DVS as a Centralized File System in CLE

Jason Temple, *CSCS – Swiss National Supercomputing Center*

ABSTRACT: *Cetnralized File Systems are quickly becoming popular in the High Performance Computing field. With the advent of Cray's new Compute Linux Environment (CLE), and the move away from sysio_init, there is a need for a replacement, as wel as a new paradigm for file access. In this presentation, I will discuss Cray's Data Virtualization Service, or DVS, which is set to be this needed component that is missing with the retiring of Catamount. I will discuss the past, with sysio_init, other potential options, as well as installation, configuration, and general usage and experiences we have had with the new service. Also, I will touch upon what the future may or may not hold for DVS on Cray systems.*

KEYWORDS: Cray XT, DVS, centralized filesystems, Lustre

1. Introduction

Centralized File Systems

Most people who work in High Performance Computing have grown accustomed to having a centralized filesystem. This is usually provided by the center they work with, and has been historically available on most of the large scale systems supported by that center. In the days of Cray's Catamount operating system, the centralized filesystem was accessed via the High Speed Network through the mechanism known as sysio_init, in conjunction with IOBUF. This allowed users to keep their apps, small data files and input files in one location, regardless of the system they were using. Now, with the advent of the Compute Linux Environment, or CLE, this functionality is no longer avaiable. So, Cray has developed a new solution, DVS.

DVS

DVS is an acronym that stands for Data Virtualization Service. It was designed to allow the compute nodes access to centralized filesystems much like during the Catamount days. At this time, most CLE users have not opted to use DVS, and have instead decided to use Lustre (or rarely GPFS) locally for each system.

From a systems administration viewpoint, not having a centralized filesystem is fine. The user can 'simply' just copy their files and datasets to Lustre, and run their jobs from there. There is no added layer that the systems admin has to pay attention to. However, from a user's standpoint, this is inconvenient. Often times, systems administration is a balance between convenience and system stability, with a heavy emphasis on convenience, which is usually the major desire of the user.

Having a centralized filesystem makes sense when it comes to compiling, pre-and-post-processing and generally saving time. With the steady progression of file set size and exponential growth of compute nodes (which produce ever larger amounts of data), 'simply' copying data from one area to another is not only inconvenient, it is viewed by the user as a waste of valuable time. Coupled with the fact that most Lustre filesystems have a very frequent and stringent purge policy on scratch, most users feel that the loss of access to their data in one place is unacceptable.

Solutions

What solutions are there, other than asking Cray to produce a whole new technology that is not only difficult to write, but will most probably be a source of bugs and various headaches? At the moment, we are left with Lustre. Until the advent of external Lustre, this is inadequate. From the recent Cray presentations, it is apparent that the future of Lustre with Cray lies with external Lustre servers, connected by InfiniBand or FibreChannel to I/O servers. If all of your systems are attached to this external Lustre filesystem, you are left with a parallel, high bandwidth, low latency and scalable centralized solution. While this sounds promising, it is not a viable solution for centers that have already invested large amounts of money in pre-external Lustre systems, yet want the support that Cray offers with CLE. Clearly, Catamount is being phased out, and everyone will be forced to upgrade if they want a maintained system by Cray.

There are other efforts being made with other parallel filesystems, namely GPFS and perhaps Panasas. Currently, GPFS is limited to 512 clients, and licensing is based on a per-core basis, which is financially unfeasable for most centers. This is not a solution for any center that has more than that small number of nodes. Panasas, another “hihg-performance filesystem” currently does not scale to the numbers of clients most centers are interested in, nor does it offer the bandwidth and latency that is desirable. So until much work is done by both 3rd party vendors, they will not be a realistic replacement for sysio_init.

DVS as a Solution

Because supercomputing is a user-oriented field, it is perfectly logical that Cray needed to come up with a replacement for sysio_init. This is where DVS comes in. DVS is a set of kernel modules, not unlike Lustre, that allow normal I/O operations on the projected underlying filesystems. It is not a clustered fileystem, it is rather a mount forwarding service. It resides between the filesystem and the IP stack, forwarding normal I/O operations from the compute nodes to the filesystem, allowing for the usual calls such as open(), read(), write() and other such calls without modification. Metadata and disk allocation are handled by the filesystem, and ACL's native to the underlying filesystem are available as well. DVS is basically filesystem agnostic, and by design is unaware of the mount points it projects.

