Final report for DE-FC02-10ER26033; SC0005428 "Adding Data Management Services to Parallel File Systems"
PI: Scott Brandt, University of California, Santa Cruz

1. DOE award # and name of the recipient (Institution)

DOE Award #: DE-FC02-10ER26033; SC0005428
***Recipient*:** University of California, Santa Cruz

2. Project Title and name of the PI

***Project Title*:** "Adding Data Management Services to Parallel File Systems,"
***PI*:** Scott Brandt

3. Date of the report and period covered by the report

***Report date*:** 3/4/2015
***Period covered by the report*:** 9/1/2010 to 8/31/2014 (including the no-cost time extension of one year)

4. A brief description of the progress/accomplishments during the funding periods.

Abstract: The objective of this project, called DAMASC for "Data Management in Scientific Computing", is to coalesce data management with parallel file system management to present a declarative interface to scientists for managing, querying, and analyzing extremely large data sets efficiently and predictably. Managing extremely large data sets is a key challenge of exascale computing. The overhead, energy, and cost of moving massive volumes of data demand designs where computation is close to storage. In current architectures, compute/analysis clusters access data in a physically separate parallel file system and largely leave it scientist to reduce data movement.

Over the past decades the high-end computing community has adopted middleware with multiple layers of abstractions and specialized file formats such as NetCDF-4 and HDF5. These abstractions provide a limited set of high-level data processing functions, but have inherent functionality and performance limitations: middleware that provides access to the highly structured contents of scientific data files stored in the (unstructured) file systems can only optimize to the extent that file system interfaces permit; the highly structured formats of these files often impedes native file system performance optimizations.

We are developing Damasc, an enhanced high-performance file system with native rich data management services. Damasc will enable efficient queries and updates over files stored in their native byte-stream format while retaining the inherent performance of file system data storage via declarative queries and updates over views of underlying files.

Damasc has four key benefits for the development of data-intensive scientific code: (1) applications can use important data-management services, such as declarative queries, views, and provenance tracking, that are currently available only within database systems; (2) the use of

these services becomes easier, as they are provided within a familiar file-based ecosystem; (3) common optimizations, e.g., indexing and caching, are readily supported across several file formats, avoiding effort duplication; and (4) performance improves significantly, as data processing is integrated more tightly with data storage.

Our key contributions are: SciHadoop which explores changes to MapReduce assumption by taking advantage of semantics of structured data while preserving MapReduce's failure and resource management; DataMods which extends common abstractions of parallel file systems so they become programmable such that they can be extended to natively support a variety of data models and can be hooked into emerging distributed runtimes such as Stanford's Legion; and Miso which combines Hadoop and relational data warehousing to minimize time to insight, taking into account the overhead of ingesting data into data warehousing.

FY11: This has been a productive year for the Damasc project. We have quickly ramped up the project team to include four faculty, 1 formal senior collaborator at Livermore National Laboratory, 1 post-doctoral scholar, four funded Ph.D. students, two additional synergistically funded Ph.D. students at Los Alamos National Laboratory and Livermore National Laboratory, and a large number of other collaborators (as detailed below). Our research efforts have already begun bearing fruit, with five publications submitted and/or accepted for publication and several others in the pipeline. Our early results have generated strong interest in the community, especially SciHadoop. We are actively pursuing a number of collaborations with other researchers, including synergistic research with other DOE-funded projects.

FY12: Similar to last year, the project team includes four faculty, 1 formal senior collaborator at Livermore National Laboratory, 1 post-doctoral scholar, four funded Ph.D. students, one additional synergistically funded Ph.D. student at Los Alamos National Laboratory, and a large number of other collaborators (as detailed below). Our research efforts have continued, with five publications submitted and/or accepted for publication and several others in the pipeline. Our early results have generated strong interest in the community, especially SciHadoop. We are actively pursuing a number of collaborations with other researchers, including synergistic research with other DOE-funded projects.

FY13-14: Similar to the other years, the project team includes four faculty, 1 formal senior collaborator at Livermore National Laboratory, 1 post-doctoral scholar, four funded Ph.D. students, one additional synergistically funded Ph.D. student at Los Alamos National Laboratory, and a large number of other collaborators (as detailed below). Our research efforts have continued, with 20 publications accepted for publication and several others in the pipeline. Our early results have generated strong interest in the community, especially Miso, SIDR, and SciHadoop. We are actively pursuing a number of collaborations with other researchers, including synergistic research with other DOE-funded projects. The following sections discuss project staffing, results, and progress and plans.

