Contract Year 2 Interim Status Report

Award #: **DE-SC0008717**
Recipient: **Intel Federal LLC**
Project Title: **TRALEIKA GLACIER X-STACK**
PI: **Shekhar Borkar**
Report Date: **March 14, 2014**
Period Covered by Report: **September 1, 2013 to February 28, 2014**

Acknowledgment: This material is based upon work supported by the Department of Energy [Office of Science] under Award Number DE-SC0008717.

Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Introduction

This report combines the executive summaries and publications of milestone status reports previously supplied by Intel to DOE:

- Milestone 5 Status Report, provided 12/23/13
- Milestone 6 Status Report, provided 3/10/14

[redacted]

September 2013 – November 2013 Progress

We are shifting our efforts from infrastructure building—major thrust last year—to researching the X-Stack software using the infrastructure.

We have identified the Conjugate Gradient (CG) computation as a key kernel, being relevant for both CESAR (CG is dominant in the Nekbone proxy app) and ExMaT co-design center's apps, and have started the investigations. We are also focusing on the CoMD proxy application, and on track to demonstrate refactored Lulesh on the research software stack in the upcoming applications workshop (formerly Hackathon).

Porting of OCR (the open community runtime) to our simulator (FSim) has started with the aggressive goal of getting it ready for the workshop by mid-January. OCR was also highlighted at the Birds of a Feather session at SC13, with demonstration of the OCR runtime running the Unbalanced Tree Search benchmark. The novelty is in adaptation to changes in the environment, such as cores going offline or runtime goals changing (high performance to low power consumption, etc).

The programming system is making good progress. The CnC (Concurrent Collections) tuning system is working, provides 19% improvement on an unturned asynchronous execution on a shallow platform with today's costs. We expect more benefits on a deeper system or with the future costs. PNNL worked on a high-level representation of Lulesh in CnC. Furthermore, we have also implemented some of the NAS benchmarks in HTAs (Hierarchical Tiled Arrays).

We introduced ISA modifications to more compactly encode integer signed and unsigned arithmetic, saving about 50 instructions, or about 10% of the opcode space. These changes were reflected by updating the LLVM compiler and binutils to keep the tool chain current. We also continue to integrate the APIs for power and for software managed caches with the compiler and the FSim simulator.

To investigate introspection and self-awareness, we have added capabilities and infrastructure to the FSim simulator required for basic feedback loops needed for management algorithms. These include counters and registers in the hardware description, as well as capability of automatically simulating heat generation and heat transfer for any chip configuration and ambient temperature. We expect demonstration of a simple feedback loop in the near future.

To improve tuning of the system, we now have a visualization tool that allows viewing of energy usage throughout the system. This tool can display energy usage in a variety of ways, allowing filtering by EDTs, by hardware component, or by time. As an example, tracking data movement in OCR has allowed us to generate datasets that can be fed into this visualization tool to demonstrate the effects of local memory size on total energy usage.

December 2013 – February 2014 Progress

An application Workshop brought together the members of the Traleika Glacier project as well as the co-design centers for three days. We discussed current progress and got much better understanding of the science behind the applications of interest to DoE. Specifically, we discussed the work on LULESH and started exploring combustion and its adaptive mesh refinement (AMR) and multi-grid (mini GMG) components. As a result, the applications team is now focused on two applications to port on OCR and FSim: Conjugate Gradient (CG-kernel) and CoMD, both are working on OCR-x86 and will be targeted to FSim.

The open community runtime (OCR) port on an x86 platform is very stable. The OCR port on TG architecture now runs on an x86 platform for quick validation and debug; we are making steady progress to run it on the TG architecture simulator (FSim) this quarter.

The specification for a new 64 bit ISA is released to the wider Traleika-Glacier team, which incorporates feedback from applications, compiler development, runtime, and hardware. Specific key features include: 64 bit encoding, support for transcendental functions, enhanced DMA and hardware queue operations, and interrupt capabilities. To evaluate the architecture with TG's software stack, ETI focused on improving the simulator to match the changes. Their data movement model shows that although memory size does have a large effect on data movement energy, it is overshadowed by the constant leakage energy, with little effect on the total energy dissipation of the system.

To support 64 bit transition, we have updated binutils, and ongoing compiler support. We have started our efforts to apply R-Stream for generating optimized OCR versions of the proxy application programs, and understanding the miniGMG application benchmark to facilitate the mapping of miniGMG through R-Stream to generate an optimized OCR version of the miniGMG code.

Our high-level programming model, CnC, and our low-level programming model, OCR, are very consistent in their asynchronous, event-driven, task-based execution. Transition of the user whiteboard version of LULESH to a CnC graph specification was straightforward. We both validated the ease of use of CnC and also uncovered some new optimization potential. The Hierarchical Tiled Array (HTA) work shows potential for programming productivity with performance matching that of the tuned OpenMP code.

