Log monitoring
- Headaches
- Where do we go from here?
- Conclusion

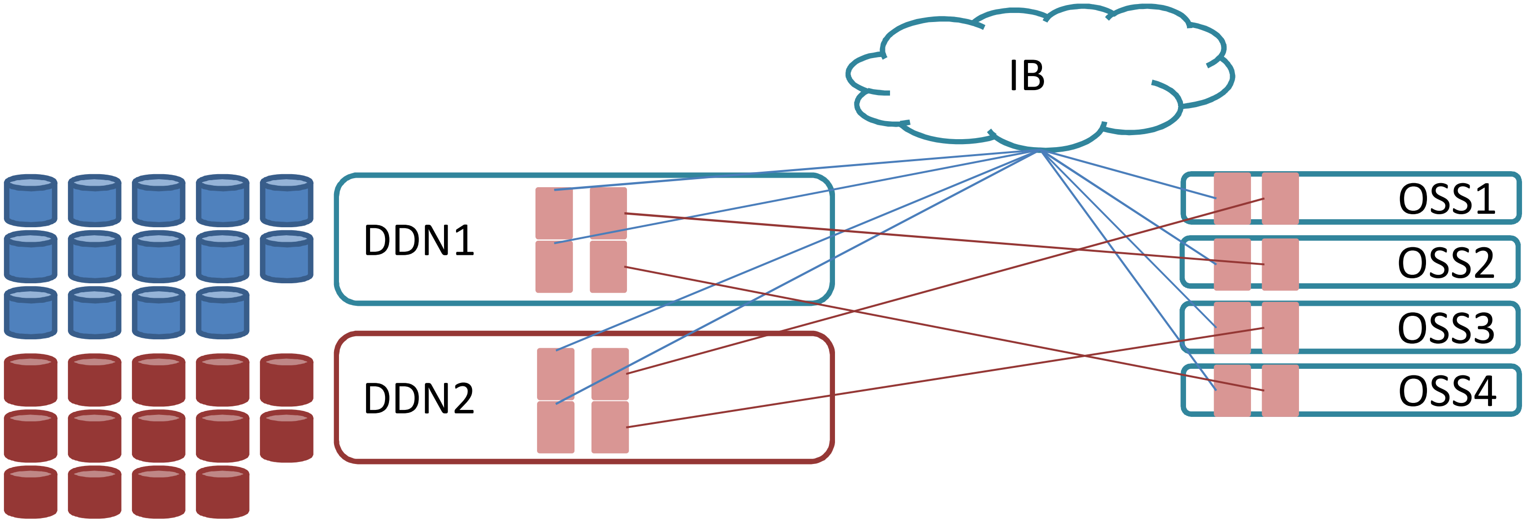
Spider Architecture

- 197 Lustre servers
  - 192 OSS, 4 MDS, 1 MGS
- 192 LNET routers on XT5
- DDR IB connects routers and Lustre servers



