Data Center for Library of Congress Digital Holdings: A Pilot Project

Final Report

Library of Congress
Martha Anderson
Babak Hamidzadeh
Michael Ashenfelder
Andrew Boyko
Phil Michel

CACI
Terry Harrison

SDSC
Richard Moore
David Minor
Jim D'Aoust
Bryan Banister
Chris Jordan
Adam Lathers
Steve Meier
Reagan Moore
Donald Thorp
Emilio Valente
Alex Wu
Bing Zhu
John Schroeder
George Kremenek

UCSD Libraries
Ardys Kozbial
Arwen Hutt

Group of Jewish children with a teacher. Samarkand, Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-21861

October 17, 2007

1. Communication Logistics and Tools 4
1.1 Communication tools 4
1.2 Documentation Tools 5
1.3 Trust Elements 6
2. Deliverable: Data Transfer 6
2.1 Setting up the Network 6
2.2 Setting up the Storage Infrastructure at SDSC 8
2.3 Transferring the Data to SDSC 10
2.4 Checksum Verification 11
2.5 Linkage Issues 12
2.6 Transfer Back to LC 12
2.7 Trust Elements 13
3. Data Monitoring and Logging 15
3.1 Data Monitoring - General SDSC Environment 15
3.2 Data Monitoring - Writing Scripts 16
3.3 Script Roll-Out 17
3.4 Data Monitoring - An Unplanned Event 18
3.5 Trust Elements 19
4. Data Access and Manipulation 20
4.1 Front End Design 20
4.2 File Manipulation Scenarios 23
4.3 Reporting Issues 23
4.4 Questions on the Nature of the Archive 24
4.5 Trust Elements 25
5. Web Archives 25
5.1 Working in Parallel 25
5.2 Trust Elements 29
Conclusions 30

Introduction

Between May 2006 and October 2007, the Library of Congress (LC) and the San Diego Supercomputer Center (SDSC) conducted data-transfer and storage tests. At the heart of the project was the issue of trust, specifically how the LC could trust SDSC to reliably store several terabytes of the LC's data. By what means could SDSC prove to the LC that the data was intact, preserved, and well-cared for? What tests could the LC devise, and what metrics could SDSC produce, to guarantee the integrity of their remotely stored data?

The two main objectives of the project were:

- For SDSC to host LC content reliably and return it intact at the end of the project
- For LC to be able to remotely access, process, analyze, and manage that content

The content consisted of two different types of digital data from two divisions within the Library:

- From the Office of Strategic Initiatives (OSI): approximately 6 TB of harvested Web sites. The tens of thousands of individual files that comprise a Web site were bundled and compressed into 500 MB ARC-formatted files.
- From the Prints and Photographs (P&P) division: approximately 580 GB of digital image files – high-density masters TIFF files and their less-dense derivatives – and associated files that support the display of those images. These files were part of the LC's Prokudin-Gorskii exhibit (http://www.loc.gov/exhibits/empire/).

Inspired by SDSC's staggering technological potential, the LC had devised several scenarios for the data tests. But ultimately, as the project progressed, the LC opted to keep its goals simple: data transfer, storage, and file manipulation.

In the end, both partners were happy with the project's success. The project also produced lessons and unexpected results, some of which will have deep implications for all cultural institutions regarding transfer and storage of their digital assets.

.

1. Communication Logistics and Tools

Group of workers harvesting tea. Greek women. [Chakva], Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-

1.1 Communication tools

Early in the life of the project it became clear that communication was going to be a key element in all processes. Because there were a large number of staff members involved, many of them working simultaneously on different issues, keeping everyone on track was going to be a challenge. There also needed to be coordination between the project managers at the two sites, to make sure that the work being done met the expectations of the PIs.

With these issues in mind, a three-part *meeting plan* was used:

- A weekly teleconference between staff at the Library of Congress (LC), San Diego Supercomputer Center (SDSC) (for the purposes of this report, SDSC includes the UCSD Libraries), and CACI. This was a large-group meeting where goals, progress and issues were discussed. It was important to have this meeting every week at the same time - people added it to their regular schedules and it became part of their routine.

- A weekly Project Coordinator meeting between the two project managers. This was a “behind the scenes” discussion where the long-term agendas and plans for the project were discussed and finalized. One element which the managers worked on during this time was an ongoing Work Breakdown Structure (WBS) which helped to keep track of the work being done. The meeting also allowed for an atmosphere where frank discussions of progress and issues could be discussed outside of the larger group.

- A weekly “internal” meeting at SDSC. This was an important facet of the process for the staff at SDSC. It allowed them to get together and discuss the work that was ongoing. It also provided a time when the SDSC project manager could make sure that everything was on track in preparation for the weekly teleconference with the LC.

These meetings represent just the formally defined processes. There were also a number of side-meetings which were put together *ad hoc* as needs arose.

There were several other *communication tools* used in the project. These included:

- A mailing list setup by the LC. This list allowed a convenient way to communicate with the whole group at once. It was the main vehicle by which updates, requests for service and general questions were disseminated. It was also the channel used to send out meeting agendas, meeting minutes, and similar things.

- A Sharepoint internet site. Sharepoint is a Microsoft product which allows for the management of documents, calendars, discussion items, etc. The Sharepoint site was setup and maintained at SDSC on the recommendation of CACI. Use of Sharepoint for communication is probably the main tool which would be reviewed in a future project. The interface proved to be cumbersome and using the site in general was less than ideal. Future work might be better served by a lighter-weight tool such as a Wiki.

In general terms, because of these communication tools, the progress and cooperation among the disparate groups was outstanding. The project was able to quickly progress from a “drawing board” phase, through a design and setup phase, and into an ingestion and implementation phase. Moving through these phases required that project members work quickly and with an eye on the calendar.

The main thing learned from the communication processes was the importance of having regularly-scheduled meetings which were the core of the process. Having meetings every week focused the disparate groups and forced people to think about what work they were doing on a constant, expected basis.

It was also clear that any tools used in the communication process must not hinder that process. The Sharepoint site was a good idea, but the staff doing the actual work in the project found it to be extra work, and often not worth using.

1.2 Documentation Tools

Each of the SDSC staff members responsible for tasks was asked to prepare documentation outlining their work. They did this by filling out a milestone “template” form so that the information would be consistent for all reports. These forms included information about time to completion, costs, surprises and problems, and future iterations of the same tasks. One of the most time-consuming and onerous parts of any enterprise is documentation. By providing staff with pre-formed templates, it was hoped that they could quickly fill-in relevant pieces without having to spend a lot of time on narrative passages. The

forms also provided an easy way for the project managers to quickly scan through the documents for comparison. It was also stressed to the people filling out the forms that they should do so as soon as possible after completing the work in question, to ensure that the work they did was fresh in their mind and documented as accurately as possible.

1.3 Trust Elements

Regarding the mechanics of "getting things done," there were several issues of trust. The largest one arose because of the cross-institutional nature of the project: staff at the LC and CACI needed to have full confidence that staff at SDSC understood their requests, acted on them in a timely fashion, and communicated any relevant items for consideration. This need can be summed up as one of *transparency*. All actions within the project needed to be transparent to all participants. There could be no black holes where actions were occurring without being reported.

The second major trust issue involves the issue of *timeliness*. Since SDSC was providing a service for the LC, there were expectations that all work be done on time and with clear explanations of any delays. Staff at the LC needed to have confidence that work would be completed when expected or have an explanation for why it might not be.

2. Deliverable: Data Transfer

Nomadic Kirghiz. Golodnaia Steppe, Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-21854

2.1 Setting up the Network

The Statement of Work for the project called for terabytes of data to be transferred. In the initial planning stages for the project the plan was to ship all of the data from the LC to SDSC on hard drives. The goal was to ship just over 4 TB of data using 500 GB drives. This was a common way for the LC to send and receive data from its various partners. In contrast, SDSC typically sends data using high speed networks, notably Internet2. (SDSC is one of the founding partners in Internet2 and forms a part of the network's backbone.)

