

# Blue Matter: Approaching the Limits of Concurrency for Classical Molecular Dynamics

Blake G. Fitch\*   Aleksandr Rayshubskiy†   Maria Eleftheriou‡   T.J. Christopher Ward§

Mark Giampapa¶   Michael C. Pitman||

Robert S. Germain\*\*

IBM Research Division

IBM Thomas J. Watson Research Center

1101 Kitchawan Rd Route 134 Yorktown Heights, NY 10598

## Abstract

This paper describes a novel spatial-force decomposition for N-body simulations for which we observe  $O(\sqrt{p})$  communication scaling. This has enabled Blue Matter to approach the effective limits of concurrency for molecular dynamics using particle-mesh (FFT-based) methods for handling electrostatic interactions. Using this decomposition, Blue Matter running on Blue Gene/L has achieved simulation rates in excess of 1000 time steps per second and demonstrated significant speed-ups to  $O(1)$  atom per node. Blue Matter employs a Communicating Sequential Process (CSP) style model with application communication state machines compiled to hardware interfaces. The scalability achieved has enabled methodologically rigorous biomolecular simulations on biologically interesting systems, such as membrane-bound proteins, whose time scales dwarf previous work on those systems. Major scaling improvements will require exploration of alternative algorithms for treating the long range electrostatics.

**Keywords:** Molecular Dynamics, N-body Simulations, Parallel Programming

## 1 Introduction

---

\*bgf@us.ibm.com

†arayshu@us.ibm.com

‡mariae@us.ibm.com

§tjcw@uk.ibm.com

¶giampapa@us.ibm.com

||pitman@us.ibm.com

\*\*rgermain@us.ibm.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SC2006 November 2006, Tampa, Florida, USA  
0-7695-2700-0/06 \$20.00 ©2006 IEEE

The ability of biomolecular simulations to make contact with experimental data is directly related to the time-scale accessible to the simulation. This necessitates strong scaling of a fixed size N-body problem, typically a system containing tens of thousands to hundreds of thousands of particles, onto a massively parallel computer with thousands or tens of thousands of nodes. For example, to achieve a simulation rate of one microsecond every two weeks requires a single time step to complete within 1.2 milliseconds, equivalent to fewer than one million machine cycles on Blue Gene/L. Continuing to improve overall time-to-solution for these N-body simulations with correct treatment of long range electrostatic interactions while decreasing the ratio of atoms per node to on the order of one atom per node is a tremendous challenge for a parallel application.

Success in meeting this challenge enables detailed atomistic simulations of biologically interesting systems at time-scales and in ensemble sizes that were previously unattainable including 26 hundred nanosecond trajectories of a G-Protein Coupled Receptor (GPCR), Rhodopsin, in a realistic membrane environment[Grossfield et al. 2006] and multiple micro-second scale simulations of that system and others. Furthermore, since the path to increased capability now seems to require increased concurrency, even working with larger systems with hundreds of thousands of atoms may require scalability to relatively small ratios of atoms per node. While there have been some theoretical studies of scaling in this limit[Taylor et al. 1997], this work represents the first implementation of classical biomolecular simulation to demonstrate scaling to this degree.

### 1.1 Background on Blue Matter

The Blue Matter effort was undertaken for two reasons. First, to address one of the primary goals of IBM's Blue Gene project[Allen et al. 2001]: To use the unprecedented computational resource developed during the course of the project to attack grand challenge life sciences problems such as advancing our understanding of biologically important processes, in particular, the mechanisms behind protein folding. Second, to provide a concrete context for the explo-

ration of the algorithmic techniques and programming models required to exploit the massive parallelism of the Blue Gene architecture. Blue Matter has been used in production by computational scientists on the Blue Gene project since 2003, initially on SP2 hardware and later on Blue Gene/L hardware[Swope et al. 2004; Pitman et al. 2004; Pitman et al. 2005a; Pitman et al. 2005b; Suits et al. 2005; Grossfield et al. 2006; Eleftheriou et al. 2006a].

## 2 Classical Biomolecular Simulation

### 2.1 N-body Problem, Long Range Electrostatics

Classical biomolecular simulation includes both Monte Carlo and Molecular Dynamics[Frenkel and Smit 1996]. The focus of our work has been on Molecular Dynamics although the Replica Exchange or Parallel Tempering Method[Sugita and Okamoto 1999] which combines Molecular Dynamics with Monte Carlo-style moves has been implemented in Blue Matter as well[Eleftheriou et al. 2006b]. Classical molecular dynamics is an N-body problem in which the evolution of the system is computed by numerical integration of the equations of motion. At each time step, forces on particles are computed; and then the equations of motion are integrated to update the velocities and positions of the particles.

Because models of proteins have components with large partial charges, the long ranged electrostatic forces cannot be approximated simply by cutting off interactions between pairs of particles further apart than some cut-off distance. The most commonly used techniques for handling these long range interactions are based on the Ewald summation method and particle mesh techniques that divide the electrostatic force evaluation into a real-space portion that can be approximated by a finite range cut-off and a reciprocal space portion that involves a convolution of the charge distribution with an interaction kernel. This convolution is evaluated using a Fast Fourier Transform (FFT) method in the Particle-Particle-Particle-Mesh (P3ME) technique[Deserno and Holm 1998] used by Blue Matter. In this case the  $O(n^2)$  dependence on particle number is reduced to  $O(n \log n)$ . The global data dependency for each time-step is imposed by the convolution step with the evaluation of the three dimensional FFTs.

We describe molecular dynamics as comprised of four major modules:

- real-space non-bonded (finite range pair interactions)
- k-space (FFT-based)

- bonded (graph-based)
- integration (per particle)

### 2.2 Inherent Concurrency of Molecular Dynamics

Before attempting to scale an algorithm onto many thousands of nodes, it is useful to estimate how much concurrency is inherent in various components of that algorithm. First, consider the anatomy of a molecular dynamics time step starting with the availability of the coordinates and velocities of all of the particles in the system ( $\mathbf{r}_i, \mathbf{v}_i$ ):

- Compute forces on each particle due to bonded (intramolecular) interactions.
  - Bond stretches
  - Angle bends
  - Torsions
- Compute forces on each particle due to non-bond interactions (assuming periodic boundary conditions)
  - Dispersion and Van der Waals forces (usually represented by a Lenard-Jones 6-12 potential function that is smoothly switched off beyond some cutoff distance  $r_c$ )
  - Electrostatic forces ( $1/r^2$  forces) (most commonly evaluated using the Ewald summation technique or its mesh-based variants[Deserno and Holm 1998])
- Accumulate the total force on each particle and use that force along with the current position and velocity of particle to propagate the motion of the particle forward in time by some small increment

It is possible to view a molecular dynamics time step (or any parallel computation) as the successive materialization of distributed data structures on which local computation takes place. Given a choice of granularity below which no parallelism will be attempted and taking into account the data dependencies in the algorithm, one can estimate the number of independent computations required at each phase. An example of such an analysis is pictured in Figure 1 for the non-bonded forces in a Molecular Dynamics simulation using the P3ME method to treat the long-range electrostatic forces. One can afford a lack of concurrency in components that impose very little computation or communication burden, but eventually even these will become bottlenecks (Amdahl's Law).

