SDM Center FY 2010 Annual Report

http://sdmcenter.lbl.gov

Table of Contents

SDM Center FY 2010 Annual Report

http://sdmcenter.lbl.gov

Introduction

Managing scientific data has been identified by the scientific community as one of the most important emerging needs because of the sheer volume and increasing complexity of data being collected. Effectively generating, managing, and analyzing this information requires a comprehensive, end-to-end approach to data management that encompasses all of the stages from the initial data acquisition to the final analysis of the data. Fortunately, the data management problems encountered by most scientific domains are common enough to be addressed through shared technology solutions. Based on community input, we have identified three significant requirements. First, more efficient access to storage systems is needed. In particular, parallel file system and I/O system improvements are needed to write and read large volumes of data without slowing a simulation, analysis, or visualization engine. These processes are complicated by the fact that scientific data are structured differently for specific application domains, and are stored in specialized file formats. Second, scientists require technologies to facilitate better understanding of their data, in particular the ability to effectively perform complex data analysis and searches over extremely large data sets. Specialized feature discovery and statistical analysis techniques are needed before the data can be understood or visualized. Furthermore, interactive analysis requires techniques for efficiently selecting subsets of the data. Finally, generating the data, collecting and storing the results, keeping track of data provenance, data post-processing, and analysis of results is a tedious, fragmented process. Tools for automation of this process in a robust, tractable, and recoverable fashion are required to enhance scientific exploration.

Our approach is to employ an evolutionary development and deployment process: from research through prototypes to deployment and infrastructure. Accordingly, we have organized our activities in three layers that abstract the end-to-end data flow described above. We labeled the layers (from bottom to top):

- Storage Efficient Access (SEA)
- Data Mining and Analysis (DMA)
- Scientific Process Automation (SPA)

The SEA layer is immediately on top of hardware, operating systems, file systems, and mass storage systems, and provides parallel data access technology, and transparent access to archival storage. The DMA layer, which builds on the functionality of the SEA layer, consists of indexing, feature identification, and parallel statistical analysis technology. The SPA layer, which is on top of the DMA layer, provides the ability to compose scientific workflows from the components in the DMA layer as well as application specific modules.

Over the last six months we have focused on enhancing and using the existing SDM tools in response to application scientists' requirements. These advances were possible since we continue to have close interactions with many scientific application users, and their feedback to the efficacy and usefulness of our tools drives our enhancements and developments. This report only covers progress in the period of October 1, 2009 to March 31, 2010. Comprehensive reports on progress in previous years are available at the SDM center web site.

Selected Highlights of Achievements

- **The book on Scientific Data Management was published.** Members of the SDM center edited and contributed chapters to the book entitled "Scientific Data Management: Challenges, Existing

Technology, and Deployment" [SR09]. In six out of thirteen chapters, the lead authors are members of the SDM center, and additional members contributed to the content of these chapters.

- **Textbook titled "Practical Graph Mining with R" written entirely by students to be published by Chapman & Hall/CRC Press under the Data Mining and Knowledge Discovery Series**, has a final delivery date of January 31, 2011 (http://www.csc.ncsu.edu/news/1071). The lead editor and co-editors of the book are members of the SDM center. Proceeds will go to the NC State Department of Computer Science.
- **High productivity in the SDM center, a large number of papers published.** During the last year (October 2009 – October 2010) members of the SDM center published **65** papers (see publication list), and organized and presented numerous tutorials, invited talks, or invited sessions.
- **SDM center mentoring of high school students led to the 2nd place win at the National Siemens Math & Science Competition.** We have mentored a team of two high school students who developed a data analysis pipeline for front detection and tracking in XGC fusion plasma simulation data generated by the CPES fusion project. The students won the 2nd place at the National Siemens Math & Science Competition and a $50,000 scholarship with their project entitled "Supercomputing Analytical Discovery of Plasma Instabilities in Fusion Energy Reactors."
- **Study on I/O performance at leadership scale validates scaling of SDM I/O software.** In conjunction with the Argonne Leadership Computing Facility (ALCF), the SDM team recently completed a comprehensive study of I/O performance at leadership scales. It uncovers bottlenecks in the system, examines performance of high-level I/O libraries, and highlights the performance characteristics of a set of HPC application I/O kernels. The study validated that SDM software (PnetCDF, ROMIO MPI-IO, and PVFS) performed well at leadership scales (see details in section 1.1).
- **Preliminary results show potentials for great improvement of I/O efficiency by taking advantage of multi-core processors.** We recently implemented an opportunistic data compression scheme that can leverage spare cycles on multi-core clients to improve the efficiency of I/O. A flexible storage mapping facilitates storage of this compressed data in multiple files. The observed improvement on a small number of cores ranges from 2 times to 8 times better I/O bandwidth (see details in section 1.2).
- **Data aggregation significantly improves the I/O bandwidth of GCRM applications**. The SDM team recently developed a new data aggregation feature in Parallel NetCDF library that greatly leverages the I/O performance for the applications of Global Cloud Resolving Models (GCRM). A 140% improvement over the currently best I/O method was observed. (See details in section 1.3).
- **A new tool for automatic discovery of front detection developed.** We created an analytical methodology for automatic discovery of turbulent patterns, namely front detection and tracking, both in space and time, in the electrical potential fluctuation by plasma turbulence data from the XGC fusion simulations. The tool uses Automatic Parallelization of Data-Parallel Statistical Computing Codes (see details in section 2.1).
- **Orders of magnitude (100000 fold) speedup achieved for All Pairs Similarity Search (APSS).** The scalable algorithm was developed with specialized indices and heuristic optimization over data sets with millions of records in high dimensional spaces. We developed an open source library of algorithms for fast, incremental, and scalable all pairs similarity searches (see details in section 2.1).
- **A huge (1000 fold) speedup achieved with specialized FastBit structures for Gyrokinetic Fusion region identification.** We developed a specialized bitmap index FastBit structure by directly utilizing the mesh structure of Gyrokinetic Transport Code (GTC) for simulating the magnetically confined fusion plasma. Consequently, we were able to improve the speed of identifying regions of interest by nearly 1000-fold (see details in section 2.2).

- **Deeper insights into fusion data achieved.** The initial analysis by the SDM center of coherent structures in Gyrokinetic Simulation of Energetic Particle (GSEP) SciDAC center's fluid data is providing previously unexplored insights into the statistics of the structures in the ion heat flux variable. It discovered that there are some small structures with negative ion heat flux that need further investigation to determine if they are due to noise or physics (see details in section 2.3).
- **The FIESTA framework applied to a new code for real-time code monitoring**. We deployed an initial version of the Pixie workflow to fusion scientists at ORNL. This workflow ties the results of the Pixie code runs into the ORNL Dashboard through the VisIT command line interface, making the results quickly available to scientists (see details in section 3.6).
- **New "portable" dashboard developed.** The SDM center dashboard tool has been found to be extremely useful to scientists in viewing the results of simulation. Consequently, based on user's requests, a "portable" version has been developed, so it can be applied in any user facility. An initial version of the portable dashboard has been released, which allows scientists to download and interact with (parts of) their data sets while in a disconnected environment, such as on a plane during flight (see details in section 3.3).
- **"Resource-aware" workflow provenance recorder released.** We have issued a new release of the provenance recorder which is sensitive to application resource requirements. This new version has local data stores and is aware of potential competition for resources, including CPU and the network, and responds by slowing down its collection speeds. Thus, it is able to work effectively with larger and more computationally intensive workflows (see details in section 3.4).
- **The Adaptable I/O system (ADIOS) has been released for public use**. As a result, several application teams, including the Chombo and S3D teams, are now using ADIOS within their code frameworks. Several applications are now using ADIOS for in-situ code coupling and analysis tasks (see details in sections 1.3 and 3.6).
- **Simplified generic workflow components, patterns and templates were developed**. Advances in the understanding of end-users needs allowed us to develop generic actors, patterns and templates (i.e. workflow components and processes) which can be automatically generated and applied to a broad category of scientific problems; this includes release of generic actors, as well as of a prototype of a template-based workflow generation wizard. (See section 3.2)
- **New provenance collection and analysis tools developed**. We developed a flexible provenance collection and analysis infrastructure relevant to a DOE workflow environment. We also assisted in transfer of that technology to a broader community. (See section 3.4)
- **New workflow reliability and fault tolerance tools were developed in Kepler**. Development of workflow reliability and fault tolerance included development of a new model, and of the specifications and pilot versions of Kepler-based alternative actor for recovery from issues. (See section 3.5)

1. Storage Efficient Access (SEA)

The core I/O functionality on today's high-performance computing (HPC) systems consists of a collection of I/O software that provides a convenient and efficient interface to the available I/O hardware. The projects in this layer focus on this core I/O functionality, and they have two complementary goals. First, we develop and support a collection of highly-scalable and freely available I/O software components that

are used in production applications by scientists, and we actively engage the community to help application scientists better understand how to use these tools. Second, through our interactions with the community we identify specific deficiencies in functionality, performance, and usability that we then work to address. Successful improvements are subsequently integrated into production releases, ensuring that these benefits are made widely available.

Overall, our work can be placed in four categories, discussed in the following sections:

- File system benchmarking and application I/O behavior
- Parallel I/O infrastructure development
- Application interfaces and data models
- Outreach and education

1.1 File System Benchmarking and Application I/O Behavior

The high peak rates of HPC I/O systems simply do not translate into adequate sustained performance for computational science applications. The root cause of this performance gap is the mismatch between the requirements of the system's applications and the capabilities of I/O hardware and software. Systematic evaluation of both I/O system capabilities and application requirements provides much needed insight into the efficient use of existing systems and help guide the design of next-generation I/O systems.

The objective of this work is to study file system characteristics that have significant impacts to the parallel I/O operations and evaluate the relative performance of the file systems available to important SciDAC applications on DOE compute platforms. The performance, functionality, and scalability of MPI-IO, parallel netCDF, and HDF5 are critical for many applications.

Progress to Date

We started an investigation on the impact of various file access patterns to the file system performance in hope to understand the file system's locking behavior so that we can construct the best data redistribution policies in parallel I/O libraries, such as MPI-IO. We devised the benchmark using different numbers of I/O servers as well as client processes. The benchmark contains various mapping between the two sides, including round-robin interleaving, single-block contiguous, multiple-block interleaving, and N-server-to-M-client mapping. The purpose of this work is to understand the gap of I/O performance obtained in practice and in theory under the production parallel file systems that employ a distributed lock mechanism. We are developing this software on Franklin at NERSC and Abe at NCSA.

In conjunction with the Argonne Leadership Computing Facility (ALCF) we have recently completed a study of the I/O system of the 557 TFlop IBM Blue Gene/P at Argonne [LCL+09]. In this work we detail the performance of individual components that make up the system and then examine how the performance of various components impacts overall bandwidth for a variety of access patterns, including patterns exhibited by SciDAC and INCITE applications. This is to our knowledge the most comprehensive study of I/O performance at this scale. Following on this study and building on the Darshan I/O Characterization tool [CLR+09], developed at Argonne, we have completed the data-gathering phase of a multi-month study of I/O patterns on the ALCF system and worked with the ALCF team to improve performance for one of the underperforming codes identified in the study. The results of this work have been written up and submitted to the IPDPS 2011 symposium.

Working with SciDAC UltraVis Institute collaborators at the University of Tennessee, Knoxville, we have applied our parallel I/O knowledge in the context of climate data analysis, assisting in accelerating the data ingest process in a query-based climate data analysis tool [KGH+09]. This tool dramatically accelerates the analysis of observational data and provides a familiar, query-based interface atop a highly parallel analysis infrastructure.

1.2 Parallel I/O Infrastructure Development

Multiple parallel file system options are now available, and most HPC systems now include a rudimentary I/O software stack. However, the performance of the I/O stack on many systems is much lower than possible given the hardware available. As HPC systems scale and application complexity increases, extracting the highest possible performance from the I/O hardware is critical to the overall effectiveness of the system. The objective of this work is to improve the state of parallel I/O support for HPC. The Parallel Virtual File System (PVFS) and ROMIO MPI-IO implementations are in wide use and provide key parallel I/O functionality. This work builds on these two components by enhancing them in order to ensure these capabilities continue to be available as systems continue to scale. In addition to improvements to these tools, special attention is paid to Cray systems using the Lustre parallel file system.

Progress to Date

Through collaboration with the Argonne Leadership Computing Facility (ALCF) we have ensured that the I/O system on the Blue Gene/P system will meet performance and reliability goals. This includes aiding in the specification of the storage hardware, porting and deployment of PVFS at large scale [LCL+09], and working with IBM to solve a significant functionality problem in early versions of their MPI-IO software for the system. We implemented a lock-free driver for the Blue Gene that enables PVFS use, improved the scalability of some metadata operations, and integrated IBM's changes back into the ROMIO source tree.

Working with Cray and Sun Microsystems (now responsible for Lustre), we incorporated successful research efforts into the production **ROMIO MPI-IO library** [TGL99], including Lustre-specific improvements, file domain, and strided I/O optimizations. These optimizations are critical to MPI-IO performance on Cray XT systems, such as those at NERSC and ORNL.

