Multi-core Processor Architectures

Polychronis Xekalakis and Josep M. Codina

Intel Barcelona Research Center

Aula Empresa, Facultat d'Informàtica de Barcelona, February 2010

Why Parallel Computing?

Simple task: Eat a pizza with 6 slices as fast as possible!!

Why Parallel Computing?

1x  **= 6 Time Units**

Why Parallel Computing?

1x  **= 6 Time Units**

6x  **= 1 Time Unit**

Why Parallel Computing?

1x  **= 6 Time Units**

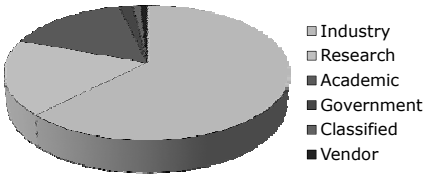
6x  **= 1 Time Unit**

Faster … but hungrier !!

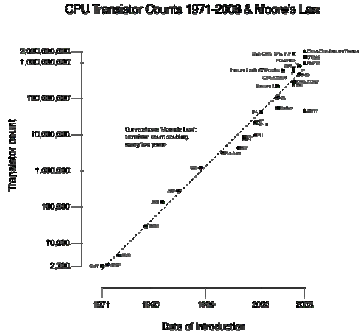
Parallel Computing Promises

- Solve larger problems (Human genome deciphering)
- Concurrency (big datacenters)
- More economic (Google – cluster computing)
- Power efficiency (Embedded computing)

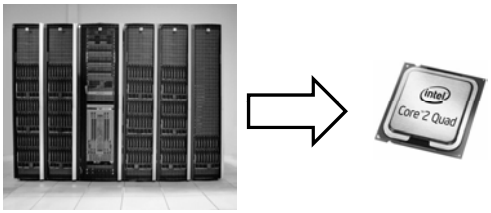
Who is Using Parallel Computing



Moore's Law for Multi-Cores



Shift from Parallel Computing to Multi-cores



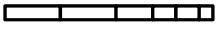
Why Chip Multiprocessors?

- For companies:
  - Time-to-market (Simply replicate cores)
  - Easier to extract perf. (Programmers have to do it ☺)
  - Power efficient (Low-cost thermal sinks)
- For individuals:
  - Solve larger problems (maybe ..)
  - Concurrency (Watch a movie, while burning a DVD)
  - Power efficiency (Embedded computing)

Agenda

- Motivation
- Parallel Architectures
- Keeping the Caches Consistent
- Interconnecting the cores
- Multithreading
- Concluding Remarks

Progress Bar



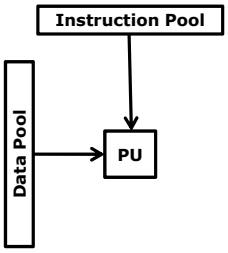
Parallel Architectures

Instruction Stream \ Data Stream		
	Single Inst. Single Data SISD	**Single Inst. Multi. Data SIMD**
	Multi. Inst. Single Data MISD	**Multi. Inst. Multi. Data MIMD**

Categorization by FLYNN

Single Instruction Single Data

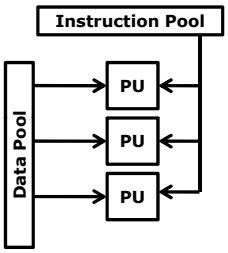
- Serial systems
- Deterministic execution
- First machines to be built
- Most common even today



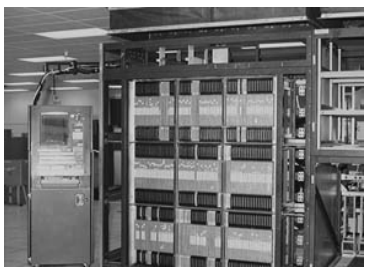
Univac I

Single Instruction Multiple Data

- Best suited for "regular" apps.
  - graphics/image processing
- Simple SIMD units in unip.
  - MMX/SSE/ .. / AVX

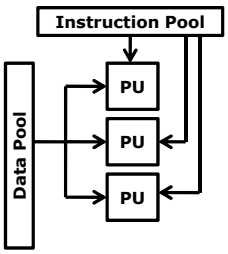


ILLIAC IV



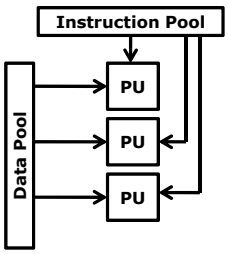
MISD

- Not many examples of this
- Hard to program ..



