Avalon: Champagne Computing on a Beer Budget Extended Abstract

Michael S. Warren *Aric A. Hagberg*
J. David Moulton *David Neal*

Theoretical Division and Center for Nonlinear Studies
Los Alamos National Laboratory

John K. Salmon

Center for Advanced Computing Research, California Institute of Technology

{msw,aric,moulton,dneal}@lanl.gov
{johns}@cacr.caltech.edu
http://cnls.lanl.gov/avalon/

Abstract

Avalon is a 140 processor Alpha/Linux Beowulf cluster constructed entirely from commodity personal computer technology and freely available software. Computational Physics simulations performed on Avalon resulted in the award of a 1998 Gordon Bell price/performance prize for significant achievement in parallel processing. Avalon ranked as the 113th fastest computer in the world on the November 1998 TOP500 list, obtaining a result of 47.8 Gigaflops on the parallel Linpack benchmark.

Avalon currently provides over 15,000 node-hours of production computing time per week, split among about 10 production users. Obtaining an equivalent amount of computing through Los Alamos institutional sources would cost a minimum of $45,000 per week. The machine also supports code development for another 50 users. The largest single simulation was performed in April and May 1998 using the SPaSM molecular dynamics code, which computed a total of 1.12×10^{16} floating point operations. This simulation is among the few scientific simulations to have ever involved more than 10 Petaflops of computation.

The price of hardware and final assembly labor for Avalon totalled $313,000 dollars. The monetary cost of the development and

OS software used for the applications mentioned above was $0. Perhaps most extraordinary, all of the hardware and software maintenance on the machine is performed in the spare time of four people, averaging less than 10 man-hours of labor per week overall.

1 Introduction

Linux is the primary technology which has enabled the practical use of parallel computers built from commodity hardware. Its importance was recognized from the very beginning the Beowulf project [**?**]. Our experience first with Loki [**?**] and now with Avalon has served as a clear example of the advantages of Linux, and we are proud to have contributed to its increasing recognition. A more extensive introduction will be included in the full version of this paper.

2 Zen and the Art of Beowulf

Instruction #1 — Assembly of Beowulf requires great peace of mind.

It is possible to enumerate a set of guidelines about how to construct a Beowulf, filled with details about disks and CPUs and network performance. This, however, misses the point of what makes these machines different. What the machine is made of is far less important than what it is capable of doing, and one should keep in mind that the capabilities of a system as complicated as a parallel computer can not easily be determined by simply looking at its component parts.

2.1 Hardware

Avalon was initially constructed from 70 nodes for a total cost of $152,175. All of the operating system software (RedHat Linux 5.0), software tools (GNU) and compilers (egcs) used are freely available. The individual nodes were purchased from Carrera Computers and delivered to Los Alamos on April 10 completely assembled with the operating system already installed and configured. We also used four 3Com SuperStack II 3900 36-port fast ethernet switches, with 2 Gigabit uplink modules added to each one. This provides 3 Gigabit links on each switch, which are trunked together and attached to a 3Com SuperStack II 9300 12-port Gigabit Ethernet switch. Overall, this provides a switched network of 144 fast ethernet ports, at a cost of about $300 per port.

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. Avalon was doubled in size to 140 nodes on September 10, and a memory upgrade also doubled the memory per node to 256 Mbytes. The final configuration of the machine is found in Table 1.

The machine has been more reliable than even our initially optimistic expectations. During a three month period during the summer of 1998 not a single hardware compo-

Qty.	Price	Ext.	Description
140	1701	238140	DEC Alpha 164LX 533 MHz 21164A, with 2x128Mb SDRAM DIMM ECC memory (256 Mbyte/node), Quantum 3240 Mbyte IDE Hard Drive, Kingston 100 Mb Fast Ethernet PCI Card, cables , assembly, Linux install, 3 year parts/labor warranty
70	285	19950	128 Mb Memory upgrade for initial 70 nodes
4	6027	24108	3Com SuperStack II 3900, 36-port Fast Ethernet
8	968	7744	Gigabit uplink modules for 3900s
1	10046	10046	3Com SuperStack II 9300, 12-port Gigabit Ethernet
5	1055	5275	Cyclades Cyclom 32-YeP serial concentrators
140	10	1400	Serial cables (20 ft)
7	117	819	Shelving
56	100	5600	Final assembly labor
Total		$313,082	$2236 per node 1.066 Gflops peak per node

Table 1: Avalon architecture and price.

nent was replaced on any of the nodes. The downtime of the machine has been completely dominated by failures which were beyond our control. An unneeded upgrade of the air conditioning units in our machine room (which then failed) required us to shut down much of the machine on several occasions for several days. The full version of this paper will include detailed failure statistics.

