UP1.43

# Plasma Simulation on a Cluster of PCs[1]

Charles Karney, Darin Ernst,
Douglas McCune, and Ernest Valeo

Plasma Physics Laboratory
Princeton University

APS DPP Meeting, Seattle, November, 1999

[1]Work supported by US DoE Contract No. DE–AC02–76–CH03073

*This poster is available online at*
http://w3.pppl.gov/puffin/

## Abstract

With advances in the speeds of personal computers, it is now possible to perform large computing tasks on a cluster of PCs. We are constructing a small cluster with two objectives: to benchmark the system with a variety of plasma codes in production use at PPPL [2]; and to ascertain the ease with which such a system can be maintained and used. The system will consist of 9 dual-processor 400MHz Pentium PCs, each with 512MB of RAM and a 14GB disk and connected via a dedicated 100Mb Ethernet network. The computers will use the Linux [3] operating system which is the basis of many so-called Beowulf [4] clusters, including the NERSC PC Cluster [5]. A PC cluster potentially has the computational throughput of a specialized massively-parallel machine costing ten times as much; but the PC cluster is missing the high-speed interconnect of the parallel machine. Thus, the actual performance will be dictated by issues of network bandwidth and latency, both of which we will attempt to quantify. Codes which use one of the standard message-passing interfaces, PVM [6] or MPI [7], and which have small communication requirements are expected to run efficiently on this system.

## Motivation

The processing speed of PC chips is now competitive with the chips in supercomputers.

A parallel computer system built out of commodity PCs *may* offer performance / price ratios an order of magnitude greater than special-purpose parallel computers.

Large parallel computers are often purchased to serve the needs of a large group of users, and allocation of resources is frequently contentious.

The puffin cluster was assembled to assess the usefulness of a cluster of PCs for solving computational problems in plasma physics.

## Goals

- to determine the computational throughput for plasma codes
- to identify bottlenecks in the system
- to measure the overall reliability (software and hardware)
- to minimize the administrative overhead

## Hardware

8 worker PCs (unit price $1875) with

- Dual Intel Boxed Pentium III 450MHz CPU
- ASUS P2B-D dual processor motherboard
- Integrated 512k ECC on-chip L2 cache per CPU
- 512MB PC100 ECC SDRAM, 2 DIMMs, 8ns
- IBM 14.4GB Ultra DMA EIDE drive, 7200 RPM
- SVGA graphics card
- Teac 1.44MB floppy drive
- Med tower ATX case with 6 drive bays, 250W power
- 3COM Fast EtherLink XL PCI (3C905) 10/100BaseTX
- Redhat Linux 6.0 pre-configured w/SMP kernel
- Two year replacement warranty

Front-end PC (unit price $3300) configured as above with

- Symbios 53C895 Ultra2 Wide SCSI controller
- 2nd HD: Seagate Cheetah 18GB 10k RPM SCSI drive
- Toshiba XM-6502B 40X speed ATAPI CDROM drive
- ATI Xpert 98 AGP, 8MB, 230MHz (replaces SVGA card)
- Full tower ATX case with 10 drive bays, 300W power
- Logitech MouseMan PS/2 3 button mouse
- KeyTronic 2000 104-key PS/2 keyboard
- 2nd 3COM Fast EtherLink XL PCI (3C905) 10/100BaseTX
- Redhat 6.0 Linux CDs/Documentation

PC vendor is SW Technologies [8]

8-port KVM Switch, Midrange Solutions, including cables (price $1000).

24-port 100Mb switched Cabletron Ethernet hub (price $3497). (Cheaper models are available. This model was dictated by compatibility with other network equipment at PPPL.)

21in Sony GDM-500PS Monitor (price $1000).

