E.T. International, Inc.

Pacific Northwest
NATIONAL LABORATORY

Reservoir Labs

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

DynAX X-Stack Project Overview

Guang R. Gao

DynAX Principle Investigator

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

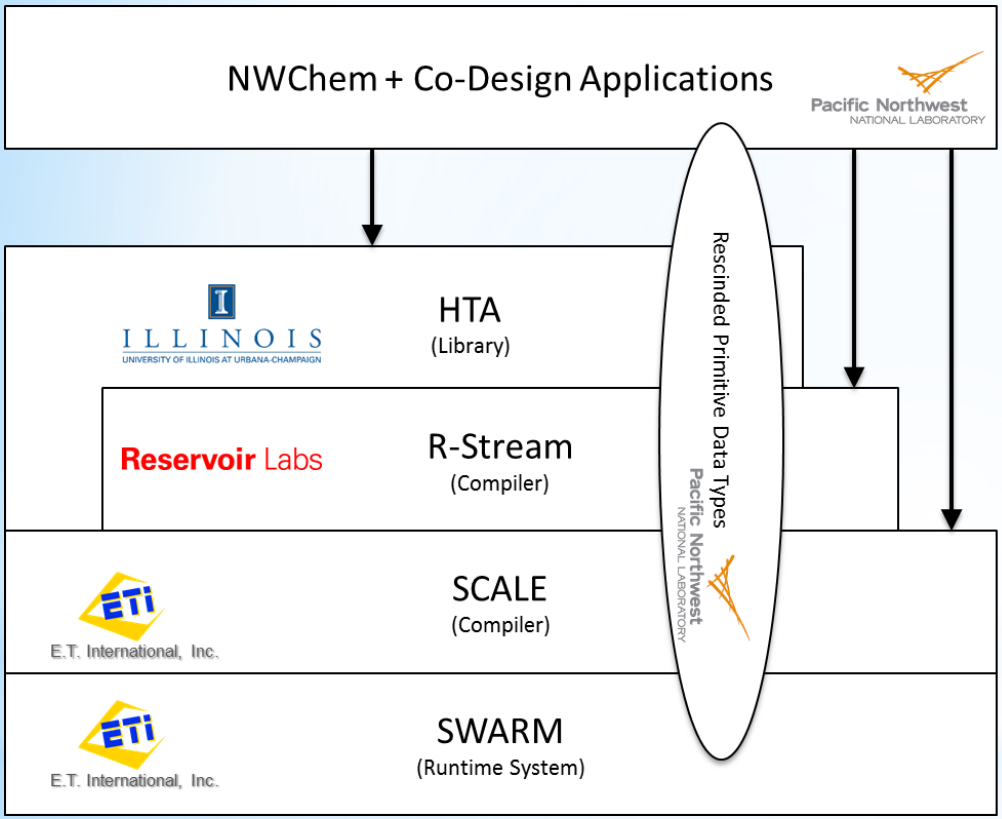
E.T. International, Inc. Reservoir Labs ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN Pacific Northwest NATIONAL LABORATORY

Acknowledgements

- Co-PIs:
  - Benoit Meister (Reservoir)
  - David Padua (Univ. Illinois)
  - John Feo (PNNL)
- Other team members:
  - ETI: Mark Glines, Chris Lauderdale, Sam Kaplan, etc.
  - Reservoir: Rich Lethin
  - Univ. Illinois: Adam Smith, Chih-Chieh Yang
  - PNNL: Andres Marquez, Joseph Mazano
- DOE
  - Sonia Sachs, Bill Harrod
- Other Collaborators
  - US: UDEL/CAPSL, Intel, IBM, DOE Labs
  - International: EU/Teraflux, Others
- Collaborators and Consultants: several are in the audience
- Jaime Arteaga for helping preparing the presentation

DynAX X-Stack

Software Stack



Overview

Challenges:

- Scalability
- Locality
- Programmability
- Portability
- Energy efficiency
- Resiliency
- Interoperability
- Co-Design Applications

Approach:

- Enhance scalability and locality with codelet execution model
- Address locality using data/memory-aware techniques
- Enhance productivity, portability and energy efficiency with novel compilation and runtime technology
- Focus on X-Stack co-design applications

DynAX X-Stack: Progress/Results

- **Applications** (On Track)
  - Completed codelet design and optimization of NWChem SCF module
  - Completed codelet design and optimization of Cholesky Decomposition
  - Development of translator to convert NWChem TCE equations to Codelet source code
- **High-Level Programming Model for Codelets: HTA** (On Track)
  - Developed HTA as a C library and a compiler that compiles HTA programs into ETI SCALE.
  - Preliminary performance evaluation conducted on shared memory machine using NAS Parallel Benchmarks implemented in HTA notation
- **R-Stream Compiler Targeting Codelets** (On Track)
  - Auto-parallelization to SWARM **implemented** and work well
  - Optimally scalable synchronization for codelets implemented and tested for feasibility
  - Communication layer optimization scheme in place
- **Adaptive SWARM Runtime for Performance/Energy Optimization** (On Track)
  - Power-Efficient Data Abstraction Layer (PEDAL) designed and tested for feasibility
  - By Group Locality (GL) designed and tested for feasibility
  - Compiler/Runtime Toolchain on SWARM in place

Sponsor's Instructions

- One slide with the vision of your project, showing how application developers use the technologies that you are researching and developing in order to write new applications and/or migrate existing ones to exascale platforms. Please include a visual representations of the vision.