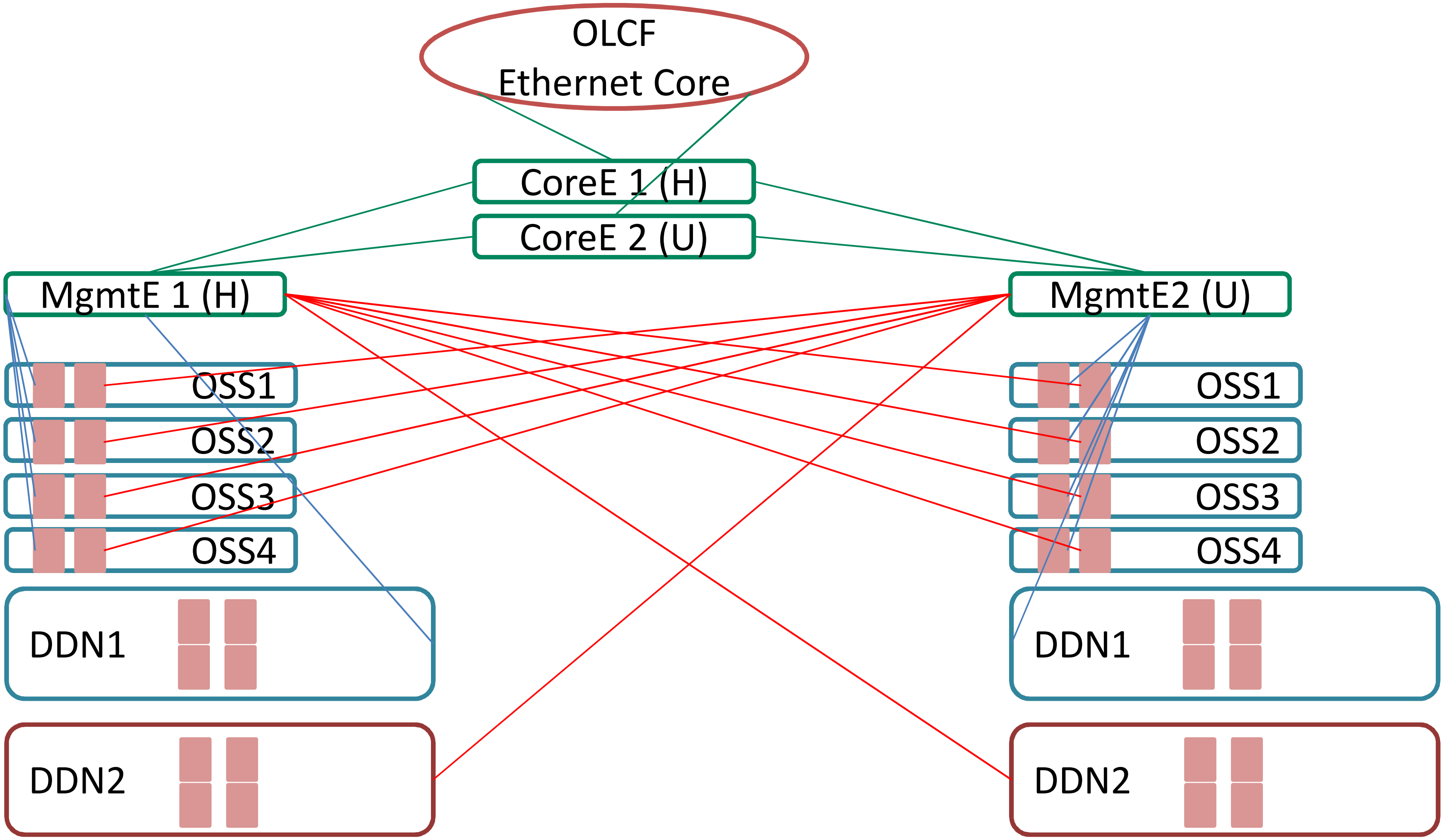
Spider Architecture

- Scalable Units
  - 1 DDN 9900 couplet, 4 OSS nodes
- Scalable Clusters
  - 3 SU's
- 16 Scalable Clusters in all

