JÜLICH
FORSCHUNGSZENTRUM

Amdahl meets Exascale

Status and Perspectives of Scientific Computing

Legnaro, INFN Laboratories

Thomas Lippert

Jülich Supercomputing Centre

What I was asked to talk about:

EUROPEAN HPC DEVELOPMENT PROJECTS

- **PAST**
  - APE (…, 100, mille, next)
  - QPACE
- **RIGHT NOW**
  - PRACE: Co-development of SC component technology
  - PRACE PP, PRACE 1IP, PRAVE 2IP
  - PRACE 1IP: 6 projects: GPU, I/O, BGQ,Cooling
- **SOON TO COME ?**
  - EU Exascale Call 9/2010:

    → 3 projects (decision March 2011)

    AXIO, DEEP, MONT BLANC, etc in the competition
  - FET Flagships that need HPC at scale:
    - *Human Brain Project: three arch. Lines to follow*
    - *FutureICT (Social Computing)*

IDC Recommendations Report: For EU HPC Leadership In 2020

October 2010

Authors: Earl Joseph, Steve Conway and Jie Wu

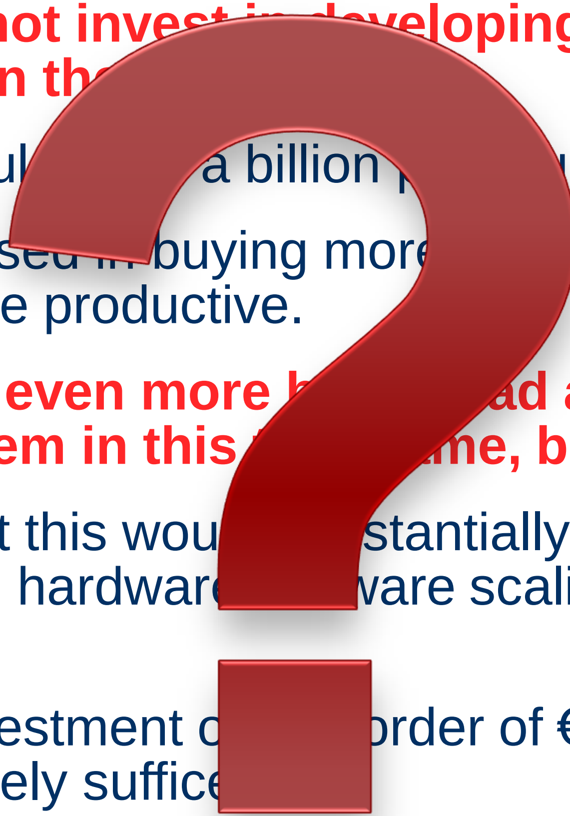
Impression

- The IDC report gives a host of very important recommendations
- It strengthens the role of PRACE and emphasizes the importance of large investments in HPC systems
- It emphasizes to concentrate on visible strengths → HPC software

Actions Required To Achieve The Vision: Purchase vs. Develop Exascale Systems

IDC Analyze the Future

- **IDC recommends that the EU buys the 4th or 5th exascale system, and does not inve[illegible]loping the first 1, 2 or 3 exascale systems in th[illegible]**
  - This strategy coul[illegible] a billion [illegible]uros
  - –> that can be use[illegible] buying more[illegible]ems and making EU researchers more productive.
- **Europe could save even more [illegible]ad aiming to purchase a near-exascale system in this [illegible]me, but 1 year later.**
  - We estimate that this wou[illegible]stantially reduce the technical challenges (e.g., hardware[illegible]ware scaling) and associated costs.
  - Such that an investment o[illegible]order of €150 to €200 million per system would likely suffice[illegible]
  - The resultant near-exascale supercomputer would still sustain unrivaled performance in the targeted application domain, and still attract the best researchers and collaborations.



PROSPECT‘s VIEW

- This is a recommendation for Europe not to strive for being one of the 10 top players worldwide
- It sounds reasonable on a short and maybe midterm timescale
- It would be disastrous on the long run
- We should not develop a consumer mentality!

Encouragement by EC Commission

- In a meeting that took place on September 2nd, 2010, in Brussels, PROSPECT has been encouraged to proceed and to speed up its preparation of the ETP HPC, approaching all relevant stakeholders and starting to draft a vision paper soon to be followed by a joint European research agenda.

EC ETP Report 2005

Technology Platforms: Overall Concept

Stakeholders, led by Industry, getting together to define a Strategic Research Agenda on a number of strategically important issues with high societal relevance where achieving Europe's future growth, competitiveness and sustainability objectives is dependent upon major research and technological advances in the medium to long term.

Steps towards ETPs

- STAGE 1: Stakeholders getting together in order to establish their "vision" for the future development of the field concerned and to set up the technology platform;
- STAGE 2: Stakeholders define a Strategic Research Agenda setting out their common views on the necessary medium to long term research, development and demonstration needs for this technology;
- STAGE 3: Implementation of the Strategic Research Agenda - for which, in many instances, it is anticipated that significant public and private investments will need to be mobilised.

Major Challenges for Scientific Computing

1. SCALABILITY
2. ENERGY CONSUMPTION

JUGENE:
10^{15}: World Record in Scalability

QPACE: World Record in Energy Efficiency 2008-2010

JÜLICH
FORSCHUNGSZENTRUM

Best Energy Efficiency Worldwide

Developped by IBM – Böblingen, European Universities and Helmholtz Partners

Energy

- **Exascale in 2018?**
- **Moores Law might give factor 100 per processor**
- **Need to lower power by factor 10 per processor**


JÜLICH
Sun
microsystems
JUROPA
5

Supercomputer @ Jülich

1956	Erster Computer in Jülich		
1983	Cray XMP/22	0.0004	Teraflop/s
1989	Cray YMP	0.003	Teraflop/s
1996	Cray T3E	0.8	Teraflop/s
2003	IBM p690	9	Teraflop/s
2006	BGL: JUBL	46	Teraflop/s
2008	BGP: JUGENE	223	Teraflop/s
2009	JuRoPA	200	Teraflop/s
	HPC-FF	100	Teraflop/s
	BGP: Peta	1000	Teraflop/s

?

SCALABILITY

10^{18}

Did we forget what Amdahl and Gustafson have told us?

16. February 2011 | Thomas Lippert

1967: Gene Amdahl

- **Validity of the single processor approach to achieving large scale computing capabilities**
- **Gene M. Amdahl**
- **IBM Sunnyvale California**

Gene Amdahl - 1967

$$S = \frac{s + p}{s + \frac{p}{N}}$$

Notation

- *S:* speedup compared to single core
- *N:* # of parallel cores
- *s*: portion computed sequentially
- *p*: portion computed in parallel
- *s*: *O(1)* concurrency
- *p*: *O(N)* concurrency

