

# High-Performance Scientific Computing

## Lecture 1: Intro

MATH-GA 2011 / CSCI-GA 2945 · September 5, 2012

# Today

About this class

HPC: A look around

A taste of what's to come

Extra stuff

# Outline

About this class

HPC: A look around

A taste of what's to come

Extra stuff

## Course Goal

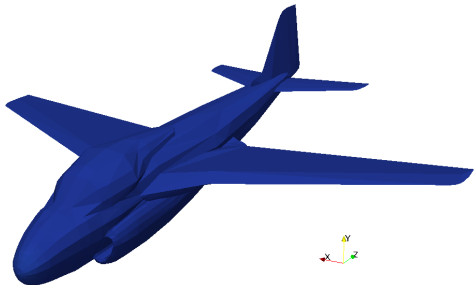
Slow code goes in.

Speedy code goes out.

# Define slow?

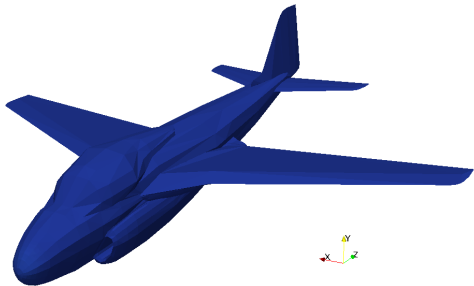
## Define slow?

- Took about 30 days on a single PC.



## Define slow?

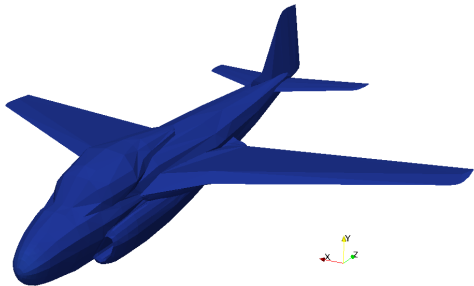
- Took about 30 days on a single PC.
- Took about a day on a GPU.



## Define slow?

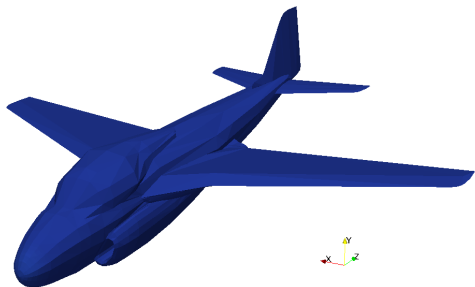
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.





## Define slow?

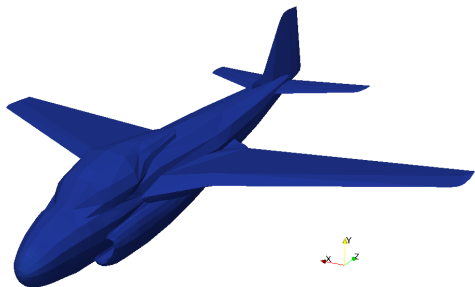


- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

## Define slow?



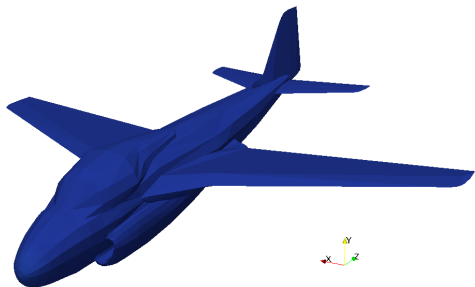
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements.  
Now?

## Define slow?



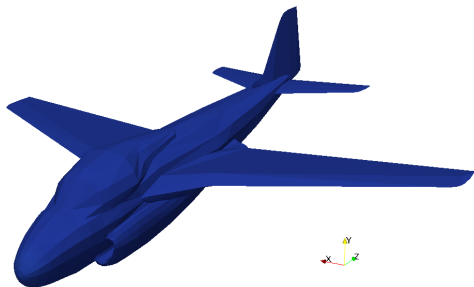
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements.  
Now?
- Anything else?

# Define slow?



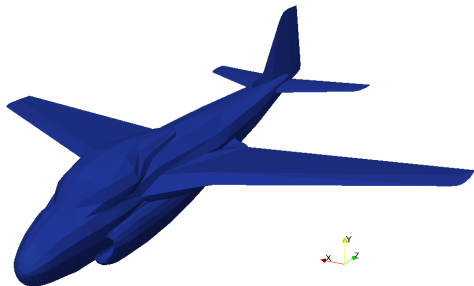
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements. Now?
- Anything else?
- $16\times$  the cost!

## Define slow?



Realistic (high-fidelity) problems are big.

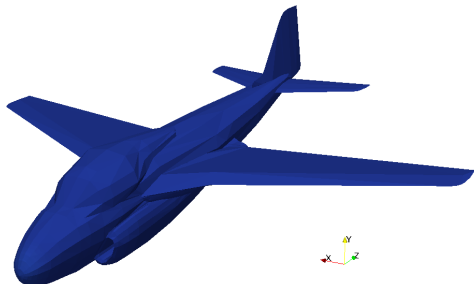
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements. Now?
- Anything else?
- $16\times$  the cost!

## Define slow?



Realistic (high-fidelity) problems are big.

→ You'll need a bigger hammer.