We continue to develop various optimizations for the **I/O delegate and caching system**, an initial work from [NLC08]. It is a software layer in MPI-IO where certain tasks, such as file caching and consistency control are delegated to a small set of compute nodes, collectively termed as I/O delegate nodes. This layer is implemented at the bottom layer of ROMIO where it intercepts all the system I/O requests initiated by ROMIO and redirects them to delegate nodes. We incorporated a static file domain assignment approach proposed in our earlier work on MPI-IO file domain in [LC08]. We also explored the opportunity of using more than one delegate processes in a multi-core compute node machine. We anticipate this work can greatly benefit the performance of MPI independent I/O whose optimizations have been considered difficult by the parallel I/O community and almost none exists. We conducted our experiments using the FLASH and S3D I/O kernels on Lustre. Testing and development were performed on several parallel machines: Abe at NCSA and Franklin at NERSC. Our experiments show that using MPI independent I/O functions in the two application kernels can outperform the same kernels using the collective ones. The observed improvement ranges from 2 times to 8 times better I/O bandwidth.

Parallel NFS (pNFS) is touted as an emergent standard protocol for parallel I/O access in various storage environments. Several pNFS prototypes have been implemented for initial validation and protocol examination. Previous efforts have focused on realizing the pNFS protocol to expose the best bandwidth potential from underlying file and storage systems. Recently we performed an initial characterization of two pNFS prototype implementations, lpNFS (**a Lustre-based parallel NFS** implementation [YDV09]) and spNFS (another reference implementation from Network Appliance, Inc.) [Yu10]. We show that both lpNFS and spNFS can faithfully achieve the primary goal of pNFS, i.e., aggregating I/O bandwidth from many storage servers. However, they both face the challenge of scalable metadata management. Particularly, the throughput of spNFS metadata operations degrades significantly with an increasing number of data servers. The lpNFS architecture overcomes many of these deficiencies.

For its low-latency, high bandwidth, and low CPU utilization, RDMA (Remote Direct Memory Access) has established itself as an effective data movement technology in many networking environments. This includes recent incarnations of InfiniBand-based RDMA on long-distance networks. However, the transport protocols of grid run-time systems, such as GridFTP in Globus, are not yet capable of utilizing RDMA for fast data movement. We recently examined the architecture of GridFTP for the feasibility of enabling RDMA. An RDMA-capable XIO (RXIO) framework is designed to extend its XIO system to match the characteristics of RDMA in terms of both connection establishment and communication progress. An initial proof-of-concept implementation of RXIO is realized on InfiniBand. Our experimental results demonstrate that, compared to IPoIB and 10GigE, RDMA can significantly improve the performance of GridFTP, reducing the latency by 32% and increasing the bandwidth by more than three times. In achieving such performance improvements, RDMA dramatically cuts down CPU utilization of GridFTP clients and servers. These results demonstrate that RXIO is effectively designed and implemented to exploit the benefits of RDMA for GridFTP. This work provides a good prototype to further examine and prepare GridFTP on wide-area RDMA networks as they become available.

1.3 Application Interfaces and Data Models

In order to make applications more nimble with respect to their I/O behavior, more effort must be spent on the applications and the interfaces that they use to interact with the I/O system. The objective of this work is to improve the usability and observed I/O throughput for applications using parallel I/O by enhancements to or replacements for popular application interfaces to parallel I/O resources. This task was added in response to a perceived need for improved performance at this layer, in part due to our previous work with the FLASH I/O benchmark. Because of their popularity in the scientific community we have focused on the NetCDF and HDF5 interfaces, and in particular on a parallel interface to NetCDF files.

Progress to Date

Significant work has gone into making the **Parallel netCDF (PnetCDF)** [LLC+03] software ready for production. PnetCDF now supports large datasets, such as arrays with more than 4 billions of elements. The original UCAR netCDF format supports up to 2 billion elements (due to 32-bit integer data type limitations). We have developed an extension (that uses 64-bits for sizes), the “CDF-5” format, to allow one to store variables of effectively unlimited size. We are synchronizing these changes with the serial netCDF team so that serial tools can interoperate. PnetCDF also incorporates a Fortran90 module to better support Fortran90 programs for argument data type checking, which reduces programmers’ development and debugging time.

PnetCDF version 1.2.0 was released in August 2010, which is augmented with several new features, aiming to improve both productivity and performance. One feature is to enable data aggregation optimization in the non-blocking I/O APIs [GLC+09]. In the current form of PnetCDF, as in serial netCDF, I/O is carried out one variable at a time. For applications with a large number of small variables, accessing these variables results in small file accesses and poor performance. This new feature aggregates small requests into large one and can significantly enhance the I/O performance. Other new features include the file layout alignment for file header and all variables, a more scalable data consistency control, and a new parallel tool to compare two netCDF files.

We continue to support PnetCDF users, including the Geodesic Parallel I/O library (GIO) developed by the PNNL team lead by K. Schuchardt. GIO is used by the Global Cloud Resolving Models (GCRM), a large-scale climate simulation code. We developed an additional I/O method in GIO library that uses the new non-blocking I/O feature and observed 140% of write bandwidth improvement over the best I/O method in GIO. Below is the performance chart presented in the Workshop on High-Resolution Climate Modeling 2010.

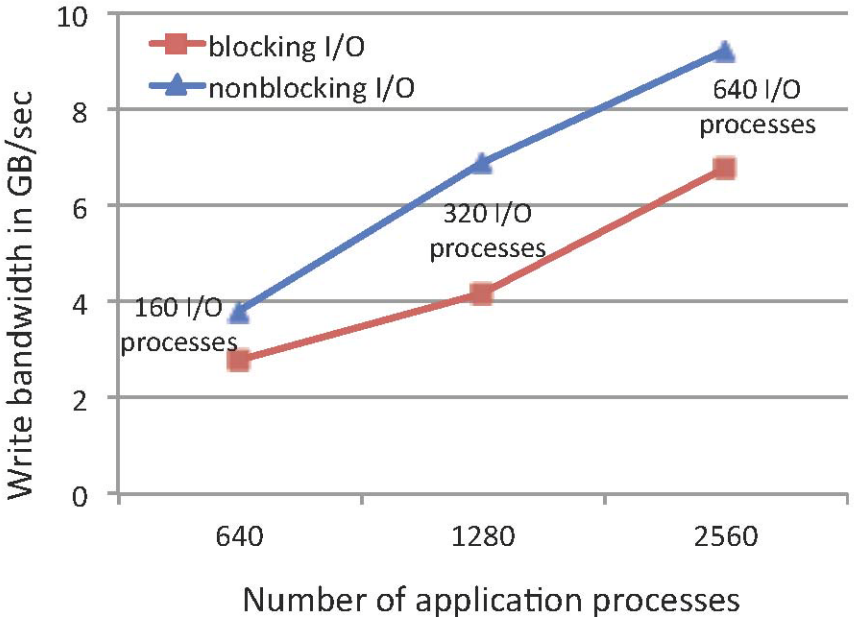

Figure 1.1 Improved I/O performance in the Geodesic Parallel I/O library (GIO)

We also improved the PnetCDF method for one of the GCRM tools that extracts and generates image data from the simulation output files. Likewise, we have worked with the FLASH astrophysics team to adopt the aggregation features of PnetCDF 1.2.0 in their checkpoint routines. This has resulted in substantial speedups, and a paper on these improvements is in progress.

We have completed implementations of **Disk Resident Multidimensional Extendible Array (DRXTA)** functions in C to create, read, write and manipulate out-of-core arrays stored in Unix file systems [OR07]. Support for accessing Parallel Extendible Array files in Global Array applications has been implemented as well, providing an alternative to Disk Resident Arrays. Array chunks are cached in and out of memory using the cache pool implementation from BerkeleyDB, and we have experimented with alternatives to LRU for cache replacement in this system. We have also implemented skip-list access methods for chunked dense arrays, also using the BerkeleyDB cache pool module underneath. This is a more common method for organizing this type of data, and using this implementation we performed a performance comparison of DRXTA storage of dense arrays with skip-list indexing of dense array chunks. Currently we are evaluating the DRXTA approach as an alternative storage organization for use in storing chunked data within the HDF5 library.

The ADaptable I/O System (ADIOS) is an I/O framework developed to address both the differing performance characteristics as HPC codes move from one platform to another as well as to provide a flexible framework for integrating various I/O related mechanisms to enhance productivity and performance. It affords the end user different I/O methods for different groups of data in an application, such as restarts and diagnostics, so that the best performance can be achieved by employing the most efficient methods based on I/O patterns and run-time platform characteristics, all without changing the source code. ADIOS has been successfully deployed on Jaguar, the Cray XT5 system at Oak Ridge National Laboratory, and the Franklin Cray XT4 at NERSC both using Lustre file system, with proved performance improvement. We recently evaluated the alternative I/O methods of ADIOS for high performance I/O on both file systems on the very different architecture of Blue Gene/P. Initial performance results indicate that ADIOS is able to deliver an efficient and scalable I/O solution for the Blue Gene platform. We also tuned ADIOS methods for better I/O performance on BG/P.

Advances in multicore technologies lead to processors with tens and soon hundreds of cores in a single socket, resulting in excessive computing power compared to the available memory and I/O bandwidth for

data handling. It would be desirable if some surplus in computing power can be transformed into gains in the efficiency of I/O. We recently designed and implemented an **opportunistic data compression scheme**, called NEarline data COmpression and DECompression (neCODEC), for data-intensive parallel applications. Several techniques are introduced in neCODEC, including an elastic file representation and hierarchical metadata management. A neCODEC file consists of an elastic number of data files (a.k.a subfiles) and a metafile that stores metadata to the data chunks in the subfiles. Hierarchical metadata management is designed to provide scalable management of metadata records for data chunks of these subfiles.

1.4 Outreach and education

We take outreach very seriously. We have presented 2 full-day tutorials on topics related to storage and parallel I/O in the last year months at the SC conference series and as well as invited talks at conferences, workshops, and universities. We actively participate in DOE Exascale workshops and other application-oriented meetings to help educate the community on I/O best practices, and we continue to help organize the annual HEC FSIO meeting, helping guide research into file systems and I/O for high-end computing [GNB+09]. This year's meeting was August 2-4 in Arlington, VA.

We have recently led the development of two chapters on parallel I/O for books on data management and analysis as well [RCG+09, RCM09], now in publication, and two sections in the upcoming *Encyclopedia of Parallel Computing* [Latham10,Ross10], edited by Dr. David Padua (UIUC).

We organized the second international workshop on Interfaces and Architectures for Scientific Data Storage (IASDS) (http://www.mcs.anl.gov/events/workshops/iasds10/), in Heraklion, Crete, Greece, providing a forum for engineers and scientists to present their most recent work in this area. This was held in conjunction with the IEEE Cluster conference.

Talks and Tutorials

R. Ross, "Preparing for Exascale: Understanding HPC Storage Systems," Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS), Heraklion, Crete, Greece, September 2010.

R. Ross, "Data Models and Data Analysis at Exascale," High-End Computing File Systems and I/O Conference, Arlington, VA, August 2010.

Robert Latham, "Parallel I/O in Practice," CScADS Workshop on Leadership-class Machines, Petascale Applications, and Performance Strategies, Snowbird, UT, July 2010.

Robert Latham, "Parallel I/O in Practice," Big Data for Science Workshop, Virtual School of Computational Science and Engineering, July, 2010.

R. Ross, "Scientific Computing at Extreme Scale," University of Connecticut, Storrs, CT, June 2010.

R. Ross, "Storage in an Exascale World," IEEE International Workshop on Storage Network Architecture and Parallel I/Os (SNAPI), Incline Village, NV, May 2010.

R. Ross, "Applications, Data, and the Future of Storage in Computational Science," SCI Institute, University of Utah, Salt Lake City, UT, May 2010.

Robert Latham, "Middleware Libraries for Parallel I/O," Carnegie-Mellon University, Pittsburgh, PA, April 2010.

Robert Latham, "Making the Most of the I/O Software Stack," Extreme Scale I/O and Data Analysis Workshop, Austin, TX, March 2010.

Robert Ross, "Input/Output (I/O) in Computational Science," University of Chicago, Chicago, IL, February 2010.

Alok Choudhary, "Detailed Analysis of I/O Traces of Large Scale Applications", The International Conference on High-Performance Computing, India, Dec. 2009.

Robert Latham, Robert Ross, Marc Unangst, and Brent Welch, "Parallel I/O in Practice," SC 2009, Portland, OR, November 2009.

William Gropp, Ewing Lusk, Robert Ross, and Rajeev Thakur, "Advanced MPI," SC2009, Portland, OR, November 2009.

Robert Ross, "Extreme Scale I/O Systems," IEEE Nuclear Science Symposium Data-Intensive Workshop, Orlando, FL, October 2009.