As the project's early stages progressed, the LC's own Internet2 connection became available. It was thus proposed that at least some of the data should be sent using this connection. However, the simple presence of the high-speed network didn't guarantee that it worked correctly. There were a number of issues that needed to be solved, including making configuration changes to the data transfer boxes at the LC, working through a number of firewall issues within the LC, and making a number of other changes.

The networking staff at SDSC ran a number of tests from San Diego to the transfer box at LC. It was immediately apparent that this box was only capable of transferring data at a theoretical maximum of 100Mb/s. In real world transfers this box was only able to achieve 5Mb/s. This was in contrast to tests being run from San Diego to other Internet2 sites around the country, which were able to reach transfer levels as high as 730Mb/s.

Troubleshooting this issue was made more difficult because of the presence of Firewalls and a VPN at the LC. These are common issues that often come up in processes like this and are part of standard troubleshooting issues. In order to get around these issues in the short term, SDSC networking staff setup a computer at SDSC which was a duplicate of the one at the LC. This allowed them to test all of the parameters and get it working outside of other issues. These changes were then made at the machine located at the LC.

Once these changes were made, transfer speeds reached a high of 88Mb/s over the 100Mb/s network. While still relatively slow, it represented a very good rate given the environment. Soon after this, the LC networking staff installed a gigabit uplink to Internet2, increasing the theoretical limit first to 660 Mb/s and eventually to 1Gb/s. Making a number of configuration changes, the project team was able to achieve a network transfer rate of 388Mb/s. (This rate was reported using the iPerf network tool, a standard tool for this kind of work. Note that this rates is a theoretical maximum for the network. It is not the rate that can be expected for actual data movement in real world circumstances.)

At this point in the process the network was deemed "ready" for data transfer. Networking staff at SDSC wanted to continue working on the network setup, for two primary reasons: 1) The 388Mb/s rate was still about half of what it should be. SDSC sees rates much closer to 1Gb/s to many sites across the country; 2) the SDSC networking staff contended that in order to keep the high speed network functioning there needed to be a more permanent network monitoring process in place. Networks in general need constant hand holding for reliable performance. This is even more important in a high-tuned, high-speed network.

It was determined however that work on improving and monitoring the network would cease at this point. LC and CACI staff contended that further work would be outside of the scope of the project, and would take the LC's scarce technical resources away from their designated job of data transfer. It was reiterated that

the project was about building trust and moving and storing data reliably, not networking. Since the rates that had been achieved would be more than adequate to transfer the amount of data in the designated collections, work could commence on the real work that needed to be done.

2.2 Setting up the Storage Infrastructure at SDSC

Before any data could be moved to SDSC, decisions had to be made about how and where it would be stored. Fortunately the project was able to leverage much of the existing SDSC infrastructure and processes. Generally speaking the storage for this project was divided into three parts: a) an ingestion machine, which initially hosted the data arriving from LC; b) the back-end storage, which comprised a large amount of storage including live disks and tape-based archival storage systems; c) access machines, which housed the various mechanisms used by LC staff. This model was based on the standard storage model used by SDSC's Data Central group. Figure 1 shows a general view of this model.

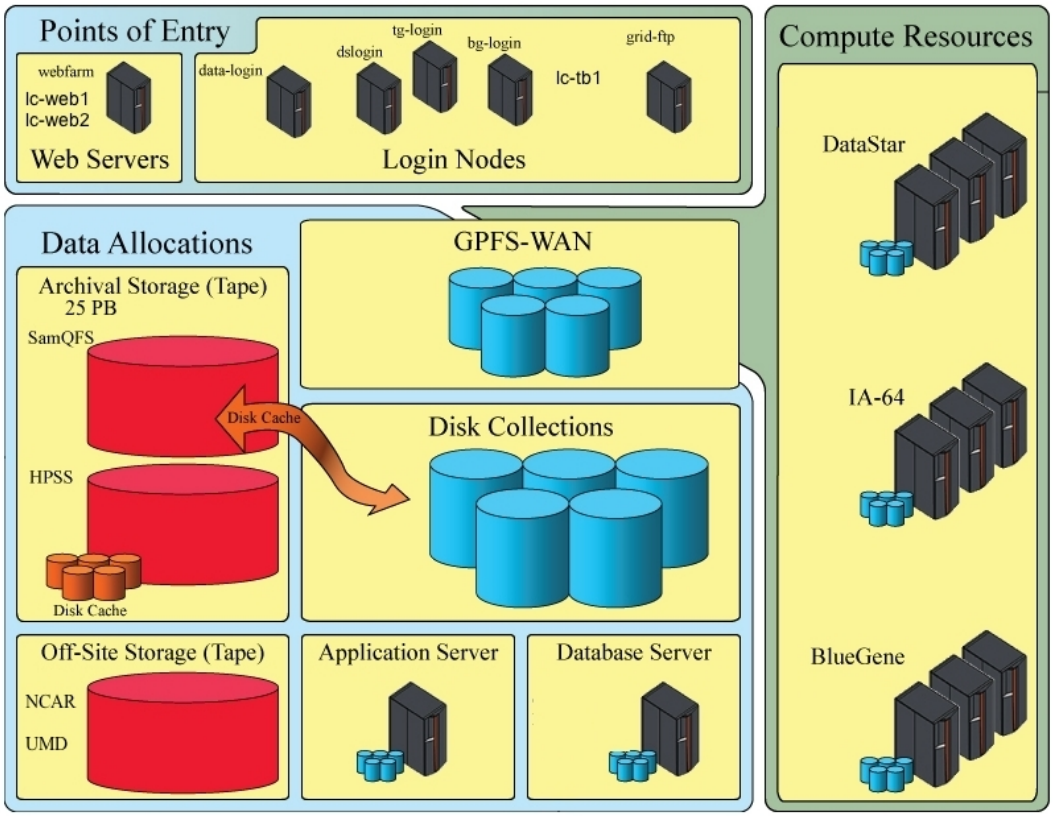


Figure 1: Storage Infrastructure at SDSC

There are several items to note in this diagram: first, SDSC has several different storage management systems. The two most important are the High Performance Storage System (HPSS) and SAM-QFS. HPSS was developed by IBM in conjunction with several DOE laboratories. SAM-QFS is a product from

Sun Microsystems. While they have slightly different feature sets, both systems provided the same functions for this project: the ability to store large amounts of data on both disks (for fast access) and archival tapes (for long-term reliability). SDSC decided to store identical copies of all LC data on both of these systems for reliability. It also provided some unique tests for sharing and logging data among systems.

During the course of the project, SDSC acquired a piece of test equipment from COPAN Systems: a MAID storage system. MAID stands for Massive Array of Idle Disks and is a proprietary technology from COPAN. The unit that SDSC acquired is one of the first production models in the world. With the agreement of staff at the LC, SDSC put a third copy of all the LC data on the MAID system to see how it performed.

Another important thing to note from Figure 1 above is that the separate “pieces” of the storage infrastructure are indeed pieces: that is, they are functionally separate from each other. The ingest machines are completely separate from the storage machines which are completely separate from the front-end machines. This provides a great deal of “safety” in the storage environment. One of the goals for high-reliability storage is the elimination of single points of failure. By segregating the sections of the infrastructure, there are fewer opportunities for one event to take everything down.

At the same time, however, this complex infrastructure requires more management. In SDSC’s case, in fact, the different pieces of the environment are managed by different groups. While this normally works very well, there was an instance where this complexity led to some confusion and lack of access for the LC. (This is discussed below.)

Finally, there is a third point to be taken from Figure 1: during the course of this project SDSC increased its storage presence at two remote sites: the National Center for Atmospheric Research (NCAR) in Boulder, CO, and the Pittsburgh Supercomputer Center (PSC). As a demonstration of enhanced data storage, SDSC replicated a section of the LC data at these sites. When this was done, the data was considered to be a static archive: that is, it was not accessible by anyone at the LC. The purpose was for this copy of the data to be available only if needed in an emergency situation.

