Chapter-10

Multicore Computers

A **multi-core processor** is a single computing component with two or more independent actual central processing units (called "cores"), which are the units that read and execute program instructions. The instructions are ordinary CPU instructions such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs enabling parallel computing. Manufacturers typically integrate the cores onto a single integrated circuit die (known as a chip multiprocessor or CMP), or onto multiple dies in a single chip package.

Processors were originally developed with only one core. Multi-core processors were developed in the early 2000s by Intel, AMD and others. Multicore processors may have two cores (dual-core CPUs, for example AMD Phenom II X2 and Intel Core Duo), four cores (quad-core CPUs, for example AMD Phenom II X4, Intel's i5 and i7 processors), six cores (hexa-core CPUs, for example AMD Phenom II X6 and Intel Core i7 Extreme Edition 980X), eight cores (octo-core CPUs, for example Intel Xeon E7-2820 and AMD FX-8350), ten cores (for example, Intel Xeon E7-2850), or more.

A multi-core processor implements multiprocessing in a single physical package. Designers may couple cores in a multi-core device tightly or loosely. For example, cores may or may not share caches, and they may implement message passing or shared memory inter-core communication methods. Common network topologies to interconnect cores include bus, ring, two-dimensional mesh, and crossbar.Homogeneous multi-core systems include only identical cores, heterogeneous multi-core systems have cores that are not identical.

Multi-core processors are widely used across many application domains including general-purpose embedded, network, digital signal processing(DSP), and graphics.

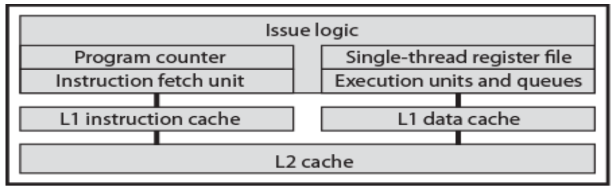
Hardware Performance Issues

Increase in Parallelism

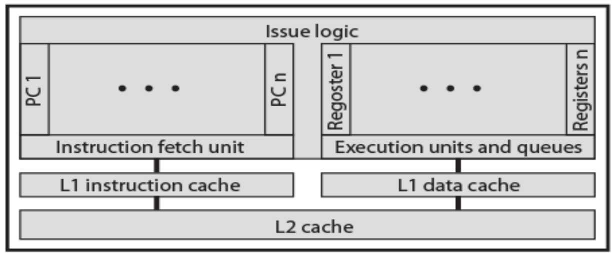
For enhancement of multicore processors design has been focused on increasing instruction level parallelism, so that more work could be done in clock cycle. These changes include:

- Pipelining: Instructions are executed through pipeline of stages so that while one instruction is executing in one stage of pipeline, another instruction is executing in another stage of pipeline. Simple three-stage pipelines were replaced by pipelines with five stages, and many more. There is a limit in increasing pipelining stages as increased stage requires more logic, interconnections and control signal.
- Superscalar: Multiple pipelines are constructed replicating the resource. This enables parallelism in parallel pipeline. Performance can be increased by increasing number of parallel pipelines.
- Simultaneous multithreading(SMT): Registers are replicated so that multiple threads can share the use of pipeline resources. Managing multiple threads over a set of pipelines limits the number of threads and number of pipelines that can be effectively utilized.

