Lustre hands-on - Let's have fun with Lustre :)

Make sure you have access to a terminal and try to connect to the system.

Understanding the environment

Storage

Some questions suppose you can connect to the storage system.

1. What type of storage is the system using ? What's the connection backend ?
2. What type of drives ? SAS, NL-SAS, SATA, SSD ?
3. How would you determine the maximum throughput of a drive ?
4. Define and briefly explains what RAID6 is
5. How do you determine the maximum theoretical throughput of a RAID6 made of 10 drives ?
6. How would determine the maximum throughput of a storage system ? What are the limiting factors (if any) ?

Identify the different types of servers

This exercise suppose you are able to connect to Lustre servers.

1. Where is the MGS ?
2. Where is the MDS ? Is there more than one ?
3. Where are the OSS ? How many are there ?
4. Can you identify the MGT ? A MDT ? An OST ?
5. Can you find the LNET used by any of the servers ? Find a server's NID(s).

Lustre client

You will have to connect to a Lustre client for this exercise.

1. Lustre provides a user-space CLI (Command Line Interface), where is it located ? Can you find its man pages ?
2. Check if a Lustre filesystem is mounted and give the total size of the filesystem (data only).
3. How many files can you create on this filesystem ?
4. Try to create a directory. If the filesystem as more than one MDT, create a new directory on each MDT.

5. Create an empty file, look at its striping information. What is the stripe size ? What is the OST count ?
6. Create a new file with striping information that you choose, e.g. stripe size of 16MB, stripe count of 10.
7. How many LNET is your client using ? What is this client's NID(s) ?

Application I/O profiling

This exercise suppose you have a working MPI environment (with a compiler).

1. Retrieve the application source code and compile it. You should obtain multiple binaries.

SimpleIO

We are going to start with the application called "simpleio". This is a very simple application which writes integers to a binary file.

1. Run the application on one node, one thread. It will be interesting to track the application execution time.

If possible, connect to the OSS hosting the target where the file data resides and use different commands to monitor system resources (disks, CPU, memory, network).

2. Repeat with more than one thread and if possible more than one node.

Read the application source code. Can you explain what's wrong with the write pattern ? With the read pattern ?

3. How could you optimise the access pattern ?

You can change the filesize by editing simpleio.h, the constant "FILESIZE" is expressed in bytes.

SimpleIO_SSF

"simpleio_ssf" use a single shared file access pattern (all threads access the same file).

1. Run the application on one node, one thread. Try to track the application execution time.

If possible, connect to the OSS hosting the target where the file data resides and use different commands to monitor system resources (disks, CPU, memory, network).

2. Repeat with more than one thread and if possible more than one node.

Read the application source code. How would optimize the access pattern so each processes perform better ?

You can change the filesize by editing simpleio.h, the constant "FILESIZE" is expressed in bytes.

SimpleIO_MD

Read the application source code. Can you explain what's wrong about the metadata access ?

1. Run the application on one node, one thread. Look at the metadata activity on the MDT. Does the application hit a bottleneck ?
2. Repeat with more than one thread and if possible more than one node.

What can you conclude about metadata access with Lustre ?

3. By reading the source code, suggest a better way to handle metadata access.

Bonus questions

1. How would you determine the maximum number of IOPS (I/O Operations Per Second) of a drive ?
2. Would SSD drives help to improve metadata performance with Lustre ? Would that make a big difference with SAS drives ?
3. From your point of view, what is the biggest problem with the Lustre filesystem architecture ?
4. Lustre is licensed under GPLv2, if you develop an extension to one of its modules can you make that code proprietary ? Does this apply to application using the Lustre API ?