PETASCALE DATA STORAGE INSTITUTE

Peter Honeyman, Co-Principal Investigator

University of Michigan, Center for Information Technology Integration

535 W. William St., Suite 3100, Ann Arbor, MI 48103-4978

734-763-4413 (-2929 Karen Kitchen, assistant), honey@citi.umich.edu

DOE/Office of Science Program Office: Advanced Scientific Computing Research (ASCR)

DOE/Office of Science Program Office Technical Contact:
Anil Deane, Thomas Ndousse, Fred Johnson, Gary Johnson, Mary Anne Scott, Yukiko Sekine

DOE Grant Number: N/A

Collaborating Institutions/PI:

- Carnegie Mellon University (Institute lead), Garth Gibson (Institute PI)
- University of California at Santa Cruz, Darrell Long
- University of Michigan at Ann Arbor, Peter Honeyman
- Los Alamos National Laboratory, Gary Grider
- National Energy Research Scientific Computing Center, William Kramer
- Oak Ridge National Laboratory, Philip Roth
- Pacific Northwest National Laboratory, Evan Felix
- Sandia National Laboratory, Lee Ward

Abstract:

Petascale computing infrastructures for scientific discovery make petascale demands on information storage capacity, performance, concurrency, reliability, availability, and manageability. The last decade has shown that parallel file systems can barely keep pace with high performance computing along these dimensions; this poses a critical challenge when petascale requirements are considered. This proposal describes a Petascale Data Storage Institute that focuses on the data storage problems found in petascale scientific computing environments, with special attention to community issues such as interoperability, community buy-in, and shared tools. Leveraging experience in applications and diverse file and storage systems expertise of its members, the institute allows a group of researchers to collaborate extensively on developing requirements, standards, algorithms, and development and performance tools. Mechanisms for petascale storage and results are made available to the petascale computing community. The institute holds periodic workshops and develops educational materials on petascale data storage for science.

The Petascale Data Storage Institute is a collaboration between researchers at Carnegie Mellon University, National Energy Research Scientific Computing Center, Pacific Northwest National Laboratory, Oak Ridge National Laboratory, Sandia National Laboratory, Los Alamos National Laboratory, University of Michigan, and the University of California at Santa Cruz.

I. Introduction

A. Executive Summary

The Petascale Data Storage Institute brings together high performance file and storage system expertise and experience meeting the high performance storage requirements of today's DOE terascale scientific discovery through advanced computing for the purpose of identifying, resolving and setting in motion solutions for the storage capacity, performance, concurrency, reliability, availability and manageability problems arising from petascale computing infrastructures for scientific discovery.

Led by Carnegie Mellon University, the Petascale Data Storage Institute membership also includes University of California at Santa Cruz, University of Michigan at Ann Arbor, Los Alamos National Laboratory, National Energy Research Scientific Computing Center, Oak Ridge National Laboratory, Pacific Northwest National Laboratory, and Sandia National Laboratory.

The Institute's work will be organized into six projects:

- Petascale Data Storage Outreach: (Type: Dissemination) Development and deployment of training materials, both tutorials for scientists and course materials for graduate students; support and advise other SciDAC projects and institutes; and development of frequent workshops drawing together experts in the field and petascale science users.
- Protocol/API Extensions for Petascale Science Requirements: (Type: Dissemination) Drive deployment of best practices for petascale data storage systems through development and standardization of application programmer interfaces and protocols, with specific emphasis on Linux APIs; validate and demonstrate these APIs in large scale scientific computing systems.
- Petascale Storage Application Performance Characterization: (Type: Data Collection) Capture, characterize, model and distribute workload, access trace, benchmark and usage data on terascale and projected petascale scientific applications, and develop and distribute related tools.
- Petascale Storage System Dependability Characterization: (Type: Data Collection) Capture, characterize, model and distribute failure, error log and usage data on terascale and projected petascale scientific systems, and develop and distribute related tools.
- Exploration of Novel Mechanisms for Emerging Petascale Science Requirements: (Type: Exploration) In anticipation of petascale challenges for data storage, explore novel mechanisms such as global/WAN high performance file systems based on NFS; security aspects for federated systems, collective operations, and ever higher performance systems; predictable sharing of high performance storage by heavy storage load applications; new namespace/search and attribute definition mechanisms for ever large namespaces; and integration and specialization of storage systems for server virtualization systems.
- Exploration of Automation for Petascale Storage System Administration: (Type: Exploration) In anticipation of petascale challenges for data storage, explore and develop more powerful instrumentation, visualization and diagnosis methodologies; data layout planning and access scheduling algorithms; and automation for tuning and healing configurations.

The Institute's budget is spread approximately evenly over the eight collaborating organizations and approximately evenly over the three project types. Each of the projects has a project lead coordinating with the overall PI to manage execution. Deliverables will be fully and freely available to all potential users. Long term support strategy is based on achieving deployment in open source packages, especially Linux distributions, and in commercial products.

B. Budget Summary

This proposal requests $12.5M over a five-year program for the Petascale Data Storage Institute. These monies would be spread over FY07 through FY11 with $2.5M distributed among the contributing organizations each year. The eight collaborating universities and laboratories are requesting approximately $300,000 annually, with the exception of the lead institution, Carnegie Mellon University, which is requesting approximately $400,000 annually to accommodate its larger leadership coordination role in addition to its full research and outreach roles. The Institute's work will be organized into six projects with the following approximate annual budgets: (1) Petascale Data Storage Outreach, $380K; (2) Petascale Storage Application Performance Characterization, $640K; (3) Petascale Storage System Dependability Characterization, $200K; (4) Protocol/API Extensions for Petascale Science Requirements, $420K; (5) Exploration of Novel Mechanisms for Emerging Petascale Science Requirements, $470K; and (6) Exploration of Automation for Petascale Storage System Administration, $390K.

C. Management Plan

The Petascale Data Storage Institute is a university-led distributed center involving multiple institutions, including universities and DOE National Laboratories. It is led by Carnegie Mellon University with Garth Gibson as its Principal Investigator with overall coordination responsibility. The Institute's work will be organized into six projects of three types, each with its own project leader:

- Project 1: Petascale Data Storage Outreach
  Leader: Garth Gibson, CMU; Type: Dissemination
- Project 4: Protocol/API Extensions for Petascale Science Requirements
  Leader: Gary Grider, LANL; Type: Dissemination
- Project 2: Petascale Storage Application Performance Characterization
  Leader: Bill Kramer, NERSC; Type: Data Collection
- Project 3: Petascale Storage System Dependability Characterization
  Leader: Gary Grider, LANL; Type: Data Collection
- Project 5: Exploration of Novel Mechanisms for Emerging Petascale Science Requirements
  Leader: Darrell Long, UCSC; Type: Exploration
- Project 6: Exploration of Automation for Petascale Storage System Administration
  Leader: Greg Ganger, CMU; Type: Exploration

Each project lead is responsible for the execution of that project, and coordinates with the overall PI to achieve overall execution of the Institute's objectives. Most projects will be further subdivided and delegates from each contributing organization will be identified and assigned specific tasks. Monthly conference calls will be used to manage the execution of the distributed team. Workshops offered for outreach will also be used to gather the Institute's staff for face-to-face reviews of the execution of the Institute's projects. The project numbering is not well aligned with the project types for historical reasons.

Source code for software deliverables will be fully and freely available for use and modification throughout the scientific computing community unless otherwise stated in a project statement of work. The open source license used may vary across deliverables, but will be one of the licenses approved and certified by the Open Source Initiative (http://www.opensource.org), such as the classic GPL, LGPL, BSD and MIT licenses.

Our strategy for long term support for many of our deliverables is to achieve adoption by the Linux community. While this cannot be guaranteed a priori, the Institute's projects seeking this strategy for long term support of software will use best efforts to accommodate the requests of the Linux Maintainers such that a code deliverable is acceptable by the Linux Maintainers for inclusion in a Linux distribution.

D. Background for Merit Review

As suggested in the SciDAC program notice, this section identifies where and how each merit review criterion is addressed.

1. Scientific and/or Technical Merit of the Project.

a) *Potential for significant impact on SciDAC applications.* I/O performance and data management challenge terascale systems; scaling to petascale systems requires new solutions and new approaches. Bridging research and practice is essential to making an impact on SciDAC applications. The Institute brings together leading experts in the design and implementation of large-scale storage systems in support of scientific applications.

b) *Demonstrated capabilities of the applicants.* The Institute reflects an exceptionally strong record of basic research and its transition to practice. We cite two examples. Object-based storage began as a research concept invented by Institute members and nurtured by Institute members, which led to standards, and then to products that are used worldwide to meet the data storage needs of scientific applications. Additionally, Institute members have made essential contributions to the open source community; software developed by institute members is core to Linux, Kerberos, the NSF Middleware Initiative, the Globus Toolkit, and many other open source platforms.

c) *Coupling with scientific simulation.* Institute membership spans National Labs staff who directly support massive scientific computations as well as researchers who work with scientific simulation themselves on a regular basis. (Project 2 discusses a number of the specific applications supported.)

d) *Impact on science disciplines outside of SciDAC applications.* The need to manage massive storage is a fact of life for much of large-scale science, and grows in importance as scale grows and knowledge accumulates. While the vanguard of the high-end science community requires scale before others, support for the data needs of petascale science opens the door for other science disciplines to exploit access to petascale computing resources.

e) *Approach to long-term support and transfer.* Outreach, a critical part of this Institute's plan, is the focus of Project 1. For software tool support, see the Management Plan.

f) *Broad community interaction.* Institute members span the state of the art in storage systems research and practice, featuring top National Lab staff and the leading academic storage systems research centers. All of the Institute members have enjoyed long and productive interactions with industry partners; the academic members are collaborating directly with (and supported in part by) a broad collection of the storage industry, including IBM, HP, Intel, Sun, Veritas, EMC, Microsoft, Network Appliance, Panasas, Seagate, PolyServe, and Engenio.

2. Appropriateness of the Proposed Method or Approach.

a) *Plan for coupling to emerging advances in enabling technology or to applications researchers.* Project 1 outreach activities will strengthen lines of communication between Institute members and petascale data stakeholders. Existing coupling between Institute members and the storage R&D community also facilitates these connections.

b) *Proposed work schedule and deliverables.* The Project Plan and Milestones section of this narrative lays out the schedule for work proposed in the six projects.

c) *Approach to intellectual property management and open source licensing.* The Institute intends to extend the frontier of massive-scale storage to achieve petascale science. Reference implementations, tools, and traces produced as part of the Institute's work will be shared via open source, as discussed in the Management Plan.

d) *Plan for effective collaboration among participants.* All of the proposed projects involve explicit and direct collaboration among the participants. Face to face meetings at Institute workshops will

supplement electronic communication in the development of common protocol/API extensions in Project 4. Many of the individual sub-projects in Projects 2, 3, 5, and 6 involve pair-wise or three-way collaborations in which tools developed or data gathered by one is shared with another. As examples, I/O traces gathered at a National Lab will be used for characterization and study at a University; automation tools developed at a University will be tested at a National Lab. The Management Plan above provides additional details.

e) *Plan for ensuring communication with other efforts.* The outreach effort of Project 1 is explicit about ensuring communication with other stakeholders exploring and supporting massive-scale science applications.