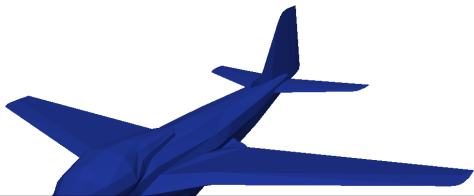
- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements. Now?
- Anything else?
- $16\times$  the cost!

## Define slow?



Realistic (high-fidelity) problems are big.

→ You'll need a bigger hammer.

**You'll need to know how to use the bigger hammer.**

- Took about 30 days on a single PC.
- Took about a day on a GPU.

That's still pretty crude-looking.

Suppose I'd like to double the resolution. (i.e. cut the mesh width  $h$  in half.)

- Had  $K$  elements. Now?
- Anything else?
- $16\times$  the cost!

# Course Outline

## Part 1: Do (~ 4)

- Write, run programs (C)
- Use tools (make, git, gdb)
- OpenMP, MPI, OpenCL
- Correctness in each

## Part 2: Understand (~3)

- Measure and understand performance
- Basic machine architecture
- CPU machine model
- GPU machine model

## Part 3: Refine (~3)

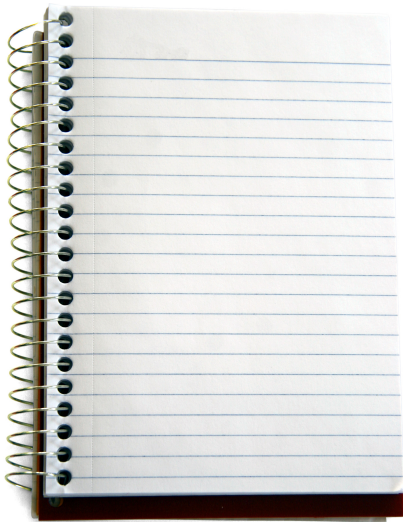
- Advanced tools & languages
- Work partitioning
- Common patterns
- Load balancing

## Part 4: Apply

- Find a project (start looking *now!*)
- Pitch it to us (5 min)
- Apply what you've learned
- Present your work (2)



# Sign-up sheet



# Survey



- Home department

# Survey



- Home department
- Degree

# Survey



- Home department
- Degree
- Longest program ever written?

# Survey



- Home department
- Degree
- Longest program ever written?
  - in C?

# Survey



- Home department
- Degree
- Longest program ever written?
  - in C?
- Parallel?

# Survey



- Home department
- Degree
- Longest program ever written?
  - in C?
- Parallel?
- Already have a project?

[illegible]

[Class advertisement](#)

01234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768697071727374757677787980818283848586878889909192939495969798991001011021031041051061071081091101111121131141151161171181191201211221231241251261271281291301311321331341351361371381391401411421431441451461471481491501511521531541551561571581591601611621631641651661671681691701711721731741751761771781791801811821831841851861871881891901911921931941951961971981992002012022032042052062072082092102112122132142152162172182192202212222232242252262272282292302312322332342352362372382392402412422432442452462472482492502512522532542552562572582592602612622632642652662672682692702712722732742752762772782792802812822832842852862872882892902912922932942952962972982993003013023033043053063073083093103113123133143153163173183193203213223233243253263273283293303313323333343353363373383393403413423433443453463473483493503513523533543553563573583593603613623633643653663673683693703713723733743753763773783793803813823833843853863873883893903913923933943953963973983994004014024034044054064074084094104114124134144154164174184194204214224234244254264274284294304314324334344354364374384394404414424434444454464474484494504514524534544554564574584594604614624634644654664674684694704714724734744754764774784794804814824834844854864874884894904914924934944954964974984995005015025035045055065075085095105115125135145155165175185195205215225235245255265275285295305315325335345355365375385395405415425435445455465475485495505515525535545555565575585595605615625635645655665675685695705715725735745755765775785795805815825835845855865875885895905915925935945955965975985996006016026036046056066076086096106116126136146156166176186196206216226236246256266276286296306316326336346356366376386396406416426436446456466476486496506516526536546556566576586596606616626636646656666676686696706716726736746756766776786796806816826836846856866876886896906916926936946956966976986997007017027037047057067077087097107117127137147157167177187197207217227237247257267277287297307317327337347357367377387397407417427437447457467477487497507517527537547557567577587597607617627637647657667677687697707717727737747757767777787797807817827837847857867877887897907917927937947957967977987998008018028038048058068078088098108118128138148158168178188198208218228238248258268278288298308318328338348358368378388398408418428438448458468478488498508518528538548558568578588598608618628638648658668678688698708718728738748758768778788798808818828838848858868878888898908918928938948958968978988999009019029039049059069079089099109119129139149159169179189199209219229239249259269279289299309319329339349359369379389399409419429439449459469479489499509519529539549559569579589599609619629639649659669679689699709719729739749759769779789799809819829839849859869879889899909919929939949959969979989991000100110021003100410051006100710081009101010111012101310141015101610171018101910201021102210231024102510261027102810291030103110321033103410351036103710381039104010411042104310441045104610471048104910501051105210531054105510561057105810591060106110621063106410651066106710681069107010711072107310741075107610771078107910801081108210831084108510861087108810891090109110921093109410951096109710981099110011011102110311041105110611071108110911101111111211131114111511161117111811191120112111221123112411251126112711281129113011311132113311341135113611371138113911401141114211431144114511461147114811491150115111521153115411551156115711581159116011611162116311641165116611671168116911701171117211731174117511761177117811791180118111821183118411851186118711881189119011911192119311941195119611971198119912001201120212031204120512061207120812091210121112121213121412151216121712181219122012211222122312241225122612271228122912301231123212331234123512361237123812391240124112421243124412451246124712481249125012511252125312541255125612571258125912601261126212631264126512661267126812691270127112721273127412751276127712781279128012811282128312841285128612871288128912901291129212931294129512961297129812991300

