Luca Tornatore - I.N.A.F.

"**Foundation of HPC" course**

DATA SCIENCE &
SCIENTIFIC COMPUTING
2019-2020 @ Università di Trieste

The “free lunch”

From ‘60s to mid of ‘90s the performance gain of a software was due essentially to 1 main factor:

the **clock** speed increase of the CPUs

Now “Free lunch” is over (*)

As first,

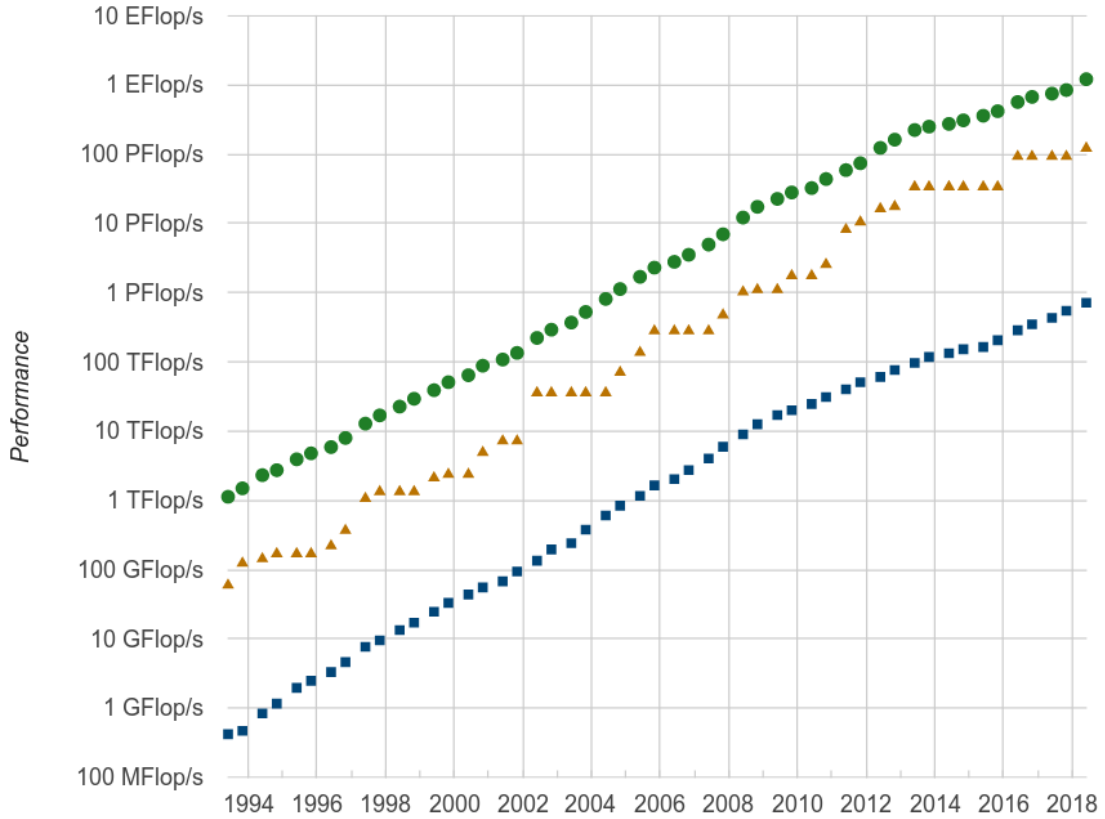
“There ain’t no such thing as free lunch”(**)

R.A. Heinlein
The Moon is a harsh mistress

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

“Free lunch” is it over ?

Top 500



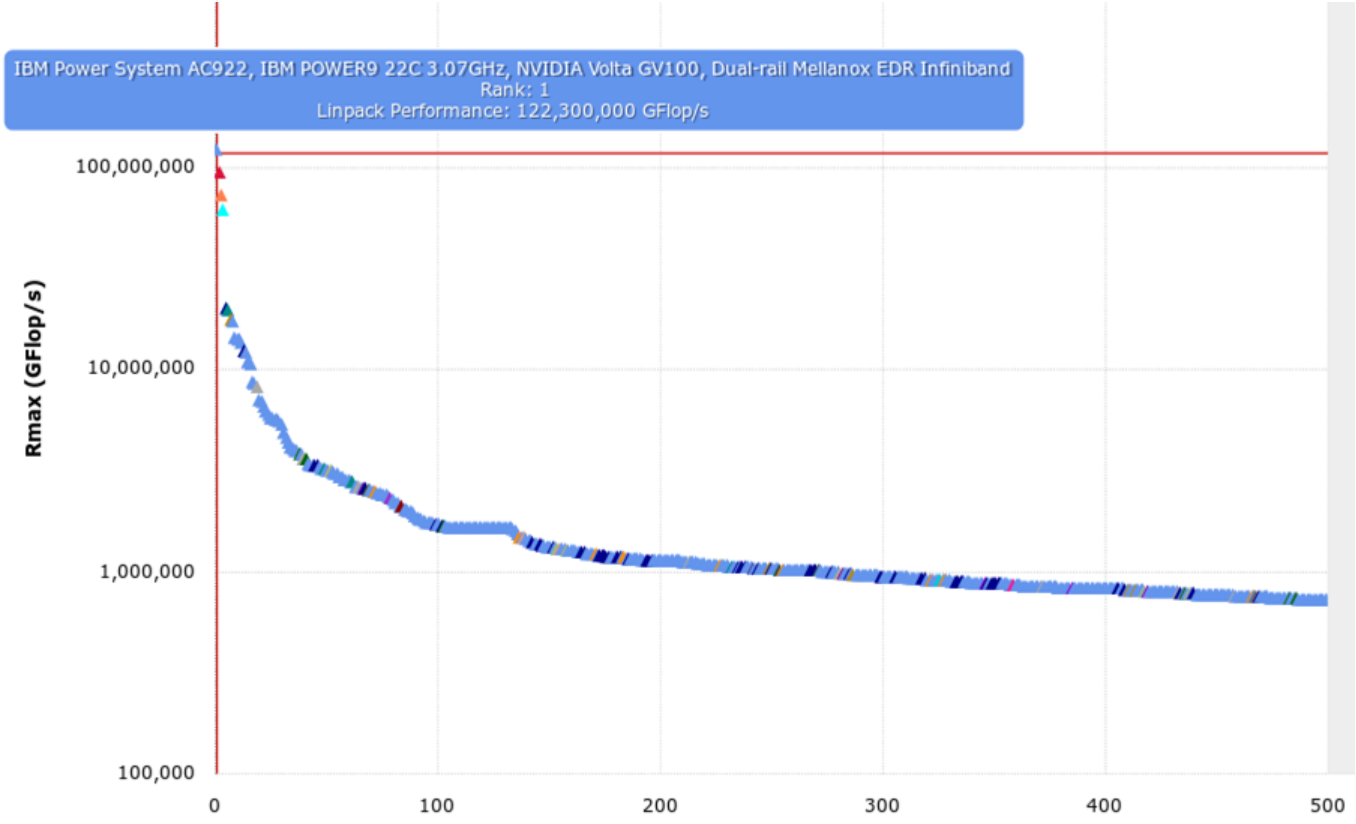
“Free lunch” is it over ?

