BOINC and CUDA: Distributed High-Performance Computing for Bioinformatics String Matching Problems

Charl van Deventer[1], Willem A. Clarke[1] and Scott Hazelhurst[2]

Department of Electrical and Electronic Sciences

University of Johannesburg[1], P. O. Box 524, AucklandPark, 2006

Tel: +27 72 155 9066

and Department of Electrical and Information Engineering

University of Witwatersrand[2]

email: landon@mweb.co.za

Abstract - Research has shown that applications ported to utilize Graphics card hardware typically show at least an order of magnitude performance increase. This is typically achieved due to suitability and scale of the problems to these processors. String matching problems, prevalent in the bioinformatics field, have a low computation to data ratio and often a huge scale that limits them to classical supercomputers. We explore an option to try and scale classical GPGPU computing to large scales through the use of desktop grids.

Index Terms—GPGPU, CUDA, BOINC, Desktop Grids, Bioinformatics, String Matching, multiple sequence alignment

I. Introduction

The core of bioinformatics sequence analysis is a simple sequence. A string of characters representing nucleotides: Adenine(A), Guanine(G), Cytosine(C) and Thymine(T). All complexity and diversity comes from the scale of these sequences. The human genome for instance is approximately 4 billion characters long with data about all observed variations being databases terabytes in size. GPU computing and Desktop grids offer an opportunity to help process this large amount of information.

Desktop grids are a form of peer-to-peer computing. It is a way for owners of desktop computers to donate spare CPU cycles to scientific or corporate projects without negatively influencing themselves. This provides access to a great resource of computing power and hard drive space that would otherwise lie dormant in the personal computers of the public at home or at work.

Desktop grids have been becoming more and more prevalent in recent years due to cheaper and faster desktop machines and faster internet interconnectivity. The first two large-scale desktop grid attempts in 1997 was distributed.net, a project dedicated to deciphering encrypted messages, and GIMPS, a project dedicated to finding large prime numbers. Since then projects such as SETI@home and Folding@home have managed to gain a lot of publicity, having been installed on hundreds of thousands of PCs worldwide(1).

More recently, GPGPU (computing of non-graphics problems on graphics hardware) has been gaining increased attention as well. Currently it has found wide-spread use in physics simulations for games, encryption and decoding of compressed movie formats for playback. This has even gained the attention of desktop grid projects such as SETI@home, which has started utilizing the power of graphics cards to improve their throughput.

II. Desktop grid versus Grid Computing

Grid Computing refers to interconnected clusters set up at various locations that share their computing power for common goals. Unlike Desktop Grids though, these computers usually have similar operating systems, several cores, large throughput network connections and managed by the group that owns it. Desktop Grids on the other hand can be a mix of various operating systems and applications, managed by whoever owns the desktop and with the computing project set to a low priority. This means that individual desktops part of a desktop grid would have less efficiency than a dedicated grid computing computer, but the lower barrier of entry allows many more computers to join the project(1).

III. BOINC

BOINC (Berkeley Open Infrastructure for Network Computing) is a distributed grid middleware platform. This means that it hosts distributed applications and provides client and server software to allow new desktop grid computers to be added to a project with a minimum of configuration(1).

It has a simple to understand and use API for client-side applications. SETI@home has decided to through its weight behind BOINC and now uses the BOINC middleware for their project.

At the time of this writing BOINC boasts 36 projects listed on their webpage, over 300 000 volunteers providing over 550 000 computers, totaling a 24-hour average of over 5 PetaFLOPS.

IV. The Problem

The specific problem that this project is concerned with is that of EST sequence clustering. EST's are short (usually 80-800 characters long) nucleotide sequences from various mRNA sources. There are multiple copies of each source mRNA and each one has been randomly fragmented. The objective is to perform many-to-many overlap detection that allows us to group all the EST's from the same source into the same cluster. This vastly simplifies the later objective or reassembling the EST's into their original complete sequence.

