A Review of Applications of Cluster Computing

Lucas Hope and Edmund Lam

School of Computer Science and Software Engineering
Monash University
Clayton Campus, Melbourne, Australia
Email: {lhope,epl}@csse.monash.edu.au

Abstract

There are many applications which can benefit from parallelisation. Employing clusters of computers provides a method to utilise commodity components, minimising cost and and maximising longevity of the individual parts.

We present three popular applications; web-servers, ray tracing algorithms and data mining, and report how cluster technology can be applied.

1 Introduction

Clusters of computers have become more popular in recent years as the power of commodity processors has approached that of workstation processors. Yet at the same time, the increased usage of computer technologies has meant that costs of processors as well as other computer sub-systems (such as motherboard, memory, hard disks, network cards) has benefited from the scale-of-economy. The standardisation of network hardware and software protocols has also improved the viability of cluster computing.

Since the invention of computers, there has been a requirement for greater processing power. Solving differential equations or running Monte Carlo simulations increases the hunger for processing power. An appetite that that increases yet again when satisfied. A cluster designed to meet these needs is a "high performance" cluster.

However, not all problems are amenable to cluster solutions. Depending on the communication costs of the algorithm used to solve the problem, employing a cluster may bring zero or negative benefits. On the other hand, there is a wide class of problems which can be cheaply and effectively solved using clusters. Parametric-modeling is one such example, as well as "embarrassingly parallel" problems such as brute-force cracking of encryption keys.

Another reason why clusters have become more popular recently is society's growing dependence on computer systems. As we grow more dependent, the cost of any failures increases dramatically. Combined with computer systems' reputation for poor reliability, (in contrasted to equipment such as pen and paper) has meant that there has been a great impetus to develop and deploy cluster computer solutions to ensure that computer systems stay up twenty-four hours a day. A cluster designed to meet these needs is a "high availability" cluster.

Another feature which cluster systems offer, which has not been traditionally found in computer systems, is gradual degradation, whereby a system fails slowly and with prior warning so that remedial action can be taken in time to avoid a catastrophe.

This is often enhanced by placing the nodes of the clusters in separate physical locations. This means when artificial or natural disasters — such as power failure, earthquake, fire, flood and riots — occurs, the computer systems can continue to operate. Other measures can be taken to reduce the computer system's points of failure such that (ideally), no single component failure (hardware/software/human) can bring down the system (ie eliminate "single-point-of-failures").

Within the limitations of the cluster framework (ie some problems will not be more efficient even if more processing power were available), there are other benefits of clustering. For example, clusters offer greater scalability than traditional SMP computers. With additional hardware, the performance of the cluster can increase substantially. A cluster of machines is also more attuned to the requirements of the user as additional computing power, which is not yet required, can be deferred for purchase at a later date. In contrast with individual workstations a user is forced to purchase a computer substantially more powerful than their current requirements to avoid obsolescence a few years down the track. Finally, heterogeneous clusters (those which contains multiple architectures) can be built. For example, an initial cluster was built using Alpha processors. Recently x86 processors have become more viable and the newer machines have x86 processors instead. This means that the user is not necessarily locked into a proprietary solution.

List of works

- University of Bristol (Computer Graphics Group) - Parallel ray tracing
  `<http://www.cs.bris.ac.uk/Research/Graphics/>`

- Kwangju Institute of Science and Technology (Information Systems Group) - Parallel ray tracing
  `<http://parallel.kjist.ac.kr/projects.htm>`

- Ray tracing FAQ
  This document contains introductory material about ray tracing

<http://www.faqs.org/faqs/graphics/raytrace-faq/part1/>

- PAPIA - PArallel Protein Information Analysis
  a site which uses data mining techniques to analyse protein
  <www.rwcp.or.jp/papia>

- Parallel Data Mining - A short piece on data mining
  <http://isi-cnr.deis.unical.it:1080/~talia/PDM.html>

- PADMA - PArallel Data Mining Agents
  This uses clusters to data mine unstructured text.
  <http://www.eecs.wsu.edu/~hillol/padma.html>

- David Skillicorn's parallel data mining page
  Summary of types of data mining.
  <http://www.qucis.queensu.ca/home/skill/cascon.html>

- Google - Web-search engine
  Utilises a four thousand computer cluster
  <http://www.google.com>
  <http://www.newsalert.com/bin/story?StoryId=Cotm8WbKbyte2mtu>

- Eddieware - High performance, high availability web server
  A flexible multi-platform commercial-grade web server with load balancing.
  It offers high performance and high availability.
  <http://www.eddieware.org>

- Linux kHTTPd - Kernel-based web server
  A very high performance web server for static pages
  <http://www.kernel.org>
  <http://kernelnotes.org/lnxlists/linux-kernel/lk_9906_01/msg00856.html>

2 Web Serving

When the Internet exploded upon the public consciousness, the World Wide Web became synonymous with the Internet. Invented in 1989, the hypertext markup language and the associated hypertext transport protocol has brought about communication which is vastly different from the traditional one-dimensional novel.

