Pathways into Large Parameter Search Spaces: Experiences with Molecular Hyperdynamics

Justin M. Wozniak, Santanu Chatterjee, Paul Brenner, Douglas Thain, Aaron Striegel and Jesús A. Izaguirre
University of Notre Dame

A great deal of systems software is available to enable researchers to utilize complex computation and data grids to perform and visualise large batches of scientific jobs [1]–[6]. However, the dependency structure for molecular hyperdynamics simulation [7]–[9] does not fall into well-studied categories. Without user interaction to steer the workflow structure, a brute force implementation could consume months of computer time. In this report, a distributed scientific filesystem is employed upon which the researcher may perform distributed computations while enabling the browsability required to maintain control of the virtual experiment. We demonstrate the computation time savings available through user interaction with batches of jobs running on an unreliable resource fabric, an unwieldy combination of volunteer computing sites as well as a cooperative, controlled storage network of borrowed disk space: common aspects of these systems are highlighted in Table I. These benefits are enabled through the use of opportunistic computing and storage systems driven by a reusable methodology and software system called the *steerable parameterized workflow*, the primary topic in this discussion.

Storage Abstraction: The Scientific Filesystem

Modern e-Science applications motivate efforts to bridge the gap between the management of computation and storage. The combination of a variety of technologies including worldwide computing resources, task/data co-schedulers, and workflow development tools provides great opportunities for computational research, but system designers must remain grounded in the need for scientists to obtain understandable, reproducible results. Our approach to the problem formulates tangible, steerable *data sweeps*, constituting a scientific exploration of a parameter space. This provides a cooperative framework that drives well-defined computational transformations making up the work units of a scientific simulation or other computation.

In this framework, typical molecular simulation applications involve the generation of simulated molecular trajectories over a range of input parameters. Figure 1(a) diagrams this method by indicating simulation *segments* - restartable chunks of a simulation - as functions of the simulation parameters. The hyperdynamics application described in this paper differs in that only a subset of the whole parameter space will be explored- it is too large to fully explore and only a part is of interest. However, the area of interest is not known in advance and must be determined by user inspection of previously computed segments. The interesting areas of the parameter space are entered by *branching* from the existing segments, creating a new trajectory that differs from the unmodified sequence in that a new bias force is applied to the system. Thus, additional metadata must be stamped on each segment to record the location of the segment in the search tree, as diagrammed in Figure 1(b). These stored segments conflate concepts such as workflow node, application checkpoint, and pre-staged input data source.

Steerable Metadata-Driven Execution

Many parameter search applications must address the two challenges posed by *breadth* and *depth*. Expanding search breadth allows the algorithm to explore many search paths in parallel. Depth allows a search to proceed for a long period of time, assuming that proceeding down a narrow pathway is expected to come to the desired result. The researcher may run a long simulation and wait for occurrences of interesting events; or the researcher may run an ensemble of short simulations that attempt to find the same events. Long runs on unreliable resources require regular checkpoints; and benefit from a storage system that allows structured and tagged application-aware checkpoints. Broad runs on highly parallel resources are easily developed and synchronized on the parameterized data storage system.

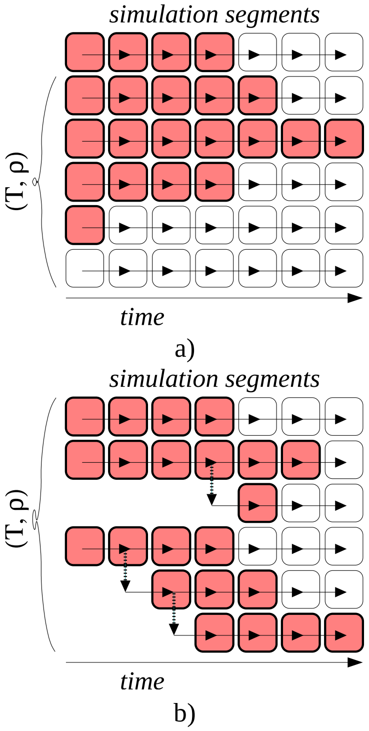
In hyperdynamics simulation, enhanced storage organization and rapid data access for spacious parameter sweeps are insufficient features for the effective investigation of system behavior. This steered method allows the user to bias the simulation into areas of conformational space yet to be explored. The observed local distribution of entropy of visited microstates determines when and how the bias should be applied. These timescale and entropy histograms visually indicate if the simulation has progressed to the point at which applying another bias potential level would be beneficial and free from serious error. Since there is no analytical method to make this determination, tools to enable *ad hoc* exploration of the parameter space are required. As a demonstration of the nontriviality of this process, Figure 2 diagrams the timescale performance as a function of branch location on the time axis. Thus, the researcher controlling the simulation must monitor the output histograms for error and smoothness while selecting branch points that maximize simulation efficiency in terms of timescale.

Summary: In certain exploratory applications, selecting all workflow targets in advance may not be efficient, desirable, or even possible. The workflow techniques presented here improve the ability to browse intermediate simulation results and guide simulation progress.