3. Competency of Applicant's Personnel and Adequacy of Proposed Resources.

The Institute brings together leading researchers from the top academic storage systems research centers and lead staff managing large-scale storage at National Labs. The CVs describe their extensive qualifications. The resources requested will complement existing resources to allow major progress in defining, exploring, solving, and disseminating knowledge in meeting the data storage needs of petascale computing.

4. Proposed Budget

See above.

II. Project 1: Petascale Data Storage Outreach

A. Summary

Scalable global parallel file systems were identified as critical for terascale scientific computing almost a decade ago [SGPFS99]. Coupling with the object storage research at CMU in the mid 90s (then called Network Attached Secure Disks [Gibson98]), a major thrust in novel storage architectures for increased scalability ensued [Lee96, Gibson97, Thekkath97, VanMeter98, Keeton98, Gibson99, Lustre02, Azagury03, Brandt03, Ghemawat03, Gibson03, Rodeh03, Saito04, Gibson04, Nagle04, Hildebrand05, ANSI04, Wang04]. The members of this Institute had a very large role in most of that major new thrust in storage technology, and can be expected to play major roles in getting from today's terascale solutions to the soon-to-be-needed petascale solutions. With this project, the Petascale Data Storage Institute asks these scalable storage system leaders to make themselves and their knowledge available to others in a variety of venues. Broadly stated, the goal of this project is to educate the community on best practices for efficient use of large scale storage systems, and specifically, to jump start community preparation for effective use of petascale systems.

B. Categories of Investigation

1. Workshops

To reach out and engage the scientific computing community in the emerging problems of petascale storage system performance, dependability, application programmer interfaces, novel problems and mechanisms and automation of management, this project will develop and chair an annual petascale storage workshop collocated with a scientific computing conference. The targeted conference is SCxy: the International Conference for High Performance Computing, Networking, Storage and Analysis (http://www.supercomp.org). This project will also engage the academic computer science community by targeting the USENIX Conference on File and Storage Technologies (FAST) (http://www.usenix.org/events/byname/fast.html), or, as appropriate, the IEEE Mass Storage Systems Technical conference (MSST) (http://storageconference.org).

Other workshops will be sought or accepted when and where appropriate. For example, ORNL plans to host a special topics workshop in year three of the Institute.

2. Tutorials

To communicate the techniques, mechanisms, programming practices and tools to the broader communities of scientific computing, academic computer science and industrial storage systems development, this project will develop and deploy multiple tutorials. Included in the scope of these tutorials will be advice to scientific discovery application developers on the strategies for maximizing the effectiveness of petascale storage access. Target venues for these tutorials include conferences such as SCxy, FAST, Storage Networking World, USENIX Annual Technical Conference, LISA, DSN, IEEE MSST and others.

3. Course work

The effective development of solutions for the petascale systems of the next decade depends on the development of the human resources that will be needed to design, operate and manage these systems. This project will create classroom materials covering the scope of the Institute and deploy them in at least the graduate programs at all three of the Institute's university members. Possible courses to be augmented include advanced operating and distributed systems, advanced storage systems, security systems, advanced scientific algorithms, and others.

4. Support for other SciDAC Activities

As a community of experts on storage systems for large scale applications, this project will make available resources to assist other SciDAC activities. For example, we have already been contacted by another SciDAC proposal, the Institute for Petascale Computing, to serve on their steering group and represent the need for new research in file and storage systems or application use of file and storage systems. Services such as these will be performed when possible and appropriate.

III. Project 4: Protocol/API extensions for Petascale Science Requirements

A. Summary

The increasing use of global and parallel file systems places a great burden on vendors and open source authors to support a diverse and rich collection of client platforms. Most of the tasks accomplished by any such file system are similar, though. An active collaborative research program into guiding efforts to generate common, well-accepted APIs and protocols is required for the health of the HPC storage industry. Additionally, as new concepts in storage and I/O mature beyond prototypes, existing APIs must be enhanced or new APIs need to be developed and validated with real science applications. Such research would strive to support the most important workloads while retaining an ability to be easily extended so that competition via unique features and capabilities is retained.

One of the strengths of this SciDAC Institute is in its extremely knowledgeable and influential member institutions and staff. The Institute will draw on the science applications experience and diverse file and storage systems experience at the national laboratory partners and the scalable storage research experience of its university partners. Many of these organizations are already collaborating to guide some important standards and API related activities. Additionally, many of the Institute's staff have a track record for successfully influencing industry accepted standards and APIs including technologies like RAID; IETF NFSv4, pNFS, iSCSI; ANSI/T10 OSD; and others. The establishment of this SciDAC Institute will enable much more extensive collaborative exploration into trade-offs which will provide the basis for guiding the development of standards and APIs for petascale storage. The Institute will also enable reference implementations of HPC storage standards and APIs and facilitate validation of these using real science applications. This powerful combination of knowledge, influence, reference implementations, and meaningful validation gives this Institute the unique capability to move HPC storage related standards and APIs forward within the standards communities and industry.

B. Categories of Investigation

1. POSIX performance and management extensions

It is widely agreed to within the HPC I/O community that the POSIX I/O API [UNIX] is "unnatural" for high-end computing applications. Opportunity abounds to make the POSIX I/O API more effective for HPC, high concurrence, and parallelism. The entire set of operations should be carefully and consistently reviewed, and enhancements suggested for high-end computing needs. Possible POSIX I/O API issues that will be investigated for enhancement include data and operation ordering; coherence management; missing capabilities like atomic vectored seek, read and write operations; operations performed by groups of processes; locking semantics for parallelism and high concurrence; attribute management including scalability; portability of underlying network topology hinting; portability of hinting about storage geometries, size, and types of upcoming I/O activity; and interfaces for storing, managing, and searching richer metadata than file systems currently manage [NITRD].

Dealing with concurrency and parallelism

Since science applications are parallel in nature and becoming more so as time goes on, the amount of parallel or concurrent processing against the I/O subsystem goes up. Perhaps the biggest weakness of the POSIX I/O-file system API is its lack of ability to deal with highly concurrent access from different client processes on different machines. Things like cache coherence management; attribute coherence management; group oriented metadata operations like locks, opens, closes; and something other than a stream of byte oriented access method such as distributed vector or matrix operations all need to be addressed.

Quality of Service

To ease data management for and improve the productivity of scientific application users, supercomputing facilities are implementing global parallel file systems. As global parallel file systems at these sites become more prevalent, where one common parallel file system is shared in a scalable way between multiple terascale or petascale computational platforms, the need for Quality of Service (QoS) becomes acute. Since parallel applications progress at the pace of the slowest participating process, multiple parallel applications simultaneously accessing a common file system could lead to massive non-deterministic behavior leading to horrible inefficiencies. Initial research into storage quality of service has been done [Uttamchandani04, Wu04]. If a QoS mechanism existed, there would need to be an interface for applications and computing platforms to request a particular QoS. The Institute will do collaborative research into if and which QoS extensions to the POSIX I/O API are needed.

Network topology

As storage area networks become more complex, parallel file systems need to be able to adapt to these complexities to avoid transfer bottlenecks. While the ultimate desire for any science user doing I/O in an application is for the network to be transparent, it may be beneficial for file systems to be informed about network topology facts. This does not mean that the science user has to know something about the network topology, it simply means that there may be information that the storage area network can provide to the file system that allows the file system to plan for optimal use of the network. Data paths at terascale are quite complex and at petascale the storage area network will be even more complex [Hospodor04, Xin05b, Khalsa06]. Communicating the network topology to a file system requires an interface standard.

Layout and geometry control

Currently, in some parallel file systems, applications can query and even control to some extent the underlying geometry of how a file is or will be stored [Isaila01]. Geometric attributes like width, depth, and stride can often be queried and set by the science application. Often I/O middleware can provide the mapping of geometry between application and file system [HDF5]. But there is currently no standard for interpreting, querying or setting these attributes. A standard in this area would simplify I/O middleware and also would simplify science applications as they seek to fully exploit petascale systems.

Hinting about upcoming I/O operations

High-performance scientific applications sometimes know well in advance which files will be accessed, which portions of the files will be accessed, and what the access patterns will be. This information can be known hours or more in advance of the application run-time, as access to high-performance systems is typically scheduled well in advance. All of this knowledge could be used to significantly improve system and application performance. The Institute will explore ways to extend POSIX I/O interfaces to allow far-in-advance "pre-fetch" information to be passed to the file system so that the underlying file/storage system can take advantage of this knowledge. Specific performance improvements possible with this information include metadata pre-staging and pre-creation, and data pre-staging and pre-fetching.

Data filtering

In some science applications, in situations when accessing only portions of a file, for instance specific fields in a repeating data structure, significant system resources can be wasted caching and transferring unused portions of the blocks containing the desired data. The institute will address this through application-specific data filtering. We will investigate effective interfaces for specifying data templates, develop efficient locking, caching, and other data management techniques on partial blocks, and develop a set of use cases demonstrating the effectiveness of this mechanism and providing examples for application developers to use in developing their own templates.

Richer Metadata API

As the HPC community approaches petascale, the amount of data and metadata being dealt with by a science application user will require new organizational tools. Tools currently provided by the file system for helping scientific application users manage their data are very simplistic: names for files, dates about files, and a few size and permission attributes about files. Outside of the file system there are some libraries that provide data formatting and annotation information using science terms about files [Gertz01]. This mix of simple tools for data/file management and the lack of integration of these tools makes data management for a scientist difficult. What is needed is a rich metadata standard with common update and query capabilities that are highly integrated with the file system.

Archive

Accessing data from file systems is very similar to accessing data in archives; using the POSIX I/O-file system API for accessing archive data is a natural desire. While the XDSM DMAPI standard [OPEN] was designed to be a file system "oriented" interface for accessing archived data, commonly known as hierarchical storage management (HSM), this standard has never been supported broadly, mostly due to the operating system kernel complexities required to make this API available. Given this situation, another serious attempt at making archives available via file system interfaces is needed. Extensions to the POSIX I/O interface to deal with the "offline" nature of archive data are needed to enable a variety of less kernel dependent HSM solutions. Additionally, the ANSI T10 Object Storage Device protocol [ANSI04] may be leveraged in this archive interface standardization endeavor [Dingshan06]. The experience some sites within the Institute have in implementing file system interfaces to petascale archives will be used to redefine the next generation interfaces.

