HPI Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

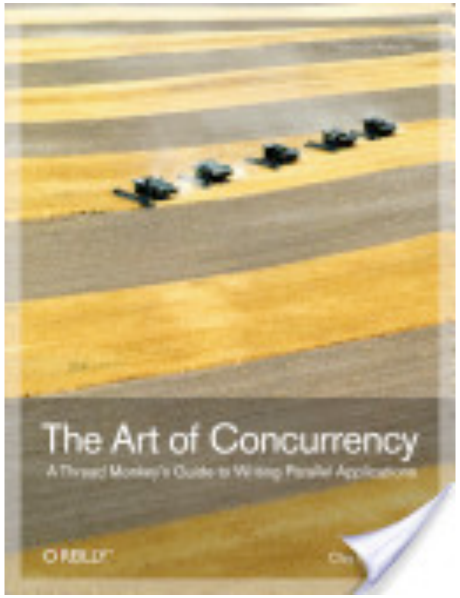
Programmierung Paralleler und Verteilter Systeme (PPV)

Sommer 2015

Frank Feinbube, M.Sc., Felix Eberhardt, M.Sc.,
Prof. Dr. Andreas Polze

Course Design

- Lectures covering theoretical and practical aspects of distribution, concurrency and parallelism in hardware and software
- This is a course about concepts,
  not a programming tutorial !
- Practical assignments
  - □ Earn extra 3 ETCS credits
  - □ Implementation of parallel algorithms with various programming models
  - □ Presentation at OSM research seminar
- 30 minutes oral exam / September
- Literature list on course home page



*The Art of Concurrency:
A Thread Monkey's Guide to
Writing Parallel Applications*

Clay Breshears
O'Reilly Media, Inc.
2009

Course Topics

- Motivation
- Terminology
- Workload & Metriken
- Konzepte der Parallelverarbeitung
  - Coroutinen, Fork & Join, ParBegin/ParEnd, expl. vs. impl. Parallelität
  - Shared Address Space vs. Message Passing
  - Datenparallelität vs. Kontrollparallelität
  - idealisierte Parallelrechner: PRAM, LogP, BSP

Course Topics (contd.)

- Synchrone Parallelität
  - SIMD-Rechner: Aufbau, Datenparallelität, Virtuelle Prozessoren
  - CM-2, MasPar, DAP 610
- Kommunikation
  - Verbindungsstrukturen
  - Datenaustausch, Vektorreduktion
- Probleme bei synchroner Parallelität
  - virtuelle vs. physische Prozessoren
  - I/O-Problem, Netzwerk-Bandbreite
  - Mehrbenutzerbetrieb, Fehlertoleranz
- High Performance Fortran
- Parallaxis - Beispiel für datenparallele Programmierung

Course Topics (contd.)

- Asynchrone Parallelität
- MIMD-Rechner, SPMD-Ansatz
  - Synchronisation und Kommunikation in MIMD-Systemen
  - Softwarelösung, Hardwarelösung, Semaphore, Monitore, Nachrichten, RPC
- Probleme bei asynchroner Parallelität:
  - inkonsistente Daten, Verklemmungen, Lastbalanzierung
- Shared Memory Programmierung
- Advanced Shared Memory Programmierung
- GPU Computing mit OpenCL

Course Topics (contd.)

- Parallelität in verteilten Systemen – Überblick

- Modelle für Shared Nothing Computing
- Parallelität in verteilten Systemen
    - MPI / PVM
    - Object Space / Linda / Koordinationssprachen
    - Responsive Cluster Computing
- Trends / Ausblick

Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Why Parallel ?

Programmierung Paralleler und Verteilter Systeme (PPV)

Sommer 2015

Frank Feinbube, M.Sc., Felix Eberhardt, M.Sc.,
Prof. Dr. Andreas Polze

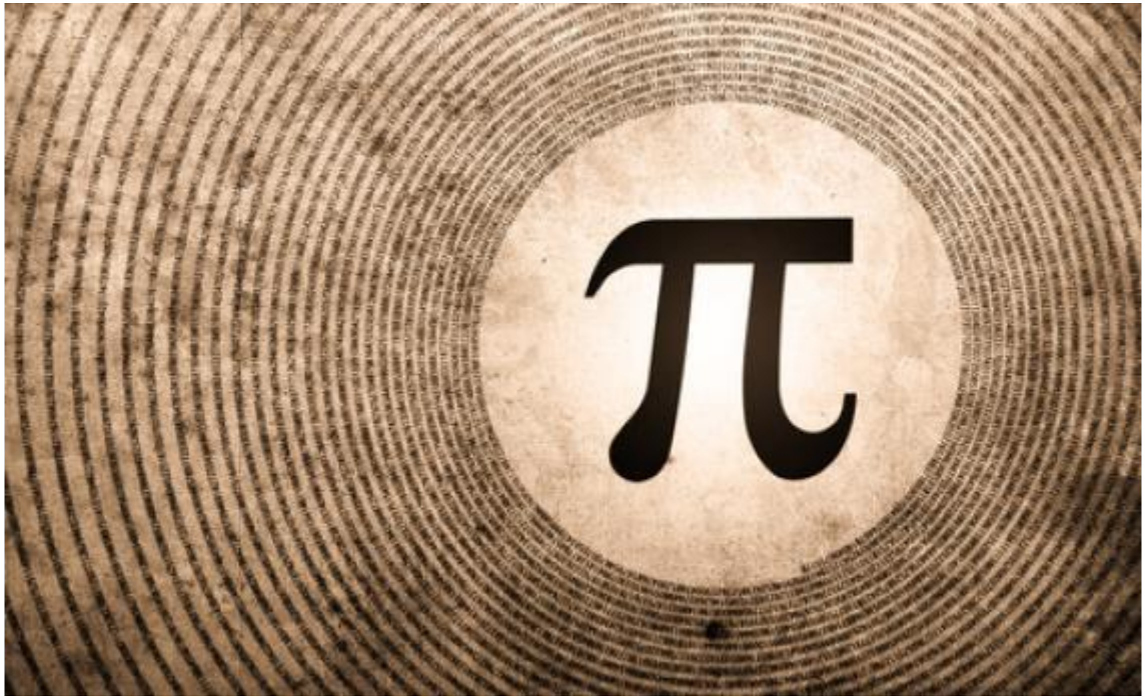
Computer Markets

- Embedded Computing
  - Real-time systems, nearly everywhere
  - Power consumption and price as major issue
- Desktop Computing
  - Home computers
  - **Performance / price** ratio as major issue
- Servers
  - **Performance** and availability is key
  - Business service provisioning as major goal
  - Web servers, banking back-end, order processing, ...

Awesome Applications

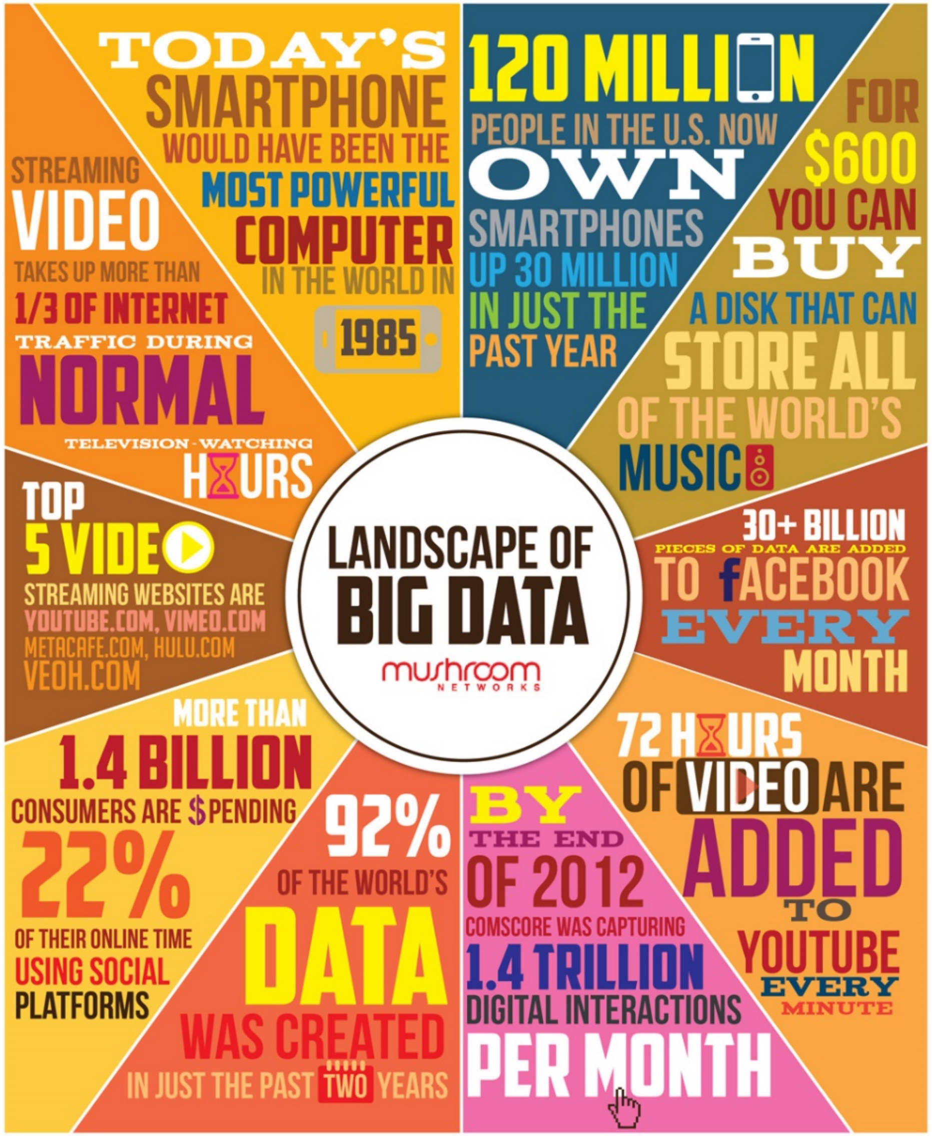
- Some problems always benefit from faster processing
  - Simulation and modeling (climate, earthquakes, airplane design, car design, vehicle traffic patterns, ...)
  - Data mining (big data), transaction processing
  - Web search
  - Social networks
  - Modern computer games
  - Next-generation medicine (DNA sequencing, simulation of drug effects)
  - Business data processing
  - Graphic effects on consumer devices
  - …

Laws of this Universe: π



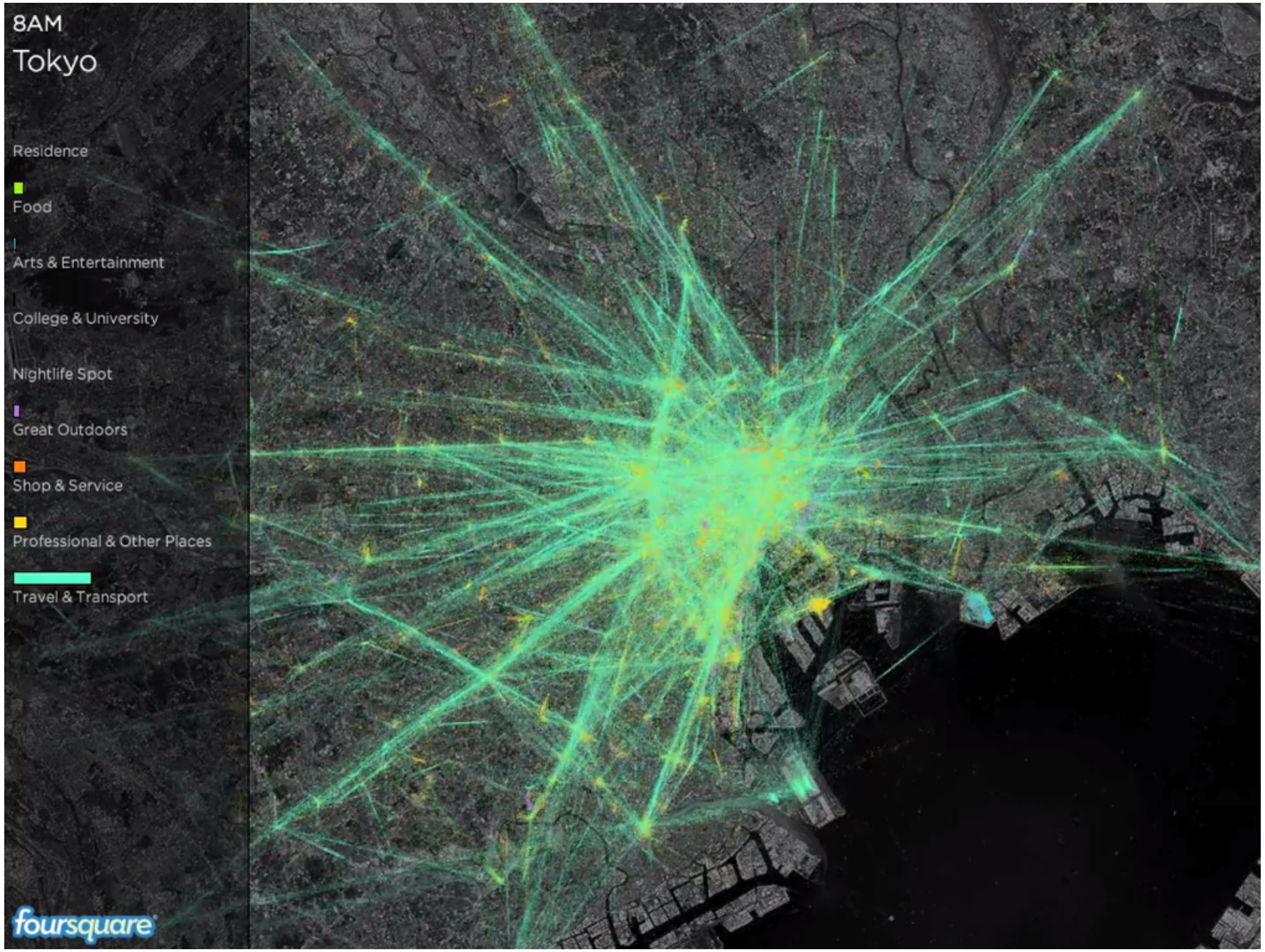
In 2011, pi was computed out to **10,000,000,000,000** decimal places. It only takes 39 digits of pi to draw a circle the size of the universe down to the accuracy of a hydrogen atom.

