

Grid Middleware Services for Virtual Data Discovery, Composition, and Integration

Yong Zhao¹, Michael Wilde², Ian Foster^{1,2}, Jens Voeckler¹,
Thomas Jordan³, Elizabeth Quigg³, James Dobson⁴

ABSTRACT

We describe the services, architecture and application of the GriPhyN Virtual Data System, a suite of components and services that allow users to describe virtual data products in declarative terms, discover definitions and assemble workflows based on those definitions, and execute the resulting workflows on Grid resources. We show how these middleware-level services have been applied by specific communities to manage scientific data and workflows. In particular, we highlight and introduce *Chiron*, a portal facility that enables the interactive use of the virtual data system. *Chiron* has been used within the QuarkNet education project and as an online “educator” for virtual data applications. We also present applications from functional MRI-based neuroscience research.

Categories and Subject Descriptors: D.2.6 [Programming Environments]: Integrated Environments. **General Terms:** Management, Design. **Keywords:** Virtual Data, Data Grid, Provenance, Data Discovery, Data Integration, Workflow, Portal

1. INTRODUCTION

The virtual data concept is motivated by next generation data-intensive applications in which scientists distributed worldwide need to extract scientific information from large collections of data, and would like to share both data products and the resources needed to produce and store them [1,2]. Virtual data mechanisms enable the declarative specification of the recipes used to derive data, so that requests for data products can be transparently mapped into computation and/or data access across multiple Grid computing and storage locations. The information recorded to support this reuse of computational results can also be used for other purposes, such as explaining provenance and managing workflows [3].

The GriPhyN virtual data system [4] implements such virtual data mechanisms. Definition and query operations are expressed in a virtual data language (VDL), and information about data and computation procedures is stored in a virtual data catalog (VDC).

The virtual data lifecycle (Figure 1) begins when a virtual data definition, described in VDL, is entered in the VDC. This definition can then be discovered by virtual data query operations, and the scientific analysis procedures associated with it can be executed and validated. New algorithms and procedures can be derived from the knowledge gathered, and another round of the lifecycle starts when the derived virtual data is published into the VDC or recorded by the virtual data system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2nd Workshop on Middleware for Grid Computing Toronto, Canada
Copyright 2004 ACM 1-58113-950-0 ...\$5.00.

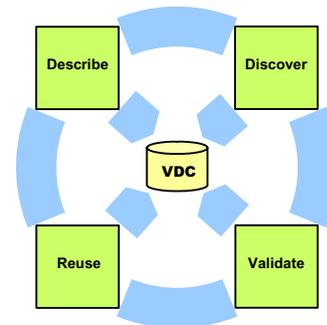


Figure 1: Lifecycle of Virtual Data

The Virtual Data System has been applied to a variety of data-intensive applications, including physics event simulation [5], galaxy cluster finding [6], genome analysis [7], and biomedical image analysis [8]. However, the initial virtual data interfaces (a Java API with command-line wrappers), while powerful for scripting-level integration, proved cumbersome for both deeper application integration and direct end-user use. Many user communities require the ability to integrate virtual data capabilities more deeply into science applications, and to encapsulate the tasks of setting up and configuring the virtual data system and its associated grid compute and storage resources. End users need an interactive environment in which they can easily discover and share virtual data products, compose workflows, and monitor workflow executions across multiple grid sites.

These considerations have led us to develop, apply, and evaluate a set of interfaces that provide greater access to virtual data system data structures. On top of these, we developed the *Chiron* portal for convenient interaction with the virtual data system. *Chiron* allows users to:

- manage user accounts and track user activities;
- describe and publish applications and datasets;
- discover, validate, share and reuse applications and datasets published by other users;
- configure and discover Grid resources;
- compose workflows and dataset derivations;
- execute workflow either locally or in a Grid environment and monitor job progress; and
- record provenance information.

With these enhancements, the virtual data system now provides both user-level and service (middleware)-level functionality that portal users can integrate to build customized virtual data interfaces for specific user communities and science applications.

