Three Case Studies of Large-Scale Data Flows

William Y. Arms[2] Selcuk Aya[2] Manuel Calimlim[2,4] Jim Cordes[1] Pavel Dmitriev[2]
Johannes Gehrke[2,4] Chris Jones[3] Dave Lifka[4] Julia Deneva[1] Lawrence Gibbons[3]
Mirek Riedewald[2] Dan Riley[3] Anders Ryd[3] Gregory J. Sharp[3] Valentin Kuznetsov[3]

[1]Departments of Astronomy [2]Department of Computer Science
[3]Department of Physics [4]Cornell Theory Center
Cornell University

Abstract

We survey three examples of large-scale scientific workflows that we are working with at Cornell: the Arecibo sky survey, the CLEO high-energy particle physics experiment, and the Web Lab project for enabling social science studies of the Internet. All three projects face the same general challenges: massive amounts of raw data, expensive processing steps, and the requirement to make raw data or data products available to users world-wide. However, there are several differences that prevent a one-size-fits-all approach to handling their data flows. Instead, current implementations are heavily tuned by domain and data management experts.

We describe the three projects, and we outline research issues into opportunities to integrate Grid technology into these workflows.

1 Introduction

The Grid with its associated tools holds great promise for simplifying the development and deployment of large-scale data-driven workflows. At Cornell, domain scientists from astronomy and physics and computer scientists are working together on three large-scale workflows: the Arecibo sky survey, the CLEO high-energy particle physics experiment, and the Web Lab project for studying the evolution of the World Wide Web. All three projects involve with massive amounts of data which are growing rapidly; they all have sophisticated data processing pipelines that meld raw data through expensive processing steps into finished data products. For all project, the consumers of both the raw data and the data projects is a distributed community of scientists who are located all over the globe.

However, despite these similarities, there are also striking differences between the workflows, and at this point, each group has built its own custom solution. In this paper, we survey the existing solutions and our experience with the three workflows, and we outline research challenges derived from these problems. In particular, it is the goal of this paper to stimulate discussions at the workshop about novel research directions motivated by real applications.

The remainder of this paper is organized as follows. We first survey the workflows associated with the three projects: The Arecibo Sky Survey (Section 2), the CLEO High-Energy Particle Research (Section 3), and the Web Lab (Section 4). We conclude with next steps (Section 5).

2 The Arecibo Sky Survey for Neutron Stars

This project makes use of the Arecibo Telescope in Puerto Rico, the world's largest radio aperture, as the source of data for several astronomical surveys. The upgrade of the telescope and installation of a 7-feed array mounted at the focus (the Arecibo L-band Feed Array, ALFA), operating at 1.4 GHz makes the pulsar survey the most sensitive ever done. The survey commenced in early 2005 and will continue for at least five years, producing about a Petabyte of raw data. Processing to identify pulsars and transients yields data products about one to a few percent the size of the raw data that, in turn, are subjected to a meta-analysis that discriminates between and classifies terrestrial interference and astrophysical signals.

2.1 Arecibo Data Flow and Challenges

The Arecibo data flow consists of several major data acquisition, transport, and processing steps:

1. Acquisition of dynamic spectra at the telescope and recording to local disks.
2. Initial local processing for quality monitoring and for making preliminary discoveries.

3. Transport of raw data to the Cornell Theory Center (CTC) for archiving, processing and dissemination to other processing sites.
4. Further processing of the data using proprietary algorithms at several member institutions of the "Pulsar ALFA" (PALFA) Consortium (including Cornell); member institutions are world-wide.
5. Consolidation of processing data products at the CTC for meta analysis.
6. Incorporation of data products into a database that facilitates the meta analysis.
7. Long term archiving of raw data and data products for reprocessing, which is common for pulsar surveys, and for cross wavelength studies now and in the indefinite future. This involves connection with the National Virtual Observatory.

Data are obtained in observing sessions of 3 hours, once or twice a day for periods of one to two weeks, yielding tens of Terabytes of raw data. To ensure data quality against spectrometer functionality, proper signal levels, and interference that contaminates signals to highly-varying degree, data is analyzed locally at the Arecibo Observatory. There is some interest to perform full pulsar-search processing at Arecibo on some of the data. Primary reasons are (a) one or more pulsar astronomers involved with the project will be in residence at Arecibo who have an interest in being involved with the processing; (b) local processing reduces demands on observatory staff to ship all data promptly and (c) any initial pulsar candidates found could be confirmed during the same telescope session. For the most part, however, processing of raw data will require off-island resources primarily because the resource demands are high.

