Grid Enabled Interactive Molecular Dynamics Simulations

Peter V. Coveney and Shantenu Jha,
Centre for Computational Science, University College London
Stephen Pickles,
Manchester Computing, The University of Manchester

The importance of computational approaches in providing quantitative information as well as qualitative insight has been widely acknowledged. For biomolecular systems, classical molecular dynamics (MD) simulations have the greatest ability to provide insight into specific aspects of a system at a level of detail not possible for other simulation techniques and often not even accessible experimentally [1]. However, the ability to provide such specific information comes at the price of making such simulations extremely intensive computationally. Although so far high-performance computational approaches have addressed this requirement to a great extent and in the process have played a critical role in enhancing our understanding of biomolecular systems, it has been shown [2] that if larger and more "real systems" are to be simulated over meaningful timescales, then advances in both the algorithms and the computing approach used are imperative. In this article we will focus on new computing approaches that facilitate *interactive* simulations of large-scale systems and will discuss some of the issues that require attention when attempting to use such an approach effectively and in a routine manner. By interactive simulations, we mean simulations with which the end-user can interact though a visualization component (visualizer) and/or computational steering client in near real-time.

Using Interactive Simulations to Compute Free Energy Profiles: An Exemplar

The transport of biomolecules like DNA, RNA and poly-peptides across protein membrane channels is of primary significance in a variety of areas. Although there has been a flurry of recent activity, both theoretical and experimental [3, 4] aimed at understanding this crucial process, many aspects remain unclear.

A back-of-the-envelope estimate of the computational resources required helps us to appreciate why there has not been any significant computational contribution to understanding the dynamical aspects of the translocation problem. The physical time scales for translocation of large bio-molecules through a trans-membrane pore is typically of the order of tens of microseconds. It currently takes approximately 24 hours on 128 processors to simulate one nanosecond of physical time for a system of approximately 300,000 atoms. Thus it takes about 3000 CPU-hours on a tightly coupled machine to simulate 1ns. Therefore a straightforward "simple" MD simulation will take 3×10^7 CPU-hours to simulate 10 microseconds – a prohibitively expensive amount. Thus approaches that are "smarter" than vanilla classical equilibrium MD simulations are required.

As part of the SPICE project[1] we are implementing a method (here referred to as SMD-JE, see Refs. [2, 6] for more details) to compute the free energy profile (FEP) along the vertical axis of the protein pore. By adopting the SMD-JE approach, the net computational requirement for the problem of interest can be reduced by approximately a factor of 50-100. The methodology however, requires the introduction of two new parameters, with a corresponding uncertainty in the choice of the values of these parameters. The situation can be addressed by performing a set of "preprocessing simulations", which, along with a series of interactive simulations, help determine an appropriate choice of the parameters. To benefit from the potential advantages of the SMD-JE approach and to facilitate its implementation at all levels — interactive simulations for such large systems, the pre-processing simulations and finally the production simulation set — we use techniques developed in the RealityGrid project[2] to make optimal use of the computational resources of a federated trans-Atlantic grid.

Making Interactive Simulations Possible: Some Technical Issues

As the model being studied is very large and complex, a large number of processors are needed to provide sufficient compute-power for the simulation to be interactive. Thus, it is rather unlikely that all the computational and visualization resources required for interactive simulations will be available in one place, and even less likely that these will be co-located with the user.

[1] SPICE is the UK part of a joint UK-US high-end computing project funded by the EPSRC in the UK and the NSF in the US that shows the advantages of grid-enabled interactive simulations. SPICE will be exhibiting at SC05 on the University of Manchester and UK e-Science booths, amongst others.

[2] See http://www.realitygrid.org.

As a consequence, we have to deal with a geographically distributed set of resources, and the performance of high-end interactive simulations becomes dependent on the performance of the network. Furthermore, since these interactive simulations involve bi-directional communication - there is a steady flow of data from the simulation to the visualizer, as well as flow in the reverse direction when, for example, the user applies a force to a subset of atoms through the visualizer — unreliable communication can lead not only to a possible loss of interactivity, but also to a slowdown of the simulation if it has to wait for data from the visualization [6]. Thus interactive MD simulations place rather unique and demanding quality of service (QoS) requirements (low latency, jitter and packet loss) on the network linking the simulation and visualization components. The recently deployed UKLight [7] network and the optically networked Global Lambda Infrastructure Facility [8] provide us with the necessary QoS. The CSAR systems – Newton and Green – are already connected to UKLight and the connection of HPCx should be in place by Supercomputing 2005.

