Institute of Computing Technology
Chinese Academy of Sciences

Godson-T
A High-Efficient Many-Core Architecture for Parallel Program Executions

Dongrui Fan
Advanced Micro-System Group
Institute of Computing Technology
Chinese Academy of Sciences

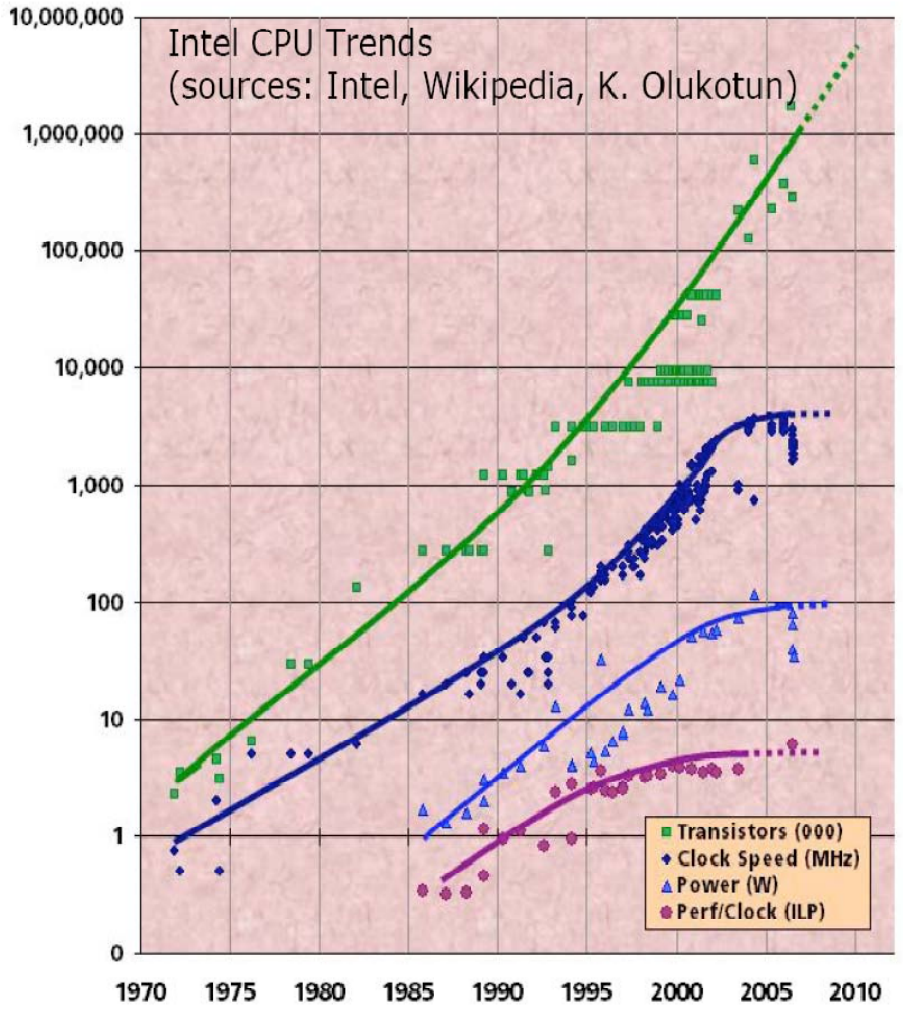
Advanced Micro System Group

Outline

- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

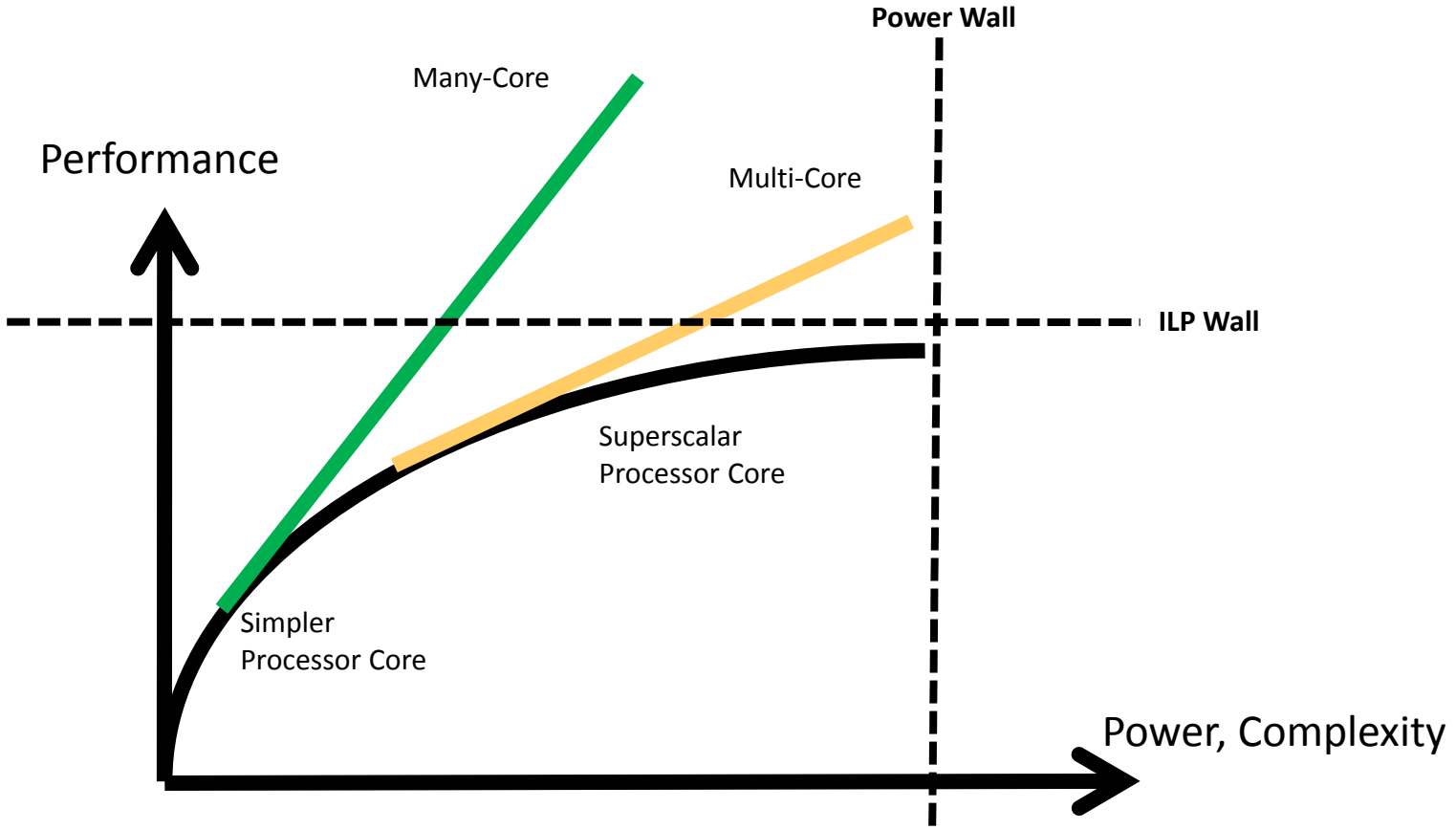
What will be the Next Moore's Law?

- Moore's Law is still in effect
- Processor microarchitecture does not scale well
  - Frequency scaling ends
  - Aggressive ILP exploitation becomes inefficient
  - Power dissipation meets limitation
- → Parallel Microprocessor



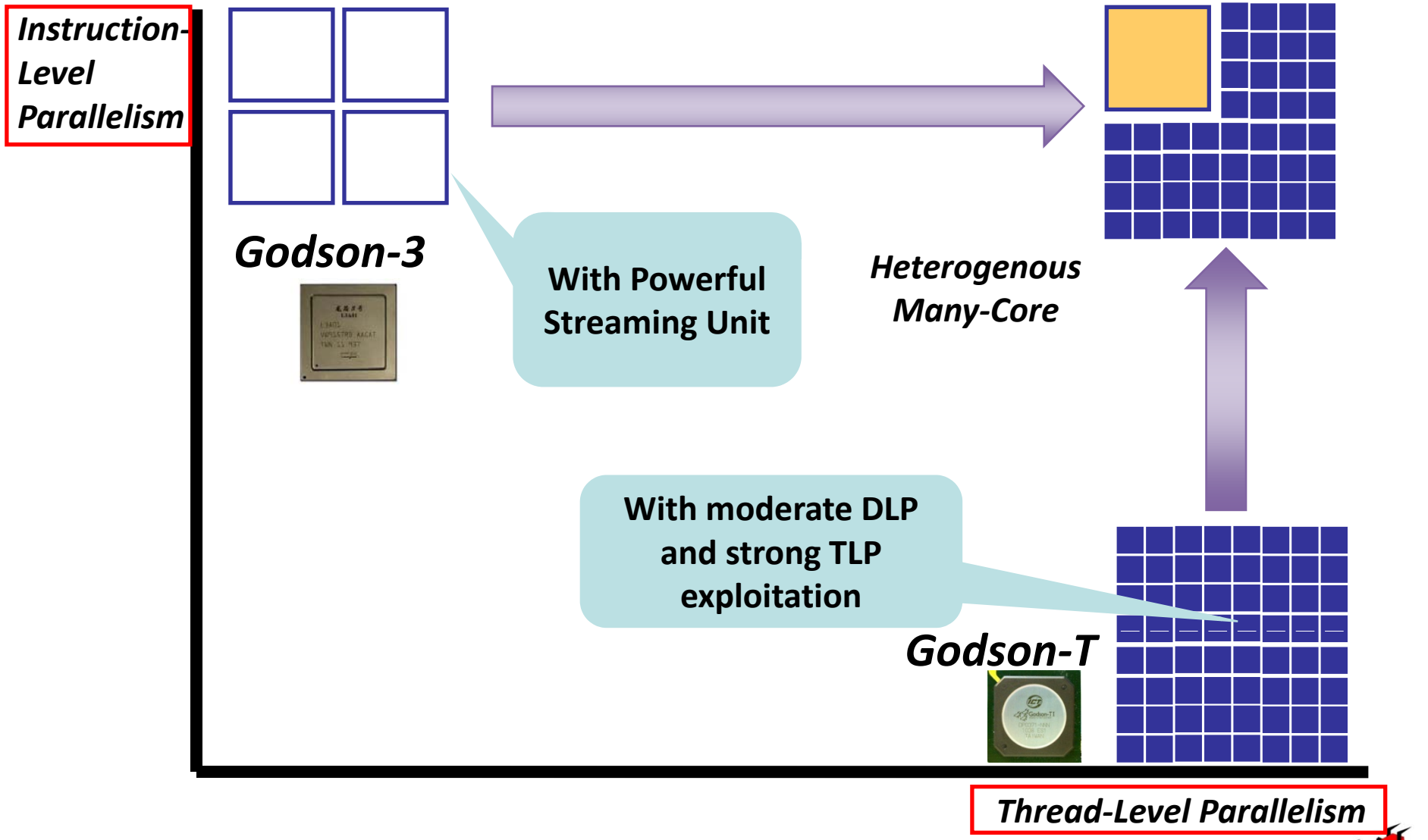
Microprocessor Architecture Revolution

- Memory wall + Power wall + ILP wall = Brick wall !
  What parallel architecture will be successful?



