

A Scalable Directory Architecture for Distributed Shared-Memory Chip Multiprocessors

Huan Fang and Mats Brorsson

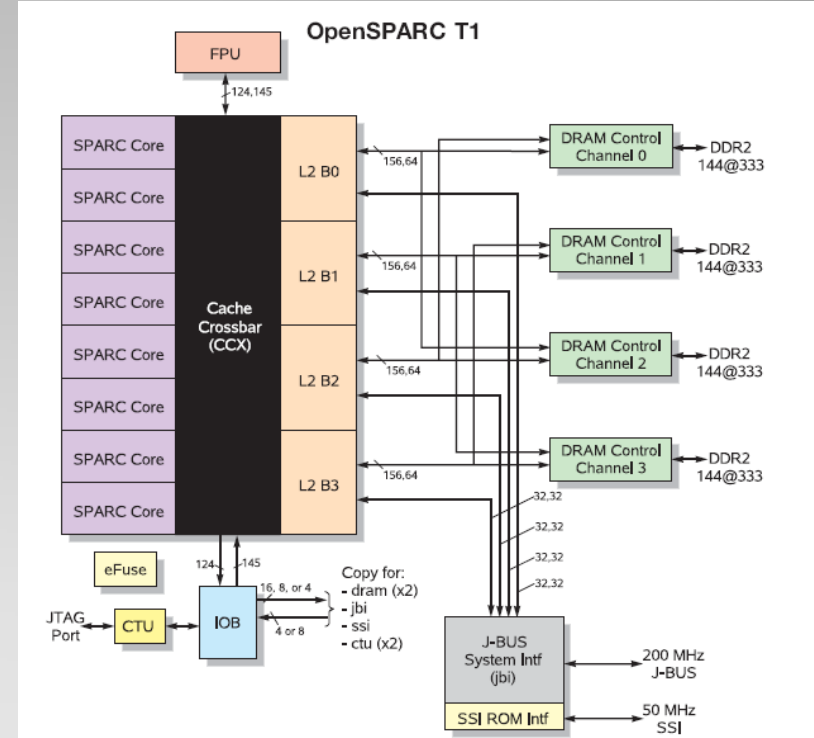
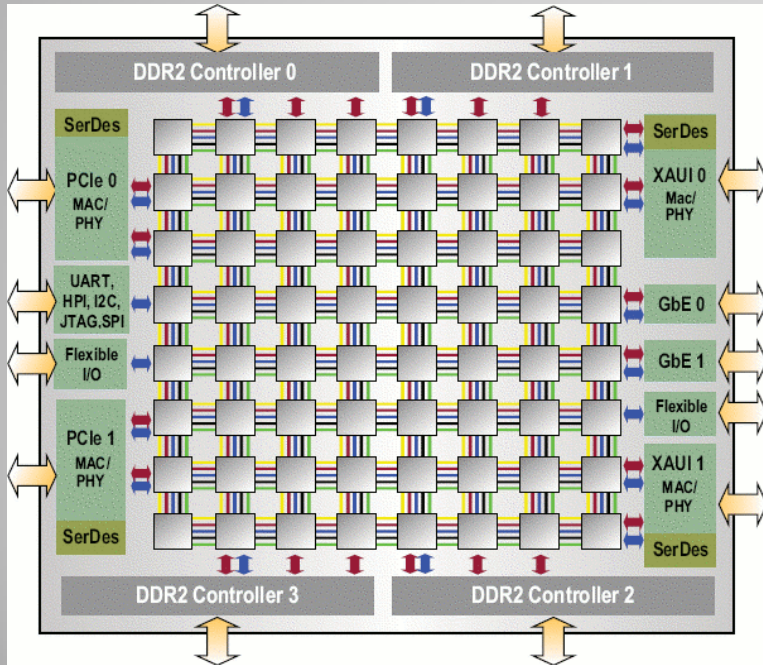
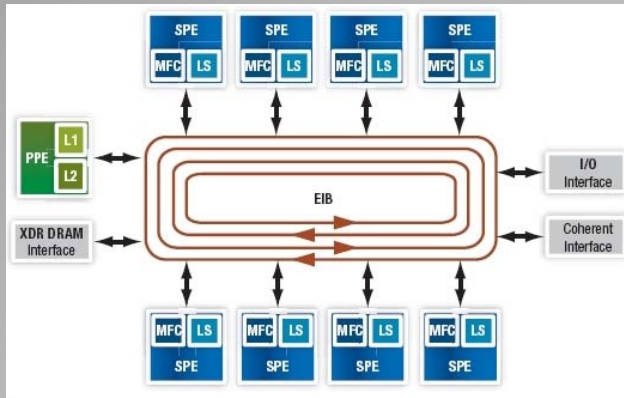
KTH, School of Information and Communication Technology
{huanf, matsbror}@kth.se



**ROYAL INSTITUTE
OF TECHNOLOGY**

Chip Multiprocessor

- A commercialized technique in CPU design that combines two or more processor cores on a single piece of silicon chip.
- To build high-performance processors by exploiting parallel processing capacities while keeping the power dissipation low without increasing clock frequencies.



Cell BE TILE 64 Processor SUN Niagara

- **Micro Architecture**

How do we manage on-chip resources like caches, memory controllers and routers.

- **Interconnection**

A more scalable on-chip network is needed rather than global broadcast technique for future CMPs.