¹Department of Computer Science, University of Chicago

²Math and Computer Science Division, Argonne National Laboratory

³Fermi National Accelerator Laboratory

⁴Department of Psychological & Brain Sciences, Dartmouth College

The rest of this paper is organized as follows: in Section 2, we describe Chiron system architecture; in Section 3, we discuss virtual data discovery; in Section 4, we present virtual data integration examples from the QuarkNet project and functional MRI experiments; in Section 5, we compare with related work; and we conclude in Section 6.

2. CHIRON SYSTEM ARCHITECTURE

Chiron is based on the GriPhyN virtual data system and implemented in terms of virtual data system APIs and libraries, standard commodity Web technologies, databases, and Grid technologies. We show the system architecture in Figure 2.

A Chiron client only needs a web browser that supports SSL for security, DHTML and Java scripts for interactivity, and cookies for customization. The Chiron server comprises an Apache Tomcat web server, Java Server Pages (JSP), the virtual data system API and libraries (jar files), backend databases for the virtual data system, GraphViz for graph visualization, workflow engines for local and Grid execution, and Condor-G [9] for Grid job submission and execution.

2.1 System Configuration

The Chiron portal code and libraries are deployed into a Tomcat servlet engine. Chiron extends the virtual data system “properties” file to configure various runtime parameters, such as database driver and schema, MDS (monitoring and discovery service), and RLS (replica location service) host information, file transfer mode, logging options, and scheduler-specific configuration. The backend database leverages Java reflection and dynamic class loading to support several database management systems.

2.2 User Management

All portal users must create a user account and should have a valid Grid certificate. We use user-ID and password authentication via secure HTTP connections, so that the client side only needs a web browser to access the portal. Different users are assigned different roles at the portal, such as guest, administrator, and super user, and different levels of controls are associated with each different role. The portal also keeps track of sessions, user preferences, and user activities. Thus, we may be able to find analysis patterns by leveraging data mining technologies.

2.3 Job Submission

Once users have registered and published data derivation recipes in the VDC, they can request that these virtual datasets be materialized. We term the process of mapping these requests to the execution of computation procedures “planning” as it is suggestive of query planning in database systems. Following a request, the virtual data system finds all datasets and procedures required to derive the requested dataset, and generates an “abstract” derivation workflow: abstract in the sense that it is not tied to specific executables, dataset replicas, and grid sites. This workflow is represented in an XML format that can be interpreted and executed by different planners and workflow engines.

Chiron supports two different planners. The *local planner* takes a workflow definition and produces a set of shell scripts that invoke the required programs. This planner is designed to run derivations

on a single host and thus is useful for initial debugging of applications and for determining whether specific errors are caused by the Grid, virtual data system, or the application.

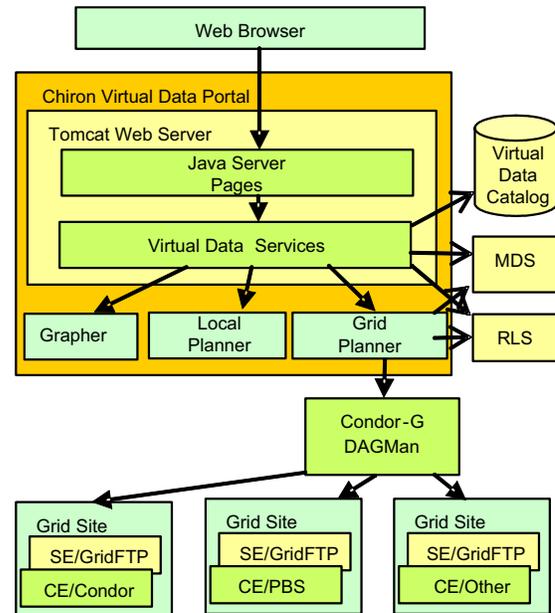


Figure 2 Virtual Data System Architecture

The *Pegasus planner* [10] uses information from the Grid information service (MDS) and replica location service (RLS) to convert the abstract plan into a concrete execution plan. That concrete plan is then submitted to DAGMan and executed on remote Grid sites via Condor-G and Globus. A Grid site usually has compute elements (CE) managed by a local scheduler (Condor, PBS, LSF, etc), and storage elements (SE).

The portal also displays execution progress and job status once the workflow is converted into a concrete plan and executed. Upon successful completion, requested datasets can be displayed as images or text, depending on their format.

