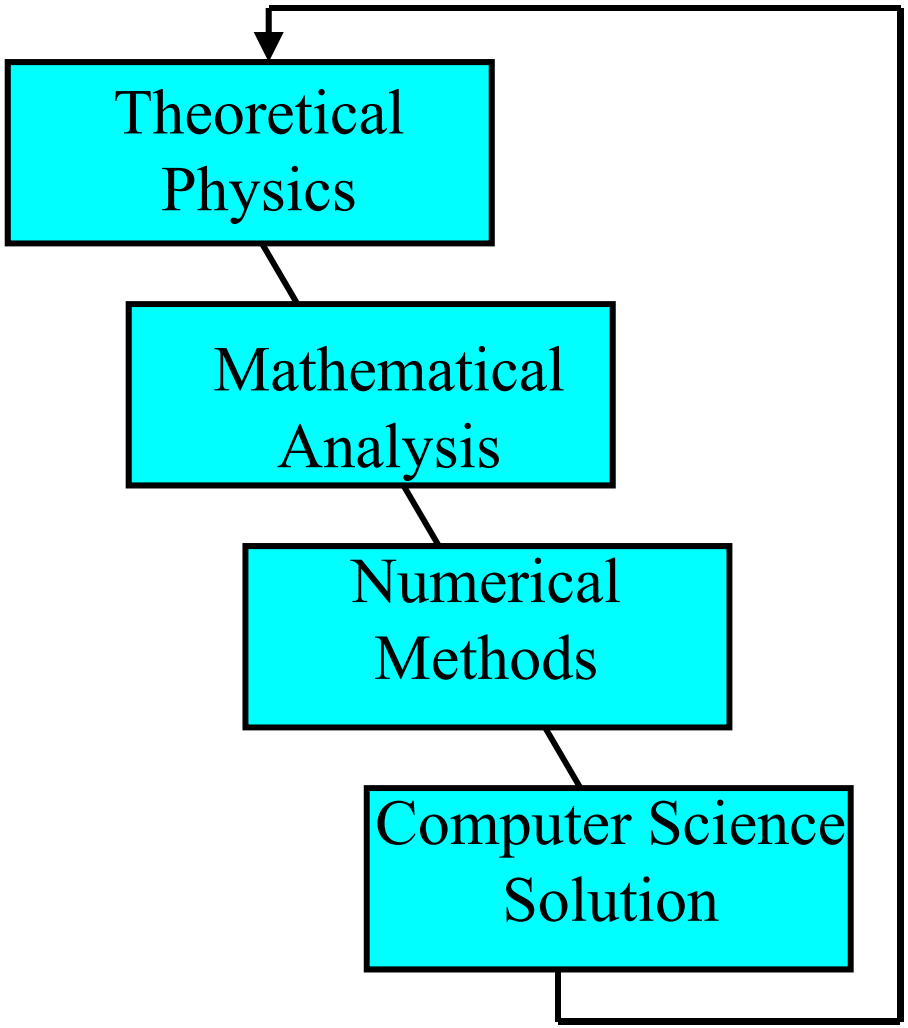
Parallel Programming

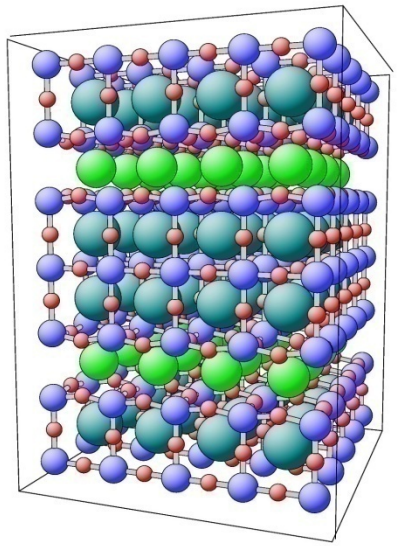
Part 1:
Introduction

foils by R. Buyya, F. Ercal, T. Fahringer, I. Foster, M. Gerndt, W. Jalby, B. Wilkinson

Parallel Processing: The Origins

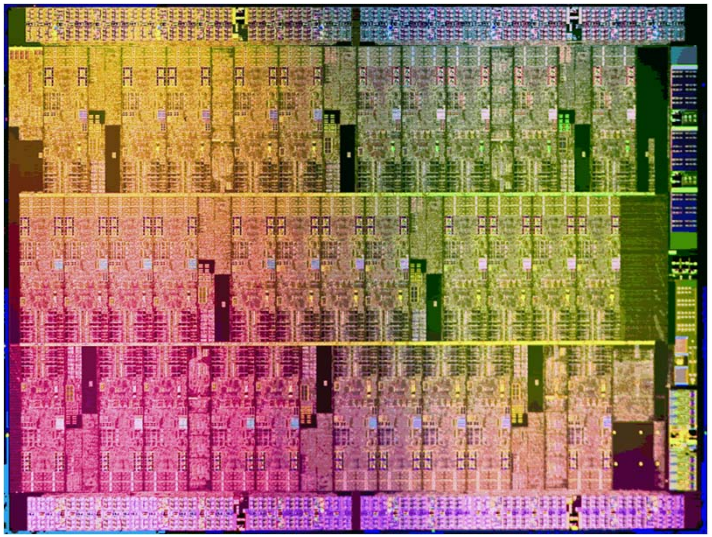


- Electronic Structure Computation
  F(C)C = SCE
- Ab initio molecular dynamics
- Density Functional Theory
- MPI version
- Limited to Scientific Applications that required lots of computations and data.
- for brave people

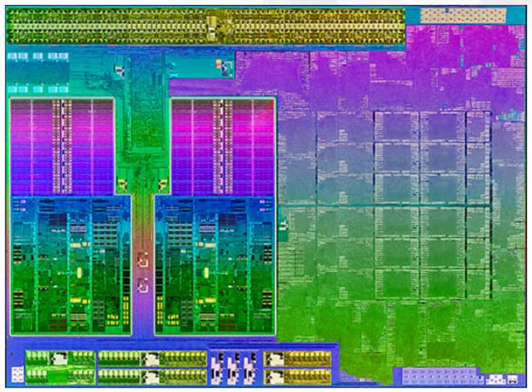


Parallel Processing Becomes Mainstream

- Parallel processing became mainstream
  - Sony Playstation 3 (2005)
- Multi-core architectures
  - Nr. of transistors double every 12 to 18 months