University of Delaware team demonstrated the ability to model heat dissipation and transfer in a multi-block simulation (9-blocks) with the capability of the simulator (FSim) to trace energy and heat. The demonstration was performed using a matrix-vector kernel as a basis. This is a start to incorporate introspection and self-awareness in the system software.

On the architecture front, we have continued to integrate the APIs for energy management and software managed caches in the Fsim simulator and automate their use with the compiler to demonstrate that the compiler can generate code for these API automatically and efficiently.

Schedule Status

Our status above represents progress against milestones 5 and 6 below. We consider ourselves on track overall. We employ Shekhar Borkar's PI leadership, regular weekly PI and technical meetings, monthly rolling wave milestone planning meetings, semi-annual application workshops, a collaboration wiki, and central code repository to keep the team focused on priorities to achieve a successful X-Stack.

#	Due	Milestone	Lead
1	11/30/12	Architecture V2 spec & preliminary apps kernel identified for evaluation	Intel
2	3/1/13	Simulators V2 functional, tools (C + binutils) in place, IRR V1 identified	ETI, Reservoir
3	5/31/13	Selected kernels evaluated for 0(compute)	Intel
4	8/30/13	Basic timing in simulator, intelligent scheduling in Exec model, tools (LLVM, etc)	ETI, Rice, Reservoir
5	11/27/13	Selected kernels evaluated for 0(com), select apps coded with PGM system for IRR	UCSD
6	2/28/14	Architecture V2.5 spec, system evaluation of V2.0	Intel, UIUC
7	5/30/14	Simulators V2.5 functional, tools for V2.5 released	ETI, Reservoir
8	8/29/14	System evaluation of V2.5	UIUC
9	11/26/14	Arch V3.0 spec, selected apps evaluation with Exec model & PGM system for V2.5	Intel, UCSD, Rice, Reservoir
10	2/27/15	Simulators V3.0 functional, tools for V3.0 released	ETI, Reservoir
11	5/29/15	Release OCR (Open Collaboration Runtime) V1.0	Rice
12	8/28/15	Evaluation of all X-Stack technologies and report	Intel

[redacted]

Publications September 2013 – February 2014

The following were presented at the CnC'13 workshop September, 2013. This was the fifth annual CnC workshop. It was co-located with Languages and Compilers for Parallel Systems (LCPC) in Santa Clara, CA.

- "Compiler Optimization of an Application-Specific Runtime". Kathleen Knobe (Intel) and Zoran Budimlic (Rice)*.
- "The CnC tuning capability", Sanjay Chatterjee (Rice), Zoran Budimlic (Rice), Vivek Sarkar (Rice), Kathleen Knobe (Intel).
- "Automatic Selection of Distribution Functions for Distributed CnC", Kamal Sharma (Rice), Kathleen Knobe (Intel), Frank Schlimbach (Intel), Vivek Sarkar (Rice)*.
- "CnC on Open Community Runtime", Alina Sbirlea (Rice) and Zoran Budimlic (Rice).
- "Bounded Memory Scheduling of CnC Programs", Dragos Sbirlea (Rice), Zoran Budimlic (Rice) and Vivek Sarkar (Rice). *
- "CDSC-GL: A CnC-inspired Graph Language", Zoran Budimlic (Rice), Jason Cong (UCLA), Zhou Li (UCLA), Louis-Noel Pouchet (UCLA), Vivek Sarkar (Rice), Alina Sbirlea (Rice), Mo Xu (UCLA), Pen Zhang (UCLA).*
- “Implementing Asynchronous Checkpoint/Restart for CnC”, Nick Vrvilo and Vivek Sarkar (Rice University) Kath Knobe and Frank Schlimbach(Intel)
- “Automatic CnC generation from a sequential specification”, Nicolas Vasilache (Reservoir Labs, Inc.)

  Note: Asterisked (*) presentations are supportive of the Traleika Glacier X-Stack strategic aims and objectives but not directly under the statement of work.

Reservoir Labs submitted a paper for publication to PPoPP; unfortunately, this contribution was rejected. We will produce a technical report taking into account the reviewers’ comments. We circulated this publication internally to members of the OCR core team and to members from Intel.

Submitted for publication: Title: Compiler Support for Software Cache Coherence
Authors: Sanket Tavarageri, Wooil Kim, Josep Torrellas, and P Sadayappan Pacific Northwest National Labs (John Feo, Andres Marquez)

"Bounded memory scheduling of dynamic task graphs”. Dragos Sbirlea, Zoran Budimlić, Vivek Sarkar. Submitted to IPDPS 2014.

St. John, T. et al.: “T2: ASAFESSS: A Scheduler-Driven Adaptive Framework for Extreme Scale Software Stacks”, 4th International Workshop on Adaptive Self-tuning Computing Systems 2014, Vienna Austria. (Best paper award).

Marquez, A. et.al, “ACDT: Architected Composite Data Types Trading-in Unfettered Data Access for Improved Execution,” submitted to the 23rd International ACM symposium on High Performance Parallel and Distributed Computing 2014, Vancouver Canada.

Subject Inventions

None.