GPU Programming Why parallelism?



Why parallelism?





More efficient programs!



[1] J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach, sixth edition. Morgan Kaufmann, 2017.

| AMD Athlon, 2.6 GH Intel Xeon EE 3.2 GHz |
|------------------------------------------------------------------------------------------------|
| cessor with Hyper-Threading Technology) 6,0 |
| IBM Power4, 1.3 GHz 4,195 |
| rd, 1.0 GHz Pentium III processor n XP1000, 667 MHz 21264A 6/575, 575 MHz 21264 1,267 |
| 0, 600 MHz 21164 649 600, 500 MHz 481 |
| 800 MHz 280 |
| Hz 183 |
| • 117 |
| |
| 52%/year |
| |
| |
| |
| |
| |
| |
| 1994 1996 1998 2000 2002 : |

© Christian Lessig, 2018



Year

integrated circuits, "Cramming more components onto , vol. 38, no. 8, p. 114 ff, 1965. G. Moore, " Electronics, '



[1] J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach, sixth edition. Morgan Kaufmann, 2017.

| AMD Athlon, 2.6 GH Intel Xeon EE 3.2 GHz |
|------------------------------------------------------------------------------------------------|
| cessor with Hyper-Threading Technology) 6,0 |
| IBM Power4, 1.3 GHz 4,195 |
| rd, 1.0 GHz Pentium III processor n XP1000, 667 MHz 21264A 6/575, 575 MHz 21264 1,267 |
| 0, 600 MHz 21164 649 600, 500 MHz 481 |
| 800 MHz 280 |
| Hz 183 |
| • 117 |
| |
| 52%/year |
| |
| |
| |
| |
| |
| |
| 1994 1996 1998 2000 2002 : |



"The La-Z-Boy programmer era of relying on hardware designers to make their programs go faster without lifting a finger is officially over."

Hennessy & Patterson [2017]

100,000

10,000

780) ance Perf





[1] J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach, sixth edition. Morgan Kaufmann, 2014.







https://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law_-_2011.svg







https://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law_-_2011.svg





https://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law_-_2011.svg

© Christian Lessig, 2018



0

2016

© Christian Lessig, 2018



© Christian Lessig, 2018



© Christian Lessig, 2018



(almost) all current "computers" are parallel

Parallel programming?





What makes parallel programming difficult?

Parallel programming?

"[Serial] algorithms have improved faster than clock over the last 15 years. [Parallel] computers are unlikely to be able to take advantage of these advances because they require new programs and new algorithms."

© Christian Lessig, 2018

Gordon Bell (1992)

G. Bell, "Massively parallel computers: why not parallel computers for the masses?," in The Fourth Symposium on the Frontiers of Massively Parallel Computation, 1992, pp. 292-297.



© Christian Lessig, 2018

Why parallel: the hardware side

© Christian Lessig, 2018



i7-2600K Clockspeed versus Power-Consumption

https://forums.anandtech.com/threads/power-consumption-scaling-with-clockspeed-and-vcc-for-the-i7-2600k.2195927/

R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," IEEE J. Solid-State Circuits, vol. 32, no. 8, pp. 1210–1216, 1997.

$P = C \cdot V^2 \cdot f$

R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," IEEE J. Solid-State Circuits, vol. 32, no. 8, pp. 1210–1216, 1997.

capacitance voltage

R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," IEEE J. Solid-State Circuits, vol. 32, no. 8, pp. 1210–1216, 1997.

capacitance voltage

© Christian Lessig, 2018

i7-2600K Clockspeed versus Power-Consumption

https://forums.anandtech.com/threads/power-consumption-scaling-with-clockspeed-and-vcc-for-the-i7-2600k.2195927/

© Christian Lessig, 2018

 \sim

Watts/cm

1000 100 hot plate 10 Intel Pentium Pro Intel Pentium i386 • i486 0.5μ 1μ

After: http://research.ac.upc.edu/HPCseminar/SEM9900/Pollack1.pdf

nuclear reactor

Intel Pentium III Intel Pentium II

0.25µ 0.125µ

© Christian Lessig, 2018

 \sim

Watts/cm

After: http://research.ac.upc.edu/HPCseminar/SEM9900/Pollack1.pdf

 n^2 Watts/cm

After: http://research.ac.upc.edu/HPCseminar/SEM9900/Pollack1.pdf

processor heat limit

© Christian Lessig, 2018

D energy

Energy is critical: Handheld: major factor for customer satisfaction Warehouse scale computing: major cost factor

Energy is critical:
Handheld: major factor for customer satisfaction
Warehouse scale computing: major cost factor

... and to keep our planet alive.

