High-Performance Computing: Cluster Computing

Adrian F. Clark: alien@essex.ac.uk

2015–16

Introduction

SETI

Though it may seem strange, we're going to start with SETI, the Search for Extra-Terrestrial Intelligence

In 1959, Cocconi & Morrison published an article in *Nature* pointing out that microwaves could be used for inter-stellar communications

Frank Drake had reached the same conclusion and in 1960 aimed the 85-foot Green Bank radio telescope in the USA at nearby stars, tuned to the 21 cm (1.42 GHz) hydrogen line — he found nothing but static

Over the next 20 years or so, small-scale trials were carried out in Russia and the USA, and in 1988 a monitoring programme was funded in the USA; this funding was terminated a year later

However, SETI researchers are determined people and used private funding to continue monitoring — at Arecibo, the Allen Telescope Array, and Parkes Observatory

Processing the radio signals involves Fourier transformation and processing, quite slow in those days, and the SETI people did not have enough computers to do it

So they asked the public to help, and SETI@home was born

SETI@home

In the Windows world, SETI@home is a screensaver that downloads a chunk of SETI data over the Internet, processes it (and displays the result on the screen), then returns the result

Under Unix, where process switching is normally faster than under Windows, SETI@home can be run as a low-priority process without graphics, meaning it is scheduled whenever the processor has nothing else to do

These days, SETI@home is an application that makes use of the NSF-funded BOINC software; other citizen science tasks built on it include studying global warming and finding pulsars

BOINC

What characteristics must the data have to be processed using BOINC?

Condor (now HTCondor)

HTCondor is a management system for compute-intensive jobs – we used to run it here before we had to re-boot all our machines each night for Windows and anti-virus software updates

It is intended for batch (non-interactive) jobs, providing a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management

The idea is that one runs a Condor server on one's computer; this monitors things like mouse clicks and keystrokes and, when the machine is deemed to be idle, schedules a batch job on it

When the machine is used again, Condor checkpoints the batch job and can migrate it to a different machine

If the program requires data access, Condor can either migrate the data along with the program or arrange for data accesses to be transferred over the network to the originating machine

What constraints does the use of Condor place on jobs?

What type of jobs will work well with Condor and what poorly?

How would you go about implementing this kind of system yourself?

Building a cluster

Cycle-stealing approaches such as BOINC and Condor are good at making best use of resources — but you need to be patient

To achieve higher throughput — and especially if interaction or guaranteed performance are required — a dedicated resource is required

A cluster is basically a load of computers networked together — so what factors influence its design?

Beowulf

The computer that initiated interest in cluster-building was Beowulf, built by Thomas Sterling and Donald Becker at NASA in 1994, though the basic idea was older (and some of us were already doing this kind of thing)

This used a large number of cast-off computers (of a range of speeds), connected by a dedicated network and running Linux

Why a dedicated network?

Why Linux?

A typical modern cluster

Rack-mounted PCs, often multi-core, each having a large amount of memory and a local disk

Machines are connected via a Gigabit Ethernet switch inside a single cabinet

One machine has two network cards, and acts as the external access point

This runs hot, and so needs to be kept in an air-conditioned room

This is excellent for both compute-intensive and database search kinds of tasks

A ‘distributed cluster’

Buy a number of PCs that are low-end but ‘souped up’ by expanding memory and with a fast processor

Connect groups of about four machines together with small network hubs and put them on individual researchers’ desks

These groups are inter-connected by a switch, and one person’s desktop machine also acts as the ‘cluster controller’

PXE (boot-on-LAN) lets individual parts of the cluster be switched on or off remotely

Although this has less high-end performance, each person is able to use ‘their’ cluster on small tasks and others’ clusters (or all of them) can be used for larger tasks

We have found that this approach also works well when the task has some real-time aspect (*e.g.*, processing images or videos from a mobile robot)

An alternative approach

PC-class machines are fast but fairly power-hungry, and need things like graphics cards that are of little use in a cluster

There is a small but definite trend towards smaller, commodity processors that offer a good price–performance ratio

Can you offer suggestions as to suitable hardware?

