Welcome!

Virtual tutorial starts at 15:00 GMT

archer

epcc

THE UNIVERSITY OF EDINBURGH

Lustre and I/O Tuning

Performance Tips

Reusing this material

Slides 5, 6 and 14 have been provided by our colleagues at Cray. The content on these slides may not be used without express permission.

Lustre Fundamentals

Lustre is the parallel file system used for the /work spaces on ARCHER (/fs2, /fs3, /fs4)

Consists of:

- Object Storage Target (OST): RAID array of HDDs
- Object Storage Server (OSS): server managing one or more OSTs
- MetaData Server (MDS): handles individual file metadata (one MDS per file system)

Files are striped over OSTs – high performance is achieved through accessing these in parallel

ARCHER file systems

Connected to the Cray XC30 via LNET router service nodes

Infiniband Network

/fs2
12 OSSs
48 OSTs
480 HDDs
4TB per HDD
1.4 PB Total

/fs3
12 OSSs
48 OSTs
480 HDDs
4TB per HDD
1.4 PB Total

/fs4
14 OSSs
56 OSTs
560 HDDs
4TB per HDD
1.6 PB Total

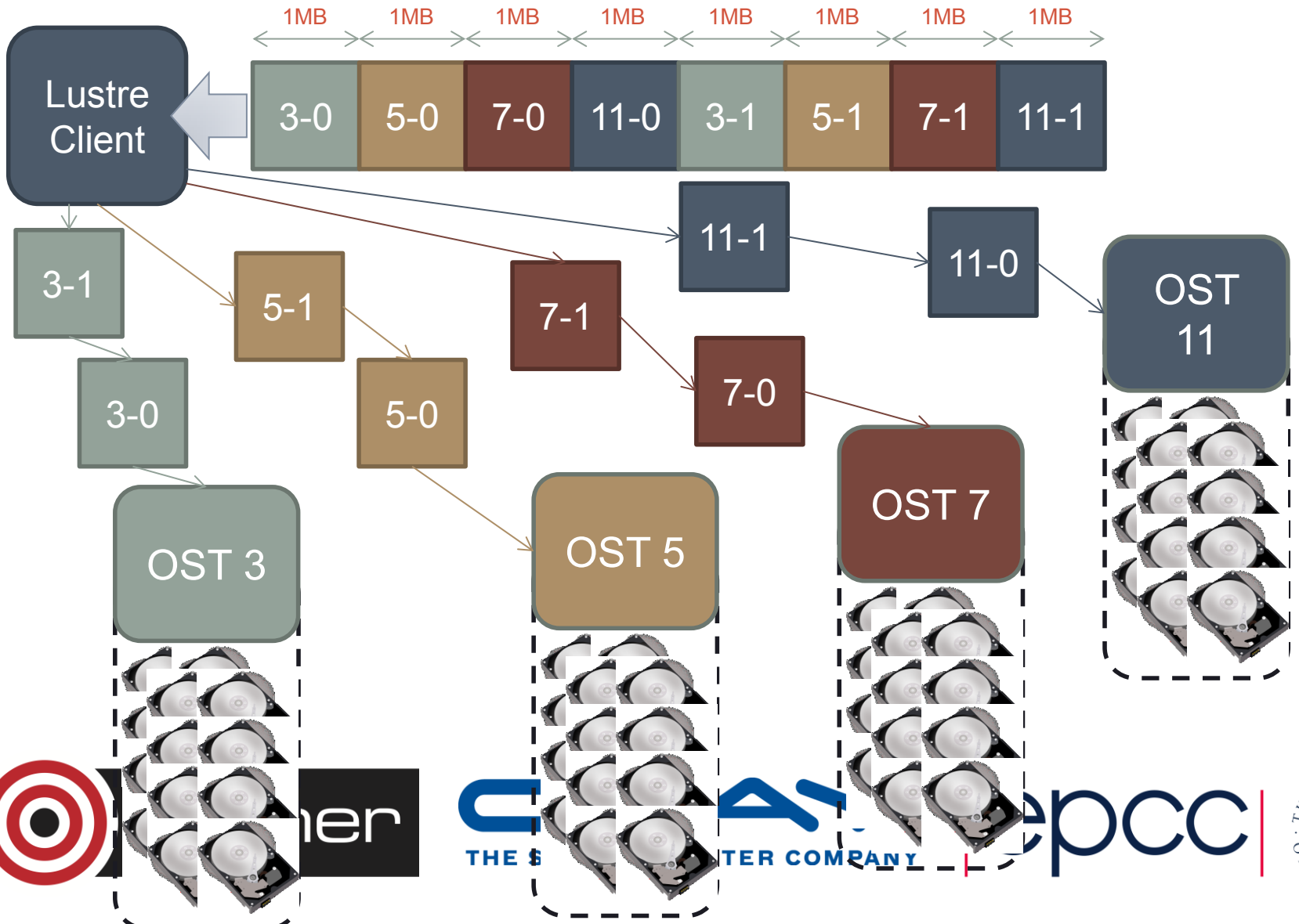
archer

CRAY THE SUPERCOMPUTER COMPANY

epcc

THE UNIVERSITY OF EDINBURGH

File decomposition – 1 Megabyte stripes



Tuning Lustre

Lustre can (and should) be configured by the user to fit their usage pattern

Most common commands:

- lfs getstripe
- lfs setstripe

Others available (check documentation):

```
$> lfs help
Available commands are:
        setstripe
        getstripe
        setdirstripe
        getdirstripe
        …
```

lfs getstripe

Returns the stripe count for a file or a directory (recursive)

Example use:

```
$> lfs getstripe default-striped-dir
default-striped-dir
stripe_count:   4 stripe_size:    1048576 stripe_offset:  -1
default-striped-dir/myfile
lmm_stripe_count:   4
lmm_stripe_size:    1048576
lmm_pattern:        1
lmm_layout_gen:     0
lmm_stripe_offset:  15
        obdidx           objid           objid           group
            15       114711232       0x6d65ac0               0
             7       114277483       0x6cfbc6b               0