$$s_r = \frac{s}{s+p} \qquad p_r = \frac{p}{s+p}$$

$$s_r + p_r = 1$$

$$S = \frac{1}{1 - p_r + \frac{p_r}{N}}$$

Maximum Speed Up

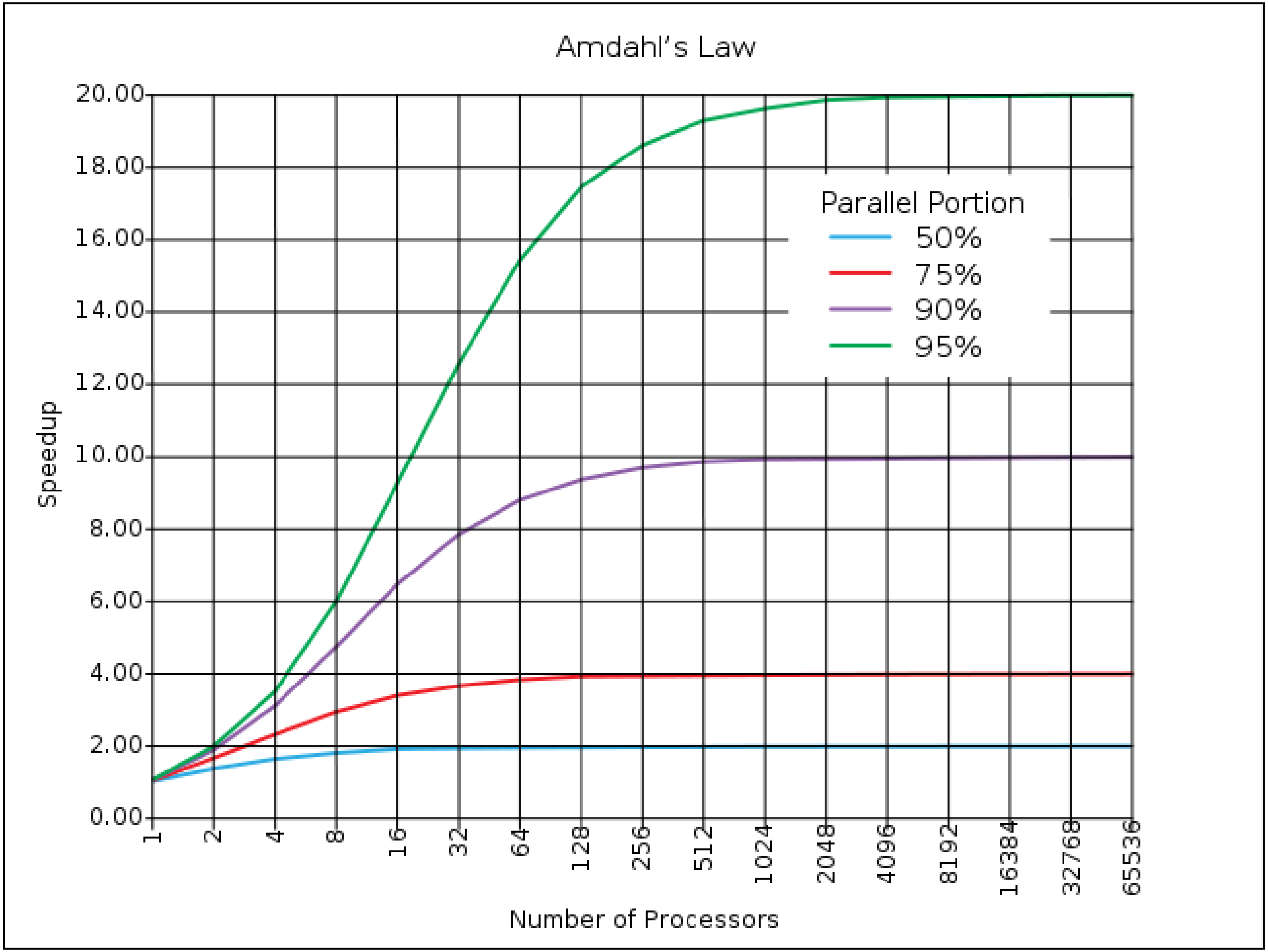
$$N \to \infty \qquad S \to \frac{1}{s_r}$$

s_r must be as small as possible

$$s_r = \frac{1}{2} \to S = 2$$

s_r=1/2

Cores (N)	Speedup Factor	
1	1.00 (baseline)	sequential · potentially O(N) parallelizable · sequential
2	1.33	sequential · core 1 / core 2 · sequential
4	1.60	sequential · core 1 / core 2 / core 3 / core 4 · sequential
∞	2.00	sequential · sequential


Amdahl's Law
Speedup
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
Parallel Portion
50%
75%
90%
95%
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
Number of Processors

SOLUTION ?

Strategies to Break the Speed Limit

s **and** ***p*** **are no fixed quantities!**

1. **Gustafson's Law (1988)**
   - Increase the work done in *O(N)* concurrency (only!!!)
   - → increase *p*
2. **Add or go to other *O(N)* parallelizable features**
   - Switch to different computational model
3. **Optimize for *O(K)* concurrency portions**
   - Many problems are *O(K)* dominated not *O*(1)

1. Gustafson 1988

- **Runtime, not problem size is the constraint of a computational scientist**
- **On larger machines, we work on larger problems**
- **Weak scaling**
- **In contrast Amdahl keeps the workload fixed → strong scaling**
- **Total work *w* (# of cycles) to be done in a fixed time on *N* cores is accordig to Gustafson's model**

$$w = s + pN$$

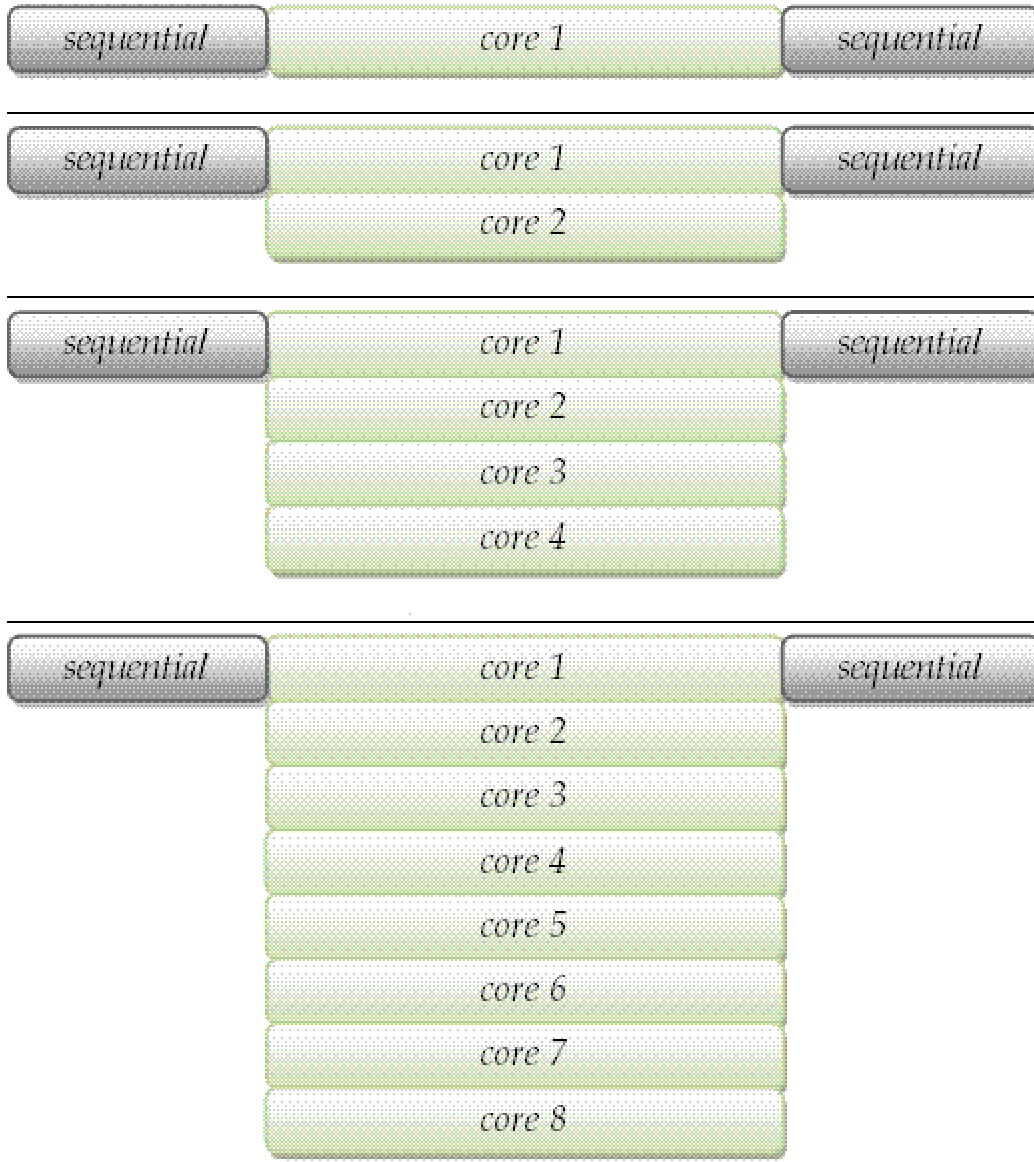
- ***s* not to be increased with *N***

$$S = \frac{s + Np}{s + p}$$

$$S \xrightarrow[N \to \infty]{} N$$

s_r=1/2

Cores (N)	Total Work
1	1.0 (baseline)
2	1.5
4	2.5
8	4.5



Pros and Cons

- **Pro**
  - In principle we can do infinite amount of work with
    $N \to \infty$
  - Proof of concept by highly scalable systems (BGP) ??
- **Caveats**
  - Can we just rely on Gustafson for Exascale?
  - Some problems have fixed size, want to execute them faster
  - Nonlinear algorithms: $O(N^3)$ algorithm means that double the concurrency gives only about a 26% increase in problem size
  - For many problems, *s* grows with *N*
  - Maximal problem sizes limited by memory and I/O

JÜLICH
FORSCHUNGSZENTRUM

An Analogy

Amdahl's Law:

- A car is traveling between two cities 60 miles apart
- It has already spent one hour traveling half the distance at 30 mph
- No matter how fast you drive the last half, it is impossible to achieve 90 mph average before reaching the second city
- Since it has already taken you 1 hour and you only have a distance of 60 miles total; going infinitely fast you would only achieve 60 mph.

