ASCR Project Final Report

"*A Fault-oblivious Extreme-scale Execution Environment*"

OSU PI: Ponnuswamy Sadayappan

OSU Project #: DE-SC0005034
Project Start: 09/01/2010
Project End: 08/31/2013
Total Funding Received at OSU: $225,000
External Partners: Ronald Minnich, Sandia National Laboratory ***(LEAD)***, Sriram Krishnamoorthy, Pacific Northwest National Laboratory

Statement of Work/Abstract:

Exascale computing systems will provide a thousand-fold increase in parallelism and a proportional increase in failure rate relative to today's machines. Systems software for exascale machines must provide the infrastructure to support existing applications while simultaneously enabling efficient execution of new programming models that naturally express dynamic, adaptive, irregular computation; coupled simulations; and massive data analysis in a highly unreliable hardware environment with billions of threads of execution.

We propose a radically new approach to the data and work distribution model provided by system software based on the unifying formalism of an abstract file system. The proposed hierarchical data model provides simple, familiar visibility and access to data structures through the file system hierarchy, while providing fault tolerance through selective redundancy. The hierarchical task model features work queues whose form and organization are represented as file system objects. Data and work are both first class entities. By exposing the relationships between data and work to the runtime system, information is available to optimize execution time and provide fault tolerance. The data distribution scheme provides replication (where desirable and possible) for fault tolerance and efficiency, and it is hierarchical to make it possible to take advantage of locality. The user, tools, and applications, including legacy applications, can interface with the data, work queues, and one another through the abstract file model. This runtime environment will provide multiple interfaces to support traditional Message Passing Interface applications, languages developed under DARPA's High Productivity Computing Systems program, as well as other, experimental programming models. We will validate our runtime system with pilot codes on existing platforms and will use simulation to validate for exascale-class platforms.

In this final report, we summarize research results from the work done at the Ohio State University towards the larger goals of the project listed above.

Significant Studies and Results:

Load balancing via Resource Sharing Barriers: A new load balancing mechanism called a resource-sharing barrier (RSB) was implemented in the GA (Global Arrays) framework, and its use was demonstrated in improving the performance of a production Monte Carlo application in the NWCHEM computational chemistry suite - Dynamic Nucleation Theory Monte Carlo (DNTMC). The DNTMC application utilizes two levels of parallelism: i) concurrent "walkers"

independently progress through a number of steps before a barrier synchronization across walkers, and ii) each walker uses a process group with several GA processes to perform energy calculations in parallel. Load imbalance between different walkers often results since the total work performed by different walkers can be different. The Resource Sharing Barrier is a novel mechanism to temporarily allocate processor resources from process groups waiting at a barrier to other groups that are still active. The RSB-based implementation of DNTMC demonstrated significant performance improvement on a full computation chemistry application, requiring minimal change to existing code. The work was published at IPDPS 2012 [1].

Optimization of Distributed Tensor Contraction Expressions

General matrix multiplication or tensor contractions are an integral part of various computational methods used for modeling problems in science. In *ab initio* computational quantum chemistry, correlation methods such as CC, CI, EOM-CCSD, MRCC, MBPT, active space methods all rely heavily on being able to calculate well-defined sequences of tensor contractions. Similar methods in nuclear physics are emerging. Hence, it is important to develop efficient algorithm for performing sequences of tensor contractions. Two complementary aspects in this regard were addressed in our research:

1. Given a single large tensor contraction, how can it be efficiently executed on a distributed-memory computer system?

2. Given a tensor contraction expression involving a large collection of tensor contractions with inter-contraction dependences, how can the collection of contractions be efficiently executed on a distributed-memory computer system?