- **Scalability**

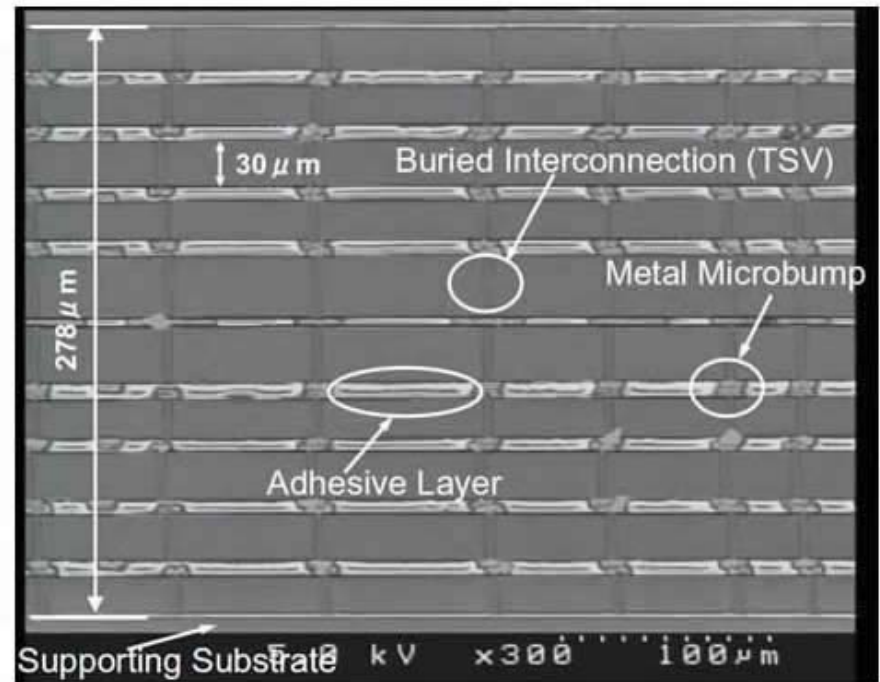
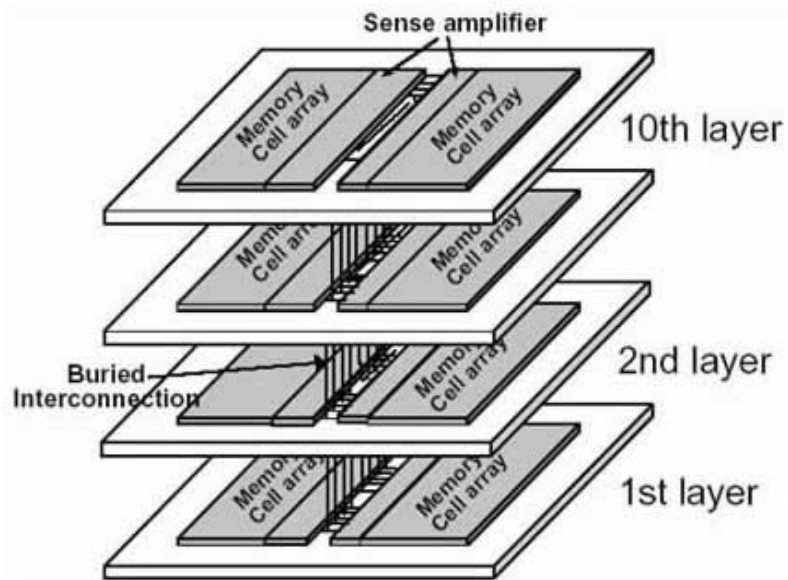
A system whose performance improves after adding hardware, proportionally to the capacity added, is said to be a scalable system.

Considerations...

In this work ...

- A novel directory-based protocol is designed and tested to verify if it is suitably efficient and practical when applied to large situations.
- New cache and directory structure
 - Comparable performance to an existing SCMP protocol*
 - The memory overhead of directory is reduced to 10% as that of SCMP.*

