Avalon: An Alpha/Linux Cluster Achieves 10 Gflops for $150k

Submitted for the 1998 Gordon Bell Price/Performance Prize

Michael S. Warren [*] Timothy C. Germann [†] Peter S. Lomdahl [‡] David M. Beazley [§]
John K. Salmon [¶]

Abstract

We present two calculations from the disciplines of condensed matter physics and astrophysics. The simulations were performed on a 70 processor DEC Alpha cluster (Avalon) constructed entirely from commodity personal computer technology and freely available software, for a cost of 152 thousand dollars. Avalon performed a 60 million particle molecular dynamics (MD) simulation of shock-induced plasticity using the SPaSM MD code. This simulation sustained approximately 10 Gflops over a 44 hour period, and saved 68 Gbytes of raw data. The resulting price/performance is $15/Mflop, or equivalently, 67 Gflops per million dollars. This is more than a factor of three better than last year's price/performance winners. This simulation is very similar to that which won part of the 1993 Gordon Bell performance prize using a 1024-node CM-5.

Avalon also performed a gravitational treecode N-body simulation of galaxy formation using 9.75 million particles, which sustained an average of 6.78 Gflops over a 26 hour period. This simulation is exactly the same as that which won a Gordon Bell price/performance prize last year on the Loki cluster, at a total performance 7.7 times that of Loki, and a price/performance 2.6 times better than Loki.

1 Introduction

Building upon the foundation of the BEOWULF project [1] and our own success with Loki [10], it has become possible to construct high-performance computers entirely out of commodity components and free software, thus obtaining a significant price/performance advantage over typical parallel machines. Last year Loki and Hyglac, clusters of 16 Pentium Pro processors, were the first such machines to win a Gordon Bell price/performance prize [15]. This year, changing to the the DEC Alpha microprocessor (which is also found in the Cray T3E series) and using a more advanced fast ethernet switch, we have improved total performance by almost a factor of ten, and improved price/performance by over a factor of three.

In 1992, Warren and Salmon were awarded a Gordon Bell Performance Prize [11] for "Astrophysical N-body Simulations Using Hierarchical Tree Data Structures." In 1993, Lomdahl and Beazley were awarded a Gordon Bell Performance Prize [3] for "50 GFlops Molecular Dynamics on the Connection Machine 5." It is now possible to run similar simulations on a machine constructed out of mail-order parts and free software for a cost of $152k. We emphasize that the simulations we report here are only a few of the programs which we have run in the three week lifetime of this machine. While not directly relevant to this entry, some benchmarks with which the judges may be familiar should demonstrate that Avalon is a general-purpose parallel machine with superior price/performance on many applications: Parallel Linpack on 68 processors runs at 19.7 Gflops. Results for the NAS NPB version 2.3 Class B benchmarks on 64 processors include: BT 2222 Mflops, SP 1038 Mflops, LU 3030 Mflops, and MG 2080 Mflops. These results are all within factors of two or three in absolute performance of 64-processor parallel supercomputers such as the SGI Origin 2000, the Cray T3E, and the IBM SP2, all of which cost a million dollars or more. In particular, a 64 processor 250 Mhz SGI Origin 2000 with 8 Gbytes of memory has a list price of over 1.8 million dollars.

[*] Theoretical Astrophysics, Los Alamos National Laboratory, Los Alamos, NM 87545, Email: *msw@lanl.gov*
[†] Condensed Matter Theory, Los Alamos National Laboratory, Los Alamos, NM 87545, Email: *tcg@lanl.gov*
[‡] Condensed Matter Theory, Los Alamos National Laboratory, Los Alamos, NM 87545, Email: *pxl@lanl.gov*
[§] Department of Computer Science, University of Utah, Salt Lake City, UT 84112, Email: *beazley@cs.utah.edu*
[¶] CACR, California Institute of Technology, Mail Code 206-49, Pasadena, CA 91125, Email: *johns@cacr.caltech.edu*

2 Avalon

As a co-operative venture at the Los Alamos Center for Nonlinear Studies (CNLS) and Theoretical Division, Avalon was constructed from 70 nodes for a total cost of $152,175 as described in Table 1. All of the operating system software (RedHat Linux), software tools (GNU) and compilers (egcs-1.02) used for these results are freely available. MPI was used for the message passing layer, using our own TCP socket based implementation of the basic MPI functions (SWAMPI). Copies of the sales invoices for all components are available from the authors. The ethernet equipment was bought under government contract, but a quick search found at least three Internet mail-order companies offering this hardware to the public for nearly the same price as we paid, and at least one offered it for less. The individual nodes were delivered to Los Alamos on April 10 completely assembled with the operating system already installed and configured. The only labor required to complete assembly of the cluster was unpacking the nodes from their shipping boxes and the attachment of power and network cables. This took 28 man-hours of labor, which we have included in the price at $100/hour. The machine was operational on April 13, three days after delivery. Further information and pictures of the machine are available at `http://cnls.lanl.gov/avalon`.