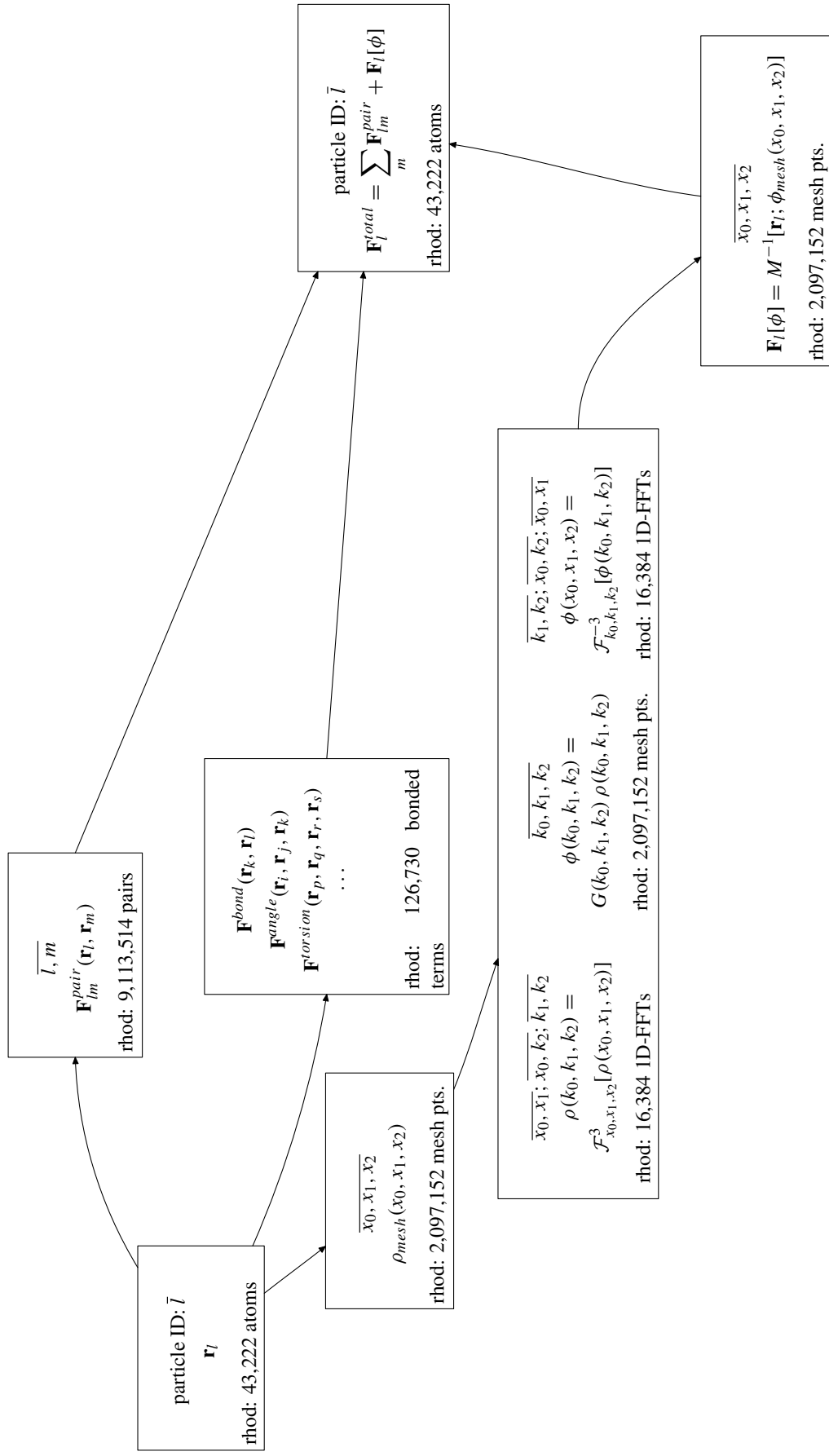


Figure 1: Diagram showing the data dependencies and opportunities for concurrency in various stages of the non-bonded force calculations in a Molecular Dynamics time step using the P3ME method. For each step, the actual number of independent work items are displayed for one of the molecular systems, Rhodopsin, benchmarked in this paper. Three separate threads of computation are shown: finite-ranged pair interactions, bonded interactions, and long-ranged electrostatic interactions computed via the P3ME method. The large box at the bottom of the figure represents the convolution step which is evaluated by first Fourier-transforming the meshed charge distribution, then multiplying by a kernel (Green's function), and then inverse Fourier-transforming the result to obtain the meshed electrostatic potential. A detailed explanation of the P3ME method shown here can be found in [Deserno and Holm 1998].

## 3 Parallel Decomposition

### 3.1 Introduction

Our explorations of parallel decompositions have included replicated data methods that leverage the hardware facilities of Blue Gene/L to globalize and reduce data structures[Fitch et al. 2003; Germain et al. 2005b] and spatial decompositions. Prior to the present work, the most recent spatial decomposition, which we will refer to as V4, used geometric criteria to determine where the real-space non-bonded interaction between two particles would be computed, namely on whichever node contained the point half-way between the two particles. This provided a large number of distinct units of the computational burden that could be distributed by partitioning space using an Optimal Recursive Bisection (ORB) scheme[Fitch et al. 2005; Germain et al. 2005a; Fitch et al. 2006]. The implementation of this method entailed the broadcast of a particle's position to a sphere with radius  $R_{eff}$ . Nominally  $R_{eff}$  will be half the molecular dynamics cut-off distance for real-space non-bond interactions for both V4 and V5. However, to allow the preservation of particle assignments to nodes over several steps requires the introduction of a guard zone that increases the  $R_{eff}$  beyond half the molecular dynamics cut-off. The size of the guard zone is a tuning parameter. This also provides the particle positions required by the k-space and bonded force modules.

In contrast to the purely geometric approach of V4, the method described here, referred to as V5, uses geometry primarily as a heuristic to prime the set-based optimization process that follows. Where V4 managed data distribution (of particle positions) and reduction (of forces) via a data "push" and caching module, V5 specializes communications between the integrator module and the three force computation modules described above. Where the V4 push/cache method required a distinct communications phase, V5 allows overlap of one module's communication and/or computation with another module's communication and/or computation. On Blue Gene/L, which has two processors per node, modules are scheduled to cores to maximize overlap. Currently the scheduling is static and we place the longest-running module on its own core.

With Blue Matter V5, a programming model resembling Communicating Sequential Processes (CSP) has evolved. Although still rough in implementation, it is clear that the four main modules in Blue Matter are connected by data channels defined by application specific protocols[Hoare 1985]. This model helps in two ways. First it enables the application to provide as much knowledge as possible to minimize communication overhead. Second, integrated application/communication state machines can be compiled directly to hardware interfaces further reducing overheads.

For each time step, after the integration module has run,

a new set of atom positions are sent to each of the force-generating modules. Each force-generating module returns force partial sums to the integration module at the end of an operational phase. The force-generating modules manage work distribution using an initialization or planning phase to configure structures that will be stable for many iterations as well as dynamic activities that occur more frequently to manage the diffusion of particles. Periodically, the lists used to manage the pair interactions that must be computed (the Verlet lists), must be regenerated because of particle diffusion. Less frequently, diffusion also requires the assignment of particles to nodes to be updated. It should be noted that load balancing for both V4 and V5 currently takes place only during a set-up phase and can lead to degradation of load balance over time. In practice, we have found the performance degradation between job restarts is negligible.

