

Anton, a Special-Purpose Machine for Molecular Dynamics Simulation

By David E. Shaw, Martin M. Deneroff, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, Kevin J. Bowers, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J.P. Grossman, C. Richard Ho, Douglas J. Jerardi, István Kolossváry, John L. Klepeis, Timothy Layman, Christine McLeavey, Mark A. Moraes, Rolf Mueller, Edward C. Priest, Yibing Shan, Jochen Spengler, Michael Theobald, Brian Towles, and Stanley C. Wang

Abstract

The ability to perform long, accurate molecular dynamics (MD) simulations involving proteins and other biological macromolecules could in principle provide answers to some of the most important currently outstanding questions in the fields of biology, chemistry, and medicine. A wide range of biologically interesting phenomena, however, occur over timescales on the order of a millisecond—several orders of magnitude beyond the duration of the longest current MD simulations.

We describe a massively parallel machine called Anton, which should be capable of executing millisecond-scale classical MD simulations of such biomolecular systems. The machine, which is scheduled for completion by the end of 2008, is based on 512 identical MD-specific ASICs that interact in a tightly coupled manner using a specialized high-speed communication network. Anton has been designed to use both novel parallel algorithms and special-purpose logic to dramatically accelerate those calculations that dominate the time required for a typical MD simulation. The remainder of the simulation algorithm is executed by a programmable portion of each chip that achieves a substantial degree of parallelism while preserving the flexibility necessary to accommodate anticipated advances in physical models and simulation methods.

1. INTRODUCTION

Molecular dynamics (MD) simulations can be used to model the motions of molecular systems, including proteins, cell membranes, and DNA, at an atomic level of detail. A sufficiently long and accurate MD simulation could allow scientists and drug designers to visualize for the first time many critically important biochemical phenomena that cannot currently be observed in laboratory experiments, including the “folding” of proteins into their native three-dimensional structures, the structural changes that underlie protein function, and the interactions between two proteins or between a protein and a candidate drug molecule. Such simulations could answer some of the most important open questions in the fields of biology and chemistry, and have the potential to make substantial contributions to the process of drug development.

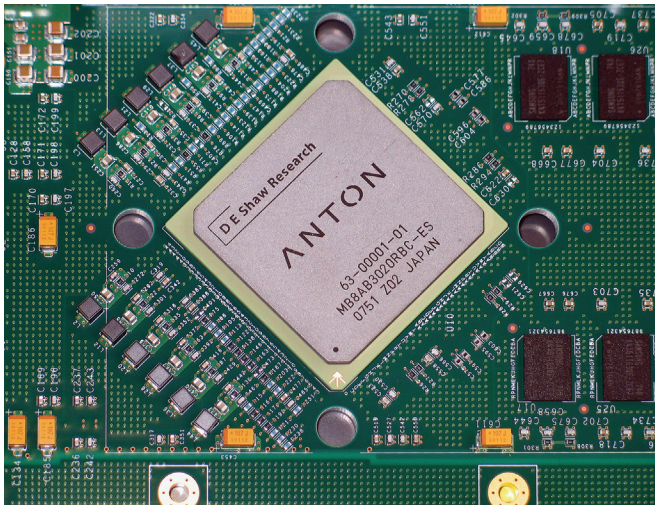
Many of the most important biological processes occur over timescales on the order of a millisecond. MD simulations on this timescale, however, lie several orders of magnitude beyond the reach of current technology; only a few MD

runs have thus far reached even a microsecond of simulated time, and the vast majority have been limited to the nanosecond timescale. Millisecond-scale simulations of a biomolecular system containing tens of thousands of atoms will in practice require that the forces exerted by all atoms on all other atoms be calculated in just a few microseconds—a process that must be repeated on the order of 10^{12} times. These requirements far exceed the current capabilities of even the most powerful commodity clusters or general-purpose scientific supercomputers.

This paper describes a specialized, massively parallel machine, named Anton, that is designed to accelerate MD simulations by several orders of magnitude, bringing millisecond-scale simulations within reach for molecular systems involving tens of thousands of atoms. The machine, which is scheduled for completion by the end of 2008, will comprise 512 processing nodes in its initial configuration, each containing a specialized MD computation engine implemented as a single ASIC. The molecular system to be simulated is decomposed spatially among these processing nodes, which are connected through a specialized high-performance network to form a three-dimensional torus. Anton’s expected performance advantage stems from a combination of MD-specific hardware that achieves a very high level of arithmetic density and novel parallel algorithms that enhance scalability by reducing both intra- and inter-chip communication. Figure 1 is a photograph of one of the first Anton ASICs.

