

# Managing Biomolecular Simulations in a Grid Environment with NAMD-G

Michelle Gower\*    Jordi Cohen†    James Phillips†    Richard Kufrin\*  
Klaus Schulten†‡

May 3, 2006

## Abstract

Experiences designing and deploying NAMD-G, an infrastructure for executing biomolecular simulations using the parallel molecular dynamics code NAMD within the context of a Computational Grid, are described. The effort is motivated by a general outline of the tasks involved in conducting research of this class as traditionally undertaken, followed by a description of the enhancements offered by current developments in Grid technologies. Specifics of the initial implementation of NAMD-G are given and a brief example of the use of the system in real-world scientific investigations simulating gas permeation in proteins is provided.

## 1 Introduction

As the computation of biological phenomena inevitably becomes increasingly complex, the need for high-level simulation tools also becomes increasingly important. Biologically-inspired simulations often study one specific system at a time, in great detail. In most scientific disciplines, this approach is often sufficient and even satisfactory. For many biological problems, however, studying one system is a great limitation. Let us consider the case of proteins, which are the micromachines and building blocks of living organisms. A given protein is always just one small entity in a greater family, consisting of proteins of related functions and/or structures. Furthermore, any protein of a specific species in the tree of life will have homologs in different species, each of which

is slightly different in its function and/or structure. Lastly, our protein can only perform its role by interacting with a large number of other molecules, including other proteins. One quickly realizes that the understanding arising from the study of a protein in isolation is limited, and that a protein must be studied in the context of its protein superfamily, its protein homologs, and its interactions with its partners. Consequently, a large number of simulations must be examined. Similarly, if one is interested in studying a specific biological function that is common to many classes of proteins, then again a large number of cases must be studied. We believe that the main barrier to computationally studying many related proteins, thereby taking advantage of biological diversity and complexity, is not one of methodology or of lacking resources, but rather is caused by the nature of the tools currently used by scientists. While tools are readily available for performing multiple protein sequence and structure comparisons, tools for comparing protein dynamics across many different systems are lacking.

Distributed computational resources collectively referred to as “Computational Grids” offer great promise as platforms that offer transparent and seamless access to the most powerful processors and networks available, and support the extreme-scale requirements necessary to conduct cutting edge scientific investigation. While advanced facilities such as those made available through the National Science Foundation TeraGrid facility provide basic building blocks for the latest offerings in hardware and software, they do not provide complete *end-to-end solutions* that are directly applicable to the routine activities of the scientists. This is not a failing of the distributed computational environments’ design but, rather, is a consequence of the large number of diverse activities that can be usefully pursued within

---

\*National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

†Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign.

‡To whom correspondence should be addressed.

them. It is more fruitful to rely on the knowledge, expertise, and resources of the individual scientific communities in order to define and build the software infrastructures that directly correspond to their needs. NAMD-G, the software system described in this paper, is the result of such an effort to create an infrastructure for biologically-inspired simulations.

Software developed by the Theoretical and Computational Biophysics Group (TCBG)<sup>1</sup>, such as the molecular dynamics code NAMD [13], has enjoyed increasing usage by a growing international community that benefits from the excellent parallel scalability that NAMD has been shown to achieve [12]. As an example of the aggregate computational requirement for NAMD-based simulations, during a six-month period spanning from April through September 2005, NAMD was the highest usage application at the National Center for Supercomputing Applications (NCSA) based on service units delivered on the Center’s production systems.

Even with the availability of state-of-the-art application software packages such as NAMD, there remains an intensive human-directed effort that is necessary to manage every technical aspect of each individual investigation. Regardless of the size of the system of interest, researchers must prepare at a minimum input data files appropriate for the target simulation. The MD simulation itself will be directed, monitored, and analyzed by the researcher and may involve management of a collection of input and output data files that can easily exceed hundreds of gigabytes. These files may reside locally (on the researcher’s “home” system) or may be created on remote supercomputers many thousands of miles away.

In practice, managing remote batch jobs can be a time-consuming activity that amounts to “baby-sitting” a simulation through to completion. This problem is exacerbated when large numbers of simulations need to be concurrently performed. The time spent in these mundane activities is therefore time unavailable for performing far more useful tasks. The NAMD-G system removes the burden of this from the researcher and, to a great extent, automates the entire process from beginning to end, monitoring progress, handling repeated remote batch job submissions, and finally transferring all files necessary both to and from their desired locations.

