There is no more Free Lunch

Luca Tornatore - I.N.A.F.

“**Foundation of HPC - basic”** **course**

DATA SCIENCE &
SCIENTIFIC COMPUTING
2022-2023 @ Università di Trieste

The end of “Free lunch” era

Why your life would have been easier 30 years ago and why it is not so

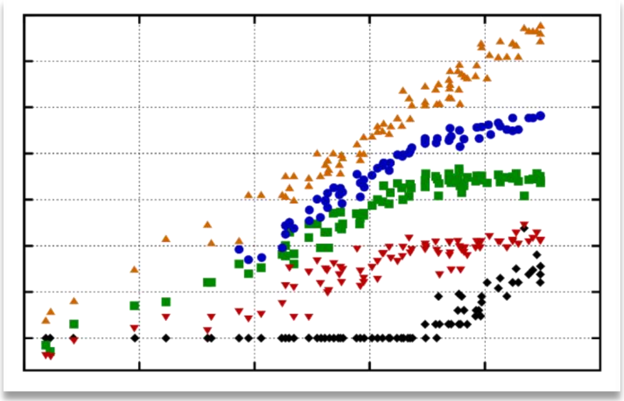
The deep roots of the shift in computer architecture

The “free lunch”

From ‘60s to mid of ‘90s the performance gain of a software was due essentially to the **constant increase** of:

1. the **clock** speed of the CPUs
2. the “**capability**”of CPUs
   (the available instruction set)

The “Free lunch”(*)



The “ *Moore’s law* ” predicted an exponential growth of transistor density on chips:

every 18 months the density of transistors will double *at the same manufacturing cost* (*).

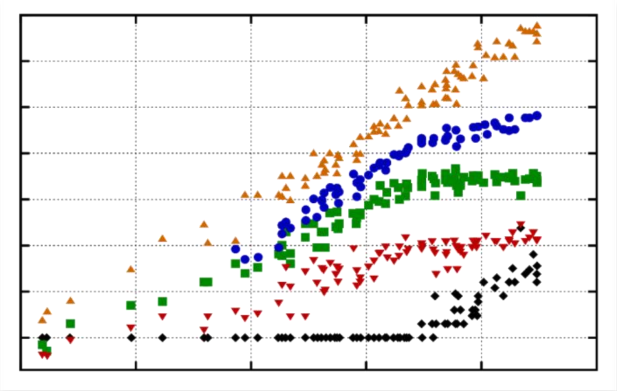
meaning that every 18 months you could buy a commodity CPUs with 2× logics than the previous generation, at the same cost

(*) there have been even faster growths in other fields, for instance in data storage density

(*) expression taken from an article by H. Sutter in *Dr. Dobb’s Journal*, 2005

The “Free lunch”

“ *Moore’s law* ” predicted exponential growth of transistor density on chips:

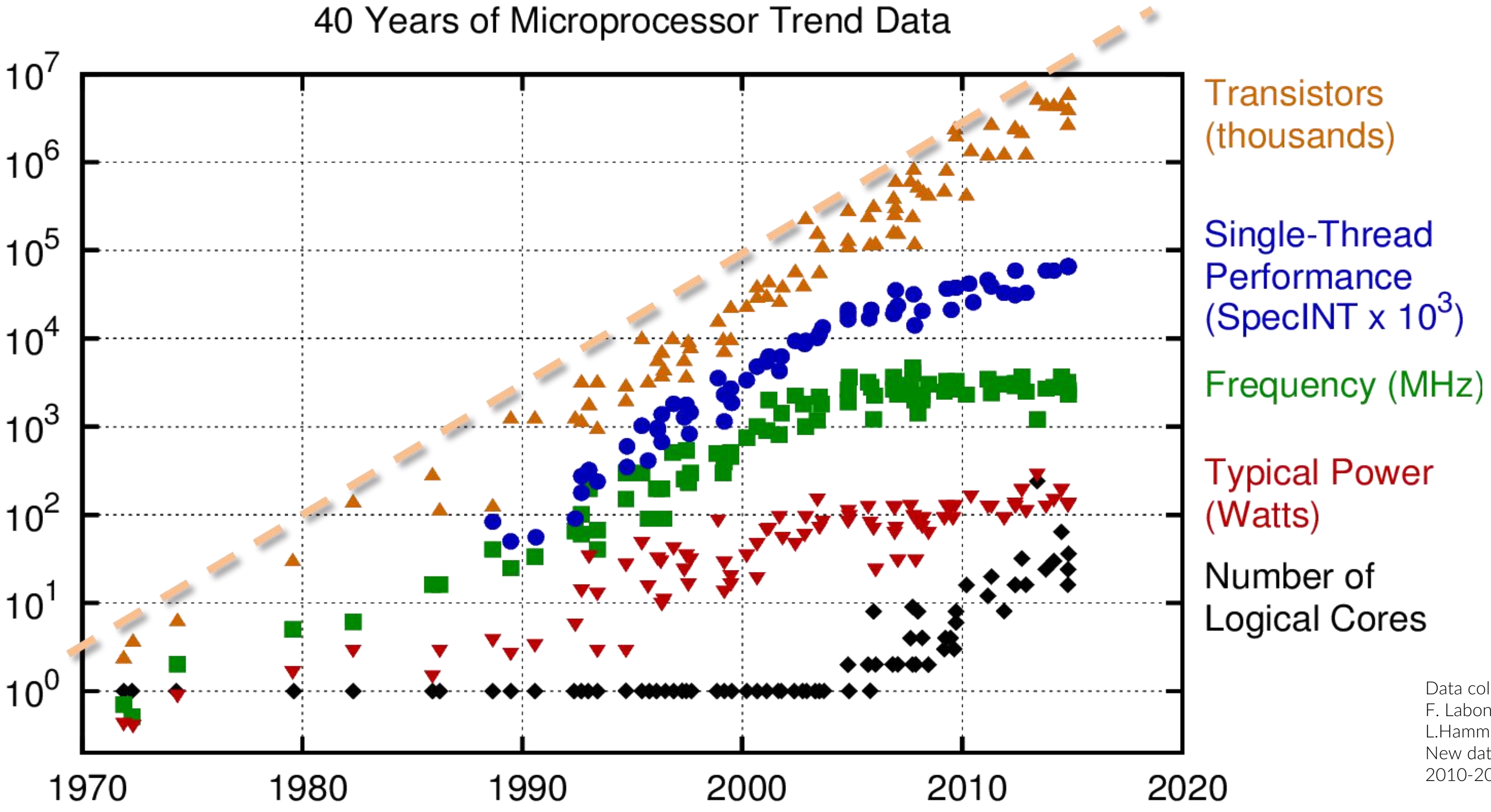


every 18 months the density of transistors will double *at the same manufacturing cost* (*).

Having twice the transistors enhances the CPU capabilities, having it faster means that the same operations are executed in less time, without any effort on the programmers’ side.

That’s why it has been called “free lunch”.
However, all exponential growths get to their end..

Moore’s law, is it over ?