In designing Anton and its associated software, we have attempted to attack a somewhat different problem than the ones addressed by several other projects that have deployed significant computational resources for MD simulations. The Folding@Home project,¹⁶ for example, has obtained a number of significant and interesting scientific results by using as many as 250,000 PCs (made available over the Internet by volunteers) to simulate a very large number of separate molecular trajectories, each of which is limited to the timescale accessible on a single PC. While a great deal can be learned from a large number of independent MD trajectories, many other important problems require the examination of a single, very long trajectory—the principal task for which Anton is designed. Other projects, such as FASTRUN,⁶ MDGRAPE,²² and MD Engine,²³ have produced special-purpose hardware to accelerate the most computationally expensive elements of an MD simulation. Such hardware reduces the cost of MD simulations, particularly for large molecular systems, but Amdahl’s law and communication bottlenecks prevent the efficient use of enough such chips in parallel

Figure 1: Anton ASIC. One of the first Anton ASICs, which arrived in January 2008.



to extend individual simulations beyond microsecond timescales.

Anton is named after Anton van Leeuwenhoek, whose contributions to science and medicine we hope to emulate in our own work. In the 17th century, van Leeuwenhoek, often referred to as the “father of microscopy,” built high-precision optical instruments that allowed him to visualize for the first time an entirely new biological world that had previously been unknown to the scientists of his day. We view Anton (the machine) as a sort of “computational microscope.” To the extent that we and other researchers are able to increase the length of MD simulations, we would hope to provide contemporary biological and biomedical researchers with a tool for understanding organisms and their diseases on a still smaller length scale.

2. MD COMPUTATION ON ANTON

An MD computation simulates the motion of a collection of atoms (the *chemical system*) over a period of time according to the laws of classical physics.¹ Time is broken into a series of discrete *time steps*, each representing a few femtoseconds of simulated time. A time step has two major phases. *Force calculation* computes the force on each particle due to other particles in the system. *Integration* uses the net force on each particle to update that particle’s position and velocity.

2.1. Force calculation

Interatomic forces are calculated based on a *molecular mechanics force field* (or simply *force field*), which models the forces on each atom as a function of the spatial coordinates of all atoms. In commonly used biomolecular force fields,^{9, 11, 15} the forces consist of three components: *bond forces*, involving groups of atoms separated by no more than three covalent bonds; *van der Waals forces*, computed between pairs of atoms separated by less than some cutoff radius (usually chosen between 5 and 15 Å); and *electrostatic forces*, which are the most computation-

ally intensive as they must be computed between all pairs of atoms.

Anton uses the *k*-space Gaussian split Ewald method (*k*-GSE)¹⁸ to reduce the computational workload associated with the electrostatic interactions. This method divides the electrostatic force calculation into two components. The first decays rapidly with particle separation and is computed directly for all particle pairs separated by less than a cutoff radius. We refer to this contribution, together with the van der Waals interactions, as *range-limited interactions*. The second component, *long-range interactions*, decays more slowly, but can be computed efficiently by mapping charge from particles to a regular mesh (*charge spreading*), taking the fast Fourier transform (FFT) of the mesh charges, multiplying by an appropriate function in Fourier space, performing an inverse FFT, and then computing forces on the particles from the resulting mesh values (*force interpolation*).

To parallelize range-limited interactions, our machine uses an algorithm we developed called the *NT method*.¹⁹ The NT method achieves both asymptotic and practical reductions in required interprocessor communication bandwidth relative to traditional parallelization methods. It is one of a number of *neutral territory methods* that employ a spatial assignment of particles to nodes, but that often compute the interaction between two particles using a node on which neither particle resides.^{4, 7, 10, 14, 17, 21}

2.2. Integration

The integration phase uses the results of force calculation to update atomic positions and velocities, numerically integrating a set of ordinary differential equations describing the motion of the atoms. The numerical integrators used in MD are nontrivial for several reasons. First, the integration algorithm and the manner in which numerical issues are handled can have a significant effect on accuracy. Second, some simulations require the integrator to calculate and adjust global properties such as temperature and pressure. Finally, one can significantly accelerate most simulations by incorporating constraints that eliminate the fastest vibrational motions. For example, constraints are typically used to fix the lengths of bonds to all hydrogen atoms and to hold water molecules rigid.

3. WHY SPECIALIZED HARDWARE?

A natural question is whether a specialized machine for molecular simulation can gain a significant performance advantage over general-purpose hardware. After all, history is littered with the corpses of specialized machines, spanning a huge gamut from Lisp machines to database accelerators. Performance and transistor count gains predicted by Moore’s law, together with the economies of scale behind the development of commodity processors, have driven a history of general-purpose microprocessors outpacing special-purpose solutions. Any plan to build specialized hardware must account for the expected exponential growth in the capabilities of general-purpose hardware.

