LFC Consolidation for the US Cloud: Dependencies and Impacts

B. Ball, R. Gardner, H. Ito, P. McGuigan, S. Williams

February 7, 2012

Introduction

Since September 2009 there have been discussions and efforts to consolidate and simplify the DDM infrastructure, including DQ2 site services and LFC catalogs. This originally started with Massimo Lamana but was taken up by Simone Campana in December 2010 (1). This document summarizes the following:

1. Description of the current LFC deployment in the US, including a table listing the # files by site, giving a sense of the scale of each LFC catalog.
2. Description of possible historical "structure" in the catalog that may need to be cleaned up before migration including permissions ACLs, directories, etc.
3. Examination of the utilities currently in use - and any needed changes implied if the LFC is consolidated at CERN. Eg. CCC.py and proddiskcleanse.py, etc. Impact for sites based on SE type, if any.
4. Any dependencies or implied changes in the compute node environment?
5. What is the immediate benefit of consolidation to ADC operations? to sites? (will this speed up central deletion or consistency checking?)

Current LFC Deployment

The below table summarizes the current status of LFC deployments in the US Cloud.

Table 1: LFC instances in the US Cloud

Site	LFC instance	# Replica/Meta	Consistency checking frequency	Consistency method
BNL Tier 1 NOTE: ORACLE Based	lfc.usatlas.bnl.gov t3lfcv01.usatlas.bnl.gov	76M/110M 4M/6M	When needed vary	central T3s check/central
AGLT2	lfc.aglt2.org	7.7M/10.3M	Daily	ccc_pnfs.py (v1.14)
MWT2	uct2-grid5.uchicago.edu	9.7M/11M	As needed	ccc_pnfs.py (v1.13)
NET2	atlas007.bu.edu	2.8M/3.7M	As needed	egg/custom scripts
SWT2_CPB	gk02.atlas-swt2.org	2M/4.7M	As needed	ccc_generic.py (v1.17)
SWT2_UTA	gk03.swt2.uta.edu	180K/360K	As needed	ccc_generic.py

				(v1.17)
SWT2_OU	tier2-05.ochep.ou.edu	700K/975K	As needed	ccc_generic.py
WT2	atl-lfc.slac.stanford.edu	7.8M/10.8M	As needed	custom scripts/dark data cleaning only

Historical Structures

At AGLT2 one of the otherwise unused LFC fields is used to hold the PNFSID of the file. This also used to be the case at BNL, but may no longer be true. A cron task runs once every 20 minutes to insert new values. This field is used to allow quicker access to files by only using information in the LFC, and not querying the dCache databases. As far as we are aware, there are no other such issues.

Benefits of LFC Consolidation

- Driving factors behind consolidation
  - For DDM operation, accessing multiple LFCs takes some efforts.
  - Access to the remote LFCs are slow due to larger latency.
  - Consistency checking among location service catalog and LFC will be much faster if LFC are consolidated.
  - Some T1s had major problem due to the backend ORACLE issue, resulting on shutdown of entire cloud for significant amount of time.
    - The consolidation of all LFCs to one place at CERN will result in entire ATLAS computing shutdown in the event of the problem at CERN.
- Expected benefits of consolidation
  - Sites have one less services to run.
    - LFC has been a trouble free service. The overhead of running LFC is quite low.
  - Consistency checking between the DDM location service and LFC will be faster.
    - On the other hand, the consistency checking between LFC and SE will be more difficult (see the next section.)

Impact on Consistency Checking

USATLAS has a history of being concerned with consistency checking that started even before the adoption of the centralized DDM service. This concern continued with full adoption of the tiered data model and DDM within ATLAS where USATLAS initially deployed Local Replica Catalogs (LRC's) before converting to the Local File Catalogs (LFC's). Each Tier 2 site in USATLAS maintains at least one LFC. Initially, the concern for consistency centered on the full utilization of disk resources by finding physical files that would never be used, ie dark data. As DDM matured, along with the Tier2 sites, so did the consistency checking within the Tier2's. Tier2 sites currently use the ccc_pnfs.py for sites employing DCache storage or ccc_generic.py for sites using other storage backends.

