Parallel Computing 34 (2008) 652–660

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Biomolecular committor probability calculation enabled by processing in network storage

P. Brenner [a], J.M. Wozniak [a], D. Thain [a], A. Striegel [a], J.W. Peng [b], J.A. Izaguirre [a,*]

[a] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
[b] Department of Chemistry and Biochemistry, University of Notre Dame, Notre Dame, IN 46556, USA

ARTICLE INFO

Article history:
Received 15 May 2007
Received in revised form 30 June 2008
Accepted 21 August 2008
Available online 12 September 2008

Keywords:
High throughput computing
Distributed systems
Computational biophysics

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, organizational, and security challenges inherent to the wide-area computation and storage grid. PINS is presented as an effective and scalable scientific simulation framework to meet the unbounded requirements of a 'user of infinite need'. The novel hybrid database–filesystem architecture enables the high throughput computation and data generation required by our scientific target. Biomolecular simulation methods are correlated with the primary PINS components, including: client tools, hybrid database/file management service (GEMS), computation engine (Condor), virtual file system adapter (Parrot), and local file servers (Chirp). Performance for the PINS prototype is reported for the committor probability calculation of a solvated protein domain requiring 500 independent simulations and the generation of over 1,000,000 output files.

1. Introduction

The design of scalable modular frameworks to meet grand challenge computational efforts such as those referenced in the National Science Foundation report "Cyberinfrastructure Vision for 21st Century Discovery" [1] requires design based on the recognition of new challenges in computational science. The identification and analysis of atomic scale reaction coordinates relevant to the molecular function of a protein is a pertinent example. This process requires exploratory simulation using numerous targeted algorithms and reaction coordinates. The example simulation set discussed here required the generation of 500 trajectories and over 1 million output files; however, it is only a small portion of the superset of simulations undertaken in the investigation of this protein by the authors and our collaborators. For scientific challenges such as these, it is clear that a conceptual 'user of infinite need' is a practical and scientifically essential target. For this user, the incites revealed through computational simulation and analysis are bounded only by the limitations in resources, interfaces, and scalability correlations between the science and computational framework.

Computationally complex and highly parallel simulation and post-processing frameworks such as SETI@home [2] and Folding@home [3] have been successfully mapped to wide-area resources in order to overcome computational resource limitations of individual scientists with particular requirements. However, characteristics such as heterogeneous resources (compute, storage, network), autonomous owners, and segmented organizational authentication present special challenges for application agnostic frameworks to efficiently produce and manage large data sets. The design of modular frameworks to

* Corresponding author. Tel.: +1 574 261 0988.
E-mail address: izaguirr@nd.edu (J.A. Izaguirre).

0167-8191/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.parco.2008.08.001

meet these computational efforts will require architectures built on common core components, each designed to function in heterogeneous deployments.

To meet this challenge we introduce processing in network storage (PINS) a novel and practical distributed framework to address the many scientific 'users of infinite need' faced with a real world of limited finite resources. PINS holds a unique position in the spectrum of scientific computation frameworks between homogeneous high performance computing (HPC) and specialized application wide-area implementations. PINS is a hybrid offering high throughput scientific computation by effectively and efficiently utilizing shared HPC and wide-area resources in an application agnostic manner.

In this work, we first introduce our target scientific application of atomic scale biomolecular simulation, specifically, committor probability calculation of the WW protein domain. We then present the PINS model which allows a novel and efficient mapping of simulation to grid computation and storage, facilitating high throughput data creation and analysis. Our implementation of the framework is described, followed by a discussion of the improved simulation scalability through decentralized storage and utilization of heterogeneous resources. Current committor probability results for our target protein are presented. In conclusion, the novel PINS capabilities are summarized in a brief review of related work.

2. Protein domain simulation

Atomic scale biomolecular simulation based on molecular dynamics is the primary scientific application of our team's computational scientists. 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 [4–6]. Two areas of intense research in the field are biomolecular sampling and transition path analysis [7–12]. Both are computation and data intensive since they involve long trajectories with expensive inter-atomic force computations at each time step. In this work we present distributed transition simulations of a protein domain, focused on the calculation of committor probabilities.