TOTAL COST $24,000

Note well: to save costs the worker PCs are ordered *without*:

- monitor
- keyboard
- mouse
- sound card
- CDROM drive
- fancy graphics card

The equipment was located in the PPPL computer center which has the necessary

- space: allow 0.15 $m^2$ per PC (with 6 PCs per rack);
- power: each PC consumes about 250W;
- heat: all that power ends up as heat; adequate air-conditioning is essential;
- network: the whole cluster requires just one 100Mb connection to the outside world.

## Hardware setup

The PCs were installed in 2 laboratory racks from PLT.

hub
switch

puffin03
puffin04

puffin05
puffin06

puffin07
puffin08

puffin01
puffin02

common
monitor

puffin00
monitor

puffin00

The KVM switch allows a single Keyboard, Video monitor, and Mouse to be switched between the 8 worker machines. Switching is either via a rocker switch on the front panel or via keyboard commands.

(We should have purchased the 16-port version to allow for expansion!)

## Network configuration

For simplicity all PCs are connected to the hub and the hub is connected to a port on the PPPL network. All PCs have published IP addresses (puffin00 thru puffin08) and are visible to the outside world. This is the most straightforward setup and requires no special configuration.

Other possible configurations would be to connect the second network card on the front-end machine, puffin00, to the PPPL network and then either

- have the front-end machine route packets between the worker machines and the outside world, or
- assign private IP numbers to the worker machines and have the front-end machine do IP masquerading.

We may investigate these configurations later.

## Software setup

The PCs were delivered with Linux pre-installed. Minimal configuration was required to make them usable for computations

- use DHCP to determine IP number at boot time;
- configure NIS, this allows user accounts and home directories to be shared with other Unix machines at PPPL;
- load AFS (Andrew File System) software to allow common software to be accessed (Fortran compilers, libraries, etc.) to be accessed in `/usr/local`. AFS is a caching file system, so accessing files via AFS typically has a minimal impact on the network.

## Configuration summary

Total time to configure the hardware and the software to provide a useful stable platform was about 3 days.

The task was made easier by prior experience with Linux machines (both at PPPL and at home). This meant

- we were familiar with setting up Linux machines
- we already had a wealth of software ready to use (via AFS)

## User environment

Users of puffin user their regular Unix accounts and home directories. Each machine has a 11 GB `/local` partition for user files. The front-end machine also has a 17 GB `/scratch` partition which is NFS-mounted all the cluster machines.

This presents a very familiar environment to users. In most cases a simple recompilation is all that is required to run jobs on puffin.

## Additional software

The machines came with Redhat Linux installed on them. This includes many compilers and tools. In addition we had a lot of software already installed on our systems from our previous usage of Linux

Free software installed:

- MPI, a parallel library (MPICH and LAM)
- PVM, an older parallel library

Commercial software installed:

- NAG, numerical library
- Fortran compilers (NAG and Portland Group)
- NCAR, graphics library
- AFS, caching file system

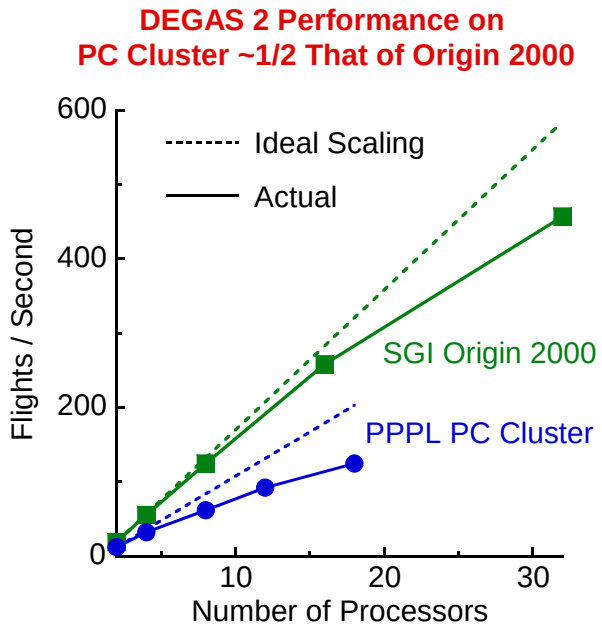
(We have several other commercial products. However these are not relevant to our operation of puffin.)