We estimate that Avalon consumes about 18 kilowatts of power while under load. The cost of electricity for the machine works out to about $15k per year. With space charges at Los Alamos of about $50 per square foot, the charge for the space to put the machine adds an additional $6k per year. Scaling the purchase price by Moore's Law, when the machine is about 6 years old, it will cost more per year for power and space than the value of the computation produced by the machine. This puts a clear upper limit on the useful lifetime of the machine. More expensive commercial machines do not reach this limit, since they start out being 10 times as expensive.

2.2 OS install

OS install on an Alpha is somewhat different than Intel-based machines. One needs linload.exe, milo and a kernel. All three will fit on a single floppy, but milo and kernels are specific to each particular Alpha motherboard.

We use two methods of OS install. The first is a "bootstrap" method, where one starts from scratch with a clean disk and a RedHat CD-ROM or ftp image. The second is "cloning," where one takes a current

disk image and duplicates it to a new disk. The bootstrap is clearly required initially. We considered using a bootstrap method exclusively, but found that cloning had advantages such as keeping new nodes exactly consistent with software updates that have already been performed on the working disk images.

The RedHat "kickstart" method is the usual method of unattended installation, but we were unfamiliar with its control syntax, and it was not flexible enough to partition the disks the way we wanted them. The Avalon node disks each contain 7 partitions. A dos /boot partition for linload, milo and the kernel, a swap partition, /tmp, /var, /usr, /home and /. The rationale behind many partitions is primarily to insure that corruption in one partition does not affect others, and to allow OS upgrades to be performed without affecting user data in /home.

Rather than use kickstart, we created a small 10 Mbyte nfsroot partition on a server with `yard`. We used a kernel with nfsroot support to boot using this partition, and ran a small shell script which partitioned the hard drive, and installed the desired RPM packages directly. One could use a ramdisk-based filesystem to bootstrap the OS install without an NFS server, but the limited space provided by a floppy makes things more difficult than necessary. One subtlety in this process is that RPMS can not be installed all at once (a bug in `rpm`?). A bit of trial and error determined that two `rpm -i` calls would work, as long as the first one installed the appropriate 20 packages.

The cloning procedure simply requires a short shell script which makes disk partitions, and uses `tar` to copy the data off of another disk. The other disk can be on the same machine, or on another machine on the network (in this case one must provide some mechanism to boot the machine with the empty disk in it). The final task of the script is to modify the two network configuration files in order to give the new disk the correct identity.

3 Applications

The first major simulation performed by Avalon was a 60 million particle molecular dynamics (MD) simulation of shock-induced plasticity using the SPaSM MD code [**?**]. This simulation ran for a total of 332 hours on Avalon, computing a total of 1.12×10^{16} floating point operations. Also for the Gordon Bell prize entry, Avalon 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. Both of these simulations are reported in more detail in [**?**].

During the shakedown of the initial system, Avalon made the largest contribution of any group in the world to the solution of the Certicom Cryptographic Challenge, organized by Robert J. Harley of the Institut National de Recherche en Informatique et Automatique (INRIA), France. The solution to

the ECC2K-95 problem was found after 21.6 trillion elliptic curve operations, carried out in 25 days by 47 people. The $4000 prize for finding the solution was donated to the Free Software Foundation.

We will report on the other major applications currently running on Avalon (Nonlinear Dynamics, Partial Differential Equations, Phase Transitions in the early Universe, Eigenvalue solvers, Monte Carlo simulations of 3d spin glasses, 3d supernova simulations) in the full version of this paper.

4 Software

4.1 Prsh

Prsh is a script we have developed which implements a "parallel" rsh. Prsh runs a command on a list of remote processors with optional timeouts, output flushing, status reports, etc. It is implemented with about 200 lines of perl. It has turned out to be a tremendously powerful and flexible way to extend the usual UNIX command-line interface across a parallel machine. A typical prsh command would be:

```
prsh -- uptime
```

This shows the uptime on nodes defined by the environment variable `PRSH_NODES`. The last argument to prsh can be any valid command you would ordinarily give on the command-line. `ls, cp, shutdown, zap, date`, etc.

A more explicit way to specify nodes may also be used:

```
prsh a00 a01 a02 a03 -- uptime
```

Typing machine names quickly becomes tiresome on a large machine, but another small perl script comes to the rescue. `nseq` simply creates a string of node names. The example above is equivalent to `prsh ‘nseq 0 3‘ -- uptime`, which is not much shorter than the explicit command, but `prsh ‘nseq 0 140‘ -- shutdown -h` certainly is.

Prsh calls `rsh` asynchronously, so all commands execute in parallel. The prsh command in practice feels very responsive, with commands running across 140 Avalon processors completing in less than 2 seconds. prsh also can be told to use ssh as an option, so all system maintenance commands are performed using ssh-agent authentication and prsh from the front-end.