Both versions of the program provide a check between the replica information in an LFC and the

physical files found on storage. The replica information is retrieved from the local LFC by means of an SQL query, as opposed to LFC client commands and the physical files are discovered by querying the individual data servers for files. The check identifies files that are found in the LFC but not in the storage (missing files) or files found in the storage but no matching replica exists in the LFC (extraneous files). The tests are based on existence and do not cover more extensive characteristics such as size or checksums.

Missing files identified by the program are problematic as it indicates that a registered replica no longer exists on storage. We recover from this situation by downloading a new copy of a file or declaring the file lost. The extraneous files are typically not as serious but they do represent storage space that can be reclaimed. Recovery is accomplished by deleting the physical file. The program takes a time window into account so that recent inconsistencies are ignored. These types of inconsistencies are expected in a dynamic system like DDM. Files that are in the process of being delivered will show up in the physical files but will not be registered yet in the LFC. Similarly files being deleted may be removed from disk before the replica information is removed from the LFC.

A second type of consistency check done by either program is for checking the LFC replica information against the central DQ2 catalogs by space token to identify files that registered in the LFC but are no longer expected to exist at a site, or files that have not been delivered as part of a complete dataset. The check is performed by querying the DQ2 central catalogs for the expected datasets for one or more space token end points. The contents of each dataset is then queried to build up a list of expected files by GUID. The program caches the list of dataset contents so that subsequent runs can avoid performing this query again. The list of expected GUIDs is then compared to the contents of the LFC to determine if files are missing or extraneous. Missing files can be recovered by resubscribing the dataset to the affected endpoint. Extraneous files can be removed by declaring the files to the central deletion service. The ccc_pnfs.py program performs an additional consistency by checking physical files found on disk against PNFS ID's. This check allows for the discovery of both missing and extraneous files within DCache.

Both versions of ccc require information from an LFC and from the storage system. The program is mature enough that it can access these data either “live” and query the systems as it runs or can operate from flat files that provide dumps of either the LFC, the storage system, or both. For a Tier2 in the US it may take a few minutes to query the LFC for the entire dump necessary to run ccc. Dumping the contents of a storage system, generally involves walking the filesystem (similar to the Linux find command) and collecting data. There are several storage backends utilized in the US and each dump can be expected to vary based on size of storage, distribution of the filesytem (e.g. how many dataservers/pool nodes can be querried in parallel), and current load on the storage system. For an Xrootd based backend containing more than 2 million replicas spread over 16 dataservers, the query of the filesystem contents takes only a few minutes.

The check to compare LFC contents to the storage contents takes only a few minutes and is

largely based on reading in the data either live or from pre-captured flat files provided that the machine running ccc has adequate memory to hold both the LFC dump and storage dump in memory. The check to compare the LFC contents to the expected contents based on the DQ2 central catalogs can take significant time the first time the program is executed. The program must perform the equivalent of “dq2-list-files” for each dataset for each provided space token. This may take on the order of 24 hours for the initial run but due to caching of dataset contents subsequent runs will only need to query newly subscribed datasets, and take on the order of 3 hours for an SSD-based Chimera DataBase. For an iScsi based DB as at AGLT2, the time for subsequent scans is of order 8 hours.

USATLAS maintains a single LFC instance, hosted at BNL, for Tier 3 sites that have DQ2 endpoints. The Tier 3 sites do not have access to the LFC's database backend like the Tier 2's which maintain their own LFC and this makes running ccc problematic. A similar system is in place that allows a Tier3 to perform consistency checks and corrections using a program named storageManager.py

Tier 3's receive periodic dumps of the LFC as an SQLite file. The file is delivered usually to the LOCALGROUPDISK token area at the Tier3 and contains replica information and associated metadata for files registered in the LFC for the specific Tier3. The storageManager.py program will discover the contents of storage system through a find command. The program can then discover missing and extraneous files by comparing parsed replica names against physical filenames. The program also offers the Tier3 adminstrator the opportunity to fix the inconsistency by deleting data from the LFC (through LFC client libraries) or deleting physical files via SRM commands.

A few Tier3 sites are also used in ATLAS production activities and the storageManager.py program can be used for cleaning the PRODDISK token area by removing old files. This program is less refined than the equivalent Tier 2 program in that it will remove any file in the PRODDISK area that is older than a given time window, normally set to twenty-one days. This will delete both files staged to the Tier3 as inputs for production through the PandaMover process as well as remove lingering production outputs.

