

Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets

1 Introduction

The term “the Grid” denotes a distributed computing infrastructure for advanced science and engineering. Grid computing has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and high-performance orientation. [1]

The astronomy community has an abundance of imaging data (i.e. SDSS [2], GSC-II [3], 2MASS [4], POSS-II [5], etc) at its disposal which are essentially the “crown jewels”; however the terabytes of data makes the analysis of these datasets a very difficult process traditionally. Large astronomy datasets are generally very large (terabytes +) and contain many objects (100 million +) separated into many files (100,000+).

We propose to use grid computing as the main mechanism to enable the dynamic analysis of large astronomy datasets. There are five reasons why analyzing these large datasets is not trivial: 1. *large size of the datasets* (TB+ in size, 100M+ objects); 2. *large number of users* (1,000s); 3. *large amount of resources* needed to have adequate performance (potentially 1,000s of processors and 100s TB of rotating storage); 4. *dispersed geographic distribution of the users and resources*; and 5. *heterogeneity of the resources*.

The key question we will answer by the successful implementation of this proposal is: “*How can the analysis of large astronomy datasets be made a reality for the astronomy community using Grid resources?*” Our answer is the “AstroPortal”, a science gateway to grid resources that is specifically tailored for the astronomy community.

Some of the interesting and innovative research work that will be the building blocks of the AstroPortal will be: 1. *resource provisioning* (advanced resource reservations, resource allocation, resource de-allocation, and resource migration); 2. *data management* (data location, data replication, and data caching); and 3. *distributed resource management* (coupling data and processing resources together efficiently, and distributed resource management for scalability).

The AstroPortal will be a real implemented system that will give the astronomy community a new tool to advance their research and to open new doors to opportunities never before possible. At the same time, the building blocks of the AstroPortal should uncover new approaches to resource and data management that are specifically tailored for the efficient and successful dynamic analysis of large scientific datasets.

2 Related Work & Background Information

This section’s purpose is to describe related work similar to the AstroPortal and to cover background material (existing astronomy datasets, the TeraGrid testbed, and science gateways) necessary to make this proposal as self contained as possible.

2.1 Astronomy Datasets

Astronomy faces a data avalanche, with breakthroughs in telescope, detector, and computer technology allowing astronomical surveys to produce terabytes (TB) of images. There are several well known large astronomy datasets that could potentially be used by the AstroPortal. Some of these include the Sloan Digital Sky Survey (SDSS) [2], the Guide Star Catalog II (GSC-II) [3], the Two Micron All Sky Survey (2MASS) [4], and the Palomar Observatory Sky Survey (POSS-II) [5]. There are other astronomy datasets, but this small sample shows that astronomy datasets are generally very large (TB+) and contain many objects (200M +). Specifically, SDSS has 200M objects in 6TB of data; GSC-II has 1000M objects with 8TB of data; 2MASS has 500M objects in 10TB of data; and POSS-II has 1000M objects in 3TB of data.

2.2 Related Work: Analysis of Large Astronomy Datasets

Although there might be more related projects under development, two relatively large efforts with very similar goals to our own proposed system are the NSF National Virtual Observatory (NVO) [6] and Montage [7] which is tightly coupled with the NVO project. NVO is a multiyear effort to build tools, services, registries, protocols, and standards that can extract the full knowledge content of these massive, multi-frequency data sets. The goal is to use the computational resources of the TeraGrid [8] combined with the

NVO to enable astronomers to explore and analyze large datasets. Montage [7] is a project that will deploy a portable, compute-intensive, custom astronomical image mosaic service for the NVO. The authors make use of grid computing as their key element of the Montage portal architecture. We differentiate our efforts from those in the NVO and Montage mostly by offering optimizations at different levels, from resource provisioning, data management, to distributed resource management.

2.3 TeraGrid & Science Gateways

The TeraGrid (TG) [8] is an open scientific discovery infrastructure combining leadership class resources at eight partner sites to create an integrated, persistent computational resource. The deployment of TeraGrid brings over 40 teraflops of computing power and nearly 2 petabytes of rotating storage, interconnected at 10-30 gigabits/second via a dedicated national network. The initial prototype will be deployed at The University of Chicago / Argonne National Laboratory (UC/ANL) site in the TeraGrid, with future implementation iterations (the distributed version) being deployed over the entire TeraGrid's 8 sites distributed throughout the US.