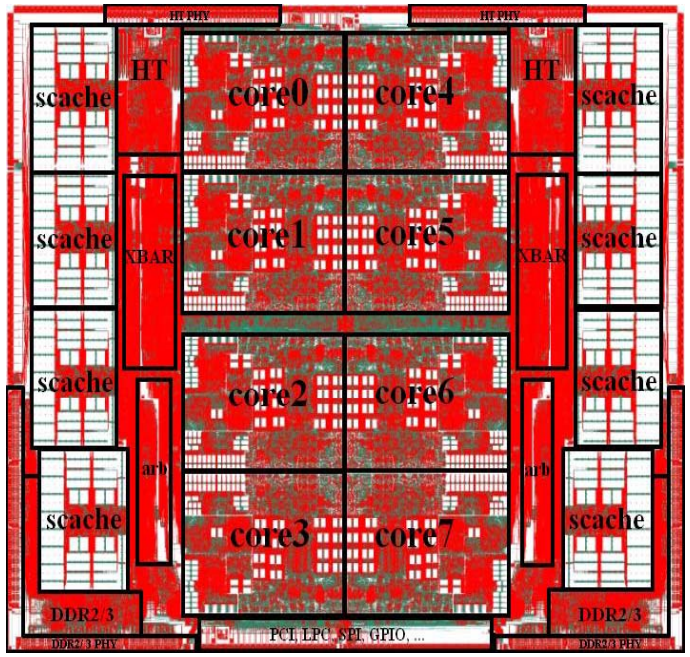
- Parallel architectures (TLP) come to rescue, superscalar processor is being replaced by on-chip multi-core / many-core processor

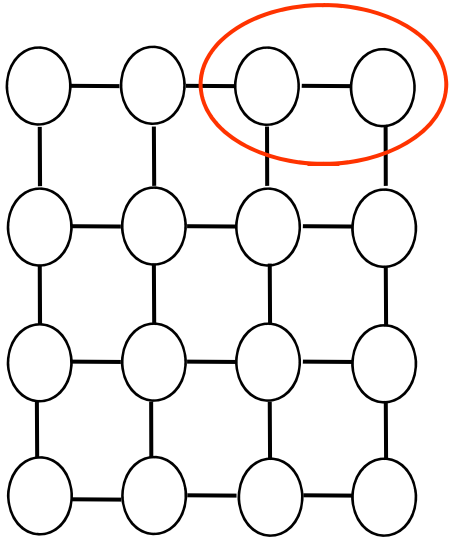
Exploiting Polymorphic Parallelism

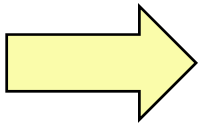


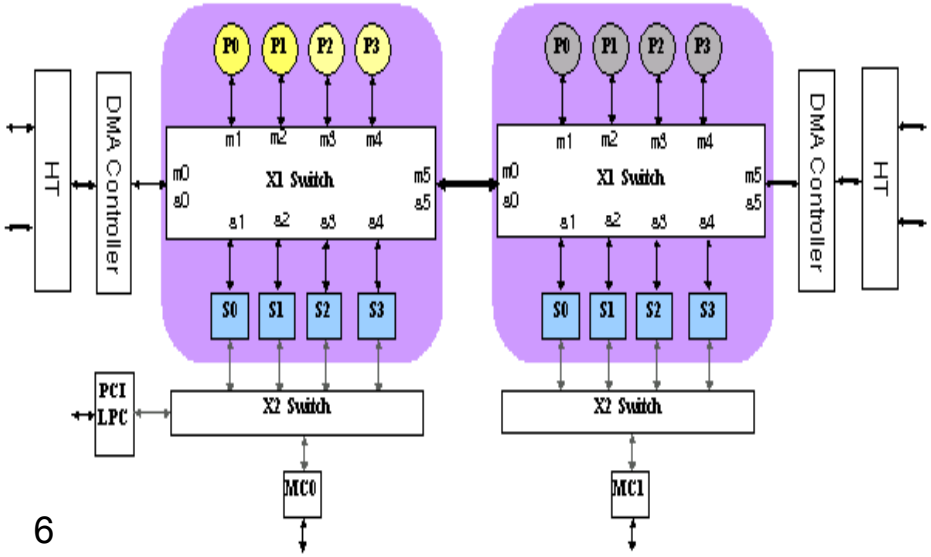
8 Core Godson-3B

- 8 four-issue 64-bit core
- 2*256-bit Vector Ext. per core
- 1.0GHz@65nm
- 128GFLOPS@40W
- 2 DDR3, 2 HT Controllers
- Taped out 2010.5



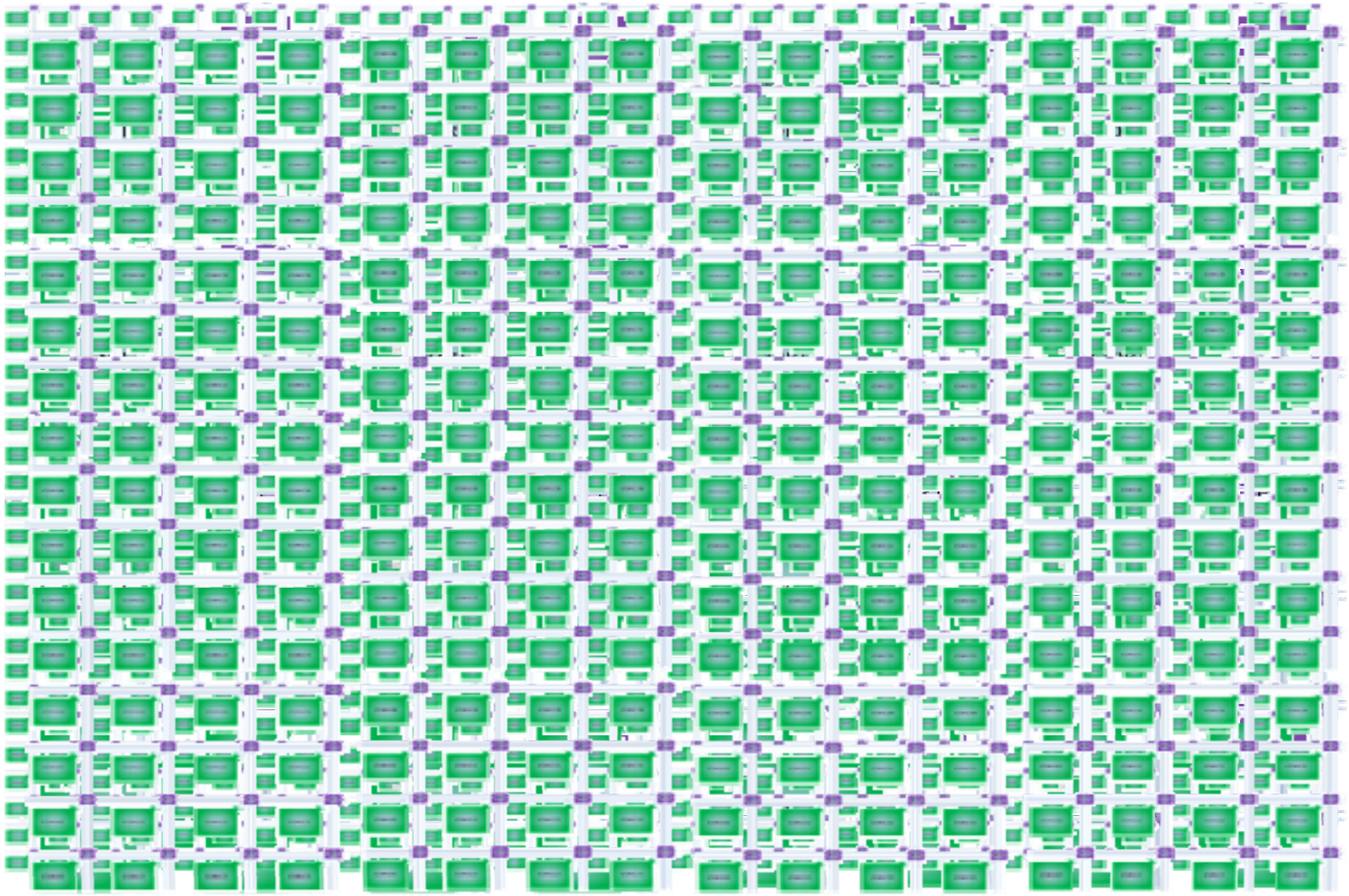






Comparison among State-of-art Processors

Processors	Freq. (GHz)	Process(nm)	Core Num	Die Size(mm2)	Power (W)	Double Precision (GFLOPS)	Energy Efficiency (GFLOPS/W)
Intel Core i7 980 XE	3.2	32	6	240	130	107.55	0.827
Intel Sandy Bridge	3 ~ 4	28	8	370	130	256	1.96
AMD Opteron X12	2.4	45	12	346	130	152	1.16
IBM Power7	3~4.14	45	8	567	100	264.96	2.64
IBM PowerXCELL	3.2	45	9	221	80	100	1.25
Fujstu SPARC fxVIII	2.2	42	8	513	50	128	2.56
Godson-3B	1.0	65	8	300	40	128	3.2



Many-Core Challenges: The 5 P's

- **P**ower efficiency challenge
  - Performance per watt is the new metric – Dark Silicon will appear in 2020 when facing 11nm process technology
- **P**erformance challenge
  - How to scale from 1 to 1000 cores – the number of cores is the new Megahertz
- **P**rogramming challenge
  - How to provide a converged many core solution in a standard programming environment
- **P**arallelism challenge
  - How to exploit parallelism through software and hardware codesign
- **P**latform challenge
  - How to maximize the usability, such as, machine, runtime system (OS), compiler & library, and debug mechanism

What we could help.....

Productivity Side:

- Most sequential programmers are not ready to switch into parallel programming
  - conservative programming model and make incremental improvement?
- Unique programming model cannot solve all problems efficiently;
  - support multiple programming features efficiently?
- Locks are messy
  - new way to eliminate deadlock?

……

Performance Side: ***Take this carefully, because it may affect productivity !***