Clearly the storage of LC data at SDSC was robust and complex. For the purposes of data management within SDSC, the decision was made to use the Storage Resource Broker (SRB). The SRB (Figure 2) is a well-known and fully-featured collection management system in use in many instances at SDSC. The SRB provides all of the data management, replication, logging and auditing services called for by the project proposal. Because the data sets within the project had different structures, they were stored in SRB in different ways, but to the “end user” accessing the data these differences were irrelevant.

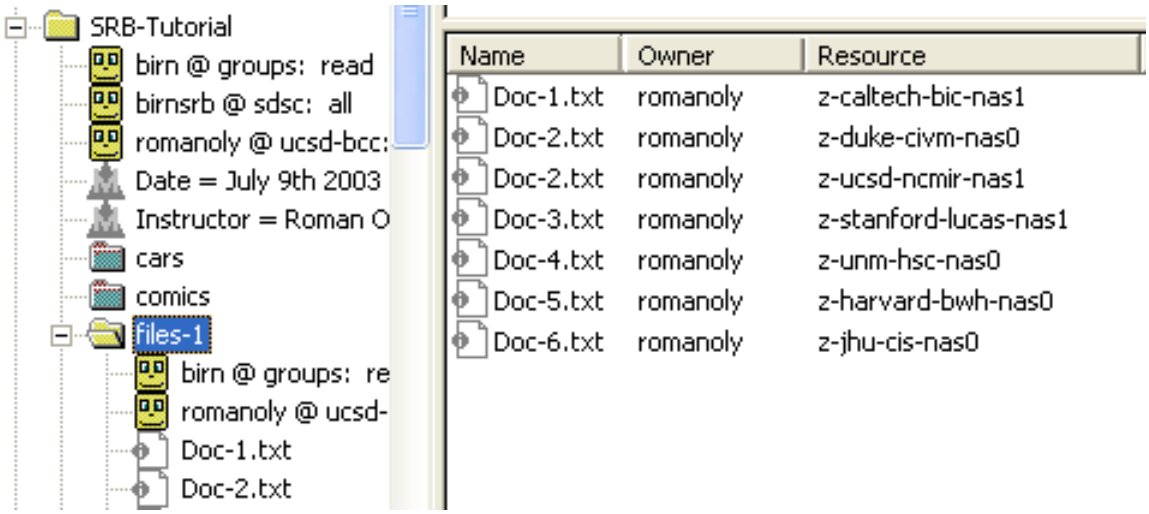


Figure 2: Example of data listed in SRB storage. Note that this is storage for a different collection, not that of the current project. The column on the right-hand side lists the physical storage location for a copy of data. In this example, the storage resources included systems at Caltech, Duke, UCSD, Stanford, UNM, Harvard, JHU for a bio-informatics project (BIRN).

2.3 Transferring the Data to SDSC

While the network and data infrastructures were being readied, data transfer commenced. The first bit of data transferred was actually sent physically on hard drives. This was done because it was a well-known process for LC staff and because it allowed work to progress without waiting for the network to be ready. Nineteen drives were sent to SDSC. This process was fairly labor-intensive on both ends: first two disks had to be purchased for the P&P transfer. Then they had to be connected to the LC storage system containing the data. Then they were mailed across country. Finally, SDSC took the disks and connected them to the local ingest machines and transferred the data. In total this process took several weeks. While time-consuming, the process was well-understood and generally encountered few problems.

When the high-speed network was configured enough to allow acceptable transfers, this method was used. It allowed several terabytes of data to be transferred in a much shorter amount of time. This process was also labor-intensive however, mainly because of configuration problems between the two sites. SDSC's transfer application of choice is GridFTP. GridFTP is a high-performance transfer protocol optimized for high-bandwidth wide-area networks. It is written and maintained by the Globus Alliance. It is based upon the Internet FTP protocol, with extensions for high performance operation. GridFTP is a very common transfer tool in the high performance computing world.

While SDSC staff were very comfortable using GridFTP, it represent a significant barrier for LC staff. This was because they had not used it before and they were not experienced in the general “grid” infrastructure that the application uses. The situation was also complicated because the LC computing infrastructure is firewall and password-based. GridFTP depends on a certificate-based infrastructure. This difference meant that staff at both SDSC and LC had to do additional work to configure the transfer setup.

In addition the network transfers suffered because, as noted above, the network setup was not a focus of this project. Because of this time was not devoted to maintaining a reliable link. This meant that it was often hard to determine when problems were related to the transfer application or the network itself.

This difference in security environment represents one important lesson learned in this project: organizations can have very different configurations and expectations for security and access; these differences can have very tangible effects on many work processes. For instance, setting up the grid infrastructure is complicated, time-consuming and demands input from a wide range of participants within an organization. Because of this there needs to be clear understanding that the work being done will have a strong benefit on the larger process.

It is also an example of a question or set of questions that an organization like the LC needs to ask when starting to work with other institutions. Not all potential partners will have the same expectations as the LC, and they may also have different combinations of security environments.

2.4 Checksum Verification

There must be a way to confirm that data transferred to a remote storage destination is the same as the data at the source. Checksums are one way of confirming that the data has not changed. In basic terms, a checksum is a redundancy check which provides a simple way to determine data’s integrity by detecting errors. A checksum adds up the data’s bits and stores the sum. At the other end of the transfer you must produce a checksum again and compare the results to the original checksum value. If the sums match, this confirms that the data has not changed.

For all data being transferred, LC staff prepared checksums and manifests. The manifests were simple text files that listed all of the data being transferred, the checksums for the individual files, and the directories storing them. This is a very easy way to do a first pass check on data reliability. As data arrived at SDSC, staff used the checksums and manifests to verify the validity of the transfers.

Almost immediately a problem arose: there were a large number of discrepancies in the checksums between the organizations. Staff in both locations worked hard to find the problem, which turned out to be the software used to generate the checksums. Specifically the software (MD5deep) worked correctly on two of the computing platforms used in this process (Solaris and Linux) but not in the third (AIX). Once this problem was discovered, fixes and workarounds were put in place, the most important being a bug-fixed version of the software. All subsequent data verified OK as it was ingested into the SDSC storage system. SDSC performed a second checksum verification all data was migrated to its permanent storage locations. These checksums formed the basis of future tests of validation within the SDSC storage system, noted below.

2.5 Linkage Issues

Another issue related to differences in the storage environments came up after the data was transferred to SDSC. This problem was also related to the multiple versions of operating systems at the two sites. Within the storage system at the LC, there were a number of links and file system mappings that had been created for the LC staff to work with the data efficiently. This is a common practice in many storage resources. When file systems are moved from one storage resource or another, however, the default action is to take links and create real files from them. Unless the person transferring the files is aware of the presence of links and makes explicit workarounds for it, the resultant filesystem will not exactly match the original. For the strict purposes of the project, this was not germane for SDSC to demonstrate ability to preserve and monitor data, but replicating those linkages was necessary so that they could provide access (under GridSphere ultimately). The important point is that the operating system environment needs to be considered a variable for third party storage. Mappings and link files are not relevant for pure and simple bit storage, but it might be (as it was in this project) if any operations, services or workflows are to be provided.

2.6 Transfer Back to LC

During the course of the project it was decided that testing transfers back to the LC would be beneficial. This was referred to alternatively as the “Doomsday” scenario - what processes would need to be considered if data needed to be handed back to the LC at a moment’s notice, for instance in an emergency situation. There were several issues which were part of this process, including making sure that the network between the sites continued to function and being able to verify the data once it arrived back at the LC, to prove both that it had not changed during the transfer process and also that it could be used in the same systems that had originally generated it.

The transfer was made using the same two methods as the original from the LC to SDSC: first, several terabytes were transferred using GridFTP over the high speed network; at the same time a hard drive was prepared at SDSC and data was copied over to it. This drive was then mailed to the LC – this was done as a proof of concept to show another possible method.

For the network transfer there were some delays because the transfer applications had not been used for several months. In the interim changes had been made to the systems at both sites. This meant there were some configuration tweaks that needed to be made to get things running as fast as possible. Once these were done the data was transferred at an average speed of 60Mb/s. The data sent in this manner was a subset of the web archive data.