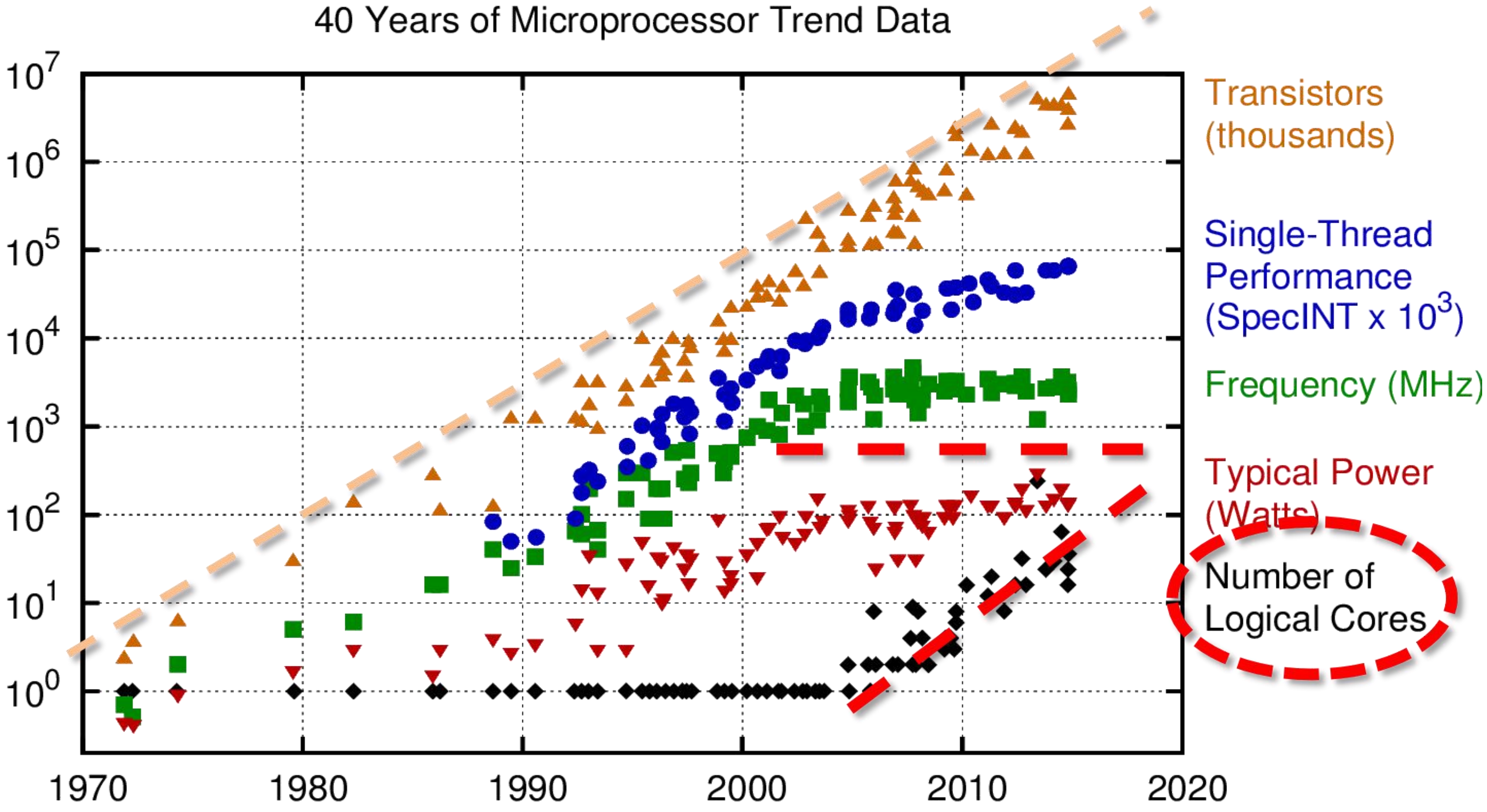
40 Years of Microprocessor Trend Data

The Moore’s law is still active in current technological loop.

The number of transistor is increasing accordingly (note the log scale on the *y* axis)

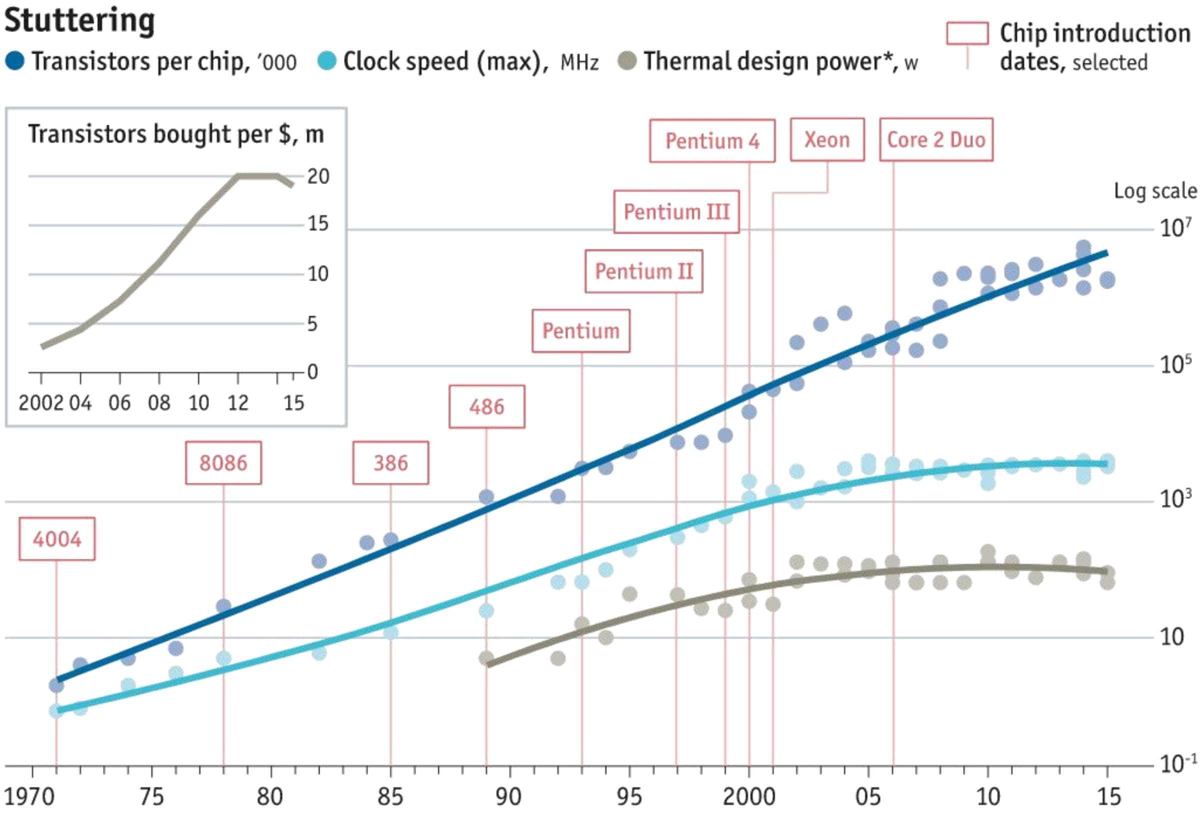
Data collected up to 2010 by M.Horowitz, F. Labonte, O. Schacham, K. Olukotun, L.Hammond and C. Batlen.
New data up to 2015 collected by 2010-2015 by K. Rupp.

Moore’s law, is it over ?