2. High-impact

Aug 21, 2012

If you're fit  
open to sto

Gradation/Evaluation [\[Edit\]](#)

If you will be taking the class for credit, the

- Weekly homework

- Material
- [\[Edit\]](#)

Books [\[Edit\]](#)

© Fingerprint Pro

- For OpenGL and GPU programming, we will also be referring to the following sources:

OpenCL in Action: How to Accelerate Graphics and Computation	<a href="#">@ from NYU net</a>
OpenCL Programming Guide	<a href="#">@ from NYU net</a>

[Primary source material](#) [\[Edit\]](#)

These are the technical standards on which

- **CPS specification**

- Collected Wisdom
- [\[Edit\]](#)

- [Tool Chest Sheet](#)

- Previous Ed

• [QFHE'18](#)





# Class web page

TeachingHPCFall2012

High-Performance Scientific Computing [Edit]

Class Time/Location: Wednesday 5-7pm, Room 522 Warren Weaver Hall

Instructor: @Martha Berger, @Andreas Kloeckner

Email: @martha.berger@nyu.edu, @andreas.kloeckner@nyu.edu

Office: Courant Institute, Warren Weaver Hall, Rooms 1123, 1150A

Office Hours: Andrius: Wednesdays, 2-4pm WFH 1205A

Class Meetings: @nyu.ted.lytchak

Class Location: WH

Class advertisement

Curriculum

1. Introduction to Scientific Computing
2. C++
3. C++
4. C++
5. C++
6. C++
7. C++
8. C++
9. C++
10. C++
11. C++
12. C++
13. C++
14. C++
15. C++
16. C++
17. C++
18. C++
19. C++
20. C++
21. C++
22. C++
23. C++
24. C++
25. C++
26. C++
27. C++
28. C++
29. C++
30. C++
31. C++
32. C++
33. C++
34. C++
35. C++
36. C++
37. C++
38. C++
39. C++
40. C++
41. C++
42. C++
43. C++
44. C++
45. C++
46. C++
47. C++
48. C++
49. C++
50. C++
51. C++
52. C++
53. C++
54. C++
55. C++
56. C++
57. C++
58. C++
59. C++
60. C++
61. C++
62. C++
63. C++
64. C++
65. C++
66. C++
67. C++
68. C++
69. C++
70. C++
71. C++
72. C++
73. C++
74. C++
75. C++
76. C++
77. C++
78. C++
79. C++
80. C++
81. C++
82. C++
83. C++
84. C++
85. C++
86. C++
87. C++
88. C++
89. C++
90. C++
91. C++
92. C++
93. C++
94. C++
95. C++
96. C++
97. C++
98. C++
99. C++
100. C++

Updates [Edit]

Aug 21, 2012

Aug 9, 2012

Grading/Evaluation [Edit]

Material [Edit]

Books [Edit]

For OpenCL and GPU programming, we will also be referring to the following resources:

OpenCL: A Beginner's Guide to Accelerated Graphics and Computation @nyu.ted.lytchak @nyu.ted.lytchak

OpenCL: Programming Guide @nyu.ted.lytchak @nyu.ted.lytchak

Heterogeneous Computing with OpenCL @nyu.ted.lytchak @nyu.ted.lytchak

Primary source material [Edit]

These are the technical standards on which the class will be based. Write yourselves a bit technical, these documents define whether the program

Collected Wisdom [Edit]

Previous Editions of the Class [Edit]

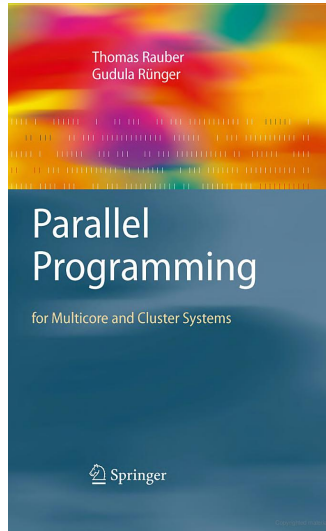
bit.ly/hpc12

Posted: Virtual machine image (instructions in HW1)

