Project Closure Report

Frontier Distributed Database Caching System (July 2006 - June 2013)

Table of Contents

1. Approvals

CMS Representative:	**Signature:**	
	Print Name:	Liz Sexton-Kennedy
	Title:	
	Date:	
Sponsor:	**Signature:**	
	Print Name:	Lothar Bauerdick
	Title:	
	Date:	
Project Leader:	**Signature:**	
	Print Name:	Dave Dykstra
	Title:	
	Date:	

2. Document Change Log

Revision	Date	Change Description	Prepared By	Approved By
V 1.0	9/17/13	First complete version	Dave Dykstra	

3. Project Abstract

The Frontier Distributed Database Caching System supplies detector calibration and other conditions data from central databases to millions of computing jobs, via http proxy caches, so the jobs can use the data to properly analyze separately-supplied collision events. During the July 2006 to June 2013 phase of the project, the project team made the application and operational systems ready for production use by all computing jobs in the CMS experiment and by most of the computing jobs in the ATLAS experiment, and made ongoing fixes and enhancements to both the application and the operational systems software. The operational systems included managing a network of http proxy cache servers at all Worldwide LHC Computing Grid (WLCG) sites worldwide that support the ATLAS and CMS experiments. The team also helped the proxy caches to become an essential part of other grid-wide applications, most notably the CernVM FileSystem (CVMFS). In addition, the development team did a majority of the operations of the CMS central servers and the CMS+ATLAS monitoring servers for most of the time period, and that has now transitioned to a separate Operations team.

4. Project Documentation

This project was not managed as a formally planned project with an itemized deliverable list, so there are no project planning documents. Rather, the project was managed with an agile methodology where priorities were dynamically determined as the project progressed, and progress reports were sent regularly to management. The goal of the original project as written in CS-doc-790-v1, a presentation made for the International Symposium on Grid Computing 2005, was to “assemble a toolkit, using standard web technologies, to provide high-performance, scalable, database access through a stateless, multi-tier architecture.” The Frontier application was first put into production for CDF before the period covered by this project closure report document, and an early demonstration version for CMS was already in place at the beginning of the time as well, but the system required a lot of improvements for robust production operation by CMS and later ATLAS.

5. Supporting Documentation

Most of the supporting documentation is available on the project web site,
http://frontier.cern.ch

In addition, the following papers were published for the project:

- B Blumenfeld, D Dykstra, L Lueking, and E Wicklund, "CMS Conditions Data Access using FronTier", CHEP 2008 *J. Phys.: Conf. Ser.* **119** 072007, http://dx.doi.org/10.1088/1742-6596/119/7/072007
- D Dykstra and L Lueking, “Greatly Improved Cache Update Times for Conditions Data with Frontier/Squid”, CHEP 2010 *J. Phys.: Conf. Ser.* **219** 0720342009, http://dx.doi.org/10.1088/1742-6596/219/7/072034
- D Dykstra, “Scaling HEP to Web Size with RESTful Protocols: The Frontier Example”, CHEP 2011, *J. Phys.: Conf. Ser.* **331** 042008 http://dx.doi.org/10.1088/1742-6596/331/4/042008
- D Barberis, et. al., “Evolution of grid-wide access to database resident information in ATLAS using Frontier”, CHEP 2012, *J. Phys.: Conf. Ser.* **396** 052025 http://dx.doi.org/10.1088/1742-6596/396/5/052025

- B Blumenfeld, et. al., "Operational Experience with the Frontier System in CMS", CHEP 2012, *J. Phys.: Conf. Ser.* **396** 052014 http://dx.doi.org/10.1088/1742-6596/396/5/052014
- D Dykstra, "Comparison of the Frontier Distributed Database Caching System to NoSQL Databases", CHEP 2012, *J. Phys.: Conf. Ser.* **396** 052031 http://dx.doi.org/10.1088/1742-6596/396/5/052031
- Abstracts entitled "Security in the CernVM File System and the Frontier Distributed Database Caching System" and "Squid monitoring tools – a common solution for LHC experiments" have been accepted to CHEP 2013 and will be in its proceedings

6. Reason for Closing the Project

This system is still in active use, and ongoing maintenance of the application software is still expected, but at this point the responsibility for the operations and for maintenance of the operations systems has been turned over to a separate CMS Computing Operations team. The functionality goals for Frontier have mostly been reached, and the remaining features are going to be added during maintenance, as detailed in section 12 below.

