

MOLECULAR DYNAMICS COMES OF AGE: 320 BILLION ATOM SIMULATION ON BlueGene/L

KAI KADAU

*Theoretical Division, Los Alamos National Laboratory
MS G756, Los Alamos, New Mexico 87545, USA
kkadau@lanl.gov*

TIMOTHY C. GERMANN

*Applied Physics Division, Los Alamos National Laboratory
MS F663, Los Alamos, New Mexico 87545, USA
tcg@lanl.gov*

PETER S. LOMDAHL

*Theoretical Division, Los Alamos National Laboratory
MS B214, Los Alamos, New Mexico 87545, USA
pxl@lanl.gov*

Received 1 November 2006

Revised 1 December 2006

As computational power is increasing, molecular dynamics simulations are becoming more important in materials science, chemistry, physics, and other fields of science. We demonstrate weak and strong scaling of our classical molecular dynamics code *SPaSM* on Livermore's BlueGene/L architecture containing 131 072 IBM PowerPC440 processors. A maximum of 320 billion atoms have been simulated in double precision, corresponding to a cubic piece of solid copper with an edge length of 1.56 μm .

Keywords: Molecular dynamics; Blue Gene Light; high performance computing (HPC); *SPaSM*; large scale.

PACS Nos.: 07.05.Tp, 83.10.Rp, 71.15.Pd.

1. Introduction

Since the invention of molecular dynamics (MD) simulations in 1957 by Berni Alder and Tom Wainwright,¹ the peak computational power has increased from about 1000 floating point operations per second (Flops) of early vacuum tube machines, like the UNIVAC at Livermore or the MANIAC at Los Alamos, to 360 TFlops on architectures like the IBM BlueGene/L (BGL) at Livermore.^{2,3} With this increase of computational power, the early 100 particle hard sphere system has grown into more sophisticated particle interactions — like smooth pair and many body

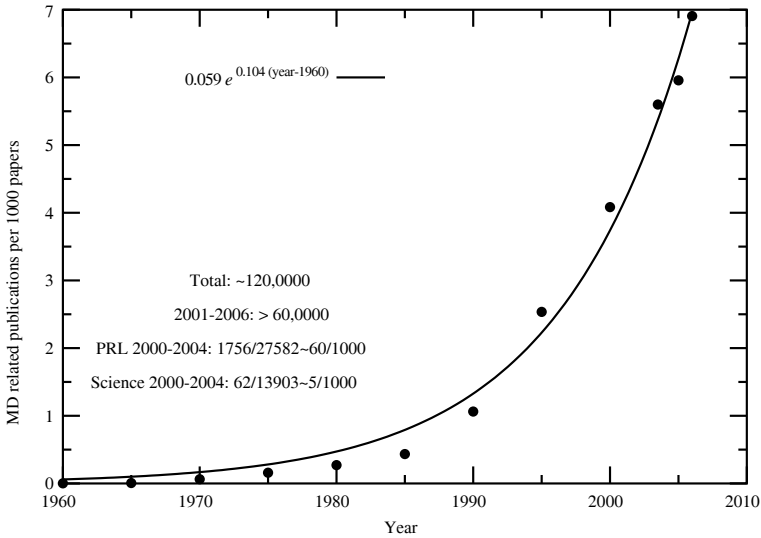


Fig. 1. Evolution of the literature fraction of MD related publications (i.e., molecular dynamics is a keyword or in the abstract). The literature search was performed using the following data bases: BIOSIS, Engineering Index, Inspec, ISI Proceedings, ISI SciSearch, ISI Social SciSearch. An interdisciplinary journal such as *Science* follows this general trend, while specific physics journals such as *Physical Review Letters* have a significant larger fraction of MD related publications. An exponential fit represents the general trend of the data.

potentials, as well as *ab initio* methods calculating the interatomic forces based on quantum mechanical principles — and the accessible system size has increased to many billions of particles. The importance of MD in science has evolved with this increase in capabilities, as can be illustrated by the dramatic growth in the MD-related fraction of the scientific literature (see Fig. 1). Today, 7 out of 1000 published scientific articles are related to MD — if looking at a specific physics journal such as *Physical Review Letters* this number is about 10 times larger — and still growing exponentially.

Here, we demonstrate the scalability of the *SPaSM* (Scalable Parallel Short-range Molecular dynamics) algorithm for up to 320 billion atoms, interacting via a Lennard-Jones pair potential, on Livermore’s BGL system containing 131 072 IBM PowerPC 440 processors with a clock speed of 700 MHz.^{4–6} Together with the analysis and visualization methods described earlier,⁷ this enables scientific simulations in 3D on the micron length scale, with full atomistic resolution.