Big Data


TODAY'S SMARTPHONE WOULD HAVE BEEN THE MOST POWERFUL COMPUTER IN THE WORLD IN 1985
120 MILLION PEOPLE IN THE U.S. NOW OWN SMARTPHONES UP 30 MILLION IN JUST THE PAST YEAR
FOR $600 YOU CAN BUY A DISK THAT CAN STORE ALL OF THE WORLD'S MUSIC
30+ BILLION PIECES OF DATA ARE ADDED TO FACEBOOK EVERY MONTH
72 HOURS OF VIDEO ARE ADDED TO YOUTUBE EVERY MINUTE
BY THE END OF 2012 COMSCORE WAS CAPTURING 1.4 TRILLION DIGITAL INTERACTIONS PER MONTH
92% OF THE WORLD'S DATA WAS CREATED IN JUST THE PAST TWO YEARS
MORE THAN 1.4 BILLION CONSUMERS ARE $PENDING 22% OF THEIR ONLINE TIME USING SOCIAL PLATFORMS
TOP 5 VIDEO STREAMING WEBSITES ARE YOUTUBE.COM, VIMEO.COM METACAFE.COM, HULU.COM VEOH.COM
STREAMING VIDEO TAKES UP MORE THAN 1/3 OF INTERNET TRAFFIC DURING NORMAL TELEVISION-WATCHING HOURS
LANDSCAPE OF BIG DATA
mushroom NETWORKS

http://www.mushroomnetworks.com/infographics/landscape-of-big-data

Cities pulse via Foursquare check-ins



http://flowingdata.com/2013/09/30/cities-pulse-via-foursquare-check-ins/

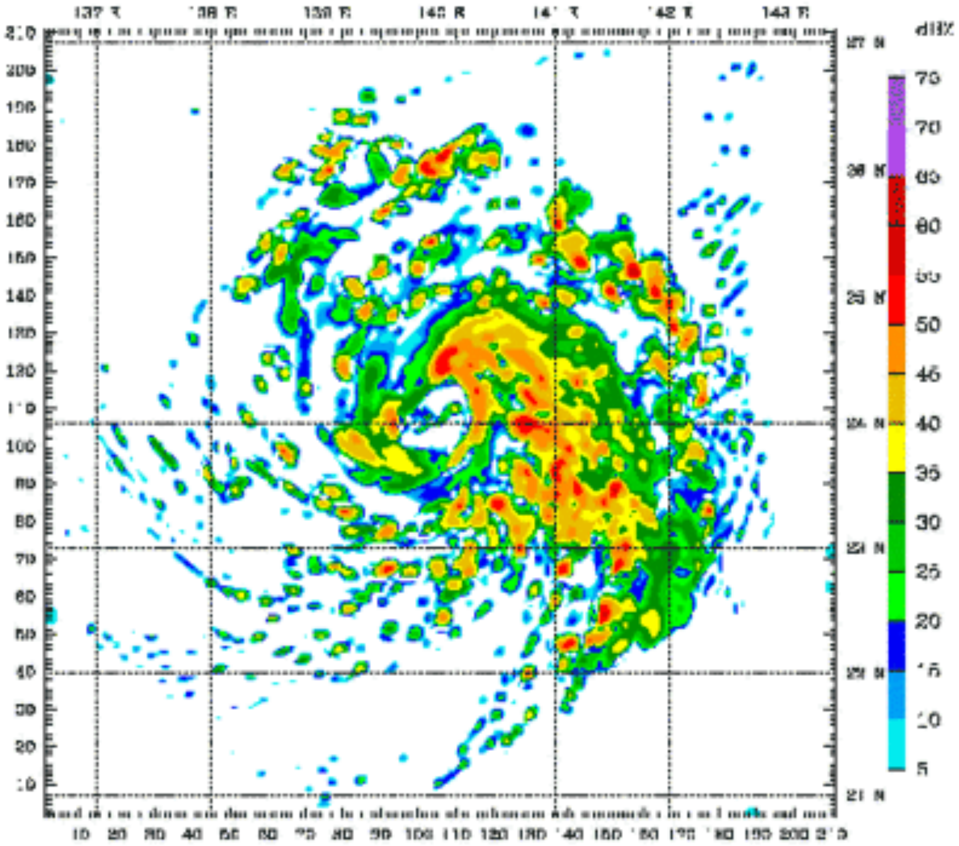
Real-World Simulations

http://www.gimpchat.com/viewtopic.php?f=11&t=5232

"In August 2011, several areas of London experienced episodes of large-scale disorder, comprising looting, rioting and violence. In this article, we present a **mathematical model of the spatial development of the disorder**, which can be used to examine the effect of varying policing arrangements…" [Davies et al.]

Numerical Weather Prediction

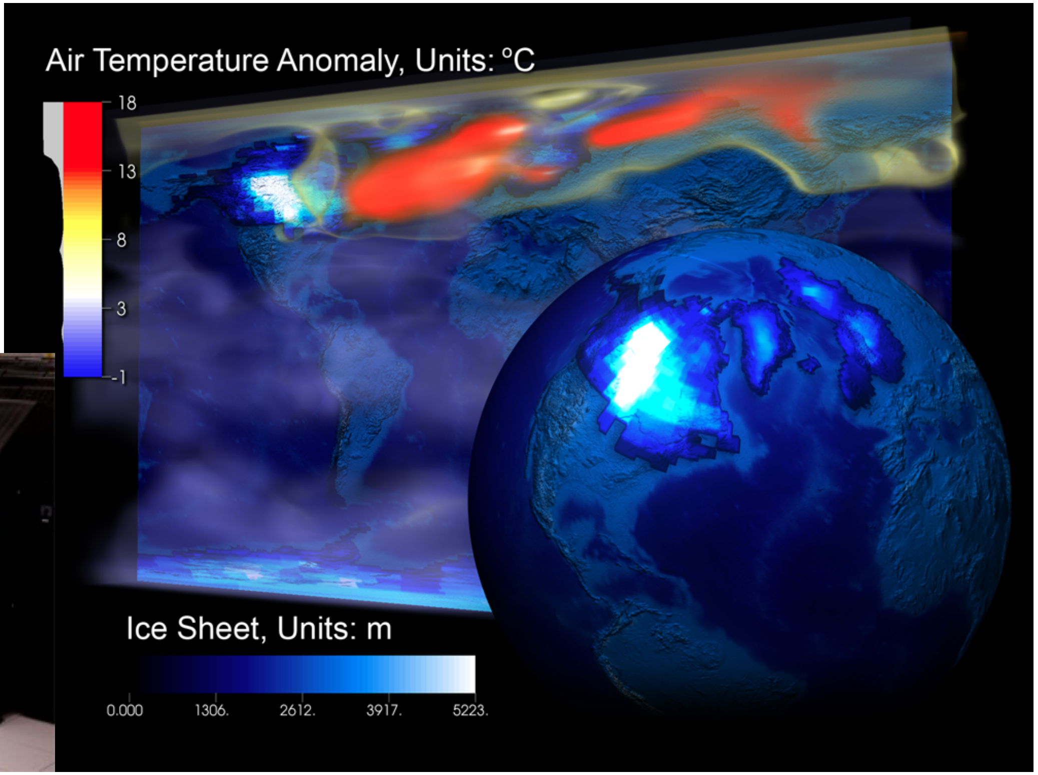
- Calculation of all physical factors driving the atmosphere
- 1959: UK Met Office had state-of-the-art hardware (3000 FLOPS)
- 1980: European Centre for Medium Range Weather Forecasts installed a Cray 1 (250 million FLOPS)
- 2014: New Cray XC30 systems for German weather service with 17.500 cores and 85 TB of main memory
- Today 6-16 days of prediction into the future



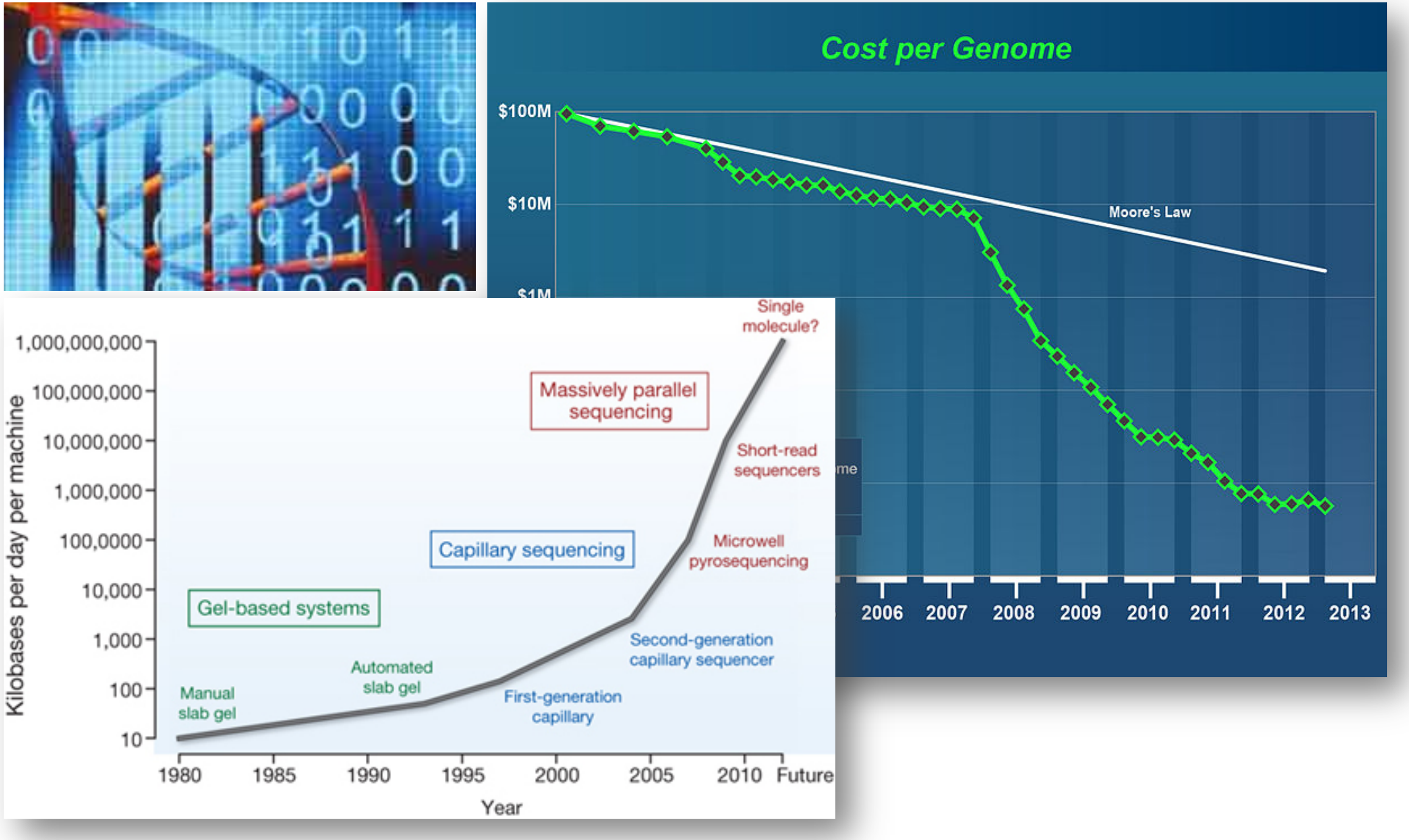
Climate Simulation

- Simulation of abrupt climate change
- From 14.000 years ago to 200 years in the future
- 4 Million processor hours in 3 years on Cray XT („Jaguar“)
  - □ 200 cabinets
  - □ 224.256 cores
  - □ 2.3 Petaflops