### 3.2 Real Space Non-bond Module

The real space non-bond (RSNB) module must efficiently compute the non-bonded interactions for each pair of atoms within a specified cutoff distance. This is a type of N-body problem that involves periodic boundary conditions, diffusion of atoms, and significant density fluctuations over the course of the simulation. The work of computing the force terms between pairs of atoms is generally the finest granularity considered for parallel distribution and is called an "interaction". Each interaction must be computed on each timestep requiring that the interaction be assigned to a specific node and that arrangements be made for this node to receive the positions of and return forces on the interacting atom pair.

The relationship between how particles are distributed and how interactions are assigned to nodes determines the communication and load imbalance costs of an RSNB algorithm and therefore its scaling characteristics. The Blue Matter research effort has considered three major classes of parallel decomposition.

**Particle decompositions** distribute all particle positions to all nodes (replicated data) and then freely assign interactions to nodes to achieve near perfect load balance.

**Interaction or force decompositions** focus on minimizing communications by assigning interactions to condense use of particle position data on each node.

**Spatial decompositions** map the simulation space to nodes in order to reduce communications by achieving data locality and are expected to be particularly beneficial when mapped to mesh-type machine network topologies.

Force decompositions with good parallel scaling characteristics were first described in [Hendrickson and Plimpton

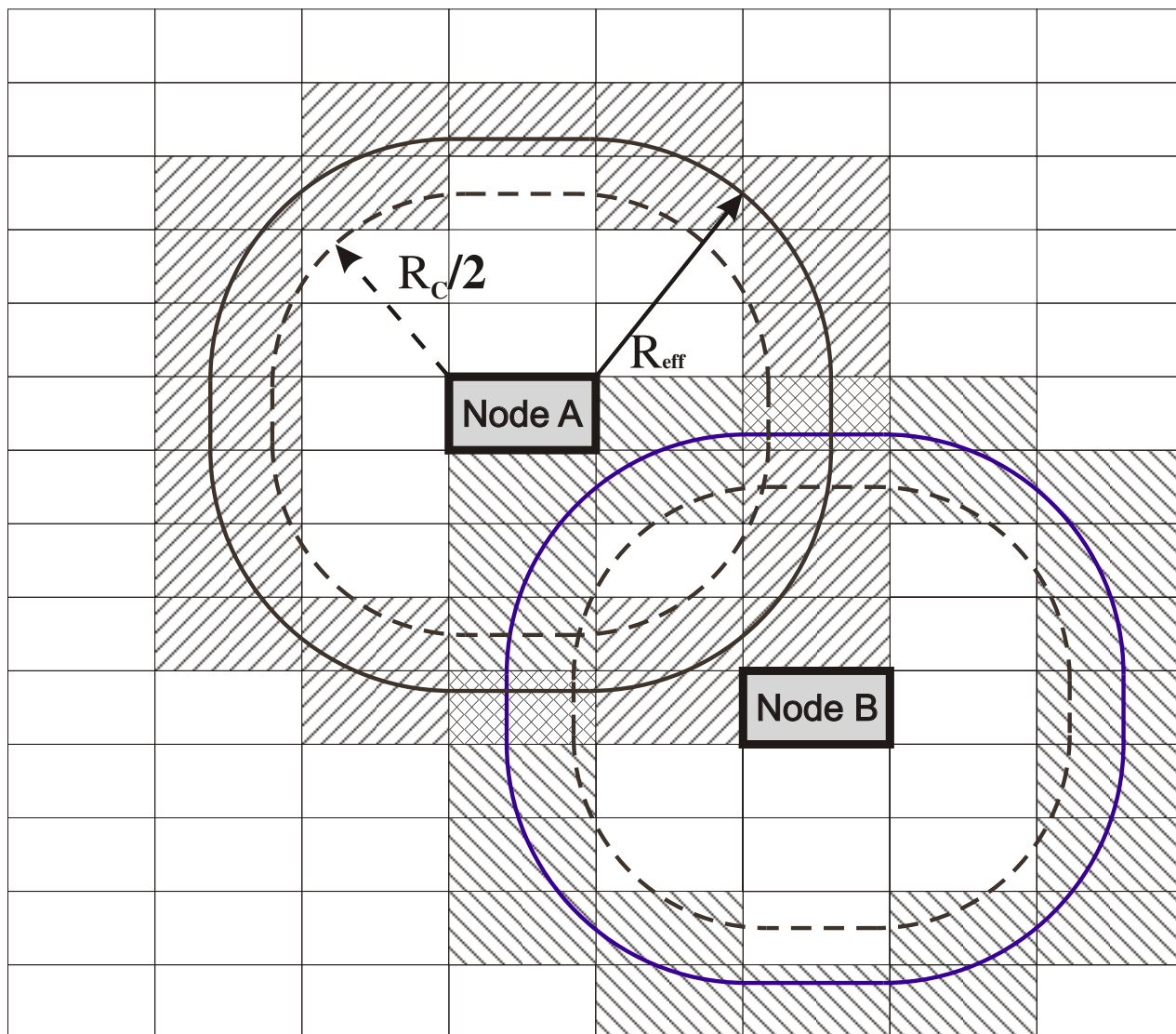


Figure 2: Spatial decomposition showing the “surface interaction sets” for two nodes superimposed on the spatial decomposition of the domain onto all nodes (two-dimensional view for simplicity). The interaction surfaces of nodes A and B are drawn with solid lines at a distance  $R_{eff}$  from the boundaries of the node. The surface sets consist of nodes that are intersected by the interaction surface. The two surface sets are shown in two types of hatching with cross-hatching used to indicate “surface intersection set”. The effective radius  $R_{eff} > R_c/2$  where  $R_c$  is the molecular dynamics cutoff radius. The interaction between a particle stored on Node A and a particle stored on Node B can be computed on any node in the surface intersection set (the nodes with cross-hatching).

1995; Plimpton and Hendrickson 1996] and have the useful property that the number of communicating partners required by each node is  $O(p^{1/2})$  rather than the  $O(p)$  behavior of spatial or replicated data decompositions. This seminal work inspired subsequent research into parallel decompositions that retain the communication scaling behavior of the force decomposition in combination with the data locality achievable by a spatial decomposition [Snir 2004; Shaw 2005]. Through this same conceptual progression, the Blue Matter effort developed what we call the “Wagon Wheel” decomposition independently developed and described by Shaw [Shaw 2004] as the “Neutral Territory” method. Both the Wagon Wheel/Neutral Territory and Snir’s “Hybrid” algorithm achieve attractive theoretical communication characteristics in the extreme strong scaling regime. Force decomposition-inspired hybrid algorithms present implementation challenges in the areas of Verlet list management, coverage of bonded/P3ME communications, and load balancing.

Blue Matter V4 [Fitch et al. 2005; Germain et al. 2005a] was conceptually derived from spatial decompositions and attempts to retain spatial data locality while reducing communication costs to acceptable levels. The V4 decomposition requires that each node send particle positions to a set of nodes (the send-to set) owning any portion of simulation space within  $R_{eff}$  (the broadcast volume). For any pair of nodes within  $2 R_{eff}$  there will be one or more nodes in the intersection of their send-to sets, one of which must be chosen to compute the interaction. Our best V4 assignment method assigns interactions to the node owning space at a point in space midway between the interacting particles. To achieve load balance, the interaction space is assigned to nodes via an optimal recursive bisection (ORB) procedure. V4 is a geometrically assigned spatial-force decomposition.

