Multi processor architecture

A short introduction

Grigory Rechistov

Intel–MIPT lab, MDSP project

March 2011

Introduction

Moving from single CPU systems to multiprocessor ones requires much more effort than it would seem.

1. Topology questions
2. Resource sharing
3. Message passing
4. Platform startup

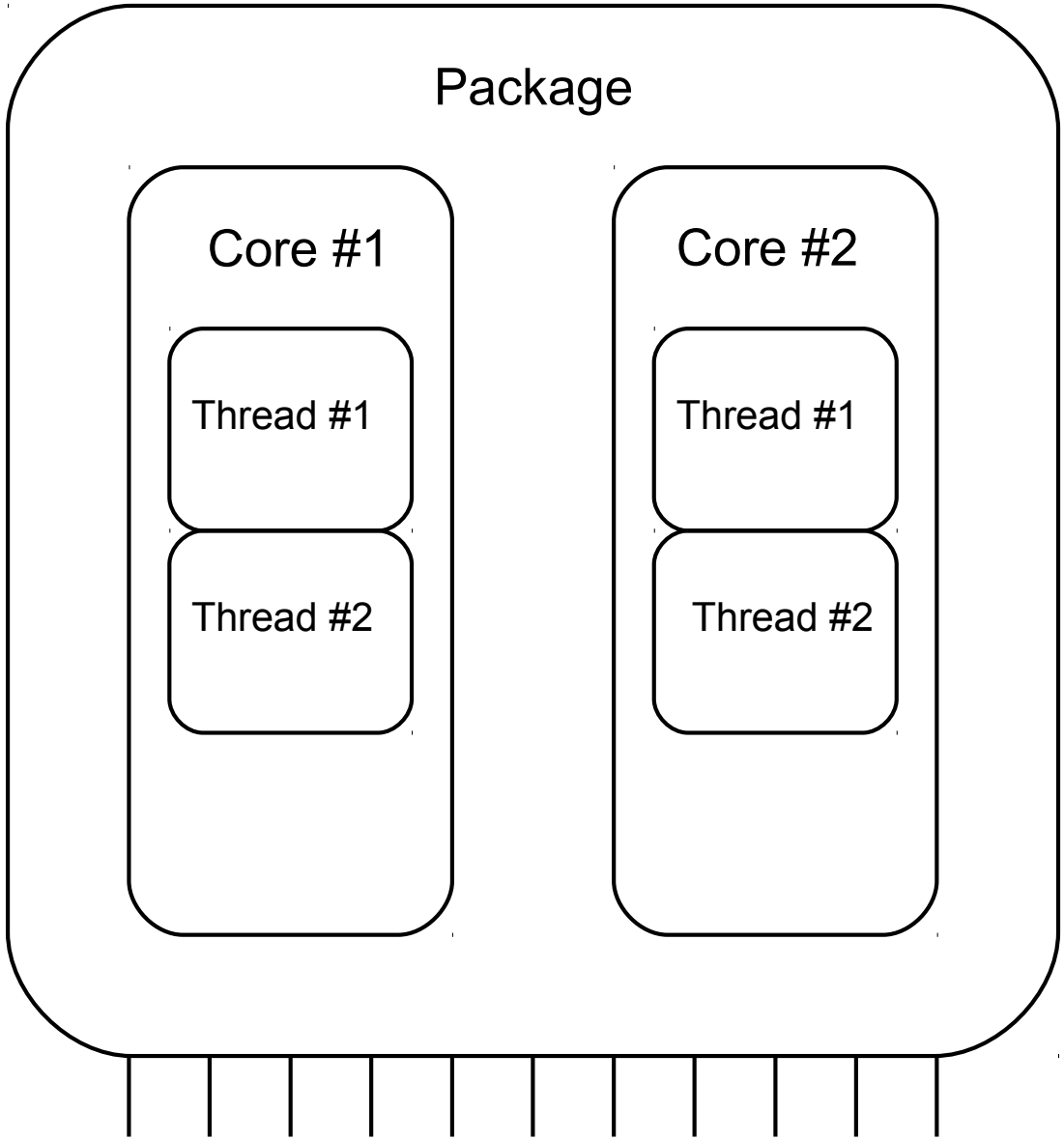
Introduction

Moving from single CPU systems to multiprocessor ones requires much more effort than it would seem.