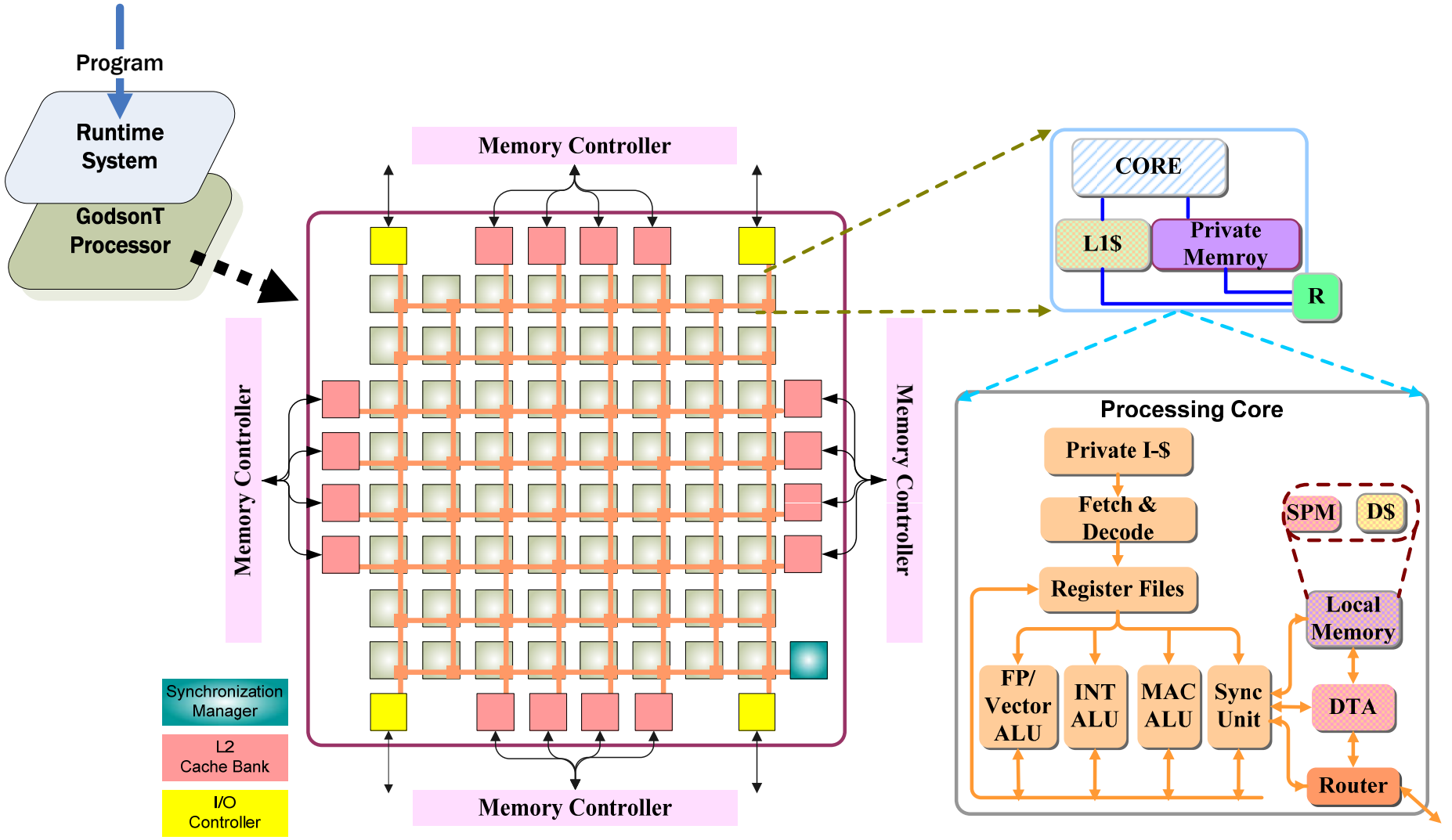
- On-chip synchronization is fast;
  - handle all synchronizations on-chip?
- Flops are cheap, memory communications are expensive;
  - trade flops for communication latency?
- Fine-grained parallelism should be taken into consideration;
  - enable data-driven thread execution on chip?

……

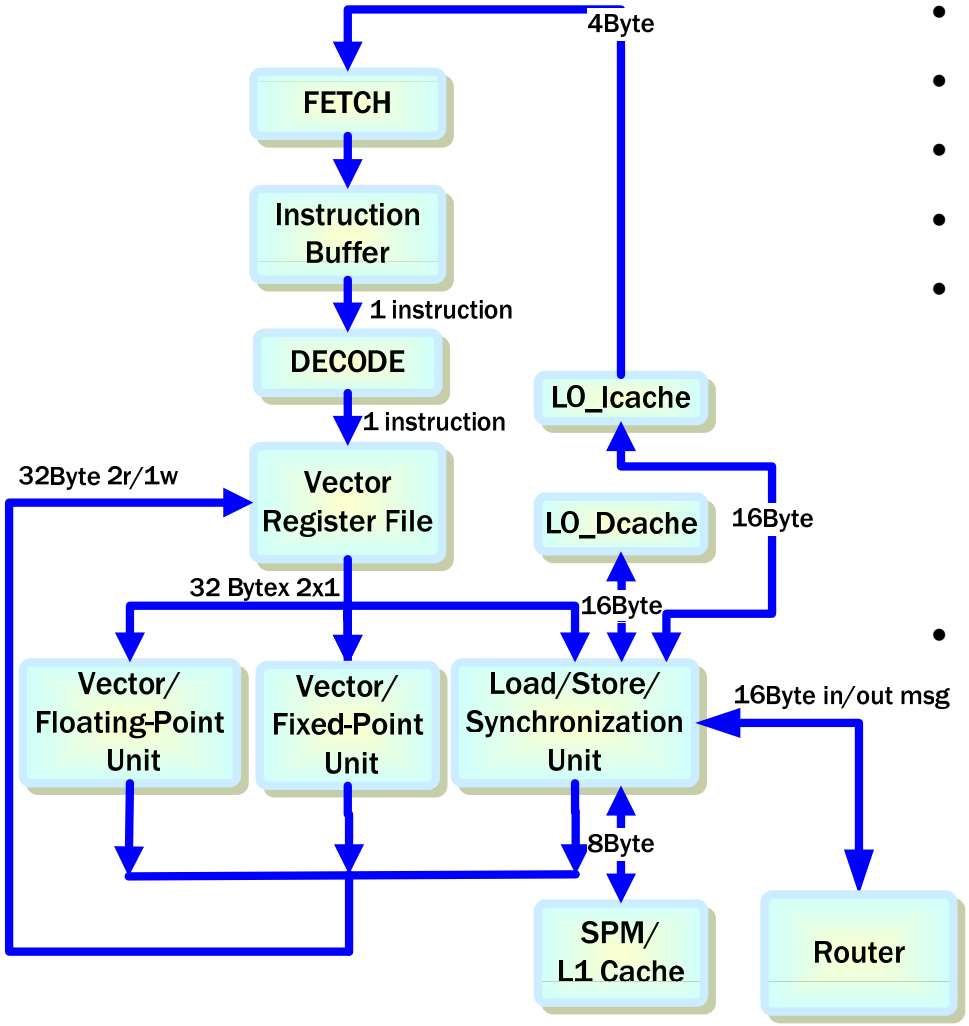
Outline

- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

Overview of Godson-T Architecture



Processing Core

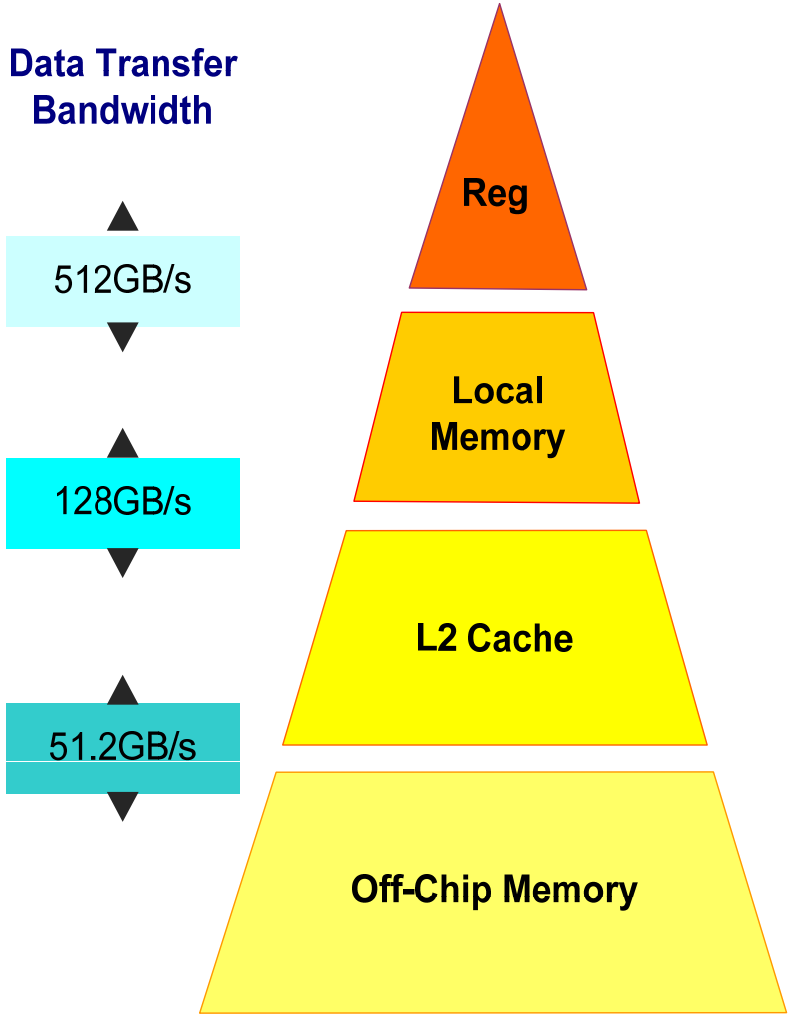


- ISA: MIPS (user), SIMD-ext., sync-ext.
- 8-stage pipeline
- Dual-issue per thread
- Fast level-1 memory
- 16KB private memory
  - Automatically mapped into stack address space
  - Full/empty bit tagged on each 64-bit slot enables efficient producer-consumer style synchronization
- Communication with external modules through message packets

Interconnection

- Separated routers with each processing core
  - Static XY wormhole routing
  - Round-Robin arbitration
  - Two independent physical networks
  - Duplex 128bit link for each network
- Guaranteed in-order point-to-point communication
  - Deadlock-free & livelock-free
  - Scalable and power-efficient for MESH topology
  - Low latency core-to-core communication
  - Separated networks allowing traffic segregation and tolerating burst DMA transfer

Memory Hierarchy



Each processing core with 32 fixed-point and 32 floating-point registers

32KB local memory, including L1$D, SPM

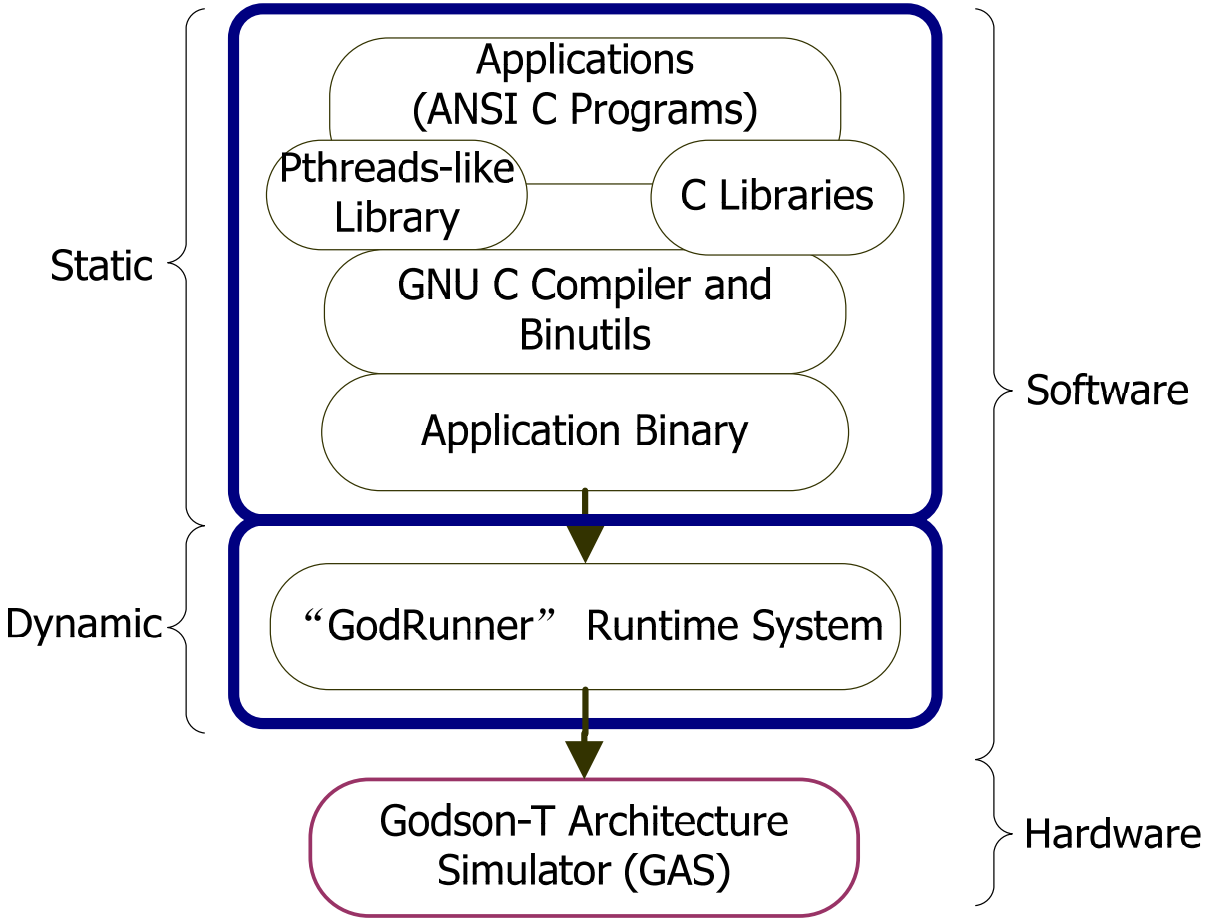
16 address-interleaved L2 cache banks, 128KB each

