CLUSTER ARCHITECTURES USED TO PROVIDE CERN CENTRAL CVS SERVICES

M. Guijarro, S. Lopienski, I. Reguero, CERN, Geneva, Switzerland

Abstract

There are two cluster architecture approaches used at CERN to provide central CVS services. The first one (http://cern.ch/cvs) depends on AFS for central storage of repositories and offers automatic load-balancing and fail-over mechanisms.

The second one (http://cern.ch/lcgcvs) is an N + 1 cluster based on local file systems, using data replication and not relying on AFS. It does not provide either dynamic load-balancing or automatic fail-over. Instead a series of tools were developed for repository relocation in case of fail-over and for manual load-balancing.

Both architectures are used in production at CERN and project managers can chose one or the other, depending on their needs. If eventually, one architecture proves to be significantly better, the other one may be phased out. This paper presents in detail both approaches and describes their relative advantages and drawbacks, as well as some data about them (number of repositories, average repository size, etc).

INTRODUCTION

What is CVS

CVS, or the Concurrent Versions System [1], is the dominant open-source version control system. CVS allows developers of software projects to store their source code, to keep track of different versions (revisions) of a file and changes done by fellow developers, and to work on the same file concurrently. It is used world-wide by thousands of open-source and commercial projects, being useful for everyone from individual developers to large, distributed teams.

A CVS repository (an equivalent of a project) contains modules, which in turn have subdirectories and files. Basic CVS operations are import (adding a new module), check-out (downloading files from a repository to a local working directory) and commit (checking-in locally modified files back to a repository). One can also tag (attach a tag to files in the repository), add and remove files from an existing module, view differences between file revisions etc.

There are CVS servers and clients available for many platforms; CVS clients are often built-in to IDEs [2], like JDeveloper, Eclipse etc.

Software development at CERN

As we prepare for the LHC (Large Hadron Collider) era, there is a lot of software development being carried on at CERN. Accelerator groups are preparing software for Controls and Operations. Physicists are writing programs to analyze data coming from detectors (Atlas, Alice, CMS and LHCb). And IT engineers are developing a software and hardware infrastructure to handle the challenges (data storage and computing needs) that will come with LHC. The latter includes the LCG (LHC Computing Grid project) and EGEE (Enabling Grids for E-science in Europe) project. For all these software projects CVS is a perfect tool to store, maintain and share source code.

CENTRAL CVS SERVICE

The project to set up a Central CVS Service was started in early 2002. The requirements were collected and several possible designs discussed. Reliability was one of the key requirements to be met, along with availability of a variety of authentication methods and platform independence.

Architecture

It was decided to set up a cluster of several machines (initially two, now four PCs running Linux). Each node of the cluster runs CVS server software (and other software, like a web server, etc.). There is a DNS alias, isscvs.cern.ch, which points to one of the machines in the cluster. The alias is refreshed (and possibly changed) every 30 seconds - a machine that is available and least used is chosen every time to be addressed by the alias (see Figure 1).

The repositories are stored centrally, on AFS, and are available from all cluster members. This solution ensures data consistency even if several consecutive operations on a given repository are performed on different nodes of the cluster. Data consistency could otherwise be a major problem, requiring some kind of a data replication mechanism to be put in place.

Features

This type of architecture provides the following features: load-balancing and automatic fail-over. Both are offered gratis with the DNS alias mechanism: when a machine is down, or more busy than other nodes in the cluster, it won't be addressed by the alias. This mechanism has proven to be very reliable and easy to administer.

