Commercial Viability of 'Off-Peak' Distributed Computing at a Tertiary Institution – a Progress Report

Mark Caukill
Information Systems
Faculty of Humanities and Business
Universal College of Learning
Palmerston North, NZ
markc@ucol.ac.nz

Did you ever wonder what happens to the computers at work once everyone goes home? Not much – except for a few individuals helping to look for extraterrestrial signals! Ever think about what a waste it is to have all those computers doing nothing? All that money - in most cases - doing nothing! So, why not use this wasted resource? Working together, these many PC's can make a very powerful computer – potentially a super computer!

This large processing capability is accomplished by distributed computing. Distributed computing is basically any computing that involves multiple computers across a network, where each computer (or node) has a role in solving computationally intensive problems.

UCOL - Palmerston North has many hundreds of computers throughout its campus. Each night these computers sit idle or are off. There is considerable potential to exploit these computers during the time when they are not in use to earn an income through the selling of computing power; or at the very least partner with researchers or industry for a symbiotic relationship.

Distributed computing is not new to academia. Academic institutions are the fertile ground from whence they sprung. But what is not common, is the use of academia's core business (i.e. education) computers being used by industry for a profit. This is where this project is aimed.

As the project has been going for a while, but is far from over, a progress report is in order.

This paper will give a snapshot of where the project is at by:

> Explaining the basics of distributed computing

> Showing the initial aim of the project

> Discussing project directions that have been successful and not so successful

> Looking ahead at "where to from here"

1. THE NUMBER EIGHT WIRE BASICS OF DISTRIBUTED COMPUTING

Trying to delineate a single definition of distributed computing is akin to asking how long the proverbial string is. If one does a Google™ (www.google.com) search using keywords of "define:Distributed Computing", at least two pages of descriptions result – none of which exactly agree. Add into the mix other labels like: cluster computing, grid computing, parallel computing, and the definitions become even more varied and contrary. It comes down to who is formulating the definition, their perspective on the subject and how detailed or pedantic they choose to be for their target audience.

This section on the basics of distributed computing is a melange of definitions put together as the author's understanding on the subject. It is not, nor does it try to be a definitive, precise summary, but a viewpoint from which others may become familiar with a rather complex and daunting subject.

Simplistically, distributed computing can be characterised as a compute task that is spread amongst multiple computers within a network.

Within this loose framework there is a set of jobs and devices that need identifying. Generally, a management server, or master node, acts to distribute the necessary raw data out to the nodes on the network. It may also act as a centralised storage device for both raw data and 'solution' data. So the task puzzle is broken up and sent out to the nodes where the application processes the data and the computed solution data is sent back. A depiction of this process can be seen in Figure 1.

The next step in understanding distributed computing is to consider what is effectively the geographic relationship between the two distinctly different types of distributed approaches: Grid Computing and Cluster Computing.

Grid Computing is distributed computing at a typically large geographic scale, linking a number of clusters over a telecommunications infrastructure. In some cases this is the Internet but more often than

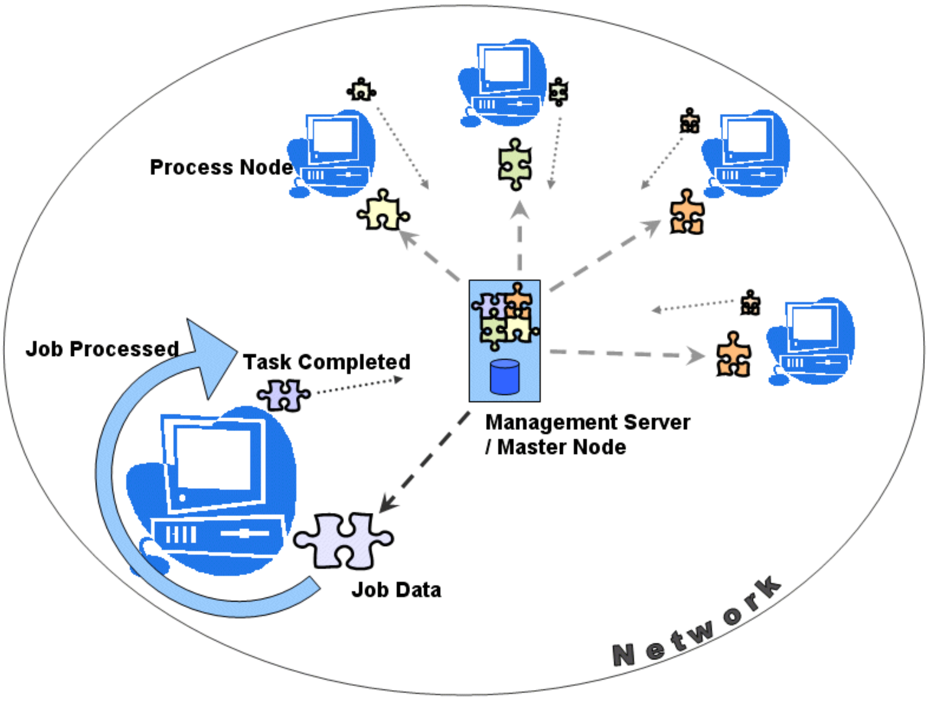

Figure 1: Distributed Computing Simplified (adapted from http://www.pcmag.com/image_popup/0,3003,s=0&iid=15742,00.asp)