            23       114253166       0x6cf5d6e               0
            40       110783889       0x69a6d91               0
```

Directory at the default stripe count on ARCHER (4) containing one file striped over OSTs with IDs 15, 7, 23 and 40. Stripe size is 1 MB (the default on ARCHER).

lfs setstripe

Sets the stripe for a file or a directory

Example use:

```
$> lfs setstripe -c 1 single-striped
$> lfs getstripe single-striped
single-striped
stripe_count:   1 stripe_size:    1048576
stripe_offset:  -1
```

Stripe count set to 1 on directory named “single-striped”

Any files and subdirectories created here will inherit the parent’s stripe settings

lfs setstripe

Note: it is not possible to change a file's stripe settings once it is created:

```
> lfs setstripe -c -1 myfile
error on ioctl 0x4008669a for 'myfile' (3):
stripe already set
error: setstripe: create stripe file 'myfile'
failed
```

File must be copied to a directory of new desired stripe size

Count of -1 indicates file/directory should use all available OSTs

Lustre Key Points

Lustre is designed to deliver high bandwidth access to a small number of files

Not intended for working with large number of small files:

- Causes bottlenecks at MetaData Server (MDS)
  - Don't have 1000s of files in one directory
- Overhead of repeatedly querying OSTs
  - File size information is stored on the OSTs

Lustre is not impervious to failure:

- Loss of an OST means all files using that OST are inaccessible
- /work is NOT BACKED UP!

Selecting Best Stripe Counts

Default striping on ARCHER is 4, unlikely to be optimal for all cases

Two extremes:

One file written to by multiple processes, stripe over all OSTs:

```
lfs setstripe -c -1
```

One file per process (and number of processes > number of OSTs), stripe over a single OST:

```
lfs setstripe -c 1
```

Always let the system decide which OSTs to stripe over

Quantifying Performance

What is good performance on ARCHER?

