Yearly Project Progress Report

DynAX: Innovations in Programming Models, Compilers and Runtime Systems for Dynamic Adaptive Event Driven Execution Models

Award Number: DESC0008716
Dates of Performance: 9/1/2014 to 8/31/2015
Report Date: 12/31/15

Principal Investigator
Guang Gao, ET International, Inc.

CoPIs:
Benoit Meister, Reservoir Labs, Inc.
David Padua, University of Illinois Urbana Champaign
Andres Marquez, Pacific Northwest National Laboratories

Table of Contents

Introduction
Accomplishments
    ETI Accomplishments
    TCE
    Resilience (containment domains)
    MPI Interoperability
    PNNL Accomplishments
    Reservoir Labs Accomplishments
    UIUC Accomplishments
Further Details
    Technologies Delivered
        Q9-Q12
Presentations
Publications
Websites

Introduction

This report outlines the work that was done in the four quarters (i.e. Q9, Q10, Q11, Q12) of year 3 by the DynAX Team (ET International, Reservoir Labs, UIUC, and PNNL). Given that there is a NCE upto December 31st, here we document mainly the work done on Y3 upto to August 31st. For the NCE period the reader should refer to the final project report (see: the websites listed below for a link to DynAX deliverables).

Accomplishments

The accomplishments for the year are broken down into four sections corresponding to each of the four institutions : ETI, Reservoir, UIUC and PNNL. Note that these accomplishments have already been reported in more details in our Q9, Q10, Q11 and Q12 reports. The Year 3 report is more of an extended summary with highlights.

ETI Accomplishments

In Y3, ETI finished study of the TCE application provided by PNNL. Additionally, throughout the year ETI incorporated and studied resilience techniques within SWARM and how to achieve MPI interoperability.

TCE

ETI has adapted the Tensor Contraction Engine (TCE) provided by PNNL in order to make it more suitable for exploring our research questions related to data placement, data movement, memory access models and scheduling.

In Q9 and Q10 , we completed the block-parallel implementation of TCE using OCR and moved to implementing multiple versions using SWARM. We completed a task-parallel version in SWARM. We have observed that a block-parallel version similar to the OCR version should be feasible with a few additional modifications. However OCR continued to evolve and has not been a stable target at the time when the Y3 of DynAX project has been completed. The progress and results of TCE related work accomplished by ETI can be found in the

corresponding quarterly reports in Q9 and Q10, as well as the background information on TCE in relevant quarterly reports of Y2.

Resilience (containment domains)

In Q9, we began research into resilience and fault tolerance using containment domains. We developed a research plan with insight and feedback from Mattan Erez and his UT Austin team. We began work on a prototype implementation of containment domains within SWARM. At this stage the prototype was capable of creating containment domains containing codelets, preserving data within domains, checking for errors upon completion, and rerunning the corresponding containment domain if errors were detected.

In Q10, we improved upon our initial implementation of containment domains within SWARM. We extended the implementation to support multiple perservations per containment domain, allowing multiple containment domains to be active at a time (for concurrent domain execution), and enhanced the flexibility of application programmers to place containment domains within applications. Additionally, we incorporated the ability to nest domains within the prototype. Finally, we began work on an initial implementation of Cholesky using containment domains within SWARM.

In Q11, we collected and analyzed the results of Cholesky within SWARM and produced a detailed technical report on resilience. Through our results we demonstrated the feasibility of the approach by showing low overhead and high adability within the SWARM framework.

In Q12 and further extended into the NCE period, we continued our design studies of the containment domain approach within SWARM-like program execution and programming models. We explored the design space and confines of containment domains within SWARM, specifically the properties of well-behavedness and loop constructs. We additionally published a full paper to the Mini-Symposium on Energy and Resilience in Parallel Programming (ERPP 2015) held in conjunction with the International Conference on Parallel Computing (ParCo 2015). The reader is referred to the task 8.1 in the Q12 report for more details, as well as, additional updates found in our NCE report (see: the websites listed below for a link to DynAX deliverables).