The Blue Matter V5 algorithm introduced here is a set-based spatial-force decomposition which is conceptually derived from V4 but employs set based optimization techniques to do interaction assignment, minimize communicating node pairs, and achieve load balance. The development of V5 was driven by the fact that the V4 communication pattern requires exchanging data with a rapidly increasing number of nodes,  $O(p)$ , in the strong scaling limit. Analysis of the V4 midpoint assignment method shows that communicating node pairs that are close together do far fewer interactions on each particle received than nodes farther away, representing inefficiency. Outside of the special time steps during which particle migration occurs, essentially all of the data communicated between nodes is consumed in useful computational work.

Since V4 allows assignment of a node pair’s interactions to any node in the intersection of their broadcast volumes, it is possible to create geometric assignment algorithms which assign interactions only to nodes near the skin of the broadcast volume, reducing the RSNB requirement to distribute

particles to nearby nodes. We experienced three problems with attempting to implement V5 purely as a geometric interaction assignment method within the V4 framework:

1. Decrease in the size of the send-to set was less significant than we had hoped.
2. Load balancing using the V4 ORB technique was ineffective because assignment was no longer geometrically continuous (as is the case with assignment of interactions to the point mid-way between the pair).
3. Modules handling the bonded and P3ME force computations no longer likely to receive the particle positions they require as a side effect of the RSNB algorithm and require exchanging data with additional nodes.

These significant issues drive the following principles of V5:

1. Nodes are nominally only added to the send-to set when they own space intersected by a surface, the “interaction surface”, defined by a radius swept from the volume of space owned by the sender; the surface-defining radius is  $R_{eff}$ .
2. Assignment and load balancing is now node based—each pair of nodes owning space within cut-off will have particle interactions done by single node.
3. Each of the force generating modules (bonded, non-bonded, P3ME) will require specialized communications since real space non-bond send-to sets will not cover bonded and P3ME requirements.

V5 uses the interaction surface as a heuristic to prime set-based operations which define an efficient send-to set. A node’s Surface Node Set (the nominal V5 send-to set) is comprised of those nodes which own space intersected by the interaction surface. For each pair of nodes within cutoff, intersecting their surface node sets will produce an Interaction Assignment Option Set (to be defined below), any member of which may be assigned the node-pair’s interactions.

The size of the V5 surface node set scales like the surface of a sphere,  $O(p^{2/3})$ , which is an improvement over the V4  $O(p)$  volumetric scaling behavior. This provides an upper bound for the number of communicating partners for a node. For a given particle count, V5 will compute all of the interactions involving any particular particle using a smaller number of nodes than V4. However, if the program is dominated by communication it may be desirable to further reduce the number of nodes in the send-to set. We have developed sparse surface methods to reduce the final send-to node set in V5 using set-based manipulations.

The most effective sparse surface method is described in the next section. This method assigns the task of computing all of the interactions between two nodes to one of the nodes in the intersection of their surface node sets and makes the

choice in such a way as to minimize the total size of each node's final send-to set.

### 3.2.1 A Method for Constructing Sparse Surface Sets

This section describes the data structures used by V5 in the construction of the sparse surface sets and provides a description of the procedures whereby they are created.

In the implementation of the Blue Matter V5 algorithm, a straightforward domain decomposition of the simulation space onto the node mesh is used so that every node “owns” an equal volume of the simulation space. The set of nodes will be denoted by  $\mathcal{P}$ . The algorithmic description provided below uses a pair of methods to compute distances between node volumes, which we will refer to as node volumes or nodes interchangeably from here on:

$\text{minDistance}(\mathbf{i}, \mathbf{j})$  computes the minimum distance between two nodes  $i$  and  $j$ .

$\text{maxDistance}(\mathbf{i}, \mathbf{j})$  computes the maximum distance between two nodes  $i$  and  $j$ .

**Surface Node Set** There is one such container for every node in the system. The elements in this container are node identifiers of those nodes whose volumes contain any portion of the surface comprising points that are some effective radius,  $R_{eff}$  from the volume of space owned by the “central” node.

{Populate the Surface Node Set,  $\mathcal{S}$  for node  $i$ }

```

for  $j = 0$  to  $P - 1$  do
  if  $j$  is enabled for real space non-bond computation
  then
    if  $\text{minDistance}(i, j) \leq R_{eff} \wedge$ 
       $\text{maxDistance}(i, j) > R_{eff}$  then
       $\mathcal{S}.\text{insert}(j)$ 
    end if
  end if
end for

```

**Interaction Assignment Option Set** This is a two dimensional array of lists where there is one such list for every pair of nodes, called the target node pair, that are within  $2R_{eff}$ . The elements of each list comprise those nodes that could compute an interaction belonging to the target node pair. This is the set of nodes in the intersection of the Surface Node Sets of the nodes in the target node pair.

{Populate the InteractionAssignmentOption Set}

```

for  $i = 0$  to  $p - 1$  do
  for  $j = i$  to  $p - 1$  do
     $C = \text{SurfaceNodeSet}[i] \cap \text{SurfaceNodeSet}[j]$ 
    for all  $k \in C$  do
       $\text{InteractionAssignmentOptionSet}[i][j].\text{append}(k)$ 
    end for
  end for
end for

```

```

end for
end for

```

**SparseSendToNode Set** This is a container identical in structure to the Surface Node Set, but whose elements are a subset of those in the corresponding Surface Node Set and which represent the nodes to which the central node actually has to send particle positions to and from which it receives computed forces.

```

{Construct Sparse Send-to Node Set}
Initialize all elements of InteractingPairAssignment to  $-1$ 
Let sequence  $S = \{(i, j) \in \mathcal{P} \times \mathcal{P} | (i < j)\}$ 
sort  $S$  according to  $\|\text{InteractionAssignmentOptionSet}[i][j]\|$ 
{the size of the assignment option set, smallest first}
for  $k = 0$  to  $\|S\| - 1$  do
   $(i, j) = S[k]$ 
   $C = \text{InteractionAssignmentOptionSet}[i][j]$ 
  if  $\exists a \in \mathcal{P} | a \in (C \cap \text{SparseSendToSet}[i] \cap$ 
     $\text{SparseSendToSet}[j])$  then
    {No need add any nodes to Sparse Send-to Node Sets}
  else if  $\exists a \in \mathcal{P} | a \in (C \cap \text{SparseSendToSet}[i]) \vee (C \cap$ 
     $\text{SparseSendToSet}[j])$  then
    Choose the element  $a$  that appears in the smallest number of SparseSendToNode sets and append it to SparseSendToSet[ $i$ ] and to SparseSendToSet[ $j$ ]
    {One of these appends will be a no-op because  $(a \in \text{SparseSendToSet}[i]) \vee (a \in \text{SparseSendToSet}[j])$  already}
  else
    From  $a \in C$  choose  $a$  |  $a$  appears in the smallest number of SparseSendToNode sets and append it to SparseSendToSet[ $i$ ] and to SparseSendToSet[ $j$ ]
  end if
end for

```

We have observed that the V5 Sparse Surface Send-to Set is substantially smaller than the Surface Node Set set because there is substantial redundancy in the Interaction Assignment Option Set. This redundancy allows our method for constructing Sparse Send-to Node sets to omit a substantial number of nodes from the Surface Node Set. This technique has produced communicating partner node counts of order  $p^{1/2}$  as shown in Figure 3.