The capabilities of NAMD-G go beyond simply

<sup>1</sup>TCBG is part of the Beckman Institute at the University of Illinois and operates the NIH Resource for Macromolecular Modeling and Bioinformatics. More information and software downloads are found at <http://www.ks.uiuc.edu/>

running a sequence of runs specified by the researcher. It is nearly always the case that executing a simulation from start to finish requires substantially more total computing time than is available within limits imposed by the remote batch system. In the absence of a system like NAMD-G, the researcher is forced to periodically check the status of submitted jobs, verify that the simulation is progressing satisfactorily, and restart the simulation from an intermediate checkpointed state. This process has to be repeated until the total desired simulation time has been reached.

## 2 User Experience

In order to study a specific protein function in a wide variety of proteins, a scientist would use NAMD-G in the following way. A library of dozens of protein systems is selected based on their relevance to the problem at hand. All-atom structures are then generated for each system and serve as inputs for the simulations. For each protein system, a directory is created that contains these input files, consisting of the initial atomic coordinate and structure files. The scientist then edits a “run file”, which contains a short textual description of the simulation, as well as information such as the types and ordering of the various sequential runs to perform, the machine name, user email, *etc.*, which are used by NAMD-G to create and run the simulation. A typical run file is shown in Figure 1, and illustrates the simplicity of the setup. A number of run types, such as pre-equilibrations, equilibrations, and NVT and NPT ensemble production runs are pre-defined and will automatically work for any system without user intervention. The scientist can furthermore easily extend this setup by defining

```
SYSNAME myoglobin
MACHINE NCSA-Cobalt
EMAIL your_email@domain.edu

#RUN CONFIG_FILE PROCS WALLHOURS
RUN pre1.namd 24 4
RUN pre2.namd 24 4
RUN equil.namd 24 12
RUN npt.namd 24 12
```

Figure 1: A simple NAMD-G run file. The run file is edited by the user and provides information, such as the system name, the remote machine to run on, the user’s email, a list of the sequential runs that comprise the simulation, and other optional parameters.

their own run types. When this is done, the scientist simply types `ngsubmit` in a terminal, in the desired simulation directory.

Once NAMD-G is invoked, all the steps in the simulation are automatically managed and monitored until the simulation is complete. For our initial application (see Section 4), a typical simulation consisted of two pre-equilibration steps, a 1 ns equilibration and a 6 ns production NPT run. The two pre-equilibrations and the 1 ns equilibration each correspond, on average, to 1 multiple-processor job, and the 6 ns production runs average 4 to 7 such jobs running in sequence on the remote supercomputer. Each of these processes requires an initial transfer of input files, the storage to backup and to local disk of output files, and the conditional creation and transfer of input files and submission of the next job, which itself depends on the state of completion of the just-run job. For each studied system, this sequence of operations (file transfer, simulation creation, job submission and monitoring) must be performed 7 to 11 times just for one simulation to complete. For the case of simulating 25 different systems, this corresponds to 250 job submissions and 750 file transfer events (each job requiring 3 sets of transfers: input files, and output files to mass storage and to the local machine) in a short time, for which there is no allowance for error, and each of which have their independent timelines and dependencies. As the simulation runs complete, the output files gradually appear on the scientist’s local machine and progress is communicated by means of email reports. The scientist can also monitor (through the `ngstat` command), remove (`ngdel`), and modify and resubmit halted or failed jobs (`ngrestart`). At no point of this process does the scientist need to directly access remote accounts or files. The scientist can then perform his analysis of each protein system on his local machine. Having these tasks automated by NAMD-G allows scientists to tackle experiments which involve large numbers of independent simulations, as in the example above.

### 3 Implementation

There were other prerequisites for NAMD-G beyond performing management tasks for the scientist. The first goal was not to require changes in the existing NAMD software. Another requirement was to utilize stable software such that its own failures would not keep the scientists from getting their work done.

