High-Performance Computing
State and Trends

James Davenport J.H.Davenport@bath.ac.uk

16 January 2012

Background: Amdahl's Law

If a fraction P of a program can be improved (parallelised), and the speed-up on this is S, then the achieved speed-up overall is

$$\frac{1}{(1-P)+\frac{P}{S}}.$$

In particular, if $S=\infty$, we have $1/(1-P)$.
So what part of your program can't/don't you parallelise?
Often the I/O!
Tends not to be a major problem at Bath, with our applications and core count, except for `Gaussian`!

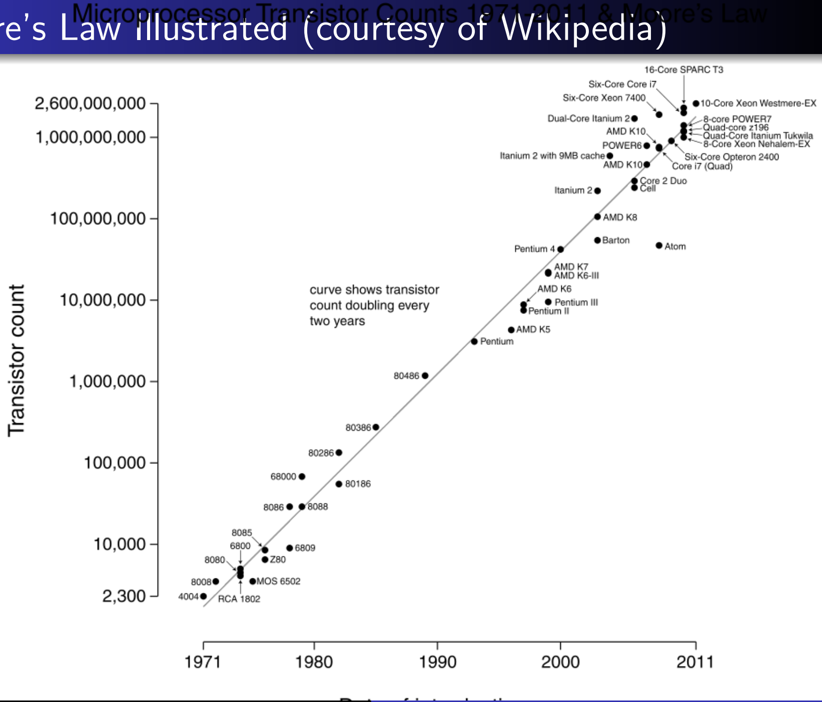
Background: Moore's Law

Moore was a co-founder of Intel, and predicted in 1965: "for the next ten years"

> *the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.*

Uncannily accurate

Moore's Law illustrated (courtesy of Wikipedia)



Moore's Law (Common Version)

> *the speed of computers doubles approximately every* $1\frac{1}{2}$ *years* $=$ *18 months.*

Actually due to an Intel marketing executive, who converted number of transistors into speed (true at the time, and essentially true today if we measure in terms of total throughput of a chip), and converted 2 years into $1\frac{1}{2}$ "because circuits are also getting faster" (no longer true)

What's a GigaFLOP?

- **Clearly** 10^9 floating point operations per second
- `c:=c+a*b` is *clearly* two floating point operations (multiply and add)

But Most processors can execute this as one instruction — Fused Multiply–Add

And can execute two (or four in single-precision) simultaneously

Hence a 3GHz core is a 12Gflop peak machine

If all it does is pairs of FMAs (as matrix multiply might)

- Random floating-point would be lucky to see 25% of this.

Also there are 2 or 3 levels of cache between the chip and main memory

and a latency factor of 30 or more,

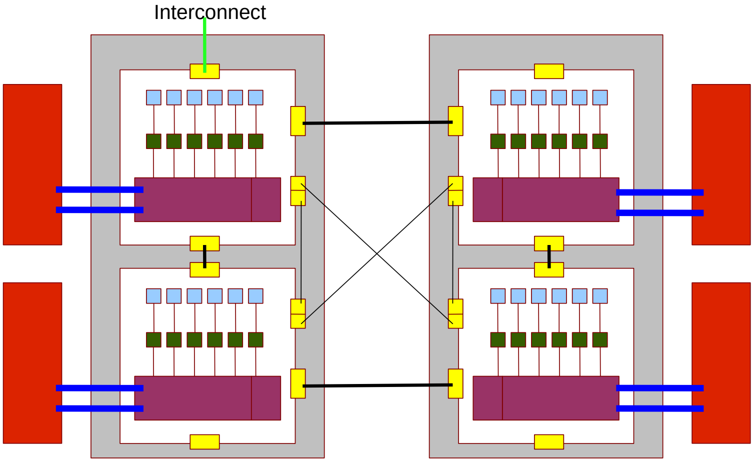
So random floating-point from random locations will see $< 1\%$