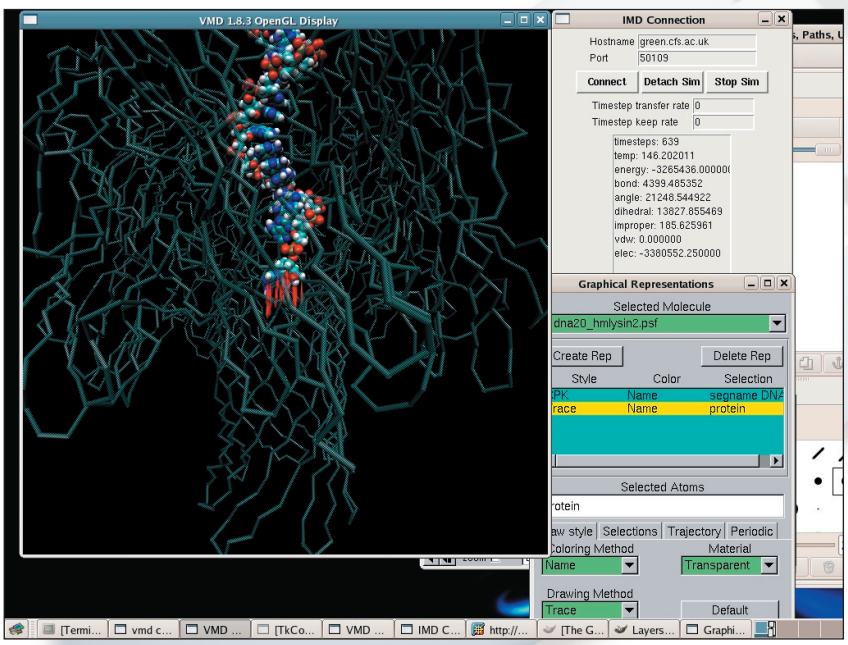

Figure 1: Snapshot of an interactive session in progress. A single stranded DNA polymer is beginning its translocation through the alpha-hemolysin protein pore. The red arrow-headed lines represent the forces that are applied to the end residues of the DNA to guide and speed up its translocation through the pore. When grid infrastructure is not used the user is responsible for providing host and port information. The use of RealityGrid tools simplifies this situation and the user only need select the simulation of interest. The steering infrastructure [5] then handles establishing the details of communication and exchanging this information in a manner that is essentially invisible to the user.

This style of interactive computing requires co-allocation of computation and visualization resources as well as network circuits of sufficient bandwidth and QoS between the simulation and visualization components so that the simulation does not stall waiting for the visualization and interactivity can be maintained. The need for community-wide agreements and protocols for co-allocation is obvious, but today, *ad hoc*, pair-wise agreements are the norm. For example, co-allocation [9, 10] is currently handled by manually reserving the necessary resources (we are grateful for the co-operation of HPCx, CSAR and TeraGrid staff in this matter). The current mode of operation is cumbersome and error-prone (one of the authors had to exchange about a dozen emails correcting three distinct errors introduced by two different administrators for one reservation request!) and is not in general a scalable solution. The situation gets even more complicated when attempting to schedule non-compute resources like network light-paths simultaneously. At present, we find ourselves in the fortunate situation where we have a quasi-persistent connection between our visualization machine (at UCL) and most of our compute resources (at CSAR and on the TeraGrid) and thus have to worry only about co-allocating compute resources. But sooner or later, demand for switched light-paths will increase and we will be faced with the more general situation of having to worry about coordinating and co-scheduling network circuits with compute resources, for it is of no use if the circuit between two machines is provisioned only after the user's reservation on the two machines has expired! There have been encouraging attempts and progress has been in this direction [11, 12, 13], but much remains to be done. This is a challenging problem and will need to be addressed by all the communities involved, for any bespoke or partial solution will present problems when used in a true grid context.

