A Cluster Based Solution for High Performance Hmmpfam Using EARTH Execution Model

Weirong Zhu, Yanwei Niu, Jizhu Lu, Chuan Shen and Guang R. Gao

Computer Architecture & Parallel Systems Laboratory

University of Delaware

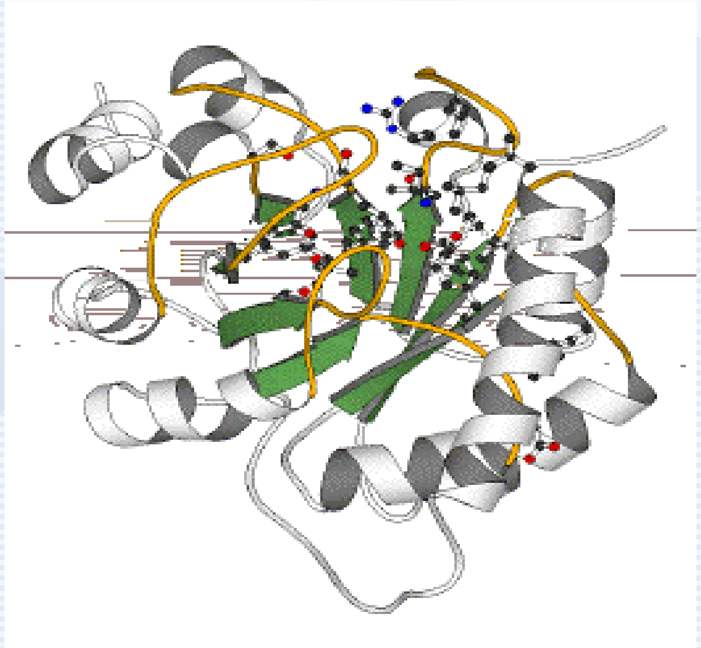
2003-12 -2

Outline

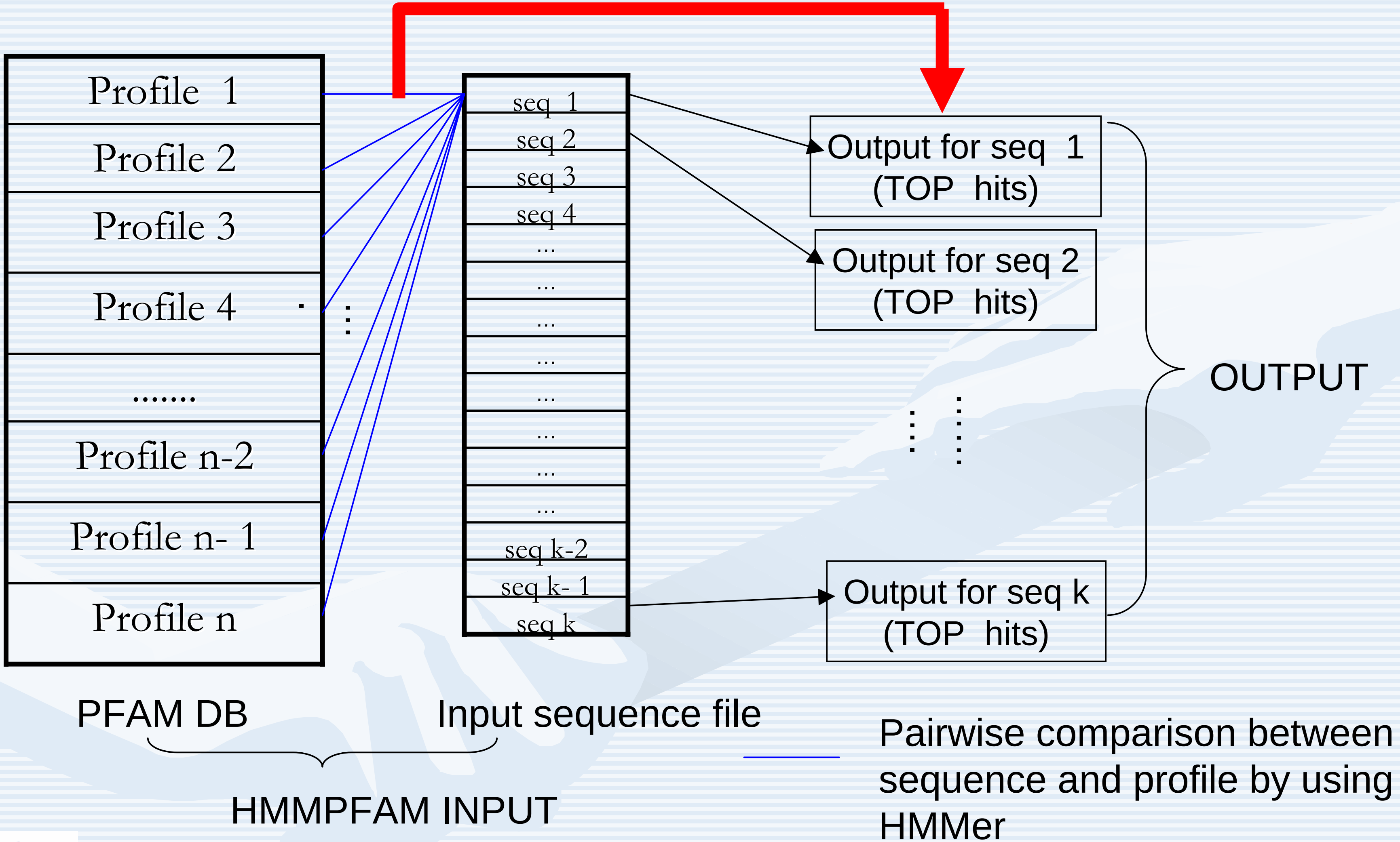
- Introduction
- Background: HMMPFAM & EARTH
- A Parallelization Scheme
- Experiments and Results
- Conclusion & Future work
- Acknowledgement

Introduction to HMMPFAM

- Profile Hidden Markov Models (profile HMMs) can be used to do sensitive database searching.
- HMMER is a freely distributable implementation of profile HMM software for protein sequence analysis, developed by Sean Eddy Lab at Washington University in St. Louis. (http://hmmer.wustl.edu/)
- Hmmpfam is a tool in HMMER package used to look for known domains in a query sequence, by searching a single sequence against a library of HMMs.
- The PFAM HMM library is a single large file, containing several hundred models of known protein domains.