However, we observe a consistent growth in the number of logical cores found in the cpus, while other curves are flattening (namely, the frequency and the absorbed power, as well as the “performance” of single cores)

Moore’s law, is it over ?



Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

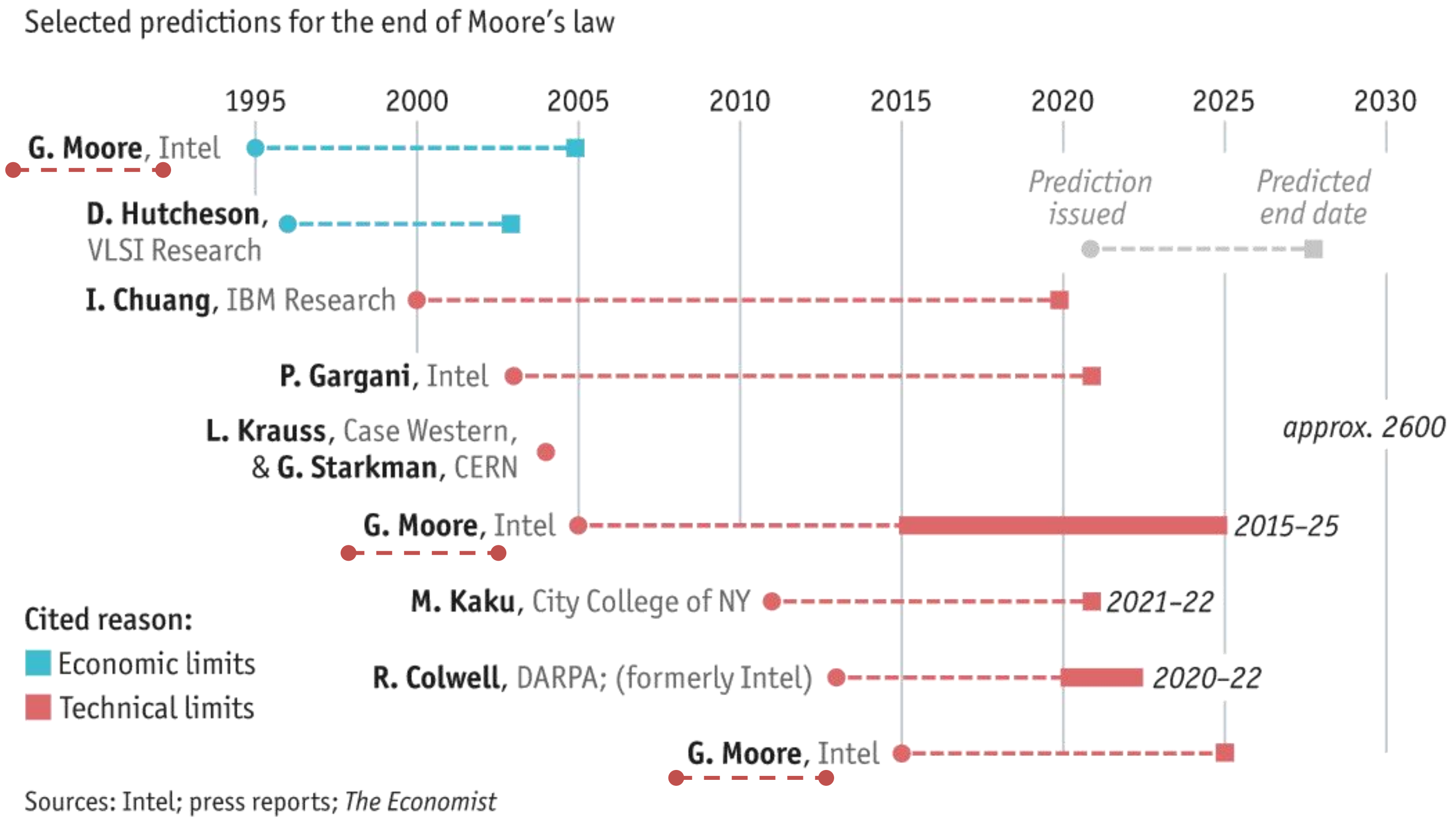
*Maximum safe power consumption

Here is a different plot of 3 key quantities, tagged with some architecture commercial name (only Intel, though. Do not forget other vendors, like AMD, ARM, IBM).

Moore’s law, is it over ?

So the Moore’s law is (not yet) over.

But -a curiosity- there have often been rumours about that in the past


Selected predictions for the end of Moore’s law

Sources: Intel; press reports; *The Economist*

Summary

GOOD NEWS:
processors has continued to become “more powerful”

OTHER NEWS:
Hints are that they are “differently” more powerful.
Let’s see how differently, and what is the impact for us.

The transition from the “free lunch”

Until half of the ‘00s, engineers succeeded in gaining performance by essentially 3 ways:

1. Increasing **clock** speed
2. Optimizing **execution**
3. Enlarging/improving **cache**

The transition from the “free lunch”

Until half of the ‘00s, engineers succeeded in gaining performance by essentially 3 ways:

1. Increasing **clock** speed →
2. Optimizing **execution**
3. Enlarging/improving **cache**

You get more cycles per unit time; more or less that means doing the same bunch of instructions faster

The transition from the “free lunch”

Until half of the ‘00s, engineers
gaining performance by essent

1. Increasing **clock** speed
2. Optimizing **execution** →
3. Enlarging/improving **cache**

- More powerful instructions
- Pipelining
- Branch predictions
- Out-of-order execution
- …

Under enormous pressure, CPUs manufacturers risked (and did) to break the semantic of your code. Or introduce horrible bugs..
(*have you heard about **Meltdown** and **Spectre** ?)*

The transition from the “free lunch”

Until half of the ‘00s, engineers
gaining performance by essent

1. Increasing **clock** speed
2. Optimizing **execution** →
3. Enlarging/improving **cache**

- More powerful instructions
- Pipelining
- Branch predictions
- Out-of-order execution
- …

Under enormous pressure, CPUs manufacturers risked (and did) to break the semantic of your code. Or introduce horrible bugs..
(*have you heard about* ***Meltdown*** *and* ***Spectre*** ?)

The transition from the “free lunch”

Until half of the ‘00s, engineers succeeded in gaining performance by essentially 3 ways:

1. Increasing **clock** speed
2. Optimizing **execution**
3. Enlarging/improving **cache** → More on that later..

Why there is no more “free lunch”?

Applications no longer get more performance for free without significant re-design, since ≳15 years

Since 15 years, the gain in performance is essentially due to **fundamentally different factors**:

1. Multi-core + Multi-threads
2. Enlarging/improving **cache**
3. Simultaneous Multithreading

   (SMT, known as *hyperthreading* in Intel’s language)

Why there is no more "free lunch"?

2 Cores at 3GHz are basically 1 Core at 6GHz.. ?

False, for many reasons. Among them:

- × Cores coordination for cache-coherence
- × Threads coordination
- × Memory access
- × Increased algorithmic complexity

Since 15 years, the gain in performance is essentially due to **fundamentally different factors**:

1. Multi-core + **Multi-threads**
2. Enlarging/improving **cache**
3. Hyperthreading *(smaller contribution)*

Why there is no more “free lunch”?