Since the surface radius may be arbitrarily greater than the required half cut-off distance, the number of nodes intersected and thus the level of redundancy of assignment options can be expanded for all interacting node-pairs. This tends to increase the distance from the root to nodes in the Sparse Send-to Set. With the ability to reliably produce assignment option sets with many members, it is possible to remove nodes according to a regular pattern. This effectively limits the number of nodes that will be used by the V5 real space module. For example, all odd numbered nodes could be removed from the assignment option sets condensing real space computation to half the nodes. We have tried using a

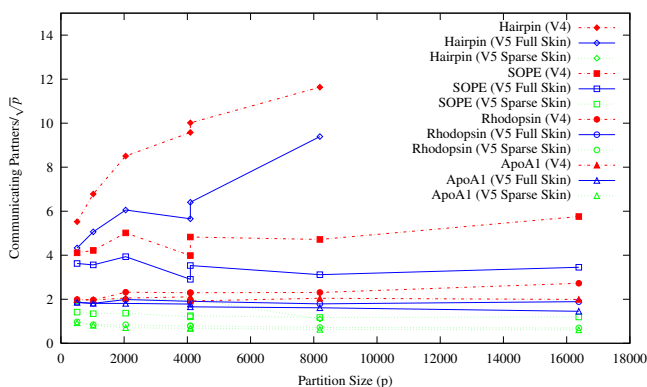


Figure 3: Average number of communicating partners for each node during the neighborhood broadcast and reduce collective operations. The plot shows the communicating partner count divided by  $\sqrt{p}$  as a function of node count,  $p$ , to facilitate comparisons with the scaling behavior of the number of communicating partners in the force decomposition technique invented by Hendrickson and Plimpton [Hendrickson and Plimpton 1995]. The results are shown for the V4 (communication within a spherical volume), the V5 Surface Node Set (communication with a spherical shell), and the V5 Sparse Send-to Node Set algorithms.

$2 \times 2 \times 2$  template which we index using the low-order bits of the torus coordinates of the nodes to mask off eighths of the nominal option set. This has effectively reduced the number of nodes in the send-to sets but has not yet yielded an improved time-to-solution because of increased real space load and imbalance.

### 3.2.2 Interaction Assignment and Load Balancing

In addition to the data structures described in the previous section, we also need the following to assign interactions:

**Interaction Computation Cost** This structure can be represented as an upper triangular matrix indexed by node identifiers whose elements represent the cost in cycles of computing all of the interactions between particles homed in the first node with those homed in the second node.

**Receive-from Set** For every node  $i$  this is the set of nodes that send positions to  $i$ . In effect, this is the transpose of the Sparse Send-to Node Set.

**Interacting Pair Assignment** For every node  $i$  this container is a map keyed by a pair of node identifiers  $(m, n)$ . The node identifiers appearing in the key are all members of the Receive-from Set for  $i$  and the element is a boolean indicating the assignment of  $(m, n)$  to  $i$ . That is, if the value is “true”, all of the interactions between particles homed on  $m$  and  $n$  will be computed on  $i$ .

Given an initial assignment of interactions between pairs of nodes (typically randomized), their associated Interaction Computation Cost, and the Interaction Assignment Option Set described previously, we run a minimization algorithm to create an assignment of node interactions which has the smallest difference between the most heavily burdened and the most lightly burdened nodes. This process yields the Interacting Pair Assignment. Other techniques like the ORB used in V4 could be used but have thus far proved unnecessary.

## 3.3 K-space Module

The V5 k-space module has a distribution function set by a naturally mapped fully distributed FFT [Eleftheriou et al. 2005]. The FFT represents the bulk of the communication time of P3ME, but additional communication is required to send atom position data to those nodes where they will contribute to FFT mesh points and force partial sums resulting from contributions of those mesh points must be returned to the node owning the originating atom at the end of the phase. Since atoms diffuse during the simulation, an atom may contribute to a different set of FFT mesh points during each time step, requiring its position to be sent to different nodes. Ideally, atom positions will only be sent to those nodes managing the appropriate mesh points.

The communication protocol designed for the k-space module on Blue Gene/L takes advantage of the hardware global collective network [Gara et al. 2005; Giampapa et al. 2005]. When the k-space module begins, it receives new position values from the integration module and for each, determines the list of target nodes owning relevant FFT mesh points. Each node then enters a loop sending positions via the hardware torus network to target nodes and receiving positions from other sources. As each node completes sending all local positions to their targets, it contributes a value of the number of torus packets sent to an asynchronous hardware supported integer reduction. The nodes then continue to receive hardware torus packets while polling the global reduction hardware. The first time a global integer reduction completes, each node knows the total number of packets sent. Subsequently and until the receive phase is done, each time an integer reduction completes, each node contributes the number of torus packets received to the next reduction. The phase ends when the global number of torus packets received equals the global number of packets sent. The incremental cost of this method of terminating the communication phase is approximately that of a single integer all-reduce on the global collective network.

### 3.4 Bonded Force Module

The V5 bonded force module is responsible for executing force generating computations on sets of atoms that are connected by covalent bonds (representable as graphs). Because atoms are migrated through the spatial decomposition without regard to these graph based interactions, after atom migration each node with an atom that participates in graph operations that are not fully local must discover which nodes to exchange data with in support of these operations. Since graph based operations can involve up to four atoms and three bonds, the spatial range of nodes that share graph based computations is limited. When atoms are migrated to maintain the spatial decomposition, positions are sent to all nodes which might share a graph based operation. The majority of this set will not share an interaction and this knowledge is preserved for future bonded module communication phases until the next migration time-step. The actual number of nodes exchanging data on non-migration time-steps is the exact number required by the assignment of graph based operations typically an order of magnitude fewer than are in range.

### 3.5 Communication Channel Collectives

We have described Blue Matter V5 as having four main communicating modules. Communication occurs primarily when particle positions are sent from the integration module to the bonded, real space non-bonded, k-space modules, and within the FFT subroutine of the k-space module. Communication also occurs during particle migration and guard-zone violation signal distribution.

While good theoretical communication scaling properties are necessary for strong scaling, it is also essential to reduce any non-scalable overheads. To this end, we have:

1. Preserved communication meta-data which is reused over many time-steps
2. Implemented communication collectives using low level interfaces
3. Overlapped communications with computation by placing force generating modules on both cores of the Blue Gene/L chip

With Blue Matter V4, a single module handled all communications and acted to distribute particle positions and reduce forces for all force generating functions. Overlapping communications with other communications or with computation was not possible since collectives ran as a phase. In V5, conceptually, positions are sent from the integration module on each node to force generating modules on other nodes using “channel collectives”. A channel collective is functionally equivalent to a set of point-to-point communication channels but is implemented to efficiently perform a broadcast or

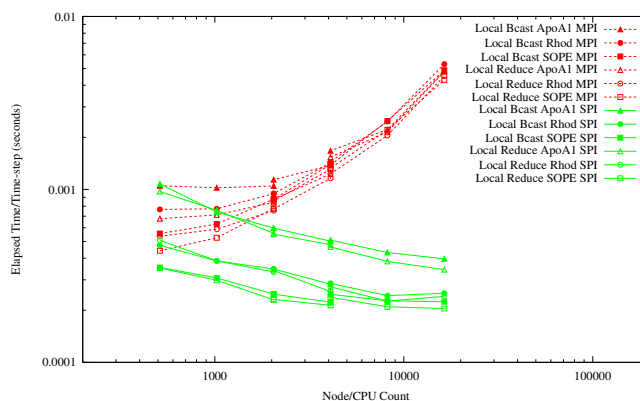


Figure 4: Performance data for the neighborhood broadcast and reduction collectives used in Blue Matter V4 using MPI and BG/L ADE SPI communications interfaces. The measurements were made via trace-points embedded in the Blue Matter application around calls to the neighborhood collectives and represent averages over all the broadcasts and reductions in the entire n-body system and over a number of time steps.

reduce. Overlapping channel collectives may be driven concurrently from multiple threads of execution on a single chip; currently this is achieved by partitioning hardware resources. In V5, each force generating module receives positions and sends forces on channel collectives with behavior appropriate to that module’s requirements. Currently, the scheduling of force generating modules to processors is static and V5 schedules integration and P3ME on one processor and RSNB and bonded on the other processor which works well for the molecular system sizes and parallel partitions sizes we are targeting.