MPI Interoperability

In Q9, we began investigating MPI interoperability within SWARM. A number of important conclusions were reached: (1) interoperability should not degrade performance, (2) legacy codes can be straightforwardly parallelized between MPI calls in a manner similar to how OMP and MPI interoperate today, (3) the former method has limitations due to only the main MPI

thread being able to make MPI calls. To this end, we began studying ways of allowing MPI calls in non-blocking functions with codelet semantics.

In Q10, we considered numerous methods of achieving MPI interoperability with SWARM. These include incorporating SWARM calls into a traditional MPI program, incorporating MPI calls within a SWARM program, creating an MPI compatibility layer within SWARM, and fully rewriting MPI programs within SWARM.

In Q11, we produced a technical report summarizing results of our study of various methodologies to enable MPI interoperability with SWARM. This report demonstrates a number of case-studies using small codes.

In Q12, we consolidated our ideas on interoperability and focused mainly on how to incorporated SWARM calls into traditional MPI programs. To this end, we extended our original case studies beyond those reported in prior reports with a practical example of a matrix multiplication application. Additionally, we conducted a comparative study of MPI+X work within the field. During the NCE period, the PI has continued to give more thoughts on how to continue this research to further XStack goals.

PNNL Accomplishments

During Y3 of the DYNAX project, PNNL concentrated on improving management of data movement and resource underutilization within the Group Locality framework. Group Locality takes care of creating scattering functions to increase resource utilization and data locality at compile time. Additionally, PNNL enhanced the Jagged Tiling approach with different tile shapes that help expose additional inner parallelism in the lower levels of the tiling hierarchies. Our base architecture for this framework is the Intel Xeon Phi architecture with selected and representative stencil kernels. We show improvements ranging from 5.58% to 31.17% over existing state-of-the-art techniques.

PNNL also introduced a data restructuring framework within Group Locality that uses access patterns for a group of threads -- represented under a Polyhedral formulation – to move and restructure data. We start with hierarchical tiled code, as developed in our previous research under this program, and apply data transformations at each level to improve data residence. The main contributions of this methodology include a collaborative data restructuring for group reuse and a low overhead transformation technique that exploits locality. We used an exemplar many core architecture, Tilera TileGX, to show improvements over optimized OpenMP code: performance increase of up to 31% in GFLOPS. The restructuring framework also yielded improvements over our own previous work (the fine grained tiling techniques) for selected kernels.
For more in-depth details, please refer to Brandywine Q9, Q10, Q11 and Q12 reports.

Reservoir Labs Accomplishments

During Year 3, we have developed support for distributed computing and for distributed block-sparse computation in R-Stream, as well as UIUC's HTA/PIL as a front-end to R-Stream.

We first developed a backend and runtime to R-Stream for parallelization of dense array computations on clusters, based on a PGAS abstraction. The PGAS runtime was implemented directly on Global Arrays (GAs). A paper on the backend and runtime was accepted at HPEC'15, along with a communication-reducing optimization based on data duplication.

We then moved on to implementing an R-Stream mapping path, backend and runtime to produce block-sparse computations on clusters, still based on a PGAS abstraction. The underlying runtime supports hardware-backed Remote Direct Memory Accesses (RDMAs), and avoids communications *and* computations due to sparsity. The API used to manage the PGAS distributed memory has the form of an extended Direct Memory Access (DMA) API. The compiler performs optimizations that minimize the runtime overhead.

We also completed support for HTA and PIL as a front-end to R-Stream. Very encouraging results obtained by combining HTA with R-Stream in a hierarchical fashion are reported in the UIUC section below.

UIUC Accomplishments

In the Y3 of the Dynax project, the UIUC team worked on implementing the SPMD execution of Parallel Intermediate Language (PIL) and Hierarchically Tiled Arrays (HTAs) on the SWARM runtime system. We developed a strategy to map HTA program executions onto the SWARM runtime system in fork-join fashion during Y1 and Y2. However, when programs execute in the fork-join mode, parallel operations have implicit global barriers in them. Since global barriers are time consuming and greatly restrict asynchronous execution of codelets and can as a result exacerbate load imbalance, we decided to experiment with the SPMD mode where processes can synchronize with each other through point-to-point synchronization primitives and global barriers can be avoided. To evaluate the performance of both execution modes, we implemented six of the NAS Parallel Benchmarks including EP, IS, FT, CG, MG, LU. We described the performance results of SPMD execution mode in the Q11 report.