2 Cores at 3GHz are basically 1 Core at 6GHz.. ?

False, for many reasons.
Among them:

- × Cores coordination for cache-coherence
- × Threads coordination
- × Memory access
- × Increased algorithmic complexity

Since ance
is ess
funda

1. M
2. E
3. H ution)

To have just a glance of the increased complexity for the programmer, consider the code shown below.
Two threads from the same process, running on two cores are writing a memory location and reading the the memory location written by the other thread: which is the correct order of the operations?

Core 1	Core 2
`mov [X], value`	`mov [Y], value`
`mov reg1, [Y]`	`mov reg2, [X]`

Why there is no more “free lunch”?

Let’s discuss what has led to this big change in paradigm.

Why there is no more “free lunch”?

Moore’s law	Dennard’s scaling (in MOSFET technology)
Manufacturing cost/area is constant while the transistors’ dimension halves every ~2 years → **The number of transistors doubles in a CPU every ~2 years**	- *voltage*, *capacitance*, *current* scale with λ - Transistor power scales as λ^2 → **Power density remains constant**

Power consumption:

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

Why there is no more "free lunch"?

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

C is the capacitance, scales as the area
V is the voltage, scales as the linear dimension
f is the frequency

so, the linear size of transistors shrinks and so do the voltage; if the area remains equal (more transistor on the same die), then the frequency can become larger

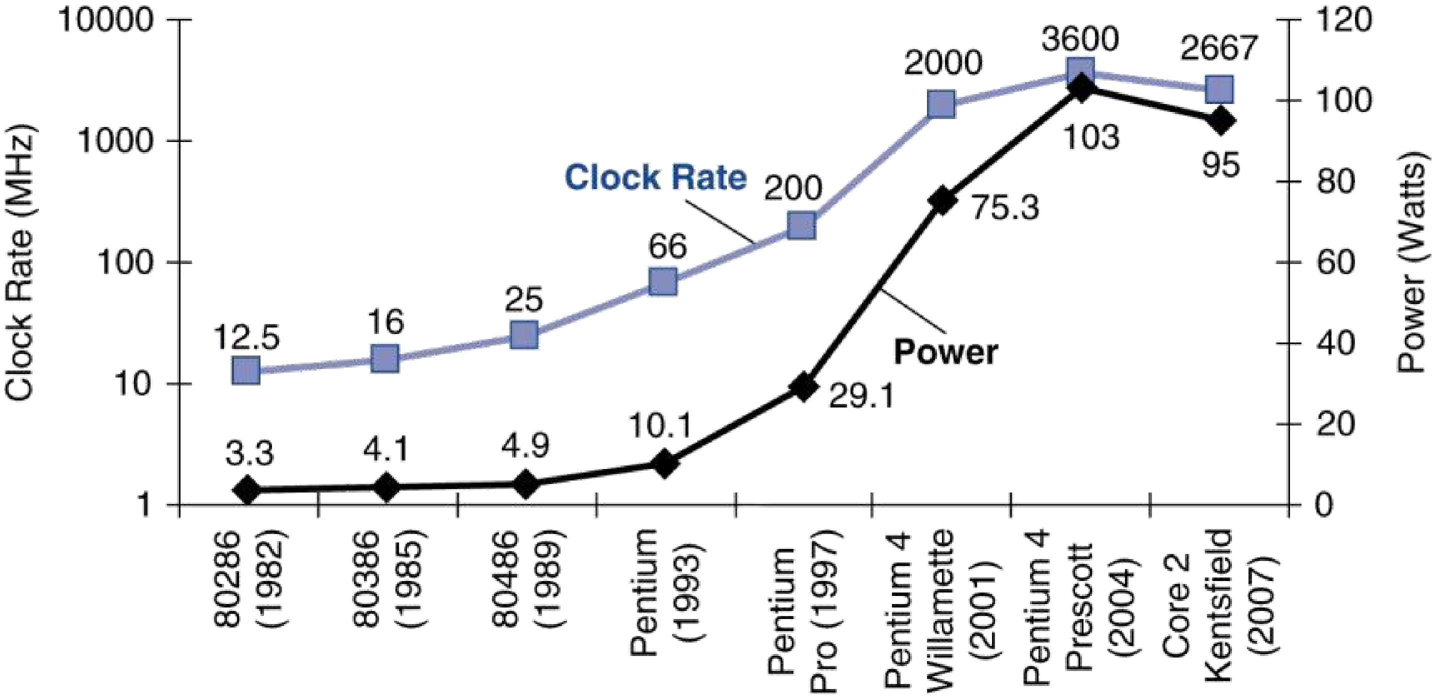
Why there is no more “free lunch”?

$\times \sim 1000$

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

$\times \sim 30$

5V → 1V



Why there is no more “free lunch”?

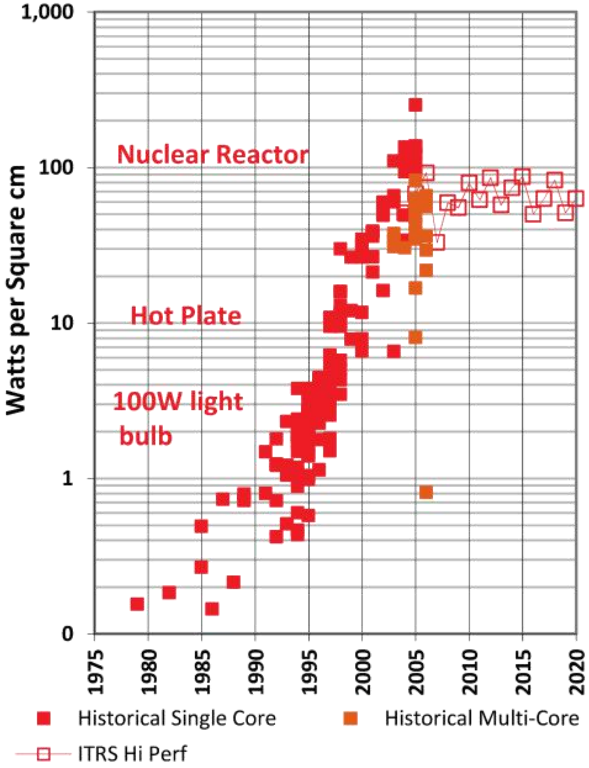
In summary, due essentially to quantum effects

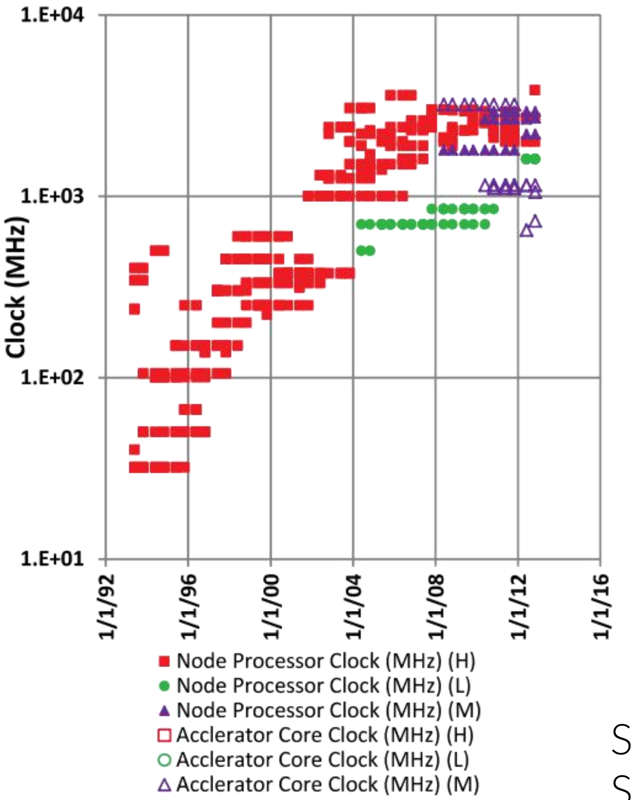
- Leakage current
- Threshold voltage
- Physical limits ad atomic scales

the *Dennard’s scaling is broken*, while the Moore’s law could still work (for a while at least).
So, as transistors get smaller **the power density actually increases**.
The result is a “**power wall**” that prevented the clock frequency to increase beyond ~3GHz since ~12 years

A dramatic change in the technological paradigm would be needed to revert the trend.