## Benchmarks

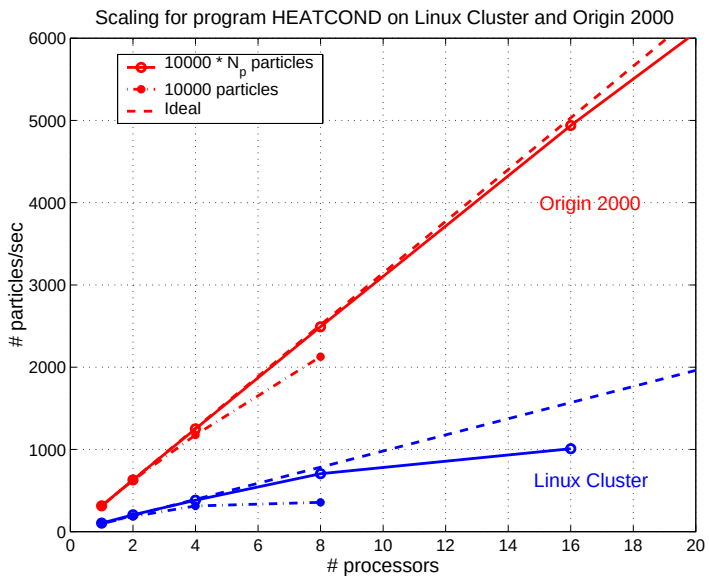
We have been able to run MPI codes on puffin with few or no modifications necessary. The single-CPU performance is comparable to that of the T3E (mcurie.nersc.gov) and about 3 times slower than the Princeton Origin 2000 (hecate.princeton.edu, 64 CPUs, 300 MHz). (These ratios are *very* problem dependent. Your mileage may vary.)

The slower interconnect means that as the number of processors is increased with a given problem size puffin saturates before specialized massively parallel computers.

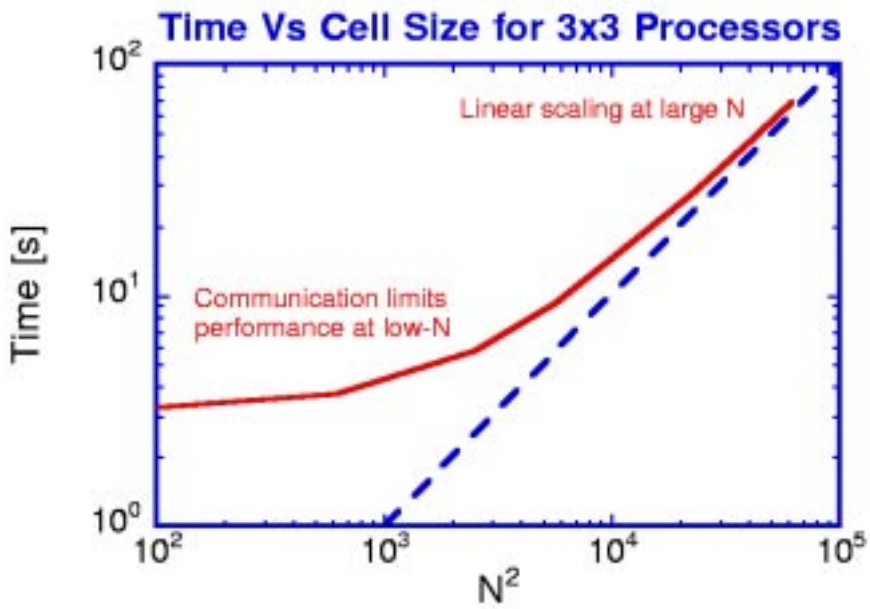
degas2 (data courtesy of Daren Stotler).



heatcond (data courtesy of Stephan Brunner)




wave (data courtesy of Raffi Nazikian)



## Access control

In order that puffin may be used for its intended purpose, running parallel plasma codes, we need to prevent access by casual users and by malicious intruders.