2.4 Workflow Visualization

Chiron uses GraphViz, a graph drawing tool developed by AT&T Laboratories, to visualize workflows. Figure 3 shows thumbnail-level workflow graphs from four different domains. (More detailed views of workflows are presented later). Parallelograms and ellipses represent datasets and computation procedures, respectively. When displayed in Chiron, a user can click on a dataset node to show all derivations that process the dataset; clicking on a procedure node displays detailed information about the corresponding transformation. These graphs provide insights into both the workflow process in the large and individual steps.

3. VIRTUAL DATA DISCOVERY

The keys to virtual data reuse and sharing are efficient description and discovery. Chiron allows users to search for virtual data by definition, metadata, and provenance. Definitions organize virtual data in a structured hierarchy, metadata associates virtual data products with arbitrary annotations, and provenance records the processing lineage of virtual data products.

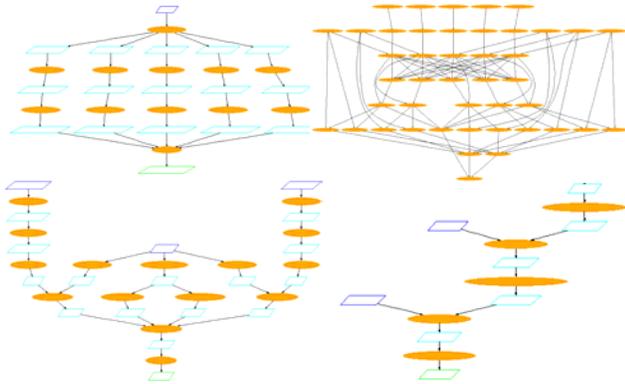


Figure 3: Workflow thumbnails for (clockwise): biology, astronomy, physics education, and geospatial experiments.

3.1 Definition

A definition is a declarative specification of some data analysis procedure. It can be either a transformation (TR), which is similar to a function definition in that it defines a function name and formal parameters, or a derivation (DV), which is like a function call with the actual parameters and datasets identified. [3,4]

Definitions can be organized hierarchically by namespaces, which might be associated with, for example, institutions, projects, groups, or users. Chiron displays definitions in a tree view that can be customized to show namespace, TR, and/or DV (Figure 4). A user can explore different namespaces and display the detailed definitions of each TR and DV by exploring the tree. Definitions can also be shown in VDL [3,4] for advanced users.

Chiron also provides a search bar for quick exploration. A user can search for TRs, DVs, datasets, and metadata using keyword and wildcard search. These definitions provide an interface-level understanding of data analysis procedures.

Attribute Name	Type	Value
startdate	date	2003-02-01 13:32:04.0
type	string	split
detectorID	string	057
school	string	IJMSA
project	string	cosmics
juliandate	string	2452672
startdate	date	2003-02-01 13:30:30.0
uploaddate	date	2004-02-18 00:00:00.0
blessed	boolean	true
author	string	TheEinsteinsGroup

Figure 4: Tree View and Metadata

3.2 Metadata

Metadata—information about data—is essential to discovery and understanding of datasets and procedures stored in the VDC. We allow five categories of annotation to virtual data objects: datasets, TR, DV, TR parameters, and TR calls in compound TRs. (A compound TR is one that calls other TRs.) Annotations are represented by (attribute_name, type, attribute_value) tuples, where type can be integer, float, string, date, or boolean.

Typical metadata about a dataset includes cataloging information, such as description, creator, creation date, and publisher; content

information such as data format, data structure, and size; and provenance information, i.e., how data is acquired, transformed, transferred. For code and procedures, besides information about their inputs and outputs, and execution environments, we find it useful to include functionality description, prerequisites, and constraints on parameters.

We define a metadata query language to search for virtual data objects that have specific metadata. A user does not need to specify the types of the attributes being searched: a query parser takes care of type conversion and parses it into an abstract syntax tree, which is then optimized and translated into SQL or XML XPath or XMLquery statements. The query statements are submitted directly to the backend database and the results are interpreted and then returned in the appropriate format.

The Chiron portal provides web interfaces for associating metadata with different virtual data objects, displaying metadata about a specific virtual data objects, and searching for virtual data objects with specific metadata.

