File System Pitfalls, Lessons and Options for OSG Services

Terrence Martin

Site Admin

UCSD CMS T2 Center

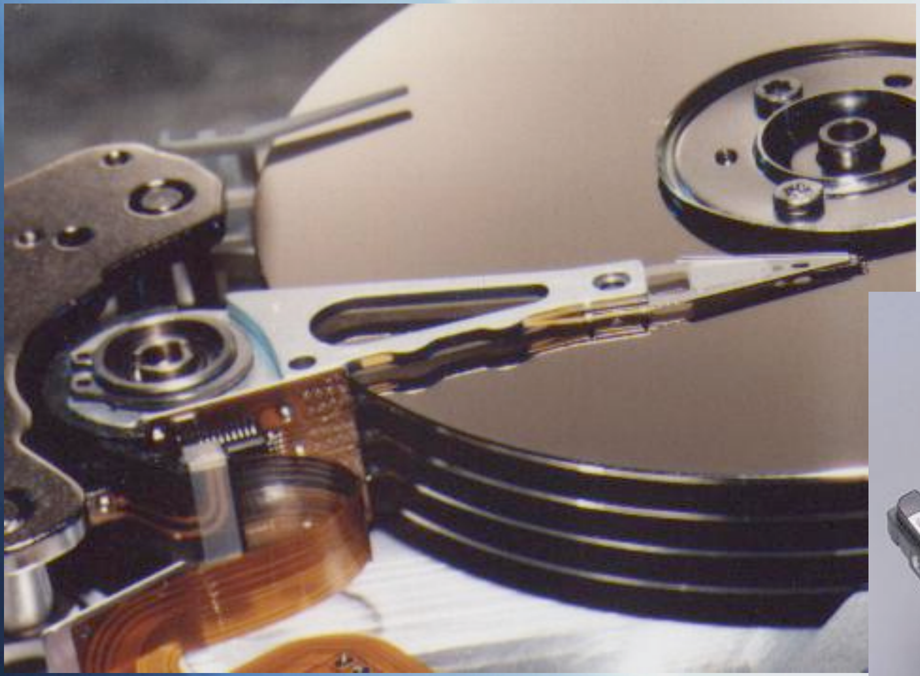
Common File Systems in the OSG

- Local file systems of various flavors (ext3, XFS, JFS, tmpfs, …)
- Network file systems (NFS, CIFS)
- Distributed Storage Systems (dcache)

Physical Storage Mediums

- Hard Disk Drives
- Memory Cache File Systems

Hard Disk Drives



Hard Disk Technology

- Serial Devices that use one or more queues for reads and writes
- Reading and Writing are separate operations
- Queue overhead can severely limit throughput
- Parallel IO operations from the OS and higher impacts disk queue performance
- Capacity increases throughput
- Higher Rotation speed improves access
- Neither have kept up with capacity