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 [13]. Similar WW domains are found as components of many larger proteins and participate in binding reactions affecting protein regulation of a variety of functions [14]. We are studying the motion and rates of the domain's recognition loop (Fig. 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.

Sampling of the system was performed using the replica exchange method [15–17] with a target temperature of 278 K for subsequent correlation with experimental NMR results. Fig. 2a shows the ARG12 (amino acid cf. Fig. 1) dihedral distribution results of one such test. One dominant state *A* is apparent along with a path to a second state *C* populated only marginally greater than the intermediate points found along the connecting path. The designation of a state *B* is reserved for a separation of state A into two adjacent states as observed in separate sampling tests. In Fig. 2b the states are overlaid on a

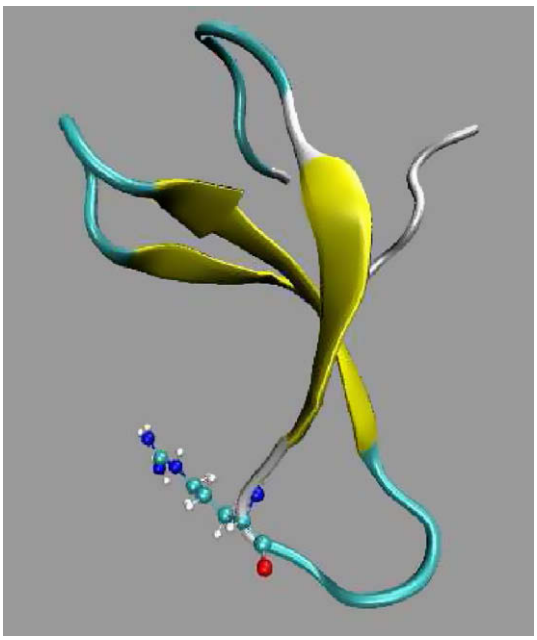

Fig. 1. WW domain with ARG12 shown. PDB ID 1I6C. Six residues have been truncated off the flexible linker end [18–20].

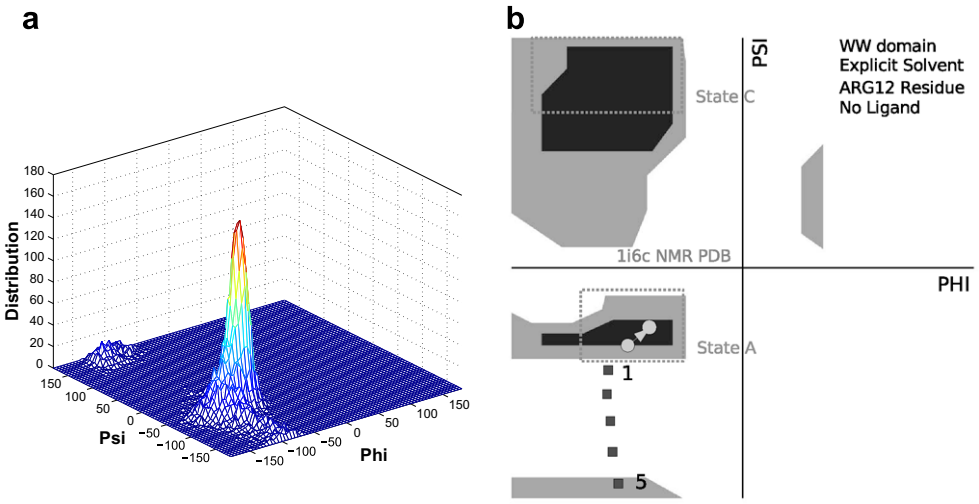

Fig. 2. (a) WW domain ARG12 dihedral distribution from explicitly solvated replica exchange simulation with a target temperature of 278 K. (b) Labeled Ramachandran for WW domain ARG12. The five intermediate conformations are ordered from state *A* to state *C*. The circles in state *A* represent the ARG12 value as found in the 1I6C PDB structure and the arrow represents movement to a new value after equilibration of the initial PDB structure.

Ramachandran plot marked with the standard expected population as observed for the set of all residues. In addition, five intermediate points along the path between states *A* and *C* are denoted and serve as the points from which the committor probabilities are calculated as described in the next section.