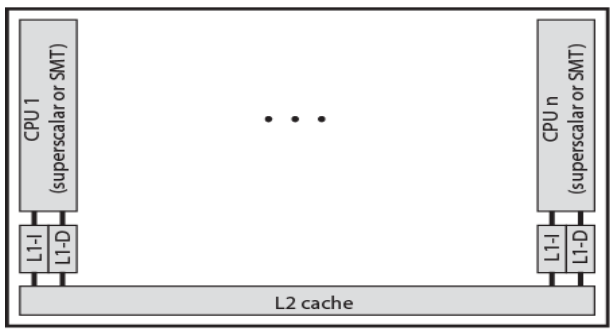
Alternative chip Organization



(a) Superscalar



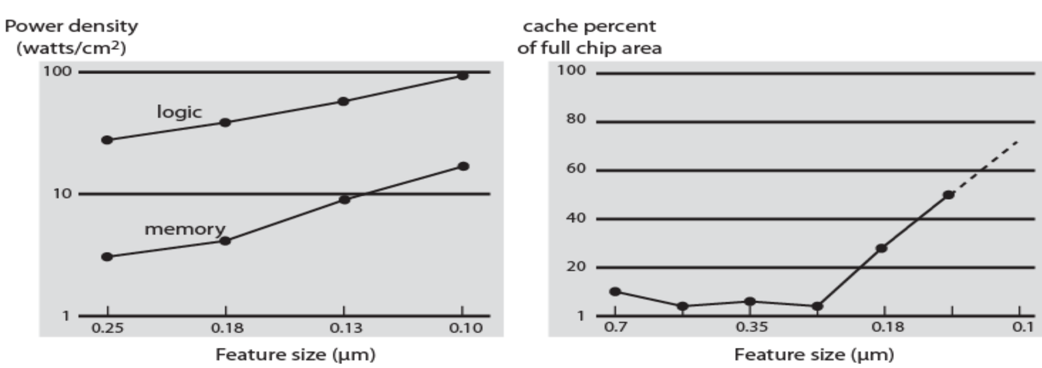
(b) Simultaneous multithreading



(c) Multicore

Power Consumption

As the number of transistors per chip and clock frequencies increased power requirements grown exponentially. Power density can be minimized by using multiple chips on cache memory as memory transistors are smaller and have a power density lower than that of logic transistors.



(a) Power density

(b) Chip area

Fig: Power and Memory Considerations

Power consumption trend is increasing in every year. By 2015 we can expect processors chips with 100 billion transistors on a 30mm^2 die. Assuming 50-60% of the chip devotes to memory supporting 100 MB of cache and leave about over 1 billion transistors available for logic.Pollack's rule states that performance is roughly proportional to square root of increase in complexity i.e. if we double the logic in a processor core, then it delivers only 40% more performance.Multicore has potential for near-linear improvement with the increase in number of cores. Power consideration provides another motivation for multicore organization as the chip has such a huge amount of cache memory.

Software performance issues

Performance benefits of multicore organization depend on effective exploitation of parallel resources. Software suffers overhead as a result of communication, distribution of work to multiple processors and cache coherence overhead. This results in high performance at the beginning and goes on decreasing as burden of overhead of using multiple processors increases.

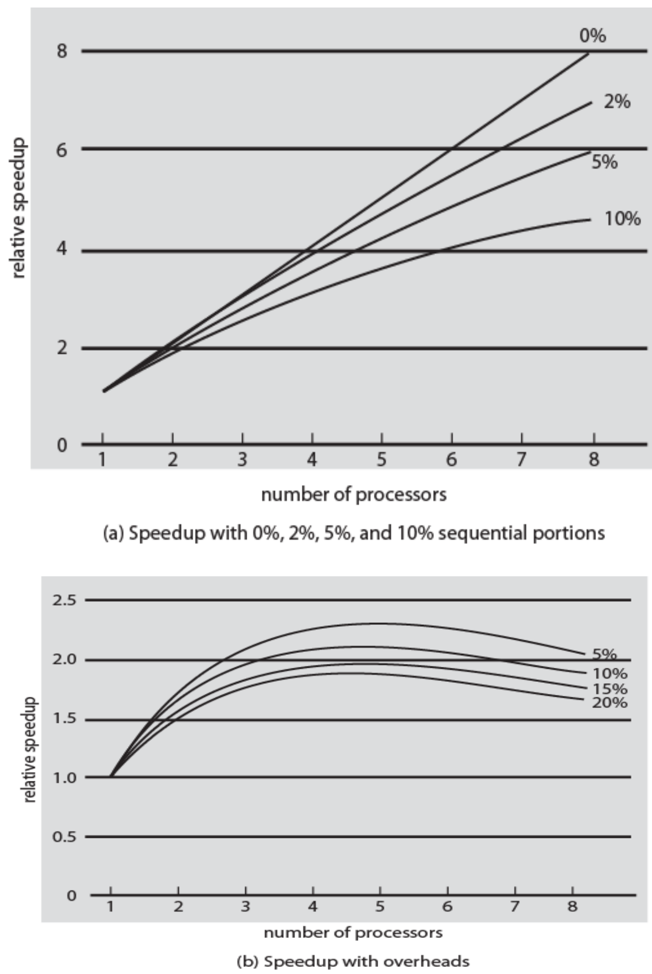


Fig: Performance Effect of Multiple Cores

In addition to general-purpose server software a number of classes of applications benefits directly. Some examples include:

- Multithreaded native applications: These applications are characterized by having a small number of highly threaded processes.

