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

Brandywine Team:

Rishi Khan, ET International, Inc.
Benoit Meister, Reservoir Labs, Inc.
David Padua, UIUC
John Feo, PNNL

http://www.etinternational.com/xstack

Background

In the Brandywine X-Stack project, we will address the issues of **scalability, programmability, portability, resilience, energy efficiency**, and **interoperability** through the use of a novel execution model (*codelet execution model*), a runtime system supporting the execution model (*SWARM*), a language/compiler to express parallelism and locality (*SCALE*), a higher level compiler to automatically parallelize and localize (*R-Stream*), a library of objects and algorithms to represent arrays and parallel operations (*Hierarchical Tiled Arrays [HTA]*), and new programming constructs for resilience (*Containment Domains [CD]*) and energy efficiency (*Rescinded Primitive Data Types [RPDT]*). This work will be guided by applications from the co-design centers and facilitated through the efforts of PNNL. We will also prominently investigate NWChem codes, derived from DOE's premier computational chemistry software.

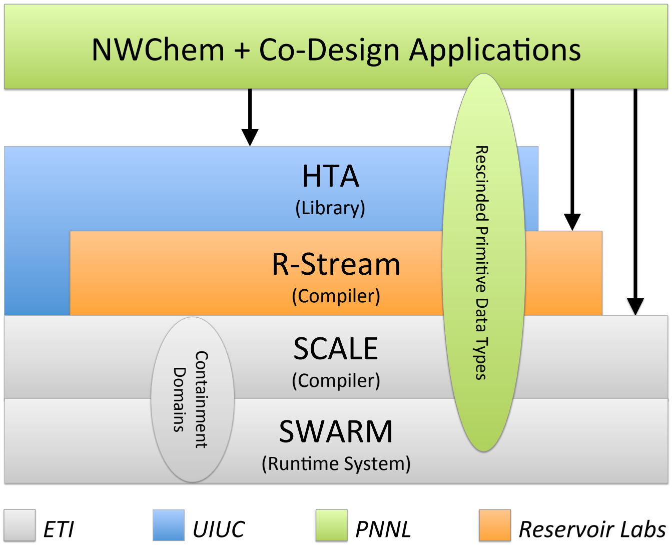

Figure 1: Brandywine Team Software Stack

Impact

In answering the key questions, the Brandywine Team project will create impact in the following areas:

- **Scalability**: We will leverage SWARM to express and exploit new forms of parallelism currently hidden by over-synchronization, language expressibility limitations, and static binding constraints. We will study novel ways of seamlessly scheduling tasks across the system as if it were one system rather than hundreds of disparate pieces. We will also study mechanisms of symmetric access semantics across heterogeneous devices.
- **Locality**: Recognizing that data movement will be a key contributor to energy, we will provide mechanisms to express locality as a first-class citizen. We will focus on mechanisms to expose the memory hierarchies to the compiler (and programmer) and provide data types and accesses such that the programmer can think of the system as one system than hundreds of disparate memories.
- **Programmability**: We will create easier ways of expressing asynchrony thereby enabling programmers to write more scalable programs. We will also use R-Stream to automatically extract parallelism and locality from common idioms. We will provide data types and algorithms that provide high-level representations of arrays mapped to the memory hierarchy and the algorithm hierarchy for automatic parallelization and data placement.
- **Applications**: We will support the NWChem application and provide others domain expertise in this co-design application. We will use NWChem as our exemplar application to optimize through the Brandywine team software stack. We will also use the other co-design applications to guide our work, but the focus will be on NWChem.
- **Portability**: We will demonstrate a software stack that is portable to multiple architectures provided a C compiler. SWARM will support a platform abstraction layer, which will allow it to operate on multiple heterogeneous architectures. We will work with Xpress on the XPI interface to show application portability between runtime systems.
- **Energy Efficiency**: We will describe mechanisms to reduce energy consumption by more intelligently collocating execution and data, dynamically load

balance execution based on resource availability and dynamically scale resources based on load. We will provide new programming constructs (Rescinded Primitive Data Types) that allow compressed data formats at higher memory levels to minimize data transfer costs.

- **Resilience**: We will demonstrate a hierarchical, distributed asynchronous, multi-grained checkpoint / verify / restart mechanism (leveraging containment domains) that allows for graceful degradation in the face of exascale-level faults and a framework for software validation of soft faults. Containment domains and their extensions will be integrated into the SWARM runtime system and SCALE compiler.
- **Interoperability**: We will work with XPRESS on XPI interoperability with legacy codes such that all X-Stack runtime systems and all X-Stack applications can benefit from Evolutionary/Revolutionary runtime system interoperability.

Key Questions

To achieve the objectives and impact, this project will conduct research to answer these questions:

Q1. **How can work be scheduled** across cores, sockets, nodes, and peripherals to enable seamless task sharing and stealing, multiple ways of scheduling in space and time?

Q2. **How can data be managed** across memory hierarchies that are loosely bound to compute resources to enable data locality-aware scheduling?

Q3. **How can loops be transformed** into tiles with schedules and placements to ensure optimal execution?

Q4. **What new constructs can be added to high-level language** to express locality and scheduling in time and space?

Q5. What **compiler strategies** will be implemented to translate high-level languages into efficient SWARM code?

Q6. **How can key co-design center kernels be ported** to the X-Stack research platform to test and direct research efforts?

Q7. How can the X-Stack software stack be **designed for portability across exascale systems**?

Q8. **How can fault tolerance be integrated into the X-Stack software stack** with user specified checkpoints?

Q9. How can the X-Stack software stack (high level interpreter, compiler, and runtime system) **integratively support energy efficiency**?

Q10. How will the X-Stack software stack **interoperate with legacy code**?

Collaborations

XPRESS: We will collaborate on XPI definitions such that application/compiler programmers can leverage multiple X-Stack runtimes. MPI interoperability may be incorporated in XPI.

LBL: We will work on understanding and incorporating containment domains into the co-design applications with the LBL team. We will also leverage LBL for containment domain integration into the runtime system.

Co-Design centers: We will collaborate with the co-design center post-docs to understand the parallelism and locality in the co-design applications.

Traleika Glacier: We will collaborate with Intel using the Runnemede simulator for power and next-gen architecture studies.

Your team here: We'd love to collaborate with you where our goals are synergistic!

ETI

Reservoir Labs

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Pacific Northwest
NATIONAL LABORATORY