2. Scientific Data Mining and Analysis (DMA)

The Data Mining and Analysis (DMA) layer provides the data-understanding technologies necessary for efficient and effective analytics of complex scientific data. This is accomplished through the development and deployment of the three core technologies:

- High performance parallel statistical computing
- High-Dimensional Indexing Software
- Feature extraction and tracking for scientific applications

2.1 High performance parallel statistical computing

Data produced by the extreme-scale simulations are not only massive in size but also inherently complex due to the generally nonlinear, multi-scale and dynamic nature of the underlying physical phenomena. Yet, techniques for analyzing these complex signals are in their infancy. In particular, the dynamics of large, or *meso-scale*, turbulence patterns and structures in fusion plasma has not been seriously addressed. These issues are relevant to galactic dynamics simulations, and so are of interest beyond magnetic fusion energy. To address some of these issues, we focused on the following data analysis problems:

- *Analysis of fluctuation energy distribution*–the observed composite energy signal is distributed in space and time. We focused on discovery of turbulent patterns in the $dphi^2$ XGC energy data, where $dphi^2$ is the square of electrical potential fluctuation by turbulence.
- *Mapping and tracking the evolution of turbulence in space and time*–we developed a methodology for *multi-resolution* (across space and time) analysis of turbulence, especially, through front-tracking to establish such dynamics.
- *Efficient large-scale analytical data processing*—even if the tools for solving these data analysis problems existed, performing such knowledge discovery tasks for trillions of particles across thousands of time steps presents a computational challenge. To address this issue, we improved the execution efficiency of such tools. Specifically, we developed an advanced middleware for automatic parallelization of data analysis tasks and scalable execution of these tasks in hybrid parallel, multi-node (with multiple processors), multi-core (with multiple units within a processor) environments, often found in supercomputers.

Progress to Date

To date, the framework that aims to enrich and optimize the knowledge discovery cycle has three major components, which parallel the three core steps of the knowledge discovery cycle:

Automatic Spatio-Temporal Turbulent Front Detection and Evolution in Fusion Plasma

Few would argue that fusion energy has been the Holy Grail of renewable energy efforts. The grand challenge is to produce more energy through a fusion reaction than that required to initiate the process in a reactor. A key bottleneck is the turbulence, or unstable motion, of the fusion plasma. Turbulence influences the degree of energy lost by plasma during the fusion process; therefore, controlling the turbulence is critical to viable energy production. Discovery of dynamic turbulent patterns and trends from the data produced by a computer-simulated fusion reaction offers a potential to reveal ways to control the turbulence. Yet, it presents a challenge: how to effectively and efficiently analyze the massive amounts of data, which is inherently complex, noisy, and high-dimensional. To address this challenge, we created an analytical methodology for automatic discovery of turbulent patterns, namely front detection

and tracking, both in space and time, in the electrical potential fluctuation by plasma turbulence data from the XGC simulations (see Figure 1). This work was conducted in collaboration with Dr. C.S. Chang, NYU. This process can potentially predict the structure, dynamics, and function of fusion plasma turbulence. It could also enable similar analyzes required in other disciplines, such as astrophysics and oceanography.

One strategy is the one of *reduced, yet informative, data representation* for the target data analysis task. For a fixed time-step, t_0, we approximated with line segments in a spatial region around the point of interest, r. The points corresponding to the fronts are the points, where the line segments change their slope from the direction almost parallel to the *x*-axis (green) to the direction almost parallel to the *y*-axis (blue and red) (see Fig. 1.a). Such an approach only required the slope and intercept of the approximating line segments for a few sequential sliding windows. The other steps of the end-to-end front detection and tracking process (see Fig. 1.b) have been local, by nature, and have utilized *pRapply*() method for a multi-node multi-core parallel execution with an ideal speed-up, as described next.

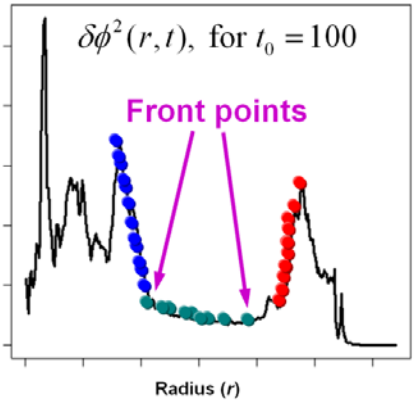

Fig. 1. Front detection and tracking in fusion simulation data.

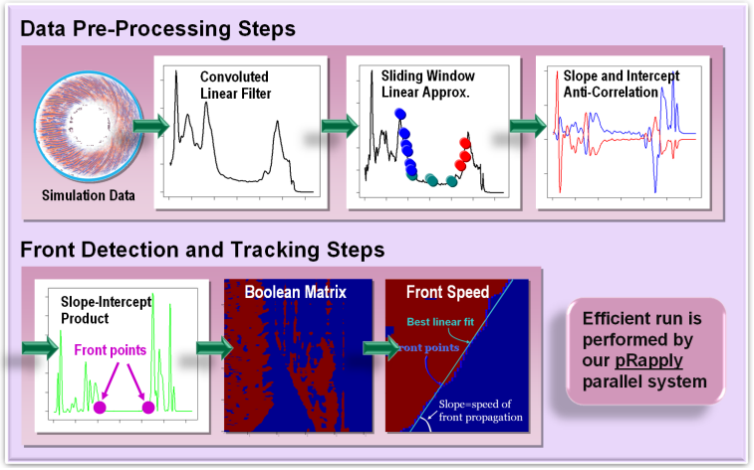

Fig. 2. A multi-step turbulent front detection and tracking process

Automatic Parallelization of Data-Parallel Statistical Computing Codes with pRapply in Hybrid Multi-Node Multi-Core HPC Environments

The increasing size and complexity of modern scientific data sets challenge the capabilities of traditional statistical computing. High-Performance Statistical Parallel Computing is a promising strategy to address these challenges, especially as multi-core parallel computing architectures become increasingly prevalent. However, parallel statistical computing introduces implementation complexities and, therefore, an automatic parallelization approach would be ideal. Data-parallel statistical computations that aim to evaluate the same function on different subsets of data represent natural candidates for automatic parallelization due to their inherent inter-process independence.

In FY09, we extended the *pR* middleware [BH+09] with *pRapply*() function for the *R* open-source statistical environment to support automatic parallelization of data-parallel tasks in multi-node, multi-core, and hybrid environments [BK+09]. *pR* requires few or no changes to existing serial codes, offers a linear speed-up with the increasing number of processors, and yields over 50% end-to-end execution time improvements in our tests, compared to the commonly used *snow R* package. We released *pRapply*() as open source software.

We also extended the capability of RScaLAPACK library to support openMPI back-end in response to multiple users' requests, eased RScaLAPACK's installation via improved autoconf, provided processor grid manipulation routines, provided both static and dynamic MPI library support, etc. The updated version was released on the R's CRAN web-site.

Fast, Incremental, and Scalable All Pairs Similarity Search.

Searching pairs of similar data records is an operation required for many data mining techniques like clustering and collaborative filtering. As the scale of the data has been increasing to several millions or billions of records in a high dimensional space, enabling fast and incremental similarity search over such data sets has become a formidable task. To address this challenge, we developed an open source library of algorithms for fast, incremental, and scalable all pairs similarity searches through improved indexing, systematic heuristic optimizations, and parallelization.

First, we designed a sequential algorithm for all pairs similarity search (APSS) that involves finding all pairs of records having similarity above a specified threshold. Our proposed fast matching technique speeds up APSS computation by using novel tighter bounds for similarity computation and indexing data structure [AS09a]. It offers the fastest solution known to-date with up to 6X speed-up over the state-of-the-art existing APSS algorithm.

We further addressed the incremental formulation of APSS problem, where APSS is performed multiple times over a given data set while varying the similarity threshold. The goal is to avoid redundant computations across multiple invocations of APSS by storing history of computation during each APSS. Depending on the similarity threshold variation, our proposed history binning and index splitting techniques achieve speed-ups from 2X to over 10^5 X over the state-of-the-art APSS algorithm [AS09b, ASB09]. To the best of our knowledge, this is the first work that addresses this problem.

Finally, we designed scalable parallel algorithms for APSS that take advantage of modern multi-processor, multi-core architectures to further scale-up the APSS computation. Our proposed index sharing technique divides the APSS computation into independent tasks and achieves ideal strong scaling behavior on shared memory architectures [AS10]. We also propose a complementary incremental index sharing technique, which provides a memory-efficient parallel APSS solution while maintaining almost linear speed-up. Performance of our parallel APSS algorithms remains consistent for datasets of various sizes. To the best of our knowledge, this is the first work that explores parallelization for APSS. We demonstrate the effectiveness of our techniques using real-world million record data sets.

Future Plans

- Provide an initial support of analytics functions inside of the ADIOS I/O library in collaboration with Scott Klasky.
- Explore the ways to incorporate fusion science analytical pipelines such as front tracking (not described here) through *in-situ* data analysis in staging.

2.2. High-Dimensional Indexing Software

Many scientific applications are generating large amounts of data; however, the key to gain insight is often associated with a relatively small number of records. For example, in a study of turbulent combustions, these special data records might be called ignition kernels. In a study of laser-wakefield particle accelerators, these special data records might be called particle bunches. Sifting through mountains of data to locate these "interesting" data records is a significant challenge. To meet this challenge, we have been developing and expanding an indexing software package, called FastBit.

FastBit is based on a database indexing technique called the bitmap index. This type of index is well suited for searching scientific data, where the data records usually remain unchanged after they are created. Taking advantage of this "read-only" nature, FastBit indexes answer queries fast by sacrificing some efficiency in updating the indexes. This allows us to package the indexing data structures tightly, reduce the I/O requirement when answering a query. Additionally, FastBit is designed to work with user data in their existing formats, instead of demanding the user data to be transformed into a particular format or loaded into a database management system. This flexibility makes it possible for users to accelerate their search operations with a minimal amount of change to their existing data analysis framework.

Progress to Date

We briefly describe the work in the last year as two sets of tasks: software development and application support.

Software development

In the past year, we have implemented three new features, 64-bit offsets for index sizes, approximate string matching, and support for recursive queries. The first feature allows the index sizes to grow beyond 2^{31} bytes (2GB), which is important for data set with larger sizes. The original design demands the users to break up their data into smaller units called *partition*s. However, there are many cases when explicitly breaking a data set into partitions is inconvenient. Increasing the internal offsets to be 64-bit long allows the index sizes to grow beyond the current limit, and makes it easier to work with larger datasets.

The second new feature allows the wildcard characters to be used with the SQL operator LIKE, such as 'sequence LIKE %ACGTTA%,' to find any string value (named sequence) containing the substring 'ACGTTA.' This is useful in applications involving string values. The current implementation contains basic implementation; additional work is required to improve the performance of such queries.

The third new feature is the support for recursive queries. This is implemented via in-memory data partitions and allows these partitions to be queries in the same way persistent data partitions are allowed. There are a number of SQL queries that can be more conveniently expressed as this type of recursive queries.

Application support

The FastBit developers are continuing the collaboration with the Visualization group and the LOASIS laser plasma accelerator group, both at LBNL, to improve the visual data analysis software for exploring the simulation data produced from the laser wakefield particle accelerator simulation [RGC+09]. The underlying software, called H5Part, has been significantly expanded recently to expose more functionality from FastBit. In particular, we have improved the interface between FastBit and H5Part to reduce the cost of using 3D histograms. For large histograms involving millions of bins, the new light-weight interface reduces the overall analysis time by a factor of two [Rub09].

We have also been working with a fusion application to accelerate the exploration of coherent spatial objects known as regions of interest. By incorporating the special mesh structure used in the gyrokinetic transport code (GTC) for simulating magnetically confined fusion plasma, we are able to turn the FastBit search results into regions of interest much more efficiently than earlier approaches. Preliminary testing shows that the new approach can identify regions nearly 1000 times faster than the commonly used approach. A report on this work is in preparation.

We are also supporting a number of external users. Here are two interesting examples. Prof. Siqueira and colleagues have recently published an extensive study of their Spatial Bitmap Index SB-Index in Journal

of the Brazilian Computer Society, 2009; 15(1):19-34 (http://www.scielo.br/pdf/jbcos/v15n2/v15n2a03.pdf), where they showed that the SB-Index is much more compact than competing join indexes, and at the same time, answers queries much faster. In some cases, it can be a hundred times faster. Dr. Luca Deri and his NTOP team have also been using FastBit for a number of analysis tasks for network traffic data collected through their NTOP system. They have also published a report of their work in an international conference on network security.

Another application of FastBit is in the area of biology. Protein identification is one of the important objectives for proteomic and medical sciences, as well as for pharmaceutical industry. With recent large-scale automation of genome sequencing and the explosion of protein databases, protein identification, it is important to exploit latest data processing technologies and design highly scalable algorithms to expedite the process of protein identification. We have designed, implemented, and evaluated a new software tool, Bitmapped Mass Fingerprinting (BMF), that can efficiently construct a bitmap index for short peptides, and quickly identify candidate proteins from leading protein databases [YWX+10]. BMF is developed by integrating the FastBit indexing technology and the popular Message Passing Interface (MPI) for parallelization. By exploiting FastBit for peptide mass fingerprinting across protein boundaries, we are able to accomplish parallelized computation and I/O for a scalable implementation of protein identification. Our experimental results show that BMF brings dramatic performance improvement for protein identification from various protein databases. In particular, we demonstrate that BMF can effectively scale up to 8,192 cores on the Jaguar Supercomputer at Oak Ridge National Laboratory, achieving superb performance in identifying proteins from the NCBI non-redundant (NR) protein database.

