Biomolecular Path Sampling Enabled by Processing in Network Storage

P. Brenner[1], J. M. Wozniak[1], D. Thain[1], A. Striegel[1], J. W. Peng[2], and J. A. Izaguirre[1]

[1]University of Notre Dame
Dept. of Computer Science and Engineering
Notre Dame, IN 46556 USA
izaguirr@nd.edu

[2]University of Notre Dame
Dept. of Chemistry and Biochemistry
Notre Dame, IN 46556 USA
jpeng@nd.edu

Abstract

Computationally complex and data intensive atomic scale biomolecular simulation is enabled via Processing in Network Storage (PINS): a novel distributed system framework to overcome bandwidth, compute, storage, and security challenges inherent to the wide area computation and storage grid. High throughput data generation requirements for our scientific target are overcome through novel aggregate bandwidth capabilities. Biomolecular simulation methods are correlated with the client tools, hybrid database/file server (GEMS), computation engine (Condor), virtual file system adapter (Parrot), and local file servers (Chirp). PINS performance is reported for the path sampling of a solvated protein domain requiring over 1000 simulations with total output data generation on the order of 1TB.

1 Introduction

Computationally complex and highly parallel biomolecular sampling methods such as Folding@home [8] have been successfully mapped to grid resources in order to overcome computational resource limitations of individual simulation scientists. However grid characteristics such as heterogeneous resources (compute, storage, network), autonomous owners, and segmented organizational authentication present special challenges for the efficient production and management of large data sets. Our study of molecular transition paths, requiring the generation of nearly 1000 trajectories, will be enabled by the use of grid computation only when coupled with an efficient grid storage framework.

We first introduce our target scientific application of atomic scale biomolecular simulation, specifically, path sampling of the WW protein domain. We then present the Processing In Network Storage (PINS) model which efficiently maps molecular path sampling to the grid, allowing high throughput data creation. Our implementation of the framework is described, followed by a discussion of the improved simulation scalability through reduction of bandwidth consumption. Current path sampling statistics are presented and we conclude with a short discussion of the novel PINS capabilities in a brief review of related work.

2 Protein Domain Simulation

Atomic scale biomolecular simulation based on molecular dynamics is the primary scientific application for the current phase of PINS development. This simulation method has already proven to be a crucial tool in the revelation of fundamental protein, nucleic acid, and cell boundary properties. A promising future holds hope for longer time scale analysis enabled through computational advances in architecture and algorithms [12, 17, 10]. Two areas of intense research in the field are biomolecular sampling and path identification [24, 6, 2, 15, 4, 7]. Both are computation and data intensive since they involve long trajectories with expensive inter-atomic force computation per step. In this work we present grid distributed path identification simulations of a protein domain.

The protein under analysis is the WW protein domain of the PIN1 enzyme. The PIN1 enzyme binds to a subset of proteins, playing a role in the regulation of their function. Most notably the up-regulation of PIN1 may be involved in certain cancers, and the down-regulation may be involved in Alzheimer's disease [14]. Similar WW domains are found as components of many larger proteins and participate in binding reactions affecting protein regulation of a variety

1-4244-0910-1/07/$20.00 ©2007 IEEE.

