Programming and Compiling for TiNy Threads (*TNT*) -- *Experience with Cyclops-64 Architecture*

Guang R. Gao

Endowed Distinguished Professor

Electrical & Computer Engineering

University of Delaware

ggao@capsl.udel.edu

Outline

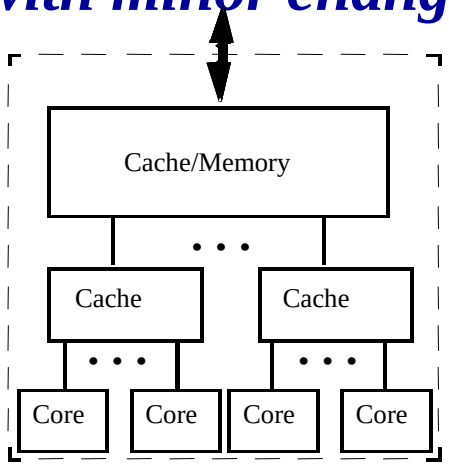
- Introduction
- A New Era – Multi-Core Chip Technology
- IBM Cyclops-64 Architecture/Software
- Programming/Compiling for Cyclops-64
- Looking Beyond Cyclops-64
- Summary

Two Types of Multi-Core Architecture Trends

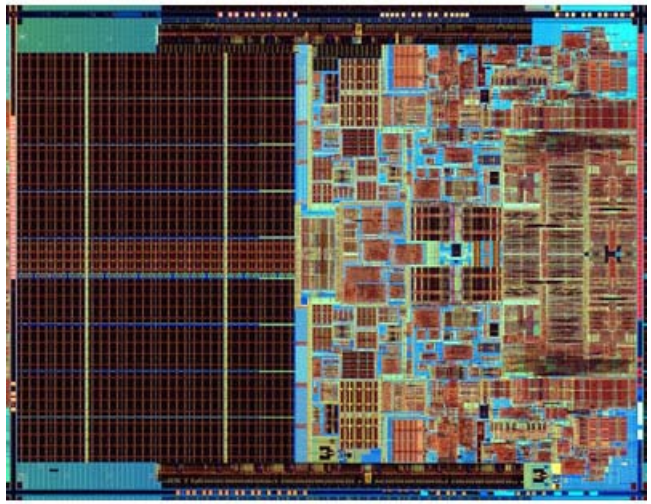
- **Type I**: Glue "heavy cores" together with minor changes
- **Type II**: Explore the parallel architecture design space and searching for most suitable chip architecture models.

Multi-Core Type I:

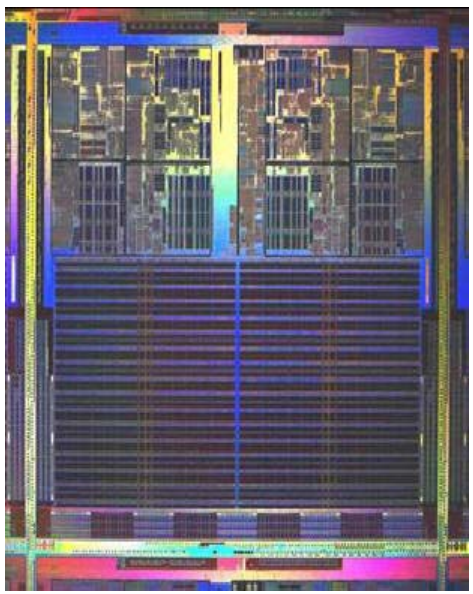
"Glue" together existing "heavy cores" with minor changes



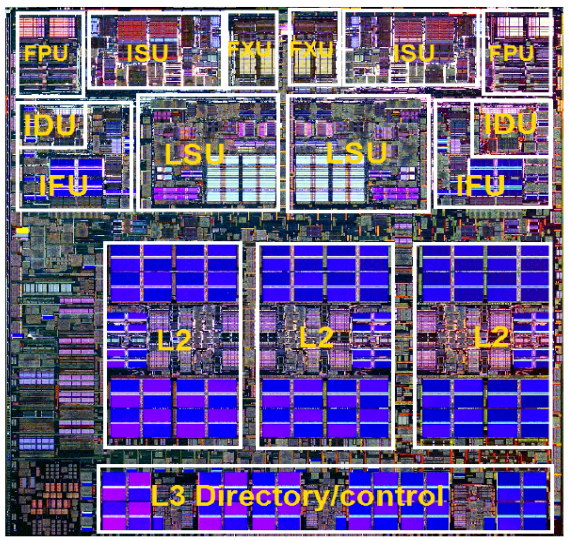
(a) Hierarchical Designs



Intel Core 2 Duo



AMD Opteron Dual-Core



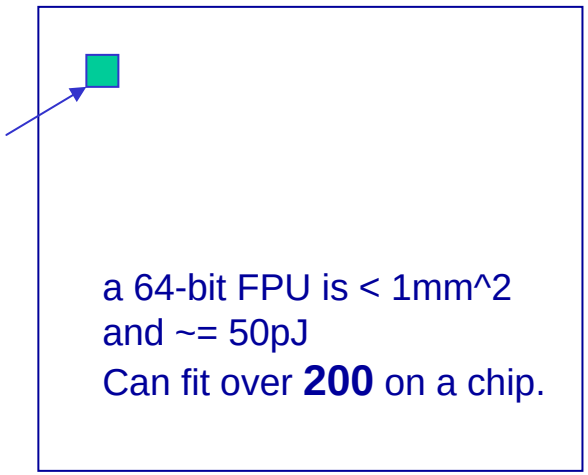
IBM Power5

Multi-Core Type II

- New factors to be considered
  - Flops are cheap!
  - Cache-coherence is expensive!
  - On-chip bandwidth can be enormous!
  - Examples: Cyclops-64, and others

Flops are Cheap!