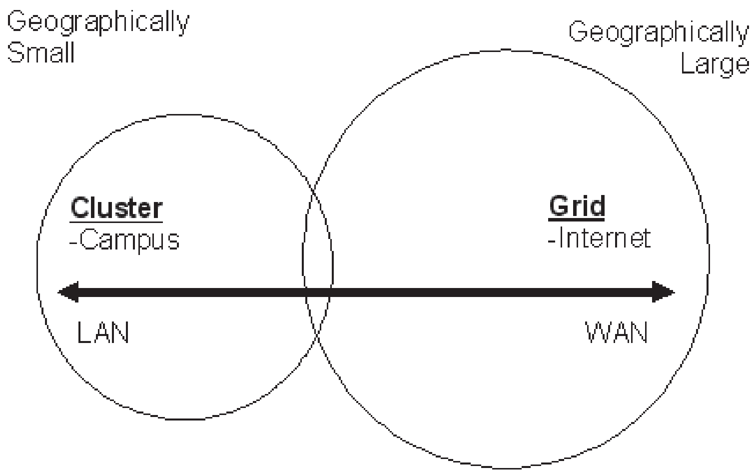

Figure 2: Geographical differences in distributed computing.

not is over a dedicated long distance network (WAN).

At the campus sized, Local Area Network (LAN) end of the scale, distributed computing tends to be called Cluster Computing. Figure 2 gives an overview.

As the project this paper is about focuses on clusters, they will be considered in more detail.

Again, in a 'one-dimensional' view of the issue, clusters come in two flavours: Purpose Built Clusters and Desktop Clusters.

Purpose built clusters are just that – built for the sole purpose of creating a powerful compute platform; in many cases, for a specific task. A purpose built cluster will generally have a fast, high bandwidth interconnect for efficient movement of data. Each node also tends to be homogeneous in terms of hardware configuration and operating system, and have a large amount of memory to keep I/O operations at a minimum.

Desktop clusters are where a number of existing desktop computers on a LAN infrastructure, are used together to perform a compute task. In this format, a cluster is often working on the compute task using otherwise unused processor cycles (called 'cycle scavenging). The business environment that a desktop cluster exists within, means that the nodes

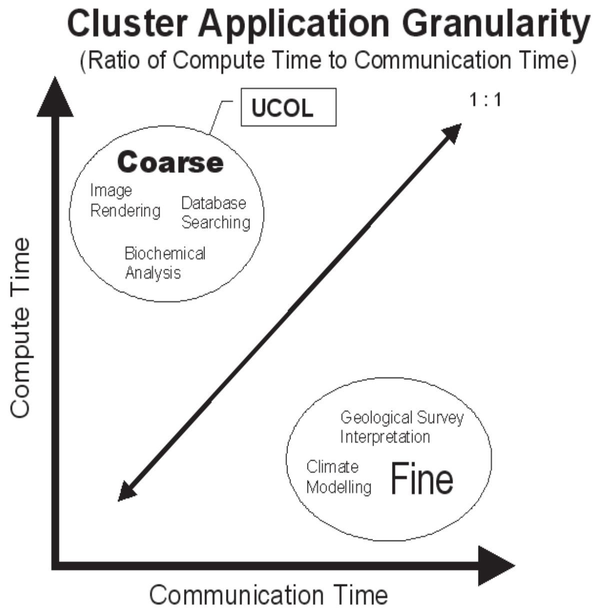

Figure 3: Application granularity graph.