1. Topology questions
2. Resource sharing
3. Message passing
4. Platform startup

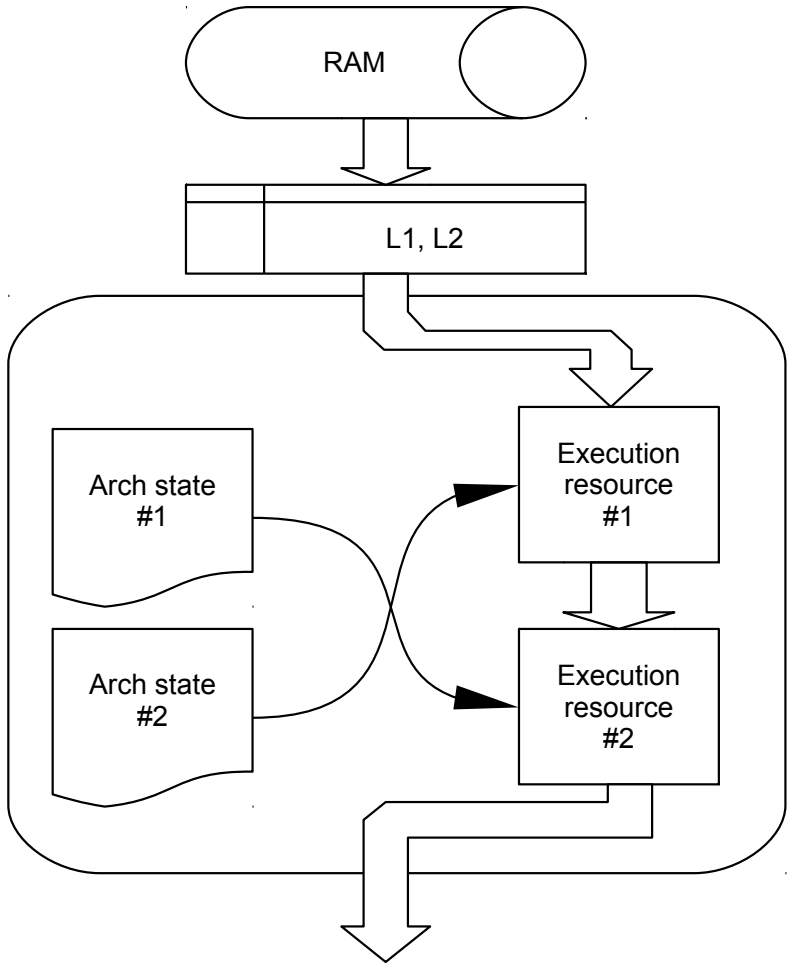
We don't consider the issues of programming MP systems.

Processor units hierarchy



What is hyperthreading

HT [1] works by duplicating certain sections of the processor – those that store the architectural state – but not duplicating the main execution resources.



HT: pros and cons

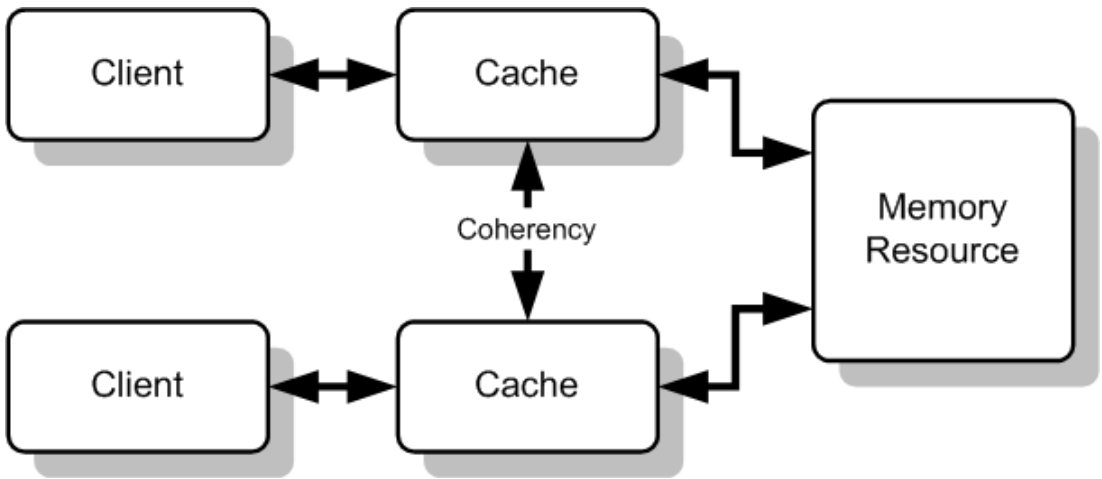
Pros:

- Utilizes resources that might otherwise stall.
- Up to 30% faster than same CPU without HT.
- Uses only 5% more of die area.