An example to illustrate design tradeoffs:



14mm x 14mm chip (130nm and 1GH)

Curtsey: Numbers are due to Steve Scott [PACT2006 Keynote]

- If fed from small, local register files:
  - 3200 GB/s, 10 pJ/op
  - < $1/Gflop (60 mW/Gflop)
- If fed from global on-chip memory:
  - 100 GB/s, 1nJ/op
  - ~ $30/Gflop (1W/Gflop)
- If fed from off-chip memory:
  - 16 GB/s
  - ~$200/Gflop (many W/Gflop)

500 FPUs on a chip Is possible!

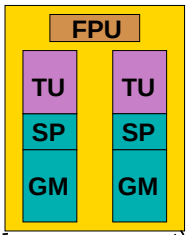
[M. Denneau: private communication]

Outline

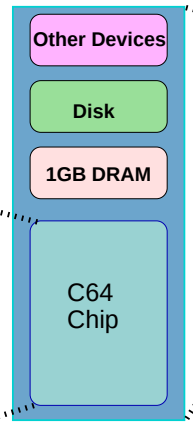
- Introduction
- A New Era – Multi-Core Chip Technology
- IBM Cyclops-64 Architecture/Software
- Programming/Compiling for Cyclops-64
- Looking Beyond Cyclops-64
- Summary

