High Performance Computing using a Parallella Board Cluster

PROJECT PROPOSAL

March 24, 2015

Michael Johan Kruger
Rhodes University
Computer Science Department
g12k5549@campus.ru.ac.za

Principle Investigator

Michael Johan Kruger
3 st Aidens
Grahamstown, 6139
079 982 8583
g12k5549@campus.ru.ac.za
Supervised by: Dr. Karen Bradshaw

Project Title

The proposed title of the project is:
High Performance Computing using a Parallella board Cluster

Problem Statement

Is it possible to build a high performance computer using a cluster of Parallella boards. The release of the Parallella board provides a cheap and energy efficient computer that requires minimal setup. Cluster computing is becoming more and more popular as it is a low cost way to create high performance computers. Using multiple separate computers over a network it is possible for them to work together as one system to solve problems [5].

Objectives

The proposed objectives of this project are as follows:

- Build a high performance computer using multiple Parallella boards connected over a network
- install Operating systems and software to set up the cluster
- Compare performance of the Parallella cluster with other similarly priced systems
- Discover the limitations of the Parallella cluster

Background

High Performance Computing

High Performance Computing (HPC) is the term for very fast systems aimed at processing large amounts of information quickly. High performance computers are made up of multiple processors as the speed of a single processor has reached its limit due to physics [6]. HPC is most cheaply obtained using cluster computing, as most places needing large

amounts of processing power have multiple computers readily available [5]. HPC is used to run high cost simulations that would be to expensive or difficult to do physically, these require large amounts of computational power to be completed in a timely manner [11] [10] so more powerful machines are always being built. The High performance computing has evolved over time and the amount of cores in 1 computer is in the millions with performance in the Peta FLOPS (10^{15} floating-point operations per second) [3].

Parallel and Concurrent Computing

Due to the current size and speed constraints on single core processors it is found to be more efficient and faster to run multiple slower cores. Algorithms need to take into account ways in which to split the work load evenly between multiple processors if they want to get faster speeds in this architecture. [6]. Many compilers have special flags so that they will try optimise for parallel computation [9]. But this is a minor boost in speed when compared with an algorithm that splits the work up. According to David Geer [6] to take advantage of having multiple cores programs need to be rewritten so that they may run on multiple threads, each thread may then be assigned to a processor.

Cluster Computing

With the need for large amounts of processing power, ways of creating super computers cheaply have appeared. Clusters of computers connected on a network can be purposed to work together as a super computer. With the increased speed and decreased latency of the Internet, it is possible to create a cluster using computers from all over the world, this has lead to programs and applications that allow a computer to connect to a pool of other computers and add its processing power to the computation. There are factors limiting the effectiveness of cluster computing such as: Building a switch to keep up with the speed of a single core processor, creating compilers that make good use of multiple processors. Clusters communicate with each other using a message passing interface (MPI) which provides a thread safe application programming interface (API) that allows the work to be effectively delegated to multiple computers on the network [**?**]. The MPI allows each individual processor to communicate with the others allowing them to pass information to each other [**?**].

Parallella board

The Parallella board is an "affordable, energy efficient, high performance, credit card sized computer"[2]. its function is to provide a platform for developing and implementing high performance parallel processing. The 66-core version of the Parallella board manages over 90 GFLOPS (10^9 Floating Point Operations Per Second) while the 16 core version can reach 32 GFLOPS using only about 5 Watts. The Parallella is provided with an SD card that contains a customised ARM implementation of Linux (Ubuntu 14.04) which is also provided on the Parallella website [2]. The Parallella has a 1gbps Ethernet port allowing large amounts of information to be passed quickly over the network, this increases its

ability to work in a cluster as it can pass information to its peers faster provided that the switch is capable of handling the 1gbps bandwidth.