A. Staffing:

FY11: The DAMASC project team includes four faculty, 1 senior collaborator at LLNL, 1 post-doctoral scholar at LLNL, four funded Ph.D. students, two additional synergistically funded Ph.D. students (one at LANL), and other collaborators detailed below.

FY12: During this year there were minor changes in our team: The post-doctoral scholar is funded elsewhere since spring 2012 but continues to collaborate with us on this project, and the synergistically funded Ph.D. student at Livermore graduated fall 2011 and is now funded as post-doctoral scholar by the LLNL portion of this project.

FY13-14: During these years there were the following changes in our team: two of the Ph.D. students graduated, Joe Buck and Dimitrios Skourtis.

i. Senior Personnel:

1. Prof. Scott Brandt (PI), Professor, Computer Science, UC Santa Cruz

2. Prof. Carlos Maltzahn (Co-PI), Adjunct Professor, Computer Science, UC Santa Cruz

3. Prof. Neoklis Polyzotis (Co-PI), Associate Professor, Computer Science, UC Santa Cruz

4. Prof. Wang-Chiew Tan (Co-PI), Associate Professor, Computer Science, UC Santa Cruz

5. Dr. Maya Ghokale (LLNL PI), Computer Scientist, Center for Applied Scientific Computing, LLNL

6. Dr. Kleoni Ioannidou, Post-Doctoral Scholar, Computer Science, UC Santa Cruz.
   Research: Theoretical aspects of divergent indexing and parallel query execution/scheduling. Dr. Ioannidou left this position in spring 2012 to pursue a career in industry but will continue to collaborate with us.

7. Dr. Sasha Ames (at and funded by LLNL), Post-Doctoral Scholar, Center for Applied Scientific Computing, LLNL.
   Research: File system metadata management and querying. Genome database indexing.

ii. Ph.D. Students:

The DAMASC team includes 6 Ph.D. students.

FY11: Four are funded by the DAMASC project (#2,3,5,7 below) and two are funded by synergistic funding sources (#1,4).

FY12: Four are funded by the DAMASC project (#2,3,5,7 below) and two are funded by synergistic funding sources (#1,4). In spring 2012 we started to involve Adam Crume in SciHadoop focusing on semantic compression, and we started to involve Noah Watkins in programmable file systems (details below).

FY13-14: Three are funded by the DAMASC project (#2,3,6 below) and three are funded by synergistic funding sources (#4,5,7). Noah Watkins received synergistic funding from the LANL/UCSC Institute for Scalable Scientific Data Management (ISSDM) and Jeff LeFevre received synergistic funding from industry. We started to invole Dimitrios Skourtis predictable performance. In fall 2012 Adam Crume started working on behavioral modeling of storage

devices, a topic closer to his original topic of modeling parallel file systems. We requested to extend the scope of the Damasc project to include this line of research. This request was granted by Lucy Nowell per email on 10/18/2012.

1. Sasha Ames (at and funded by LLNL)
   Research: File system metadata management and querying
   Dr. Ames graduated in fall 2011 and joined this project's senior personnel (see above).

2. Joe Buck (at UCSC)
   Research: Query execution and operator optimization.

3. Adam Crume (at UCSC)
   Research: Behavioral modeling of storage devices

4. Latchesar Ionkov (at and funded by LANL)
   Research: Specification and scalable implementation of a scientific data transformation language

5. Jeff LeFevre (at UCSC)
   Research: Automatic indexing tuning, divergent indexing of replica, opportunistic design

6. Dimitrios Skourtis (at UCSC)
   Research: Predictable performance, especially for flash

7. Noah Watkins (at UCSC)
   Research: Parallel query execution and scheduling and Programmable file systems.

B. Results:

Our research efforts have been very productive, with 32 papers accepted for publications and several more in various stages of preparation. Our results have generated strong interest in the community, especially SciHadoop, DataMods, and Miso (details below).

i. Publications and Submissions

Thirtyone papers (#2-32) have been accepted for publication in the period covered by this report. Several others are in the pipeline and will be submitted in the coming months.

1. Scott Brandt, Carlos Maltzahn, Neoklis Polyzotis, and Wang-Chiew Tan, "Fusing Data Management Services with File Systems", Petascale Data Storage Workshop (PDSW), 2009. Based on work in our proposal, but presented before the start of the award.