Top 500

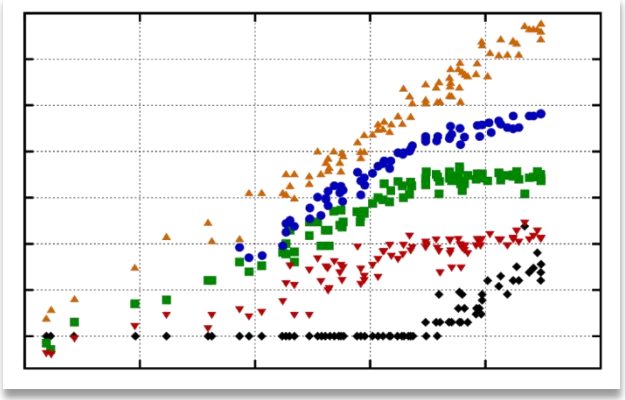
Summit performs with
LINPACK
at ~122 Pflop/s.

However, it performs with
HPCH
at ~3 Pflop/s.

Piz Daint, 5th in the list,
performs at ~0.5 Pflop/s



“Free lunch”, is it over ?



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

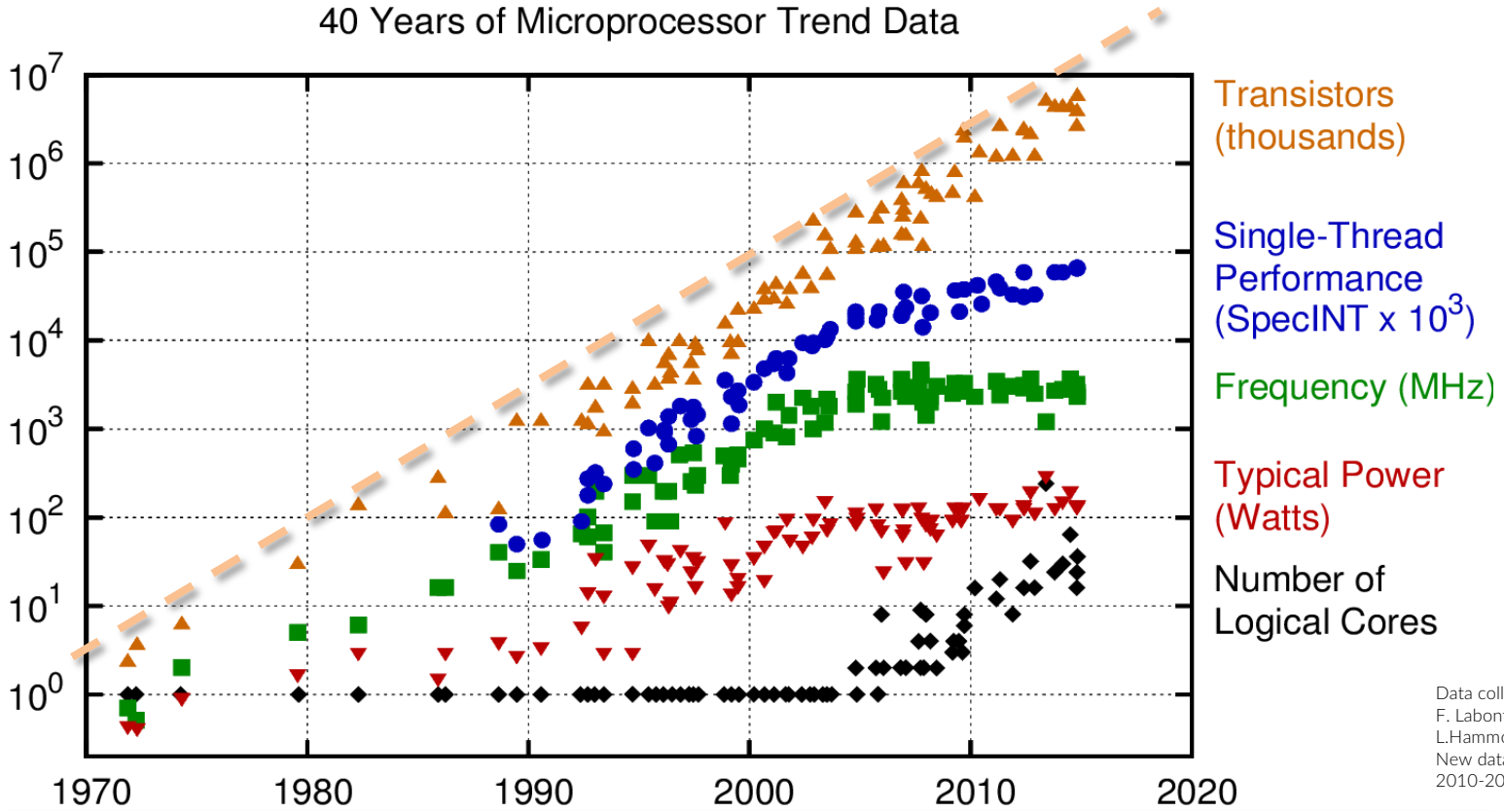
every 18 months the density of transistor will double *at the same manifacturing cost* (*).