Intel MIC processor with 60+ x86 cores

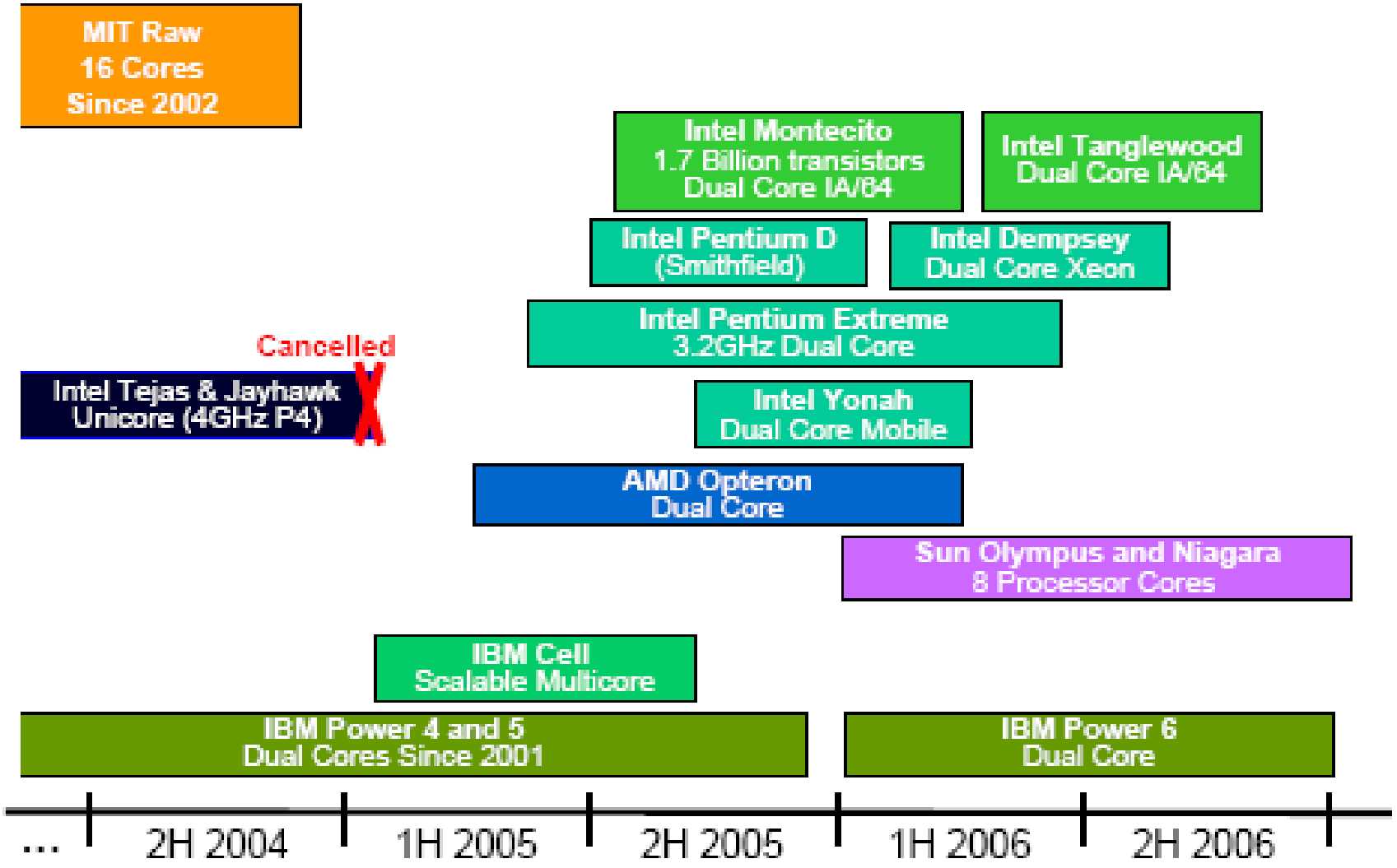


AMD Trinity APU with 4 CPU cores and 384 stream processors

Why Parallel Computers?

- Performance growth of single processors cannot be sustained indefinitely:
  - speed of light
  - limits of miniaturization (nr. of transistors per chip)
  - economic limitations
  - thermodynamics
  - DRAM latency
  - wire delays
  - diminishing returns on more instruction level parallelism
- Connecting multiple off-the-shelf processors is a logical way to gain performance beyond that of a single processor.
- Or putting multiple cores onto a single chip

Unicores are on the verge of extinction
Multicores dominate



Grand Challenge Problems

One that cannot be solved in a reasonable amount of time with today's computers. Obviously, an execution time of 10 years is always unreasonable.

Examples

- Modeling large DNA structures
- Global weather forecasting
- Modeling motion of astronomical bodies.

Motivating Example: Modeling Motion of Astronomical Bodies

- Each body attracted to each other body by gravitational forces. Movement of each body predicted by calculating total force on each body.

- With *N* bodies, *N* - 1 forces to calculate for each body, or approx. N^2 calculations. ($N \log_2 N$ for an efficient approx. algorithm.)

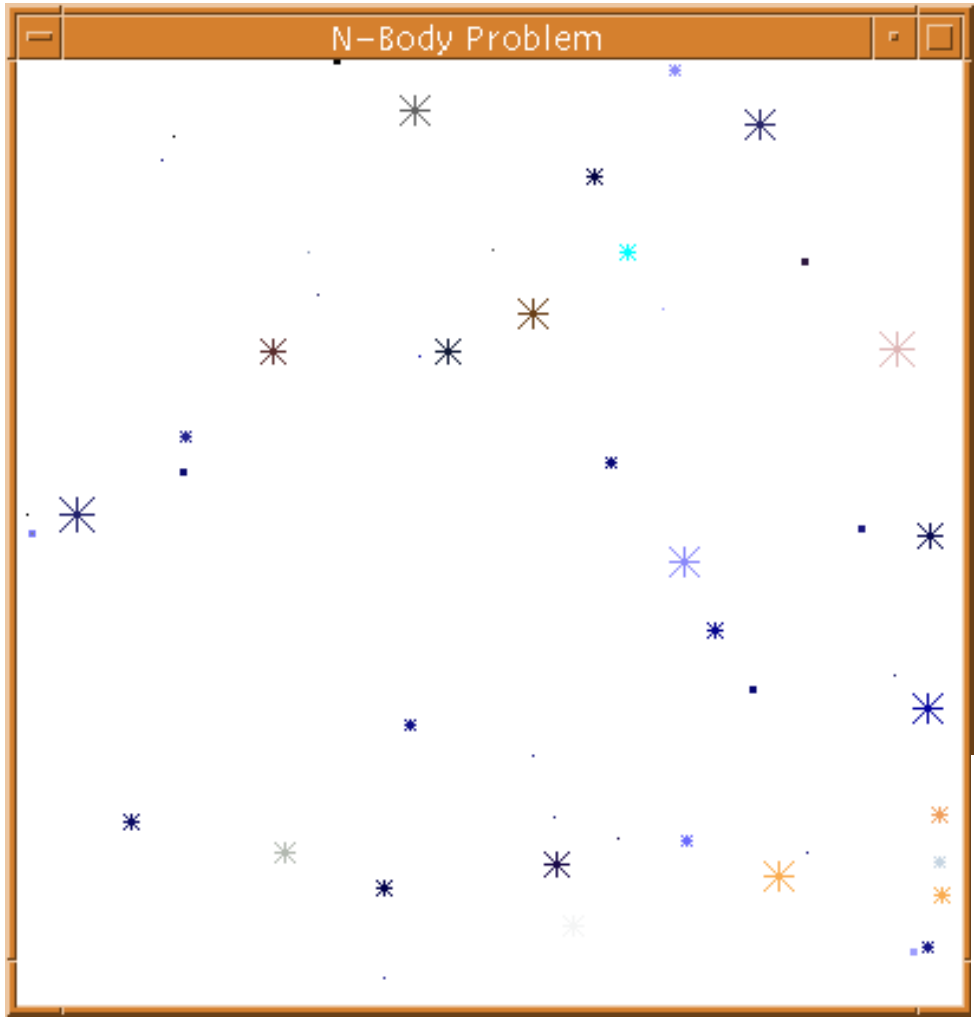
- After determining new positions of bodies, calculations repeated.