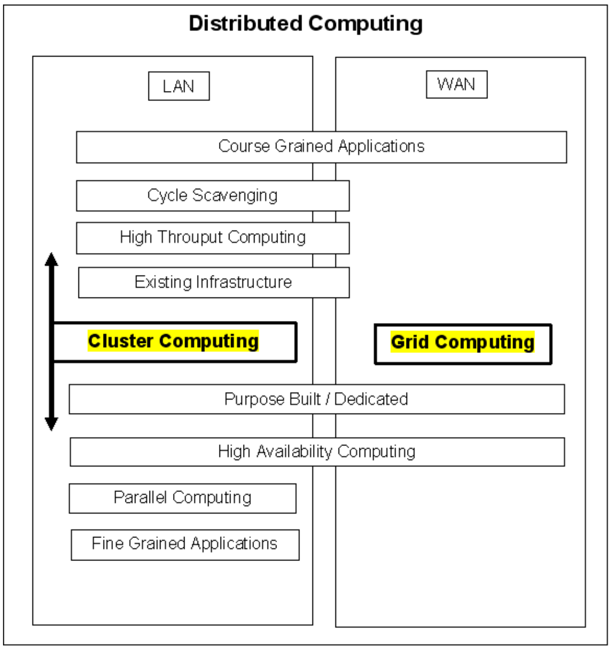

Figure 4: An extreme graphical summary of the interrelationships between the various issues discussed above. Note: some concepts have not been covered.

are heterogeneous, varying greatly in hardware configurations and operating systems. The network that these desktops are connected with is not an optimum interconnect for distributed computing, as it has most likely been built to support a business model.

It is now appropriate to consider the types of compute tasks that are suitable to run on the various distributed platforms.

One does not normally build a cluster and then search for an application to run on it, as the cluster cannot be optimally suited to every application. In a purpose built cluster, the compute task decides what algorithm is written and the specific format of hardware. Unfortunately, for a desktop style cluster, this is a luxury that is not available. The compute tasks must be chosen to fit the existing system within the limitations already present. This includes operating system or systems (which will primarily influence the algorithm used), memory, network interconnect, etc. The cluster management software is the only real choice for this type of system.

The granularity of a compute task will determine which type of distributed system should be used for the purpose. Granularity is the ratio of compute time to communication time[1] (Intel Corporation [Intel], 2002). If the compute time to communication time ratio is high, then the task is considered to be a coarse grained application. These applications spend more time computing than communicating and in this situation one can get away with an inefficient (ie: slow business focussed) network interconnect. Where the ratio is low, the task is a fine grained application. A highly optimised, fast, wide bandwidth network interconnect is necessary in this situation as the compute task algorithm spends much of its time communicating with other nodes in the cluster.

Examples of fine grained applications are climate modelling, and virtual nuclear weapons testing. Any application where the interaction of the parts plays an important role in the outcome is effectively fine grained. In the coarse grained area are jobs that can be broken down into parts that are only a smaller representation of the whole. Examples include image rendering and data-mining. A graphical representation of granularity can be seen in Figure 3.

Figure 4 summarises the preceding abridged discussion of distributed computing.

For the author, this very generic picture has developed over time and is an attempt to show the interrelationship of the various definitions.

2. THE INITIAL AIM OF THE PROJECT

It was anticipated that the project would be in three parts:

2.1 Research and analysis.

Several factors must be addressed at this initial stage.

- What are the various types of distributed computing and how do they work.
- Which model fits best with UCOL's hardware infrastructure
- Is there a market (globally or nationally) that would pay for a secure distributed processing platform. If so – how much.
- Is it possible given the existing UCOL PC equipment? If so, how and what issues need addressing? An initial list might start with:
  - o Security
  - o Operating System on node PC's
  - o Distributed Computing management software (client & server)
  - o Operations and Maintenance (human resources, etc)
  - o Extra hardware necessary
- Is it possible given the existing computer maintenance and support service level agreement? Could this type of support even be an important factor in its (distributed computing grid) success?

2.2 Proof of concept.

A small-scale trial (or multiple simultaneous trials) if research and analysis indicate it is feasible.

2.3 Full implementation

Should parts 1 and 2 be successful. This would most likely be implemented on a rolling basis to avoid the project difficulties, should they arise, getting 'out of hand.'

From the outset it was intended for any cluster within UCOL to only operate initially during the non-core business hours; approximately 10pm to 7am each night.

The project was effectively split into two areas of development. The Apple Computer Macintosh's within the multimedia environs as one cluster, with the x86 computers across the rest of the campus as a second larger cluster.

The Mac cluster (renderfarm) was to be treated as a multimedia based cluster, nominally for rendering images and movies. There was no expectation of income from this cluster *per se* but a relationship was sought from players in the multimedia industry. This relationship would be potentially beneficial to UCOL as it would provide an important link with industry, guest speakers, student scholarships, etc. The advantage for the cluster client would be in the use of the computing power without the associated capital and maintenance costs.

The larger x86 based cluster is considered to be the bigger technical and business challenge, with the potential for reasonable financial returns. A multimedia approach was initially taken with this computer base but this has since turned towards more of a 'technical' problem solving focus such as biochemical analysis. This is where the author believes the cluster's best chances of success are.

3. PROJECT SUCCESSES AND DISAPPOINTMENTS

Ignorance is sometimes a wonderful thing. When one does not know what cannot be done, the possibilities are limitless. It can also lead to overly grandiose expectations! This has been the case with this project. To date barriers have outnumbered bridges.