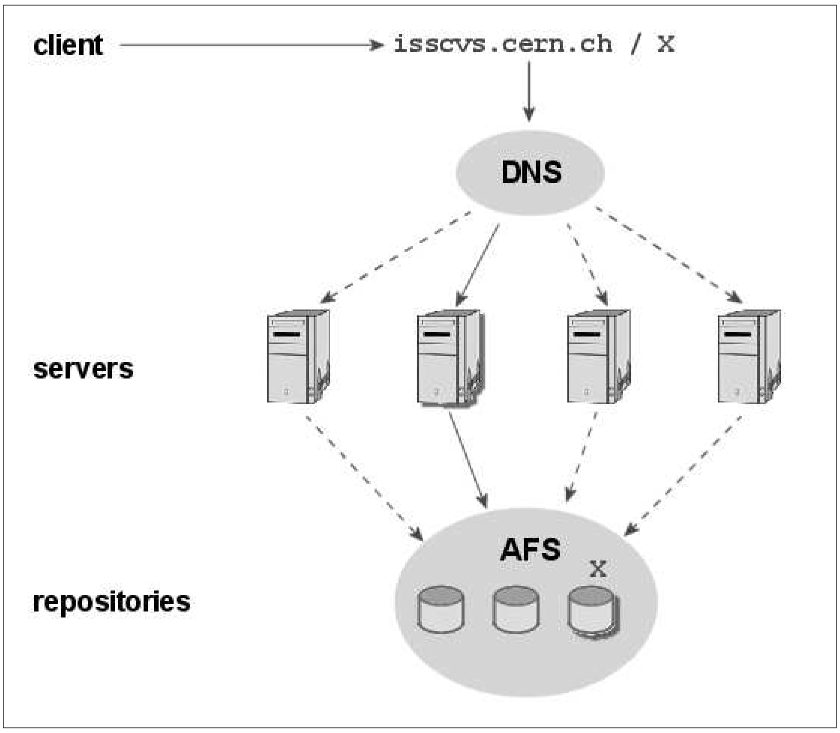

Figure 1: Architecture of the CERN Central CVS Service.

Dependence on AFS and DNS is a clear drawback of the architecture - if either of these systems stops working, users won't be able to access their repositories. Nevertheless, AFS is a core part of the computing infrastructure at CERN, working stably and securely over years, and as such can be assumed as reliable. Also, DNS is used implicitly by almost any every-day networking software, so using it for the CVS Service can't be regarded as a great risk.

The service has also a group of web tools implemented in PHP that allow users to request a new repository, change parameters of an existing one, and access contents of a repository (CVSWeb and ViewCVS web interfaces). Full user documentation is available on the Service web page [3].

CVS SERVICE FOR LCG

Architecture

LCG (LHC Computing Grid) project asked for a CVS Service that would be independent from AFS. Following this request, a dedicated non-AFS CVS service was designed and implemented for LCG-related projects. It uses an N + 1 cluster architecture (Figure 2). Each CVS repository is hosted on one of N members of the cluster. An additional server (called slave server) keeps an up-to-date copy of all CVS repositories. The periodic update is done using CVSUP (a public-domain CVS-optimized Network File Distribution System [4]).

Each repository is accessed via a unique DNS alias (*projectname*.cvs.cern.ch), which points to the server currently serving it. Therefore, users are unaware of the physical location of the repository they use. In case of failure of a cluster member, all repositories hosted on that machine can be relocated to another one, using the copy from the slave-server. The relocation operation, among other things, modifies the DNS tables, so that address *projectname*.cvs.cern.ch is redirected to the new node serving that repository. This operation is manually triggered through a Web interface by service administrators.

Data replication keeps the slave-server heavily busy and causes some overload in both the network and each cluster member. Regardless of the replication frequency, there is always a risk of data loss in case of failure. Since most CVS operations are read operations, above-mentioned problems are minimized and drawbacks of data replication are acceptable.

Features

Having CVS repositories stored in a local file system offers best possible performance for single CVS accesses.

As for the Central CVS service, ViewCVS access and a set of CVS web tools are provided. Additionally, a Web SOAP interface has been developed for project relocation. CVS Service for LCG has its dedicated webpage with full documentation [5].