Future Plans

- Start to work with ADIOS team to make FastBit indexing functions available for in situ index creation.
- Continue the collaboration with the Visualization group on integrating FastBit into HDF5 and applying the resulting software to more applications.

2.3 Feature extraction and tracking for scientific applications

As the data from scientific simulations, observations, and experiments approach the petascale and beyond, scientists are interested in extracting and tracking features of interest in these data. This topic area focuses on the development and application of scientific data mining techniques for such analysis. We use techniques from image and video processing, machine learning, statistics, and pattern recognition, to find useful information in massive, complex data sets [Kam09]. Our goal is two-fold – to improve our understanding of scientific phenomena and, as appropriate, to deploy our solutions for use by application scientists. Over the years, we have worked with a number of application projects, initiated by the SDM center or at the request of the domain scientists.

Progress to Date

The problems we focus on are driven by applications scientists. Each problem presents different challenges and requires different techniques. The challenge is not only to discover the combination of techniques that addresses the problem at hand, but also to discover new approaches for previously unsolved problems. Some projects involve analysis that had never been done before; any results have to be carefully analyzed by both the data analysis and the domain experts to ensure that the conclusions drawn are scientifically correct and meaningful. This is achieved by working closely with the application scientists, understanding their problems, providing solutions, and iterating the process. In the last year (October 2009-September 2010), we had great success in addressing two problems as described below.

The first project is a collaboration with the GSEP SciDAC (Zhihong Lin, PI). The analysis goal is to identify coherent structures in GSEP simulation fluid and particle data and to understand the non-linear

interactions between the two. This is difficult as: (i) there is no definition of coherent structures; (ii) they vary extensively over time making it difficult to identify robust algorithms; (iii) the fluid data are on a twisted toroidal mesh while the particle data are unstructured, making existing algorithms inapplicable; and (iv) the data are currently in terabytes, with petabytes expected in the future. In FY09, we had implemented an initial algorithm that used several variables to identify the structures in the fluid data. Discussions with Zhihong Lin and Yong Xiao indicated that the approach also made sense from the physics viewpoint. The data from the smaller Ion Temperature Gradient (ITG) simulation were then analyzed at select time steps to extract statistics on the structures. This simulation has 32 toroidal planes with 40,000 grid points per plane. The initial results for a plane were very interesting, prompting the analysis of all planes at a time step. The preliminary conclusions from this analysis were i) that the event size distribution needs further analysis to confirm the type of distribution and ii) there are some small structures with negative ion heat flux that need further investigation to determine if they are due to noise or physics. To investigate these issues, a larger ITG simulation was run, with 64 poloidal planes and 600,000 grid points per plane. The results were consistent with the smaller data set.

These results prompted a comparison with the Collisionless Trapped Electron Mode (CTEM) simulation to understand the physics better. The distribution of the ion heat flux in CTEM was different from ITG, requiring a different algorithm to identify the structures. We also found that there were a lot more negative structures which alternated with positive flux structures along a flux surface (see Figure 2.1). Both the statistics on the structures and a visual tracking of the structures over time indicated that from an analysis viewpoint, these structures were not due to noise. The issue is being further investigated by GSEP physicists. The results on ITG simulations were presented in a poster at the 2010 Sherwood Fusion Theory Conference in April 2010 [KXL10], while the results of both ITG and CTEM analysis were presented at the GSEP Annual Meeting [KXL10a]. This work is partially supported by the GSEP SciDAC Center.

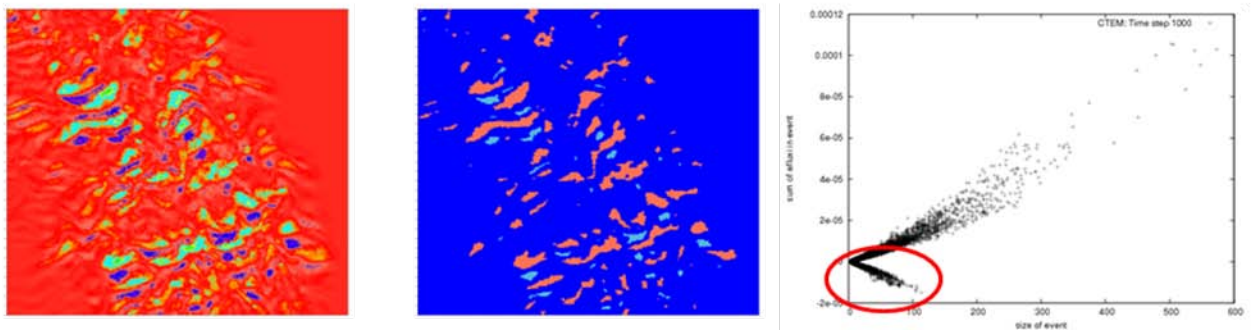

Figure 2.1: (a) A subset of the data on a2-D poloidal plane from the CTEM simulation showing the ion heat flux variable. (b) The structures in (a) with the positive structures in red and negative ones in blue. (c) A plot of the sum of ion heat flux in a structure vs. the size of the structure, clearly showing the large number of structures with negative flux.

The second project, started in early 2009, is a collaboration with the WindSENSE project funded through EERE. This opportunity in renewable energy resulted from the 2007 DOE CRNARE workshop which focused on EERE missions and Office of Science capabilities. Since wind energy is intermittent, it is difficult to schedule it on the power grid while maintaining its reliability. In particular, ramp events, where the wind power increases or decreases by a large amount in a short time, are becoming a major problem. We are using data mining techniques to understand wind ramp events better. This work is being done using data from Southern California Edison (SCE) and Bonneville Power Administration (BPA). In FY09, we had obtained some early statistics on the ramp events for Columbia Basin for 2007-2008. Our FY10 work included the analysis of 2009 data from BPA and the 2007-2008 data from SCE. The BPA analysis showed that with increasing generation, the extreme events became more frequent and severe. While some observations from 2008 carried over to 2009, others did not. For example, in 2008, negative

ramp events rarely occurred in the morning, but this was no longer true in 2009. Also, while some months (such as March, June, and August) tended to have more ramps in both years, some months, such as November, which had few ramps in 2008, had more ramps in 2009. In addition, through detailed analysis, we showed that there were no substantial differences in the three commonly used definitions of ramp events. These results were summarized in a technical report and were presented at the IEEE T&D conference in April 2010 [Kam10a, Kam10b]. This work is mainly supported by the WindSENSE project.

In addition, as part of our outreach effort, we presented our work at the Office of Science Graduate Fellowship meeting held at ANL in August to introduce recent graduate student fellows to the work being done in the Office of Science [Kam10c].

Plans for the Future

In the near future, we will:

- Identify the distributions of different statistics extracted from the ITG and CTEM simulations and track the structures over time for "volume" statistics.
- Complete the journal paper on classification of orbits; identify and extract "X points" in the separatrix orbits and deploy the code for use by fusion scientists.
- Complete the identification of the blobs in NSTX images for the shorter sequences, extract relevant features, track the blobs by exploiting the overlapping between frames [GK08], and extend the approach to the longer sequences of 7000-8000 frames.

This work is being done in close collaboration with the domain scientists. In addition, we will continue to explore opportunities for data analysis in additional Office of Science applications.

3. Scientific Process Automation (SPA)

Effectively generating, managing, and analyzing scientific data requires a comprehensive, end-to-end approach that encompasses all stages from the initial data acquisition to the final analysis of the data. As part of the SPA thrust area, we are developing a suite of tools and frameworks that integrate into a robust and auditable system for automation of scientific processes to enhance and speed up scientific discovery [CRI09]. Our technologies provide run-time management of the workflow processes, provenance collection, and analysis and display of results. This has led to the deployment of production workflows that allow scientists to a) monitor, in near-real-time, complex tasks such as the execution of large simulation codes, and b) facilitate complex analyses of the process metadata and of the simulation results. This has resulted in significant savings in scientists' time, in more efficient use of resources, and in a more cost-effective scientific discovery process overall.

Workflow technologies have a long history in the database and information systems communities [GHS95]. Similarly, the scientific community has developed a number of problem-solving environments, most of them as integrated solutions [HRG+00]. Component-based solution support systems are also proliferating [CL02, CCA06]. Scientific workflows merge advances in all these areas to automate support for sophisticated scientific problem-solving [LAB+06, LG05, DOE04, ABB+03, BVP00, VS97, SV96]. We use the term scientific workflow as a blanket term describing a series of structured activities and computations (called workflow components or actors) that arise in scientific problem-solving as part of the discovery process. This description includes the actions performed (by actors), the decisions made (control-flow), and the underlying coordination, such as data transfers (dataflow) and scheduling, required to execute the workflow. In its simplest case, a workflow is a linear sequence of tasks, each one implemented by an actor. An example of a scientific workflow is: transfer a configuration file to a large cluster, run a simulation passing this file as an input parameter, transfer the results of the simulation to a secondary system (e.g. a smaller cluster), select a known variable, and generate a movie showing how this

variable evolves over time. Scientific workflows can exhibit and exploit data-, task-, and pipeline-parallelism. In science and engineering, process tasks and computations often are large-scale, complex, and structured with intricate dependencies [DOE04, DBN+96, EBV95, Elm66].

We have not only developed a considerable body of software, but we have transferred our technology to a number of ongoing science projects, published numerous papers, and conducted several tutorials and workshops. The challenge is to provide adequate tools and support for four categories of user levels:

- Level 1: Scientist (uses workflows, uses parameterized templates and web-based **wizards** to adapt existing workflows for own use)
- Level 2: Advanced Users (writes more complex new workflows using existing tools, including Kepler-level graphical user interface)
- Level 3: Workflow, template, and actor developer, very advanced workflows
- Level 4: Workflow framework and engine developer

Level 3 and 4 tools and environments are well understood and we have a comprehensive suite of tools and educational materials for those two levels. We have completed studying level 1 and 2 interface and interaction needs, and we are in the process of developing a comprehensive suite of wizards and educational tools for those to level. In the past year, our contributions to advancing the state-of-the-art in scientific workflows have focused on the following areas. Progress in each of these areas is described in subsequent sections.

- **Workflow development.** The development of a deeper understanding of scientific workflows "in the wild" and of the requirements for support tools that allow easy construction of complex scientific workflows;
- **Generic workflow components, patterns and templates.** The development of generic actors, patterns and templates (i.e. workflow components and processes) which can be broadly applied to scientific problems;
- **Dashboard development.** The development of a one-stop-shopping workflow monitoring and analytics dashboard;
- **Provenance collection and analysis**. The design of a flexible provenance collection and analysis infrastructure within the workflow environment; and
- **Workflow reliability and fault tolerance.** The improvement of the reliability and fault-tolerance of workflow environments.

3.1 Workflow development

The original base-line contribution of the SPA team has been to co-found the Kepler project – an open source workflow support environment [http://www.kepler-project.org]. Kepler is now a widely accepted scientific workflow development and execution environment that powers a number of research and production projects all over the world. The SPA researchers and engineers continue to regularly contribute to Kepler. We are constantly working with the Kepler Core team [e.g., AJB+04, LAB+06, GBA+07] to enhance Kepler at all levels including the user interface, documentation, and tutorials. Our work has led to a significant reduction in the effort required to generate real workflows [ABC+06, SAC+07]. We also actively partner with science teams to transfer technology to their projects and develop and deploy their workflows. This provides real-life case-studies which are then used to enhance Kepler requirements, to identify Kepler enhancements, generic functionalities, and canonical generic workflow solutions, and to improve user interfaces.

In general we distinguish workflow for three stages of scientific computational processes: a) preparation workflows – those used to prepare data and environments for simulations that will run on supercomputers; b) run-time simulation workflows – those that manage launching of the jobs and run-time monitoring, data collection, and steering, and c) post-processing workflows – those that facilitate output management, viewing, post-run analytics and knowledge creation, archiving, and other post-processing activities. Of course, we also have end-to-end workflows, those that encompass preparation, launching, monitoring and post-processing.

Progress to Date

As active participants, and founding members, in the Kepler research community we contribute to the development of the Kepler environment. Specifically, we participate in Kepler and Ptolemy workshops that have produced significant enhancements in the underlying workflow environment, and we have identified and implemented new requirements for scientific workflows, fixed bugs, and improved the overall software development environment, as well as execution-time interfaces and data collection practices. Most recently, much of our focus has been in hardening the current workflows used in the fusion and combustion simulations. We have also extended the workflows to allow ADIOS in situ methods to communicate to the Kepler data base, allowing users to perform in memory and file based coupling using the same Kepler workflow. This capability was demonstrated at the Fusion Simulation Project meetings.