Why there is no more “free lunch”?





Source: Kogge and Shalf, IEEE CISE

“Free lunch” is over

In the Top 500 the performance is missing,
when it comes to consider “real” applications

	Rpeak (Pflop/s)	HPL / Rpeak	HPCG / Rpeak	HPL rank	HPCG rank
FRONTIER	1685	0.66	-	1	-
FUGAKU	514	0.8	0.026	2	1
LUMI	214	0.7	0.009	3	3
SUMMIT	201	0.74	0.015	4	2
SIERRA	126	0.75	0.014	5	5

From the top500 ranking, June 2022

“Free lunch” is over

) the performance is missing,
to consider “real” applications

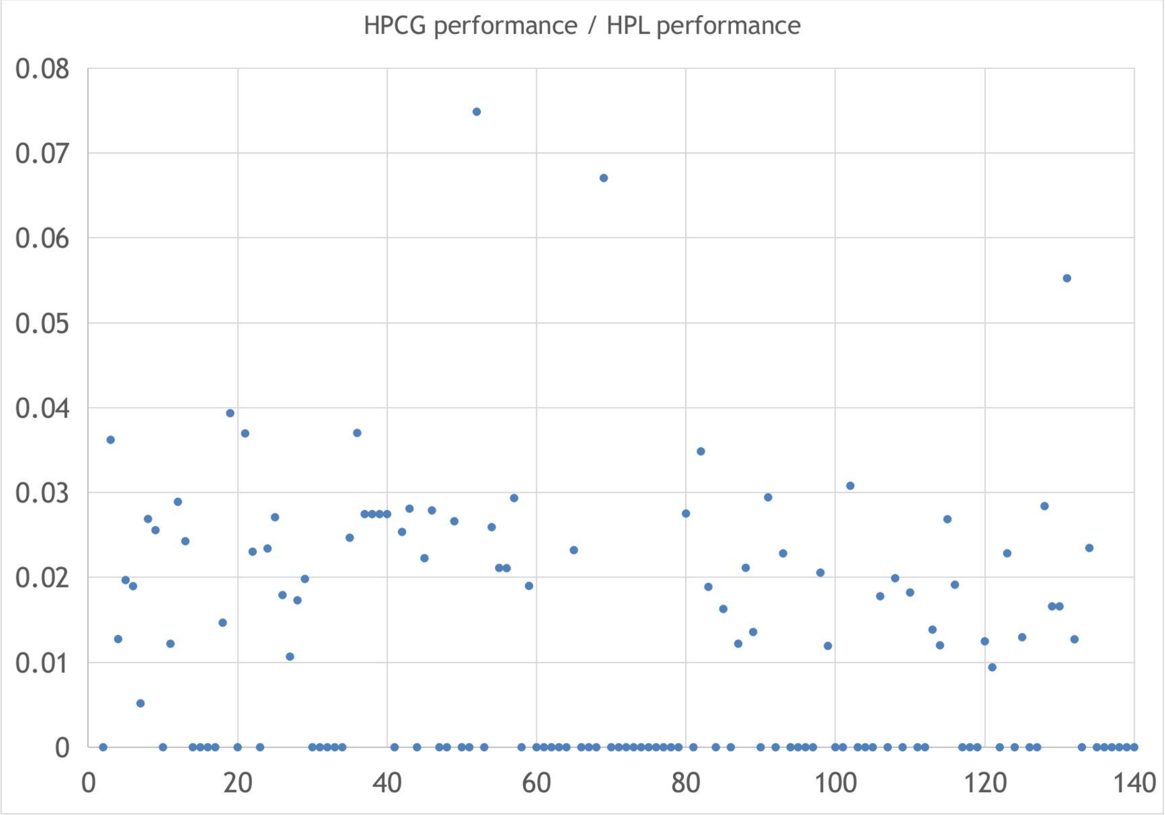
Note that the ratio between the performance in HPL and HPCG and the theoretical peak performance (Rpeak) is *(i)* not close to one even for HPL and *(ii)* **much** smaller for HPC.

Remind:
HPL : solution of a dense set of linear eqaution; very arithmethic intense, moves little memory

HPCG: calculate conjugate gradient; move much more memory

HPL / Rpeak	HPCG / Rpeak	HPL rank	HPCG rank
0.66	-	1	-
0.8	0.026	2	1
0.7	0.009	3	3
0.74	0.015	4	2
0.75	0.014	5	5

"Free lunch" is over

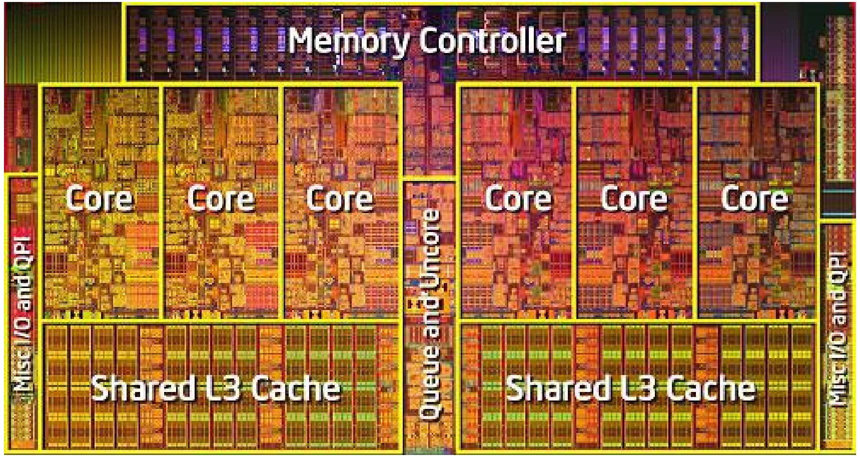


This is what happens when you compare the Flop performance estimated by **HPL** with that estimated by **HPCG**

HPCG is a different code with a much smaller arithmetic intensity and, above all, it moves memory!