7. Project Schedule

Since this project was managed with an agile methodology, there was no planned schedule for specific deliverables beyond having the system ready for production when the LHC experiments began taking data. Instead, the team continuously reevaluated priorities and implemented the things that they deemed to have the most positive impact using the available resources.

8. Project Accomplishments

Below is a summary of the major accomplishments of the project. There are more details in the release notes of some parts online:

- Frontier client: http://frontier.cern.ch/dist/clientreleasenotes.txt
- Frontier servlet: http://frontier.cern.ch/dist/servletreleasenotes.txt
- Squid tarball: http://frontier.cern.ch/dist/frontier-squid-releasenotes.txt
- Squid rpm: http://frontier.cern.ch/dist/rpms/frontier-squidRELEASE_NOTES

Dates	Accomplishments
7-06 to 6-07	Made 11 Frontier client releases and 8 Frontier servlet releases, including adding compression of responses, adding a feature to configure all options through a single string passed in from higher layers, replacing forced cache refreshes from every client with a "short" cache time (queries identified by client but time defined by servlet), re-using TCP connections for multiple queries, and reducing timeouts for faster recovery but allowing long queues for the database by adding "keepalive" messages from servlet to client.
7-06 to 6-07	Integrated with higher levels of software that used Frontier, including supplying 6 updates to CMS software related to Frontier configuration. Also contributed patches to CORAL to improve Frontier integration and performance, and to work around crashes.
6-07	Added a feature to the frontier-squid package so it could run more than one squid on the same machine on the same network interface (squid2 is single-threaded). The feature required making a patch to the original squid package;

	the patch was submitted and accepted by the open source project.
9-06 to 6-07	Greatly improved the performance of loading conditions data, including documenting 10 significant improvements and 6 studies at https://twiki.cern.ch/twiki/bin/view/CMS/FrontierPerformanceImprovements The throughput at the beginning was very low with even a modest number of clients, but by the end of this period, demonstrated 430 MBytes/sec throughput sustained from two squid machines each with two 1 Gbit/sec bonded interfaces and two squid processes. Identified cases where CMS programmers had used a badly performing data structure that resulted in over 27,000 queries per job; when those were fixed all those were queries were reduced to only 3.
1-07 to 12-07	Participated in the CMS code performance improvement task force, attending most of the meetings and supplying suggestions.
7-06 to 6-07	Upgraded and/or installed the machines used for operating and monitoring the Frontier service, including 8 servers at CERN and 2 servers at Fermilab by the end of the period. On the CERN servers, managed not only the application but also the operating system by owning the cluster configuration templates and reorganizing them for easiest management of common components. Upgraded the older machines from SLC3 to SLC4 to match the newer machines.
5-07 to 4-08	Made a squid rpm for all CMS Online High Level Trigger worker nodes that automatically configured itself into a hierarchy so each node sent data to (typically) four other nodes. In that way, all 640 nodes (each with 4 or more jobs/cores) were able to load all the conditions data simultaneously in less than the required 60 seconds. Also made a tomcat rpm for installing the Frontier server software on two nodes to read from a local Oracle database. Tested all the installations and did performance measurements. These rpms enabled re-use of all the same code for conditions loading from Offline into the Online environment, compared to ATLAS which uses a completely separate implementation developed from scratch using considerably more effort.
12-07 to 2-08	Made a prototype implementation extending CORAL, Frontier client, and Frontier servlet to support writes to the database, authenticated with grid proxies and encrypted with https, in addition to the normal (cached) reads. This was not expected to be high performance but it was intended to be a replacement for CERN IT plans to implement a separate CORAL proxy server. Performance measurements showed it to be twice as slow as directly writing to Oracle on local connections, but long distance (Fermilab to CERN) it only slowed down another factor of two where Oracle slowed down a factor of ten.
7-07 to 6-08	Made 4 Frontier client releases and 3 servlet releases, including adding client-based load balancing, improving the retry strategy after failures, improving memory usage, and adding a local client cache to eliminate duplicate queries from the same client (a temporary measure that was removed later when the source of the duplicates was removed).
7-07 to 6-08	Contributed several more improvements to the CMS software including eliminating use of POOL “catalogs”, which were a major maintenance headache. Also contributed more fixes and performance improvements to CORAL.
7-07 to	Analyzed and diagnosed many problems that occurred in the Frontier