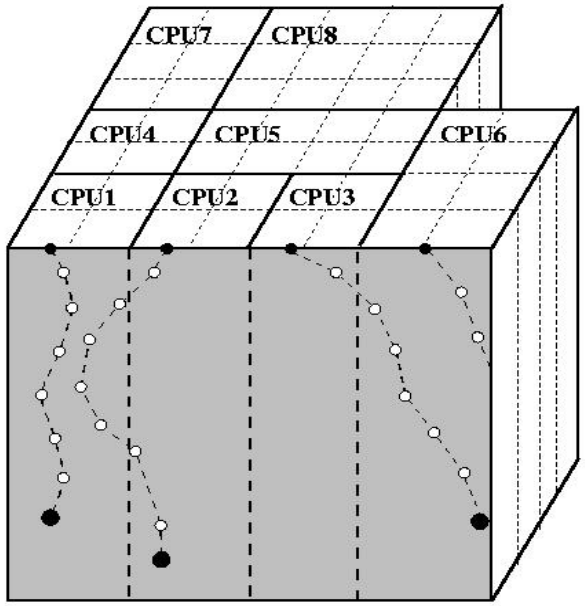
- A galaxy might have, say, 10^{11} stars.

- Even if each calculation done in 1 ms (extremely optimistic figure), it takes 10^9 years for one iteration using N^2 algorithm and almost a year for one iteration using an efficient $N \log_2 N$ approximate algorithm.

- a difficult problem to parallelize

- Scaling is very hard





Why use Parallel Computing?

- save time – wall clock time
- overcoming memory constraints
- solve larger problems
- possible better fault tolerance
- cost savings
- scientific interest

Speedup Factor

$$S(p) = \frac{\textit{Execution time with 1 processor}}{\textit{Execution time with n processors}} = \frac{t_s}{t_p}$$

where t_s is execution time on a single processor and t_p is execution time on a multiprocessor.

$S(p)$ gives increase in speed by using multiple processors.

Use best sequential algorithm with single processor system instead of parallel program run with 1 processor for t_s. Underlying algorithm for parallel implementation might be (and is usually) different.

Maximum Speedup

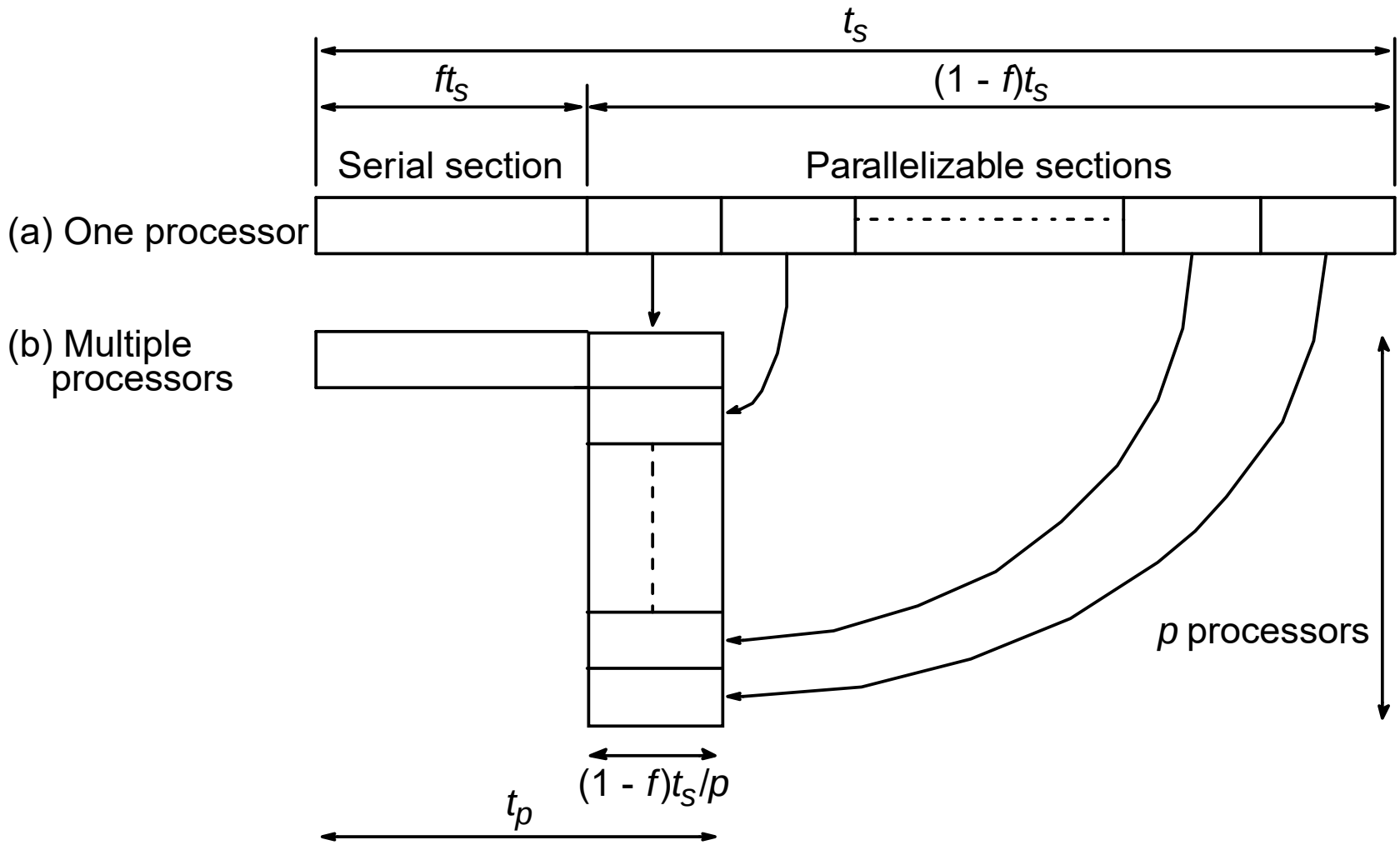
Maximum speedup is usually p with p processors (linear speedup).

Possible to get superlinear speedup (greater than p) but usually a specific reason such as:

- Extra memory in multiprocessor system
- Nondeterministic algorithm

Maximum Speedup

Amdahl's law

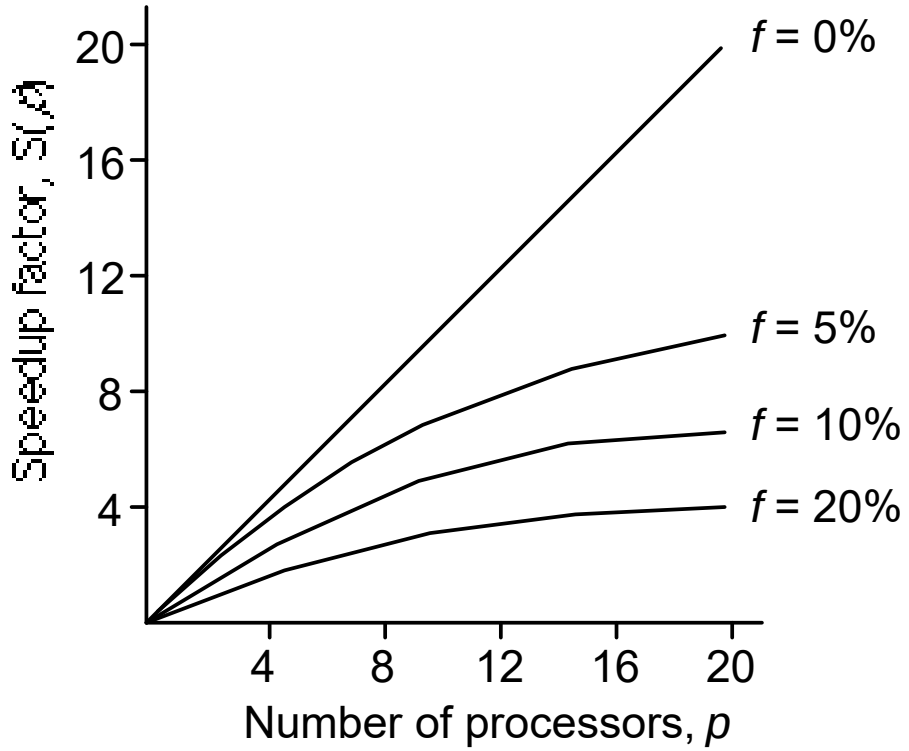


Speedup factor is given by:

$$S(p) = \frac{t_s}{ft_s + (1-f)t_s/p} = \frac{p}{1 + (p-1)f}$$

This equation is known as Amdahl's law

Speedup against number of processors

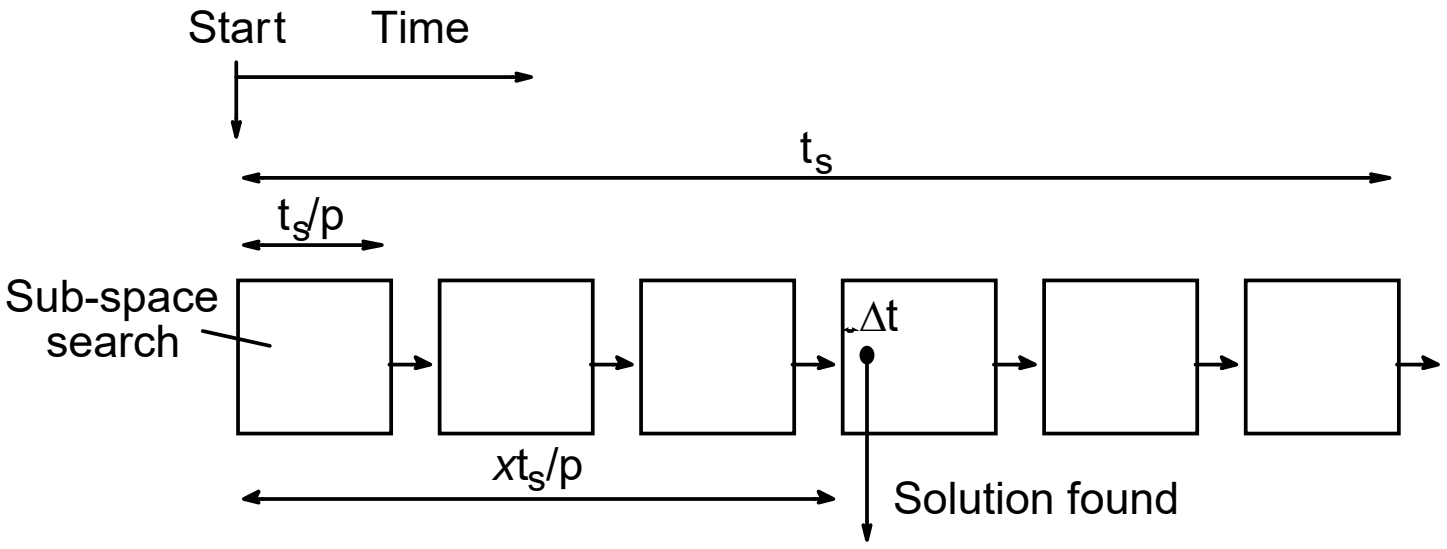


Even with infinite number of processors, maximum speedup limited to 1/f .

Example: f = 5% => maximum speedup is 20,
f = 10 % => maximum speedup is 10,
f = 20 % => maximum speedup is 5
irrespective of number of processors.

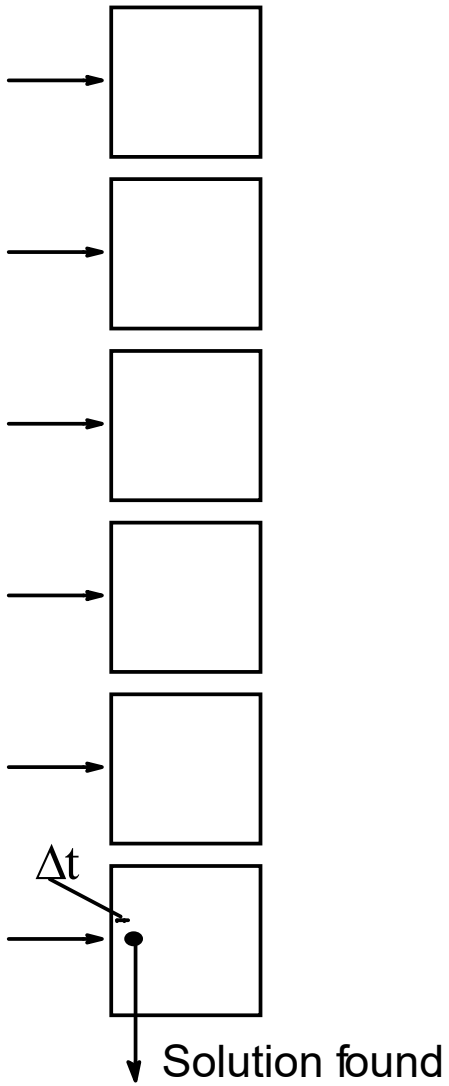
Superlinear Speedup example - Searching

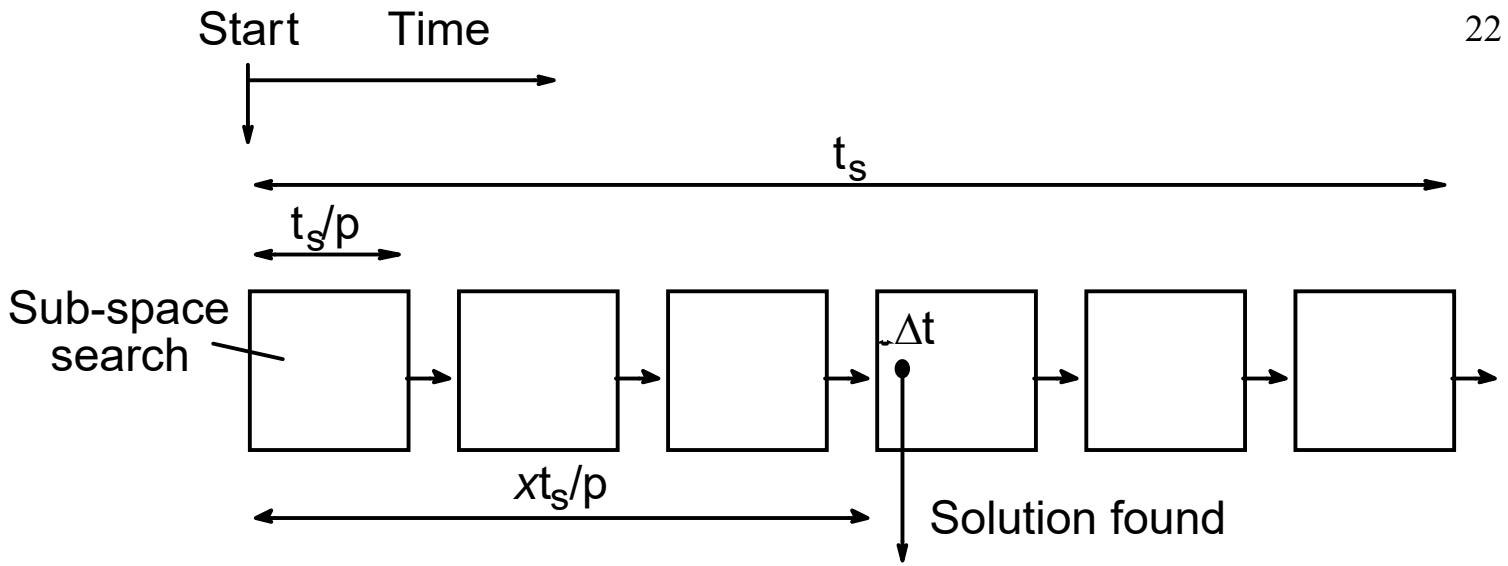
(a) Searching each sub-space sequentially