We concluded that special-purpose hardware is warranted in this case because it leads to a much greater improve-

ment in absolute performance than the expected speedup predicted by Moore's law over our development time period, and because we are currently at the cusp of simulating timescales of great biological significance. We expect Anton to run simulations over 1000 times faster than was possible when we began this project. Assuming that transistor densities continue to double every 18 months and that these increases translate into proportionally faster processors and communication links, one would expect approximately a tenfold improvement in commodity solutions over the five-year development time of our machine (from conceptualization to bring-up). We therefore expect that a specialized solution will be able to access biologically critical millisecond timescales significantly sooner than commodity hardware.

To simulate a millisecond within a couple of months, we must complete a time step every few microseconds, or every few thousand clock ticks. The sequential dependence of successive time steps in an MD simulation makes speculation across time steps extremely difficult. Fortunately, specialization offers unique opportunities to accelerate an individual time step using a combination of architectural features that reduce both computational latency and communication latency.

For example, we reduced computational latency by designing:

- Dedicated, specialized hardware datapaths and control logic to evaluate the range-limited interactions and to perform charge spreading and force interpolation. In addition to packing much more computational logic on a chip than is typical of general-purpose architectures, these pipelines use customized precision for each operation.
- Specialized, yet programmable, processors to compute bond forces and the FFT and to perform integration. The instruction set architecture (ISA) of these processors is tailored to the calculations they perform. Their programmability provides flexibility to accommodate various force fields and integration algorithms.
- Dedicated support in the memory subsystem to accumulate forces for each particle.

We reduced communication latency by designing:

- A low-latency, high-bandwidth network, both within an ASIC and between ASICs, that includes specialized routing support for common MD communication patterns such as multicast and compressed transfers of sparse data structures.
- Support for choreographed "push"-based communication. Producers send results to consumers without the consumers having to request the data beforehand.
- A set of autonomous direct memory access (DMA) engines that offload communication tasks from the computational units, allowing greater overlap of communication and computation.
- Admission control features that prioritize packets carrying certain algorithm-specific data types.

We balance our design very differently from a general-purpose supercomputer architecture. Relative to other high-performance computing applications, MD uses much communication and computation but surprisingly little memory. The entire architectural state of an MD simulation of 25,000 particles, for example, is just 1.6 MB, or 3.2 KB per node in a 512-node system. We exploit this property by using only SRAMs and small L1 caches on our ASIC, with all code and data fitting on-chip in normal operation. Rather than spending silicon area on large caches and aggressive memory hierarchies, we instead dedicate it to communication and computation.

It is serendipitous that the most computationally intensive parts of MD—in particular, the electrostatic interactions—are also the most well established and unlikely to change as force field models evolve, making them particularly amenable to hardware acceleration. Dramatically accelerating MD simulation, however, requires that we accelerate more than just an "inner loop."

Calculation of electrostatic and van der Waals forces accounts for roughly 90% of the computational time for a representative MD simulation on a single general-purpose processor. Amdahl's law states that no matter how much we accelerate this calculation, the remaining computations, left unaccelerated, would limit our maximum speedup to a factor of 10. Hence, we dedicated a significant fraction of silicon area to accelerating other tasks, such as bond force computation, constraint computation, and velocity and position updates, incorporating programmability as appropriate to accommodate a variety of force fields and integration methods.

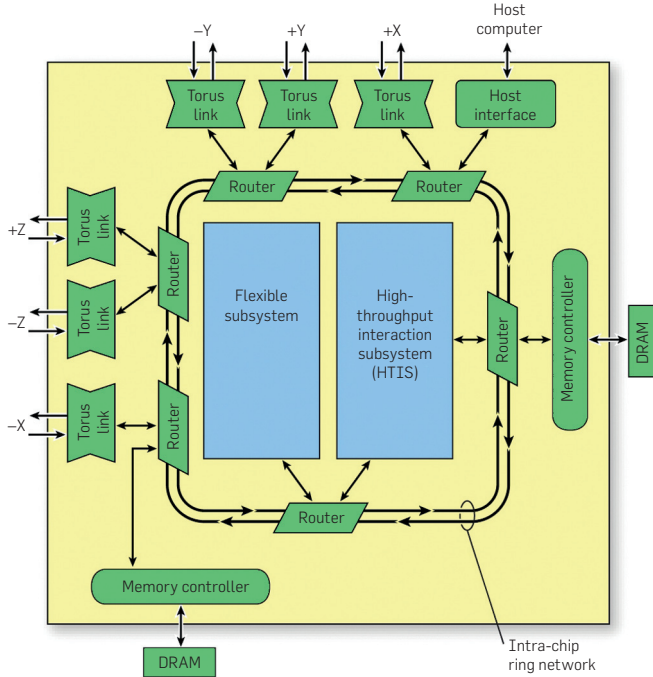
4. SYSTEM ARCHITECTURE

The building block of the system is a node, depicted in Figure 2. Each node comprises an MD-specific ASIC, attached DRAM, and six ports to the system-wide interconnection network. Each ASIC has four major subsystems, which are described briefly in this section. The nodes, which are logically identical, are connected in a three-dimensional torus topology (which maps naturally to the periodic boundary conditions frequently used in MD simulations). The initial version of Anton will be a 512-node torus with eight nodes in each dimension, but our architecture also supports larger and smaller toroidal configurations. The ASICs are clocked at a modest 400 MHz, with the exception of one double-clocked component in the *high-throughput interaction subsystem* (HTIS), discussed in the following section.

