JÜLICH
FORSCHUNGSZENTRUM

HPC Architectures

D. Pleiter (JSC)| Lattice Practices | 7 March 2014

JÜLICH FORSCHUNGSZENTRUM

Overview

Trend towards more parallelism

- Long-term trend and technological background

Parallel computing devices

- Many-core processors and GPUs

Memory architectures

- Providing sufficient data to computing devices
- Holding all required data

Network technologies and architectures

- Link technologies and network topologies

Selected examples of high-end HPC systems

- Selection out of 10 “fastest” systems

Power limits and energy efficiency

Top500 List

Performance metric

- Floating-point operations per time unit while solving dense linear set of equations

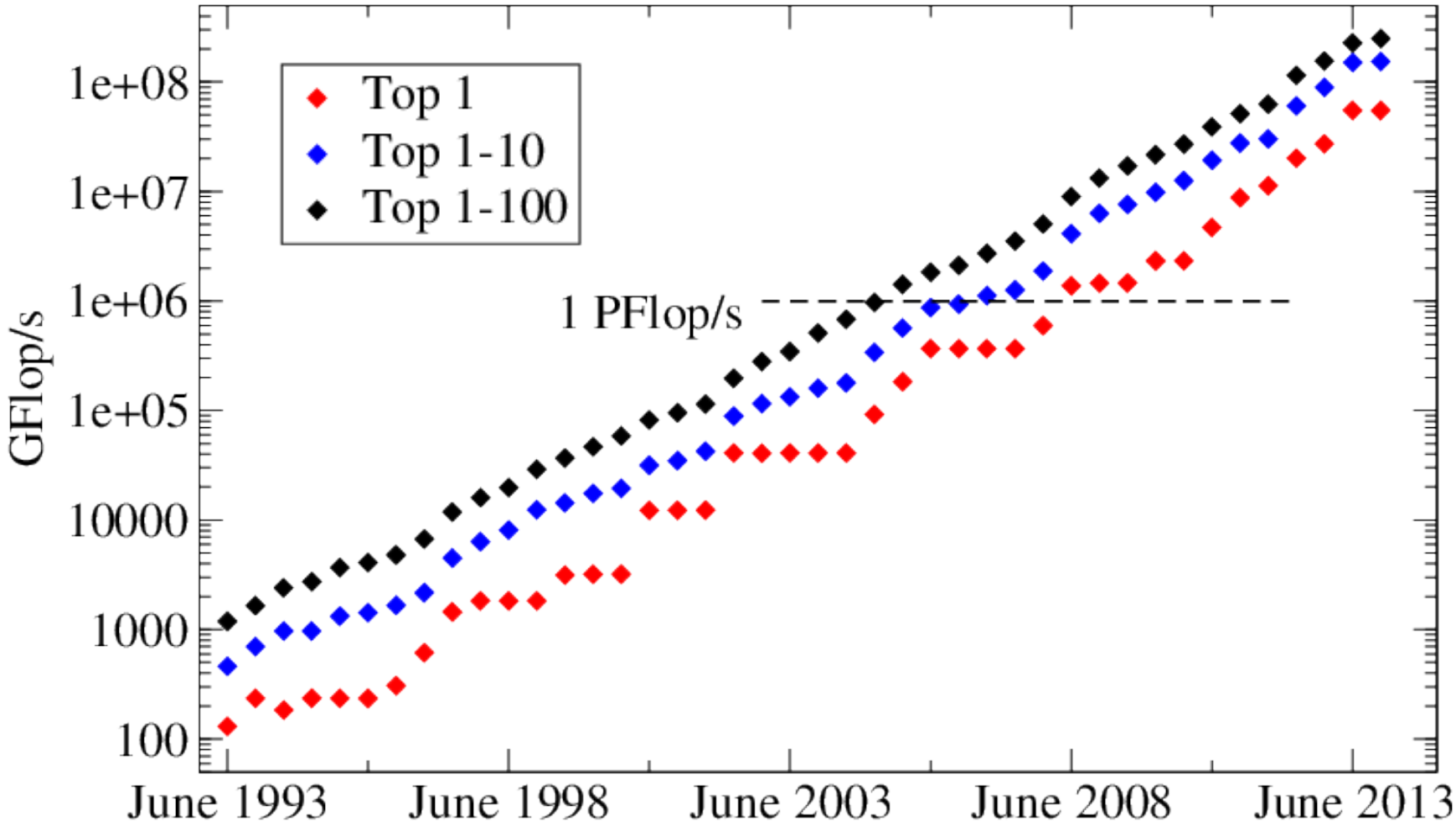
Criticism

- Work-load not representative
- Problem size can be freely tuned

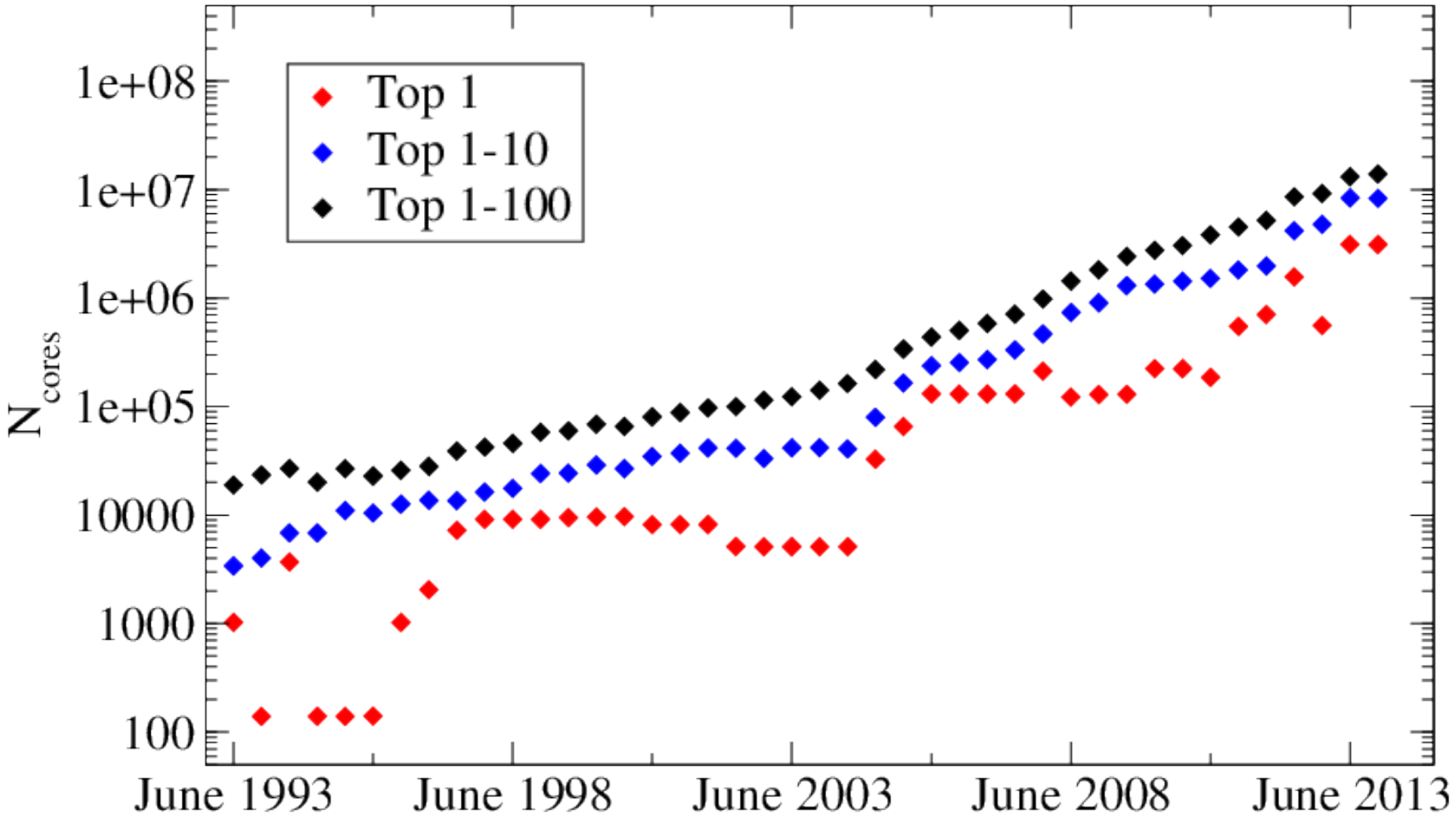
But: Allows for long-term comparison

- First list published in June 1993

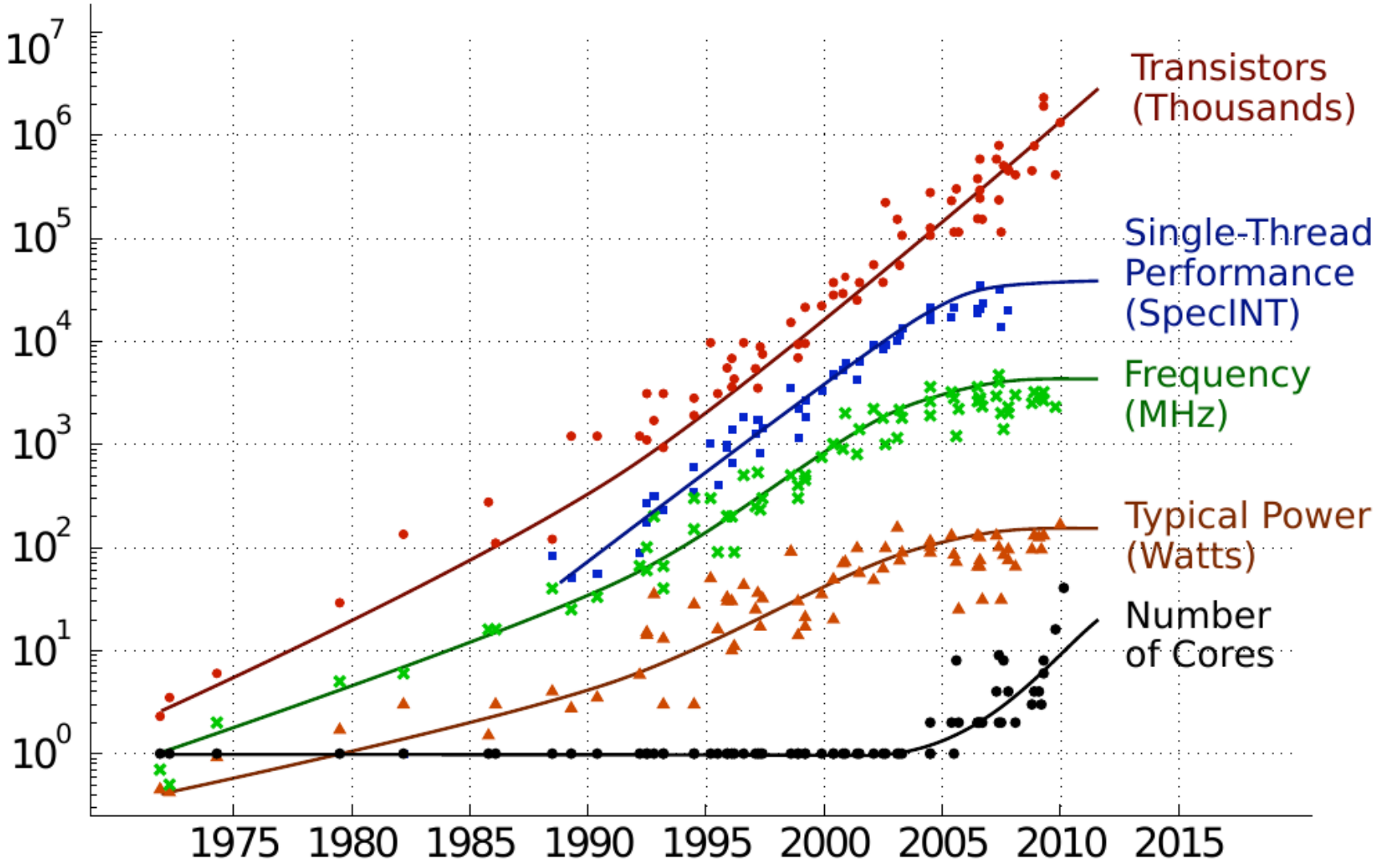
Top500 trends: Peak performance



Top500 trends: Number of “cores”



Processor Performance Trends



Multi-Level Parallelism

Micro-architecture level

- Instruction-Level Parallelism (ILP)
- Single Instruction Multiple Data (SIMD)
- Simultaneous Multi-Threading (SMT), Single-Instruction, Multiple-Threads (SIMT)

Processor level

- Multi-core, many-core

Node level