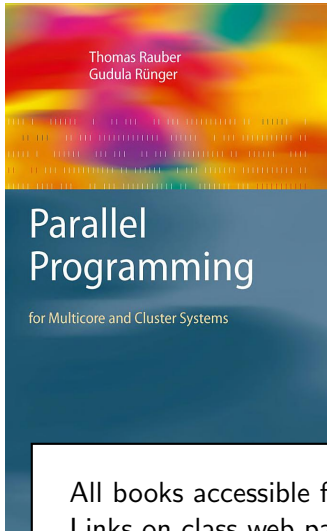
Posted: Homework set 1  
(C warm-up, git, mechanics)  
*Due next week.*

hpc12@tiker.net

# Book 1

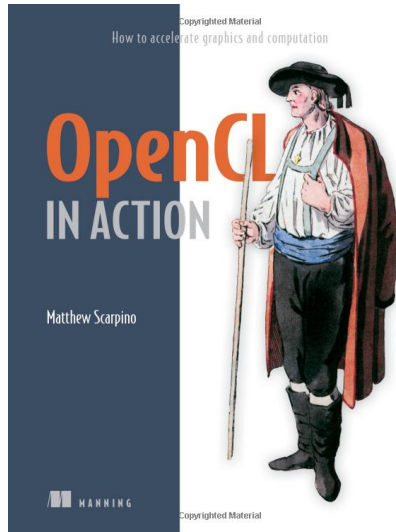


# Book 1

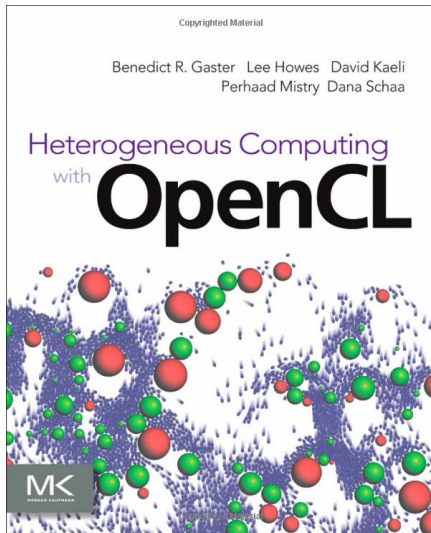


All books accessible from NYU network.  
Links on class web page.

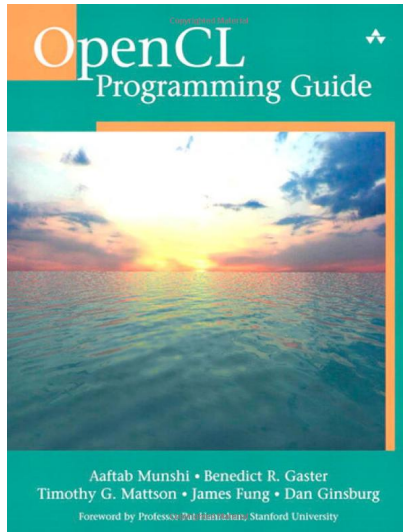
## Books 2–4



## Books 2–4



## Books 2–4



# Grading

- 60% Weekly homework
- 40% Final project



Smile! You're on camera



Lecture video will be posted soon after each class.

# Outline

About this class

**HPC: A look around**

A taste of what's to come

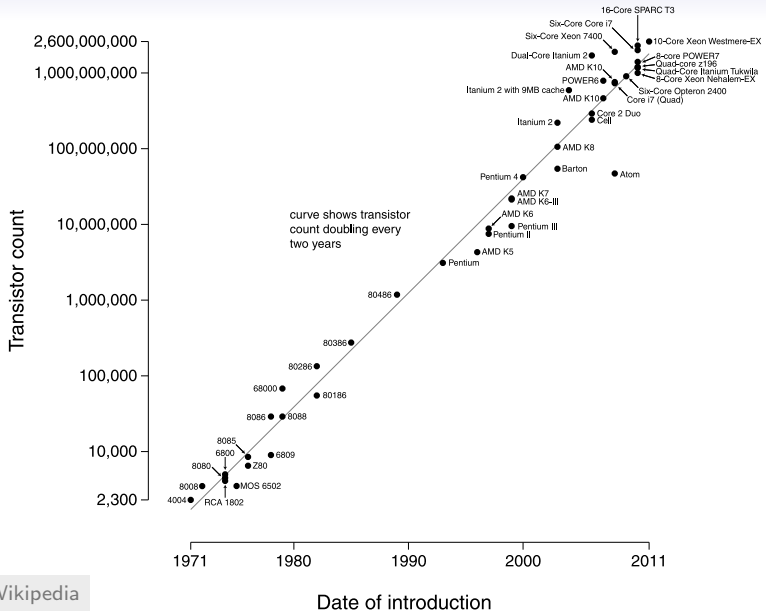
Extra stuff

My program is taking too long.

My program is taking too long.

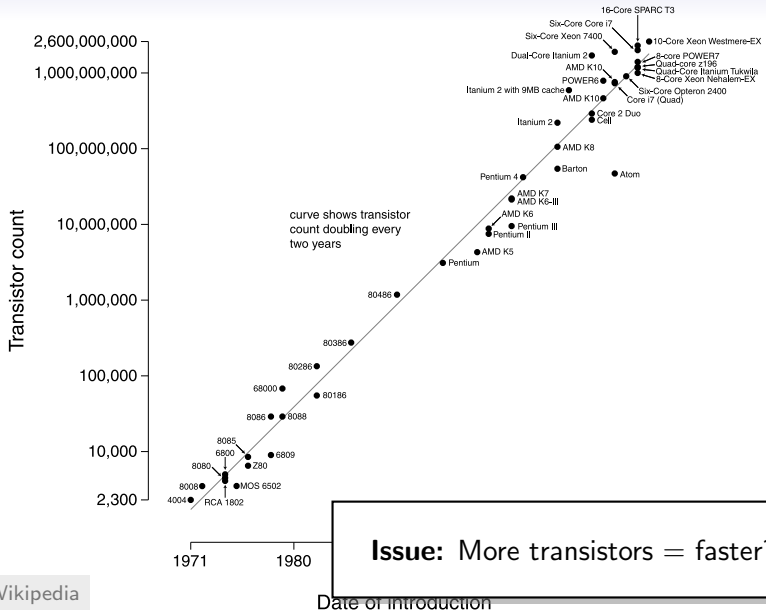
Maybe it'll get faster if I wait long enough?