4.1. High-throughput interaction subsystem

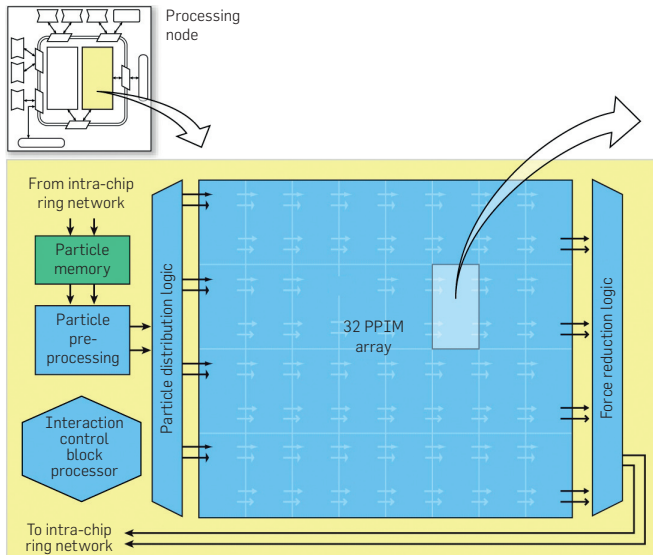
The HTIS calculates range-limited interactions and performs charge spreading and force interpolation. The HTIS, whose internal structure is shown in Figure 3, applies massive parallelism to these operations, which constitute the bulk of the calculation in MD. It provides tremendous arithmetic throughput using an array of 32 *pairwise point interaction modules* (PPIMs) (Figure 3), each of which includes a force calculation pipeline that runs at 800 MHz and is capable of computing the combined electrostatic and van der

Figure 2: Anton processing node. The HTIS performs the most demanding calculations in an MD simulation. The flexible subsystem performs the remaining MD calculations, coordinates MD time step activity, and manages housekeeping tasks.



Waals interactions between a pair of atoms at every cycle. This 26-stage pipeline (Figure 4) includes adders, multipliers, function evaluation units, and other specialized datapath elements. Inside this pipeline, we use customized numerical precisions: functional unit width varies across the different pipeline stages but still produces a sufficiently accurate 32-bit result.

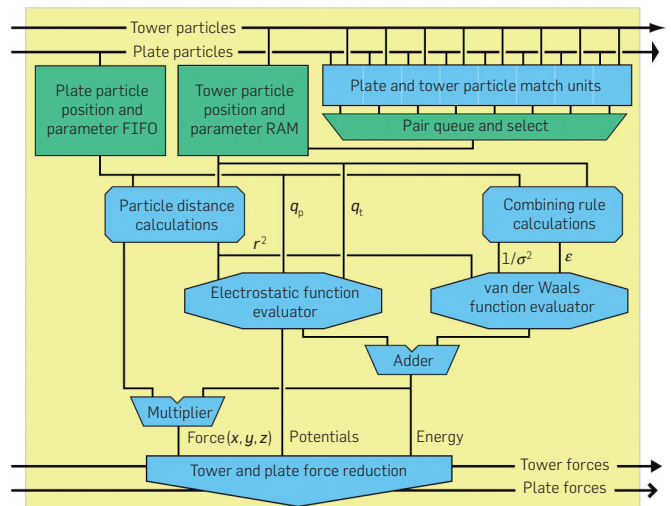
Figure 3: High-throughput interaction subsystem. The HTIS comprises an array of 32 PPIMs and an embedded control processor to coordinate the distribution of particles to the PPIM array.



In order to keep the pipelines busy with useful computation, the remainder of the HTIS must determine pairs of atoms that need to interact, feed them to the pipelines, and aggregate the pipelines' outputs. This proves a formidable challenge given communication bandwidth limitations between ASICs, between the HTIS and other subsystems on the same ASIC, and between pipelines within the HTIS. We address this problem using an architecture tailored for *direct product selection reduction operations* (DPSRs), which take two sets of points and perform computation proportional to the product of the set sizes but only require input and output volume proportional to the sum of their sizes. The HTIS considers interactions between all atoms in a region called the *tower* and all atoms in a region called the *plate*. Each atom in the tower is assigned to one PPIM, while each atom in the plate streams by all the PPIMs. Eight *match units* in each PPIM perform several tests, including a low-precision distance check, to determine which pairs of plate and tower particles are fed to the force calculation pipeline. Because the HTIS is a streaming architecture, with no feedback in its computational path, it is simple to scale the PPIM array to any number of PPIMs. The HTIS also includes an *interaction control block* processor, which controls the flow of data through the HTIS. More detail about the HTIS and about DPSR operations can be found in the proceedings of this year's HPCA conference.¹³

The PPIMs are the most hard-wired component of our architecture, reflecting the fact that they handle the most computationally intensive parts of the MD calculation. That said, even the PPIMs include programmability where we anticipate potential future changes to force fields. For instance, the functional forms for van der Waals and electrostatic interactions are specified using SRAM look-up tables, whose contents are determined at runtime.