3D packaging



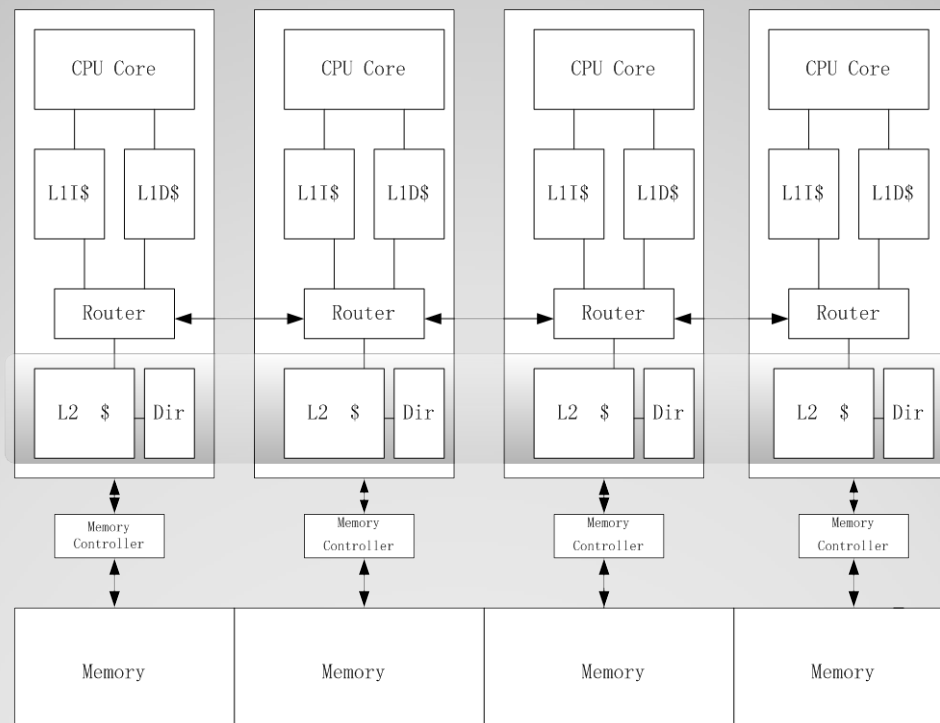
Snoop-based protocols

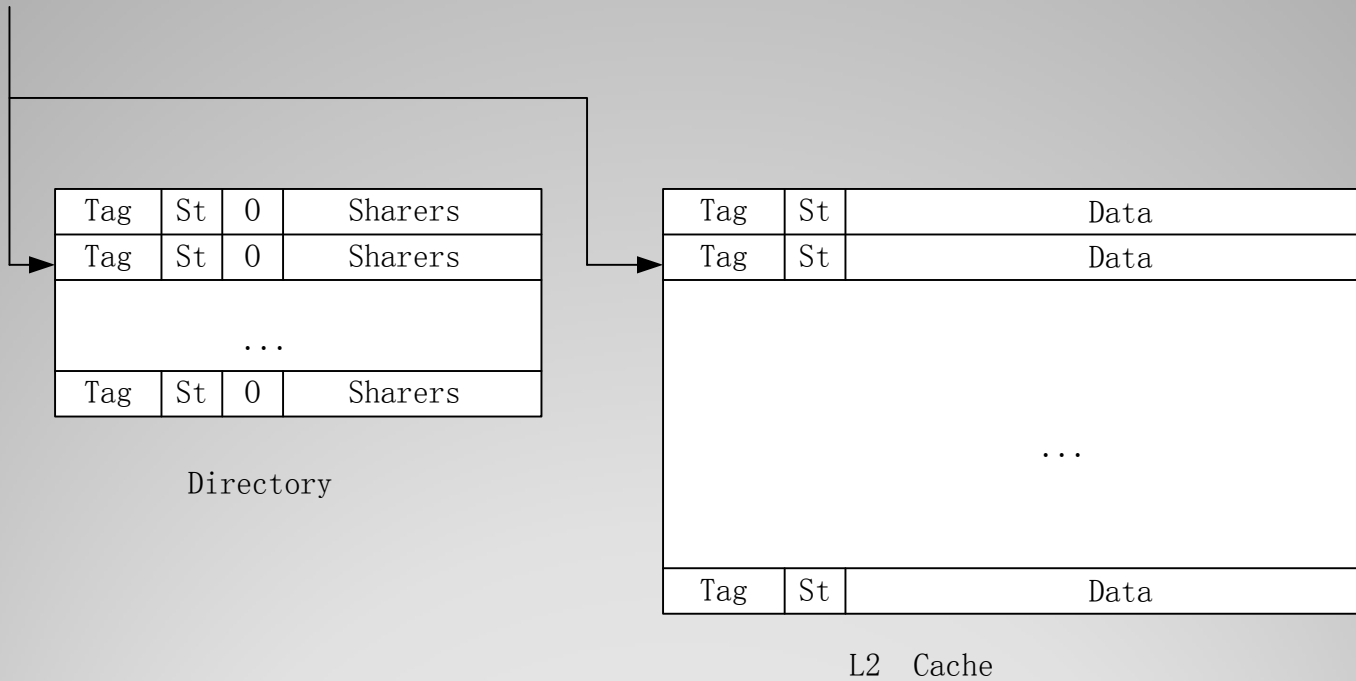
- *Every cache that has a copy of the data from a block of physical memory also has a copy of the sharing status of the block, but no centralized state is kept.*
- *The caches are all accessible via some broadcast medium (a bus or a switch), and all cache controllers monitor or snoop on the medium to determine whether or not they have a copy of a block that is requested on a bus or switch access.*

Directory-based protocols

- *The sharing status of a block of physical memory is kept in just one location, called the directory. Information in the directory includes which caches have copies of the block, whether it is dirty, and so on.*