x indeterminate

(b) Searching each sub-space in parallel





x indeterminate

Speed-up then given by

$$S(p) = \frac{x\frac{t_s}{p} + \Delta t}{\Delta t}$$

Worst case for sequential search when solution found in last sub-space search. Then parallel version offers greatest benefit, i.e.

$$S(p) = \frac{\frac{p-1}{p} t_s + \Delta t}{\Delta t} \rightarrow \infty$$

as Δt tends to zero with p as low as 2

Least advantage for parallel version when solution found in first sub-space search of the sequential search, i.e.

$$S(p) = \frac{\Delta t}{\Delta t} = 1$$

Actual speed-up depends upon which subspace holds solution but could be extremely large.

Hardware Diversity 2022

Intel processors

Alder Lake chips will allow users to max turbo boost the processor up to 5.2 GHz and as many as 16 cores and 24 threads.

AMD RYZEN™ IN 2022

CONSUMER NOTEBOOK PROCESSOR LINEUP

H-SERIES

FOR GAMERS & CREATORS

AMD Model			Cores/ Threads	Max Boost (Base)	L2+L3 Cache	GPU Cores (Max Boost)	Node	TDP
RYZEN™ 9 6980HX	"Zen 3+"	RDNA 2	8/16	5.0 (3.3)	20MB	12 (2.4GHz)	6nm	45W+
RYZEN™ 9 6980HS	"Zen 3+"	RDNA 2	8/16	5.0 (3.3)	20MB	12 (2.4GHz)	6nm	35W
RYZEN™ 9 6900HX	"Zen 3+"	RDNA 2	8/16	4.9 (3.3)	20MB	12 (2.4GHz)	6nm	45W+
RYZEN™ 9 6900HS	"Zen 3+"	RDNA 2	8/16	4.9 (3.3)	20MB	12 (2.4GHz)	6nm	35W
RYZEN™ 7 6800H	"Zen 3+"	RDNA 2	8/16	4.7 (3.2)	20MB	12 (2.2GHz)	6nm	45W
RYZEN™ 7 6800HS	"Zen 3+"	RDNA 2	8/16	4.7 (3.2)	20MB	12 (2.2GHz)	6nm	35W
RYZEN™ 5 6600H	"Zen 3+"	RDNA 2	6/12	4.5 (3.3)	19MB	6 (1.9GHz)	6nm	45W
RYZEN™ 5 6600HS	"Zen 3+"	RDNA 2	6/12	4.5 (3.3)	19MB	6 (1.9GHz)	6nm	35W

U-SERIES

THIN & POWERFUL MOBILITY

AMD Model			Cores/ Threads	Max Boost (Base)	L2+L3 Cache	GPU Cores (Max Boost)	Node	TDP
RYZEN™ 7 6800U	"Zen 3+"	RDNA 2	8/16	4.7 (2.7)	20MB	12 (2.2GHz)	6nm	15-28W
RYZEN™ 5 6600U	"Zen 3+"	RDNA 2	6/12	4.5 (2.9)	19MB	6 (1.9GHz)	6nm	15-28W
RYZEN™ 7 5825U	"Zen 3"	Vega	8/16	4.5 (2.0)	20MB	8 (1.8GHz)	7nm	15W
RYZEN™ 5 5625U	"Zen 3"	Vega	6/12	4.3 (2.3)	19MB	7 (1.6GHz)	7nm	15W
RYZEN™ 3 5425U	"Zen 3"	Vega	4/8	4.1 (2.7)	10MB	6 (1.5GHz)	7nm	15W

* See endnotes: GD-150.

Unlocked 12th Gen Intel® Core™ Desktop Processors

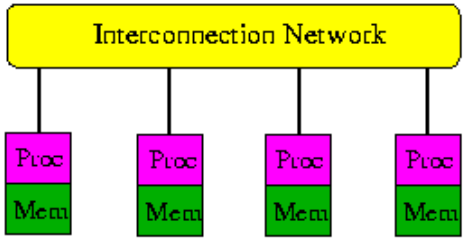
Processor Number	Processor Cores (P+E)[3]	Processor Threads[4]	Intel® Smart Cache (L3)	Total L2 Cache	Processor Turbo Frequency			Processor Base Frequency		Unlocked[1]	Processor Graphics[7]	Total CPU PCIe Lanes	Max Memory Speed[2]	Memory Channels	Maximum Memory Capacity[2]	Processor Base Power (W)	Maximum Turbo Power (W)	RCP Pricing (USD 1K)
					Intel® Turbo Boost Max Technology 3.0 Frequency (GHz)[4]	P-core Max Turbo Frequency (GHz)[5]	E-core Max Turbo Frequency (GHz)[5]	P-core Base Frequency (GHz)[5]	E-core Base Frequency (GHz)[5]									
Socket LGA 1700 Performance																		
i9-12900K	16 (8P + 8E)	24	30MB	14MB	Up to 5.2	Up to 5.1	Up to 3.9	3.2	2.4	√	Intel® UHD Graphics 770	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	241	$589
i9-12900KF	16 (8P + 8E)	24	30MB	14MB	Up to 5.2	Up to 5.1	Up to 3.9	3.2	2.4	√	n/a	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	241	$564
i7-12700K	12 (8P + 4E)	20	25MB	12MB	Up to 5.0	Up to 4.9	Up to 3.8	3.6	2.7	√	Intel® UHD Graphics 770	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	190	$409
i7-12700KF	12 (8P + 4E)	20	25MB	12MB	Up to 5.0	Up to 4.9	Up to 3.8	3.6	2.7	√	n/a	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	190	$384
i5-12600K	10 (6P + 4E)	16	20MB	9.5MB	n/a	Up to 4.9	Up to 3.6	3.7	2.8	√	Intel® UHD Graphics 770	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	150	$289
i5-12600KF	10 (6P + 4E)	16	20MB	9.5MB	n/a	Up to 4.9	Up to 3.6	3.7	2.8	√	n/a	20	DDR5 4800 MT/s DDR4 3200 MT/s	2	128GB	125	150	$264

AMD processors

AMD Ryzen 6000H series is entirely made of Zen3+ SKUs. The lists featured up to Ryzen 9 6980HX with a boost clock up to 5.0 GHz and TDP at 45W+.