Cleaning up PRODDISK

In April 2011, responsibility for most PRODDISK cleanup reverted to central DDM. This left only Panda Mover cleanup in the hands of the T2 site. This responsibility will continue as long as Panda Mover is still in use by US T2s. The script used in this is pandamover-cleanup.py.

This script runs daily, and (typically at AGL) removes several hundred GB of data older than the default 14 days. Such files, if left untended, would fill dCache storage at a rate exceeding 3TB/ week, a disastrous rate. Any LFC central migration path in the US Cloud would need to provide a mechanism for rapid cleaning of such T2 files for as long as Panda Mover remains in use. As noted above, storageManager.py can perform this function.

Impact of LFC Consolidation on US Facilities

- LFC structure – what is lost and what is gained if structural changes are required
    - Have identified only one such use, a convenience/speed of access measure.
- WN dependencies (firewalls, software)
    - SLAC (WT2) and possibly NET2 firewall issues affecting LFC access
    - Jobs could fail due to the network overload.
        - At BNL, worker nodes are required by cyber-security policy to be behind the firewall. BNL used to have the LFC service exclusively on OPN network (outside the firewall), resulting in job failures by LFC when the network load was high. Since then, with one backend database, BNL has an LFC instance running within the firewall as well as on the outside to prevent any communication between client and the server crossing the firewall. This can not be done if the LFC is at CERN.
- Schedconfig and ToA changes: -> very minor
    - Panda schedconfig and TiersOfAtlas must be updated to point to the BNL/CERN LFC server for each site. This should be a minor effort.
- Ghost file and dark data cleaning
    - New tools to identify missing and extraneous files must be developed
- Exchange of local LFC issues for set of remote LFC issues
    - BNL (or CERN) personnel would now have to handle any LFC issue that arises.
    - Provision of flat files, sql dumps, etc, to T2 sites on a regular basis would have to be planned and executed. See discussions above on Consistency Checking and ProdDisk cleanup.
- Single point of failure
    - BNL/CERN ORACLE outage will result in complete blackout of entire ATLAS.
- SQL limitations, if any
    - BNL T1 LFC concurrency is currently limited to 100
- PandaMover vs DDM
    - PandaMover does not register files in the LFC. See discussion on ProdDisk cleanup.
- Comparison of stage-in and stage-out times between US (local LFC) and, eg, CA (central LFC) clouds
    - No hard data on such delays is currently available.
- Considerations concerning storage type at US sites (dCache, BestMan, Lustre, …)
- Some sites were editing replica for certain reason (fix full/short surl, etc.) Although this is still possible through LFC API, it is different. Also, identifying such a file is not easily possible without access to DB (sql)
    - Best solution would appear to be a global ATLAS fix that would result in such edits being no longer necessary. This should happen before any LFC migration takes place.

Impact on Federated Xrootd Infrastructure

The LFC is used by the federated xrootd system when translating file names from the global

namespace to the physical location on disk. The component which accomplishes this is XrdOucName2NameLFC.cc (http://repo.mwt2.org/viewvc/xrd-lfc/). The deployment of on sites is described in Reference 5. The load on the LFC can be reduced by caching lookups (using ttl and cachesize parameters); these may need study following consolidation.

Consolidation Test Plan

Possible consolidation test and final process

- One T2 LFC (AGLT2?) can be run at BNL.
    - The backend database of MySQL at AGLT2 can be replicated on live via rubyrep program to MySQL running at BNL. The program allows the continous replication, resulting in relative short delay (<1min).
    - LFC instanace running at AGLT2 will be changed to readonly (or stopped).
    - LFC instance will be run at BNL using the replicated MySQL.
    - Change schedconf and ToA for AGLT2 site(s)
    - Run jobs/DDM to see if there are any negative impacts. eg. Jobs failure due to LFC related cause
- If successful, one can migrate MySQL contents to BNL T1 LFC. The some development is needed. Also, it is possible that BNL T1 LFC might need a update to accomodate more than 100 concurrent connections. 100 is the max allowed.
    - possible process/code.
        - stop AGLT2 LFC running at BNL.
        - make dump/copy
        - insert triggers to detect insert,delete,update from the dump
        - start AGLT2 LFC running at BNL
        - migrate the dump to T1 LFC
        - run migration continously to check update via triggers.
        - stop AGLT2 LFC running at BNL.
        - clean remaining update
        - change schedconf/ToA for AGLT2
- If AGLT2 is successful, the remaining T2s can be consolidated to BNL LFC.
- BNL LFC can be migrated to CERN LFC.
- Extra: BNL might get CERN LFC as a back up site. The test of ORACLE replication has been discussed.