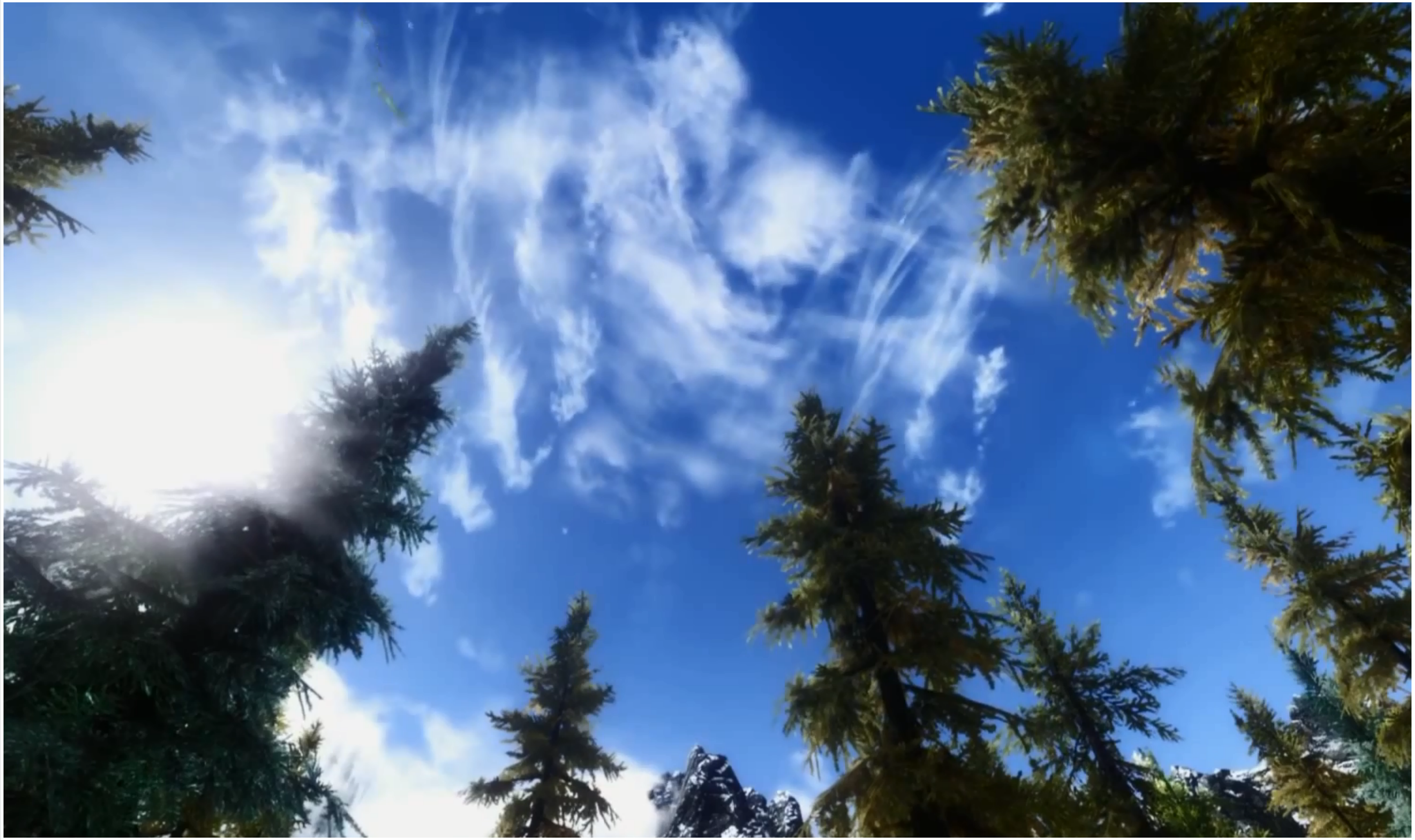
http://www.olcf.ornl.gov



DNA Sequencing

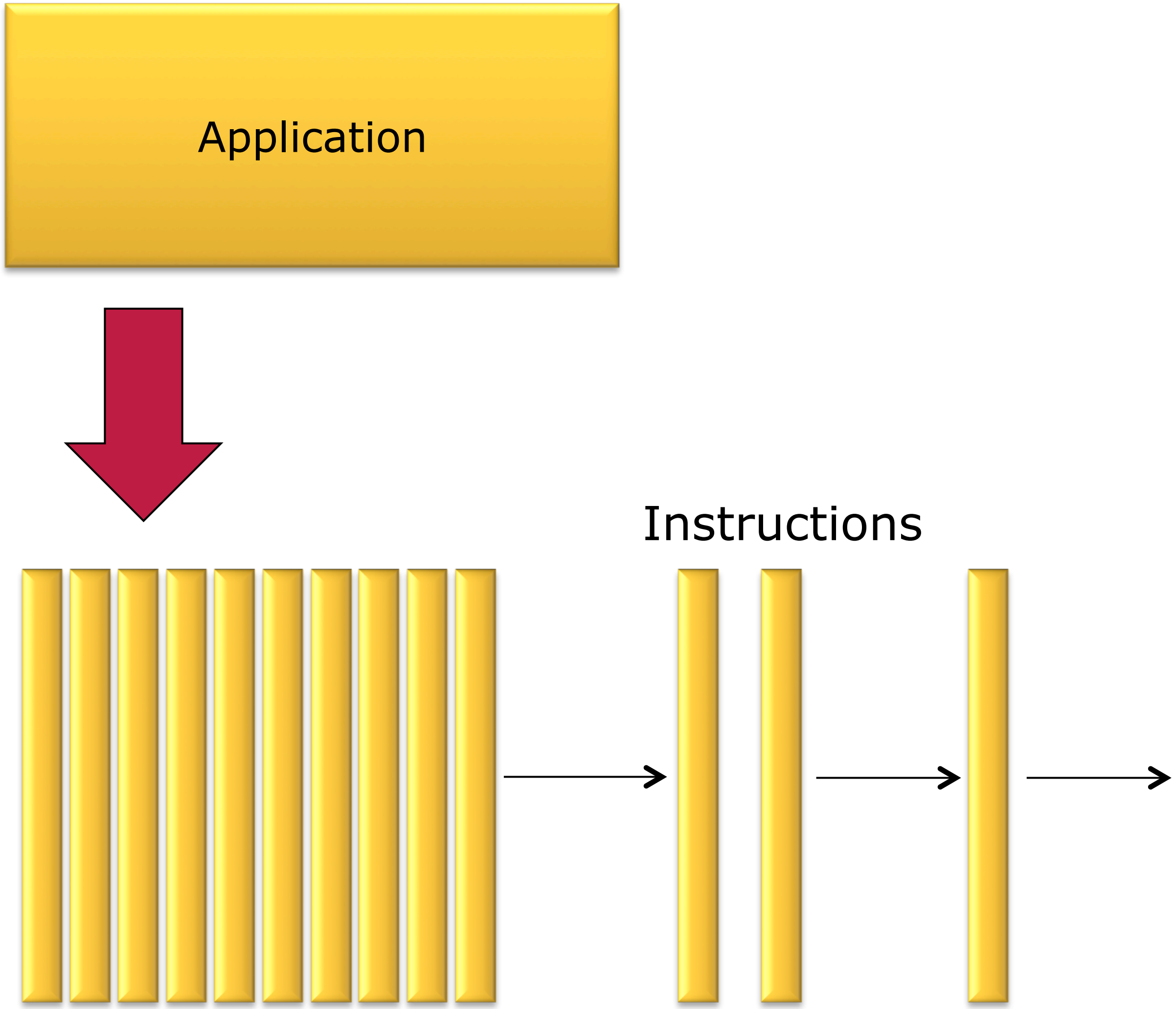


Computer Games



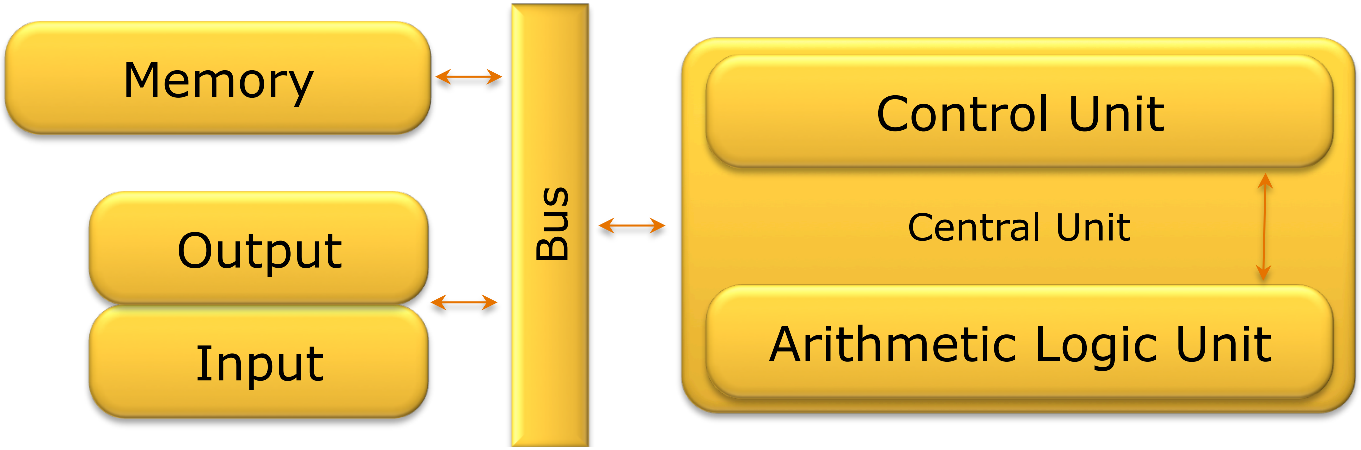
SkyRim with texture mods

Running Applications

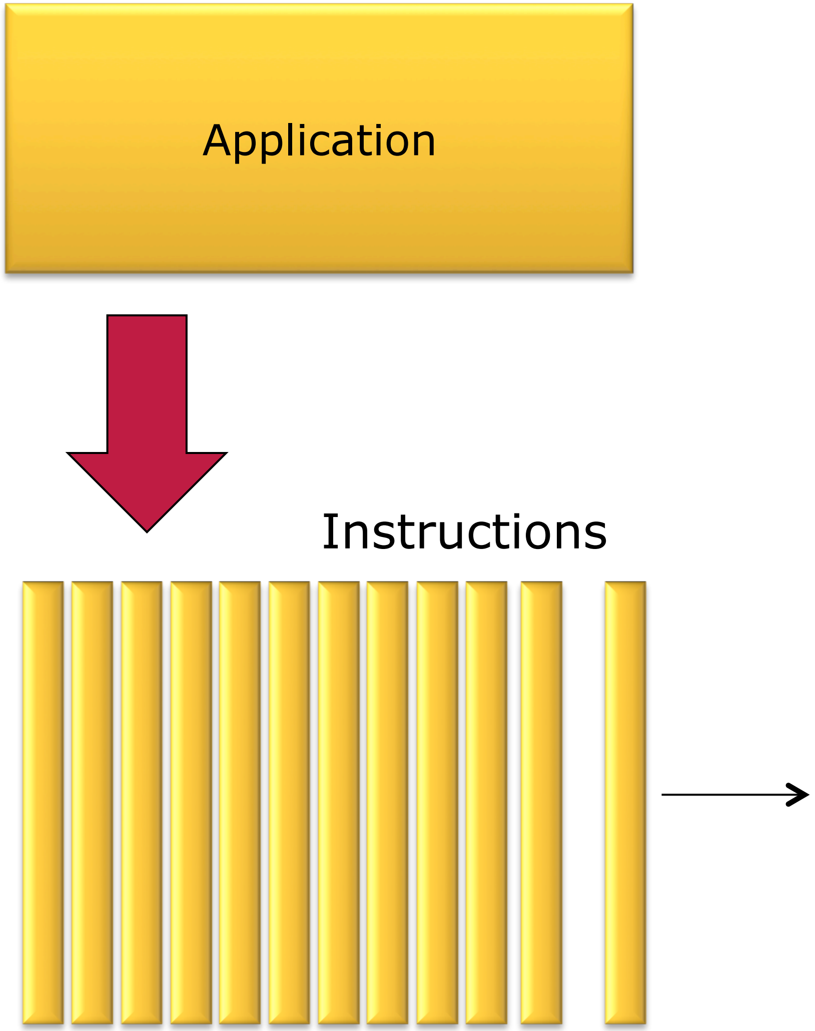


Machine Model

- First computers had fixed programs (electronic calculator)
- **von Neumann architecture** (1945, for EDVAC project)
  - Instruction set for control flows stored in memory
  - Program is treated as data, which allows the exchange of code during runtime and self-modification
  - Introduced the **von Neumann bottleneck**
- CPUs are built from logic gates, which are built from transistors
- Multiple CPUs (SMP) were always possible, but exotic



Three ways of doing anything faster [Pfister]