Processing consists of data unpacking, dedispersion, Fourier analysis, harmonic summing, threshold tests to identify candidates, reprocessing of dedispersed time series to signal average at the spin period of a candidate signal, and investigation of the time series for transient signals that may be associated with astrophysical objects other than pulsars. In addition, interference from terrestrial sources needs to be at least identified and most likely removed from the data. To do so requires development of new algorithms that simultaneously investigate dynamic spectra for each of the 7 ALFA beams and apply tests of different kinds. Overall about 50 to 200 processors would be needed to keep up with the flow of data. However, these numbers are only for the basic analysis and do not include additional, possibly substantial overhead from Radio Frequency Interference (RFI) excision. Finally, another level of complexity comes from addressing pulsars that are in binary systems, for which an acceleration search algorithm also needs to be applied.

To further refine pulsar candidate signals, usually about 0.1% of the raw data volume, before they are confirmed on the telescope, a meta-analysis is needed to cull those candidates that appear in multiple directions on the sky. With past experience, we find that spurious signals take a wide range of forms, those that are obvious and easy to recognize in the first stages of the analysis, to subtle cases that are appear very sporadically and can mimic astrophysical signals uncannily well.

Storage requirements during the different processing steps are even more challenging. A useful data block consists of the ~ 400 telescope pointings obtained in one week, or about 35 hours of telescope time. The corresponding raw data require 14 Terabytes of storage. Dedispersion entails summing over the frequency channels with about 1000 different trial values of the "dispersion measure," each yielding a time series of length equal to the original number of time samples. These time series require storage about equal to that of the original raw data. The processing is iterative, requiring operations on both the dedispersed time series and the raw data, so a minimum of 30 Terabytes of storage is required instantaneously.

2.2 Current Solutions and Future Challenges

As discussed in the previous section, most of the processing has to happen off-site. Unfortunately, because of Arecibo's limited network bandwidth to the outside world, for the forseeable future, network transport of raw data is infeasible. We therefore have developed a system based on transport of physical ATA disks with raw data. The main issues of data transport are: personnel requirements; assessment and maintenance of data integrity; tracking and logging of data contents; ensuring no data loss; and developing a database of utility into the indefinite future.

The raw data disks are transported to the CTC, where their contents are archived to a robotic tape system and retrieved for processing. Some of the raw data and data products are also distributed to participating PALFA member organizations. The large number of data products (data diagnostics and plots, test statistics, candidate lists, confirmation analyses, etc.) that are created for each telescope pointing, are loaded into a MS SQLServer database system at the CTC. The database is accessed through a Web-based server and will provide the tools for meta-analyses. It currently supports interactive groupings of candidate signals, tests for correlation or uniqueness of the candidates, and generation of appropriate plots for accomplishing the combination of pattern recognition and statistical analysis required. Eventually the entire processing pipeline will be controllable from the Web-based system.

The current transport and processing pipeline, as described above, is adequate but requires a great deal of intervention by personnel at Arecibo, in the Astronomy Department at Cornell, and at the CTC. We need to continue to automate more of the steps. The largest challenge is to evolve the entire system into a sustainable structure that will allow (a) room for growth when data rates using a new spectrometer will increase; (b) provision of tools for the diverse analyses that PALFA Consortium members will want to conduct over the duration of the survey; and (c) development of infrastructure for linking the PALFA data sets to the National Virtual Observatory (NVO). Connecting the CTC database system with the NVO requires particular XML-based protocols that have been developed by the NVO Consortium. We are currently developing tools that use these protocols.

The raw data and data products will be valuable for the indefinite future, providing tremendous opportunities for multi-disciplinary astrophysical studies that will link satellite telescope data with the Arecibo data. For example, the next generation gamma-ray telescope (the Gamma-ray Large Aperture Space Telescope, GLAST), to be launched in 2007, will be a discovery and analysis instrument with direct interest for the Arecibo project. We must ensure flexibility and transparency of the tools for use by a broad constituency of researchers.