- Multiple sockets per node

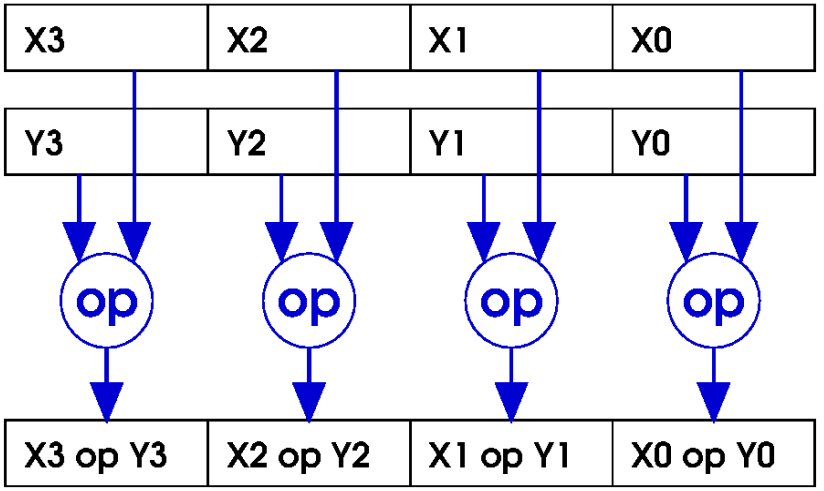
System level

- Many nodes per system

SIMD Parallelism

Exploit data level parallelism by operating on data in parallel, e.g. vector add:

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} \leftarrow \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$



Advantages

- More performance at lower die area and power costs

Challenges

- Vectorization of operations
- Data alignment or data gather/scatter

SIMD ISA Examples

ISA	Width [bits]	Vendors
SSE	128	Intel/AMD
AVX	256	Intel/AMD
AVX-512	512	Intel
VSX	128	IBM
QPX	256	IBM
NEON	128	ARM and others

SMT and SIMT Parallelism

Simultaneous Multi-Threading (SMT)

- Simultaneous execution of different sets of instructions

☞ **Helps to increase Instructions Per Cycle (IPC)**

Single Instruction, Multiple Threads (SIMT)

- Model introduced in NVIDIA Tesla
- Instruction scheduler only manages groups of threads (warps)
- Multiple warps within single Streaming Multiprocessor

☞ **Allows for high IPC with high level of concurrency**

Parallel Programming

MPI = Message Passing Interface

- Parallelisation at node and/or system level
- Works for non-shared memory spaces

Multi-threading: POSIX threads, OpenMP

- Parallelisation at micro-architecture and/or node level
- Mandates shared memory space