Gustafson's Law:

- Suppose a car has already been traveling for some time at less than 90 mph.
- Given enough time and distance to travel, the car's average speed can always eventually reach 90 mph, no matter how long or how slowly it has already traveled.
- For example, if the car spent one hour at 30 mph, it could achieve this by driving at 120 mph for two additional hours, or at 150 mph for an hour

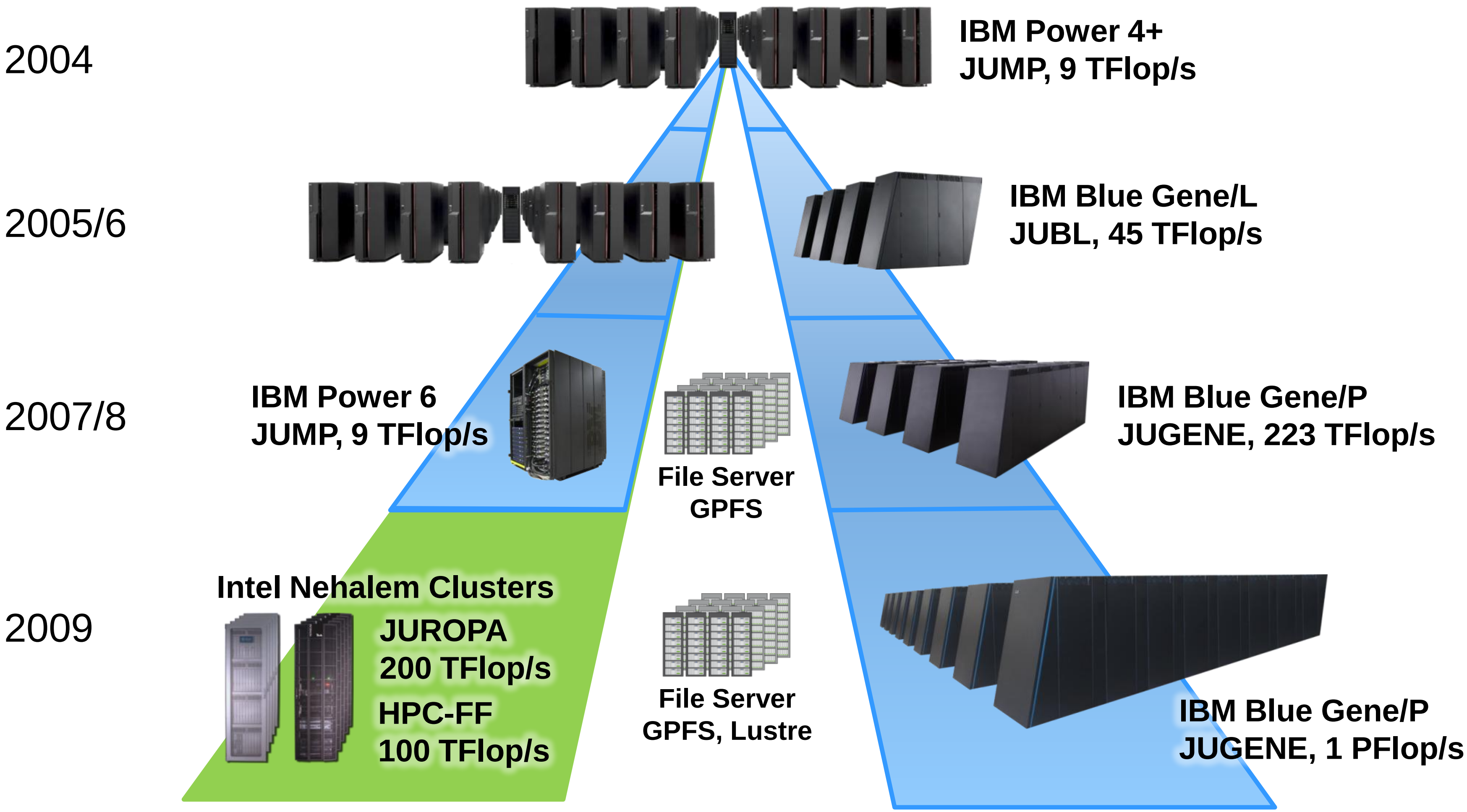
2. Switch to Models with more *O(N)* Cycles

- **Those models and theories will win which have better efficiency on Exascale systems ?**
- **Computer development will certainly transform science and engineering**
- **This is not just an algorithmic problem!**
- **We are already deep within this process**

3. Reduce *s* by Means of *O(K)* Concurrency

- Even purely sequential programs can be (moderately) parallelized by injecting *O(K)* concurrency in the form of caching, prefetching, instruction reordering, and pipelining
- Many problems' worst scaling is in fact *O(K)* and not *O(*1*)*
- What are the consequences as to Amdahl's Law?

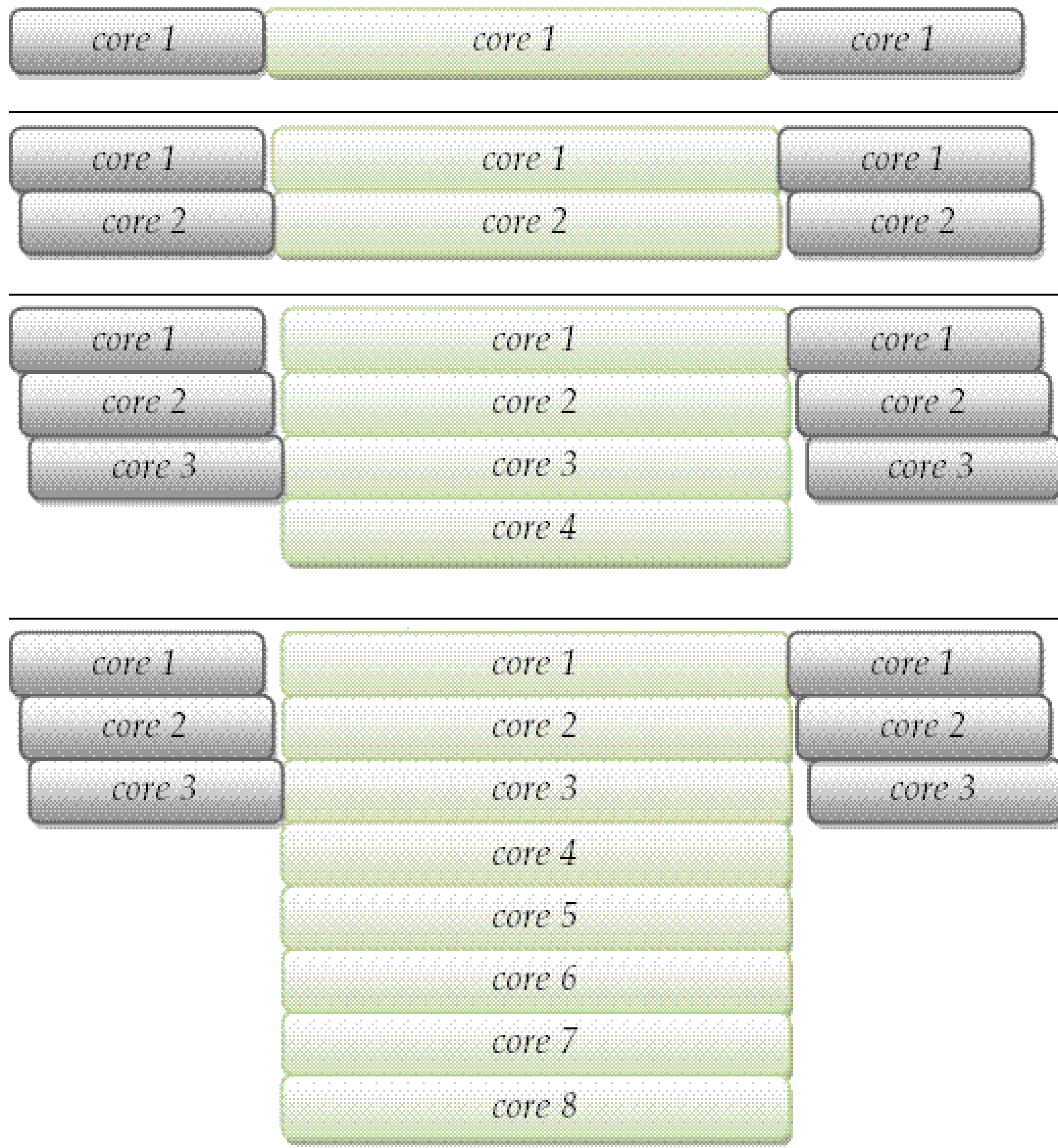
Jülich Dual Concept: Distinguish Between K-Concurrency and N-Concurrency on Code Basis