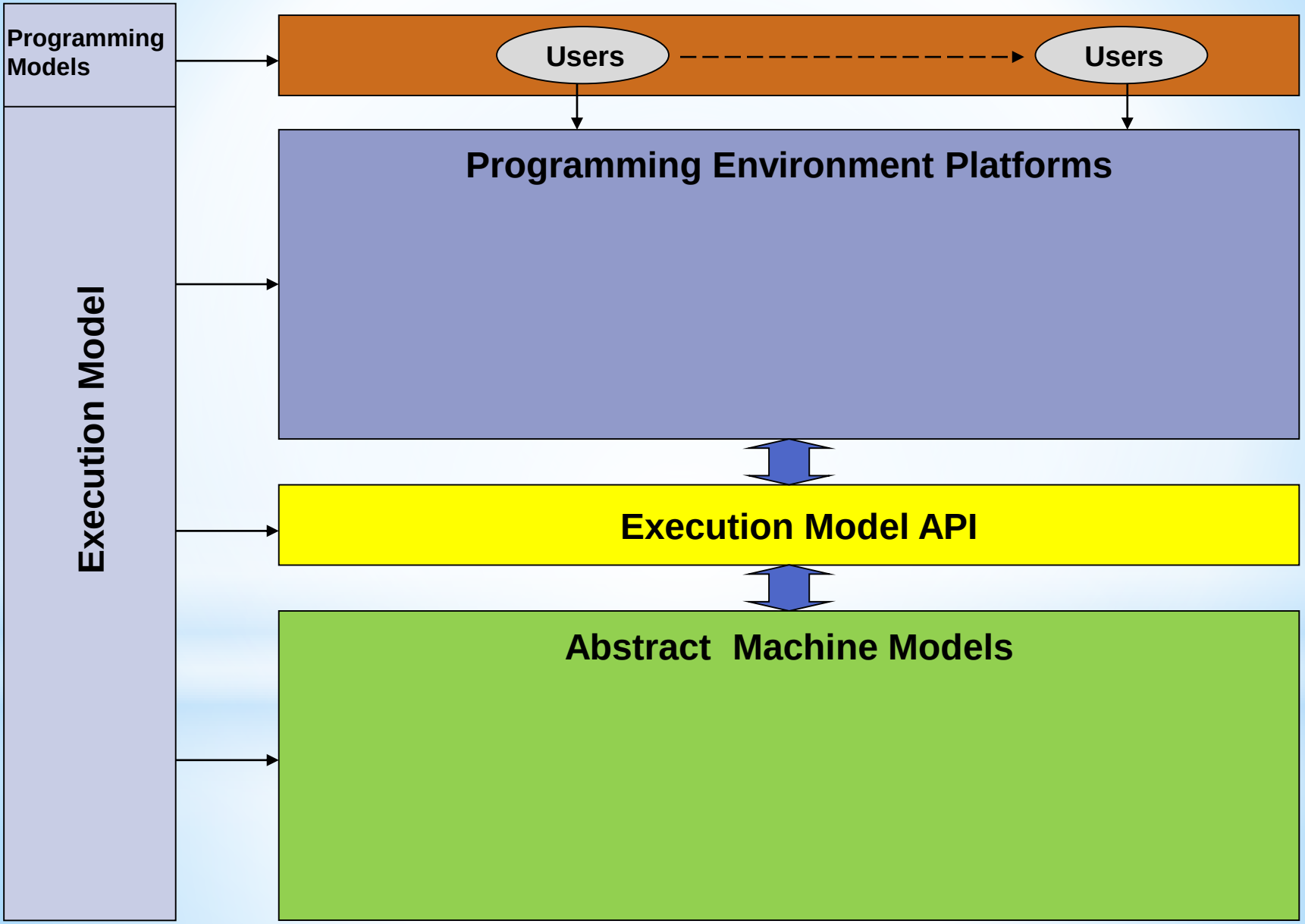
- One "use case" slide, showing how at least one application benefits from the technologies under R&D.

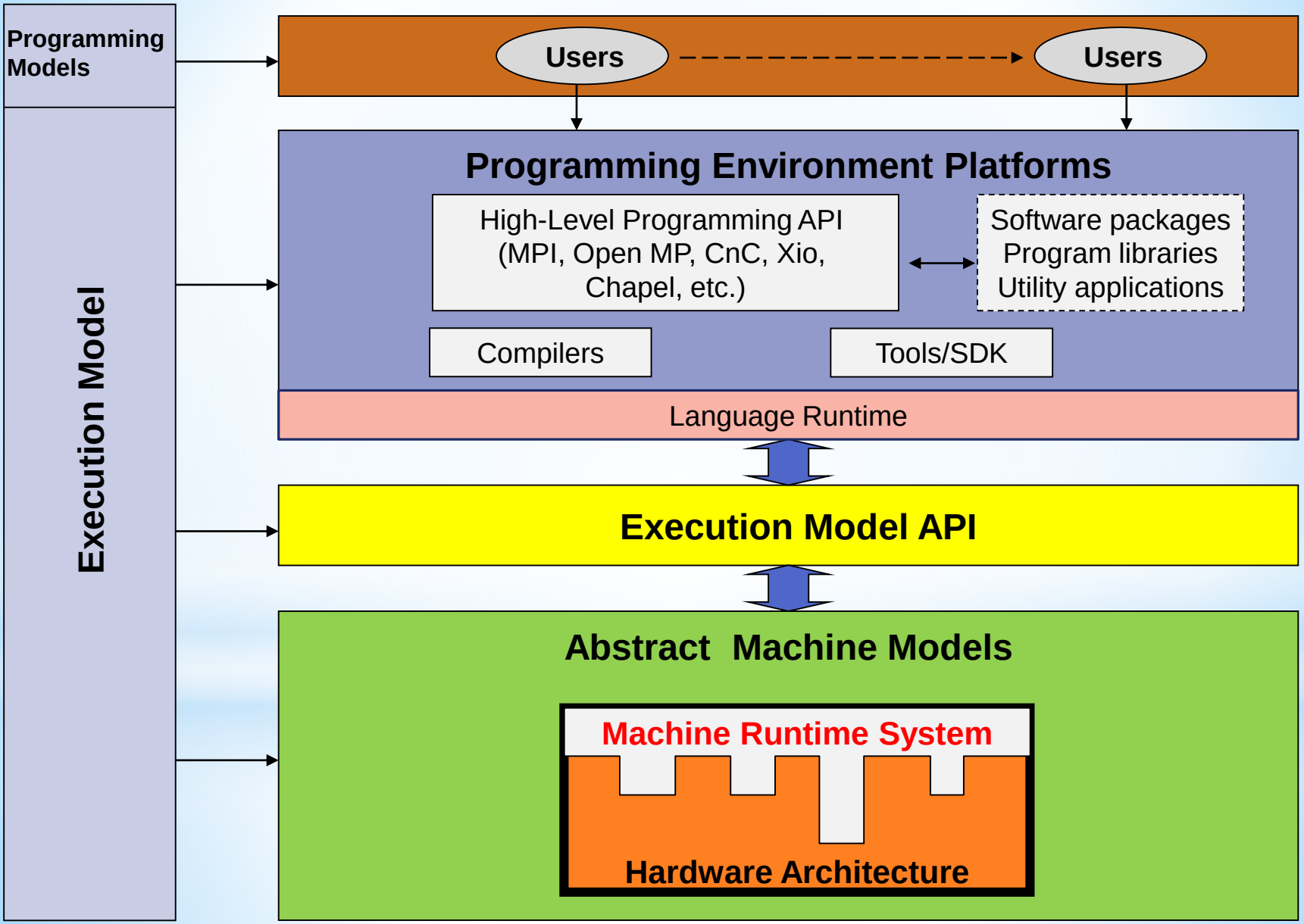
- One slide showing the status of your project, overlayed with your vision slide.