The transfer on disk was straightforward but did take the expected time necessary to copy the files, package the disk and have it sent. Once it was received at the LC, a machine to which it could be attached was identified and the data copied to it. The data sent on this disk was the entirety of the Prokudin-Gorskii (P-G) data.

Once the data was resident again at the LC, it was verified to be readable. This was done in a manual process by simple visual interaction with it: for instance the manager for the P-G data looked it over and verified that it appeared correct and was in the correct directory structure. Several of the files were opened and verified to be correct and uncorrupted.

In the future, it would be highly beneficial to make this process less manual and more rigorous. This would better insure that the data could be actually used by the LC applications.

2.7 Trust Elements

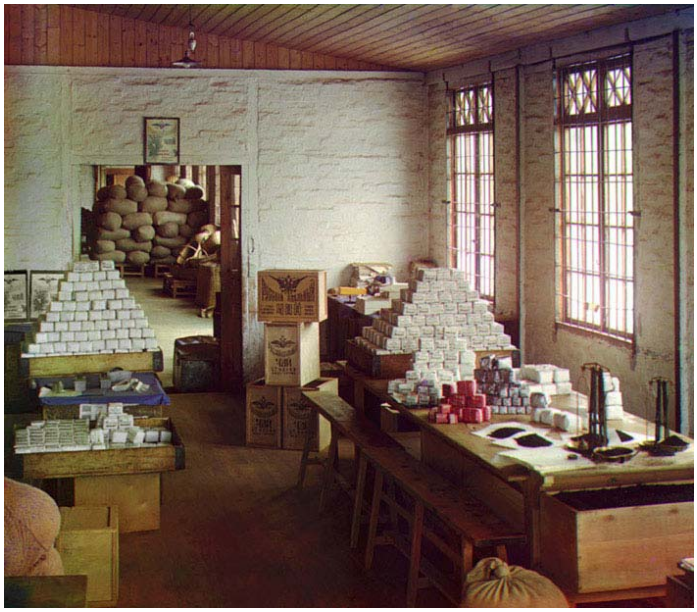
There were several important trust elements for this section of the project. The most notable were:

- *Have multi-institutional issues been solved?*
  Clearly transferring data between LC and SDSC exposed a number of differences in both physical setup and expectations. Because of its history as one of the founding members of the Grid Computing community, SDSC has been used to working with a specific set of applications and services. The LC (and indeed most of the organizations in this arena) do not have the same background. Thus it is incumbent on SDSC to be flexible and non-dogmatic in its work if it wants to continue to expand its relationships with similar institutions. This is something that might well change from situation to situation (e.g. what comprises an “acceptable” network), and

SDSC needs to be prepared to work through these issues.

- *Does the new infrastructure actually improve the overall process?*
  Clearly using high-speed networks to transfer data is a benefit. It's not necessary, however. This project could have been completed just as successfully with the data being sent completely on hard drives. In addition, the work that was done to setup the high speed transfers was time-consuming and labor-intensive. The question then needs to be asked: have the changes put in place been of benefit generally to the LC? Or were they just a lot of work that could have been spent on more productive matters? This is a question that only the LC can answer.

- *Has a long-term solution been found?*
  As noted above, the work setting up the data transfer environment was significant. It is important that the end result is something that the LC can use again in the future without significant changes or significant additional work. The solutions investigated should be *reusable* and *reproducible* by the LC. If not, then the LC cannot trust that SDSC has produced something of benefit.



Weighing section. [Chakva tea factory], Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-21514

- *Is solution useful for other organizations?*
  Finally, when large-scale setup is done between organizations, it should not be ad hoc. It should not work just for that particular relationship. Instead, whatever solutions are devised need to be applicable in other situations. In fact this is of prime importance in this particular case, because the LC is hoping to accomplish similar transfers with other organizations. If SDSC has not produced a solution which can be shared in some way with these other processes, then it should not be trusted with future work.

3. Data Monitoring and Logging

3.1 Data Monitoring - General SDSC Environment

Generally speaking, the current SDSC model for verifying data integrity is:

- Checksums are managed for each file in separate database systems / metadata catalogs
- Synchronization mechanisms are used to validate the checksums, and replace bad replicas
- Replications of files are stored across multiple storage systems
- Integrity checks verify files have associated state information in the metadata catalogs
- Integrity checks are done to verify all records in the metadata catalogs point to valid files.
- Files and state information are replicated into independent data grids run at another site
- Federation of data grids synchronize collections housed in two independent data grids.

For this project SDSC aggressively validated the integrity of the collections. Not all projects require this level of integrity-checking. Often data stored is “write-once, read-never.” Data in this conception is really a just-in-case backup which is there for unspecified peace of mind. This is not the case in a digital preservation environment, strictly defined.

One simple approach to data monitoring is to show that there is a verification procedure that can be run after any operation performed upon the data. This could happen after “known, good” data changes, or to simulate failures. Some example failures:

- Data transmission problems - problems with checksums and bad disks.
  - Detect that there is a change; verify that it was unintentional.
  - Run a synchronization mechanism to replace changed file with original
  - Verify process worked.
- Destroy a file (system administrator command at the root level) in a random location
  - Detect that file was destroyed; verify it was unintentional.
  - Run a synchronization mechanism to rebuild the corrupted file.
  - Verify replication worked.

- Change a file so that it differs in two locations (e.g. in two live disk locations)
  - Live monitoring by system detects that there is a difference.
  - Decision is made (either manually or automatically) which is the correct version
  - Appropriate change is made

At each of the designated points where problems may occur, SDSC can provide a standard way to verify integrity:

- Access file and generate checksum
- Compare checksum with the saved value
- List all files for which checksums are bad
- Synchronize replicas such that files with bad checksums are replaced with good files

The list of designated failure points defines when the above procedure should be invoked. Identifying the type of corruption is harder, requiring a systems administrator to examine logs, run hardware integrity checks, and determine if the problem is repeatable. This requires the development of a procedure to identify the cause of each corruption.

The synchronization step identifies when files are deleted from any storage system and when they are corrupted.

As with any organization, there are systemic dependencies within SDSC. An example is the authentication system which is mounted at each computer using NFS. When the NFS file system goes down, we are unable to access any of the computers. A question is whether there are other systemic dependencies that may affect data integrity. Possibilities are:

- Water release damaging all tapes
- Fire in a tape silo
- Tape library controller corruption
- Bad microcode in tape drives

3.2 Data Monitoring - Writing Scripts

To monitor all data interactions as closely as possible, SDSC created a data testing and verification team for the project. The initial tasks for this group were to create procedures to monitor the Prokudin-Gorskii (P-G) data on the designated storage areas within SDSC.

For the P-G data, the requirements are spelled out specifically within the proposal:

- Locate desired files in SDSC storage system
- Replace an existing file stored at SDSC with an updated version
- Delete specified files from storage
- Add files into storage
- Change properties of sets of digital files rapidly

SDSC provided the auditing and logging capabilities for these actions to show exactly what happened when users interacted with the data. SDSC took a

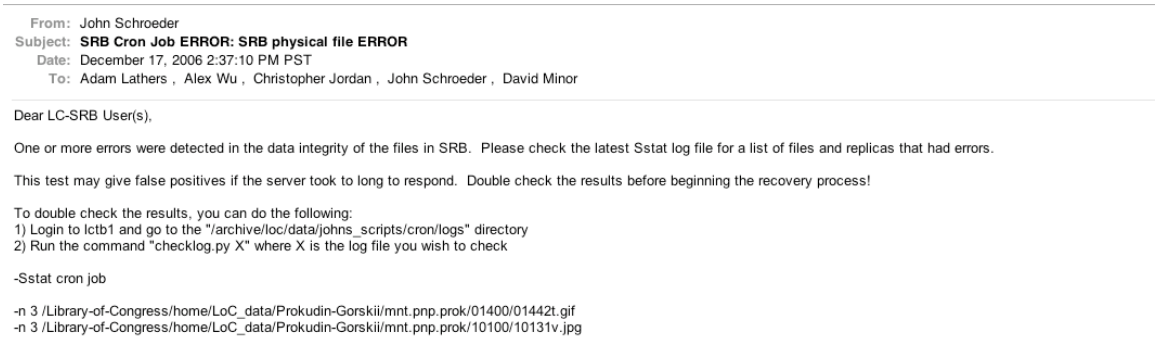
straightforward approach and designed verification procedures that ran automatically at preset intervals. These intervals ranged in length from once a day to once a month.