System Architecture Hardware Model





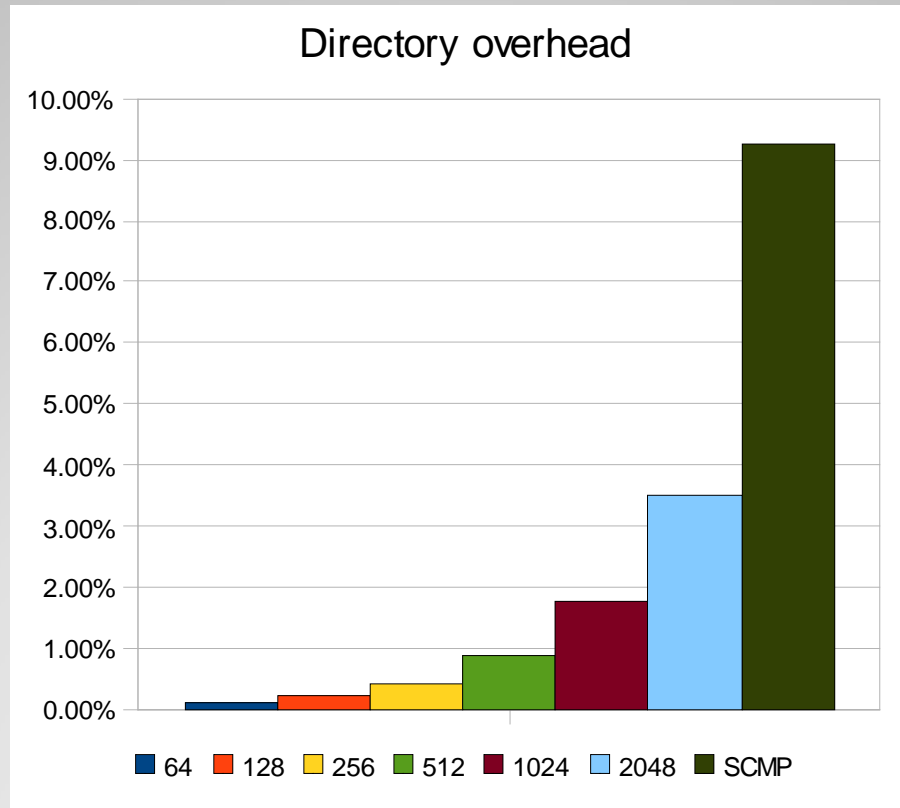
Directory and L2 cache organization

Tag	St	0	Sharers	Data
Tag	St	0	Sharers	Data
...				
Tag	St	0	Sharers	Data

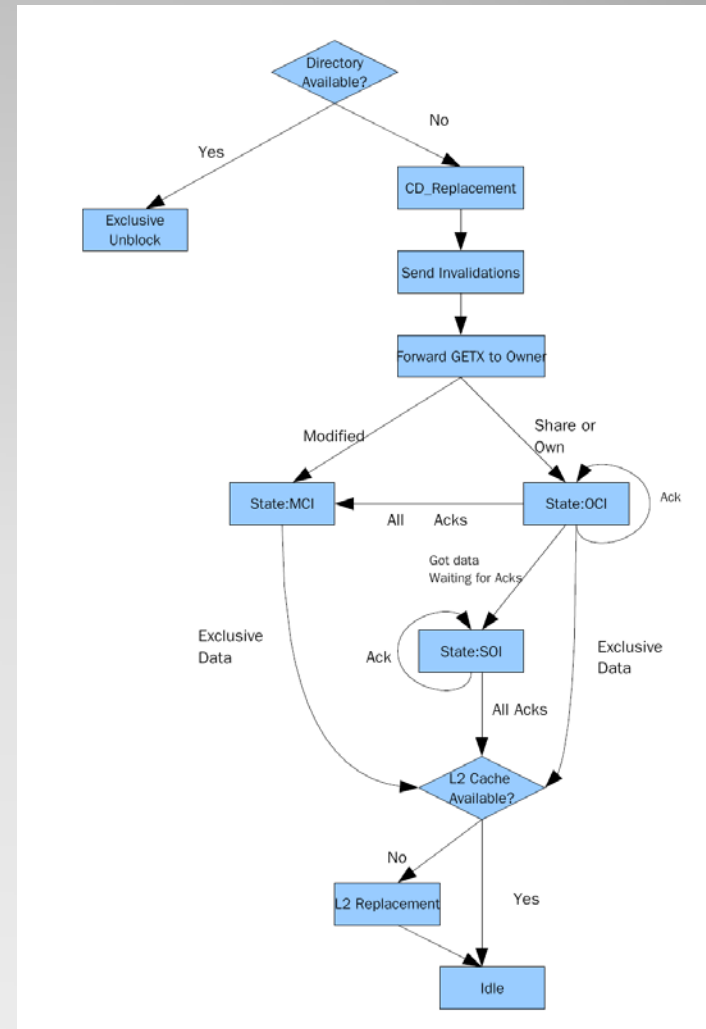
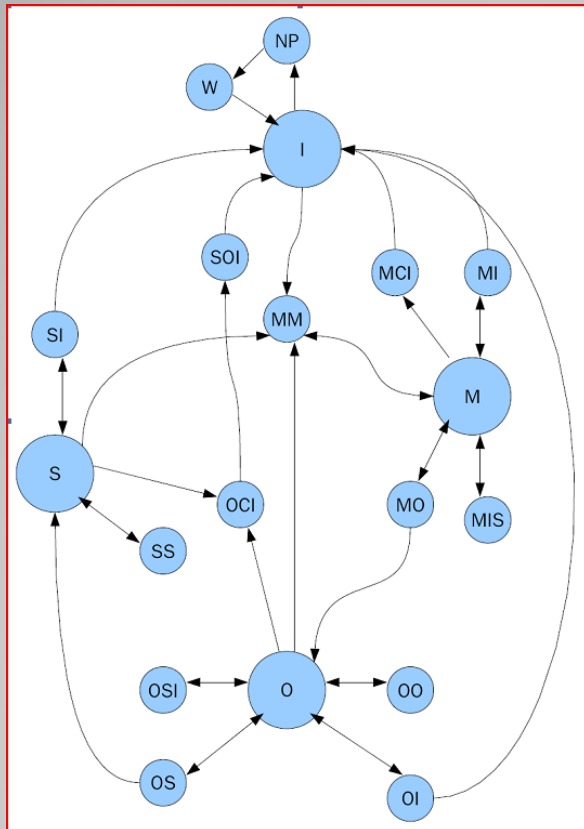
L2 Cache

SCMP Protocol for comparison

Overhead comparison



L2 State Machine



State transitions on directory allocation

Simulation Environment

Simics

- *Simics is a full-system simulator used to run unchanged production binaries of the target hardware at high-performance speeds.*
- *By default, Simics does not model any cache system.*

GEMS and ruby model

- *They leverage Simics as the basis around which to build a set of timing simulator modules for modeling the timing of the memory system and microprocessors.*
- *Ruby is a timing simulator of a multiprocessor memory system that models: caches, cache controllers, system interconnect, memory controllers, and banks of main memory.*

Benchmark Suite

- **PARSEC** ---- The Princeton Application Repository for Shared-Memory Computers
- *It includes not only a number of important applications from the RMS suite but also several leading-edge applications from Princeton University, Stanford University and the open-source domain. The goal is to create a suite of emerging workloads that can drive CMP research.*

- Blackscholes has least instruction counts and minimum data requests.
- Swaptions has most instruction counts and medium working set.
- Canneal has medium instruction counts but unbounded working set.

Program	Application Domain	Parallelization		Working set	Data Usage	
		Model	Granularity		Sharing	Exchange
blackscholes	Financial Analysis	data-parallel	coarse	small	low	low
swaptions	Financial Analysis	data-parallel	coarse	medium	low	low
canneal	Engineering	unstructured	fine	unbounded	high	high

PARAMETER SETTINGS

	4-core [⊖]	16-core [⊖]	64-core [⊖]
Processor [⊖]		1 GHz, 2 IPC [⊖]	
L1 I & D cache [⊖]		32 KB, 4-way [⊖]	
Total shared L2 cache [⊖]	2 MB, 4-way, 4 banks [⊖]	4 MB, 4-way, 16 banks [⊖]	8MB, 4-way, 64 banks [⊖]
L1/L2/Dir block size [⊖]		64 Bytes [⊖]	
Directory associativity [⊖]		4-way [⊖]	
Memory size [⊖]		4 GB DRAM [⊖]	
On-chip link latency [⊖]		1 cycle [⊖]	
L1 hit/miss latency [⊖]		1/2 cycles [⊖]	
L2 hit/miss latency [⊖]		2/4 cycles [⊖]	
Directory latency [⊖]		4 cycles [⊖]	
Memory latency [⊖]		60 cycles [⊖]	
Topology [⊖]		2D Mesh [⊖]	