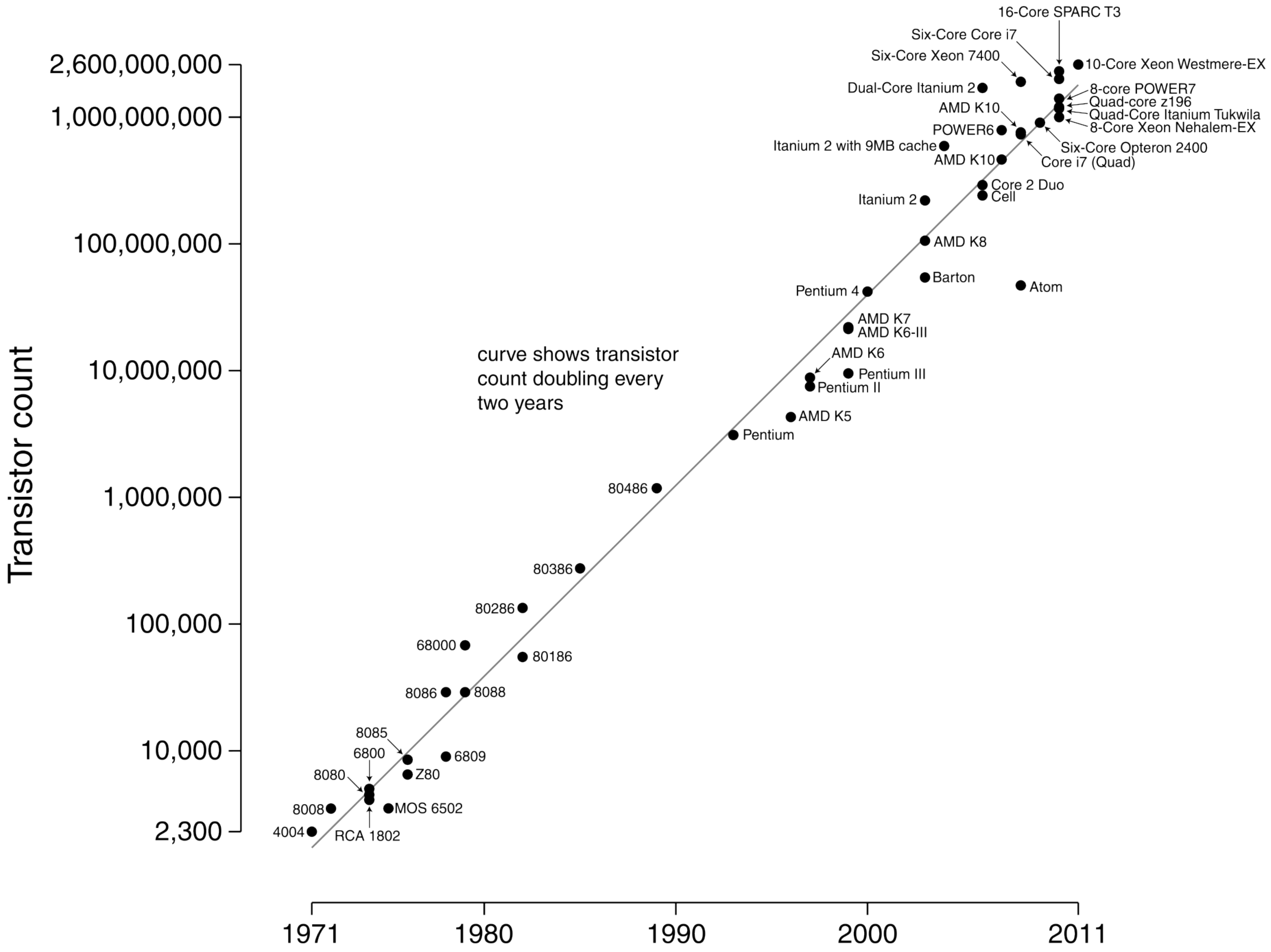


- Work Harder (clock speed)
- Work Smarter (optimization, caching)
- Get Help (parallelization)

Moore’s Law

- *„...the number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every two years. ...“ (Gordon Moore, 1965)*
    - Rule of exponential growth
    - Applied to many IT hardware developments
    - Sometimes misinterpreted as performance indication
    - Meanwhile a self-fulfilling prophecy
    - May still hold for the next 10-20 years

Moore’s Law

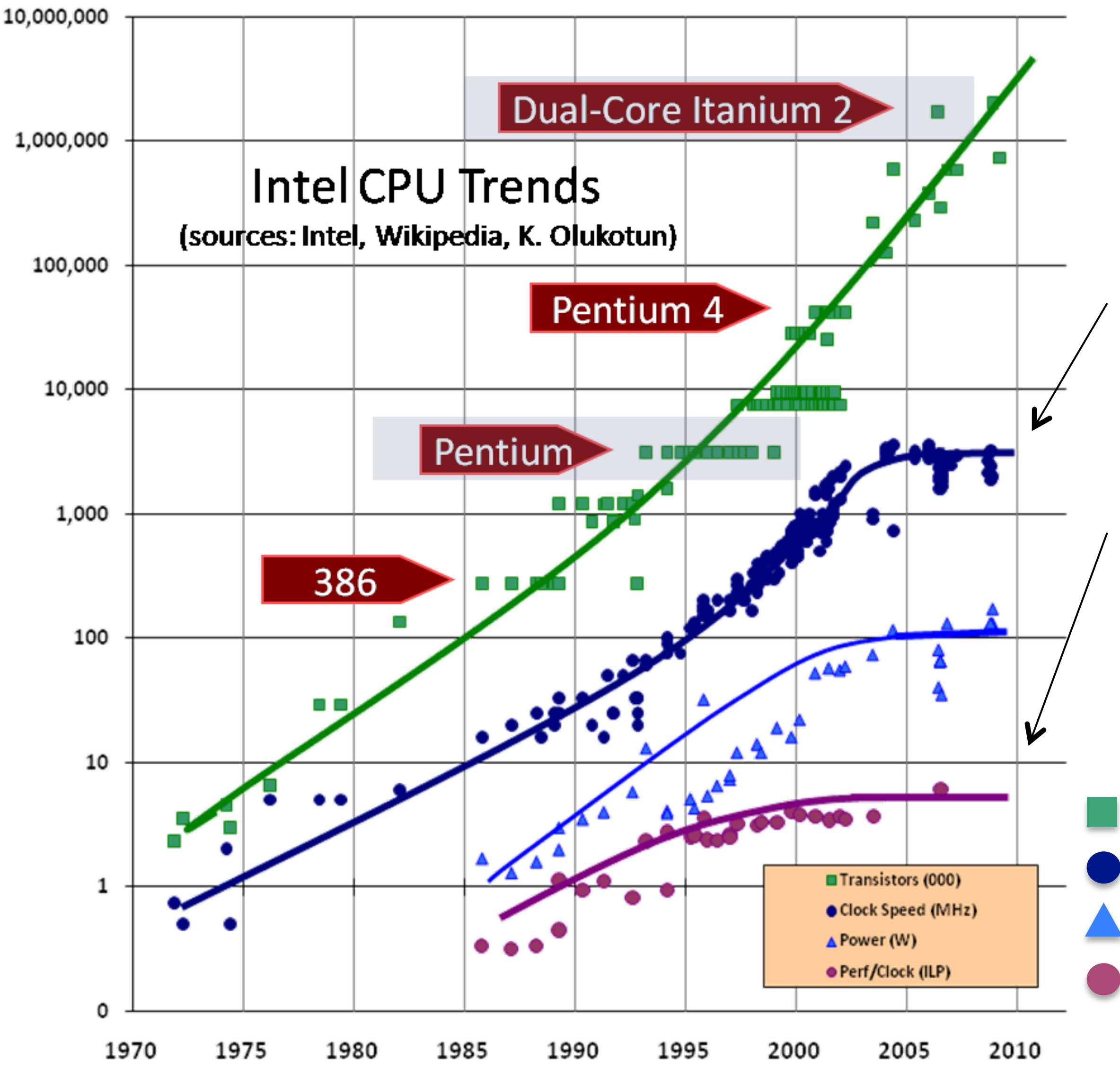


[Wikimedia]

Moore’s Law vs. Software

- Gate’s law: *“The speed of software halves every 18 months.”*
- Wirth’s law: *“Software is getting slower more rapidly than hardware becomes faster.”*
- May’s law: *“Software efficiency halves every 18 months, compensating Moore's Law.”*
- Jevons paradox:
  “Technological progress that increases the efficiency with which a resource is used tends to increase (rather than decrease) the rate of consumption of that resource.”
- Zawinski's Law of Software Envelopment:
  “Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can.”

Processor Speed Development



„Work harder"

„Work smarter"

🟩 Transistors #

● Clock Speed (Mhz)

▲ Power (W)

● Perf/Clock (ILP)

[Herb Sutter, 2009]

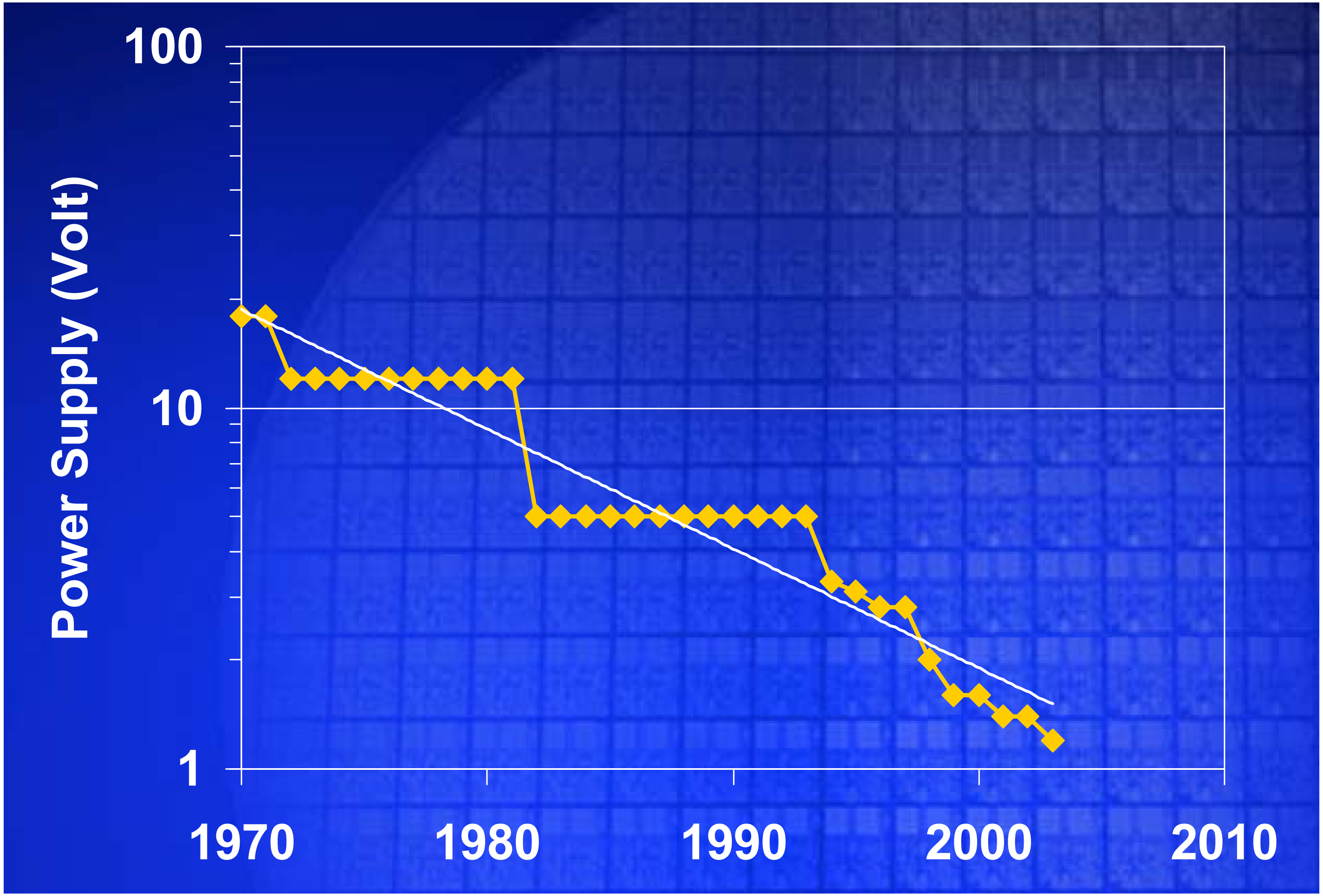
A Physics Problem

- Power: Energy needed per time unit
  - Power density: Watt/mm^2 → Cooling
- **Static power**: Leakage of transistors while being inactive
- **Dynamic power**: Energy needed to switch a gate

Dynamic Power ~
Number of Transistors (N) x Capacity (C) x
Voltage^2 (V^2) x Frequency (F)

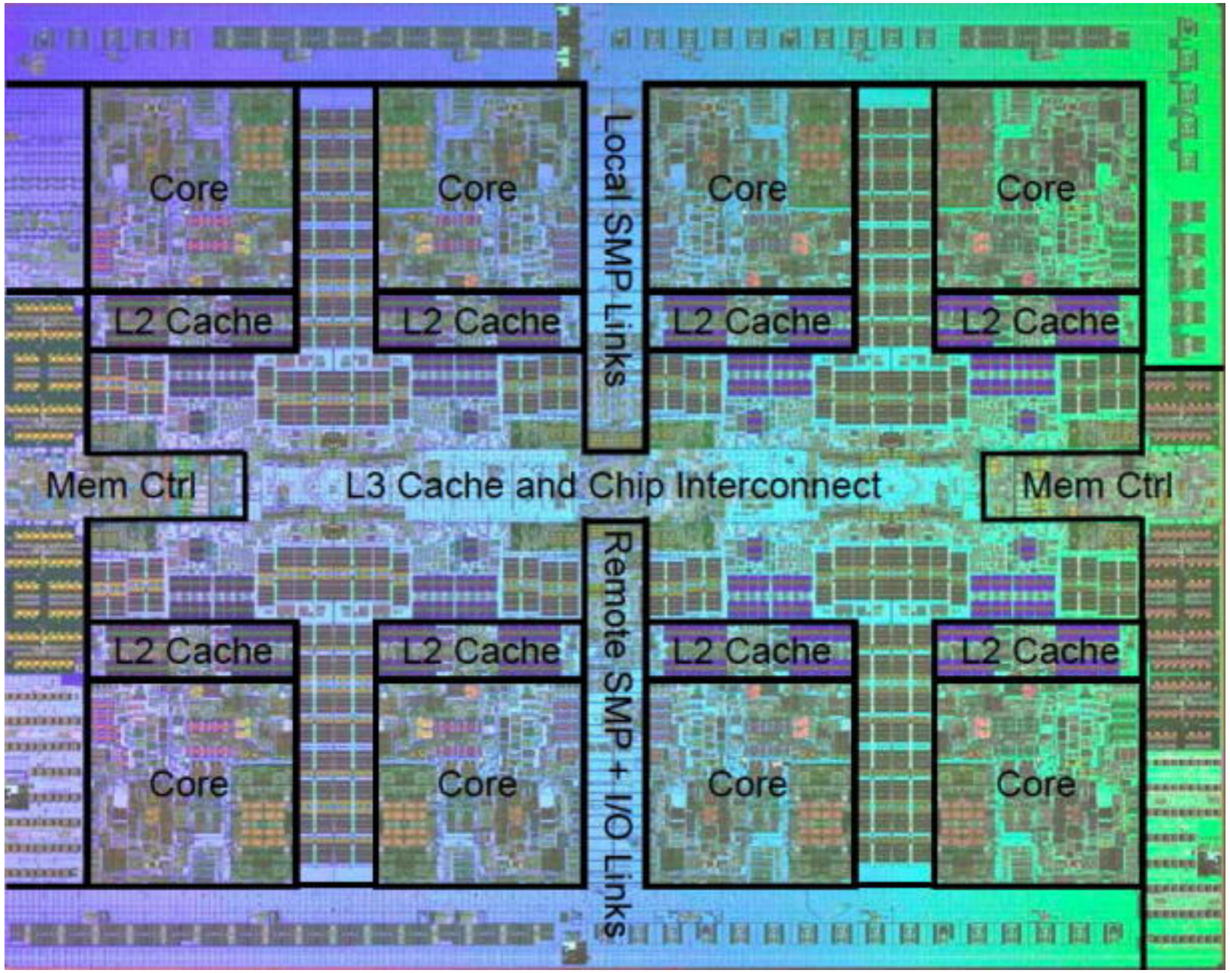
- Moore's law: N goes up exponentially, C goes down with the size
- The trick
  - Bringing down V reduces energy consumption, quadratically
  - Don't use all the N for gates (e.g. caches)
  - Keeps the dynamic power increase moderate
  - We can happily increase F with N for faster computation