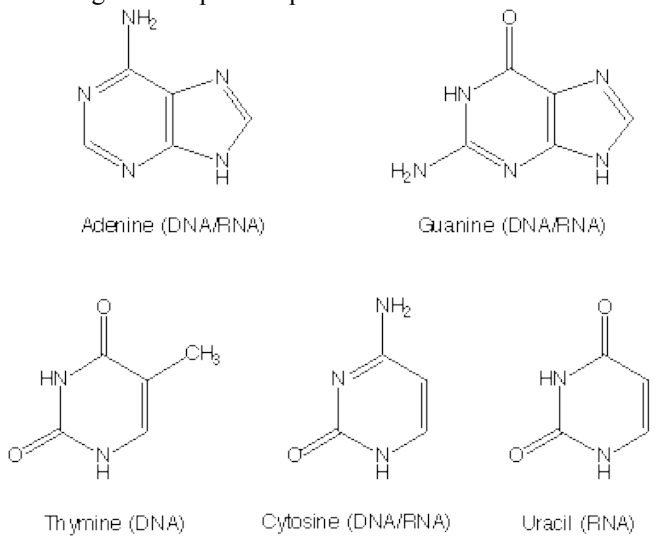

Figure 1: The Common Nucleotide Bases

A. D² Algorithm

The primary algorithm being implemented is called d², with the implementation based on the wcd tool by professor Scott Hazelhurst(2). The key to d² is the preprocessing step where all possible combinations of words of a specific length is counted. This allows a very quick and efficient many-to-many comparison:

$$d_k^2(x,y) = \sum_{i=1}^{4^k} (c_x(w_i) - c_y(w_i))^2$$

The function $c_x(w)$ is used to refer to the count of the occurrence of a particular word(sequence of specific characters) in the sequence x.

This provides a simple and quick 'distance metric' that can be tested to compare sequence similiarity, which is then used to group sequences in clusters.

B. FFT Algorithm

The FFT sequence alignment algorithm is also being considered for this project. It is an alignment-free comparison that utilizes FFT to test for the frequency of words.(3)

A quick example is given below, assume the string x.

x: ACGTNA

A: 100011

C: 010010

G: 001010

T: 000110

The string is decomposed into bit arrays, each bit signifying the occurance of a base.

The algorithm comes from the realisation that the basic comparison as given here can be simplified:

$$d(k) = \sum_{i=1}^{|x|} x(i)y(i+k)$$

where $|x|$ is the length of the string represented by x, $x(i)$ is the ith character of the string represented by x and $y(i+k)$ is the $i+k$th character represented by the string y.

This is a basic convolution, and as such can be simplified by converting the sequence into the frequency domain, performing a multiplication, then converting back into the time-domain. This converts the above O(n²) algorithm into a O(n log n) algorithm and allows one to make use of CUDA's built-in FFT libraries.

V. Experimental Hardware

Before suitability of this project over multiple computers can be tested, the experimental application must first be verified on a single machine. The testbed for this is a linux computer with a 3GHz Athlon II X2 250 processor, 4GB of RAM and a GeForce GTX 260 graphics card. Once verified, 8 similar computers with a mix of linux and windows operating systems will be tested and one high performance computer vision research laboratory GPU cluster with 7 GeForce GTX 295 graphics cards, testing a practical limit for multiple graphics cards.

VI. Limitations and Conclusion

It is hoped that this range of operating systems and hardware will show suitability of this or a similar project for large scale projects. The primary limitation is the latency between computation request and result, making desktop grids very unsuited for real-time applications. Security concerns for important projects and the need for validation of results also means that many calculations often need to be done several times on different computers.

However, the ability to repurpose existing desktop machines as opposed to buying cluster or supercomputer hardware has definite cost advantages, especially if combined with the great processing power that the average desktop can leverage due graphics cards that has become standard.

REFERENCES

(1) D. P. Anderson "BOINC: A system for Public-Resource Computing and Storage" *5th IEEE/ACM International Workshop on Grid Computing. November 8, 2004*

(2) Hazelhurst "Algorithms for clustering expressed sequence tags: the wcd tool" *South Africa Computing Journal* (2008) **40**

(3) K. Katoh, K. Misawa, K. Kuma, T. Miyata "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform" *Nucleic Acids Res.* **30** 3059-3066

Charl van Deventer received his undergraduate Computer Sciences degree in 2007 and his Electronic Engineering degree in 2008 from the University of Johannesburg and is presently studying towards his Master of Engineering degree at the same institution. His research interests include GPGPU, High Performance Computing, Bioinformatics and Computer Game Development.