- One slide comparing your new technologies with the current state-of-the-art, particularly MPI+OpenMP programming. We need to crisply identify advantages over current state-of-the-art.

Outline

- **DynAX Project Vision Overview:**
  - ***DynAX Position: parallel execution/abstract machine model***
  - **DynAX Technology: The Codelet Model and SWARM Abstract Machine**
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**


Programming Models
Execution Model
Users
Users
Programming Environment Platforms
Execution Model API
Abstract Machine Models

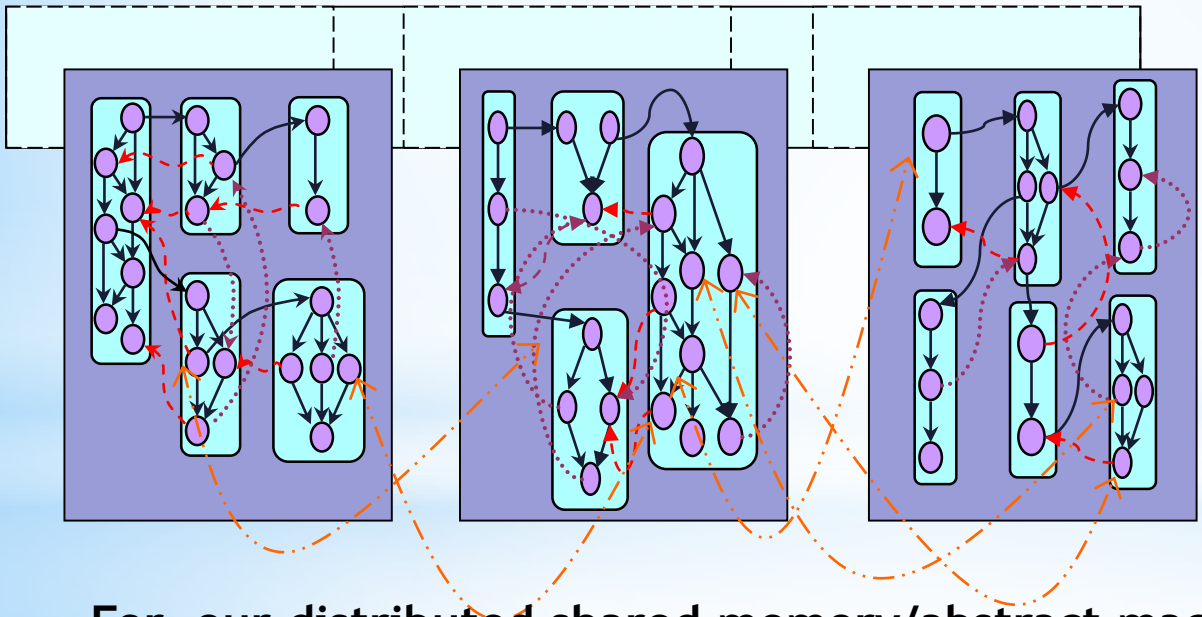

Programming Models
Execution Model
Users
Users
Programming Environment Platforms
High-Level Programming API (MPI, Open MP, CnC, Xio, Chapel, etc.)
Software packages
Program libraries
Utility applications
Compilers
Tools/SDK
Language Runtime
Execution Model API
Abstract Machine Models
Machine Runtime System
Hardware Architecture

Outline

- **DynAX Project Vision Overview:**
  - **DynAX Position: parallel execution/abstract machine model**
  - ***DynAX Technology: The Codelet Model and SWARM Abstract Machine***
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**

A Dynamic Multithreaded Execution Model and Abstract Machine

Global Distributed Shared Address Space
Tree of "trunks"
"write-once semantics" + LC model



Large-Grain Tasks (**LGT**)