Back to the future

Take-home message
Many-cores CPUs are here to stay



- Concurrency-based model programming (different than both *parallel* and *ILP*): work subdivision in as many independent tasks as possible

- Specialized, heterogeneous cores

- Multiple memory hierarchies

The Energy Challenge

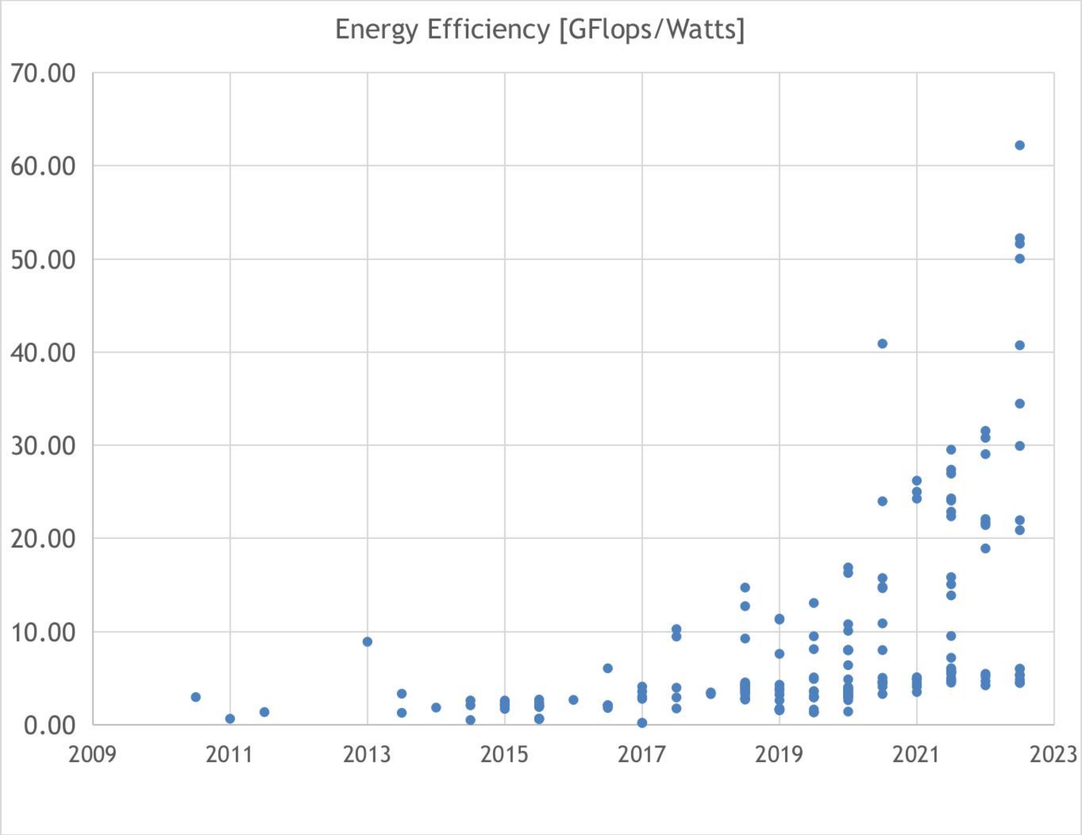
The #1 in top500, Frontier @ Oak Ridge National Laboratory, absorbs ~**21 MW** of electrical power to exhibit the HPL-based exaflop.

Assuming that HPCG is a better proxy of a “real” performance, and considering a fiducial value

$$\mathrm{HPCG}_{\mathrm{performance}} \,/\, \mathrm{HPL}_{\mathrm{performance}} \sim 0.03$$

Then, to express a HPCG-exaflop it would require more than ~ **600 MW** of power, i.e. more or less a dedicated nuclear power plant.

The Energy Challenge



The energy efficiency is a topic that raised attention only recently in the HPC sector.

Now it is of paramount importance.

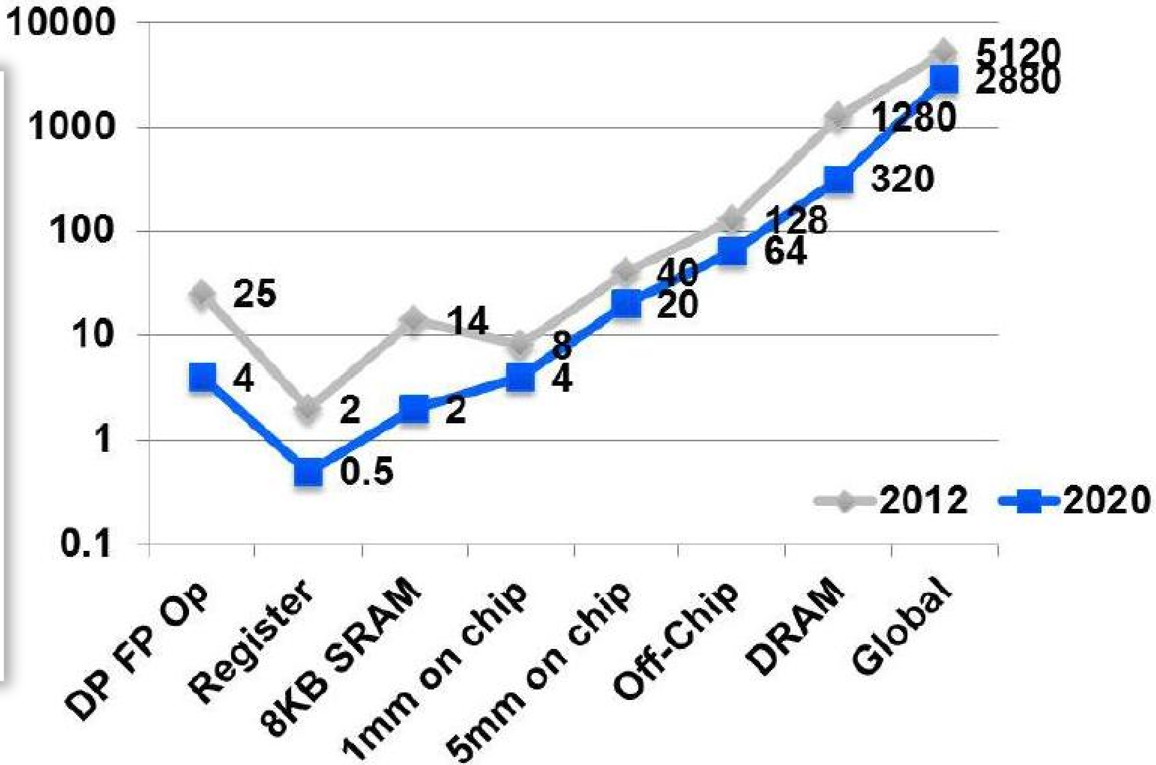
The Energy Challenge

(∗)

Message II

Moving memory is a very expensive operation.
A FP op could cost ~4pJ while reading the data to be operated from memory ~700pj.