As more members of our society embrace the web, the characteristics of network traffic (eg fractal traffic with long-range dependence) requires that popular web sites have plenty of excess capacity to spare in order to serve waiting customers. In fact, this phenomena has been named the ‘slashdot-effect’ <http://ssadler.phy.bnl.gov/adler/SDE/SlashDotEffect.html>.

Roughly speaking, this effect occurs when an otherwise unpopular site is linked by a popular site. As web surfers follow the link, they overwhelm the web server whose web-master could not have predicted the peak in network traffic in advance. The web-surfers in effect perform a Distributed Denial of Service (DDoS) attack, although there is the critical distinction that the attack is not malicious and is truly distributed. And since the people ‘attacking’ are valid customers the only alternatives are to close the web site or increase the capacity of the web server and/or the network throughput.

Of course, a web site could also be attacked by a malicious DDoS attack. This is where a few crackers utilising many web clients overwhelm the web server by superfluous (and possibly invalid) requests. Sometimes it is possible to defend or at least reduce the impact of such an attacks, but increasing the network throughput and ensuring the web server has enough processing power is also often an effective defense.

Finally, as the Internet has become more popular, our society has grown dependent on its existence. Although computer technologies are relatively unreliable, companies need to ensure that their web site, which is after all their public face/facade, has an excellent uptime. This is due to various reasons. For example, if Altavista could not be accessed for a period of time, they would lose valuable advertising revenue. If Microsoft was taken down, their public image becomes even more tarnished and their reputation sullen in the public eye. If Coles Myer’s site failed, their customers and suppliers may decide to go elsewhere, causing the loss of valuable business. In the current climate of ‘Internet-mania’, building a ‘critical-mass’ of customers is vital and losing customers is suicidal.

Web serving is a task that varies in complexity. It can be as simple as responding to requests for static pages. A task which is readily parallelisable, though the bottleneck often isn’t the processor, but the network throughput. In such cases, there are specific software whose specialty is to only serve static pages (Linux kernel patch kHTTPd <http://www.kernel.org> or <http://kernelnotes.org/lnxlists/linux-kernel/lk_9906_01/msg00856.html>).

A task which requires more processor usage is the service of dynamic web pages, whose content changes either very frequently, or change depending on the user environment (such as the browser used). These pages often take the form of a web page being compiled from a database of content and formatting. It might also be created using Perl <http://www.perl.org>, PHP <http://www.php.com>

or another type of CGI-scripts. All of which will take additional system resources such as processor cycles as well as memory.

Finally, there are some web pages which are interfaces to web-applications. Examples include database searches like search engines. Not only does the application read from the database, it may also update the database creating problems of coherency, race-conditions and the like. Such applications may also require vast amounts of processor power (such as Babelfish <http://world.altavista.com>).

Clearly, if a web server requires vast amounts of processing power in order to perform its duties and the problem is readily parallelisable, then a cluster solution often makes sense. In other situations whereby the processing power isn't the bottleneck or when a problem does not have a corresponding parallel algorithm, a cluster cannot be used to bring about greater performance.

Nonetheless, additional computers forming a cluster can still be used to provide redundancy as well as load balancing. The web site <http://www.msn.com> has seven mirrors ensuring that if one server crashes, another is able to take its place. This means that hopefully, users will never notice any downtime with the web site. In another example, <http://www.us.kernel.org> has fourteen web sites to cater for the legions of Linux users downloading the latest kernel or patch.

An example of a cluster-capable web server is Eddie <http://www.eddieware.org> [1]. It is a web server designed to be both high performance as well as high availability. It does this by having a scalable and flexible design combined with a load balancer. Both the performance and availability comes from having multiple servers at different locations. There is also support for automatic detection of server failure and the subsequent rerouting of the failed server's task onto another server.