Small-Grain Tasks (**SGT**)
Threaded Procedures

Tiny-Grain Tasks(TGT)
Codelets

Invoke an SGT/Sync a TGT within same SGT

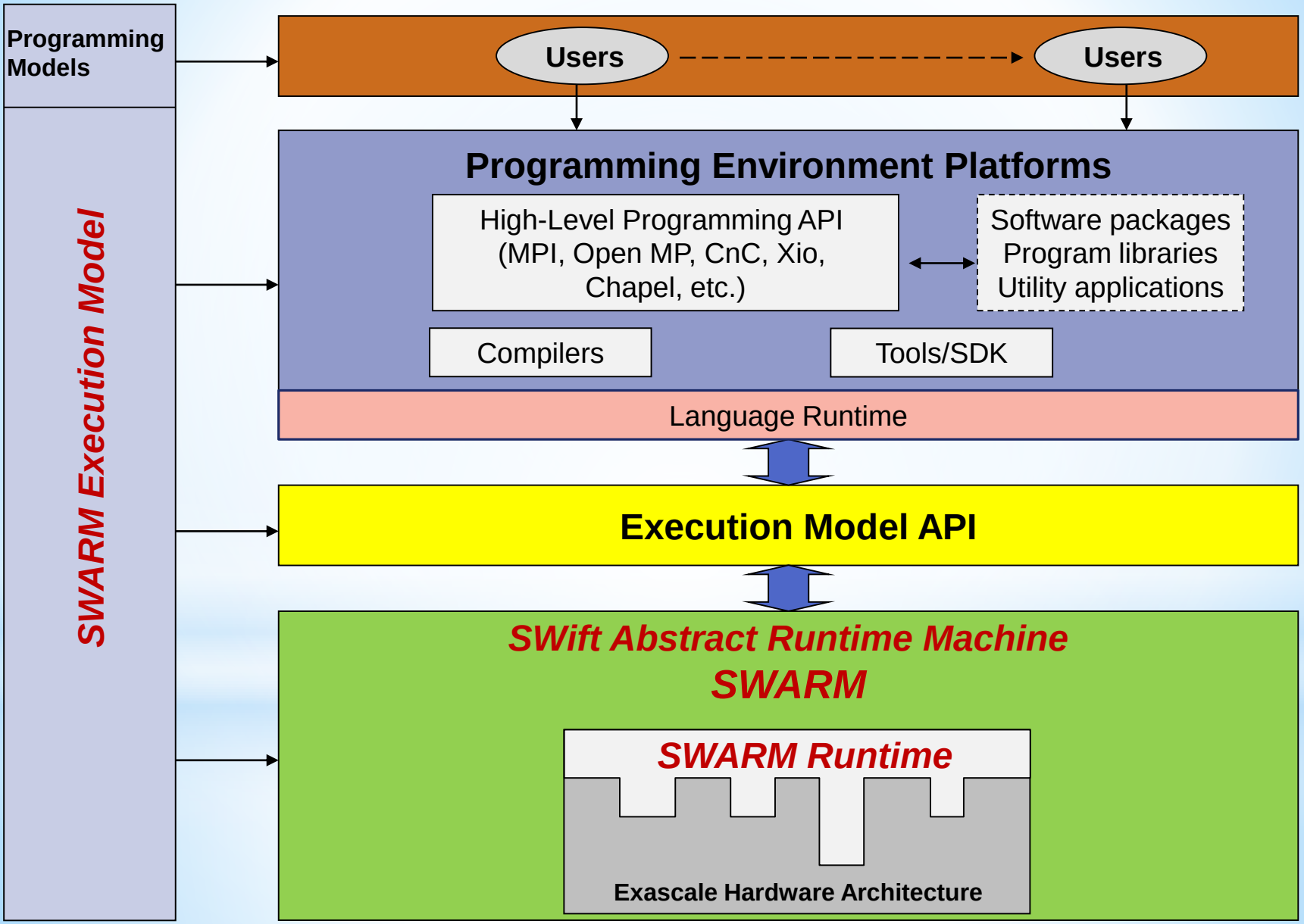
SYNC ops

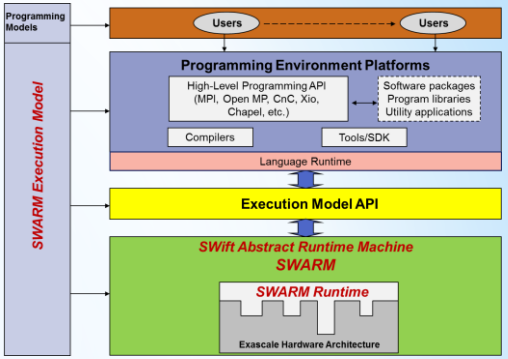
Data-SYNC ops

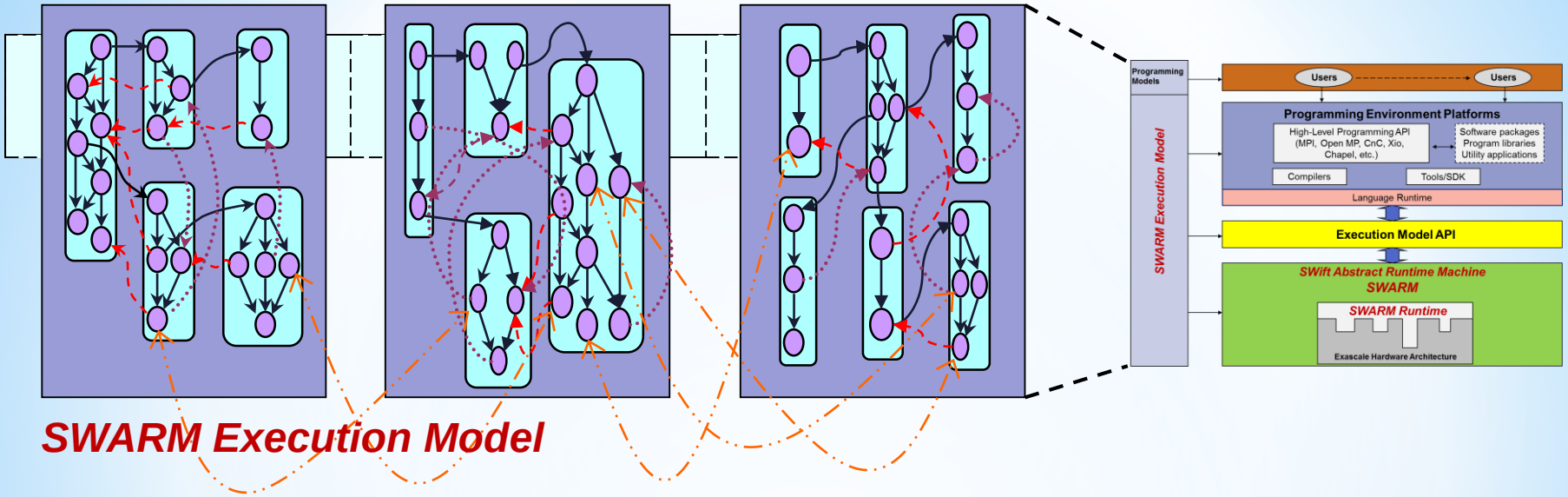
Inter-LGT Communication & Synchronization

Note: the lower two levels of the tasks are *fine-grain*

For our distributed shared memory/abstract machine model - see "Massive Multi-Core System and Virtual Memory", [Gao et.al 2011]Dennis and Gao [2011, 2014].


Programming Models
SWARM Execution Model
Users
Users
Programming Environment Platforms
High-Level Programming API (MPI, Open MP, CnC, Xio, Chapel, etc.)
Software packages
Program libraries
Utility applications
Compilers
Tools/SDK
Language Runtime
Execution Model API
SWift Abstract Runtime Machine
SWARM
SWARM Runtime
Exascale Hardware Architecture


Programming Models
SWARM Execution Model
Users
Users
Programming Environment Platforms
High-Level Programming API (MPI, Open MP, CnC, Xio, Chapel, etc.)
Software packages
Program libraries
Utility applications
Compilers
Tools/SDK
Language Runtime
Execution Model API
SWift Abstract Runtime Machine
SWARM
SWARM Runtime
Exascale Hardware Architecture



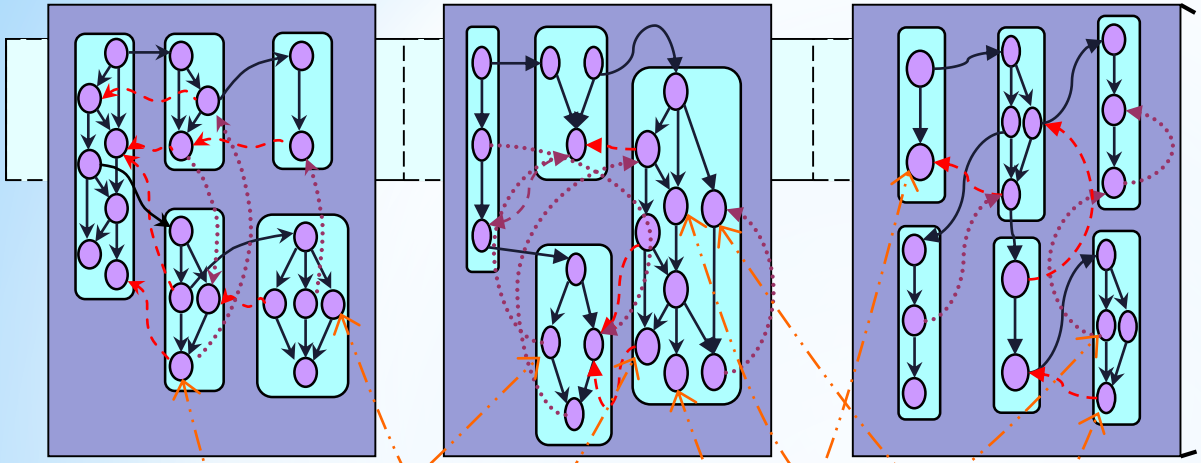
SWARM Execution Model

E.T. International, Inc.

