Supercomputing

TMA4280—Introduction to Supercomputing

NTNU, IMF

January 12. 2018

Outline

Context: Challenges in Computational Science and Engineering

Examples: Simulation of turbulent flows and other applications

Goal and means: parallel performance improvement

Overview of supercomputing systems

Conclusion

Computational Science and Engineering (CSE)

What is the motivation for Supercomputing?

Solve complex problems fast and accurately:

— efforts in modelling and simulation push sciences and engineering applications forward,

— computational requirements drive the development of new hardware and software.

Computational Science and Engineering (CSE)

Development of computational methods for scientific research and innovation in engineering and technology.

Covers the entire spectrum of natural sciences, mathematics, informatics:

— Scientific model (Physics, Biology, Medical, . . .)
— Mathematical model
— Numerical model
— Implementation
— Visualization, Post-processing
— Validation

→ Feedback: virtuous circle

Allows for larger and more realistic problems to be simulated, new theories to be experimented numerically.

Outcome in Industrial Applications

Figure: 2004: “The Falcon 7X becomes the first aircraft in industry history to be entirely developed in a virtual environment, from design to manufacturing to maintenance.” Dassault Systèmes

Evolution of computational power

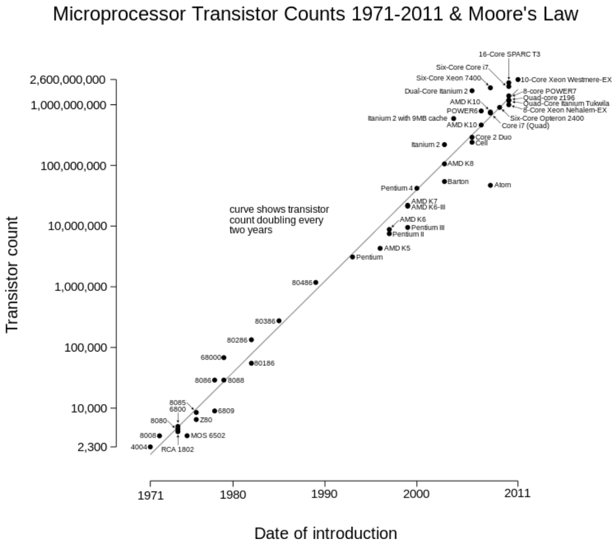


Figure: Moore's Law: exponential increase of number of transistors per chip, 1-year rate (1965), 2-year rate (1975). WikiMedia, CC-BY-SA-3.0

Evolution of computational power

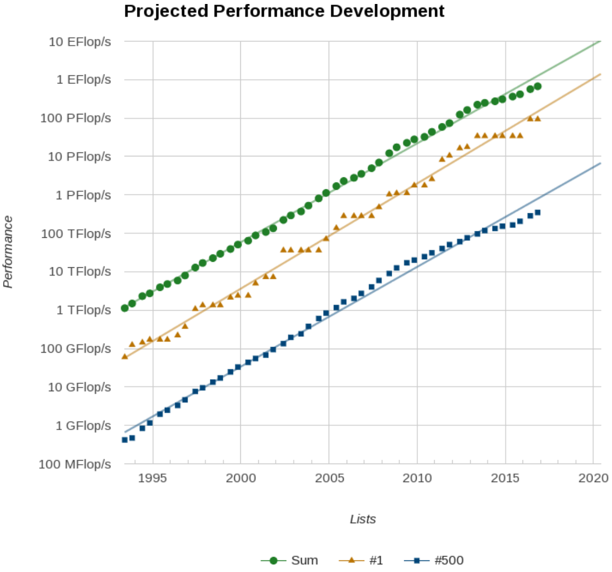


Figure: Top500: worlwide ranking of supercomputers. Source: top500.org

Progress in numerical methods and software

Progress cannot be achieved only by raw performance:

— improvement of linear solvers,

— model reduction,

— uncertainty quantification,

— development of new programming models,

— ...