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

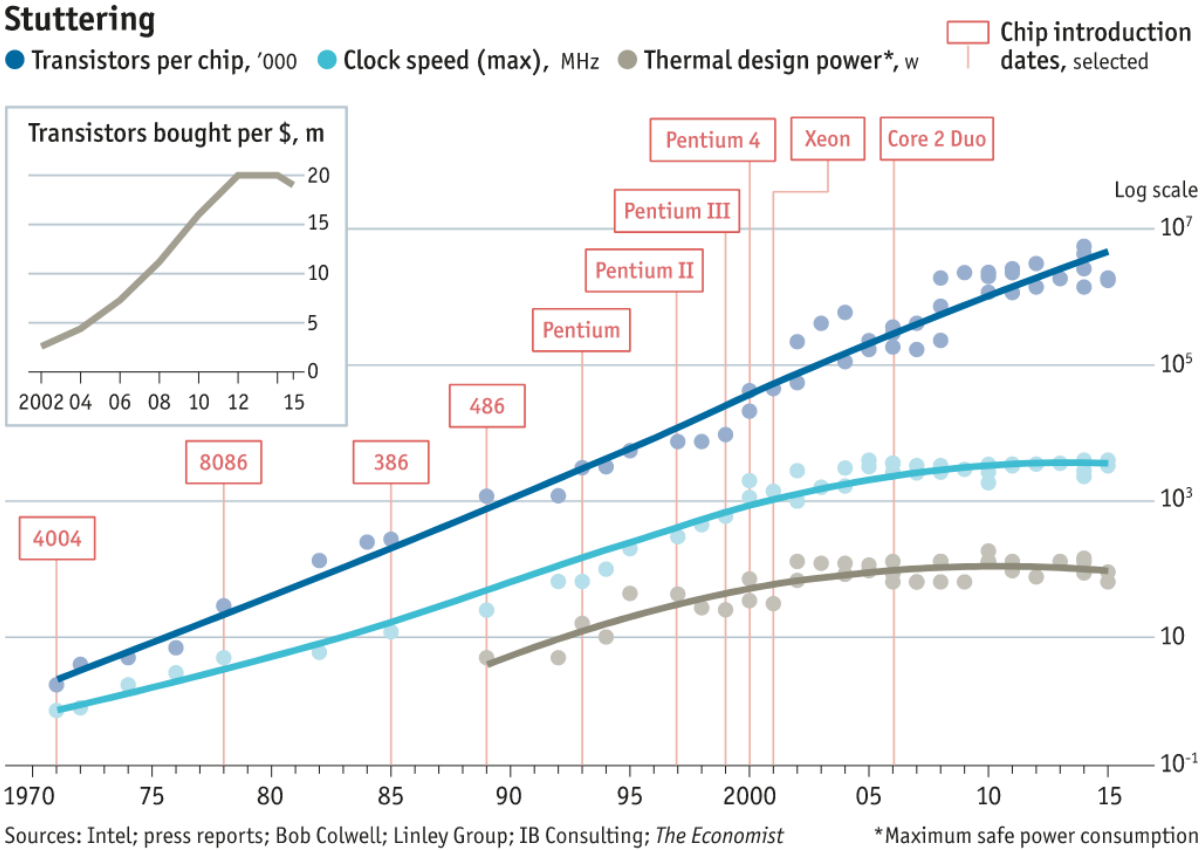
All exponential growths get to their end..

Moore's law, is it over ?