Figure 4: PPIM detail. This figure gives a sense of the numerical calculation units in a PPIM. The top portion of the figure shows the match units and particle memories. The lower portion shows the general structure of the force calculation pipelines.



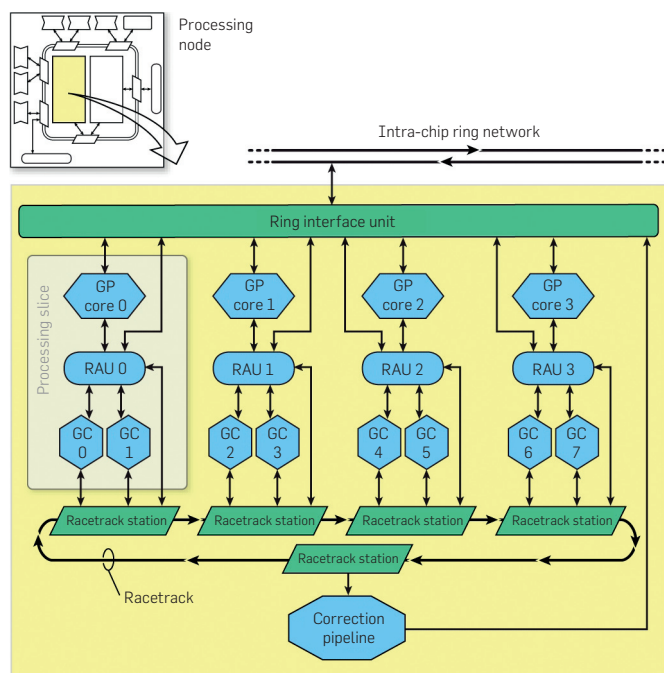
4.2. Flexible subsystem

The *flexible subsystem* controls the ASIC and handles all other computations, including the bond force calculations, the FFT, and integration. Figure 5 shows the components of the flexible subsystem. Four identical *processing slices* form the core of the flexible subsystem. Each slice comprises a general-purpose core with its caches, a *remote access unit* (RAU) that performs autonomous data transfers, and two *geometry cores* (GCs), which are programmable cores that perform most of the flexible subsystem's computation. The RAU is a programmable data transfer engine that enables the flexible subsystem to participate in "push" communication, both of flooding messages sent from the processor cores and tracking incoming messages to determine when work is ready to be done. Each GC is a dual-issue, statically scheduled, 4-way SIMD processor with pipelined multiply accumulate support and instruction set extensions to support common MD calculations. Other components of the flexible subsystem include a correction pipeline, which computes force correction terms; a racetrack, which serves as a local, internal interconnect for the flexible subsystem components; and a ring interface unit, which allows the flexible subsystem components to transfer packets to and from the communication subsystem. More detail about the flexible subsystem is given in a second paper at this year's HPCA conference.¹²

4.3. Communication subsystem

The *communication subsystem* provides high-speed, low-latency communication both between ASICs and among

Figure 5: Flexible subsystem. It is a collection of four identical processing slices (one of which is indicated by a box at the left) and a correction pipeline unit. The processing slices communicate with each other and with the correction pipeline via the racetrack. The various components communicate with the intra-chip communication ring via the ring interface unit shown at the top of the figure.



the subsystems within an ASIC. Between chips, each torus link provides 5.3GB/s full-duplex communication with a hop latency around 50 ns. Within a chip, two 256-bit, 400 MHz communication rings link all subsystems and the six inter-chip torus ports. The communication subsystem supports efficient multicast, provides flow control, and provides class-based admission control with rate metering. The communication subsystem also allows access to an external host computer system for input and output of simulation data.

4.4. Memory subsystem

The *memory subsystem* provides access to the ASIC's attached DRAM. In addition to basic memory read/write access, the memory subsystem supports accumulation and synchronization. Special memory write operations numerically add incoming write data to the contents of the memory location specified in the operation. These operations implement force, energy, potential, and spread charge accumulations, reducing the computation and communication load on the flexible subsystem. By taking advantage of the attached DRAM, Anton will be able to simulate chemical systems with billions of atoms.