Each stage of the development of a new method requires care:

— Implementation: unit testing

— Algorithms: verification

— Conceptual model: validation

— Error propagation, Reliability: uncertainty quantification

Outline

Context: Challenges in Computational Science and Engineering

Examples: Simulation of turbulent flows and other applications

Goal and means: parallel performance improvement

Overview of supercomputing systems

Conclusion

Context

Why is it necessary to

— solve larger problems,

— improve accuracy,

and how to work towards these goals?

Example: Large-Eddy Simulation

Incompressible Navier–Stokes Equations (NSE):

Find $\hat{\boldsymbol{u}} \equiv (\boldsymbol{u}, p)$ such that:

$$\begin{array}{rcll} \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p - 2\nu \nabla \cdot \varepsilon(\boldsymbol{u}) & = & \boldsymbol{f} & \text{in } \Omega \times [0, T] \\ \nabla \cdot \boldsymbol{u} & = & 0 & \text{in } \Omega \times [0, T] \\ \hat{\boldsymbol{u}}(\cdot, 0) & = & \hat{\boldsymbol{u}}^0 & \text{in } \Omega \end{array}$$

— Scope: 3D turbulent flows, high Reynolds number $Re = 10^5 - 10^6$

— Applications: design of turbines, aircrafts, climate modelling, . . .

— Discretization: finite dimensional problem, N degrees of freedom,

— Modelling unresolved scales: physical subgrid model, stabilization (Implicit LES), filtering . . .

Number of degrees of freedom to resolve scales, Kolmogorov Law:

$$N \sim Re^{9/4}$$

AIAA Benchmark: Complex Landing Gear (BANC-II)

- Meshes: initial 3.6M elements, final 23.8M elements
- Resources: adaptive stages 900K core.h, final 600K core.h

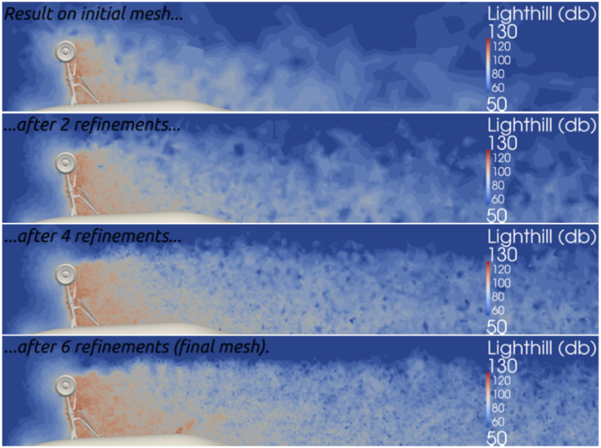


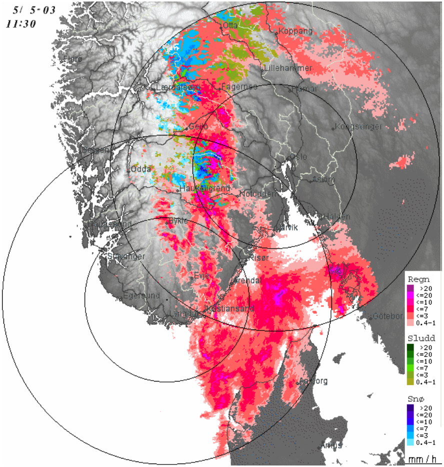
Figure: Mesh adaptation stages, Lighthill tensor (noise generation). Vilela De Abreu/N. Jansson/Hoffman, 18th AIAA Aeroacoustics Conference, 2012

Applications

Computationally intensive problems:

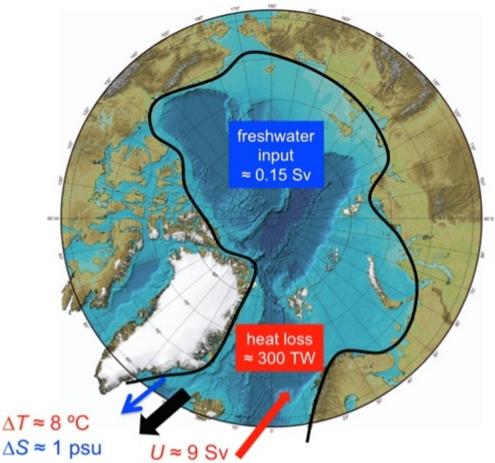
- Physical phenomena involving a large range of scales
- Systems with many degrees of freedom
- Methods with high dimensionality (Monte–Carlo)
- Big Data

