

Parallel Implementation of the Replica Exchange Molecular Dynamics Algorithm on Blue Gene/L

M. Eleftheriou¹, A. Rayshubski¹, J. W. Pitera², B. G. Fitch¹, R. Zhou¹, R. S. Germain¹

¹IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598-0218 USA
{mariae,arrayshu,bgf,ruhongz,rgermain}@us.ibm.com

² IBM Almaden Research Center
St. Jose, CA 95120-6099 USA
{pitera@us.ibm.com}

Abstract

The Replica Exchange method is a popular approach for studying the folding thermodynamics of small to modest size proteins in explicit solvent, since it is easily parallelized. However, Replica Exchange can become computationally expensive for large-scale studies, due to the number of replicas needed as well as interprocessor communication requirements both between and within replicas. In this paper we discuss an implementation of Replica Exchange Molecular Dynamics on Blue Gene/L for performing large scale simulation studies of systems of biological interest. The algorithm is tuned with an awareness of the physical network topology and hardware performance features of the Blue Gene/L architecture. Performance measurements for Replica Exchange using the Blue Matter Molecular Dynamics application are presented on Blue Gene/L hardware with up to 256 replicas simulated on 8,192 compute nodes. Both scalability and performance are achieved with this implementation.

1 Introduction

The mission of the Blue Gene/L science effort is to provide useful insight in understanding protein folding and biomolecular dynamics through large scale computer simulations. As part of this effort the Blue Gene application and science team has developed a Molecular Dynamics framework, Blue Matter [11, 14], to tackle this problem using massively parallel supercomputers, such as Blue Gene/L [13].

One of the scientific projects currently running on the Blue Gene/L systems within IBM is the simulation of protein conformational changes via an enhanced sampling method called Replica Exchange Molecular

Dynamics (*REMD*) [16, 19, 23, 30, 22, 28, 29].

Molecular simulations of protein systems at the relatively low temperatures relevant for biology are very difficult because the molecules can become trapped in multiple local minima [4, 8]. A commonly used computational method for efficient sampling of the phase space of large biomolecular systems with rough potential energy surfaces is Replica Exchange Molecular Dynamics [12, 21, 20]. The main idea of REMD is to perform concurrent simulations of M different replicas of the molecular system, each running under different thermodynamic conditions, in our case, temperature. Molecular systems at high temperatures are generally able to explore a larger volume of the phase space than at low temperatures. Replica exchange achieves better sampling by allowing systems at different temperatures to exchange configurations. Replica exchange simulations alternate between phases of conformational sampling and swapping. During the sampling phase, each replica undergoes normal molecular dynamics or Monte Carlo sampling at its assigned temperature. During the swapping phase, replicas are exchanged between temperatures by a stochastic process that uses a Metropolis-like acceptance criterion. However, the alternating phases of the calculation introduces a weak coupling between different replica runs. This weak coupling may introduce inefficiencies in the simulation, since it requires all the processors to be synchronized.

Traditionally, Replica Exchange simulations, due to the relatively small communication requirements between the replicas, are performed on distributed computing clusters with each replica running on single CPU. Although this is sufficient for small to modestly sized molecular systems, it becomes a problem when one wants to run large size protein systems for long simulation times. For such simulations, parallel exe-

cution of the individual replicas is required which in turn necessitates a high performance interconnect between the CPUs used for each replica. If the individual replicas execute sufficiently quickly, even the communication step between replicas can introduce significant inefficiencies since the swapping step requires synchronization between all replicas [22].

We are currently interested in simulating molecular systems of size $\approx 70\text{K}$ atoms using the replica exchange method with 256 replicas. Such simulations exercise both the capability and the capacity aspects of the BG/L supercomputer. Replica exchange simulations of this scale can be performed on a large cluster system such as Mare Nostrum, but almost the whole machine would be required to obtain performance equivalent to that of Blue Gene/L [25]. This illustrates the capacity aspect of Blue Gene/L, since it permits the use of 8,192 nodes in order to enable simulations using 256 or more replicas simultaneously. Long simulation times (on the order of ($\approx 100\text{ns}$)) for a relatively large protein system, $\approx 67,000$ atoms, are enabled by parallelization of the individual replicas.