3.2 Generic workflow components, patterns and templates

Scientific workflows are a set of actions performed in a given sequence in scientific problem solving. The workflows of interest to the SDMC deal with huge amounts of data, and one of the issues that arises is data-movement and whether computations should go to the data or vice versa. Currently, construction of complex workflows using available workflow capturing and development tools, languages and engines is very much an art. On the other hand, Level 1 (scientists) and Level 2 (advanced users) users prefer to focus on their domain, and use workflows provided they can construct them in the simplest possible manner. We are developing a set of patterns and templates suitable for of scientific workflows.

Many workflows contain sections with very similar functionality but subtle differences in how that functionality is obtained. For example, transferring a file between machines, submitting a job to a batch processing system, monitoring the execution of a running job, and remotely executing a command are found in almost all of the scientific workflows we have developed. However, the actual implementation of these capabilities varies dramatically depending on features such as the specific machine configurations (e.g., which batch processor is used), the security requirements (e.g. ssh or rsh, certificates or one-time-passwords), and the workflow requirements (e.g. failover options, fault tolerance requirements, validation options). Tools that provide the instantiation of a case-specific workflow or workflow component from more general templates or components would substantially improve the ability to reuse existing solutions, leading to greater productivity when developing workflows [e.g., DKV97, YGN09],

In software engineering, a design pattern [e.g., GHJ+05, DKV97] is a reusable abstracted solution for, or an approach to solution of, different variations of frequently occurring problems. A template, on the other hand, is a more specific [e.g., BBC+94]. It focuses on explicit details of the solution, explicit parameterization, and even explicit codes – a template can be composed of multiple patterns. In the case of small scale design patterns, templates and design patterns often can be used interchangeably. Patterns make life simpler by providing faster solutions that are known to work thus minimizing rework by allowing us to reduce a problem to a known solution. Design patterns in general, can be described as a canonical solution to a specific problem, which then needs to be refined, modified and customized to suit the problem. The customization of a design pattern depends on the input parameters and the specifications of the problem. Pattern oriented model development is a well-known and widely accepted initiative in workflow and traditional application development [e.g., GHJ+95].

Progress to Date

Despite KEPLER being employed for the implementation of most SDM processes, the design and development of scientific workflow models are still not well understood by scientists and workflow developers in general. Pattern oriented model development is a well-known and widely accepted initiative in workflow and traditional application development. Basically, a pattern is the abstraction from a concrete form, which keeps recurring in specific non-arbitrary contexts. Workflow patterns help to study "the suitability of a particular process language or workflow system for a particular project, assessing relative strengths and weaknesses of various approaches to process specification, implementing certain requirements in a particular process-aware information system, and as a basis for language and tool development".

We have developed an initial set of 15 understandable workflow patterns and different versions for each pattern from the previous SDM workflows. Currently, we have been developing an application platform that will allow non-specialist scientists to create their workflows using those patterns without dealing with low-level implementation details of Kepler. In addition to the abstract patterns, we have implemented a number of templates in cooperation with project collaborators at NCSU that abstract many of the most widely used steps in simulation workflows, i.e., job submission, check out of a code.

In relation with the scientific workflow patterns work, we developed an infrastructure extension for Kepler that allows workflow designers to automate the definition of rescue fragments. Thus, with respect to the predefined error types and error tokens, the rescue fragments can be automatically defined without making redundant designs. We have developed a set of recovery strategies appropriate for SDM workflows. The implementation of these strategies is tightly coupled with the operation of generic and non-generic actors (i.e. error tokens, error types that they provide). The first version of the framework has been presented in the related publications. In addition to the framework, we have developed a set of failure patterns in relation with SDM workflows. The failure patterns and the corresponding automated recovery techniques.

We have developed a new dataflow optimization technique for scientific workflows called "X-CSR" (for XML-Cut-Ship-Reassemble) which minimizes data transfer between different workflow stages. The optimization is based on a static analysis of the data structure that is flowing through the processing pipeline and the actor configurations (read scopes), which define the relevant part of the stream for any particular actor. The core idea of the optimization is to "cut" the data stream into fragments such that relevant fragments can by-pass intermediate actors which do not need those fragments, and instead forwarded them to the downstream actors that consume the stream fragments (Figure 3.1). We have also developed a new approach for exploiting data parallelism in XML processing pipelines through novel compilation strategies to the MapReduce framework [12]. Pipelines in our approach consist of sequences of processing steps that receive XML-structured data and produce, often through calls to "black-box" (scientific) functions, modified (i.e., updated) XML structures. Our evaluation uses the Hadoop MapReduce system as an implementation platform and shows that execution times of XML workflow pipelines can be significantly reduced using our compilation strategies. These efficiency gains, together with the benefits of MapReduce (e.g., fault tolerance) make our approach ideal for executing large-scale, compute-intensive XML-based scientific workflows. We also developed a new data aggregation actor which can evaluate window-based aggregation queries on data streams (Figure 3.2).

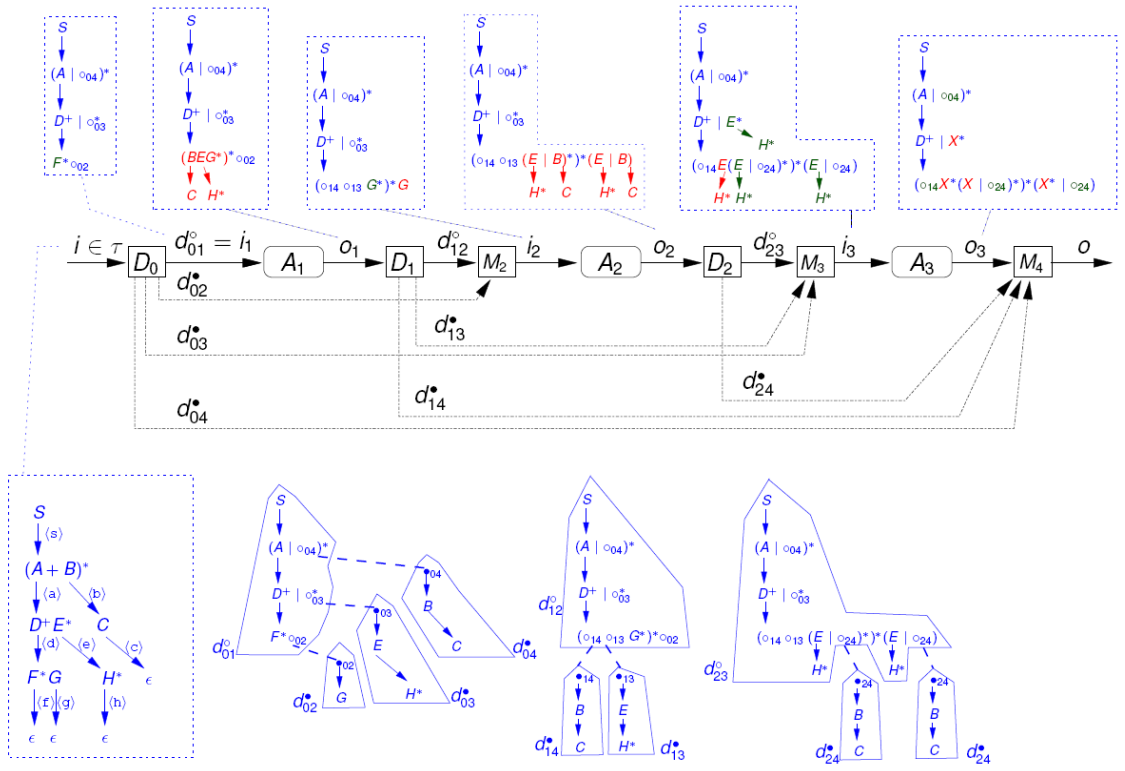

Figure 3.1: X-CSR: XML Cut-Ship-Reassemble Dataflow Optimization.

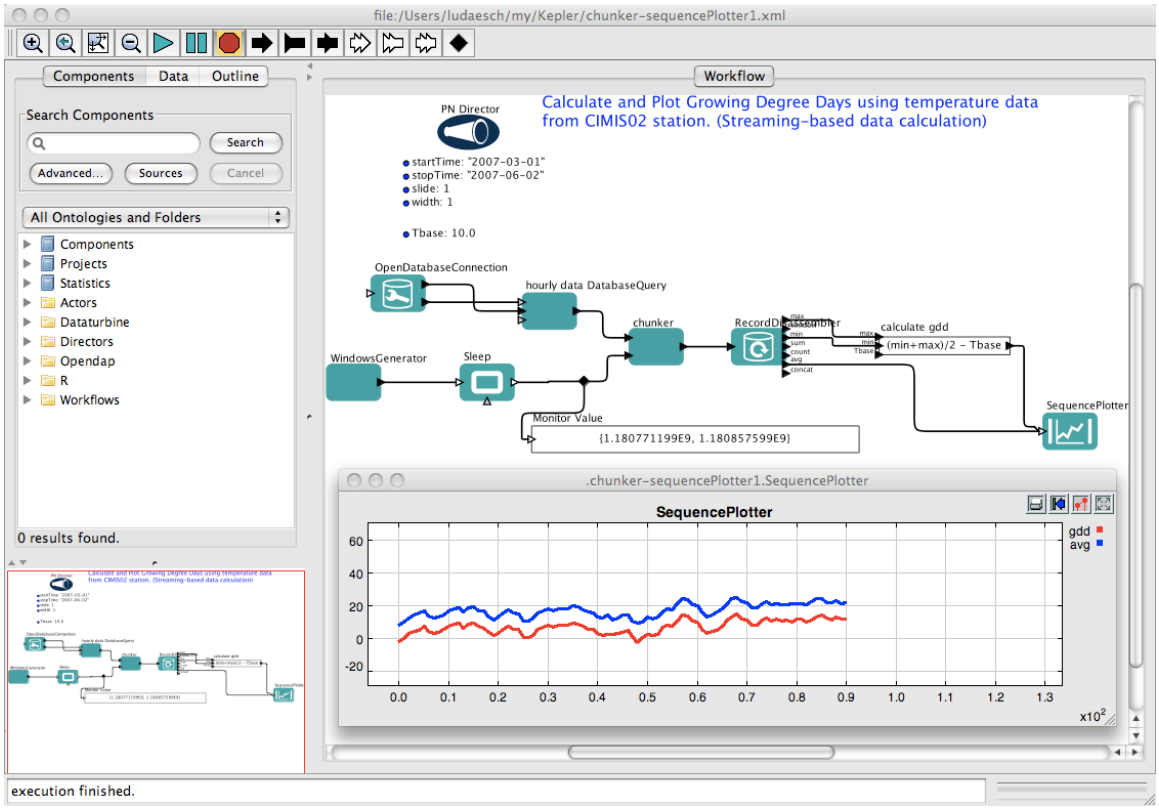

Figure 3.2: Stream-based Data Aggregation

We have also continued to develop and deploy generic actors, which provide general capabilities across underlying representations. These generic actors allow the development and deployment of workflows that effectively run on a variety of platforms – dramatically simplifying the use of workflows to support science. In addition to the previously deployed generic remote execution actor and job submission actor we have now released a generic file transfer actor. This actor utilizes available protocols, including FTP, SCP and SRM-Lite, to reliably transfer large numbers of files between machines and is capable of pattern matching, copying directories, and performing recursive transfers. This new actor was motivated by groundwater workflow developers who needed to transfer multiple files and folders between machines, while identifying the appropriate files and folders using regular expressions. Since each of the underlying protocols have different constraints on regular expressions, this generic actor masks these differences and provides a uniform interface to the workflow designer – allowing portable workflow development.

3.3 Dashboard development

The emergence of leadership class computing is creating a tsunami of data from petascale simulations. Results are typically analyzed by dozens of scientists. In order for the scientist to digest the vast amount of data being produced from the simulations and auxiliary programs, it is critical to automate the effort to manage, analyze, visualize, and share this data. One aspect of automation is to provide an easy-to-use web-based mechanism to monitor the progress of simulations, and view and compare the results generated with the use of the workflow system. A second aspect is to leverage the collective knowledge and experiences of the scientists collaborating on a project through a scientific social network. This can be achieved through a combination of parallel back-end services, provenance capturing and analysis, and an easy-to-use front-end tool. The SDM Center Dashboard is one such tool [e.g., CRI09, BCK+07, BKM09].

Progress to Date

Work on the eSiMon dashboard development is ongoing, as the dashboard team is continuing to make improvements for production use. The first release of eSiMon is set for early December, 2010. The alpha release of eSiMon is currently available at http://www.olcf.ornl.gov/center-projects/esimon/. A major part of our work includes the hardening of this version for a broader public release. The new version includes analysis API's which will can integrate into the simulations, and talk to eSiMon through a login-node. We have also developed a GUI to upload and run Matlab scripts, and this uses the SGI infrastructure at ORNL. Additional effort has been into research ideas for creating a new UI for an enhanced user experience.