Cons:

- Is not *true* multicore.
- Requires OS awareness to correctly utilize HT.
- Poses security vulnerability [2].

Multicore systems and their caches

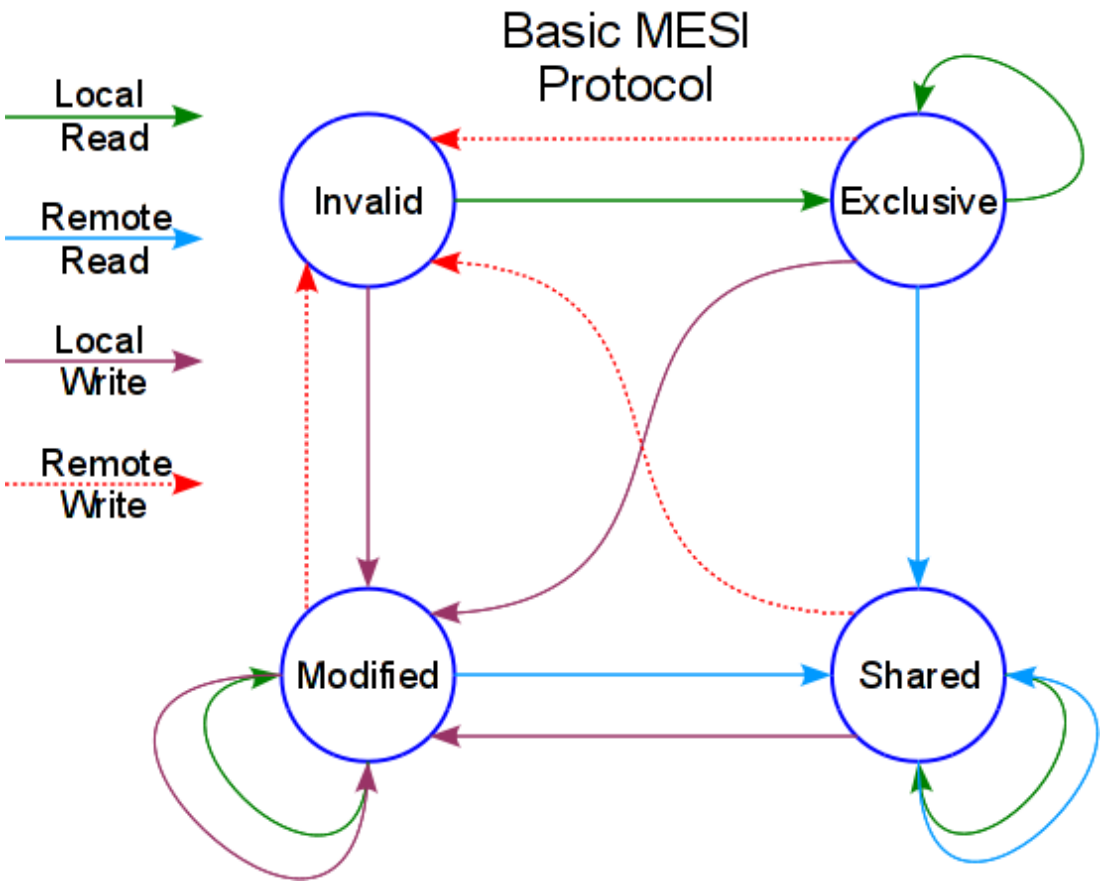


Type of cache sharing depends on the system: it can be L2 that is shared etc.