4 DDR3-1600 memory controllers

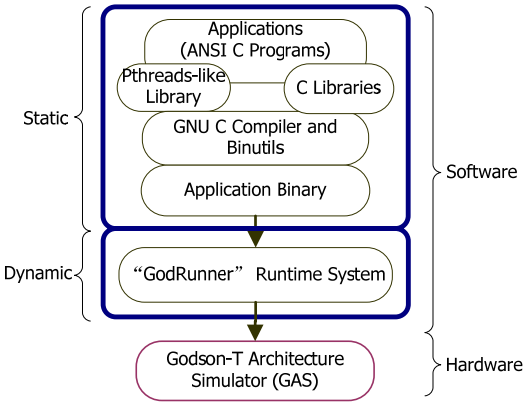
Outline

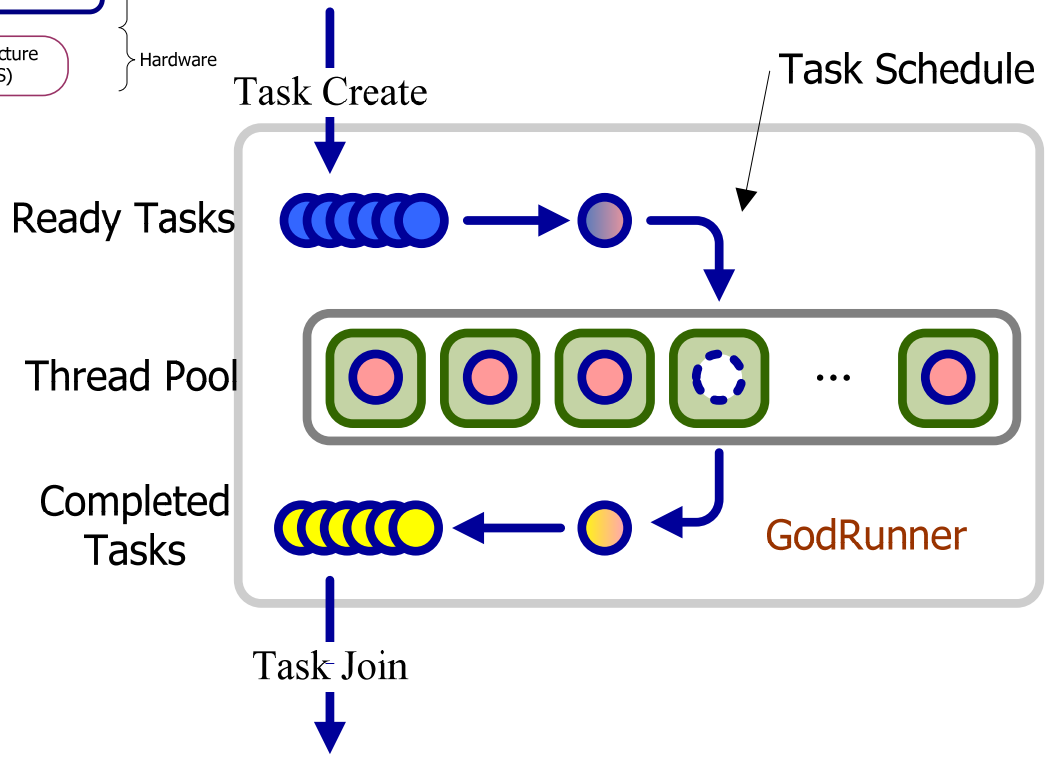
- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

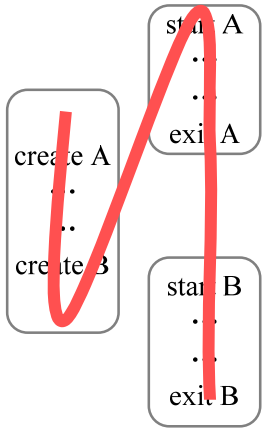
GodRunner: Software Runtime System

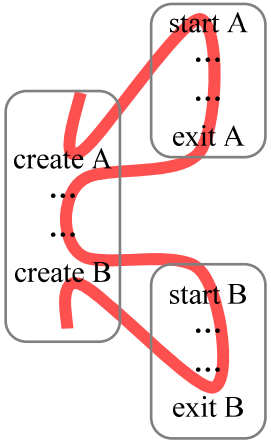


GodRunner: Software Runtime System









Outline

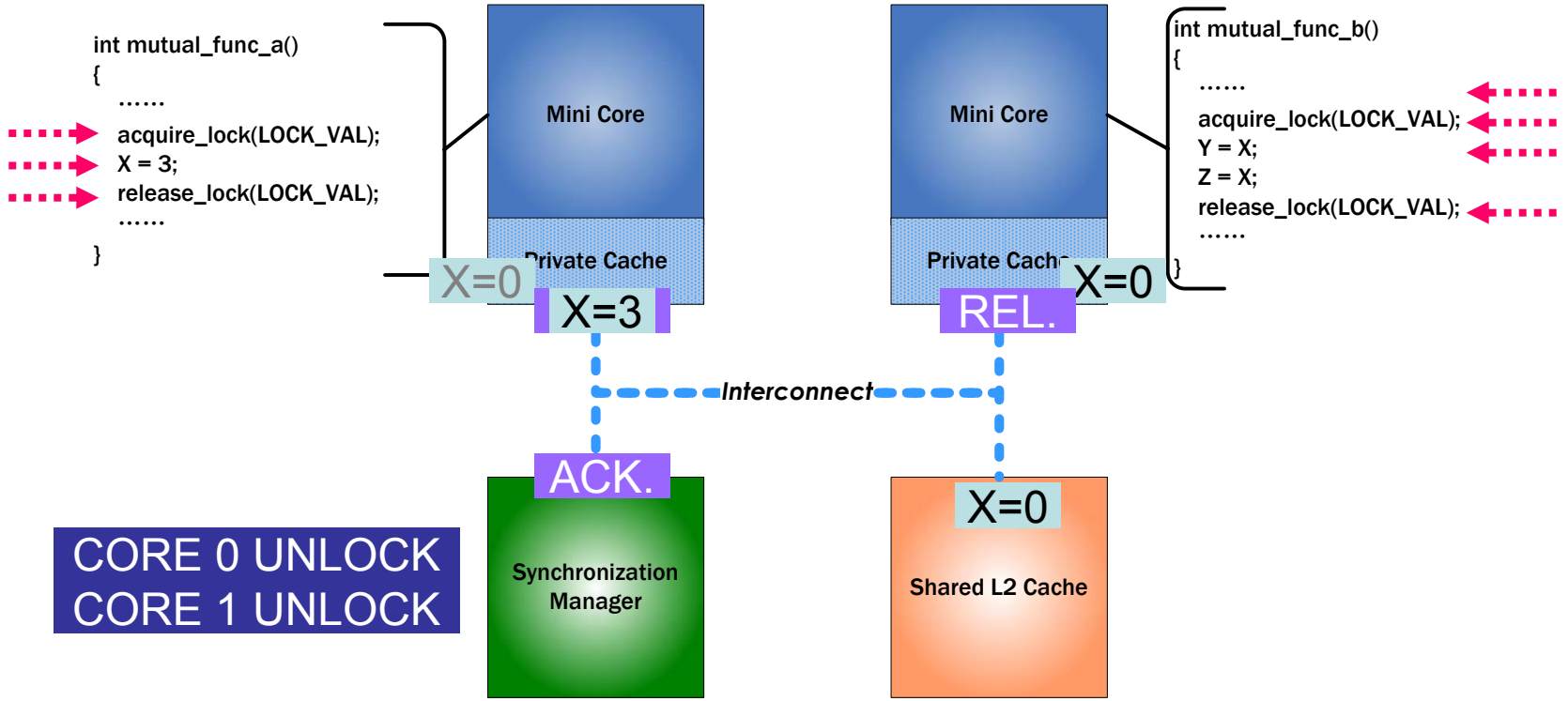
- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

Architectural Supports for Multithreading

- Thread Communication
  - *Cache Coherence Protocol*
  - *Orchestrating Data Movement*
- Thread Synchronization
  - *Synchronization Without Memory*
  - *Full/Empty Bit Synchronization*

Lock-Based Cache Coherent Protocol

- **Lazy cache coherent protocol**
  - Pure mutual-exclusion synchronization instruction without memory accesses;
  - Eliminate busy-waiting for locks;
  - More scalable than bus-snoopy and directory-based cache protocol;
  - Enable hardware deadlock detecting.

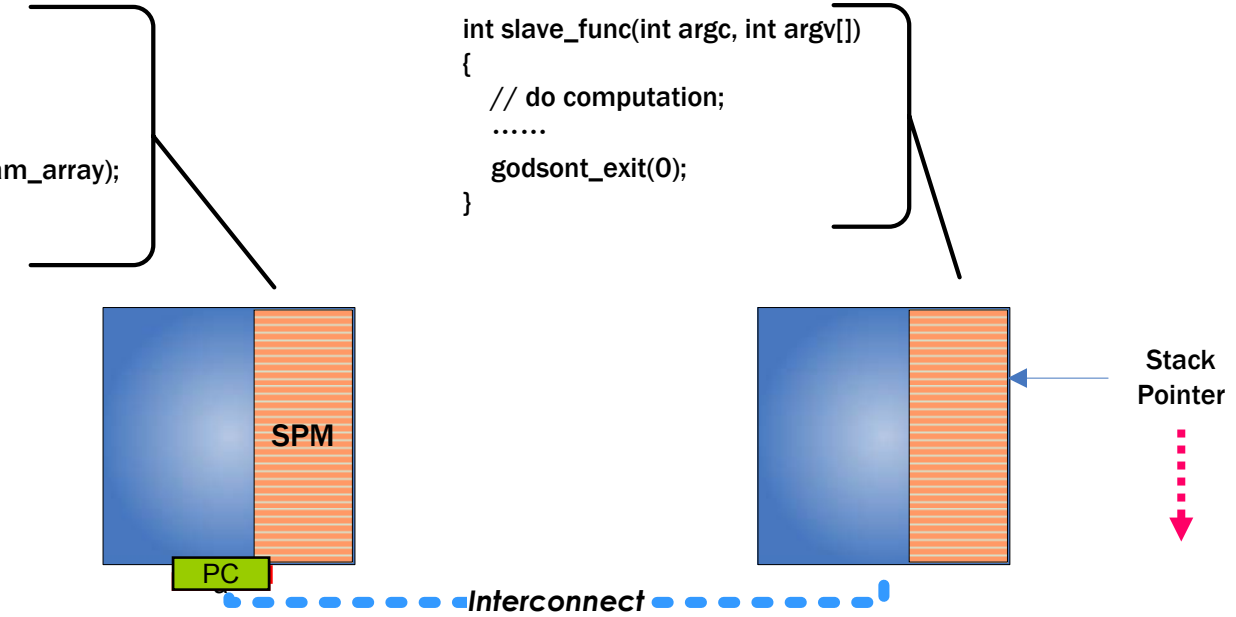


Pthread-like Thread Execution Using Private Memory

- Support master-slave style thread execution;
- Programming easily: C programming model + Pthread API;
- Compatible with large amount of conventional codes on shared memory system.

```
int mater_func()
{
   ……
   int thread_id =
     godsont_ create_thread(slave_func, 2, param_array);
   ……
}
```

```
int slave_func(int argc, int argv[])
{
   // do computation;
   ……
   godsont_exit(0);
}
```