Speed Up for *O(K)* Concurrency

$$S = \frac{1}{\frac{1-p_r}{K} + \frac{p_r}{N}}$$

Cores (N)	Total Work
1	1.0 (baseline)
2	~2.0
4	~3.5
8	~5.5



Introduce Difference in Effective Core Speed: *f*

$$S = \frac{1}{\frac{1-p_r}{Kf} + \frac{p_r}{N}}$$

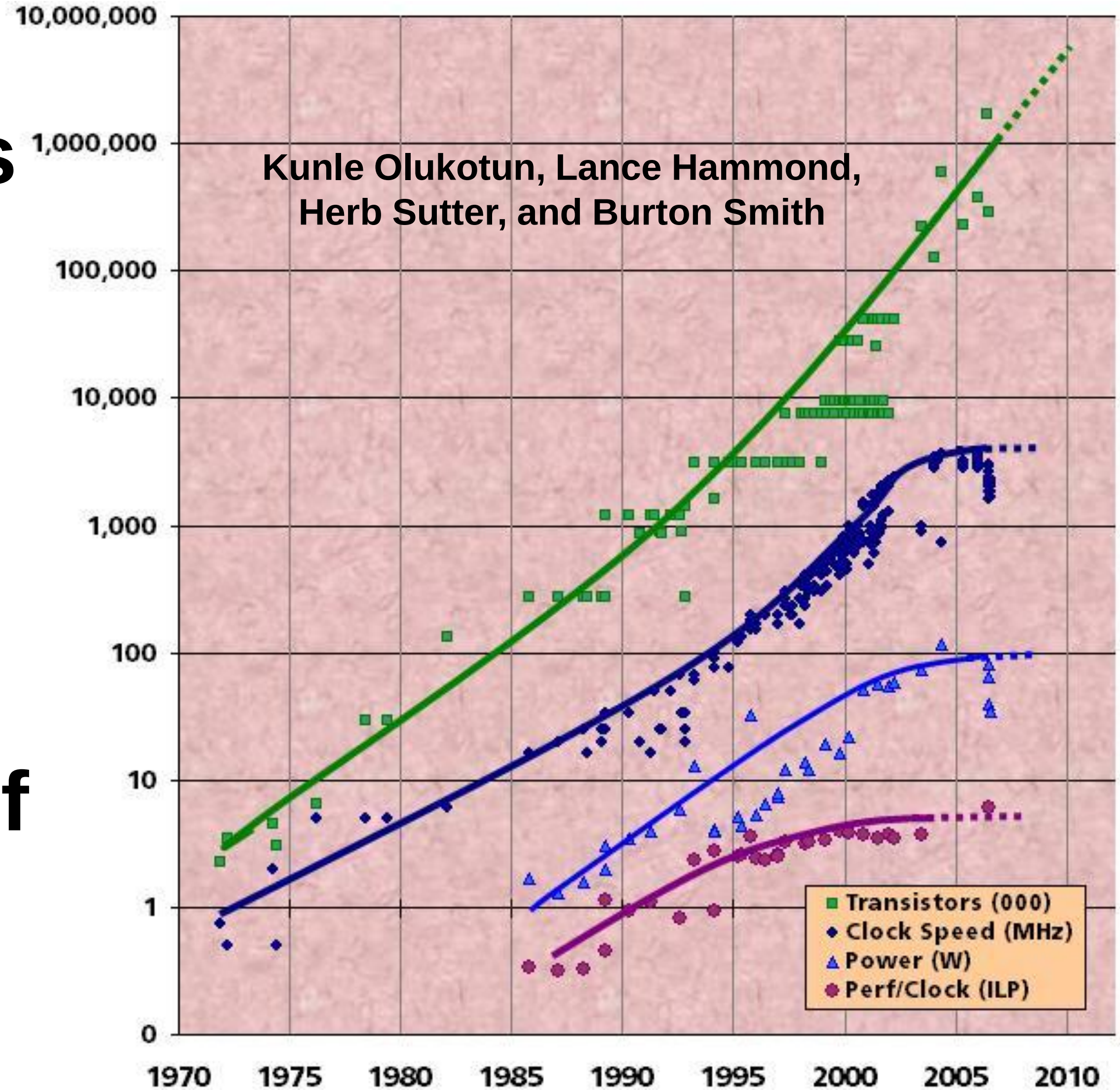
p_r=.50, N = 500.000, K = 10.000, f = 1: S = 20.000

p_r=.95, N = 500.000, K = 10.000, f = 4: S =320.000

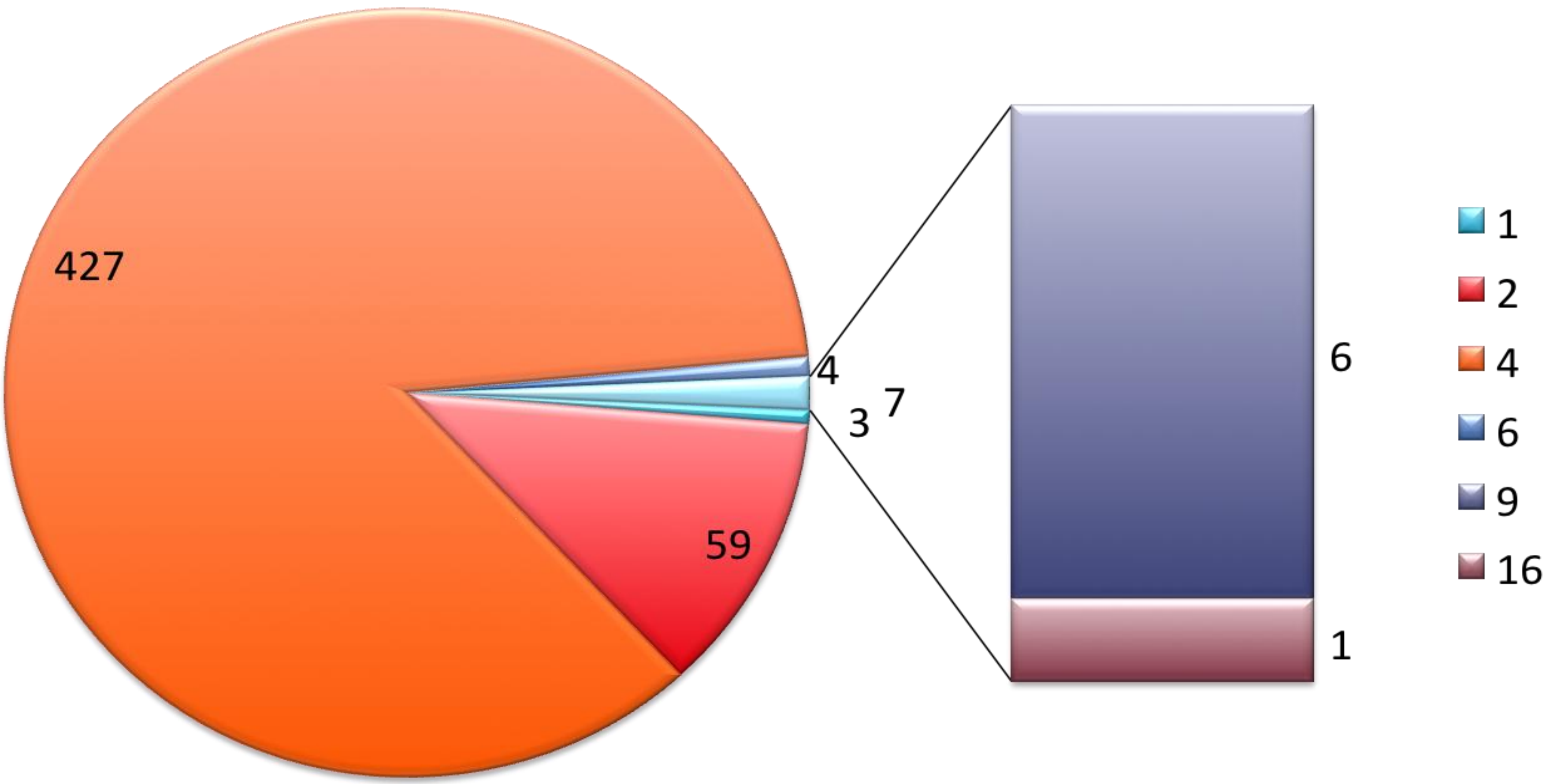
ENERGY CONSUMPTION

Exponential Growth

- # of transistors doubles in 18 month period
- Frequency increase stalled
- → increase # of cores