To better understand the execution behavior of SPMD mode, we implemented dense Cholesky factorization for detailed analysis. We chose dense Cholesky factorization because the computation is irregular, and we can easily change the parameters to observe the performance of different problem sizes and task granularities. We discovered that although the SPMD

execution allows more asynchrony, the static distribution of data tiles and owner computes rule limits the scheduling decision of the runtime system and it makes the performance suffer. This discovery led us to implement nested parallelism in PIL to enable dynamic scheduling of fine-grain codelets as a solution. In our experiments we show that in certain configurations when nested parallelism is enabled, the SPMD execution results can be better than the fork-join execution results. We described Cholesky factorization and nested parallelism in the Q11 and Q12 reports.

In Y3 we also worked with the Reservoir team on the integration of PIL generated code and R-Stream optimized code. The user can write more intuitive parallel program in PIL and use R-Stream compiler to optimize the computation kernels. With the integrated compilation flow, the user can focus on coarse-grain parallelism and leave the non-trivial and low-level optimizations to the R-Stream compiler. Since PIL and R-Stream both support OpenMP and ETI SWARM as backend, we implemented the integrated compilation flow for both backends. We conducted experiments with both the OpenMP version and the SWARM version on a 4 Intel Xeon E5-4620 processors (32-core) machine. To rule out the effects of hyper-threading, we ran the matrix-matrix multiplication benchmark using up to 32 threads. The results show that the performance scales with the amount of worker threads available. For OpenMP we observed up to 42x speedup when 32 worker threads were used and up to 27x speedup for SWARM.

Further Details

Further details on the above Accomplishments can be found in the Q10, Q11 and Q12 reports:

- Brandywine XStack Report Q9
- Brandywine XStack Report Q10
- Brandywine XStack Report Q11
- Brandywine XStack Report Q12

Technologies Delivered

Additional information on deliverables can be found on the DynAX page of the XStack wiki: https://xstackwiki.modelado.org/DynAX#Deliverables

Q9-Q12

- Prototype of nested containment domains within SWARM **(ETI Q9-Q11)**
- Case study of CD based cholesky decomposition within SWARM **(ETI Q12)**
- Case study of MPI+X **(ETI Q12)**
- NWChem TCE module:
    - Block-level parallel OCR **(ETI Q9-Q10)**
- R-Stream cluster backend based on PNNL's Global Arrays **(Reservoir Q09-Q10)**

- R-Stream block sparse cluster runtime **(Reservoir Q11-Q12)**
- PIL compiler and HTA SPMD execution mode implementation **(UIUC Q9-Q10)**
- HTA SPMD performance evaluation using NAS Parallel Benchmark and Block Cholesky Factorization **(UIUC Q11-Q12)**
- PIL SCALE backend integration with R-Stream **(UIUC & Reservoir Q12)**
- Comparison between ACDT frameworks in two fine grain runtimes: OCR versus SWARM [ICPDAS'14] **(PNNL Q10-12)**
- New loop tiling technique for parallel start applications [CGO'15] **(PNNL Q11)**
- Data restructuring framework for affine applications codes [HPCC'15] **(PNNL Q12)**

Presentations

- Presentation on Resilience and containment domains by Sam Kaplan in May 2015.
- Presentation on group locality and gregarious data restructuring in July 2015.
- Presentation on Block-sparse support in R-Stream by S. Tavarageri, A. Konstantinidis and B. Meister in July 2015.
- Presentation on containment domains within SWARM in August 2015.
- Ph.D dissertation defense on 'A Framework for Group Locality Aware Multithreading' by Sunil Shrestha in August 2015.