SIMD: auto-vectorization, intrinsics

- Parallelisation at micro-architecture level

Today: Need to mix

- Room for performance tuning

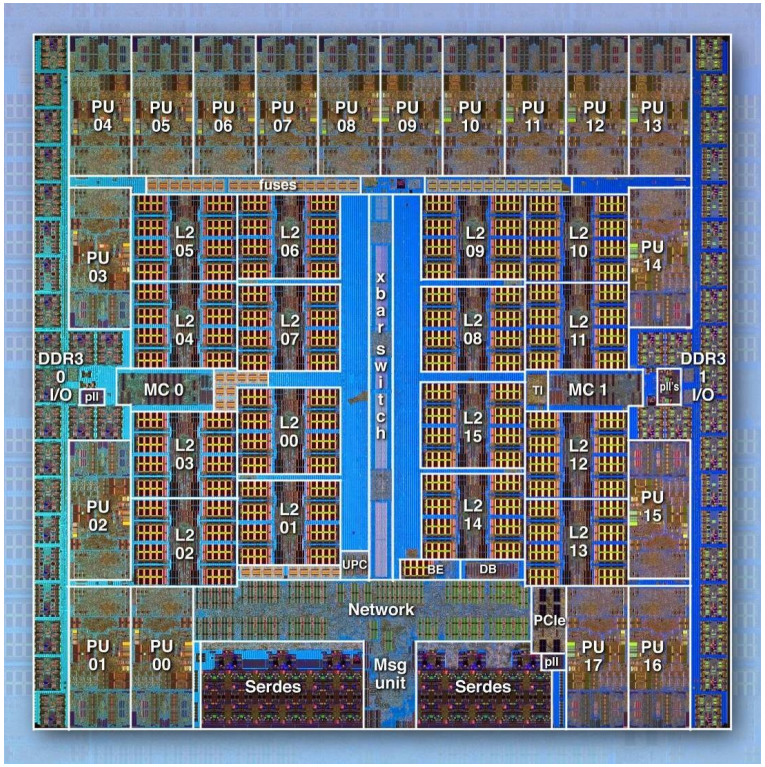
Example: Blue Gene/Q Processor

Processor core features

- 4-way Simultaneously Multi-Threaded (SMT)
- 256-bit wide SIMD unit
  - Up to 4 FMA/cycle
    → 12.8 GFlop/s @ 1.6 GHz

Processor features

- 16+1 cores
  - Up to 64 threads/processor
  - Up to 64 FMAs/processor/cycle
    → 204.8 GFlop/s @ 1.6 GHz
- 2 128-bit DDR channels
- 11 network links



