Cloud and local software deployment using CVMFS, EasyBuild, and Lmod

Eric Appelt[1], Andrew Melo[1], Davide Vanzo[1]
[1]Vanderbilt University, Nashville, TN 37203

Introduction

Installing and supporting scientific software for users of a HPC system is a time consuming and complex task. EasyBuild and Lmod are community-driven tools commonly used in combination to automatically install software and generate environment modules accessible to users in a hierarchical structure[1]. These compiled modules are typically installed on a network file system such as GPFS[2].

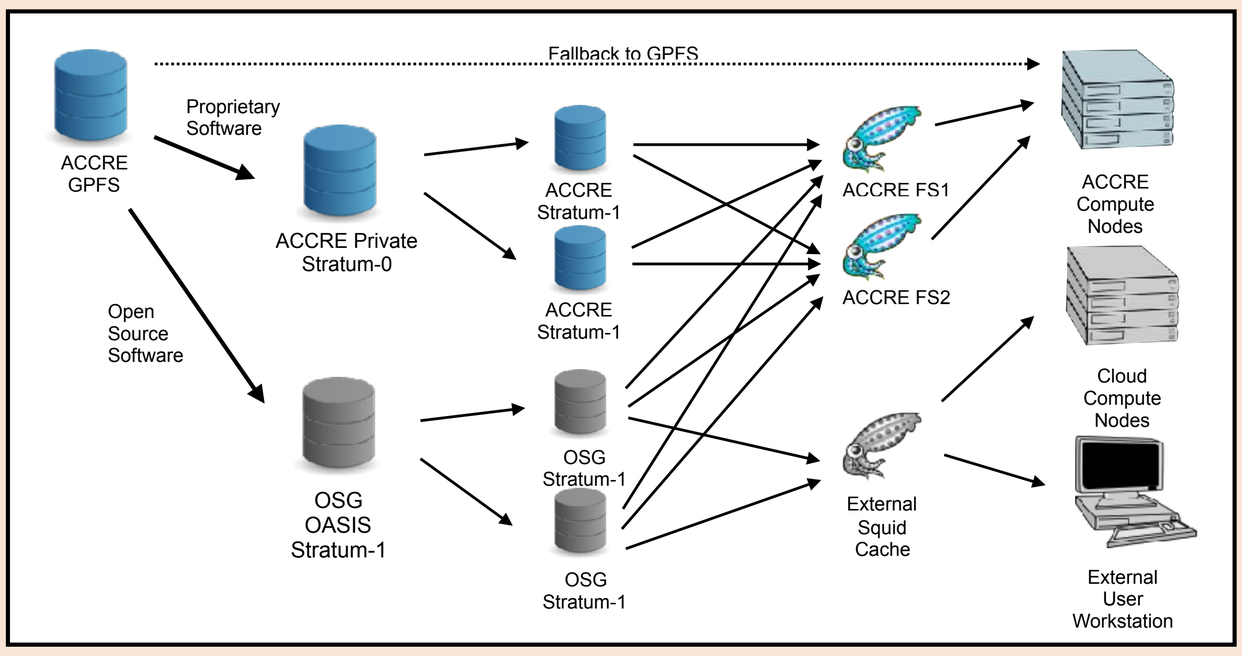
Recently, user interest has been shown for interactive and graphical usage of this software on cluster resources as well as external cloud or desktop availability of optimized software builds. Both these capabilities present challenges for networked file systems. Interactive users are very sensitive to initial software load times, which may be unsuitably slow over a network file system due to the necessity of reading and re-reading a multitude of tiny files. Delays due to local cache misses which may have been unnoticed in batch scheduled jobs may present an unacceptable wait time for the researcher. Allowing cloud or desktop access to network filesystems for researchers presents a serious administrative challenge and may be completely prohibited by privacy or export restrictions. Repackaging or installing a scientific software stack on individual compute resources or containerization also imposes a significant workload on an administrative team. The goals required to meet these demands can be summarized as follows:

- **Deploy hierarchical software modules as simply as placing them on a network filesystem**
- **Aggressively cache and avoid constraints of read-write network filesystems**
- **Send data to compute nodes on-demand, avoid local storage requirements**
- **Allow convenient access to the software data to external or cloud nodes**
- **Operationally decouple from read-write network filesystem**

CVMFS[3] is a network filesystem designed specifically for distribution of scientific software and commonly used in high energy physics collaborations, such as those at the LHC[4]. It is implemented as a POSIX filesystem in user space using a FUSE module and serves files and directories over outgoing HTTP connections. Data is transferred on demand and verified by cryptographic hash. Aggressive caching and content-addressable storage take advantage of the immutable nature of this data optimizing for the frequent reads of many small files often accessed as a group.

We present the use of CVMFS at Vanderbilt's ACCRE facility, where the scientific software stack built with EasyBuild and available to users via Lmod contains both open source and proprietary code. A symlink scheme has been developed to allow for external cloud nodes to access only the open source portion of the stack hosted on public OSG OASIS[5] servers, while a private ACCRE CVMFS instance allows only internal access to proprietary code. Additional redirection allows for individual nodes to access software specifically compiled for their CPU architecture family. This framework has allowed us to meet the above goals at our facility.

Software Distribution Architecture



Previous to the use of CVMFS in conjunction with Lmod and EasyBuild at ACCRE, Scientific software was built on the cluster in parallel on a set of nodes each with a different CPU architecture family and stored in the GPFS filesysem. The software stack is configured to reside in an /accre base directory with subdirectories configured for software optimized for various CPU families. Configuration management software ensures that each compute node is linked to the software built against its particular CPU family.