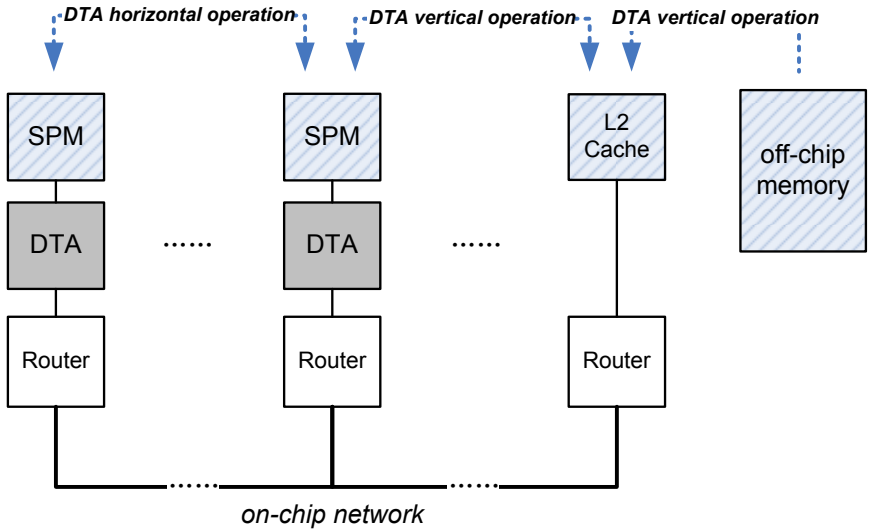


1. **Master transmits parameters to slave;**
2. **Master setups PC and register context for slave;**
3. **Slave function begins execution, private memory acts as a local stack.**

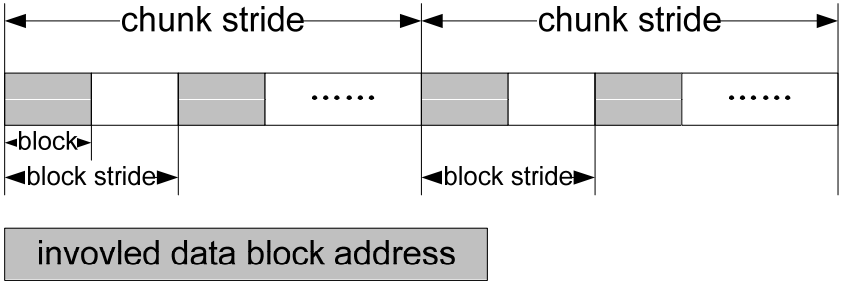
Stack v.s. L1-D$ accesses ≈ 30 **(8x8 matrix multiply, no opt.)**

Significantly reduce cache pressure in some applications!

Data Transfer Agent (DTA)



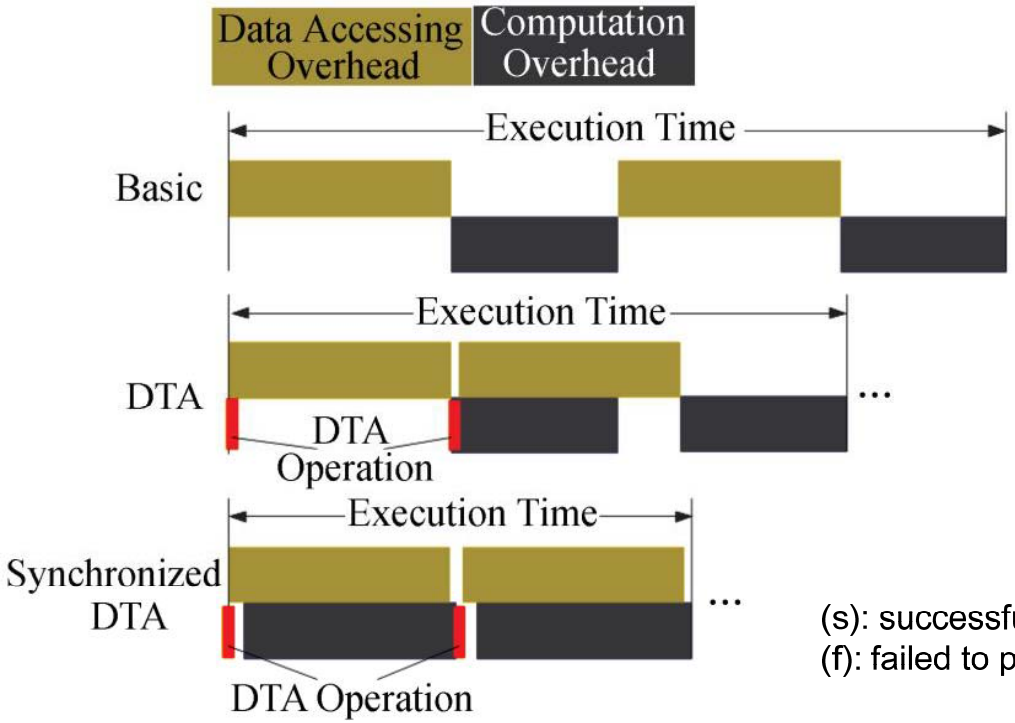
(a) vertical and horizontal DTA operations

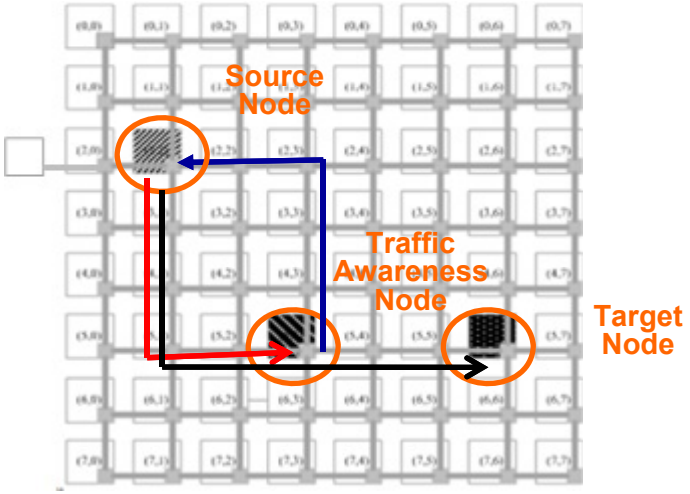


(b) 2D strided DTA operations

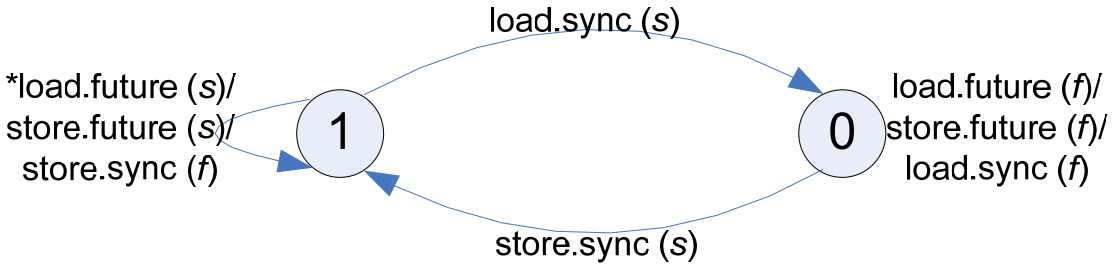
- **Programmable asynchronous data transfer agent**
  - Support vertical and horizontal DTA operations (such as prefetch)
  - Data transfers between multi-dimension addresses (such as matrix inversion)
  - Network load perception, automatically flow control (improve bandwidth-efficiency)
  - Support fine-grain synchronous operations

Synchronized DTA Operations





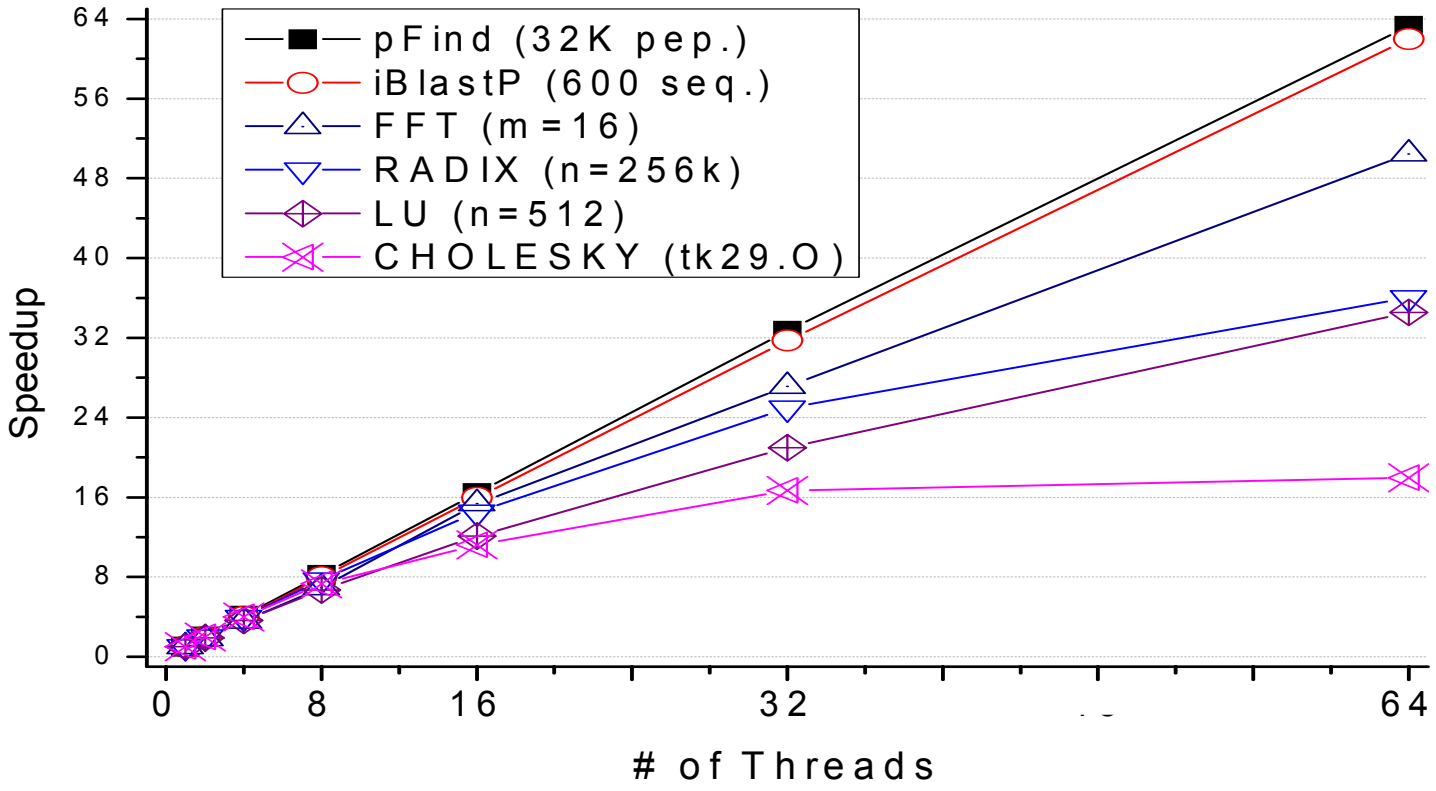
(s): successfully perform on the state of full/empty bit
(f): failed to perform on the state of full/empty bit