Example: weather forecasting



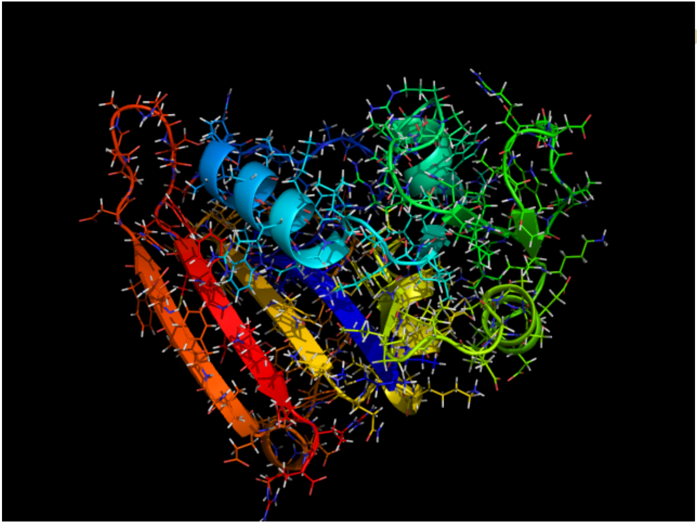
For more information, see http://met.no

Example: climate modelling



For more information, see http://www.bjerknes.uib.no.
Source: Nilsen et al. (2008)

Example: molecular dynamics



For more information, see http://www.gromacs.org.

Outline

Context: Challenges in Computational Science and Engineering

Examples: Simulation of turbulent flows and other applications

Goal and means: parallel performance improvement

Overview of supercomputing systems

Conclusion

Evaluation of performance

Measure unit: FLOPS, Floating-point Operator Per Second.

1. R_{PEAK}: theoretical operation rate handled by the hardware.
2. R_{MAX}: maximum achieved in reality for a given benchmark.

$$\text{Efficiency: } \mathcal{E} = R_{MAX}/R_{PEAK}$$

Top500: ranking using HPL (High-Performance LINPACK Benchmark)
LINPACK: library for solving dense linear systems.

Parallelization

Example of mesh distribution:

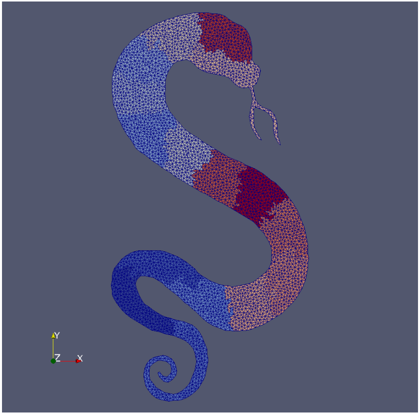


Figure: Snake demo run, 16 processes.

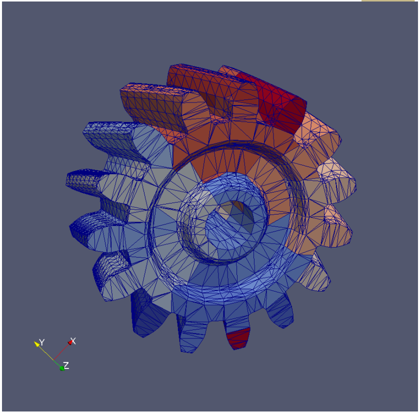


Figure: Gear demo run, 32 processes.

Expectations and limitations

- Given a fixed problem:

  Serial computation $\rightarrow$ Parallel computation on P processors

- The **wall time** T_n is the execution time on n processors.

- Speed-up:

$$S_P = T_1 / T_P$$

- Ideally, linear (strong) scaling: $S_P = P$

- Data dependencies, Communication overhead, ...