The TeraGrid Science Gateways [9] initiative is also another effort that is encouraging that this proposal is on the cutting edge of research initiatives in the grid and astronomy communities. Science Gateways signal a paradigm shift from traditional high performance computing use. Gateways enable entire communities of users associated with a common scientific goal to use national resources through a common interface.

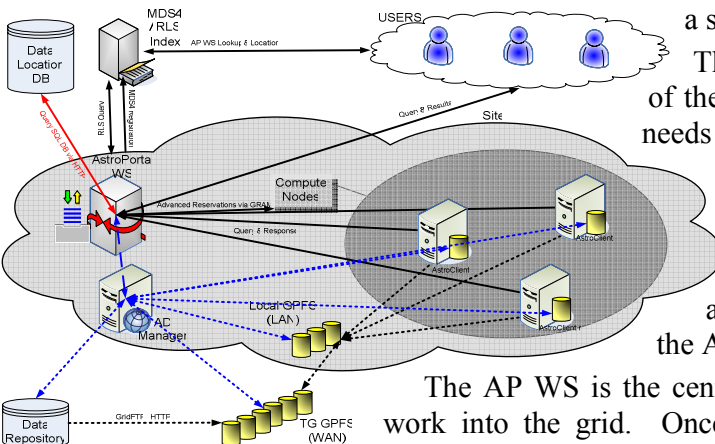
3 AstroPortal Implementation & Evaluation

The AstroPortal implementation will be based on various components of the Globus Toolkit version 4 (GT4) [10], and it will be deployed in the TeraGrid [8]. Some of the GT4 components are: WS GRAM [10], MDS4 [11], RFT [10], GridFTP [12], RLS [10], DRS [10], and WS [10]. The implementation will be done in two versions focusing on different objectives: 1) AstroPortal functionality as a science gateway, along with the needed resource management support for astronomy analysis code to be efficiently run on large datasets; 2) it will focus on a distributed resource management design that will enhance scalability and performance of the AstroPortal.

The rest of Section 3 will assume the use of the SDSS DR4 [13] dataset as it will be the first supported dataset in our prototype to be deployed on the ANL/UC TeraGrid testbed.

3.1 AstroPortal Architecture

Figure 1 shows the basic architecture of the centralized version of the AstroPortal (AP). There are several components that make the building blocks of the AP: 1) the AstroPortal Web Service (AP WS), 2) the AstroData Manager (AD), 3) the Astro Clients (AC) running on the compute nodes, and 4) the User Clients (UC). The communication between all these components would be using Web Services (WS). Furthermore, we will leverage GT4 functionality which offers persistent state storage for WS; the persistent state will make the AP WS more robust to failures as it will offer the alternative to continue execution of unfinished jobs after a system restart.



The AP WS and the AD are the two main components of the system where the resource management innovation needs to occur; furthermore, both of these components are rather generic, and with minor tuning, could be used in the analysis of other large non-astronomy related datasets. The AC and UC are specific to the astronomy community, and will offer the analysis and visualization functionality needed make the AP system useful to astronomers.

The AP WS is the centralized gateway for all UC to submit their analysis work into the grid. Once the AP WS is up through a basic bootstrapping mechanism, the AP WS would register itself with a well known MDS4 Index, so UC could dynamically find the location of the AP WS. The

Figure 1: AstroPortal architecture

UC could use many existing tools offered by the SDSS / SkyServer [14] to find the location (the sky coordinates – {ra dec band}) of the interesting objects in question. The UC then packages the list of locations along with the analysis to be performed, and it is sent to the AP WS for processing as a job. Initially, the AP WS would make some advanced reservations (via GRAM) of some predefined set of resources for a predefined duration. New resources could be reserved dynamically to increase the performance of the AP under heavy loads, and resources could be de-allocated to a minimum under light loads. Upon the AP WS receiving the work from the UC, it places the list of locations in a user queue and spawns multiple threads to find (via RLS) the necessary data within the storage hierarchy. ACs use this data to perform the appropriate operation, and sends the results back to the AP WS. When the AP WS received the results from an entire job, it packages or aggregates them depending on the particular analysis performed, and sends the results back to the UC. For relatively large results, only the location of the results will be returned via WS, and the actual results will be retrievable via GridFTP for better efficiency.