Problem	**Solution**
Cooperative, unreliable storage fabric	Data replication [10], [11]
Cooperative, unreliable compute fabric	Opportunistic computing [2], [12]
Data access for computation	Locality, collocation [13]
Science-friendly interaction with running batches	The *scientific filesystem* abstraction

TABLE I

OVERVIEW OF PROBLEMS AND SOLUTIONS IN COOPERATIVE COMMODITY SCIENTIFIC COMPUTING.



a)

b)

Fig. 1. a) In a parameter sweep [13], [14], user jobs typically fill a square parameter space. b) To allow for interactive parameter execution, the system has to allow the dynamic user creation of execution branches. Our approach starts by encapsulating workflow elements as metadata-tagged filesets distributed throughout the computing infrastructure and accessible through the parameter-driven scientific filesystem abstraction. Runtime progress reports and visualisations are easily obtained from this framework. As a result of these observations, dynamically created targets may be inserted. The parameterized workflow system fills in the dependency structure for the new target and spawns the required computation.

REFERENCES

[1] Daryl H. Hepting, "Interactive evolution for systematic exploration of a parameter space," in *Intelligent Engineering Systems through Artificial Neural Networks*, 2003.

[2] Michael Litzkow, Miron Livny, and Matt Mutka, "Condor - A hunter of idle workstations," in *Proc. International Conference of Distributed Computing Systems*, 1988.

[3] J. Cao, S. A. Jarvis, S. Saini, and GR Nudd, "GridFlow: Workflow management for grid computing," in *Proc. Cluster Computing and the Grid*, 2003.

[4] Ewa Deelman, Tevfik Kosar, Carl Kesselman, and Miron Livny, "What makes workflows work in an opportunistic environment?," *Concurrency and Computation: Practice and Experience*, vol. 18, 2006.

[5] S. G. Parker, M. Miller, C. D. Hansen, and C. R. Johnson, "An integrated problem solving environment: the SCIRun computational steering system," in *Proc. Hawaii International Conference on System Sciences*, 1998.

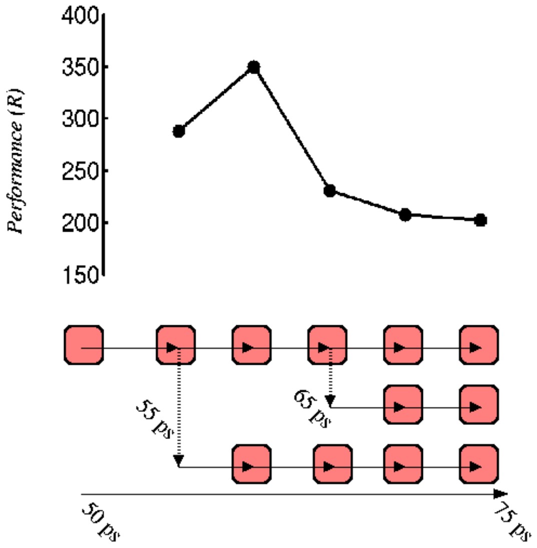


Fig. 2. Performance ratio R for various branch points. The ratio is plotted above a diagram indicating two illustrated example branches. R is plotted as a function of branch time. A higher ratio indicates more efficient exploration of the simulated conformation space. The performance ratio is only obtainable as the result of a simulation segment, necessitating interactive workflow control.

[6] Karen Schuchardt, Brett Didier, and Gary Black, "ECCE - a problem-solving environment's evolution toward grid services and a web architecture," *Concurrency and Computation: Practice and Experience*, vol. 14, 2002.

[7] A. Voter, "A method for accelerating the molecular dynamics simulation of infrequent events," *J. Chem. Phys.*, vol. 106, no. 11, 1997.

[8] A. F. Voter, "Hyperdynamics: accelerated molecular dynamics of infrequent events," *Phys. Rev. Lett.*, vol. 78, 1997.

[9] X. Zhou, Y. Jiang, K. Kramer, H. Ziock, and S. Rasenmassen, "Hyperdynamics methods for entropic systems:time-space compression and pair correlation function approximation," *Phys. Rev. E*, vol. 74, pp. 1–4, 2006.

[10] Justin M. Wozniak, Paul Brenner, Douglas Thain, Aaron Striegel, and Jesus A. Izaguirre, "Generosity and gluttony in GEMS: Grid-Enabled Molecular Simulation," in *Proc. High Performance Distributed Computing*, 2005.

[11] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, and M. Ripeanu, "Giggle: A framework for constructing scalable replica location services," in *Proc. Supercomputing*, 2002.

[12] David P. Anderson, "BOINC: A system for public-resource computing and storage," in *Proc. Workshop on Grid Computing*, 2004.

[13] Paul Brenner, Justin M. Wozniak, Douglas Thain, Aaron Striegel, Jeff W. Peng, and Jesus A. Izaguirre, "Biomolecular committor probability calculation enabled by processing in network storage," *Parallel Computing*, 2008, accepted.

[14] David Abramson, Jon Giddy, and Lew Kotler, "High performance parametric modeling with Nimrod/G: Killer application for the global grid," in *Proc. International Parallel and Distributed Processing Symposium*, 2000.