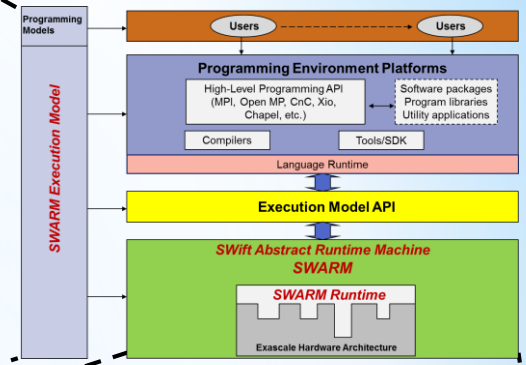
Reservoir Labs

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

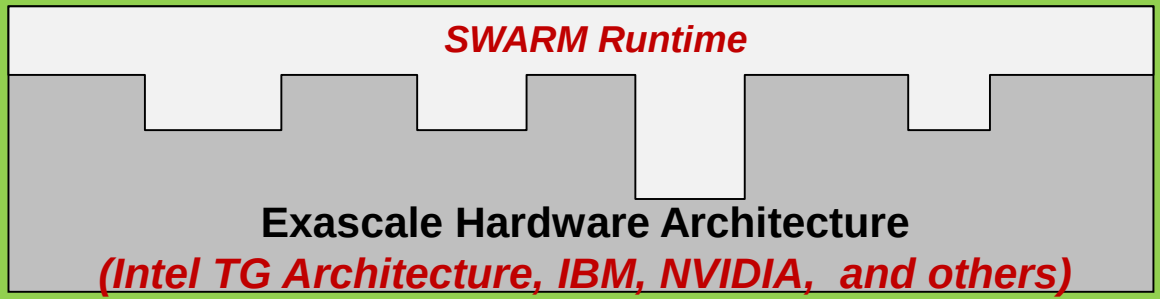
Pacific Northwest
NATIONAL LABORATORY



SWARM Execution Model



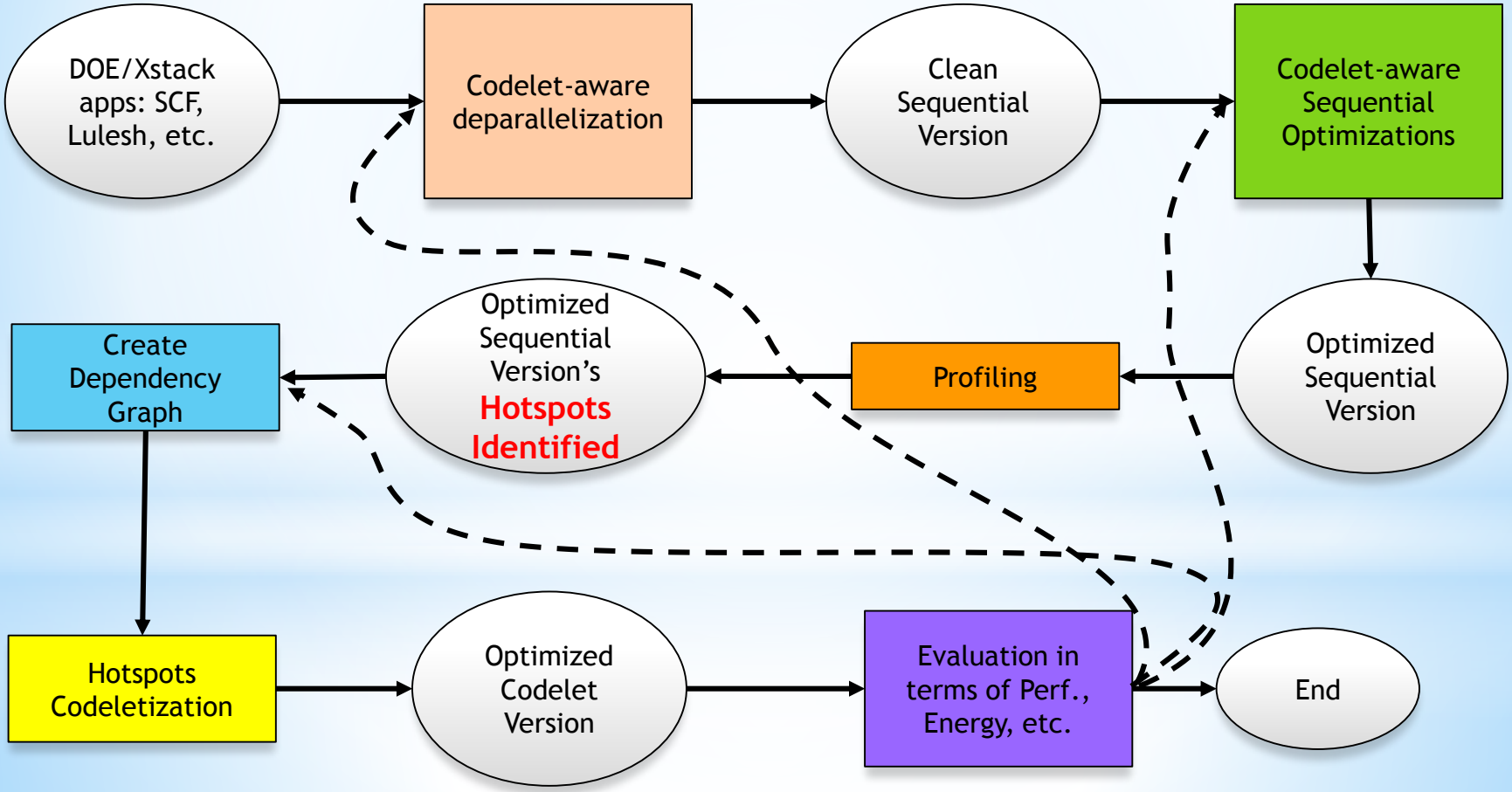
SWift Abstract Runtime Machine
SWARM



Outline

- **DynAX Project Vision Overview**:
  - DynAX Position: parallel execution/abstract machine model
  - DynAX Technology: The Codelet Model and SWARM Abstract Machine
  - ***DynAX Path to Productivity:*** *an application-focused approach to understanding scalability problems, and general solutions which many applications can use*
- A Use Case
- DynAX Project Status Report: Progress and Impact
- Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)
- Summary and Future Work (1-2 slides)