6-08	infrastructure, several of which were traced to more cases of CMS programmers introducing new uses of the badly performing data structures. Several other problems were cache coherency issues caused by data that changed in the database after it was cached, against the rules. That inspired making detailed plans for a new cache coherency mechanism to reduce the cache expiration times without overloading the infrastructure.
7-07 to 6-08	Continued to manage CMS Frontier server operations at CERN, including adding 5 new Frontier-related servers and decommissioning others. One of the new servers was frontier.cern.ch, a web server for central monitoring of squids and Frontier, and a source of information about the project.
7-08 to 6-09	Made numerous significant improvements to the Frontier client and servlet, including 3 client releases and 7 servlet releases. The most major addition was the implementation and rollout (for both CMS Offline and Online) of a new cache coherency mechanism based on modification times of database tables and the http header If-Modified-Since. This greatly reduced the amount of time it took to propagate changes out to all Offline sites without overloading servers, while at the same time improving performance. It also significantly reduced the average Run Start time for CMS Online. Other major improvements to the servlet included two fixes for automatic recovery from situations that sometimes occurred when a database went down. The main improvements to the client included drastically reducing the number of large memory allocations it did, improving the retry algorithm, making it re-lookup DNS names every 5 minutes, working around a problem found when CMS Online moved up from 4 to 7 jobs starting at once on their worker nodes, and eliminating the huge performance hit when writing debug log files to AFS.
9-08 to 6-09	Helped ATLAS begin to use the Frontier system for loading conditions to analysis jobs at Tier 2s, which gave them vastly improved performance over the direct access to Oracle at Tier 1s they had been planning. This included fixing small bugs in CORAL and Frontier client that only showed up for their application. Gave much advice on how to configure their systems, and taught one of their developers how to modify the Frontier servlet so he could begin to develop some small enhancements that they needed there.
9-08 to 6-09	Trained and provided ongoing advice to a CMS facilities operations person (a student) at CERN that administered the Frontier service part-time.
7-08 to 6-09	Analyzed and diagnosed many problems that occurred in the CMS Offline and Online Frontier infrastructures. The facilities operations person was not able to take care of non-routine problems. Discovered one problem with the Online system during the “pajama party” celebrating the first collisions in September 2008, but then damage to the collider shut it down for another year. Meanwhile the Frontier systems were all in production, processing cosmic ray and simulated (Monte Carlo) data.
7-08 to 6-09	Continued to administer CERN Frontier computers, including bringing up 6 new virtual servers (the first at CERN for VO-specific services) on two physical machines, and decommissioning 6 other servers. Set up one pair of virtual servers as a highly available pair serving frontier.cern.ch. The total number of Frontier-related CERN servers administered at that point was 12: three production “launchpad” servers, one extra for “backup production” and development, two for development, two for monitoring, and four for production squid proxies serving CERN worker nodes.
7-09 to	Maintained the Frontier client and servlet software, including 4 client releases

