Blue Matter on Blue Gene/L: Massively Parallel Computation for Biomolecular Simulation

Robert S. Germain
rgermain@us.ibm.com

Blake Fitch
bgf@us.ibm.com

Aleksandr Rayshubskiy
arayshu@us.ibm.com

Maria Eleftheriou
mariae@us.ibm.com

Michael C. Pitman
pitman@us.ibm.com

Frank Suits
suits@us.ibm.com

Mark Giampapa
giampapa@us.ibm.com

IBM Thomas J. Watson Research Center
1101 Kitchawan Road/Route 134
Yorktown Heights, NY 10598 1pc

T.J. Christopher Ward
IBM Hursley Park
Hursley, UK SO212JN
tjcw@uk.ibm.com

ABSTRACT
This paper provides an overview of the Blue Matter application development effort within the Blue Gene project that supports our scientific simulation efforts in the areas of protein folding and membrane-protein systems. The design philosophy of the Blue Gene/L architecture relies on large numbers of power efficient nodes (whose technology is derived from the world of embedded microprocessors) to enable packing of many such nodes into a small volume to achieve high performance. In order for an application to exploit the potential of this architecture, the application must scale well to large node counts. Because the scientific goals of the project entail simulating very long time-scales, up to microseconds, strong scaling of a fixed size problem to these large node counts is a requirement. In pursuit of this objective we have considered a variety of parallel decompositions and explored ways to exploit and map algorithms onto the two primary high performance interconnects provided by the Blue Gene architecture, the 3D-torus network and the collective network. Our current version of the application continues to speed up through 4096 nodes and is being used for studies of a protein/lipid system (for which some results have already been published) and for protein folding/unfolding simulations.

Categories and Subject Descriptors
J.3 [Computer Applications]: Life and Medical Sciences—Biology and Genetics; D.1.3 [Programming Techniques]: Concurrent Programming—Parallel Programming

1. INTRODUCTION
From its inception at the end of 1999, the Blue Gene project has had an application effort to support its scientific goal of using large scale simulation to improve our understanding of protein folding mechanisms and other biologically important phenomena. That application effort has always looked for ways to exploit features of the machine architecture as part of the effort to improve performance and scalability.

The scientific goals of the Blue Gene project require biomolecular simulations of modestly sized systems (10,000–100,000 atoms) for long time scales (hundreds of nanoseconds to microseconds) and because the philosophy of the Blue Gene/L hardware design has been to use massive numbers of power efficient CPUs to achieve high performance, the Blue Matter application effort is required to be able to demonstrate strong scaling of fixed size problems to large node counts. Results obtained from the exploitation of prototype Blue Gene/L hardware for production scientific use in the second half of 2004 were recently published and additional work is being initiated as more hardware becomes available.

2. BLUE GENE/L SYSTEM OVERVIEW
Blue Gene/L is a massively parallel supercomputer developed at the IBM T.J. Watson Research Center in collaboration with Lawrence Livermore National Laboratory. In contrast to other current massively parallel supercomputers, the building block of BG/L is not derived from the fastest (and highest power density) technology available, but rather from the world of embedded microprocessors. Except for off-chip double data rate (DDR) memory and link chips that enable the partitioning of the machine, all of the functionality for a BG/L node is contained on a single ASIC chip shown
schematically in Figure 1. This chip operates at a clock frequency of
700 MHz and has two PowerPC 440 CPUs with associated dual floating point units. An on-chip L3 cache comprising 4MB of DRAM is typically only 30 cycles away from registers on L1/L2 cache misses. All of the circuitry for the communications networks is also on this chip and the companion link chip. The core BG/L machine is comprised of two classes of nodes, compute nodes and I/O nodes. The I/O nodes are not counted when sizing the compute capacity of BG/L; they only act as the gateway for communication to the external world and the ratio of I/O nodes to compute nodes can be varied.

BG/L has five networks, two of which are of particular interest to the application developers: the torus network and the collective network. The three-dimensional torus has links between each node and its six neighbors while the collective network enables low latency broadcast and reduction operations as well as providing the path for I/O to external devices. Applications communicate with the outside world via I/O nodes; these are additional BG/L nodes that are connected to the collective network (but not the torus network) and also have a Gigabit Ethernet connection to the outside world.

Two modes of operation are supported by the system software: (1) “coprocessor mode” which runs a single MPI task on each compute node, and “virtual node mode” which runs two MPI-tasks on each compute node. The compute nodes run a specialized Compute Node Kernel (CNK) that supports a subset of posix calls and only includes ECC. The I/O proc is electrically neutral) with periodic boundary conditions can be included in the context of biomolecular simulation.

