

A Million Way Parallelism ----

**Issues and Challenges to
New Generation Software**

Guang R. Gao

Computer Arch. and Parallel System Laboratory

University of Delaware

PF System Challenges ?

- **Challenges**
 - Huge variations of latencies in deep/complex memory hierarchy (*$10^4 \sim 10^6$ order*)
 - Complex Memory/resource and interconnect hierarchy

PF-scale Application Challenges: *Complex and Dynamic*

- Data/control dependence: *irregular*
- Data access pattern: *irregular*
- Load evolution: *irregular*
- ***Observation I***: Challenges of “**Bad Latencies**” for **class B** Peta-scale applications !!

Challenges to One-Million Way Parallelism Exploitation?

Observation II:

- It is simply **too hard** to “fabricate” a million-way *coarse-grain* parallelism from **class B** Peta-scale apps!

Here Comes the Surprise!

*Observation III: It is not necessarily too hard to “generate” and “program” 1-million-way ***fine-grain*** parallelism (regular or irregular)!*

How to Make 1-Million Way Fine-Grain Parallelism Work?

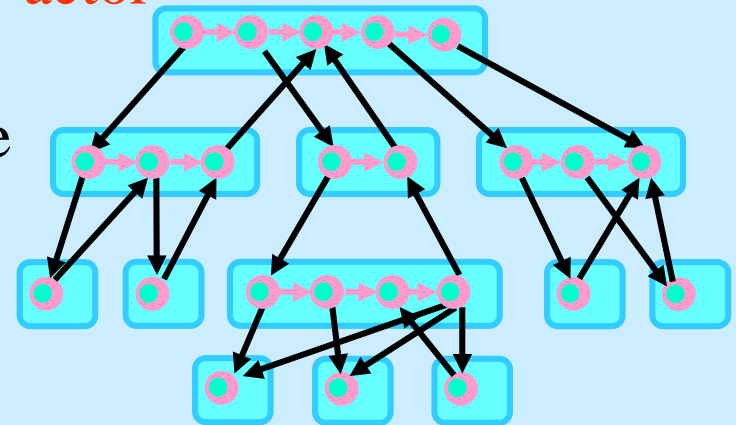
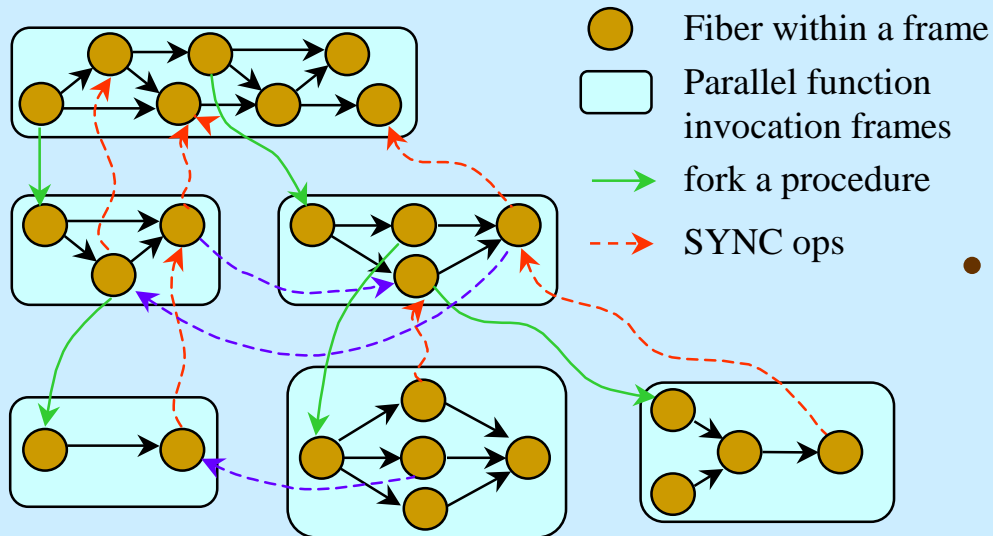
Observation IV: A Sound Program Execution Model Integrating Fine-grain Parallelism is Essential !

Call it a “**paradigm shift**” ? —Fine!

Dynamic load balancing, split-phase transactions, etc.