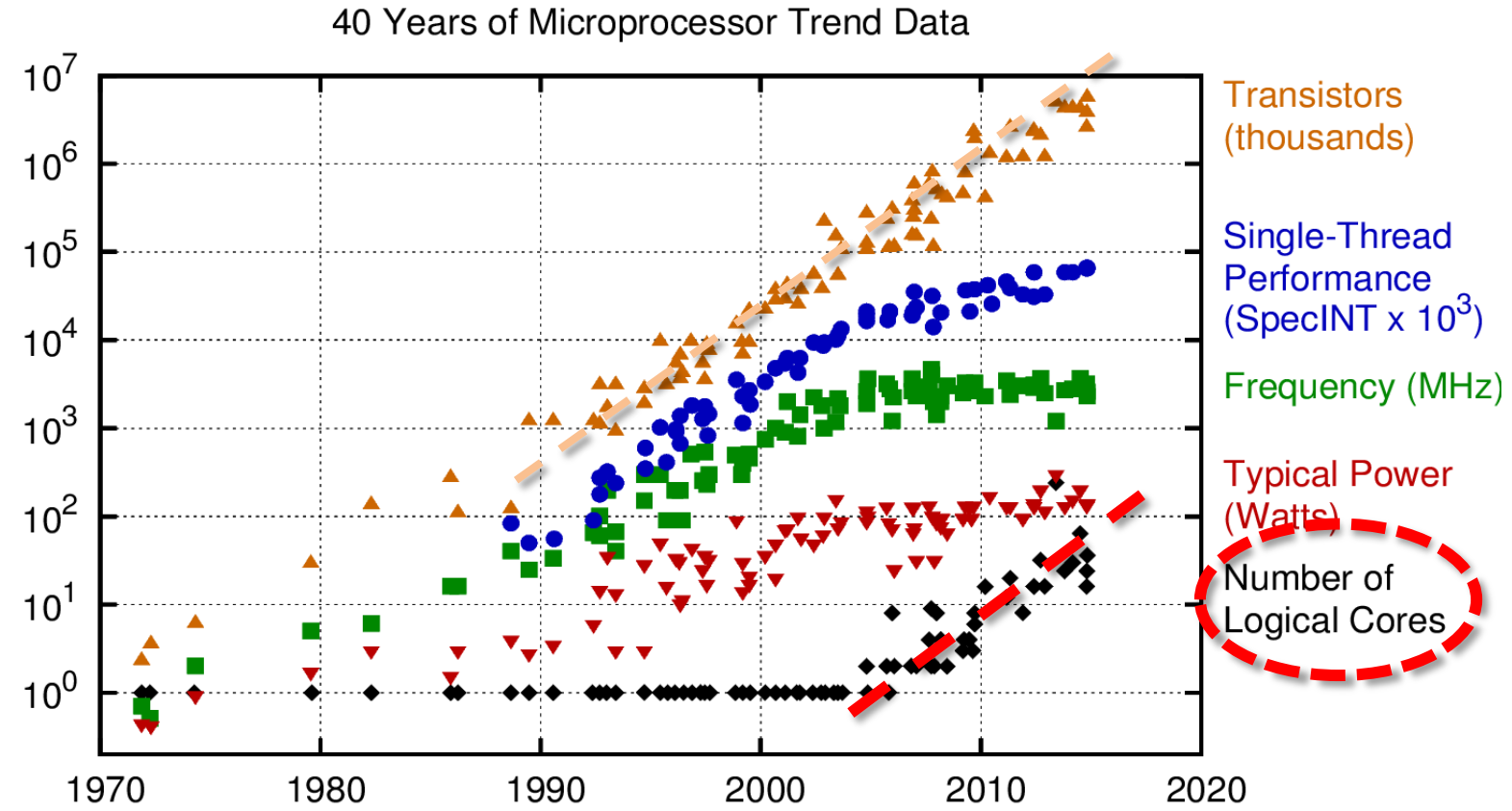


Data collected up to 2010 by M.Horowitz, F. Labonte, O. Schacham, K. Olukotun, L.Hammond and C. Batlen.
New data up to 2015 collected by 2010-2015 by K. Rupp.

Moore's law, is it over ?

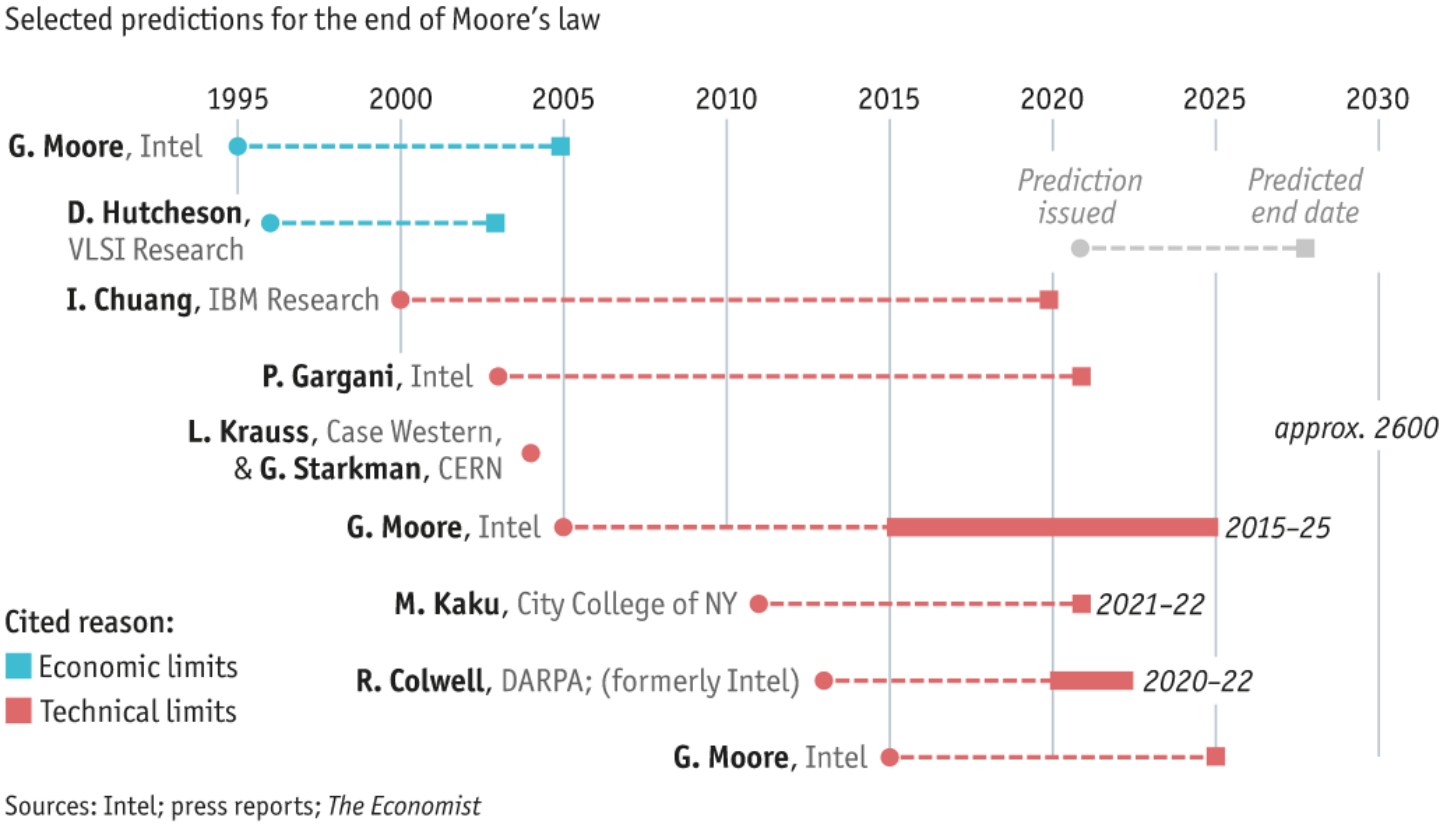


Moore's law, is it over ?



Moore's law, is it over ?

Not yet.
But there are rumors about that.



Why there is no more “free lunch”?

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

OTHER NEWS:
They are *(increasingly)* “differently” more powerful

Why there is no more “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**

Why there is no more “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

Why there is no more “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*** *?)*

Why there is no more “free lunch”?

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

1. Increasing **clock** speed
2. Optimizing **execution** →

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

- 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*** *?)*

Why there is no more “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. Hyperthreading *(smaller contribution)*

Why there is no more “free lunch”?

For instance:

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

False

- × 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)*

Will there be any more “free lunch”?

That is highly unlikely, unless some fundamental breakthrough in underlying physics appears.

Let’s summarize some physical argument in the following slides.

Why there is no more “free lunch”?