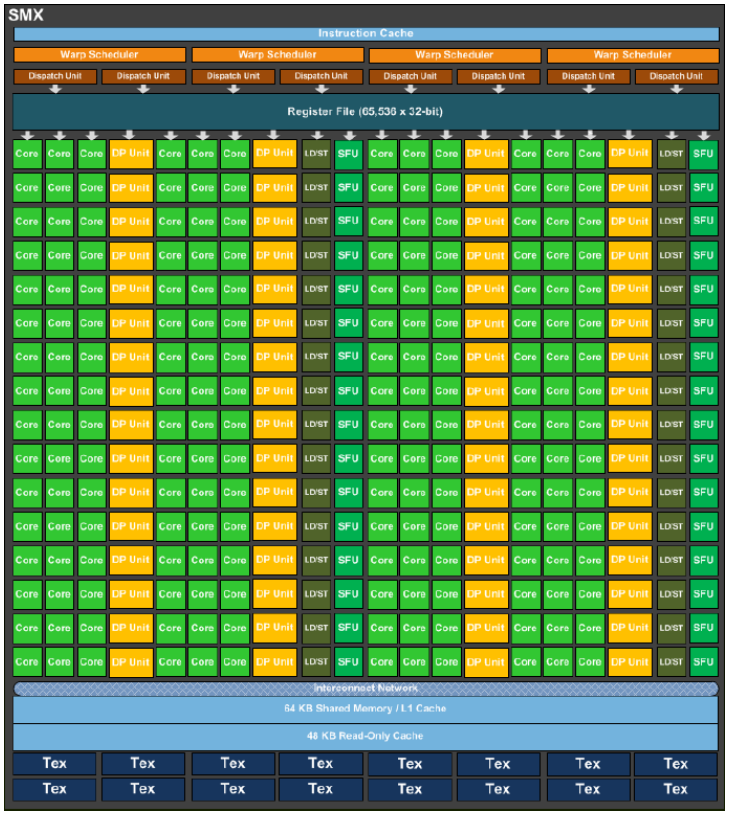
Example: NVIDIA Kepler GPU K40

Streaming Multiprocessor features

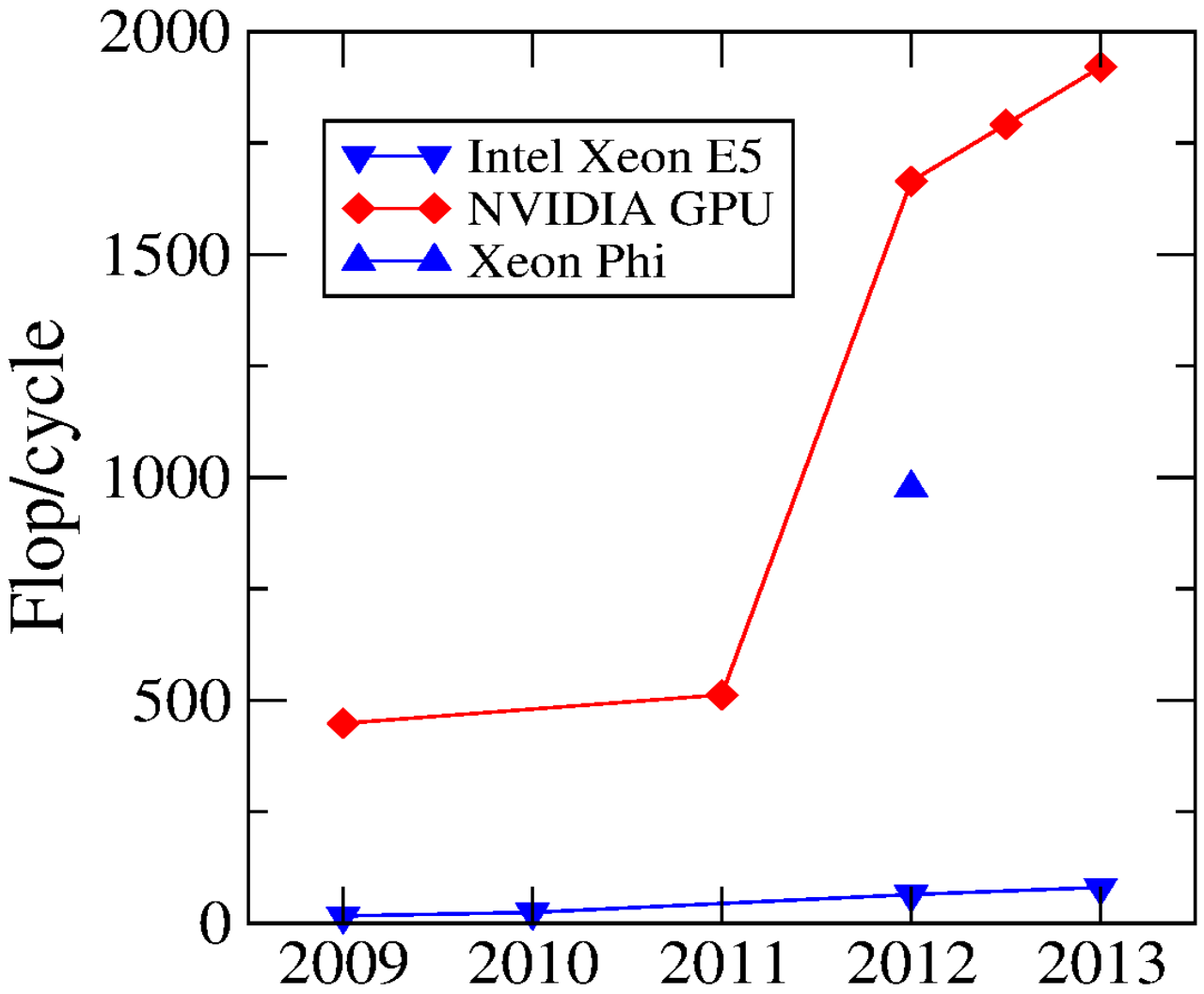
- 64 double precision units
  - Up to 64 FMA/cycle → 95 GFlop/s @ 0.745 GHz
- 192 CUDA cores
  - Up to 192 SP-FMA/cycle
    → 286 GFlop/s @ 0.745 GHz
- Up to 64 warps

Device features

- 15 SM
  - Up to 960 FMA/cycle
    → 1430 GFlop/s @ 0.745 GHz
- 6 GDDR5 channels
- 1 PCIe GEN3 port



Increasing Parallelism at Compute Device Level



Exploiting Parallelism in LQCD