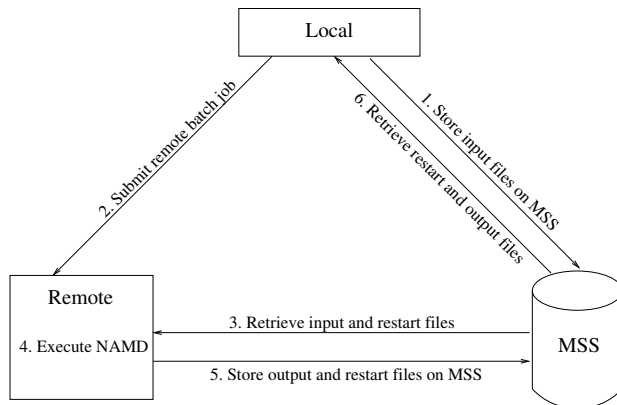


Figure 2: Overview. These are basic tasks normally done by the scientist that are managed by NAMD-G. Tasks 2–6 are repeated until the simulation is complete.

It was also desired that a solution be implemented rapidly so that the scientists could start using it to submit jobs as quickly as possible.

The basic tasks NAMD-G needs to accomplish include transferring files to and from a mass storage system (MSS) and submitting remote batch jobs until a simulation is complete, as shown in Figure 2. NAMD-G utilizes existing grid software (see Table 1) to allow the user to accomplish those tasks from their desktop.

#### 3.1 Authentication

A scientist who runs codes on one or more remote machines has to keep track of a different remote login and password for each machine. NAMD-G uses the security part of the Globus Toolkit [4] to securely replace these with a single passphrase. The user first obtains a long-term certificate from a certificate authority (CA) that is recognized by the remote host (for example the NCSA CA). From these files, the user creates a short-lived proxy certificate that will be used when authenticating to the remote machines. The remote machines have a file that maps unique information, called a Distinguished Name (DN), from this proxy certificate to the username. This process is called GSI (Globus Security Infrastructure) authentication.

Functionality	Software	Component	Location		
			Local	Remote	MSS
Authentication	Globus Toolkit	Security	X	X	X
	MyProxy		X		
Job Submission and Monitoring	Condor	Condor-G	X		
	Globus Toolkit	Pre-WS GRAM Services		X	
Workflow Management	Condor	DAGMan	X		
Data Transfer	uberFTP		X	X	
	Globus Toolkit	Pre-WS GridFTP Service		X	X
Simulation	NAMD			X	

Table 1: Software required for NAMD-G functionality on the user’s desktop (Local), the compute cluster (Remote), and the mass storage system (MSS). NAMD-G is built on standard grid services.

### 3.2 Remote Job Submission

Condor [14] is a management system created by Miron Livny’s group at the University of Wisconsin-Madison. NAMD-G uses the Condor-G component, which uses the Globus Resource Allocation Manager (GRAM) [7] architecture of the Globus Toolkit to submit jobs to remote resources. From the local machine, Condor-G converts a high-level description of the job to a Globus Resource Specification Language (RSL). Condor-G submits this RSL job description to a Globus server, called a gatekeeper, running on the remote machine. The gatekeeper first authenticates the user using GSI-authentication. Then it starts a job manager process that converts the RSL into a batch script and submits the job to the batch system.

NAMD-G also uses Condor’s DAGMan (Directed Acyclic Graph Manager), which allows the user to specify the dependencies between jobs and can re-run a job if necessary. NAMD-G takes advantage of the fact that NAMD writes checkpoint files from which it can be restarted. A typical NAMD simulation consists of several runs, each broken into multiple jobs due to batch queue runtime limits (see Figure 3). Managing this sequence of NAMD executions fits naturally into DAGMan. The scientist tells NAMD-G which NAMD configuration files to use for each run and the order in which to execute them. From this information NAMD-G creates an input file for DAGMan and lets DAGMan and Condor-G manage the remote batch jobs in the specified order.

After each remote batch job, NAMD-G needs to determine whether that job was successful or not. To see if NAMD finished gracefully, NAMD-G looks for the text that NAMD prints before exiting (for exam-

ple “End of Program”) in NAMD’s standard output. If it finds this text, it tells DAGMan that the job has finished successfully and DAGMan submits the next job in the sequence. If the job did not finish successfully, then DAGMan writes a rescue file that contains information about what jobs still need to be run. NAMD-G uses this file to restart the simulation where the job failed instead of repeating the entire simulation. In most cases, this allows the user to very easily restart the simulation where it quit by typing in a single command.

