CSCS

Swiss

National

Supercomputing

Center

DVS as a Centralized File System in CLE

Jason Temple

CUG 2009
Compute the Future

Pre-DVS history

- In the Catamount days, we grew accustomed to using sysio_init, in conjunction with IOBUF.

- The HSN was used to handle the traffic, and the bandwidth was sufficient.

Times are changing

- With the advent of CLE, we no longer have access to sysio_init and IOBUF.
- Therefore, it is necessary to come up with another solution for compute node home filesystem access.
- At this point, most sites using CLE have opted not to use DVS, as it was not really complete or robust at the time of the recent releases

Other options?

- Lustre stands out as the most obvious choice for centers that are upgrading to CLE
- It is supported by Cray, and has a proven track record for both reliability and bandwidth

Other options: GPFS

- One other intriguing option is GPFS.
- GPFS from IBM has matured a great deal in the past few years
- Nice scalability, reliability and bandwidth
- We know that Cray will be adding support for GPFS soon, and, as my predecessor has proven(?), it works now.

GPFS? Panasas?

- The real problem here is:
  - $
  - No Support (yet)
  - (One of these two doesn't scale)

The new solution: DVS

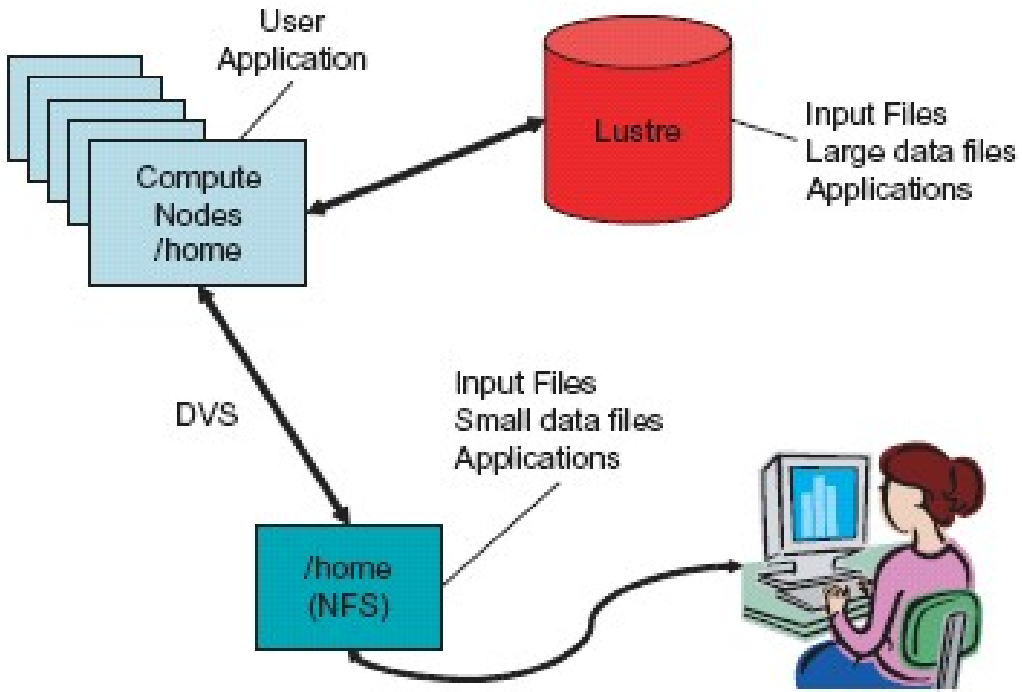
- DVS is now functioning on our systems (without many desired options)
- We are told it is scalable to a large number of nodes
- Currently can support “balancing” (not turned on yet – functionality is there)
- It is easy to install, easy to configure

Why DVS?

- Why do we need something like DVS?
- Historically, users of HPC systems have become accustomed to using their externally available home filesystems.
- **Convenience**

What is DVS?

The Cray Data Virtualization Service



What is DVS (con't)

- A set of kernel modules that allow normal i/o operations on a remotely mounted nfs filesystem
- dvspn 95088 0
- dvsof 145064 1 dvspn
- dvsrmclient 29888 2 dvspn,dvsof
- dvsutil 10936 3 dvspn,dvsof,dvsrmclient
- dvsipc 94104 4 dvspn,dvsof,dvsrmclient,dvsutil
- dvsproc 20568 4 dvspn,dvsof,dvsrmclient,dvsipc
- dvsklib 15184 6 dvspn,dvsof,dvsrmclient,dvsutil,dvsipc,dvsproc
  - Normal system calls such as open(), read(), write(), etc work without modification
  - Attributes supported by the underlying file system are available through DVS(e.g. ACLs)
  - Metadata, disk allocation, and disk I/O are all managed by the underlying file system

DVS ~= NFS

- At the moment, DVS currently only supports NFS mounts, which translates basically into NFS speeds.
- Using IOR, we can achieve basically NFS speeds, which are disappointing, naturally. And, it is, for the moment, a serial bottleneck

Installing DVS at your site

- Actually quite simple. Very little config is needed.
- Two rpms, dvs-ss and dvs-cnl, which contain the same files/modules, except for dvs–ss, which contains man pages
- These rpms install some server kernel modules and some client modules

Install con't

- Now, if the correct flag is set in Xtinstall.conf, the dvs server package is installed automatically (thanks Cray!)
- The rpm for the clients is installed at the time that you create the boot images (/tmp/shell_bootimage_system.sh)
- Once that is done, just some configuring is left.

Install con't

- If you already have a running CLE system, you can install the server rpm directly through xtopview – if you are brave enough to not use XTinstall, or don't want to run it again.
- If the dvs client rpm is not installed in the /tmp/shell_bootimage_system.sh, you can install using
- rpm -r /opt/xt-images/${BOOTIMAGE}/compute –ivh \
- /opt/xt-images/${XTRELEASE}/dvs-cnl-[0-9]*rpm

Configuring DVS server side

- First, you need to ensure the nfs mounts you want to forward are available (fstab of the server node)
- start DVS on the node you wish to use as a server:
- lappend actions { crms_exec_via_bootnode "nid00088" "root" "/etc/init.d/dvs start" }
- Or, you could just launch it by hand if you are just testing

Configure DVS compute side

- Now, you have to make sure that the compute node fstab contains the magic incantation:
- /nfs/xt3-homes/apps /nfs/xt3-homes/apps dvs path=/nfs/xt3-homes/apps,nodename=c0-0c2s6n0
- /nfs/xt3-homes/users /nfs/xt3-homes/users dvs path=/nfs/xt3-homes/users,nodename=c0-0c2s6n0
- 91@ptl:/nid00091 /lus/nid00091 lustre flock,rw 0 0
- Next, make sure the mount points are in the compute node template:
- ls nfs/xt3-homes/
- apps users
- Recreate the boot image using /tmp/shell_bootimage_system.sh -c

Current Limitations

- With yods, you used to have lots of power over your file access
- It didn't matter what filesystem was underneath, catamount happily supplied it
- Now, you are relying on standard Linux caching and i/o (which may be beneficial)
- No load balancing or parallel functionality
- NFS only
- Perception

DVS - Desires

- More varied filesystem support
- Load balancing
- Parallel I/O
- Scalability
- Failover

DVS – the Future

- Very Interesting!
- Parallel reads and writes
- Filesystem Agnostic
- Admin-tunable striping to match underlying fs
- RDMA
- Will project a file system beyond limit of underlying clustered file system
    - GPFS on Linux is limited to 512 clients

The End.

Thanks for your time!