CMS without SRM

Brian Bockelman
pre-GDB, October 2012

CMS File Movement

- CMS interacts with storage using three different applications:
  - CMSSW: The primary scientific application framework.
  - PhEDEx: The data management / movement application.
  - WMAgent: Stageout of files

The Trivial File Catalog

- CMS has a database for official files (Data Bookkeeping System, DBS) and a database for file locations (Transfer Management Database, TMDB, part of PhEDEx).
- The TMDB maps sets of files to one or more storage locations.
  - The storage location is an opaque string, typically an endpoint's hostname.
- A separate file, the trivial file catalog, helps map the logical file name to a "physical file name" used for file access.

The Trivial File Catalog

- The TFC is:
  - Maintained by the local site. If the site needs to change server hostnames, they do the change locally.
  - Installed locally in the software area, and synchronized to CMS CVS.
  - A list of rules for manipulating strings based on perl regexps.
- Given a (logical filename, protocol), provides a URL for use.
- Mechanism hasn't changed in about 6 years.

The Trivial File Catalog

- Example rules:
- <lfn-to-pfn protocol="direct" destination-match=".*" path-match=".*/LoadTest07_Nebraska_(.*)_.*_.*" result="/mnt/hadoop/user/uscms01/pnfs/unl\.edu/data4/cms/store/phedex_monarctest/Nebraska/LoadTest07_Nebraska_$1"/>
- <lfn-to-pfn protocol="direct" destination-match=".*" path-match="/+store/(.*)" result="/mnt/hadoop/user/uscms01/pnfs/unl.edu/data4/cms/store/$1"/>
- <lfn-to-pfn protocol="xrootd" destination-match=".*" path-match="/+store/(.*)" result="root://xrootd.unl.edu//store/$1"/>
- <lfn-to-pfn protocol="hadoop" destination-match=".*" path-match="/+store/(.*)" result="/user/uscms01/pnfs/unl.edu/data4/cms/store/$1"/>

CMS without SRM

- The file movement mechanisms in CMSSW and PhEDEx predate SRM.
  - PhEDEx has been heavily customized to work well with SRM.
  - While SRM can be used with CMSSW, precisely zero sites haven chosen to do this. All have preferred to use a LAN protocol (dcap, xrootd, POSIX).
- However, SRM is by no means required. Both PhEDEx and CMSSW can negotiate compatible protocols without SRM.
  - Protocol negotiation between endpoints are done prior to mapping to a URL.

PhEDEx without SRM

- PhEDEx can be run in two modes for file movement:
  - FTS: Submits transfers to PhEDEx. Has all the bells and whistles expected, used by (almost) all sites.
  - local client: Simply invokes a script to do the transfer.
    - Requires site admin to pick the number of files per transfer; timeout semantics are defined by local client. None of the SRM clients do it well :)

Other SRM uses in PhEDEx

- **File deletion**: PhEDEx invokes a user-provided script to delete files before and after transfer. No "overwrite" mode for FTS required.
- **File verification**: Gives a PFN, a file size, and a checksum; the site's script is supposed to verify the PFN.
- None of these callouts have changed in the past 6 years. CMS provides a few default ones - particularly, one which works with SRM.
  - Site picks a script and commits it to their CVS area (CVS not required for functionality, but heavily encouraged).
  - Removing SRM is as simple as providing an alternate script. Many sites have chosen to do this for efficiency reasons.

WMAgent stageout without SRM

- WMAgent provides a list of file transfer commands it implements.
  - To add a new command, it must be put into a WMAgent release - site control is minimal.
    - Expect to do medium-term planning and work.
  - There have been 2 examples of site-contributed stageout plugins.

SRM-free at Nebraska

- The GridFTP protocol provides all the necessary metadata calls (remove, mkdir, stat, checksums) necessary for PhEDEx.
- All our GridFTP servers act as proxy servers for HDFS - they can access any file in HDFS equally. No locality-based optimization.
- Basically, SRM exists to load-balance multiple servers.