Alternative process:

- T2s can skip migrating to BNL. One can consider migrating to CERN directly without test(?) migration to BNL. The advantage would be to avoid complete US Cloud downtime. The disadvantage is that US production will be down when BNL T1 LFC is migrated anyway.

Workshop notes - 3/20/12

- LFC is dropped with Rucio
- Not consolidate by T2 to cern - labor impacts

- Go at regional level, and run like that for a reasonable (long) time
- Hiro: host lfcdaemon at sites -- database at BNL. But what about topology seen by cern? Ans: would see it as today. Note: for consolidated LFC, the lookup happens at panda server for uploads, not pilot. Kaushik: actually not, pilot consults several LFCs. Athena also uses LFC.
- Is there a general obstacle for processes on compute nodes to contact the LFC at BNL?
- Sarah's proposal to run the service locally that handles authentication to the backend db at BNL, the compute nodes access mysql directly.
- Schedule
    - UTA_SWT2 production cluster. Full migration. ~ month.5
        - Solution for PRODDISK-CLEANSE
    - AGLT2 would try this hybrid consolidation (running a local LFC daemon, mysql db at BNL). Follow success of UTA
    - The remainder TBD

References

1. Simone's original report (December 2010): http://indico.cern.ch/getFile.py/access?contribId=159&sessionId=16&resId=1&materialId=slides&confId=76896
2. Graeme Stewart, April 2011, LFC Pre-merge cleaning https://indico.cern.ch/subContributionDisplay.py?subContId=0&contribId=0&confId=135782
3. LFC discussions in ADC (many), See: https://indico.cern.ch/search.py?categId=1706&p=LFC&f=&collections=&startDate=&endDate=&sortField=&sortOrder=d
4. US Cloud LFC consolidation study group: https://www.usatlas.bnl.gov/twiki/bin/view/Admins/ConsolidatingLFCStudyGroupUS
5. Description of xRootd deployment at sites: https://twiki.cern.ch/twiki/bin/viewauth/Atlas/AtlasXrootdSystems

Scraps:

Begin forwarded message:
From: Hironori Ito <hito@rcf.rhic.bnl.gov>
Date: November 30, 2010 5:34:44 AM CST
To: Simone Campana <Simone.Campana@cern.ch>
Cc: Graeme Andrew Stewart <graeme.a.stewart@gmail.com>, Vincent Garonne <Vincent.Garonne@cern.ch>, k Bos <kors.bos@cern.ch>, Michael Ernst <mernst@bnl.gov>, "Robert w. Gardner" <rwg@hep.uchicago.edu>, Alexei Klimentov <Alexei.Klimentov@cern.ch>, Dario Barberis <Dario.Barberis@cern.ch>, Charles George Waldman <cgw@hep.uchicago.edu>
Subject: Re: LFC consolidation

Hello.

Any schedule before Friday is fine with me. but, can I send some concern regarding this issue . Otherwise, I might forget it.

1. Finding Dark files.
In US, we use the content of LFC to find dark files: SE dark, LFC dark and DQ2 dark. To do this effectively, various sites have special methods which work well with their specific storage. For example, in BNL, LFC replica entry has PNFSID which is unique ID from dCache. Using this ID, the search becomes much faster (in SQL) compared with search in name space. For US T3s, there is a program which creates sqlite3 formatted file for the specific storage from US T3 LFC. The program also registers it as an dataset in DDM. And, it is shipped to a corresponding site via DDM site service. Then, there is a program which a site admin uses to find/clean dark files of various types. For US T2s, there is a similar program that utilize the LFC dump.

In all of these cases, the access to LFC database is quite crucial although some of them can be replaced with more central operation. Also, the creation of dump can increase the load in database. If consolidated, the load can certainly increases.

2. Special entry in LFC.
In BNL (also AGLT2), as I have said already, LFC replica entry has also special PNFSID. And, the file is accessed using this PNFSID instead of the name space since this method is much more efficient (fact or 10 or more). Although it is not necessary to store this information in LFC, it is the most convenient place (for time being.)