Publications

- S. Kaplan, S. Pino, A. Landwehr, G. Gao. **"Landing Containment Domains on SWARM: Toward a Robust Resiliency Solution on a Dynamic Adaptive Runtime Machine."** To Appear in the proceedings of the 2015 international Parallel Computing conference (ParCo'15). Edinburgh, Scotland, UK, September 1 – 4, 2015.
- S. Shrestha, J. Manzano, A. Marquez, S. Zuckerman, S. L. Song and G. Gao. **"Gregarious Data Re-structuring in a Many Core Architecture."** Invited paper to the 17th international conference on High Performance Computing and Communication (HPCC 2015). New York, USA, August 24 – 26, 2015.
- S. Shrestha, J. Manzano, A. Marquez, J. Feo and G. R. Gao. "**Locality Aware Concurrent Start for Stencil Applications.**" In the 2015 International Symposium on Code Generation and Optimization, San Francisco, CA, USA, February 7-11, 2015.
- A. Marquez, J. Manzano, S. Song, B. Meister, S. Shrestha, T. St. John and G. R. Gao. **"ACDT: Architected Composite Data Types Trading-in Unfettered Data Access for Improved Execution."** In the 20th IEEE International Conference on Parallel and Distributed Systems, Hsinchu, Taiwan, December 16 – 19, 2014
- S. Shrestha, J. Manzano, A. Marquez, and G. R. Gao. **"A Framework for Resource Aware Multithreading."** Poster presented at the International Conference for High Performance Computing, Network, Storage and Analysis (SC 14). New Orleans, LA, USA, November 16 – 21, 2014. Best poster nominee.

- S. Shrestha, J. Manzano, A. Marquez, J. Feo and G. R. Gao. **"Jagged Tiling for Intra-tile Parallelism and Fine-Grain Multithreading."** In the 27th International Workshop on Languages and Compilers for Parallel Computing, Hillsboro, OR, USA, September 15 – 17, 2014
- PhD Thesis Sunil Shrestha **"A framework for Group Locality Aware Multithreading."** Fall 2015.
- S. Tavarageri, B. Meister, M. Baskaran, B. Pradelle, T. Henretty, A. Konstantinidis, A. Johnson, R. Lethin. **"Automatic Cluster Parallelization and Minimizing Communication via Selective Data Replication."** In proceedings of the 2015 IEEE High Performance Extreme Computing conference (HPEC'15), 15-17 September 2015, Waltham, MA.
- A. Marquez, J. Manzano, S. Song, B. Meister, S. Shrestha, T. St. John and G. R. Gao. **"ACDT: Architected Composite Data Types Trading-in Unfettered Data Access for Improved Execution."** In the 20th IEEE International Conference on Parallel and Distributed Systems, Hsinchu, Taiwan, December 16 – 19, 2014.
- Chih-Chieh Yang, Juan C. Pichel, Adam R. Smith, David A. Padua. **"Hierarchically Tiled Array as a High-Level Abstraction for Codelets."** In the Fourth Workshop on Data-Flow Execution Models for Extreme Scale Computing, 2014.
- Chih-Chieh Yang, Juan C. Pichel, Adam R. Smith, David A. Padua. **"Hierarchically Tiled Array for Exascale Computing."** In the Fifth Workshop on Programming Abstractions for Data Locality (PADAL'15), 2015.
- Adam Smith. **"The Parallel Intermediate Language."**, Ph.D. dissertation, Computer Science Dept., University of Illinois at Urbana-Champaign, September 2015.

Websites

The URLs listed below contain STI delivered during the course of the DynAX project or software or methods relevant to delivered STI.

- Deliverables of DynAX project:
  https://xstackwiki.modelado.org/DynAX#Deliverables
- Scalapack's two dimensional block-cyclic distribution:
  http://netlib.org/scalapack/slug/node75.html
- Polyhedral library Polylib:
  http://icps.u-strasbg.fr/polylib/
- NWChem:
  http://www.nwchem-sw.org/index.php/Main_Page
- TCE correlation models:
  http://www.nwchem-sw.org/index.php/TCE#CCSD.2CCCSDT.2CCCSDTQ.2CCISD.2CCISDT.2CCISDTQ.2C_MBPT2.2CMBPT3.2CMBPT4.2C_etc._--_the_correlation_models