Because the decision had been made to use SRB for the storage procedures in the project, much of this the verification work involved setting up scripts and policies within SRB. The first step was fairly manual – physically replicate the data to the different storage systems and then verify checksums on the data to verify that nothing had changed during the process. This was done and no problems were detected. The next step was to register all of the data into SRB; this means that SRB is made aware of the data and storage locations.

The final step in the process was the creation of scripts to run within SRB so that it could monitor the data. SDSC staff wrote three initial scripts. The first script relied on the function "Sls." This test verified that files on physical file systems existed in the storage catalog. The second test relied on the function "Sstat," which verified that files in the catalog existed on physical file systems. A third verification script, “Schksum,” looked for any changes within the file on a filesystem. These three tests in combination provided a way to check all possible changes made, in SRB or directly on the filesystems.

For reporting and statistics, whenever the scripts detected changes within the system, an email was sent to the SDSC administrators with the affected files cited.

Figure 3 shows an example of an email detailing changes detected by the system:



From: John Schroeder
Subject: **SRB Cron Job ERROR: SRB physical file ERROR**
Date: December 17, 2006 2:37:10 PM PST
To: Adam Lathers , Alex Wu , Christopher Jordan , John Schroeder , David Minor

Dear LC-SRB User(s),

One or more errors were detected in the data integrity of the files in SRB. Please check the latest Sstat log file for a list of files and replicas that had errors.

This test may give false positives if the server took to long to respond. Double check the results before beginning the recovery process!

To double check the results, you can do the following:
1) Login to lctb1 and go to the "/archive/loc/data/johns_scripts/cron/logs" directory
2) Run the command "checklog.py X" where X is the log file you wish to check

-Sstat cron job

-n 3 /Library-of-Congress/home/LoC_data/Prokudin-Gorskii/mnt.pnp.prok/01400/01442t.gif
-n 3 /Library-of-Congress/home/LoC_data/Prokudin-Gorskii/mnt.pnp.prok/10100/10131v.jpg

Figure 3: Email of Detected Change

3.3 Script Roll-Out

One all of the scripts were setup, SDSC conducted several “in the blind” tests: at unannounced times several staff members logged in to the system and deleted, added or moved files. In all cases the monitoring system was able to detect the changes and report them to the proper staff people.

The second planned data verification process was to use these monitoring scripts to track the changes that the P&P staff were making during their file manipulation scenario. Since these manipulations were happening unannounced, SDSC staff had no way of knowing when they were occurring or what was being done. Thus the monitoring process became the key part of detecting this: when the P&P staff began making changes, the system alerted staff at SDSC of these changes successfully. Emails were sent to the administrators of the storage systems who were able to verify that the changes were made.

This process highlights an important lesson that this process produced: computers are very good at detecting when changes occur in a storage system. This can be done at a number of different levels. Computers are not good at determining whether these changes are *intentional* or *desired*. At almost all points in the monitoring process, LC staff asked whether things could be setup to automatically replace files that were found to be missing or changed. The problem is that without interacting with a real human, there is no way that the system can determine whether the change was intentional or not. SDSC could have setup a “master” version of the data which was always to be considered authoritative, with all changes compared to it. However, it would still need to be the case that any changes made to this authoritative copy be vetted against a human check.

With this in mind, near the end of the project, LC staff made the decision that the data at SDSC was “stable,” and should never change. Only after this decision was made could any change be automatically designated as a *bad* change.

3.4 Data Monitoring - An Unplanned Event

There was an unplanned test of data verification at SDSC during the project. At one point, one of the main storage systems (SAM-QFS) began having intermittent problems with its metadata services. This was something that had not happened previously and took several weeks to solve. During the period when it was being examined, the decision was made to take the SAM-QFS filesystem offline completely, rather than try to provide a resource which was unpredictable and unreliable.

Normally this would have been no issue for the LC’s data stored at SDSC: SAM-QFS was only one of three separate filesystems on which the data is stored. And, even on the SAM-QFS filesystem, there was never any threat of data loss or corruption – it was the metadata servers that were malfunctioning.

However, this problem exposed a flaw in how SDSC had designed access to the data for LC staff: the primary front-end for the data was itself located on SAM-QFS. This meant that whenever SAM-QFS was down, access was restricted,

even though there was nothing bad happening to the data itself. Once this flaw was found, SDSC took corrective measures to eliminate this point of failure.

This process did point out two important issues which are central lessons learned for the project:

1) There is a very real difference between high reliability and high availability for data. It can be said with certainty that the LC's data was being stored very reliably at SDSC: at no point was there any threat of data loss or corruption because of this problem. However there were periods of time when the data was not available to the people who needed it. This obviously caused frustration. Beyond any kind of semantic hand waving in the process, SDSC needs to be aware of, and plan for, any potential points of access failure which might prevent the LC from being able to work with its data. Again, the actual root cause of the access problem was trivial to fix (mere minutes in fact). The issue arose because of a planning decision. If this problem had been anticipated, the LC would never have encountered any problems accessing their data.

In the long run, there needs to be a conversation concerning "single points of failure." During the aftermath of the SAM-QFS problem, there were several discussions about what pieces of the infrastructure between the LC and SDSC can go down, and what this might mean operationally. As two large, complex organizations, there are obviously a multitude of "things that can go wrong." For long term planning it would be enlightening to be explicit about what consequences different outages might have. While eliminating all failure points between cross-continental sites is an impossible goal, there still needs to be a conversation beforehand which indicates where problems might arise.

3.5 Trust Elements

There are a number of trust elements in this part of the project. Indeed, in many respects, the interaction of the data remotely by LC staff is the *core* of the whole trust issue; it gets at the point of the enterprise: can we give you our stuff and trust you to protect it? The most obvious atomic elements of this question are:

- Can remote data be accessed?
- Can remote data be verified?
- Can remote data be retrieved and re-used?
- Can ownership be clearly defined?

All of these have implications for real world actions and they must clearly be understood between the organizations. In many ways however they are what one would expect in a process like this. The more interesting and unexpected element of trust in this area involves the interaction of machines and humans. It is in many ways definitional: who defines changes as deliberate, desired, accidental, malicious, etc. Storage systems can be designed which are infinitely

robust and sophisticated. At some level however decisions need to be made by humans outside of the mechanical system. When and how this happens needs to be well-understand by all participants and planned in advance.

4. Data Access and Manipulation

4.1 Front End Design

As noted above, SDSC put in place an accessible storage environment and active logging and monitoring procedures. While these pieces were being implemented, SDSC staff worked with staff from the Prints and Photographs division of the LC to determine the best access methods. As part of the default install at SDSC, command line access (i.e. access via a UNIX shell) was available. This would provide baseline access to the collection for anyone designated as authorized.

Emir of Bukhara. Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-21886

There were several potential problems with this basic type of access however. First, because the SDSC storage setup was actually comprised of multiple replications stored on multiple filesystems, simple command line access was actually quite complicated - it required the use of a new set of commands which were enhanced to handle the advanced capabilities. Second, because of the requirements for logging and auditing, it seemed beneficial to put together a system which provided ready access to reports and data about the collection.

For these reasons, it was decided that SDSC staff would look into front ends to the data which would be more user-friendly. LC and SDSC staff held several meetings to discuss the possibilities available for interfaces for the Prokudin-Gorskii collection. This was to insure that whatever interface was provided met the expectations and needs as outlined in the proposal. Since this scenario had a number of explicit deliverables for logging and auditing, it was critical to make sure that these were covered. After discussing various options, the group decided on the following approach:

SDSC would continue to provide a UNIX command line interface to the data which would be essentially the same as what some LC staff currently use. Documentation and assistance would be provided for this. This access was appropriate for “expert” users who are familiar with UNIX and who may need

enhanced functionality for manipulating groups of files.