The topic of web serving encompasses a wide area. Depending on the material being served, a cluster solution could be a vital component or simply a waste of good hardware. Certainly for static pages it seems that greater network bandwidth is required before the limits of current-day processors are reached. However, for web sites such as Google it is only the existence of clusters that this web-site is possible. It is clear that exciting work is done by the Eddie project which aims to improve current web-serving technologies in preparation for the continued exponential increase in traffic that has occurred in the past decade.

3 Ray Tracing

Ray tracing is the most commonly used method to render photo-realistic images. A three dimensional world is described using either a CAD package or a programming language designed for this purpose. The world contains textured objects, each with a certain reflective value, and light sources. Once this world has been fed into a ray tracer, the paths of the individual light rays emanating from their sources are simulated, resulting in a highly realistic output. This process can be further complicated when such effects as diffuse inter-reflection calculation (simulation of light diffusion) are added [2].

Ray tracing is a computationally demanding task which may consume hours or even days of processing time. The time taken is determined by world size, image size, geometry, reflectiveness of the objects rendered, and the desired quality of the rendering.

There are a number of reasons why parallel processing is beneficial with respect to ray tracing, and a number of issues which increase the difficulty of effectively utilising parallelism. Parallel processing is beneficial as it can reduce the time taken to render images. More importantly, images which need more memory than an individual processor has available can be divided between a number of individual processors, enabling the rendering of complex images which would be otherwise be impossible. Major issues include the non-trivial task of segmenting the jobs between the processing nodes and minimising the communication costs.

The two common methods of segmentation are demand driven techniques and data parallel techniques. Demand driven methods subdivide the rendering task into different sections of the image space (the two dimensional image which is seen on the screen). Each section represents a task, with tasks completed separately by the different processors. When a task is completed a processor requests a new one. This method can produce good speedups if the entire scene can be stored on each processor. However, this can not always be assumed.

If data must be distributed over all the processors, data communication, which can substantially decrease the speed-up provided by parallelism, will be necessary. Although data caching can be used, a lack of coherency between the objects can all but destroy the efficiency of this method.

Data parallel methods generally segment object space (the information representing the actual objects to be rendered). The geometry is distributed over the processors and individual ray traces are passed to relevant processors. If a ray does not intersect any objects on a given processor it is passed to a processor which operates on the neighbouring data. This process is repeated until the ray either intersects an object or leaves the environment. The data parallel methods allow for large scenes to be rendered. However, processors which include the view point and processors which include light sources will have higher loads. Also, many rays may cross between processors, resulting in a high communication cost.

Reinhard et al. [3] suggest combining the above approaches in two ways. They report that ray traversal is suited to demand driven methods with ray intersection comprising the data parallel component. However, this does not adequately balance the load. They advocate instead the use of coherency of the object data to subdivide tasks into demand driven and data parallel components. Coherent tasks require a small amount of data, and yet are complex enough to warrant demand driven scheduling. The rest of the tasks remain data parallel.

Ray tracing is an application which benefits much from the parallelism provided by cluster computing. Asides from the standard goal of speed-up, clusters can be used to distribute data which can not fit on a single machine. Scheduling of tasks and balancing the workload of processors is non-trivial, so carefully constructed algorithms are needed to utilise the power of clusters. The cost of communication, one of the major problems faced when attempting cluster computing, is very much a problem in this case, as there is no easy way to segment the data and minimise communication.

4 Data Mining

Data mining describes the use of computers to analyse large amounts of data with the intent of finding relationships and knowledge that are of interest and are too difficult to find manually. Data mining is of interest in many research areas such as banking, protein analysis and genetics [4].

Skillicorn [5] lists three main approaches to data mining. These are verification, supervised techniques and unsupervised techniques.

- **Verification**
  This is the situation where the analyst knows what he is seeking. The data is explored via testing of different hypotheses and the system responds by either verifying or partially verifying the given hypothesis. From this data, the hypothesis can be refined and the process repeated. Once the hypothesis has been sufficiently refined, it may be accepted or rejected depending on its validity.

- **Supervised Learning**
  A supervised technique is one where the system is 'trained' on a set of data which has some interesting property. This approach can then be categorised into two different areas, transparent searches and opaque searches.

  - *The opaque approach*
    The opaque approach generates a black box which can identify data with similar properties as the training data. An example of this is neural network training. Opaque supervised learning can be used for purposes such as bulk mailing and retail marketing.