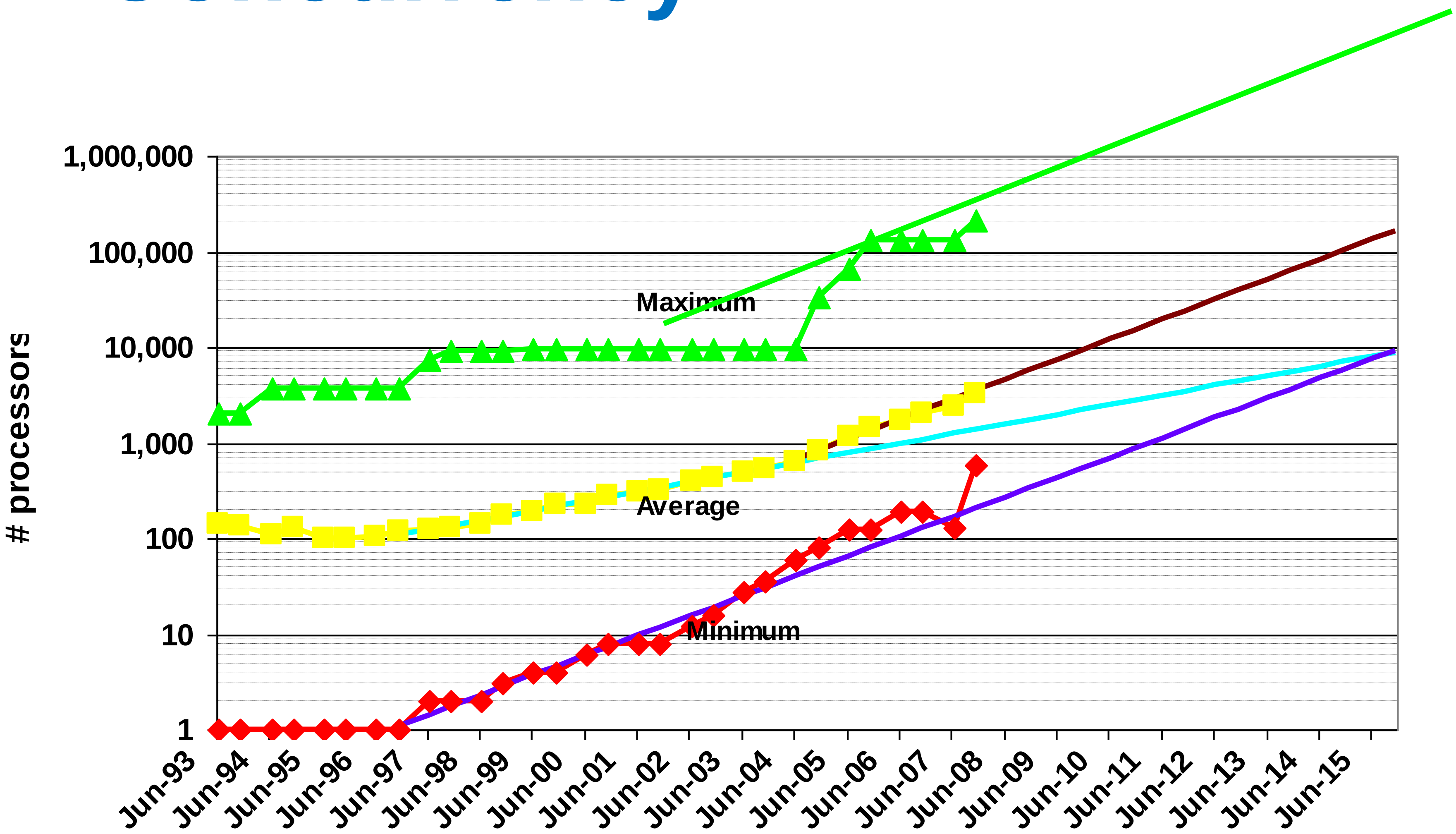
(Multi-Core)-Cores per Socket (Nov. 2009)



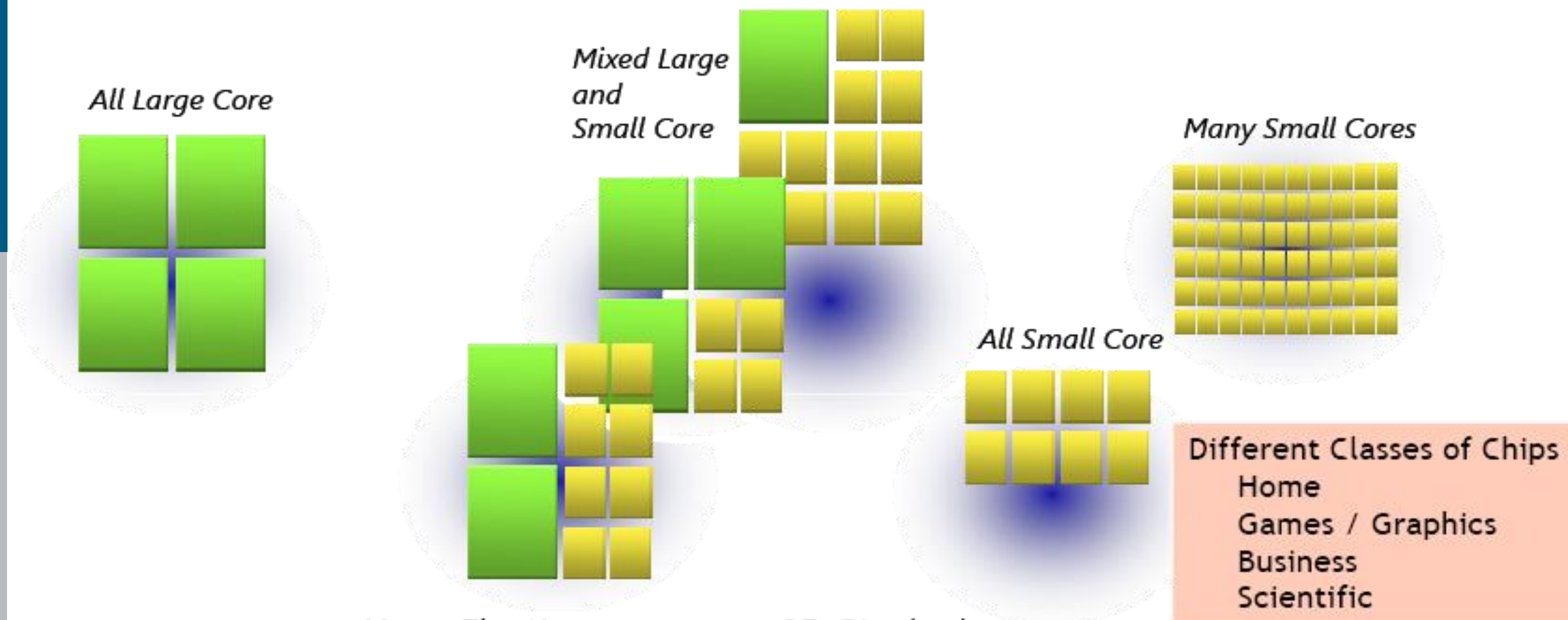
Power Envelope

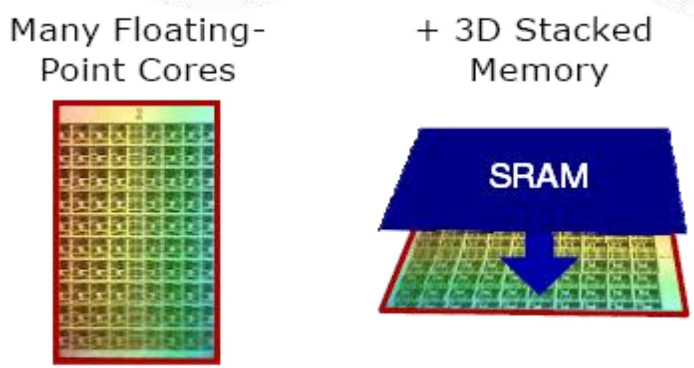
- **Processor power consumption has come to a saturation at about O(100) W**
  → dictated by desktop and laptop systems
- **Saturation of frequencies → increase concurrency**
  → multi core
- **Core sizes sufficient for HPC might be smaller than „standard cores"**
  → HPC will profit from many core
- **# of sockets per system has increased to < O(10.000)**
  → dictated by power budget of SC centres