2.1. Committor probability calculation

The target of these biomolecular simulations is to analyze 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 with respect to its Φ and Ψ backbone dihedral angles. More specifically, our study centers on the motion of the WW domain's unbound state for later comparison to that of the bound state when the ligand is present. The committor probability calculations are made based on the work by Du et al. [21,22] and subsequent related evolutions of the concept [23,24]. The results provide a strong validation of the distributions achieved via sampling and insight into the free energy surface along this pathway.

The committor probability p is defined for each of the five designated intermediate states as the probability of arriving at state *A* before arriving at state *C*. To compute p we ran numerous trajectories and determined first arrival in either *A* or *C*. The larger the number of trajectories the greater the accuracy. The notation p_{ac} denotes the probability of arriving at *A* before *C* and p_{ca} denotes *C* before *A*, respectively. The value serves as an indication of the kinetic proximity of the intermediate state to the stable state. Note that multiple intermediate states can have similar probabilities, forming an ensemble of states of similar kinetic distance. Because each trajectory's contribution to p is not qualified by time, the distribution of p can indicate either a plateau in the energy surface or a free energy maxima. This statement is conditional that the simulation time is sufficiently long for a high percentage of all trajectories to arrive at either *A* or *C*. A high percentage of trajectories not arriving at either state may indicate insufficient trajectory lengths, the presence of intermediate stable states, or an energy surface dominated by reaction coordinates sufficiently decoupled from those used in the current state definitions.

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 Fig. 3. Fully detailed example scripts specific to the molecular simulation reported here are available on-line.[1]

Fig. 4 provides a high level overview of the components and functionality. The user composes a client side script which creates a relevant directory structure from which to serve input files and submit jobs to Condor. The data generating computation (molecular dynamics) is matched with a computation resource followed by the storage of the data in the grid. A

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

```
# 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)
```

Fig. 3. Abstract simulation control.

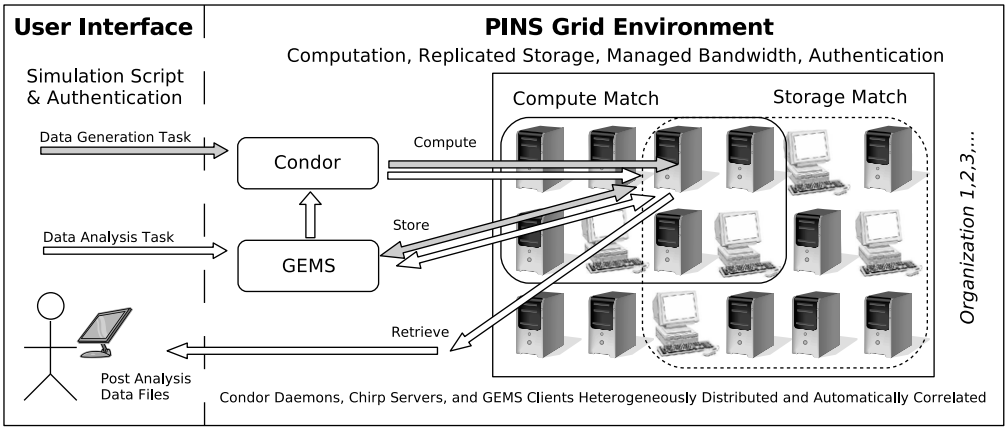

Fig. 4. PINS enabled simulation on heterogeneous resources in the wide-area.

subsequent data analysis computation initiated via client side script uses GEMS to locate 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 distributed 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 simulation algorithm.

- File management
  File management is enabled via the grid enabled molecular simulation (GEMS) hybrid file system/database [25]. 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 [26]. 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. While GEMS was originally designed as a repository for molecular structure data sets, it has been fully generalized to support multiple modes of data services, including usage as a localized, streaming data service or a widely replicated tertiary archive. During committor calculation all files: input, output, and post-processing were stored in GEMS.
- Parameterized data repository
  A fundamental feature of the GEMS repository is the ability to tag individual data sets in a scientifically meaningful manner. Thus, a previous simulation stored in the system may be located by providing the parameters used to create it. Borrowing from previous work in virtual data systems [27], this capability is expanded in the PINS framework through the use of a distributed, in-place storage fabric. The construction of scripts is simplified by the ability to refer to, look up, and even browse other data sets. The user can employ a parameterized programming and data access model, unaware of the complexity of the underlying dynamic replica placement system and corresponding name resolution and data movement details. Further the repository functions not only as an interface to the simulation engine but as a stand alone repository for archival and data sharing. Both command line and GUI Interfaces to directly access and manage data are available via command line and the GEMSview GUI client as shown in Fig. 5. In this work scripts were greatly simplified by the ability to reference files via filenames or associated metadata without consideration of low level details such as the physical location and descriptors.