Another aspect of the large Arecibo data sets is the prospect for discovering entirely new classes of signal and thus astrophysical objects. Exotica such as evaporating black holes, transient emissions from extrasolar planets, and signals from other civilizations are examples of potential serendipitous discoveries that may be made if adequate tools for exploration exist.

A key issue for these and other considerations is the migration of the data to new storage technologies as they emerge. Storage media costs undoubtedly will decrease, but manpower requirements for migrating the data are significant and care is needed to avoid loss of data.

3 The CLEO High-Energy Particle Research Project

CLEO is a high-energy particle physics (HEP) research experiment at Cornell University [1, 9]. The CLEO collaboration includes over 150 physicists from more than 20 universities, studying the production and decay of beauty and charm quarks and tau leptons produced in the Cornell Electron Storage Ring (CESR) [2, 8]. The collaboration makes some of the most sensitive tests of the Standard Model of elementary particles, key to understanding the forces of nature and the fundamental structure of matter [7].

The primary goal of CLEO software is to produce physics analysis results of the electron-positron collision events detected by the CLEO detector. Delivery of physics analysis products follows complex work and data flows that have evolved over the past 30 years.

3.1 CLEO Data Flow and Challenges

CLEO has accumulated more than 80 Terabytes of data. This is a massive data collection, but nowhere near Arecibo's Petabyte-size storage requirements. The biggest challenge in CLEO lies in its complex data processing workflow, as summarized in Figure 1 (red arrows indicate data flow). Due to space constraints, it is not possible to explain it in detail, but we will highlight important aspects.

The current data analysis software comprises several million lines of C++ code. The processing steps include

1. Acquisition of runs of particle collision measurements and initial analysis.
2. Reconstruction of the runs; followed by computation of post-reconstruction data for each run.
3. Generation of Monte-Carlo random data for each run.
4. Physics analysis, performed either locally or remotely.

These steps are explained below. They produce many different data products: raw data of the detector response to electron-positron particle collisions, detector calibration data, data from Monte Carlo simulations of the detector response, centrally produced derived data (known as reconstruction), and the output of the physics analysis that depends on all of the other data types.

Raw data are the detector response to the particle collision events measured by the CLEO detector. They are stored in units known as runs. A run is the set of records collected continuously over a period of time (typically between 45 and 60 minutes), under (nominally) constant detector conditions. A run worth analyzing typically comprises between 15K and 300K particle collision events.

A reconstructed run is produced by processing the raw data for a run. Each event in a reconstructed run comprises many sub-objects, which may in turn have sub-sub-objects. An atomic storage unit (ASU) is the smallest storable subobject of an event. An ASU will never be split into component objects for storage purposes.

In addition to the reconstructed data files, post-reconstruction values are also produced and stored. These values depend on the reconstructed data, and so cannot be calculated until after reconstruction. There are typically a dozen ASUs per event in the post-reconstruction data. For each version of the reconstructed data, there may be several versions of the post-reconstruction data.

The processes for reconstruction and physics analysis require iterative refinement. With each iteration the knowledge about the measured collision event increases, e.g., which particles were produced during the collision, and hence another iteration might be triggered. When starting a new analysis, a physicist normally wishes to use the most recently produced version of the analysis software and the