Processor Supply Voltage



[Moore, ISSCC]

Transistor Usage

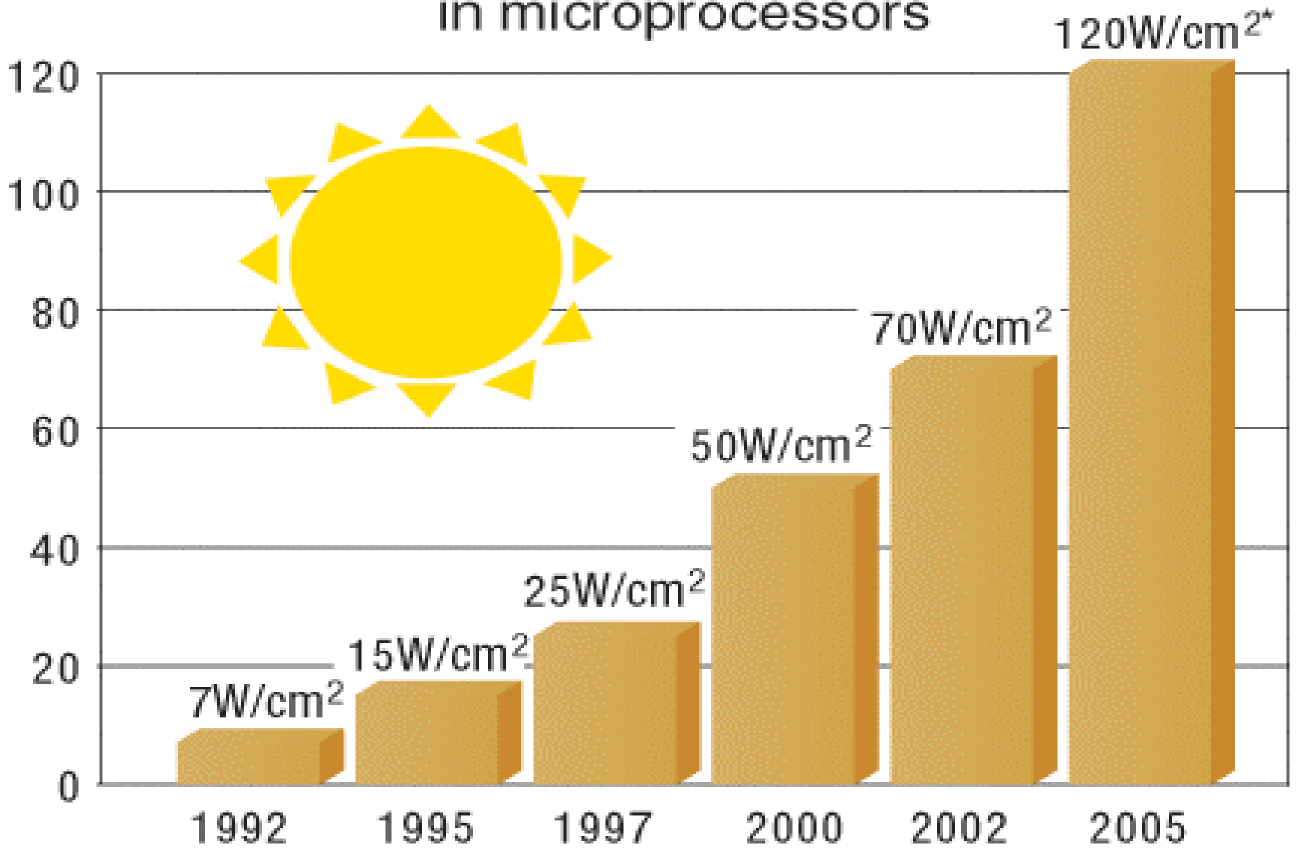


http://arstechnica.com/gadgets/2009/09/ibms-8-core-power7-twice-the-muscle-half-the-transistors/

Power Density

SIZZLING SEMICONDUCTORS

growth of watts per square centimeter in microprocessors



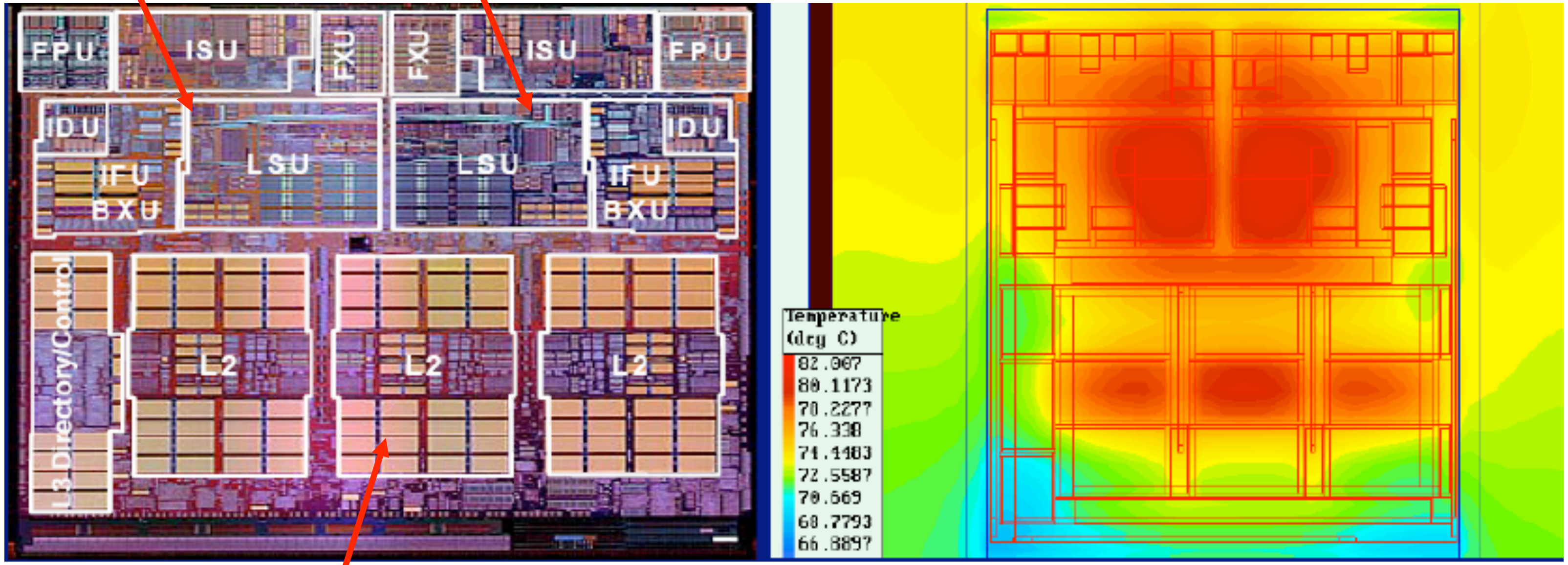
*Could be higher, depends on level of integration.

SOURCE: HEWLETT-PACKARD LABS

Power Density = Temperature

1st CPU 2nd CPU [source: Devgan’05]