MIMD

- Most common supercomputer
- Everything is MIMD ..
- CMPs are MIMD too!!



IA32 Cluster

Agenda

- Motivation
- Parallel Architectures
- Keeping the Caches Consistent
- Interconnecting the cores
- Multithreading
- Concluding Remarks

Progress Bar

Caches

- Memory can be Fast or Vast !!
- Gap between Memory and Processor grows ..

Caches

- Memory can be Fast or Vast !!
- Gap between Memory and Processor grows ..

Caches

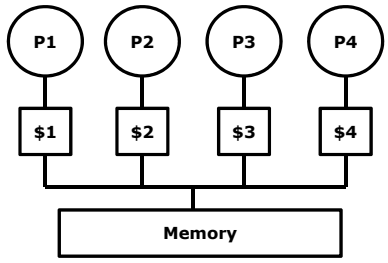
- Memory can be Fast or Vast !!
- Gap between Memory and Processor grows ..

- Caches come to the rescue ☺

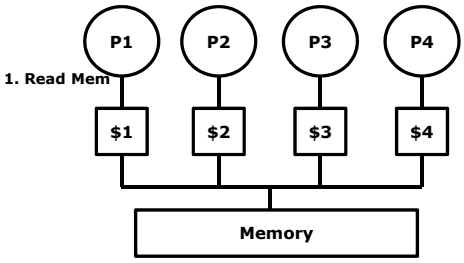
Caches for Uniprocessors and CMPs

- For single core systems caches are nice:
  - Reduce average access time
  - Bigger ➔ performance
  - Generally power efficient
- For CMPs still nice, but:
  - Cache Coherency
  - Memory Consistency

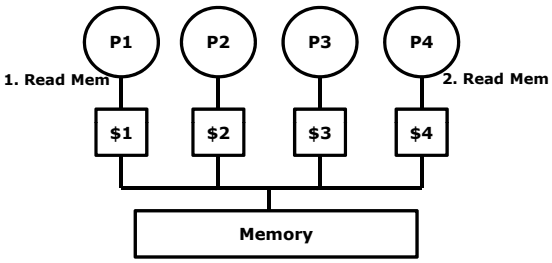
Cache Coherency



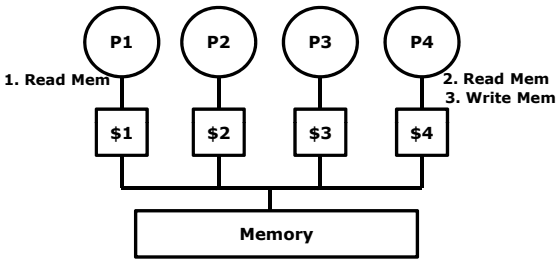
Cache Coherency



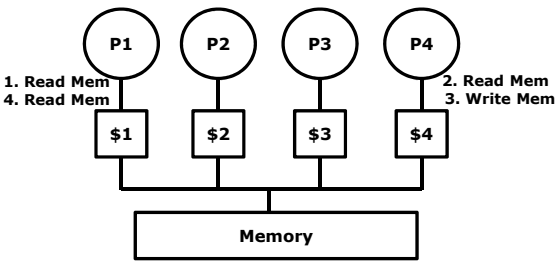
Cache Coherency



Cache Coherency



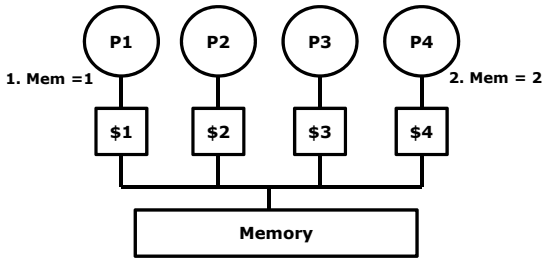
Cache Coherency



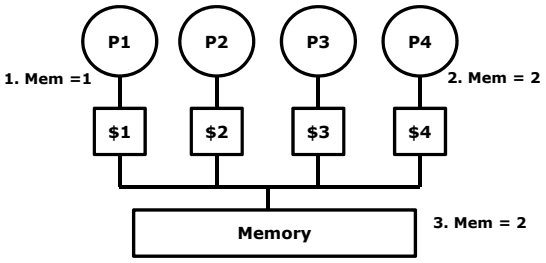
Memory Consistency

- Coherency guarantees writes visible to all
- But, in which order do they appear??