We implement this as follows:

- keep up-to-date on security patches
- limit the network daemons
- use restrictive TCP-wrappers
  - outside access only allowed to puffin00
  - access only permitted from PPPL computers
  - access only permitted with ssh
  - rsh access only permitted *within* the puffin cluster.

## System administration

In order for a PC cluster to be economical, it is essential that the administrative burden be minimal; in particular, we want to avoid operations which scale as $O(N)$ for an $N$-member cluster.

Our methodology is:

- do administrative work on one machine
- for minor updates, use rsync to update the files on the other machines
- for major updates, use rsync to clone the system partitions of the administrative machine to `/local` on the other machines, boot off a floppy disk, and copy the new system files to the system partitions.

## Conclusions

It is straightforward to set up a PC Cluster and this provides a very cost effective platform for some parallel codes.

Linux is now a first-class platform for scientific computing offering. Pretty much everything we can do on a Solaris or Digital machine we can also do on a Linux machine. Because Linux is open-source, scores of people world-wide are working to optimize the performance of Linux clusters.

## Future directions

We will monitor the software development efforts to support computing on Linux clusters [4]. At present, basic Linux PCs with MPI and PVM do a great job on many of our applications.

Gigabit networking is an attractive way of increasing the performance of a PC cluster. This would increase the bandwidth by a factor of 10 and should reduce the latency by a bigger factor. Possible options are Giganet [9] or Myrinet [10] which both offer bandwidths of 1.25 Gb. At present, this networking technology is considerably more expensive than 100 Mb Ethernet. We expect prices to drop over the next year. NERSC is working on an implementation of VIA [11] which will allow MPI to be optimized for this hardware.

We would like to investigate using an Alpha PC running Linux. Since the high speed network interfaces are fairly expensive, it might make sense to optimize with fewer more powerful machines. Issues are: cost/performance tradeoffs, availability of software for Alpha Linux.

## References


[1] http://w3.pppl.gov/puffin/
*THIS POSTER IS AVAILABLE HERE.*

[2] http://www.pppl.gov.

[3] http://www.linux.org

[4] http://www.beowulf.org

[5] http://www.nersc.gov/research/FTG/pcp/

[6] http://www.netlib.org/pvm3/

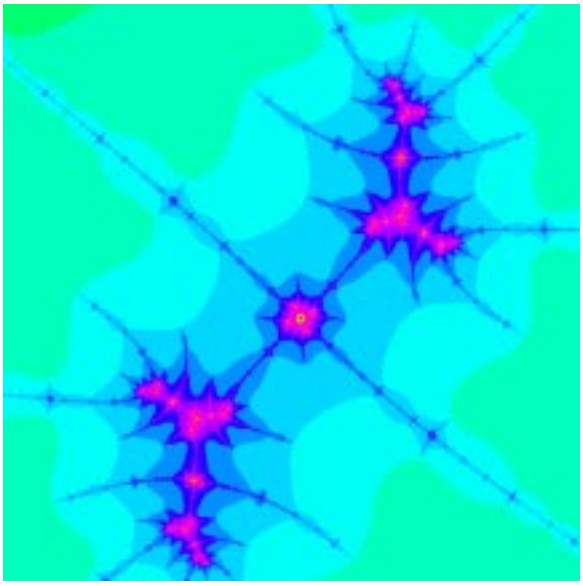
[7] http://www.mcs.anl.gov/mpi/

[8] http://www.swt.com

[9] http://www.giganet.com

[10] http://www.myri.com

[11] http://www.nersc.gov/research/FTG/via/


The Mandelbrot set computed in parallel on puffin