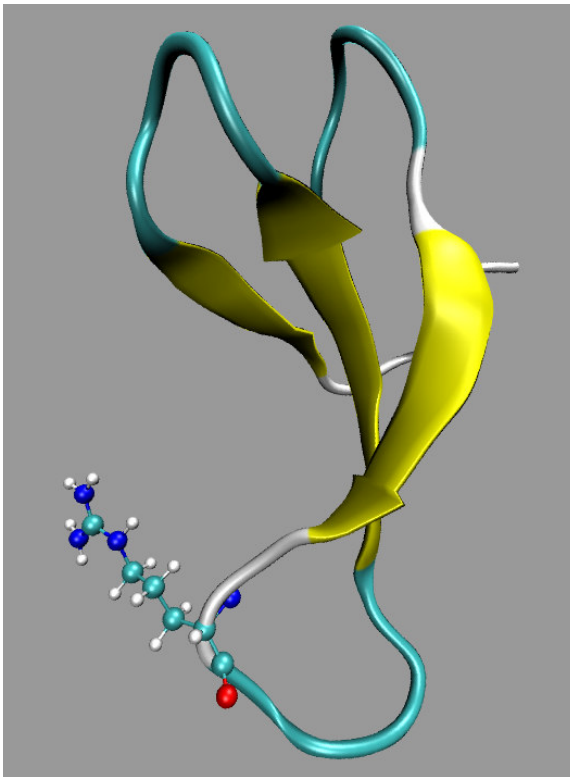

Figure 1. WW Domain with ARG12 shown. PDB ID 1I6C. Six residues have been truncated off the flexible linker end. [1, 25, 11]

of functions [19]. We are studying the motion and rates of the domain's recognition loop (Figure 1). The test system for simulation consists of an explicitly solvated 551 atom domain with 1469 TIP3P water molecules for a total size of 4958 atoms.

2.1 Path Sampling

The target of these biomolecular simulations is to identify a highly probable path of motion along a given set of reaction coordinates, and estimate the respective rates of this motion. To define a path, we focus on the motion of the Arg12 residue(amino acid cf. Figure 1) from its position in the WW domain's unbound configuration to that of the bound. The path is sampled using the Transition Path Sampling (TPS) method [4] which involves the iterative trial of numerous paths (molecular dynamics trajectories) to obtain a representative ensemble of possible paths. The method targets rare event (rapid but infrequent) transition paths which may be sampled via numerous short simulations as opposed to fewer long simulations. Depending on the specific path and length of trial trajectories 50 to 90 percent of the trajectories may fail. To generate a set of 100 accepted trajectories we expect to run approximately 1000 trials. Post processing is performed on the accepted path ensemble to determine the most probable path, estimate rates, and determine transition points.

```
# Client side script
Specify simulation parameters
Implement simulation algorithm
Submit PINS scripts
     # PINS script
     Perform Computation
     Store Simulation Data
Monitor Progress
Post Process Data (Remote Access)
```

Figure 2. Abstract Simulation Control.

3 PINS Framework

The PINS framework is built upon the close integration of multiple independent software components designed to provide client interface, file management, computation management, storage resource access, compute resource access, and secure authentication over the heterogeneous wide area computational grid. A scientist can directly harness these components, according to their respective APIs, through their scripting language of choice. The simulation execution is controlled via an interactive client side script and an 'in network' PINS script. A simple abstraction of the control script is shown in Figure 2. Fully detailed example scripts specific to the molecular simulation reported here are available online [1].

Figure 3 provides a high level overview of the components and functionality. A data generation computation (molecular dynamics) is matched with a computation resource followed by the storage of the data in the grid. A subsequent data analysis computation locates the stored files and automatically selects the available computation resource within the closest proximity. Bandwidth utilization to the remote client is reduced to communication with Condor daemons and the small output files from post processing the large 'in grid' data sets. To make this abstract representation more concrete we describe the following core components utilized in our PINS implementation. Note that the scientist's interface with each of these components is through his or her chosen scripting language and the majority of underlying functionality behind the component interfaces is abstracted to allow focus on implementation of the