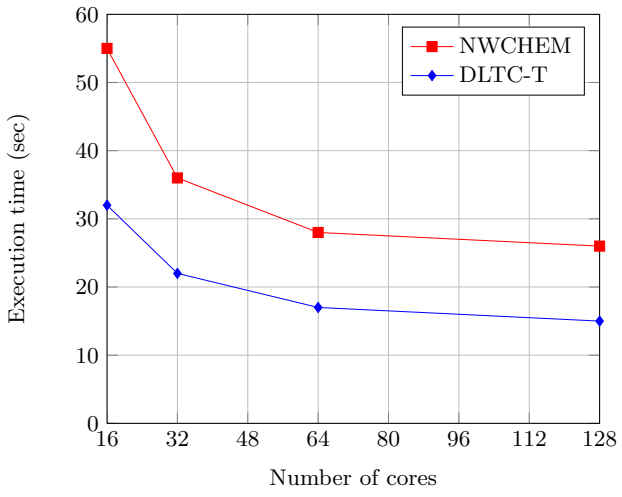
Below we summarize our contributions in addressing the above two problems:

RRR framework for distributed tensor contraction: We developed a framework with three fundamental communication operators to generate communication-efficient contraction algorithms for arbitrary tensor contractions: reduction, recursive broadcast, and rotation. We showed that for a given amount of memory per processor, the framework is communication optimal for all tensor contractions. Given any distribution of tensors, a contraction can begin immediately using a suitable combination of rotation, reduction, and recursive SUMMA. We proposed various algorithms for contracting distributed tensors of arbitrary dimensions based on their distribution. We developed a cost-driven model to obtain highly efficient algorithms based on memory constraints and interconnect topology. Tensors that appear in aforementioned areas of science are also symmetric in nature. We introduced a novel approach that avoids data redistribution in contracting symmetric tensors while avoiding redundant storage and maintaining load balance. We presented results for several contractions that appear in CCSD methods and compare the results with Cyclops Tensor Framework (CTF). Some representative results showing improvement over an NWChem proxy (a stand-alone application that implemented the distributed tensor contraction scheme employed by NWChem) is shown below for the contraction C[i,j,m,n] = A[l,j,k,l]*B[k,l,m,n].

Cores	NWChem Proxy	RRR (No Replication)	RRR (With Replication)
4096	36.89	32.63	20.43
16384	19.94	13.072	6.63
65536	9.07	4.87	2.023
262144	9.22	2.63	0.84

This research was described in a paper published at SC'14 [3]. Open source software for the RRR framework for distributed tensor contraction is available at https://github.com/samyam/DTCF/

Load Balancing of Tensor Contraction Expressions via Dynamic Task Partitioning: The second issue listed above was addressed by developing the DLTC (Dynamic Load-balanced Tensor Contractions framework, a domain-specific library for efficient task parallel execution of tensor contraction expressions. The framework decomposes each contraction into smaller units of tasks, represented by an abstraction referred to as iterators. We exploit an extra level of parallelism by having tasks across independent contractions executed concurrently through a dynamic load balancing run- time. We demonstrated the improved performance, scalability, and flexibility for the computation of tensor contraction expressions on distributed-memory parallel computers using examples from Coupled Cluster (CC) methods. A representative example of the improvement obtained by using DLTC over the production NWChem software suite is shown below, for a coupled-cluster modeling of the Uracil molecule.



More results are presented in a publication describing this research, which was published at SC'13 [2]. Open source software for DLTC is available at https://github.com/samyam/DTCF/tree/master/DLTC

Research Impact: Both the RRR framework and the DLTC framework are of significant interest to the NWChem team, and are planned to be incorporated into NWChem in the future.

[1] H. Arafat, J. Dinan, S. Krishnamoorthy, T. Windus, and P. Sadayappan, “Load Balancing of Dynamical Nucleation Theory Monte Carlo Simulations Through Resource Sharing Barriers,” International Parallel and Distributed Processing Symposium (IPDPS’12).

[2] P.-W. Lai, K. Stock, S Rajbhandari, S Krishnamoorthy, and P Sadayappan, "A Framework for Load Balancing of Tensor Contraction Expressions via Dynamic Task Partitioning" In *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13)*.

[3] S Rajbhandari, A. Nikam, P.-W. Lai, K. Stock, S Krishnamoorthy, and P Sadayappan., "A Communication-Optimal Framework for Contracting Distributed Tensors,” In *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'14)*.