6-10	and 5 servlet releases. The most significant changes on the client side were adding a new option for guaranteeing freshness of data (implemented for CMS Prompt Reconstruction) and adding support for forking of the parent process after conditions are loaded (needed first by CMS Online but both CMS & ATLAS Offline were also planning to use it). The most significant changes in the servlet were adding support for tracking modification times on all tables in multi-table queries (needed by ATLAS and the majority of the change was contributed by their developer), adding an option to send keepalive messages to clients while waiting for the database to execute long queries (also needed by ATLAS), and fixing a race condition that was causing timer threads to not get stopped properly. Many of the changes required extensive investigations to find out what the root cause was.
7-09 to 6-10	Did extensive work on squid and tomcat packages. Made a major revision to the Frontier squid distribution tarball package plus two follow-on minor releases. The biggest change was the addition of a mechanism to programmatically edit the squid configuration file so customizations do not have to be manually reapplied every time there is an upgrade. Fixed two more bugs in squid itself and submitted them to the squid package owners on the internet. Provided detailed consulting and testing for a Frontier squid rpm that someone from CERN IT (working for ATLAS) made from the Frontier squid tarball distribution, for general use by ATLAS & CMS Offline sites. For CMS Online, released two new versions of their custom squid rpm and four new versions of their custom tomcat rpm. Also created a a new distribution of the client side of the squid log-analyzing awstats software for use by ATLAS Offline Frontier “launchpad” servers to upload processed log (usage) data to the shared frontier.cern.ch web server.
7-9 to 6-10	Helped ATLAS put their Frontier network into production, and continued to meet with them weekly and advise them on improvements they needed and issues they had. Also answered numerous questions from CMS users of Frontier; that is, provided ongoing support.
7-9 to 6-10	Continued to do system administration of CERN Frontier-related servers because CMS Facility Operations still didn't have anyone able to take over. Created the first CERN CMS cluster management templates for SLC5 machines (which were drastically different than for SLC4 machines), and brought up the first SLC5 machine and the first of each application type. Wrote instructions for the CMS Facility Operations person that supported Frontier part-time to upgrade the rest of the machines. The warranty on many of the machines also expired so they were moved to new physical hardware. Virtual machines weren't yet supported on SLC5 so the 6 VMs were converted to 6 real machines. Also did most of the Frontier software upgrades of squid and tomcat servlets on the CMS Frontier Offline production servers, including adding a couple new servlets.
2-10 to 6-10	Began to support a new use of the squids deployed for Frontier: the CernVM FileSystem (CVMFS). Provided consulting for how best to configure the Frontier squid package for CVMFS, and for avoiding squid pitfalls in the CVMFS client. Wrote software to daily generate a web page list of squids at CMS sites, for CernVM to use to try to automatically locate the squids.
7-10 to 6-11	Continued to maintain the Frontier client and Frontier servlet software, including 6 servlet releases and 3 client releases. The most notable changes on the servlet side were to add the capability to load small files in addition to

	doing database queries (used by CMS for loading “sherpacks”), to put a limit on the number of threads any one servlet may use (in order to prevent interference with other servlets), and to prevent stale data from being used in squids when there's a server error. The most notable changes on the client side were to add a command line interface for loading the new files made available by the servlet, to make the retry strategies more robust such that it doesn't give up until all servers are tried with all proxies, and to improve performance by doing decompression interleaved with I/O.
11-10	Made a new release of the Frontier squid tarball distribution, the most notable changes being improved startup speed by cleaning out the old cache in a background process, increased robustness of restarts after a system crash, and increased ease of maintaining an rpm distribution.
7-10 to 6-11	Consulted extensively to the people who maintained rpms containing Frontier-related software: frontier-squid, frontier-tomcat, frontier-servlet, and frontier-awstats. In the middle of the period, the ATLAS person who had previously developed improvements to the Frontier servlet took over the maintenance of the rpms from the CERN IT person who first made them.
7-10 to 6-11	Helped CMS Online with their installation of Frontier, diagnosing quite a few cases where the runs died with frontier errors. Traced a majority of them to unexplained delays in their network. Also found that they were repeating some queries far more often than they should and that that made them much more vulnerable to the failures. Made 3 releases of their custom tomcat server rpm and one release of their squid rpm.
7-10 to 6-11	Helped with CERN CMS Offline operations, including serving as backup operations person during CERN off-hours and FNAL on-hours. The majority of the operations was done by a new person in CMS Facility operations, but she needed training and still needed expert consultation and some help with administering the machines at CERN. Designed two new monitoring systems that she implemented and that proved to be very useful: one automatically detects failovers from site worker nodes that directly connect to the Frontier servers, and the other graphs the server thread usage by servlet and sends out warnings when 75% of maximum threshold is reached. Also diagnosed a few problems that occurred on the production servers. A significant case was caused by a sudden drastic drop in bandwidth to a site in Japan that ended up tying up the central CMS Frontier servers’ database connections for a long time; solved that problem by recommending a change to the squid configuration to transfer the problem from tomcat to squid which can handle that type of situation much better.
7-10 to 6-11	Helped ATLAS Frontier Operations to improve their Frontier-based network and continued to meet with them on a weekly basis and advise them on improvements and on issues they had. Trained new people in ATLAS that were taking over more of the work, especially related to Frontier monitoring, which they were quite far behind CMS on.
7-10 to 6-11	Tested many CVMFS releases and provided the developers with a lot of advice on how to improve CVMFS based on Frontier experience.
7-11 to 6-12	Maintained the Frontier client software including 3 new releases. The Frontier servlet software was stable and didn't require any new releases. The client releases were all minor, bug fixes for four bugs that showed up under relatively rare conditions.
7-11 to	Made one new release of the Frontier squid distribution tarball, the most