How HMMPFAM works?



Pairwise comparison between sequence and profile by using HMMer

Motivation

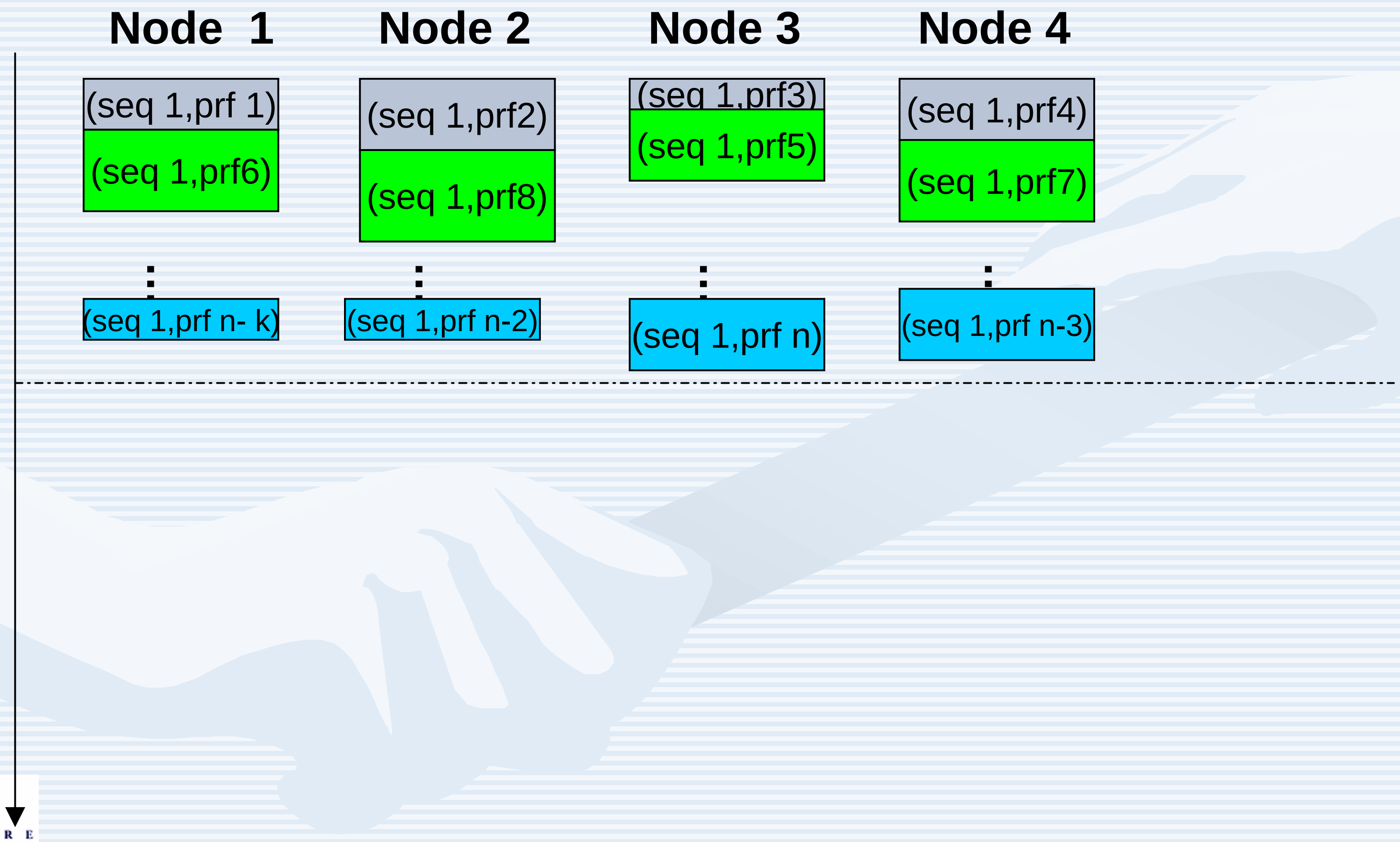
- Hmmpfam is a widely used bioinformatics tool for sequence classification

- In real situations, this program may cost a few months to finish processing large amounts of sequence data. Thus parallelization of the Hmmpfam is a present important challenge for bioinformatics researchers.