- Generally see ~500MB/s per OST
- This is the serial limit. If getting that, not achieving parallel I/O

Always benchmark and quantify bandwidth

- Use the Cray performance tools

Contention is an issue – can see huge variance in results

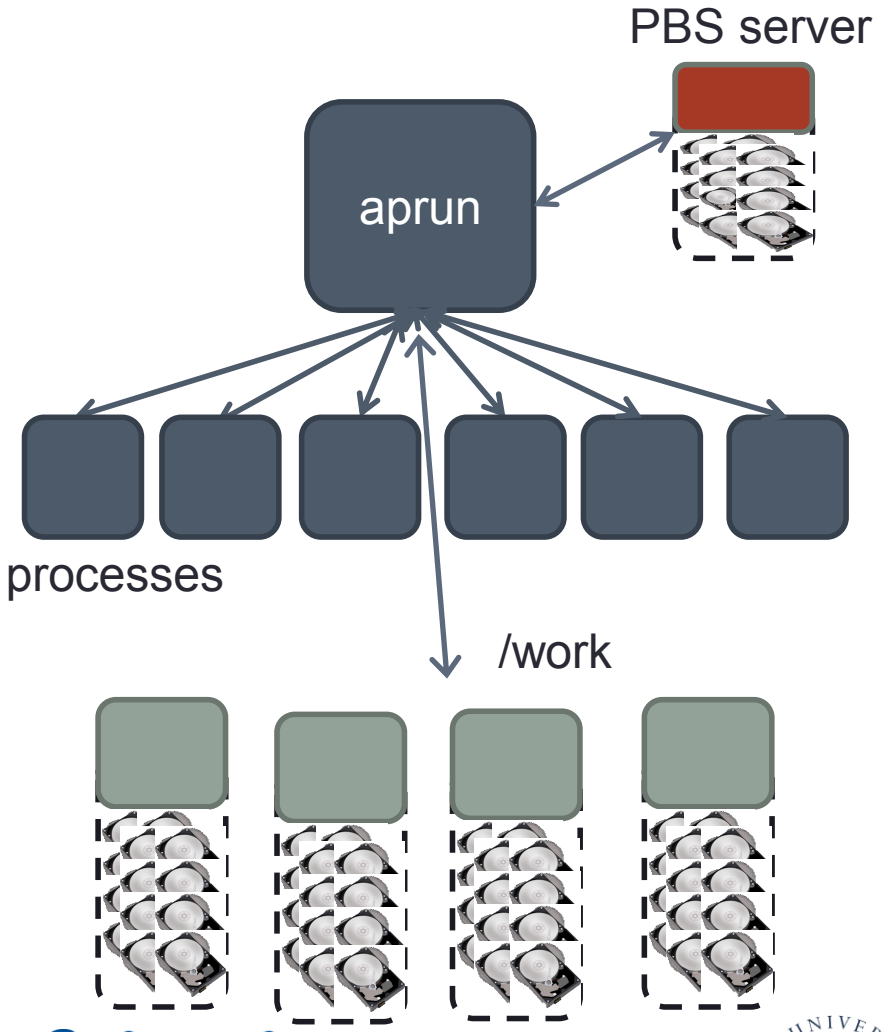
- Do multiple runs at different times of day
- Look at best and worst case

Beware of caching effects on performance results

Standard Output and Error (not Lustre, but important)

Note that STDIN, STDOUT and STDERR I/O streams are serialised through aprun

Disable all debugging print statements when benchmarking or running a simulation



Performance – Large Number of Files

"setting striping to 1 has reduced total read time for his 36000 small files from 2 hours to 6 minutes"

- comment on resolution of an ARCHER helpdesk query.