[1] http://gipse.cse.nd.edu/GEMS

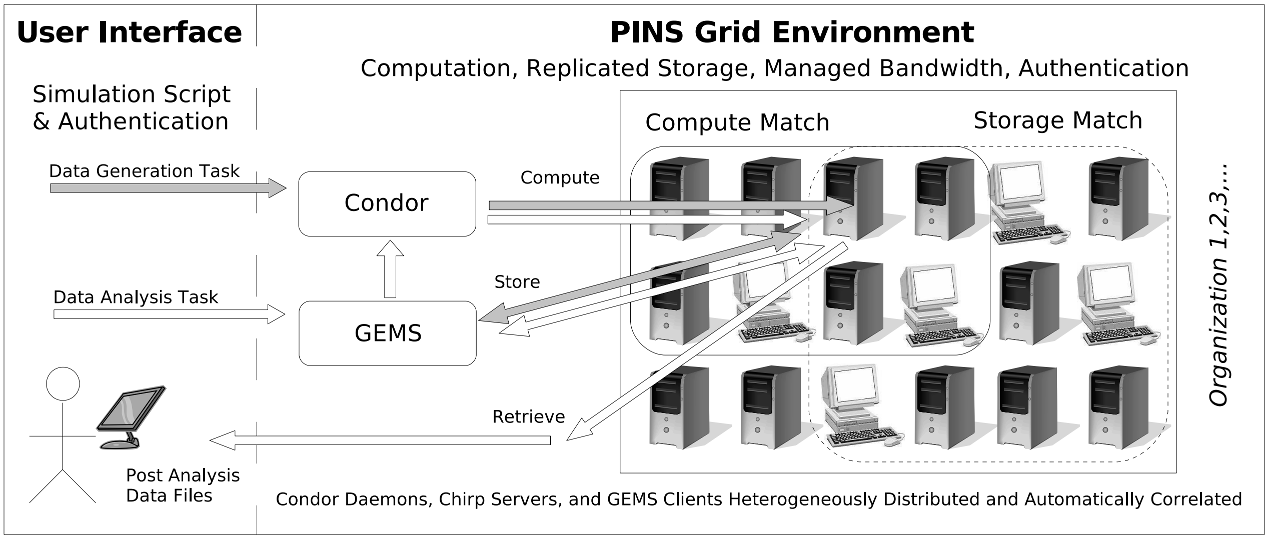
User Interface
PINS Grid Environment
Computation, Replicated Storage, Managed Bandwidth, Authentication
Simulation Script & Authentication
Compute Match
Storage Match
Data Generation Task
Condor
Compute
Store
Data Analysis Task
GEMS
Organization 1,2,3,...
Retrieve
Post Analysis Data Files
Condor Daemons, Chirp Servers, and GEMS Clients Heterogeneously Distributed and Automatically Correlated

Figure 3. PINS enabled simulation on heterogeneous resources in the wide area

simulation algorithm.

- File Management

  File management is enabled via the Grid Enabled Molecular Simulation (GEMS) hybrid file system/database [26]. GEMS provides client tools to store, query, and retrieve data over heterogeneous and autonomous storage resources. The GEMS server efficiently manages replicated files and their associated metadata to provide fault-tolerant large-capacity storage [28]. The GEMS client tools can be run on the scientist's host machine or, as in the PINS framework, called as part of the remote computation thus handling file movement and access within the grid distributed resources.

- Storage Access

  The heterogeneous and autonomous storage resources are made available by the Chirp file server [22] which provides secure authenticated access to files residing in space allocated by the storage owner through the user level installation of a Chirp server. Remote access to the distributed files is enabled by the Parrot personal file system adapter [23] which allows applications to utilize the remote files directly without modification.

- Work flow Management and Matchmaking

  Grid distributed compute resources are matched and authenticated to our simulation processes through the Condor [13] grid engine. PINS scripts which manage the storage and computation in a fault tolerant manner are executed as Condor jobs. Intermittent job failures resulting from inherent resource faults are overcome through automatic restarts from the last regularly saved simulation progress point, aided by a transactional record-keeping model.

- Secure Authentication

  Authentication and security are enabled uniformly by each component. At the highest level GEMS enforces storage and access permissions on all files over multiple domains, while Condor and Chirp provide authentication methods such as Globus credentials, hostname, and Unix/AFS permissions [27].

- High Throughput Data Generation

  Data intensive grid distributed simulation must acknowledge practical bandwidth and centralized storage limitations. The GEMS framework automatically utilizes storage and computation locality information minimizing both total and peak bandwidth consumption. Computation producing new data is correlated with distributed storage on or nearby the computation host, allowing high throughput computation to produce high throughput data without concern for overloading centralized storage. This unique capability is the key highlight in this phase of our PINS mapping to biomolecular simulation.

- Cooperation and Dynamism

  As resources in large scale computing are often owned or managed by disparate groups, we enable resource sharing and cooperation while respecting the authority of the machine owner. This authority allows changes in the available resources that are uncontrollable by the system as a whole. Operations within the system may be represented as combinations of operation type, job type, data sources, job location, and replica location. These combinations are created dynamically by the job scheduler and replica location system, enabling the effective use of uncontrolled remote resources.