SIMD instructions

- Parallelisation of local operations, e.g. multiplication with SU(3) matrices
- Domain decomposition of lattice
  → Vector elements belong to different sites

Multiple threads and processes

- Domain decomposition: different threads/processes operate on different parts of the lattice

Limits of parallelisation

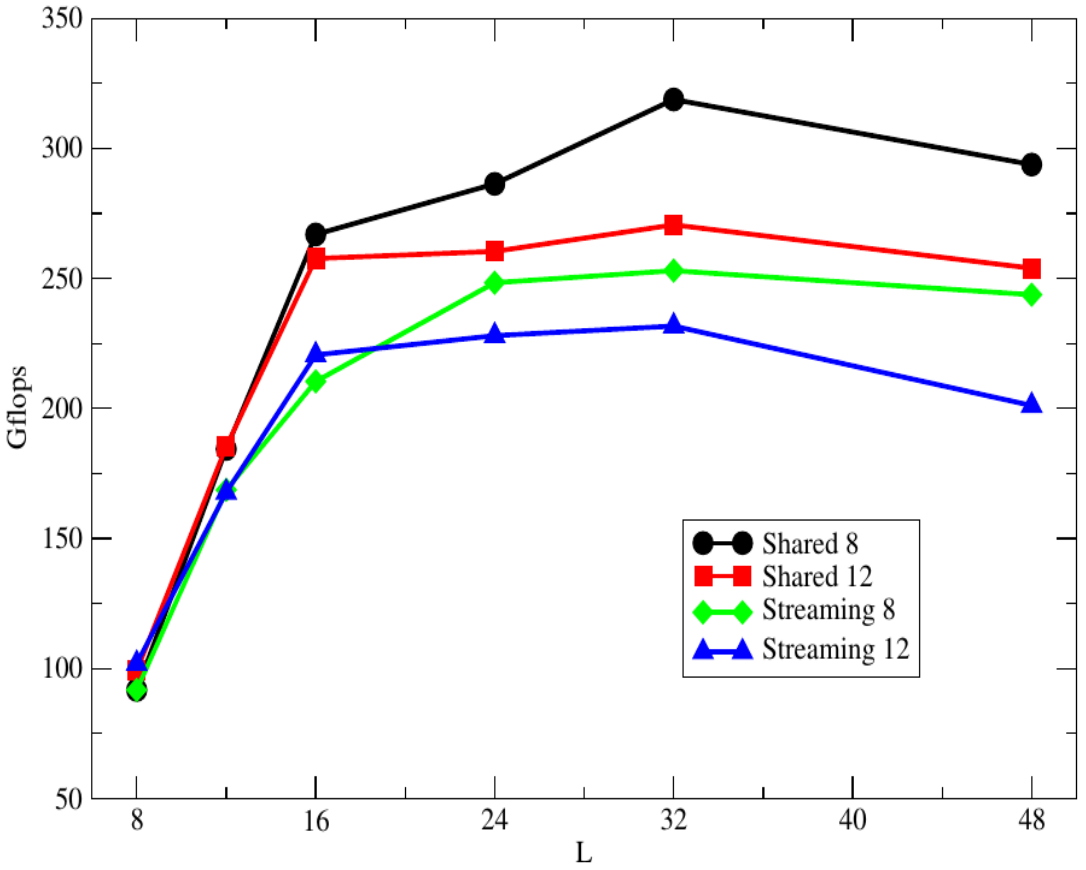
- Large lattice $L^4 = 64^4$ → concurrency of up to $O(10^7)$
- CRAY XK7 “Titan” with 18,688 NVIDIA K20 GPUs
  → $O(10^7)$ double precision units