Moore’s law	Dennard’s scaling (MOSFET)
Manifacturing cost/area is costant 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

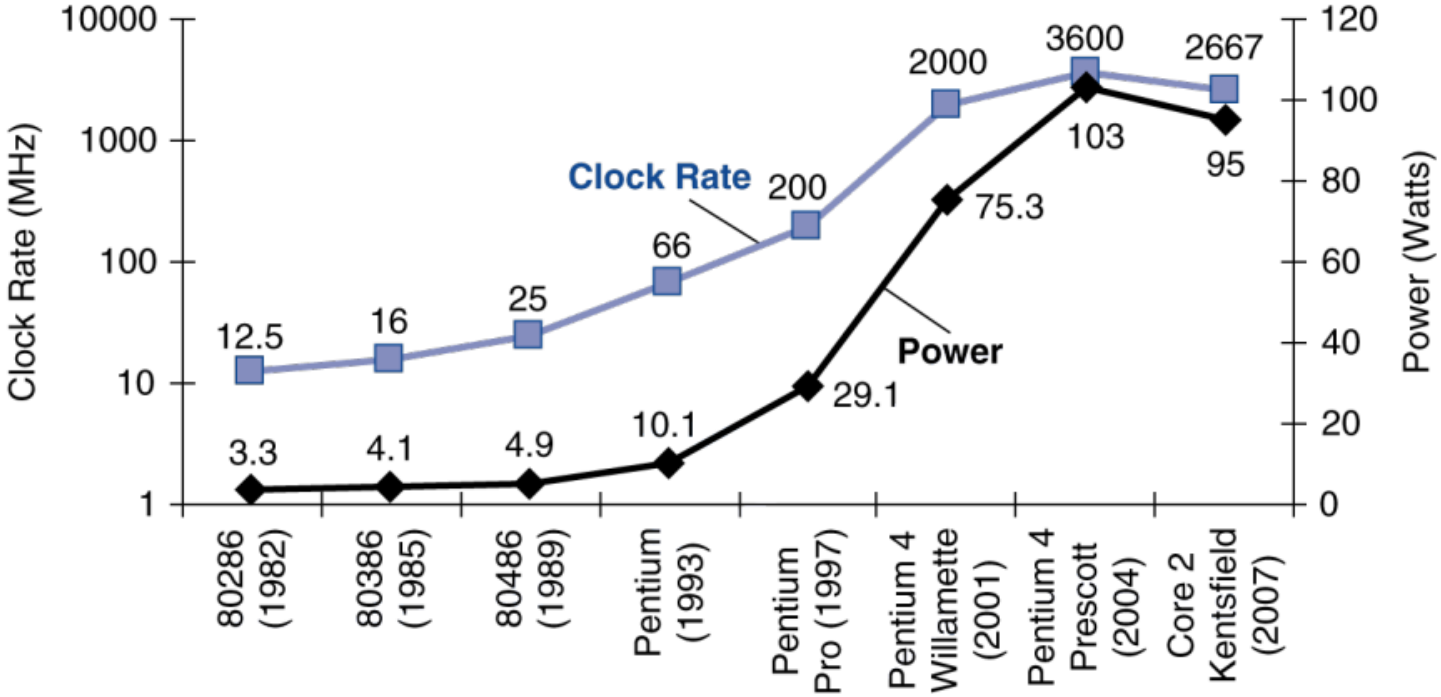
Why there is no more “free lunch”?

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

×~ 1000

×~ 30

5V → 1V



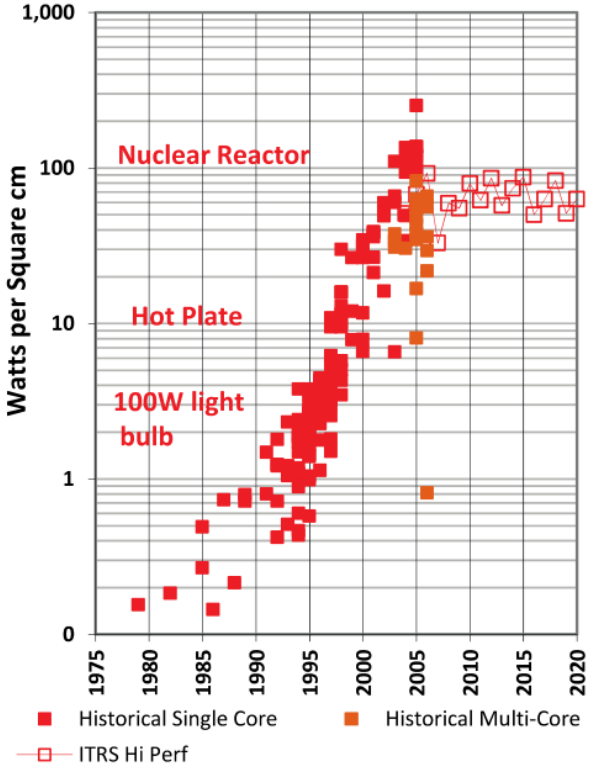
Why there is no more “free lunch”?

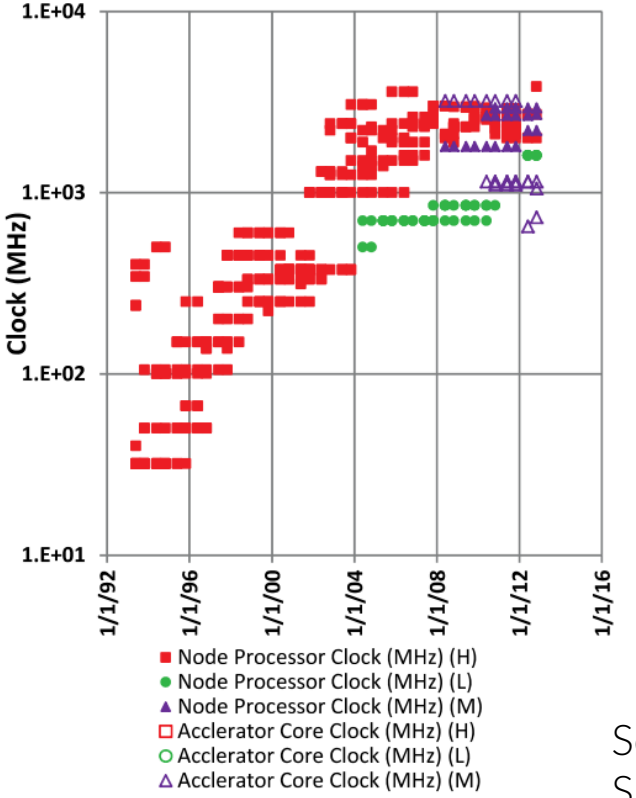
The *Dennard’s scaling* has broken down to

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

So, as transistors get smaller the power density actually increases.
The result is a “power wall” that prevented the clock frequency to increase beyond ~4GHs since ~12 years

Why there is no more "free lunch"?