Power 4 server chip

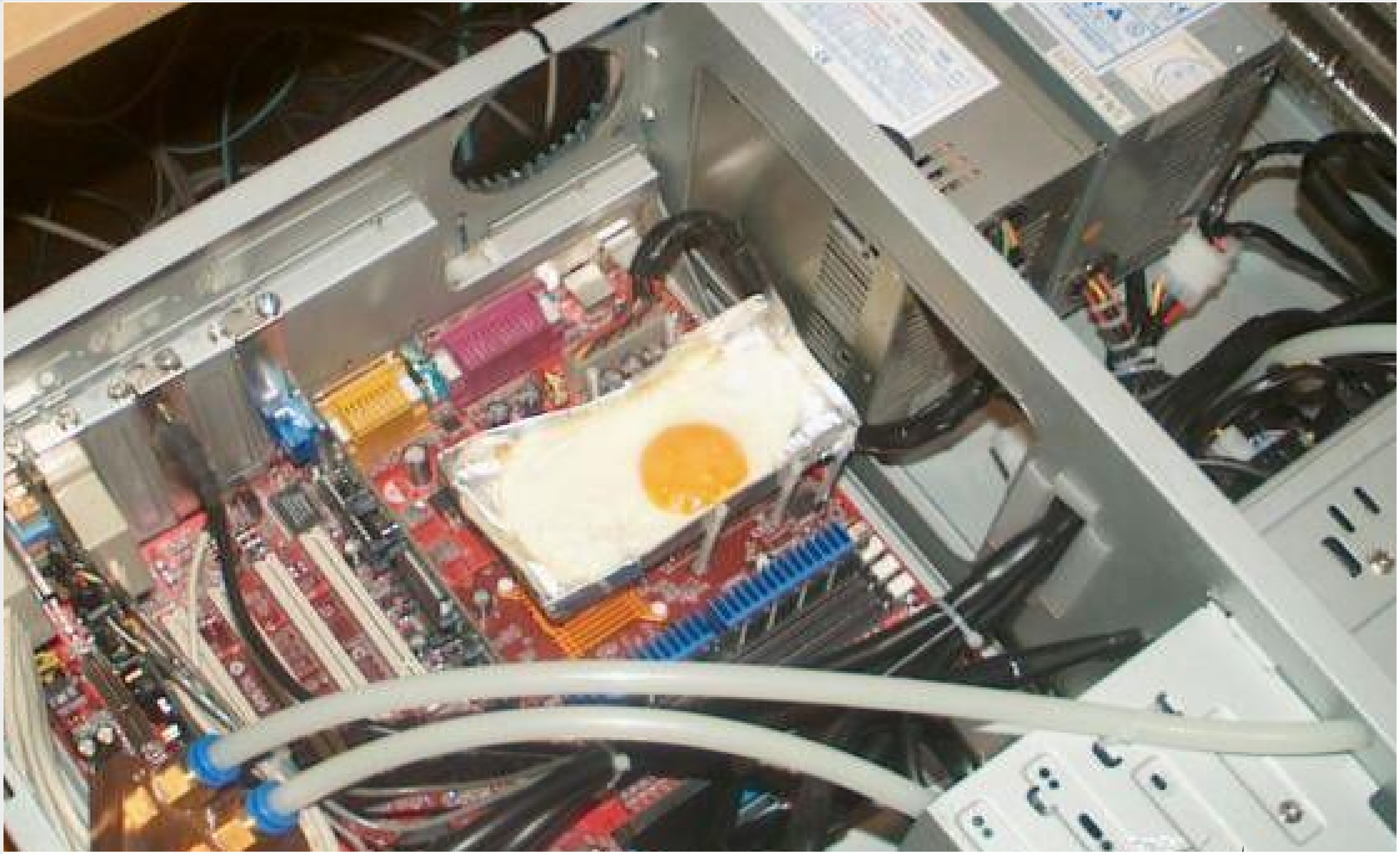


cache

thermal profile during runtime

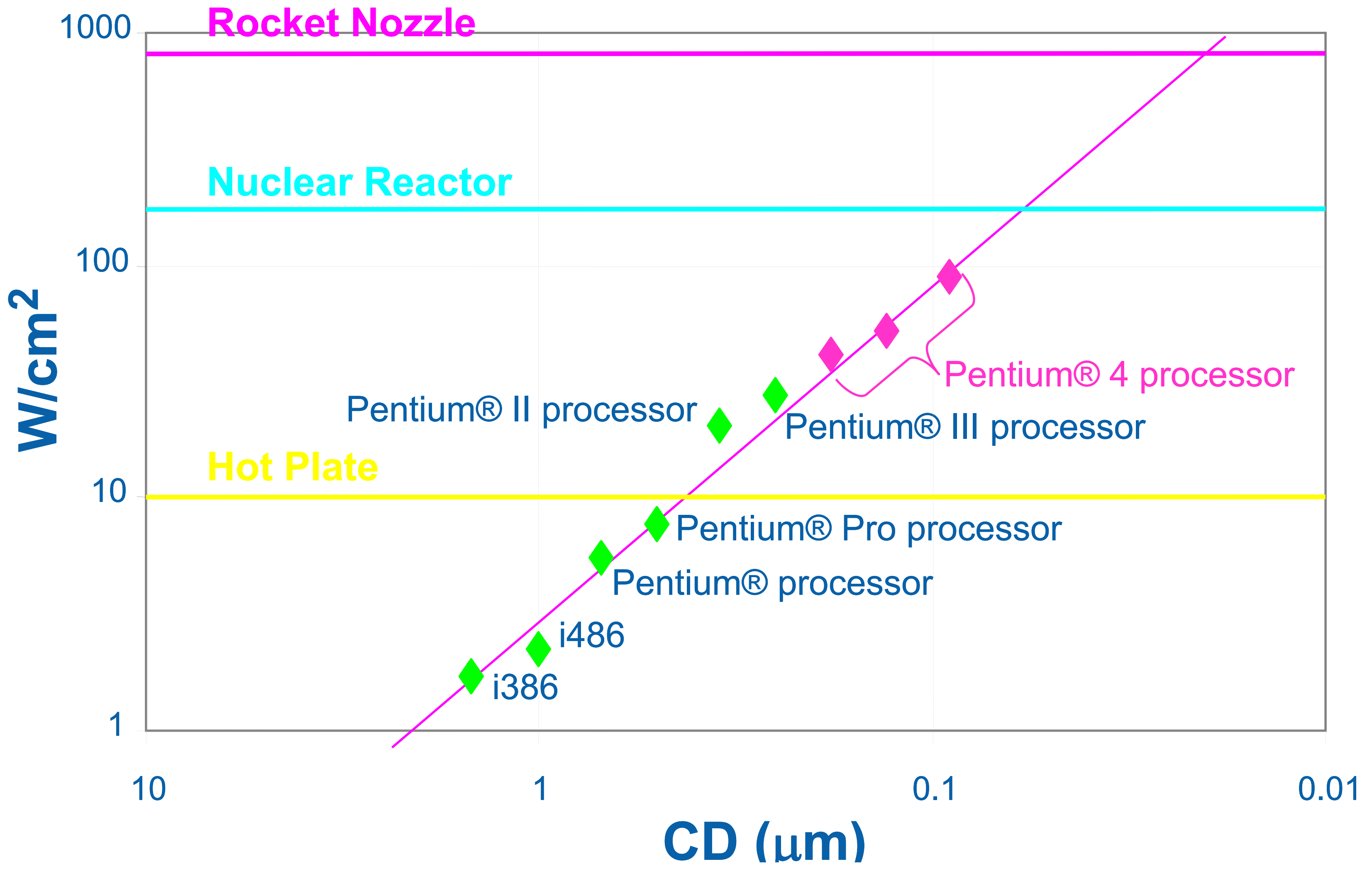
- Higher temperature leads to
  - Increased transistor leakage
  - Decreased transistor speed
  - Higher failure probability

Power Density



[Kevin Skadron, 2007]

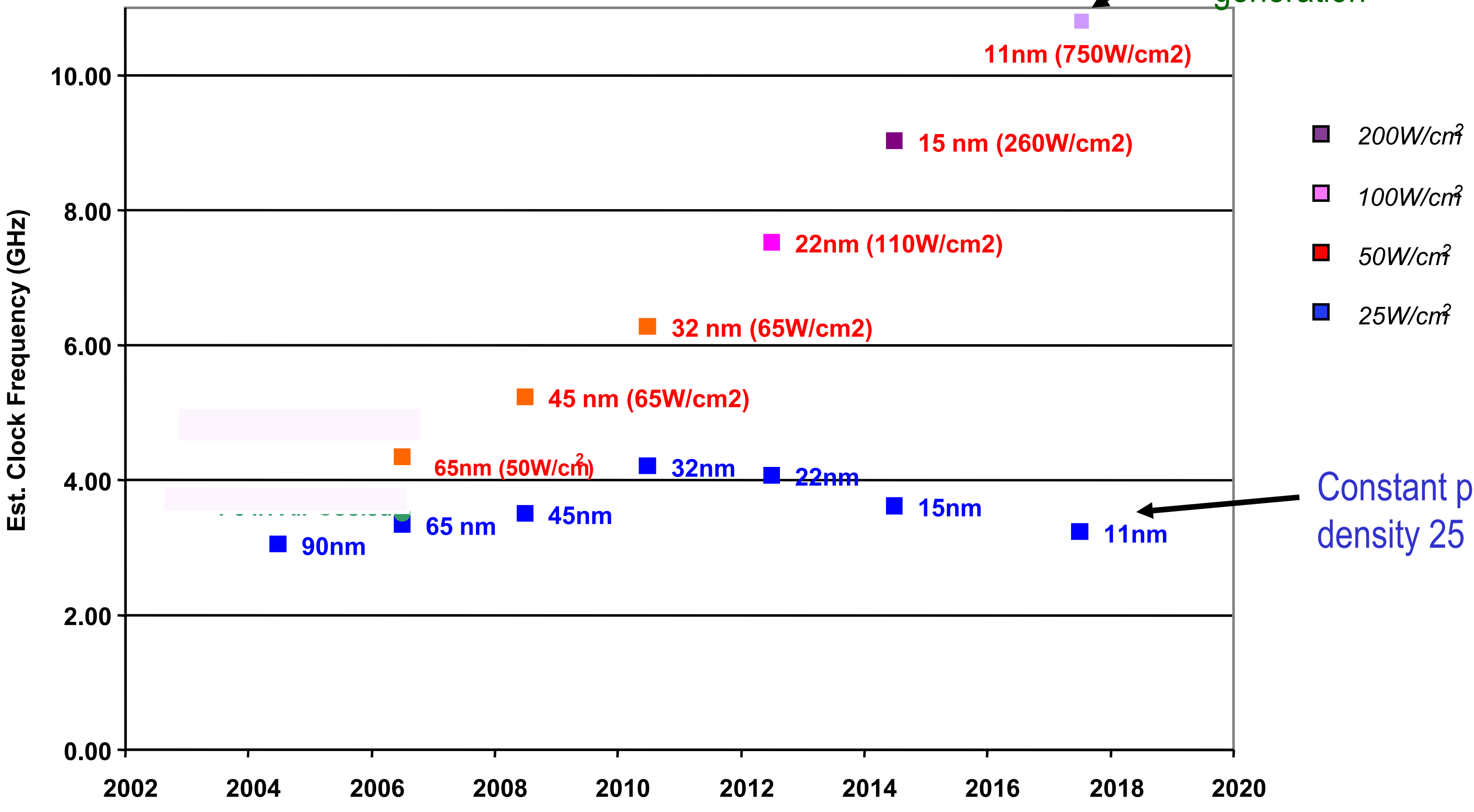
Power Density



[Taylor, 2009]

Power Density Projection

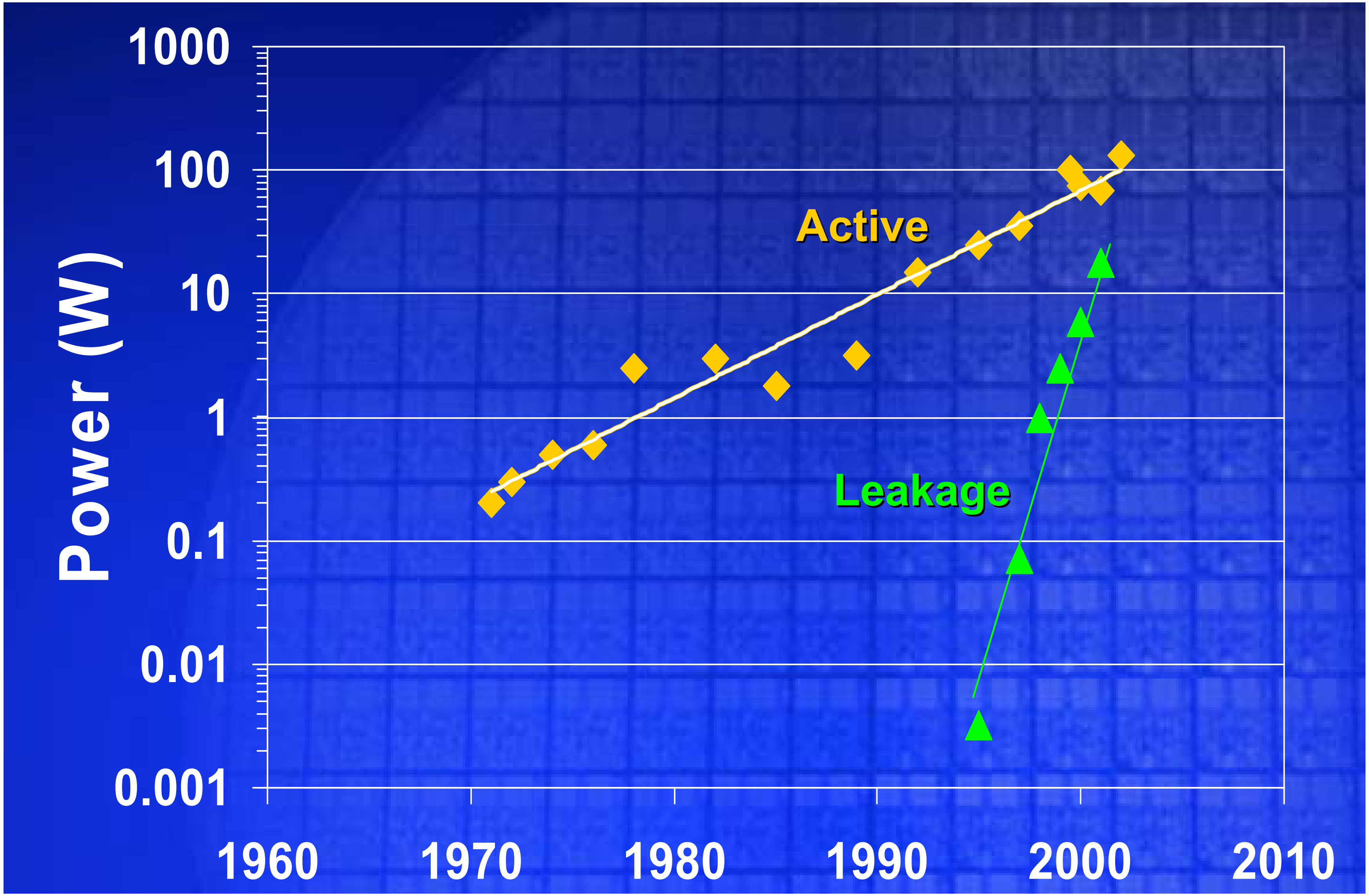
Constant performance improvement of 20% per generation



Constant power density 25 W/cm^2

Source: *D. Frank, C. Tyberg,* IBM Research

Second Problem: Leakage Increase



[www.ieeeghn.org]

A Physics Problem

Dynamic Power = N x C x V^2 x F

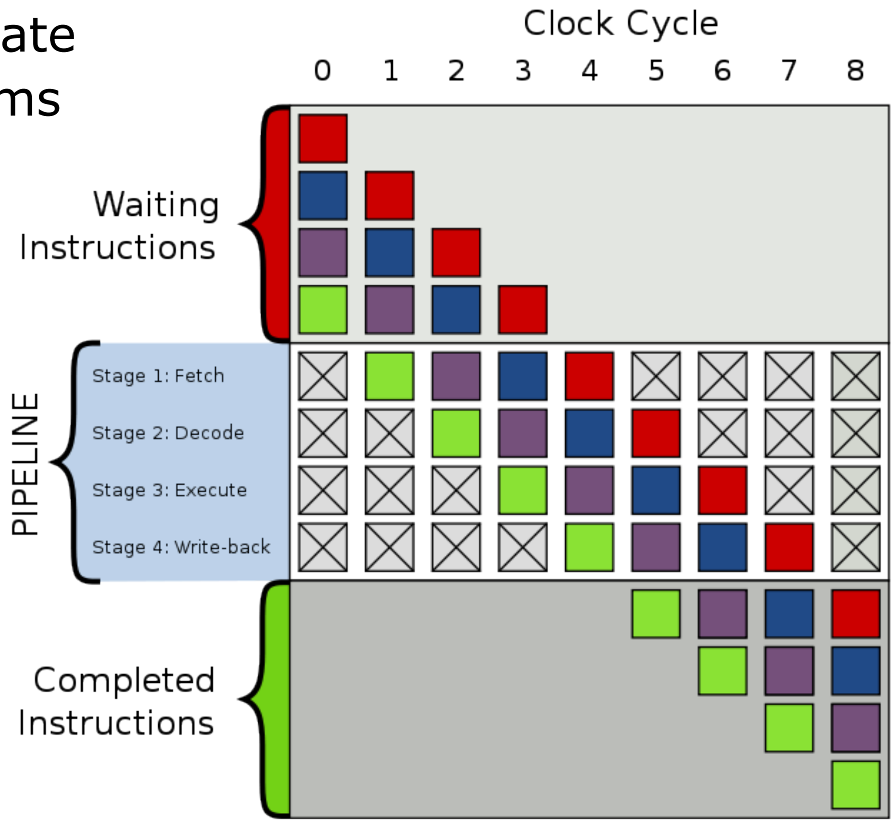
- Even if we would keep F constant
  - N continues to increase exponentially → dynamic power
  - Increasing N sums up to more leakage → static power
- Cooling performance is constant (100-125 Celsius)
  - Static and dynamic power consumption has a limit
- Further reducing V for compensating an additionally increased F
  - Also makes the transistors slower
  - We can’t do that endlessly, 0V is the limit
  - Strange physical effects
- Increasing the frequency is no longer possible
  → **“Power Wall”**
- Ok, so let’s use the additional N for smarter processors