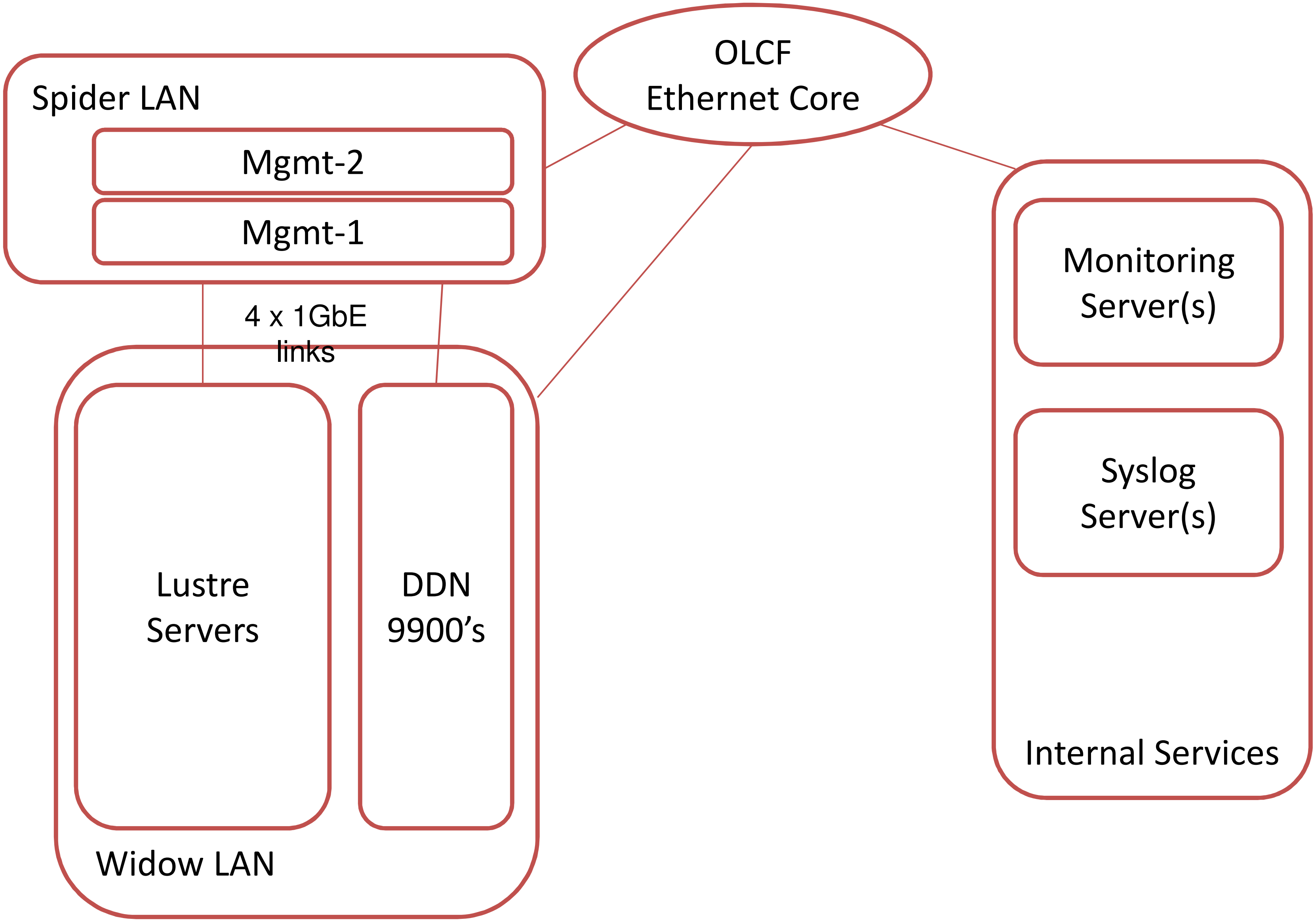


Spider Architecture

- Cross connected to single fed Ethernet switches



Monitoring Infrastructure



Monitoring Infrastructure

- Nagios Server -- HP DL 360G6
  - Quad Socket Quad Core 32 GB system memory
  - Bonded GbE (2 links)
- Gigabit Ethernet backplane from Lustre VLAN to Int-Services VLAN
- Network utilization 45 Mbit so network isn't a bottleneck
- Nagios
  - Set up parent/child relationship between Lustre servers
  - Parent/Child relationships for DDN controllers to Lustre OSSes

Monitoring Infrastructure

- SNMP monitoring
  - Most Nagios plugins/checks use snmpwalk
  - ORNL has registered OID space
  - We use custom OIDS to execute a script on the local machine
  - snmpwalk –v 2c –c mycommstring hostname OID
    - OID name /path/to/script in /etc/snmp/snmpd.local.conf
- Additionally use a SNMP trap to set downtime in Nagios
  - Relies on parent/child relationships
  - Suppress notifications for known issues