The classes of interactions typically included in a force field are enumerated in Table 3. These include forces between covalently bonded particles such as bond stretches between pairs of atoms, angle bends defined by a triple of particles, and torsions defined by a quartet of particles. In addition, non-bonded forces such as electrostatics and van der Waals (typically modeled, along with hard core repulsion, by a Lennard-Jones 6-12 potential) are also included. Current practice in biomolecular simulation involves the use of periodic boundary conditions. The bonded and Lennard-Jones interactions are weak enough at large distances that they are just set to zero beyond some cut-off distance, often using a smooth cut-off function and the periodic boundary conditions can be treated using a minimum image convention in which for each particle in the central cell, the force will be computed with the nearest image (either in the central cell or in one of the 26 adjoining cells) of every other particle within the cut-off distance. However, the same finite range cut-off treatment applied to the long-range electrostatic forces can lead to unphysical behavior and requires the use of a different technique.

The potential energy of a system of point charges (where the system is electrically neutral) with periodic boundary conditions can be written as

$$\sum_{i} \sum_{j \neq i}^{N} \frac{q_{i} q_{j}}{|\mathbf{r}_{ij} + \mathbf{n}|}$$

where $q_{i}$ is the charge on the $i$th particle and $\mathbf{n}$ belongs to the set of lattice vectors $\{l \mathbf{u} + m \mathbf{v} + n \mathbf{w}\}$ where $l,m,$ and $n$ are integers and $\mathbf{u}, \mathbf{v},$ and $\mathbf{w}$ are the lattice basis vectors. The $\mathbf{r}$ on the first sum indicates that when $\mathbf{n} = 0$, the terms in the double summation with $i = j$ should be excluded. This sum is only conditionally convergent; that is, the value of the sum depends on the summation order. The Ewald summation technique and related techniques rewrite the potential sum as a pair of summations; one in real space consisting of a summation of “screened” electrostatic potential terms that is unconditionally convergent and the second, a sum that is expressed as a sum of Fourier transformed terms (this is often called the “reciprocal space” sum). The most commonly practiced techniques for handling electrostatics with periodic boundary conditions, particle-mesh methods such as Particle-Particle-Particle Mesh Ewald (P3ME) technique, approximate the actual charge distribution by a set of weights on a uniformly spaced mesh. This allows use of the Fast Fourier Transform (FFT) in carrying out the necessary computation which is a convolution of the charge distribution with a kernel chosen to give good accuracy with the uniform mesh approximation used for the charge distribution.

4. COMPUTATIONAL CHALLENGES

In order to support the goal of using simulation to access longer time scales routinely, the Blue Matter application has to provide good scalability to thousands of nodes for fixed size problems. Key challenges exist in the areas of load balancing and global data dependencies. Load balancing challenges arise in the context of computing finite-ranged non-bond real-space interactions. Global data


<table>
<thead>
<tr>
<th>Force Class</th>
<th>Description and Examples</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonded</td>
<td>Forces between covalently bonded molecules—includes bond stretches, angle bends, and torsions.</td>
<td>Limited range within a covalently bonded molecule (typically within a graph distance of four)</td>
</tr>
<tr>
<td>Real Space Non-bond</td>
<td>Lennard-Jones: ( U_{L−J}(r) = 4\epsilon \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} ), cut-off electrostatics, real space interaction portion of Ewald and related techniques</td>
<td>Limited range in simulation space</td>
</tr>
<tr>
<td>Reciprocal Space</td>
<td>Fourier space portion of long range electrostatic forces treated by the Ewald summation technique or related Particle-Particle-Particle-Mesh (P3ME) technique</td>
<td>global communication required for computation of 3D-FFTs</td>
</tr>
</tbody>
</table>

Table 1: This table summarizes the different classes of forces that must be computed in molecular dynamics, provides some examples in each class, and specifies the communication or data dependency characteristics of each class. The bonded and real-space non-bond forces have data dependencies that are local in simulation space and hence can take advantage of an appropriate domain decomposition to achieve locality of communication. The computation of the reciprocal or Fourier space contribution to the force on a particle requires knowledge about the position of every particle in the system. This global data dependency comes about through the computation of the convolution of the charge distribution with the kernel required to solve for the effects of the long range electrostatic forces in a system with periodic boundary conditions. In the case of the Particle-Particle-Particle-Mesh (P3ME) technique used in this work, the convolution is computed by first approximating the actual charge distribution by weights on a regular mesh, then computing the convolution through use of a forward and inverse FFT.

5. CURRENT WORK AND RESULTS

Our current exploration involves a variant of a “volume” decomposition in which the three dimensional simulation domain is divided into volume elements that are mapped directly onto BG/L nodes so that locality in simulation space is equivalent to locality on BG/L’s 3D-torus network. In this case we achieve load balancing for the real-space interactions by first assigning the interaction between two atoms (or groups of atoms) to be carried out on the node containing the point (in simulation space) that lies midway between the two interacting atoms (or groups) as shown in Figure 2 and associating a cost with that position. Then we carry out successive orthogonal bisections of simulation space in such a way as to give equal loads in each section[5]. This process of orthogonal recursive bisection (ORB) is carried out until there are enough sections so that we can associate each volume section with a BG/L node and locality is preserved. The position broadcast and force reduction can now be local operations on the torus network since positions only need to be broadcast to nodes containing portions of simulation space within half the cutoff distance of the originating atom or atoms. Of course the quality of the load balance will decrease with time, but measurements indicate that the degradation is slow enough to allow the simulation to proceed for many time steps before a regeneration of the mapping is necessary.