Parallel Architectures

Distributed Memory Machines



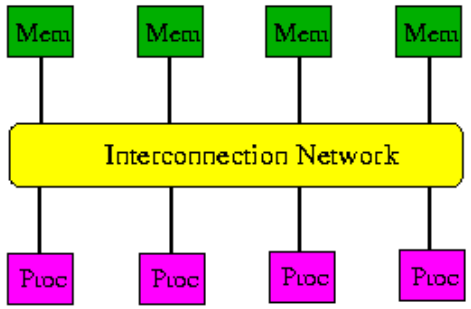
Advantages:

+ scalable
+ latency hiding

Disadvantages:

- harder to program
- program must be replicated

Shared Memory Machines



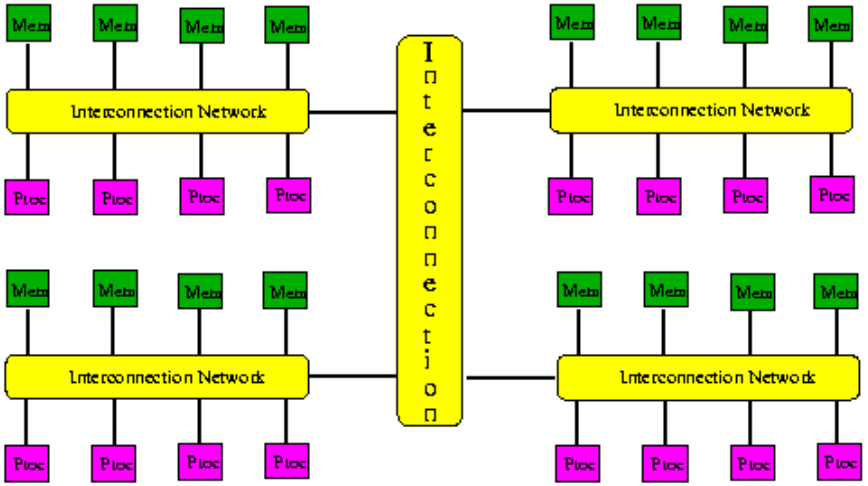
Advantages:

+ ease of programming
+ processors share code and data

Disadvantages:

- scalability problem

Cluster of Symmetric Multiprocessor Systems (SMP)



Advantages:

+ scalable

Disadvantages:

- programming paradigm unclear

Parallelism Granularity

- **Microoperations:** parallelism within an instruction exploited by control unit.
- **Instructions:** parallelism between instructions exploited by pipeline, superscalar architectures.
- **Basic Blocks:** in practice parallelism between loop iterations exploited via multithreading, multiple processors/cores.
- **Program modules**: functional parallelism exploited by multiple processors.

Regularity versus Irregularity (1)

- **Data Structures:** dense vectors/matrices versus sparse (stored as such) matrices

rows = 5
columns = 5
size = 12
capacity = 16

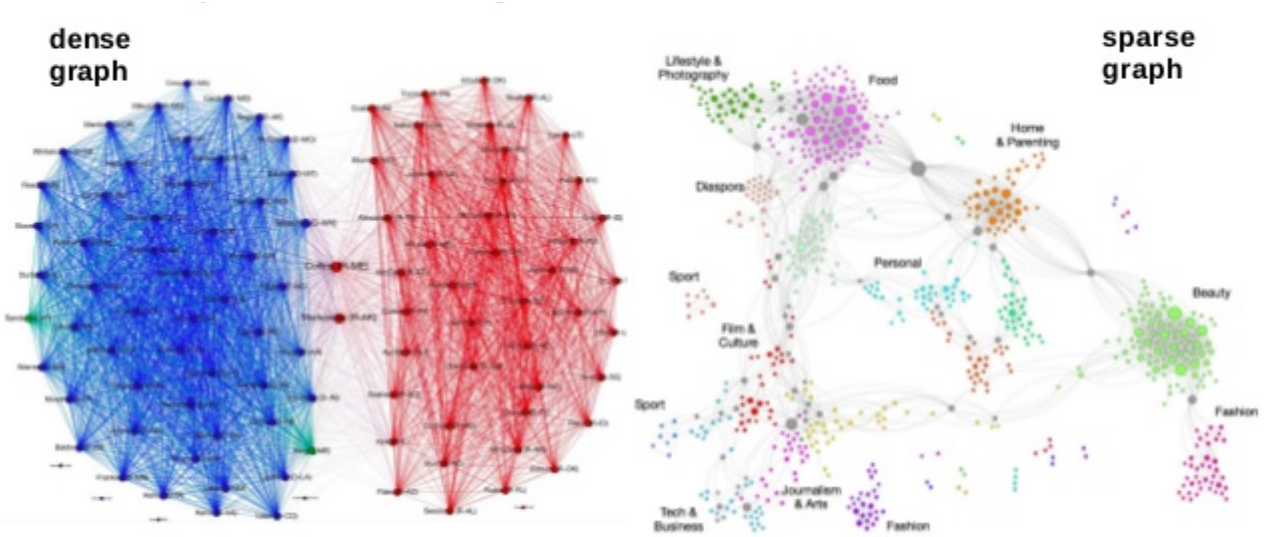
$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0.2 & 0 & 0 & 0 & 0.6 & 0 \\ 0 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.2 & 0 & 0 & 2.5 & 2.6 & 0 \\ 3.0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.7 \\ 0 & 0 & 0 & 0 & 4.4 & 0 & 0 & 0 \\ 0 & 0 & 5.2 & 0 & 0 & 5.5 & 5.6 & 0 \\ 0 & 6.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.2 & 7.3 & 7.4 & 0 & 0 & 7.7 \end{pmatrix}$$

diagonal	0	1.1	2.2	0	4.4	5.5	0	7.7

row_index	0	2	2	4	6	6	8	9	12

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
column_index	2	6	5	6	0	7	2	6	1	2	3	4				
off_diagonal	0.2	0.6	2.5	2.6	3.0	3.7	5.2	5.6	6.1	7.2	7.3	7.4				

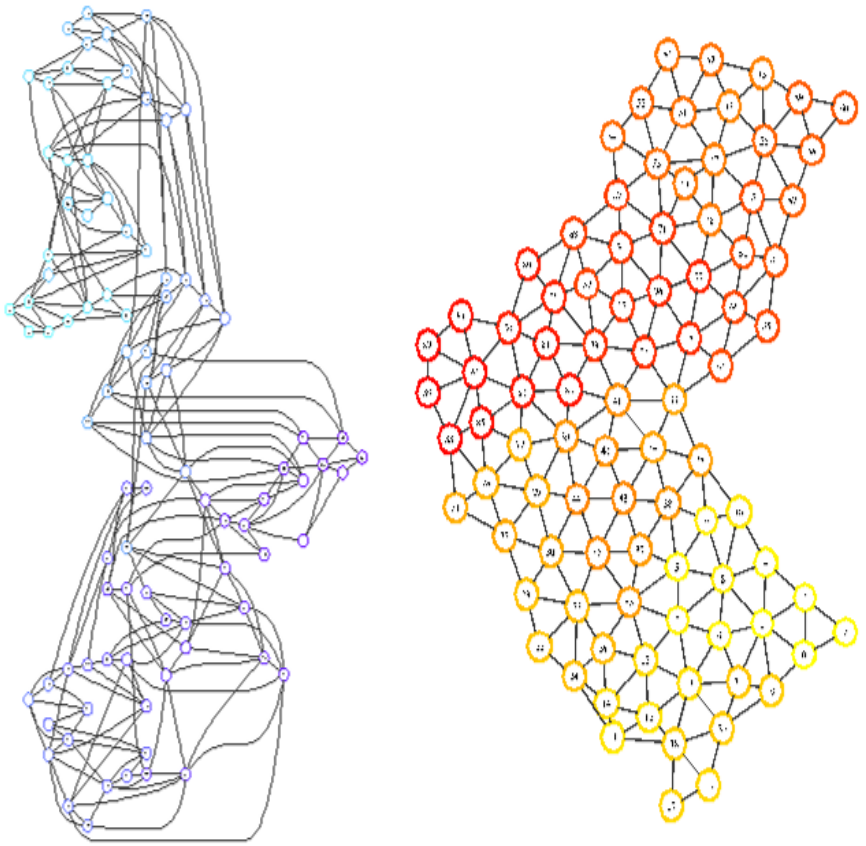
Source: D. Harder, Univ. of Waterloo



Source: Derek Greene

Regularity versus Irregularity (2)