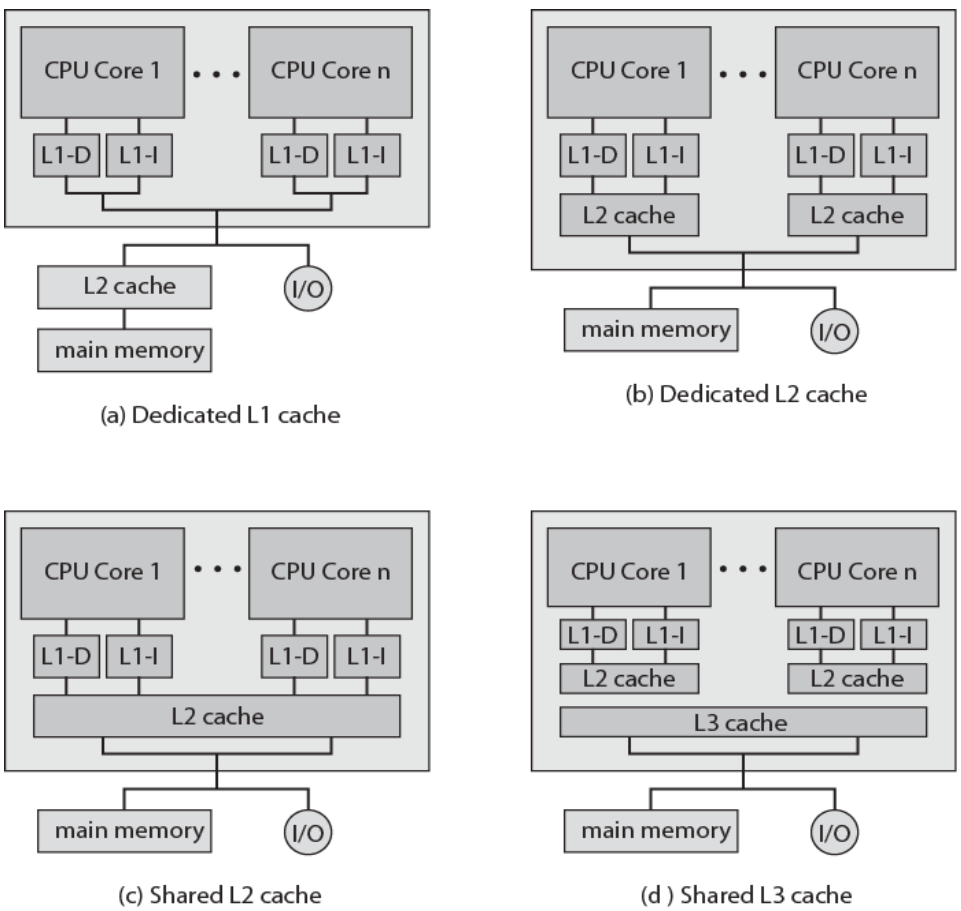
- Multi-process applications: It is characterized by the presence of many single threaded processes.
- Java applications: Java language and Java virtual machine is a multithreaded process that provides scheduling and memory management. Java applications that directly benefits from multicore resources are Sun’s Java Application Server, BEA’s Weblogic, IBM Websphere and open-source Tomcat.
- Multi-instance applications: Single application can run in multiple instances parallel.

Multicore Organization

The main variables in a multicore organization are as follows:

- The number of core processors on the chip.
- The number of levels of cache memory.
- The amount of cache memory that is shared.

Figure below shows four general organizations for multicore systems.



(a) Dedicated L1 cache

(b) Dedicated L2 cache

(c) Shared L2 cache

(d) Shared L3 cache

Figure (a) is an organization found in some of the earlier multicore computer chips. In this organization, the only on-chip cache is L1 cache, with each core having its own dedicated L1 cache. E.g is ARM11MPCore. Figure (b) is organization where there is no on-chip cache sharing. In this, there is enough area available on the chip to allow for L2 cache. E.g. AMD Operton. Figure (c) is similar to (b) but with the use of shared L2 cache. E.g. Intel Core Duo. As the amount of cache memory increases, performance considerations requires a separate share L3 cache with dedicated L1 and L2 cache for each processor. E.g. Intel Core i7.