2. The *SPaSM* Algorithm

We will consider a box of particles interacting via a short-range potential, i.e., there is a cut-off radius r_{cut} after which the interaction is neglected. There are many ways to map this physical geometry and its spatial distribution of particles to the memory of the computer.^{8,9} An efficient way is to organize the geometry into small

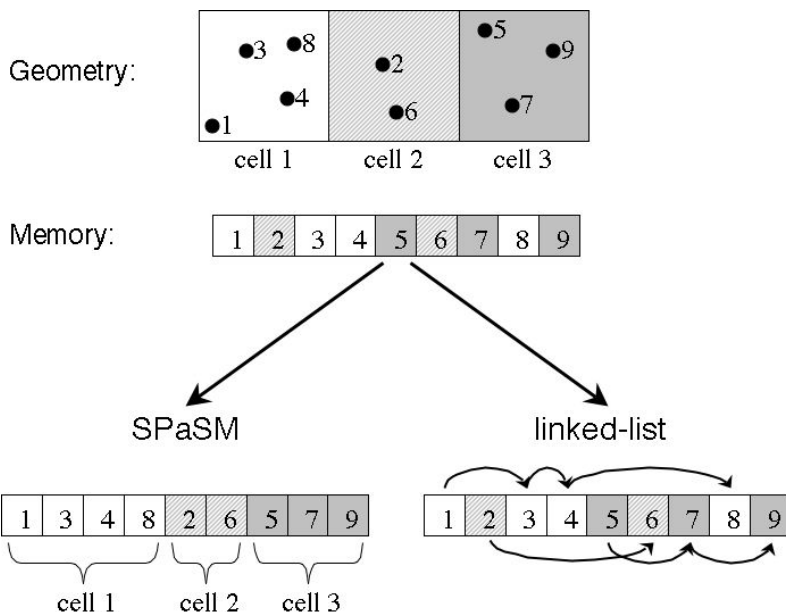


Fig. 2. Relation between the physical geometry of the simulation cell and the memory organization in *SPaSM* and a linked-list algorithm. *SPaSM* sorts the memory such that no pointers or lists are needed to connect particles that are in the same cell, only the number of particles per cell is needed.

cells that have an edge length at least as long as r_{cut} . Interactions then only need to be calculated for particles that are in the same or adjacent cells. A traditional way of organizing the memory is to link the atoms that belong to a certain cell, and each time particles change cells to reorganize those links. This so-called *linked-list* method is in contrast to the memory management of *SPaSM*, where particles that belong to the same cell are sequentially stored in memory (see Fig. 2). This way, interactions between particles can be calculated by just sequentially moving through the particle memory without having to hop through disjointed regions of memory. Also, no additional memory is needed to link the particles. In a typical MD simulation not many particles change their cell during one iteration, and therefore, the amount of time spent for sorting the particles into cells is small. However, if particles change cells often — which might be the case for certain applications — a *linked-list* method might be the better choice.

In order to use a distributed memory multi-processor system, particle information has to be sent to and received from processor to processor. A spatial decomposition is used to assign particles to processors.⁷ The Message Passing Interface (MPI) and local send- and receive buffers are handling the particle information exchange between processors.¹⁰ Therefore, the iteration time t on a distributed memory system can be expressed as

$$t = a \frac{N}{P} + b \left(\frac{N}{P} \right)^{2/3}, \tag{1}$$

where N is the number of particles (in millions) and P is the number of processors. The first term reflects that the computational effort increases linearly with the number of atoms per processor. The second term takes into account the message passing cost, which increases (under the assumption of homogeneity) as the surface area of the geometrical domain assigned to each processor.

Using the *SPaSM* algorithm, various physical challenges in solid state physics,^{11–13} fluid dynamics,¹⁴ and even epidemiology¹⁵ have been successfully tackled by large-scale particle-based simulations.