Cyclops-64 (C64) Supercomputer

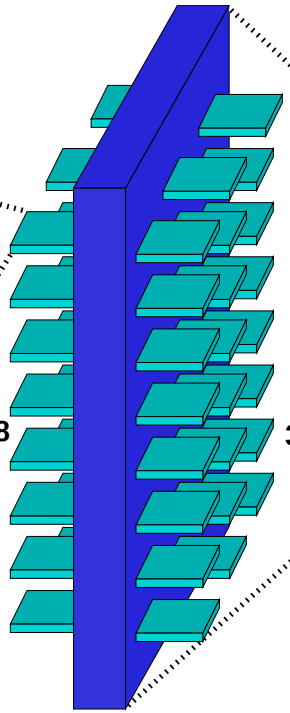
Processor(2 Threads)
1Gflops
60KB SRAM



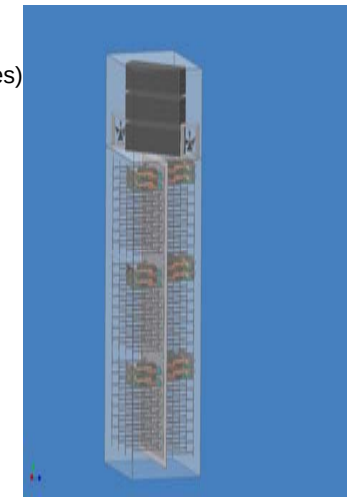
Board(1 Chip)
80Gflops
1GB DRAM

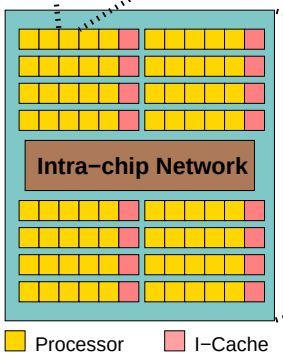


BackPlane(48 Boards)
3.84TFlops / 48GB

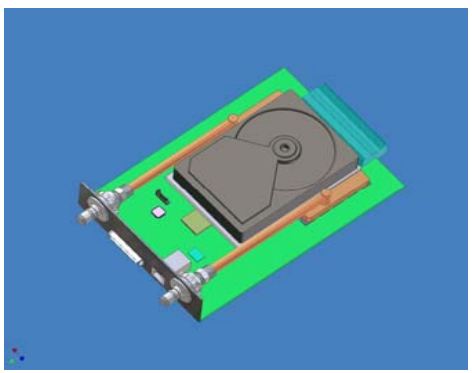


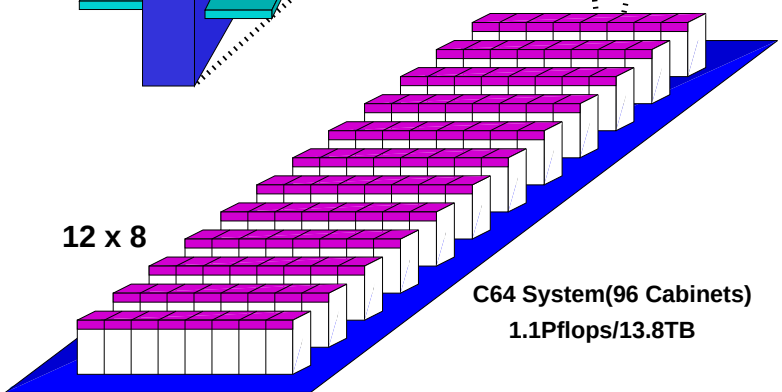
Cabinet(3 BackPlanes)
11.52TFlops/144GB





Chip(80 Processors)
80Gflops
4.7MB SRAM



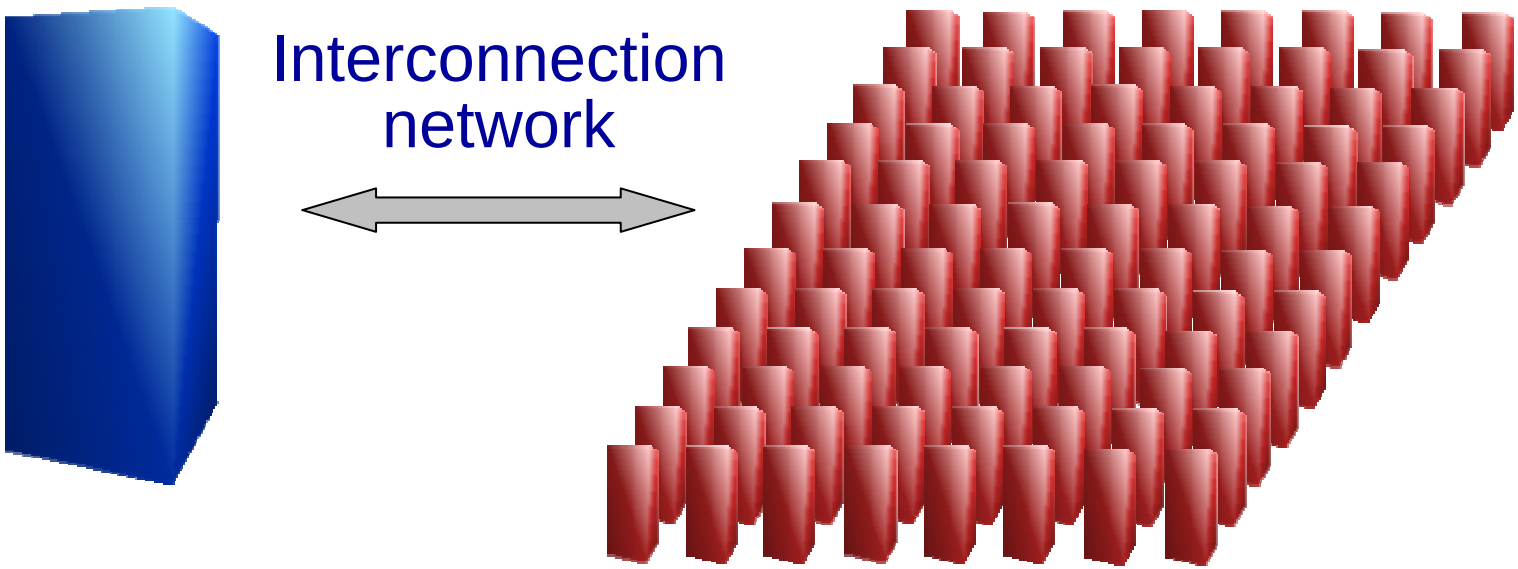


C64 System(96 Cabinets)
1.1Pflops/13.8TB

10,000 Feet High System Overview

Front-end
Programming environment

Back-end
C64 computing engine



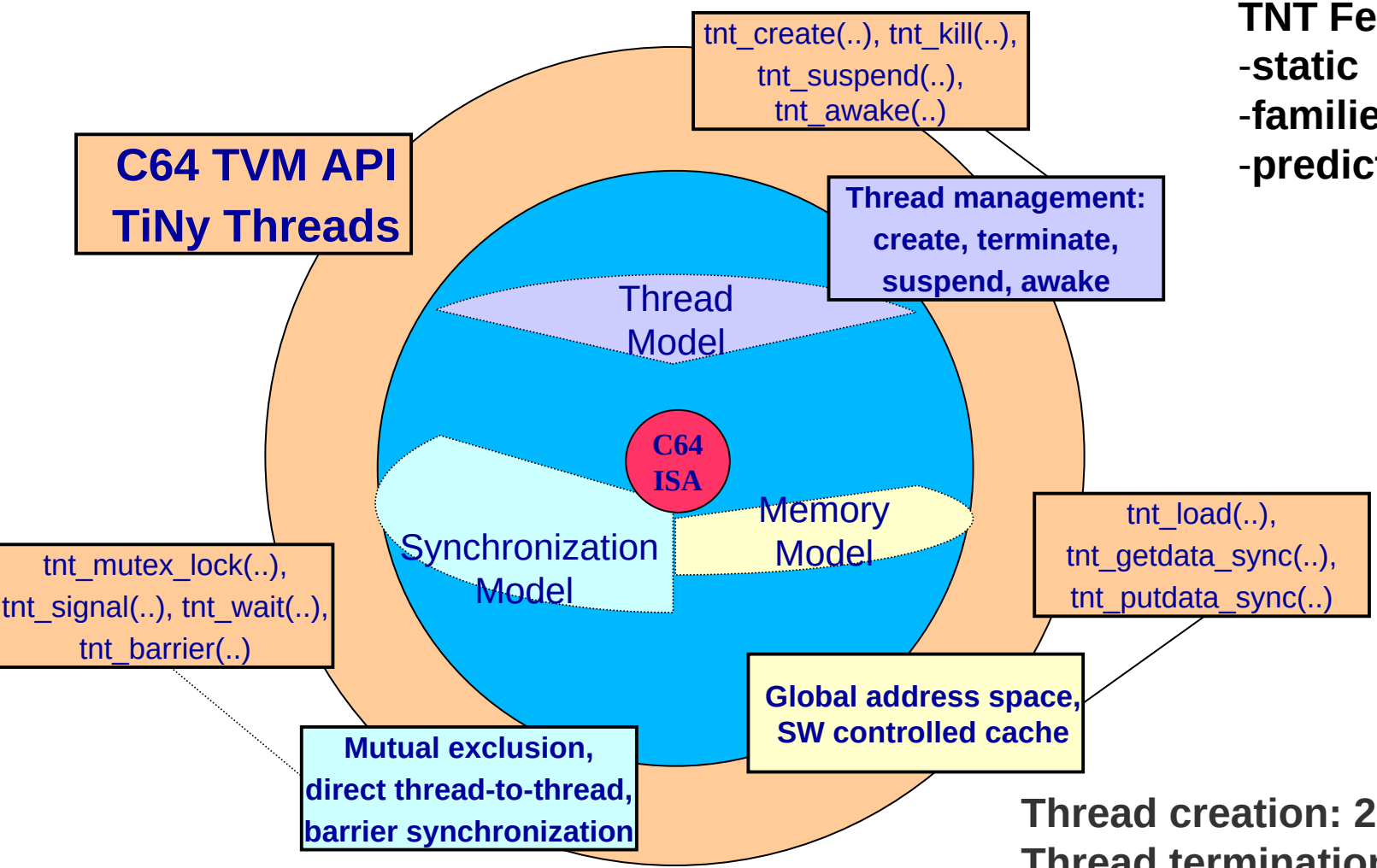
Curtsey from E.T. International Inc