snmpd.local.conf example

```
exec 1.3.6.1.4.1.341.49.5.1.3 monitor_multipath /opt/bin/monitor_multipath.pl
exec 1.3.6.1.4.1.341.49.5.1.4 monitor_ib_health /chexport/bin/monitor_ib_health.sh
exec 1.3.6.1.4.1.341.49.5.1.17 lustre_health /opt/bin/lustre_healthy.sh
exec 1.3.6.1.4.1.341.49.5.1.18 lnet_stats /opt/bin/lnet_stats.sh
exec 1.3.6.1.4.1.341.49.5.1.23 lustre_dev /opt/bin/lustre_device_check.sh widow1
exec 1.3.6.1.4.1.341.49.5.1.24 lustre_dev /opt/bin/lustre_device_check.sh widow2
exec 1.3.6.1.4.1.341.49.5.1.25 lustre_dev /opt/bin/lustre_device_check.sh widow3
exec 1.3.6.1.4.1.341.49.5.1.26 lustre_dev /opt/bin/lustre_device_check.sh widow
exec 1.3.6.1.4.1.341.49.5.1.28 aacraid /chexport/bin/check-aacraid.py
exec 1.3.6.1.4.1.341.49.5.1.35 aacraid /opt/bin/aacraid.sh -b
exec 1.3.6.1.4.1.341.49.5.1.36 aacraid /opt/bin/aacraid.sh -c
exec 1.3.6.1.4.1.341.49.5.1.37 aacraid /opt/bin/aacraid.sh -d
exec 1.3.6.1.4.1.341.49.5.1.30 collectl /opt/bin/collectl.sh
exec 1.3.6.1.4.1.341.49.5.1.34 osirisd /opt/bin/osirisd.sh
```

Lustre Health

- What is it?
  - Can users access their files?
  - Is it performance related?
  - All Lustre services/devices are “online”
    - What about failover conditions?
- We use the “online” thought process for our current monitoring
  - Also looking if users can access their files
- Also look if Storage resources are online

Storage Monitoring

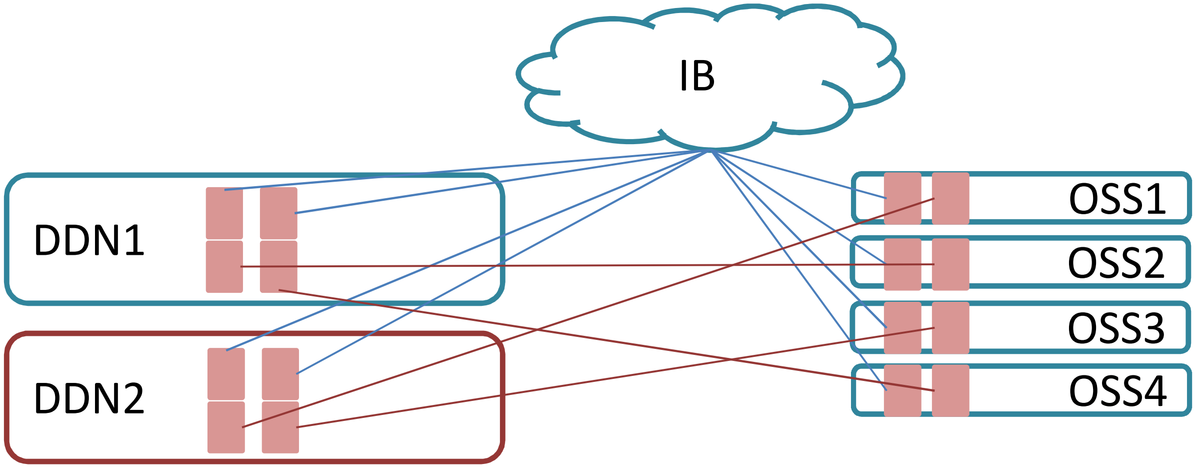
- DDN 9900's don't have much monitoring capability
  - Ping the IP of the controller
- DDN has a GUI for looking at controllers, doesn't work well for our setup
  - Serializes connections to DDN's before displaying
    - 96 couplets takes 20-30 minutes to scan
- More about this in log monitoring

Server Monitoring

- Need to establish that the hardware can communicate
  - Ping the OSS
  - sshd alive
  - OSS is alive on the IB network
- Need to establish that the server is running correct things
  - OpenSM to backend storage
  - Configuration monitoring (osiris)
  - Statistic gathering (collectl)
- Need to establish the hardware has no problems
  - Dual PSU connected to House and UPS
  - Input Voltage within range
  - Fans/Temps all within range
  - Plugin from Nagios Exchange for Dell OMSA

Server Monitoring (2)

- Multipathd checker
  - Use IB SRP to connect to DDN 9900 via two paths
  - Look at output of echo “show multipaths topology” > multipathd –k
  - Making sure that there are 2 good paths
    - Looking for active and ready paths.