Backup

Designers of systems that backup file system data have often asked for a bulk interface to file system metadata to enable efficient backup decisions to be made for files or objects in the file system. Because most file systems store the relevant metadata in sparsely distributed and indirectly discovered data structures, implementations of a bulk metadata interface are likely to be highly sensitive to the semantics of such an interface, especially as the number of files grows toward petascale. The NDMP standard provides for some representation and reordering freedom to file system implementations, but it lacks parallelism and offers no richer metadata definitions [NDMP]. For example, a log of recently changed file names has

been suggested as powerful tool for more efficient backup. Also, there is a need for standard interfaces for managing file system snap-shots, copy-on-write, and other incremental representations of a changing file system.

Active Storage extensions

Adding the ability to enhance storage systems with applications or application class specific extensions has shown great promise in dealing with mapping I/O in science applications to the underlying storage devices with breakthrough efficiencies [Gibson98, Anurag98, Riedel00, Riedel01]. The advent of the ANSI T10 Object Storage Device protocol is a good start towards enabling active storage concepts. Research, prototyping, and validation of POSIX I/O interface extensions to enable active storage capability are needed if the HPC community is ever to realize the promise of these breakthrough efficiencies in any broadly applicable way.

2. Reference implementations and validation

Enabling reference implementations of HPC storage related standards and APIs and facilitating validation using real science applications is a very important piece of the API and standards project. Without reference implementations and validation against petascale science applications, any standards are likely to be ineffective and not accepted or deployed widely.

This Institute includes researchers with access to the system development and production environments, classified and unclassified, incorporating some of the fastest, most scalable machines in the world. The opportunity to leverage these environments in what has historically been an opaque, but driving, environment is rare. The extreme size and capabilities of these machines represents an opportunity to gather data and validate these APIs and performance driven solutions far beyond the norm.

IV. Project 2: Petascale Storage Application Performance Characterization

A. Summary

We cannot predict the required characteristics of future storage and I/O infrastructure without understanding the storage and I/O demands of current applications.

However, obtaining that understanding has proven difficult for large-scale scientific applications due to lack of appropriate benchmarks, activity traces, and workload information [HEC05]. Without such information, researchers and vendors are likely to focus on workloads for which information is readily available, such as small file servers [SPEC97] and Internet services [SPEC05]. Storage systems are being designed and deployed for scientific workloads without accurate information about the nature of those workloads. These systems are poorly suited not only for today's scientific computing centers but also as starting points for future petascale systems.

This Institute is uniquely suited to address the lack of information about large-scale scientific workloads because of its members' expertise with high-end computing and storage systems, and expertise with performance characterization tools and techniques. Furthermore, the Institute includes members from several Department of Energy National Nuclear Security Administration and Office of Science sites, and these members have ties to application groups producing and employing at scale science codes such as:

- BLAST, ScalaBLAST (biology)
- CCSM (climate)
- CTH, Sierra/Salinas, Sierra/Calore (materials)
- EVH-1, RAGE (astrophysics)
- Firetec (atmospheric/combustion wildfire prediction)
- GYRO, GTC (plasma turbulence)
- MADNESS, NWChem, GA-Tools (chemistry)

- MCNP (nuclear physics)
- Sierra/Alegra (high-energy physics)
- QCD (quantum chromodynamics)

Thus the Institute has the potential to significantly impact application groups, helping them to perform breakthrough science.

The Institute's information and expertise will have its greatest impact if it is collected, normalized, and made available to all storage researchers and implementers. To facilitate this outcome, the Institute will create a public repository containing scientific application workload information, interoperable tracing and analysis tools, benchmarks, performance models, and guidance about storage and I/O best practices.

B. Categories of Investigation

For Project 2, the Institute will generate characterizations of scientific application I/O behavior, tools for collecting and analyzing such characterizations, I/O benchmarks that simulate the behavior of scientific applications, and models for predicting I/O performance. The Institute will make these artifacts available to the public via an open repository. No classified or export controlled information will be maintained in the repository. The Project 2 activities are discussed in more detail in the rest of this section.

1. Collection of scientific application I/O workload information

Understanding the behavior of storage systems under scientific application workloads is a key Project 2 activity. The artifacts of this activity will be stored in the Institute's repository in three forms:

1. Raw information about the I/O behavior of applications in the form of I/O request traces;
2. Characterizations of application I/O behavior and storage infrastructure workload, in the form of behavior profiles and summary reports; and
3. Performance models of application I/O behavior

To ease the task of working with raw workload information, I/O traces and machine-readable performance profiles will be stored using a small number of on-disk formats. Furthermore, reference implementations of support libraries for reading and writing these formats will either be included in the repository, or the repository will contain links to such libraries hosted on other sites. We will refresh our I/O characterizations and performance models when the systems running our characterized applications are upgraded or when new systems are deployed.

While gathering data is a simple task to describe, its importance cannot be oversold. Nothing advances science as well as new, extensive, raw data. The potential impact of this data collection alone on researchers outside of this Institute is very large.

2. Performance monitoring, analysis, and visualization tools

In addition to raw I/O workload information and characterizations of those workloads, the Institute will serve the storage community by making effective, easy-to-use performance tools available via its repository. Our aim is to provide tools not only for processing the raw workload information we collect, but also for collecting such information so that others can produce workload characterizations of their own applications that are comparable to the information in the repository. Because several of the tools available via our repository will be general purpose performance analysis tools, we will also provide guidance for using such tools to collect and analyze application I/O behavior.

Although there are a lack of performance tools focused specifically on characterizing and analyzing I/O performance, a few general-purpose performance tools can be used to collect and analyze application I/O behavior information. This is especially true for MPI applications that use MPI-IO. Because the MPI standard defines an easy-to-use profiling interface [MPI96] and this profiling interface includes the MPI-

IO functions, it is relatively easy for a performance tools like mpiP [Vetter01], TAU [Mohr96], and IPM [IPM06] to collect I/O event traces or profiles at the MPI-IO level. For applications that use I/O libraries such as parallel netCDF [Li03] or HDF5 [HDF05], or that use low-level I/O interfaces such as Fortran I/O calls, interposition techniques can be used to collect I/O workload information. In this approach, a performance monitoring library is interposed between an application and the I/O interface it uses. When the application makes an I/O call, control is passed instead to the monitoring library. The monitoring library generates an event record or updates its profile statistics, and then calls the actual I/O function.

This project will use or develop appropriate profiling/tracing tools as needed for gathering a wide range of pertinent I/O information.

3. Guidance

Workload data collected by the Institute has the potential to have a broad impact on the storage community. Our hope is that our efforts will encourage others to collect their own workload information, and possibly share it with the community via our storage repository. To support such activity, the Institute will work with high-end computing sites to produce best practice guides for I/O performance data collection and analysis. Like our workload data, these guidance reports will be made available via the Institute's repository.

V. Project 3: Petascale Storage System Dependability Characterization

A. Summary

The ability of petascale applications to make forward progress towards solution depends on the ability of the application to get work done between application interrupts. However, with millions of processing, storage, and networking elements involved in petascale computing, element failure will be frequent. In order to make petascale computing a worthwhile endeavor, reliability/dependability at scale delivered to the science application is paramount. Designing and implementing highly reliable systems, at petascales as well as at small scales, requires a good understanding of the underlying failure characteristics. Unfortunately, failure behavior of real systems is poorly understood, even for simple (non-terascale) systems, mostly due to lack of real data. As pointed out by many researchers, disk drive manufacturers are loathed to provide information on disk failures [Talagala99, Prabhakaran05]. Indeed, when NERSC, during the NERSC-5 procurement, asked vendors to provide "information concerning the number of defects ... for all major software and hardware components", no vendor provided meaningful data. This is partially because they do not track the data for software failures, and often do not have data organized for hardware failures either. But is it also because they have no motivation to provide the data themselves. In fact, one major vendor responded that they do "not make this data available because this data could be used to misrepresent individual products"!

Researchers are focusing on workstation and Internet service failure models, because those are the environments for which some failure data is available. Of course, these environments are completely different from large scale science computing systems in terms of consequences of failures and maintenance characteristics. Moreover, these studies are often based on only a few months of data, covering a few hundred failures [Tang90, Kalayanakrishnam99, Xu99, Oppenheimer03, Sahoo04, Nurmi05]. Many of the most commonly cited studies stem from the late 80s and early 90s, when computer systems where significantly different from today [Gray86, Iyer86, Meyer88, Gray90, Tang90, Lin90, Murphy95, Talagala99]. Given the lack of even terascale science failure data, the community will see researchers and designers aimed at the wrong targets (for scientific application needs). Working with unrelated and ill-matched failure characteristics may very well lead to designs of storage solutions that simply won't work when scaled to the sizes and setups needed to support petascale computing.

Given this lack of data about and understanding of failures for current systems, the science community is not well prepared for understanding and predicting failure at petascale. Existing research on failure in

terascale storage systems [Xin03, Xin04, Xin05a, Xin05b] assumes that failure types and frequencies are based solely on part counts and constant failure rates. As the scale increases the inaccuracy of these approximations can grow significantly. An early failure profile for petascale computing, networking, and storage would be of great benefit to help focus research funding on the most pressing problems or the problems requiring the longest lead time to solution.

Given the extremely large amount of resources employed by petascale science applications, having the ability to predict a significant portion of future failures may be required for the petascale computing infrastructure to be effectively used.

Finally, large scale file systems have a complex layer of software required to make all of the large numbers of standard components appear to users as a single, global storage system, making every significant increase in scale uncover important new classes of interacting multiple failure cases. Effective petascale global file systems need significant new reliability studies.

The Institute will address research in dependability at petascale through data collection, analysis, and modeling. The National Labs represented in the Institute have very strong ties with the entire U.S. Government funded high performance computing community via bilateral relationships and through the High End Computing/Interagency Working Group (HEC/IWG) I/O and file systems coordination activities. Through this extended community, a vast amount of failure and usage data can be collected and made available for analysis and modeling. The universities represented in the proposed institute have strong interest and experience in failure research. This Institute is strongly positioned to publish large amounts of existing terascale failure and usage data and to do ground breaking work in failure analysis, modeling, and prediction for petascale science.

B. Categories of Investigation

1. Reliability, usage, and error log data collection

The lack of publicly available failure data from real systems is a serious hindrance in designing more reliable systems, since it forces system designers and researchers to work with hypothetical models and assumptions on failure behavior, rather than hard facts. Some commonly made assumptions are, for example, that failures are not correlated [Bolosky00, Bhagwan04, Dabek04, Yu04]; that disk failure rates are constant over time [Gibson93, Ng94, Hou97, HoonBaek01]; and that failures are fail-stop, i.e. a disk either works perfectly or fails in its entirety and in a detectable manner [Schneider90]. The validity of these assumptions in practice is questionable. For example, disk drive behavior is much more complex in practice than fail-stop, including latent sector errors or corruptions of individual blocks [Kari93, Schwarz04, Prabhakaran05]. However, without the necessary data to build and justify more realistic and less tractable failure models, it is hard to go beyond these simple assumptions.