Types of coherence protocols

1. Directory-based: the data being shared is placed in a common directory that maintains the coherence between caches. The directory acts as a filter through which the processor must ask permission to load an entry from the primary memory to its cache. When an entry is changed the directory either updates or invalidates the other caches with that entry.
2. Snooping: individual caches monitor address lines for accesses to memory locations that they have cached. When a write operation is observed to a location that a cache has a copy of, the cache controller invalidates its own copy of the snooped memory location.

Caches coherency protocol example: MESI



Other protocols

- MOSI protocol
- MOESI protocol
- MESIF protocol
- MERSI protocol

SMP

Stands for **S**ymmetric **M**ulti **P**rocessing

- Identical processing units
- Single shared memory
- Single bus, mesh interconnections

Problem: scalability (systems with > 8 units will have performance problems)

NUMA

Stands for **N**on **U**niform **M**emory **A**ccess
Memory access time depends on the memory location relative to a processor. Under NUMA, a processor can access its own local memory faster than non-local memory,

- Highly scalable
- Requires special coherency protocols
- Requires OS support

AMD: Opteron (HyperTransport)
Intel: Intel Core i7-9xx (QPI)

APIC, IO-APIC

Stands for **A**dvanced **P**rogrammable **I**nterrupt **C**ontroller

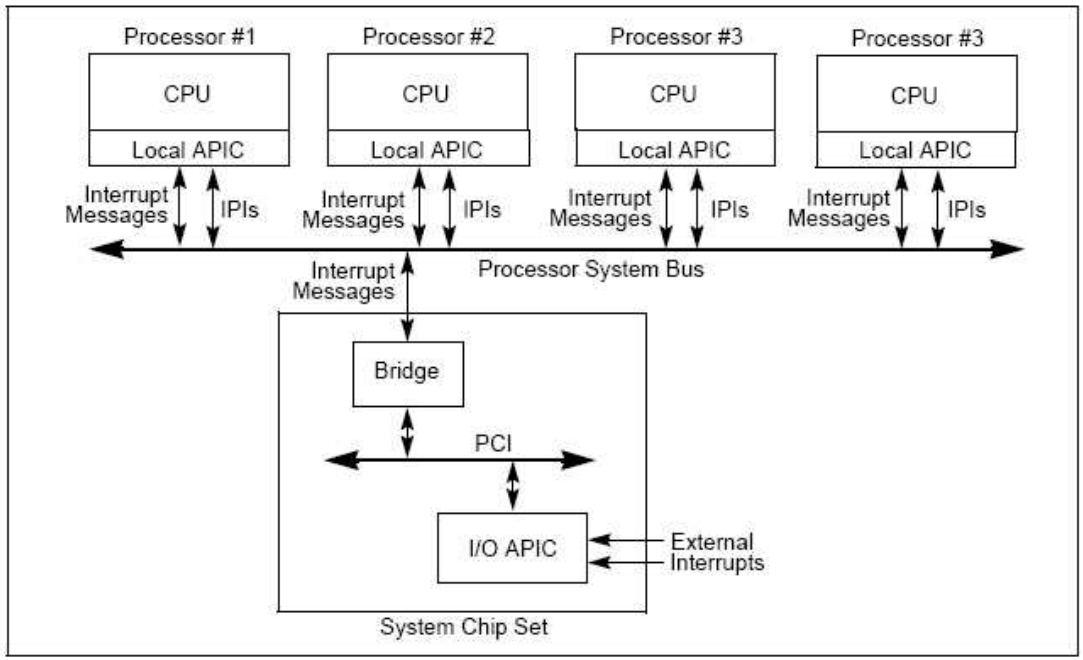


Figure 8-2. Local APICs and I/O APIC When Intel Xeon Processors Are Used in Multiple-Processor Systems

Programmable interrupt controller (PIC) is a device that is used to combine several sources of interrupt onto one or more CPU lines, while allowing priority levels to be assigned to its interrupt outputs.