In this paper we describe an efficient and scalable parallel implementation of the REMD algorithm. Unlike conventional REMD parallel implementations which assign a single processor to each replica, we distribute individual replicas over disjoint subsets of processors. That is, each replica trajectory is computed in parallel. This not only allows scalability to larger number of nodes than the number of replicas but also improves the “total time to solution” of individual replicas. Unlike many implementations of the REMD method, the communication step between replicas is performed at the application level using MPI, instead of using an external script. We also provide a comparison to the performance of AMBER8 [7]. Our implementation outperforms AMBER8’s for both the MPI and scripting approach on 8,192 compute nodes on the BG/L supercomputer. Finally, we report both weak and strong scaling measurements in the context of REMD.

The paper is organized as follows: In the next section we give a brief overview of the serial REMD algorithm. Section 3 describes the parallelization approach we took to produce a scalable algorithm. Section 4 describes the performance experiments and reports the scalability of the algorithm using different mesh topologies and numbers of processors for each replica. Finally, we present our conclusions and suggestions for further improvements.

2 Replica Exchange algorithms

The free energy landscape of a protein is very rugged due to the comparatively close packing of the atoms in the native structure. Methods such REMD were introduced to overcome difficulties in efficiently evaluating thermodynamic properties of solvated protein systems and other materials with challenging energy landscapes. We describe two commonly used versions of the algorithm below.

2.1 Ordered Replica Exchange algorithm

In this section we provide a brief discussion of REMD method based on Okamoto’s velocity rescaling approach [23, 24]. We create M separate systems each with an independent set of coordinates and velocities. The systems can all be initialized with the same configuration or with a variety of different initial configurations. Each system or replica has the same number and type of atoms, but samples a different thermodynamic state, here different temperature, T_m , where $m=1,2,3,\dots,M$. There is a one-to-one correspondence between replicas and temperatures. A permutation function maps each replica to the corresponding temperature. All the replicas are simulated concurrently and independently for N MD time steps. Every N time steps we iterate over all the temperature pairs of replica with neighboring temperatures. A swap between two neighbor temperatures (or replicas) is attempted. The swap attempts to move the lower temperature replica to the higher temperature and vice versa. The proposed swap (x') is accepted or rejected based on the Metropolis criteria with probability,

$$T(x \rightarrow x') = T(x_m^i x_n^j \rightarrow x_n^i x_m^j) \quad (1)$$

$$= \begin{cases} 1, & \text{for } \Delta \leq 0, \\ \exp(-\Delta), & \text{for } \Delta > 0. \end{cases} \quad (2)$$

$$\Delta = (\beta_m - \beta_n)(V(q^j) - V(q^i)) \quad (3)$$

$$\beta_m = \frac{1}{k_B T_m} \quad (4)$$

where T_i and $V(q_i)$ is temperature and the potential energy of the configuration q , of the i th replica, respectively. If the given swap is accepted, we update the permutation functions and rescale or reassign velocities for each replica so they are representative of the new temperatures. The acceptance rule above is appropriate for replica exchange simulations where each replica is simulated in the canonical ensemble. Different criteria are

used for other ensembles, like isothermal-isobaric. The above algorithm is repeated every N iterations during the duration of the Molecular Dynamics simulation.

2.2 Stochastic Replica Exchange algorithm

The stochastic replica-exchange method is a modification of the ordered replica-exchange, in which the higher temperature from the replica pairs used in the swapping step is picked randomly from a uniform distribution. In this approach, although the modification is trivial, the replicas can move by more than one temperature increment during the swapping phase and swaps approach the limiting distribution faster. On the order of $5 * M$ swaps are attempted during the stochastic exchange phase.

2.3 Acceptance ratio and number of replicas

Both replica exchange algorithms improve sampling by allowing each replica to sample a range of temperatures (and thus a broad range of potential energies). This requires a non-vanishing acceptance probability for the swapping moves. Typically, acceptance probabilities between 0.2 and 0.5 are considered acceptable. The acceptance probability for a swap between a pair of temperatures depends on the potential energy distributions sampled at those two temperatures. The high temperature replica needs to sample potential energies that are accessible at the low temperature, and vice versa.