5. PERFORMANCE AND ACCURACY MEASUREMENTS

In this section, we show that the performance of Anton significantly exceeds that of other MD platforms, and that Anton is capable of performing simulations of high numerical accuracy. Because we do not yet have a working 512-node segment, performance estimates for our machine come from our performance simulator. The cycle fidelity of this simulator varies across components, but we expect overall fidelity better than $\pm 20\%$.

5.1. Performance comparison

We compare the performance of various MD platforms in terms of simulation rate (nanoseconds of simulated time per day of execution) on a particular chemical system. In this section and in Section 5.2, we use a system with 23,558 atoms in a cubic box measuring 62.2 Å on a side. This system represents dihydrofolate reductase (DHFR), a protein targeted by various cancer drugs, surrounded by water.

The highest-performing MD codes achieve a simulation rate of a few nanoseconds per day for DHFR on a single state-of-the-art commodity processor core.⁸ Existing multiprocessor machines with high-performance interconnects achieve simulation rates up to a few hundred nanoseconds per day using many hundreds or thousands of processor cores.^{2, 3, 5}

We expect a 512-node Anton system to achieve a simulation rate of approximately 14,500 nanoseconds per day for DHFR, enabling a millisecond simulation in just over two months. While the performance of general-purpose machines will undoubtedly continue to improve, Anton's performance advantage over other MD platforms significantly exceeds the speedup predicted by Moore's law over the next few years. A more detailed performance comparison of Anton and other MD platforms is given in the proceedings of last year's ISCA conference.²⁰

5.2. Accuracy

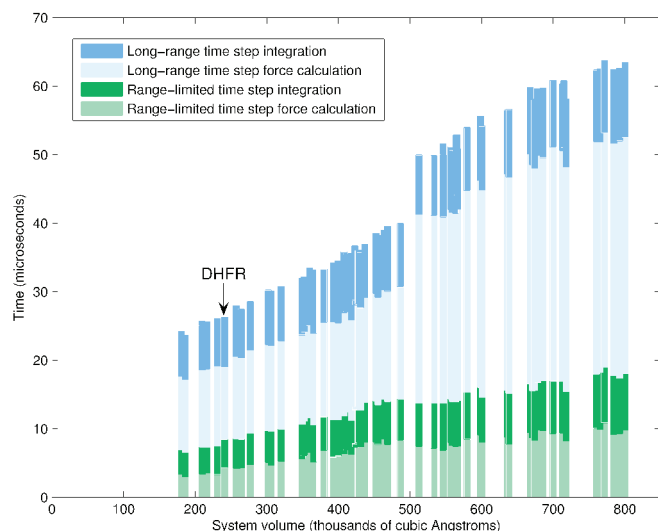
To quantify the accuracy of force computation on Anton, we measured the *relative rms force error*, defined as the rms error in the force on all particles divided by the rms force.¹⁸ For the DHFR system with typical simulation parameters, Anton achieves a relative rms force error of 1.5×10^{-4} . A relative rms force error below 10^{-3} is generally considered sufficiently accurate for biomolecular MD simulations.²⁵

We also measured *energy drift* to quantify the overall accuracy of our simulations. An exact MD simulation would conserve energy exactly. Errors in the simulation generally lead to an increase in the overall energy of the simulated system with time, a phenomenon known as energy drift. We measured energy drift over 5 ns of simulated time (2 million time steps) for DHFR using a bit-accurate numerical emulator that exactly duplicates Anton's arithmetic. While the simulation exhibited short-term energy fluctuations of a few kcal/mol (about 0.001% of the total system energy), there was no detectable long-term trend in total energy. MD studies are generally considered more than adequate even with a significantly higher energy drift.²⁴

5.3. Scaling with chemical system size

Figure 6 shows the scaling of performance with chemical system size. Within the range where chemical systems fit in on-chip memory, we expect performance to scale roughly linearly with the number of atoms, albeit with occasional jumps as different operating parameters change to optimize performance while maintaining accuracy. The largest discontinuity in simulation rate occurs at a system volume

Figure 6: Scaling of performance for a 512-node version of Anton with increasing chemical system size. The graph shows a stacked bar chart for each chemical system, with the height of each stack proportional to the simulation time, assuming that long-range forces are evaluated every other time step. Each stack represents the time required to execute two consecutive time steps; one is a "long-range time step" that includes calculation of long-range electrostatics by *k*-GSE, and the other is a "range-limited time step" that does not. The chemical systems represent proteins and nucleic acids of various sizes, surrounded by water.



of approximately 500,000 Å³ when we change from a $32 \times 32 \times 32$ FFT grid to a $64 \times 64 \times 64$ FFT grid, reflecting the fact that our code supports only power-of-two-length FFTs. This lengthens the long-range calculation because the number of grid points increases by a factor of 8. Overall, the results are consistent with supercomputer scaleup studies—as we increase chemical system size, Anton's efficiency improves because of better overlap of communication and computation, and because calculation pipelines operate closer to peak efficiency.