2. Debabrata Dash, Neoklis Polyzotis, and Anastasia Ailamaki, "CoPhy: A Scalable, Portable, and Interactive Index Advisor for Large Workloads," PVLDB 4(6): 362-372 (2011).

3. Bogdan Alexe, Balder ten Cate, Phokion G. Kolaitis, and Wang Chiew Tan, “Eirene: Interactive Design and Refinement of Schema Mappings via Data Examples,” PVLDB 4(12): 1414-1417 (2011)

4. Bogdan Alexe, Mauricio A. Hernaandez, Lucian Popa, and Wang Chiew Tan, “MapMerge: Correlating Independent Schema Mappings,” PVLDB 3(2): 81-92 (2010).

5. Joe Buck, Noah Watkins, Jeff LeFevre, Kleoni Iouannidou, Carlos Maltzahn, and Neoklis Polyzotis, and Scott Brandt, "SciHadoop: Array-based Query Processing in Hadoop," SC11, Seattle, WA, November 2011

6. Adam Crume, Carlos Maltzahn, Jason Cope, Sam Lang, Ross Rob, Phil Carns, Chris Carothers, Ning Liu, Curtis Janssen, John Bent, et al., “FLAMBES: Evolving Fast Performance Models,” Poster at SC11, Seattle, WA, November 2011

7. Brian Van Essen, Roger Pearce, Sasha Ames, and Maya Gokhale, “On the role of NVRAM in data intensive HPC architectures,” Workshop on Emerging Supercomputing Technologies (WEST) at ICS 2011, Tucson, AZ, May 2011

8. Ning Liu, Christopher Carothers, Jason Cope, Philip Carns, Robert Ross, Adam Crume, and Carlos Maltzahn, “Modeling a Leadership-scale Storage System,” 9th International Conference on Parallel Processing and Applied Mathematics (PPAM 2011), Torun, Poland, September 2011

9. Ning Liu, Jason Cope, Philip Carns, Christopher Carothers, Robert Ross, Gary Grider, Adam Crume, and Carlos Maltzahn, “On the Role of Burst Buffers in Leadership-class Storage Systems,” 28th IEEE Conference on Massive Data Storage (MSST 2012), Pacific Grove, CA, April 2012

10. Karl Schnaitter and Neoklis Polyzotis, “Semi-Automatic Index Tuning: Keeping DBAs in the Loop,” PVLDB 5(6): 478-489 (2012)

11. Mariano P. Consens, Kleoni Ioannidou, Jeff Lefevre, and Neoklis Polyzotis, “Divergent Physical Design Tuning for Replicated Databases,” 2012 ACM SIGMOD, Scottsdale, AZ, May 2012

12. Ivo Jimenez, Huascar Sanchez, Quoc Trung Tran, and Neoklis Polyzotis, “Kaizen: A Semi-Automatic Index Advisor (DEMO)”, 2012 ACM SIGMOD, Scottsdale, AZ, May 2012

13. Jun He, John Bent, Aaron Torres, Gary Grider, Garth Gibson, Carlos Maltzahn, and X.-H. Sun, “Discovering structure in unstructured I/O,” 7th Parallel Data Storage Workshop at Supercomputing 12 (PDSW 2012), Salt Lake City, UT, November 12, 2012.

14. Noah Watkins, Carlos Maltzahn, Adam Manzanares, Scott Brandt, “Datamods: Programmable file system services,” 7th Parallel Data Storage Workshop at

Supercomputing (PDSW 2012), Salt Lake City, UT, November 12, 2012.

15. Adam Crume, Joe Buck, Carlos Maltzahn, Scott Brandt, “Compressing intermediate keys between mappers and reducers in SciHadoop,” 7th Parallel Data Storage Workshop at Supercomputing 12 (PDSW 2012), Salt Lake City, UT, November 12, 2012.

16. Jun He, John Bent, Aaron Torres, Gary Grider, Garth Gibson, Carlos Maltzahn, Xian-He Sun, “I/O Acceleration with Pattern Detection,” 22nd International ACM Symposium on High Performance Parallel and Distributed Computing (HPDC’13), New York City, NY, June 17-22, 2013.

17. Jeff LeFevre, Jagan Sankaranarayanan, Hakan Hacıgümüs, Junichi Tatemura, and Neoklis Polyzotis. Towards a workload for evolutionary analytics. In SIGMOD Workshop on Data Analytics in the Cloud (DanaC), 2013. Extended version CoRR abs/1304.1838.