Although we are running in co-processor mode, we can make use of both cores to allow a limited amount of overlap between communication and computation. Because the real-space non-bond computations and the reciprocal space operations can proceed independently, we overlap them by off-loading the real-space non-bond operations (which are pure computation) to the second CPU while carrying out the communications-intensive reciprocal-space operations on the first CPU. This “dual core” mode of operation gives significant improvements when the real-space and reciprocal-space operations take comparable amounts of time.

Figure[5] shows the scalability of the major components of a time-step for a system with 43,222 atoms comprising a protein, Rhodopsin,
Interaction Centers

Node A

Node B

Spatial Decomposition

Figure 2: Figure (a) illustrates the relationship of the particle positions (large dots) and the interaction centers (small dots) in simulation space. The interaction centers are placed at the mid-point between each pair of particles that fall within the cut-off radius. A dashed circle with radius equal to the cut-off radius chosen is drawn around one of the particles. Static or “structural” load balancing is carried out by using optimal recursive bisection to partition the simulation volume into sub-volumes that contain approximately equal computational burdens. The computational burden of a sub-volume is computed by summing the computational burden of each interaction center contained within that sub-volume. Figure (b) gives a view of the spatial decomposition showing the broadcast zones for two nodes superimposed on the spatial decomposition of the domain onto all nodes (two-dimensional view for simplicity). The nodes that contain areas of simulation space within $R_b$ of the volume element assigned to Node A are shown with one kind of hatching while those nodes that contain areas of simulation space within $R_b$ of the volume element assigned to Node B are shown with another type of hatching. The region of overlap between these two areas is shown with cross-hatching. The broadcast radius $R_b > R_c/2$ where $R_c$ is the cutoff radius. The interaction between a particle stored on Node A and a particle stored on Node B can be computed on any of the nodes in the overlap (cross-hatched) region.

Figure 3: Scalability of major components of a time-step for the Rhodopsin system. The data were taken in dual core mode except for the “Real Space” data which had to be taken in single core mode because the trace facility only works on “Core 0” and in dual core mode the real space non-bond interactions are handled on “Core 1”.

Figure 3 shows that the time required to carry out the reciprocal-space operations begins to become the limiting factor for an iteration at the highest node counts.

The current MPI-based implementation of the 3D-FFT takes advantage of a highly optimized $\text{MPI\_ALLTOALLV}$ implementation developed for BG/L [4]. This MPI collective, which is functionally equivalent to a collection of point-to-point messages between nodes in a mesh-connected system, is used for all 3D-FFT computations. By using this collective, the reciprocal-space operations can be performed efficiently across the entire system, allowing for scalability to thousands of nodes.
all of the nodes, is used in the implementation of the distributed data transpose required by the 3D-FFT. For the transpose, in most regimes, the execution time is limited by the bisectional bandwidth of the machine, therefore a fixed size FFT will speed up as p increases, the execution time is limited by the bisectional bandwidth of the machine.

Figure 4 compares the execution time obtained for 64 and 128 nodes using MPI and using low level (hardware packet) interfaces derived from communications routines developed for hardware performance testing and diagnostics[11]. The data show that at very high node counts, the implementation using low level interfaces currently outperforms the MPI-based implementation.

6. SUMMARY
We have provided an overview of the Blue Matter molecular simulation application that was developed in support of and in conjunction with the Blue Gene science program. As part of that effort we have developed a new variant of spatial decomposition for n-body simulations that permits efficient load balancing of real space interactions and which is compatible with a three-dimensional mesh/torus connected machine. To address scalability challenges encountered in correctly treating long range electrostatic interactions with periodic boundary conditions, we have demonstrated a distributed 3D-FFT that, for the sizes required for our application, continues to scale well to many thousands of nodes. These developments have allowed us to achieve sub-ten millisecond time steps on a 43K atom membrane protein system using a 4096 node Blue Gene/L partition which enables running microsecond simulations in less than two months. Scientific results have already appeared resulting from this work and further scientific studies are underway on additional systems of fundamental biological interest.

7. ACKNOWLEDGEMENTS
The authors would like to acknowledge the contributions of others who have worked on the Blue Matter application and those who have used it for production scientific work, including Yuriy Zhetskov, Alan Grossfield, Jed Pitera, Bill Swope, Ruhong Zhou, Yuk Sham, and Alex Balaeff. All of this work was made enabled by the efforts of the Blue Gene/L hardware and system software teams.

8. REFERENCES