3. Timings and Performance on BlueGene/L

Multiple short performance runs were carried out on the BGL system using 4096–131 072 CPUs and 1–320 billion particles arranged in a face-centered cubic lattice and interacting via a Lennard-Jones potential with $r_{\text{cut}} = 2.5\sigma$ or $r_{\text{cut}} = 5.0\sigma$.³ The runs presented here used the so-called *virtual node mode*, which means that all available CPUs are used for calculations.^a

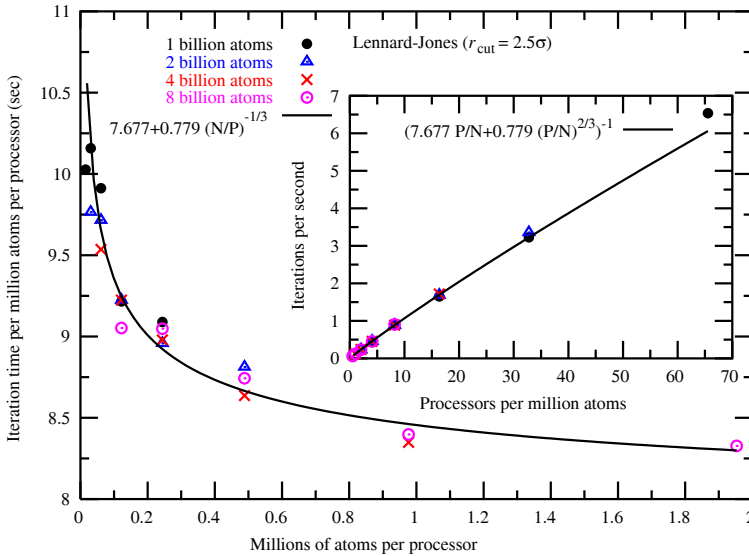


Fig. 3. Strong scaling for 1–8 billion atoms distributed on 4096–65 536 processors of the IBM BGL machine at Livermore. For a small number of atoms per processor the communication/calculation ratio gets larger and the iteration time per atom per processor increases as $(N/P)^{-1/3}$. The inset shows the speed-up as the number of processors per million atom increases.

^aOne node of BGL consists of two CPUs, one of which is normally set aside to handle communication only. In *virtual node mode* both of the CPUs are used for calculations.

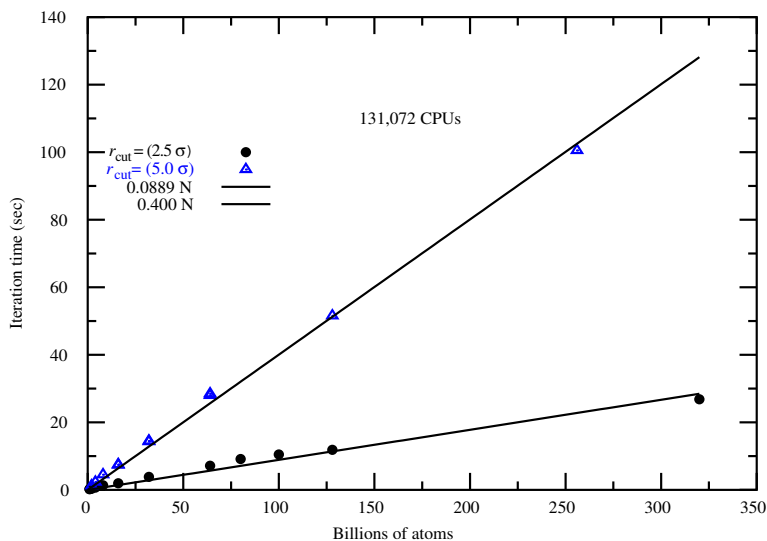


Fig. 4. Iteration time for up to 320 billion atoms on 131 072 processors of the BGL architecture at Livermore. The achieved performance for 320 billion atoms simulated by a Lennard-Jones potential with a cut-off of 2.5σ was 27.2TFlops, and 48.1TFlops for 256 billion atoms interacting via a Lennard-Jones potential with a larger cut-off of 5.0σ .

Before August 2005, only half of the 131 072 CPUs were installed, and iteration times were measured for 1–8 billion atoms on 4096–65 536 processors. The data from multiple runs nicely collapse and are fit by Eq. (1) with $a = 7.67$ sec and $b = 0.779$ sec (see Fig. 3). Weak scaling (varying the problem size and the number of CPUs, such that the problem size per CPU is constant, to verify that the execution time is constant) is to within noise perfectly achieved and not shown here. The strong scaling behavior (fixed problem size and varied processor counts, to measure the speed-up) exhibits an almost linear speed-up down to 30 000 atoms per processor (inset of Fig. 3). Based on Eq. (1), the communication time only becomes equal or greater than the calculation time for fewer than 1000 atoms per processor.