6. CONCLUSION

We are currently in the process of building a specialized, massively parallel machine, called Anton, for the high-speed execution of MD simulations. We expect Anton to be capable of simulating the dynamic, atomic-level behavior of proteins and other biological macromolecules in an explicitly represented solvent environment for periods on the order of a millisecond—about three orders of magnitude beyond the reach of current MD simulations. The machine uses specialized ASICs, each of which performs a very large number of application-specific calculations during each clock cycle. Novel architectural and algorithmic techniques are used to minimize intra- and inter-chip communication, providing an unusually high degree of scalability.

While it contains programmable elements that could in principle support the parallel execution of algorithms for a wide range of other applications, Anton was not designed to function as a general-purpose scientific supercomputer, and would not in practice be well suited for such a role. Rather, we envision Anton serving as a computational microscope, allowing researchers to observe for the first time a wide range of biologically important structures and processes that have thus far proven inaccessible to both computational modeling and laboratory experiments. C

References

- Adcock, S.A. and McCammon, J.A. Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chemical Review*, 106:1589–1615, 2006.
- Bhatele, A., Kumar, S., Mei, C., Phillips, J.C., Zheng, G., and Kale, L.V. Overcoming scaling challenges in biomolecular simulations across multiple platforms, to appear in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008)*, Miami, FL, 2008.
- Bowers, K.J., Chow, E., Xu, H., Dror, R.O., Eastwood, M.P., Gregersen, B.A., Klepeis, J.L., Kolossvary, I., Moraes, M.A., Sacerdoti, F.D., Salmon, J.K., Shan, Y., and Shaw, D.E. Scalable algorithms for molecular dynamics simulations on commodity clusters. *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*, Tampa, FL, 2006.
- Bowers, K.J., Dror, R.O., and Shaw, D.E. Zonal methods for the parallel execution of range-limited N-body problems. *Journal of Computational Physics*, 221(1):303–329, 2007.
- Fitch, B.G., Rayshubskiy, A., Eleftheriou, M., Ward, T.J.C., Giampapa, M.E., Pitman, M.C., Pitera, J.W., Swope, W.C., and Germain, R.S. Blue matter: scaling of N-body simulations to one atom per node. *IBM Journal of Research and Development*, 52(1/2), 2008.
- Fine, R.D., Dimmler, G., and Levinthal, C. FASTRUN: A special purpose, hardware computer for molecular simulation. *Proteins: Structure, Function, and Genetics*, 11(4):242–253, 1991 (erratum: 14(3):421–422, 1992).
- Germain, R.S., Fitch, B., Rayshubskiy, A., Eleftheriou, M., Pitman, M.C., Suits, F., Giampapa, M., and Ward, T.J.C. Blue matter on blue gene/L: Massively parallel computation for biomolecular simulation. *Proceedings of the Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES + ISSS '05)*, New York, NY, 2005.
- Hess, B., Kutzner, C., van der Spoel, D., and Lindahl, E. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4(2):435–447, 2008.
- Jorgensen, W.L., Maxwell, D.S., and Tirado-Rives, J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids.