The average potential energy is proportional to the number of atoms N_a in the system, and its variance is proportional to $\sqrt{N_a}$. As the system size increases, the energy distributions for the two temperatures overlap less and less, driving the acceptance probability downward. Replica exchange simulations of protein folding need to include a range of temperatures that span the folding transition temperature. Above this temperature, the protein is largely unfolded, while below this temperature it is largely confined to the folded state. To ensure that the simulations span the melting temperature, protein folding replica exchange simulations use a broad range of temperatures from below 270 to above 600 K. For a small system (5-10,000 atoms), this range can be covered by 64 exponentially spaced replicas with adequate acceptance of swaps. If the system size is increased tenfold, roughly $64 * \sqrt{10}$ replicas are required for the same acceptance ratio. The largest system used in this study requires the simultaneous simulation of 256 replicas.

3 Parallel algorithm and Implementation

During the replica exchange molecular dynamics method, a large number of simulations are run at different temperatures and periodically exchange information using a Metropolis Monte Carlo scheme. Thus, we exploit parallelism at two levels:

- Within each replica, that is, each individual Molecular Dynamics trajectory.
- Among replicas, that is, during the Monte Carlo step (swapping step)

The communication within replicas is tightly coupled, since it requires exchange of force and position data at every time step (time iteration). In contrast, the communication among replicas is more loosely coupled since it only requires a periodic global exchange of data between all of the processors.

3.1 Parallel decomposition

The BG/L processor pool is connected as a three dimensional Cartesian torus and it is partitioned into disjoint groups of processors. Each group of nodes is responsible for the simulation of a single replica. There are two natural approaches to partitioning the machine into disjoint domains: rows or columns of processors and compact rectangularly shaped groups of processors. Because certain MPI collectives are highly optimized for rectangular partitions of the torus network, this is the partitioning scheme currently implemented in REMD. Specifically, the Blue Matter application performs most of its communications with `MPI_Alltoallv` collective operations. The current `MPI_Alltoallv` collective is optimized well on the Blue Gene/L torus network [2] and it is suitable for latency limited applications since all six links on the torus can be used to send messages. The implementation used for this work doesn't use the row/column multi-cast capabilities of the hardware, that is, not well optimized for row/columns shaped machine partitions. An implementation of the all-to-all collective that leveraged hardware multi-cast might enable further improvements in performance of the `MPI_Alltoallv` on row/column shaped machine partitions.

3.1.1 Parallelism within replicas:

The machine is divided into compact M domains of size $\frac{P_x P_y P_z}{M}$, where P_i is the number of processors in a given direction i . Each domain contains the same

number of independent nodes. The shapes of these domains can be compact rectangles (currently) or single rows/columns. The processors within a domain carry out a (parallel) molecular dynamics simulation of the replica assigned to that domain. Within each replica we are using a new variant of spatial decomposition for n -body simulations, to map the machine on the 3D torus network of the BG/L supercomputer and to improve load balancing. The latter is a major challenge for n -body simulations[14]. The main difference between the conventional spatial decomposition approach and our approach is that we equally distribute the number of work units (number of interactions that need to be calculated) instead of the atoms on the machine. More specifically, after we evaluate all the possible pair chemical interactions between atoms in the molecular system, we assign each interaction to the node that is roughly located in the middle of the two nodes where the atoms involved in that pair interaction reside. To ensure that each node has the same amount of work to perform, we use an orthogonal recursive bisection (ORB) scheme to partition the total work among the processors. A more detailed overview of the Blue Matter application framework and its performance on the Blue Gene/L supercomputer is published elsewhere [15, 11].

3.1.2 Parallelism between replicas:

Early Replica Exchange implementations for the most commonly used molecular dynamics packages [17, 1, 3, 5] utilized a script based approach to perform the swapping step among the replicas. However, such approaches have significant I/O requirements and disk I/O is slow compared to the speed of execution for a time step on the Blue Gene/L supercomputer. This could be a problem if the swaps performed are relatively frequent as compared to the number of time steps (iterations) run in between swaps. We use an integrated application approach, where the data exchange between replicas is performed on top of the MPI communication layer. The swapping step requires the collection of data from all the processors. During that step, each task in the domain M performs a reduction step to calculate the potential energy $V(q)$ of its replica. Then, each master node of each group of nodes, broadcasts its potential energy value to all processes in MPI_COMM_WORLD. Alternatively, we can replace those two collective operations by using a single MPI_ALLREDUCE collective operation on MPI_COMM_WORLD. This second implementation produces significant speedups for the small messages involved.