Approach

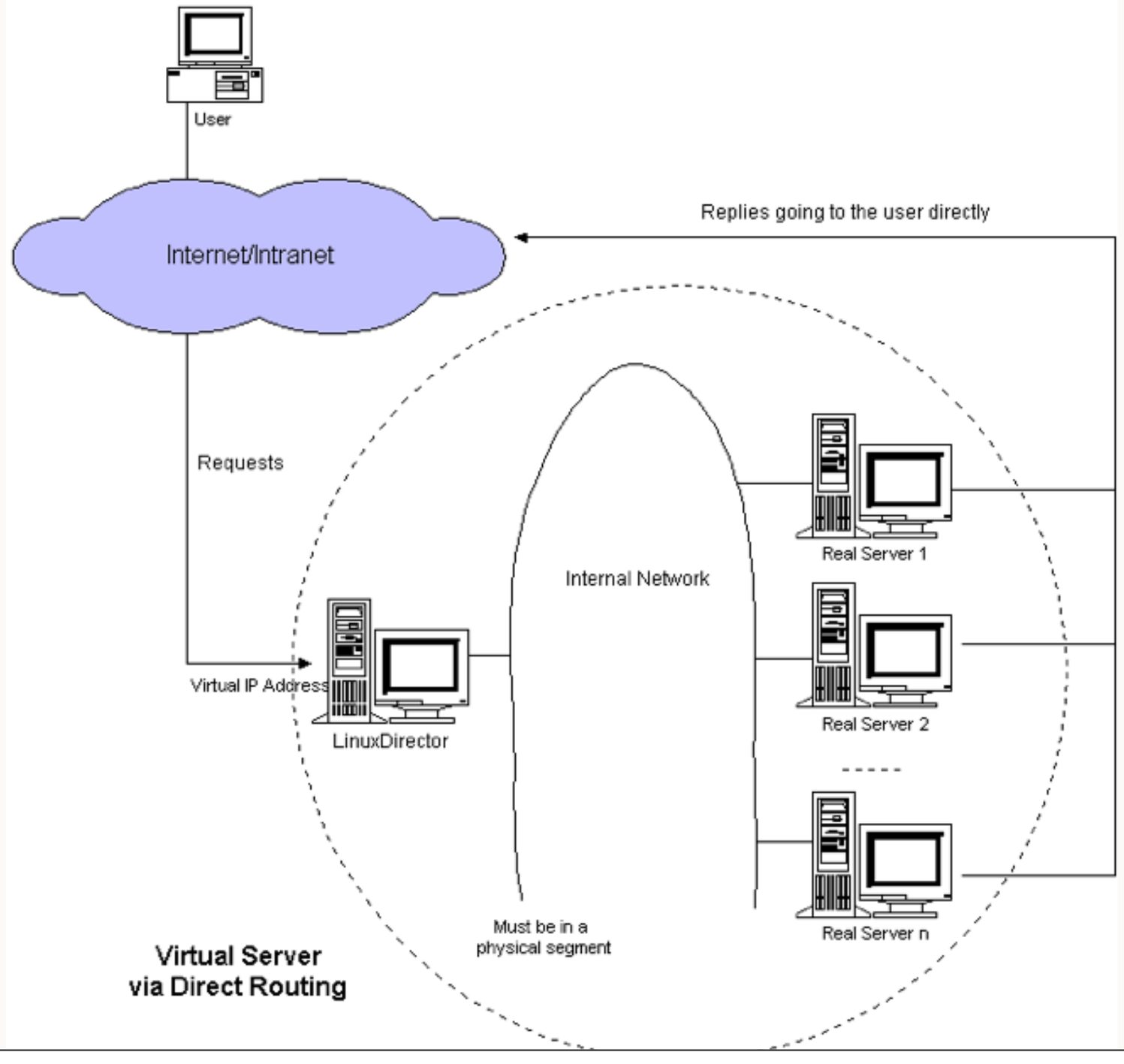
- PhEDEx: We decided to keep the FTS mode - FTS now supports GridFTP fairly transparently.
  - Thoroughly tested SRM<->GridFTP transfers.
- WMAgent stageout: invoke existing GridFTP plugin.
  - (Still needs more testing)
- CMSSW: Uses POSIX. No SRM ever was involved.

LVS-based Load Balancing

- DNS-based load-balancing is hell. For example, to put a server in maintenance, you have to wait for all external name caches to expire.
  - Oh, and CERN's DNS server is broken and may cache locations for 24 hours.
- The Linux Virtual Server project provides several generic mechanisms to do IP-based load-balancing.
  - IP-based load-balancing stays internal to the site, but (basically) requires the servers to be on the same LAN.

In a picture

(from https://indico.fnal.gov/getFile.py/access?contribId=26&sessionId=6&resId=0&materialId=slides&confId=5109)



LVS-balanced GridFTP

- All clients - internal and external - see a single IP address. No one outside the site is the wiser!
  - All servers are heartbeat-tested every 5 seconds. If they fail the test, they are removed from rotation.
- The control channel passes through the active LVS host.
  - The LVS hosts themselves have failover capability.
- The data channel passes directly from GridFTP server to client.
- All GridFTP servers share a hostcert. We do this for 12 servers. "Just Works".

LVS-based GridFTP

- All our GridFTP servers are equal - no particular efficiency by choosing a particular server for a given file.
  - Not true for DPM or dCache where, for example, you may want to select the GridFTP server on the host where the file is served.
  - Unfortunately, in GridFTP v1, the host serving the data is selected *before* specifying which file to transfer. Doesn't work well!
  - In GridFTP v2, the data host can be specified after the filename. dCache has this implemented but Globus GridFTP does not.
    - DPM uses Globus GridFTP, so they can't take advantage of this optimization!

Advertising endpoints

- For FTS to work, we advertise the GridFTP endpoint to the BDII.
  - Basically, copied the entry from CERN EOS.
- For PhEDEx, when "srm" protocol is requested, we return gridftp://-based URLs.
  - The CMS protocol name is an opaque string; nothing cares if you return a different protocol.
  - Again, all in the control of the site. Do not need to involve anyone else.

Comparing Performance

- Really, nothing much to compare here.
  - We have eliminated the overhead of SRM (admittedly rather small).
  - SRM is hardcoded to round-robin GridFTP servers - but there's only one GridFTP server in the rotation!
- Could peg the 10Gbps line before; can peg the 10Gbps line afterward.
- Have run GridFTP-only in the Debug instance for several weeks.
  - Have run LVS-load-balanced servers since March 2012.

Lessons

- In theory, CMS relies on no SRM-specific features. Sites wanting to get rid of it should realize it might take awhile to find all the "bugs" (i.e., WMAgent stageout).
  - SRM has some great features, but their usage must be implemented by sites. Sites can utilize these as desired or ignore as desired.
- This setup works well because GridFTP is well-supported in FTS and the site controls the relevant settings.
  - Once Xrootd has better FTS support, we could use this too.
- LVS approach is not as efficient for DPM sites. Should work for dCache (maybe with some elbow grease!).
  - DPM needs an additional feature from Globus GridFTP server. Globus folks appear somewhat interested in adding this. I have commit access to Globus, if necessary. :)