An Execution Model with Fine-Grain Threads

- A *fiber* acts like an **extended dataflow actor**
- An *enabled fiber* may be selected for execution when the required hardware resource has been allocated



- **Old barriers/rules between apps, compilers, RTS, OS, hardware/arch do not work well here !**

Note: The role of fiber ● under the left model

Experiments with Fine-grain Thread Models (1988-1999)

- EARTH Benchmark Suite (**EBS**)
- Adaptive mesh computation
- Sparse matrix computation
- Crack Propagation
- HTMT Application Benchmarks

What Fine-grain Parallel Execution Model Can Offer?

- *Scalability*

- *Smoothability*

($P(P_{inf})/P_{inf}$ -- see [TheGaoHen92])

- *Robustness*

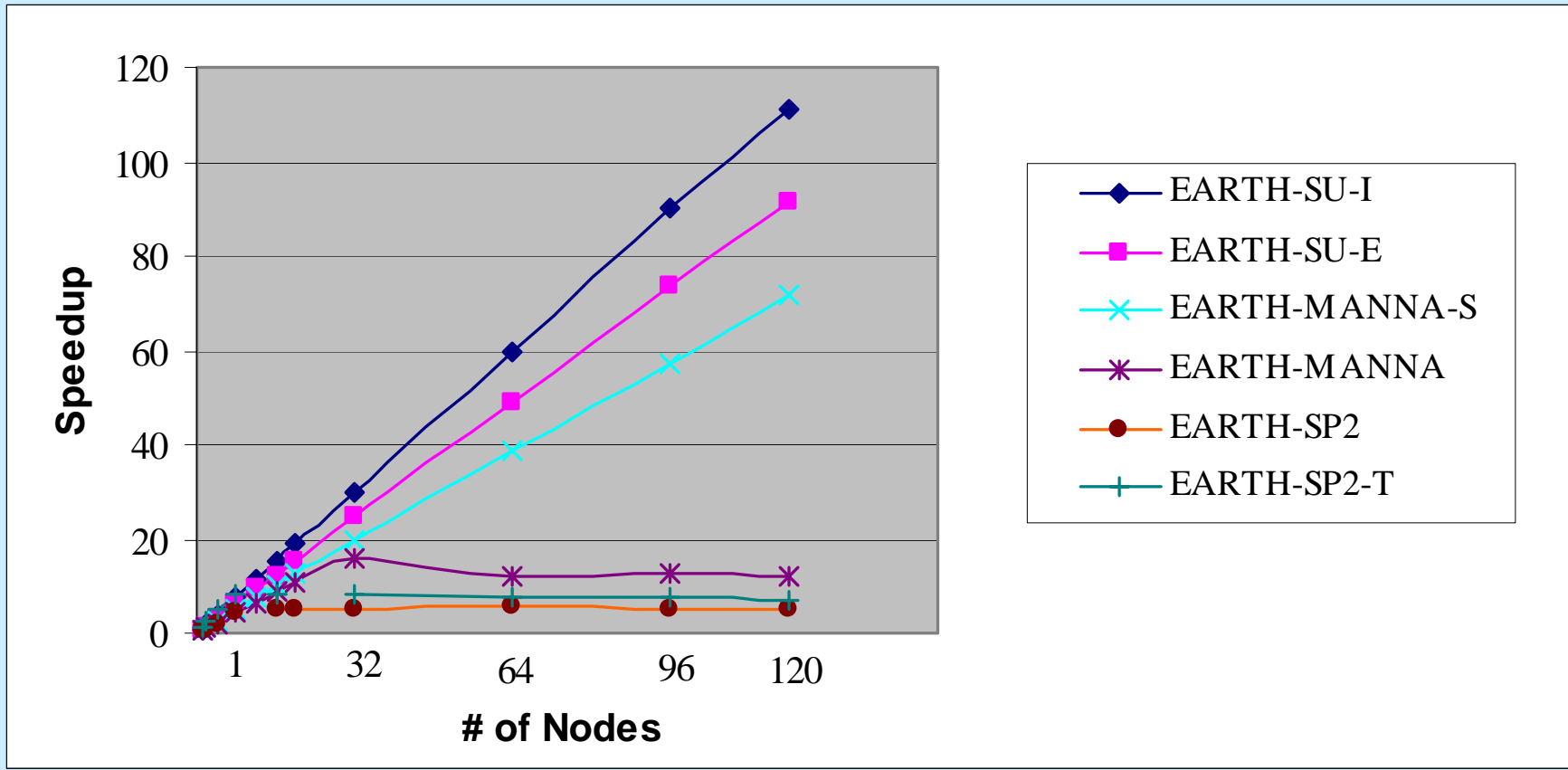
(see [Theobald99])