3.3 Provenance

Provenance information captures the derivation history of data [11]. In our case, it refers to the invocation record for a data derivation, which records when, how, and where the data is produced, what kind of resources are used, and so on. Combined with the abstract workflow produced for the data derivation, provenance information gives the audit trail of data production process. A user can use this information to repeat a derivation, to explain a result, and so forth.

An invocation record is recorded for each task in a workflow that is executed. These records can be used to provide reports on data quality, run time, resource consumption, or even computation anomalies; to re-execute past workflow or to modify/improve future workflow; and to answer ad hoc queries. Figure 5 illustrates the potential to query the VDC across the dimensions of provenance, metadata, and transformation definitions.

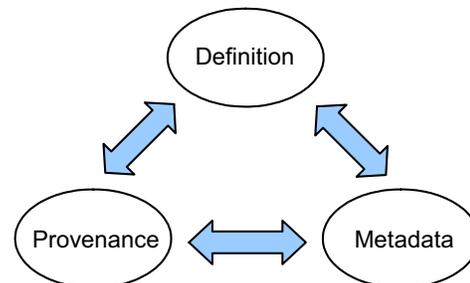


Figure 5: Discovery Space

4. VIRTUAL DATA INTEGRATION

Science communities and scientific applications must often integrate virtual data into an existing portal or Web interface framework. Chiron facilitates the development of application specific portals and Web interfaces, and provides reusable core services for accessing virtual data and Grid resources. We illustrate virtual data integration scenarios drawn from QuarkNet project and neuroscience research projects.

4.1 QuarkNet Cosmic Ray Application

The Chiron portal is being applied in the QuarkNet project [12], a NSF- and DOE-funded project that aims to create a research community of physicists, high school teacher and their students. QuarkNet engages high school teachers and students with scientific investigations about the structure of matter and the fundamental forces of nature. Students learn fundamental physics as they analyze live online data and participate in inquiry-oriented investigations, and teachers join research teams with physicists at local universities or laboratories. Currently, the project involves more than 6 large physics experiments, 52 research groups, 158 physicists, 514 high school teachers and thousands of students.

QuarkNet has created a specialized web site and web interfaces for science education. This portal, driven by virtual data middleware, allows students to launch, configure and control remote applications as though they are using a desktop computing environment. The key requirements for such web interfaces are to give students the means to a) discover and apply datasets, algorithms, and data analysis methods; b) collaborate by developing new transformations and by sharing results and observations; and c) learn data analysis methods that will prepare—and motivate—them for a scientific career.

These requirements match those for Chiron, but here it is the web applications that need to interact with Chiron. Thus, we have encapsulated relevant Chiron portal functions in into APIs so that these web applications can integrate virtual data mechanisms seamlessly into their framework.

As illustrated in Figure 6, students interact directly with the QuarkNet Web interface, which in turn accesses Chiron services to upload data and PERL code, process data using simple form-based tools, annotate and share data and results, and create posters to describe and share analysis processes. Students can use Chiron's metadata queries to discover each other's workflows and data products.

The QuarkNet Cosmic Ray Project is designed to give students hands-on experience in scientific collaboration and discovery. Students learn to construct, deploy and test cosmic ray detectors, and analyze real-time data gathered by the detectors. About 40 detectors have been installed in high school classrooms across the U.S. QuarkNet helps students understand and explore mysteries such as the origin of highly energetic cosmic rays, the source of their energy, and the clues they hold about the early universe.

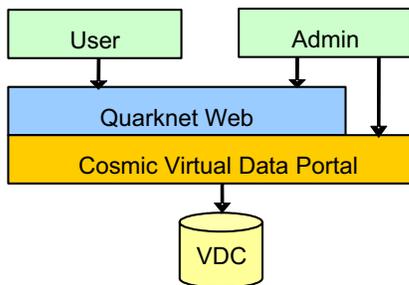


Figure 6: QuarkNet Virtual Data Integration

In Figure 7, we show the customized web page for the cosmic ray shower study. This study involves a data analysis process that combines detector and GPS data gathered from distributed geographical locations and determines whether a cosmic ray

shower has occurred in the sampled period and location. The plot is generated by clicking the ‘analyze’ button, which sends the request to the derivation script in Chiron. The script first generates the abstract workflow (Figure 8), then calls the shell planner to concretize the workflow, then executes the corresponding analysis code to generate the graph, and finally returns a link to the graph. These steps all occur transparently - the user only interacts with the QuarkNet web page.