Approach

The first step will be to find information on high performance computing (HPC) with clusters. This will require a literature review, which will provide a better understanding on how HPC is done and how cluster computing is used to create high performance systems. Information on the Parallella and what it is capable of will need to be reviewed so they can be used correctly.
The second step will be the physical building of the Parallella cluster. This involves setting up cooling on all the Parallella boards using a fan, providing power to all the boards via the mounting hole pads, setting up a router or switch to facilitate communication between the boards and mounting and connecting all the boards to power and the switch. After setting up the components, operating systems and software will need to be installed on each board or have network booting set up on each. The software will have to be configured so that the boards will be aware of each other and work together.
Once the Parallella cluster is operational, comparisons against other systems can be made using software to benchmark each systems performance. Once comparisons and benchmarks are taken ways in which to optimise the Parallella cluster can be explored and tested, this involves changing the cluster configuration, using different compiler flags and any other optimisations discovered after further research. Results and any limitations of the Parallella boards are to be recorded.

Requirements

Hardware required to meet project goals:

- Multiple Parallella boards
- Switch and Ethernet cable
- Fans to cool the Parallella boards
- Desktop with a screen to interact with the headless Parallella cluster.

Progression Time line

Friday 27th February, 2015	Formal Written Proposal completed and accepted (submit to Supervisor and cc Project coordinator)
Tuesday 3rd March, 2015	Seminar Series 1: oral presentations 8-10 mins each
Sunday 29th March, 2015	Literature Review and Plan of Action (email to Supervisor and cc Project coordinator)
Monday 20th April, 2015	Have Parallella boards powered and mounted
Saturday 25th April, 2015	Operating system installed on the Parallella Boards
Monday 20th April, 2015	have Parallella boards working as a cluster
Monday 1st June, 2015	Run benchmarks and comparisons on the cluster and similar systems
Tuesday 28th July, 2015 , 4th and 11th Aug 2015	Seminar Series 2: oral presentations 15 mins each
Monday 14th September, 2015	Short Paper Submitted (Supervisors to confirm)
26th, 27th and 28th Oct 2015	Seminar Series 3: Final oral presentations 20 mins max - assessed
Friday 30th October, 2015	12 noon Project Deadline submitted to Project Coordinator
Friday 6th November, 2015	Research Website complete
Wednesday 18th November, 2015	Final Research Oral Examination
Week 1 - 4th Term	Skeleton outline of all chapters including major section and subsection headings.
Week 2 - 4th Term	Complete structure of final thesis including summary of contents of all sections and subsections.
Week 3 - 4th Term	By now you should have at least one or two chapter drafts.
Week 4 - 4th Term	Additional drafts of new chapters . etc.

References

[1] Networkboot parallella board. `https://plus.google.com/+shogunx/posts/K5taMhPKurp`. Accessed: 2015-02-27.

[2] The parallella board. `http://www.top500.org/lists/2014/11/`. Accessed: 2015-03-01.

[3] Top 500 super computers. `https://www.parallella.org/board/`.

[4] George S Almasi and Allan Gottlieb. Highly parallel computing. 1988.

[5] Rajkumar Buyya. High performance cluster computing. *New Jersey: F'rentice*, 1999.

[6] David Geer. Chip makers turn to multicore processors. *Computer*, 38(5):11–13, 2005.

[7] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel computing*, 22(6):789–828, 1996.

[8] Mark F Mergen, Volkmar Uhlig, Orran Krieger, and Jimi Xenidis. Virtualization for high-performance computing. *ACM SIGOPS Operating Systems Review*, 40(2):8–11, 2006.

[9] David A Padua and Michael J Wolfe. Advanced compiler optimizations for supercomputers. *Communications of the ACM*, 29(12):1184–1201, 1986.

[10] KY Sanbonmatsu and C-S Tung. High performance computing in biology: multimillion atom simulations of nanoscale systems. *Journal of structural biology*, 157(3):470–480, 2007.

[11] T Tezduyar, S Aliabadi, M Behr, A Johnson, V Kalro, and M Litke. Flow simulation and high performance computing. *Computational Mechanics*, 18(6):397–412, 1996. test.