4 Experimental Results

We first present baseline aggregate data generation measures demonstrating the requirement to handle bandwidth and storage efficiently for high throughput computation of variable resolution trajectories. Then we follow by showing how our PINS framework is being successfully utilized to study the PIN1 WW domain ARG12 path with PINS WW domain sampling statistics.

4.1 Aggregate bandwidth

Our application requires both a transactional record-keeping model *and* high peak output data bandwidth to keep up with the large degree of parallelism obtained through the distributed computational resources. For example, consider the bandwidth usage recorded in Table 1 for an unsolvated WW domain. Clearly, high resolution results obtained from multiple concurrent simulations will eventually overwhelm a client machine employing a data staging method, due to limits in the available network bandwidth and operating system capabilities. Since our resource fabric consists of widely distributed commodity hardware connected by a general purpose, multiple use network, the a traditional file staging architecture will not suffice. The PINS framework removes bandwidth bottlenecks by harnessing the additional *bandwidth parallelism* available in the network of volunteer resources, and *conserves bandwidth* by employing data locality when available.

4.2 PINS statistics

We are incrementally executing 2000 molecular dynamics simulations, where half of the simulations represent the forward and reverse components of a single trajectory respectively. This yields a total of 1000 attempted trajectories. Each simulation requires approximately 65 hours of uninterrupted computation time on a Pentium 4 system. For a homogeneous system of such machines this would yield a total computation time estimate of 130,000 CPU hours. Utilizing heterogeneous and autonomous resources yields significant variance in the computation average as jobs run on numerous architectures and are subject to resource eviction. Each simulation was segmented into automatically restarting segments such that the maximum computational overhead from a single eviction was 2 hours.

The total storage requirement is estimated at 1000 MD trajectories × 350 MB × 3 replicas[2]resulting in a storage requirement of over 1TB. Each simulation is comprised of 7 input files and 8 output files. The output was broken into 100 segments for restart functionality yielding 2000 × 100 stored data records for a total of 2000 × 100 × 8 output files, not counting the replicas.

The PINS framework enables this high throughput data generation by reducing bandwidth consumption through the efficient utilization of distributed storage. During simulation the output data is stored and replicated across the distributed storage in a location aware manner. Storage is first attempted on the compute host, the fall back option looks for a storage host in the same domain, an finally any available storage host. For the initial storage of all simulations we observe negligible bandwidth utilization as opposed to an over network store of 350 GB for all 1.6 million files. Subsequent distributed replication can be managed at off peak times [28].

4.3 WWd Path Sampling

Initial trial trajectories suffered a low acceptance ratio of less than 5% prompting an adjustment of the dihedral angle order parameter specification and the random initial velocity generator. For subsequent trial trajectories we are experiencing improved acceptances approaching 10% and continue to revise the trials for more efficient sampling. Figure 4 is a Ramachandran plot showing the conformations from a 10 ps segment of a 1 ns trajectory. During this short segment a transition was captured betweens states A and B as defined by the dihedral order parameters. The trajectory generation phase will continue until a representative set of at least 100 paths is collected.

Upon completion of the path sampling phase the path ensemble will be heavily post processed in a second phase of distributed computation to determine average thermodynamic properties, transition points, and the calculation of transition rates. The second phase analysis will also require additional simulations to calculate free energy terms.

5 Related Work

There are multiple grid utilities and middleware components which could be selected to build a PINS framework. The efficiency with respect to how PINS enables computation, storage, and bandwidth utilization is highly dependent on these components. We reference the following research

[2] GEMS asynchronously and dynamically obtains storage space and creates file replicas to provide data survivability on uncontrolled resources [26].

	Number of concurrent simulations		
Output Resolution	64	128	256
1000fs	0.05 MB/s	0.1 MB/s	0.21 MB/s
100fs	0.53 MB/s	1.06 MB/s	2.12 MB/s
10fs	5.3 MB/s	10.6 MB/s	21.2 MB/s

Table 1. Output bandwidth observations for WW domain simulation. Output resolution represents regularity of simulation state output with respect to the time step.