TyNy-Thread – The API of A Cyclops-64 Thread Virtual Machine

- Multi-chip multiprocessor extension of the base C64 ISA.
- Runs directly on top of C64 HW architecture.
- Takes advantage of C64 HW features to achieve high scalability.
- Supports kernel residency and runtime system modes.
- Provides the interface that shields programmers from C64 HW complexities adding little overhead.
- Three components: thread model, memory model and synchronization model.

[Cuvillo, et. al., ISCA95 Worshop]

Cyclops-64 Thread Virtual Machine (TNT)



TNT Features:
- **static**
- **familier**
- **predictable**

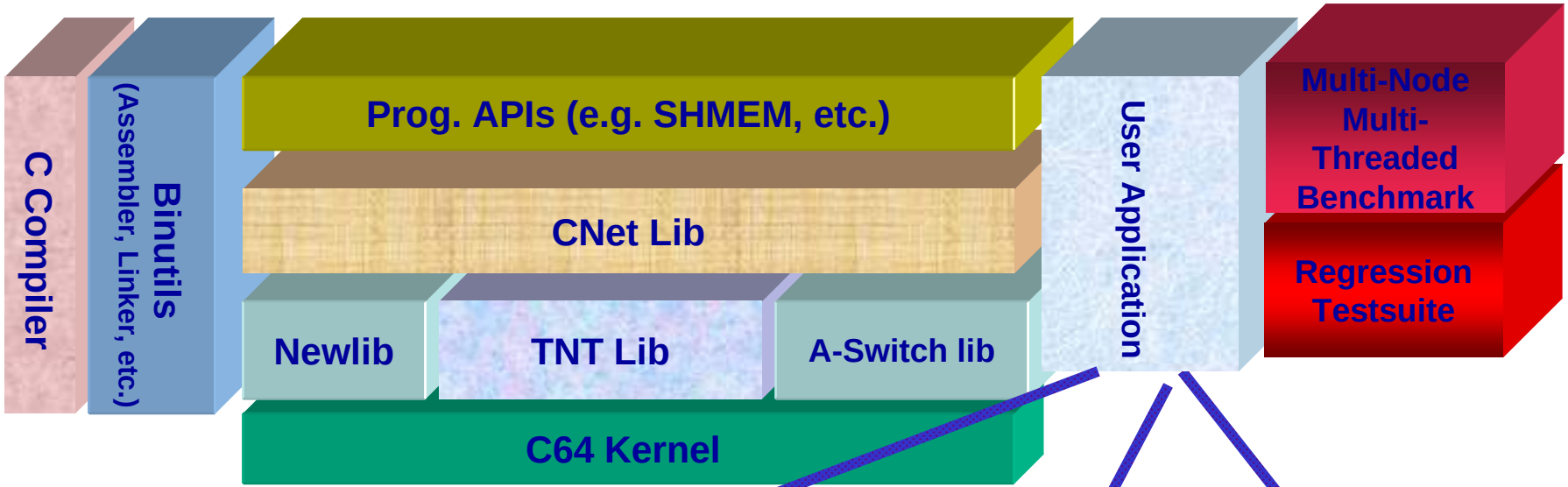
Thread creation: 280 cycles.
Thread termination: 60 cycles.
Thread reuse: 265 cycles.