Concurrency



Varieties

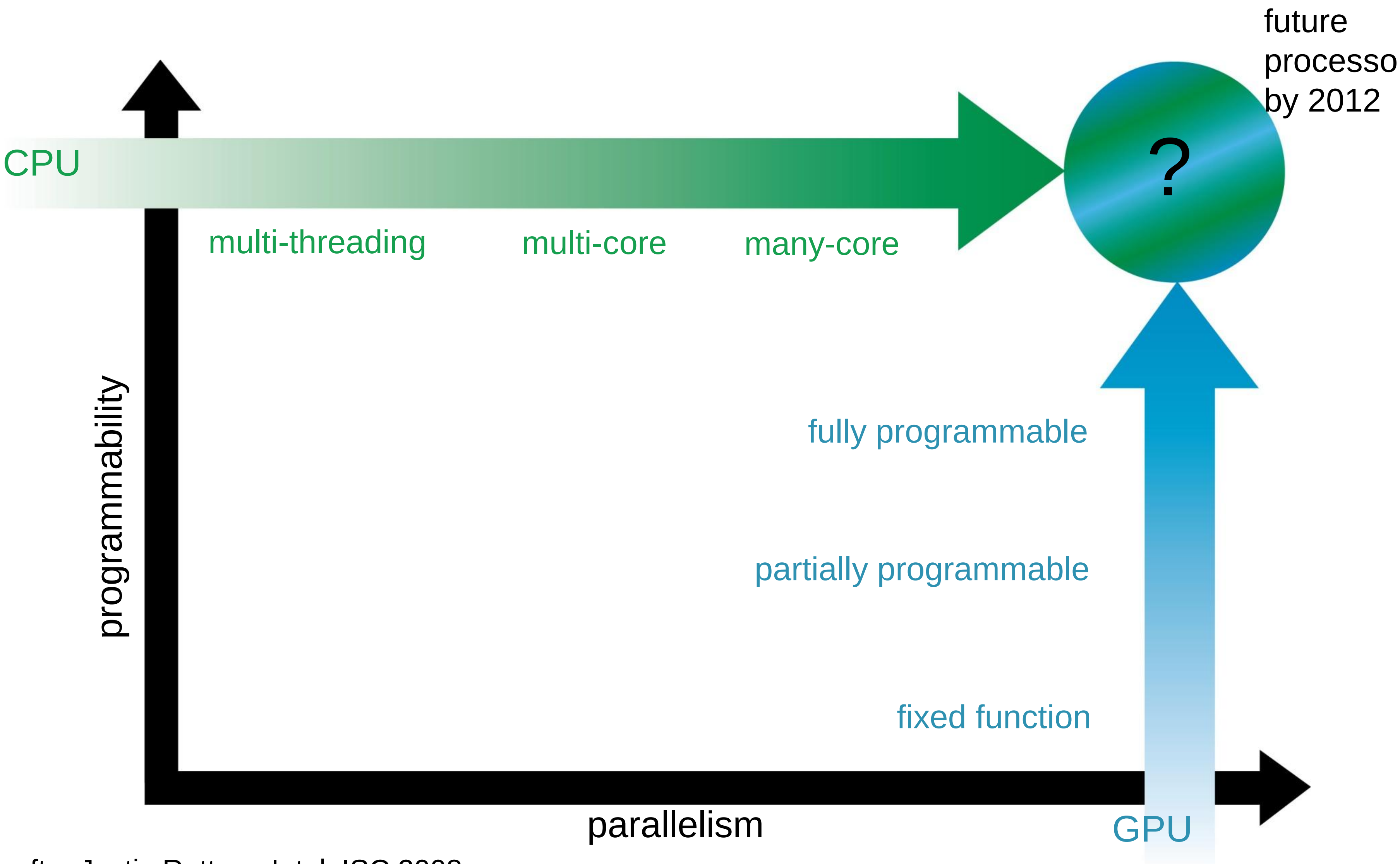




The question is not whether this will happen but whether we are ready

Source: Jack Dongarra, ISC 2008

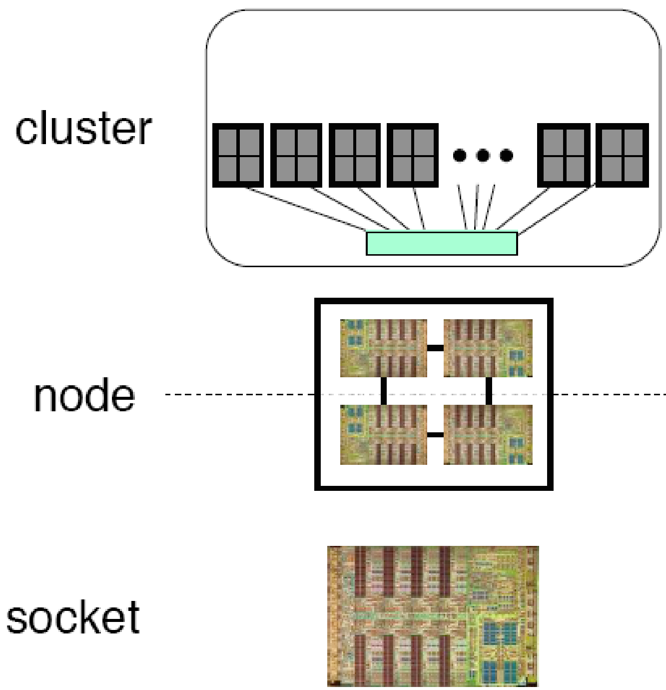
Vision 2008



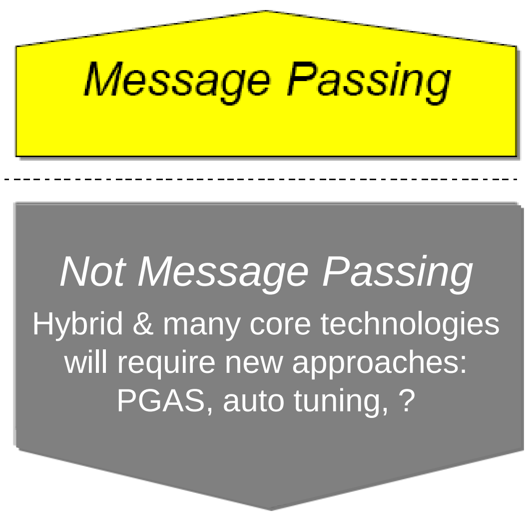
after Justin Rattner, Intel, ISC 2008

Is this the future ?

System: cluster + many core node



Programming model:
MPI+X



after Don Grice, IBM, Roadrunner Presentation,
ISC 2008

JuRoPA + HPC-FF:
Can we continue the cluster model?

Cluster computer
Bull NovaScale R422-E2
1080 nodes, 8640 cores
101 TF peak, Intel Nehalem
24 GB memory
Infiniband QDR (Mellanox)
ParaStation Cluster-OS
HPC for Fusion

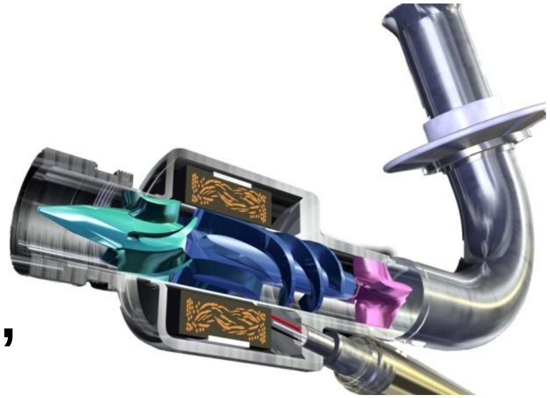
Cluster computer
SUN-blades
2208 nodes, 17664 cores
207 TF peak, Intel Nehalem
48 GB memory
Infiniband QDR (SUN M9)
ParaStation Cluster-OS
General Purpose HPC