Single Device Scaling Limits for LQCD

Large number of computing pipelines
→ large problem size

[M. Clark et al., 2012]

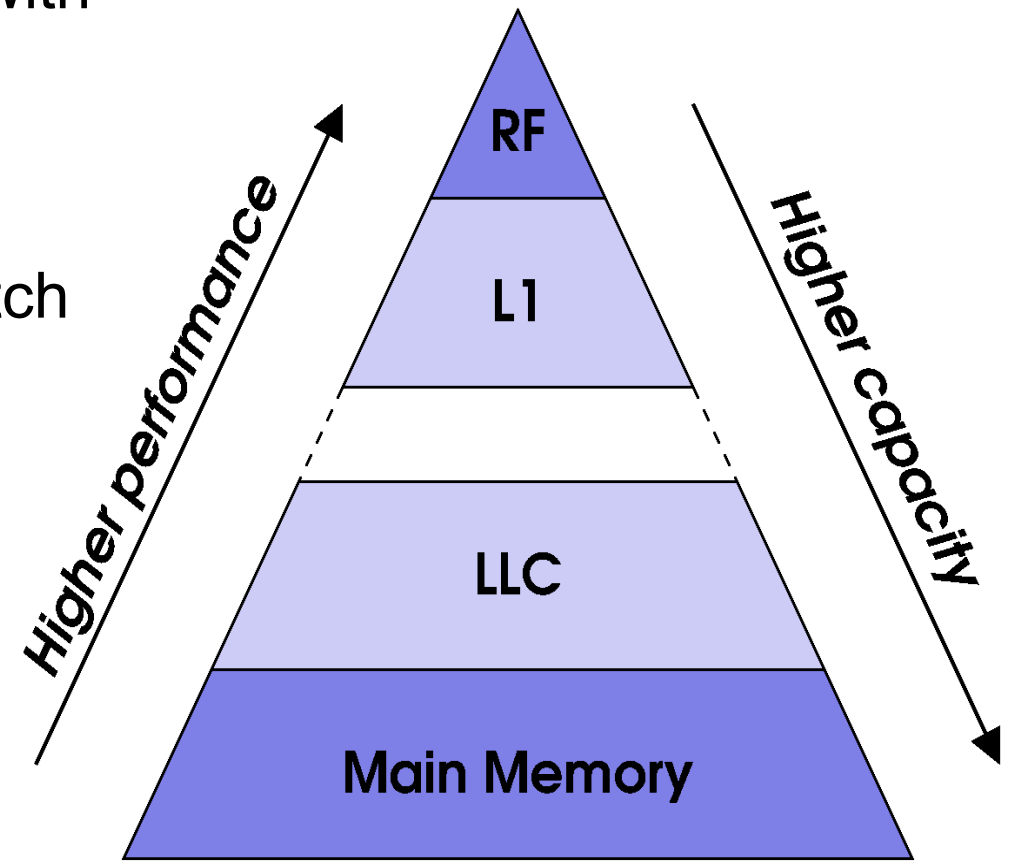
- Example: Dslash (single-precision)
- NVIDIA Fermi M2090
  - 16 SM
  - 32 cores/SM
- Need at least 128 sites per core to maximize performance



Memory Hierarchy

Strategy

- Hierarchy of memories with different performance
  - Closer to processor: faster but smaller
- Non-available data is fetch from next level