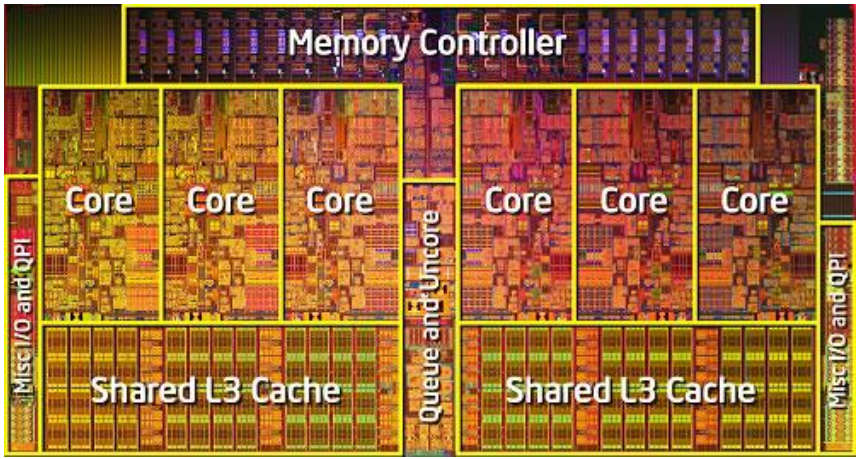




Source: Kogge and Shalf, IEEE CISE

Back to the future

Message I
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

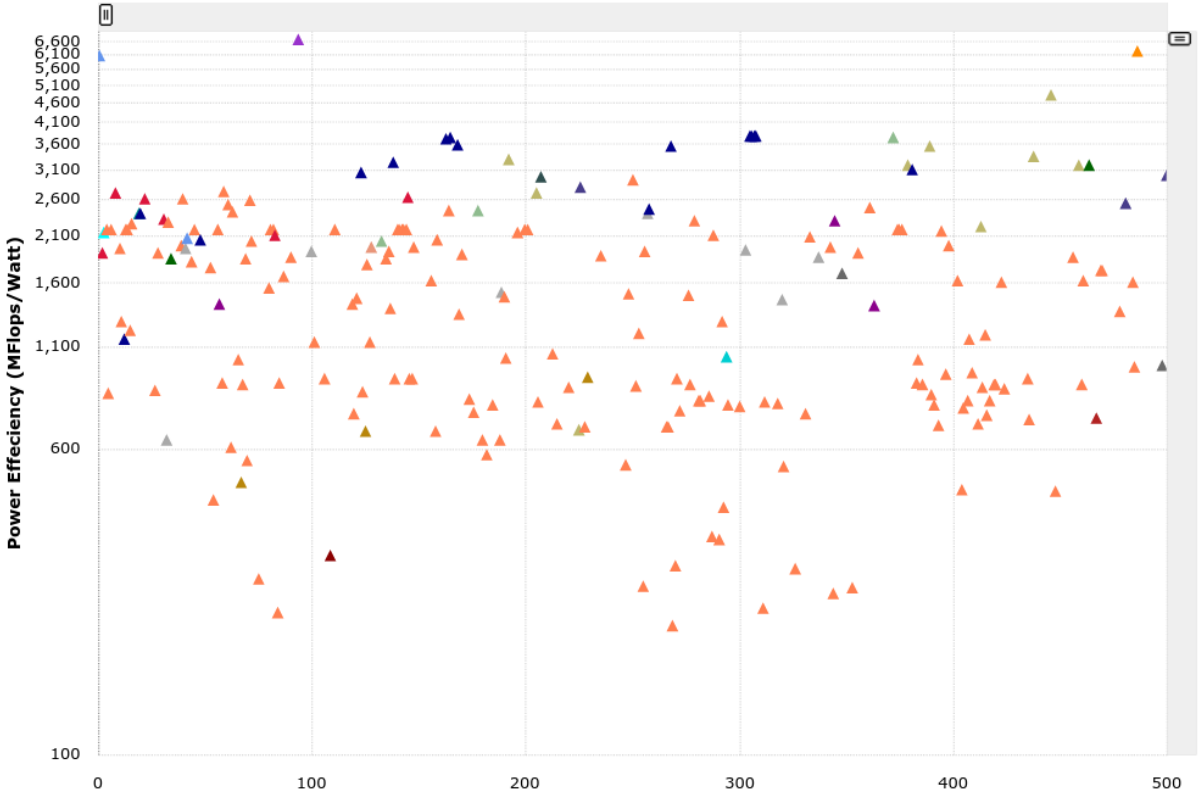
The Energy Challenge

Sunway performs for some apps at ≈ 10 Pflop/s consuming ≈ 18MW.

Simply rescaling to Eflop/s, it would consume ≈1.8GW.

The exa-scale goal is to reach Eflop/s at 20MW of electric power, i.e. 50 Gflops/W