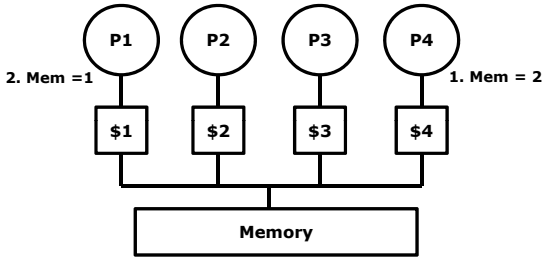
Memory Consistency



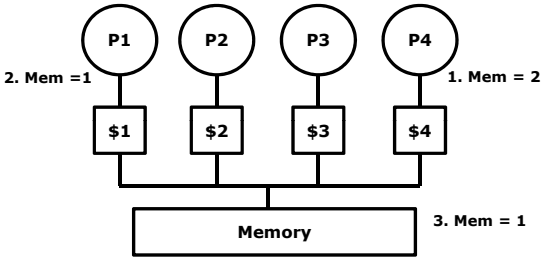
Memory Consistency



Memory Consistency



Memory Consistency



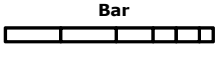
Memory Consistency Model

- Specifies constraints on the order in which memory operations (from any process) can appear to execute with respect to one another
- Implications for both programmer and system designer
  - Programmer uses to reason about correctness and possible results
  - System designer can use to constrain how much accesses can be reordered by compiler or hardware
- Contract between programmer and system

Agenda

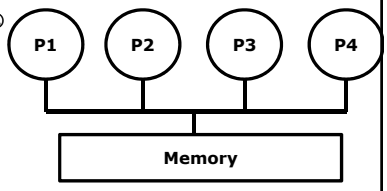
- Motivation
- Parallel Architectures
- Keeping the Caches Consistent
- Interconnecting the cores
- Multithreading
- Concluding Remarks

Progress Bar



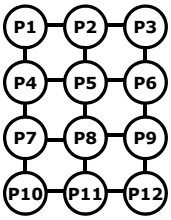
Simple Bus-Based Systems

- Fairly simple collection of "wires"
- Common medium that everyone can snoop
- Common for current CMPs
- Does not scale ☹



Mesh Networks

- Each core functions as a router
- More suitable for manycore systems
  - Single Chip Cloud Computing
- Longer latencies than bus



Networks on Chip

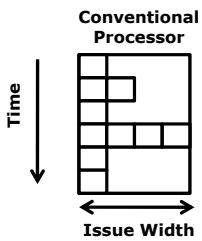
- Emerging design paradigm for interconnection within the chip
- More suitable for large-scale CMPs
  - Based on techniques used widely in networking
- NoCs provide:
  - QoS
  - Separability between computation and communication
  - Reusability

Agenda

- Motivation
- Parallel Architectures
- Keeping the Caches Consistent
- Interconnecting the cores
- Multithreading
- Concluding Remarks

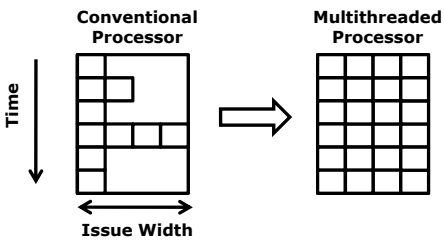
Multi-Threading Within a Core

- Cores usually underutilized …



Simultaneous Multi-Threading

- Fill it app with instructions from other threads



Speculative Multithreading

- What if threads come from same app??
- Can we have threads whose sole purpose is to help the main thread??
- Yes we can (Obama)!!
  - Helper threads (prefetch to memory)
  - Multi-path (follow all paths on a branch)
  - Thread Level Speculation (extract parallelism)
  - Etc ..

Agenda

- Motivation
- Parallel Architectures
- Keeping the Caches Consistent
- Interconnecting the cores
- Multithreading
- Concluding Remarks

Progress Bar

Summary - Conclusions

- Parallel computing is here to stay
- Today we discussed about:
  - Why we need parallelism
  - What are the types of architectures
  - How we keep the memory consistent
  - Interconnections
  - Multithreading
- Tip of the iceberg: Loads of exciting research happening!!

Multi-core Processor Architectures

Polychronis Xekalakis and Josep M. Codina

Intel Barcelona Research Center

Aula Empresa, Facultat d'Informàtica de Barcelona, February 2010