Levels of parallelism: Single processor

Core

- — pipelining
- — superscalar execution
- — vector processing (SIMD unit)
- — branch prediction
- — caching techniques
- — multithreading
- — prefetching
- — ...

Instruction-level parallelism, Concurrency

Levels of parallelism: Multi-processor

Compute node

- multiple cores on a chip
- core sharing cache memory
- affinity, locality
- accelerators
- . . .

Shared memory model

Levels of parallelism: Distributed system

Cluster (system comprised of several compute nodes)

- network topologies
- optimized libraries
- communication patterns
- ...

Distributed memory model

Outline

Context: Challenges in Computational Science and Engineering

Examples: Simulation of turbulent flows and other applications

Goal and means: parallel performance improvement

Overview of supercomputing systems

Conclusion

Historical perspective: CRAY-1

“The world’s fastest from 1976 to 1982 !”

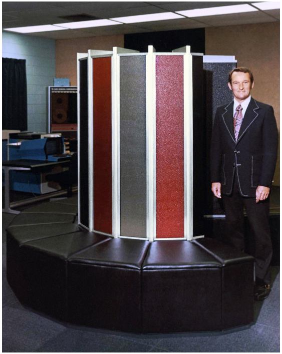


Figure: Pr. Seymour Cray and a CRAY-1. © Cray Research, Inc. (CRI)

Specifications:

- Type: Vector computer
- Processor Clock 83 MHz
- Peak performance 166 MFLOPS
- Memory 8 MB to 32 MB
- Sold for 8.8 M dollars
- Power consumption of 115 kW

Comparison:

- Intel Xeon E5-2620v3 (6 cores): 200 GFLOPS
- Nvidia Tesla: 500 GFLOPS to 1 TFLOPS

Historical perspective: evolution

Top Computers Over Time for the Highly-Parallel Linpack Benchmark.

Year	Computer	Processors	R_{MAX}
2005	IBM Blue Gene/L	131072	280600
2002	Earth Simulator Computer, NEC	5104	35610
2001	ASCI White-Pacific, IBM SP Power3	7424	7226
2000	ASCI White-Pacific, IBM SP Power3	7424	4938
1999	ASCI Red Intel Pentium II Xeon	9632	2379
1998	ASCI Blue-Pacific SST, IBM SP604E	5808	2144
1997	Intel ASCI Option Red 200MHzPentiumPro	9152	1338
1996	Hitachi CP-PACS	2048	368.2
1995	Intel Paragon XP/S MP	6768	281.1
1994	Intel Paragon XP/S MP	6768	281.1
1993	Fujitsu NWT	140	124.5
1992	NEC SX-3/44	4	20.0
1991	Fujitsu VP2600/10	1	4.0

Achieved performance R_{MAX} in GFlops.

Historical perspective: current consumer hardware

Can you estimate the computing power of a Raspberry Pi Model-B?

Historical perspective: current consumer hardware

Can you estimate the computing power of a Raspberry Pi Model-B?

LINPACK single node compute benchmark result:

Precision	R_{MAX}
Single	0.065 GFLOPS
Double	0.041 GFLOPS

A cluster of 64 Raspberry Pi Model B computers, labeled "Iridis-pi", achieved a LINPACK HPL suite result of:

1.14 GFLOPS (n=10240) at **216 Watts** for power consumption.

Supercomputers at NTNU: History

Supercomputing center established in 1983.

Year	System	Processors	Type	GFLOPS
1986–1992	Cray X-MP	2	Vector	0.5
1992–1996	Cray Y-MP	4	Vector	1.3
1995–2003	Cray J90	8	Vector	1.6
1992–1999	Intel Paragon	56	MPP	5.0
1996–2003	Cray T3E	96	MPP	58
2000–2001	SGI O2	160	ccNUMA	100
2001–2008	SGI O3	898	ccNUMA	1000
2006–2011	IBM P5+	2976	Distributed SMP	23500
2012–	SGI Altix ICE X	23040	Distributed SMP	497230