- *The transparent approach*
  Transparent approaches require training the system on the data, so knowledge about common properties existing within the data can be discovered. This is labeled transparent because black box techniques such as neural networks are not adequate. Transparent techniques are also harder to implement than opaque methods. An example of this technique is decision trees learning (eg MML induction of decision trees). Some uses are automated mortgage approval and medical billing.

- **Unsupervised Learning**
  The goal of unsupervised learning is, given a set of data, to discover likely hypotheses which are of interest. The data is generally split into different clusters, based on common properties. This is perhaps the most useful technique as it can detect previously undiscovered properties in the data. Some examples of this are association rules and inductive logic programming. Neural networks can be either supervised or unsupervised.

Parallel processing is an important tool for data mining, a computationally expensive process. Data mining requires processing of massive amounts of data. There is a demand for greater data complexity, as well capabilities to handle this more complicated data. Speed of processing is of paramount importance to commercial interests, as this can provide them with a competitive edge. Also, some datasets are already distributed across media such as the Internet and so employing data parallelism can also decrease the time taken to gather data.

There are, however, some problems inherent in data mining which parallel processing can not fix. One of the key characteristics of data mining is that the majority of the data set is processed. Also, sampling of data, for example a comparatively small random sample, may produce knowledge which is just as useful.

A major problem with sequential data mining techniques is bottlenecks in the data access. If the data is distributed this may be improved, but there is no reason why this isn't just as much a problem for parallel systems. The cost of communication, which is a problem with parallel computing in general, is no less a problem with parallel data mining.

Kargupta et al.[6] have developed a system called PArallel Data Mining Agents (PADMA). This system runs on clusters of Sun SPARC and Linux systems. The main components of the system are the data mining agents, the facilitator which coordinates them and a web-based user interface, which implements a single system image.

The agents are responsible for analysing the data and extracting useful information. These agents are the parallel component of the system, and are currently specialised to perform text classification. The agents share their information through the facilitator which also provides feedback to the user.

It is unclear as to whether cluster computing provides benefits to data mining. Data access, the major bottleneck for for sequential data mining, can be overcome without the use of a cluster. However, if communication can be minimised, parallelism will increase productivity in data mining systems. Depending on what information is desired from the raw data, as well as the algorithms used, clusters could potentially be useful.

5 Conclusions

As a result of falling prices for computer hardware for the past few decades, today's personal computers are more powerful than the biggest supercomputer of yesteryear. However, the demand for more processing power continues unabated. The commodisation of network equipment has increased the viability of communicating between separate computers by reducing latency and increasing throughput.

Although cluster computers are not a panacea for all of society's computing needs, there exists a wide range of problems for which cluster are the ideal solution. As shown in this paper, there already exists a wide variety of places (Google, PAPIA) who already depend on clusters to achieve their goals. Work which would otherwise be infeasible due to the prohibitive costs.

Depending on the material a server is web serving, a cluster solution may or may not provide performance advantages. Nonetheless, increased availability can be gained by using clusters. Ray tracing has gained much from clusters. As an example, the rendering of certain scenes in the movie Titanic were performed on a cluster of Linux-based machine. As such, it is clear that some in the entertainment industry see the potential benefits of utilising clusters to create movie scenes artificially. Data mining has not yet reached its full potential and it is not clear whether clusters can provide adequate speed-up for their cost.

References

[1] Michael Rumsewicz. Web servers for commercial environments : The imperatives and the solution. `<http://www.eddieware.org/txt/wp_1_0.ps.gz>`.

[2] Markus Kniebes (Maintainer). Graphics/raytrace faq. `<http://www.faqs.org/faqs/graphics/raytrace-faq>`, 29 April 2000.

[3] Erik Reinhard, Alan Chalmers, and Frederik W Jansen. Hybrid scheduling for parallel rendering using coherent ray tasks. In *1999 IEEE Parallel Visualization and Graphics Symposium*, pages 21–28. ACM SIGGRAPH, October 1999.

[4] Domenico Talia. Parallel data mining. <http://http://isi-cnr.deis.unical.it:1080/˜talia/PDM.html>.

[5] David Skillicorn. Parallel data mining. <http://www.qucis.queensu.ca/home/skill/cascon.html>.

[6] Hillol Kargupta, Ilker Hamzaoglu, and Brian Stafford. Parallel data mining agents. <http://www.eecs.wsu.edu/˜hillol/padma.html>.