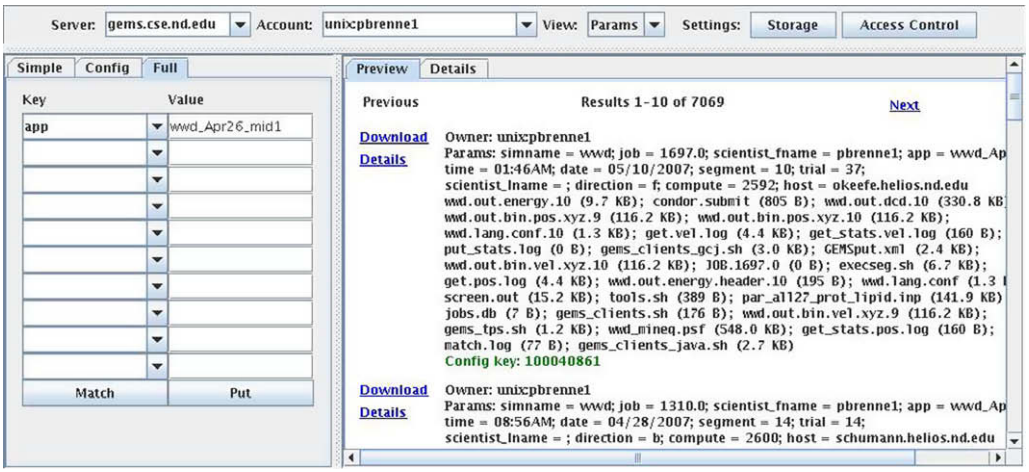

Fig. 5. Record browsing in the general-purpose GEMSview client tool. By simply specifying a batch name, all records created by that batch are obtained, including parameterized metadata and file information. Users may then iteratively specify additional key/value pairs to reduce the quantity of returned results before downloading the correct data set from a remote replica site.

- Storage access
  The heterogeneous and autonomous storage resources are made available by the Chirp file server [28] 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 [29] which allows applications to utilize the remote files directly without modification. File access for the trajectory post-processing was automatically optimized. The user generated post-processing script simply modified calls for otherwise local trajectory files with a Parrot call. GEMS automatically located the closest file and informed Parrot accordingly.
- Workflow management and matchmaking
  Grid distributed compute resources are matched and authenticated to our simulation processes through the Condor [30] 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 diverse authentication methods including Globus certificates, hostname verification, and Unix techniques [31]. The trajectory and post-processing files in this work were limited to read access only by the collaborators.
- High throughput data generation
  Data intensive grid distributed simulation must acknowledge practical bandwidth and centralized storage limitations. The GEMS replica location service automatically utilizes storage and computation locality information minimizing both total and peak bandwidth consumption. Computation producing new data is correlated with distributed storage *in the network* – on or near to the computation host – allowing high throughput computation to produce high throughput data without concern for overloading centralized storage. This unique capability is a highlight in this phase of our PINS mapping to biomolecular simulation. Trajectory files were written locally first in all cases and the majority of post-process computation was able to be correlated with the location of a trajectory replica.
- 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. Throughout the weeks of simulation associated with these data sets all storage and computation was achieved by capitalizing on compute and storage resources owned by and prioritized for scientists other than the simulation owner. Although resource availability was stochastic, trajectory generation proceeded without intervention.