SDSC would also provide a web portal to the data. This allowed staff at the LC to interact with the data without having to use the command line. This interface provided the fuller logging and auditing capabilities as required by the proposal. After reviewing a number of options, LC and SDSC staff decided that the most appropriate would be the GridSphere Portlet software (http://www.gridsphere.org/gridsphere/docs/javadocs/org/gridlab/gridsphere/portlet/package-summary.html).

Figure 2 below is a screenshot showing the GridSphere application displaying a subset of the P-G data. Of interest is that this is a detailed view of the data, providing a list of the locations where the files are located. This is an easy way for an administrator to verify the existence of the data in all of the storage locations as well as see the dates of creation.

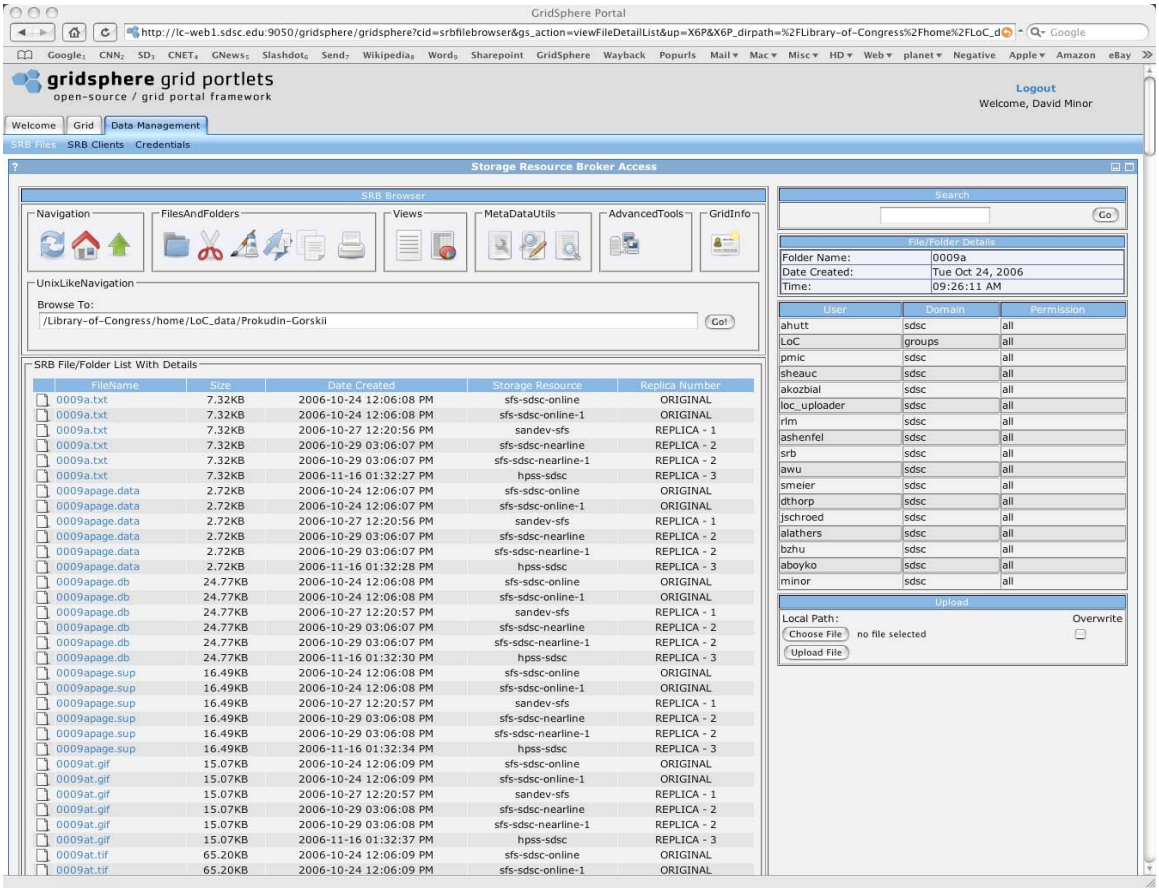


Figure 4: Detailed data view in GridSphere

The process of installing and configuring the Gridsphere interface highlighted a number of unexpected issues. For example:

It required the LC staff that would be accessing the data to generate certificate-based credentials to access data at SDSC. This process takes about 10 minutes to complete, and is comprised of several discrete steps which must be completed in very specific ways. This process was new to the LC staff and required active support to complete. The security credentials were also something that SDSC staff needed to actively monitor for changes and problems.

Because the interface displayed all copies of the data at SDSC, it quickly raised questions of which one should be the “master” copy to which all the other copies should be synced. Further, it raised the question of whether all changes to a copy at SDSC should be replicated automatically, or whether there needed to be a process to manually validate changes before they were replicated.

Even though the new front end had been tested and approved, there were still several usability issues with it. First was speed - for the users at the LC, it was slow to react at times, even for trivial tasks. Second, it didn’t always provide the amount of feedback they wanted when interacting with the data. Finally, because the front end was designed to be a simple interface to a complicated back-end, there were general questions about exactly what the interface was doing - was it changing one copy of the data at a time, or all copies?

These are examples of interface refinement which would need to be made if this was a longer term process. For instance, several people commented that the interface wasn’t sorting data in a way that was useful for them. There were also comments made that there needed to be better error messages and progress bars. This kind of feedback is a normal part of software development and would be changed if needed.

Other questions arose simply because of differences in a new system. For instance, “This doesn’t look/act/work in the way that I am used to.” Clearly this would be the case with almost any new interface or piece of software.

These issues also brought to light the fact that the creation of an interface was not an original goal of the project, and had grown organically without a precise goal in mind. Because of this there was no intention to have a detailed planning process to change the front end based on user feedback.

Given the above points, staff at SDSC was interested in finding ways that the GridSphere interface could be improved to respond to the issues raised. However, it was decided that further work on the interface was not appropriate at this time. The main reason was because the focus of the project was demonstrating that SDSC can provide reliable storage and access to this storage for the LC. Whether or not GridSphere is the specific tool that would be used for this access in the long run is an open question. It should also be noted that there are a number of allied questions here, including how this process would mesh with current LC processes for file manipulation. There were several comments

made by the P&P staff members that, “I don’t normally do this kind of thing.” If there were long term plans for implementing tools an important foundation would be the current processes of the staff members involved.

4.2 File Manipulation Scenarios

Once the storage was ready and the front end was in place, LC staff started working on the file access and manipulation scenarios. These tests were simple in concept and intended to be examples of real work that would be done with a collection at the LC:

- Locate Files (Technical Direction Letter, section 4.4.3.1)
- Replace Files (TDL section 4.4.3.2)
- Delete Files (TDL section 4.4.3.3)
- Add Files (TDL section 4.4.3.4)
- Change properties of sets of digital files (TDL section 4.4.4)

Phil Michel of the Prints and Photographs Division completed all of these tasks. Some were done in conjunction with other members of his group at the Library; some were completed by Phil working alone. The important point to note here is that the SDSC system was able to detect all changes made to the data and reported them via email to SDSC system administrators and to Phil. This was done using the logging tools noted in the above section.

4.3 Reporting Issues

As noted, LC staff were able to successfully complete all of the designated tasks for the Prokudin-Gorskii collection. In fact in many ways this part of the process was the most trivial aspect - much of the work was in setting up the system to make it useable, not actually using it. The process did raise several questions which were not expected.

First, the logging and auditing information that SDSC provided actually proved to be too much. The emails that were originally generated were very long and verbose. Because they had been designed originally for system administrators they contained more information than a normal “user” would want. Based on the first round of tests, LC staff requested that there be different levels of email that would be sent out: one level would be appropriate to someone like Phil Michel, who might want to know when and if files had been changed, and which ones they were. He then could take this information and make judgement on whether they were appropriate changes or not.