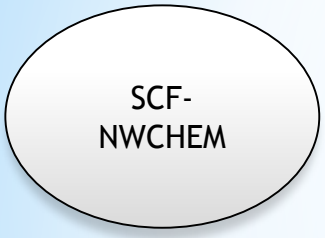
Toward A Generic Re-Engineering Procedure Migration: *MPI/OpenMP to Codelets*



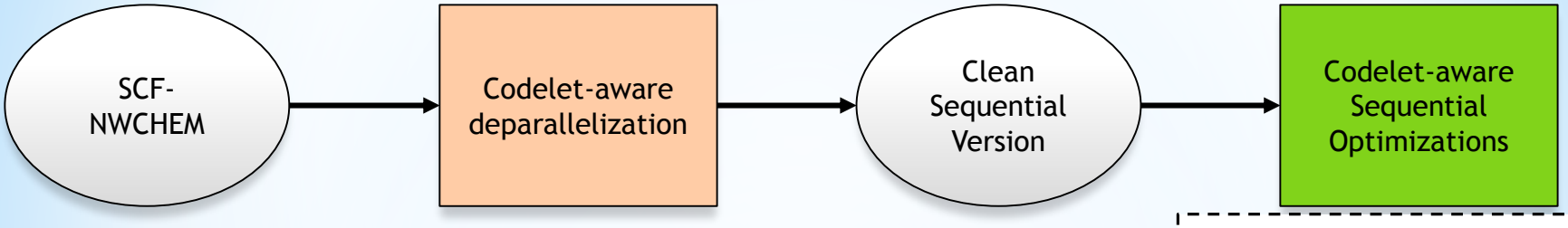
Outline

- **DynAX Project Vision Overview:**
  - **DynAx Position: parallel execution/abstract machine model**
  - **DynAX Technology: The Codelet Model and SWARM Abstract Machine**
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**

NWChem Self-Consistent Field (SCF) to Codelets



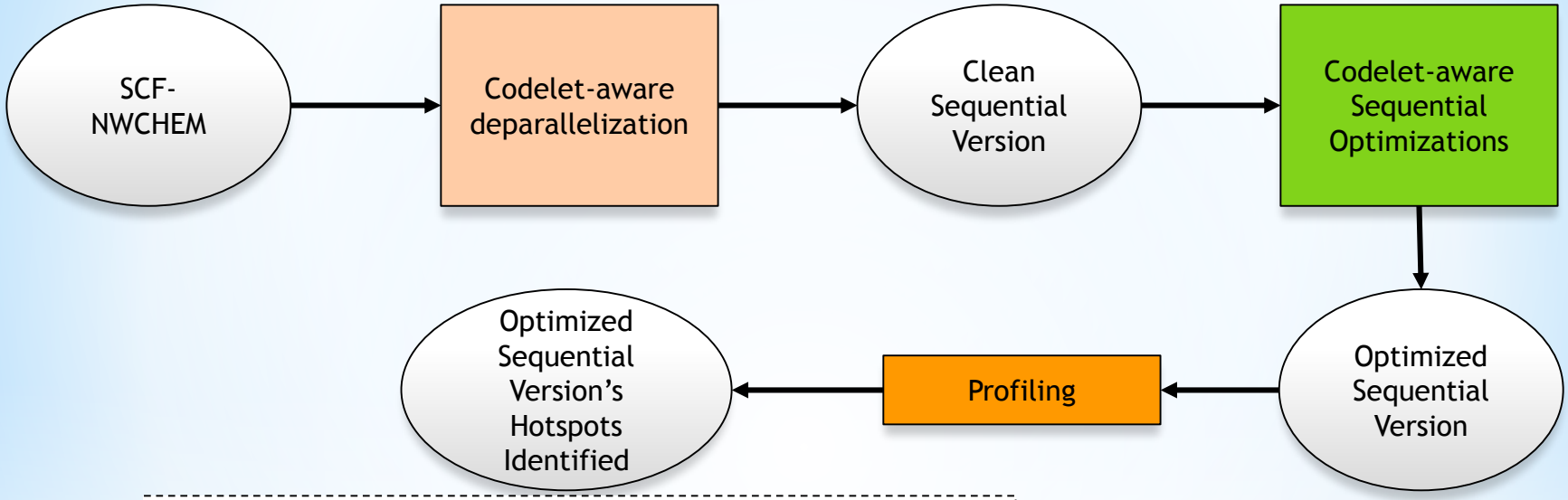
NWChem Self-Consistent Field (SCF) to Codelets

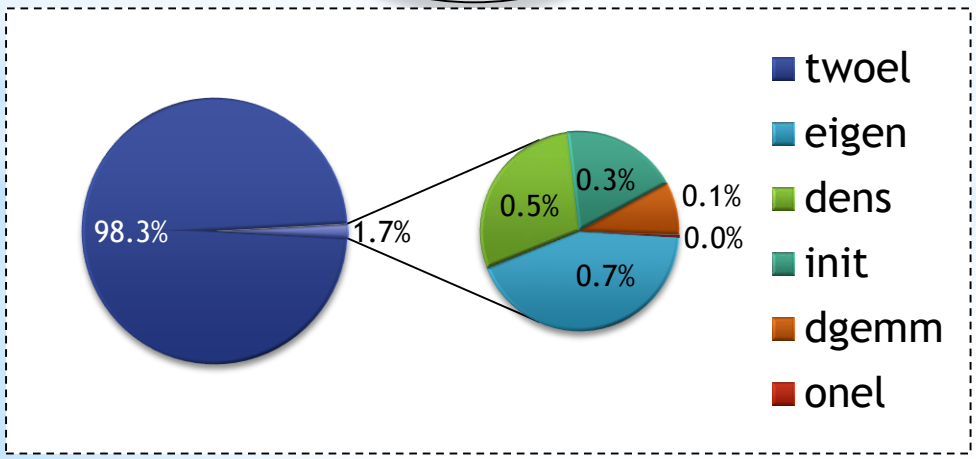


- Symmetry of g()
- Lookup Tables in g()
- BLAS/LAPACK
- Symmetry of fock and dens matrices
- Cache g() results