NAMD-G keeps the scientist informed about the progress of his simulation. The user can tell NAMD-G to send an email at the end of each batch job which includes the relevant parts of the NAMD standard output. This makes it more informative than the standard batch system email. In the case of an error, NAMD-G first looks for the standard output and error from NAMD. If it exists and can be retrieved from the remote system, NAMD-G emails this to the user. If it cannot find the NAMD output, it looks for the batch job’s standard output and error and emails those instead.

### 3.3 File Transfers

NAMD-G uses uberFTP [10], a GridFTP-enabled client developed by NCSA, for all file transfers. It was chosen over other clients for three reasons. It uses GSI-authentication, which allows NAMD-G to use it without requiring passwords as well as being one of the two methods allowed to remotely access NCSA’s mass storage system (MSS). uberFTP handles the delay in retrieval of files from NCSA’s MSS as the files are staged from tape to disk. Outside of NAMD-G, uberFTP provides an interactive FTP

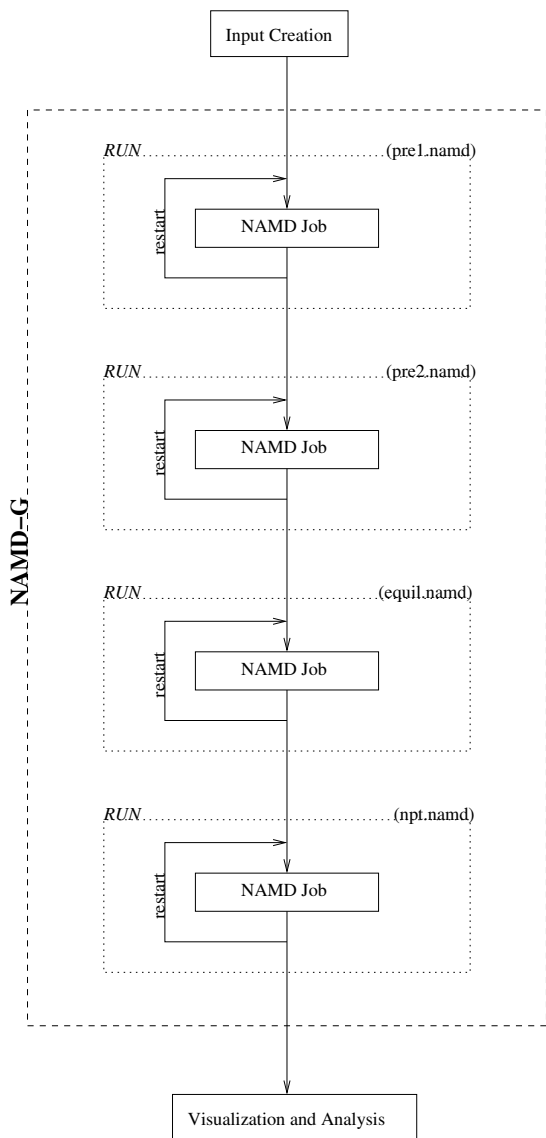


Figure 3: Typical NAMD-G workflow. This diagram shows the ordering of the high-level job-based tasks corresponding to the example run file in Figure 1. If a run is not completed successfully within a single NAMD job, due to exceeding the wallclock limit or a transient failure, then an additional NAMD job is submitted to continue the run.

interface that can easily be used by the scientist to manually retrieve backups from the MSS.

When the user submits the simulation, NAMD-G creates a tar archive of the current directory and transfers the archive to NCSA’s MSS so that it can be retrieved from inside the remote batch job. Besides the initial input files, the job will need restart files from the previous job if it is not the first job in the simulation. NAMD-G retrieves the appropriate restart files from NCSA’s MSS as well at the beginning of the remote batch job.

After a batch job has ended, NAMD-G creates two tar archives of the output files according to size to allow for separation of restart files from the larger simulation output. It ignores input and previous restart files and other miscellaneous files generated by NAMD with certain extensions (i.e. .tar, .old, .BAK, FFTW\*.txt) to minimize storing duplicate data. These tar archives are transferred to NCSA’s MSS for backup as well as back to the user’s local machine.

### 3.4 Technical Challenges

There are times when the NAMD run will take longer than the wallclock limit on the remote batch system. In this case, the batch job is aborted, and does not have any time left-over to save the output and restart files. Instead of transferring files in the remote batch job, NAMD-G submits an interactive process immediately after the job’s completion that takes care of all necessary post-job file transfers.