At the moment, disappointingly, DVS only supports serial NFS access. A sort of load balancing is available, by using multiple DVS servers and pointing different nodes at different servers to spread out the access. The bottleneck is currently NFS, which is inherently a weak networked filesystem, unsuitable for most supercomputing needs. However, what DVS does provide, it provides well. In our experience, there has been very little instability, and it is easy to install, configure and troubleshoot.

Installation

DVS is installed via two rpms, dvs-ss and dvs-cnl. Before the integration of DVS into XTinstall, it was a manual process that was nonetheless quite simple. All that needs to be done to install it is to use xtopview, rpm -ivh the dvs-ss package on the service node that will be providing the mount point.

Configuration

On the compute side, one has to make sure that the necessary mount points are available in the compute nodes' filesystem (essentially in /opt/xt-images/templates), and the corresponding fstab file contains the magic incantation that will mount the appropriate mount point at boot time. The fstab configuration is very similar to any standard mount point, set much like a Lustre mount point (type dvs instead of lustre). When /tmp/shell_bootimage.sh is called to create the compute node boot image, the rpm is installed into the compute image, then the image is created. As long as the server is running before the nodes come up, the mount will happen just as expected with Lustre.

To make sure that the mounts are working correctly, either ssh to your compute node and issue the mount command, or, run an interactive aprun session and ls your directory and/or issue the mount command. If these very simple steps are followed, then in my experience, DVS will work without fail.

Current Situation

Currently, there are limitations that make DVS an inviable solution. As mentioned before, the single server NFS mountpoint forwarding is not sufficient. No client can access the same file concurrently, so writes are nearly impossible, while reads are serial and would severely restrict usability. With Catamount, the user was given a large amount of control over I/O via IOBUF, which would allow for controlled caching and readahead that coud be tuned for a particular application. Also, the underlying filesystem was unimportant, sysio_init handled it for you. Users grew used to having access to their centralized filesystem. Now, with the advent and necessity of CLE, users are forced to use Lustre, with the added inconvenience that this entails. DVS has not progressed at the pace that the user community would have desired, but the current headway being made is promising, and leads me to believe it will be very useful in the future.

The Future

Future development of DVS, from the roadmap provided by Cray, indicates that there will be parallel writes and reads, scaling and filesystem agnostic variability. The fact that many compute nodes will be able to access a clustered filesystem, any filesystem for that matter, via DVS will allow for a larger number of clients than the filesystem itself will allow. RDMA, if the interconnect allows for it, will be supported as well. Really, DVS appears as if it may be the future of filesystem access in general on Cray systems. With the ability to transparently allow access from the compute nodes to filesystems without any modification to normal system calls, to scale to a large number of nodes, and to write and read in parallel, there seems to be a bright future for DVS, if all goes according to plan.

Conclusion

From all indications, it appears that Cray will complete the plans detailed in their DVS roadmap, and that the results will be very useful and interesting. If it scales to the level they say it will, and allows for a full utilization of the HSN bandwidth and server I/O bandwidth, the user community will be drawn to DVS as opposed to sticking with the default, Lustre, and the accompanying inconvenience of copying files from one standalone filesystem to another. It looks as if DVS may possibly be a viable centralized filesystem for Cray's new CLE, exceeding expectations.

Acknowledgments

The author would like to thank colleagues, users and vendor staff for a wealth of experiences we have shared together.

About the Author

Jason Temple is a Senior Systems Engineer at the Swiss National Supercomputing Center. He has worked in the HPC field for 10 years, and is currently the administrator for several Cray XT systems. He can be reached at Galleria 2 – Via Cantonale, 6928 Manno, Switzerland, +41916108259, jtemple@cscs.ch.