These two steps are combined in one all-reduce call as follows. Let's assume we have D machine partitions (that is, number of replicas), and $P = P_x \times P_y \times P_z$ number of nodes. Each processor in MPI_COM_WORLD allocates an array of size P , number of processors in the whole simulation. Then each node fills all elements with zeros except the element with index $myReplicaID \times \frac{P}{D}$. That element is filled with the partially potential energy of that sub-partition. An all-reduce operation on the MPI_COMM_WORLD communication with a MPI_SUM operations will make all replica's potential energies available on every node zero on all subpartitions. with the each node to have an array of potential energies at different temperatures. After all the nodes receive the potential energies for each replica, we perform the whole swapping step locally on every processor. Each node computes the swap ratio for all selected replica pairs and decides whether to swap or not. By replicating the swapping step, we avoid an additional communication phase that would be needed for broadcasting the new replica-temperature mapping.

The parallel decomposition of the replica exchange method with 4 replicas on $8 \times 8 \times 8$ mesh of nodes, is shown in Figure 3.1.2. Each replica is distributed on $4 \times 4 \times 4$ processor partition.

4 Performance Results

4.1 Benchmarked Molecular System

The protein used for these simulations is a truncated, monomeric form of the lambda repressor DNA binding domain that has been engineered to fold quickly and reversibly by the introduction of five mutations (D14A/Y22W/Q33Y/G46A/G48A) [27]. The protein consists of 80 amino acids and its folded state consists of a bundle of five alpha helices. This particular protein is of interest since it is one of the fastest folding small proteins known, with folding rates under 10 microseconds. It is also of interest because spectroscopic experiments show that lambda repressor is folding at or near the "speed limit" for a protein of its size. It thus represents an excellent opportunity to simulate the full folding process of a realistic protein using molecular dynamics.

The simulated system consists of the protein (1258 atoms), one chloride counter ion, and 21823 water molecules for a total of 66,728 atoms. The protein is modeled with the AMBER96 force field [6] and the water molecules with the TIP3P potential [26]. The system is simulated at constant volume (after a constant pressure equilibration) with cubic periodic boundary conditions. The box edge length is 88.027Å, sufficient

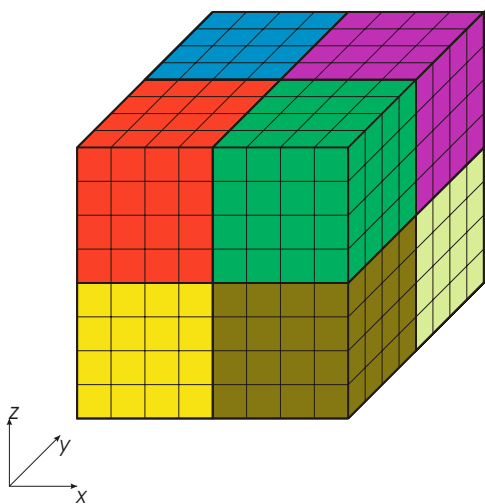


Figure 1. Example of a partition of the the three-dimensional processor mesh into replicas. The processor mesh dimensions are $8 \times 8 \times 8$ and each replica's sub-partition dimensions are $4 \times 4 \times 4$.

to encompass the expected dimensions of highly unfolded states (50-54Å diameter). Long range electrostatics are treated with Particle Mesh Ewald, and a 9/9.5Å switch was used for direct-space electrostatics and Lennard-Jones interactions.

4.2 Scalability studies

In this section we present performance measurements of the Blue Matter Replica Exchange application on the Blue Gene/L supercomputer at IBM T. J. Watson Research Center, that consists of 20,480 compute nodes. All the benchmark were done using the MPI based implementation of the Blue Matter code. Although we run the experiments in the co-processor mode, the code uses both cores to allow a limited amount of overlap between communication and computation [14]. Here we report experimental data for the lambda repressor system. All the benchmarks reported here perform a swap step every 1000 time steps (iterations).