Server Monitoring (3)

- IB health monitor
  - Sources configuration file that gives interface, speed
  - Verifies that state is active, speed correct, interface is up
  - If not correct return -2 and print text

```
# cat monitor_ib_health_oss.conf
mthca0 1 20
mthca0 2 20
mthca1 1 20
```

Lustre monitoring

- Do we have all of our devices mounted
  - Source the configuration file for the filesystem and compare with mounted devices
  - **OSTDEV[0]=“HOSTNAME:/dev/mpath/LUNNAME“**
  - Mounted at /tmp/lustre/FSNAME/OSTDEV
- What is the status of /proc/fs/lustre/health_check ?

```
if [ ! -e /proc/fs/lustre/health_check ]; then
  exit 2
fi

if [[ $(cat /proc/fs/lustre/health_check) != "healthy" ]]; then
  echo "CRITICAL: Lustre is unhealthy. Call Lustre OnCall Admin"
  exit 2
else
  echo "OK: Lustre is healthy"
  exit 0
fi
```

Lustre Monitoring (2)

- Is LNET performing okay?

```
if [ ! -e /proc/sys/lnet/stats ]; then
   exit 2
fi

last_stat=$(cat /tmp/last_lnet_stat)
curr_stat=$(/bin/awk '{print $1}' /proc/sys/lnet/stats)

# Now we set the last stat for the next run
echo $curr_stat > /tmp/last_lnet_stat

if [[ $curr_stat -lt 30000 && $last_stat -lt 30000 ]]; then
  echo " OK: Curr: $curr_stat Last: $last_stat"
  exit 0
elif [[ $curr_stat -gt 30000 && $last_stat -lt 30000 ]]; then
  echo " OK: Curr: $curr_stat Last: $last_stat"
  exit 0
else
  echo " CRITICAL: Curr: $curr_stat Last: $last_stat"
  exit 2
fi
```

Lustre Monitoring (3)

- Client side monitoring
  - ls timer
    - If greater than 30s, send mail
    - From XT5 and external sources
  - df
    - Manifested by Nagios built-in filesystem checks
  - lfs osts
    - Can look for inactive OSC's
  - lfs check servers

Log Monitoring

- Simple Event Correlator
  - DDN Syslog message parsing
  - Lustre Syslog message parsing
- Rationalized printk
  - John Hammond et. al from TACC
  - Formatting messages to programmatically parse
- Splunk
  - Use for trending and reporting as we move forward

Headaches

- Snmpwalk times out with 7000-10000 entries in process table
  - Use snmp bulk get and a starting OID to help things
  - Helps with process checks on MDSes where 4000-8000 mdt processes exist
- Current monitoring load is large as a percentage of all OLCF monitoring
  - 30% of hosts in Nagios are Lustre servers (323)
    - Not counting LNET router monitoring
  - 40% of services in Nagios are Lustre services (2441)
  - Static load average is ~10
  - Every “new” service we monitor adds (min) 210 checks to Nagios
- Aspirin : Delegate Lustre checks to another Nagios server?

Where do we go from here

- Several things we want to do
  - LMT
  - Custom DDN 9900 monitoring
    - Verify configuration is as expected, alert if not
  - Parsing server side client stats
    - Not simple with 18k clients
  - Make current monitoring failover aware
    - Could require on-the-fly Nagios configuration
  - Implementing rationalized printk

Conclusion

- Pace of current monitoring won’t stand up to next generation
- As system size grows problems arise in monitoring
- Can changes be made to snmp based monitoring?
- Explore options that are currently available (NRPE, delegation)
- Monitoring health is critical to delivering stable storage platform for compute, analysis, visualization, and data transfer to other sites

Questions?

This research used resources of the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DEAC05-00OR22725.