PVM version of HMMPFAM

MASTER NODE

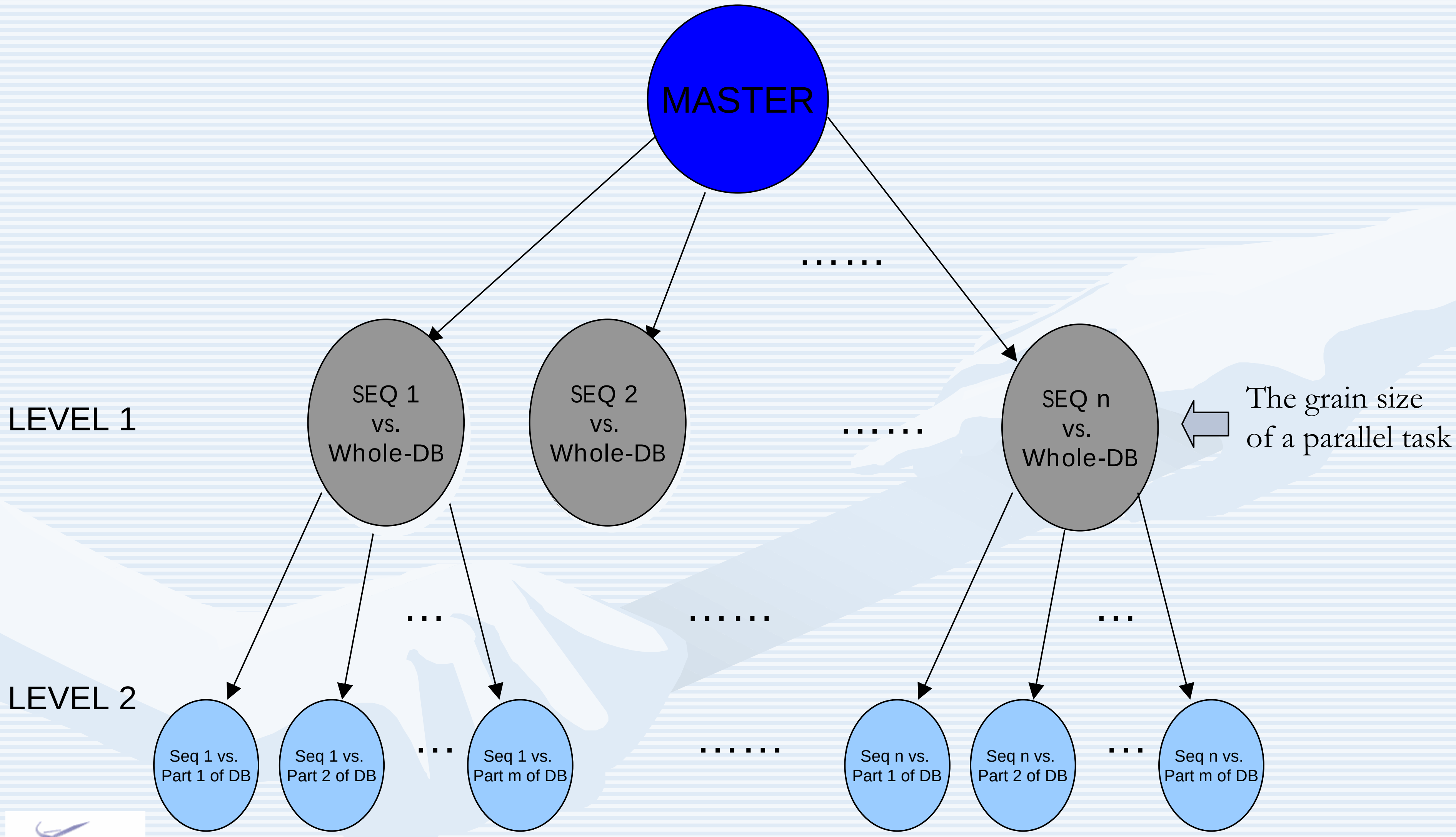
1. Distribute sequence data to slave nodes,
2. invoke process for pairwise comparison on slave nodes,
3. collect and sort result



EARTH

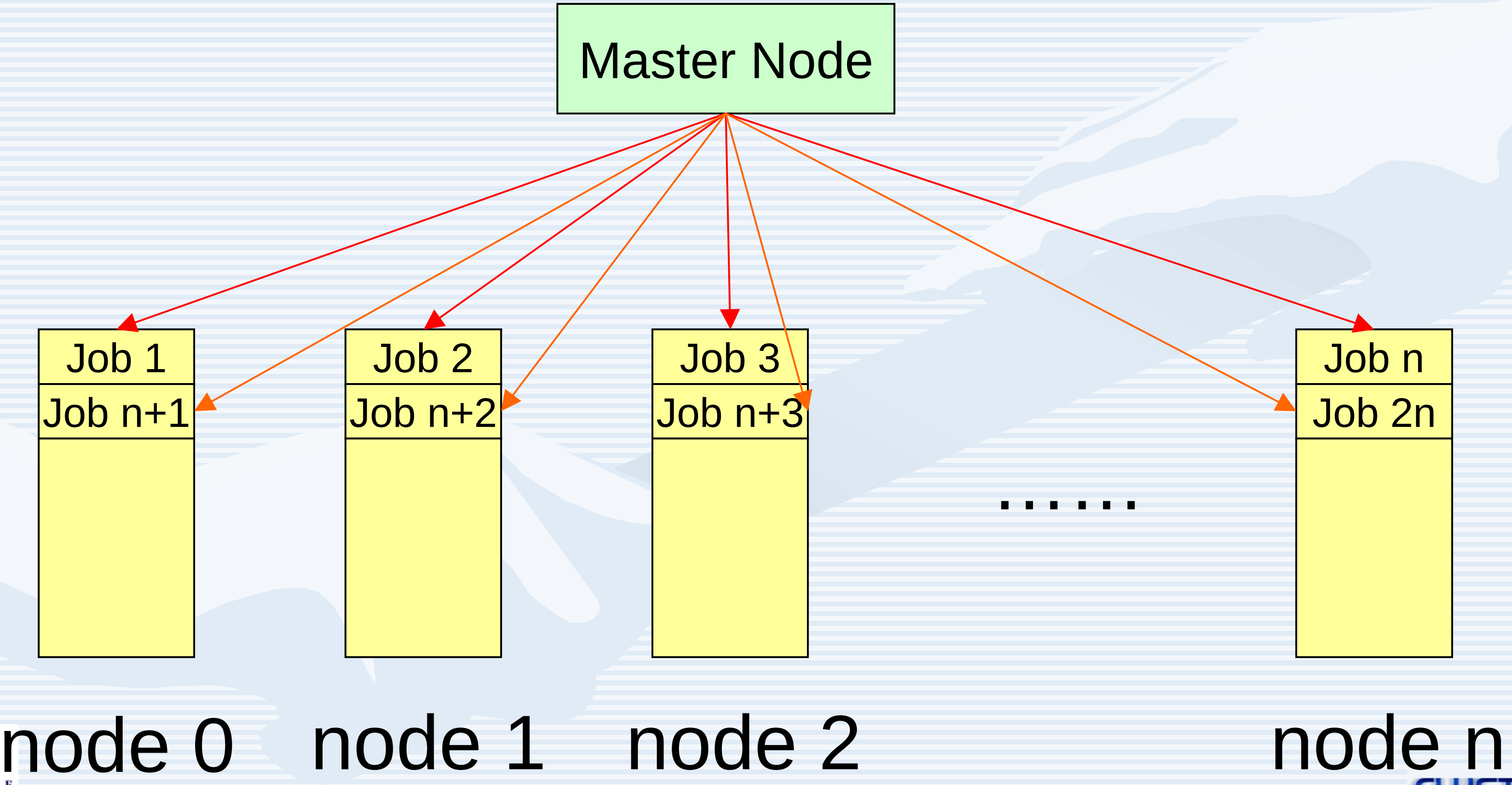
- EARTH – Efficient Architecture for Running Threads, developed by CAPSL.
- EARTH Architecture Model – EU executes threads; SU performs “EARTH OPs”
- EARTH Execution Model
  - Two-level Thread Hierarchy
  - Event Driven Fine Grain Multithreading
- EARTH Runtime System 2.5
  - assumes the responsibility to provide an interface between an explicitly multi-threaded program and a distributed memory hardware platform.