Rule of thumb: 1MW = $1M / yr

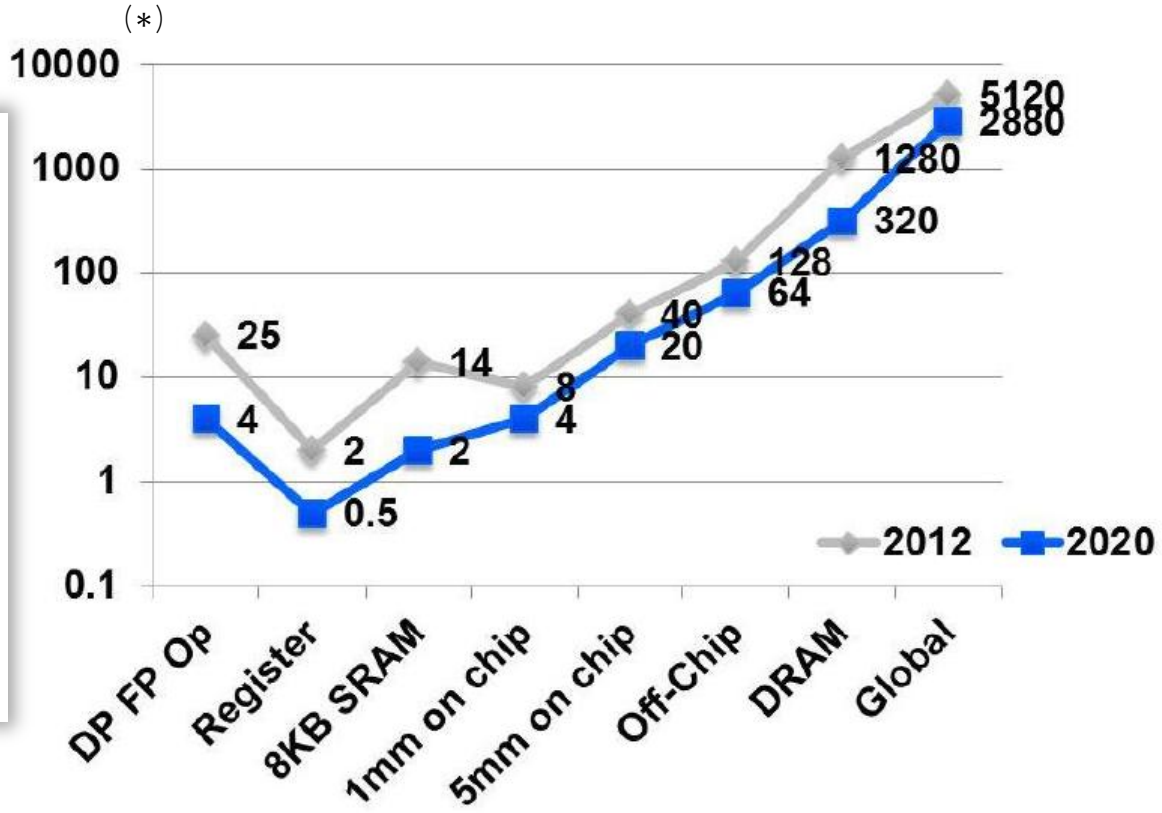


The Energy Challenge

Message II

Moving memory is among the most expensive operation.
By 2020 a FP op could cost ~4pJ while reading the data to be operated from memory ~700pj.

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



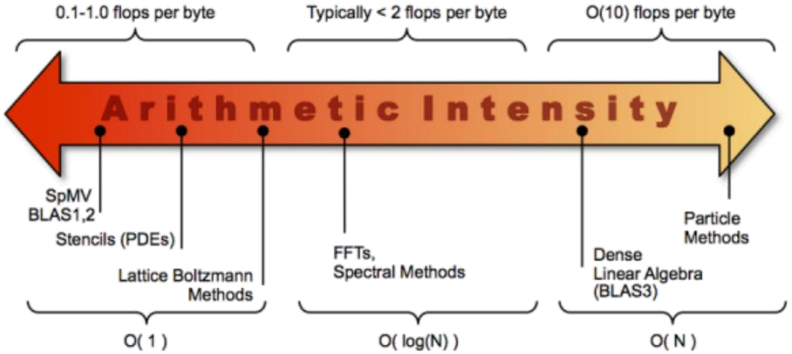
The Energy Challenge

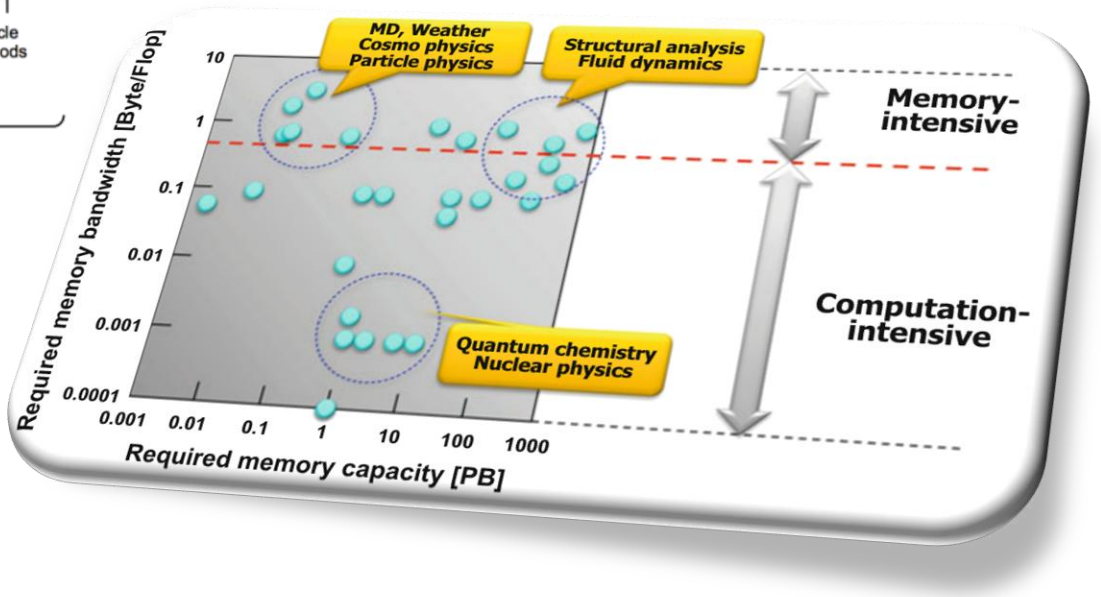
Memory power consumption
$\propto$
Bw $\times$ L^2 / Area