The LC also requested a very high level, “manager’s email,” which would contain almost no details. Instead it would contain just a few sentences indicating whether any changes to the data had been made or not. These messages would

be sent out every two weeks regardless of whether there had been work done on the data.

Functionally the creation of these emails was straightforward. They raised however an important issue which needs to be considered: this type of reporting procedure is an additional process which occurs separate from the actual storage and computing processes. This issue was brought to light at the end of the project when SDSC was ramping down its work. The scripts which were monitoring the data for the project were turned off. The email-generating scripts which were watching the log files were left running however. At the same time, staff at the LC decided that they'd like to do another test of the data without advance warning to SDSC staff. In doing so they were expecting to receive a message which indicated that they had made changes to the system. However, the email they received did not indicate the changes, because the reporting systems were no longer being maintained at SDSC. This was a minor issue and one mostly of communication at the very end of a project, but it highlights the kinds of issues that can arise within a complex environment.

4.4 Questions on the Nature of the Archive

Following from the issue of authentic changes to the data, SDSC and LC staff had extensive discussions on the nature of the remote storage system itself. One of the first possibilities brought up by LC staff was the the data at SDSC should be treated as an "inactive" copy. That is, the data stored there should never change. In this environment, any change would be defined a "bad." This would seem to solve many questions of authenticity and change management.

Obviously this kind of environment offers less function and fewer options for data stored there. Ignoring this for now, there are still a number of questions that need to be answered. The most pressing is *still* the issue of synchronization: perhaps there is no synchronization needed at the remote end, but how does the remote data stay synchronized with the original copies of the data at the LC? If the data at the Library is ever changed then there needs to be a process in place to synchronize it. (Or allow for the possibility that the data will all need to be transfered again.)

These questions highlight some of the main functional questions which need to be studied in the future: what is the role of the remote storage? Is it simply a "backup," there for disaster recovery? Is it a remote repository with shared management between the institutions? Or is it a remote data center which has its *content* owned and managed by the library but the underlying *infrastructure* managed by the remote staff. All of these questions will impact how things are managed and monitored.

4.5 Trust Elements

Most of the trust elements in this section of the project had to do with separate organizational staff working together. Creating something like a shared front end application highlights the fact that there can be widely divergent expectations on the part of both organizations. Sometimes ideas grow out of a project organically. This doesn't mean however that active planning and vetting of all parts of the process aren't needed.

Three generations. A.P. Kalganov with son and granddaughter. The last two work in the shops of the Zlatoust plant, Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-20542

5. Web Archives

5.1 Working in Parallel

The first phase of this part of the project included the transfer of approximately 5 terabytes of web crawl data. This data had originally been created by the Internet Archive for the LC and was based on items related to the 2004 Elections in the United States. In total the collection included approximately 40,000 individual ARC files. In the initial stages of the project it was decided that SDSC would take a copy of this data and index and display it. The end result had to "look" and "act" like it would normally to staff at the LC. To do this SDSC would install and configure the Heritrix and Wayback Machine software. The amount of data to be used was higher than the LC typically worked with and they were particularly interested to see if SDSC could come up with novel solutions for it.

As noted in the data transfer and storage sections above, SDSC successfully ingested this data and replicated it throughout its storage system. There were then several additional steps which were required. The first was the indexing of the data, using the Wayback Machine software. The installation of this software was fairly straightforward, but the subsequent indexing process turned out to be more difficult and involved than originally expected. There were two reasons for this:

- Several software bugs were found in the indexing software supplied by the Internet Archive. These bugs had not been encountered before because SDSC was using a newer version of the software than had been used at LC. Once these bugs were found, the information was communicated

back to the Internet Archive team, who acknowledged them. SDSC staff who found these bugs also re-wrote some pieces of the original Internet Archive source code in order to get around the bugs, rather than waiting for them to get fixed in the shipping version.

- The size of the collection and the large number of files in it proved to be a new challenge for the indexing software. The LC had not worked with a collection of this size and didn't know what to expect. In order for the collection to be indexed it had to be broken into 18 separate parts which were then indexed separately. This was a solution arrived at jointly between SDSC and IA staff.

The second item proved to be a difficult problem with an interesting solution. If SDSC had used the standard method of indexing for the web crawl collection, it would have taken approximately 40 days of compute time to complete. Instead, by breaking the collection in 18 separate indexes the indexing time dropped to about 7 days. The team arrived at using 18 different indexes after extensive testing of various configurations.

Figure 1 is a graphical representation of a default installation of the Wayback software. This is what was first installed at SDSC.

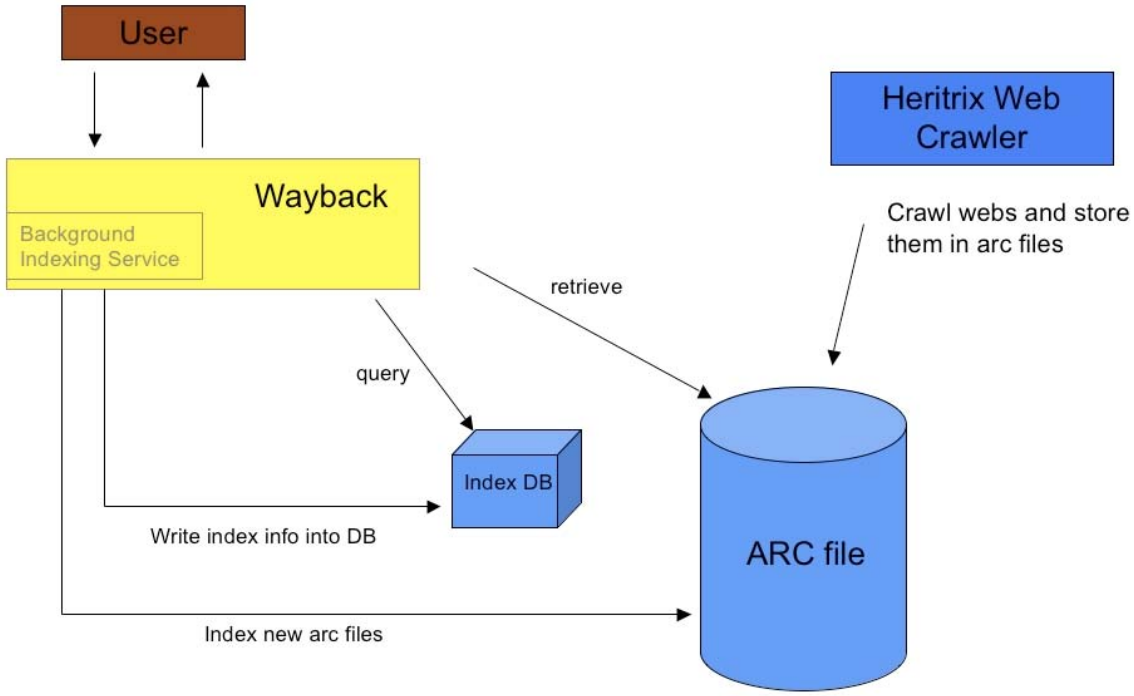


Figure 1: Default Wayback Installation

Figure 2 is the re-configured Wayback system that SDSC staff installed.