Coarse Grain Parallel Computing Strategy

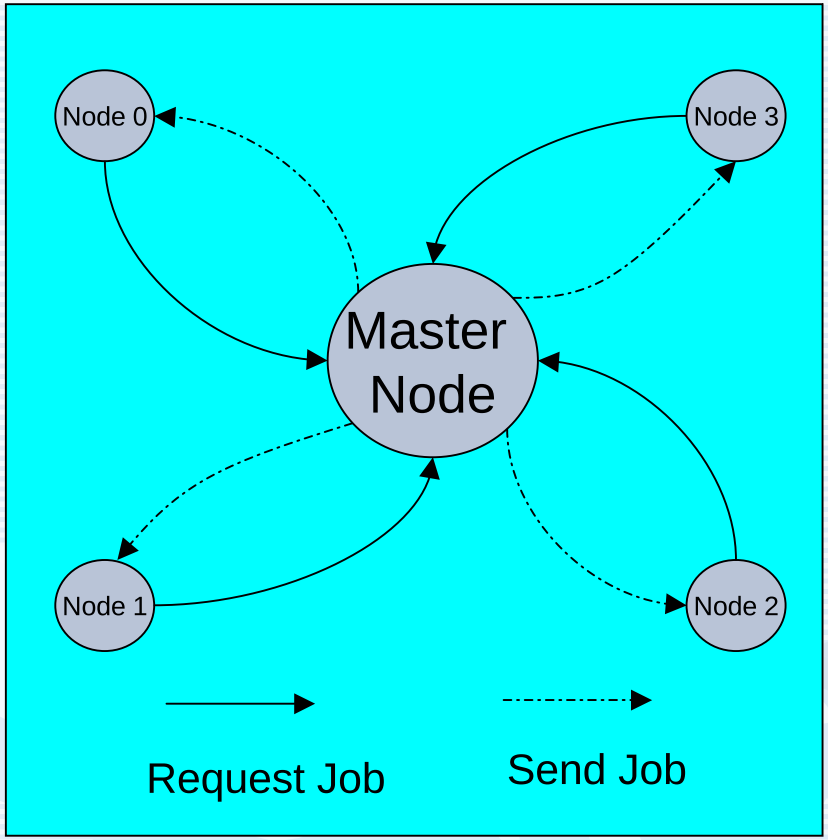


Static Load Balancing

- Job distribution is pre-determined: Programmer explicitly distributes all jobs to the ready queue of computing nodes by the Round-Robin algorithm explicitly at initiation stage.