The Institute will help to close this gap by providing large amounts of real failure data. In particular, the three major DOE Office of Science supercomputer centers and the two major DOE/NNSA supercomputer centers will enable the collection of up to, and perhaps more than, a decade of error, usage, failure, and reliability data. Since some of the sites involved have classified operations, and others have privacy issues, providing this data will require some work. However, the enormous potential of millions of publicly available failure, usage, and error records to enable breakthrough research, provides a strong motivation to put in this extra work.

Data collection at the Institute's HPC sites has the potential for wide-ranging impact. The Institute's members will also seek to motivate other HPC sites to collect and make available similar data. To further encourage and facilitate collection of failure data at other sites, the Institute will work with the a broader community on failure and usage collection best practice guides, failure and usage record formats, and failure and usage data analysis tools for petascale science clusters. Additionally, the Institute will manage a clearinghouse for these data.

2. Shared repository for dependability data

The availability of large scale computing system failure and usage data to SciDAC researchers everywhere is paramount to ensuring effective understanding of failures and interrupts at petascale. The Institute will maintain a openly shared repository for failure and usage data so that this important data is accessible to a broad community.

3. Dependability analysis, modeling and prediction

This project's collected data will provide the basis for creating more realistic models of failure properties. Important failure properties that interest system designers, administrators, and researchers alike, and require good models, include the root cause of failures, time between failures and its statistical properties, time to repair a failure and its statistical properties, correlations between failures and workload, correlations between failures in different devices, and locality of disk block failures. In addition to providing researchers and system designers with a general understanding of failure characteristics, the models derived from this project's collected data will be of immediate practical use in at least three different contexts.

The first context is *dependability benchmarking*. While many widely accepted *performance* benchmarks exist (e.g. the SPEC benchmark suite [SPEC05]), there are virtually no *dependability* benchmarks available that can be used to proactively evaluate and compare the dependability of systems. Realistic models of fault workloads provide the basis for creating realistic dependability benchmarking tools.

The second context is *system design and configuration*. When designing and configuring a system to achieve a certain level of dependability one is faced with a huge search space, including the choice of hardware components, choice and configuration of software components, choice and configuration of redundancy mechanisms such as replication or erasure codes, redundancy in network paths, etc. Accurate models of failure behavior are a prerequisite to evaluating and comparing the reliability and performance of the different design and configuration alternatives on a realistic basis.

The third context is a *predictive capability* for system management and planning. In system management, the combination of error logs, usage data, and failure data could be used to develop statistical learning techniques for failure prediction and proactive fault management. In planning, failure models based on the collected data can be used to derive a predictive modeling capability for failure at petascale. This information can be used to shape future government and industry investments in dependability solutions.

VI. Project 5: Exploration of Novel Mechanisms for Emerging Petascale Science Requirements

A. Summary

Petascale computing places very high demands on the storage system; current approaches to storage scalability are insufficient to meet these demands. We must focus on true scalability rather than simply targeting "peta" scalability to avoid facing this challenge again as scientific computing demands continue to grow.

Large national science grids carry enormous amount of information over vast areas through very high bandwidth links. As a result, the bandwidth of these networks rivals memory bandwidth, but the speed of light constraint means that latency remains an issue. The research that we will conduct on NFSv4 and pNFS will take advantage of these very high bandwidth links and mitigate the effects of latency to provide transparent high performance access to science data. We will also provide, through techniques such as object storage devices (OSD), the parallel virtualization of storage, so, as computing resources grow and the complexity of applications increase, the storage system will be able to keep pace with these advances.

For petascale systems and storage it is a necessity to avoid the need to move files to make them accessible by different machines. By reducing the number of distinct storage systems that must be maintained, a shared, scalable storage infrastructure simplifies user activities and reduces administration overheads. The research that we will conduct will develop techniques for sharing of resources in ways that do not interfere and do not negatively impact performance, but instead enhance performance and usability of the system. This approach will often require joining resources of disparate realms into a virtual organization that manages and controls resource access and allocation. The research that we will conduct on petascale computing security will encompass truly scalable algorithms and techniques.

Petascale scientific computation occurs on very complex information with inherently rich attributes. The storage system should provide support to the application so that more of those attributes can be identified, queried and modified instead of being hard-coded into the application program or embedded in an archive format based on an evolving standard. The research that we will conduct will provide support for rich metadata in an environment where it can be defined, searched, queried or modified. Through the use of OSD techniques, we will insure that this rich metadata will be accessible in a highly parallel fashion that enhance the performance of applications.

B. Categories of Investigation

1. Global access to parallel storage

NFSv4 [Pawlowski00] is designed with wide-area access in mind. NFSv4 extensions, notably pNFS [Goodson05], offer parallel access to petascale data stored in a storage area network using block, object or other parallel storage architectures. pNFS separates NFSv4 data and metadata paths, allowing data movement directly between clients and storage system elements while continuing to use NFSv4 metadata services for naming, security, etc. By bridging petascale computing and petascale data, pNFS relaxes the coupling between compute and storage plants [Hildebrand05].

Science grids carrying multiple 10 Gbps wavelengths— National LambdaRail, UltraScience Net [Rao05], StarLight, UltraLight— allow geographically dispersed clusters to communicate as fast globally as they do locally. Network backbone bandwidth is approaching that of CPU memory. For streaming I/O, this completely decouples storage placement from the compute plant. For other data access patterns, the current practice of pre- and post-staging copies of data objects can be enhanced with automated migration and/or replication of data [Zhang06]. Integrating data management with NFSv4 and pNFS provides consistent access to data as well as uniform security mechanisms.

2. Security requirements for petascale storage

Some of the demands of petascale computing can be met by extending current security mechanisms [Gibson98, Fu99, Gobioff99, Miller02, Kallahalla03, Li04]. For example, new mechanisms may be required to protect objects whose size obsoletes algorithms and data structures intended for more conventional workloads. Uniform security mechanisms for protection of petascale data are beneficial and desirable, but petascale computing often requires joining resources of disparate realms into a virtual organization that manages and controls resource access and allocation. Agile mechanisms for creating and managing the cryptographic scaffolding that corresponds to the structure of the virtual organization is a challenge faced today by the petascale computing vanguard, including the Institute members.

Another concern is the ability of existing techniques to scale to systems with thousands of clients and thousands of storage devices. Performance of security mechanisms in such an environment can be a limiting factor for parallel storage performance [Olson05], so focusing on improving security for operations including collective open and other parallel operations will allow the design of secure parallel file systems that do not trade performance for security.

Beyond the security and privacy requirements of shared data sets, petascale data often has great value just in terms of the resources expended in its production. We will explore techniques for securing petabyte-

scale file systems, both improving local file system security and facilitating the sharing of data between geographically and organizationally diverse file systems. These techniques will keep valuable data secure while preserving high-performance access, both locally and remotely.

3. Predictable sharing

By avoiding the need to move files to make them accessible by different machines and by reducing the number of distinct storage systems that must be maintained, a shared, scalable storage infrastructure simplifies user activities and reduces administration overheads.

Substantial and unpredictable inefficiencies often accompany this kind of sharing. Because it is not acceptable for a large (and very expensive) machine to be slowed by interference with activity from other systems, such sharing is not typically supported in today's high-end computing environments. Even private (per-cluster) shared storage can become a bottleneck if multiple applications execute simultaneously on subsets of the computation cluster if their I/O paths overlap. Most current approaches are focused on cluster-based storage solutions [Lee96, Gibson98, Lustre02, Ganger03, Ghemawat03, IBM04, Saito04]. A scalable storage solution naturally creates the potential to use one system, scaled as necessary, to support multiple computation servers running separate applications. Unfortunately, doing so often adversely affects performance in much greater magnitude than should be expected from simply splitting the resources: the total ends up being much less than the sum of the parts.

To promote sharing, we need to insulate from one another the I/O performance of high-end computing applications sharing a cluster storage system. In particular, such sharing should not cause unexpected inefficiency. An application sharing a cluster may enjoy less attention from the system and thus may see lower performance, yet the work accomplished should be proportional to the fraction received. Ideally, no I/O resources should be wasted due to interference between applications, and the I/O performance achieved by a set of applications should be predictable fractions of their non-sharing performance.

To eliminate inefficiencies that can arise from sharing, the resource management policies within the storage system must be designed to avoid them. Many computer system resources, such as CPU time and network bandwidth, can be shared with relatively minor interference concerns, but the two primary storage system resources—transfer bandwidth and I/O latency—cannot. Performance can be adversely affected by applications that interfere along the I/O path, affecting the ability of traditional disk and cache management policies to maintain good performance. Accomplishing performance insulation and predictability requires cache management, disk layout, disk scheduling, and storage-node selection policies that explicitly avoid interference.

Our research will focus on techniques that can promote sharing while eliminating interference between multiple applications. These issues arise in disk layout, cache management, replica selection, and disk utilization [Chambliss03, Lumb03, Karlsson04, Wu06]. We will explore techniques that allow multiple applications to use a single storage system without adversely affecting each others' performance.

4. Rich metadata in petabyte-scale storage

We propose to make rich metadata [Ames05] scalable for use in object-based petabyte-scale storage. We will build on work that resulted in the object-based storage system Ceph, and the Linking File System (LiFS) that implements primitives for rich metadata such as file attributes, relational links, and link attributes. A key challenge is the scalable maintenance of rich metadata indices while enabling scalable searching.

We assume that the access characteristics of rich metadata resemble data access characteristics more than file system characteristics. Applications already use additional files to help organize other data files. This is especially evident in HPC, where popular libraries such as HDF exist for managing groups of files. This implies that rich metadata processing is more parallelizable than traditional metadata management.

We therefore propose to extend OSDs to support rich metadata management. This has three main advantages. First, the load of rich metadata management is distributed over a large number of storage nodes instead of creating a potential bottleneck at the metadata server cluster. Second, distributed indexing technologies can be combined with hash-based object placement functions [Brinkmann02, Honicky04, Weil06] for query routing. Third, OSDs can act independently on rich metadata to perform tasks such as distributed replica placement.

5. Para-virtualization

The goal in building massively scalable storage systems is to allow bandwidth to scale linearly with the number of clients that access storage. We propose to experiment with virtualization technology for efficient, cost effective, and manageable scaling. Linear storage bandwidth scalability can potentially be achieved with COTS hardware and existing distributed file system technology by running a client and a storage server on the same hardware: this lets the combined bandwidth of local client disk subsystems be used by the global file system. If the I/O load is spread across storage subsystems—a constraint that can be effected manually, e.g., through striping, and potentially through a policy to use local storage on a virtual storage server. This distribution also has the potential to accelerate databases, a benefit to access-limited SciDAC applications, thanks to the large amount of RAM available for caching information and the large number of seek operations per second possible with many drive spindles.