Future of High-end Cluster Computing

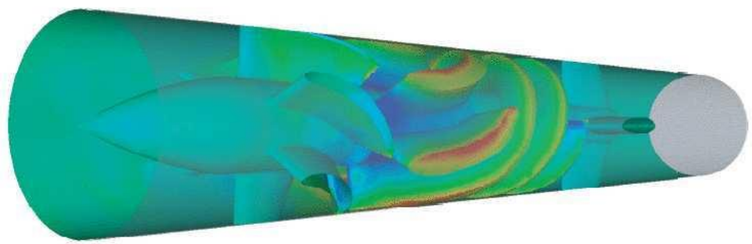
- **Standard processor speed will increase by about a factor of 4 to at most 8 in next 4 years…**
  - → Clusters need to utilize accelerators to reach Exascale
  - Current accelerators not parallelized on the node-level
  - Programming very cumbersome
  - Integrated processors expected after 2015…
- **Clusters going Exaflop/s will require virtualization elements in order to raise resilience and reliability.**
  - → Virtualization software layer
- **Flexibility**
  - Have to tolerate over/under subscription
  - Requirement of fault tolerance if accelerator fails

EXASCALE STRATEGIES

CFD: Simulation of Blood Flow in a Ventricular Assist Device (VAD)

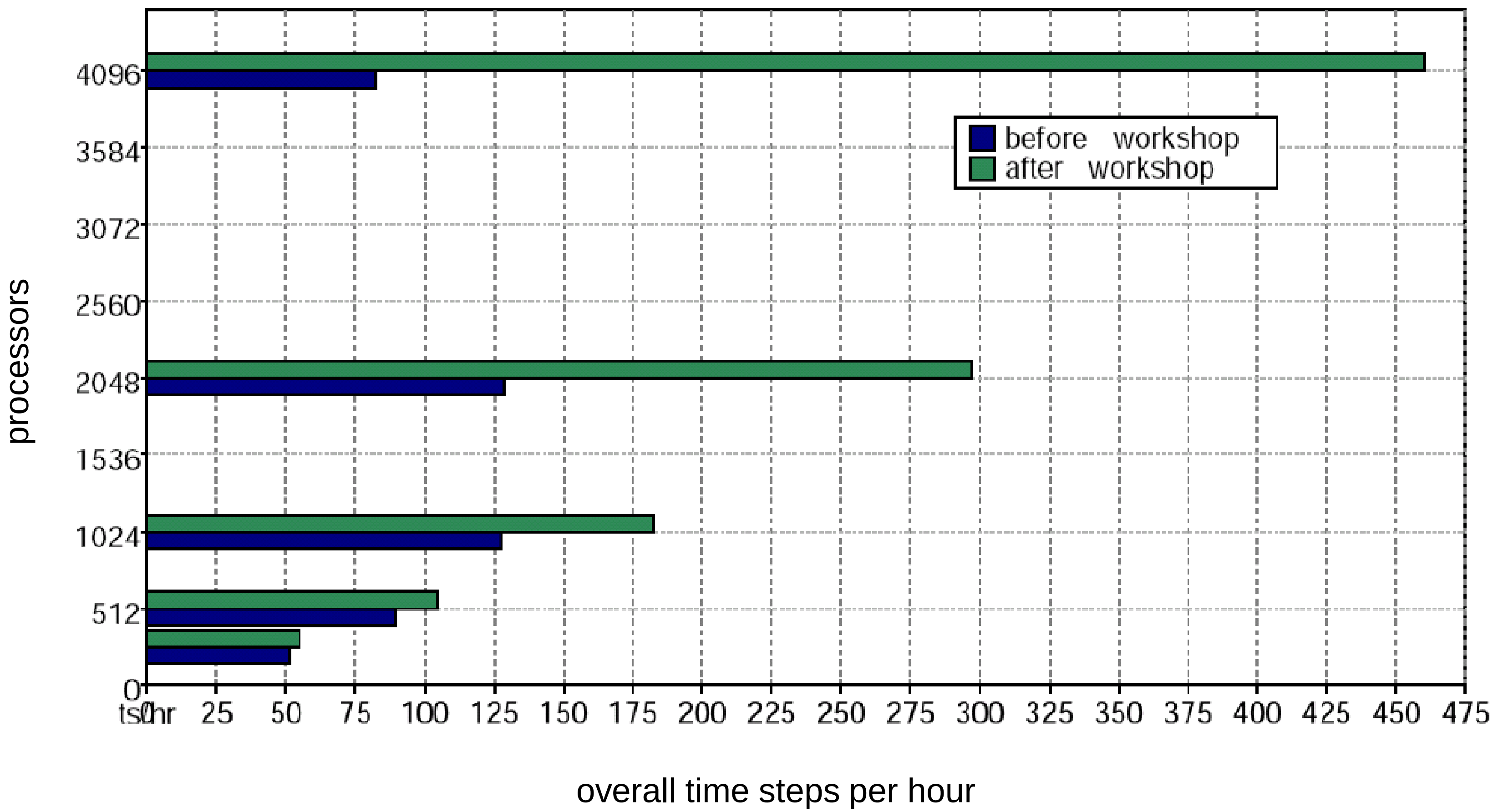


- **Research Area: CFD**
- **Code: Finite Element techniques, Distributed Memory Code (distribution of subdomains Fortran90 and C / MPI**



  - Simulation of unsteady fluid flows
  - Major problem:
    scalar communication bottlenecks

Scalability Pattern of the Underlying CFD Code

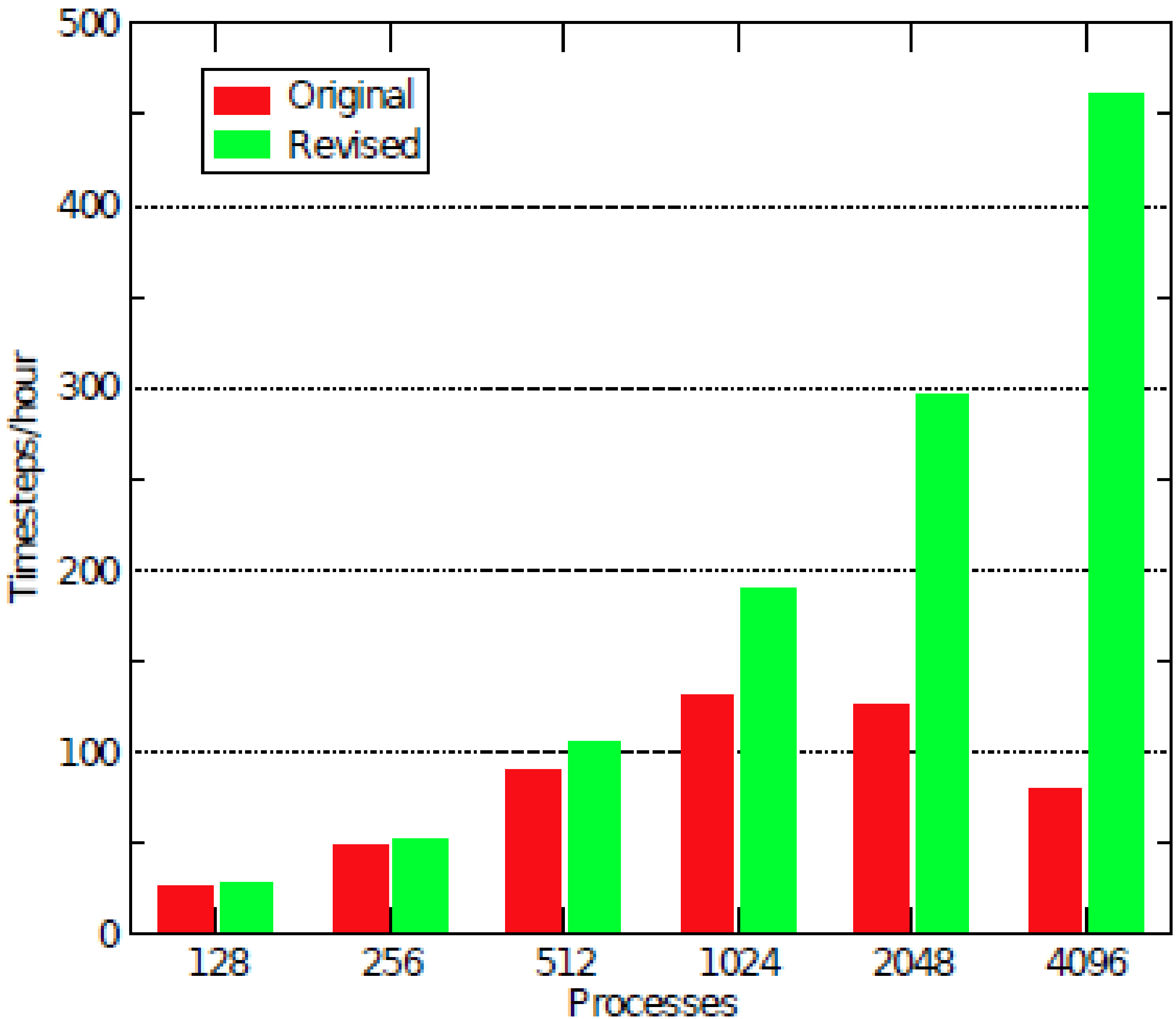


O(K) Concurrency on O(N) System

- **Running an O(K) concurrent code part on an O(N) parallel system**
  - Only use O(K) processors