Supercomputers at NTNU: Current

`vilje.hpc.ntnu.no`

- System: SGI Altix ICE
- Type: Distributed SMP
- Nodes: 1404
- Each node is a shared memory system with two octa-core chips and 32 GB memory.
- Physical cores: 22464
- Logical cores: 44928
- CPU: Intel Xeon (Sandy Bridge)
- Theoretical peak performance: 467 TFLOPS

UNINETT Sigma2 manages the supercomputing infrastructure in Norway: https://www.sigma2.no/

Supercomputers at NTNU: IDUN/Lille

— System: Dell P630

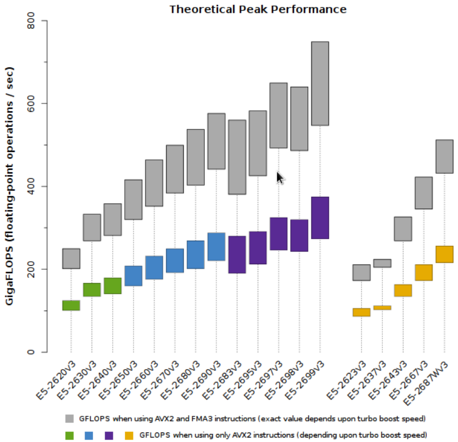
— Type: Distributed SMP

— Nodes: 27

— Each node is a shared memory system with two deca-core chips and 128 GB memory.

— CPU: Intel Xeon E5-2630v4 (about 500GFLops per CPU)

Supercomputers at NTNU: IDUN/Lille



Types of system architecture

Various existing categories:

— vector computers (CRAY X-MP)

— shared memory systems (SGI O3)

— distributed memory clusters (SGI Altix ICE X)

— compute grids (Folding@HOME)

Each of them require:

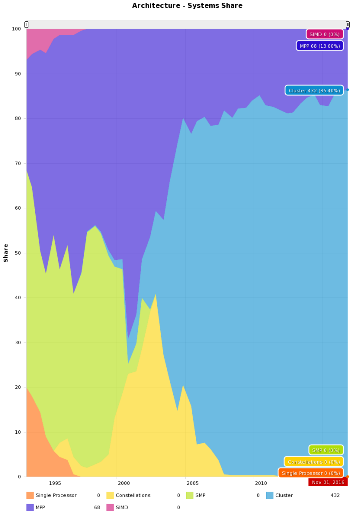
— adapting algorithms,

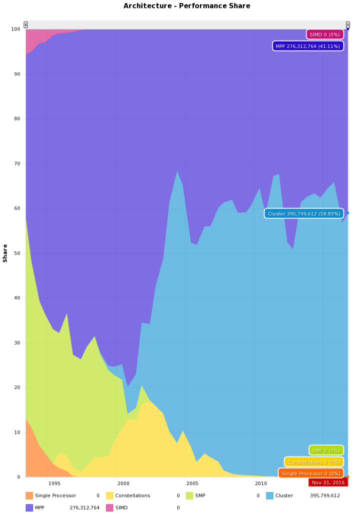
— using different programming models,

— hardware-specific implementation optimizations,

— ...