VII. Project 6: Exploration of Automation for Petascale Storage System Administration

A. Summary

As system size grows, so does system complexity, especially when one seeks maximal system efficiency. With today's terascale systems, system administration is a major contributor to cost of ownership, downtime, and inefficiency [IBM01, Kephart03]. These issues are especially acute in storage systems for critical input and result data; without expert configuration and tuning, storage systems can become a performance bottleneck. These problems will grow in importance as the science community strives to move towards petascale systems. To be feasible in practice, petascale storage systems require new tools and techniques that reduce the human cost of storage infrastructure administration.

Institute members have considerable experience with deploying and administering terascale systems as well as a research record focused on finding new approaches to automating difficult and time-consuming storage administration tasks [Ganger03], including diagnosing system problems [Wang02, Narayanan05, Thereska05], making configuration/tuning decisions [Wang04, Abd-El-Malek05], and even automated control [Salmon03, Thereska04]. This combination of practical and research experience promises innovative solutions that help with administration of large-scale storage infrastructures deployed in support of science applications. One exciting aspect of this Institute project will be early field-testing and possible adoption of promising new tools and techniques in the participating National Labs.

Project 6 has the most long-term challenges of the Institute activities. Much of the effort here will be adapting automation techniques developed in other domains to the needs of petascale storage systems. Accordingly, we focus here on identifying key challenges rather than proposing specific solutions.

B. Categories of Investigation

1. Instrumentation, visualization, and diagnosis

Scale and complexity make large-scale storage systems difficult to plan, deploy, and maintain; yet, a deep understanding of system organization and behavior is critical to addressing failure and performance. For petascale systems, it is unlikely that current approaches based on aggregated performance counters and "knowing the system inside and out" will scale. With thousands of disks, dozens to hundreds of storage

servers, and complex interconnection networks, storage system administrators striving to keep petascale storage systems serving science needs effectively require comprehensive instrumentation and new aggregation tools. Work is critically needed in three interrelated areas: instrumentation, visualization, and diagnosis.

Instrumentation is central to understanding system behavior, especially when a system misbehaves [Barham03, Thereska05]. Information is needed regarding the state, timing, and resource usage of each software and hardware component and their intercommunication. Instrumentation detail must be balanced against collection and retention overheads; the value of additional detail is directly related to the types of visualization and diagnosis that it enables.

Petascale storage systems will have far too many components to allow examination of all of the raw instrumentation data. *Visualization* tools will be needed to aggregate instrumentation data into a high-level view that lets administrators drill down so as to analyze specific parts and specific time periods in more detail. Tools that draw attention to the most important and interesting issues exposed by the instrumentation data would be of great value.

Beyond their intrinsic visual appeal, good visualization tools have great value as part of system *diagnosis*. Beyond visualization, petascale storage administrators will need tools that help automate diagnosis, e.g., by identifying misbehaving components, suggesting possible causes, and assembling supporting evidence that allows administrators to confirm and dig deeper [Chen02, Aguilera03]. Automated diagnosis is a long-term goal, but each step towards automatically focusing attention on the most likely problem areas in a large-scale system will help make maintainable petascale storage systems more feasible.

2. Planning of data distribution

With scores of servers holding thousands of disks, data distribution decisions will play a major role in determining I/O performance and, thus, likely overall performance in many petascale systems. Therefore, planning data distribution will be a crucial aspect of petascale storage system management. It includes initial decisions, as data is first created, as well as decisions about data migration over time.

Data distribution includes how data is broken up and which pieces are assigned to which servers (and to which disks within those servers). In terascale storage systems, data distribution choices affect load-balancing, parallelism, and network data flows. In petascale storage systems, reliability and competition will enter the equation. As the number of servers grows to the numbers needed to support petascale science, storing data redundantly across servers (rather than "simply" using RAID internally) becomes increasingly necessary [Wylie00, Ganger03, Saito04]. Similarly, having huge private storage systems for each large machine will become less and less viable as a deployment strategy, so understanding which data should be stored on the same servers becomes an important consideration [Chambliss03, Lumb03, Karlsson04].

Increased deployment of high-speed global optical networking creates the possibility of geographically dispersed petascale storage. Transparently managing the replication and migration of data across a collection of collaborating large-scale storage deployments will be an important capability. Doing so includes some of the local data distribution concerns as well as issues of resource allocation, access control, administrative domains, and protocol compatibility.

3. Automated control of configuration, tuning, healing

Ideally, administrative intervention for tuning and repairing can be obviated, but this is a long-term goal in which diagnosis, understanding, decision-making, and administrative action are all automated [IBM01, Kephart03]. A valuable first step is to automate the implementation of human-made decisions. This can reduce human administration overhead. Removing the human from the loop can also speed up changes and promote agile adaptation, which is especially important in petascale systems, whose myriad components and massive scale make failure and performance problems commonplace.

Achieving trustworthy automation requires the work described above (instrumentation, diagnosis, data distribution planning) as well as robust control mechanisms. For example, self-tuning consists of monitoring system operation, identifying potential improvements (e.g., via modeling of potential options), constructing a plan for making them, and implementing that plan. Hysteresis and other robustness mechanisms help achieve stability and avoid cyclic changes. As another example, self-healing consists of monitoring system operation, diagnosing failures, constructing a plan for repairing them, and implementing that plan. Errors in diagnosis or repair planning can do more harm than good, which illustrates why robustness of automation must be a primary focus.

VIII. Project Plan and Milestones

Project 1: Petascale Data Storage Outreach (Annual Budget Est.: $380K); LEAD – Garth Gibson, CMU

Year	Institution	Task/Milestone
1-2	LANL, SNL	Participate in HPC I/O and file storage systems curriculum development at participating institute universities including courses and parenthetical degrees.
1-5	ALL	Develop & host I/O and file storage workshop for science application developers and users; and for I/O and file storage researchers
1-5	LANL	Sponsor SciDAC and HEC/URA/NSF I/O and file storage R&D showcase as an extension to the HEC/IWG I/O and file storage workshop to showcase, update, and coordinate the HPC industry and university, SciDAC and HEC/IWG I/O and file storage related research and development.
1-5	PNNL	Deployment, support of HPC I/O filesystem packages for OS releases not supported by core development.
1-5	CMU, MICH	Course development integrating HEC storage examples, problems, techniques, tools; including training materials for science app programmers, HEC file system developers and grad students; publish materials for public use online.
1-5	MICH	Organize and participate in annual workshops held in conjunction with Supercomputing, Global Grid Forum, CCGrid, USENIX FAST conferences; publish reports, papers, and meeting results online
2-5	LANL, SNL	Participate in lecturing and university course delivery as a part of the HPC I/O and file storage system curriculum.
3	ORNL	Host I/O and file storage workshop for science application developers and users

Project 4: Protocol/API extensions for Petascale Science Requirements (Annual Budget Est.: $420K); LEAD – Gary Grider, LANL

Year	Institution	Task/Milestone
1-2	PNNL	Research and build upon previous Active Storage work to standardize an API and use model for Active Storage.
1-2	PNNL	Research and evaluate current API's for backup and HPSS such as NDMP, DMAPI, XFS/CXFS. Collect use scenarios for current installations at the collaborating institutions for evaluation of systems.
1-2	NERSC	Evaluate the suitability of the existing archive and backup and storage usage practice on NERSC platforms and the NERSC Global Filesystem and characterize the archive/backup extension needed to support a facility-wide file system.
1-3	UCSC, MICH	Investigate extensions to POSIX interface leading to parallelized interfaces supporting higher data rates and non-sequential data accesses, including the execution of application-specific data filters on storage nodes.
1-5	CMU	Reference implementations and evaluation of proposed POSIX API extensions for parallel science applications; release implementations for incorporation into open source NFSv4 codebase if appropriate with best effort support.
1-5	LANL, PNNL, SNL	Assist in the validation of emerging HPC storage related standards and API's such as parallel Network File System (pNFS), iSCSI Enhanced RDMA (iSER), and active storage, and enhanced POSIX I/O.
2-3	PNNL	Build a reference implementation of an Active Storage API into a publicly available parallel file system.
2-3	NERSC	Collaborate in defining the archive/backup and storage allocation requirements for facility-wide file systems to be considered for use at the Office of Science computing facilities.
3-4	PNNL	Create and use multiple Active Storage Components to improve applications that are used in the Institutes' systems.
3-5	PNNL	Assist in the definition and creation of API's or standards that allow parallel file systems and backup systems to work together using rich meta-data to provide policy based data storage and retention.
3-5	UCSC	Investigate further extensions, including exploiting the computational power in OSDs and other forms of active storage, to further widen the interface. This includes direct execution of application code on storage nodes.
3-5	MICH	Develop and distribute reference implementations and evaluation of proposed POSIX API extensions for parallel science applications.
4-5	PNNL	Evaluate uses for active storage in new standards that have emerged and utilize the Active Storage framework.
4-5	NERSC	Collaborate in developing an HSM & archive reference tool with the NERSC Global FS and HPSS that gives users the appearance of an infinite capacity by moving data transparently to HPSS-managed storage and returning it on demand.

Project 2: Petascale Storage Application Performance Characterization (Annual Budget Est.: $640K); LEAD – William Kramer, NERSC

Year	Institution	Task/Milestone
1	ORNL, NERSC	Evaluate suitability of tools available for I/O and storage characterization
1	ORNL	Characterize storage and I/O demands of at least one DOE SciDAC code on NLCF platforms
1	NERSC	Gather supercomputer, networking, and I/O and file storage usage data, including job length, size, processor usage and other usage profile data for analysis.
1	SNL	Enhance Red Storm, Sandia's user-level, virtual filesystem framework, to support efficient application I/O tracing.
1-2	ORNL	Modify tools on NLCF platforms for I/O and storage characterization
1-2	LANL, SNL	Provided parallel I/O traces of unclassified parallel applications.
1-2	PNNL	Assist in the collection of system wide file system usage, and parallel job use patterns.
1-2	LANL, SNL	Provide parallel I/O traces of synthetic parallel benchmarks as well as source for the benchmarks to enable base lining for parallel I/O trace analysis and replay research.
1-3	CMU, MICH	Consensus development of tracing best practice guides, formats, analysis tools, and replay tools for highly parallel science apps; scientific application I/O kernels for benchmarking; release tools as open source.
1-4	PNNL, ORNL	Provide visualization applications for gathered I/O traces, and signatures.
1-5	UCSC	Develop a repository of high-end computing traces.
1-5	UCSC	Examine lightweight I/O tracing techniques suitable for very large high-performance computing systems.
1-5	UCSC	Collect and analyze traces.
2-3	ORNL	Instrument and characterize data movement and storage infrastructures at Office of Science centers
2-3	ORNL	Characterize storage and I/O demands of at least three more DOE SciDAC application codes on NLCF systems
2-3	LANL, SNL	Provide unclassified and scrubbed parallel I/O traces of classified parallel applications.
2-3	NERSC	Instrument tools to characterize I/O workloads of DOE SciDAC application codes on NERSC systems.
2-4	LANL	Provide unclassified derived I/O kernels that faithfully reproduce I/O patterns of both classified and unclassified parallel science applications.
2-4	CMU	Support validation of tracing tools and benchmarks by partners with large-scale systems and HEC science users.
2-5	ORNL	Track scale & capability increases in NLCF systems by revalidating characterizations and performance models on updated systems
2-5	LANL, SNL, PNNL, NERSC	Assist in validation of trace analysis, replay, and workload/system simulation by providing access to parallel computational resource and interfacing to real science applications.
3	ORNL	Extend techniques for incorporating I/O and storage behavior into a performance prediction framework
3-4	ORNL	Express behavior models for characterized applications in the modeling framework chosen in Year 3.
4-5	CMU	Refresh tool chain and validation as needed for emerging HEC machines and science applications.
4-5	NERSC	Assist in benchmark development based on the I/O workloads of the characterized science applications.
4-5	NERSC	Track scale and capability increases in NERSC systems by re-evaluating earlier characterizations on new systems.
4-5	MICH	Develop and support tools to manage, customize, and distribute large trace collections.