Figure 4.2 shows the measured speed up for the Blue Matter replica exchange application on Blue Gene/L system. In that benchmark we increase the number of nodes/replica (decrease the number of replicas) while keeping the total number of nodes constant at 4096. This is equivalent to strong scaling and the data for performance scaling of a REMD trajectory is approximately the same as that obtained for strong scaling of an independent trajectory. Thus, individual replicas can scale to large number of nodes.

Figure 4.2 shows the weak scaling performance (constant work per compute node) for the Blue Matter Replica Exchange application. In this benchmark we increase both the number of replicas and number of compute nodes at the same rate. That is, each replica always runs on 32 compute nodes. For scientific interest, this will enable very long thermodynamical simulations of individual replicas needed to make necessary contact between experiment and simulation.

We compare the overall performance of Replica Exchange molecular dynamics method of two different Molecular Dynamics packages, Blue Matter and AMBER8. For the AMBER8 benchmark the Molecular Dynamics kernel used is SANDER, and the communication between replicas was performed using the MPI communication layer and scripts. The Replica Exchange Molecular Dynamics overall throughput per time step (iteration) is about 22msec and 34msec for Blue Matter and AMBER8 MPI-based, respectively. The amortized time per time step for the AMBER8 script-based implementation is extremely slow. More detailed explanation of the performance within replicas and between replicas is given below.

4.2.1 Performance measurements within each replica

In an attempt to characterize the difference in performance due to the topology of the subpartitions used for each replica, we performed the following simulation experiments. We measured the average execution times for REMD simulations of 128 replicas running on a 4096 node machine for different choices of sub-partition shapes. In the first benchmark all the replicas ran on $2 \times 2 \times 8$ and for the second on $2 \times 4 \times 4$ machine partitions, their average execution time per time step were 30 ms and 22 ms respectively. We are reporting these numbers to emphasize the importance of choosing the most cubic-“like” decomposition to run single replicas. A variation in the mesh topology of individual replicas may cost on the order of 30% or more in performance. The sensitivity of the performance to sub-partition shape is probably due to two factors in

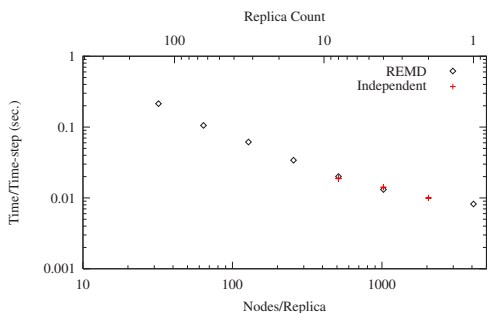


Figure 2. Strong scaling data showing the time per time-step as the number of replicas is varied for a fixed number of nodes (4096). The upper x-axis shows the number of replicas (in the REMD runs) while the lower x-axis shows the number of nodes per replica (in the REMD runs) or the number of nodes (for the “independent” strong scaling runs). Note that the performance within the REMD implementation is essentially the same as that measured for an independent trajectory.

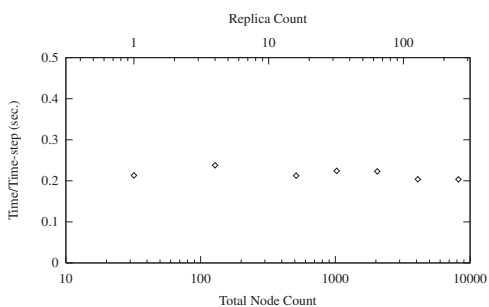


Figure 3. Weak scaling data showing the time per time-step as the number of replicas is varied for a fixed number of nodes/replica (32). The upper x-axis shows the number of replicas while the lower x-axis shows the total number of nodes.

the application software: First, the 3D FFT bisectional bandwidth [10, 9] and secondly, the geometric shape of the ORBs, that is if the ORB sub-volumes are cubic the broadcast communications are done between a smaller number of nodes.