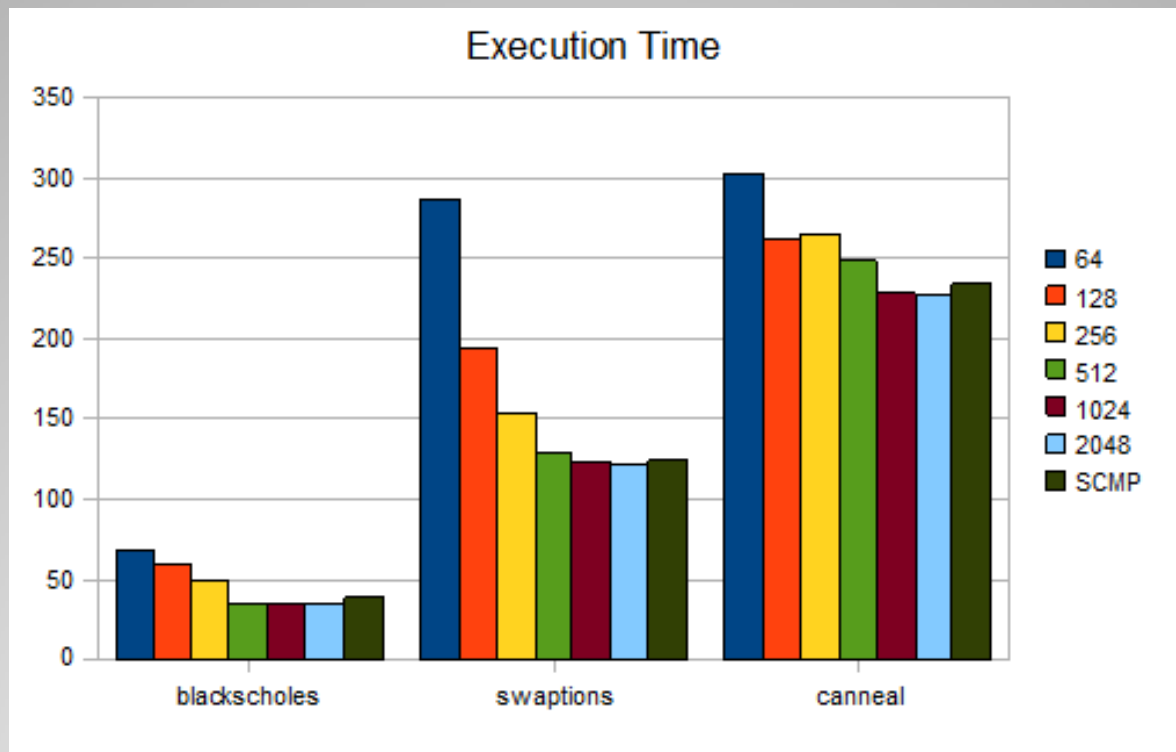
Simulation Metrics

- **Execution time:** The Ruby Cycle is our basic metric of simulated time used in the Ruby module of the GEMS simulator. The Ruby module is the basic module for the memory system configuration and interconnection network design.
- **L1 cache misses:** As the name says it represents the misses of L1 cache. It's calculated by dividing request missed by number of requests (Instruction + Data). It's an important metric for cache hierarchy.
- **L2 miss/miss rate:** This represents the total misses and miss rate of the L2 cache. It is calculated from the number of requests issued to the L2 and the misses of all banks of L2.
- **L2/Dir replacement:** Number of replacements of L2/Directory entries. It's caused by capacity misses and conflict misses.
- **Miss latency average:** Average of the L1 miss latency in Ruby cycles. It is measured from the moment a memory request is issued to the moment when the data is retrieved.
- **Memory requests:** Number of reads and writes issued to main memory

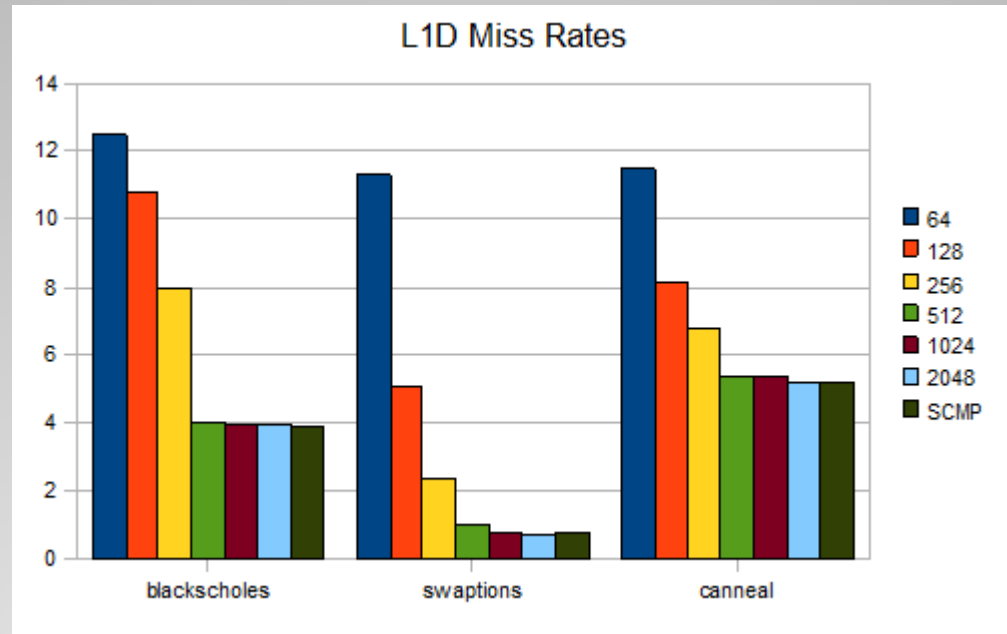
Results and Analysis

1. Influence of directory size

- First vary the directory size from 64 to 2048 entries on a four node Chip multiprocessor.
- SCMP protocol is simulated for comparison(last column).
- All simulations are configured with same cache size: 32KB L1I + 32KB L1D and 512KB L2 cache per core.