Two of the performance critical communication collectives required are:

1. Distributed transpose required by the 3D-FFT
2. Finite-ranged neighborhood broadcast and reduction required for the real space non-bond operations

The 3D-FFT algorithm used for the P3ME method employed by Blue Matter has been described elsewhere along with a performance comparison of implementations on MPI and the BG/L ADE SPI [Eleftheriou et al. 2005]. The performance of the distributed 3D-FFT implementation on the BG/L ADE SPI is significantly better than that of the MPI version at larger node counts, but both the MPI and SPI implementations continue to speed up through the limits of scalability of the FFT.

In contrast, the scaling behaviors of the MPI and the “neighborhood” broadcast and reduction are qualitatively different as shown in Figure 4. The detailed performance comparison of the MPI and BG/L ADE SPI implementations of the entire Blue Matter molecular dynamics application using the V4 al-

gorithm was reported previously[Fitch et al. 2006]. The Blue Matter V4 results have also been reproduced in Figure 5.

We have hand generated dedicated state machines for 3D FFT, neighborhood communications, all-to-all, and hardware reduce terminated one-sided communications and in each case exceeded the performance of available standard communication libraries.

## 4 Benchmarking Results

The Blue Matter benchmarking data for V5 presented in Table 4 and included in Figure 5 was obtained by averaging 180 time steps taken at the end of a 1000 time step run. Details about the parameters used in these benchmarking runs can be found in Table 4 and reflect either actual production parameters or equivalent (where possible) parameters to those used in other benchmarks. These parameters give excellent energy conservation in NVE runs with Blue Matter, e.g. Rhodopsin energy drift is about  $6 \times 10^{-4}$  K/ns or less than 1 K equivalent temperature rise over a 1  $\mu$ second simulation at a kinetic energy equivalent to 310 K. The  $\beta$ -Hairpin runs and one V5 SOPE series used a  $64 \times 64 \times 64$  FFT mesh while all the other Blue Matter runs used a  $128 \times 128 \times 128$  FFT mesh. FFTs were computed using single precision while all other operations were done in double precision. All runs were made using a velocity Verlet integrator[Swope et al. 1982] and performed the P3ME calculations on every time-step. All but one of the Blue Matter runs were made out to the largest node counts supported by our current FFT implementation, 4096 for the systems using a  $64^3$  mesh and 16,384 nodes for the systems using a  $128^3$  mesh. A single run, on 8192 nodes, of the V5 SOPE system with a  $64^3$  3D-FFT was actually run in a mode with the FFT at its scaling limit of 4096.

Figure 5 plots the computational throughput in time steps per second versus the number of atoms per node. Plotting the data as a function of atoms per node provides some degree of normalization for system size and one can observe that the scalability plots at larger values of atoms/node (lower node counts) seem to lie on a “universal” curve. This is true when the real-space non-bonded interactions are the dominant contribution to the iteration time. For comparison with the V5 results, Figure 5 includes the Blue Matter V4 data on both SPI and MPI[Fitch et al. 2006], published results on the ApoA1 system for ports of NAMD to Blue Gene/L using both MPI and a lower level messaging layer[Kumar et al. 2006], and finally benchmarking data on the same system from 2002 for NAMD on the PSC Lemieux system[Phillips et al. 2002].

Figure 6 shows scaling of selected components of the total time-step for V5 Rhodopsin with  $128^3$  3D-FFT. The transition from real-space computation-bound behavior at low

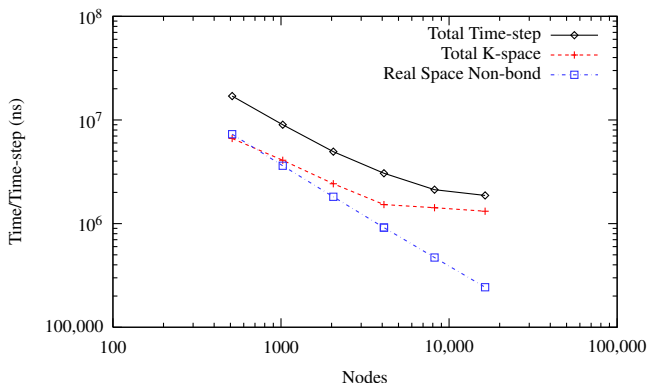


Figure 6: Scaling of Time-step Components for V5 Rhodopsin

node counts to K-space dominated behavior at high node counts is clearly visible. The K-space component itself is dominated by the execution time of the forward and reverse 3D-FFTs used to compute the convolution.

The V5 results show clear improvements over the V4 SPI benchmarks and in some cases actually approach time/time-step values limited by the k-space (FFT) module. The  $\beta$ -hairpin V5 results showed a 50% improvement over the V4 benchmarks and achieved an iteration time of 0.83 milliseconds or only 581,000 cycles on Blue Gene/L. The SOPE system also showed significant improvement, which was sufficient to expose the effects of overheads and caused speedup to end at 8192 nodes.

## 5 Summary and Conclusions

In this paper we have described a novel N-Body spatial-force decomposition which has an upper bound on communication scaling of  $O(p^{2/3})$  and for which we observe communication scaling consistent with  $O(p^{1/2})$ . This work has resulted in the demonstration of an unprecedented computation rate of over 1200 time steps per second for a small  $\beta$ -Hairpin system on 4096 nodes and continued speed-up for larger systems through 16,384 nodes. Given sufficient nodes to reduce the real-space non-bond cost, we have observed the 3D-FFT in the P3ME module becoming the rate-limiting step at the limits of concurrency for the 3D-FFT. This suggests that further performance improvements would require explorations of alternative methods for treating the long range electrostatic interactions.

The time-to-solution achieved contributes to significantly increasing the potential overlap between simulation and experiment and has already had significant impact in the area of biomolecular simulation of membrane-bound protein systems[Pitman et al. 2005a; Grossfield et al. 2006] and studies of protein folding[Eleftheriou et al. 2006a].