Until recently we were limited to raw images, and movies generated from raw images. Last year we created a tool for visualizing contour trees. While an improvement, this is very specific solution and a more general solution is required. The solution has been to add VisIt support to the dashboard. By allowing the user run small VisIt sessions in the background on already processed data, the user has a world of options opened up to them. Anything from slight modifications to the data to 3D visualizations can be done in VisIt. With VisIt support comes the question of how to wrap VisIt functionality. It is impossible to support all Visit functionality within the dashboard, so we must determine the appropriate subset. The initial approach was to allow access to the Visit command line interface through an XML file that an expert would generate. This, of course, is less than optimal as it requires an expert to build a custom file for each session. We also created a method to compare session files and generate VisIt's GUI functionality in a semi-automatic fashion. In response to requests regarding the VisIt interface, we've also worked to improve the experience by adding features such as sockets, instant refreshing, and more comprehensive widgets.

Currently the dashboard has the ability to run VisTrails workflow medleys. Maintenance for this code has been passed to a new developer who has fixed a number of bugs, ensuring support for medleys in the future.

We are now at the state where a portable version of the dashboard will be offered for installation to users at other DOE sites. The ORNL eSiMon dashboard is relatively tightly coupled to the ORNL environment. It can only be used there (through the network), and a number of items in it are "hardwired" to that environment. To enable portability and transfer of the dashboard technology to other DOE users, we have developed two versions of "portable" dashboards. One is a portable ORNL dashboard that can be taken to places where there is no network – for example, airplanes, by installing it on a laptop and by downloading some of the relevant data to the local database [MOU08]. The other is a "distributable" dashboard – a one-click installation package that can be installed in any LAMP environment and with appropriate configuration used at sites away from ORNL and even with workflow engines other than Kepler [Nag09]. We have worked on a Linux installer for the dashboard. Installation and instructions for use are at (http://www.olcf.ornl.gov/center-projects/esimon).

3.4 Provenance collection and analysis

In scientific applications, effectively managing data provenance is extremely important [e.g. CFS+05] Provenance is at the heart of almost all functionalities, capabilities, and productivity improvements offered by workflow solutions. We distinguish system, workflow, process and data provenance categories. In the first category we collect data about meta-data related to preparation, run-time and post-processing environments – things like which compiler was used to make the run-time simulation code, possibly pre-processing testing activities, run-time machine characteristics, etc. Workflow provenances is concerned with workflow preparation, history, templating, and so on, while process provenance offers information about the order and success of the processes executed on both workflow control plane – Kepler, and at sites where remote resource are being used. Data provenances is often considered most important, in our case a lot of it comes from the Kepler provenance recorder, but other sources are used as well. In the case of Kepler, data provenance can be thought of as the complete processing history (transformations, types, etc.) of a data product, for example, actor identification and invocation parameters (or application codes launched by those actors), properties such as the time, location, and userid of invocation, relevant environment and configuration parameters. This information needs to be persistent and permanently associated with a data product so that its provenance is readily available. It also needs to be searchable, so that data with certain provenance can be easily identified – for example, if a bug is identified and corrected, the provenance can help identify which runs should be repeated, or at run-time or post-processing phases, we need quick access to certain information.

Progress to Date

By consolidating provenance information or a variety of applications, we can provide a uniform environment for querying, sharing, and re-using provenance in large-scale, collaborative settings. In this context, important sources of provenance information are different application, system and other logs. For example, logs can be used to understand issues that may arise, collect information that otherwise would not be available, and in general enrich the meta-data about all provenance aspects.

The Kepler provenance architecture has been developed and deployed by the team, as reported previously. However, to ensure its continued applicability to the evolving Kepler environment, we have implemented schema upgrade utilities along with updates and bug fixes to support release of Kepler 2.1, reporting and workflow run manager modules. In collaboration with Kepler/CORE developers, we have fixed memory leaks in Kepler GUI, including the fixes required the provenance module, and implemented a provenance database migration tool.

In addition, we have developed a provenance browsing and querying tool for Kepler (Figure 3.3) that allows a user to navigate provenance graphs and inspect data lineage dependencies.

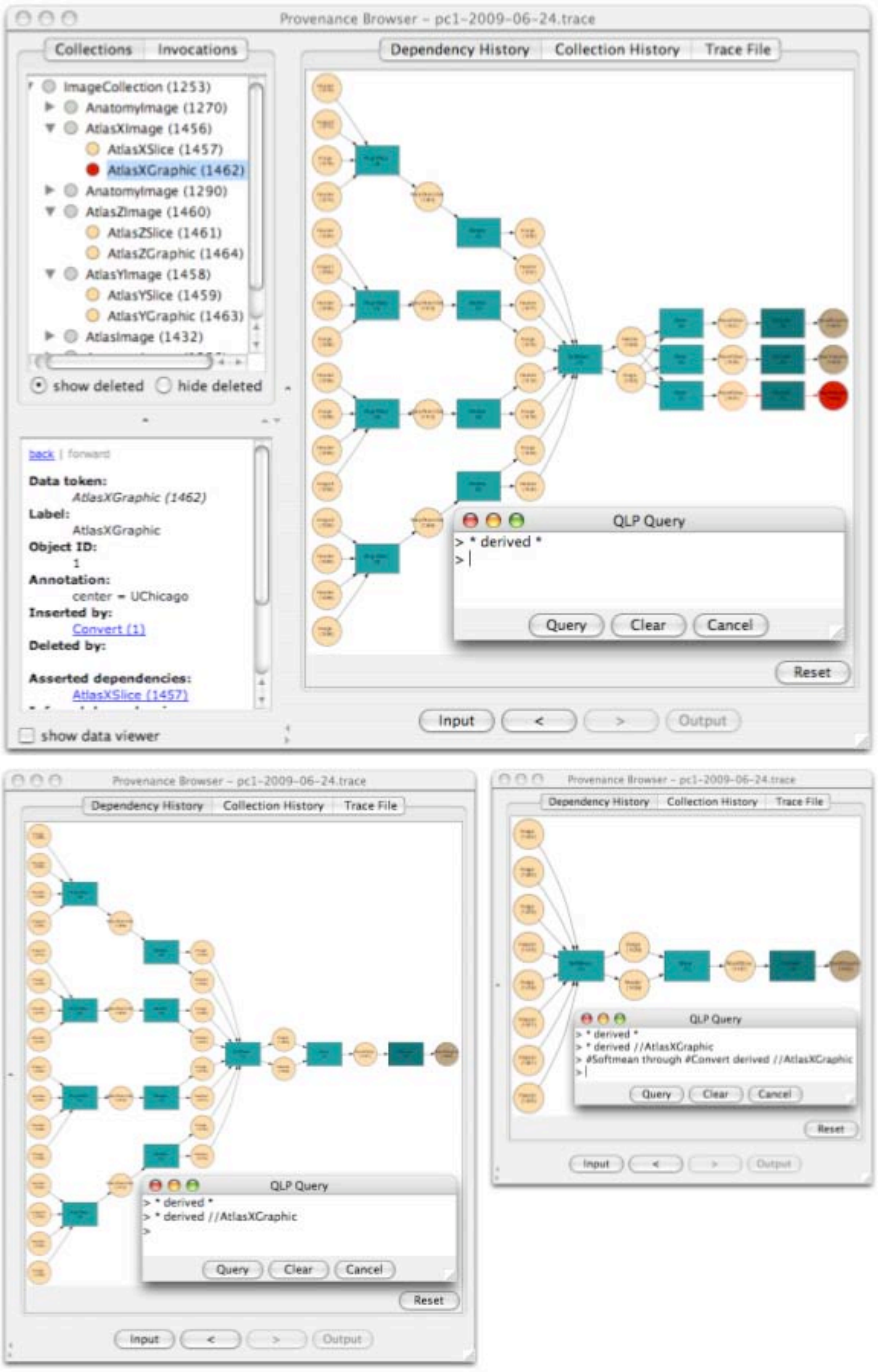

Figure 3.3: Provenance Browser and Query Facility.

3.5 Workflow reliability and fault tolerance

One of problems associated with scientific workflows is the fault-tolerant workflow modeling. There are two basic forms of run-time fault-tolerance: forward-recovery (e.g. failure masking and redundancy based failover), and backward-recovery (e.g. check-pointing) [LB90, MV06, DKV97, VOU05, VAB+07]. Exception handling is a very traditional way of managing run-time problems [MV96, VOU05]. It is also

used in the workflow-oriented environments [HA96, CCPP99]. Exception handling can involve forward-recovery, backward-recovery, or graceful termination. More recently the web services community has recognized the need for some form of standardized fault-tolerance in the service provisioning through replication [SPP+06]. An important component is collection of sufficient amount of meta-data (provenance information) about the workflow, processes, data, and environments to enable fault-tolerance actions.

At this stage of our work, it is the provenance information that is being collected through our provenance recorder that is at the heart of the fault and failure management mechanisms we are implementing. It has the capability of providing meta-data needed to detect and locate workflow run-time issues that can be handled at the Kepler control layer.

Progress to Date

As previously reported, we have implemented a Kepler-based fault-tolerance (FT) framework that leverages provenance information it collects, and provides options for forward recovery as well as backward recovery of the workflows at control plane [MCA+10, YAC+10]. The FT framework addresses the majority of the known issues and faults in current production workflows and is composed of 3 major components (figure 2):

- An error Handling Layer, which monitors the components beyond the workflow engine's direct control, such as visualization services, and reacts accordingly when an error is detected
- A contingency Actor, which provides a recovery block mechanism within the workflow.
- Checkpointing and Smart resume capabilities for when the above 2 mechanisms fail from preventing the workflow from terminating abnormally.

The extension of the model beyond the workflow control plane allows us to catch and process problem signals from environments that Kepler has no direct control over (see Figure 3.4) [MCA+10, YAC+10] . The key concept in that context is operational profiles – the frequency of usage of different Kepler and other operations, and their relationship to run-time failures. Operational profiles are an essential part of software reliability engineering. Typically they are created from the software requirements, and through customer reviews. Creation of operational profiles often is laborious and requires human intervention. Our approach builds an operational profile based on the actual usage from execution logs. The difficulty in using execution logs is that the amount of data to be analyzed is extremely large (more than a million records per day in many applications). Our solution constructs operational profiles by identifying all the possible clustered sequences of events (patterns) that exist in the logs. This is done very efficiently using suffix arrays data structure [NWV09, NVW+08].

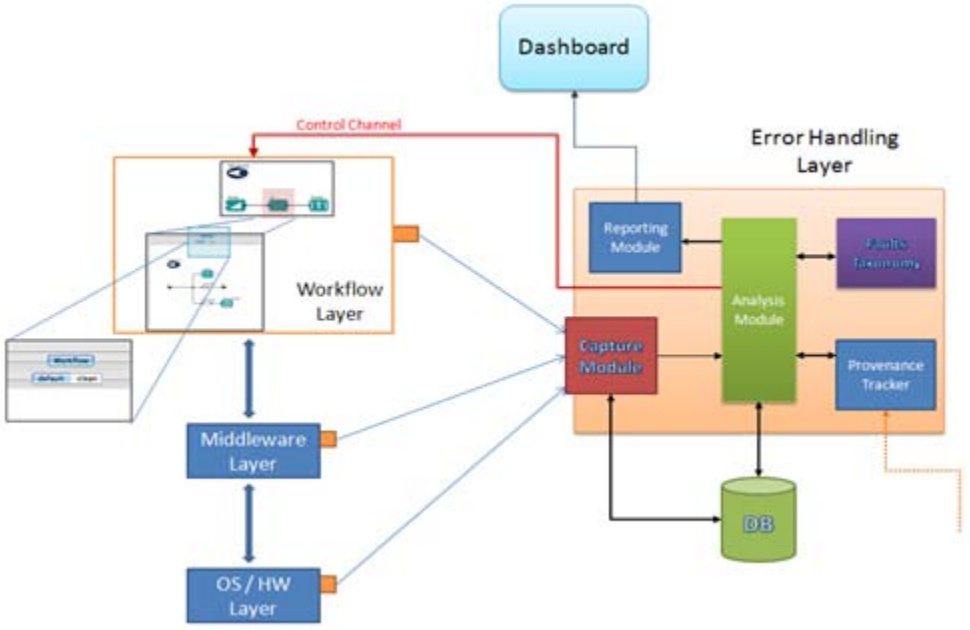

Figure 3.4 Error handling layer for SDMC scientific workflow support.

During this period, we have extended the Contingency actor to make it configurable as a finite state machine (FSM) as seen in Figure 3.5. In this new version of the Contingency actor, one tab contains a FSM, which specifies when to execute sub-workflows each of which are displayed in separate tab. We also implemented port conditions to monitor data transferred between actors (see Figure 3.6). The port condition checks for data matching user-configured expressions. When a matching condition is satisfied, an action is performed, e.g., change state in Contingency actor, pause or stop workflow execution. In collaboration with Pierre Mouallem and Mladen Vouk at NCSU, we worked on the design and initial implementation of more generalized workflow conditions and steering interfaces to allow external applications to control workflow execution. A set of fault-tolerance patterns for the SDM fault-tolerance framework has been developed.