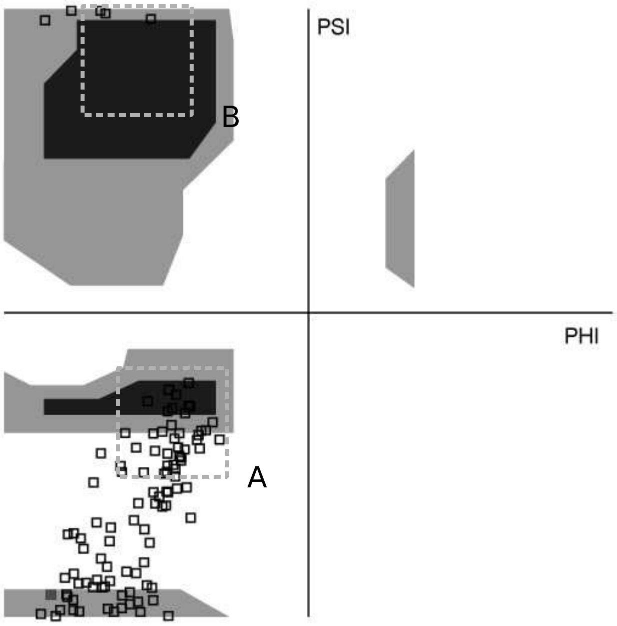

Figure 4. Ramachandran plot for a 10 ps segment of a 1 ns trajectory, capturing the transition between states A and B

projects with respect to their PINS relative function and or application for biomolecular simulation.

The Storage Resource Broker (SRB) [16] allows for the construction of unified data grids, from a variety of storage systems. SRB provides multiple user-configurable replication techniques, with appropriate metadata managed in a database [18].

The Globus [9] Replica Location Service (RLS) [5] provides the ability to map logical file names to physical file locations. Systems built upon the RLS must manage user metadata externally. Another Globus project, GASS [3], exemplifies the data staging model of job submission, using cache management strategies to reduce network bandwidth consumption.

The GFARM project [21] addresses possible collocation opportunities to enhance the I/O performance of a fully POSIX capable grid distributed file system. To meet their performance and scalability objectives, efficient replication algorithms are employed to improve data preservation and collocation over the wide (intercontinental) area.

While these these distributed architectures provide capabilities suitable for use in a PINS framework, we chose to implement PINS on top of the cited components based on the following novel utilities. First is the minimal impact on existing client applications, resources, and institutional authentication as enabled by the Parrot/Chirp [22] system. In addition, the Parrot interface design allows for a great deal of flexibility running simulation scripts because of the location independence and virtual file system that Parrot creates. Second, GEMS allows for heterogeneous and flexible scalability with no requirements for application recompilations, root privileges, or kernel modifications. Further, the hybrid architecture between a file database and a file system targets the core scientific tasks of data creation, analysis, and archive browsing allowing for optimizations not feasible in a full featured file system. Finally, Condor has a robust interface for matchmaking and job eviction ideally suited for a highly heterogeneous and autonomous set of compute resources.

A storage system designed for the application area of molecular dynamics is BioSimGrid [20]. Built upon the provision for a unified format trajectory database, BioSim-

Grid provides tools to perform analysis on its libraries of simulation data. The software architecture combines a standard database with an underlying SRB storage system. The primary differences between BioSimGrid and PINS are the PINS distributed computation interface and resiliency in the face of uncontrolled, volunteered resources, which subsequently drives a divergence in the grid distributed storage design.

6 Acknowledgments

This work was partially supported by NSF grants DBI-0450067 and CCF-0135195. Computational resources were made available by the Center for Research Computing and Department of Computer Science and Engineering at the University of Notre Dame.

References

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, pages 235–242, 2000.

[2] B. J. Berne and J. E. Straub. Novel methods of sampling phase space in the simulation of biological systems. *Curr. Opin. Struct. Biol.*, 7:181–189, 1997.

[3] J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke. GASS: A data movement and access service for wide area computing systems. In *Proceedings of the Sixth Workshop on I/O in Parallel and Distributed Systems*, 1999.

[4] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Ann. Rev. Biophys. Biophys. Chem.*, 53:291–318, 2002.

