Benchmarking Goals for Lustre and Big Data

This paper suggests a transparent methodology for benchmarking the performance of a storage system Architecture operating under the Lustre™ parallel file system for use with Big Data applications. The benchmark methods proposed here could likely be used with any related global parallel file systems – not just Lustre. The results of the benchmark will enable end users to do a fair, side-by-side comparison of various storage system Architectures that will better equip them to make more informed decisions. The set of Architectures, configurations, tests and metrics defined will benefit the majority of current or contemplated Lustre Big Data deployments; and may serve as a basis for more extensive configurations and tests. The test results and derived metrics would ideally be audited by an Auditor and made publicly available.

While some initial discussions have taken place with the OpenSFS community group on how to implement and progress such a program of testing, no specific program has been agreed.

Storage Systems Architecture

Storage system Architectures are described in Appendix 1, Architectures. These Architectures will be configured and tested in an agreed manner, then results submitted and posted.

Actual Proposed Tests

The following tests and related results are proposed. These tests will be applied to each Architecture under test.

- Aggregate Bandwidth. Aggregate Bandwidth means the average of read and write bandwidth, calculate as follows:
  Aggregate Bandwidth = (Read Bandwidth + Write Bandwidth)/2, in GB/s.

- Bandwidth Density. Bandwidth density means the Aggregate Bandwidth per U, calculated as follows:
  Bandwidth Density = Aggregate Bandwidth/U per Architecture, in GB/s/U.

- Capacity Density. Capacity Density means the storage capacity per U, calculated as follows:
  Capacity Density = Total raw disk capacity/U, in TB/U.

- Power Density. Power density means the power consumed per read , calculated as follows:
  Power Density = Total power/read, in W/GB/s.

- Price Performance. Price Performance means the price per TB of raw storage, calculated as follows:
  Price Performance = Price/Aggregate bandwidth, measured in $/GB/s.
  Price may be either published list price or discounted off list price, so long as submitter clearly states the discount percentage off of list price.

Proposed Submission and Publication

Tests would be run in-house or by an external party(ies). Test results will be included in the Submitter Package and will be submitted by the Submitter to the Auditor. The initial Auditor would need to be selected.

The Auditor would review the Submitter Package and will approve or reject the Submitter Package within thirty (30) calendar days of receiving the Submitter Package. If the Auditor approved the Submitter Package, then the auditor will notify the Submitter by providing a written Approval Notice. If the Auditor rejects the Submitter Package, then the auditor will notify the Submitter the reasons for rejection.

Dan Ferber Brief Background/Experience

Dan holds a Masters in Computer Systems from the University of St. Thomas, and has worked in testing, support, software development, and business development for Cray, SGI, Sun, Oracle, and now Whamcloud, Inc. As a storage vendor-neutral provider of Lustre and Lustre services, Dan and Whamcloud have an interest to see Lustre and storage system benchmarks of various vendors' scalable storage units that are used in conjunction with global parallel file systems. Dan's previous testing work along with his current work related to Whamcloud customers and markets, is what interests him in this area. Dan is based in Eagan, MN.

Appendix 1
Architectures

The following Architectures have been defined by the Working Group.

Architecture Number 1: Scalable Storage Unit

A Scalable Storage Unit (SSU) implementation will consist of the minimum following stand-alone components:

- One or more Object Storage Servers (OSS) with necessary channel connections
- One or more associated Object Storage Targets (OST)
- Client network port connections from the OSSs to the client network
- Linux operating system
- Lustre OSS software
- Misc transceivers, connectors, cables and rails

Additional components that are not included in the SSU but are needed to implement a functional Lustre deployment are:

- Metadata Server(s) (MDS)
- 1 x Metadata Storage (MDT)
- 1 x management server
- 1 x Ethernet management switch
- Linux operating system
- Lustre MDS software
- Misc transceivers, connectors, cables and rails

Architecture Number 2: High Availability Scalable Storage Unit

A High Availability Scalable Storage Unit (HASSU) implementation will consist of the minimum following stand-alone components:

- Two or more Object Storage Servers (OSS) with necessary channel connections
- Two or more associated Object Storage Targets (OST) that provide failover partners for each active OSS
- Client network port connections from the OSSs to the client network
- Linux operating system
- Lustre OSS software
- Misc transceivers, connectors, cables and rails

Additional components that are not included in the SSU but are needed to implement a functional Lustre deployment are:

Metadata Server(s) (MDS)
1 x Metadata Storage (MDT)
1 x management server
1 x Ethernet management switch
Linux operating system
Lustre MDS software
Misc transceivers, connectors, cables and rails

Architecture Number 3: Scalable Storage Unit Appliance

A Scalable Storage Unit Appliance (SSUA) implementation will consist of all the components of the SSU, either embedded to form a complete system or as stand-alone components integrated to form a complete system.

Architecture Number 4: High Availability Scalable Storage Unit Appliance

A High Availability Scalable Storage Unit Appliance (HASSUA) implementation will consist of all the components of the HASSU, either embedded to form a complete system or as stand-alone components integrated to form a complete system.