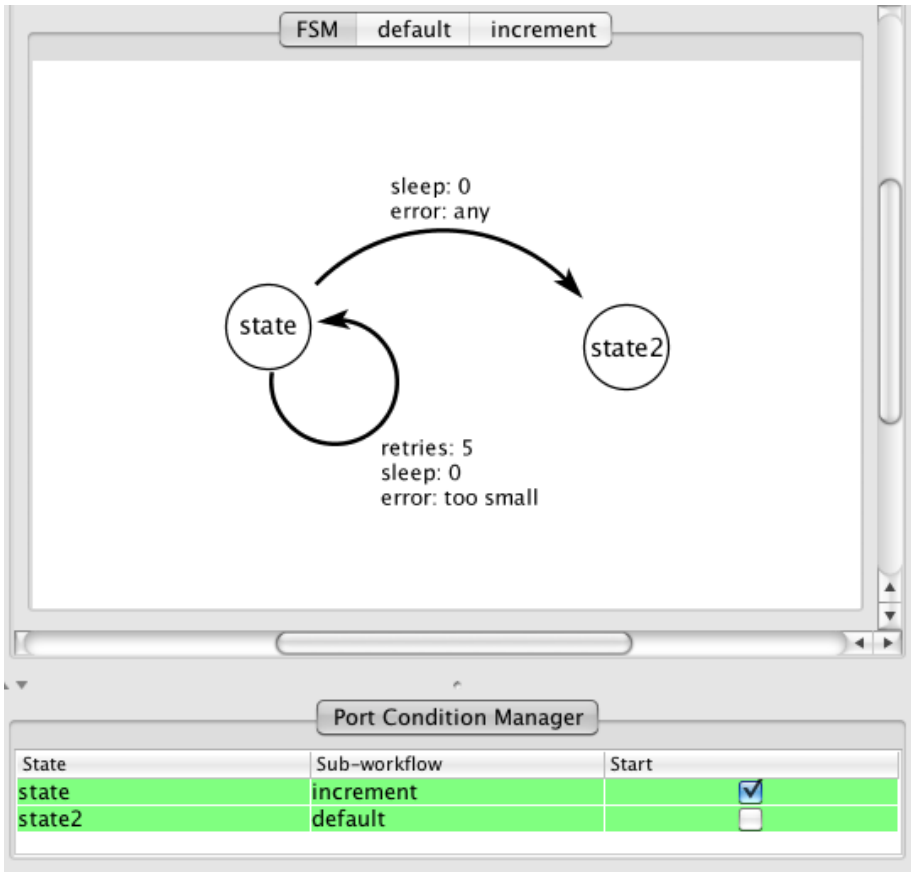

Figure 3.5: Finite State Machine tab of Contingency actor. Each state is associated with a sub-workflow that can be displayed in another tab.

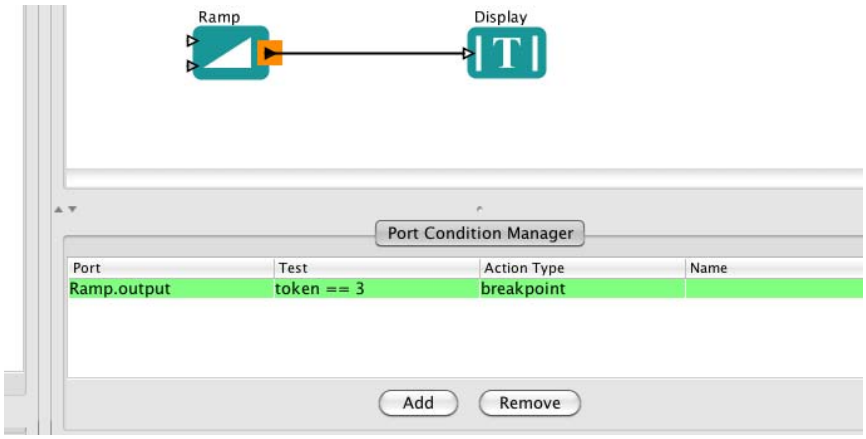

Figure 3.6: Port Condition checking output of Ramp actor.

3.6 Framework for Integrated SDM Technologies for Applications (FIESTA)

SDM center computer scientists actively collaborate with fusion scientist from the Center for Plasma Edge Simulations (CPES). The main theme of this collaborative effort is to provide enabling technologies for complex coupled simulations running from petascale computers. The technologies being developed by the SDM center for CPES are fully applicable to other projects as well, and in fact, the success of many of our techniques has led to adoption by some of the biggest data-producing codes in the DOE (*e.g.*, (*e.g.*,CHIMERA for astrophysics simulations and S3D for combustion simulations). Furthermore, there is

already a lot of interest from other FSP projects in the technologies developed within CPES, as well as collaboration with ITER simulation software developers in France.

Progress to Date

Our focus has been in innovative techniques using I/O pipelines in staging, for in situ analysis, re-organization, visualization, and I/O. At the heart of our research has been the ADIOS research platform, which allows researchers the ability to invent new methods into the I/O stack, and immediately use these in major simulations which run on the all of the major computer platforms. In fact, our codes are used in many major codes including: GTS, GTC, XGC-1, XGC-0, M3D-MPP, M3D-OMP, M3D-K, Pixie3D, GTC-P, Chimera, S3D, HFODD, PAMR; along with many new codes which scientist are just starting to use with ADIOS in the USA, Europe, and Asia. It is important to note that the ADIOS team includes both partners within the SDM Center, specifically ORNL, LBNL, and NCSU, but also external collaborators including Auburn, Georgia Tech, Sandia, and Rutgers.\

Overall research highlights by the ADIOS team include.

1. Using the staging area inside of ADIOS, researchers can now use a Service Oriented Architecture to couple codes. We have worked with the CPES SciDAC project, to couple the XGC-0, M3D-OMP, M3D-MPP code together. Using Kepler we then use the provenance tracking capability to track the provenance, and to move files that are output from the coupled codes to the Elite simulation, which runs on a different platform (first codes run on jaguar at OLCF, and the Elite code runs on ewok at the OLCF).
2. We have used codes embedded code, as part of the I/O stream, to create in situ workflows, which contain this embedded code, and we can execute this code on other nodes, during the simulation.
3. We have worked on scheduled movement of the I/O services, and we can reduce the impact of the I/O pipelines/ (disk activity) by paying careful attention to the data movement for I/O, and the data movement for local data movement inside of the HPC system.
4. We have worked on integrating file-based code coupling, alongside memory based code code, to work in concert with one-another. This has been demonstrated in several Fusion Simulation Project meetings.
5. We have coupled together more in situ visualization techniques inside the ADIOS staging area, thereby shielding the visualization code from the simulation code. Each code acts as a service, and uses the ADIOS I/O interface for the coupling mechanism.
6. We have demonstrated the value of an Application Log File format (ALF), in it's ability to place data on storage targets, thereby increasing concurrency for I/O patterns common for codes in reading.
7. We have worked closely with the S3D team to further help scale their code to 98K cores, having an I/O impact <1% to their calculation, using our sub-filing technique; including open, read, and close.
8. We have worked on new ADIOS methods which are capable of increasing the QoS for simulations, demonstrating this for "real-codes" such as XGC-1, allowing them to scale to over 240K cores.
9. We have worked closely with the DMA team, to increase the Statistics generated in ADIOS, automatically for the scientist. This includes such statistics as min/max, average, and standard deviation across each time step separately.
10. We have worked on optimal methods for the IBM BGP at the ALCF, using the GTC-P code, allowing their code to get "near-optimal" performance on the ALCF file system.

3.7 Dissemination and Outreach

Two key activities at SDMC SPA are dissemination of the results of its work through publications, presentations and participation in different forums, and transfer of its technology through outreach

activities such as tutorials and direct work with user groups. All publications and principal event (such as All Hands Meetings) meetings are on-line, along with other SDMC information

Over the last six months we have published or prepared for publication over 37 papers, reports and tutorials (see Publications subsection). We have also participated in numerous conferences, meetings, panels and other events, as outlined in the publications list.

- Provided the Fusion Simulation Project hands-on training. Slides for the training are available at http://users.nccs.gov/~sklasky/effis.pdf.
- ADIOS helped the S3D team run at scale on the Cray XT5 at ORNL. Their results generated over 60 TB, which the S3D is currently analyzing using the ADIOS-BP file format.
- Worked with the VisIt/VACET team to have the next release of VisIt reading in data from the ADIOS-BP file format.
- Made the ADIOS-BP file format to read efficiently the Chombo data file format. Working with the LBL team, we have benchmarked the Chombo code, and further integrated ADIOS into the Chombo framework.

SPA Software

- Distributable Dashboard (http://sdm7.csc.ncsu.edu/download_dashboard/)
- Portable ORNL Dashboard (https://ewok-web2.ccs.ornl.gov/portable/)
- Contributions to KEPLER (https://kepler-project.org/)

We also worked on extending our SDM outreach efforts in this period. We hosted Philippe Huynh (Nov. 2009) from the ITER group at SDSC/UCSD. Future planning for our collaboration with ITER was discussed further in the ITER meetings that co-PI Altintas has attended in June 2010. We are currently working on releasing the gridLite modules implemented by the Euforia project in the Kepler repository. In addition, we are working with the Fermilab Computing Division to apply the SDM technologies on distributed computing to process telescopic images coming from the next space-based dark energy mission.

October 2009 – October 2010 Publications by SDM center (65)

[AAC+10] Ilkay Altintas, Manish K. Anand, Daniel Crawl, Adam Belloum, Paolo Missier, Carole A. Goble, Peter M.A. Sloot. Understanding Collaborative Studies Through Interoperable Workflow Provenance. The third International Provenance and Annotation Workshop (IPAW2010). Submitted.

[ABA+10] Manish Kumar Anand, Shawn Bowers, Ilkay Altintas, Bertram Ludaescher. Approaches for Exploring and Querying Scientific Workflow Provenance Graphs. The third International Provenance and Annotation Workshop (IPAW2010). Submitted.

[ABL09] Manish Kumar Anand, Shawn Bowers, Bertram Ludäscher, A navigation model for exploring scientific workflow provenance graphs. SC-WORKS 2009

[ABL10] Manish Kumar Anand, Shawn Bowers, Bertram Ludäscher: Techniques for efficiently querying scientific workflow provenance graphs. EDBT 2010: 287-298

[ABML09a] M. Anand, S. Bowers, T. McPhillips, and B. Ludaescher, Exploring Scientific Workflow Provenance Using Hybrid Queries over Nested Data and Lineage Graphs, In 21st Intl. Conf. on Scientific and Statistical Database Management (SSDBM), New Orleans, 2009.

[ABML09b] Manish Kumar Anand, Shawn Bowers, Timothy M. McPhillips, Bertram Ludäscher: Efficient provenance storage over nested data collections. EDBT 2009: 958-969

[AKS10] H. Abbasi, S. Klasky, K. Schwan, M. Wolf, "Extracting Information ASAP!", PDSI 2010.

[BH+09] Paul Breimyer, William Hendrix, Guruprasad Kora, Nagiza F. Samatova, "pR: Lightweight, Easy-to-Use Middleware to Plugin Parallel Analytical Computing with R," IKE 2009: 667-673.

[BK+09] Paul Breimyer, Guruprasad Kora, William Hendrix, Neil Shah, Nagiza F. Samatova, "pR: Automatic parallelization of data-parallel statistical computing codes for R in hybrid multi-node and multi-core environments," IADIS AC (2) 2009: 28-32.

[CKP+10] Cummings, Klasky, Podhorszki, Barreto, Lofstead, Schwan, Docan, Parashar, Sim, Shoshani, "EFFIS: and End-to-end Framework for Fusion Integrated Simulation", PDP 2010, http://www.pdp2010.org/.

[CPP+08] J. Cummings, A. Pankin, N. Podhorszki, G. Park, S. Ku, R. Barreto, S. Klasky, C. S. Chang, H. Strauss, L. Sugiyama, P. Snyder, D. Pearlstein, B. Ludaescher, G. Bateman, A. Kritz, and the CPES Team, Plasma Edge Kinetic-MHD Modeling in Tokamaks Using Kepler Workflow for Code Coupling, Data Management and Visualization, Communications in Computational Physics, 4(3), September 2008.

[Cri09] T. Critchlow. "Scientific Process Automation Improves Data Interaction", Invited Cover Article for Scientific Computing. September 2009.

[DCK+10] C. Docan, J. Cummings , S. Klasky , M. Parashar, N. Podhorszki, F. Zhang, "Experiments with Memory-to-Memory Coupling for End-to-End fusion Simulation Workflows", ccGrid2010, IEEE Computer Society Press 2010.

[DCKP11] C. Docan, J. Cummings, S. Klasky, M. Parashar, "'Moving the Code to the Data - Dynamic Code Deployment using ActiveSpaces", submitted to IPDPS 2011.

[DPK10] C. Docan, M. Parashar and S. Klasky, DataSpaces: An Interaction and Coordination Framework for Coupled Simulation Workflows, Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 2010), Chicago, Illinois, USA June, 2010.

[DVA+10] P. Dreher, M. Vouk, S. Averitt, E. Sills, "An Open Source Option for Cloud Computing in Education and Research," 2010, to appear.

[GBA+09] Antoon Goderis, Christopher Brooks, Ilkay Altintas, Edward A. Lee, Carole A. Goble: Heterogeneous composition of models of computation. Future Generation Comp. Syst. 25(5): 552-560 (2009).