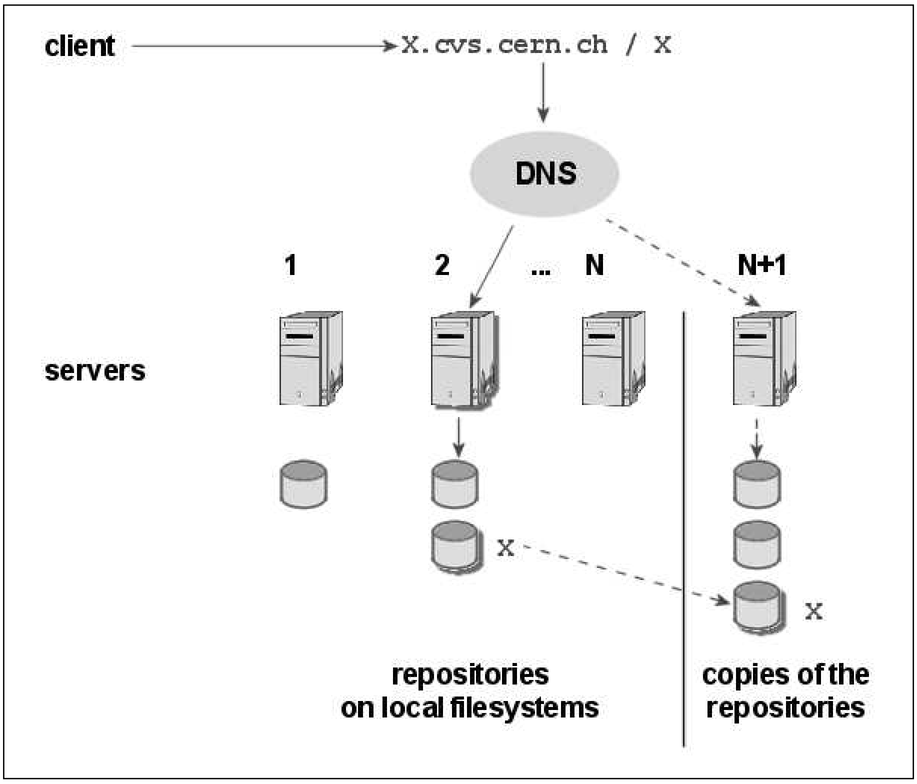

Figure 2: Architecture of the CVS Service for LCG.

COMPARISON

From the implementation point of view, the CVS Service for LCG required additional infrastructure to be created (as compared to the Central CVS Service), such as:

- Web tool for project relocation
- data replication system
- NFS server setup to export CVS repositories and users home directories

The two-year experience of providing both services shows that server failures are practically transparent for the Central CVS Service users. In case of the CVS Service for LCG, interruptions are visible by the users. On the other hand, the CVS Service for LCG has never been affected by any of the very few AFS service interruptions at CERN. Also, in current configuration and usage of both services, CVS Service for LCG is in general faster, since it is based on local file systems.

Availability logs show that both services have provided high (over 99%) availability, but the CVS Service for LCG has had a 20% higher down time than the Central CVS Service.

The Central CVS Service hosts (at the time of writing) 81 projects with a total of over 10GB of source code, and has around 780 developers with write access. The CVS Service for LCG is smaller: it hosts 23 projects with a total of 4.3GB of source code, and has around 250 registered developers. The number of projects in the Central CVS Service increases at pace of 4 new projects per month, while the number of projects in the CVS Service for LCG has been stable during last year.

SUMMARY AND CONCLUSION

There is no winning architecture to implement a CVS service. Each site should select an architecture to implement a CVS service based on already existing infrastructure. For sites where a solid and well supported Distributed File System service is already in place and DNS redirection is possible and can be done in a dynamic manner, it is worth trying to implement the Central CVS Service architecture. At CERN, it has been done using AFS, but it could also work using NFS, Samba, GFS or others.

On sites where there is not a reliable Distributed File System, the N + 1 cluster architecture is the one to be selected. N + 1 architecture can be improved by implementing:

- multiple slave-servers each of them replicating a subset of repositories,
- automatic slave-server relocation: in case of slave-server failure any other member of the cluster starts acting as slave-server.

In both architectures, the number of nodes should vary depending on the number of repositories and access rate.

ACKNOWLEDGEMENTS

The authors would like to thank Arash Khodabandeh and Helge Meinhard for ideas and feedback concerning the setup of the Central CVS Service. Gunter Folger, Andreas Pfeiffer, Werner Jank, Hans-Peter Wellisch and Sergei Sadilov provided valuable user requirements. Also, ideas, help and user support of both CVS services offered by Philippe Defert are much appreciated.

REFERENCES

[1] Concurrent Versions System http://www.cvshome.org

[2] Integrated Development Environment

[3] CERN Central CVS Service: http://cern.ch/cvs

[4] CVS Service for LCG: http://cern.ch/lcgcvs

[5] The CVS-Optimized General-Purpose Network File Distribution System: http://www.cvsup.org