18. Latchesar Ionkov, Mike Lang, Carlos Maltzahn, “DRepl: Optimizing Access to Application Data for Analysis and Visualization,” 29th IEEE Symposium on Massive Storage Systems and Technologies - Research Track (MSST 2013), Long Beach, CA, May 6-10, 2013.

19. Sasha K. Ames, David A. Hysom, Shea N. Gardner, G. Scott Lloyd, Maya B. Gokhale, Jonathan E. Allen. "Scalable metagenomic taxonomy classification using a reference genome database.” Bioinformatics, 29 (18) July 2013, 2253-2260.

20. Brian Van Essen, Henry Hsieh, Sasha Ames, Roger Pearce, Maya Gokhale. “DI-MMAP—a scalable memory-map runtime for out-of-core data-intensive applications.” Cluster Computing, October 2013

21. Joe Buck, Noah Watkins, Greg Levin, Adam Crume, Kleoni Ioannidou, Scott Brandt, Carlos Maltzahn, Neoklis Polyzotis, and Aaron Torres, “SIDR: Structure-Aware Intelligent Data Routing in Hadoop,” in SC ’13, (Denver, CO), November 2013.

22. Noah Watkins, Carlos Maltzahn, Scott Brandt, Ian Pye, and Adam Manzanares, “In-vivo storage system development,” in BigDataCloud ’13 (in conjunction with EuroPar 2013), (Aachen, Germany), August 26, 2013.

23. Dimitrios Skourtis, Dimitris Achlioptas, Carlos Maltzahn, and Scott Brandt, “High Performance & Low Latency in Solid-State Drives Through Redundancy,” in INFLOW 2013 (in conjunctions with SOSP’13), Farmington, PA, November 3, 2013.

24. Adam Crume, Carlos Maltzahn, Lee Ward, Thomas Kroeger, Matthew Curry, Ron Oldfield and Patrick Widener, “Fourier-Assisted Machine Learning of Hard Disk Drive Access Time Models,” 8th Parallel Data Storage Workshop at Supercomputing 13 (PDSW 2013), Denver, CO, November 18, 2013.

25. Jay Lofstead, Jai Dayal, Ivo Jimenez, and Carlos Maltzahn, “Efficient Transactions for Parallel Data Movement,” 8th Parallel Data Storage Workshop at Supercomputing ’13 (PDSW 2013), Denver, CO, November 18, 2013.

26. Sasha Ames, Jonathan E. Allen, David A. Hysom, G. Scott Lloyd and Maya B. Gokhale. "Design and Optimization of a Metagenomics Analysis Workflow for NVRAM." 13th IEEE International Workshop on High Performance Computational Biology. May 2014

27. Dimitris Skourtis, Dimitris Achlioptas, Noah Watkins, Carlos Maltzahn, Scott Brandt, “Flash on Rails: Consistent Flash Performance through Redundancy,” USENIX Annual Technical Conference (ATC’14), Philadelphia, PA, June 19-20, 2014.

28. Adam Crume, Carlos Maltzahn, Lee Ward, Thomas Kroeger, Matthew Curry, “Automatic Generation of Behavioral Hard Disk Drive Access Time Models,” 30th International Conference on Massive Storage Systems and Technology (MSST 2014), Santa Clara, CA, June 2-6, 2014.

29. Jeff LeFevre, Jagan Sankaranarayanan, Hakan Hacıgümüs, Junichi Tatemura, Neoklis Polyzotis, and M. J. Carey. Opportunistic physical design for big data analytics. In SIGMOD, Snowbird, UT, 2014.

30. Jeff LeFevre, Jagan Sankaranarayanan, Hakan Hacıgümüs, Junichi Tatemura, Neoklis Polyzotis, Michael J. Carey. MISO: Souping Up Big Data Query Processing with a Multistore System. In SIGMOD, Snowbird, UT, 2014.

31. Dimitrios Skourtis, Dimitris Achlioptas, Noah Watkins, Carlos Maltzahn, Scott Brandt, “Erasure Coding & Read/Write Separation in Flash Storage,” 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads (INFLOW ’14) (Workshop co-located with OSDI 2014), Broomfield, CO, October 5, 2014.