## Moore's law



Wikipedia

# Moore's law



Wikipedia

# So what does it mean?

So what does it mean?





So what *does* it mean?

$$\frac{\text{Work}}{s} = \text{Clock Frequency} \times \text{Work/Clock}$$

So what *does* it mean?

$$\frac{\text{Work}}{s} = \text{Clock Frequency} \times \text{Work/Clock}$$

# Dennard scaling of MOSFETs

Parameter	Factor
Dimension	$1/\kappa$
Voltage	$1/\kappa$
Current	$1/\kappa$
Capacitance	$1/\kappa$
Delay Time	$1/\kappa$
Power dissipation/circuit	$1/\kappa^2$
Power density	1

[Dennard et al. '74, via Bohr '07]

# Dennard scaling of MOSFETs

Parameter	Factor
Dimension	$1/\kappa$
Voltage	$1/\kappa$
Current	$1/\kappa$
Capacitance	$1/\kappa$
<b>Delay Time</b>	$1/\kappa$
Power dissipation/circuit	$1/\kappa^2$
Power density	1

[Dennard et al. '74, via Bohr '07]

$$\text{Frequency} = \text{Delay time}^{-1}$$

# Dennard scaling of MOSFETs

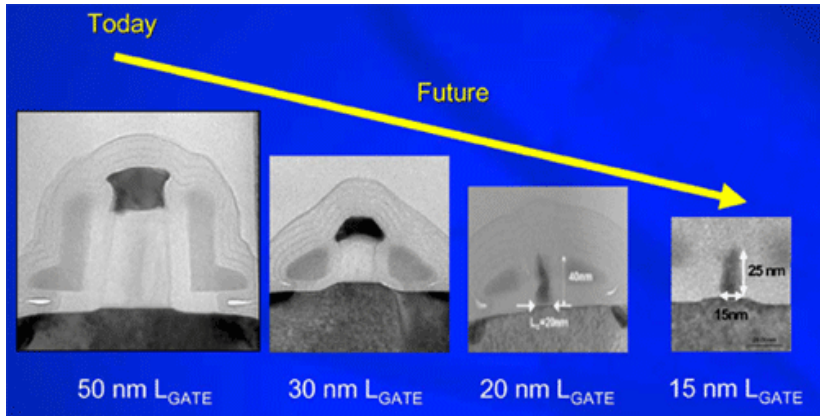
Parameter	Factor
Dimension	$1/\kappa$
Voltage	$1/\kappa$
Current	$1/\kappa$
Capacitance	$1/\kappa$
Delay Time	$1/\kappa$
Power dissipation/circuit	$1/\kappa^2$
Power density	1

[Dennard et

$$\text{Frequency} = \text{Delay time}^{-1}$$

'New' problem at small scale:  
Sub-threshold leakage (due to  
low voltage, small structure)

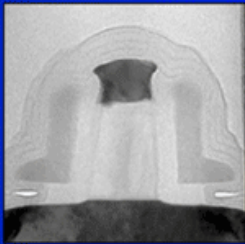
# MOSFETs



Intel Corporation

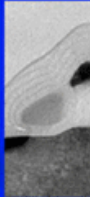
# MOSFETs

Today

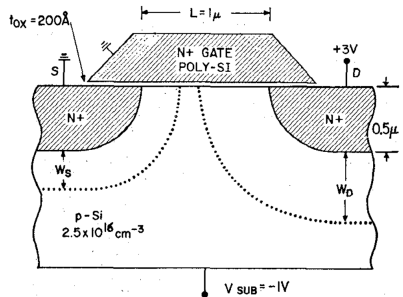


50 nm  $L_{\text{GATE}}$

Future



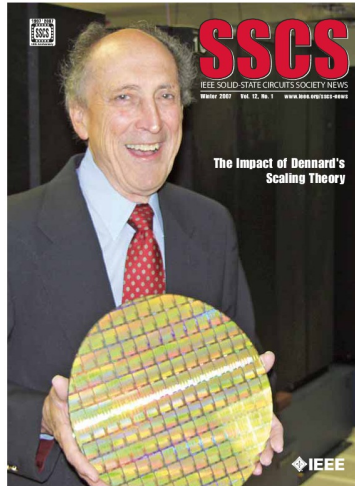
30 nm



[Dennard et al. '74]

Intel Corporation

# Robert Dennard





So what *does* it mean?

$$\frac{\text{Work}}{s} = \text{Clock Frequency} \times \text{Work/Clock}$$

## So what *does* it mean?

$$\frac{\text{Work}}{s} = \cancel{\text{Clock Frequency}} \times \text{Work/Clock}$$

So what *does* it mean?

$$\frac{\text{Work}}{s} = \cancel{\text{Clock Frequency}} \times \text{Work/Clock}$$