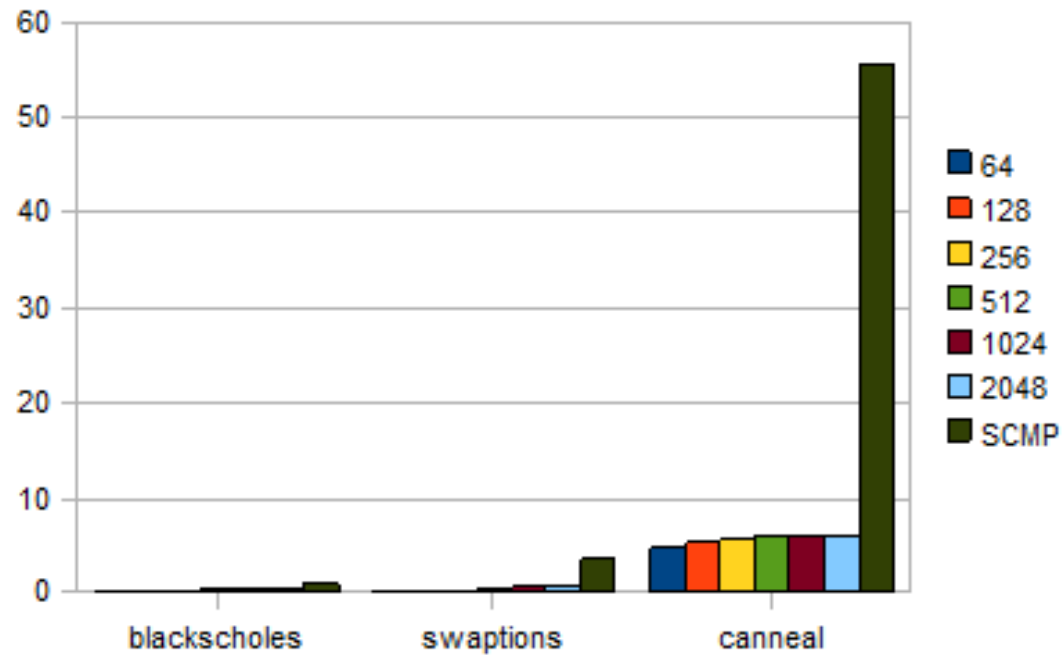


Execution time (million cycles)



L1 data cache miss rate (%)

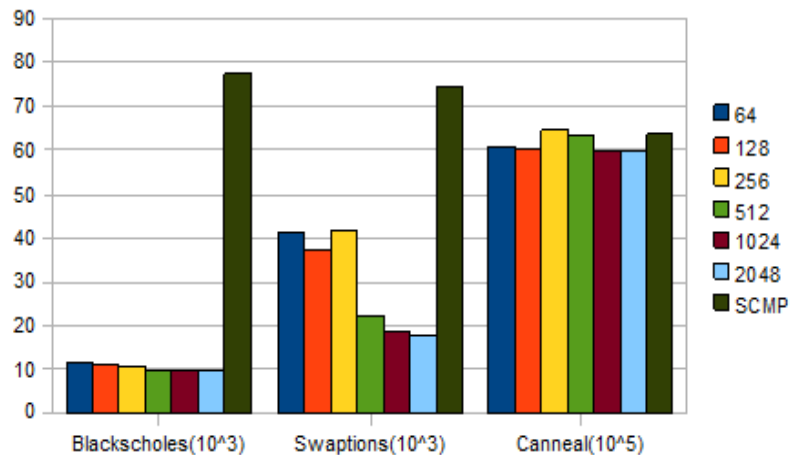
L2 Miss Rates



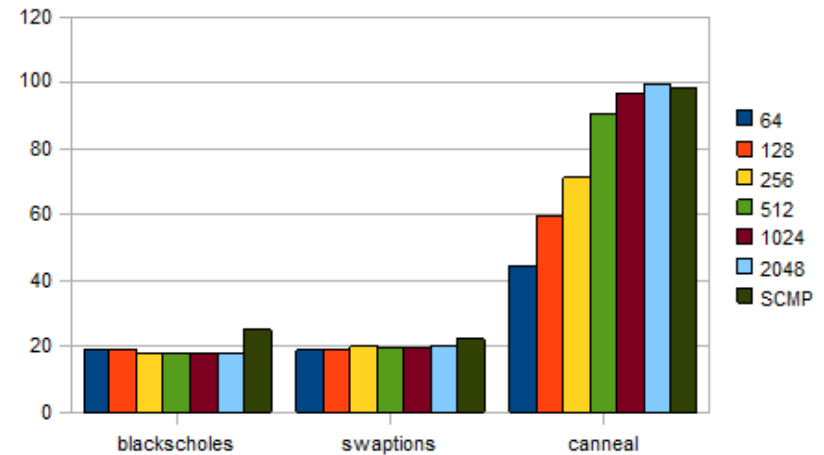
L2 miss rate (%)