	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

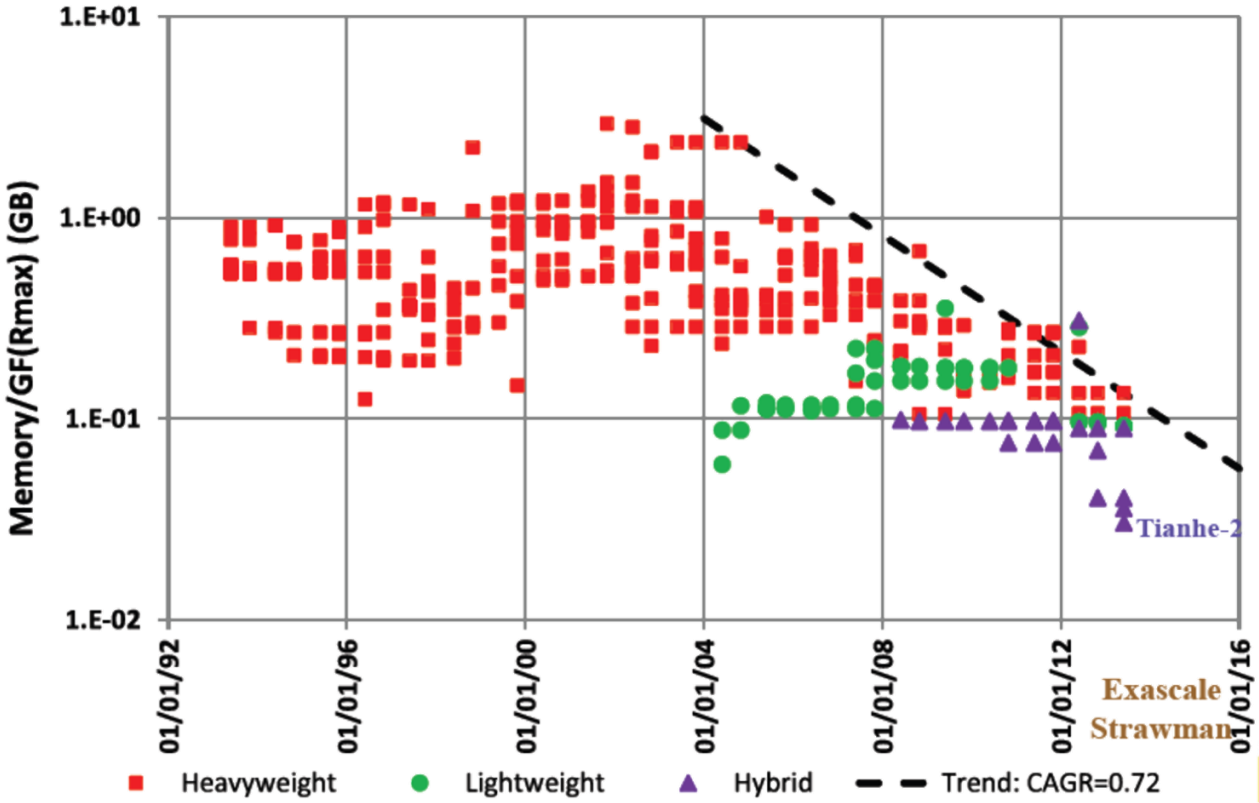




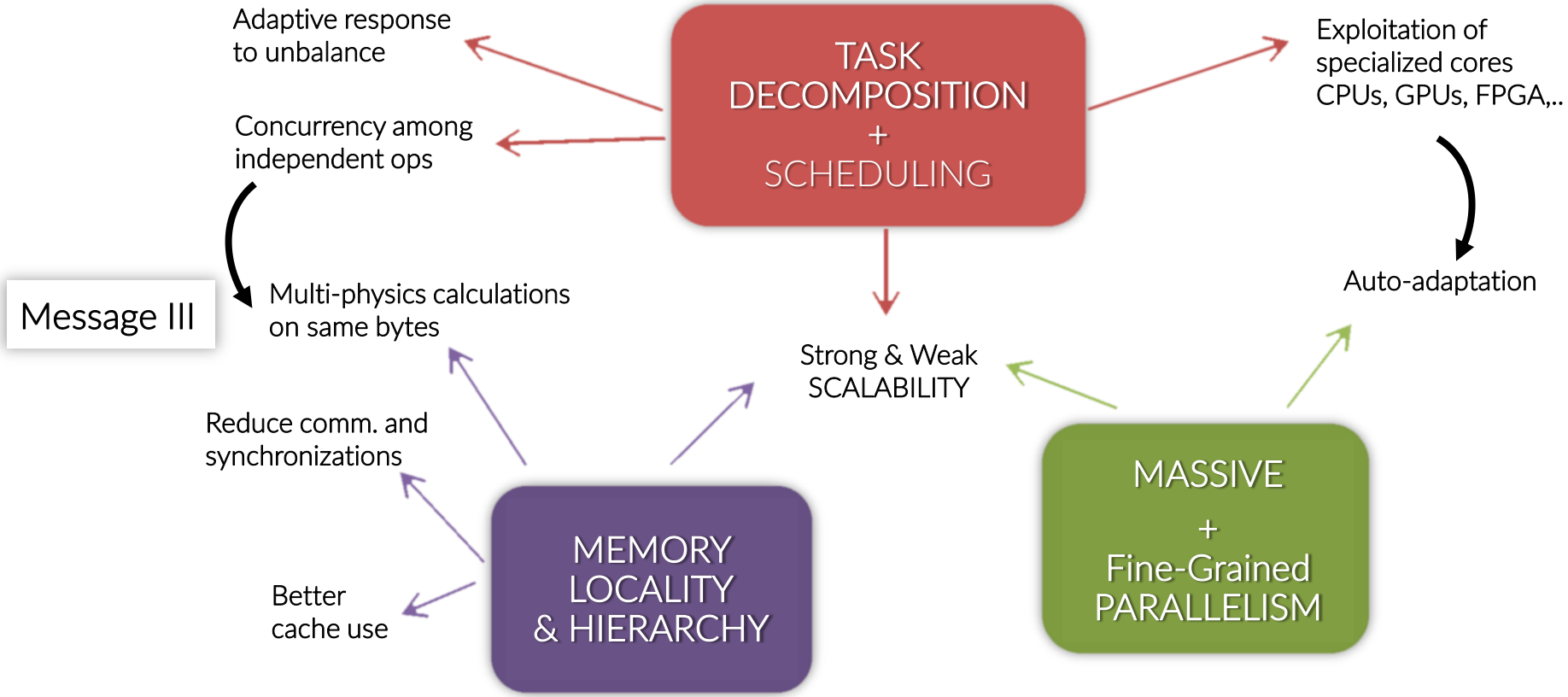
The Energy Challenge

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.



The Energy Challenge



that's all, have fun