- Journal of the American Chemical Society*, 118(45):11225–11236, 1996.
10. Kalé, L., Skeel, R., Bhandarkar, M., Brunner, R., Gursoy, A., Krawetz, N., Phillips, J., Shinozaki, A., Varadarajan, K., and Schulten, K., NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151(1):283–312, 1999.
 11. Kollman, P.A., Dixon, R.W., Cornell, W.D., Fox, T., Chipot, C., and Pohorille, A. The development/ application of a "Minimalist" organic/ biomolecular mechanic forcefield using a combination of ab initio calculations and experimental data, in *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, van Gunsteren, W.F. and Weiner, P.K. eds., Dordrecht, Netherlands:ESCOM, pp. 83–96, 1997.
 12. Kuskin, J.S., Young, C., Grossman, J.P., Batson, B., Deneroff, M.M., Dror, R.O., and Shaw, D.E. Incorporating flexibility in Anton, a specialized machine for molecular dynamics simulation. *Proceedings of the 14th International Symposium on High-Performance Computer Architecture (HPCA-14)*, Salt Lake City, UT, 2008.
 13. Larson, R.H., Salmon, J.K., Dror, R.O., Deneroff, M.M., Young, C., Grossman, J.P., Shan, Y., Klepeis, J.L., and Shaw, D.E. High-throughput pairwise point interactions in Anton, a specialized machine for molecular dynamics simulation. *Proceedings of the 14th International Symposium on High-Performance Computer Architecture (HPCA-14)*, Salt Lake City, UT, 2008.
 14. Liem, S.Y., Brown, D., and Clarke, J.H.R. Molecular dynamics simulations on distributed memory machines. *Computer Physics Communications*, 67(2):261–267, 1991.
 15. MacKerell, A.D. Jr., Bashford, D., Bellott, M., Dunbrack, R.L., Evanseck, J.D., Field, M.J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kucera, K., Lau, F.T.K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D.T., Prodhom, B., Reiher, III, W.E., Roux, B., Schlenkrich, M., Smith, J.C., Stote, R., Straub, J., Watanabe, M., Wiórkiewicz-Kucera, J., Yin, D., and Karplus, M.J. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.
 16. Pande, V.S., Baker, I., Chapman, J., Elmer, S.P., Khaliq, S., Larson, S.M., Rhee, Y.M., Shirts, M.R., Snow, C.D., Sorin, E.J., and Zagrovic, B. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68(1):91–109, 2003.
 17. Plimpton, S.J., Attaway, S., Hendrickson, B., Swegle, J., Vaughan, C., and Gardner, D. Transient dynamics simulations: Parallel algorithms for contact detection and smoothed particle hydrodynamics. *Proceedings of the ACM/IEEE Conference on Supercomputing (Supercomputing '96)*, Pittsburgh, PA, 1996.
 18. Shan, Y., Klepeis, J.L., Eastwood, M.P., Dror, R.O., and Shaw, D.E. Gaussian split Ewald: A fast Ewald mesh method for molecular simulation. *Journal of Chemical Physics*, 122:054101, 2005.
 19. Shaw, D.E. A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions. *Journal of Computational Chemistry*, 26(13):1318–1328, 2005.
 20. Shaw, D.E., Deneroff, M.M., Dror, R.O., Kuskin, J.S., Larson, R.H., Salmon, J.K., Young, C., Batson, B., Bowers, K.J., Chao, J.C., Eastwood, M.P., Gagliardo, J., Grossman, J.P., Ho, C.R., Ierardi, D.J., Kolossváry, I., Klepeis, J.L., Layman, T., McLeavey, C., Moraes, M.A., Mueller, R., Priest, E.C., Shan, Y., Spengler, J., Theobald, M., Towles, B., and Wang, S.C., Anton, a special purpose machine for molecular dynamics simulation. *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA '07)*, San Diego, CA, 2007.
 21. Snir, M. A Note on N-body computations with cutoffs. *Theory of Computing Systems*, 37:295–318, 2004.
 22. Taiji, M., Narumi, T., Ohno, Y., Futatsugi, N., Suenaga, A., Takada, N., and Konagaya, A., Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations. *Proceedings of the ACM/IEEE Conference on Supercomputing (SC03)*, Phoenix, AZ, 2003.
 23. Toyoda, S., Miyagawa, H., Kitamura, K., Amisaki, T., Hashimoto, E., Ikeda, H., Kusumi, A., and Miyakawa, N. Development of MD engine: High-speed accelerator with parallel processor design for molecular dynamics simulations. *Journal Computational Chemistry*, 20(2): 185–199, 1999.
 24. Wang, W. and Skeel, R.D. Fast evaluation of polarizable forces. *Journal of Chemical Physics*, 123(16):164107, 2005.
 25. Zhou, R., Harder, E., Xu, H., and Berne, B.J. Efficient multiple time step method for use with Ewald and particle mesh Ewald for large biomolecular systems. *Journal of Chemical Physics*, 115(5): 2348–2358, 2001.

All authors are affiliated with D.E. Shaw Research, New York.

David E. Shaw (David.Shaw@DEShawResearch.com) is also affiliated with the Center for Computational Biology and Bioinformatics, Columbia University, New York.

© 2008 ACM 0001-0782/08/0700 \$5.00

ACM TRANSACTIONS ON INTERNET TECHNOLOGY



Transactions on Internet Technology (TOIT). This quarterly publication encompasses many disciplines in computing—including computer software engineering, middleware, database management, security, knowledge discovery and data mining, networking and distributed systems, communications, and performance and scalability—all under one roof. *TOIT* brings a sharper focus on the results and roles of the individual disciplines and the relationship among them. Extensive multi-disciplinary coverage is placed on the new application technologies, social issues, and public policies shaping Internet development. **Subscribe Today!**

PRODUCT INFORMATION

ISSN: 1533-5399 **Price:** \$41 Professional Member
Order Code: (140) \$36 Student Member
 \$170 Non-Member
 \$16 Air Service (for residents
 outside North America only)

TO PLACE AN ORDER

Contact ACM Member Services

Phone: 1.800.342.6626 **Email:** acmhelp@acm.org
 (U.S. and Canada) **Mail:** ACM Member Services
 +1.212.626.0500 General Post Office
 (Global) P.O. Box 30777
Fax: +1.212.944.1318 New York, NY
 (Hours: 8:30am–4:30pm, Eastern Time) 10087-0777 USA

www.acm.org/pubs/periodicals/toit



Association for
Computing Machinery

Advancing Computing as a Science & Profession

AD28