3. Backup/mirroring LFC: technical issue for mirroring
In LFC, the backend database can be ORACLE or MySQL depending on the site. Therefore, the mirroring/copying of database contents using the specific database method can not possible between two different type of databases. For example, all US T2s use MySQL backend while BNL use ORACLE. If US T2s were to keep their LFCs while BNL makes backup copy, we can't use ORACLE (or MySQL) specific streaming method of backup. The backup must be done in SQL level instead of native database level.

Also, there is a technical issue of creating merged backup of various LFCs. The user/group entries in various LFC database are different. That is the same user(DN) can/will have different internal ID in different LFC site. The same thing can be said of file entries: for the same file/ GUID, the internal file id in various LFC can/will be different. It will be a bit tricky to make a merged backup from various sites.

Anyway, more if I come up with idea.

Hiro

On 11/30/2010 10:40 AM, Simone Campana wrote:
Hello.

We are supposed to discuss LFC consolidation this week, but if I look at the agenda I find zero slots where this can be discussed: it is completely packed. I therefore propose an LFC beer at the end of the afternoon session. May be thursday afternoon? At R1 they serve beer after 5PM.

Simone

Begin forwarded message:
From: Hironori Ito <hito@rcf.rhic.bnl.gov>
Date: December 1, 2010 4:45:14 PM CST
To: Simone Campana <Simone.Campana@cern.ch>
Cc: Graeme Andrew Stewart <graeme.a.stewart@gmail.com>, Vincent Garonne <Vincent.Garonne@cern.ch>, k Bos <kors.bos@cern.ch>, Michael Ernst <mernst@bnl.gov>, "Robert w. Gardner" <rwg@hep.uchicago.edu>, Alexei Klimentov <Alexei.Klimentov@cern.ch>, Dario Barberis <Dario.Barberis@cern.ch>, Charles George Waldman <cgw@hep.uchicago.edu>
Subject: Re: LFC consolidation

Hello.

One more point I would like to make is the connectivity issue from CE nodes to central LFC (wherever it may be located).

At BNL, we found that there are correlation between job failure caused by LFC related error with high network activity. Previously, we had LFC located outside the firewall(s) on OPN network while CE nodes are within the external and internal firewall. And, it was found that when the throughput rate (within our site) was high, there was an increased number of job failures associated with LFC (both put and get). And, it was not due to the load in LFC service itself since that has never been issue. Since we suspected the network, we have also added LFC service within the firewall of CE nodes using the same back-end database. Since then, we have not seen any LFC related errors (correlated with network load). I am afraid that, if we move LFC to anywhere (outside of BNL), the previously seen LFC related errors in panda jobs will re-appear again. We really need to do the careful evaluation and testing in under very stressful conditions, which T1s get when there are simultaneous large-scale reprocessing and

data replication.

NOTE: If one can access to Panda database, you should be able to see the validity of the above statement. We added the new LFC within the firewall around 5/26. So, you can quantitatively compare the errors in May reprocessing and that from this recent Oct/Nov one.

Hiro

Begin forwarded message:
From: Kors Bos <kors.bos@cern.ch>
Date: December 7, 2010 3:56:28 AM CST
To: ADC Operations <atlas-project-adc-operations@cern.ch>, ADC Development <atlas-project-adc-development@cern.ch>
Cc: ADC Management <atlas-adc-mgt@cern.ch>, Eric Lancon <eric.lancon@cern.ch>, Hans von der Schmitt <Hans.von.der.Schmitt@cern.ch>, Dave Charlton <Dave.Charlton@cern.ch>, Andy Lankford <Andrew.james.lankford@cern.ch>
Subject: Decisions from the Software and Computing workshop

Decisions from the Software and Computing workshop of Nov.29 - Dec.3 at CERN (http://indico.cern.ch/conferenceDisplay.py?confId=76896). The decisions were prepared during the week and finally discussed during the Friday morning session.

<snip>
File Catalogues LFCs (Simone Campana)
We have one LFC per cloud in each of the T1s plus one in the important T2s in the US. We will merge all these LFCs into one at CERN and a live backup in one or more T1s which can also be used for other purposes such as consistency checks, dumps and monitoring. Eventually this master LFC will be combined with the DDM central catalogue that now serves very similar functionality. A staged migration will allow scalability measures. We need to agree this DB service with CERN-IT and work on the migration tools. The migration will take a few days per T1 LFC and should start with some small clouds next year as early as possible.
</snip>