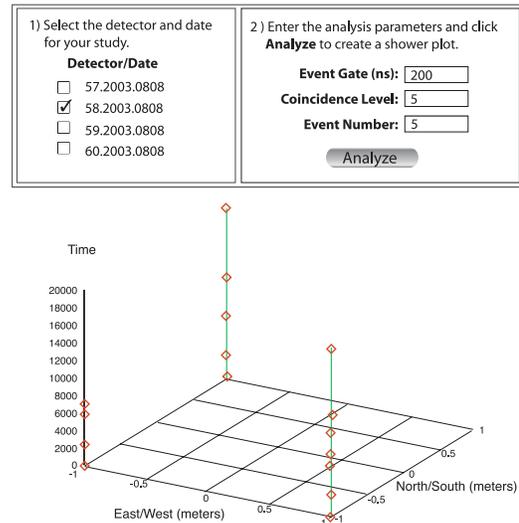


Figure 7: Cosmic Ray Shower Study

Chiron also provides mechanisms to generate application-specific web pages automatically, by using HTML templates, TR definitions, and annotations. For instance, the title of a TR can be rendered into the title of the web page, default/eligible values for TR parameters can be rendered as options, input datasets can be filtered by dataset types, and output filenames can be generated by context.

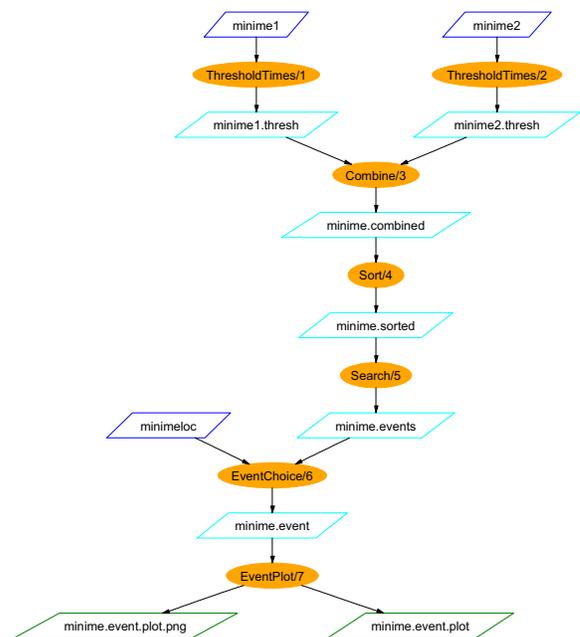


Figure 8: Cosmic ray shower study workflow

4.2 Cognitive Neuroscience Application

Automated data provenance and workflow management is a critical requirement for the functional MRI (fMRI) research community. The virtual data middleware has been applied to the process of creating population-based atlases from the fMRI Data Center's archive [18] of high resolution anatomical MR data, using a multi-stage I/O intensive pipeline that “warps” the anatomical features of the brain of multiple subjects into a standard space.