Qty.	Price	Ext.	Description
70	1701	119070	DEC Alpha 164LX 533 MHz 21164A, with 2x64Mb SDRAM DIMM ECC memory (128 Mbyte/node), Quantum 3240 Mbyte IDE Hard Drive, Kingston 100 Mb Fast Ethernet PCI Card, cables, assembly, Linux install, 3 year parts/labor warranty
2	6027	12054	3Com SuperStack II 3900, 36-port Fast Ethernet
4	968	3872	Gigabit uplink modules for 3900s
1	10046	10046	3Com SuperStack II 9300, 12-port Gigabit Ethernet
3	1055	3165	Cyclades Cyclom 32-YeP serial concentrators
70	10	700	Serial cables (20 ft)
4	117	468	Shelving
28	100	2800	Final assembly labor
Total		$152,175	$2174 per node 1.066 Gflops peak per node

Table 1: Avalon architecture and price.

3 The SPaSM molecular dynamics code

SPaSM (Scalable Parallel Short-range Molecular dynamics) is a message-passing C code originally written [2] for the CM-5, and was recognized with a 1993 Gordon Bell performance prize for achieving a sustained rate of 50 Gflops in simulating 131 million atoms on a 1024-node CM-5 [3]. Since then, we have increased the portability of the code by replacing CMMD and CM I/O calls with a set of wrapper functions, removed all assembler code, developed a visualization library, and combined these elements into an interactive package using the Python scripting language (see, e.g., [4, 5]). The critical code in the force calculation loop is written entirely in C, and hand-optimized for modern superscalar architectures by loop unrolling, macro expansion, and careful use of `register` variables. We have used SPaSM to study a number of materials science problems at an experimentally inaccesible atomistic level, including fracture mechanisms [6], dislocation interactions [7], and the mechanisms of shock-induced plasticity [8].

3.1 A 60.8 million atom shock wave simulation

To demonstrate the capability of Avalon to sustain an actual large-scale scientific simulation, on April 24-26 we carried out a simulation of 60.8 million atoms in an fcc crystal structure initially moving at a uniform velocity into an infinitely massive piston face ("momentum mirror"). As the resulting shock wave propagates, stacking faults are generated and the shock front becomes more and more non-planar. This simulation is similar to a series of 10 million

Number of	Number of Processors			
Particles	16	32	64	70
1,000,000	4.94 (2.75)	2.86 (5.22)	1.66 (9.89)	1.47 (10.93)
2,000,000	9.39 (2.79)	4.89 (5.57)	3.07 (10.27)	2.76 (10.82)
4,000,000	16.84 (2.99)	9.12 (5.73)	4.76 (11.43)	4.68 (11.70)
8,000,000	31.59 (3.07)	17.17 (5.87)	8.92 (11.74)	8.70 (11.73)
16,000,000	62.28 (3.12)	31.69 (6.12)	17.47 (12.00)	17.03 (11.99)
32,000,000		61.53 (6.30)	31.35 (12.37)	31.11 (12.47)
64,000,000			61.39 (12.64)	60.48 (12.83)

Table 2: Average time per timestep in seconds, with Gflops in parentheses.

atom simulations recently carried out [8], but with a cross-sectional area four times as large (200 × 200 fcc unit cells). We use the usual Lennard-Jones 6-12 potential, truncated at a cutoff distance of $r_{max} = 4\sigma$, where σ is the nearest-neighbor spacing.

We ran this simulation on 68 nodes for a total of 2000 timesteps, which required nearly 44 hours of simulation time. As this was an actual production run, we carried out the usual "extras," such as computing the potential and kinetic energies every 20 timesteps, generating GIF images[1] and checkpointing the data every 100 timesteps. The checkpointing was carried out independently by each processor, writing out its particle data (50 MB on average) to the local disk. To demonstrate the capability to restart from such checkpoint files, this simulation was intentionally stopped and restarted 3 times, including one restart on a different set of processors from the checkpoint (by manually transferring checkpoint files from the "old" node(s) to the "new" one(s)). Despite all of these add-ons, the entire simulation still ran at an average of 79 seconds per timestep, with a total of 1.56×10^{15} floating point operations over a wall clock time of 1.58×10^{5} seconds. (Add, subtract, multiply, and comparison are each counted as one operation, and divide as five operations.) We thus find a sustained throughput for this simulation of **9.9 Gflops** and a price/performance of $15/Mflop.

3.2 Timings and performance

We have also carried out a more extensive series of short (10 timestep) shock wave runs for timing purposes, with different numbers of particles and processors. The average wall clock times per timestep (including force calculation, timestep integration, and particle redistribution) are shown in Table 2, along with the corresponding Gflop rates obtained by counting the actual number of floating point operations in both the force and integration routines.