6-12	notable changes being the ability to run multiple squids listening on the same socket to support 10-gbit interfaces, fixed detection on RHEL6-based systems of whether or not squid was already running, improved integration with the rpm distribution so it didn't need to keep modified versions of some of the files, and improved error-checking so common error conditions are clearly identified. The RHEL6 problem turned out to be caused by a bug in the 'pidof' command; wrote a patch for pidof to fix the problem. A variation of the patch was accepted by the package owner and Redhat. Verified that the Frontier server software worked well with Oracle 11g, including with the old and new Oracle client library. Also consulted extensively for the person in ATLAS that maintained rpms containing Frontier-related software (frontier-squid, frontier-tomcat, and frontier-awstats). The consultation included designing changes to the frontier-tomcat rpm to make it ready to be used by CMS production, by pulling in the Frontier servlet into the rpm and introducing a configuration file to define the parameters for multiple servlets.
7-11 to 6-12	Helped extensively with CERN CMS Offline Frontier operations. Operations load increased because the short-term CERN Frontier operator was replaced by someone who had fewer skills. Still did much of the system administration of the CERN machines, and often had to tell the operations person what to do. Also served as a backup operations person during CERN off-hours. Dealt with two disk drives that went bad, set up backups for some of the systems, investigated numerous occasions where alarms were raised, trained the Frontier operations person on how to do some of the investigations herself, did initial setup on new replacement virtual machines, and set up two new 10-gigabit squid proxy machines. Negotiated extensively for getting a backup Oracle server & frontier launchpad servers, and agreements were reached, but the servers have not yet been set up. Also set up backup squid proxies at CERN and Fermilab which are planned to be used to keep worker nodes from directly loading the launchpad servers when site squids are down.
7-11 to 6-12	Helped with CMS Online Frontier operations. That included removing DB passwords from their custom tomcat rpm, changing the passwords, and making sure that the passwords were uploaded to their secure server. Investigated a case of CMS Online timeouts which was associated with an incorrect GlobalTag configuration that caused a large number of repeated conditions queries. Made sure the monitoring alarms for the Online Frontier launchpad servers were fixed, and set up a way to make the Frontier-specific monitoring more easily accessible by a web browser outside of the CMS Online private network. Made two new releases of their custom squid rpm and one new release of their tomcat rpm. The squid rpm changes had to do with reconfigurations of the hardware in the private network; also made a plan with a CMS Online system architect at CERN to move such reconfigurations outside of the rpm, but it hasn't been a priority for him to do his part yet.
7-11 to 6-12	Helped ATLAS continue to improve their Frontier deployment and continued to meet with them regularly (first on a weekly basis and later on a bi-weekly basis) and advise them on improvements and on issues they had. Trained new people there that took over more of the work, especially related to Frontier monitoring which they were still quite far behind CMS on. Helped them identify the cause of some of their operational problems, which ideally they would have been able to handle themselves. During this period they moved more of their Offline conditions loading off of direct Oracle and onto Frontier, so the only workflows not using Frontier by the end of this period