It is fairly straightforward to build your own cluster

Figure 1:Raspberry pi

CSEE's brambles

A compute cluster of raspberry pis ("RPis") has become known as a *bramble*

We have a pair of brambles for you to use on this module

As an RPi has 100 Mb/s Ethernet and the largest 100 Mb/s Ethernet switch available at a reasonable cost has 48 ports, each bramble has 47 RPi compute nodes

Each cluster has a gateway PC that interfaces it to the campus network

Figure 2:Brambles — raspberry pi clusters

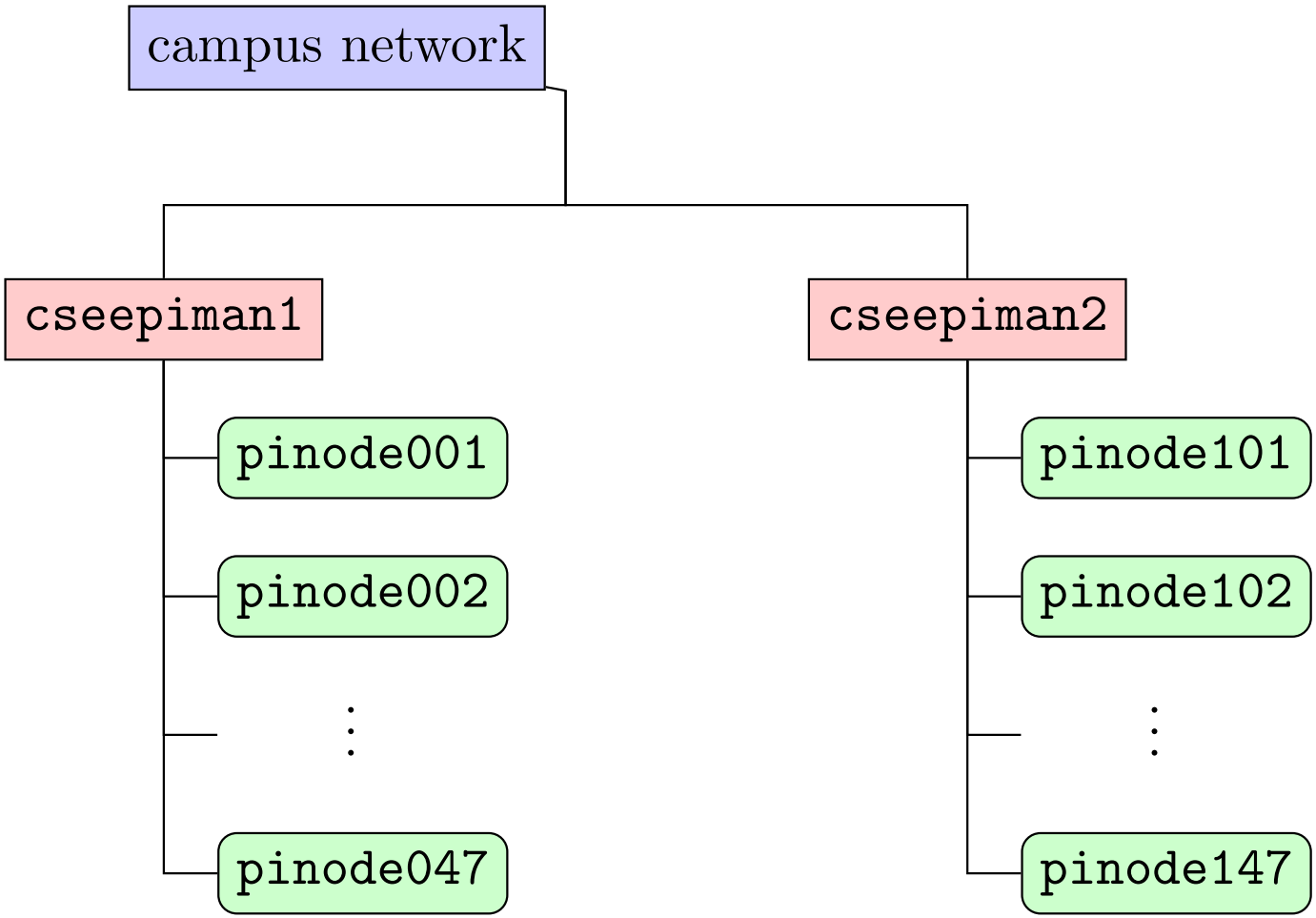

Figure 3:CSEE's brambles

Using the brambles

Use `ssh` to login on `cseepiman1` or `cseepiman2`

From there, you can `ssh` onto the `pinode`*nnn* nodes

You will need to be able to `ssh` from one `pinode`*nnn* node to another without typing your password, and you use

```
ssh-keygen -t rsa
```

on any computer to do this — when prompted for a passphrase, just hit the enter key

Ensure your public key is in the correct directory *on every machine* you wish to login on without typing your password, there is a script on the brambles called `~alien/distribute-key` that does much of the work for you

Using the brambles

`cseepiman1` and its `pinode`*nnn*s should be used for program development and debugging

`cseepiman2` and its `pinode`*nnn*s can also be used for program development most of the time

During 1400–1800 on weekdays, `cseepiman2` and its `pinode`*nnn*s can be booked with me in 30-minute slots, so that people can do their timing runs for the second assessment