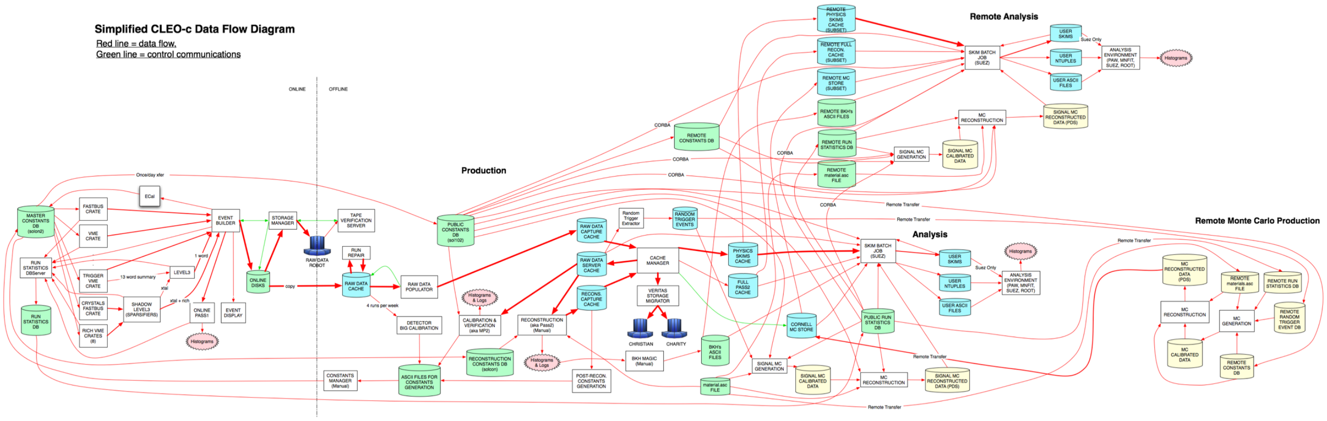

Figure 1. CLEO data flow

corresponding version of the reconstructed data. For reproducibility, it is critical that in each iteration, the physicist's analysis job makes consistent use of software versions and accesses the same information as previous iterations (unless explicitly told to do otherwise).

The tradition at CLEO has been to continue to use the version of the data an analysis was started with throughout the lifetime of that analysis, rather than periodically updating the entire analysis to newer versions (unless there is a compelling reason to repeat the analysis with a newer version of the reconstructed data). With multiple newer versions of the reconstructed data appearing during the lifetimes of some analyses, this can impose a substantial burden on the physicist to track which versions of of the data were used, particularly as data collected after the analysis began are added. Supporting provenance is challenging. The centrally managed reconstruction processes always process a run as a unit, and each processing step writes one or more complete files per run, so all the data in a given output file have identical provenance. This is not true of the later physics analysis processes, which can process individual ASUs differently. Keeping track of provenance at this level is infeasible (see also discussion below).

Similarly, physicists need fast and reliable access not only to provenance information, but to other metadata as well. For instance, often a researcher is only interested in certain types of collision events. The challenges in handling metadata are how to efficiently manage this evolving information and how to provide fast access to it.

In addition to these provenance and metadata management challenges, there is a need for making each iteration of reconstruction and analysis as fast as possible. The physicists have designed sophisticated caching and partitioning schemes to achieve this goal. For example, CLEO data are partitioned into hot, warm and cold storage units. This is a column-wise split of the event into groups of ASUs, based on usage patterns. The hot data are those components of an event most frequently accessed during physics analysis; these ASUs are typically small compared with the less frequently accessed ASUs. Discussing all these optimizations in detail is beyond the scope of this paper. Each one is based on a careful analysis of typical workloads and cannot be easily achieved by general-purpose data management tools.

3.2 Current Solutions and Future Challenges

CLEO recently implemented a new data management system, called EventStore [9]. EventStore is primarily a metadata and provenance system, designed to simplify many common tasks of data analysis by relieving physicists of the burden of data versioning and file management, while supporting legacy data formats. Data stored in the various formats are managed such that physicists conducting analyses are always presented with a consistent set of data and can recover exactly the versions of the data used previously. EventStore is accessed via a plug-in module to the data analysis software.

In order to support a variety of use cases, the CLEO EventStore comes in three sizes, tailored to the scale of the application: personal, group and collaboration. The only user interface differences between the three sizes is the name of the software module loaded, which is also the first word of all EventStore commands. Provenance data are stored in the data files using a simple extension to the standard CLEO data storage system. Other metadata about the data are stored in a relational database supporting the standard SQL query language. All but the lowest layers of the database interface code are independent of the database implementation, allowing transparent use of an embedded database (SQLite [11]) in the standalone versions and a standard relational database system (currently MySQL [6] or MS SQL Server [5]) in the larger scale systems.

The personal EventStore was originally meant to manage user-selected subsets of the data on a personal system such as a laptop or desktop. It is designed to provide the versioning and metadata query facilities of the EventStore with minimal overhead. The relational database is implemented using the embedded SQLite database, making the personal EventStore self-contained in the EventStoreModule package and supporting completely disconnected operation. To support iterative and collaborative analyses, the system is designed so that merging the results in a personal EventStore into one of the larger scale systems is a quick and reliable operation. Somewhat to our surprise, merging became the fundamental operation for adding results to the group and collaboration stores. Rather than having long-running jobs hold lengthy open transactions on the main data repository, it proved simpler to create a personal EventStore for the operation, which is merged into the larger store upon successful completion of the operation. This stratagem allowed the highest degree of integrity protection for the centrally managed data repositories with the fewest modifications to the legacy data analysis applications.