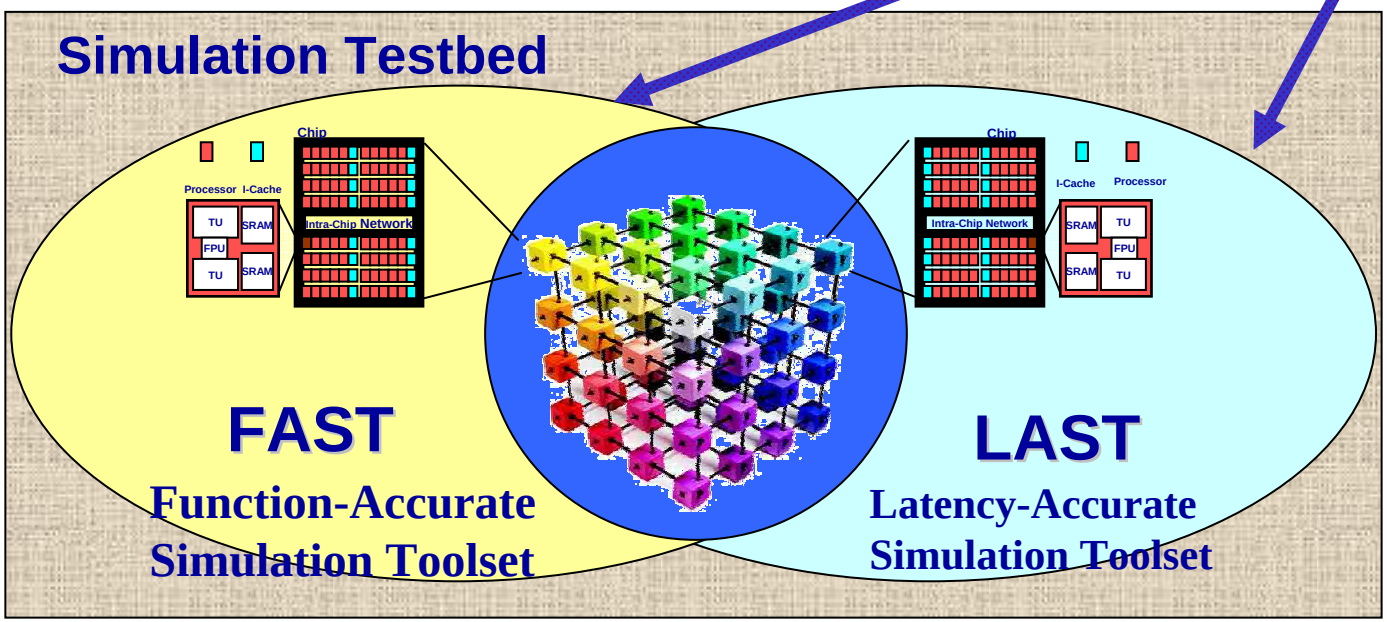
C-64 Memory Consistency Model

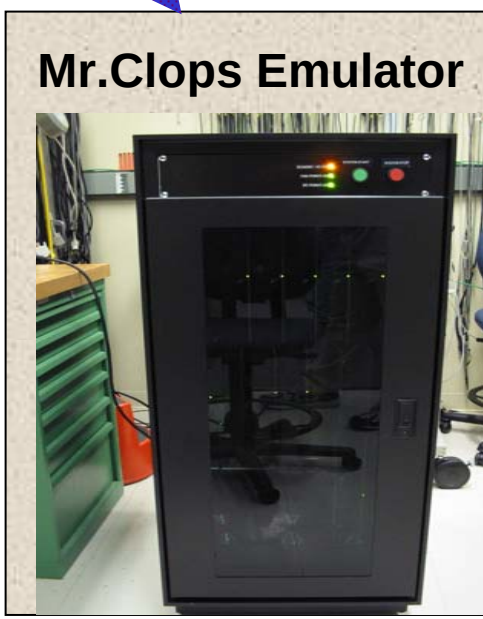
- The On-Chip SRAM is sequential consistency compliant (SCC) [**ZhangEtAl05**]
- Accesses to scratch-pad memory region do not obey SC
- A weak memory consistency model (e.g. LC consistency) is effectively studied as a natural choice for the scratch-pad memories.