Another problem with hitting the wallclock limit is that another remote batch job needs to be submitted to finish the simulation. NAMD-G first detects the job having hit the wallclock limit by searching the standard output of NAMD for lines stating that NAMD is writing restart files; this indicates that NAMD has been successfully running. NAMD-G then checks that there are no error messages. NAMD-G finally takes advantage of the fact that DAGMan repeats a job if it sees a non-zero exit code. However, the challenge arises that the same job should not be repeated with the same input and restart files. This is resolved by adopting a naming convention of *runname-jobcount* for the files. NAMD-G keeps track of the current jobcount in a text file which is also transferred to the remote system as part of the job.

One of the issues that had to be worked around while developing NAMD-G was that the local machines were located behind a firewall. This means that Globus (and thus Condor-G) could not auto-

matically retrieve the standard output and error for a job. NAMD-G works around this by making Globus store the standard output and error in temporary files which are later pulled from the remote system by NAMD-G using uberFTP.

Finally, when using GSI authentication, problems will arise if proxy certificate expires while a simulation is still being managed by NAMD-G. The default time limit for a proxy certificate created by the `grid-proxy-init` command is 12 hours. Batch jobs will almost always sit in the queue (idle or running) for longer than that. If Condor-G sees a new proxy certificate on the local machine (the result of the user manually getting a new proxy certificate), it will automatically forward it to the remote system. Condor-G can also automatically retrieve a new proxy certificate from a MyProxy [2] server when it detects that a proxy certificate will soon expire. This capability is utilized by NAMD-G.

## 4 Initial Application

A good example of tackling protein dynamics from a protein-wide point of view is the case of gas migration inside proteins. Many proteins interact with gas molecules, such as oxygen and carbon monoxide, to perform their function. Unlike those of other ligands, the reactive sites for gases are often buried deep inside proteins with no obvious entry pathway. Describing the locations of the gas pathways and the mechanism by which the gases migrate is an essential step in understanding how proteins such as globins, oxygenases, cytochrome-oxidases and many others work. The direct applications of this knowledge can be very relevant. For example, the knowledge of gas migration pathways inside the hydrogenase enzyme allows scientists to perform targeted mutations that lead to the goal of designing a novel enzyme that can be used for affordable large-scale production of  $H_2$  as a source of renewable energy [6]. Numerous simulations using different methodologies have been performed over the years to identify specific gas migration pathways inside various proteins [*e.g.*, 9, 8, 1, 11, 3, 6]. Recently, an algorithm was developed for efficiently identifying all gas migration pathways and free energy barriers in a given protein system [5], such as is shown in Figure 4 for the case of sperm whale myoglobin. Instead of performing a small human-manageable number of simulations regarding a specific system, the development of NAMD-G has created the possibility to easily and conveniently run a large number of con-

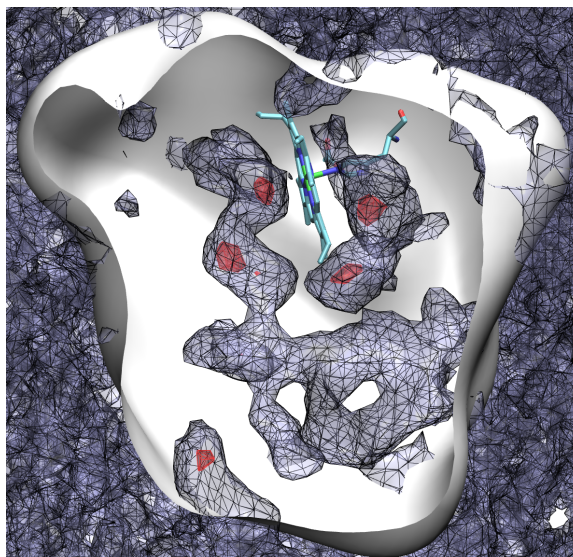


Figure 4: The potential of mean force for oxygen molecules inside sperm whale myoglobin. The very attractive holding regions are shown in red, and the areas that are relatively easily accessible to oxygen gas are surrounded by a black mesh.

current simulations to study the gas migration properties in different systems, thus allowing scientists to find the general principles that regulate gas migration in all proteins. A study of gas migration inside myoglobins has recently been published [5]. That study used NAMD-G on a small sample of four different myoglobins from different species. A larger study of the general properties of gas migration inside a broad range of proteins is currently under way.