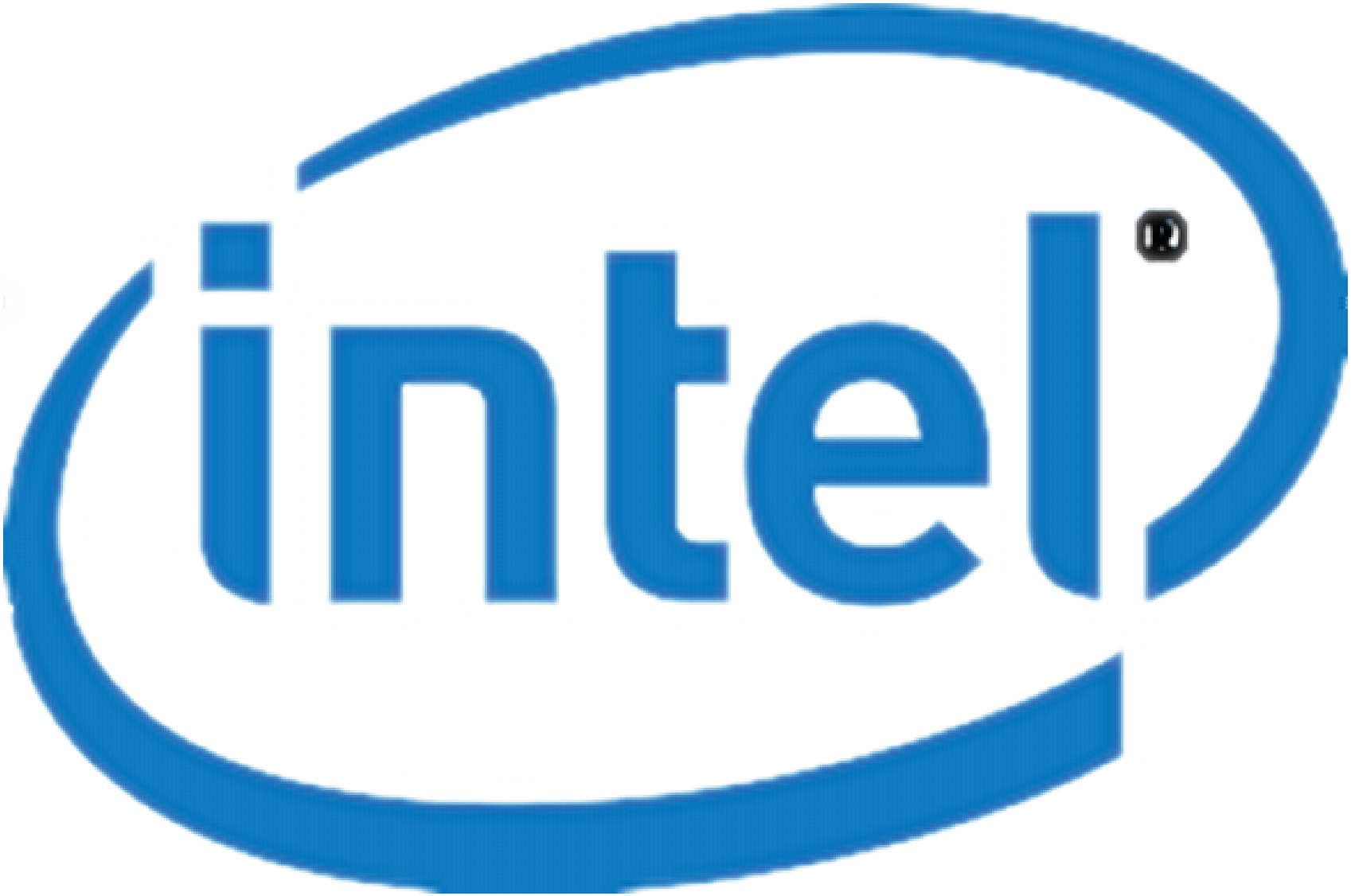
Boot process

1. The BSP[1] executes the BIOS's boot-strap code to configure the APIC environment, sets up system-wide data structures, starts and initializes the AP[2]s. When the BSP and APs are initialized, the BSP then begins executing the operating-system initialization code.
2. Following a power-up or reset, the APs complete a minimal self-configuration, then wait for a startup signal (a SIPI message) from the BSP processor. Upon receiving a SIPI message, an AP executes the BIOS AP configuration code, which ends with the AP being placed in halt state.
   [3]

[1] Boot strap processor.
[2] Application processor

More to read

- ftp://download.intel.com/technology/itj/2002/volume06issue01/vol6iss1_hyper_threading_technology.pdf
- http://www.daemonology.net/papers/htt.pdf
- http://download.intel.com/design/archives/processors/pro/docs/24201606.pdf


intel