	were Online HLT and Tier 0 Prompt Reconstruction.
7-11 to 6-12	Co-led the WLCG Database Technical Evolution Group which made medium to long-term plans for database usage in all the LHC experiments. Organized workshop and other meetings, wrote some of the report, and edited the report.
7-11 to 6-12	Continued to advise the CVMFS developer. Negotiated to have a mirror (a.k.a. Stratum One) of the CERN CVMFS distribution installed at Fermilab, and installed the reverse-proxy squids on them when the machines arrived.
7-12 to 6-13	Released two new versions of Frontier client and one new version of the frontier servlet. The major new features were an updated failure retry strategy including trying all proxies in a round-robin, support for Proxy AutoConfig (PAC) files, and verified digital signatures on responses to disallow tampering. Also wrote a command line tool called pacwget that adapts wget to use PAC files in a robust way. Made a detailed retry strategy document at https://twiki.cern.ch/twiki/bin/view/Frontier/ClientRetryStrategy
7-12 to 6-13	Took over maintenance of the frontier-squid and frontier-tomcat rpms from the ATLAS developer who had to stop contributing. Made two new versions of the frontier-tomcat rpm, and ten new versions of the frontier-squid rpm. There were many bug fixes and small new features, but the bulk of the work was making the rpm more like a standard Redhat rpm. Those improvements were necessary to get the rpm accepted into the Open Science Grid distribution. Submitted the frontier-squid rpm into the OSG distributed. Also updated the frontier-awstats rpm to be useful for monitoring CVMFS stratum one squids in addition to Frontier, and to include the software that collects performance data from the frontier-tomcat logs. The latter feature enabled completing the CMS Offline Frontier production servers transition to rpms, and enabled that very useful monitor to be used on ATLAS Frontier launchpads in addition to CMS Frontier launchpads.
7-12 to 6-13	Proposed and led the WLCG Squid Monitoring Task Force, which concluded with recommendations on how to proceed with the development of squid monitoring. In addition, proposed and led a followup WLCG Http Proxy Discovery Task Force which has published a tentative proposal based on Proxy AutoConfig (PAC) and a central server that can supply PAC configurations for all WLCG sites.
7-12 to 6-13	Participated in interviews for a new CMS Frontier Offline Operations person at CERN, and later transitioned primary operations responsibility to the new (more highly skilled, although still short-term) person that was hired. Trained him on all of the systems and tools, gave him direction for new development including moving some of the services to new WLCG Squid Monitoring virtual machines and the rest of the services to two other new VMs, and frequently consulted for him.
7-12 to 3-13	Prior to transitioning the CMS Frontier operations work to the new person, still did much of the operations. That included finishing the deployment of the 10-gigabit proxy servers at CERN, setting up a tool to merge awstats data from multiple machines into one, converting the CMS launchpad machines to be based on rpms only, and investigating a number of instances of unusual behavior on the CMS launchpad servers.
7-12 to 6-13	Continued to consult for ATLAS Offline Frontier operations people, including regarding their involvement in the development needed to implement the recommendations of the WLCG Squid Monitoring Task Force. Regularly participated in the meeting they ran every 2 weeks for Offline Frontier

	operations in both ATLAS & CMS.
7-12 to 6-13	Continued to consult for CMS Online operations people. The primary need during this period was in setting up the squids in the Online cluster to run Offline jobs during the long shutdown, because the cluster isn't needed then for recording collisions. The original plan was to set up some new squid machines with enough bandwidth similar to most Offline sites, but they didn't get the machines ready in time. Instead, helped them use the pre-existing hierarchy of squids already running on the Online nodes. Also participated in discussions early in this time period regarding ways to speed up the time in which new Conditions could be made available to the Online applications through Frontier (because they were having delay problems which they suspected might have been caused by Frontier, but they eventually concluded were not caused by Frontier).
7-12 to 6-13	Remained actively involved with CVMFS which continued to grow in deployment during this period. The WLCG Squid Monitoring Task Force explicitly included the needs of both CVMFS & Frontier. Played an important role in getting the Open Science Grid CVMFS-based OASIS service going, and in getting CVMFS approved and deployed at Fermilab. Administered the application software on the CVMFS Stratum 1 servers at Fermilab, including setting them up as a Highly Available pair. Set up an osg-cvmfs mailing list, and maintained the CVMFS software packages in the Open Science Grid software distribution. Organized a Fermilab Computing Techniques seminar by the author of CVMFS.

9. Project Team

There were many collaborators throughout the world contributing to some aspects of Frontier and Squid (especially the Operations portions). The following list covers the Fermilab collaborators.