The EventStore organizes consistent sets of data by associating a list of run ranges and a list of version identifiers for each run range with a data grade. Assignment of data to grades, particularly to the "physics" grade, is an administrative procedure performed by the CLEO officers. The evolution of a grade over time is recorded, so a consistent set of data is fully identified by the name of a grade and a time at which to "snapshot" that grade. For an analysis project, a physicist will usually specify "physics" grade data and use the date the analysis project started (e.g., 20050501) as the timestamp, so that the same consistent version will be used throughout the lifetime of the project. EventStore finds the most recent snapshot prior to the specified date, so the date specified is not limited to a set of "magic" values.

If some of the data are reprocessed, that change will not appear in the snapshot used by the analysis unless the physicist changes to a date after that modification was made. However, data added to the dataset for the first time, such as data recently taken and reconstructed for the first time, or the addition of a new object, will appear in the snapshot. This is done so that a physicist can add data collected after the beginning of the analysis without having to change to a later timestamp.

EventStore records certain basic attributes of the experiment conditions and data quality that are frequently used for selecting which data to use for a particular analysis. Since these data are independent of the processing steps in the data provenance, they can be queried via a Web Services interface from any of the different sizes of EventStore so long as an Internet connection is available. This allows most data selections to be expressed in physics terms, so for most CLEO analyses the data set used can be fully specified by a simple selector on experiment conditions and a datestamp, closely matching the terms CLEO physicists use to informally describe commonly used sets of data.

To address the data versioning and other provenance issues, EventStore attaches versioning information to the derived data, identifying how the data were produced. As an example, the version identifier Recon-20040312-Feb13_04_P2 indicates that the data were produced by the Feb13_04_P2 release of the reconstruction software, and that March 12, 2004 was the date of the most recent change to the software or inputs to the reconstruction (e.g., calibration data) that might affect the results. At each processing step we record these tags. Similar tags identify later processing steps, and these tags are accumulated at each processing step, along with enough additional information to fully specify the sequence of processing steps and data inputs.

We realized that we needed to store provenance data in much more detail. Sometimes the inconsistent use of data and software is too subtle for detection at the level of data version tags. We need a reliable audit trail of an analysis to ensure consistent and correct use of software and data versions before publishing physics results. We also need to find data products that may have used a particular data object that was later found to be defective. While version identifiers conceptually apply to individual ASUs, our current implementation requires that all the ASUs in a data file for a given run be derived by the same sequence of analysis steps. For full functionality we needed to store provenance down to the ASU level, and track exact inputs and all software parameters. However, the effort to retrofit this functionality would require changes to the core code of our analysis system and so we opted for getting most of the required functionality with just a slight modification at the data format level.

As a compromise we collect, as strings, all the software module names, their parameters, plus all the input file information and make an MD5 hash of the strings. The version strings and hash are stored in the output stream of each file written, using a simple extension to the CLEO data storage system, so that every derived data file carries a summary of its provenance. We can detect the majority of usage discrepancies by comparing the hashes. In the event of a discrepancy, the physicists can view the strings to see what has changed. Clearly, this does not provide the full semantics of tracking at the ASU level. Some inputs may not have been used in the production of some parts of the data. It only tells which ASUs might have been used in the production of this ASU. But it provides the physicist with sufficient information to make consistent use of the data and software.

Substantial changes to the EventStore and analysis code will be required to accurately detect exactly which ASUs were used in the production of an output ASU. The metadata volume to track at the ASU level will be large, and it will be inappropriate to store it in the headers of the data files. It will be stored in a metadata DB and references to it placed in the data file. We do not expect to be able to retrofit such a system to our running systems because most of the data are stored in an HSM system (i.e., they are on tape), and we cannot easily update them. Furthermore, we cannot risk substantial changes to our large software base at this stage of the CLEO experiment. We believe this will work will be relevant for our involvement in the design of the software framework for the LHC CMS experiment, which is designed to use fine-grained provenance for data selection.