  [**GaoSarkar00,SarkarGao04**]

Cyclops-64 Software Toolchain





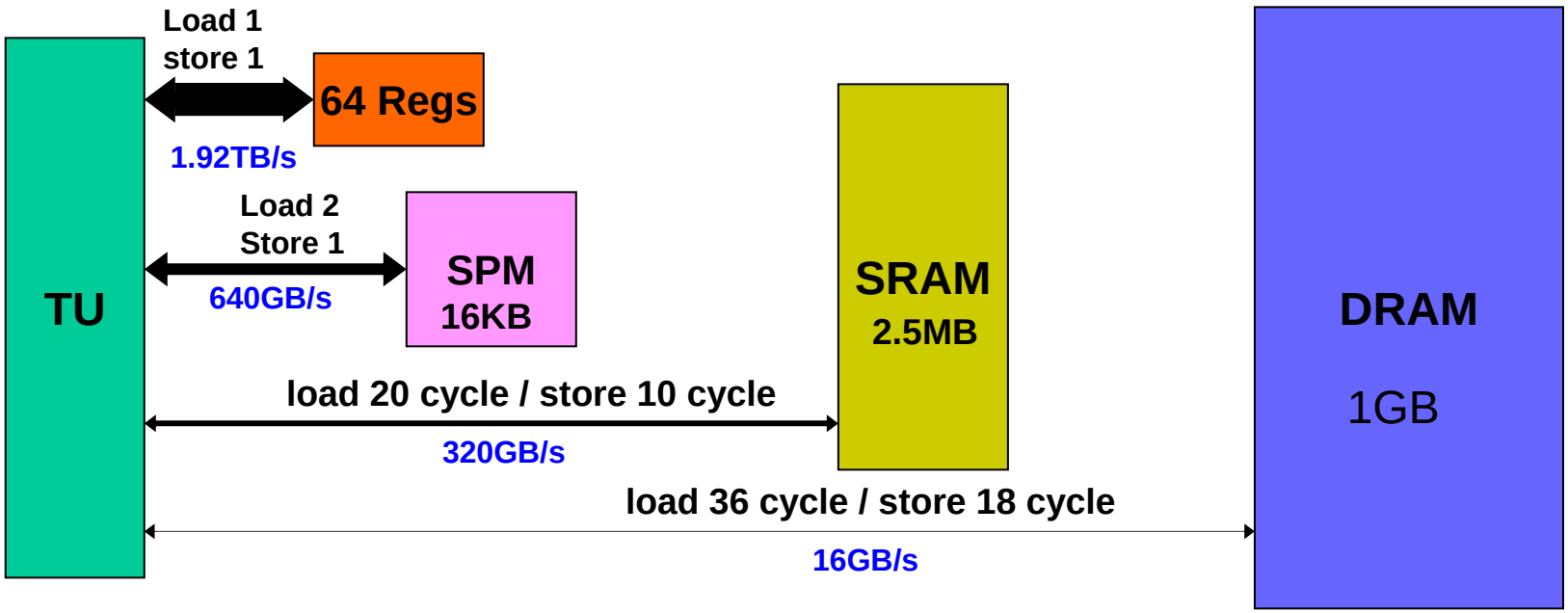


Outline

- Introduction
- A New Era – Multi-Core Chip Technology
- IBM Cyclops-64 Architecture/Software
- Programming/Compiling for Cyclops-64
- Looking Beyond Cyclops-64
- Summary

Challenges

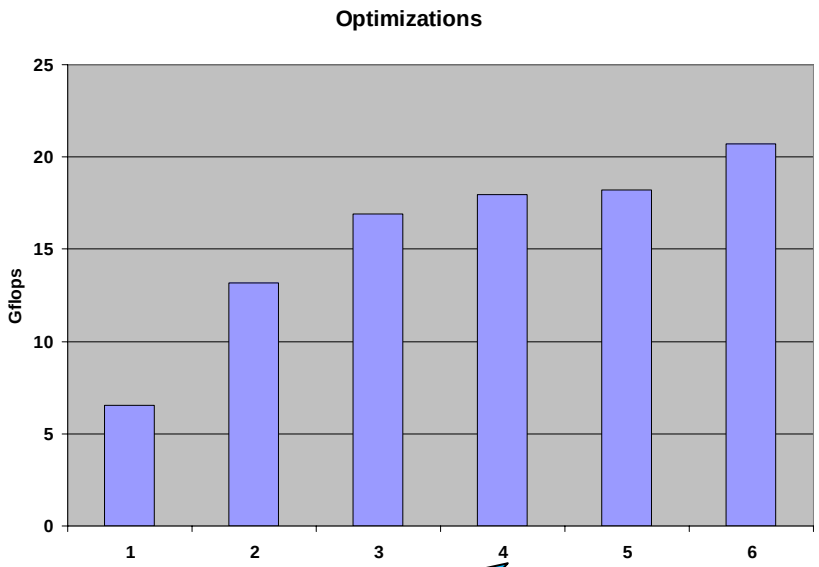
How to exploit the massive ***on-chip parallelism*** and ***bandwidth*** to tolerate ***off-chip*** memory ***bandwidth*** and ***latency***?