Instruction Level Parallelism

- Increasing transistor count was also used for more gate logic in **instruction level parallelism (ILP)**
  - **Instruction pipelining**
    - Overlapped execution of serial instructions
  - **Superscalar execution**
    - Multiple execution units are used in parallel
  - **Out-of-order execution**
    - Reorder instructions that have no data dependency
  - **Speculative execution**
    - Control flow speculation, memory dependence prediction, branch prediction
- Today’s processors are packed with ILP logic

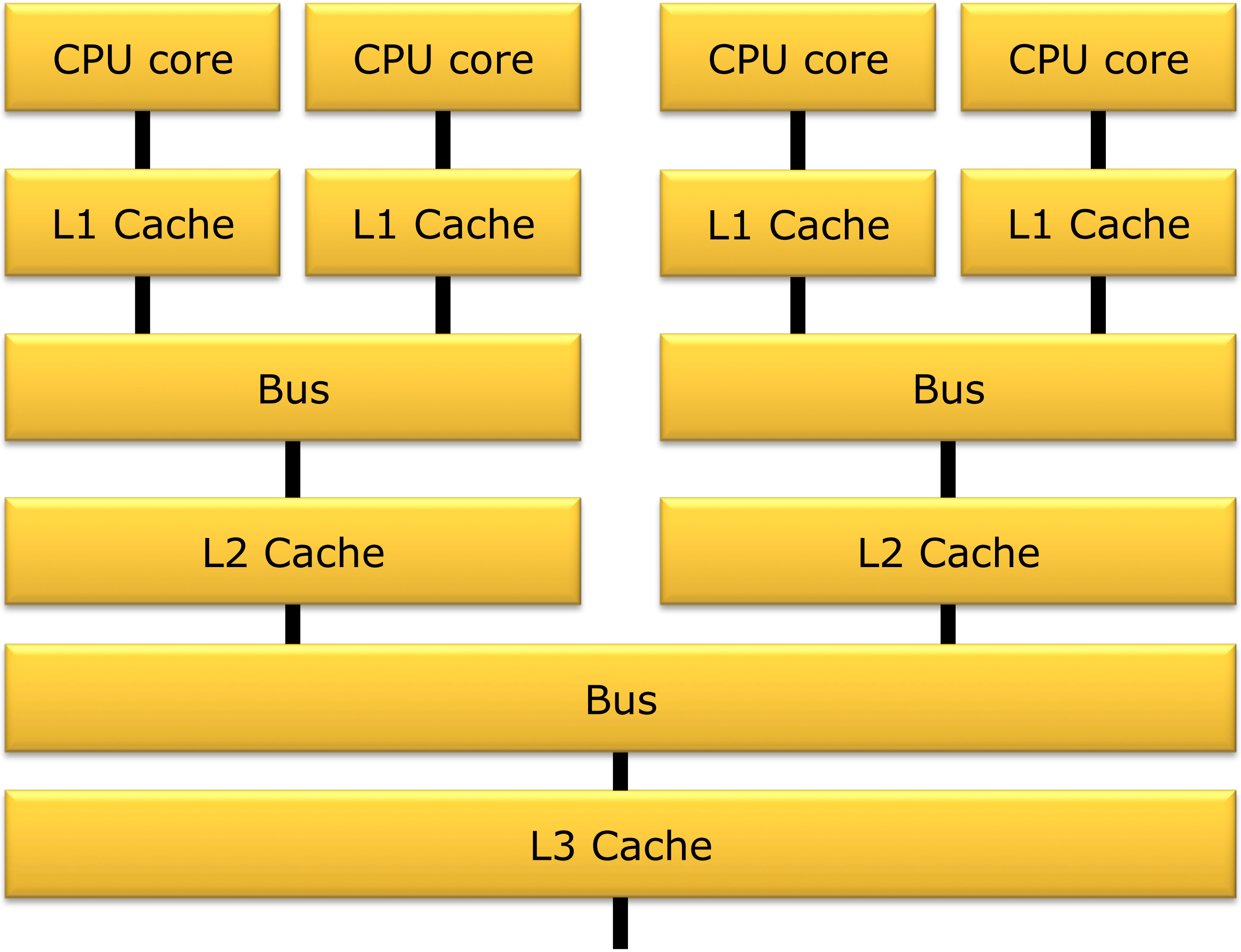
The ILP Wall

- No longer cost-effective to dedicate new transistors to ILP mechanisms
- Deeper pipelines make the power problem worse
- High ILP complexity effectively reduces the processing speed for a given frequency (e.g. mispredictions)
- More aggressive ILP technologies too risky for products due to unknown real-world workloads
- → **"ILP wall"**
- Ok, so let's use the additional N for more caches

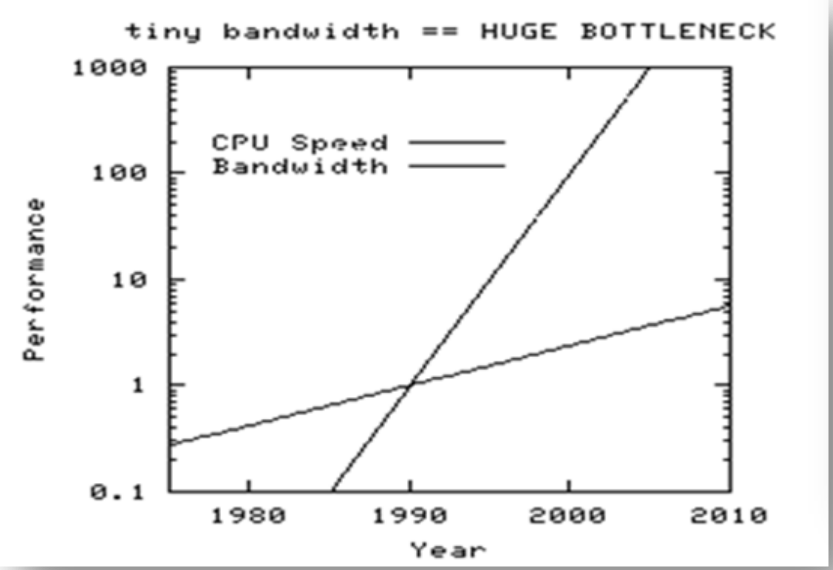


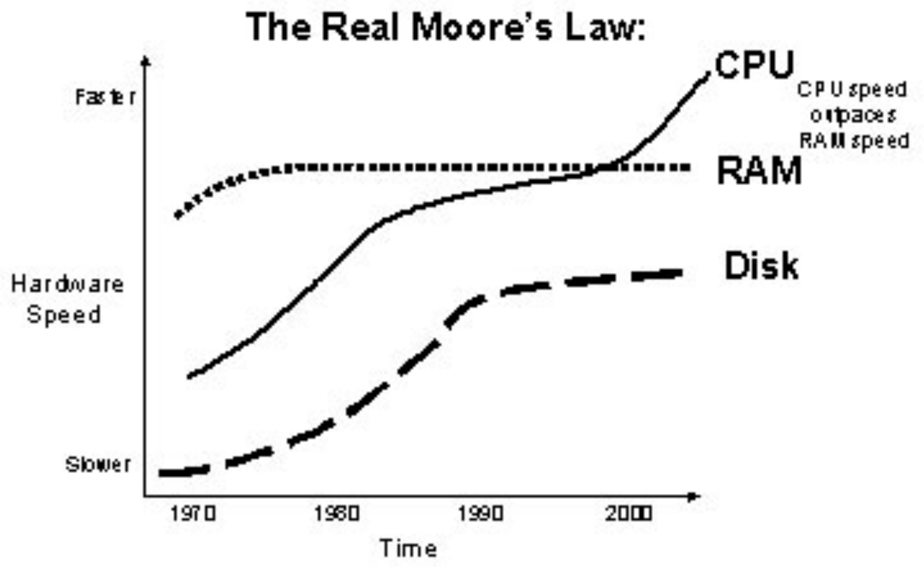
[Wikipedia]

Memory Hierarchy



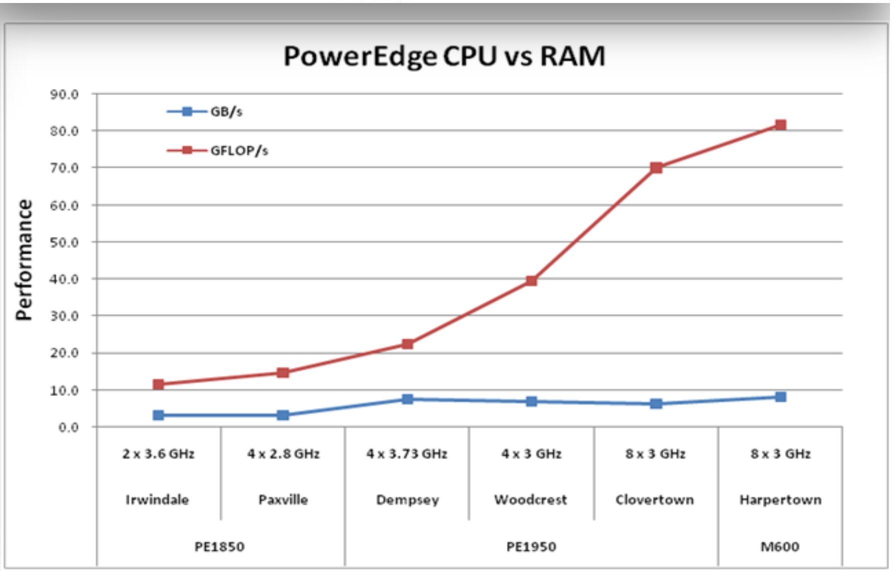
Memory Wall





http://www.dba-oracle.com/oracle_tips_hardware_oracle_performance.htm

http://en.community.dell.com/techcenter/high-performance-computing/w/wiki/2284.aspx

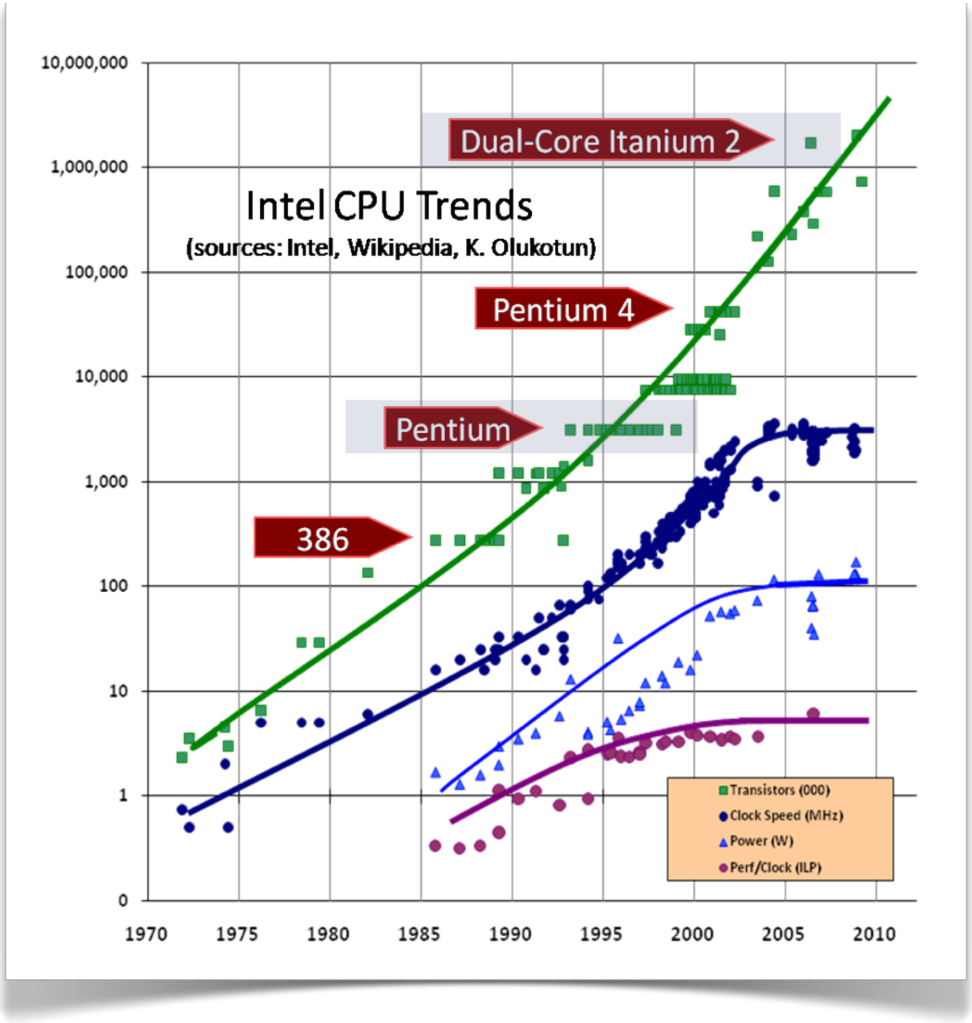


Memory Wall

- Sandia National Labs investigated the speedup achievable by increasing parallelism (ILP, multiple processors) in 2009
- Example: Number of clerks behind a supermarket counter
  - Two clerks can serve more customers than one
  - 4 ? 8 ? 16 ? 32 ? 64 ? ... 1000 ?
- The problem: Shared memory is ‚shared‛
  - Memory bandwidth
    - Memory transfer speed is limited by the power wall
    - Memory transfer size is limited by the power wall
    - Putting memory into the processor is too costly
  - Bus contention
- Another problem: Memory need kept the pace of CPU speedup
- → **"Memory wall"**