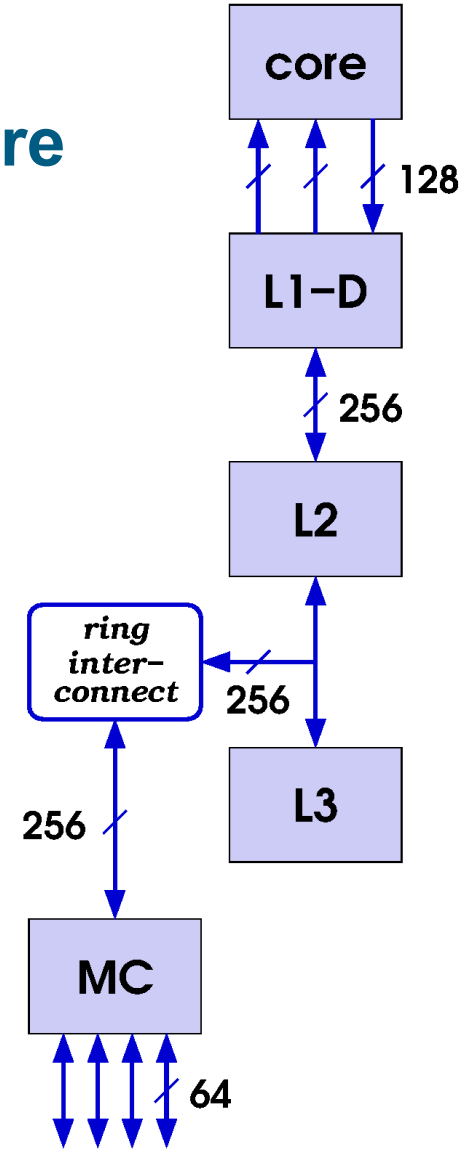
Memory Hierarchy (cont.)

Example: Intel Sandy Bridge architecture

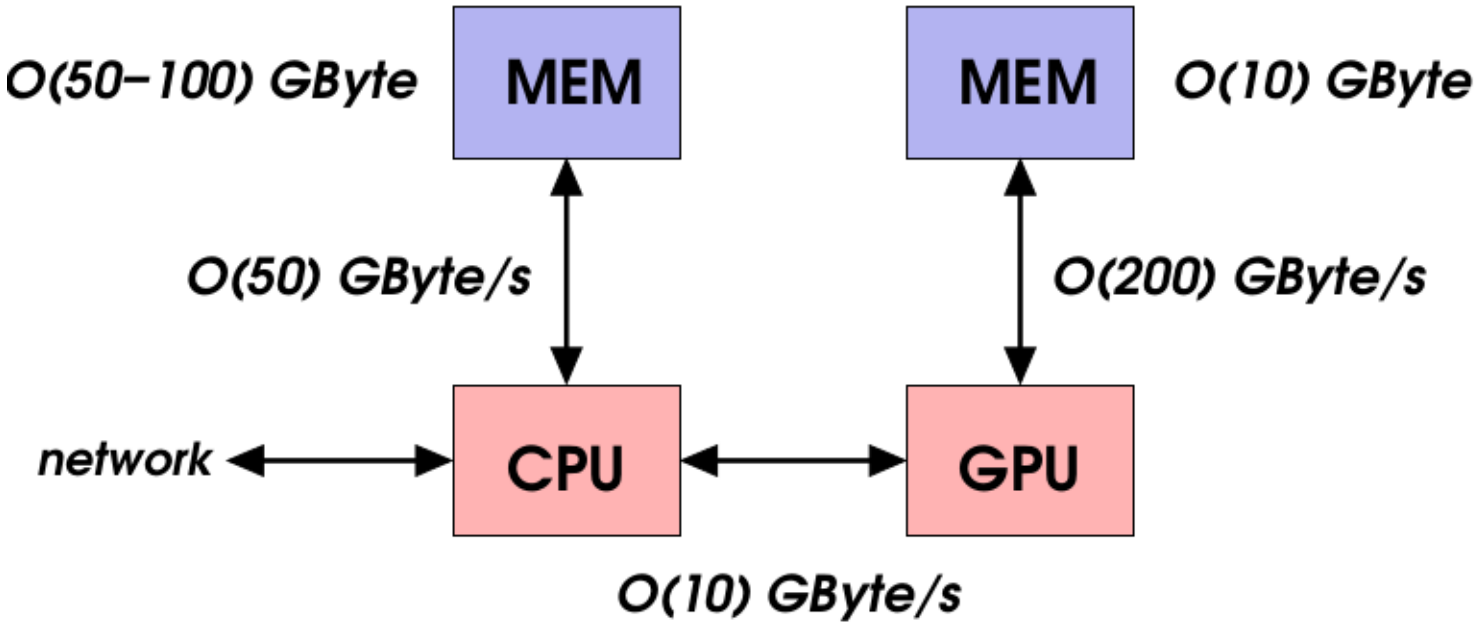
- L1 and L2 cache private to core
- Shared L3 cache
- Shared external memory interface

Access latencies

Level	Access time [cycles]
L1	4
L2	12
L3	26-31



GPU Node Architecture



Different storage devices

- Host memory attached to CPU
- Device memory attached to GPU

Capacity and performance differ by O(5 . . . 10)

Memory Technologies

Performance parameters

- Capacity C_{mem}
- Bandwidth B_{mem}
- Technologies with similar $R_{mem}=C_{mem}/B_{mem}$