Memory Requests and Miss Latency

Memory Requests

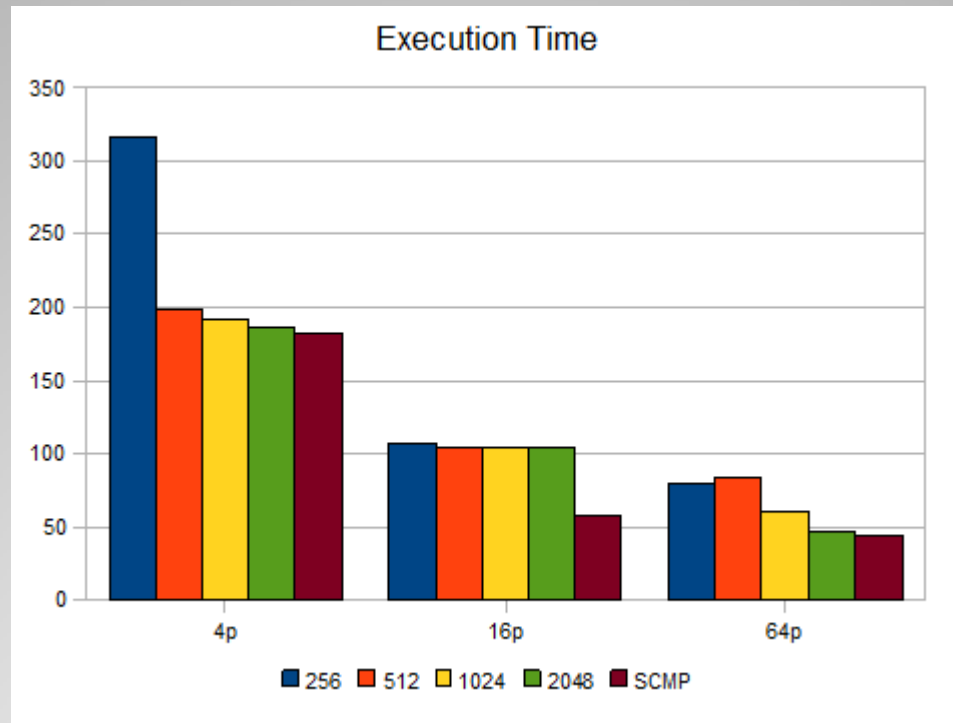


Average Miss Latency

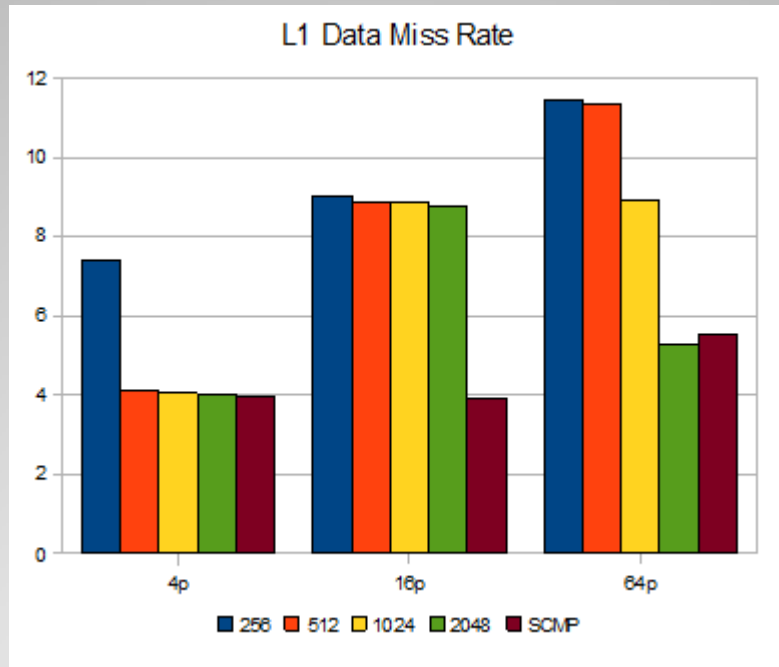


2. Scalability Research

- *A scalable directory-based protocol should work efficiently on large CMP with up to 64 cores while keep the overhead low.*
- *A PARSEC program “blackscholes” with medium working set is chosen for simulation.*
- *In order to compare with another protocol, we vary the directory size from 512 to 2048 and see whether it can achieve similar performance as the existing SCMP protocol.*