Outline

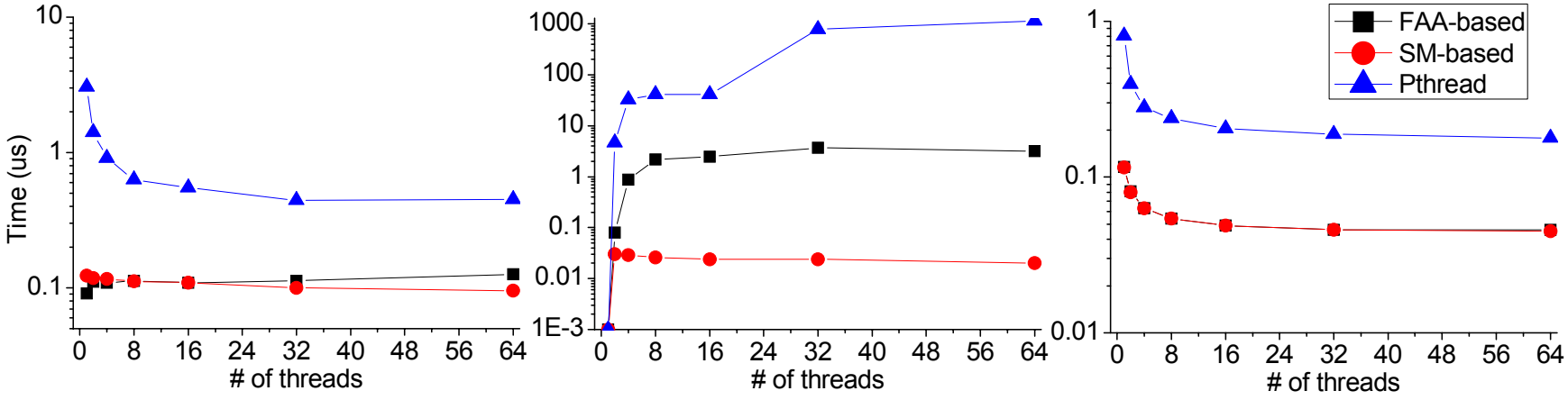
- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

Evaluating the Scalability of Performance

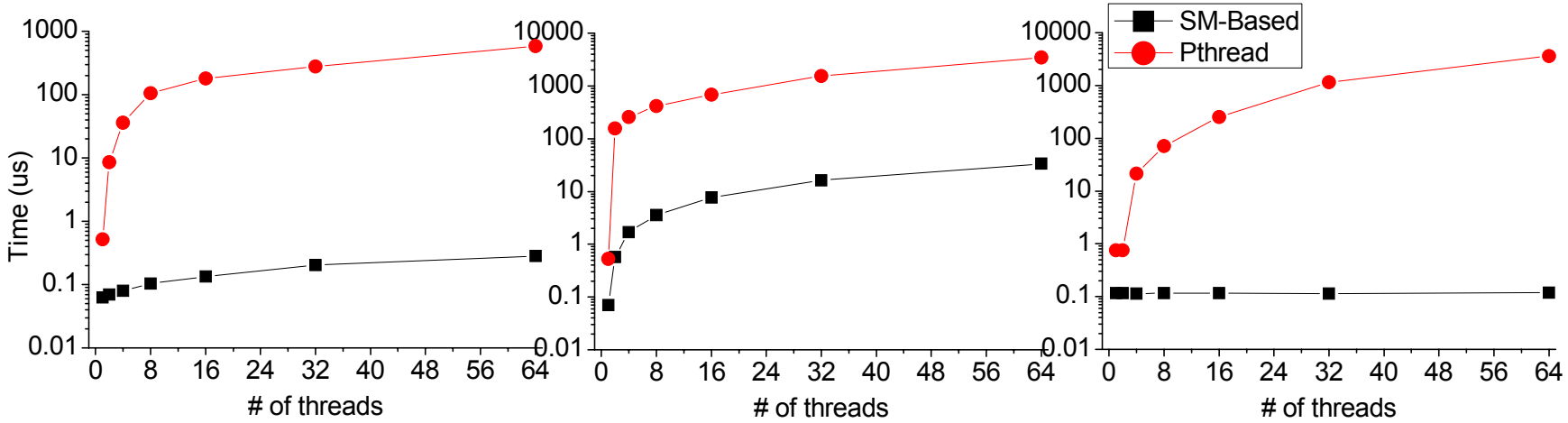


Speedup of the bioinformatics and SPLASH-2 benchmarks on Godson-T

Evaluating Synchronization without Memory

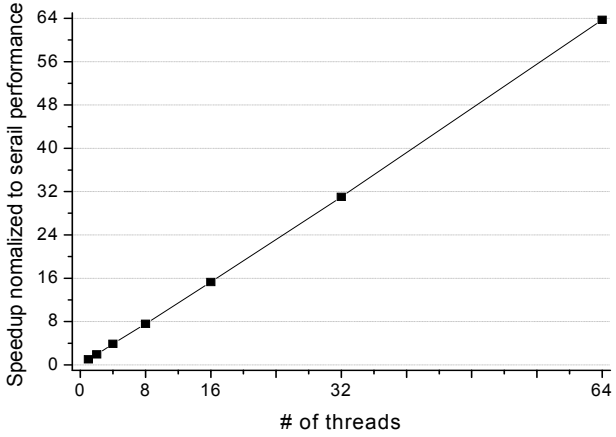


(a) lock overhead without lock contention **(b)** lock transferring overhead **(c)** average time for each load



(a) barrier overhead without workload **(b)** barrier overhead with load imbalancing **(c)** average time of each load

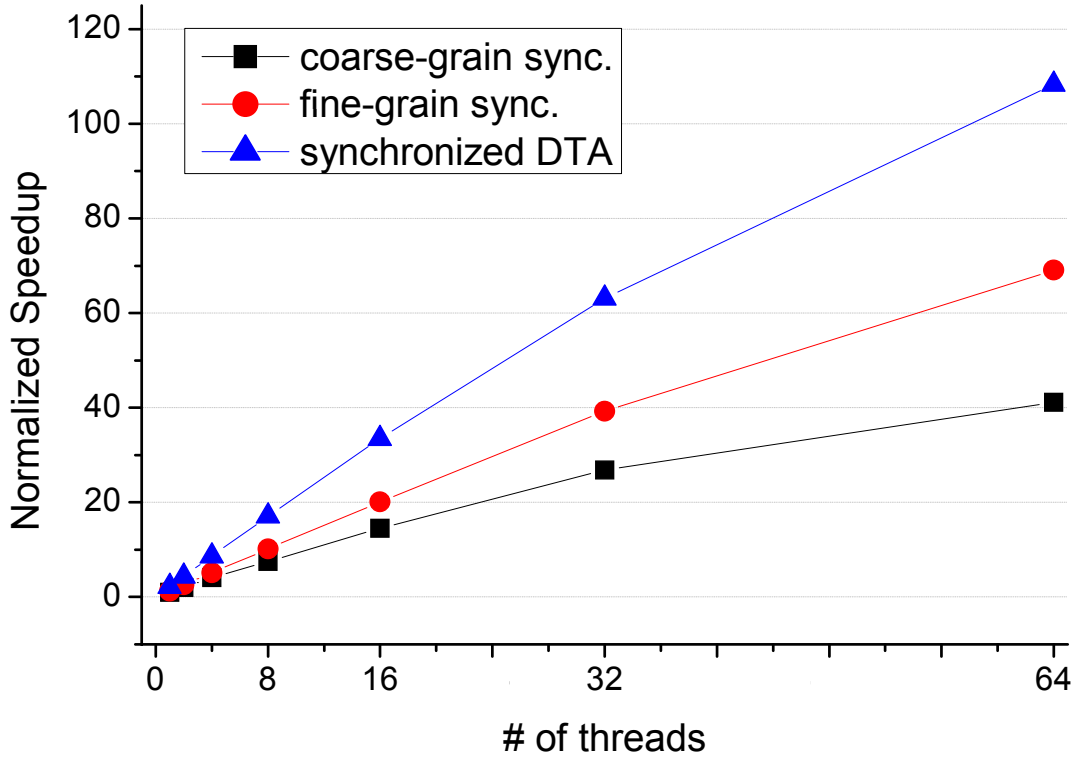
Evaluating Full-empty Bit Synchronization



Speedup of 2-D Wavefront

Livermore loop 6

```
for ( i=1 ; i<n ; i++)
    for ( k=0 ; k<i ; k++)
        W[i] += B[k][i] * W[(i-k)-1];
```