Dynamic Load balancing

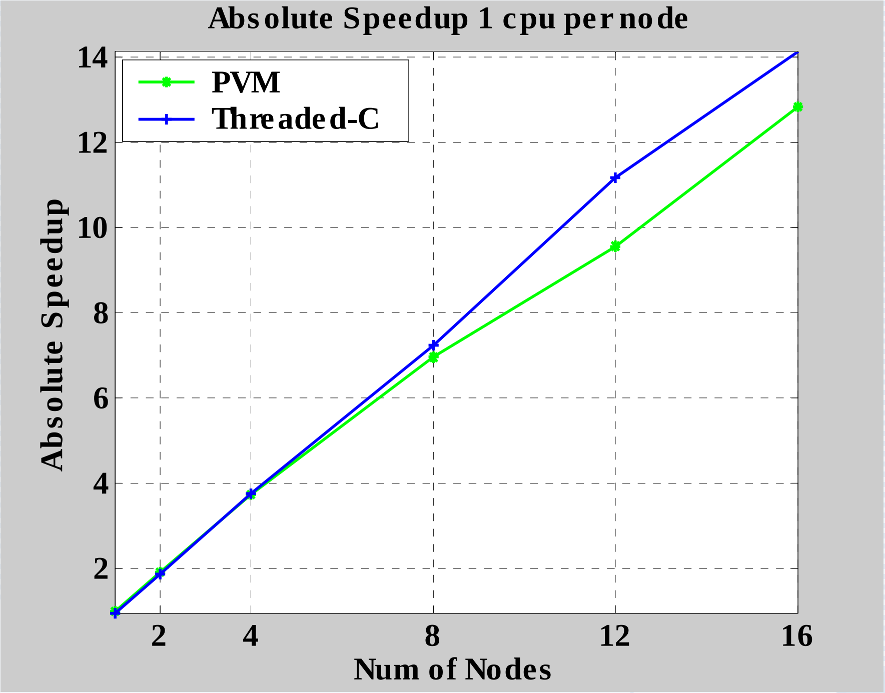


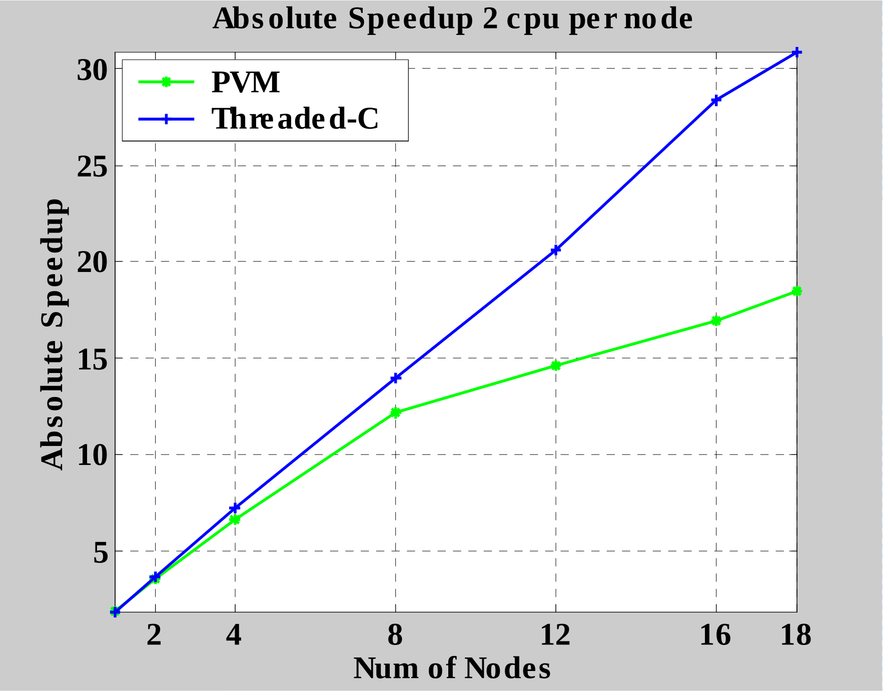
- Once a slave node finishes a job, it sends a request to master. master will respond by sending back a new job.
- The job-request and job-assignment are determined by EARTH RTS dynamically, which is transparent to programmer, thus the work of programming is simplified.
- This approach is robust in that the system won’t be stalled if some nodes are disturbed during running

Experiment Platform

- COMET at CAPSL, University of Delaware: 20 nodes, dual-CPU Athlon 1.4GHz, 512MB DDR SDRAM, Fast Ethernet
- Chiba City at Argonne National Laboratory: 256 dual-CPU Pentium III 500 MHz Computing Nodes with 512 MB of RAM and 9G of local disk. Fast Ethernet and Myrinet
- JAZZ at Argonne National Laboratory: 350 computing nodes, each with a 2.4 GHz Pentium Xeon, and 175 nodes with 2 GB of RAM, 175 nodes with 1 GB of RAM. Fast Ethernet and Myrinet 2000. Ranked 129 in 22nd TOP500 list.

Comparison of PVM version and THREADED-C version on Comet





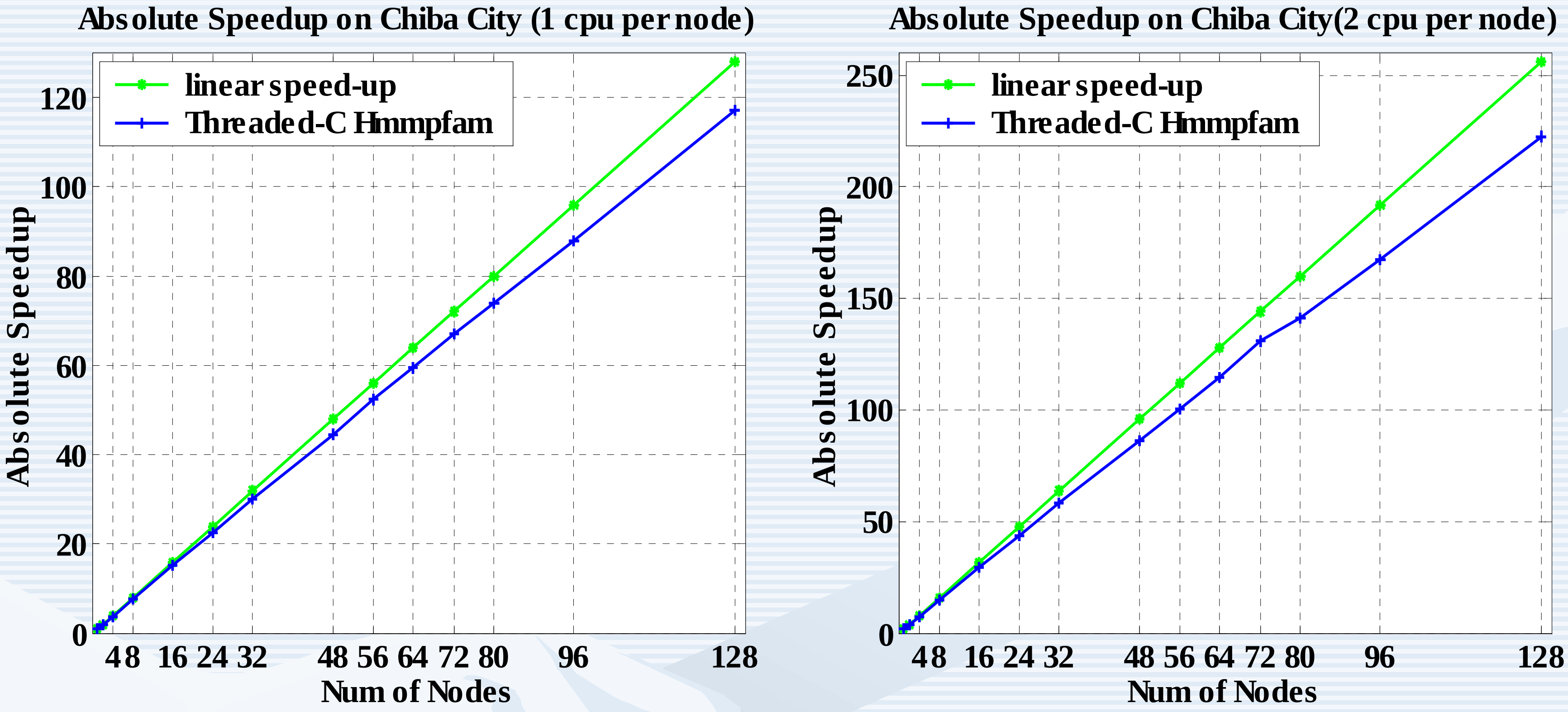
In this experiment, the data used is DB:test.bin.db(585 families), SEQ:hh1.seq(250 seqs)