32. Jay Lofstead, Jai Dayal, Ivo Jimenez, Carlos Maltzahn, “Efficient, Failure Resilient Transactions for Parallel and Distributed Computing,” The 2014 International Workshop on Data-Intensive Scalable Computing Systems (DISCS-2014) (Workshop co-located with Supercomputing 2014), New Orleans, LA, November 16, 2014.

ii. Outreach and Collaboration:

We are actively pursuing a number of collaborations with other researchers, including synergistic research with other DOE-funded projects. We are in active discussions and/or collaboration with researchers at the LANL Ultrascale Systems Research Center, SLAC (Jacek Becla), LBL (Shane Canon and Kesheng Wu), the HDF5 Group (Quincey Koziol), ANL (CODES project with Rob Ross, Phil Carns, Jason Cope, Tom Peterka), and RPI (Chris Carothers), and SNL (Lee Ward, Jay Lofstead, Ron Oldfield, Matthew Curry, Patrick Widener). In addition, our work on SciHadoop has generated very strong interest at IBM, Cloudera, Yahoo, Greenplum, SLAC, LBL, LANL (USRC), LLNL (FOX project and climate group), ORNL, and MIT.

iii. Other:

DAMASC team members participated in a number of synergistic activities during the past two years, as follows:

- PIs Brandt, Maltzahn, and Gokhale participated in the DOE PI meeting, ASCR 2011 (San Diego)
- PI Maltzahn Chaired the Petascale Data Storage Workshop (PDSW 2010).
- PI Polyzotis co-Chaired the 6th International Workshop on Self-Managing Database Systems (SMDB 2011), held in conjunction with the IEEE International Conference on Data Engineering (ICDE).
- PI Polyzotis gave invited talks on Damasc at Oracle and IBM Almaden.
- PI Tan attended Very Large Data Bases, 2011.
- PIs Brandt, Maltzahn, and Gokhale participated in the DOE PI meeting, ASCR 2012 (Portland, OR)
- PI Maltzahn was General Co-Chair (with Garth Gibson, CMU) of the 6th Parallel Data Storage Workshop (PDSW 2011 at SC11)
- PI Maltzahn gave invited talks on DAMASC at HEC FSIO 2011 (panel on Next Gen IO), Cluster 2011 (panel on Storage in HPC), HPDC Trends Workshop 2012, DIDC 2012 (at HPDC 2012, panel on Rethinking Clusters and File systems for Data Intensive Computing).
- Students Buck and Watkins presented invited talks about SciHadoop at Cloudera and Greenplum.
- Student Buck published SciHadoop as open source patch for Hadoop.
- PI Polyzotis co-Chaired the 6th International Workshop on Self-Managing Database Systems (SMDB 2011), held in conjunction with the IEEE International Conference on Data Engineering (ICDE).
- PI Polyzotis gave an invited talk on DAMASC at SAP.
- PIs Polyzotis and Tan attended Very Large Data Bases, 2011.
- co-PI Maltzahn gave invited talks on DAMASC at DIDC 2012 (Panel on "Rethinking Clusters and Filesystems for Data Intensive Computing"), at HPDC 2012 (Panel on "How big is your 'Big Data', and how can HPDC help?"), at National Center for Atmospheric Research (NCAR) (presentation on "Programmable file system services and in-vivo storage system development"), at ORNL and LANL (presentation on "Programmable Storage Systems"), and at Silicon Valley Space Business Roundtable (SVSBR) (panel on "Big Data's Next Big Move").
- co-PI Maltzahn organized the 1st Programmable File Systems Workshop (PFSW at HPDC 2014).

C. Progress and Plans:

i. Query execution and operator optimization (student: Joe Buck)

SciHadoop serves as a research platform to investigate the costs and benefits of structure-awareness in query execution and operator optimization in scientific data management. Last year we focused on optimization of mappers. One of the side effects of these investigations was the insight that the largest processing and communication overhead – and therefore most of the

potential for optimization – is actually between mappers and reducers: we found that our optimization that moves holistic functions from reducers to mappers had orders of magnitude larger impact on performance than any of our other mapper optimizations. We are now addressing these findings by using semantic compression, i.e. compression that takes advantage of the structure of data and by separating information that is known at task scheduling time from information that is produced at task run time. In particular, we are (1) creating a tighter relationship between mappers and reducers increasing the context that any mapper/reducer pair can share at scheduling time, and (2) using that a priori context to minimize the amount of data any mapper/reducer pair needs to communicate. A beneficial side effect of (1) is that reducers have much fewer dependencies to mappers and can therefore provide their results much earlier than in standard Hadoop. This is particularly advantageous when SciHadoop is used in the context of a continual stream of data.