Project 3: Petascale Storage System Dependability Characterization (Annual Budget Est.: $200K); LEAD – Gary Grider, LANL

Year	Institution	Task/Milestone
1-2	LANL, PNNL, NERSC	Collect up to a decade of supercomputer, high performance networking, and I/O and file storage system reliability data including machine and environment configuration information, mean time to interrupt/mean time to repair (MTTI/MTTR) and failure cause data.
1-2	LANL, PNNL	Collect up to a decade of supercomputer, high performance networking, and I/O and file storage system usage data, including job length, size, processor usage and other usage profile data.
1-2	NERSC	Evaluate different approaches to proactively assess reliability issues.
1-3	CMU	Consensus development of failure & usage collection best practice guides, record formats, data analysis tools for large scale HEC clusters; release tools as open source with support.
2-3	NERSC	Instrument tools for proactively assess system reliability.
2-4	CMU	Collaborate on analysis of HEC system data collected by Institute partners for configuration and tuning.
3-4	NERSC	Assist in defining highly dependable, highly scalable data movement and storage reference architecture or implementation to be considered for use at Office of Science computing facilities.
3-5	NERSC	Assist in defining highly dependable, highly scalable data movement and storage reference architecture or implementation to be considered for use in the wide area between computing facilities.
4-5	CMU	Refresh tool chain and data analysis as needed for emerging HEC machines and science applications.
4-5	NERSC	Assist in validation of system design and monitoring tools for petascale storage system.

Project 5: Exploration of Novel Mechanisms for Emerging Petascale Science Requirements (Annual Budget Est.: $470K); LEAD – Darrell Long, UCSC

Year	Institution	Task/Milestone
1	UCSC	Investigate need for and appropriate extensions to metadata in high-end computing storage systems.
1-2	UCSC	Research lightweight authentication techniques appropriate for large-scale distributed storage systems
1-3	CMU	Explore performance benefits of integrating knowledge of network topology and multipath selection into file system transfer planning; release implementations for inclusion into open source NFSv4 if appropriate with best effort support.
1-3	CMU	Explore scalability of NFSv4 extensible attributes if being used for fine grain QoS in HEC storage with both high bandwidth and high create rate files; release for inclusion into open source NFSv4 if appropriate with best effort support.
1-3	CMU	Develop and experiment with approaches to allowing shared usage of petascale file systems by multiple high-end machines and clusters without unpredictable interference
1-3	MICH	Investigate and characterize data production and consumption patterns of petascale computing applications. Develop pNFS-based data management, placement, and replication strategies for petascale data storage.
1-3	MICH	Investigate structural properties of collaborating groups that affect virtual organization construction and management. Develop tools for virtual organization formation, management, and dissolution.
1-4	PNNL	Provide a system for efficient use of para-virtualization technology to allow a distributed file-system to scale in bandwidth with number of clients in a cost effective and manageable manner.
1-5	SNL	Research, collaborate with other institute members, and potentially develop solutions for parallel I/O over long-haul pipes (WAN), and scaling metadata performance.
2-4	UCSC	Implement and analyze metadata extensions.
2-5	LANL	Assist in research to define an API for how scientific applications could provide application specific metadata to be stored in the parallel file system and how applications could query this extended metadata information.
2-5	NERSC	Assist in research to define an API or an extension to DMAPI to utilizing parallel data transfer paths for data movement between online filesystem and archiving storage such as HPSS.
3-4	UCSC	Research lightweight encryption techniques appropriate for large-scale distributed storage systems
3-5	CMU	Based on exploration results define consensus APIs for standardized communication of network topology to file system and standardized interpretations for specific extended attributes.
3-5	MICH	Develop algorithms for automated storage placement in petascale computations, integrating cost, performance, priority, authorization, and resource availability.
3-5	MICH	Refine tools and portals for controlled resource sharing within virtual organizations.
5	UCSC	Push for standardization of appropriate metadata extensions; and of storage security techniques and algorithms

Project 6: Exploration of Automation for Petascale Storage System Management (Annual Budget Est.: $390K); LEAD – Greg Ganger, CMU

Year	Institution	Task/Milestone
1	UCSC	Research unmet storage needs of representative scientific applications in terms of placement, migration, and replication planning.
1-3	PNNL	Research methods used by collaborators for the current methods of data-movers and archives, and their use or misuse.
1-3	CMU	Develop and experiment with new approaches to instrumentation that scale while also providing deeper insight into file system usage patterns, internal performance characteristics, and component status.
1-3	MICH	Explore resource costs and trade-offs in replicating petascale data for automated failure handling.
2-3	ORNL	Develop automated methods for adaptive optimization of data storage and I/O infrastructures on NLCF systems
2-4	CMU	Develop and experiment with visualization techniques for condensing instrumentation data from large-scale systems to focus attention on the most interesting effects.
2-5	LANL, SNL	Participate in the creation of autonomic systems designs and management visualization tools for easily finding and viewing exception data from the massive amount of operational data generated by petascale file storage systems.
2-5	LANL, SNL	Assist in validation of autonomic system design, management visualization tools, and at-scale failure and usage analysis by providing access to parallel computational resource and interfacing to real production computation, networking, and storage systems management personnel.
2-5	CMU	Develop and experiment with approaches to automating aspects of configuration and tuning for petascale file systems, including automatic data placement and reorganization as applications requirements and access patterns change.
2-5	NERSC	Assist in research automated methods for improving system dependability of data storage and I/O infrastructures for large-scale systems within a computing facility or between computing facilities.
2-5	MICH, UCSC	Develop and experiment with approaches to automating aspects of configuration and tuning for petascale file systems, including automatic data placement and reorganization as applications requirements and access patterns change.
3-5	PNNL	Design an API for a cluster job-scheduler to specify when files need to be moved, at what QOS level to use.
3-5	CMU	Develop/experiment with automating aspects of performance problem and failure diagnosis in petascale file systems.
3-5	MICH	Develop tools for automated replication of petascale data. Explore failover strategies that build on these tools.
3-5	MICH	Develop and experiment with approaches to automating aspects of performance problem and failure diagnosis in petascale file systems.
4-5	ORNL	Modify and evaluate storage infrastructure investigated in Years 2-3 with adaptive optimization techniques

Appendix A, Part 1: References

[Abd-El-Malek 05] Michael Abd-El-Malek, William V. Courtright II, Chuck Cranor, Gregory R. Ganger, James Hendricks, Andrew J. Klosterman, Michael Mesnier, Manish Prasad, Brandon Salmon, Raja R. Sambasivan, Shafeeq Sinnamohideen, John D. Strunk, Eno Thereska, Matthew Wachs, Jay J. Wylie. Ursa Minor: Versatile Cluster-based Storage. Proceedings of the 4th USENIX Conference on File and Storage Technology (FAST '05). San Francisco, CA. December 13-16, 2005.

[Aguilera03] Marcos K. Aguilera, Jeffrey C. Mogul, Janet L. Wiener, Patrick Reynolds, Athicha Muthitacharoen. Performance debugging for distributed systems of black boxes. SOSP03, Bolton Landing, NY, 19-22 Oct. 2003. p74-89, 2003.

[Ames05] Alexander Ames, Nikhil Bobb, Scott A. Brandt, Adam Hiatt, Carlos Maltzahn, Ethan L. Miller, Alisa Neeman, Deepa Tuteja. Richer File System Metadata Using Links and Attributes. MSST05. Apr, 2005. Monterey, CA.

[ANSI04] SCSI Object-based Storage Devices (OSD), ANSI INCITS 400-2004, Dec. 15, 2004, http://www.t10.org/ftp/t10/drafts/osd/osd-r10.pdf

[Anurag98] Anurag Acharya, Mustafa Uysal, and Joel Saltz "Active Disks: Programming Model, Algorithms and Evaluation" ASPLOS98, August 1998.

[Azagury03] A. Azagury, V. Dreizin, M. Factor, et. al., Towards an Object Store, MSST 03.

[Barham03] Paul Barham, Rebecca Isaacs, Richard Mortier, Dushyanth Narayanan. Magpie: online modelling and performance-aware systems. HOTOS03. Lihue, HI, 18-21 May 2003. 79- 84.

[Bhagwan04] Ranjita Bhagwan, Kiran Tati, Yu-Chung Cheng, Stefan Savage, Geoff M. Voelker. Total Recall: System Support for Automated Availability Management. Proceedings of Symposium on Networked Systems Design and Implementation (NSDI), 2004.

[Bolosky00] William J. Bolosky, John R. Douceur, David Ely, Marvin Theimer. Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs. SIGMETRICS Perform. Eval. Rev., V28N1, 2000.ISSN0163-5999, 34—43.

[Brandt03] S. A. Brandt, L. Xue, E. L. Miller, et al., Efficient Metadata Management in Large Distributed Storage Systems, MSST03.

[Brinkman02] Andre Brinkmann, Kay Salzwedel, Christian Scheideler. Compact, Adaptive Placement Schemes for Non-Uniform Capacities. SPAA02, August 2002, 53-62. Winnipeg, Manitoba, Canada.

[Chambliss03] David D. Chambliss, Guillermo A. Alvarez, Prashant Pandey, Divyesh Jadav, Jian Xu, Ram Menon, Tzongyu P. Lee. Performance virtualization for large-scale storage systems. Symposium on Reliable Distributed Systems. Florence, Italy, 06-08 October 2003. 109-118.

[Chen02] Mike Y. Chen, Emre Kiciman, Eric Brewer. Pinpoint: Problem Determination in Large, Dynamic Internet Services. International Conference on Dependable Systems and Networks (DSN'02). Washington, DC, 23-26 Jun. 2002. p 595-604.