Advantages of using shared Cache

1. It can overcome the miss rates. That is if a thread on one core references memory, the frame is brought into shared cache memory. Now, if a thread from another core references the same frame than it is already present in cache memory.
2. Data shared by multiple cores is not replicated at same cache level.
3. With proper frame replacement algorithm, the amount of shared cache allocated to each core is dynamic.
4. Inter-processors communication is easy to implement through shared memory locations.
5. The use of shared L2 cache confines the cache coherency to L1 cache.

Dual Core and Quad Core Processors

The dual-core CPU is a processor with two execution cores in a single integrated circuit. These cores act as a single unit, but they will each have their own controller and cache, allowing them to perform faster than single-core processors.

Quad-core CPUs contain four processors. Depending on the manufacturer, this can mean that four cores are on the same integrated circuit or the same chip package. Since these are a little more complicated that dual-core CPUs, there are some variations on this design. For example, some chips may or may not share resources like caches. It's also possible to have a quad-core processor with the same types of chips (homogeneous multi-core systems) or different chips (heterogeneous multi-core systems).

A quad-core processor is faster than a single- or dual-core processor only when it's running an application that's been developed to take advantage of its abilities. Most applications today are still written for single- or dual-core processor and this simply means the quad-core's extra power goes unused.

A smartphone with a dual-core processor may perform beautifully, as it enables you to quickly switch tasks. One core may be at work on your web browsing while the other is basically just standing by. A call comes in, and the second core gets to work. Both your web browsing and your phone call continue without a hitch.

A quad-core processor allows your smartphone to run the latest, greatest, most complex apps, such as an intricate, graphics-heavy videogame. It can also mean business-oriented functions that require speed perform more smoothly. Video calls have better image clarity and solid audio, and the processor also has the ability to handle other functions in the background.

Advantages of dual core processors

1. Performs tasks faster: Dual-core CPUs run faster than single-core ones, especially in instances where there are multiple processes at one time. When a single-core processor has to do two different things, it must stop what it's doing and then switch to the next task. This switching is what creates lags, and in dual-core processors, this switching between tasks is reduced because there are two processors that can do the task at once.

2. Reduced costs: Even before there were dual-core CPUs, users were able to build dual-processor units that had the computing power of two computers. However, this also meant it was like purchasing another computer, because that means buying a separate processor and motherboard sockets. A dual-core processor can fit into a single motherboard and computer.

Disadvantages of dual core processors

1. Wasted computer power: Checking email, surfing through text-based sites, and typing documents don't actually use up a lot of power but in graphics or video programs, the dual-core almost seems excessive. Thus waste of computer power.
2. Compatibility with software: Sure, dual-core CPUs will run any software, but the software itself has to be programmed for the dual-core CPU. Programmers have to tell the software that when one CPU is overloading, it needs to switch some of the tasks over to the second CPU.

Advantages of quad core processors

1. Multitasking: The quad-core system is one of the best systems for multi-tasking. With so many cores, it can do many processes at once, while still maintaining the integrity of the system.
2. Run intensive applications: Applications that use up a lot of resources, such as graphics programs, video editors, and anti-virus programs, can run smoothly at the same time.
3. Less heat and power consumption: Most of the newer quad-core chips are so small and efficient that they can actually use less power and generate less heat than single-core systems.
4. Use for long term: The problem with Moore's Law is that it practically guaranteed that your computer would be obsolete in about 24 months. Since few software programs are programmed to run on dual-core, much less quad-core, these processors are actually way ahead of software development.

Disadvantages

1. Lowers battery life: Depending on the type of applications, quad-core systems can drain batteries faster.
2. Available software: Software needs to be programmed to take full advantage of quad-core CPUs, so not all programs can utilize the four processors.
3. Hardware compatibility: Multi-core processors are compatible with certain motherboards, so it's not as simple as swapping out the old CPU with a brand new one. Purchase of a new motherboard may also necessitate the purchase of other components that are compatible with the motherboard.