## Instructions per clock: Intel

CPU	IPC	Year
Pentium 1	1.1	1993
Pentium MMX	1.2	1996
Pentium 3	1.9	1999
Pentium 4 (Willamette)	1.5	2003
Pentium 4 (Northwood)	1.6	2003
Pentium 4 (Prescott)	1.8	2003
Pentium 4 (Gallatin)	1.9	2003
Pentium D	2	2005
Pentium M	2.5	2003
Core 2	3	2006

Charlie Brej, <http://brej.org/blog/?p=15>

# Instructions per clock: AMD

CPU	IPC	Year
K6 II	1.1	1998
K6 III	1.3	1999
Athlon B	1.9	1999
Athlon XP	2	2001
Athlon 64	2.3	2003
Athlon 64 X2	2.5	2005

Charlie Brej, <http://brej.org/blog/?p=15>

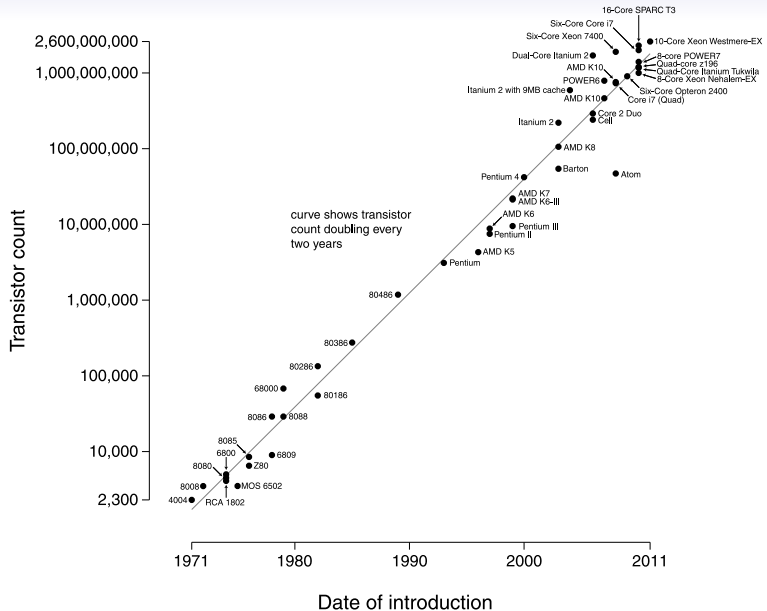
## Instructions per clock: AMD

CPU	IPC	Year
K6 II	1.1	1998
K6 III	1.3	1999
Athlon B	1.9	1999
Athlon XP	2	2001
Athlon 64	2.3	2003
Athlon 64 X2	2.5	2005

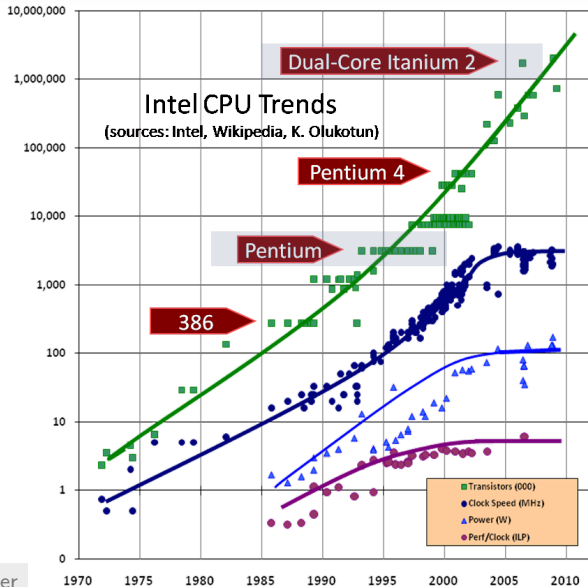
A failure of the programming model!

Charlie Brey, <http://brey.org/brey>

# Processor Evolution



# Processor Evolution



Herb Sutter



# Parallel Programming

High-performance computing *is* parallel computing. (...)

Parallel programming is ...

- *inevitable* (if you'd like maximal throughput)
- *hard*

**Problem:** People don't think 'that way'.

“Automatic parallelization” has largely been a failure.

→ People have to be taught to think that way.

# Parallel Programming

High-performance computing *is* parallel computing. (...)

Parallel programming is ...

- *inevitable* (if you'd like maximal throughput)
- *hard*

**Problem:** People don't think 'that way'.

“Automatic parallelization” has largely been a failure.

→ People have to be taught to think that way. :)

# Parallel Programming

High-performance computing *is* parallel computing. (...)

Parallel programming is ...

- *inevitable* (if you'd like maximal throughput)
- *hard*

**Problem:** People don't think 'that way'.

“Automatic parallelization” has largely been a failure.

→ People have to be taught to think that way.

# Parallel Programming

High-performance computing *is* parallel computing. (...)

Parallel programming is ...

- *inevitable* (if you'd like maximal throughput)
- *hard*

**Problem:** People don't think 'that way'

"Automatic parallelization" has

→ People have to be taught to

**Bad news:** Parallelism might not even be our worst problem.

Don't just need to compute, also need to transmit information (to memory, say)

## More bad news from Dr. Dennard

Parameter	Factor
Dimension	$1/\kappa$
Line Resistance	$\kappa$
Voltage drop	$\kappa$
Response time	1
Current density	$\kappa$

[Dennard et al. '74, via Bohr '07]

- The above scaling law is for on-chip interconnects.
- Off-chip: Similar consideration.  
Current  $\sim$  Power vs. response time

## More bad news from Dr. Dennard

Parameter	Factor
Dimension	$1/\kappa$
Line Resistance	$\kappa$
Voltage drop	$\kappa$
Response time	1
Current density	$\kappa$

[Dennard et al.]