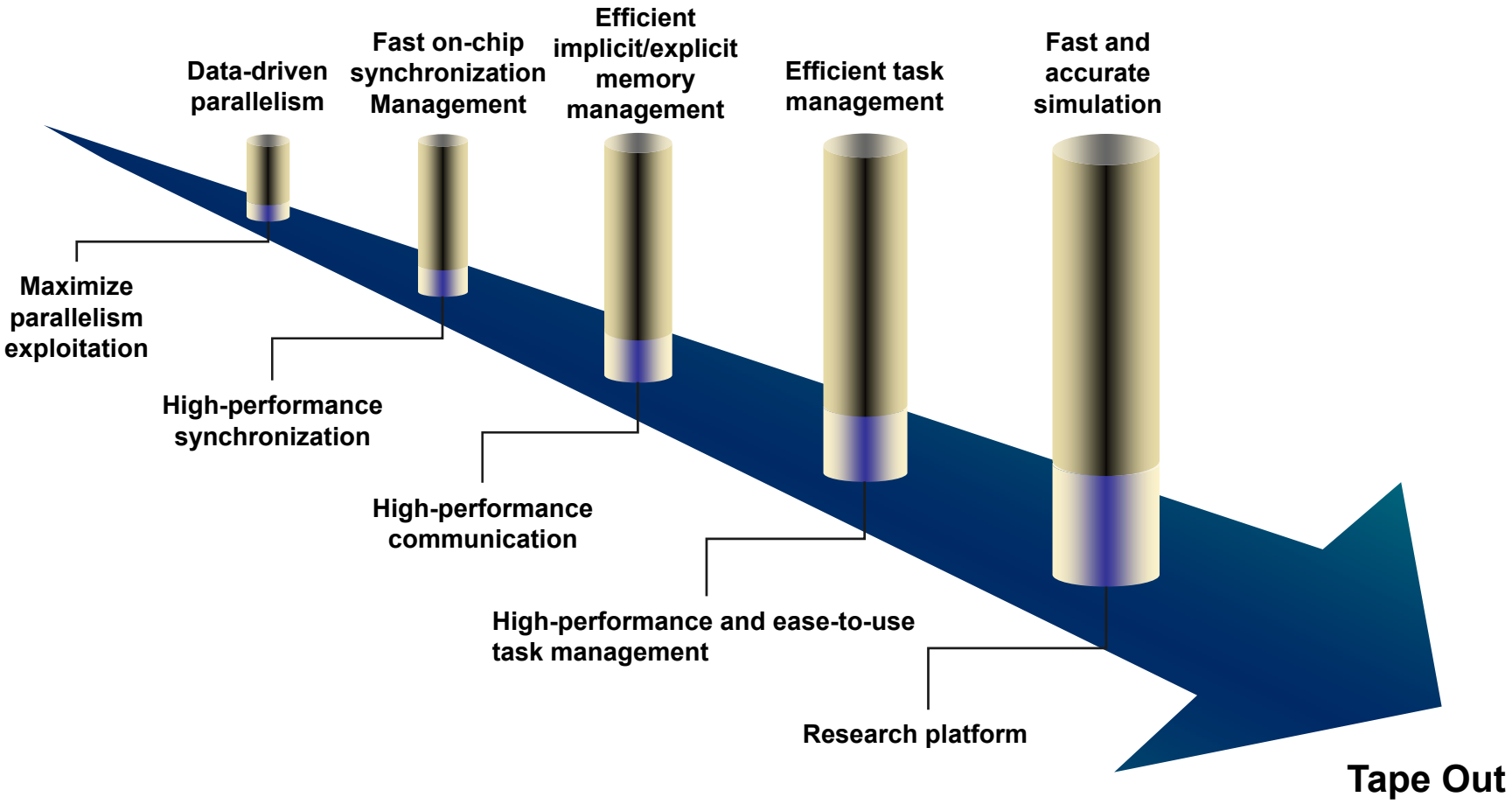


Speedup of Livermore Loop 6

Outline

- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

Godson-T Design and Implementation



曙光 DAWNING

Top1 Supercomputer in Asia
Top2 Supercomputer in the World

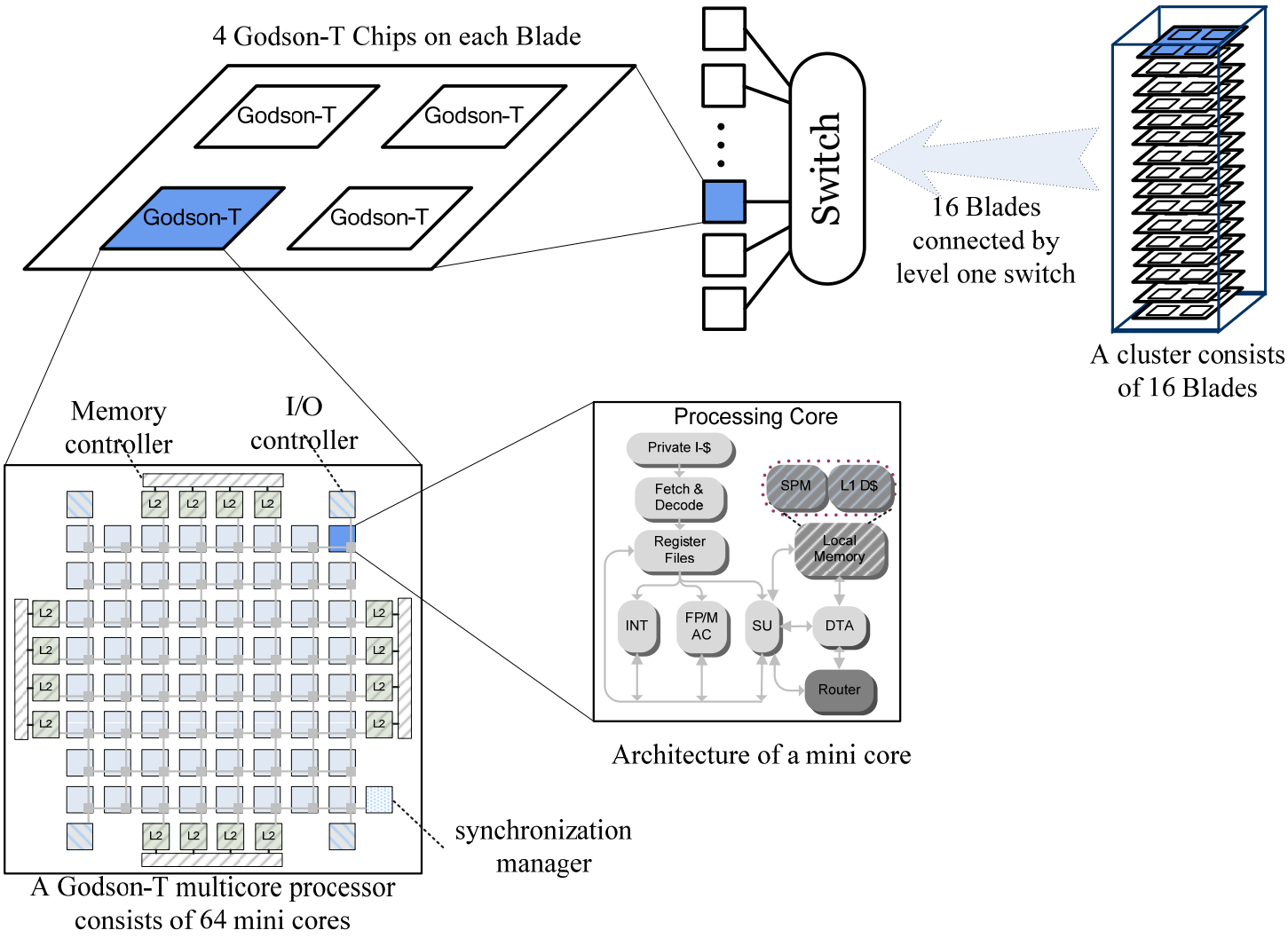
Dawning Nebulae
A Supercomputer of Superior Flexibility and Capability

Unmatched flexibility and capability along with easy integration into your existing data center infrastructure: Dawning Nebulae supercomputers are designed to meet your most demanding business requirements. Its power can be easily harnessed into hundreds or thousands of nodes and systems for large-scale applications and deployments in HPC, Grid Computing and Web Infrastructure, such as governments, education, finance, telecommunications, Internet, biology, meteorology, petroleum, material sciences, aerospace, and various other industries. The remarkable achievement of Dawning Nebulae represents the highest level of China's HPC technologies and innovations.

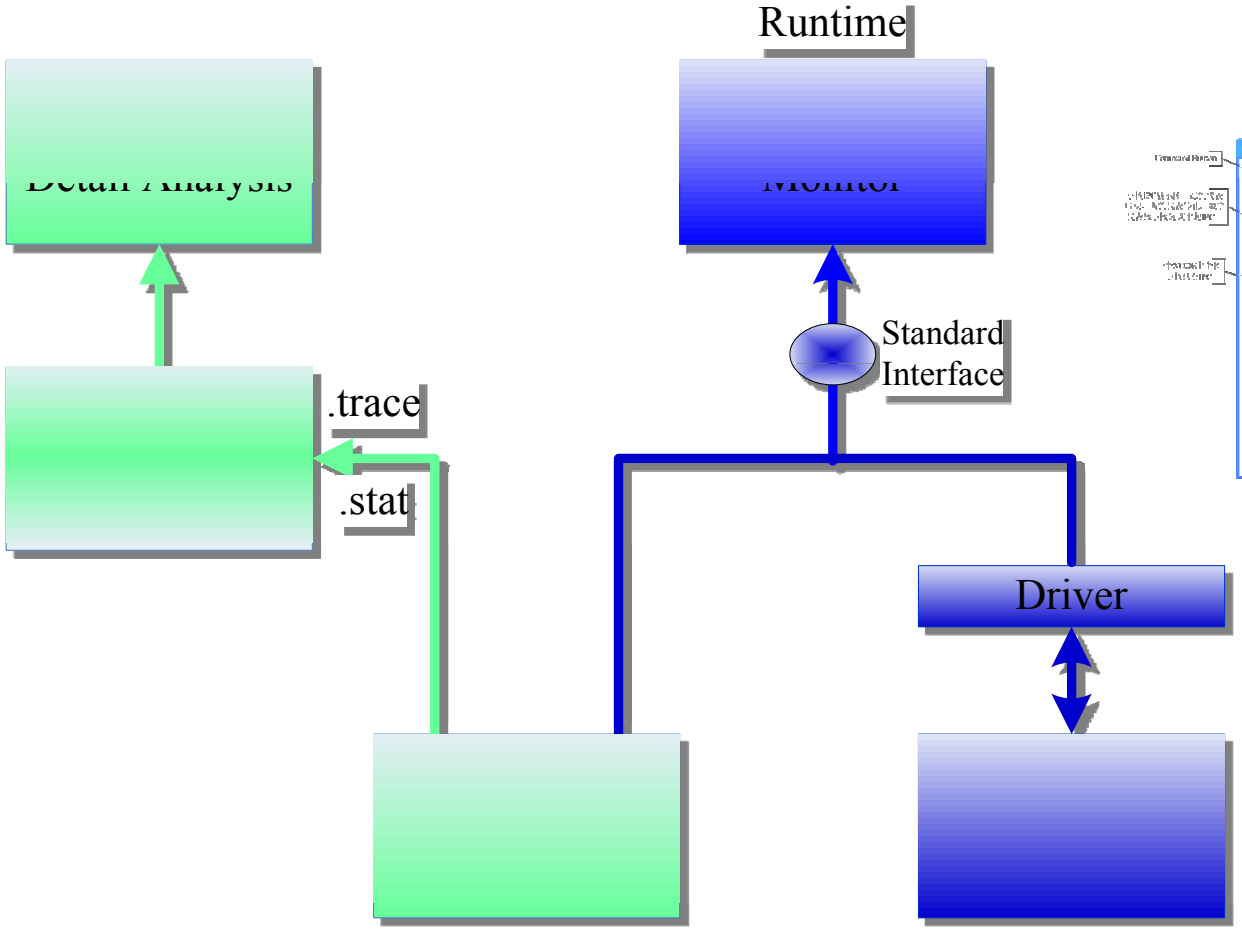
未来，"云"知道

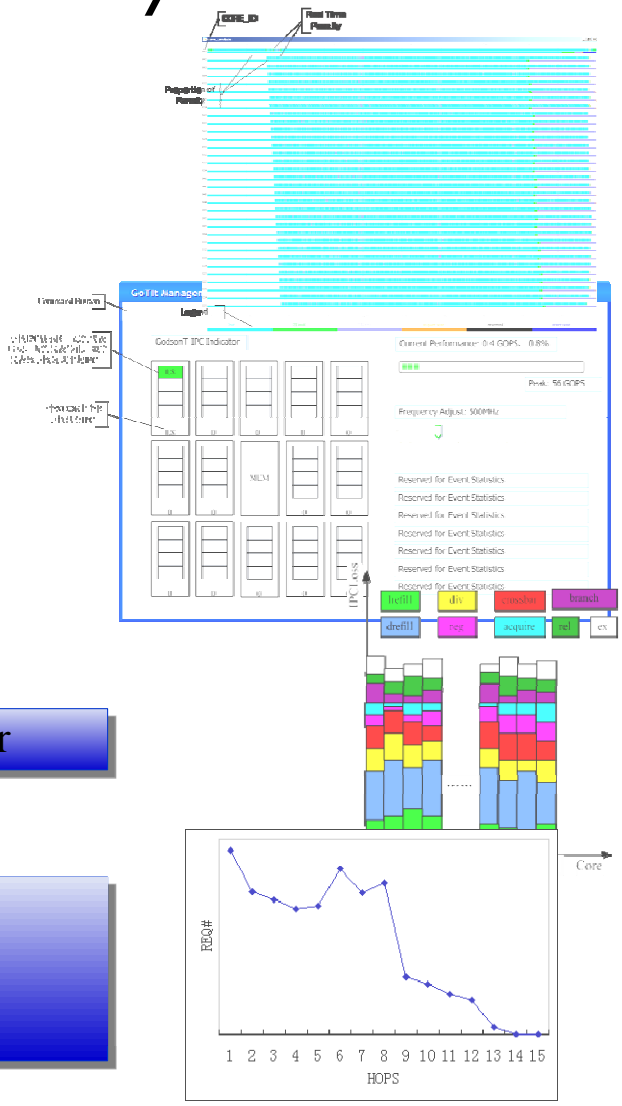
Dawning Information Industry Co., Ltd
www.dawning.com.cn