How to get around heat limit?

How to get around heat limit?

(or be as energy efficient as possible)

© Christian Lessig, 2018

processor heat limit

specialize

parallelize

© Christian Lessig, 2018

processor heat limit

specialize

parallelize

instruction level

data-parallel

multi-core

processor heat limit field programm. graphics gate array processor specialize tensor neuromorphic processing

computing unit

© Christian Lessig, 2018

parallelize

instruction level

data-parallel

multi-core

https://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law_-_2011.svg

Parallelism to avoid heat limit (and increase energy efficiency)

Clock frequency

Power

FP throughput

https://wiki.rice.edu/confluence/download/attachments/4435861/comp322-s16-lec1-slides.pdf

© Christian Lessig, 2018

Nvidia Fermi Nvidia Kepler (2012)(2010)1.0 GHz 1.3 GHz 195 Watt 250 Watt 1310 GFlops 665 GFlops

© Christian Lessig, 2018

Input Device

https://en.wikipedia.org/wiki/Von_Neumann_architecture

Von Neumann architecture

© Christian Lessig, 2018

Input Device

https://en.wikipedia.org/wiki/Von_Neumann_architecture

Von Neumann architecture

https://en.wikipedia.org/wiki/Von_Neumann_architecture

Von Neumann architecture

© Christian Lessig, 2018

2000

Data: https://en.wikipedia.org/wiki/CAS_latency, http://www.intel.com/pressroom/kits/quickreffam.htm

© Christian Lessig, 2018

frequency 1 GHz

bandwidth latency 20 ns 100 MT/s

2000

2003

Data: https://en.wikipedia.org/wiki/CAS_latency, http://www.intel.com/pressroom/kits/quickreffam.htm

© Christian Lessig, 2018

bandwidth frequency latency 20 ns 100 MT/s1 GHz 333 MT/s 2 GHz 15 ns

2000

2003

2007

Data: https://en.wikipedia.org/wiki/CAS_latency, http://www.intel.com/pressroom/kits/quickreffam.htm

frequency 1 GHz 2 GHz 4.5 GHz

bandwidth latency 100 MT/s20 ns 333 MT/s 15 ns 800 MT/s 10 ns

1.0 8.0 0.6 0.4 0.2

Data: https://en.wikipedia.org/wiki/CAS_latency, http://www.intel.com/pressroom/kits/quickreffam.htm

normalized performance

© Christian Lessig, 2018

How to get around von Neumann bottleneck?

von Neumann bottleneck

caching

von Neumann bottleneck pipelining caching

Why parallel? Caching:

© Christian Lessig, 2018

main memory

Why parallel? Caching:

© Christian Lessig, 2018

main memory

von Neumann bottleneck pipelining caching

© Christian Lessig, 2018

© Christian Lessig, 2018

© Christian Lessig, 2018

© Christian Lessig, 2018

© Christian Lessig, 2018

compute memory

© Christian Lessig, 2018

memory

© Christian Lessig, 2018

memory

typically realized by compiler or hardware

Instruction level parallelism: exploit indepence at assembler level Pipelining • Different arithmetic units

Functional unit

Integer ALU

FP add

FP divide (also integer divide)

J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach, Seventh ed. Morgan Kaufmann, 2017, p. C-53

© Christian Lessig, 2018

Data memory (integer and FP loads)

FP multiply (also integer multiply)

| Latency | Initiation interval |
|---------|---------------------|
| 0 | 1 |
| 1 | 1 |
| 3 | 1 |
| 6 | 1 |
| 24 | 25 |

Instruction level parallelism: exploit indepence at assembler level Pipelining

• Different arithmetic units

=> Exploited since 1980s but no longer significant improvements possible

Further reading

- er-1997.pdf

 http://cacm.acm.org/magazines/2009/5/24648-spending-moores-dividend/fulltext

• J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach, fourth edition. Morgan Kaufmann, 2007. http://cva.stanford.edu/classes/cs99s/

 http://research.ac.upc.edu/HPCseminar/SEM9900/Pollack1.pdf http://groups.csail.mit.edu/cag/raw/documents/Waingold-Comput-