- The above scaling law is for on-chip
- Off-chip: Similar considerations  
Current  $\sim$  Power vs. re

Getting information from

- processor to memory
  - one computer to the next
- is
- slow (in *latency*)
  - power-hungry

# Summary

Main problems for this class:

1. Express parallelism
2. Express communication/synchronization
3. Analyze, understand run time  
Both theoretically and practically (by measurement)

1 and 2 are **language issues!**

# Summary

Main problems for this class:

1. Express parallelism
2. Express communication/synchronization
3. Analyze, understand run time  
Both theoretically and practically (by measurement)

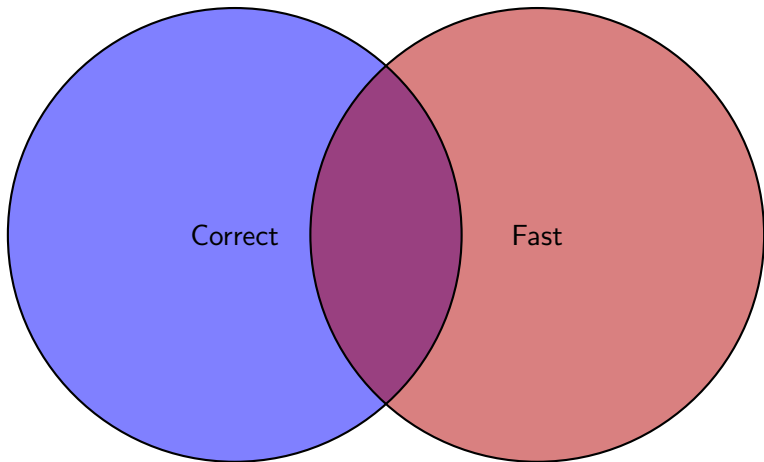
1 and 2 are **language issues!**

A little bit of terminology:

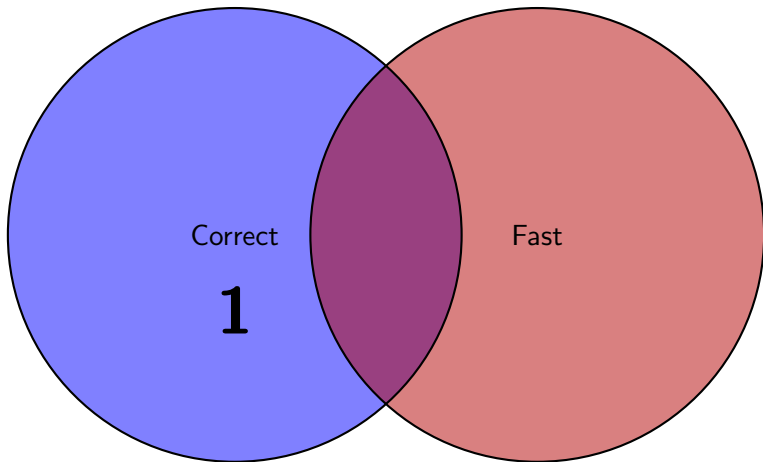
- Speedup, Efficiency, Scaling
- “Amdahl’s law”:  
Speed up 10% of your program by a factor of 10?



