Parallel Programming: Moore’s Law and Multicore

Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

Increasing Transistor Density -- Moore’s Law

"Transistor density doubles every 1.5 years." Note: Log scale!


Offentimes people have (incorrectly) equivalenced this to: "Clock speed doubles every 1.5 years."

Increasing Clock Speed?

Note: Log scale!

Clock speed has hit a plateau, largely because of power consumption and dissipation.

\[
\text{PowerConsumption} \propto \text{ClockSpeed}^2
\]

Once consumed, that power becomes heat, which much be dissipated somehow. In general, compute systems can remove around 150 W/m^2 without resorting to exotic cooling methods.
And, speaking of “exotic”, recently, AMD set the world record for clock speed (8.429 GHz) using a Liquid Nitrogen-cooled CPU.

What Kind of Power Density Dissipation Would it Have Taken to Keep up with Clock Speed Trends?

MultiCore -- Multiprocessing on a Single Chip

So, to summarize: Moore’s Law of transistor density is still going, but the “Moore’s Law” of clock speed has hit a wall. Now what do we do?

We keep packing more and more transistors on a single chip, but don’t increase the clock speed. Instead, we increase computational throughput by using those transistors to pack multiple processors onto the same chip.

This is referred to as multicore.

Vendors have also reacted by adding SIMD floating-point units on the chip as well. We will get to that later.

MultiCore and Multithreading

Multicore, even without multithreading too, is still a good thing. It can be used, for example, to allow multiple programs on a desktop system to always be executing concurrently.

Multithreading, even without multicore too, is still a good thing. Threads can make it easier to logically have many things going on in your program at a time, and can absorb the dead-time of other threads.

But, the big gain in performance is to use both to speed up a single program. For this, we need a combination of both multicore and multithreading.

Multicore and Multithreading:

Multithreading

Multicore

Multicore is a very hot topic these days. It would be hard to buy a CPU that doesn’t have more than one core. We, as programmers, get to take advantage of that.

We need to be prepared to convert our programs to run on MultiThreaded Shared Memory Multicore architectures.

Intel’s Approach to Multicore and Multithreading

Note: these are not the same!!

“HT” stands for HyperThreading.

Four Cores with Two Hyperthreads per Core

Note that this is upside-down from our usual convention. Sorry. I got this from someone else.