UML scaling under single namespace

Introduction
The XOR Universal MediaLibrary is an IT-based storage technology engineered to support media applications. This paper describes ways to scale a UML in order to expand both storage capacity and performance under a single namespace. This discussion is limited to UML scaling as a local storage within a single facility. For a discussion on UML scaling across multiple, geographically dispersed facilities under one global namespace, please see our white paper *Private Cloud Storage for Media Applications*.

Volume, Volume Group, File System
There are three software components that form the basis for UML provisioning and scaling:

Volume
A volume is a logical entity that converts a set of raw physical disk drives into logical storage space. A UML a volume is also called a LUN (Logical Unit Number) and represents a RAID group. By default all volumes on a UML are configured as 8+2 RAID 6 groups, each able to tolerate two drive failures. Unlike volumes on other storage systems, a UML volume is truly active active – simultaneously accessible by both UML heads – which is the key to achieving load balancing and high performance for media applications.

Volume group
A volume group is a logical grouping of a set of volumes. A volume group can group a set of volumes in two different ways based on performance and data redundancy requirements:

Striped
Data is striped and simultaneously written to all the volumes in the volume group. The same is true for reads albeit in the opposite direction. This configuration ensures that the overall volume group performance is the aggregate of the performance of all the volumes in the volume group. By default all the volume groups in a UML are configured as Striped.

Concatenated
The volume group converts its set of volumes into one logical sequential storage space. Data is written to the first volume. Once it is filled up, data is written in the second volume, and so on. This configuration aggregates the capacity of all the volumes in the volume group, but not the performance.

File system
A file system maintains files, folders, and their space allocation on a volume group or across multiple volume groups. New volume groups can be added to an existing UML file system to expand its capacity and performance. UML file system is a distributed file system, capable of managing a single namespace across multiple volume groups residing on different physical machines (called UML nodes). There are three ways to provision and add volume groups under one UML file system:

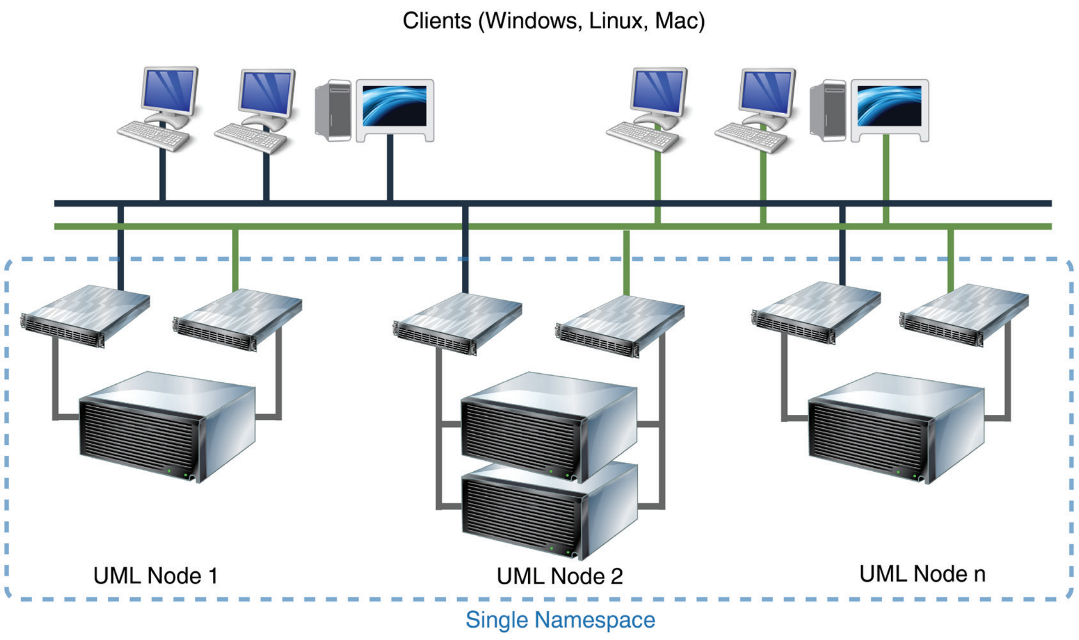
Striped
Each file is striped and simultaneously written to all the volume groups in the file system. The same is true for reads albeit in the opposite direction. This configures ensures that the overall file system performance is the aggregate of the performance of all the volume groups in the file system.

Concatenated
The file system converts its set of volume groups into one logical sequential file system space. Files are written to the first volume group, and then, once the first is filled up, to the second group, and so on. The configuration aggregates the capacity of all the volume groups in the file system, but not the performance.

Round Robin
The file system rotates all file creations across all the volume groups. Each file resides on one volume group so the file has no dependency on the other volume groups. In a typical environment where there are thousands or millions of files, space allocation and performance are statistically load balanced across all the volume groups. Round Robin configuration strikes a happy medium in terms of redundancy and load balancing between Striped and Concatenated. By default, a UML file system is configured as Round Robin.

Scaling a UML

A UML can be expanded by adding a new volume group to an existing file system (single namespace). This operation takes less than 10 seconds and a couple of commands. The following figure illustrates the scale up and scale out options when expanding a UML Grid:



Scale up

A storage expansion can be added to an existing UML. A storage expansion comes with pre-configured volumes representing 8+2 RAID groups. First, a new volume group is created to stripe across all the volumes in the expansion. Then, the volume group can be added to the existing UML file system as either striped, concatenated, or round robin (default).

Scale out

A UML node can be added to an existing UML. A UML node comes with pre-configured volumes representing 8+2 RAID groups. First, a new volume is created to stripe across all the volumes in the node. Then, the volume group can be added to the existing UML file system as either striped, concatenated, or round robin (default).

Conclusion

There are a number of options in scaling a UML file system. Customers should decide on the most appropriate configuration based on the performance and redundancy requirements of their applications. Striped configuration guarantees performance aggregation while round robin is a good way to strike a happy medium between redundancy and load balancing.

32 Mill Street, Greenville 03048
New Hampshire, USA
T: +1603 878 5055 | F: +1603 878 4012
www.xor-media.com