To give an idea of how large the fraction of time spent in the force calculation is, the 64 million atom calculation on 70 nodes required 57.27 s to compute forces, 1.27 s for the (leapfrog Verlet) integration, and 1.97 s to redistribute particles after the integration step. The force calculation alone, which involves message-passing of cells of particles between processors (5287 message passing calls per timestep), performs at an overall rate of **13.52 Gflops**, which drops to the quoted value of 12.83 Gflops after the time spent in the integration and redistribution steps is included. We measure a total message-passing time of 12.65 s, 21% of the total 60.48 s. As the number of particles is reduced this fraction increases, to 38% for the 1 million particle run. Despite this, the overall performance is still nearly 11 Gflops, and the iteration time of 1.47 s represents a speedup of 3.4 over the 16 node run, or 77% parallel efficiency. For larger problems the parallel efficiency is between 80 and 90%, demonstrating that the near-perfect scalability of the SPaSM code found on the CM-5 [3] is reasonably well preserved. For an additional point of reference, an identical 64 million particle calculation on 70 nodes of a Cray/SGI Origin 2000 (with 195 MHz R10000 CPUs) requires 76.22 seconds per timestep, or 10.18 Gflops (26% slower than Avalon). The same code was used in both cases, with use of the best available compiler optimization flags. Erring on the side of SGI, we assume that with 250 Mhz processors an Origin 2000 on this code would perform at the same rate as Avalon. Taking the ratio of the list prices, we find that Avalon has a price/performance advantage of about 12 for this code. A discount from list price might lower this factor to about 9. These figures do not take into account the the $125k/yr hardware maintenance contract for the Origin.

[1] Selected images are available for viewing at: `http://bifrost.lanl.gov/MD/MD.html`

4 The Hashed Oct-Tree Library

5 Conclusion

We believe that the simulations reported above are clearly of the "supercomputer" class, requiring nearly 8 Gbytes of memory, and performing in the range of 10^{15} floating point operations. The absolute performance of Avalon (running irregular production-type simulations) is factors of two and ten higher than the two price/performance winners last year, while improving overall price/performance by over a factor of three. Even more extraordinary, on these applications Avalon demonstrates price/performance an order of magnitude superior to commercial machines of equivalent performance.

References

[1] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer. BEOWULF: A parallel workstation for scientific computation. In *Proceedings of the 1995 International Conference on Parallel Processing (ICPP)*, pages 11–14, 1995.

[2] D.M. Beazley and P.S. Lomdahl. Message-passing multi-cell molecular dynamics on the Connection Machine 5. *Parallel Computing* **20***, 173-195 (1994).*

[3] P.S. Lomdahl, P. Tamayo, N. Grønbech-Jensen, and D.M. Beazley. 50 GFlops Molecular Dynamics on the Connection Machine 5. In *Proceedings of Supercomputing 93* (IEEE Computer Society Press, Los Alamitos, CA, 1993), pp. 520-527.

[4] D.M. Beazley and P.S. Lomdahl. Lightweight Computational Steering of Very Large Scale Molecular Dynamics Simulations. In *Proceedings of Supercomputing 96*.
`http://www.supercomp.org/sc96/proceedings/SC96PROC/BEAZLEY/INDEX.HTM`

[5] D.M. Beazley and P.S. Lomdahl. Controlling the data glut in large-scale molecular-dynamics simulations. *Computers in Physics* **11***(3), 230-238 (1997).*

[6] S.J. Zhou, D.M. Beazley, P.S. Lomdahl, and B.L. Holian. Large-Scale Molecular Dynamics Simulations of Three-Dimensional Ductile Failure, *Phys. Rev. Lett.* **78***, 479 (1997).*

[7] S.J. Zhou, D.L. Preston, P.S. Lomdahl, and D.M. Beazley. Large-Scale Molecular Dynamics Simulations of Dislocation Intersection in Copper, *Science* **279***, 1525-1527 (1998).*

[8] B.L. Holian and P.S. Lomdahl. Plasticity Induced by Shock Waves in Nonequilibrium Molecular-Dynamics Simulations. Submitted to *Science (1998), LAUR-98-702.*

[9] Alan H. Karp. Speeding Up N-body Calculations on Machines without Hardware Square Root. *Scientific Programming*, 1:133–140, 1993.

[10] M. S. Warren, D. J. Becker, M. P. Goda, J. K. Salmon, and T. Sterling. Parallel supercomputing with commodity components. In H. R. Arabnia, editor, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'97)*, pages 1372–1381, 1997.

[11] M. S. Warren and J. K. Salmon. Astrophysical N-body simulations using hierarchical tree data structures. In *Supercomputing '92*, pages 570–576, Los Alamitos, 1992. IEEE Comp. Soc.

[12] M. S. Warren and J. K. Salmon. A parallel hashed oct-tree N-body algorithm. In *Supercomputing '93*, pages 12–21, Los Alamitos, 1993. IEEE Comp. Soc.

[13] M. S. Warren and J. K. Salmon. A parallel, portable and versatile treecode. In *Seventh SIAM Conference on Parallel Processing for Scientific Computing*, pages 319–324, Philadelphia, 1995. SIAM.

[14] M. S. Warren and J. K. Salmon. A portable parallel particle program. *Computer Physics Communications*, 87:266–290, 1995.

[15] M. S. Warren, J. K. Salmon, D. J. Becker, M. P. Goda, T. Sterling, and G. S. Winckelmans. Pentium Pro inside: I. a treecode at 430 Gigaflops on ASCI Red, II. Price/performance of $50/Mflop on Loki and Hyglac. In *Supercomputing '97*, Los Alamitos, 1997. IEEE Comp. Soc.