- *Programmability*

Performance of EARTH-MANNA on N-Queens(12)

Parafin
protein
folding

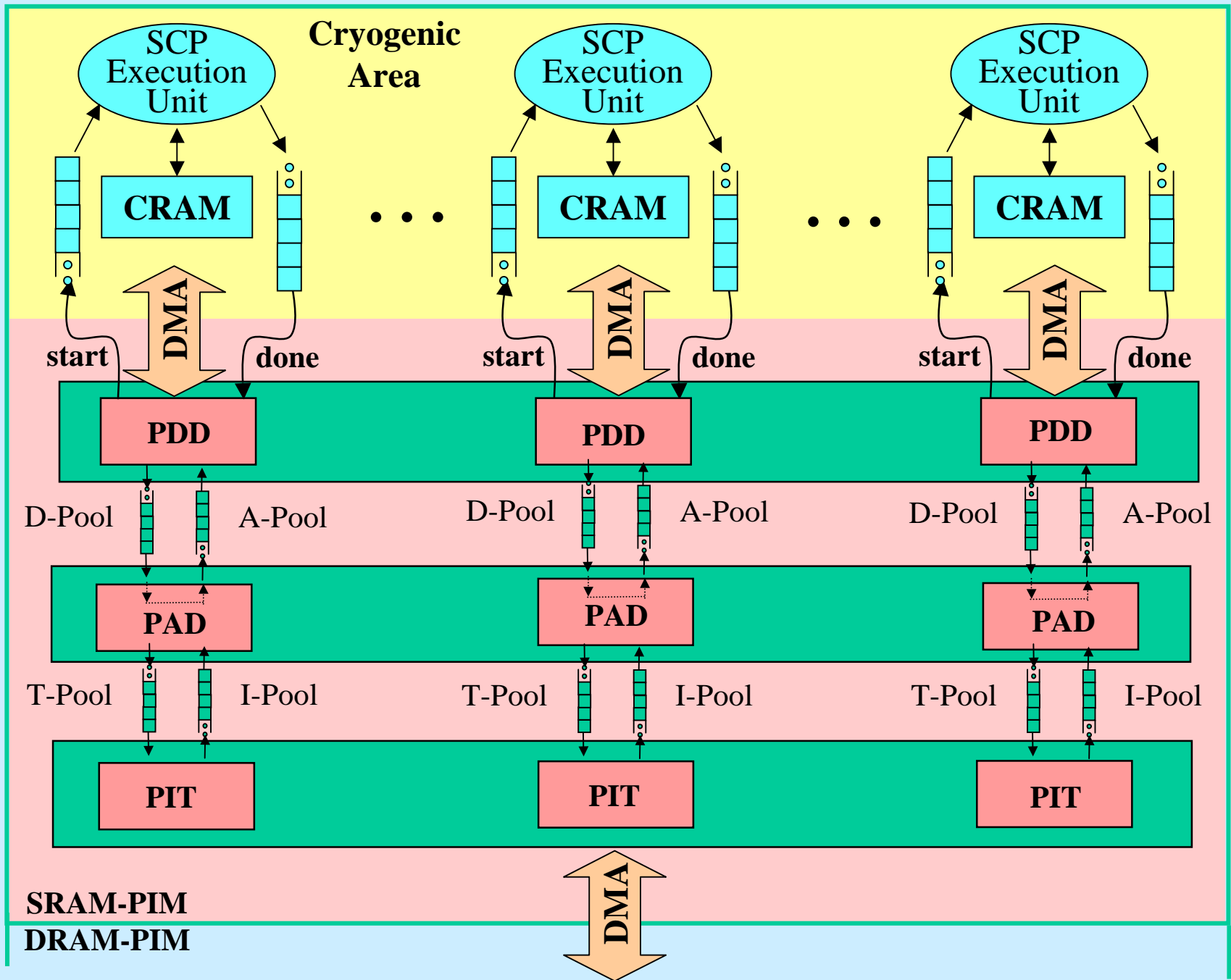
1,637,099 tokens are generated!