[GLC+09] Kui Gao, Wei-keng Liao, Alok Choudhary, Robert Ross, and Robert Latham. Combining I/O Operations for Multiple Array Variables in Parallel NetCDF. In the Proceedings of the Workshop on Interfaces and Architectures for Scientific Data Storage, held in conjunction with the IEEE Cluster Conference, New Orleans, Louisiana, September 2009.

[IBC+09] Florin Isaila, Francisco Javier Garcia Blas, Jesus Carretero, Wei-keng Liao, and Alok Choudhary. A Scalable Message Passing Interface Implementation of an Ad-Hoc Parallel I/O System. In the International Journal of High Performance Computing Applications, October 5, 2009.

[IVY10] Harini Iyer, Mladen Vouk and Ustun Yildiz, Automation of Scientific Workflow Construction Using Templates & Patterns, In 1st International Workshop on. Workflow Approaches to New Data-centric Science. held in conjunction with SIGMOD'2010 Submitted.

[Kam10a] C. Kamath, "Understanding wind ramp events through analysis of historical data," IEEE PES Transmission and Distribution Conference, New Orleans, April 2010.

[Kam10b] C. Kamath, "Using simple statistical analysis of historical data to understand wind ramp events," LLNL Technical report LLNL-TR-423242, February 2010.

[KGH+09] Wesley Kendall, Markus Glatter, Jian Huang, Thomas Peterka, Robert Latham, and Robert Ross. Terascale data organization for discovering multivariate climatic trends. In Proceedings of Supercomputing, November 2009.

[KLJ+10] S. Klasky, Qing Liu, Jay Lofstead, Norbert Podhorszki, Hasan Abbasi , CS Chang , Julian Cummings, Divya Dinakar, Ciprian Docan, Stephane Ethier , Ray Grout , Todd Kordenbrock, Zhihong Lin , Xiaosong Ma , Ron Oldfield, Manish Parashar, Alexander Romosan , Nagiza Samatova , Karsten Schwan, Arie Shoshani , Yuan Tian, Matthew Wolf, Weikuan Yu , Fan, Zhang , Fang Zheng, "ADIOS: powering I/O to extreme scale computing", SciDAC 2010.

[KXL10] C. Kamath, Y. Xiao, and Z. Lin, "Analysis of structures and event size statistics in plasma turbulence: Preliminary results," Sherwood Fusion Theory Conference, Seattle, April 2010.

[LAB+09] B. Ludäscher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure, J. Freire, C. Goble, M. Jones, S. Klasky, T. McPhillips, N. Podhorszki, C. Silva, I.

Taylor, and M. Vouk. In A. Shoshani and D. Rotem, editors Scientific Process Automation and Workflow Management, Scientific Data Management: Challenges, Existing Technology, and Deployment, Computational Science Series, chapter 13. Chapman & Hall/CRC, 2009.

[Latham10] Robert Latham, "Parallel netCDF," in the Encyclopedia of Parallel Computing, David Padua, editor, Springer, 2010 (expected).

[LBM09] B. Ludaescher, S. Bowers, and T. McPhillips. M. T. Özsu and L. Liu, editors, Scientific Workflows, Encyclopedia of Database Systems. Springer, 2009

[LCL+09] Samuel Lang, Philip Carns, Robert Latham, Robert Ross, Kevin Harms, and William Allcock, "I/O Performance Challenges at Leadership Scale," Proceedings of Supercomputing, November 2009.

[LPG+11] J. Lofstead, M. Polte, G. Gibson, S. Klasky, R. Oldfield, J. Bent, A. Manzanares, Q. Liu, N. Podhorszki, M. Wingate, M. Wolf, "Data Districts: Data Organization for High End-to-End Performance of Extreme Scale I/O", submitted to FAST 2011.

[LWM+09] Scientific Workflows: Business as Usual? , B. Ludaescher, M. Weske, T. McPhillips, S. Bowers. In 7th Intl. Conf. on Business Process Management (BPM), Ulm, Germany, 2009.

[LZZ+10] J. Lofstead, F. Zheng, Q. Liu, S. Klasky, R. Oldfield, T. Kordenbrock, K. Schwan, M. Wolf, "Managing Variability in the IO Performance of Petascale Storage Sytems", accepted ACM/IEEE SC 2010 Conference (SC'10), 2010.

[MBZL09] Timothy M. McPhillips, Shawn Bowers, Daniel Zinn, Bertram Ludäscher: Scientific workflow design for mere mortals. Future Generation Comp. Syst. 25(5): 541-551 (2009)

[MCA+10] Pierre Mouallem, Daniel Crawl, Ilkay Altintas, Mladen Vouk and Ustun Yildiz, A Fault-Tolerance Architecture for Kepler-based Distributed Scientific Workflows, In 22nd International Conference on Scientific and Statistical Database Management, (SSDBM'2010).

[Nag10] M.Nagappan, "Analysis of Execution Log Files". Accepted at the Doctoral Symposium track of ICSE 2010, May 4th Cape Town SA

[NJC+10] A. Ngu, A. Jamnagarwala, G. Chin Jr. , C. Sivaramakrishnan, T. Critchlow, "Context-Aware Scientific Workflow Systems Using KEPLER", To appear in the International Journal of Business Process Integration. 2010.

[NV10a] M. Nagappan, M.A. Vouk, "Abstracting Log Lines to Log Event Types for Mining Software System Logs." Accepted as short paper in the 7th IEEE Working Conference on Mining Software Repositories (MSR), 2-3, May, 2010, Cape Town, South Africa.

[NV10b] M. Nagappan, M.A. Vouk, "Adaptive Logging: A Case Study of Logs from a Cloud Computing Environment". Submitted to 9th IEEE International Symposium on Network Computing and Applications (IEEE NCA10), 15-17th July, Cambridg,e MA, USA.

[NWV09] M. Nagappan, K. Wu, M.A. Vouk,. 2009. "Efficiently Extracting Operational Profiles from Execution Logs using Suffix Arrays." 20th International Symposium on Software Reliability Engineering, 16-19 Nov, 2009, Mysuru, India. pp. 41 - 50.

[RCG+09] Robert Ross, Alok Choudhary, Garth Gibson, and Wei-Keng Liao. Parallel data storage and access. In Arie Shoshani and Doron Rotem, editors, Scientific Data Management: Challenges, Technology, and Deployment. Chapman & Hall/CRC, 2009.

[RCM09] Robert Ross, Philip Carns, and David Metheney. Parallel file systems. In Yupo Chan, John Talburt, and Terry Talley, editors, Data Engineering: Mining, Information and Intelligence. Springer, October 2009.

[RGC+09] Oliver Rübel, Cameron G R Geddes, Estelle Cormier-Michel, Kesheng Wu, Prabhat, Gunther H Weber, Daniela M Ushizima, Peter Messmer, Hans Hagen, Bernd Hamann, Wes Bethel, Automatic beam path analysis of laser wakefield particle acceleration data. 2009 Comput. Sci. Disc.

[Ross10] Robert Ross, "Parallel File Systems," in the Encyclopedia of Parallel Computing, David Padua, editor, Springer, 2010 (expected).

[SHT+10] E Stephan, T Halter, T Critchlow, P Pinheiro Da Silva, and L Salayandia. "Using Domain Requirements to Achieve Science-Oriented Provenance ." Short paper in The 3rd International Provenance and Annotation Workshop (IPAW'2010). June 2010.

[SKR10] A. Shoshani, S. Klasky, R. Ross, "Scientific Data Management Challenges and Approaches in the Extreme Scale Era", SciDAC 2010.

[SR09] A. Shoshani and D. Rotem (Editors), Scientific Data Management: Challenges, Technology, and Deployment, Chapman & Hall/CRC Computational Science Series, December 2009.

[TKL+11] Y. Tian, S. Klasky, J. Lofstead, R. Grout, N. Podhorszki, Q. Liu, Y. Wang, W. Yu, "Data reordering Using Hilbert Space Filling Curve to Improve the Read Performance for Scientific Applications", submitted to IPDPS 2011.

[TKP+10] R. Tchoua, S. Klasky, N. Podhorszki, B. Grimm, A. Khan, E. Santos, C.T. Silva, P. Mouallem, M. Vouk. "Collaborative Monitoring and Analysis for Simulation Scientists", In Proceedings of The 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010), 2010

[VSD10] M.A. Vouk, E. Sills, P. Dreher, "Integration of High-Performance Computing into Cloud Computing Services, Handbook of Cloud Computing, Ed. B. Furht, to appear, 2010.

[WCA+09] Jianwu Wang, Daniel Crawl, Ilkay Altintas. Kepler + Hadoop – A General Architecture Facilitating Data-Intensive Applications in Scientific Workflow Systems. In Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science (WORKS09) at Supercomputing 2009 (SC2009) Conference. ACM 2009, ISBN 978-1-60558-717-2.

[WKK+10] Jianwu Wang, Prakashan Korambath, Seonah Kim, Scott Johnson, Kejian Jin, Daniel Crawl, Ilkay Altintas, Shava Smallen, Bill Labate, Kendall N. Houk. Theoretical Enzyme Design Using the Kepler Scientific Workflows on the Grid. Accepted by 5th Workshop on Computational Chemistry and Its Applications (5th CCA) at International Conference on Computational Science (ICCS 2010).

[WSS10] K. Wu, A. Shoshani, and K. Stockinger. Analyses of Multi-Level and Multi-Component Compressed Bitmap Indexes. ACM TODS v35, Article 2, 2010

[XHZ+10] Y. Xiao, I. Holod, W. L. Zhang, S. Klasky, Z. H. Lin, “Fluctuation characteristics and transport properties of collisionless trapped electron mode turbulence”, Physics of Plasmas, 17, 2010.

[YAC+10] Ustun Yildiz, Ilkay Altintas, Daniel Crawl, Pierre Mouallem and Mladen Vouk, “Fault-Tolerance in Dataflow-based Scientific Workflow Management”, Submitted to SWF 2010

[YGC09c] U. Yildiz, A. Guabtni, and A. H. H. Ngu, Business versus Scientific Workflow: A Comparative Study, 2009. Research Report, Department of Computer Science, University of California, Davis, CSE-2009-3, http://www.cs.ucdavis.edu/research/tech-reports/2009/CSE-2009-3.pdf.

[YGN09a] U. Yildiz, A. Guabtni, and A. H. H. Ngu, Towards scientific workflow patterns," in Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, In conjunction with Super Computing, SC (USA), ACM Press, 2009.

[YGN09b] U. Yildiz, A. Guabtni, and A. H. H. Ngu, Business versus scientific workflows: A comparative study," in Proceedings of the IEEE Third International Workshop on Scientific Workflows, SWF (In conjunction with 7th IEEE International Conference on Web Services (ICWS 2009)), (USA), IEEE Computer Society Press, 2009.

[YPM_10] Ustun Yildiz, Pierre Mouallem, Mladen Vouk, Daniel Crawl and Ilkay Altintas, Fault-Tolerance in Dataflow-based Scientific Workflow Management, In 4th IEEE International Workshop on Scientific Workflows (SWF@ICWS 2010). Submitted.

[YTV10] W. Yu, Y. Tian, and J. S. Vetter, Efficient Zero-Copy Noncontiguous I/O for Globus on InfiniBand, Proceedings of the Third International Workshop on Parallel Programming Models and Systems Software for High-end Computing. Held in Conjunction with ICPP10, San Diego, CA, 2010.

[YV10] Weikuan Yu and Jeffrey Vetter, “Initial Characterization of Parallel NFS Implementations,” the Sixth International Workshop on System Management Techniques, Processes, and Services (SMTPS10), Atlanta, GA, April 2010.

[YWX+10] Weikuan Yu, Kesheng Wu, Cong Xu, Arie Shoshani, Wei-Shinn Ku, BMF: Bitmapped Mass Fingerprinting for Fast Protein Identification, Auburn University Technical Report AU-CSSE-PASL/10-TR0, 2010. Available at http://pasl.eng.auburn.edu/pubs/pasl-2010-11-bmp.pdf

[ZAD+10] F. Zheng, H. Abbasi, C. Docan, J. Lofstead, Q. Liu, S. Klasky, M. Parashar, N. Podhorszki, K. Schwan, M. Wolf, “PreDatA - Preparatory Data Analytics on Peta-Scale Machines”, IPDPS 2010, IEEE Computer Society Press 2010.

[ZBML09] Daniel Zinn, Shawn Bowers, Timothy M. McPhillips, Bertram Ludäscher: X-CSR: Dataflow Optimization for Distributed XML Process Pipelines. ICDE 2009: 577-580

[ZSML09b] Daniel Zinn, Shawn Bowers, Timothy M. McPhillips, Bertram Ludäscher: Scientific workflow design with data assembly lines. SC-WORKS 2009

[ZSSL10] Zinn Daniel, Shawn Bowers, Sven Köhler, Bertram Ludäscher: Parallelizing XML data-streaming workflows via MapReduce. J. Comput. Syst. Sci. 76(6): 447-463 (2010)