[5] A. L. Chervenak, N. Palavalli, S. Bharathi, C. Kesselman, and R. Schwartzkopf. Performance and scalability of a replica location service. In *Proc. High Performance Distributed Computing*, 2004.

[6] R. Elber. Novel methods for molecular dynamics simulations. *Curr. Opin. Struct. Biol.*, 6(2):232–235, 1996.

[7] R. Elber. Long-timescale simulation methods. *Curr. Opin. Struct. Biol.*, 15:151–156, 2005.

[8] V. P. et al. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68, 2003.

[9] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11, 1997.

[10] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, San Diego, 2nd edition, 2002.

[11] W. F. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *J. Mol. Graphics*, 14:33–38, 1996.

[12] A. R. Leach. *Molecular Modelling: Principles and Applications*. Prentice Hall, 2nd edition, Mar. 2001.

[13] M. Litzkow, M. Livny, and M. Mutka. Condor - A hunter of idle workstations. In *Proc. International Conference of Distributed Computing Systems*, 1988.

[14] K. P. Lu. Pinning down cell signaling, cancer, and alzheimer's disease. *Trends in Biochemical Sciences*, 29:200–209, 2004.

[15] F. Nardi and R. Wade. *Molecular Dynamics. From Classical to Quantum Methods*, chapter 21, pages 859–898. Elsevier Science B.V., first edition, 1999.

[16] A. Rajasekar, M. Wan, R. Moore, G. Kremenek, and T. Guptill. Data grids, collections and grid bricks. In *Proc. Mass Storage Systems and Technologies*, 2003.

[17] T. Schlick. *Molecular Modeling and Simulation - An Interdisciplinary Guide*. Springer-Verlag, New York, NY, 2002.

[18] G. Singh, S. Bharati, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A metadata catalog service for data intensive applications. In *Proc. Supercomputing*, 2003.

[19] M. Sudol, K. Sliwa, and T. Russo. Functions of ww domains in the nucleus. *FEBS Letters*, 490:190–195, 2001.

[20] K. Tai, S. Murdock, B. Wu, M. Ng, S. Johnston, H. Fanghor, S. J. Cox, P. Jeffreys, J. W. Essex, and M. S. P. Sansom. BioSimGrid: towards a worldwide repository for biomolecular simulations. *Org. Biomol. Chem.*, 2, 2004.

[21] O. Tatebe, N. Soda, Y. Morita, S. Matsuoka, and SatoshiSekiguchi. Gfarm v2: A grid file system that supports high-performance distributed and parallel data computing. In *Proceedings of the 2004 Computing in High Energy and Nuclear Physics (CHEP04)*, 2004.

[22] D. Thain, S. Klous, J. Wozniak, P. Brenner, A. Striegel, and J. Izaguirre. Separating abstractions from resources in a tactical storage system. In *Proc. Supercomputing*, 2005.

[23] D. Thain and M. Livny. Parrot: Transparent user-level middleware for data-intensive computing. In *Workshop on Adaptive Grid Middleware*, September 2003.

[24] W. F. van Gunsteren, T. Huber, and A. E. Torda. Biomolecular modelling: Overview of types of methods to search and sample conformational space. volume 330, pages 253–268. AIP, 1995.

[25] R. Wintjens, J. Wieruszeski, H. Drobecq, P. Rousselot-Pailley, L. Buee, G. Lippens, and I. Landrieu. 1h NMR study on the binding of PIN1 trp-trp domain with phosphothreonine peptides. *J. Biol. Chem.*, 276:25150–25156, 2001.

[26] J. M. Wozniak, P. Brenner, D. Thain, A. Striegel, and J. A. Izaguirre. Generosity and gluttony in GEMS: Grid-Enabled Molecular Simulation. In *Proc. High Performance Distributed Computing*, 2005.

[27] J. M. Wozniak, P. Brenner, D. Thain, A. Striegel, and J. A. Izaguirre. Access control for a replica management database. In *Proc. Workshop on Storage Security and Survivability*, 2006.

[28] J. M. Wozniak, P. Brenner, D. Thain, A. Striegel, and J. A. Izaguirre. Applying feedback control to a replica management system. In *Proc. Southeastern Symposium on System Theory*, 2006.