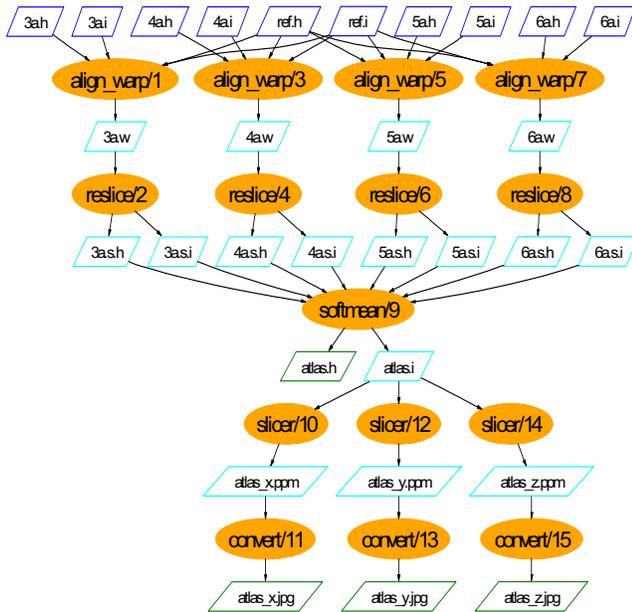


Figure 9: fMRI workflow

The VDL workflow for the creation of a brain atlas (Figure 9) involves several derivations that employ the AIR (automated image registration) suite [19,20,21] to create an “averaged” brain from a collection of high resolution anatomical data. The workflow warps each anatomical volume to a reference volume, using a 12th order nonlinear 1,365 parameter model. The brain structures were warped to define a similar shape, then each volume was resliced using the warp values, and the resulting high resolution volumes were averaged using AIR’s *softmean*.

In the context of this atlas-generation use case, we experimented with applying the middleware services of the virtual data system to various problems of discovery. We have preliminary experience with the following science-driven queries and processes. For each, we show an example query and the corresponding response.

Which TRs can process a subject image?

```
Q: xsearchvdc -q tr_meta dataType
      subject_image input
A: fMRIDC.AIR::align_warp
```

Which TRs can create an ATLAS?

```
Q: xsearchvdc -q tr_meta dataType
      atlas image output
A: fMRIDC.AIR::softmean
```

Is there a TR that can convert an Analyze-format image to a Nipbm .ppm file?

```
Q: xsearchvdc -q tr_meta dataType image
      input
A: fMRIDC.FSL::slicer

Q: xsearchvdc -q tr_meta dataType ppm
      output
A: fMRIDC.FSL::slicer
```

List anonymized subject-images for young subjects. This query searches for files based on their metadata.

```
Q: xsearchvdc -q lfn_meta
      dataType subject_image
      privacy anonymized
      subjectType young
A: 3472-4 anonymized.img
```

For a specific patient image, 3472-3, show all DVs and files that were derived from this image, directly or indirectly.

```
Q: xsearchvdc -q lfn_tree
      3472-3 anonymized.img
A: 3472-3_anonymized.img
      3472-3_anonymized.sliced.hdr
      atlas.hdr
      atlas.img
      ...
      atlas_z.jpg
      3472-3_anonymized.sliced.img
```

Similar queries can be used for data validation processes, and to construct scripts that can compose new virtual data workflows based on existing ones, and then catalog, run, and record them.

5. RELATED WORK

The GridAnt [13] system provides a flexible workflow model workflow based on Jakarta Ant and Grid execution. However, it does not address provenance tracking, planning support, or the declarative representation of application logic.

At a much larger scale, the SAM system deployed by the Collaboration for Multiscale Chemical Science [14] is based on the uniform action model of the WebDAV protocol [15], and has a well developed “laboratory notebook” model of provenance tracking. However, its middleware approach for Grid execution has the same location-dependence limitations as GridAnt.

The Grid Access Portal for Physics Applications[16] supports job submission, job status checking, and resource management. Each application is packaged as a *notebook* comprising web pages and editable parameterized scripts. Notebooks can be published and stored in web based archives for others to retrieve and modify. The notebook concept is similar to our TR and DV descriptions, but our virtual data descriptions are more generalized and decoupled from physical entities in order to support community-wide share and reuse.

The MyGrid project [17] has developed powerful distributed execution semantics and enactment engines, visual workflow specification, and provenance tracking. However, they do so only for web services, not standard application programs such as those used by the majority of today’s scientific applications.

6. CONCLUSIONS

We have described a virtual data system interface, Chiron, that allows users to describe, discover, share, and reuse scientific processes and data. Chiron has been applied successfully within the QuarkNet education project to provide service-level virtual data functionalities to specialized data analysis web applications.

Chiron also serves as a virtual data training tool. We loaded several virtual data applications, including high energy physics, cosmic ray detection, bioinformatics, and fMRI as examples into the portal, thus allowing first-time and novice virtual data users to explore core virtual data concepts and technologies without being distracted by configuration details.

We plan to leverage knowledge representation techniques to enhance metadata annotation and search, and transform service-oriented scripts into web service interfaces for advanced virtual data integration.

7. ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation GriPhyN project under contract ITR-086044. The GriPhyN Virtual Data System was implemented by Gaurang Mehta, Karan Vahi, Jens Voeckler, and Yong Zhao. Yong Zhao implemented the Chiron portal; Tom Jordan, Liz Quigg, Paul Nepwoda, Evgeni Peryshki, Nick Dettman, Yun Wu, and Eric Gilbert implemented the QuarkNet Cosmic Ray Collaboration portal. James Dobson provided the fMRI workflows and assisted in their testing. The Grid planners used in the Chiron portal are the work of Ewa Deelman's team at the USC Information Sciences Institute.

8. REFERENCES

- [1] Avery, P. and Foster, I. *The GriPhyN Project: Towards Petascale Virtual Data Grids*, 2001. www.griphyn.org.
- [2] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*. Journal of Network and Computer Applications, 23:187-200, 2001.
- [3] Foster, I., Voeckler, J., Wilde, M. and Zhao, Y., *The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration*. First Biennial Conference on Innovative Data Systems Research, Jan., 2004.
- [4] Foster, I., Voeckler, J., Wilde, M. and Zhao, Y., *Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation*. in 14th Conference on Scientific and Statistical Database Management, (2002).
- [5] A. Arbre, P. Avery, D. Bourilkov, R. Cavanaugh, S. Katageri, J. Rodriguez, G. Graham, J. Vöckler, M. Wilde. *Virtual Data in CMS Productions*. In Proceedings of Computing in High Energy and Nuclear Physics, March 2003.
- [6] Annis, J., Zhao, Y., Voeckler, J., Wilde, M., Kent, S. and Foster, I., *Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey*. in SC'2002, (2002).
- [7] A. Rodriguez, D. Sulakhe, E. Marland, V. Nefedova, M. Wilde, N. Maltsev. *Grid Enabled Server for High-Throughput Analysis of Genomes*, Workshop on Case Studies on Grid Applications, March 13, 2004, Berlin, Germany In conjunction with GGF10.
- [8] K. Leung, R. Heckemann, N. Saeed, K. Brooks, J. Buckton, K. Changani, D. Reid, D. Rueckert, J. Hajnal, M. Holden, D. Hill, *Analysis of serial MR images of joints*, IEEE International Symposium on Biomedical Imaging (ISBI) 2004, page 221-224.
- [9] J. Frey, T. Tannenbaum, I. Foster, M. Livny, S. Tuecke. *Condor-G: A Computation Management Agent for Multi-Institutional Grids*. Cluster Computing, 5(3):237-246, 2002.
- [10] E. Deelman, et al., "Pegasus: Mapping Scientific Workflows onto the Grid," Proc. 2nd EU Across Grids Conf., Cyprus, 2004.
- [11] Buneman, P., Khanna, S. and Tan, W.-C., *Why and Where: A Characterization of Data Provenance*. In International Conference on Database Theory, 2001.
- [12] The QuarkNet Project. <http://quarknet.fnal.gov/>, (June 2004).
- [13] K. Amin, et.al, GridAnt: A Client-Controllable Grid Workflow System. 37th Hawaii International Conference on System Sciences (HICSS-37-2004)
- [14] J. Myers et.al., A Collaborative Informatics Infrastructure for Multi-scale Science (pdf), Proceedings of the Challenges of Large Applications in Distributed Environments (CLADE) Workshop, June 7, 2004, Honolulu, HI
- [15] Stein, G., Web Digital Authoring and Versioning (WebDAV) Resources Community Website, <http://www.webdav.org/>, 2004
- [16] S. Krishnan et al., *The XCAT Science Portal*. In SC'2001, November 2001
- [17] Goble, C., Pettifer, S. and Stevens, R. Knowledge Integration: In silico Experiments in Bioinformatics. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2004
- [18] J. Van Horn, Online Availability of fMRI Results Images, Journal of Cognitive Neuroscience, 15(6):769-770, 2003.
- [19] Woods RP, et.al. Automated image registration: I. General methods and intrasubject, intramodality validation. Journal of Computer Assisted Tomography 1998;22:139-152.
- [20] Woods RP, et.al.. Automated image registration: II. Intersubject validation of linear and nonlinear models. Journal of Computer Assisted Tomography 1998;22:153-165.
- [21] AIR 5 Suite: <http://bishopw.loni.ucla.edu/AIR5/>