Other challenges remain. where we could use grid utitlities. We generate much of the CLEO simulated data offsite. We are currently implementing a system where the simulated data are stored in a ”personal EventStore” as they are produced, shipped to Cornell on USB disks, and merged into the collaboration EventStore. This process could be automated to a much greater extent if we could use grid data movement utilities and web services interfaces to EventStore. We would also like to make a fully web-based CLEO analysis environment, for the first passes through the data for physics analyses and for outreach purposes, and the natural way to do that now would be via web services interfaces and grid tools.

4 The WebLab Project

Since 1996, the Internet Archive has been collecting a full crawl of the Web every two months [3]. The total volume of data collected up to August 2005 is 544 Terabytes, heavily compressed, or about 5 Petabytes uncompressed. In summer 2005, we began work on transferring a major subset of this data to Cornell and organizing it for researchers, with a particular emphasis on supporting social science research.

User studies with social science researchers have identified a number of patterns in the research that they would like to do on the Web. A common theme is that researchers wish to extract a portion of the Web to analyze in depth, not the entire Web. Almost invariably they wish to have several time slices, so that they can study how things change over time. The criteria by which a portion of the Web is chosen for analysis are extremely varied. Some use conventional metadata, e.g., specific domains, file type, or date ranges. Others are empirical. For example, one researcher has combined focused Web crawling with statistical methods of information retrieval to select materials automatically for an educational digital library. Others plan to extend research on burst detection, which can be used to identifying emerging topics, to highlight portions of the Web that are undergoing rapid change at any point in time, and to provide a means of structuring the content of emerging media like Weblogs.

Of the specific tools that researchers want, full text indexes are highly important, but need not cover the entire Web. The link structure is of great interest because of its relationship to social networking. There are ambitious plans by computer scientists and social scientists working together to use methods of natural language processing to analyze the content of Web pages e.g., by extracting types of opinions.

Many social science research groups are reasonably strong technically, but they do not wish to program high-performance, parallel computers. The expectation is that most researchers will download sets of partially analyzed data to their own computers for further analysis. Therefore, access to the WebLab is provided via a Web Services interface to a dedicated Web server. The underlying API provides a generalization of the capabilities of the Google API. General services provided include a ”Retrobrowser” to browse the Web as it was at a certain date, a facility to extract subsets of the collection and store them as database views, and tools for common analyses of subsets, such as extraction of the Web graph and calculations of graph statistics.

The conventional architecture for providing heavily used services on the Web distributes the data and processing across a very large number of small, commodity computers. Examples include the Web search services, such as Google and Yahoo, and the Internet Archive’s Wayback Machine. While highly successful for production services, large clusters of commodity computers are inconvenient for researchers who carry out Web-scale research, either on the Web itself or on the social phenomena that the Web provides a record of. For instance, it would be extremely difficult to extract the types of subset required by social science researchers from the Internet Archive. Researchers studying the Web graph typically study the links among billions of pages. It is much easier to study the graph if it is loaded into the memory of a single large computer than distributed across many smaller ones.

4.1 Architecture

Access to the WebLab is provided via a Web Services interface to a dedicated Web server. General services provided include a ”Retrobrowser” to browse the Web as it was at a certain date, a facility to extract subsets of the collection and store them as database views, and tools for common analyses of subsets, such as extraction of the Web graph and calculations of graph statistics.

The conventional architecture for providing heavily used services on the Web distributes the data and processing across a very large number of small, commodity computers. Examples include the Web search services, such as Google and Yahoo, and the Internet Archive's Wayback Machine. While highly successful for production services, large clusters of commodity computers are inconvenient for researchers who carry out Web-scale research, either on the Web itself or on the social phenomena that the Web provides a record of. For instance, it would be extremely difficult to extract a stratified sample of Web pages from the Internet Archive. Researchers studying the Web graph typically study the links among billions of pages. It is much easier to study the graph if it is loaded into the memory of a single large computer than distributed across many smaller ones.

For these purposes, the decision was made to separate link information and metadata about pages from their content, and store the meta-information in a relational database on a single high-performance computer. The current machine is a 16-processor Unisys Server ES7000/430 with 64 GB of shared memory. By the end of 2007 it will have 240 TB of RAID disk storage.

Transferring the data from the Internet Archive to Cornell and loading it online places heavy demands on three parts of the system: the network connection, preprocessing the raw data, and the database load process. In addition, archiving the raw data, logging, backup, and restore operations can easily become a burden.