The synthesis of these features comprises a viable framework with good scaling properties for computational experiments. The provided tool set allows ‘users of infinite need’ to approach larger problems by integrating additional resources into the cooperative framework. As the data sets grow, file management and parameterized organization are required to provide record-keeping capabilities. This enables seamless storage access utilities and restartable work flows. Expanding user

Table 1
Four selected subsets of heterogeneous volunteer resources used in the PINS experiment

Category	CPUs	Jobs	MIPS	Description	Util. (%)	I/O (%)
LOCO	32	272	1500	Homogeneous cluster	72	5.2
HELIOS	86	355	2500	Desktops in student labs	54	12
SC0	64	364	3100	Homogeneous cluster	59	9.4
CVRL	64	62	2200	Homogeneous cluster	87	5.2

Each category of machines executed the given number of jobs as selected by the Condor system. Utilization is defined as the ratio of time spent doing useful work versus the time spent idling. The I/O ratio is the ratio of time spent performing data location or movement activity versus time spent performing a simulation. Statistics are obtained from the experiment diagrammed in Fig. 6.

groups are not centrally managed, enabling user-managed distributed access control. Data movement is contained within local networks where possible. These features, implemented in a lightweight manner on multiple-use resources, promote cooperative infrastructure models in which it is easy to integrate additional heterogeneous hardware to handle more users, more problems, and larger workloads.

4. Experimental results

4.1. PINS statistics

4.1.1. Simulation requirements

The protein analysis performed consisted of 500 molecular dynamics simulations. Each simulation requires approximately 84 h 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 42,000 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 approximately 1 h. The client side and 'in network' scripts included control logic specific to the submission of MD trajectories, segmentation of these trajectories, and subsequent post-processing to calculate committor probabilities.

The total storage is estimated at 500 MD trajectories $\times$100 segments $\times$28 files $\times$3 replicas[2] resulting in the requirement to storage, catalog, and manage 4,200,000 input and output files. For this particular simulation the output file of dominant size (the trajectory file) was reduced in size by removing the explicit waters as their positions were not necessary for the desired calculation. The removal of explicit water from the trajectory file reduced the file size by an order of magnitude bringing the total replicated storage requirement for this simulation to roughly 50 GB as opposed to 500 GB. In this case the primary benefit in reduced file size relates to improved performance of the external post-processing applications.

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, and 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 50 GB for all 1.4 million original files. Subsequent distributed replication can be managed asynchronously at off peak times [26], in a pairwise manner that utilizes the parallelism of the network.

4.1.2. Performance observations

Researchers intending to obtain the highest utility from their available resources must consider the utilization of the systems on which they are operating. Jobs running in the PINS framework may be clearly dissected into component tasks, and a consideration of the time spent in each will indicate to the programmer the performance value of the use of the target volunteer resources.

A sample of the jobs in this study were analyzed further to determine the utilization of computation and communication resources. Jobs spend their time in one of three significant states: time spent processing, time spent storing or reading simulation data, and idle time. *Idle time* – as perceived by the PINS user – may be caused by a variety of factors including eviction or suspension of the computation resource by the owner, or other problems. Idle time varies among different machine categories or clusters; for example, the heavily used, regularly rebooted desktops in a university laboratory (HELIOS) are only available to PINS 46% of the time. The sum of the component times gives the overall turnaround or wall-clock time, measured as the difference between two timestamped records in the GEMS system. Thus the idle time considered here is only the time that counts as interruption and delay between segments of batch runs.

Of the whole experiment, 1053 jobs were isolated for further analysis. Each job produced a *segment*, a span in simulation time for the appropriate trajectory corresponding to roughly two hours of computation time. The resulting segment is stored

[2] GEMS asynchronously and dynamically obtains storage space and creates file replicas to provide data survivability on uncontrolled resources [25].

in the system, so that a complete simulation may be obtained by concatenating a list of consecutive segments or restarting a simulation from a given segment. The jobs were found to have executed on 120 of the available computation sites, 94 of which were members of sizable categories of similar hardware. Four categories were chosen corresponding to the computational capability measured by Condor in million instructions per second (MIPS) and location, as described in Table 1.