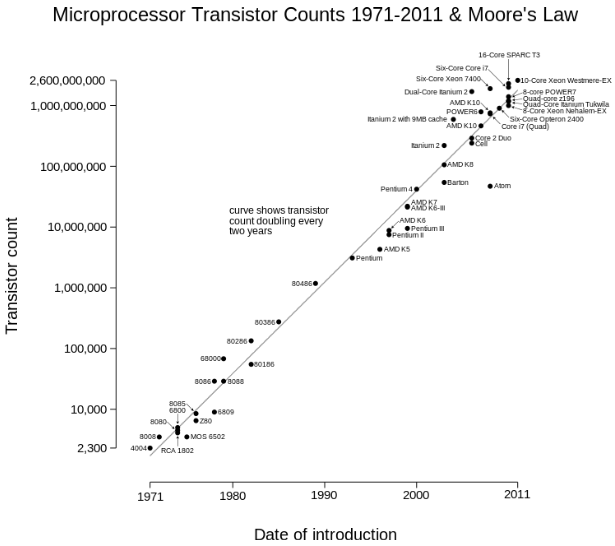
Evolution: system architecture





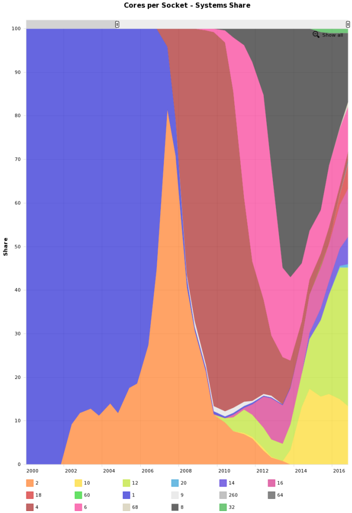
Comparison of the evolution w.r.t system and performance share.

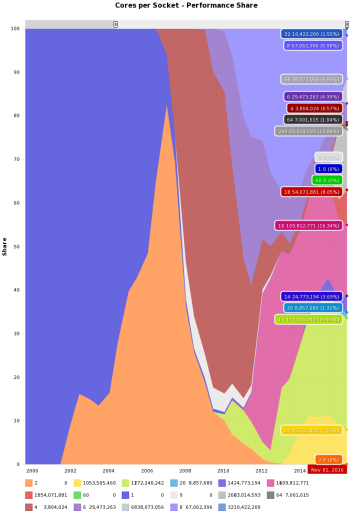
Back to Moore’s Law



Limitations to improving single-core performance → multi-core.

Evolution: number of cores per socket





Comparison of the evolution w.r.t system and performance share.

Network

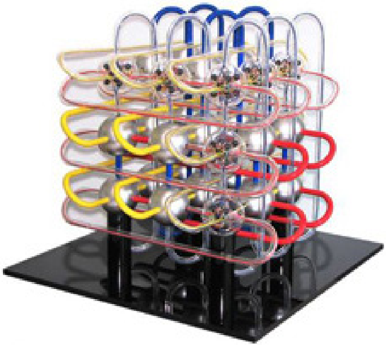


Figure: K-Computer, 6D mesh/torus TOFU network topology, Fujitsu.

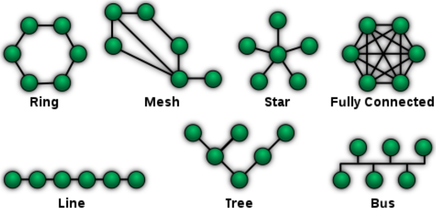
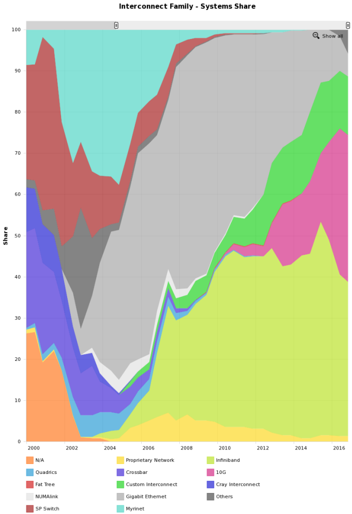


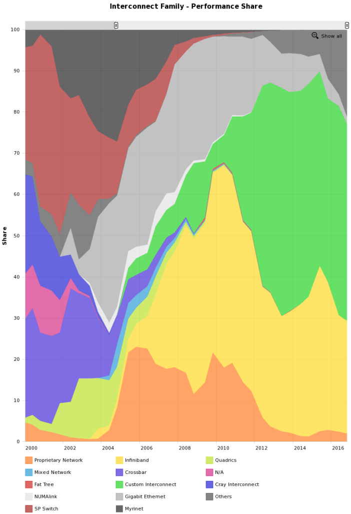
Figure: Network topologies, WikiMedia, Public Domain

Factors influencing the performance of communications:

- nature of the wiring (copper cable, optical fibre)
- topology of the network to reduce the transmission path
- protocols, algorithms (packet ordering, error checking)

Evolution: interconnect





Comparison of the evolution w.r.t system and performance share.

