FYP14020

Better Algorithms in Bioinformatics Applications

Project Plan

Supervisor: Prof. Lam Tak Wah
Ng Tsz Ho (3035017832)

Project abstract

BALSA, a DNA alignment and analysis tool, exploits the power of GPGPU. However, current version of BALSA is bound to run on local machine only. This project takes one step forward in developing a BALSA which runs in a cluster, with the technology of remote GPU driving.

1. Project Background

DNA sequencing and alignment are main topics in bioinformatics. They contribute greatly to clinical diagnosis and generic risk prediction. With next advancement in next generation sequencing (NSA), alignment software becomes applicable in the real world. Announced by the Illumia recently, sequencer cost is approaching the rate of $1,000 per whole genome sequencing. Many laboratories and hospitals nowadays routinely generate terabytes of NGS data daily.

On the other hand, alignment becomes the bottleneck along the pipeline. The challenges now become building faster and more cost-effective alignment software to deal with the large amount of sequenced data. HKU has been putting great endeavor in developing high-performance and practical alignment and analysis software. SOAP3-DP, as well as BALSA, which built on top of the former, are two of the state-or-art software in the world. They exploit the power of parallelization in general-purposed GPU to realize a reliable and high-throughput alignment channel; while at the same time maintaining a low hardware and operation cost.

1.1 The rCUDA technology

Traditional general-purposed GPU programming is bound to execute on local node. To drive GPUs on remote nodes and enable inter-node communication, message passing tools such as Message Passing Interface (MPI), Parallel Virtual Machine (PVM), etc. need to be used. However, as they are not optimized for GPU-programming, overheads will be heavy. Besides, difficulty of implementation will grow with the complexity of existing program, as well as large cost in maintaining and expanding the cluster.

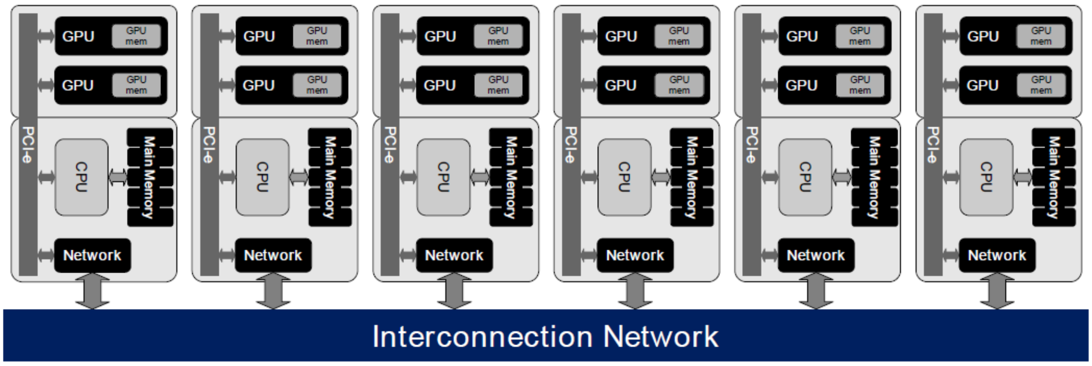
A new technology, remote rCUDA technology (rCUDA) remedies some of the problems. The rCUDA framework enables the usage of remote CUDA-compatible devices. This framework creates virtual CUDA-compatible devices on those machines without a local GPU. These virtual devices represent physical GPUs located in a remote host offering GPGPU services.

1.2 The motivation

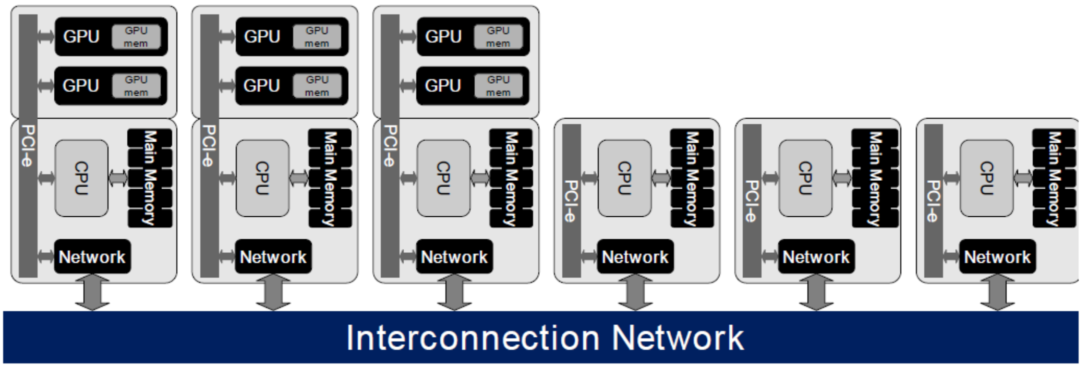
With BALSA proved to be efficient in single-node environment, the next step will be about cluster environment. It seems trivial that rCUDA is a perfect tool to start with. In the context of high-performance computing (HPC) clustering, adopting rCUDA in BALSA may lead to several advantages:

1. Reduces the number of GPUs installed in the computing facility. This reduces the rate of idleness of GPU, i.e. increasing CPU utilization and to energy savings, as well as other savings like acquisition costs, maintenance, space, cooling, etc.
2. By building a high-speed networked cluster, the overheads is minimized

A major concern is the overheads of longer distance to remote GPU. With a rough implementation of rCUDA to SOAP3-DP under a normal TCP/IP network, the overheads are around 28% without any optimization and network configuration. Therefore, it is optimistic to achieve a single-node-like performance.



(a) When rCUDA is not deployed into the cluster, one or more GPUs must be installed in those nodes of the cluster intended for GPU computing. This usually leads to cluster configurations with one or more GPUs attached to all the nodes of the cluster. Nevertheless, GPU utilization may be lower than 100%, thus wasting hardware resources and delaying amortizing initial expenses.



(b) When rCUDA is leveraged in the cluster, only those GPUs actually needed to addre overall workload must be installed in the cluster, thus reducing initial acquisition costs an overall energy consumption. rCUDA allows sharing the (reduced amount of) GPUs preser in the cluster among all the nodes.

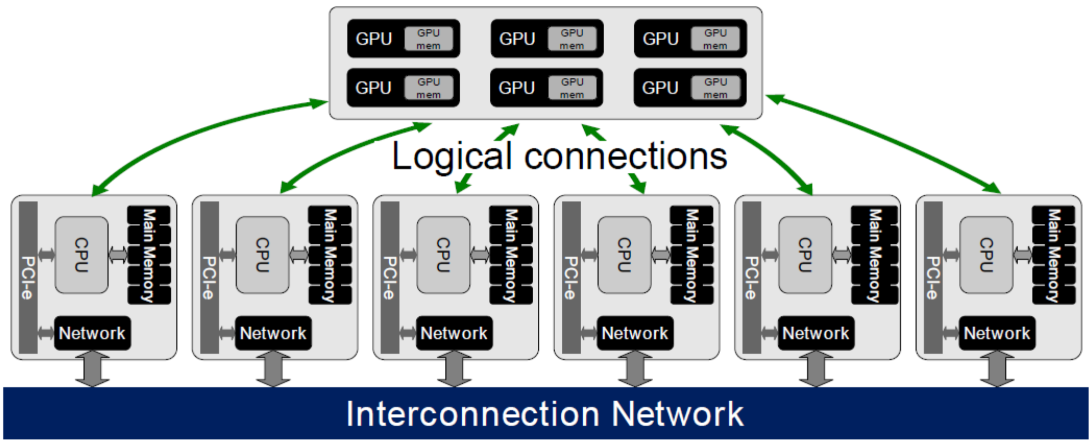


Figure 1. Principles of rCUDA

2. Project Objective

Objectives of the project are to

(1) Build a remote-gpu version of BALSA by integrating techniques of rCUDA
(2) Build a high-speed interconnected cluster as an environment
(3) Explore possibility of rCUDA in improving the scalability, and performance of current software

3. Project Methodology

The project will begin by integrating rCUDA into BALSA in the current TCP/IP network. As rCUDA at the time does not support CUDA driver API, alteration of existing codes should be made in order to be adapt to the limitation.

In the second stage, an InfiniBand network, or a Gigabit Ethernet network (GbE) will be built. This is to increase the communication speed between the server nodes and client nodes, thus the speed of the software.

Next, optimization will be done to further tune up the performance of software. Benchmarking, analysis, as well as documentation will be the last steps.

4. Project Schedule and Milestones

September	Preliminary setup
Oct – Dec	- BALSA implementation - Cluster construction
Jan	First Presentation and preparation
Feb	Interim Report
Feb – Apr	Optimization
Apr – May	- Analysis - Benchmarking - Documentation
May	- Final Report and Presentation Preparation - Final Presentation - Exhibition Preparation - Project Exhibition