- **Data access:** regular vector access (with strides) versus indirect access (scatter/gather).
- **Computation:** uniform computation on a grid versus highly dependent upon grid point computation.
- **Communication:** regular communication versus highly irregular graphs.



source: A. Bhatele, Urbana Champaign

Supercomputer
Nr. 5 on Top 500

Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband

Specifications and Features

Processor: IBM POWER9™ (2/node)

GPUs: 27,648 NVIDIA Volta V100s (6/node)

Nodes: 4,608 Node Performance: 42TF Memory/node: 512GB DDR4 + 96GB HBM2 NV Memory/node: 1600GB

Total System Memory: >10PB DDR4 + HBM + Non-volatile

Interconnect Topology: Mellanox EDR 100G InfiniBand, Non-blocking Fat Tree

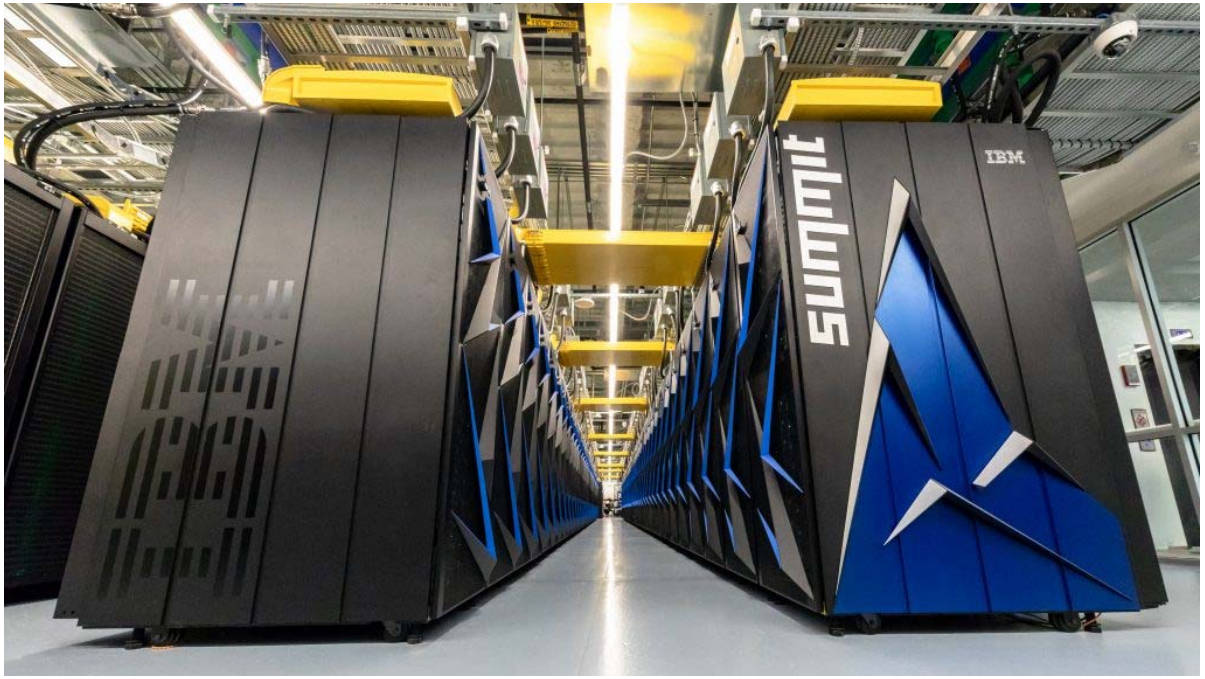
Peak Power Consumption: 13MW

2,4 M cores

Peak perf.: 200 Pflops

HPL: 148 Pflops

Power: 10 MWatt

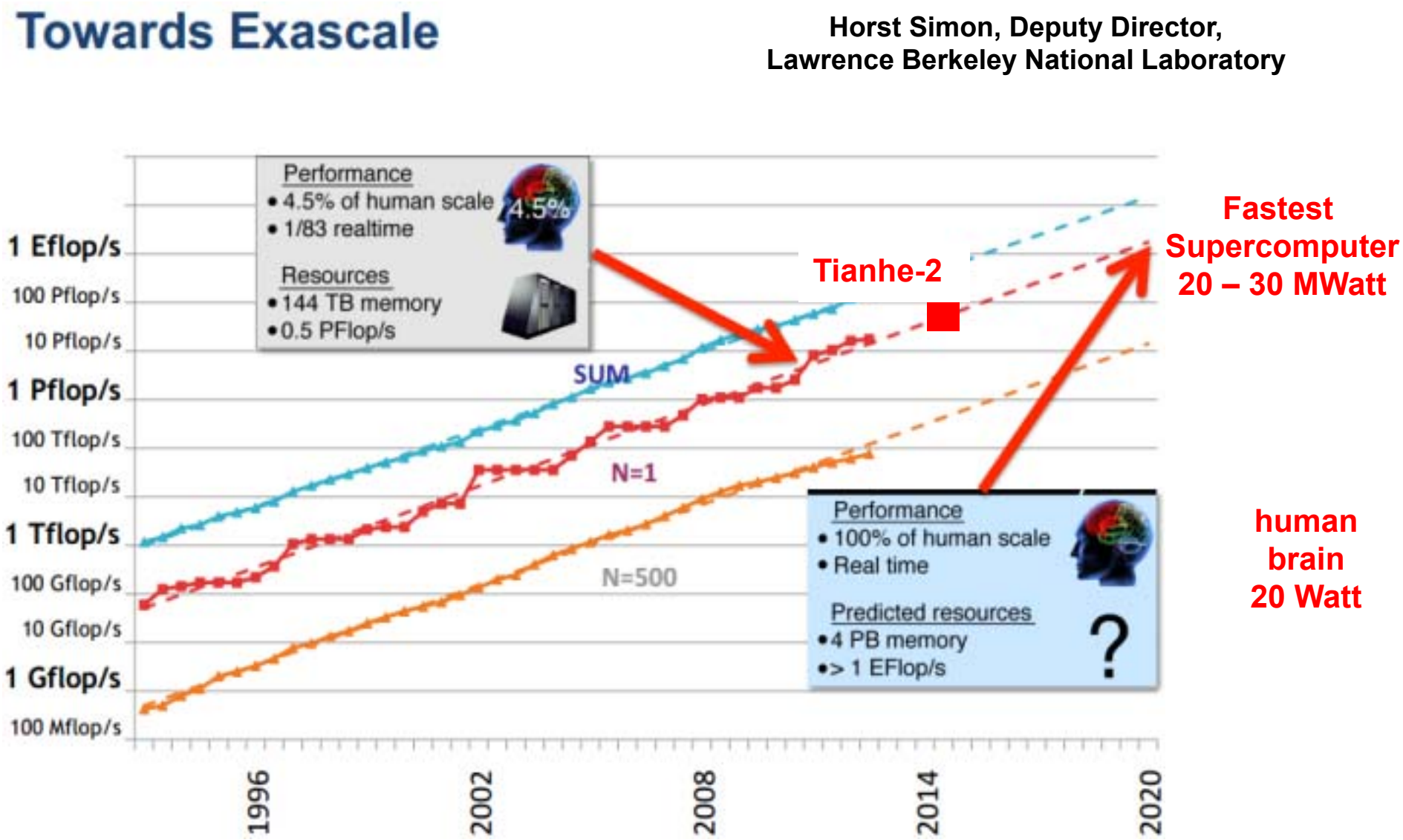


Smartphone, PC, Supercomputer

	Smartphone	PC	Supercomputer
			
	iPhone 11 max pro	Intel Core i9-9900K	Summit - IBM Power System AC922, IBM POWER9 22C
Cores	6(2High+4Light)	16	2,414,592
Memory	4 Gbyte RAM	128 GByte	2.801 PB
Performance	134 Gflops	550 FP2 Gflops	200.795 PFlop/s
Energy consumption	18 Watt	95 Watt	12 MWatt
Costs	>1000 Euro	590 Euro	325 Mio USD

- all parallel computers, programmed in a similar style

Supercomputer and a Human Brain



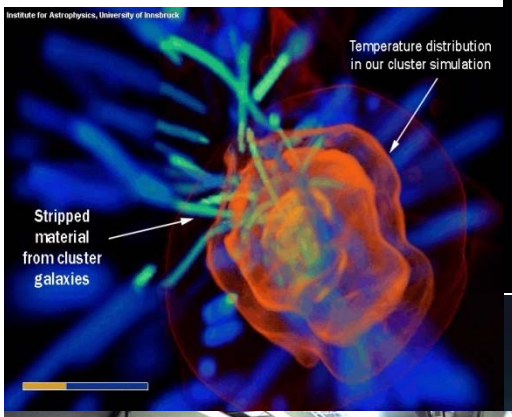
Parallel Processing at the Distributed and Parallel Systems Group, UIBK
dps.uibk.ac.at

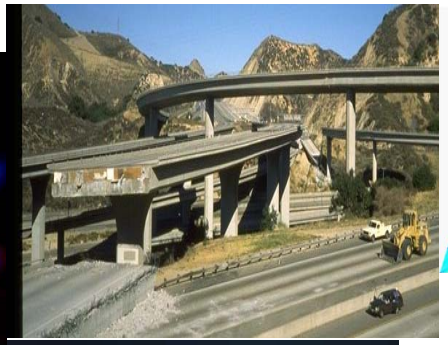
- Insieme: an optimizing compiler based on machine learning for MPI/OpenMP/OpenCL/SYCL C/C++ codes
- Celerity compiler and runtime system for SYCL based GPUs (clusters)
- Performance Tools and Technologies