Results are diagrammed in Fig. 6. The most notable result is that I/O operations are nearly insignificant compared to the time spent idling. This indicates that the implementation of the database-like distributed storage system adds no significant overhead to the overall turnaround time of each segment. Performance penalties are largely due to the idle time mandated by system owners, an unavoidable occurrence on a shared resource fabric. For example, note that the SC0 processors outperform the CVRL processors, yet better user experience is enjoyed on the CVRL system due to its high availability and resulting low turnaround time. The CVRL results indicate that the framework can make good utilization of an isolated highly available cluster, while maintaining the flexibility to integrate multiple clusters into a unified infrastructure. However, the unpredictable HELIOS machines produced a greater number of completed jobs (355), showing that the ability to integrate large numbers of unreliable computational resources is beneficial.

4.2. WWd committor probabilities

In Table 2 the calculated committor probabilities are shown based on one hundred, 1 ns trajectories for each of the five intermediate points. The simulations were carried out in the canonical ensemble at a temperature of 278 K, using explicit solvation. The data columns indicate the values for the probability of committing to state A before C (p_{ac}), and committing to state C before A (p_{ca}). In this particular case the number of observed trajectories which committed to neither state was zero for all intermediate points. This indicates sufficiently unhindered mobility along the reaction coordinate and validates the selection of a 1 ns trajectory to observe committors. Of key interest is the finding that intermediate points 1 and 2 always commit first to state A whereas points 3, 4, and 5 all exhibit similar behaviors in terms of the percentage split between commitments to A or C. This significant difference in the mobility between points 1,2 and 3,4,5 may indicate an additional reaction coordinate (hydrogen bonding) present in initial conformations 3,4,5 that facilitates motion along the ARG12 dihedral coordinates.

In a visual interpretation of the observed motion, Fig. 7a superimposes the dihedral motion from all trajectories starting at intermediate point 3, onto the distribution obtained from the REM sampling shown previously. Fig. 7b shows the same trajectory data from point 3 in contour form. We observe the highest density of dihedral values for the trajectory set is found near the initial point, and note that despite that nearly 80% of trajectories commit first to state A, the majority of dihedral values lie on the path from point 3 toward state C. This prompts further investigation into additional reaction coordinates

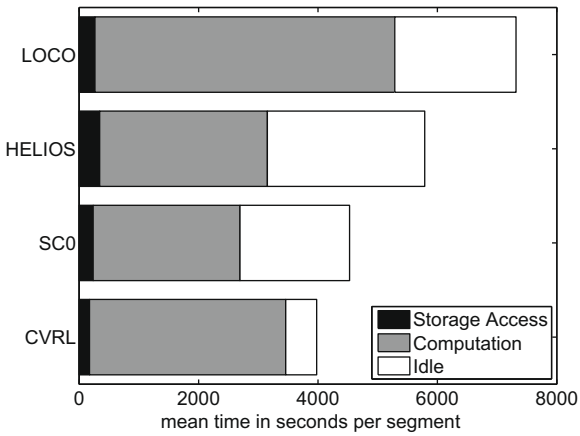

Fig. 6. Performance profile of PINS operation.

Table 2
Calculated committor probability values for five intermediate initial points

	p_{ac}	p_{ca}
Intermediate point 1	100	0
Intermediate point 2	100	0
Intermediate point 3	79.2	20.8
Intermediate point 4	77	23
Intermediate point 5	79	21

Based on 100 trajectories computed from each point. Values are a percentage.