We published these results at SC'11 ("SciHadoop: Array-based Query Processing in Hadoop"), PDSW'12 at SC'12 ("Compressing Intermediate Keys between Mappers and Reducers in SciHadoop", and SC'13 ("SIDR: Structure-Aware Intelligent Data Routing in Hadoop"). Joe Buck graduated with a Ph.D. in May 2014 and now works for Amazon.com, Inc.

ii. Scalable parallel file system simulator (student: Adam Crume)

A key accomplishment of this work is the implementation of a fast disk simulator that is about 300 times faster than DiskSim at an error rate less than 5% with a memory footprint of just a few bytes. This would either allow the simulation of systems with a million storage devices or the efficient parallel simulation of smaller systems over a large parameter space. Simulations of these kinds are very useful for investigating new data management system architectures. The fast disk simulator is mostly an analytical function combined with a small amount of state (much reduced compared to DiskSim). The analytical function fits a particular disk model and is obtained using a genetic algorithm approach—the genetic algorithm approach is necessary because it turns out that error minimization curves have a lot of local minima and maxima. Once fitted, the fast disk simulator can be used to simulate thousands of disks with little overhead. Adam also started working on fast RAID-5 simulation using a similar approach.

In Fall 2012 we hit a roadblock that so far we were not able to overcome: while it was easy to get an *average* 5% error rate for a particular workload, it turns out be very difficult to achieve a similar fidelity on a request-by-request level. The key problem is that disks reorder requests internally depending on proprietary topologies of ever increasing complexity.

After spending months working with machine learning and statistician experts, co-PI Maltzahn started working with SNL (Lee Ward, Ron Oldfield, Matthew Curry, and Patrick Widener) and was strongly encouraged by Lee to continue work on behavioral modeling of disk drives. In October 2012 permission was given to extend the scope of the DAMASC project so we can fund Adam Crume to work on this topic. While collaborating with the SNL team, he achieved a major breakthrough by (1) initially limiting the modeling to co-located tracks in an area of 1% of the disk's total area, and (2) taking phase information of major frequencies into account. Major frequencies are discovered by frequency analysis of the block access traces. Phase information is particularly beneficial because it turns out that machine learning approaches are generally very weak in identifying periodic behavior.

Adam published two papers, one at PDSW'13 at SC'13 and another at MSST'14, and passed his thesis proposal in fall 2014. Most recently we discovered a way to extract important frequencies directly from the traces without separate frequency analysis. The direct method is faster and more reliable. A publication on this is underway. We also started collaborating with a new UCSC faculty, Professor Seshadhri, a theoretical computer scientist who specializes on property checking and sublinear algorithms. Adam Crume's project can be viewed as discovering structure in a very large space through extremely sparse sampling. We hope to gain insights on the theoretical limits of our approach. Adam plans to defend in spring or summer 2015.

iii. Predictable performance (student: Dimitrios Skourtis)

An important aspect of data management in file systems is performance predictability. This becomes particularly important when extending file system functionality with various data management functionalities. To achieve predictable performance without over-provisioning, a system must provide isolation between processes. Throughput-based reservations are challenging due to the mix of workloads and the stateful nature of disk drives, leading to low reservable throughput, while existing utilization-based solutions require specialized I/O scheduling for each device in the storage system.

At the same time, virtualization and many other applications such as online analytics and transaction processing often require access to predictable, low-latency storage. Hard-drives have low and unpredictable performance under random workloads, while keeping everything in DRAM, in many cases, is still prohibitively expensive or unnecessary. Solid-state drives offer a balance between performance and cost, and are becoming increasingly popular in storage systems, playing the role of large caches and permanent storage. Although their read performance is high and predictable, SSDs frequently block in the presence of writes, exceeding hard-drive latency and leading to unpredictable performance. Many systems with mixed workloads have low latency requirements or require predictable performance and guarantees. In such cases the performance variance of SSDs becomes a problem for both predictability and raw performance.

We introduced Rails, a flash storage system based on redundancy, which provides predictable performance and low latency for reads under read/write workloads by physically separating reads from writes. More specifically, reads achieve read-only performance while writes perform at least as well as before. We evaluate our design using micro-benchmarks and real traces, illustrating the performance benefits of Rails and read/write separation in flash.