    → Efficiency goes with K/N
  - Distribute on O(N) processors

    → Efficiency becomes even worse in many cases



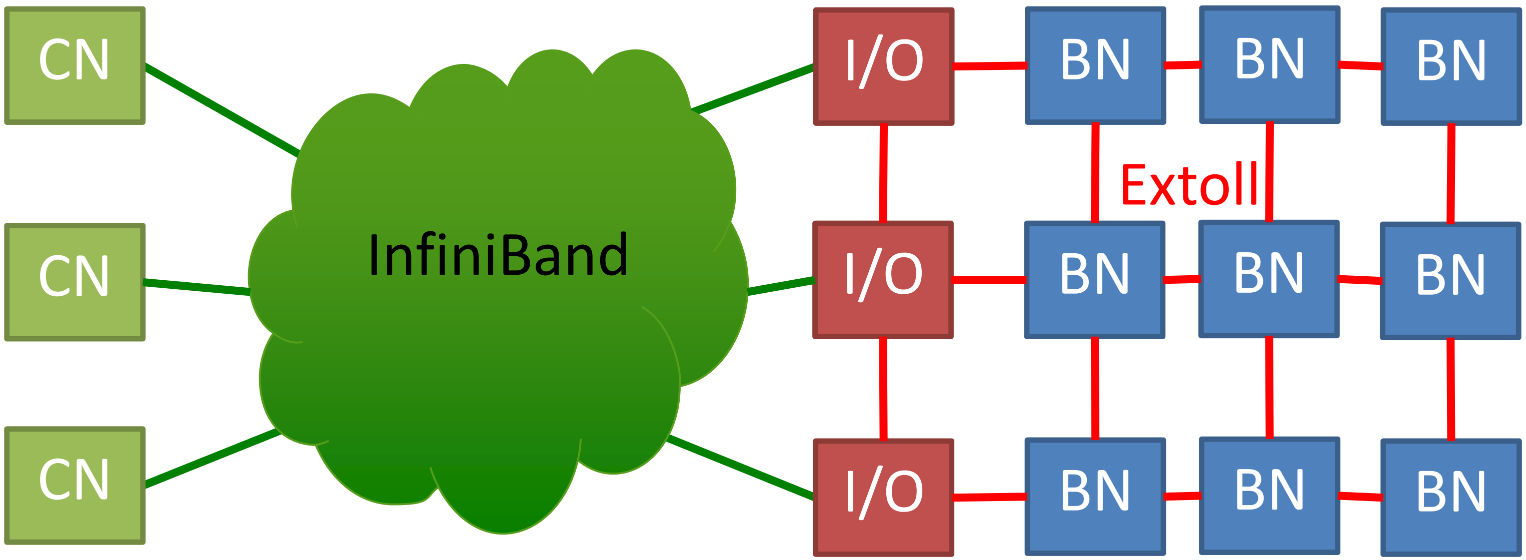
Original
Revised
Timesteps/hour
0
100
200
300
400
500
128
256
512
1024
2048
4096
Processes

Recipe

- **Choose *N* as large as necessary**
- **Enlarge fraction p_r? → Gustafson's Law**
- **Try to separately execute K-concurrency and N-concurrency complex code kernels**
  - On connected *K* and *N* architectures → Overlap K and N kernels
  - On an *N* architecture exploit internal storage

1. Spatial Break UP

Cluster-Booster Architecture (*K*--*N* Concurrency)



- Complex kernels to be offloaded are expected to have regular communication patterns
- → Data exchange expected to scale better than booster part

→ DEEP EXASCALE PROPOSAL

BOSTER Advantages

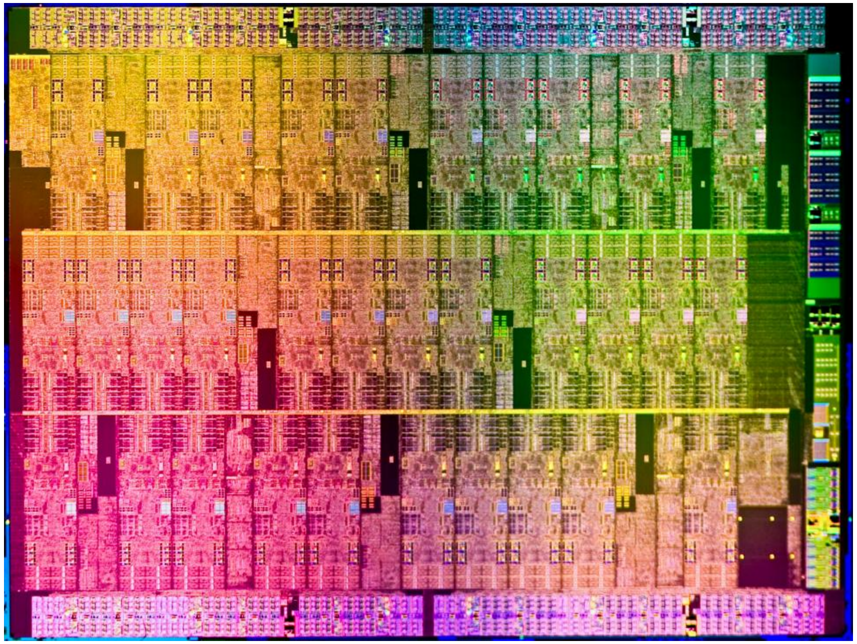
- **Dynamic and static BN-to-CN assignment**
- **Virtualization of cluster not hampered**
- **Exploit accelerator parallelism**
- Accelerator allocation follows application needs
- Fault tolerance in case of accelerator failure
- Potential for O(100) PF in 2015

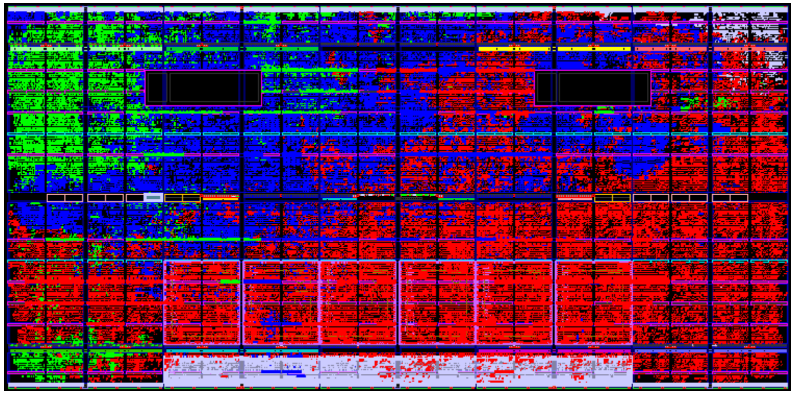
Requirements and Tasks

- BN-nodes should follow existing programming models to guarantee continuity
- IB network extension required
- Specific very fast network among accelerators required
- Specific boards for booster to be developed
- Enabling middleware layer, math libraries, compiler technology required

Technology Components for Cluster-Booster

- **Intel Knights Corner > 50 Core Server Chip**
  - > 1 TF
  - 100 PF = > 5 mio cores
- **EXTOLL (for booster)**
  - 120 Gbit per link unidir
  - 1440 Gbit/card bidir, 3d
  - 0.3 µs latency
- **Mellanox IB (for cluster)**
  - State-of-the-art interconnect
- **ParaStation cluster OS**
- **Intel Compiler and Tools**





2. Temporal Break UP

Architecture (*N* -- *nK* Concurrency)

IBM
ACTIVE
STORAGE
WITH
BLUE GENE /Q

→AXIO
EXASCALE PROPOSAL

- Several codes run in sequence
- Fast local storage allows swapping
- Different *N*-concurrency code portions run in sequence
- Different *K*-concurrency code portions run at the same time

CONCLUSION

- **Amdahl's Law refers to a fixed problem size and an *O(*1*)* lower concurrency**
- **Most problems show *O(K)* lower concurrency**
- **Larger problems are run on larger systems**
  → Gustafson's Law
- **Two solutions proposed:**
  - Spatial break up on Cluster-Booster → DEEP
  - Temporal break up on highly scalable system → AXIO
- **Two conceptual paths towards the Exascale**