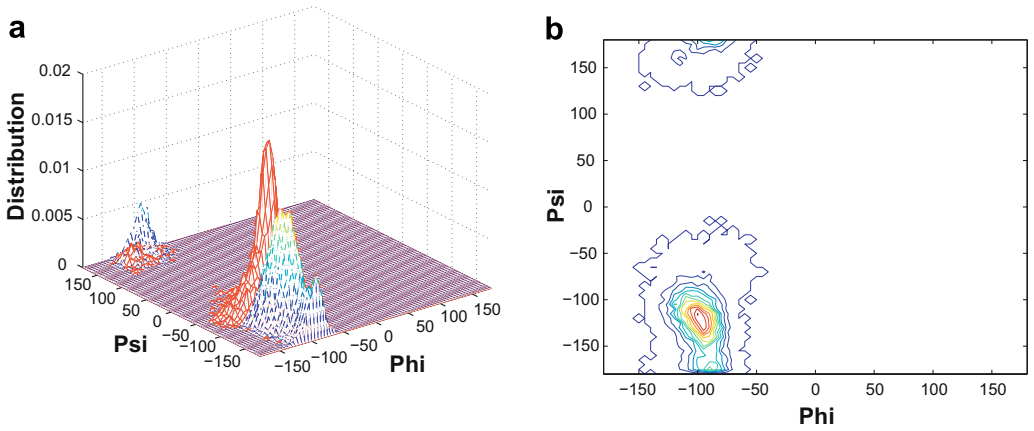

Fig. 7. (a) Distribution of ARG12 dihedral values for 100 trajectories started from intermediate point 3, superimposed onto the ARG12 dihedral values obtained from the REM sample. The REM data has a solid color mesh and contains the dominant peak. (b) Contour of the distribution of ARG12 dihedral values for 100 trajectories started from intermediate point 3.

that might enable what appears to be relatively fast diffusive motion along the path and away from the dominant state *A* low energy well.

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

The storage resource broker (SRB) [32] 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 [33].

The Globus [34] replica location service (RLS) [35] 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 [36], exemplifies the data staging model of job submission, using cache management strategies to reduce network bandwidth consumption.

The GFARM project [37] 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 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 [28] 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 [38]. Built upon the provision for a unified format trajectory database, BioSimGrid 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.

Acknowledgements

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. This work is the extension and evolution of research presented by the authors at HiCOMB 2007 [39].