The Free Lunch Is Over

- Clock speed curve flattened in 2003
  - Heat
  - Power consumption
  - Leakage
- 2-3 GHz since 2001 (!)
- Speeding up the serial instruction execution through clock speed improvements no longer works
- We stumbled into the **Many-Core Era**



[Herb Sutter, 2009]

Conventional Wisdoms Replaced

Old Wisdom	New Wisdom
Power is free, transistors are expensive	„Power wall“
Only dynamic power counts	Static leakage makes 40% of power
Multiply is slow, load-and-store is fast	„Memory wall“
Instruction-level parallelism gets constantly better via compilers and architectures	„ILP wall“
Parallelization is not worth the effort, wait for the faster uniprocessor	Performance doubling might now take 5 years due to physical limits
Processor performance improvement by increased clock frequency	Processor performance improvement by increased parallelism

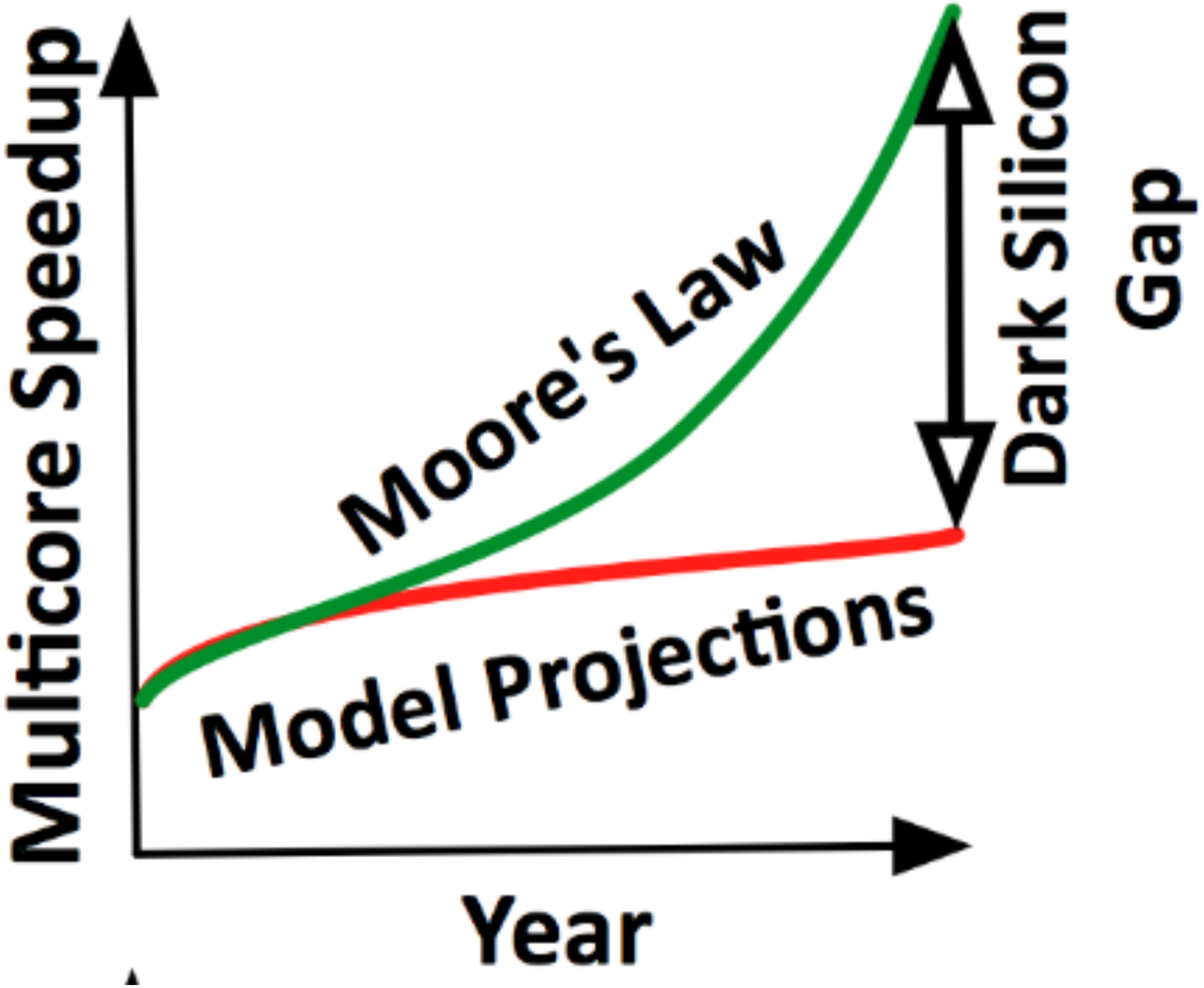
[Asanovic et al., 2006]

Memory Hierarchy

(C) Chevance, approx. values in 2005

Technology	Access Time	Human Scale	Capacity	Price
Processor Register	100 ps	0.1 s	64x64 Bits	part of CPU
Processor Cache	L1: ~1 ns L2-L3: 4-16 ms	16 s	kB - MB	part of CPU
RAM	~150 ns	~ 25 min	>= 1 GB	~0.1 $/MB
Disk	~6 ms	~700 days	> 70 GB / disk	~0.005 $/MB
Tape Robot	~10 s	~3200 years	~100 GB / tape	<0.001 $/MB

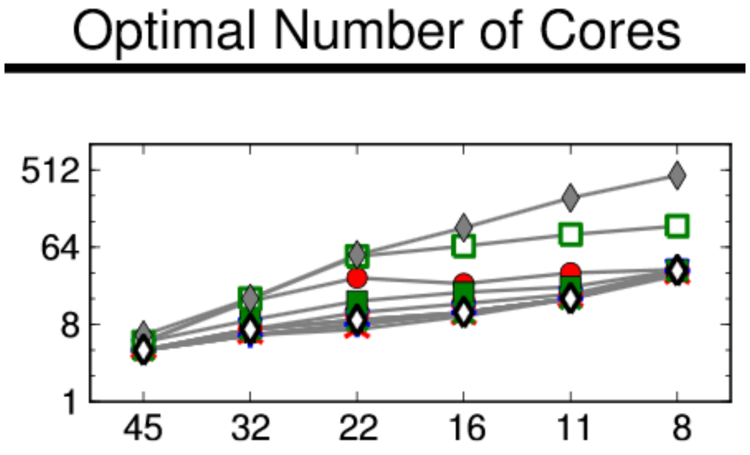
Dark Silicon = Power Wall 2.0



Power Wall 2.0

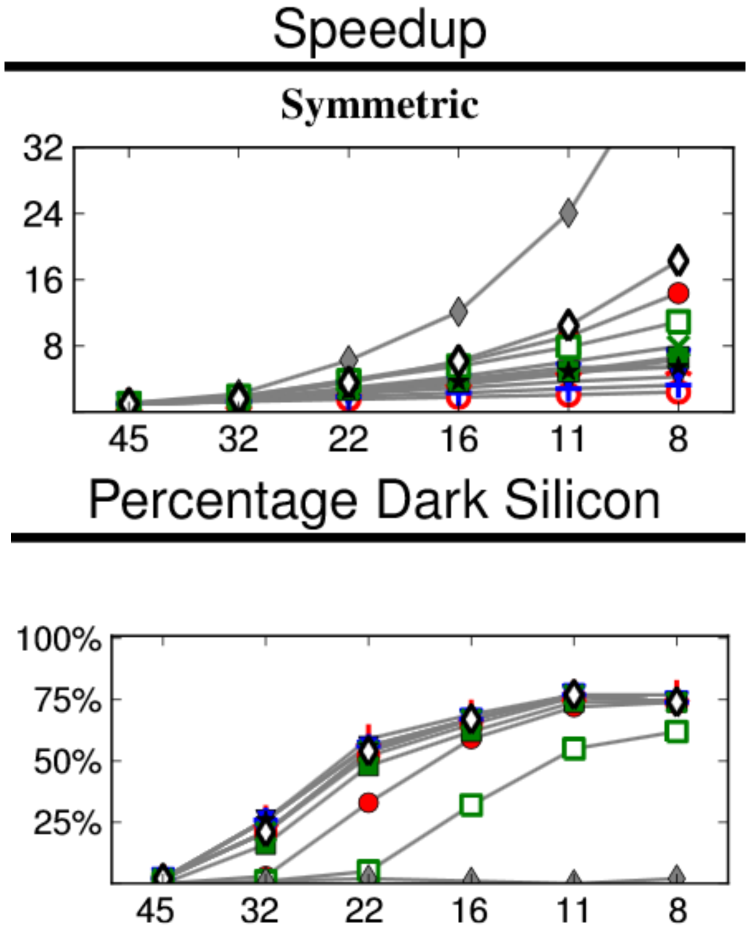
- Power consumption increases with Moore‘s law, even under constant frequencies
- Cooling is a constant factor
  - Maximum temperature of 100-125 C
  - Hot spots make it worse
- Next-generation processors need to use less power
  - Lower the frequencies
  - Dynamic frequencies scaling (see latest Intel products)
  - Minimize ‚power per bit of I/O‘ [Skadron 2007]
  - Better cache locality, stop moving stuff around
  - Start to use specialized co-processors and accelerators

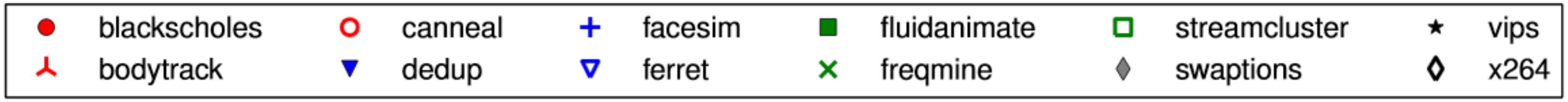
Power Wall 2.0 = Dark Silicon



"Dark Silicon and the End of Multicore Scaling"

by Hadi Esmaeilzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, Doug Burger

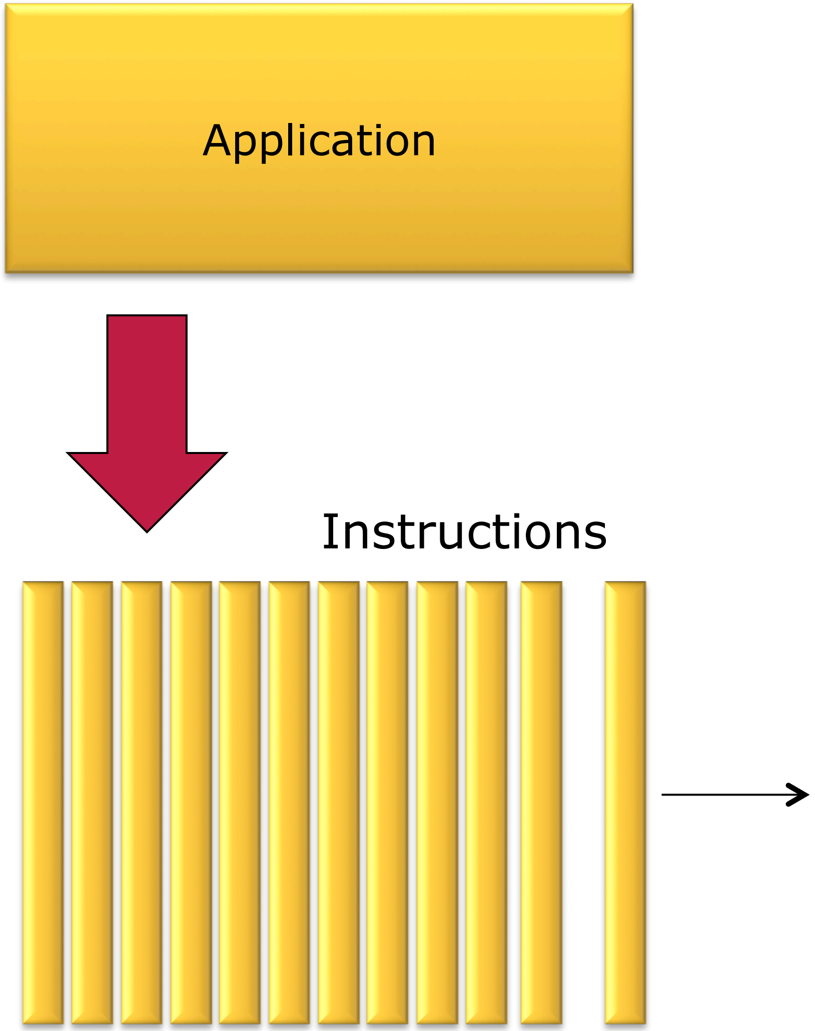




The Situation

- Hardware people
  - Number of transistors N is still increasing
  - Building larger caches no longer helps (memory wall)
  - ILP is out of options (ILP wall)
  - Voltage / power consumption is at the limit (power wall)
    - Some help with dynamic scaling approaches
  - Frequency is stalled (power wall)
  - Only possible offer is to use increasing N for more cores
- For faster software in the future ...
  - Speedup must come from the utilization of an increasing core count, since F is now fixed
  - Software must participate in the power wall handling, to keep F fixed
  - Software must tackle the memory wall

Three ways of doing anything faster [Pfister]



- Work Harder (clock speed)
- Work Smarter (optimization, caching)
- **Get Help (parallelization)**

Getting Help

- Parallelization not only in computer science
  - □ Building construction, car manufacturing, large companies
- The basic idea is easy to understand
- Meanwhile tons of options for parallel processing
  - □ Languages, execution environments, patterns
- Parallelism is a hardware property that must be exploited by software
  - □ *„A parallel computer is a set of processors that are able to work cooperatively to solve a computational problem.“* (Foster 1995)