User was performing I/O on 36000 separate files of ~300KB with 10000 processes

Had set parallel striping to maximum possible (48 OSTs / -1) assuming this would give best performance

Overhead of querying every OST for every file dominated the access time

Moral: more stripes does not mean better performance

Performance – Large Number of Files 2

15GB consisting of 5500 1.5-4MB files

Effect of striping on serial “tar” operation:

```
$> time tar -cf stripe48.tar stripe48
real    31m19.438s
…

$> time tar -cf stripe4.tar stripe4
real    24m50.604s
…

$> time tar -cf stripe1.tar stripe1
real    18m34.475s
…
```

~40% reduction in operation time between 48 and 1 stripe

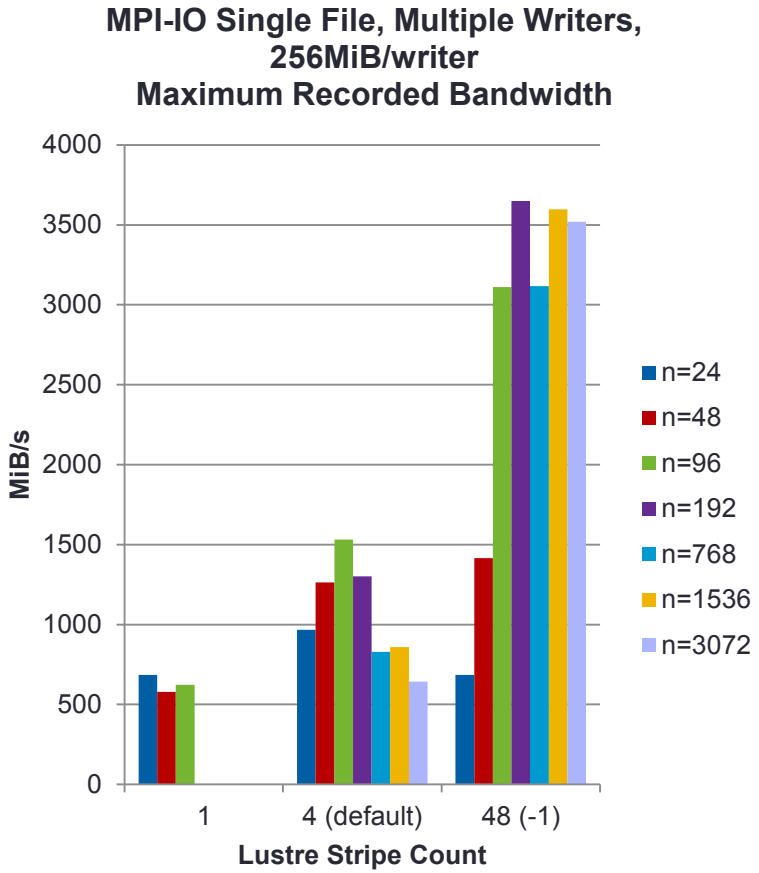
Still bottlenecks at MDS. This access pattern is not recommended, but it is common.

Performance – Single File

Results up to 3072 cores (128 ARCHER nodes)

Single stripe only tested to 4 nodes (96 cores) – fairly consistent ~500MiB/s

4 stripes peak at 1500 MiB/s for 96 cores of followed by drop off to 640 MiB/s for 3072 - attributed to contention



Parallel I/O Libraries

Lustre alone is not enough for parallel I/O, the application must be architected properly

Strongly encouraged to use a library to handle I/O rather than reinventing the wheel with POSIX I/O

Library will be more portable and better optimised

MPI-IO, HDF5, NetCDF are all comprehensive solutions and fully supported on ARCHER

Summary

Files are striped over multiple OSTs on ARCHER's /work Lustre file system

Lustre should be tuned to application file usage pattern – more stripes are not always better

File system contention can cause huge variance in results

Investigate libraries when designing programs

Links and References

ARCHER Best Practice Guide – I/O Tuning
http://www.archer.ac.uk/documentation/best-practice-guide/tuning.php#sec-6.7

Manage Lustre for the Cray Linux Environment (April 2015)
http://docs.cray.com/books/S-0010-5203/S-0010-5203.pdf

Lustre Operations Manual
https://wiki.hpdd.intel.com/display/PUB/Documentation

HDF5
https://www.hdfgroup.org/HDF5

netCDF
http://www.unidata.ucar.edu/software/netcdf