Open Research Highlights of a New Program Execution Model

- **Thread Percolation Model:** how to manage resource for 1-million way fine-grain parallelism ?
- **Memory Model:**
 - replication vs. consistency
 - efficiency vs. programmability
- **Load/Data Mobility and Balancing**
- **And their hardware support**

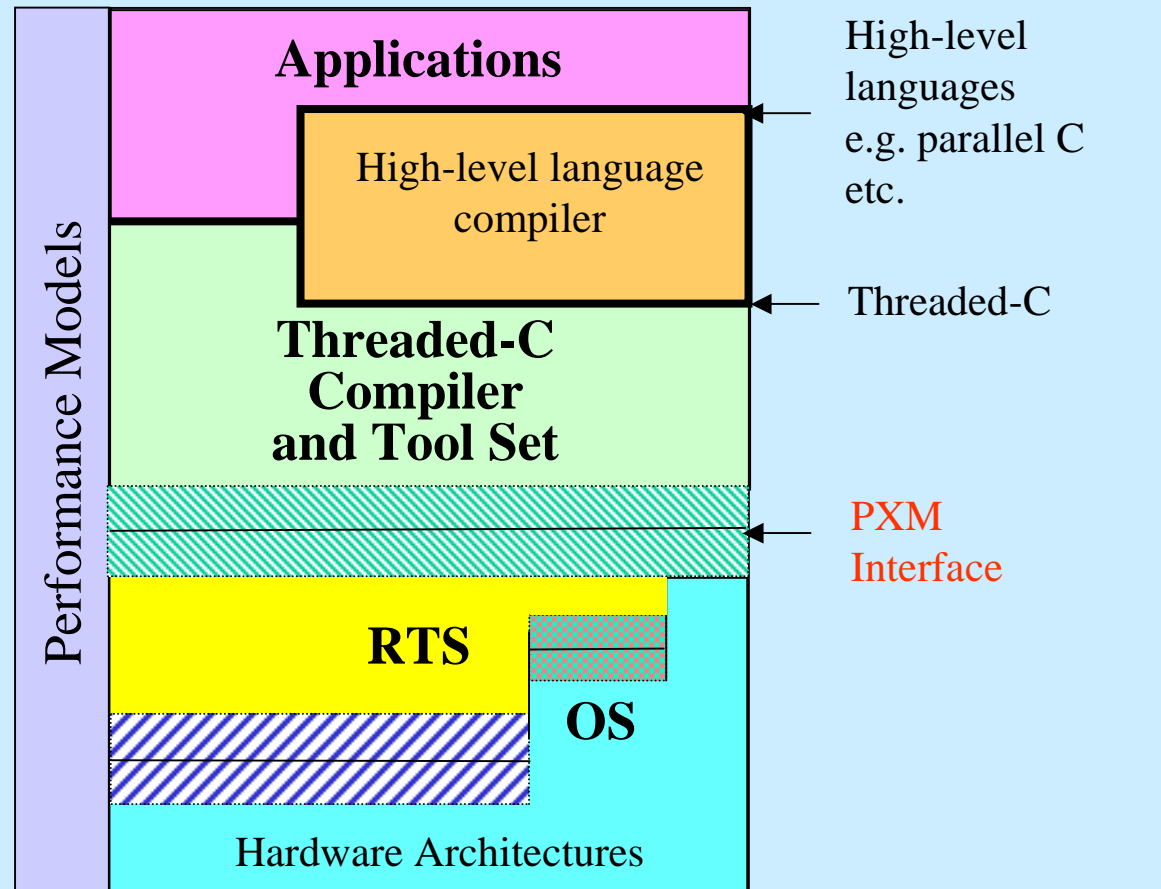
HTMT Percolation Model



An *Integrated* Software Architecture


Note:

- The threaded-C compiler has part of its functions embedded in RTS
- The RTS will work with architecture and OS layers to provide the PXM interface
- The performance models are defined across all layers
- *Elimination unnecessary barriers between different layers*



 Threaded-C Compiler - RTS interface

 RTS-OS interface

 RTS-hardware architecture interface