3.1 The Mac Renderfarm

Over a period of months during 2003 and 2004, a conversation was held with Rising Sun Pictures (RSP) (www.rsp.com.au) in Australia. They are a visual effects vendor credited with, amongst other notable achievements, the shuttle landing sequence in the movie "The Core" (Paramount Pictures. 2002). RSP choose not to make use of the Mac renderfarm for the reasons discussed in the following paragraphs. This combination of factors was specific to them but are also reasons that this idea would likely be unsuccessful with other multimedia/visual effects companies.

Render licenses for 3rd party software are extremely expensive. Making them work on a remote network would be expensive and demanding for both 3D and 2D rendering. Essentially every processor used for rendering would require the purchase of a license for software. This is not a problem for some packages but 'serious' commercial software costs around \$2k-\$5k per processor[2]. This is prohibitively expensive in most cases.

There are practical issues in making a network work, and the creation of filesystems and server setups are nontrivial and involve proprietary knowledge. This would require considerable resources from both ends. This was one of the areas where RSP was not able to arrive at a cost / benefit equation that worked for them.

Movie companies are extremely paranoid about security. There are two basic security problems – one technical, the other socio-political. The technical problem is in ensuring that those who shouldn't have access to the images doesn't (and therefore can't distribute them). The socio-political problem is that no matter how secure the process is, convincing the studios that it is all 'OK' would not be easy. In a process where the studio 'sell' is very difficult, they did not want to introduce more obstacles.

Bandwidth is another problematic area. Rendering for motion pictures involves terabytes of data. The initially considered option of increasing the 'pipeline' into the renderfarm was dismissed due to the extreme costs involved in doing so. The second option of achieving a higher bandwidth into the building involves the concept behind the networking colloquialism of: "Don't underestimate the bandwidth of a station wagon full of data tapes.[3]" If the urgency of turnaround was low then a Network Access Storage box (NAS), or even a data tape, could be couriered over (~one-two day transit), plugged in, raw data off, finished data on over the period of processing, then couriered back (~one-two day transit). Unfortunately, with rendering, part of the issue is quick access to the results is critical because things always go wrong. So the concept of turning tapes around was not an option. Also, projects more often than not on tight turnaround, so introducing a delay in the turnaround on rendering would create problems.

3.2 The x86 Cluster

In early 2003 communications were started with Weta Digital Ltd (Weta) with the idea of assisting the company with rendering on "Return of the King" (New Line Cinema. 2004.). This concept was quickly dashed as the Mac renderfarm was tiny by comparison, and the x86 computers on the UCOL network were not up to the required specifications[4]. This has however led to an excellent relationship between UCOL and Weta with ideas and support flowing both ways.

To date there has not been any trials on the core business computers. In general, it has been difficult to advance the project beyond the theory stage to date. Working in isolation has also been a contributing factor for the slow advancement of the project.

4. LOOKING AHEAD

Two cluster solutions, Condor (http://www.cs.wisc.edu/condor) and Thin-OSCAR (http://thin-oscar.ccs.usherbrooke.ca), have been chosen to move the project forward. Condor is suited to the UCOL computer environment from the standpoint of requiring little or no alteration to the existing computer images. This option is not really considered 'real clustering' by the hard core distributed computing community but is used widely for effective high throughput computing. Thin-OSCAR fits well within the accepted definition of a clustering solution but requires a paradigm shift in the way UCOL's computers are operated. A separate operating system would be required and as such the computers would be dual-boot out of necessity. This severely complicates implementation in the business environment but may be worth the trouble as it could potentially be a better, more effective solution.

Other potential project targets include:

- Providing Weta with processing backup
- An open multimedia render farm for up and coming artists
- Academic partnership
- Partnership with a commercial entity initially existing to get the project on its feet but eventually leading to a commercial income.

REFERENCES

1. Intel Corporation. (2002). White Paper: Building high-performance computing clusters with Intel architecture, Part 1: Planning HPC clusters (p7).
2. Low cost packages considered were the Lightwave renderer (http://www.newtek.com), and the 3DSMax native renderer (http://www.discreet.com). Bigger commercial packages include such as Renderman (https://renderman.pixar.com) and Mental Ray (http://www.mentalimages.com) and the Shake renderer (http://www.rfx.com).

3. Don't laugh! In March 2004, an experiment was done with pigeons carrying memory modules. The effective bandwidth was better than the available commercial DSL service. See http://www.notes.co.il/benbasat/5240.asp
4. The servers used on the "Return of the King" were IBM dual 2.0 & 2.4GHz Pentium IV Xeons with approximately 3Gigabytes of RAM per processor. The interconnect network was Gigabit Ethernet to each server with a 10Gigabit Ethernet backbone.