## 5 Conclusions

The development and application of NAMD-G has demonstrated that Grid technologies can improve the efficiency of scientists performing biomolecular simulations, and can dramatically transform the types of calculations that they can undertake. Throughout the development process, the authors have found that maintaining a close relationship between the developers and end-user scientists is essential, as regular direct interchanges are absolutely necessary to produce a useful tool. The existing middleware Grid technologies have proven quite valuable as infrastructure components, enabling a single developer to assemble a functional system in under a year by wrapping an

application-specific facade around already-available Grid services. In addition to this unified interface, scientific users demand that new tools integrate with their regular workflow at the desktop. This integration is aided by continuous direct access to the user's local file system, an advantage of NAMD-G over web-based "portal" interfaces that require user intervention to download output files.

## Acknowledgments

This work was supported in part by the National Science Foundation grants SCI-0451538, SCI-0504064, and SCI-0438712. Funding for the Resource for Macromolecular Modeling and Bioinformatics is provided by the National Institutes of Health grant NIH P41 RR05969. The authors gladly acknowledge supercomputer time provided by the National Center for Supercomputing Applications and the Pittsburgh Supercomputer Center via Large Resource Allocation Committee grant MCA93S028.

## References

- [1] P. Amara, P. Andreoletti, H. M. Jouve, and M. J. Field. Ligand diffusion in the catalase from *Proteus mirabilis*: A molecular dynamics study. *Prot. Sci.*, 10:1927–1935, 2001.
- [2] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository. *Software: Practice and Experience*, 35(9):801–816, July 2005.
- [3] C. Bossa, A. Amadei, I. Daidone, M. Anselmi, B. Vallone, M. Brunori, and A. D. Nola. Molecular dynamics simulation of sperm whale myoglobin: Effects of mutations and trapped CO on the structure and dynamics of cavities. *Biophys. J.*, 89:465–474, 2005.
- [4] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. A national-scale authentication infrastructure. *IEEE Computer*, 33(12):60–66, 2000.
- [5] J. Cohen, A. Arkhipov, R. Braun, and K. Schulten. Imaging the migration pathways for O<sub>2</sub>, CO, NO, and Xe inside myoglobin. *Biophys. J.*, 2006. Submitted.
- [6] J. Cohen, K. Kim, P. King, M. Seibert, and K. Schulten. Finding gas diffusion pathways in proteins: Application to O<sub>2</sub> and H<sub>2</sub> transport in CpI [FeFe]-hydrogenase and the role of packing defects. *Structure*, 13:1321–1329, 2005.
- [7] K. Czajkowski, I. Foster, N. Karonis, S. Martin, W. Smith, and S. Tuecke. A resource management architecture for metacomputing systems. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 62–82. Springer Verlag, 1998. Lect. Notes Comput. Sci. vol. 1459.
- [8] R. Czerminski and R. Elber. Computational studies of ligand diffusion in globins: I. leghe-moglobin. *Proteins: Struct., Func., Gen.*, 10:70–80, 1991.
- [9] R. Elber and M. Karplus. Enhanced sampling in molecular dynamics: Use of the time-dependent Hartree approximation for a simulation of carbon monoxide diffusion through myoglobin. *J. Am. Chem. Soc.*, 112(25):9161–9175, 1990.
- [10] National Center for Supercomputing Applications. UberFTP GridFTP-enabled FTP client. <http://dms.ncsa.uiuc.edu/set/uberftp/>, 2004.
- [11] D. R. Nutt and M. Meuwly. CO migration in native and mutant myoglobin: Atomistic simulations for the understanding of protein function. *Proc. Natl. Acad. Sci. USA*, 101(16):5998–6002, 2004.
- [12] J. Phillips, G. Zheng, S. Kumar, and L. Kale. NAMD: Biomolecular simulation on thousands of processors. In *Proceedings of the IEEE/ACM SC2002 Conference, Technical Paper 277*. IEEE Press, 2002.
- [13] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten. Scalable molecular dynamics with NAMD. *J. Comp. Chem.*, 26:1781–1802, 2005.
- [14] D. Thain, T. Tannenbaum, and M. Livny. Conductor and the grid. In F. Berman, G. Fox, and A. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc., April 2003.