A Parallel 4096 Cores Cluster Simulator

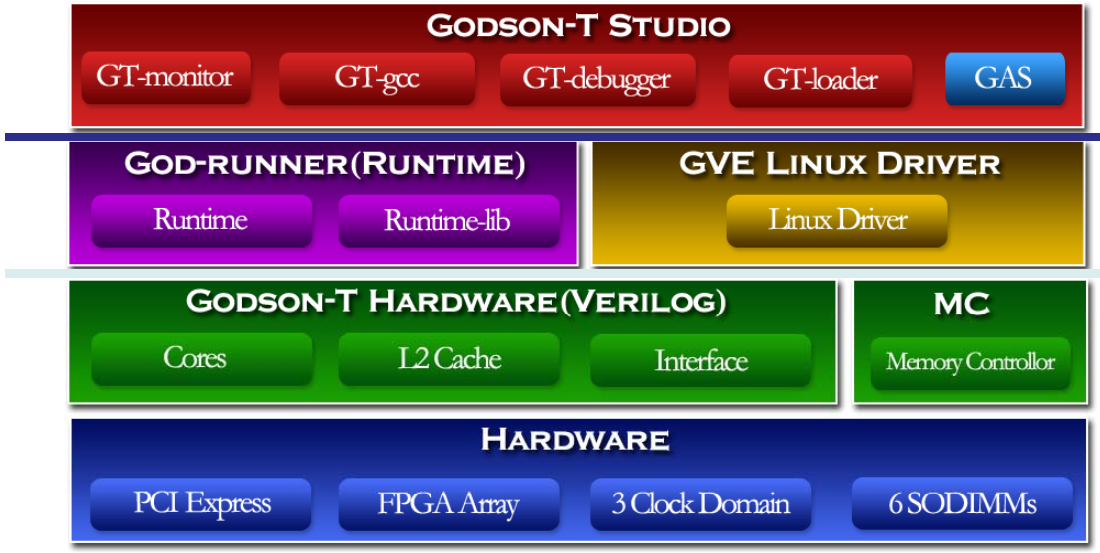


Monitor/Analysis/Evaluation/ Platform





Software



GVE Software Stack

GT-Monitor

Watch GT Running Status

GT-Gcc

Compiler for Godson-T

GT-Debugger

Troubleshooting tools

GT-Loader

Load Runtime/Program

Runtime

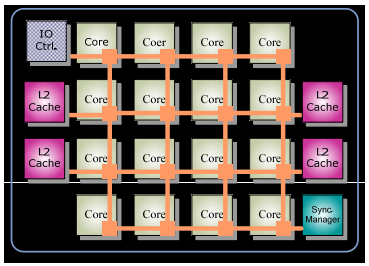
Schedule Tasks on Cores

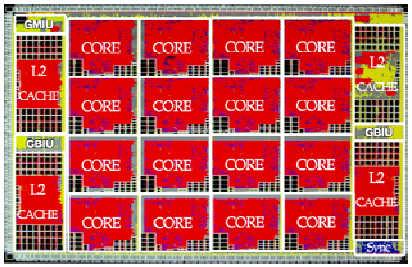
Linux Driver

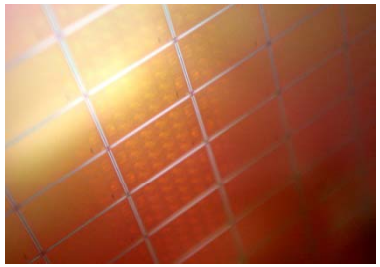
For PCI-E Communication

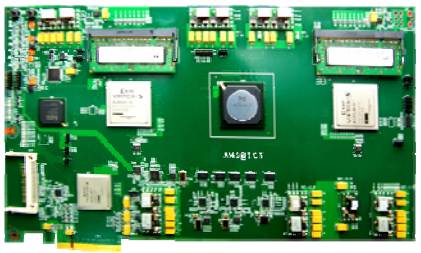
Godson-T Many-Core Prototype

- 64 light-weight processing mini cores currently
- 16-core sample:130nm, 230mm^2
- Target to domain-specific parallel acceleration
- Data Transfer Agent (DTA)
- Share scattered L2 Cache
- High Speed on-Chip interconnect
- 2 FMAC in One Core









Chip On Mind Concept
RTL

Chip on EDA
PHY Design

Chip on Silicon
Wafer

Chip in Package
Package

Chip on Board
System

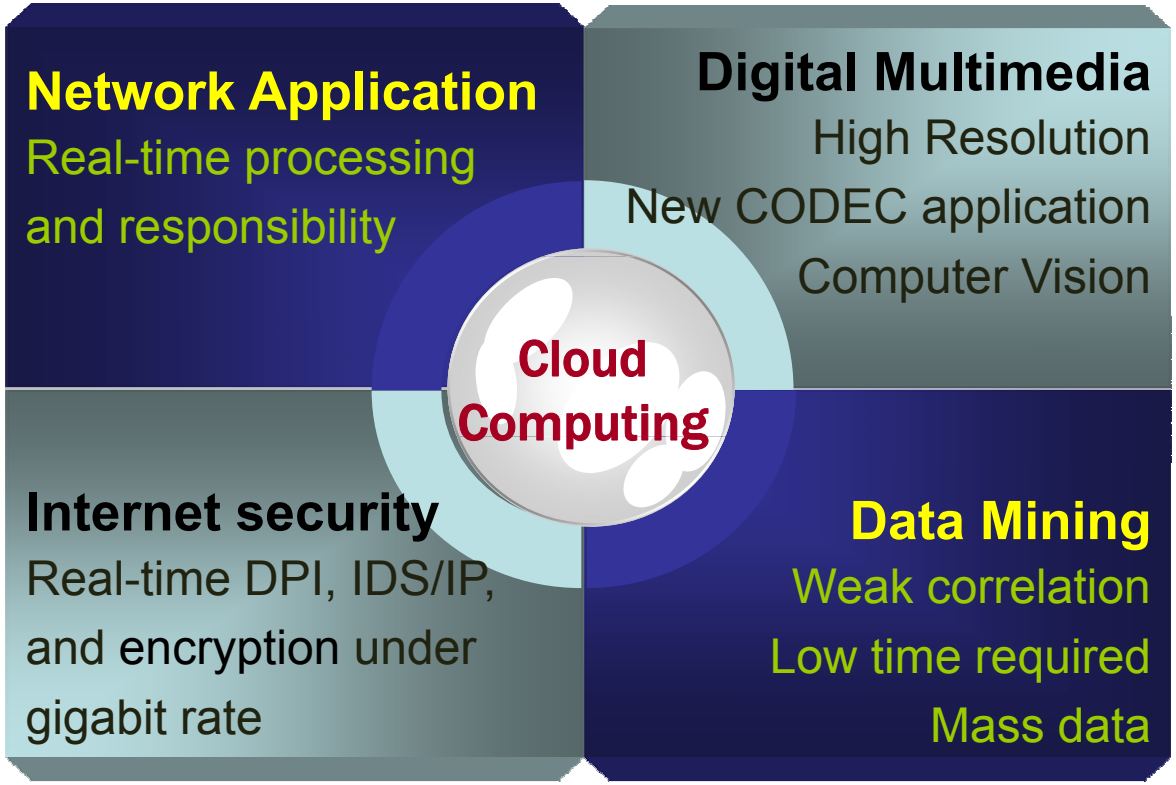
Outline

- Motivation
- Godson-T Architecture and Evaluation
  - Godson-T architecture overview
  - Software runtime system
  - Architectural supports for multithreading
  - Experimental results analysis and evaluation
- Godson-T Design and Implementation
- Conclusions and Future Work

Conclusions of Godson-T

- Scalable solution to achieve performance without losing programmability
  - Support hybrid programming models efficiently
- Built a reasonable many-core architecture framework for further study and research
  - Efficient runtime system
  - Cache coherence protocol
  - Asynchronous data transfer agents and hardware-supported synchronization mechanisms
  - Pthreads-like programming model, and versatile parallel libraries
- Efficient architecture exploiting abundant DLP/TLP in applications

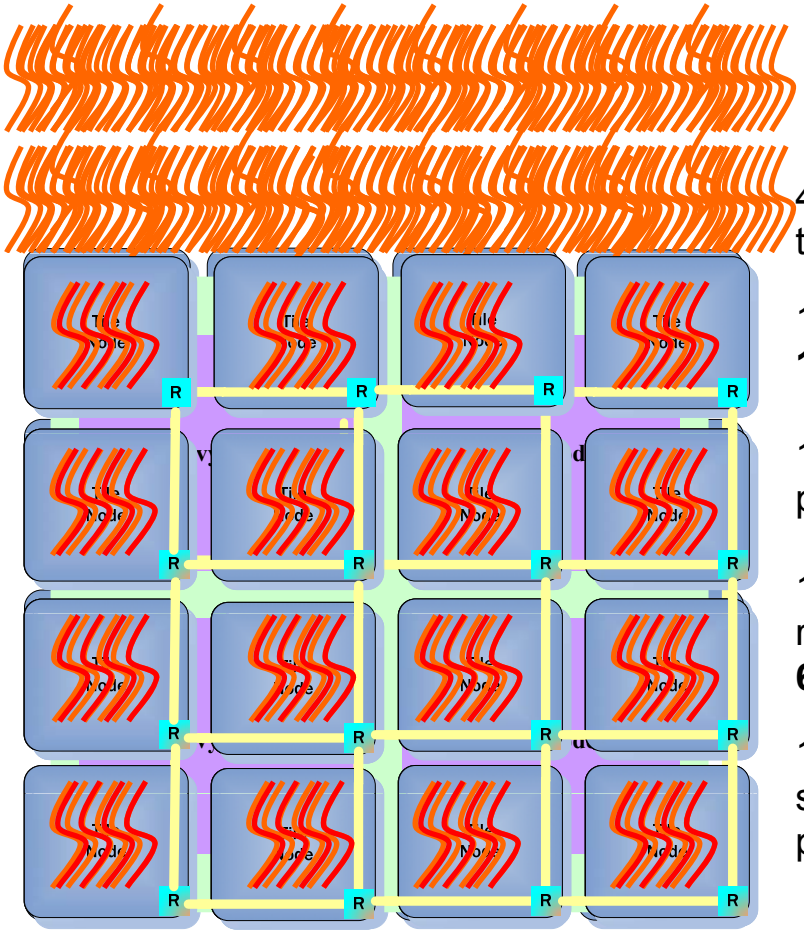
Future Work



Data-Intensive＋CPU-Bound → **Data-Intensive＋Transaction Intensive**

High Performance → **High Throughput**

Thousand Threads Running on Chip



4 1GHz heavy cores process **8** threads in Ts'

16 1GHz light cores process **16** threads in Ts'

16 1GHz SMT light cores process **4X16** threads in Ts'

16 1GHz SMT light cores with redundancy reducing process **6X16** threads in Ts'

16 1GHz SMT light cores with statement and redundancy reducing process **8X16** threads in Ts'

Thanks
&
QA