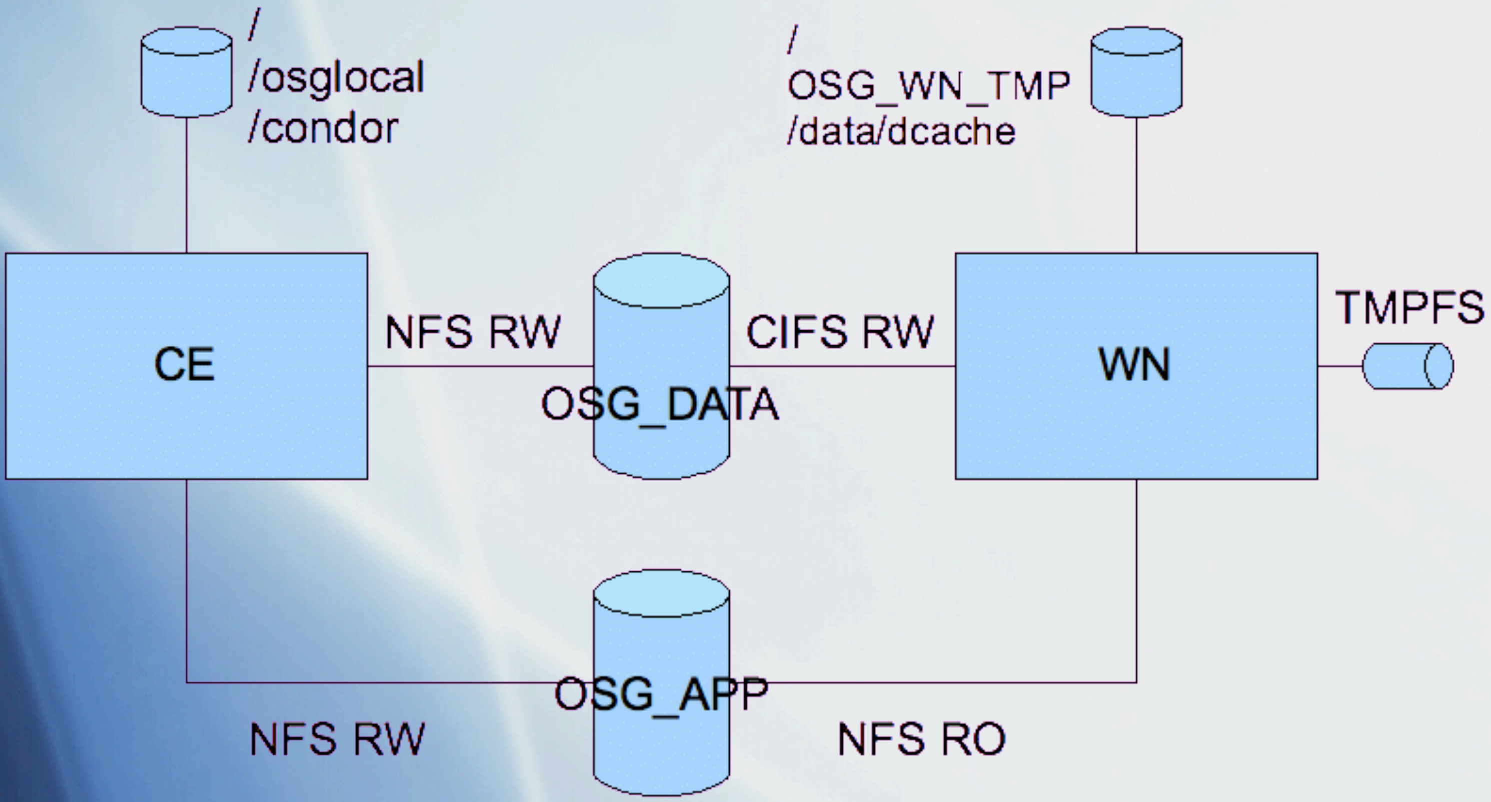
Multiple Spindle Systems

- These include RAID arrays and other multiple disk systems
- Increase throughput by spreading IO operations across multiple disks
- Mitigate parallel access limitations but do not eliminate them
- Do not scale linearly
- Still depend on mechanical hard drives
- Heavily dependent on the OS IO queue being efficient

Memory Cache File Systems

- Relies on available virtual memory capacity
  - VM capacity includes RAM and swap
- Is purged on reboot
- Can be very high performance
- More flexible than RAMDISK
- Potentially Suitable for some Temporary areas
- Can be strictly limited in size

UCSD File System Mounts



NFS Lite

- NFS Lite eliminates a traditional network mount between the WN and the CE
- Relies on the batch system to handle standard IO, scratch contents and proxies
- Currently NFS Lite in OSG only available for condor
- Significantly reduces IO load on the CE
- Deployed in some form at many of the larger OSG sites
- Currently available as an unsupported package in OSG 0.6.0

UCSD CE FSMounts

- Root and /osglocal local file systems
- NFS mounts OSG_DATA (RW)
- NFS Mount OSG_APP (RW)
- 2 - 4 Spindles using RAID1 or RAID5 on CE disk systems