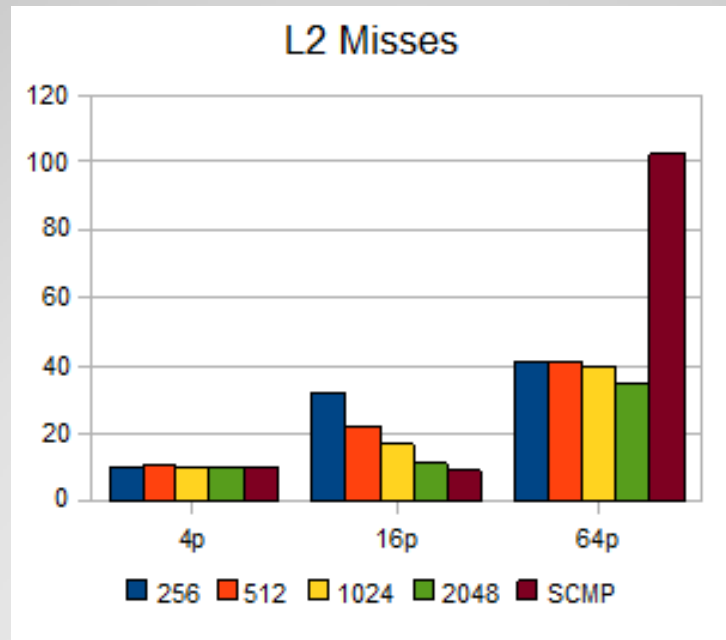


Execution Time

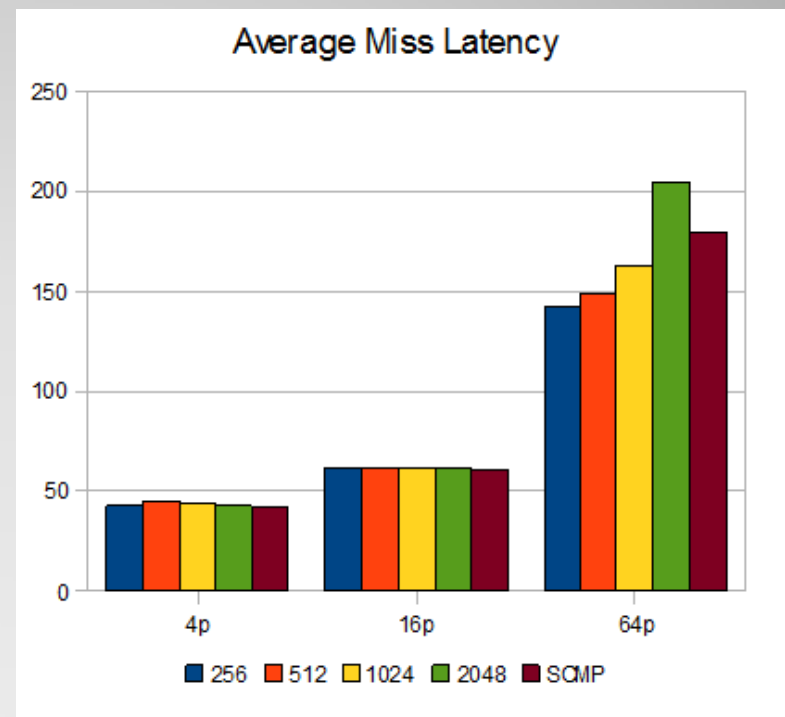
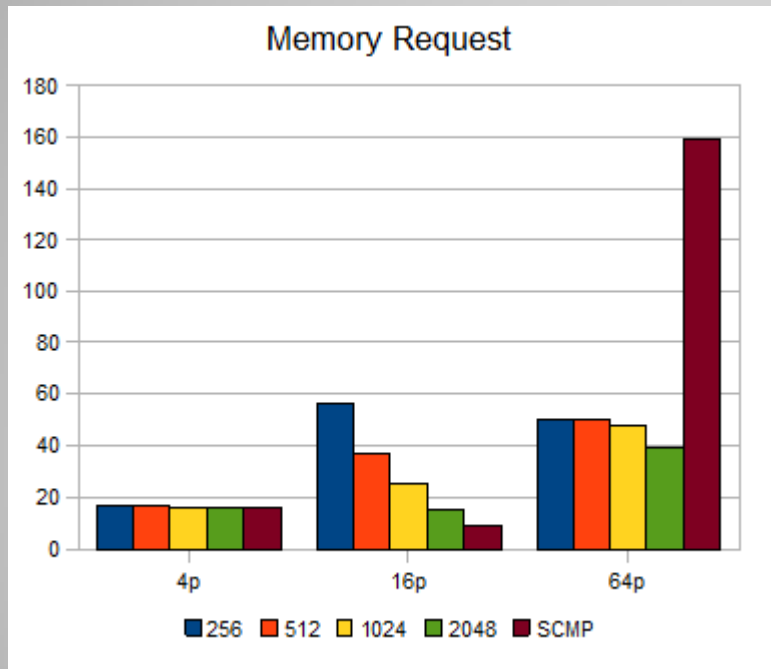


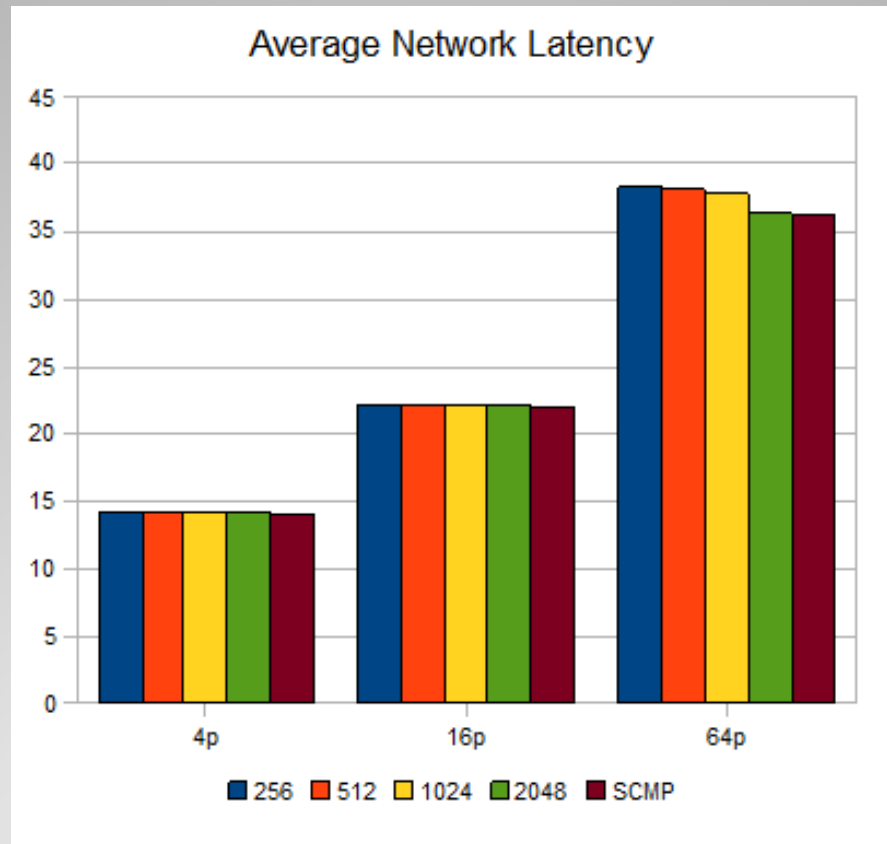
L1D Miss Rate

L2 misses(10^3)



Memory Requests & Average Miss Latency





Average Network Latency

CONCLUSION

- The protocol achieved good performance for applications with different characteristics.
- Compared to SCMP, our protocol gets similar L1 miss rates with only 512 directory entries.
- Less L2 miss rates and memory requests make our protocol attractive for memory intensive applications.
- Greatly reduced memory overhead.
- Visible scalability from 4 to 64 cores. The speedup is 1.8x from 4 to 16 cores and 2x from 16 to 64 cores

FUTURE WORK

- More tests

The simulation result would be more representative if more benchmark programs involved.

- Further reducing directory size

Directory size can be further reduced in width by utilizing compressed sharing code. However it implies modifications to GEMS itself since it's not supported natively.

- More parameter settings

We may also change parameters like network topology defined in ruby to see how the system performance is influenced.

Thank you

谢谢