[Dabek04] Frank Dabek, Jinyang Li, Emil Sit, James Robertson, M. Frans Kaashoek, Robert Morris. Designing a DHT for Low Latency and High Throughput. NSDI 2004: 85-98.

[Dingshan06] Dingshan He, Xianbo Zhang, David H.C. Du, Gary Grider, "Coordinating Parallel Hierarchical Storage Management in Object-base Cluster File Systems", MSST06, College Park, Maryland USA, May 15-18, 2006. (To appear).

[Fu99] Kevin Fu. Group Sharing and Random Access in Cryptographic Storage File Systems. MIT Masters Thesis, June 1999.

[Ganger03] Gregory R. Ganger, John D. Strunk, Andrew J. Klosterman. Self-* Storage: Brick-based Storage with Automated Administration. CMU Technical Report, CMU-CS-03-178, August 2003.

[Gertz01] M. Gertz, K. Sattler: A Model and Architecture for Conceptualized Data Annotations. Technical Report, Department of Computer Science, University of California, Davis, 2001.

[Ghemawat03] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google file system. ACM SOSP. Lake George, NY, 10-22 October 2003, 29-43, ISBN 1-58113-757-5.

[Gibson93] Garth A. Gibson, David A. Patterson. Designing Disk Arrays for High Data Reliability. J. Parallel Distrib. Comput. 17(1-2): 4-27 (1993).

[Gibson97] G. A. Gibson, D. F. Nagle, K. Amiri, et. al. , File server scaling with network-attached secure disks, Proceedings of the 1997 ACM SIGMETRICS, p.272-284, June 15-18, 1997, Seattle, WA.

[Gibson98] Garth A. Gibson, David F. Nagle, Khalil Amiri, Jeff Butler, Fay W. Chang, Howard Gobioff, Charles Hardin, Erik Riedel, David Rochberg, Jim Zelenka. A cost-effective, high-bandwidth storage architecture. ACM ASPLOS. San Jose, CA, 3-7 October 1998

[Gibson03] G. A. Gibson, B. B. Welch, D. F. Nagle, B. C. Moxon, Object Storage: Scalable Bandwidth for HPC Clusters, Proc. of the ClusterWorld Conference and Expo June 23-26, 2003, in San Jose, CA.

[Gibson04] G. Gibson, B. Welch, W. Jones, Managing Scalability in Object Storage Systems for HPC Linux Clusters, IEEE MSST04.

[Gobioff99] Howard Gobioff. Security for a High Performance Commodity Storage Subsystem. Carnegie Mellon University Ph.D. Dissertation, July 1999. Also Technical Report CMU-CS-99-160.

[Goodson05] G. Goodson, B. Welch, B. Halevy, D. Black, and A. Adamson, "NFSv4 pNFS Extensions," Internet Draft draft-ietf-nfsv4-pnfs-00.txt (October 2005).

[Gray86] J. Gray. Why do computers stop and what can be done about it. Proc. of the 5th Symposium on Reliability in Distributed Software and Database Systems. 1986.

[Gray90] J. Gray. A census of Tandem system availability between 1985 and 1990. IEEE Transactions on Reliability, V39N4, 1990.

[HDF05] HDF5 User's Guide, http://hdf.ncsa.uiuc.edu/HDF5/doc/UG/, 2005.

[HDF5] http://hdf.ncsa.uiuc.edu/HDF5/PHDF5/parallelhdf5hints.pdf

[HEC05] HEC-IWG File Systems and I/O R&D Workshop 2005, Grapevine, TX. See also http://www.nitrd.gov/subcommittee/hec/workshop/20050816_storage/fiscal.pdf.

[Hildebrand03] D. Hildebrand, P. Honeyman, NFSv4 and High Performance File Systems: Positioning to Scale, NFS Extensions for Parallel Storage Workshop (NEPS), December 2003, Ann Arbor, MI.

[Hildebrand05] D. Hildebrand and P. Honeyman, "Exporting Storage Systems in a Scalable Manner with pNFS," IEEE MSST05, Monterey (April 2005).

[Hildebrand05a] D. Hildebrand and P. Honeyman, "Scaling NFSv4 with Parallel File Systems," in Proc. CCGrid 2005, Cardiff (May 2005).

[Honicky04] R. J. Honicky, Ethan L. Miller. Replication Under Scalable Hashing: A Family of Algorithms for Scalable Decentralized Data Distribution. IPDPS04, April 2004, Santa Fe, NM.

[HoonBaek01] Sung Hoon Baek, Bong Wan Kim, Eui Joung Joung, Chong Won Park. Reliability and performance of hierarchical RAID with multiple controllers. PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing. 2001. ISBN1-58113-383-9, 246-254.

[Hospodor04] Andy Hospodor and Ethan L. Miller, "Interconnection Architectures for High-Performance File Systems," IEEE MSST 2004, College Park, MD, April 2004, pages 273–281.

[Hou97] Robert Y. Hou and Yale N. Patt. Using Non-Volatile Storage to Improve the Reliability of RAID5 Disk Arrays. FTCS '97: Proceedings of the 27th International Symposium on Fault-Tolerant Computing (FTCS '97). 1997. IEEE Computer Society, Washington, DC, USA.

[IBM] IBM Corporation. IceCube—A System Architecture for Storage and Internet Servers. http://www.almaden.ibm.com/StorageSystems/Advanced_Storage_Systems/Intelligent_Bricks/

[IBM01] International Business Machines Corp. Autonomic Computing: IBM's Perspective on the State of Information Technology, October 2001. http://www.research.ibm.com/autonomic/manifesto/.

[IPM06] IPM: Integrated Performance Monitoring. http://www.nersc.gov/projects/ipm, 2006.

[Isaila01] Florin Isaila, Walter F. Tichy, "Clusterfile: A Flexible Physical Layout Parallel File System," p. 37, 3rd IEEE International Conference on Cluster Computing (CLUSTER'01), 2001.

[Iyer86] R. K. Iyer, D. J. Rossetti, M. C. Hsueh. Measurement and modeling of computer reliability as affected by system activity. ACM Trans. Comput. Syst.,V4N3, 1986.

[Kallahalla03] Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, Kevin Fu. Plutus: Scalable Secure File Sharing on Untrusted Storage. FAST03, 2003, San Francisco, CA, 29-42.

[Kalyanakrishnam99] M. Kalyanakrishnam, Z. Kalbarczyk, R. Iyer. Failure Data Analysis of a LAN of Windows NT Based Computers. Proc. of the 18th IEEE Symposium on Reliable Distributed Systems, 1999.

[Kari93] H. H. Kari, H. Saikkonen, F. Lombardi. Detection of Defective Media in Disks. IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems, October 27-29, 1993, 49-55.

[Karlsson04] Magnus Karlsson, Christos Karamanolis, Xiaoyun Zhu. Triage: Performance Isolation and Differentiation for Storage Systems. International Workshop on Quality of Service. Montreal, Canada, 07-09 June 2004. 67-74.

[Keeton98] Keeton, K., Patterson, D.A. and Hellerstein, J.M. "The Intelligent Disk (IDISK): A Revolutionary Approach to Database Computing Infrastructure" White Paper, University of California Berkeley, May 1998.

[Kephart03] Kephart & Chess Jeffrey Kephart and David Chess, "The Vision of Autonomic Computing," Computer, 36, 1, January 2003.

[Khalsa06] S. Khalsa, A. Matthews, G. Gibson, "PaScal – A New Parallel and Scalable Server IO Networking Infrastructure for Supporting Global Storage/File Systems in Large-size Linux Clusters," 25th IEEE Int. Performance Computing and Communications Conference, Pheonix, AZ., April 2006. (to appear).

[Lee96] E. K. Lee, C. A. Thekkath, Petal: distributed virtual disks, ACM ASPLOS, p.84-92, October 01-04, 1996, Cambridge, MA.

[Li03] J. Li, W-k. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, M. Zingale, "Parallel netCDF: A High-Performance Scientific I/O Interface," Proceedings of SC2003, Phoenix, Arizona, USA

[Li04] Jinyuan Li, Maxwell Krohn, David Mazieres, Dennis Shasha. Secure Untrusted Data Repository (SUNDR). USENIX OSDI04, Dec 2004, San Francisco, CA.

[Lin90] T.-T. Y. Lin and D. P. Siewiorek. Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis. IEEE Transactions on Reliability, V39, 1990.

[Lumb03] Christopher R. Lumb, Arif Merchant, Guillermo A. Alvarez. Facade: virtual storage devices with performance guarantees. USENIX FAST. San Francisco, CA, 31 March-02 April 2003. 131-144.

[Lustre02] Cluster File Systems Inc., Lustre: A scalable high-performance file system, lustre.org/documentation.html.

[Meyer88] J. Meyer and L. Wei. Analysis of workload influence on dependability. Proc. International Symposium on Fault-tolerant computing, 1988.

[Miller02] Ethan L. Miller, Darrell D. E. Long, William E. Freeman, Benjamin C. Reed. Strong Security for Network-Attached Storage. USENIX FAST02, Jan 2002, Monterey, CA, 1-3.

[Mohr96] B. Mohr, A.D. Malony, and J.E. Cuny, "TAU: Tuning and Analysis Utilities for Portable Parallel Programming," In Parallel Programming Using C++, G. Wilson, Ed., MIT Press, Cambridge, Massachusetts, 1996.

[MPI94] Message Passing Interface Forum, "MPI: a Message Passing Interface Standard," International Journal of Supercomputing Applications 8, 3/4, Fall/Winter 1994.

[Murphy95] B. Murphy and T. Gent. "Measuring System and Software Reliability using an Automated Data Collection Process." Quality and Reliability Engineering International, V11N5, 1995.

[Nagle04] D. Nagle, D. Serenyi, A. Matthews, The Panasas ActiveScale Storage Cluster – Delivering Scalable High Bandwidth Storage, SC2004, November 6-12, 2004, Pittsburgh, PA, USA

[Narayanan05] Dushyanth Narayanan, Eno Thereska, Anastassia Ailamaki. Continuous Resource Monitoring for Self-predicting DBMS. Proceedings of the 13th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2005), Atlanta, GA, September 25-27, 2005.

[NDMP] http://www.ndmp.org/download/sdk_v4/draft-skardal-ndmp4-04.txt

[Ng94] S. W. Ng. Crosshatch disk array for improved reliability and performance. ACM ISCA,1994, 255-264.

[NITRD] http://www.nitrd.gov/subcommittee/hec/workshop/20050816_storage/fiscal.pdf

[Nurmi05] Daniel Nurmi, John B., R. Wolski. "Modeling Machine Availability in Enterprise and Wide-Area Distributed Computing Environments. Euro-Par'05, 2005.

[Olson05] Christopher A. Olson and Ethan L. Miller. Secure Capabilities for a Petabyte-Scale Object-Based Distributed File System. Proceedings of the 2005 ACM Workshop on Storage Security and Survivability. Nov 2005, Fairfax, Virgina, USA.