After the full configuration was completed in August 2005, runs with as many as 320 billion atoms were performed (see Fig. 3).³ Either due to changes in system timers, or to the increase in total processor count from 2^{16} to 2^{17} (requiring an `int` rather than a `short int` to identify processors), we found a slight increase in measured iteration times, and therefore Flops, as compared to the pre-August configuration. Therefore, our 25 TFlops performance on 65 536 processors translated into only 48.1 TFlops on the full machine.

The calculations were all performed in double precision which means that each particle structure consisted of 88 bytes (position, velocity, and force vectors [24 bytes each], integer for type [4 bytes] and a tag [4 bytes], and a double to analyze/characterize the atoms [8 bytes]). Hence, 320 billion atoms consume 26 226 GB for the particle structure information alone. Additional memory is needed for the

cell structure, various buffers, and the executable and operating system on each node. This additional memory only amounts to less than 20% of the whole memory consumption (Livermore's BGL has 32 768 GB main memory). These numbers suggest that *SPaSM* with its efficient memory management would be able to perform with almost a trillion atoms in single precision on ASC Purple, which has a main memory of 48832 GB.¹⁶

4. Conclusions

Sophisticated use of the ever growing computational power makes it possible to investigate challenges in physics and other areas of science on the fundamental atomic level. The 320 billion atoms simulated on BGL could represent a cubic piece of metal with an edge length well over a micrometer. With one iteration only taking slightly more than 20 seconds, a physical simulation on the micro scale in 3-dimensions is now in principle possible in a matter of weeks. One of the biggest challenges to make these ultra-large scale production runs feasible is the analysis and the input/output of the enormous data produced on long term storage such as huge parallel file systems. The input/output is also very important for restart capabilities since time to failure of these enormous multi-processor machines is usually only on the order of days. Another challenge — as represented by Los Alamos' Roadrunner project — is the efficient use of future large-scale hybrid architectures involving Streaming SIMD Extensions (SSE) processors and IBM Cell Broadband Engine (Cell BE) accelerators.

Acknowledgments

We would like to thank Steve Louis, Michel McCoy, and James Peery for enabling early access to BGL; and Ryan Braby, Jeff Fier, Robin Goldstone, Fred Streitz, and Aidan Thompson for their assistance in overcoming various hardware and software related hurdles. Finally, we like to thank Hans Herrmann and Dietrich Stauffer for encouraging us to write this report. This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under contract No. DE-AC52-06NA25396, with funding by ASC and LDRD-20050066DR.

References

1. B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).
2. A. Gara *et al.*, *IBM J. Res. & Dev.* **49**, 195 (2005).
3. T. C. Germann, K. Kadau and P. S. Lomdahl, *Supercomputing '05, SC05 IEEE ACM* 1-59593-061-2/05/0011 (2005).
4. P. S. Lomdahl, P. Tamayo, N. Grønbech-Jensen and D. M. Beazley, *Proceedings of Supercomputing 1993*, ed. G. S. Ansell (IEEE Computer Society Press, Los Alamitos, CA, 1993), p. 520.
5. D. M. Beazley and P. S. Lomdahl, *Parallel Computing* **20**, 173 (1994).

6. D. M. Beazley and P. S. Lomdahl, *Comput. Phys.* **11**, 230 (1997).
7. K. Kadau, T. C. Germann and P. S. Lomdahl, *Int. J. Mod. Phys. C* **15**, 193 (2004).
8. J. Roth, F. Gähler and H.-R. Trebin, *Int. J. Mod. Phys. C* **11**, 317 (2000).
9. D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 1995).
10. <http://www.unix-mcs.anl.gov/mpi>
11. B. L. Holian and P. S. Lomdahl, *Science* **280**, 2085 (1998).
12. T. C. Germann, B. L. Holian, P. S. Lomdahl and R. Ravelo, *Phys. Rev. Lett.* **84**, 5351 (2000).
13. K. Kadau, T. C. Germann, P. S. Lomdahl and B. L. Holian, *Science* **296**, 1681 (2002).
14. K. Kadau, T. C. Germann, N. G. Hadjiconstantinou, P. S. Lomdahl, G. Dimonte, B. L. Holian and B. J. Alder, *Proc. Natl. Acad. Sci.* **101**, 5851 (2004).
15. T. C. Germann, K. Kadau, I. M. Longini, Jr. and C. A. Macken, *Proc. Natl. Acad. Sci.* **103**, 5935 (2006).
16. <http://www.top500.org>