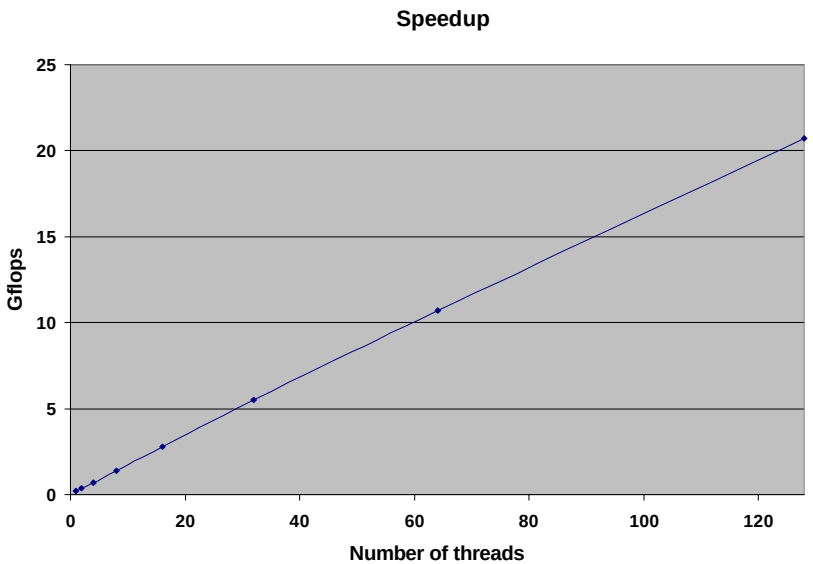


C64 Architecture Features

- Shared Memory Hierarchy (***hotel vs. cache***)
  - No *data cache* is used
  - Explicit memory regions:
  - No hardware VM manager
  - In-Memory atomic operations (non blocking)
- Hardware multi-threading model (“***nap***” vs. “***sleep***”)
  - Efficient nap (vs. sleep) and wakeup
  - Plenty of threads
- Hardware synchronization support

FFT: Performance with Various Optimizations





1 – Base parallel version (2-point work unit)

2 – + Using 8-point work unit

3 – + Special approach in the first 4 stages

4 – + Eliminate redundant memory operations (twiddle factors)

5 – + Loop unrolling (bit-reversal permutation)

6 – + Register allocation & Instruction scheduling (manually)

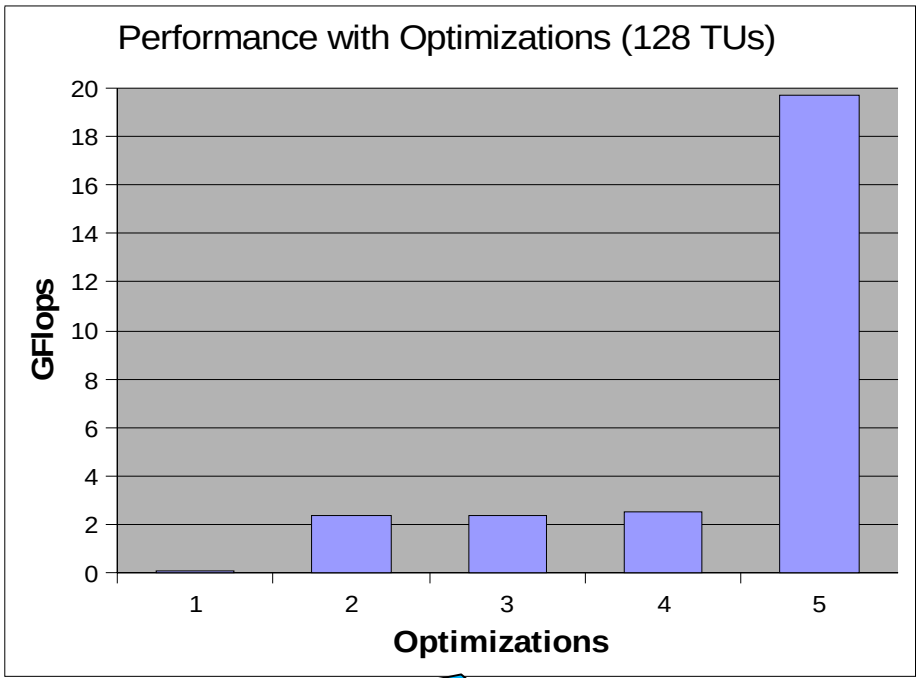
* The experiments were conducted with 128 threads.