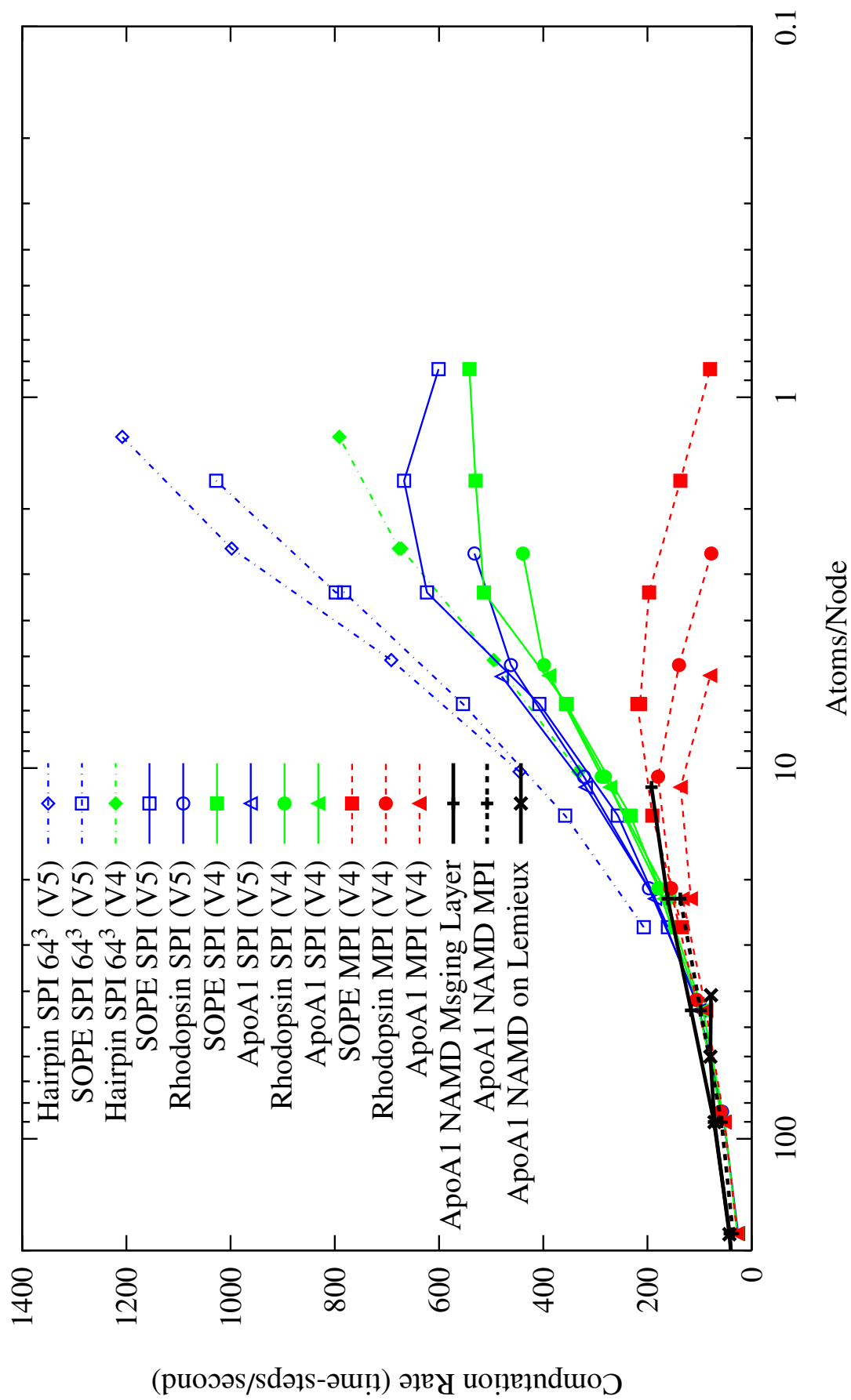


Figure 5: This plot shows computational rates on a number of molecular systems as a function of the number of atoms per node. This facilitates comparisons between systems of different sizes and also explicitly shows the degree of strong scaling achieved. Results for Blue Matter using the V5 method running on the low level communications System Programming Interface (SPI) provided by the Blue Gene/L Advanced Diagnostic Environment [Giampapa et al. 2005] are plotted along with results of a prior decomposition (V4) implemented on both MPI and SPI and data for NAMD on Blue Gene/L [Kumar et al. 2006] and on the PSC Lemieux system [Phillips et al. 2002].

System	Total Atoms	Cutoff/Switch (Å)	P3ME Mesh	Time Step (fs)
Hairpin	5239	9.0/1.0	64 <sup>3</sup>	1
SOPE	13,758	9.0/1.0	64 <sup>3</sup> , 128 <sup>3</sup>	1
Rhodopsin	43,222	9.0/1.0	128 <sup>3</sup>	2
ApoA1	92,224	10.0/2.0	128 <sup>3</sup>	1

Table 1: Details about the systems benchmarked with Blue Matter. All runs were made with the velocity Verlet integrator [Swope et al. 1982], used the Particle-Particle-Particle Mesh (P3ME) technique to handle long range electrostatic interactions, and were constant particle number, volume, and energy (NVE) simulations. All runs performed the P3ME calculation on every time-step. Except for the SOPE (64<sup>3</sup>) data, these choices are those used in production scientific work (Hairpin, Rhodopsin) or attempt to match benchmarking conditions reported elsewhere (ApoA1). For Rhodopsin, the measured total energy drift is approximately  $6 \times 10^{-4}$  K/ns, where the energy change has been expressed as an equivalent temperature rise for that system.

Total	$P_x$	$P_y$	$P_z$	Time/time-step (milliseconds)				
				$\beta$ -H	SOPE 64 <sup>3</sup>	SOPE 128 <sup>3</sup>	Rhod.	ApoA
512	8	8	8	2.25	4.83	6.22	17.52	38.42
1024	8	8	16	1.45	2.80	3.89	9.48	18.95
2048	8	16	16	1.00	1.80	2.45	5.07	9.97
4096	8	32	16	1.12	1.29	2.11	3.21	5.44
4096	16	16	16	0.83	1.26	1.60	3.11	5.39
8192	16	32	16		0.97	1.50	2.16	3.14
16384	32	32	16			1.66	1.88	2.09

Table 2: Performance in time/time step as a function of node count (and partition geometry) for  $\beta$ -Hairpin (5239 atoms), SOPE (13,758 atoms) with FFT sizes of 64<sup>3</sup> and 128<sup>3</sup>, Rhodopsin (43,222 atoms), and ApoA (92,224).

This work suggests future research in two areas. First, although Blue Matter V5 exploits the BG/L hardware to achieve speedups through the practical limits of scalability of the 3D-FFT, it would be possible to improve efficiency at moderate node counts by optimizing all the communications and computations using techniques similar to ones described here for the real-space non-bond. We also anticipate formalizing our CSP-like programming model using dedicated channel protocols and light-weight processes compiled to hardware interfaces in our next version of the code (V6).

**Acknowledgements:** We would like to thank the members of the Blue Gene hardware team including P. Heidelberger, A. Gara, M. Blumrich, J. Sexton as well as others who have made use of and contributions to the Blue Matter code over the years, particularly Y. Zhestkov, Y. Sham, F. Suits, W. Swope, J. Pitera, A. Grossfield, and R. Zhou.