Our recent studies established that a good balance between the various parts of the system is achieved by setting an initial target of downloading one complete crawl of the Web for each year since 1996 at an average speed of 250 GB/day. For this, the network connection uses a dedicated 100 Mb/sec connection from the Internet Archive to Internet2 [4], which can easily be upgraded to 500 Mb/sec. The Cornell connection will move to the TeraGrid early in 2006. First versions of the two processing components were developed during fall 2005. Each has been tested at sustained rates of approximately 1 TB per day, when given sole use of the system. Experiments will be carried out during winter 2006 to determine the best mix of jobs to run in production.

4.2 WebLab Data Flow Summary

4.2.1 Preload subsystem

The Internet Archive stores Web pages in the ARC file format. The pages are stored in the order received from the Web crawler and the entire file is compressed with gzip. Each compressed ARC file is about 100 MB big. Corresponding to an ARC file there is a metadata file in the DAT file format, also compressed with gzip. It contains metadata for each page, such as URL, IP address, date and time crawled, and links from the page. The DAT files vary in length, but average about 15 MB.

The preload subsystem takes the incoming ARC and DAT files, uncompresses them, parses them to extract relevant information, and generates two types of output files: metadata for loading into the relational database and the actual content of the Web pages to be stored separately. The design of the subsystem does not require the corresponding ARC and DAT files to be processed together.

As of December 2005, a basic workflow for this design has been implemented and tested. First indications are that the performance will comfortably meet the required performance goals. Extensive benchmarking is required to tune many parameters, such as batch size, file size, degree of parallelism, and the index management.

5 Next Steps

Despite the application specific data flow tuning in each of the three projects, there are clear commonalities. All three projects are moving towards using relational databases for data management. The second trend are steps towards the usage of Web Services. For example, Arecibo is on the verge of joining the NVO, federating their data with other data resources from the Astronomy community.

The CLEO experiment will end within two years, but the Wilson Lab will participate in the next round of high-energy particle physics experiment, the Large Hadron Collider. This participation involves even more collaborators and much larger datasets. The goal is to take advantage of Grid technology like the tools developed by the Open Science Grid.

The WebLab is already in the process of connecting to TeraGrid for accessing data from the Internet Archive. Currently, Internet2 is used as a pipe for bulk transfer of data to the WebLab. However, the very high performance of the TerraGrid will allow another level of distributed research. A social science researcher will be able to analyze data, some of which is stored at Cornell, some in San Francisco at the Internet Archive, and some on a local computer. When extracting subsets for detailed research a social scientist will be able to combine relational queries at Cornell with text searches using the full text indexes being built by the Internet Archive.

All project would benefit from usage of the Grid to efficiently disseminate data and to mirror data to increase fault tolerance and availability.

6 Acknowledgments

This research was supported by Research Infrastructure Grant Number CNS-0403340 from the National Science Foundation (NSF), by NSF Grants AST-0206035, AST-0507747, IIS-0084762, IIS-0121175, DUE-0127308, and

SES-0537606, and by an E-Science grant and a gift from Microsoft Corporation. Any opinions, finding, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

We thank Nicholas Gerner, Wei Guo, Chris Sosa, and Samuel Benzaquen Stern for help with the implementation of tools for the WebLab. We thank Jim Gray for his support, and we thank Alex Szalay for permissions and support for using his software.

References

[1] The cleo collaboration. http://www.lepp.cornell.edu/public/CLEO.

[2] Cornell electron storage ring. http://www.lepp.cornell.edu/public/CESR.

[3] Internet archive. http://www.archive.org/.

[4] Internet2. http://www.internet2.edu/.

[5] Microsoft sql server. http://www.microsoft.com/sql.

[6] Mysql. http://www.mysql.com/.

[7] The particle adventure – the fundamentals of matter and force. http://ParticleAdventure.org/.

[8] R. Briere et al. CLEO-c and CESR-c: A new frontier of weak and strong interactions. http://www.lns.cornell.edu/public/CLEO/spoke/CLEOc.

[9] D. R. C.D. Jones, V. Kuznetsov and G. Sharp. Eventstore: Managing event versioning and data partitioning using legacy data formats. In *International Conference on Computing in High-Energy Physics and Nuclear Physics (CHEP 2004), Interlaken, CH*, September 2004.

[10] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM International Conference on Data Mining*, 2004.

[11] http://www.sqlite.org/.