Threaded-C version achieve better speedup for both 1-CPU and 2-CPU node organizations. It's performance is significantly better than the original PVM code in 2-CPU node organization

THREADED-C HMMPFAM on large clusters

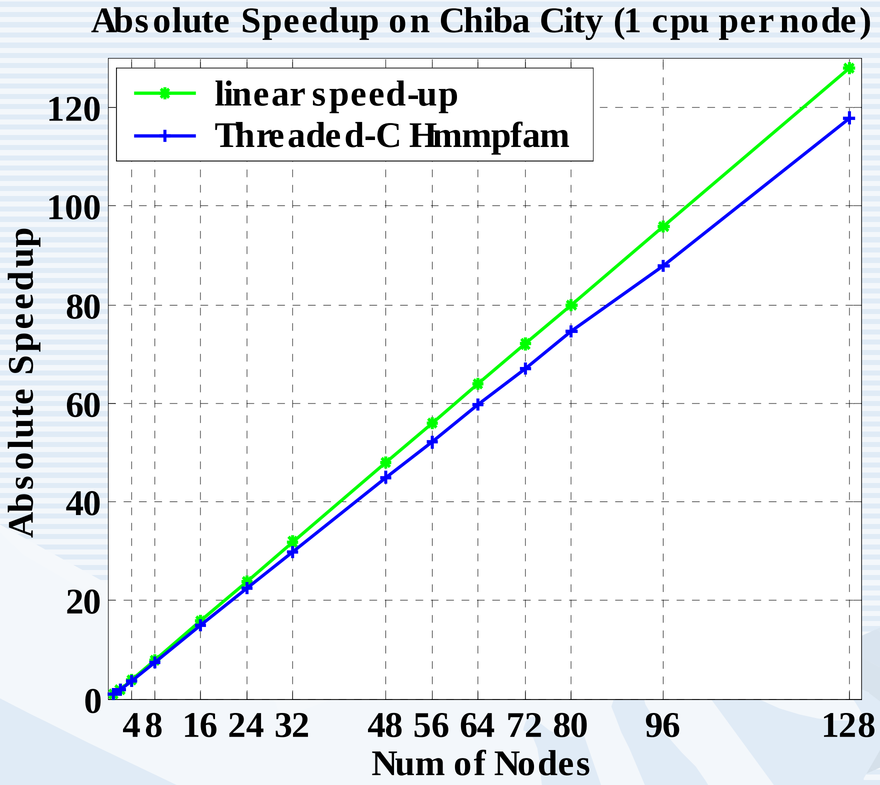
- **Experiment Data:**
  - HMM Database: 50 families
  - Input Sequence file : 38192 sequences
  - At Chiba City, the serial version will cost 15.9 hours to complete
  - At Jazz, the serial version will cost 4.9 hours to complete