Data and projections:
R. Leland et al., 2014

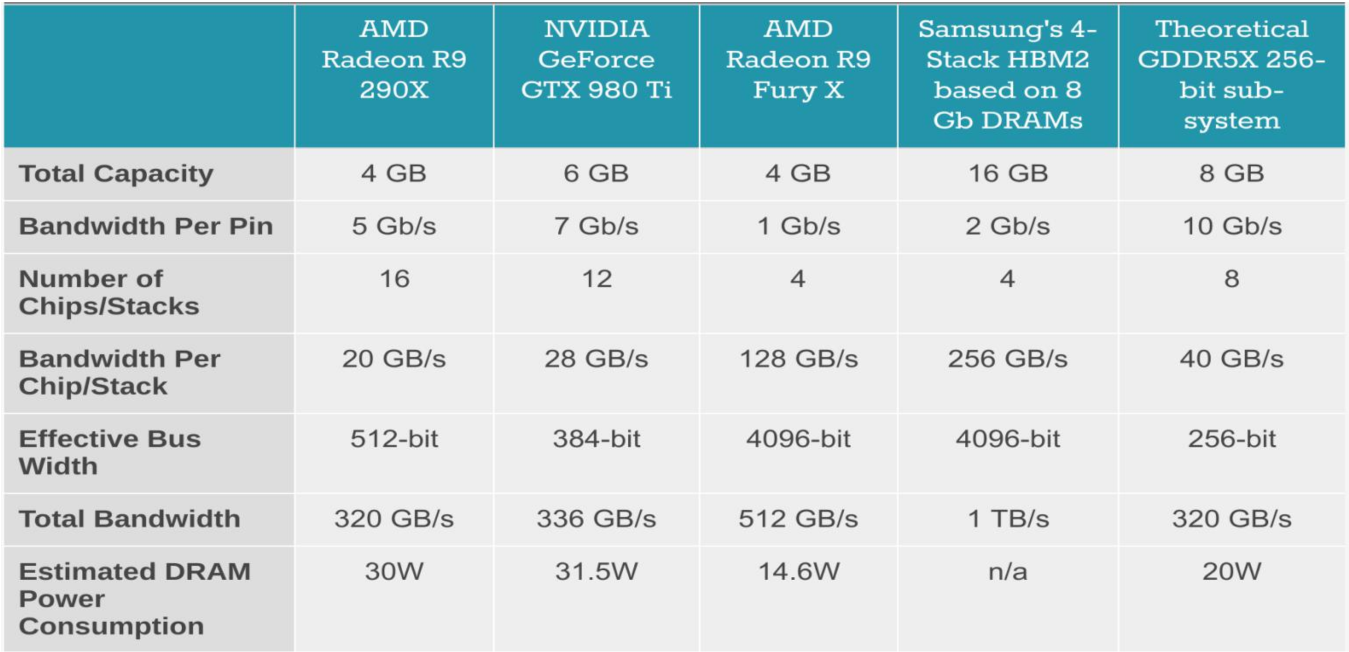


The Energy Challenge

Memory power consumption

$\propto$

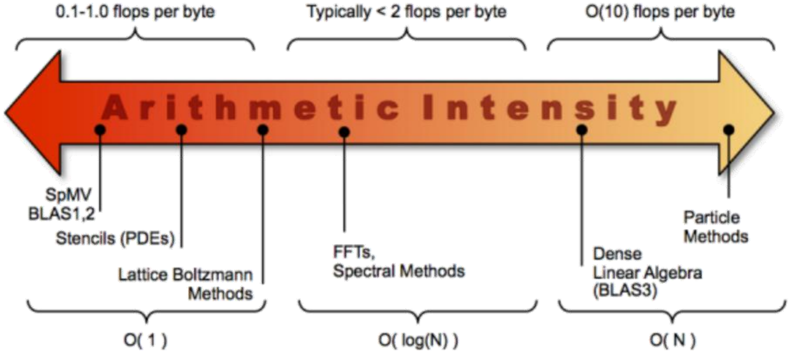
$Bw \times L^2 / Area$

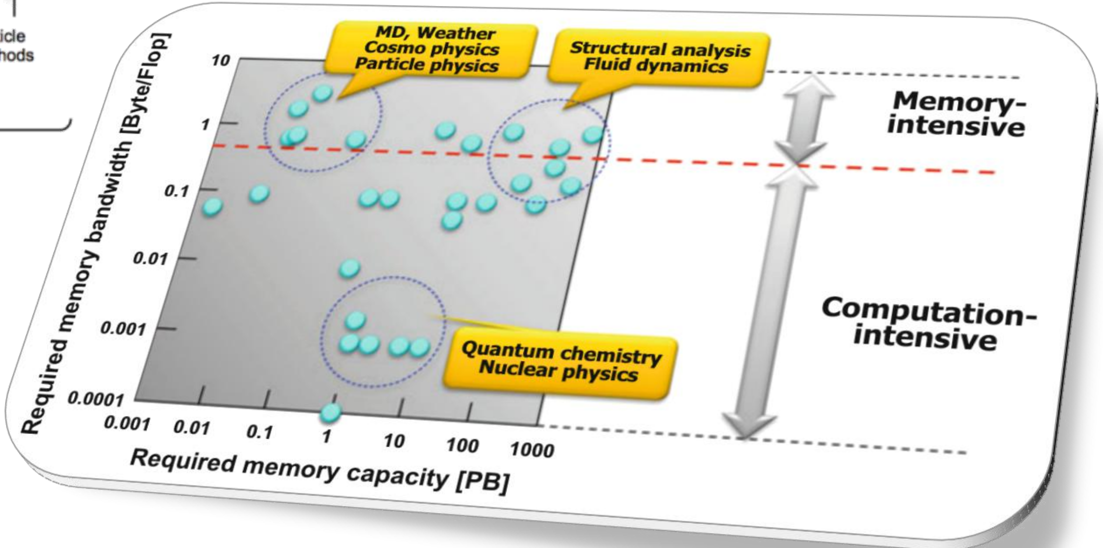


	AMD Radeon R9 290X	NVIDIA GeForce GTX 980 Ti	AMD Radeon R9 Fury X	Samsung's 4-Stack HBM2 based on 8 Gb DRAMs	Theoretical GDDR5X 256-bit sub-system
Total Capacity	4 GB	6 GB	4 GB	16 GB	8 GB
Bandwidth Per Pin	5 Gb/s	7 Gb/s	1 Gb/s	2 Gb/s	10 Gb/s
Number of Chips/Stacks	16	12	4	4	8
Bandwidth Per Chip/Stack	20 GB/s	28 GB/s	128 GB/s	256 GB/s	40 GB/s
Effective Bus Width	512-bit	384-bit	4096-bit	4096-bit	256-bit
Total Bandwidth	320 GB/s	336 GB/s	512 GB/s	1 TB/s	320 GB/s
Estimated DRAM Power Consumption	30W	31.5W	14.6W	n/a	20W