4.2 SWAMPI

Since about 1990 we have maintained a small message passing library which we have used to run our parallel codes on networks of workstations. The library was initially implemented with UDP, since UDP performance on the machines and operating systems at that time was much superior to TCP performance. The library implemented only about 10 basic communication functions, but those were sufficient to run our parallel codes, since we believed (or soon discovered) that depending on anything beyond the basics would not run reliably on most parallel machines.

With the discovery that Linux TCP bandwidth over fast ethernet was equivalent to UDP performance, we were able to substantially simplify our simple message passing li-

brary (since we no longer had to handle message sequencing and retries ourselves). While doing this we also took the opportunity to align the code more closely with the MPI standard. This rewrite was done during a couple of weeks in early 1997, and was the message-passing library used by our codes almost exclusively Loki. This library (named SWAMPI) is implemented in about 2000 lines of code, and implements the 24 most commonly used MPI functions. This may be contrasted with about 100,000 lines of code in the Ohio State LAM implementation [**?**], and 250,000 in the MPICH distribution [**?**] (of which 40k lines are examples, and 100k lines are device specific). Needless to say, it is considerably easier to understand what is happening in 2000 lines of our own code vs 100,000 of somebody else's.

SWAMPI was used on Avalon from the beginning, showing increased reliability and performance over MPICH and LAM. During the few days we had to optimize the Parallel Linpack benchmark, we were able to insert SWAMPI at the core level of MPICH and improve overall performance by over 20%. We were unable to use SWAMPI alone, because it did not provide the MPI functionality required by the numerous abstraction layers in the SCALAPACK and BLACS code which implemented the parallel Linpack algorithm.

We are now re-examining the choice of TCP, since it appears that using a connection oriented protocol results in scaling difficulties when using more than 100 processors at once. Particularly, the select() system call does not scale well when there are many open file descriptors. What would be ideal would be a connectionless yet reliable protocol.

5 Some Perspective

Avalon ranks at #113 on the TOP500 supercomputers list [**?**], at 48,600 Mflops. TOP500 is based on Parallel Linpack performance [**?**]. Avalon peak performance is 149,400 Mflops, and Avalon would rank around #50 in the world in a list based on peak performance.

To put Avalon's speed in perspective, only 12 countries in the world had machines faster than Avalon in November, 1997. Avalon is almost twice as fast as the fastest machine in Central or South America. While Avalon is thirty times slower than the fastest computer in the USA, the rapid growth of Linux in countries such as Mexico will likely lead to dramatic changes in the supercomputing landscape in those nations. Another view of the same data is that only 10 commercial enterprises in the world have ever built machines faster than Avalon. The two fastest machines listed under Compaq (formerly DEC) in the TOP500 list are actually Linux clusters designed at Los Alamos and Sandia National Labs.

Why are commercial machines so expensive? It is certainly not because supercomputer companies are charging much more than they should, as is borne out by the long list of vendors which have declared bankruptcy. Are Beowulf machines filling a small niche, or is something deeper involved? It is at least possible that the answer lies entirely in software. It may cost 10 times as much as developing the hardware to develop

a one-of-a-kind OS which attempts to be all things to all supercomputer users.

6 Outstanding Problems

While Avalon performs adequately, there are of course many areas in which improvements would be valuable [**?**]. The final version of this paper will include discussion in the areas of running DEC Unix binaries under Linux legally, queuing, problems with very large (greater than 50 Gbyte) filesystems and gigabit ethernet.

7 Appendix: Avalon Timeline

1996

16 May – Proposal for LANL Research funds to build a cluster turned down
13 Aug – "Commodity Computing" memo to T-Division Director received warmly
Sep – Loki constructed, 16 processor Pentium Pro cluster for $63,000

1997

Aug – Spot prices show Loki could now be built for less than $25,000
20 Nov – Loki wins 1997 Gordon Bell Price/Performance Prize
28 Nov – Loki appears on the cover of Linux Journal
12 Dec – "Loki II" memo proposes a larger cluster to T-Division Director