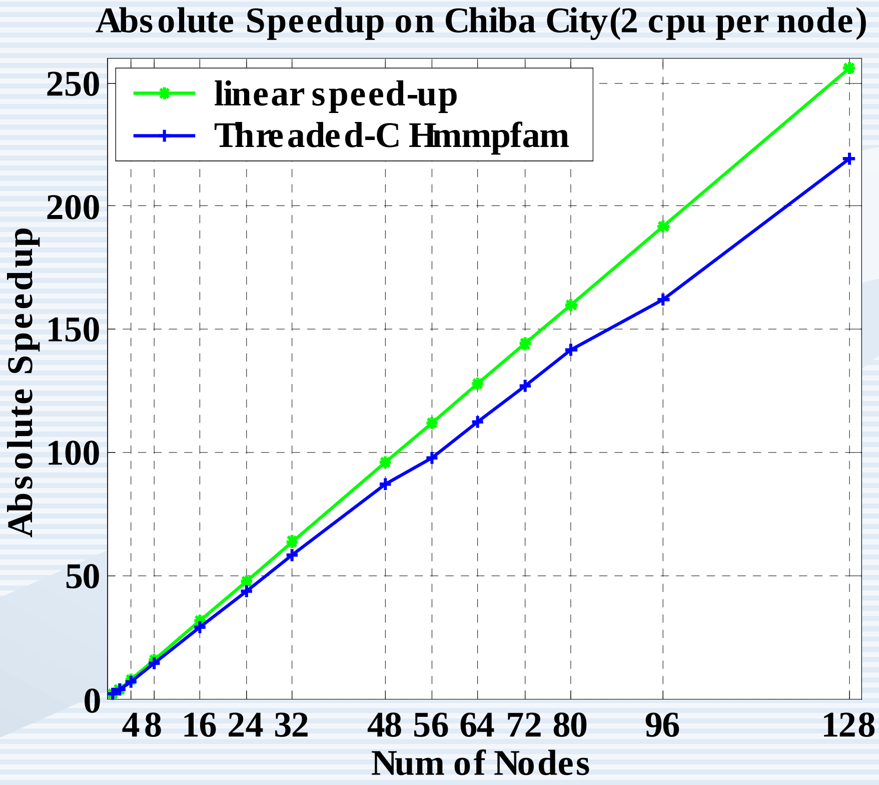
Result of Static Load Balancing



On Chiba City, with 128 dual-CPU nodes, our version can reduce the execution time of one representative test bench from 15.9 hours to 4.3 minutes.

Results of Dynamic Load Balancing



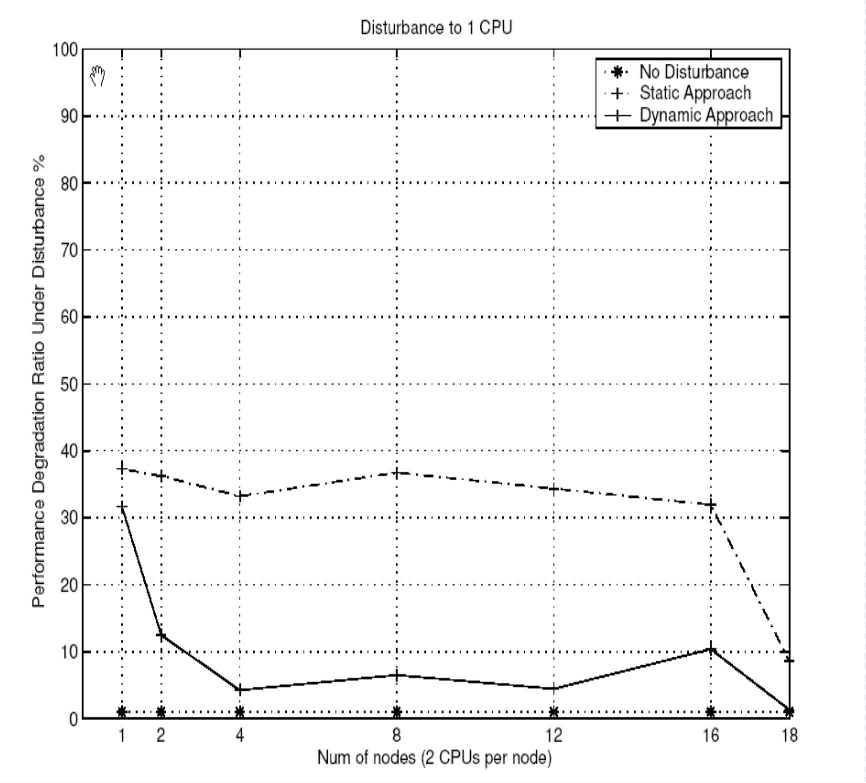


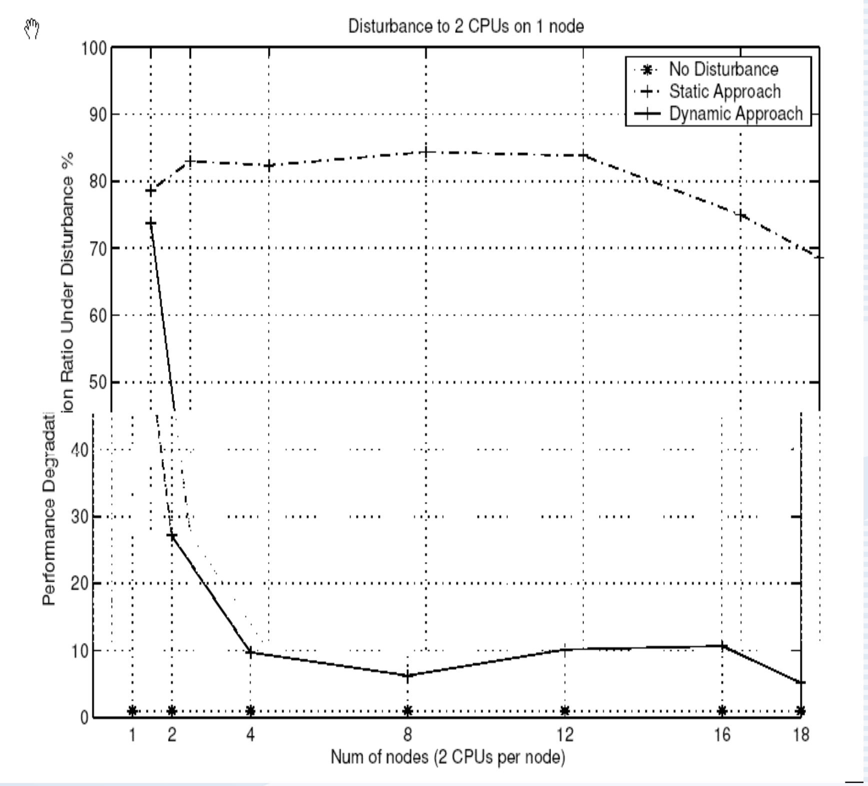
On Chiba City

[redacted]