  - SCALEA, SCALEA-G, Aksum, PerformanceProphet, Zenturio, P3T+
- Programming Paradigms and Environments
  - AllScale, Celerity, OpenMP+, libWater, JavaSymphony, AFCL
- Runtime Environment
  - Scheduling, Resource Brokerage, multi-objective optimization
- Parallelisation of real-world Applications
  - Material Science, Finance Modeling, Photonics, Astrophysics, River Modeling, Simulation of semiconductor devices, games

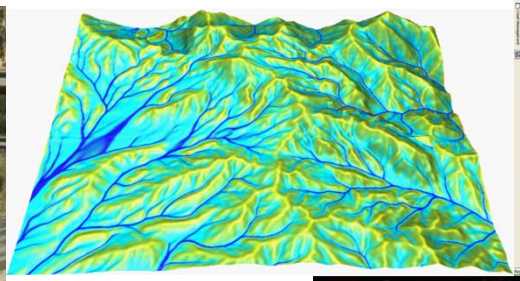
Celerity, Insieme, Askalon

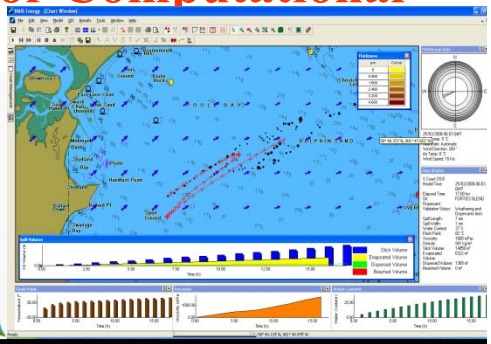
www.insieme-compiler.org www.askalon.org

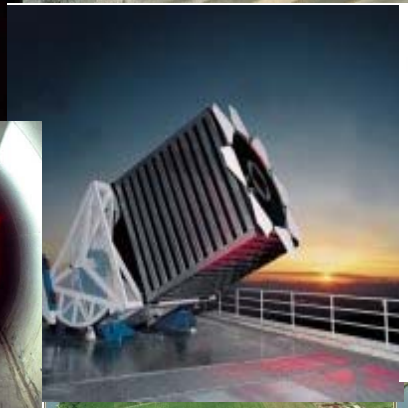
Application Development, Analysis, Optimization, Execution for Computational Science and Beyond

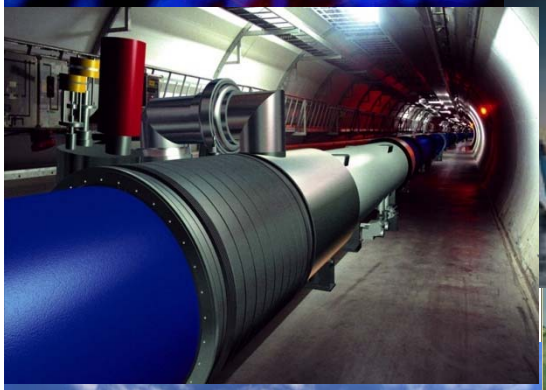












Books

- **Parallel Programming in C with MPI and OpenMP**
  Michael J. Quinn, McGraw Hill

- **Parallel Computer Architecture: A Hardware/Software Approach**
  David E. Culler, Jasweinder Pal Singh, Anoop Gupta, Morgan Kaufmann, 1999

- **Designing and Building Parallel Programs**
  Ian Foster, www.mcs.anl.gov/dbpp

Sources for Slides (Acknowledgements)

- S. Amarasinghe, MIT
- R. Buyya, University of Melbourne, Australia
- J. Dongarra, University of Tennessee, USA
- F. Ercal, University of Missouri-Rolla
- T. Fahringer, University of Innsbruck, Austria
- Ian Foster, Argonne National Lab, USA
- M. Gerndt, Department of Computer Science, TU Munich, Germany
- W. Jalby, University de Versailles, France
- K. Kennedy, Rice University, USA
- M. J. Quinn, Oregon State University, Corvallis, USA
- X. Sun, Illinois Inst. of Technology, USA
- M. Voss, University of Toronto, Canada
- B. Wilkinson and M. Allen, University of North Carolina, USA