References

[1] C. Council, Cyberinfrastructure vision for 21st century discovery, Tech. Rep. 1, National Science Foundation, March 2007.
[2] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer, SETI@home: an experiment in public-resource computing, Communications of the ACM 45 (11) (2002).
[3] V.S. Pande, I. Baker, J. Chapman, S.P. Elmer, S. Khaliq, S.M. Larson, Y.M. Rhee, M.R. Shirts, C.D. Snow, E.J. Sorin, B. Zagrovic, Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing, Biopolymers 68 (1) (2008) 91–109.
[4] A.R. Leach, Molecular Modelling: Principles and Applications, second ed., Prentice-Hall, 2001.
[5] T. Schlick, Molecular Modeling and Simulation – An Interdisciplinary Guide, Springer-Verlag, New York, NY, 2002.
[6] D. Frenkel, B. Smit, Understanding Molecular Simulation, second ed., Academic Press, San Diego, 2002.
[7] W.F. van Gunsteren, T. Huber, A.E. Torda, Biomolecular modelling: overview of types of methods to search and sample conformational space, in: AIP Conference Proceedings, vol. 330, The first European conference on computational chemistry (E.C.C.C.1), 1995, pp. 253–268, nancy, France.
[8] R. Elber, Novel methods for molecular dynamics simulations, Curr. Opin. Struct. Biol. 6 (2) (1996) 232–235.
[9] B.J. Berne, J.E. Straub, Novel methods of sampling phase space in the simulation of biological systems, Curr. Opin. Struct. Biol. 7 (1997) 181–189.
[10] F. Nardi, R. Wade, Molecular Dynamics. From Classical to Quantum Methods, first ed., Elsevier Science B.V., 1999 (Chapter 21).
[11] P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Transition path sampling: throwing ropes over rough mountain passes, in the dark, Ann. Rev. Biophys. Chem. 53 (2002) 291–318.
[12] R. Elber, Long-timescale simulation methods, Curr. Opin. Struct. Biol. 15 (2005) 151–156.
[13] K.P. Lu, Pinning down cell signaling, cancer, and Alzheimer's disease, Trends Biochem. Sci. 29 (4) (2004) 200–209.
[14] M. Sudol, K. Sliwa, T. Russo, Functions of ww domains in the nucleus, FEBS Lett. 490 (2001) 190–195.
[15] K. Hukushima, K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations, J. Phys. Soc. Jpn. 65 (6) (1996) 1604–1608.
[16] U.H. Hansmann, Parallel tempering algorithm for conformational studies of biological molecules, Chem. Phys. Lett. 281 (1997) 140–150.
[17] D.J. Earl, M.W. Deem, Parallel tempering: theory, applications, and new perspectives, Phys. Chem. Chem. Phys. 7 (2005) 3910–3916.
[18] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucl. Acids Res. (2000) 235–242. <http://www.pdb.org/>.
[19] R. Wintjens, J. Wieruszeski, H. Drobecq, P. Rousselot-Pailley, L. Buee, G. Lippens, I. Landrieu, 1h NMR study on the binding of PIN1 trp–trp domain with phosphothreonine peptides, J. Biol. Chem. 276 (2001) 25150–25156.
[20] W.F. Humphrey, A. Dalke, K. Schulten, VMD – visual molecular dynamics, J. Mol. Graph. 14 (1996) 33–38.
[21] R. Du, V.S. Pande, A.Y. Grosberg, T. Tanaka, E. Shakhnovich, On the transition coordinate for protein folding, J. Chem. Phys. 108 (1998) 334.
[22] V. Pande, A. Grosberg, T. Tanaka, D. Rokhsar, Pathways for protein folding: is a new view needed?, Curr Opin. Struct. Biol. 8 (1998) 68–79.
[23] Y.M. Rhee, V.S. Pande, One-dimensional reaction coordinate and the corresponding potential of mean force from commitment probability distribution, J. Phys. Chem. B 109 (2005) 6780–6786.
[24] Y.M. Rhee, V.S. Pande, On the role of chemical detail in simulating protein folding kinetics, J. Chem. Phys. 323 (2006) 66–77.
[25] J.M. Wozniak, P. Brenner, D. Thain, A. Striegel, J.A. Izaguirre, Generosity and gluttony in GEMS: grid-enabled molecular simulation, in: Proceedings of the High Performance Distributed Computing, 2005.
[26] J.M. Wozniak, P. Brenner, D. Thain, A. Striegel, J.A. Izaguirre, Applying feedback control to a replica management system, in: Proceedings of the Southeastern Symposium on System Theory, 2006.
[27] I. Foster, J. Voeckler, M. Wilde, Y. Zhao, Chimera: a virtual data system for representing, querying, and automating data derivation, in: Proceedings of the Scientific and Statistical Database Management, 2002.
[28] D. Thain, S. Klous, J. Wozniak, P. Brenner, A. Striegel, J. Izaguirre, Separating abstractions from resources in a tactical storage system, in: Proceedings of the Supercomputing, 2005.
[29] D. Thain, M. Livny, Parrot: transparent user-level middleware for data-intensive computing, in: Workshop on Adaptive Grid Middleware, 2003.
[30] M. Litzkow, M. Livny, M. Mutka, Condor – a hunter of idle workstations, in: Proceedings of the International Conference of Distributed Computing Systems, 1988.
[31] J.M. Wozniak, P. Brenner, D. Thain, A. Striegel, J.A. Izaguirre, Access control for a replica management database, in: Proceedings of the Workshop on Storage Security and Survivability, 2006.
[32] A. Rajasekar, M. Wan, R. Moore, G. Kremenek, T. Guptill, Data grids, collections and grid bricks, in: Proceedings of the Mass Storage Systems and Technologies, 2003.
[33] G. Singh, S. Bharati, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, L. Pearlman, A metadata catalog service for data intensive applications, in: Proceedings of the Supercomputing, 2003.
[34] I. Foster, C. Kesselman, Globus: a metacomputing infrastructure toolkit, Int. J. Supercomput. Appl. 11.
[35] A.L. Chervenak, N. Palavalli, S. Bharathi, C. Kesselman, R. Schwartzkopf, Performance and scalability of a replica location service, in: Proceedings of the High Performance Distributed Computing, 2004.
[36] J. Bester, I. Foster, C. Kesselman, J. Tedesco, 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.
[37] O. Tatebe, N. Soda, Y. Morita, S. Matsuoka, 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.
[38] K. Tai, S. Murdock, B. Wu, M. Ng, S. Johnston, H. Fanghor, S.J. Cox, P. Jeffreys, J.W. Essex, M.S.P. Sansom, BioSimGrid: towards a worldwide repository for biomolecular simulations, Org. Biomol. Chem. 2.
[39] P. Brenner, J.M. Wozniak, D. Thain, A. Striegel, J.W. Peng, J.A. Izaguirre, Biomolecular path sampling enabled by the PINS distributed simulation framework, in: Proceedings of the Sixth IEEE Workshop on High Performance on Computing in Biology HiCOMB 2007, 2007.