Center for Technology for Advanced Scientific Component Software: Distributed CCA

State University of New York, Binghamton, NY, 13902

Summary

The overall objective of Binghamton's involvement is to work on enhancements of the CCA environment, motivated by the applications and research initiatives discussed in the proposal. This year we are working on re-focusing our design and development efforts to develop proof-of-concept implementations that have the potential to significantly impact scientific components. We worked on developing parallel implementations for non-hydrostatic code and worked on a model coupling interface for biogeochemical computations coded in MATLAB. We also worked on the design and implementation modules that will be required for the emerging MapReduce model to be effective for scientific applications. Finally, we focused on optimizing the processing of scientific datasets on multi-core processors.

Research Details

We worked on the following research projects that we are working on applying to CCA-based scientific applications.

1. *Non-Hydrostatic Hydrodynamics*: Non-static hydrodynamics are significantly more accurate at modeling internal waves that may be important in lake ecosystems. Non-hydrostatic codes, however, are significantly more computationally expensive, often prohibitively so. We have worked with Chin Wu at the University of Wisconsin to parallelize non-hydrostatic code. We have obtained a speed up of about 26 times maximum. Although this is significant progress, we hope to improve the performance further, such that it becomes a practical alternative to hydrostatic codes.

2. *Model-coupling for water-based ecosystems*: To answer pressing questions about water resources requires that physical models (hydrodynamics) be coupled with biological and chemical models. Most hydrodynamics codes are written in Fortran, however, while most ecologists work in MATLAB. This disconnect creates a great barrier. To address this, we are working on a model coupling interface that will allow biogeochemical computations written in MATLAB to couple with Fortran codes. This will greatly improve the productivity of ecosystem scientists.

2. *Low overhead and Elastic MapReduce Implementation Optimized for Memory and CPU-Intensive Applications*: Since its inception, MapReduce has frequently been associated

*607-777-4904, mgovinda@cs.binghamton.edu

with Hadoop and large-scale datasets. Its deployment at Amazon in the cloud, and its applications at Yahoo! for large-scale distributed document indexing and database building, among other tasks, have thrust MapReduce to the forefront of the data processing application domain. The applicability of the paradigm however extends far beyond its use with data intensive applications and diskbased systems, and can also be brought to bear in processing small but CPU intensive distributed applications. MapReduce however carries its own burdens. Through experiments using Hadoop in the context of diverse applications, we uncovered latencies and delay conditions potentially inhibiting the expected performance of a parallel execution in CPU-intensive applications. Furthermore, as it currently stands, MapReduce is favored for data-centric applications, and as such tends to be solely applied to disk-based applications. The paradigm, falls short in bringing its novelty to diskless systems dedicated to in-memory applications, and compute intensive programs processing much smaller data, but requiring intensive computations. In this project, we focused both on the performance of processing large-scale hierarchical data in distributed scientific applications, as well as the processing of smaller but demanding input sizes primarily used in diskless, and memory resident I/O systems.

We designed LEMO-MR [1], a Low overhead, elastic, configurable for in- memory applications, and on-demand fault tolerance, an optimized implementation of MapReduce, for both on disk and in memory applications. We conducted experiments to identify not only the necessary components of this model, but also trade offs and factors to be considered. We have initial results to show the efficacy of our implementation in terms of potential speedup that can be achieved for representative data sets used by cloud applications. We have quantified the performance gains exhibited by our MapReduce implementation over Apache Hadoop in a compute intensive environment.

3. *Cache Performance Optimization for Processing XML and HDF-based Application Data on Multi-core Processors:* It is important to design and develop scientific middleware libraries to harness the opportunities presented by emerging multi-core processors. Implementations of scientific middleware and applications that do not adapt to the programming paradigm when executing on emerging processors can severely impact the overall performance. In this project, we focused on the utilization of the L2 cache, which is a critical shared resource on chip multiprocessors (CMP). The access pattern of the shared L2 cache, which is dependent on how the application schedules and assigns processing work to each thread, can either enhance or hurt the ability to hide memory latency on a multi-core processor. Therefore, while processing scientific datasets such as HDF5, it is essential to conduct fine-grained analysis of cache utilization, to inform scheduling decisions in multi-threaded programming. In this project, using the TAU toolkit for performance feedback from dual- and quad-core machines, we conducted performance analysis and recommendations on how processing threads can be scheduled on multi-core nodes to enhance the performance of a class of scientific applications that requires processing of HDF5 data. In particular, we quantified the gains associated with the use of the adaptations we have made to the Cache-Affinity and Balanced-Set scheduling algorithms to improve L2 cache performance, and hence the overall application execution time [2].

* 607-777-4904, mgovinda@cs.binghamton.edu

References:

1. Zacharia Fadika, Madhusudhan Govindaraju, ``MapReduce Implementation for Memory-Based and Processing Intensive Applications'', accepted in *2nd IEEE International Conference on Cloud Computing Technology and Science*, Indianapolis, USA, Nov 30 - Dec 3, 2010.

2. Rajdeep Bhowmik, Madhusudhan Govindaraju, ``Cache Performance Optimization for Processing XML-based Application Data on Multi-core Processors'', in proceedings of *The 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 17-20, 2010, Melbourne, Victoria, Australia.

Contact Information:

Madhusudhan Govindaraju
Binghamton University
State University of New York (SUNY)
mgovinda@cs.binghamton.edu
Phone: 607-777-4904

---

*607-777-4904, mgovinda@cs.binghamton.edu