With the introduction of CVMFS, the workflow of building software is unchanged, and ACCRE compute nodes can still access the software stack through GPFS as a fallback. Configuration management software will instead check to ensure that CVMFS is functional on the node and that the required ACCRE repository within OSG OASIS is available, and then move as necessary symlinks within /accre to point to the same files hosted by CVMFS. Should CVMFS become unavailable on the node, the links will revert to the identical files hosted on GPFS.

To synchronize software from GPFS to CVMFS after builds are complete, a script is triggered that will copy the new builds to CVMFS stratum-0 servers. For the open source software, it is written to the ACCRE repository within OSG OASIS. Proprietary software is synchronized to a private ACCRE CVMFS instance not accessible outside the internal network. The directory structure remains unchanged from the GPFS copy from the viewpoint of the client, as symlinks are created in the OSG OASIS repository to point to proprietary software within the private ACCRE repository.

To provide fault tolerance and handle client load, a set of stratum-1 CVMFS servers may be configured to synchronize with the stratum-0, providing load balancing and fault tolerance. Finally, a set of SQUID[6] caching proxy servers are configured to proxy requests from ACCRE compute nodes to the respective CVMFS servers.

The above diagram shows how ACCRE compute nodes access both open source and proprietary software from CVMFS or GPFS, and how external cloud compute nodes or user workstations can access only the open source software from outside the ACCRE network.

Interactive Performance

Interactive cluster usage places new demands on filesystem performance in the initial loading of scientific software files. While GPFS performance has been suitable for batch jobs on ACCRE, demands on a read-write filesystem will often lead to delays on the order of a few seconds to a few minutes for small-file data not yet cached on an individual compute node. These delays may be effectively unnoticed for batch submissions for which delays in scheduling are often much longer than delays in initial software loading. For interactive usage, such as that provided by the OSC Open OnDemand visualization portal[7], these delays are often unacceptable to the researcher, resulting in help desk requests and additional work for cluster administrators.

In the table below, the difference in performance of initial loading times of three popular software frameworks that are commonly used interactively is shown for a node with and without data locally cached. For python, the Anaconda distribution was used with an import of the numpy, matplotlib, and pandas libraries. For MATLAB and R, the software was started to run a no-operation script. In each case the wall clock time of the process is shown. In each case, the performance of GPFS and CVMFS is comparable for the warm cache. However, when a cache miss is encountered, the difference is dramatic, with software load times increased by approximately 20 to 50 times for GPFS. In the case of slow loading packages such as MATLAB, this delay was considered unacceptable by our clients.

STARTUP TIME (seconds)		Python	MATLAB	R
Cold Cache	GPFS	89.474	391.830	10.234
	CVMFS	1.828	18.359	0.389
Warm Cache	GPFS	0.640	7.960	0.148
	CVMFS	0.789	8.755	0.215

Cloud Deployment

Proof-of-concept cloud deployment of the open source portion of the software stack was tested on an Amazon EC2 t2.medium node with 2 virtual CPU cores and 4 GB of memory, using a stock CentOS 7 base image provided by Amazon. Installation of CVMFS was performed using the OS package manager and the CVMFS configured in just a few lines to connect to OSG OASIS directly. For larger cloud deployments, additional SQUID proxy servers should be configured.

Lmod 7.8 was installed manually and the appropriate symlinks created for the CPU architecture of the node. This process may be performed in a few minutes. Once configured, the software stack is accessible via the command line exactly as in the ACCRE cluster, except that the proprietary software modules are neither available nor displayed.

With a simple installation script, a new virtual machine could be provisioned and ready on AWS in under 10 minutes!

Conclusions

The use of EasyBuild and Lmod on the ACCRE cluster GPFS filesystem has already provided users with a easily accessible hierarchical system of optimized software modules, and is manageable by a small team of administrators and developers. However, demands for external cloud usage and interactive visual platforms have necessitated a more robust solution for distributing software modules. Our initial experience moving the compiled software stack to CVMFS has resulted in dramatically reduced load times for un-cached software modules, and allowed for use of non-proprietary software on compute resources outside of the ACCRE cluster.

The operational experience at ACCRE with hosting software modules over CVMFS has so far shown this to provide better consistency, reliability, and improved cloud capability over traditional deployment to read/write network filesystems. While Lmod and EasyBuild have commonly been used in combination in HPC environments, here we suggest that the further combination of Lmod, EasyBuild, and CVMFS forms a complete solution for managing and distributing scientific software to cluster users and allows users to also access the same software configuration from external workstations or other resources.

REFERENCES

[1] M Geimer, K Hoste, and R McLay; 2014 HUST workshop, "Modern Scientific Software Management Using EasyBuild and Lmod"
[2] F Schmuck and R Haskin; 2002 FAST Conf. Proc., "GPFS: A Shared-Disk File System for Large Computing Clusters"
[3] J Blomer et al.; 2011 J. Phys.: Conf. Ser. 331 042003, "Distributing LHC application software and conditions databases using the CernVM file system"
[4] L Evans and P Bryant; 2008 J. Inst.: 3 S08001, "LHC Machine"
[5] B Bockelman et al.; 2014 J Phys.: Conf. Ser. 513 032013, "OASIS: a data and software distribution service for Open Science Grid"
[6] D Wessels and K Claffy; 1998 IEEE J Sel. A. Comm.: 16 345-57, "ICP and the Squid web cache"
[7] D Hudak et al.; 2018 J Open Source Software: 3(25), 622, "Open OnDemand: A web-based client portal for HPC centers"

FURTHER INFORMATION

See https://www.vanderbilt.edu/accre/cs18/cvmfs or scan the QR code at the bottom right.