4.2.2 Performance between replicas

We made measurements to compare the performance of the replica swapping step using a Perl script to MPI based implementation on Blue Matter using the 1LMB

molecular system with 256 replicas. Both benchmarks were done on 8192 BG/L nodes. The script overhead during the swapping step for 256 replicas is 1 – 2min versus the 0.0016sec for the MPI-based approach in Blue Matter. Most of this large overhead is due the file i/o required by the script-based approach (to store and retrieve simulation data). Inefficient swapping can potentially become a significant performance problem if the frequency of attempted Monte Carlo steps is high.

5 Conclusion

We have described the implementation of Replica Exchange in the Blue Matter application framework on Blue Gene/L. This implementation exploits parallelism at two levels, one within the replicas and the other among the replicas. Using that technique, one can scale up efficiently to large numbers of nodes.

The performance results demonstrate that massively parallel computers such as Blue Gene/L are suitable for large scale Replica Exchange simulations. The parallelization of Replica Exchange Molecular Dynamics provides two advantages: significant speed up of the computation of a single replica and efficient performance of the computation of the swapping step. Scalability of individual replicas to large numbers of nodes is necessary to extend the simulation time. In addition, new methods such as REST [18] which require a smaller number of replicas for equivalent size systems can benefit the most because individual replicas can run on 512 nodes or more. Efficient performance of the swapping step implies that we can increase the number of replicas without significant overhead. Overall, our parallelization approach makes the simulation of large proteins systems with extended simulation time and with large number of replicas feasible on massively parallel machines such as Blue Gene/L.

Future work includes the incorporation of additional Replica Exchange methods into the BlueMatter framework and characterizing their performance on Blue Gene/L.

Acknowledgments: We would like to acknowledge Hao Yu and Gheorghe Almasi on useful discussion on the performance of the MPI collective operations.

References

- [1] Impact. <http://www.schrodinger.com>.
- [2] G. Almaši, C. Archer, C. Chris Eway, Philip Heidelberg, X. Martorell, J. E. Moreira, B. D. Steinmacher-Burow, and Yili Zheng. Optimization of MPI collective operations on Blue-Genesystems. In *International Conference on Supercomputing archive Proceedings of the 19th annual international conference on Supercomputing*, pages 253 – 262, 2005.
- [3] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.
- [4] C. L. Brooks, J. N. Onuchic, and D. J. Wales. Taking a walk on a landscape. *Science*, 293:612–613, 2001.
- [5] D.A. Case, D.A. Pearlman, J.W. Caldwell, T.E. Cheatham III, J. Wang, W.S. Ross, C.L. Simmerling, T.A. Darden, K.M. Merz, R.V. Stanton, A.L. Cheng, J.J. Vincent, M. Crowley, V. Tsui, H. Gohlke, R.J. Radmer, Y. Duan, J. Pitnera, I. Massova, G.L. Seibel, U.C. Singh, P.K. Weiner, and P.A. Kollman. *AMBER 7*. University of California, San Francisco, 2002.
- [6] W.D. Cornell, J.W. Caldwell, and P.A. Kollman. Calculation of the phi-psi mpas for alanyl and glycyl dipeptides with different additive and non-additive molecular mechanical models. *Journal Del Chimie Physique Et De Physico-Chimie Biologique*, 94:1417–1435, 1997.
- [7] et. al. D. A. Case. *AMBER8*. University of California, San Francisco, 2004.
- [8] K. A. Dill and H. S. Chan. From levinthal to pathways to funnels. *Nature Struc. Bio.*, 4:10–19, 1997.
- [9] M. Eleftheriou, B. Fitch, A. Rayshubskiy, T.J.C. Ward, and R.S. Germain. Performance measurements of the 3d FFT on the Blue Gene/L supercomputer. In J.C. Cunha and P.D. Medeiros, editors, *Euro-Par 2005 Parallel Processing: 11th International Euro-Par Conference, Lisbon, Portugal, August 30-September 2, 2005*, volume 3648 of *Lecture Notes in Computer Science*, pages 795–803. Springer-Verlag, 2005.
- [10] M. Eleftheriou, B.G Fitch, A. Rayshubskiy, T.J.C. Ward, and R.S. Germain. Scalable framework for 3d FFTs on the Blue Gene/L supercomputer: Implementation and early performance measurements. *IBM Journal of Research and Development*, 49(2/3):457–464, 2005.
- [11] B.G. Fitch, R.S. Germain, M. Mendell, J. Pitnera, M. Pitman, A. Rayshubskiy, Y. Sham, F. Suits, W. Swope, T.J.C. Ward, Y. Zhestkov, and R. Zhou. Blue Matter, an application framework for molecular simulation on Blue Gene. *Journal of Parallel and Distributed Computing*, 63:759–773, 2003.
- [12] E. Gallicchio, L. Y. Zhang, and R. M. Levy. Free energy surfaces of beta-hairpin and alpha-helical peptides generated by replica exchange molecular dynamics with the agbnp implicit solvent model. *Proteins*, 56:310–321, 2004.
- [13] A. Gara et al. Overview of the Blue Gene/L system architecture. *IBM Journal of Research and Development*, 49(2/3):195–212, 2005.
- [14] Robert S. Germain, Blake Fitch, Aleksandr Rayshubskiy, Maria Eleftheriou, Michael C. Pitman, Frank Suits, Mark Giampapa, and T.J. Christopher Ward. Blue Matter on Blue Gene/L: massively parallel computation for biomolecular simulation. In *CODES+ISSS '05: Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 207–212, New York, NY, USA, 2005. ACM Press.
- [15] R.S. Germain, Y. Zhestkov, M. Eleftheriou, A. Rayshubskiy, F. Suits, T.J.C. Ward, and B.G. Fitch. Early performance data on the Blue Matter molecular simulation framework. *IBM Journal of Research and Development*, 49(2/3):447–456, 2005.
- [16] K. Hukushima and K. Nemoto. Exchange monte carlo method and application to spin glass simulations. *J. Phys. Soc. Japan*, 65:1604–1608, 1996.
- [17] E. Lindahl, B. Hess, and D. van der Spoel. Gromacs 3.0: A package for molecular simulation and trajectory analysis. *J. Mol. Mod.*, 7:306–317, 2001.
- [18] Pu Liu, Byungchan Kim, Richard A. Friesner, and B.J. Berne. Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proceedings of the National Academy of Sciences*, 102:13749–13754, 2005.