The storage hierarchy is one of the key design choices that differentiates the AstroPortal from other related work. The storage hierarchy consists of 4 layers: remote data repository (REMOTE), TeraGrid GPFS (WAN), ANL/UC GPFS (LAN), and local storage (LOCAL). Each storage layer gets the data closer and closer to the computational resources making the analysis run faster. The AP WS could use RLS to maintain a coherent state between the replica location among the different layers (LOCAL, LAN, WAN, REMOTE). Ideally, as the data gradually flows in (from REMOTE, to WAN, to LAN, to LOCAL) as AC access the data, jobs would run faster over time. This would be true if the set of resources was static, however we are targeting a dynamic system which has a variable pool of resources. The REMOTE layer will offer persistent storage, the WAN GPFS and local GPFS (LAN) should offer relatively persistent storage, but the LOCAL disk storage will be fairly dynamic, as worker resources will start-up and terminate regularly.

In order for the AP to reach that best case scenario performance, there would be a need for a worker resource to efficiently transfer its state (work queues and locally cached data) from one resource (i.e. node) to another. As the system is used, it is possible that this transferring of state take longer due to a growing local cache of data. This is high cost of transferring state is OK as long as it does not occur too often, but that means that the system will not be very dynamic and will not be able to respond to "hot-spots" of large number of queries for a short period of time without wasting resources. We believe some innovative resource migration mechanisms could help keep the LOCAL layer available for longer periods, and hence improving the likelihood that data is read from the fastest layer (LOCAL) during the analysis.

We envision that a natural evolution to the AP will be distribute the resource management decisions of the AP WS, offering a more scalable architecture! The majority of the intra-site communication remains unchanged, with the exception that the MDS4 Index need not be specifically associated with any particular site. Each AP WS from each site would register with the MDS4 Index; when users query the MDS4 Index, users could pick a possible AP WS at random, or based on some simple metrics (i.e. AP WS load, latency from the AP WS to the user, etc) that MDS4 exposes to the users about each AP WS. The key to the enhanced performance is the ability to harness all the resources across various sites in the TG; the interaction between the various AP WS from each of the various sites is critical. Each AP WS would get some work from UC, and it would have a choice of completing it locally or forwarding the work to another site that might offer faster performance due to data locality, more available resources, faster hardware, etc.

3.2 Large Dataset Analysis Support

There are many different analysis/operations that the astronomy community can apply to astronomy datasets. One simple operation would be the GET operation, in which the input would be a list of locations that need to be retrieved, and the output would be the images corresponding to the input locations. The GET operation could be used if the user wanted to run some custom analysis not offered by the AP on a subset of the original dataset. Another operation could be MONTAGE, in which the input would be the coordinates to a rectangular area (4 set of coordinates), and the output would be an image that represented the entire rectangular area stitched together from smaller images. The MONTAGE operation could be useful for the visualization of the sky at different levels of detail. We will focus on the STACKING operation, in which the input would consist of a list of locations, and the output would be a single image corresponding to the stacked images. Stacking

could be used to enhance faint objects that would otherwise have not been detected. In our initial prototype, we plan on supporting the GET operation and the STACKING operation. To the best of our knowledge, there is no system out there offering a STACKING like service for astronomy datasets. We do not plan to implement the MONTAGE operation since there currently exists a system (Montage [7]) that will be deployed on the TeraGrid as part of the NVO project.

3.3 Evaluation Methodology

We intend to thoroughly test the AstroPortal performance, scalability and robustness. Our initial evaluation will be conducted via DiPerF [15, 16, 17], a DIstributed PERformance testing Framework, aimed at simplifying and automating service performance evaluation.