## References

- ALLEN, F., ET AL. 2001. Blue Gene: a vision for protein science using a petaflop supercomputer. *IBM Systems Journal* 40, 2, 310–327.
- DESERNO, M., AND HOLM, C. 1998. How to mesh up Ewald sums. i. a theoretical and numerical comparison of various particle mesh routines. *J. Chem. Phys.* 109, 18, 7678–7693.
- ELEFThERIOU, M., FITCH, B., RAYSHUBSKIY, A., WARD, T., AND GERMAIN, R. 2005. Performance measurements of the 3d FFT on the Blue Gene/L supercomputer. In *Euro-Par 2005 Parallel Processing: 11th International Euro-Par Conference, Lisbon, Portugal, August 30-September 2, 2005*, Springer-Verlag, J. Cunha and P. Medeiros, Eds., vol. 3648 of *Lecture Notes in Computer Science*, 795–803.
- ELEFThERIOU, M., GERMAIN, R., ROYYURU, A., AND ZHOU, R. 2006. Thermal denaturing of mutant lysozyme with both oplaa and charmm force fields. submitted to *J. Am. Chem. Soc.*
- ELEFThERIOU, M., RAYSHUBSKIY, A., PITERA, J. W., FITCH, B. G., ZHOU, R., AND GERMAIN, R. S. 2006. Parallel implementation of the replica exchange molecular dynamics algorithm on Blue Gene/L. In *Fifth IEEE International Workshop on High Performance Computational Biology*.
- FITCH, B., GERMAIN, R., MENDELL, M., PITERA, J., PITMAN, M., RAYSHUBSKIY, A., SHAM, Y., SUITS, F., SWOPE, W., WARD, T., ZHESTKOV, Y., AND ZHOU, R. 2003. Blue Matter, an application framework for molec-

- ular simulation on Blue Gene. *Journal of Parallel and Distributed Computing* 63, 759–773.
- FITCH, B. G., RAYSHUBSKIY, A., ELEFThERIOU, M., WARD, T. C., GIAMPAPA, M., ZHESTKOV, Y., PITMAN, M. C., SUITS, F., GROSSFIELD, A., PITERA, J., SWOPE, W., ZHOU, R., GERMAIN, R. S., AND FELLER, S. 2005. Blue matter: Strong scaling of molecular dynamics on Blue Gene/L. Research Report RC23688, IBM Research Division, August.
- FITCH, B. G., RAYSHUBSKIY, A., ELEFThERIOU, M., WARD, T. C., GIAMPAPA, M., ZHESTKOV, Y., PITMAN, M. C., SUITS, F., GROSSFIELD, A., PITERA, J., SWOPE, W., ZHOU, R., FELLER, S., AND GERMAIN, R. S. 2006. Blue Matter: Strong scaling of molecular dynamics on Blue Gene/L. In *International Conference on Computational Science (ICCS 2006)*, Springer-Verlag, V. Alexandrov, D. van Albada, P. Sloot, and J. Dongarra, Eds., vol. 3992 of *LNCS*, 846–854.
- FRENKEL, D., AND SMIT, B. 1996. *Understanding Molecular Simulation*. Academic Press, San Diego, CA.
- GARA, A., ET AL. 2005. Overview of the Blue Gene/L system architecture. *IBM Journal of Research and Development* 49, 2/3, 195–212.
- GERMAIN, R. S., FITCH, B., RAYSHUBSKIY, A., ELEFThERIOU, M., PITMAN, M. C., SUITS, F., GIAMPAPA, M., AND T.J. CHRISTOPHER WARD, T. C. 2005. 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*, ACM Press, New York, NY, USA, 207–212.
- GERMAIN, R., ZHESTKOV, Y., ELEFThERIOU, M., RAYSHUBSKIY, A., SUITS, F., WARD, T., AND FITCH, B. 2005. Early performance data on the Blue Matter molecular simulation framework. *IBM Journal of Research and Development* 49, 2/3, 447–456.
- GIAMPAPA, M., BELLOFATTO, R., BLUMRICH, M. A., CHEN, D., DOMBROWA, M. B., GARA, A., HARING, R. A., HEIDELBERGER, P., HOENICKE, D., KOPCSAY, G. V., NATHANSON, B. J., STEINMACHER-BUROW, B. D., OHMACHT, M., SALAPURA, V., AND VRANAS, P. 2005. Blue Gene/L advanced diagnostics environment. *IBM Journal of Research and Development* 49, 2/3, 319–332.
- GROSSFIELD, A., FELLER, S. E., AND PITMAN, M. C. 2006. A role for direct interactions in the modulation of rhodopsin by omega-3 polyunsaturated lipids. *PNAS* 103, 13, 4888–4893.
- HENDRICKSON, B., AND PLIMPTON, S. 1995. Parallel many body simulations without all to all communication. *Journal of Parallel and Distributed Computing* 27, 1, 15–25.
- HOARE, C. 1985. *Communicating Sequential Processes*. Prentice-Hall.
- KUMAR, S., HUANG, C., ALMASI, G., AND KALE, L. V. 2006. Achieving strong scaling with NAMD on Blue Gene/L. 20th IEEE International Parallel and Distributed Processing Symposium, IEEE. <http://charm.cs.uiuc.edu/papers/NAMDIDPDS06.pdf>.
- PHILLIPS, JAMES C. AND ZHENG, G., KUMAR, S., AND KALE, L. V. 2002. NAMD: biomolecular simulation on thousands of processors. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, IEEE Computer Society Press, Los Alamitos, CA, USA, 1–18.
- PITMAN, M. C., SUITS, F., MACKERELL, ALEXANDER D., J., AND FELLER, S. E. 2004. Molecular-level organization of saturated and polyunsaturated fatty acids in a phosphatidylcholine bilayer containing cholesterol. *Biochemistry* 43, 49, 15318–15328.
- PITMAN, M. C., GROSSFIELD, A., SUITS, F., AND FELLER, S. E. 2005. Role of cholesterol and polyunsaturated chains in lipid-protein interactions: Molecular dynamics simulation of rhodopsin in a realistic membrane environment. *Journal of the American Chemical Society* 127, 13, 4576–4577.
- PITMAN, M. C., SUITS, F., GAWRISCH, K., AND FELLER, S. E. 2005. Molecular dynamics investigation of dynamical properties of phosphatidylethanolamine lipid bilayers. *Journal of Chemical Physics* 122, 24, 244715.
- PLIMPTON, S., AND HENDRICKSON, B. 1996. A new parallel method for molecular dynamics simulation of macromolecular systems. *Journal of Computational Chemistry* 17, 3, 326–337.
- SHAW, D. 2004. An asymptotic improvement in the parallel evaluation of pairwise particle interactions. Presented at Philadelphia American Chemical Society meeting, September.
- SHAW, D. E. 2005. A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions. *Journal of Computational Chemistry* 26, 13, 1318–1328.
- SNIR, M. 2004. A note on n-body computations with cut-offs. *Theory of Computing Systems* 37, 295–318. DOI: 10.1007/s00224-003-1071-0.
- SUGITA, Y., AND OKAMOTO, Y. 1999. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* 314, 141–151.

- SUITS, F., PITMAN, M. C., AND FELLER, S. E. 2005. Molecular dynamics investigation of the structural properties of phosphatidylethanolamine lipid bilayers. *Journal of Chemical Physics* 122, 24.
- SWOPE, W., ANDERSEN, H., BERENS, P., AND WILSON, K. 1982. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *Journal of Chemical Physics* 76, 637–649.
- SWOPE, W. C., PITERA, J. W., SUITS, F., PITMAN, M., ELEFTHERIOU, M., FITCH, B. G., GERMAIN, R. S., RAYSHUBSKIY, A., WARD, T. J. C., ZHESTKOV, Y., AND ZHOU, R. 2004. Describing protein folding kinetics by molecular dynamics simulations. 2. example applications to alanine dipeptide and a  $\beta$ -hairpin peptide. *J. Phys. Chem. B* 108, 21, 6582–6594.
- TAYLOR, V., STEVENS, R., AND ARNOLD, K. 1997. Parallel molecular dynamics: implications for massively parallel machines. *Journal of Parallel and Distributed Computing* 45, 2 (September), 166–175.