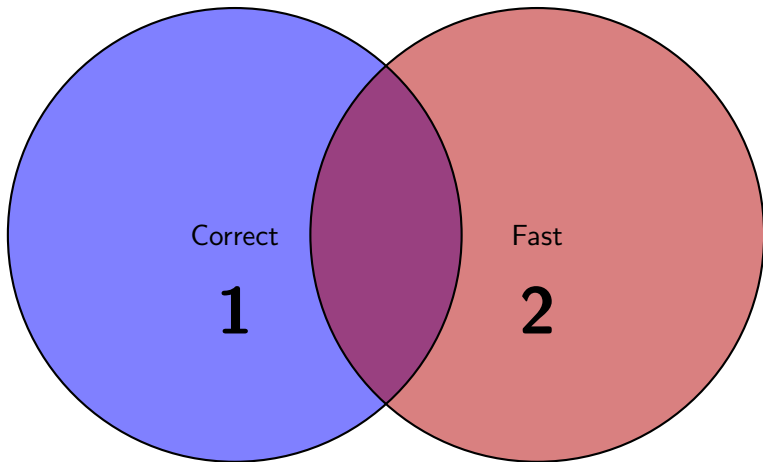
# Parallelism as a Language Question



# Parallelism as a Language Question



# Parallelism as a Language Question



# Outline

About this class

HPC: A look around

**A taste of what's to come**

Extra stuff

# Demo time!

# Outline

About this class

HPC: A look around

A taste of what's to come

Extra stuff

# HPC as a Spectator Sport

Rank ↕	Rmax Rpeak (Pflops) ↕	Name ↕	Computer design Processor type, interconnect ↕	Vendor ↕	Site Country, year ↕	Operating system ↕
1	16.324 20.132	<i>Sequoia</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Lawrence Livermore National Laboratory United States, 2011	Linux (RHEL and CNL)
2	10.510 11.280	<i>K computer</i>	<b>RIKEN</b> SPARC64 VIIIfx, Tofu	Fujitsu	RIKEN Japan, 2011	Linux
3	8.162 10.066	<i>Mira</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Argonne National Laboratory United States, 2012	Linux
4	2.897 3.185	<i>SuperMUC</i>	<b>iDataPlex DX360M4</b> Xeon E5-2680, Infiniband	IBM	Leibniz-Rechenzentrum Germany, 2012	Linux
5	2.566 4.701	<i>Tianhe-1A</i>	<b>NUDT YH Cluster</b> Xeon 5670 + Tesla 2050, Arch[4]	NUDT	National Supercomputing Center of Tianjin China, 2010	Linux
6	1.941 2.627	<i>Jaguar</i>	<b>Cray XT5</b> Opteron 6274 + Tesla 2090, Cray Gemini	Cray	Oak Ridge National Laboratory United States, 2009	Linux (CLE)
7	1.725 2.097	<i>Fermi</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	CINECA Italy, 2012	Linux
8	1.380 1.677	<i>JuQUEEN</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Forschungszentrum Jülich Germany, 2012	Linux
9	1.359 1.667	<i>Curie</i>	<b>Bullx B510</b> Xeon E5-2680, Infiniband	Bull	TGCC at CEA, and GENCI France, 2012	Linux (bullx)
10	1.271 2.984	<i>Nebulae</i>	<b>TC3600 Blade</b> Xeon 5650 + Tesla 2050, InfiniBand	Dawning	National Supercomputing Center in Shenzhen NSCS China, 2010	Linux

<http://top500.org>

# HPC as a Spectator Sport

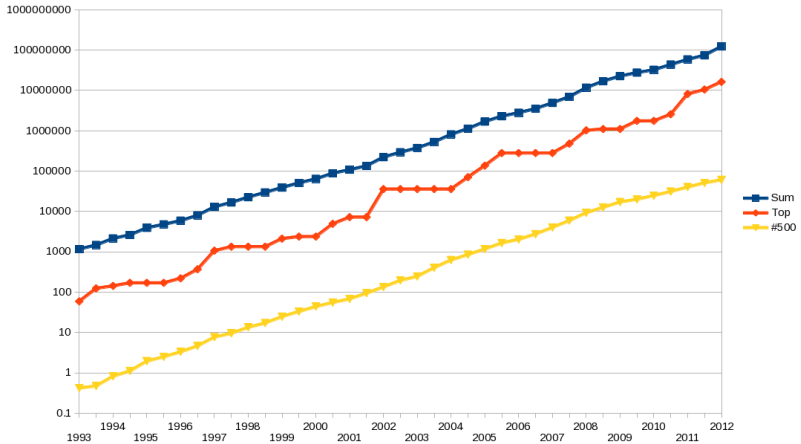
Rank ↕	Rmax Rpeak (Pflops) ↕	Name ↕	Computer design Processor type, interconnect ↕	Vendor ↕	Site Country, year ↕	Operating system ↕
1	16.324 20.132	<i>Sequoia</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Lawrence Livermore National Laboratory 🇺🇸 United States, 2011	Linux (RHEL and CNL)
2	10.510 11.280	<i>K computer</i>	<b>RIKEN</b> SPARC64 VIIIfx, Tofu	Fujitsu	RIKEN 🇯🇵 Japan, 2011	Linux
3	8.162 10.066	<i>Mira</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Argonne National Laboratory 🇺🇸 United States, 2012	Linux
4	2.897 3.185	<i>SuperMUC</i>	<b>iDataPlex DX360M4</b> Xeon E5-2680, Infiniband	IBM	Leibniz-Rechenzentrum 🇩🇪 Germany, 2012	Linux
5	2.566 4.701	<i>Tianhe-1A</i>	<b>NUDT YH Cluster</b> Xeon 5670 + Tesla 2050, Arch[4]	NUDT	National Supercomputing Center of Tianjin 🇨🇳 China, 2010	Linux
6	1.941 2.627	<i>Jaguar</i>	<b>Cray XT5</b> Opteron 6274 + Tesla 2090, Cray Gemini	Cray	Oak Ridge National Laboratory 🇺🇸 United States, 2009	Linux (CLE)
7	1.725 2.097	<i>Fermi</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	CINECA 🇮🇹 Italy, 2012	Linux
8	1.380 1.677	<i>JuQUEEN</i>	<b>Blue Gene/Q</b> PowerPC A2, Custom	IBM	Forschungszentrum Jülich 🇩🇪 Germany, 2012	Linux
9	1.359 1.667	<i>Curie</i>	<b>Bullx B510</b> Xeon E5-2680, Infiniband	Bull	TGCC at CEA, and GENCI 🇫🇷 France, 2012	Linux (bullx)
10	1.271 2.984	<i>Nebulae</i>	<b>TC3600 Blade</b> Xeon 5650 + T InfiniBand		National Supercomputing Center in	

<http://top500.org>

Know your gigas, teras, petas, and exas.



# HPC as a Spectator Sport



<http://top500.org>

# Questions?

?

## Image Credits

- Notebook: [sxc.hu/abeall](http://sxc.hu/abeall)
- Question mark: [sxc.hu/svilen001](http://sxc.hu/svilen001)
- Camera: [sxc.hu/Kolobsek](http://sxc.hu/Kolobsek)
- Gordon Moore: Wikipedia