The AP will be first deployed at the ANL/UC site in the TeraGrid, while the distributed AP will be deployed in the entire TG across eight different sites geographically distributed across the US. Our experiments will involve both very controlled experiments on dedicated resources within the TG, and more realistic scenario experiments with the UC running in another testbed, PlanetLab [18]. PlanetLab will offer real Internet conditions as the 500+ nodes are geographically distributed all over the world with relatively poor connectivity in comparison to the TG testbed.

4 Open Research Questions

We believe that there are at least three main areas with open research problems that the architecture design of the AstroPortal exposes. These areas are all in the broad context of resource management; they include: *resource provisioning, data management, and distributed resource management.*

Resource provisioning includes everything from advanced reservations, to resource allocation, to resource de-allocation issues in large scale systems. Different techniques and heuristics will apply for managing efficiently the set of resources depending on the problem we are addressing; some of the important things will be workload characteristics, number of users in total and number of concurrent users, data set size and distribution, computational intensive analysis, and I/O intensive analysis. The resource provisioning will be very important in order to achieve efficient use of existing resources, yet maintain a responsive and good performance system.

Data management: Data location, data caching, and data replication: Since one of our first two operations the AstroPortal will support is STACKING, we will focus on our motivation for the storage hierarchy described in Section 3.1, and the data management optimizations we hope to investigate. In a preliminary performance evaluation of the various data access methods, we found that there is a wide range of performance differences between the various different access methods. For example to complete 100K crops (needed for either the GET or the STACKING) on 100 nodes, the best case scenario is getting the data from the LOCAL layer which takes between at most 30 seconds. The next best performance is delivered when getting the data from the LAN layer, taking at most 200 seconds. The worst performance was the WAN layer, with times as high as 3000 seconds. Notice the difference in performance among the different layers in the storage hierarchy, which could open opportunities for good data access optimizations. There are some very interesting problems around **data management**, in which we have a very large data set that we want to break up among various sites, but also do some level of replication among the sites for improved performance. Furthermore, doing data movement based on past workloads and access patterns might prove to offer significant performance gains. We envision that the AstroData Manager will keep track of usage statistics on each object from the dataset, which will later be used to keep the most likely items in the fastest and smallest layer, optimizing the time to access the more popular data. Another significant challenge will be resource migration, in which our goal is to perform efficient state transfer among worker resources while maintaining a dynamic system.

Distributed Resource Management: The inter-site communication among the AstroPortal Web Service and its effects on the overall system performance is very interesting; work can be performed at the local site, or it could be delegated to another site that in theory could complete the work faster; the algorithms, the amount of state information, and the frequency of state information exchanges all contribute to how well and evenly the workload is spread across the various AP WS, which ultimately decides the response time that the user

observes, and the aggregate throughput the entire distributed AstroPortal system can sustain. The successful implementation the distributed AP WS and the optimization of the use of both the data storage and the computational resources could lead to a scalable system supporting large numbers of concurrent users while providing very fast response times in comparison to traditional single server implementations. The use of the GT4 throughout the architecture will allow the system to interoperate with other system easily, and provide a standard method of accessing the system.

5 Conclusion

The key question we will answer by the successful implementation of this proposal is: “How can the analysis of large astronomy datasets be made a reality for the astronomy community using Grid resources?” Our answer is the “AstroPortal”, a science gateway to grid resources that is specifically tailored for the astronomy community. The AstroPortal will be a real implemented system that will give the astronomy community a new tool to advance their research and to open new doors to opportunities never before possible.

We believe that we are in a good position to tackle the proposed project because of our previous work in resource management [19, 20]; we also believe we have the means of evaluating the proposed system via more previous work in distributed performance measurements [15, 16, 17]. We are also closely working Dr. Alex Szalay, an Alumni Centennial Professor in the Department of Physics and Astronomy at Johns Hopkins University. Dr. Szalay is a pioneer in large astronomy datasets, especially within the SDSS [2] and the NVO [6] efforts in the TeraGrid.