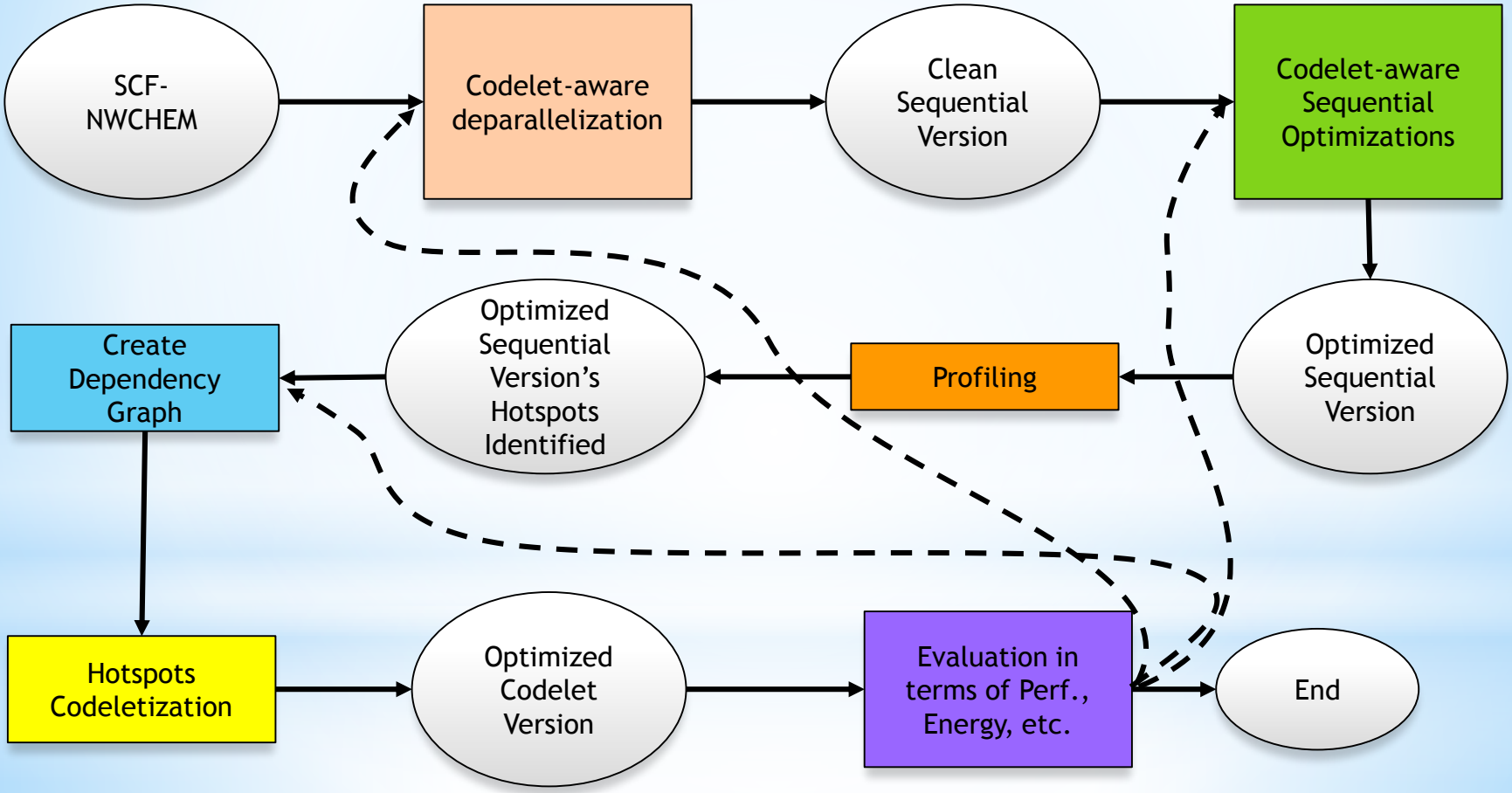
E.T. International, Inc. Reservoir Labs ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN Pacific Northwest NATIONAL LABORATORY

NWChem Self-Consistent Field (SCF) to Codelets





NWChem Self-Consistent Field (SCF) to Codelets



More About The Case Study

- The migration path/principle demonstrated should have a broad applicability to other DOE applications.
- It is important to realize that this path stressed the following principles:
  - Component reusability
  - Plug-and-Play
  - Portability
  - Interface to modern compiler technology
- Toward a smooth path to *interoperability* and *co-existensibility* with state-of-the-art MPI/OpenMP programming execution environment

Outline

- **DynAX Project Vision Overview:**
  - **DynAx Position: parallel execution/abstract machine model**
  - **DynAX Technology: The Codelet Model and SWARM Abstract Machine**
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**

