CAMPAIGN: an open-source library of GPU-accelerated data clustering algorithms

Kai J. Kohlhoff1,∗, Marc H. Sosnick2, William T. Hsu2, Vijay S. Pande3,4,5 and Russ B. Altman1,4,6

1Department of Bioengineering, Stanford University, Stanford CA 94305-5448, 2Department of Computer Science, San Francisco State University, 1600 Holloway Avenue, San Francisco, CA 94132-4163, 3Department of Chemistry, Department of Computer Science, 4Department of Structural Biology and 5Department of Genetics, Stanford University, Stanford CA 94305-5448, USA

ABSTRACT
Motivation: Data clustering techniques are an essential component of a good data analysis toolbox. Many current bioinformatics applications are inherently compute-intensive and work with very large datasets. Sequential algorithms are inadequate for providing the necessary performance. For this reason, we have created Clustering Algorithms for Massively Parallel Architectures, including GPU Nodes (CAMPAIGN), a central resource for data clustering algorithms and tools that are implemented specifically for execution on massively parallel processing architectures.

Results: CAMPAIGN is a library of data clustering algorithms and tools, written in ‘C for CUDA’ for Nvidia GPUs. The library provides up to two orders of magnitude speed-up over respective CPU-based clustering algorithms and is intended as an open-source resource.

New modules from the community will be accepted into the library and the layout of it is such that it can easily be extended to promising future platforms such as OpenCL.

Availability: Releases of the CAMPAIGN library are freely available for download under the LGPL from https://simtk.org /home/campaign. Source code can also be obtained through anonymous subversion access as described on https://simtk.org /scm/?group_id=453.

Contact: kjk33@stanford.edu

Received on March 10, 2011; revised on May 30, 2011; accepted on June 8, 2011

1 INTRODUCTION

Data clustering algorithms have been useful in many fields of computer science and are of ever increasing importance in the life sciences. For instance, data analysis in modern day biology and bioinformatics often involves the extraction of patterns from diverse sources such as gene expression data and protein structures (Andreopoulos, 2009; Belacel, 2006; Chodera, 2008; Daxin, 2004; Zemla, 2007). This can require substantial amounts of processing time, putting a limit on the amount and quality of information that can be derived in reasonable time. To allow the development of more sophisticated analysis protocols, we present Clustering Algorithms for Massively Parallel Architectures, Including GPU Nodes (CAMPAIGN); a library of GPU-accelerated clustering algorithms for large-scale datasets. Equipped with an initial set of tools and GPU-ports of well-established algorithms, including K-means, K-centers and hierarchical clustering, CAMPAIGN is intended to form the basis for devising new parallel clustering codes specifically tailored to the GPU and other massively parallel architectures.

The ‘C for CUDA’ parallel computing platform from Nvidia provides a framework for accessing CUDA-enabled graphics cards. The C-like syntax facilitates writing and maintaining code. We thus chose Nvidia’s CUDA parallel computing engine for developing the first release of the library. Providing the code as individual modules allows easy modification and extensibility, for example by future modules using OpenCL.

2 APPROACH

For the first release of the CAMPAIGN clustering library, we selected five popular clustering algorithms. For each algorithm, we implemented a serial CPU reference version and a GPU-accelerated version:

2. K-medoids (Ng, 1994).
3. K-centers (a K-medoids variant in which medoids are placed only once according to a heuristic) (Dasgupta, 2005; Gonzalez, 1985).
5. Self-organizing map (Kohonen, 1982).

We initially included a selection of Euclidean, Manhattan and Chebychev distance metrics. In general, we made an effort to ensure that the CPU and GPU versions produced the same result clusters. However, differences may occur because of rounding errors. For hierarchical clustering, substantial rearrangements of the dataset during run-time are required for our GPU version to be efficient. A CPU version that is intended to reproduce the same sequence of merging in the case of equidistant clusters is provided for comparison, but by default a more efficient sequential version is chosen. For K-means on the GPU, a basic implementation and an additional variant using parallel sorting for improved efficiency
Although there is substantial interest in fast clustering algorithms, previously published implementations are in most cases not readily available as either binaries or source code. This lack of availability hinders the evolution and incremental improvement of algorithms. In addition, such black-box implementations can be frustrating to use or understand and cannot be customized. Besides offering ready-to-use implementations of GPU-based clustering algorithms, CAMPAIGN can serve both as a readily available benchmark against which to test future implementations of such algorithms, as well as a seed for the creation of a more exhaustive library of clustering codes. The current library is limiting the size of usable datasets to the amount of memory available on the GPU, such as 4GB in case of the Tesla GPU used here for testing, with less memory available on more low-end graphic cards.

5 CONCLUSION

We have introduced CAMPAIGN, an open-source library of data clustering algorithms and tools that aims to grow through contributions by the scientific and technical communities. The first batch of algorithms distributed with the initial release of the library offers one to two orders of magnitude speed-up as compared with CPU reference implementations.

ACKNOWLEDGEMENTS

The authors are grateful to Imeen Hague and the staff at the Simbios NIH Center for Biomedical Computation for helpful suggestions.

Funding: National Institutes of Health through the NIH Roadmap for Medical research (U54 GM072970); a National Institutes of Health grant (LM-05652, to R.B.A.); Stanford School of Medicine Dean’s Fellowship (to K.J.K.).

Conflict of Interest: none declared.

REFERENCES

K.J.Kohlhoff et al.

Doxygen-generated HTML files are provided.

3 RESULTS

We executed the parallel clustering codes on a single Nvidia Tesla C1060 GPU and compared the results with the serial versions running on one core of an Intel Xeon E5420 2.93GHz CPU. We obtained performance improvements of one to two orders of magnitude for the CAMPAIGN GPU codes over sequential CPU reference implementations. Performance improvements over popular software packages such as MatLab were up to three orders of magnitude. In terms of accuracy, we find that for K-means, K-centers, K-medoids and hierarchical clustering, outputs for the different variants match closely with exception of machine-precision-related rounding errors, whereas for self-organizing map initial minor differences in rounding between two architectures are greatly amplified in successive steps.

We tested the algorithms with a biological dataset; a selection of 9461 gene array measurements of 20,099 human genes from NCBI GEO (Engroft, 2010). A number of performance comparisons are summarized in Figure 1.

4 DISCUSSION

Although there is substantial interest in fast clustering algorithms, previously published implementations are in most cases not readily available as either binaries or source code. This lack of availability...