By facilitating the coupling of the many different components involved — geographically distributed visualization, computational and software resources in distinct administrative domains — grid infrastructure enables us to circumvent traditional constraints of batch computing style HPC by allowing distributed large-scale interactive simulations. The computation

of FEP is an example of a large-scale problem that benefits tremendously from using HPC grids. In particular it is a good example of the advantages — both quantitative and qualitative — that steering simulations of large biomolecular systems can provide. The use of interactive simulations to explore further interesting and important problems, as well as its uptake as a generally useful computational approach, will require a stable and easy to use infrastructure that addresses satisfactorily the issues discussed and thus makes performing large-scale steered simulations more convenient and thereby routine.

Acknowledgements

This work has been funded as part of the RealityGrid, RealityGrid-Platform, ESLEA and SPICE projects and has been possible thanks to the assistance of many people. In particular we'd like to mention Andrew Porter, Robin Pinning and Rob Haines at the University of Manchester for their work on the software infrastructure. We would like to thank Nicola Pezzi, Peter Clarke and John Graham for facilitating the use of UKLight. Finally we'd like to acknowledge Matt Harvey from UCL for assistance in the installation and deployment of the required infrastructure at many levels.

Contact Details

Peter V. Coveney and Shantenu Jha
Centre for Computational Science,
University College London
20 Gordon Street
London
WC1H 0AJ

Stephen Pickles
Manchester Computing,
The University of Manchester
Oxford Road
Manchester
M13 9PL

Further Reading

Volume 363, Number 1833 / August 15, 2005 of *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*: http://www.journals.royalsoc.ac.uk/link.asp?id=h88056h75554

Volume 7, No. 5 and 6 of *Computing in Science and Engineering,* Guest Editors: Bruce Boghosian and Peter Coveney: http://www.computer.org/portal/site/cise/

References

[1] M. Karplus and J. McCammon. *Nature Structural Biology*, 9(9):646–652, 2002.

[2] S. Jha, P. V. Coveney, M. Harvey, and R. Pinning. *Proceedings of ACM/IEEE Supercomputing Conference,* 2005, in press. http://sc05.supercomputing.org/schedule/pdf/anal109.pdf

[3] D. K. Lubensky and D. R. Nelson. *Phys. Rev E*, 31917 (65), 1999; Ralf Metzler and Joseph Klafter. *Biophysical Journal*, 2776 (85), 2003; Stefan Howorka and Hagan Bayley, *Biophysical Journal,* 3202 (83), 2002.

[4] A. Meller et al, Phys. Rev. Lett., 3435 (86) 2003; A. F. Sauer-Budge et al. *Phys. Rev. Lett.* 90(23), 238101, 2003.

[5] S. M. Pickles, R. Haines, R. L. Pinning and A. R. Porter, A Practical Toolkit for Computational Steering, *Phil Trans. R. Soc.*, 363, 1843-1853, 2005.

[6] S. Jha, M. J. Harvey, P. V. Coveney, N. Pezzi, S.Pickles, R. L. Pinning, and P. Clarke. *Proceedings of the UK e-Science All Hands Meeting*, 2005. http://www.allhands.org.uk/2005/proceedings/papers/455.pdf.

[7] http://www.uklight.ac.uk.

[8] http://www.glif.is.

[9] B. Boghosian and P. V. Coveney. Guest Editors' Introduction: Scientific Applications of Grid Computing – part ii. *Computing in Science and Engineering,*. DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/MCSE.2005.123.

[10] Karl Czajkowski, Stephen Pickles, Jim Pruyne, Volker Sander, Usage Scenarios for a Grid Resource Allocation Agreement Protocol, GRAAP-WG, Global Grid Forum, 2003.

[11] G-lambda: Coordination of a grid scheduler and lambda path service over GMPLS. http://www.gtrc.aist.go.jp/g-lambda/.

[12] http://www.cct.lsu.edu/personal/maclaren/CoSched/.

[13] https://forge.gridforum.org/projects/graap-wg.