We also introduced eRails, a scalable flash storage system on top of Rails that achieves read/write separation using erasure coding without the storage cost of replication. To support an arbitrary number of drives efficiently we describe a design allowing us to scale eRails by constructing overlapping erasure coding groups that preserve read/write separation. Through benchmarks we demonstrate that eRails achieves read/write sep- aration and consistent read performance under read/write workloads. We demonstrate that the guaranteeable performance in SSDs is low without consistent performance. Using Rails and time-based scheduling we demonstrate that we can achieve performance guarantees that are close to optimal.

Dimitrios Skourtis published his work at INFLOW 2013 (co-located with SOSP 2013) ("High Performance & Low Latency in Solid-State Drives Through Redundancy"), USENIX ATC 2014 ("Flash on Rails: Consistent Flash Performance through Redundancy"), and INFLOW 2014 (co-located with OSDI 2014) ("ErasureCod- ing & Read/Write Separation in Flash Storage"). Dr. Skourtis graduated in September 2014.

iv. Specification and scalable implementation of a scientific data transformation language (student: Latechsar Ionkov)

This task is focused on creating efficient languages for specifying data replicas with different layouts and for specifying views over replicas. The intent is to optimize performance by using data-safety replicas to store data in different access-specific formats, allowing different accesses to use whichever replica is organized for that type of access. Additional replicas, views, may provide specialized access to some or all of the data for performance reasons. These replicas are not required for data safety, but are created purely for performance reasons. As a result, they may be incomplete copies of the data and may be the result of some degree of processing, in addition to reorganization.

Latchesar has implemented a specification language that is very general. The implementation has uncovered a number of complexities, e.g. the language currently allows the specification of data structures with arbitrarily complex substructures, a feature that turns out to greatly complicate implementation. In all cases, these complexities can be avoided by limiting the expressivity of the language. We are collaborating with science application stakeholders within LANL to either find use cases justifying complex language features or making the case of limiting language expressivity. At the same time, we are ensuring that the language is expressive enough that it enables the implementation of data transformation successes such as LANL's Parallel Log-structured File System (PLFS). Latchesar published this work at MSST 2013 and passed his thesis proposal in fall 2014.

v. Divergent index tuning for query processing (student: Jeff LeFevre)

The purpose of this task is to leverage data replication, which is applied in distributed systems for fault tolerance, in order to specialize replicas for different subsets of application workloads. For the time being, we are examining a specific variant of the problem where the data is stored in a relational database system, the workload consists of SQL queries and updates, and the specialization is achieved by installing the right set of data indexes on each replica. We have developed an algorithm that tries to find the best specialization for each replica, while ensuring that the system can remain load balanced up to some configurable degree, set by the system administrator. One interesting feature of the algorithm is that it is somewhat agnostic to the specialization scheme. For instance, it can work equally well when the data is stored in format-specific files, e.g., NetCDF or HDF5, the queries come through the corresponding library, and the specialization is achieved by reorganizing the physical storage in each replica, e.g., storing the data in row- or column-major order. Experimental results using a replicated database on Amazon EC2 have shown that significant performance gains. The results of this work have been published in the proceedings of the 2012 ACM SIGMOD International Conference on

Management of Data.

We then examined the integration of this scheme in the SciHadoop project, to optimize the physical layout of file data replicated in HDFS and queried by SciHadoop. This resulted in the introduction of the concepts of *opportuntistic design* and *multistore design*.

In opportunistic design, the MapReduce system materializes intermediate results during query processing to support fault-tolerance. We treat these results as opportunistic materialized views and utilize them toward physical design. However, MapReduce queries often contain arbitrary code in the form of user defined functions (UDFs) which creates a challenge for re-using these views. We developed a semantic model for UDFs and a novel query rewrite algorithm to search the large space of views, allowing us to reuse these views and improve performance for exploratory queries on big data.

In multi-store design we use hybrid architectures that combine the scalability of a big data store (e.g., MapReduce) with the query processing power of an RDBMS. However, to take advantage of the RDBMS, big data queries must migrate and load their data into the RDBMS, a process that has a high cost. We developed techniques to dynamically tune the physical design of both stores simultaneously, periodically reorganizing the data in each store by migrating views between them. This approach enables big data queries to utilize the RDBMS and gain significant speedup.