Name	**Project Role**	**Time on project**	**Ramp-down Plan**
Dave Dykstra	Project lead	7-06 to 9-08 - 100% 10-08 to 9-09 – 80% 10-09 to 9-10 – 75% 10-10 to 9-12 – 60% 10-12 to 6-13 – 70%	Spend 25% time on Frontier and Squid application maintenance and support until the demand reduces
Lee Lueking	Supervisor	7-06 to 12-08 (percent unknown)	
Eric Wicklund	Tester	7-07 to 9-08 (percent unknown)	

10. Outstanding Risks

These are the identified outstanding risks:

1. CERN primarily only hires people for operations on a one year contract that is only extendable to two years, what they call “Category A” people. The number of permanent employees is very small, and there is no long-term Operations Engineer. The CMS Operations group responsible for Frontier operations may face the risk that the trained Category A person may not properly hand off knowledge and duties to the new person.
2. The Frontier & Squid systems are in very active use and, as technologies change and new uses come along, there will continue to be a need for changes and support. Unless maintenance effort is properly included into the budget for future years, there is a risk that support may be slow to be provided.

11. Operations and Support

The operations of the CMS Frontier system has now been transferred to the CMS Computing Operations group. There is expected to be a need for ongoing maintenance and support of the Frontier & Squid applications for both CMS & ATLAS for as long as they remain in active use.

12. Next Steps

Following the closure of this phase of the project, a maintenance phase at a reduced level of effort will proceed. The reduced level is feasible primarily because the Operations responsibility has been transferred. The following are the major currently known work items that need to be done:

1. Add python API bindings (which were developed by someone else in CMS) to the Frontier client release.
2. Get the pacwget tool and underlying tool pacparse into the open source pacparser library (which is the library that the Frontier client uses to implement Proxy Auto Config). The owner of the library has agreed in principle to accept these tools but they need to be submitted in a form that is very easy for him to integrate.
3. Support multi-threading in the Frontier client. This is something that CMS is interested in to speed up initialization of multi-core jobs.
4. Update the frontier-tomcat rpm to upgrade from Java 6 to Java 7 and from Tomcat 6 to Tomcat 7. Java 6 has reached the end of life of its support so upgrading Java is more urgent than upgrading Tomcat.
5. Bring the frontier-tomcat rpm more in compliance with Redhat rpm standards. There are a significant number of ways that it currently does not follow the standards.
6. Support the new CMS Database schema which is planned to put more data into fewer database tables. That will have an impact on the effectiveness of the If-Modified-Since optimization, because those fewer tables will be modified much more often. The Frontier servlet (in cooperation with the CMS application) will probably need to support timestamps stored in the database along with each table row in order to keep the load on the infrastructure manageable.

7. Support IPv6 in the Frontier client and in Squid. For Squid, this will require upgrading to squid3 which is a major rewrite of squid2 in a new programming language. Even though squid3 has been released for years, it is still missing sufficient support for two key features needed by Frontier: collapsed-forwarding and If-Modified-Since. Both features are high on the Squid team's priorities, but depending on the timing, we may need to develop the support ourselves and contribute it to the Squid project.
8. Extend the Frontier feature of reading files to also be able to read web pages over http. There has been talk of an application of this for CMS, and it could be used to extend the reach of existing Intensity Frontier http-based databases.
9. Demonstrate the feasibility and evaluate the advantages or disadvantages of the various ways of using the Frontier Distributed Database System for some of the larger Intensity Frontier experiments. There is general agreement that at least a distributed Squid architecture should be used when loading conditions on the Open Science Grid, but it hasn't been decided whether it would be better to use Frontier for that or to transfer some of the knowledge learned from Frontier's grid-scale database distribution into a different implementation.
10. Provide consultation to the implementers of the WLCG Squid Monitoring Task Force recommendations.
11. Conclude the WLCG Http Proxy Discovery Task Force, and either implement its recommendations or provide consultation to the implementers.

13. Lessons Learned

These are some of the lessons learned from this project:

1. It would have helped greatly to have been able to fully transfer the Operations to another team much earlier. The team at CERN wasn't able to take full responsibility for four years. It seems to have helped a lot to carefully screen applicants, but because CERN only hires people for a very short period, this process will have to be repeated many times.
2. Site-based automated caches are an excellent method of distributing data to grid jobs when there is much duplication of data. Using Http as the protocol is a good way to do it, not only because it is a good design for the wide area but even more because we can use standard software packages to do the caching and much of the monitoring of the caches. On the other hand, it is quite tricky to understand the failure retry strategy needed for best resiliency in the presence of multiple proxy caches and multiple servers, any of which can experience failures.
3. It takes a lot of effort and patience to collaborate with a large group of mostly independent people distributed around the world.