UCSD WN FS Mounts

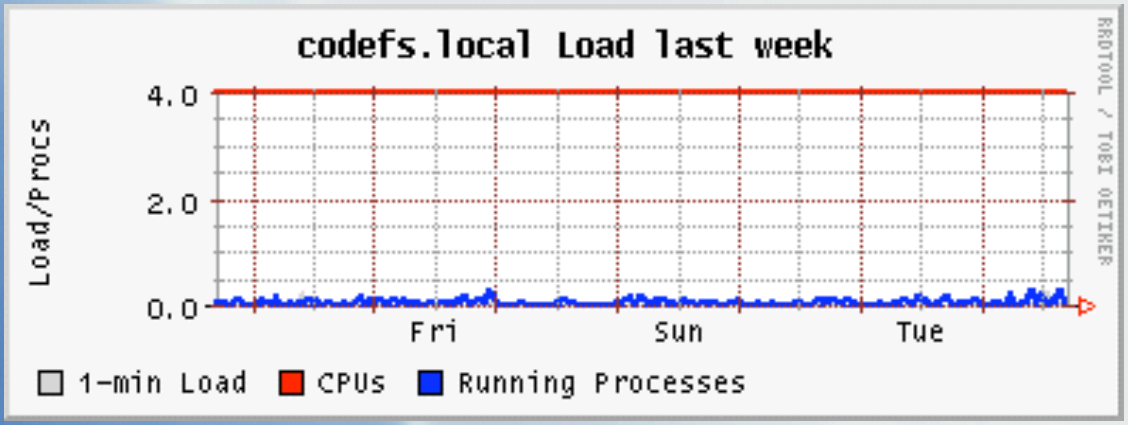
- Local File Systems Root and /state/data which is the local work disk area
- /state/dcache locally mounted for dcache pool usage
- NFS File system OSG_APP (RO)
  - mounted via autofs
- CIFS File system OSG_DATA (RW)
  - custom mount wrapper
- TMPFS file system (replaces /tmp, hard limit 256MB/job slot)
- Majority of nodes use RAID0 Striping of 4 spindles

Network File System Hardware

- OSG_DATA
  - Dual CPU Xeon
  - 1U Chassis
  - 3ware 4 Disk RAID5 array
- OSG_APP
  - Dual CPU Xeon
  - 1U Chassis
  - 3ware 4 Disk RAID5 Array

VO Usage of OSG_APP at UCSD

- Several VO make use of OSG_APP for load install software
- Load is fairly consistent and not generall high
- Local users share OSG_APP with cluster
- OSG_APP typically not loaded



VO Usage of OSG_DATA

- VO typically use OSG_DATA to
  - Stage in data for processing
  - Store interim data files in complex workflows
  - Store final job output for eventual retrieval
- Load is heavily dependent on the particular VO currently running at site
- VO can overload the system we have deployed
- Isolation of OSG_DATA prevents overload from affecting other systems and VO

OSG_APP/OSG_DATA Utilization Experience

- Currently deployed hardware has proven sufficient based on utilization patterns
- OSG_APP is high priority due to heavy use by CMS VO (primary sponsor)
- OSG_DATA is low priority due to light (none) use by sponsoring Vos
- Your site may vary

OSG_DATA Purpose Duplicated by SRM/Dcache

- Both systems provide data stagein/stageout
- SRM/Dcache typically can scale better than typical NFS access to OSG_DATA
  - Comes at the cost of mount point access
- SRM/Dcache can be deployed using a variety of hardware arrangements
  - Fewer large spindle count disk arrays vs many low spindle count nodes

Other OSG_DATA Alternatives

- Depending on sponsor VO needs it may be necessary or desirable to deploy a mountable file system capable of handling parallel access load at the scale of SRM/Dcache
  - Some possible commercial and Open Source options
- Use of high performance networks and direct stage-in and stage-out using VO central store
  - Typically cost efficient
  - Networks handle parallel activity very effectively
  - Does require additional resources on the VO side
  - Can be assisted by squid and other caching technologies
    - Caching works best for small identical data files or application code
    - Proxy can be used to assist OSG_APP as well

Squid Cache

- Squid cache can be used to assist VO to stage some files or data blobs directly to nodes without overload their central servers
- Bypasses site OSG_APP and possibly een OSG_DATA
- Squid itself is very reliable and difficult to overload
  - Tests at UCSD showed even when serving hundreds of parallel files the squid server was stable, the primary limitation was network capacity

WN Local File Systems

- Primarily locally installed hard disk drives
  - Single or multiple spindle arrays
  - UCSD uses multiple spindle RAID 0 arrays for all local FS except for / which is a single disk
- Performance and capacity should match typical VO requirements
  - UCSD deploys 100-150GB/WN shared between the job slots
- Tmpfs may be used to replace some disk file systems.
  - At UCSD each job slot gets their own private /tmp area that is mounted via tmps.

Decisions

- Determine the requirements of sponsor VO
- Determine how your site can support flexibility for additional VO use of the site
  - Can guest VO use sponsor VO storage? Is that desirable?
- Develop strategies for how to isolate guest VO so they do not negatively impact other guest and sponsor VO