Jeff LeFevre published these results at SIGMOD Workshop on Data Analytics in the Cloud ("Towards a workload for evolutionary analytics"), and at SIGMOD 2014 ("MISO: Souping Up Big Data Query Processing with a Multistore System" and "Opportunistic Physical Design for Big Data Analytics"). Dr. LeFevre graduated in May 2014.

vi. Programmable File Systems (student: Noah Watkins)

In the last two years we focussed on extending file systems to support parallel querying and scheduling. The first step is to come up with file systems that can support scientific data models. We noticed that different parallel file systems implement a common set of distributed services, such as a name service, a data placement service, a failure management service, asynchronous task scheduling service, and a data storage service. For these services to be scalable they maintain certain invariances, e.g. the size of an inode has to remain small even for very large files in order to ensure scalability of the name service. These services are usually only available indirectly as they are used to implement a file system interface with a byte stream data model. We are now investigating how we can generalize these parallel file system services as an implementation platform for scientific data models such as HDF5 without breaking scalability invariances, and identify new services (such as asynchronous indexing, index compression, and schedulers that trade off between sequential scanning and index utilization) that are of benefit to all data model implementations. We termed this approach "DataMods" which will result in programmable parallel file systems. Noah presented this work at PDSW 2012. We then started to receive synergistic funding from the LANL/UCSC Institute for Scalable Scientific Data Management to investigate how this approach relates to the FastForward I/O effort. Most recently we started a project with LANL (Galen Shipman, Brad Settlemyer, Pat McCormick,

Gary Grider) and Stanford (Michael Bauer, Alex Aiken) that investigates new abstractions to enable efficient interactions between the Legion runtime and storage systems.

Another aspect of programmability is the ability to evolve storage systems interfaces while maintaining full availability. Noah Watkins is exploring this by integrating scripting languages into the storage system that allow the modification of interfaces in a life system. To control this change, Noah is borrowing concepts from source code management, including versioning and workspaces. Noah presented preliminary results at BigDataCloud 2013 (in conjunction with EuroPar 2013).

Noah Watkins passed his thesis proposal exam in fall 2014 and plans to defend in fall 2015.

vii. Modeling and theoretical analysis (postdoc: Kleoni Ioannidou)

The purpose of this task is modeling and theoretical analysis of the DAMASC components and architecture. We have identified an appropriate language and model for DAMASC and refined the DAMASC architecture based on that model. We have also begun using the model to examine performance optimizations achievable be altering physical placement of data. Studying how different physical allocations of data affect performance is a challenging question because any given workload may have queries with contradictory effects on cost objectives. Furthermore, altering physical placement has a cost of its own that needs to be carefully accounted for and compared to the expected benefits. We have identified some tradeoffs between query costs caused by different physical placements. Although there has been some progress in formalizing this problem, we have learned that optimizing physical placement should follow performance studies given a fixed physical placement (which is a more natural starting point and can serve as a baseline).

Early results in modeling of SciHadoop and a theoretical analysis of indexing and pushed-based parallel querying have allowed us to develop several baseline scenarios upon which dynamic physical placement of data can be explored.

Ongoing efforts include additional work on other theoretical aspects related to other ongoing research in the group including divergent indexing (which has been published in the proceedings of the 2012 ACM SIGMOD International Conference on Management of Data.) and parallel querying using push-based approaches (which is targeted for VLDB 2013). Workloads with a large number of queries push-based approached have shown to be preferable compared to push-based (for amortized throughput). From a theoretical perspective, we want to identify efficient push-based and pull-based query executors and potential hybrid solutions. We will compare such solutions in terms of throughput and latency to identify their benefits for given practical scenarios. Research on cache management may help to identify ways to evaluate and compare current push- based approached and hybrids, possibly described as different prefetching versions.

Our group has also been exploring memory management to ensure QoS guarantees in a predictable manner without compromising system performance. We plan to examine our prior work in buffering as a solution to providing "predictable" memory management for prefetching, writeback, and different types of cache hits. This work could lead to effective use of caches and

may be a step towards enhancing DAMASC with the ability to enforce application QoS, enabling more efficient and effective use of the resources.

5. Unexpended funds: Indicate the amount of unexpended funds, if any that are anticipated to be left at the end of the current budget period. If the amount exceeds 10 percent of the funds available for the budget period, provide information as to why the excess funds are anticipated to be available and how they will be used in the next budget period.

***Unexpended funds*:** None.

Explanation: N/A