And that’s only a single core

- Intel Core Duo was basically just two complete processors on one chip
- (but sharing L2 cache, and bus)
- The Core 2 Quad was two complete Core 2 Duos, on *two* dies (chips), but in one package
- Whereas `aquila` has four cores on a single chip (and two chips on a node)
- HECToR had ‘Magny Cours’ 12-core Opteron packages, but that was in fact 2 6-core dies, each accessing memory separately
- And with two of them in a node, we actually had 4 memory banks per node, with complicated interconnects
- HECToR currently is 2 packages/node, with 2 dies/package, 4 modules/die and 2 cores (but only one FPU)/ module.

It will get worse

HECToR 2b (courtesy of NAG)



🟦 Single core (incl. 64KB L1)

🟩 L2 Cache 512KB

🟪 L3 Cache 6MB (~1MB used by HT Assist)

□ Hex-core die

▨ G34 socket – Magny-Cours Opteron

🟨 Hyper Transport port

— 16-bit Hyper Transport 1 link, 6.4GB/s

— 16-bit Hyper Transport 3.1 link, 25.6GB/s

— 8-bit Hyper Transport 3.1 link, 12.8GB/s

🟥 8GB DDR3 Memory @ 1333 MHz, 85.3GB/s across node

— Memory channel

There's also GPGPUs

- General Purpose Graphical Processing Units —- outgrowths of the old graphics cards
- Single Instruction Multiple Data processors, capable of awesome raw FLOP rates
- 400 cores/GPU at 2.5GFLOP/core = 1 TFLOP/card, or 16TFLOP/box!
- A box is therefore *apparently* faster than `aquila`, at 10% of the cost
- Until recently, only single-precision, but that's changing.
- *Really good* at homogeneous, small-data, Monte Carlo computations, where 400 cores ≡ 400 scenarios in parallel
- *Slower than the CPU driving it* when it comes to unstructured data communication

For most applications, still pie in the sky"

And now for a league table! (Nov. 2011)

#	Country	Total C	GPU C	Rmax	Rpeak	Eff(%)	Mflops/Watt
1	Japan	705024	0	10510000	11280384	93.17	830.18
2	China	186368	100352	2566000	4701000	54.58	635.15
3	US	224162	0	1759000	2331000	75.46	253.09
4	China	120640	64960	1271000	2984300	42.59	492.64
5	Japan	73278	56994	1192000	2287630	52.11	852.27
6	US	142272	0	1110000	1365811	81.27	278.89
7	US	111104	0	1088000	1315328	82.72	265.24
8	US	153408	0	1054000	1288627	81.79	362.2
9	France	138368	0	1050000	1254550	83.70	228.76
10	US	122400	110160	1042000	1375776	75.74	444.35
11	US	112800	0	919100	1173000	78.35	297.44
12	Germany	113472	0	831400	1043942	79.64	
13	Germany	294912	0	825500	1002701	82.33	363.98
14	China	137200	0	795900	1070160	74.37	741.06
15	US	46208	0	773700	961126	80.50	837.19
16	China	53248	28672	771700	1342750	57.47	668.1
17	US	65536	0	677104	838861	80.72	1988.56
18	Russia	33072	21756	674105	1373060	49.10	240.75
19	UK	90112	0	660243	829030	79.64	

Some observations on the league table

- Efficiency is Rmax/Rpeak (i.e. what it could do if every cycle was doing the maximum number of floating-point operations)
- #1 (Japan) is SPARC, 10, 13, 17 are PowerPC variants, the rest are x86 architectures
- `aquila` is 800 cores, with an Rpeak of 11000, and an efficiency of 85%. It does about 237 MFLOP/Watt
- With the exception of machine 10 (PowerXCel: doing something rightI), all machines with GPU cores have significantly lower efficiencies, and Rmax is the throughput on linear algebra, which is about the best case requiring any communication!