Feeding 1B / flop for 10^{18} flop/s ~28 MW ~60 MW

The Energy Challenge

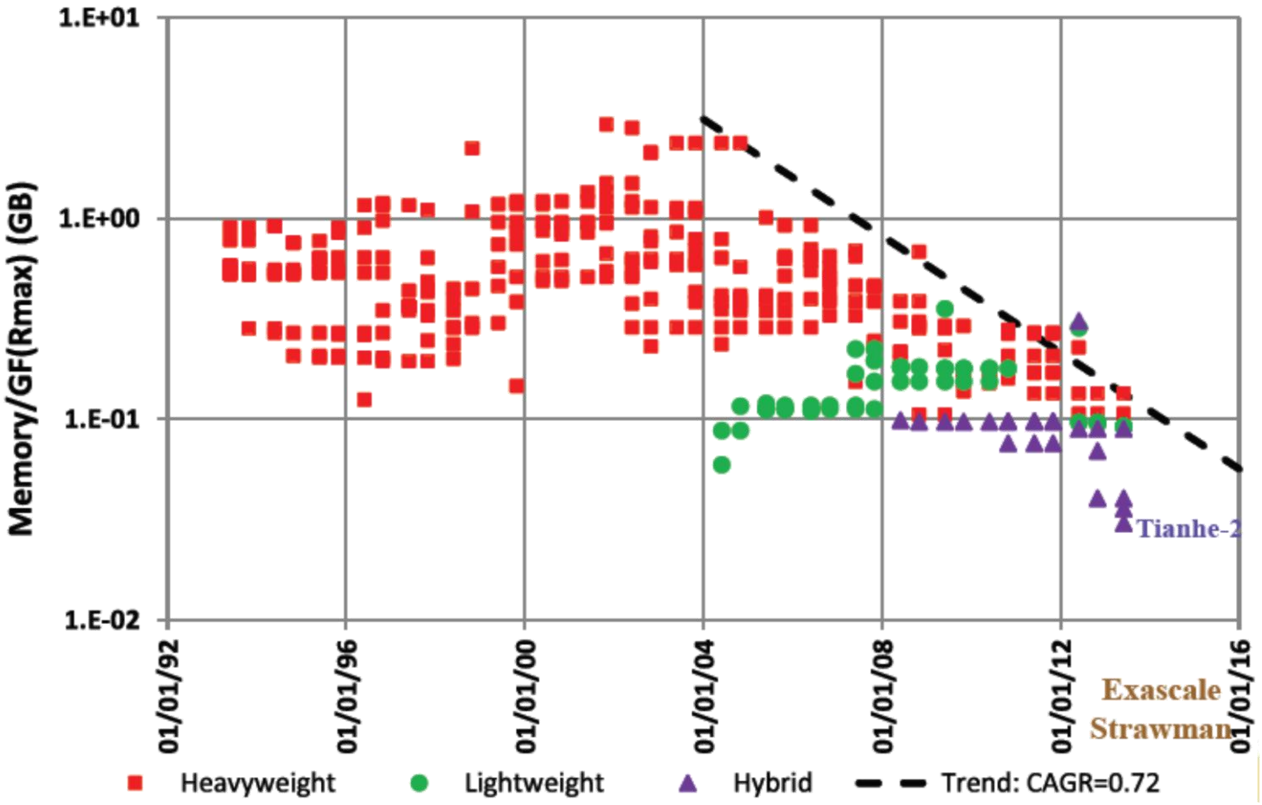




The Energy Challenge

The machines at the top of the TOP500 **do not have sufficient memory to match historical requirements of 1B/Flop**, and the situation is getting worse.

This is a big change: it places the burden increasingly on **strong-scaling** of applications for performance, rather than on **weak-scaling** like in tera-scale era.



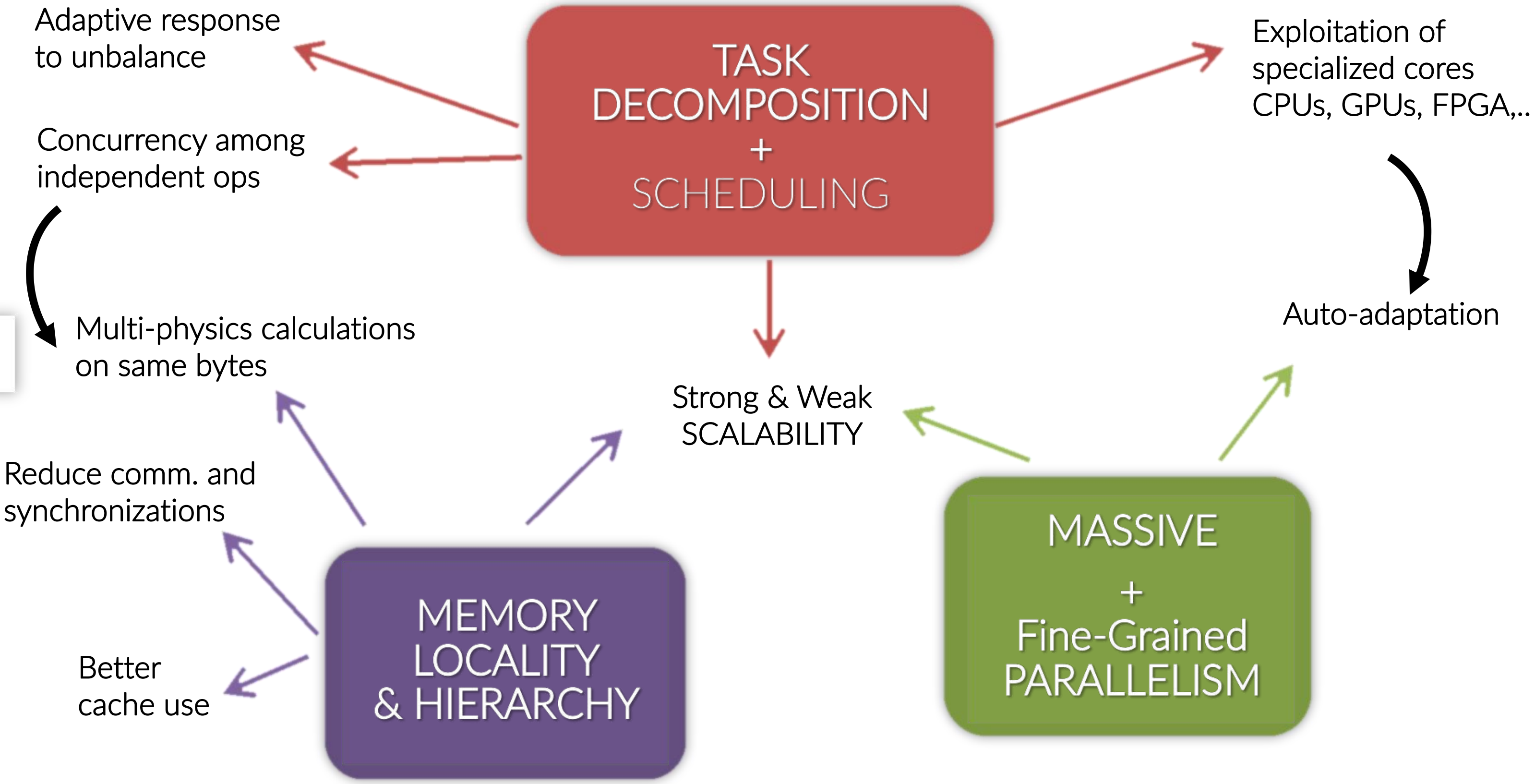
No more “free lunch”

All in all it means that the times in which you just had to wait and rely on the hardware evolution to get performance is over.

Now *you* have to write a good code if you really want to squeeze High Performance, or Extreme Performance, out of a supercomputer

No more “free lunch”

Message III



TASK DECOMPOSITION + SCHEDULING

Adaptive response to unbalance

Concurrency among independent ops

Multi-physics calculations on same bytes

Exploitation of specialized cores CPUs, GPUs, FPGA,..

Auto-adaptation

Strong & Weak SCALABILITY

MEMORY LOCALITY & HIERARCHY

Reduce comm. and synchronizations

Better cache use

MASSIVE + Fine-Grained PARALLELISM

that’s all, have fun