Edit distance problem on a GPU-cluster

Antoine Veenstra
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
a.j.veenstra@student.utwente.nl

ABSTRACT

In this paper, we describe how the edit distance problem can be distributed over a GPU-cluster using MPI.

Keywords

OpenCL, Edit distance problem, GPU-cluster, C++, case study, GPGPU program, MPI, Message Passing Interface, case study

1. INTRODUCTION

As the amount of data increases the need for parallel does so too. The edit distance problem is used in various fields of research[?]. Fields such as Computational Biology, Signal Processing, and Text Retrieval. This algorithm has already been implemented on a single Graphics Processing Unit (GPU)[?], but to decrease the processing time even further a logical step is to increase the number of GPUs[?]. These GPUs allow the processing of larger amounts of data in parallel than is possible on a CPU.

The existing implementation of the edit distance problem uses a dynamic programming algorithm, which is well-suited for general-purpose computing on graphics processing units (GPGPU). The implementation was written in C++ using OpenCL which can run on both NVIDIA and AMD GPUs. An alternative would have been CUDA, which has been developed by NVIDIA and runs exclusively on NVIDIA GPUs. OpenCL has been and will be chosen over CUDA to guarantee compatibility with both types of GPUs.

MPI

By implementing the edit distance problem on a GPU-cluster instead of a single GPU the processing time could be reduced as the performance of a cluster exceeds that of a single unit[?]. The goal of the research described in this paper is to implement the edit distance problem on a GPU-cluster using MPI. This goal gives us the research question mentioned in the following section.

2. RESEARCH QUESTIONS

The research question of this proposal is:

How much can the processing time needed to calculate the edit distance problem be reduced using a GPU cluster which uses MPI?

A possible division in subquestions is:

1. How can the algorithm be divided in separate processes?
2. How can the algorithm be run on multiple devices using MPI?
3. What is the optimal number of GPUs when considering cost and efficiency?

3. BACKGROUND

3.1 OpenCL

3.2 GPGPU programming

3.3 MPI

3.4 Edit distance

The edit distance problem is way of measuring how much two strings differ from each other. The distance between two strings is measured by inserting, removing, and rearanging characters. The operations considered

4. RELATED WORK

4.1 Edit distance problem on GPU

4.2 Benchmark on a GPU-cluster

5. METHOD

6. CONCLUSIONS

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[2] J. Braams. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat*, 12(2):291–301, June 1991.

[3] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.

[4] M. Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.

[5] L. Lamport. *LaTeX User's Guide and Document Reference Manual.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[6] S. Salas and E. Hille. *Calculus: One and Several Variable.* John Wiley and Sons, New York, 1978.

13th Twente Student Conference on IT June 21st, 2010, Enschede, The Netherlands.