We differentiate our efforts from those in the NVO [6] and Montage [7] mostly by offering optimizations at different levels, from resource provisioning, data management, to distributed resource management. Our goal is to have that our implementation offer better performance from three different perspectives: 1) offer a more responsive system while still maintaining good resource utilization; 2) offer faster data access by managing the data pro-actively based on past workloads in a 4-layer storage hierarchy; and 3) provide a more scalable system due to the distributed nature of the architecture. We hope that the innovative work from the AstroPortal project can eventually be integrated into projects such as the NVO so the astronomy community at large can benefit from the same improvements in performance that the AstroPortal found.

6 References

- [1] I Foster, C Kesselman, S Tuecke, “*The Anatomy of the Grid*”, International Supercomputing Applications, 2001.
- [2] SDSS: Sloan Digital Sky Survey, <http://www.sdss.org/>
- [3] GSC-II: Guide Star Catalog II, <http://www-gsss.stsci.edu/gsc/GSHome.htm>
- [4] 2MASS: Two Micron All Sky Survey, <http://irsa.ipac.caltech.edu/Missions/2mass.html>
- [5] POSS-II: Palomar Observatory Sky Survey, <http://taltos.pha.jhu.edu/~rrg/science/dposs/dposs.html>
- [6] R Williams, A Connolly, J Gardner. “*The NSF National Virtual Observatory, TeraGrid Utilization Proposal to NRAC*”, <http://www.usvo.org/pubs/files/teragrid-nvo-final.pdf>
- [7] JC Jacob, DS Katz, T Prince, GB Berriman, JC Good, AC Laity, E Deelman, G Singh, MH Su. “*The Montage Architecture for Grid-Enabled Science Processing of Large, Distributed Datasets*”, Earth Science Technology Conference, 2004.
- [8] TeraGrid, <http://www.teragrid.org/>
- [9] TeraGrid: Science Gateways, http://www.teragrid.org/programs/sci_gateways/
- [10] I Foster. “*A Globus Toolkit Primer*,” 02/24/2005. http://www-unix.globus.org/toolkit/docs/development/3.9.5/key/GT4_Primer_0.6.pdf
- [11] JM Schopf, I Raicu, L Pearlman, N Miller, C Kesselman, I Foster, M D’Arcy. “*Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4*”, under review at IEEE HPDC 2006.
- [12] B Allcock, J Bresnahan, R Kettimuthu, M Link, C Dumitrescu, I Raicu, I Foster. “*The Globus Striped GridFTP Framework and Server*”, IEEE/ACM SC 2005.
- [13] SDSS Data Release 4 (DR4), <http://www.sdss.org/dr4/>
- [14] Sloan Digital Sky Survey / SkyServer, <http://cas.sdss.org/astro/en/>
- [15] C Dumitrescu, I Raicu, M Ripeanu, I Foster. “*DiPerF: an automated Distributed Performance testing Framework*”, IEEE/ACM GRID2004, Pittsburgh, PA, November 2004, pp 289 - 296
- [16] I Raicu. “*A Performance Study of the Globus Toolkit® and Grid Services via DiPerF, an automated Distributed Performance testing Framework*”, University of Chicago, Computer Science Department, MS Thesis, May 2005, Chicago, Illinois.
- [17] I Raicu, C Dumitrescu, M Ripeanu, I Foster. “*The Design, Performance, and Use of DiPerF: An automated Distributed Performance testing Framework*”, under review at Journal of Grid Computing.
- [18] B Chun, D Culler, T Roscoe, A Bavier, L Peterson, M Wawrzoniak, and M Bowman, “*PlanetLab: An Overlay Testbed for Broad-Coverage Services*,” ACM Computer Communications Review, vol. 33, no. 3, July 2003.
- [19] C Dumitrescu, I Raicu, I Foster. “*DI-GRUBER: A Distributed Approach for Grid Resource Brokering*”, IEEE/ACM Super Computing 2005
- [20] C Dumitrescu, I Raicu, I Foster. “*Extending a Distributed Usage SLA Resource Broker to Support Dynamic Grid Environments*”, under review at EuroPar 2006.