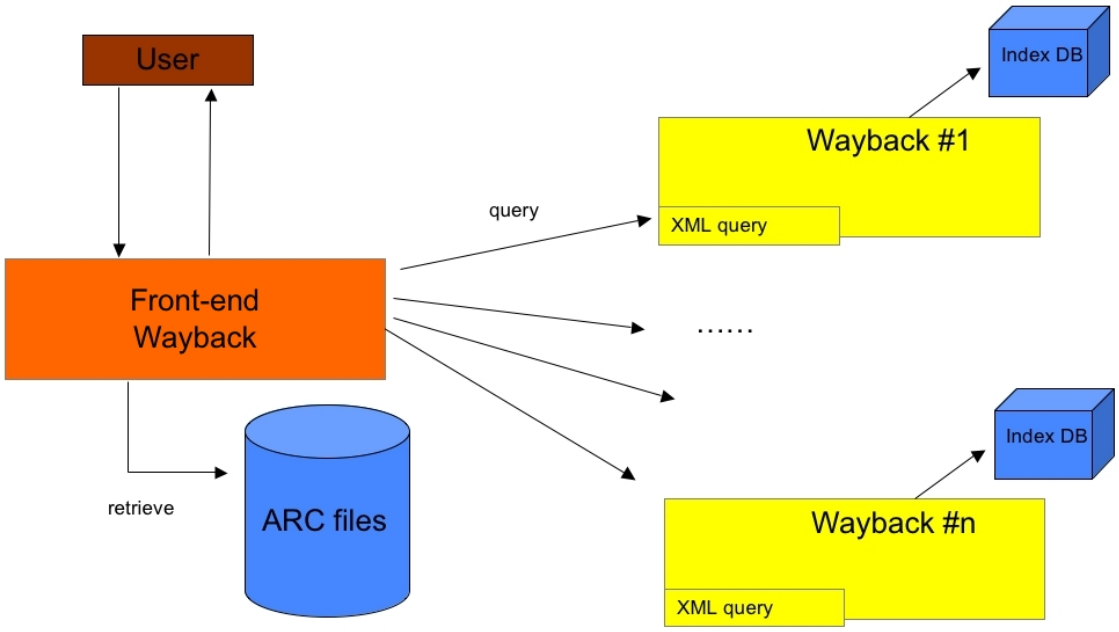


Figure 2: Re-configured Wayback

While these indexing problems were being solved, the web-based front end to the data was installed and configured to point to the 18 different Wayback machines. This process went fairly smoothly.

After the indexes and software were installed, several stress tests were conducted on various aspects. First, LC and SDSC staff held a joint “real world” test which consisted of 12 separate users accessing the data via the internet. The results of this data were tabulated and summarized. The broadest summary of the results is that there was no discernable difference whether users were “seated” at SDSC or at remote sites. There was however a noticeable speed drop off when more than 2 or 3 users were accessing the same files at the same time. Questions remained whether 12 people accessing the same files at the same time represents something likely to happen in the real world, but it still provided useful information.

SDSC staff also tested the web crawl data using an application, JProfiler. More information about JProfile can be found at http://www.ej-technologies.com/products/jprofiler/overview.html.

Running this tool provided a very interesting result: when requests for archive files were made, there were enormous numbers of individual file requests made. For a single web page call, as many as 263 requests could be made to the indexes. This large number of requests would certainly explain the slowness which more users produced.

In order to fix this problem, SDSC staff tried two approaches. First, they merged the 18 separate indexes into one large one. This had the advantage of simplicity; however it also meant that there was a very large database file (over a terabyte in size) which must be searched every time a request is made. This created a slow process for users trying to access individual files.

The other approach was a slight reconfiguration of the way that requests are handled. In this new configuration, when a user queried the archives for a web site, the front-end Wayback software queried the backend Waybacks for index information. The software then merged the queries and only a single page request was actually re-directed to its corresponding site in the original Wayback. Note that the backend Waybacks were normal Wayback machines without any modification; only the front-end Wayback interfaces had to be modified. This solution worked well and eliminated most of the multiple-redirection problems.

Figure 3 shows this new approach.

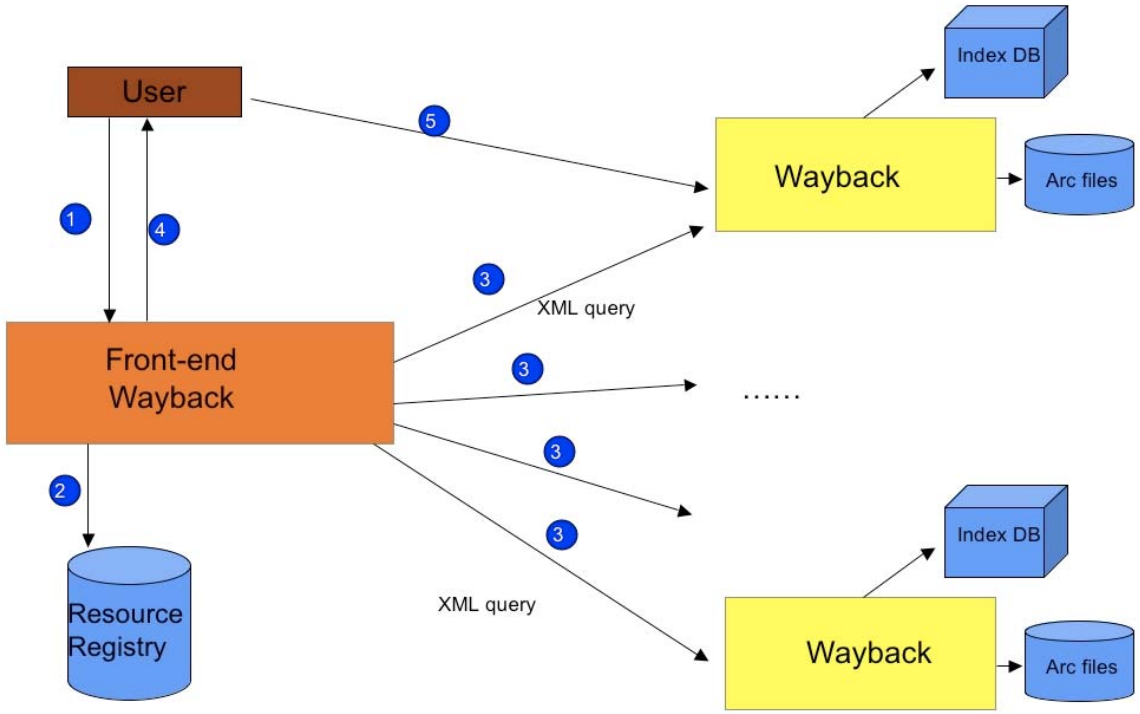


Figure 3: New method of indexing

This updated method of accessing web crawl data proved to be very interesting to the Internet Archive staff. They looked at several different parts of the solution and added them into the shipping version of the Wayback software. They felt that the solutions the LC/SDSC team came up with were important because more and more organizations are going to be working with data collections of this size and will need similar solutions.

5.2 Trust Elements

There were several elements of trust involved with Wayback section of the project. The primary ones were:

- *Are the final results the same?*
  Because much of the work with the web archive material was actual computations, the first question which always needs to be asked is, “are the results from the different processes the same?” Regardless of how fast things are completed, the results are the only barometer that truly matters.
- *Can the results be reached in a better way?*
  While it would have been useful for SDSC to exactly duplicate the processes already in place at the LC, it wouldn’t have been particularly interesting. It also highlights an important cross-organizational issue: can any new institution demonstrate a familiarity and facility with the LC’s data that allows them to recommend new solutions? In other words, is the institution able to provide services which are smart and contribute to the LC’s own processes?
- *Can a new organization bring new expertise?*
  Another way of stating the above point: can any new organization bring to bear new expertise to an established process which improves or enhances it? Does this new organization bring value-added services to the relationship that allow the LC to consider new work or better services for its own customers?
- *Can a new organization work with your partners*?
  Finally, does a new organization have the sensitivity to work with the LC’s existing partners? In this case, SDSC worked closely with the staff of the Internet Archive to provide new solutions. It should never be the case that an organization comes on board and alienates or frustrates existing partners and thus strains relationships.

Conclusions

In conclusion, this paper attempts to summarize the work that was completed for the project. It has been written specifically from the SDSC perspective. That is to say, it discusses what SDSC, as the provider of services,has learned from the project. Staff at the LC or CACI might indeed have the same recommendations and experiences, but no assumption is made to this effect.

Staff at SDSC have learned greatly from this project and are looking forward to an ongoing relationship with the Library of Congress and its partners.

On the handcar outside Petrozavodsk on the Murmansk railway, Sergei Mikhailovich Prokudin-Gorskii Collection, LC-DIG-prokc-20245