Balancing

Robustness of the dynamic load balancing





- *degradation ratio under disturbance* (PDRD)
  PDRD = (T1 – T2)/T2
  (T1 – Exec Time With Disturbance, T2 – Exec Time Without Disturbance)
- Disturbance Source is the "blastall" program, which is a computation intensive application

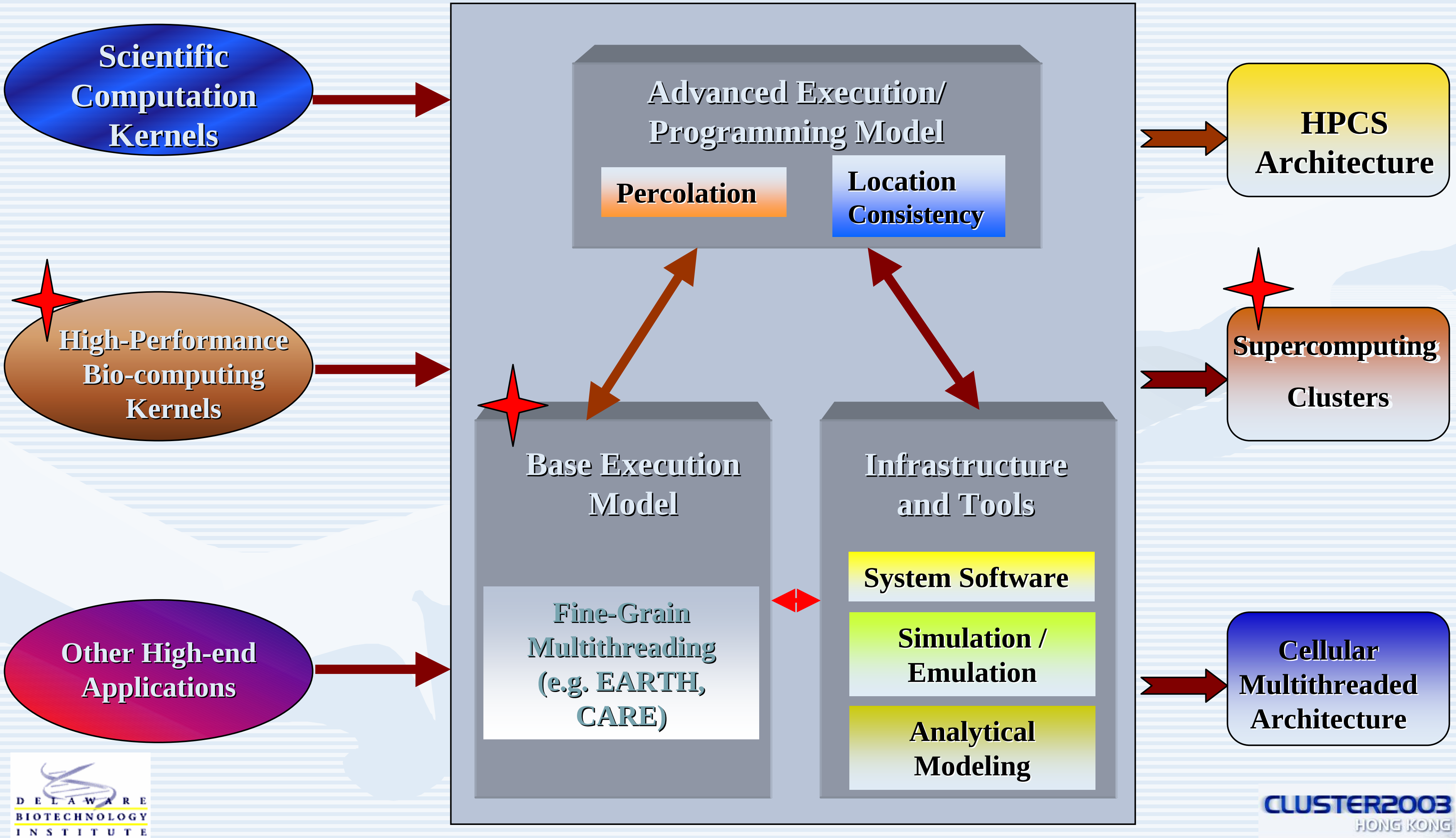
Conclusion And Future Work

- In this research, we implement a new parallel version of hmmpfam on EARTH (Efficient Architecture for Running threads) and demonstrate significant performance improvement over another parallel version based on PVM. On a cluster of 128 dual-CPU nodes, the execution time of a representative test bench is reduced from 15.9 hours to 4.3 minutes.

- Future research directions include further exploiting the fine grain parallelism mechanism of EARTH and compare different parallel scheme.

- Porting of Hmmpfam to EARTH model provides very promising results, thus further research includes the porting of other bioinformatics applications to large clusters.

Research Overview in CAPSL


Scientific Computation Kernels
High-Performance Bio-computing Kernels
Other High-end Applications
Advanced Execution/ Programming Model
Percolation
Location Consistency
Base Execution Model
Fine-Grain Multithreading (e.g. EARTH, CARE)
Infrastructure and Tools
System Software
Simulation / Emulation
Analytical Modeling
HPCS Architecture
Supercomputing Clusters
Cellular Multithreaded Architecture

Acknowledgement

- We acknowledge NSF, DOE, other US Federal agencies, and state of Delaware (via DBI) for their supports.
- We would like to thank the Bioinformatics and Microbial Genomics Team at the Biochemical Sciences and Engineering Division within DuPont CR&D for the useful discussions during the course of this work.
- We thankfully acknowledge the experiment platform support from the HPC Systems Group and the Laboratory Computing Resource Center at Argonne National Laboratory.
- Thanks to all CAPSL members