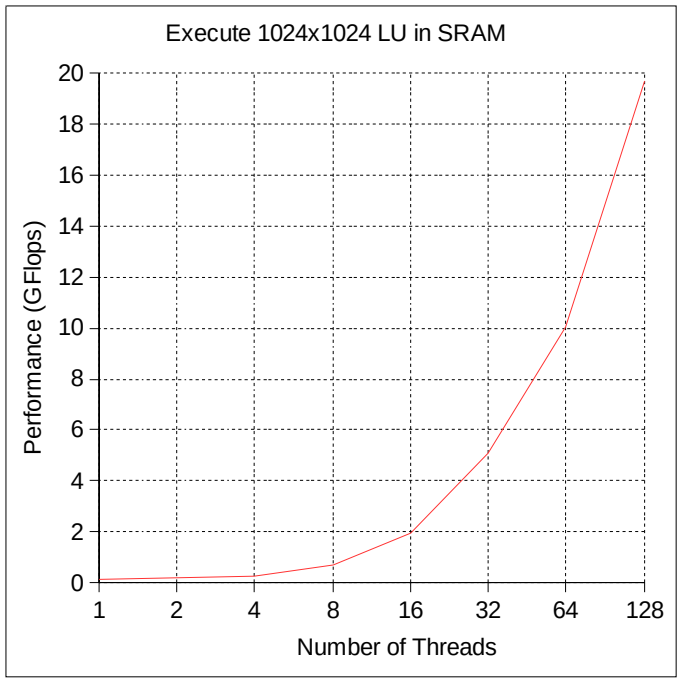
Data size – 2^{16} double precision 1D FFT

Experiments were conducted on ETI Cyclops-64 Toolchain 1.6.2.

ACK: Mike Merrill

Performance with Optimizations - LU





1 – Base Parallel Version

2 – + Dyn. Repartitioning + Recursio

3 – + Processor Adaptation

4 – + Hardware Barrier

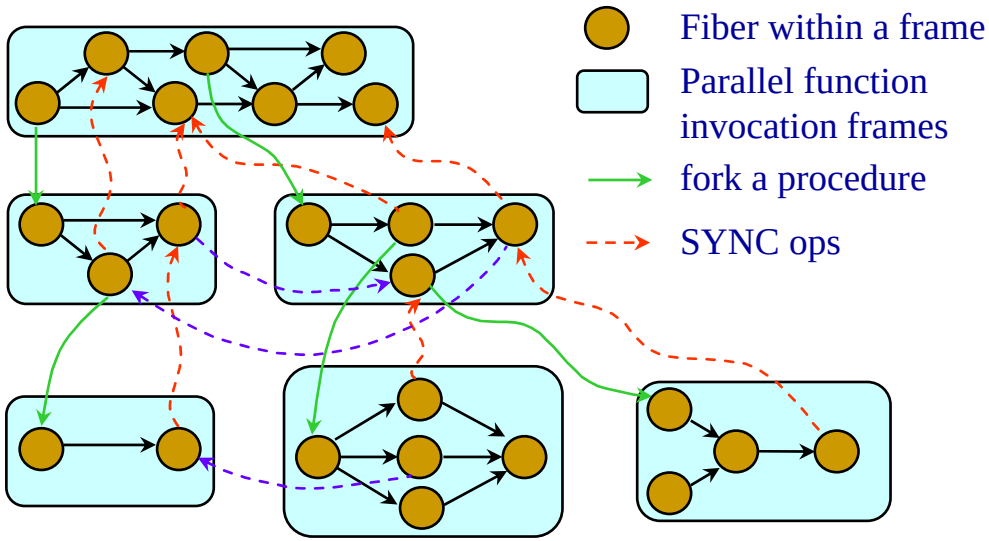
5 – + Reg. Tiling (manually)

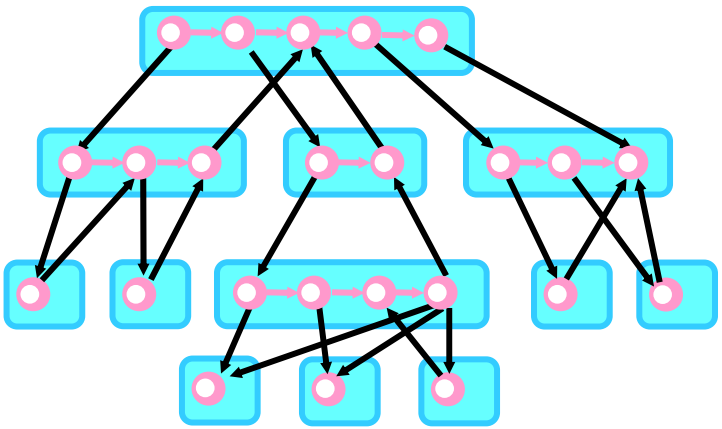
Work in Progress

Outline

- Introduction
- A New Era – Multi-Core Chip Technology
- IBM Cyclops-64 Architecture/Software
- Programming/Compiling for Cyclops-64
- Looking Beyond Cyclops-64
- Summary

EARTH vs. CILK





EARTH Model

CILK Model

Note: EARTH has it origin in static dataflow model

Outline

- Introduction
- A New Era – Multi-Core Chip Technology
- IBM Cyclops-64 Architecture/Software
- Programming/Compiling for Cyclops-64
- Looking Beyond Cyclops-64
- Summary

Summary

- It is the *program execut*
  important
- The "software problem'
  problem to be solved by s

*I am dismayed by the tendency to **split** computer science education into “programming" and "hardware" with so little brought in about the way they interact.*

- Jack B. Dennis

Acknowledgements

- Our Sponsors
- IBM Cyclops-64 Team (Monty Denneau, et. al.)
- ETI Cyclops-64 Team
- Members of CAPSL
- Other Collaborators
- My Host