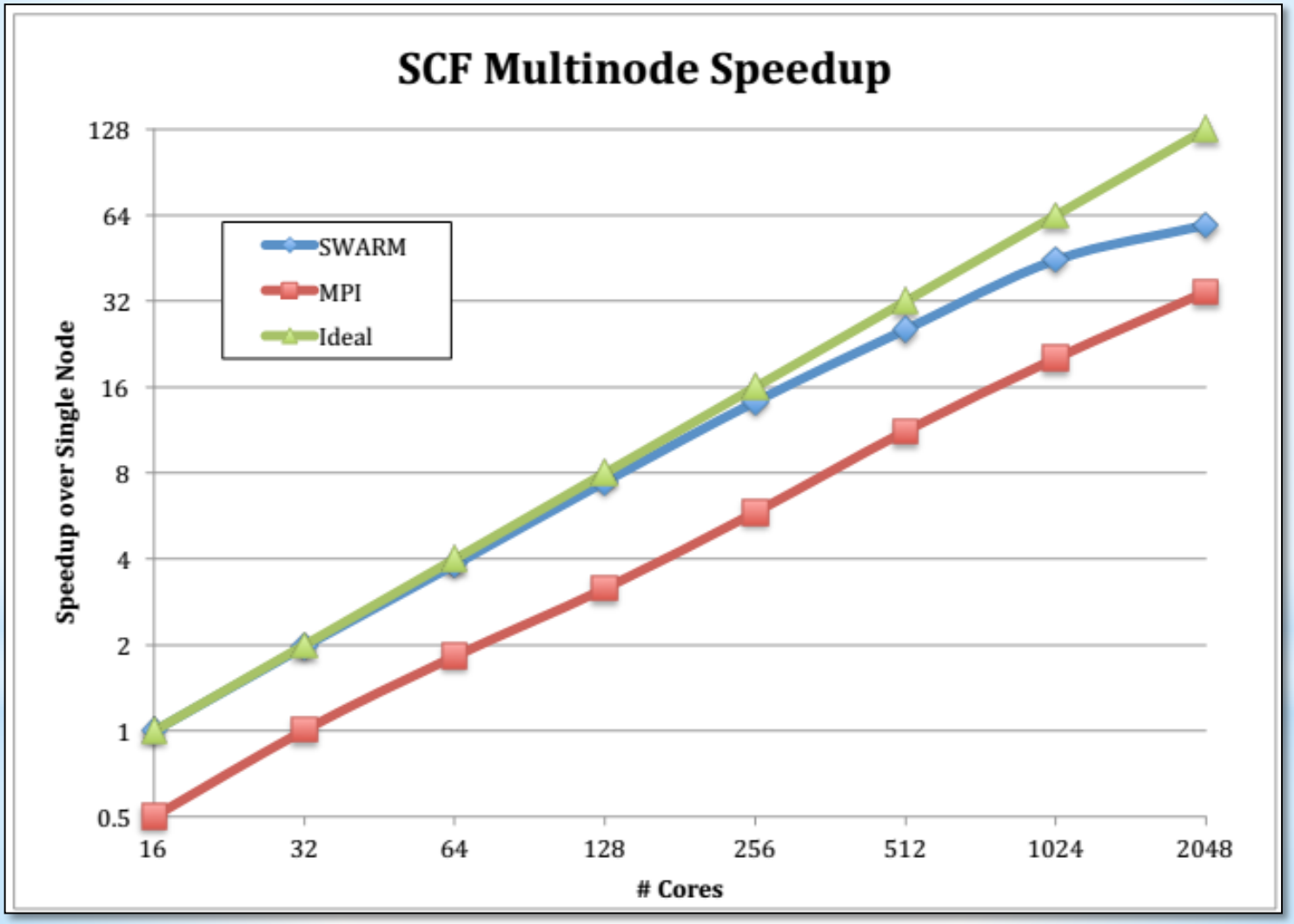
DynAX X-Stack: Impact

- General re-engineering method established and demonstrated successfully mapping DOE applications under codelet execution model and SWARM runtime
- Programmability & Productivity: Enabling the conjunction of high-level programming in HTA with low-level parallelization and optimization through R-Stream and SWARM/SCALE.
- Automatic communication reduction and avoidance through the use of smart abstract communication layers (PEDAL, Virtual DMAs) working with adaptive SWARM runtime
- DynAX project has already produces positive influence on other X-Stack runtime design (e.g., cross-pollination with OCR)
- Demonstrate the value of participation by US based small businesses (like Reservoir and ETI) in technology innovation and commercialization (like successful SBIR)

Outline

- **DynAX Project Vision Overview:**
  - **DynAx Position: parallel execution/abstract machine model**
  - **DynAX Technology :The Codelet Model and SWARM Abstract Machine**
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**

Multi Node Parallelization of twoel()



Source code available at https://xstackwiki.modelado.org/DynAX

Dynax Technology vs. MPI/OpenMP – Our Position

- The Codelet model based technology have unique features that differentiate it from the BSP model (MPI/OpenMP)
- These features can and should be productively exploited in the Exa-Scale landscape to reach certain goals beyond that are reachable with BSP model alone.
- Codelet runtime can co-exist with MPI/OpenMP runtime
- We have demonstrate a migration path from MPI/OpenMP programs to exploit the codelet model

Outline

- **DynAX Project Vision Overview:**
  - **DynAx Position: parallel execution/abstract machine model**
  - **DynAX Technology: The Codelet Model and SWARM Abstract Machine**
  - **DynAX Path to Productivity:** an application-focused approach to understanding scalability problems, and general solutions which many applications can use
- **A Use Case**
- **DynAX Project Status Report: Progress and Impact**
- **Comparison with Current State-of-The-Art (e.g. MPI+OpenMP)**
- **Summary and Future Work (1-2 slides)**

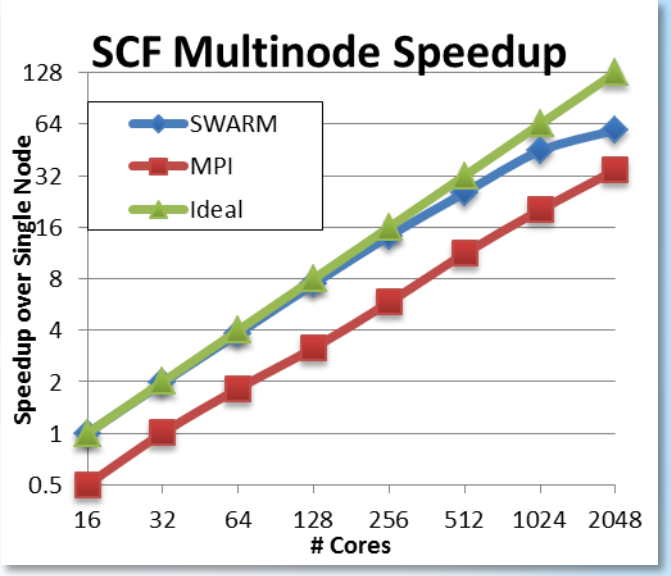
Summary and Questions

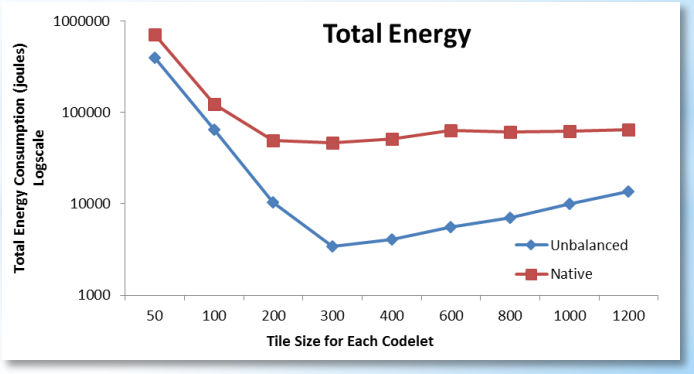
Any Questions?

E.T. International, Inc. | Reservoir Labs | ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN | Pacific Northwest NATIONAL LABORATORY

DynAX: Addressing Exa-Scale Application Challenges - Towards A Dynamic/Adaptive Runtime Machine

- Problem
  - Applications for Exascale are facing scalability and productivity challenges
- Solution
  - Enhance scalability with codelet execution model
  - Enhance productivity and efficiency with novel compilation and runtime technology
  - Focus on X-Stack co-design applications
- Accomplishments
  - Demonstrate increased performance and efficiency for SCF co-design app
  - Show greater power efficiency and performance using data/memory-aware techniques
- Impact
  - General re-engineering methods to apply codelet execution model to more relevant applications
  - Codelet ideas have positive influence on other runtime design (e.g., cross-pollination with OCR)
  - A DOE SBIR phase-I award for exploring commercial impact





Boeing's BCSSTK36 Dataset SWARM Cholesky Factorization native (red) versus ACDT (blue)