- [19] E. Marinari, G. Parisi, and J. J. Ruiz-Lorenzo. A. p. young (ed.), spin glass and random fields. *World Scientific, Singapore*, page 59, 1998.
- [20] H. Nymeyer and A. E. Garcia. Interfacial folding of a membrane peptide: Replica exchange simulations of walp in a dppc bilayer. *Biophys. J.*, 84:381A, 2003.
- [21] J. Pitera and W. Swope. Understanding folding and design: replica-exchange simulations of the trp-cage miniprotein. *Proc. natl. Acad. Sci. U.S.A.*, 100:7587–7592, 2003.
- [22] Y. M. Rhee and V. S. Pande. Multiplexed-replica exchange molecular dynamics method for protein folding simulation. *Biophys. J.*, 84:775–786, 2003.
- [23] Y. Sugita and Y. Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.*, 314:141–151, 1999.
- [24] Y. Sugita and Y. Okamoto. Replica-exchange multicanonical algorithm and multicanonical replica-exchange method for simulating systems with rough energy landscape. *Chem. Phys. Lett.*, 329:261–270, 2000.
- [25] William Swope. private communication.
- [26] W.L.Jorgensen, J. Chandrasekhar, J.D. Maduar, R.W. Impey, and M.L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79:926–935, 1983.
- [27] W. Y. Yang and M. Gruebele. Folding at the speed limit. *Nature*, 423:103–197, 2001.
- [28] R. Zhou. Trp-cage: Folding free energy landscape in explicit water. *Proc. Natl. Acad. Sci. U.S.A.*, 100:13280–13285, 2003.
- [29] R. Zhou, X. Huang, C. J. Margulius, and B. J. Berne. Hydrophobic collapse in multidomain protein folding. *Science*, 305:1605–1609, 2004.
- [30] R.H. Zhou, B.J. Berne, and Germain R. The free energy landscape for beta hairpin folding in explicit water. *Proc. Natl. Acad. Sci. USA*, 98(26):14931–36, December 2001.