1998

14 Jan – linux.lanl.gov comes on-line
16 Jan – T-Division director Richard Slansky unexpectedly leaves us
2 Feb – Meeting at CNLS to discuss building Alpha Cluster of about 64 nodes
3 Feb – Preliminary DEC Alpha quotes from DCG and Aspen
4 Feb – Two SX systems ordered from DCG for evaluation
10 Feb – Preliminary specs to send to BUS for official quotes from 6 vendors
10 Mar – 50 PC164 LXs ordred from Carerra
17 Mar – 20 more ordered
23 Mar – Avalon chosen as a name
30 Mar – Linpack: .2 GFlops (on Alpha PX164SX node)
6 Apr – New libdgemm from Goto
6 Apr – Linpack: .4 Gflops (on Alpha PX164SX node)
8 Apr – One SX sacrificed for serial controller
8 Apr – Confirmed April 15 as TOP500 deadline
10 Apr – First shipment form Carerra arrives at CNLS (70 nodes)
10 Apr – Discover EtherpowerII cards don't work with new board (rev C0)
10 Apr – Try Kingston cards, work, order cards for replacement
10 Apr – 8 machines up and running with borrowed Kingston cards
11 Apr – Wiring all nodes, last gasp to get Etherpower IIs to work
11 Apr – Linpack: 2.1 Gflops (4x2) 9600x9600
13 Apr – Linpack: 11.2 Gflops (8x8)
13 Apr – Linpack: 13.7 Gflops (4x16)
14 Apr – 2.1.96 kernel, All 70 nodes working
14 Apr – Linpack: 15.2 GFlops (5x14)
14 Apr – Linpack: 16.5 GFlops (4x17)
15 Apr – Linpack: 16.8 GFlops (4x17) 30464x30464 matrix, submitted to Top500

15 Apr – Computations for Gordon Bell prize started
15 Apr – cnls.lanl.gov/avalon up
15 Apr – Linpack: 19.3 GFlops (4x17) with swampi, amended numbers to Top500
16 Apr – Celebrate with beers at one of the only two bars in Los Alamos
17 Apr – avalon@lanl.gov list formed
27 Apr – avalon appears on Linpack list
30 Apr – Gordon Bell entry complete, computations and paper
30 Apr – News: Slashdot
1 May – Avalon begins production mode
1 May – Avalon FAQ
5 May – Avalon starts crunching Certicom challenge in background
21 May – Avalon wins $500 for ECC2K-95 crypto challange, $4000 to FSF
27 May – 20.0 Gflops with 2.1.103 kernel
18 Jun – Avalon makes #315 on Top500 list. First Linux cluster to make list.
18 Jun – LANL press release "Los Alamos Mail-order Supercomputer Among World's Fastest"
25 Jun – Avalon has first node failure in 47 days.
1 Jul – Money secured to upgrade to 140 nodes
5 Jul – "Do-It-Yourself Supercomputers" in Wired
10 Jul – News: PC World Today, CNN, Slashdot
17 Jul – 20 more nodes ordered from Carerra
28 Jul – Avalon in Forbes article "For the Love of Hacking"
31 Jul – First hint of AC troubles: water on floor
5 Aug – Avalon has 35 users
21 Aug – AC not working, apparently fixed by power cycling unit
21 Aug – 20 nodes arrive from Carrera
1 Sep – 50 more units shipped from CA
9 Sep – power installed for new nodes, based on work order issued in March
9 Sep – 2.1.120 kernel on 90 nodes
10 Sep – 50 nodes arrive
11 Sep – All 140 nodes running
11 Sep – Linpack: 34.5 Gflops (5x28) (some nodes still 128M)
11 Sep – Memory arrives, all nodes upgraded to 256MB or higher
12 Sep – Bad memory discovered in a few nodes
12 Sep – 1GB nodes don't work, downgrade to 512MB
12 Sep – Linpack: 47.7 Gflops (5x28), submitted to Top500
15 Sep – Avalon upgrade complete,140 nodes 36GB memory
15 Sep – Production runs restart

15 Sep – 30 GFlops on MD code, 18GFlops on treecode,
15 Sep – amended Gordon Bell entry
17 Sep – Bad SCSI disk in front end wastes lots of time
24 Sep – Avalon down for air conditioner replacement in machine room
24 Sep – Nodes moved to allow room for AC units to get through
24 Sep – Nodes covered with plastic to protect from construction debris
6 Oct – Avalon moved again
6 Oct – One AC unit working, not enough to cool entire machine room
13 Oct – AC functioning
13 Oct – 140 nodes up with 2.1.125 kernel
13 Oct – 250GB RAID online
13 Oct – Linpack: 48.6 GFlops (5x28)
20 Oct – New AC unit failing, most of Avalon down
21 Oct – AC 'fixed', Avalon on
22 Oct – AC not working, most of Avalon off
29 Oct – AC 'fixed', part replaced
2 Nov – Avalon all on
3 Nov – Slashdot again
5 Nov – Ranked #114 on Top500 list
12 Nov – Avalon receives Gordon Bell prize for price/performance
26 Nov – Ranked #113 on amended Top500 list
15 Dec – Avalon in Science Magazine "From Army of Hackers, an Upstart Operating System"
17 Dec – AC not working
18 Dec – Half of Avalon shut down due to AC problems
19 Dec – AC 'fixed',
21 Dec – Avalon to remain half off during holiday closure because of AC problems

1999

4 Jan – All nodes back on
5 Jan – Development nodes upgraded to kernel 2.2.0-pre4
8 Jan – AC not working, all nodes remain on
12 Jan – AC 'fixed'
15 Jan – Extended Abstract submitted for Linux Expo, world domination to follow