Efficiency: findings patterns

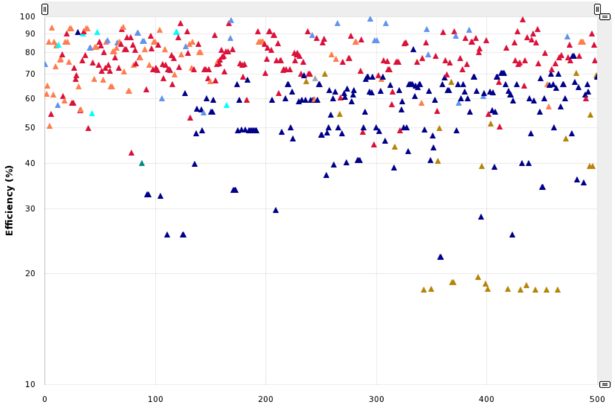


Figure: Top500: Efficiency vs Interconnect

Power efficiency: finding patterns

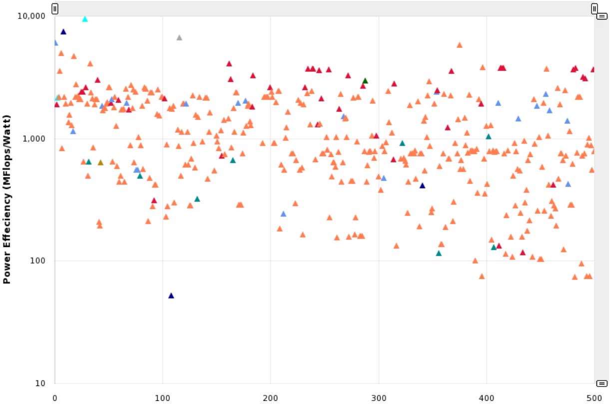


Figure: Top500: Power Efficiency vs Accelerator

Cost: top supercomputers' energy cost per year is $\sim$ million dollar. If energy consumption scales linearly: 1/10 nuclear power plant per Exascale supercomputer.

Performance considerations

November 2016 HPCG Results

Rank	Site	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)	Fraction of Peak
1	RIKEN Advanced Institute for Computational Science Japan	K computer – SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10.510	7	0.6027	5.3%
2	NSCC / Guangzhou China	Tianhe-2 (MilkyWay-2) – TH-IVB-FEP Cluster, Intel Xeon 12C 2.2GHz, TH Express 2, Intel Xeon Phi 31S1P 57-core NUDT	3,120,000	33.863	2	0.5801	1.1%
3	Joint Center for Advanced High Performance Computing Japan	Oakforest-PACS – PRIMERGY CX600 M1, Intel Xeon Phi Processor 7250 68C 1.4GHz, Intel Omni-Path Architecture Fujitsu	557,056	13.555	6	0.3855	1.5%
4	National Supercomputing Center in Wuxi China	Sunway TaihuLight – Sunway MPP, SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93.015	1	0.3712	0.3%
5	DOE/SC/LBNL/NERSC USA	Cori – XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries Cray	632,400	13.832	5	0.3554	1.3%
6	DOE/NNSA/LLNL USA	Sequoia – IBM BlueGene/Q, PowerPC A2 1.6 GHz 16-core, 5D Torus IBM	1,572,864	17.173	4	0.3304	1.6%
7	DOE/SC/Oak Ridge Nat Lab USA	Titan – Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray	560,640	17.590	3	0.3223	1.2%

Figure: High Performance Conjugate Gradients (HPCG) Benchmark project

Complement to LINPACK: dense vs sparse matrix computations.

Outline

Context: Challenges in Computational Science and Engineering

Examples: Simulation of turbulent flows and other applications

Goal and means: parallel performance improvement

Overview of supercomputing systems

Conclusion

Conclusion

- Supercomputing is crucial for Computer Science and Engineering to address new problems.
- Development of computational power is enabling but poses challenges.
- Adapting algorithms to compute on concurrent and parallel systems is non-trivial: application-dependent, low-level.
- Various hardware architecture, programming models, languages available.
- Communication is the key element in scalability.