[OPEN] http://www.opengroup.org/onlinepubs/9657099/chap5.htm

[Oppenheimer03] D. L. Oppenheimer, A. Ganapathi, D. A. Patterson. "Why Do Internet Services Fail, and What Can Be Done About It?" USENIX Symposium on Internet Technologies and Systems, 2003.

[Pawlowski00] B. Pawlowski, S. Shepler, C. Beame, B. Callaghan, M. Eisler, D. Noveck, D. Robinson, and R. Thurlow, "The NFS Version 4 Protocol," in Proc. 2nd Intl. Conf. on System Administration and Network Engineering, Maastricht (May 2000)

[Prabhakaran05] Vijayan Prabhakaran, Lakshmi N. Bairavasundaram, Nitin Agrawal, Haryadi S. Gunawi, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau. "IRON file systems" SOSP, 2005.

[Rao05] N.S.V. Rao, W.R. Wing, S.M. Carter, and Q. Wu, "UltraScience Net: Network Testbed for Large-Scale Science Applications, IEEE Optical Communications 43:11 (November 2005)

[Riedel00] Erik Riedel, Christos Faloutsos and David Nagle "Active Disk Architecture for Databases" Technical Report CMU-CS-00-145, April 2000.

[Riedel01] Erik Riedel, Christos Faloutsos, Garth A. Gibson and David Nagle "Active Disks for Large-Scale Data Processing" IEEE Computer. June 2001.

[Rodeh03] O. Rodeh, U. Schonfeld, A. Teperman, zFS - A Scalable distributed File System using Object Disks, MSST, 2003.

[Sahoo04] R. K. Sahoo, A. Sivasubramaniam, M. S. Squillante, Y. Zhang. "Failure Data Analysis of a Large-Scale Heterogeneous Server Environment". Proc. of the 2004 international Conference on Dependable Systems and Networks (DSN'04), 2004.

[Saito04] Yasushi Saito, Svend Frund, Alistair Veitch, Arif Merchant,Susan Spence, FAB: building distributed enterprise disk arrays from commodity components, ASPLOS, Oct 07-13, 2004, Boston, MA, USA.

[Salmon03] B. Salmon, E. Thereska, C. A. N. Soules, and G. R. Ganger. A two-tiered software architecture for automated tuning of disk layouts. Algorithms and Architectures for Self-Managing Systems (San Diego, CA, 11 June 2003), pages 13–18. ACM, 2003.

[Schneider90] Fred B. Schneider. Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Comput. Surv., V22N4, 1990, 299-319.

[Schwarz04] Thomas J. E. Schwarz, Qin Xin, Ethan L. Miller, Darrell D. E. Long, Andy Hospodor, Spencer W. Ng. Disk Scrubbing in Large Archival Storage Systems. MASCOTS 2004, 409-418.

[SPEC05] Standard Performance Evaluation Corporation. SPEC's Benchmarks and Published Results. http://www.spec.org/benchmarks.html.

[SPEC97] SPEC SFS (System File Server) benchmark: SFS97: http://www.spec.org/sfs97r1.

[SGPFS99] ASCI/NSA Scalable Global Parallel File System (SGPFS) Workshop, Santa Fe, NM, Sept 23-24, 1999, www.lanl.gov/asci/sgpfs/kickoff-workshop.html

[Talagala99] N. Talagala and D. Patterson. "An analysis of error behaviour in a large storage system". IEEE Workshop on Fault Tolerance in Parallel and Distributed Systems. 1999.

[Tang90] D. Tang and R. K. Iyer and S. S. Subramani. "Failure analysis and modelling of a VAX cluster system", Proc. International Symposium on Fault-tolerant computing, 1990.

[Thekkath97] Chandramohan A. Thekkath, Timothy Mann, Edward K. Lee, Frangipani: a scalable distributed file system, ACM SOSP, p.224-237, October 05-08, 1997, Saint Malo, France.

[Thereska04] Eno Thereska, Jiri Schindler, John Bucy, Brandon Salmon, Christopher R. Lumb, Gregory R. Ganger. A Framework for Building Unobtrusive Disk Maintenance Applications. FAST '04. San Francisco, CA. March 31, 2004.

[Thereska05] Eno Thereska, Dushyanth Narayanan, Gregory R. Ganger. Towards self-predicting systems: What if you could ask "what-if"? 3rd International Workshop on Self-adaptive and Autonomic Computing Systems. Copenhagen, Denmark, August 2005.

[UNIX] http://www.unix.org/version3/apis.html

[Uttamchandani04] Sandeep Uttamchandani et al. Polus: Growing storage QoS managment beyond a four-year old kid," FAST '04 (April 2004)

[Van Meter98] R. Van Meter, G. G. Finn, S. Hotz, VISA: Netstation's virtual Internet SCSI adapter, ASPLOS, Oct 1998.

[Vetter01] J.S. Vetter and M.O. McCracken, "Statistical Scalability Analysis of Communication Operations in Distributed Applications," Proceedings of the 2001 ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming (PPoPP'01), Snowbird, Utah, USA, pp. 123-132.

[Wang04] Feng Wang, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, OBFS: A File System for Object-Based Storage Devices, MSST04.

[Weil06] S. Weil, S. A. Brandt, E. L. Miller, and C. Maltzahn, CRUSH tech report, 2006.

[Wu04] J. Wu and S. Brandt. "Storage Access Support for Soft Real-Time Applications", 10th IEEE Real-Time and Embedded Technology and Applications Symposium RTAS '04), Toronto, Canada, May 2004. 108

[Wu06] Joel C. Wu and Scott A. Brandt, "The Design and Implementation of AQuA: an Adaptive Quality of Service Aware Object-Based Storage Device," IEEE MSST 2006, May 2006, to appear.

[Wylie00] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, and P. K. Khosla. Survivable information storage systems. IEEE Computer, 33(8):61–68. IEEE, August 2000.

[Xin03] Qin Xin, Ethan L. Miller, Thomas Schwarz, Scott A. Brandt, Darrell D. E. Long, and Witold Litwin, "Reliability Mechanisms for Very Large Storage Systems," IEEE MSST 2003, San Diego, CA, April 2003, pages 146–156.

[Xin04] Qin Xin, Ethan L. Miller, and Thomas J. E. Schwarz, "Evaluation of Distributed Recovery in Large-Scale Storage Systems," Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13), Honolulu, HI, June 2004, pages 172–181.

[Xin05a] Qin Xin, Thomas J. E. Schwarz, and Ethan L. Miller, "Disk Infant Mortality in Large Storage Systems," Proceedings of the 13th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2005), Atlanta, GA, September 2005.

[Xin05b] Qin Xin, Ethan L. Miller, Thomas J. E. Schwarz, and Darrell D. E. Long, "Impact of Failure on Interconnection Networks in Large Storage Systems," IEEE MSST 2005, Monterey, CA, April 2005.

[Xu99] J. Xu and Z. Kalbarczyk and R. K. Iyer. "Networked Windows NT System Field Failure Data Analysis". Proc. of the 1999 Pacific Rim International Symposium on Dependable Computing,1999.

[Yu04] Haifeng Yu. "Signed quorum systems". PODC 2004: 246-255.

[Zhang06] J. Zhang and P. Honeyman, "Naming, Migration, and Replication for NFSv4," to appear in Proc. 5th Intl. Conf. on System Administration and Network Engineering, Delft (May 2006).

Appendix A, Part 2: University of Michigan Statement of Work

A. University of Michigan (CITI) summary of tasks and milestones

The Petascale Data Storage Institute is a proposal for a five-year distributed institute from CMU (lead institution), UCSC, U. Michigan, NERSC, PNNL, ORNL, SNL, and LANL.

This distributed institution will draw on the expertise of its member institutions. For the proposed five-year SciDAC2 Petascale Data Storage Institute, the distributed institution will draw on the University of Michigan (CITI) for in-depth knowledge of Internet middleware and expertise in building prototype and production software. As principal developer and steward for the Linux-based, open-source reference implementation of NFSv4, CITI is at the vanguard of the development of NFSv4 protocol extensions for network transparency in petascale data storage, including global naming, agile credential management, transparent migration, consistent replication, and flexible monitoring. Near- and intermediate-term requirements for global access to petascale data will demand leadership and consensus among the producers, consumers, and stewards of petascale data. CITI will apply its twenty years of experience building research and development partnerships with academic and scientific institutions to help achieve this consensus.

CITI is committed to open source development. Institute activities are collaborative with other member institutions and all tools, data, and results will be shared with institute partners, SciDAC centers for technology, SciDAC application researchers, HEC/URA/NSF I/O researchers, and other researchers without restriction.

B. CITI tasks and milestones

Level of effort is 20% of Peter Honeyman, 35% of William A. (Andy) Adamson, 40% of J. Bruce Fields, and 100% of 1 Graduate Student Research Assistant.

Project 1: Petascale Data Storage Outreach

- Years 1–5: Participate in curriculum development and workshop organization for petascale storage developers and stakeholders.
- Years 1-5: Organize and participate in annual workshops held in conjunction with Supercomputing, Global Grid Forum, CCGrid, USENIX FAST conferences; publish reports, papers, and meeting results online

Project 2: Petascale Storage Application Performance Characterization

- Year 1-3: Assist in development and application of trace analysis tools; assist with trace format refinement.
- Year 4-5: Develop and support tools to manage, customize, and distribute large trace collections.

Project 4: Protocol/API extensions for Petascale Science Requirements

- Years 1-3: Develop experimental applications to validate interfaces for supporting higher data rates and non-sequential data accesses, assist with refinement of advanced protocols and APIS for supporting petascale data storage.
- Years 3-5: Develop and distribute reference implementations and evaluation of proposed POSIX API extensions for parallel science applications.

Project 5: Exploration of Novel Mechanisms for Emerging Petascale Science Requirements

- Years 1-3: Investigate and characterize data production and consumption patterns of petascale computing applications. Develop pNFS-based data management, placement, and replication strategies for petascale data storage.
- Years 3-5: Develop algorithms for automated storage placement in petascale computations, integrating cost, performance, priority, authorization, and resource availability.
- Years 1-3: Investigate structural properties of collaborating groups that affect virtual organization construction and management. Develop tools for virtual organization formation, management, and dissolution.
- Years 3-5: Refine tools and portals for controlled resource sharing within virtual organizations.

Project 6: Exploration of Automation for Petascale Storage System Management

- Year 1-3: Explore resource costs and trade-offs in replicating petascale data for automated failure handling.
- Year 3-5: Develop tools for automated replication of petascale data. Explore failover strategies that build on these tools.
- Year 2-5: Develop and experiment with approaches to automating aspects of configuration and tuning for petascale file systems, including automatic data placement and reorganization as applications requirements and access patterns change.
- Year 3-5: Develop and experiment with approaches to automating aspects of performance problem and failure diagnosis in petascale file systems.