High-performance memory technologies → small R_{mem}

- Examples: GDDR5, HMC, HBM

DDR SDRAM

- Huge market
- Capacity grows faster than bandwidth

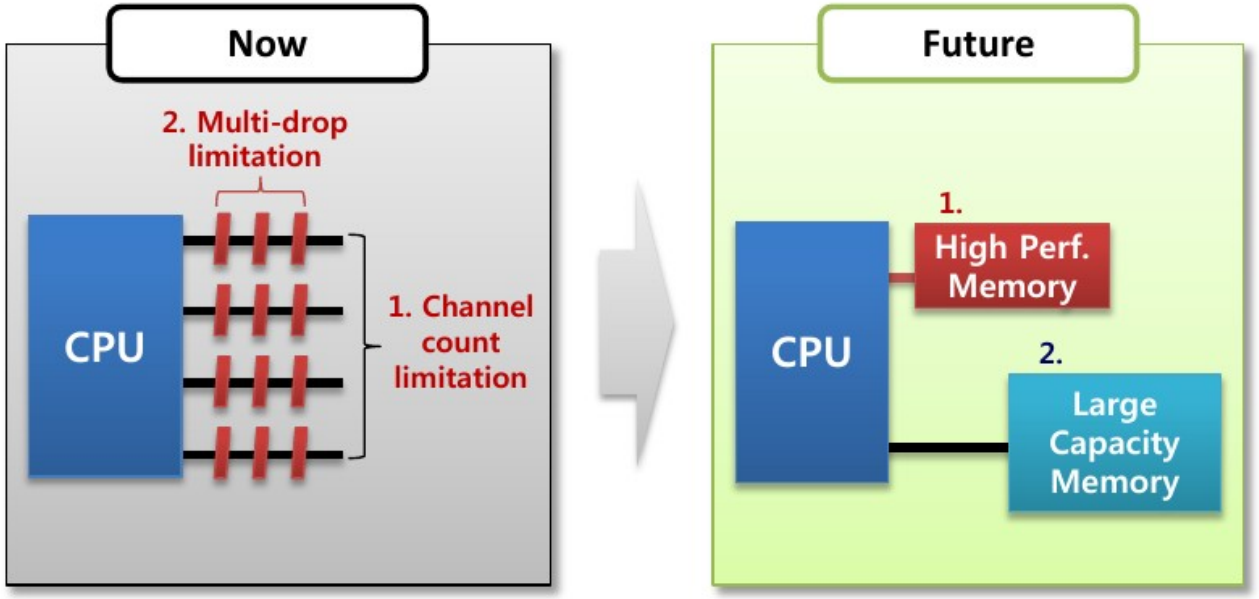
Dense memory technologies → large R_{mem}

- Today: non-volatile NAND flash memory

Richer Memory Hierarchies

Expect different memory technologies to be integrated in future processor/ node architectures

- High-bandwidth memory
- Large-capacity memory



[J. Jeong (Samsung), ISC'13]

Knights Landing Architecture

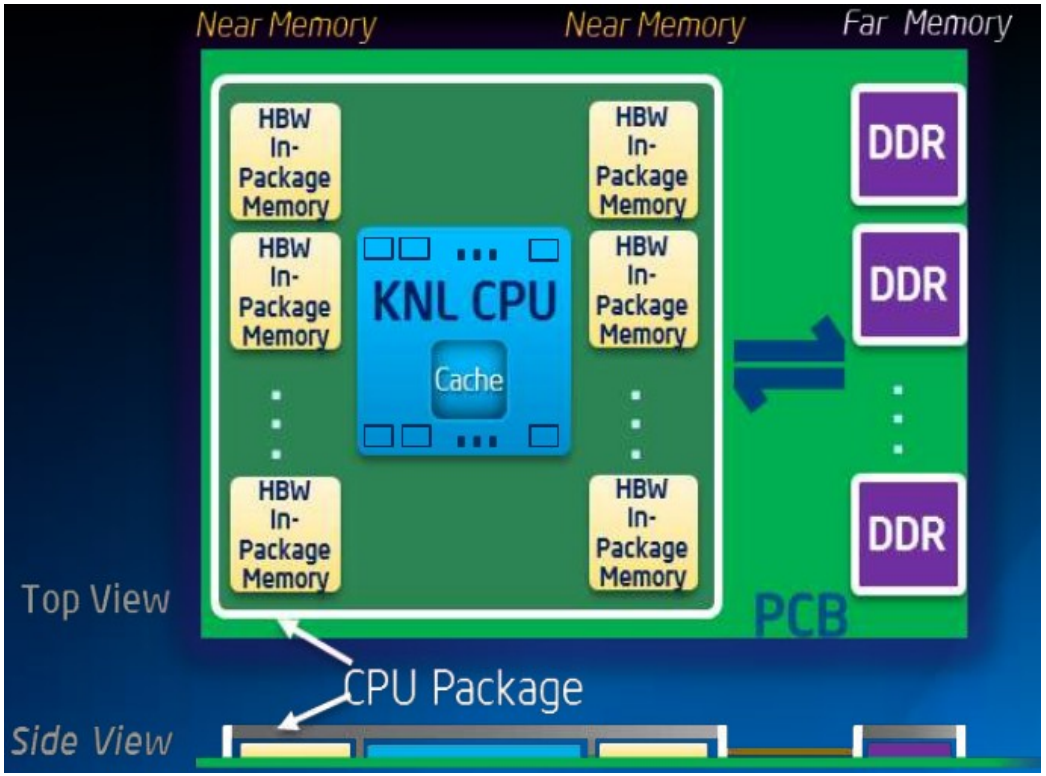
Next generation Xeon Phi

- O(60) cores
- 512-bit wide SIMD units

New memory hierarchy

- L1+L2 caches
- Near memory
  - On-package
  - High bandwidth, low capacity
- Far memory
  - Off-package
  - Low bandwidth, high capacity

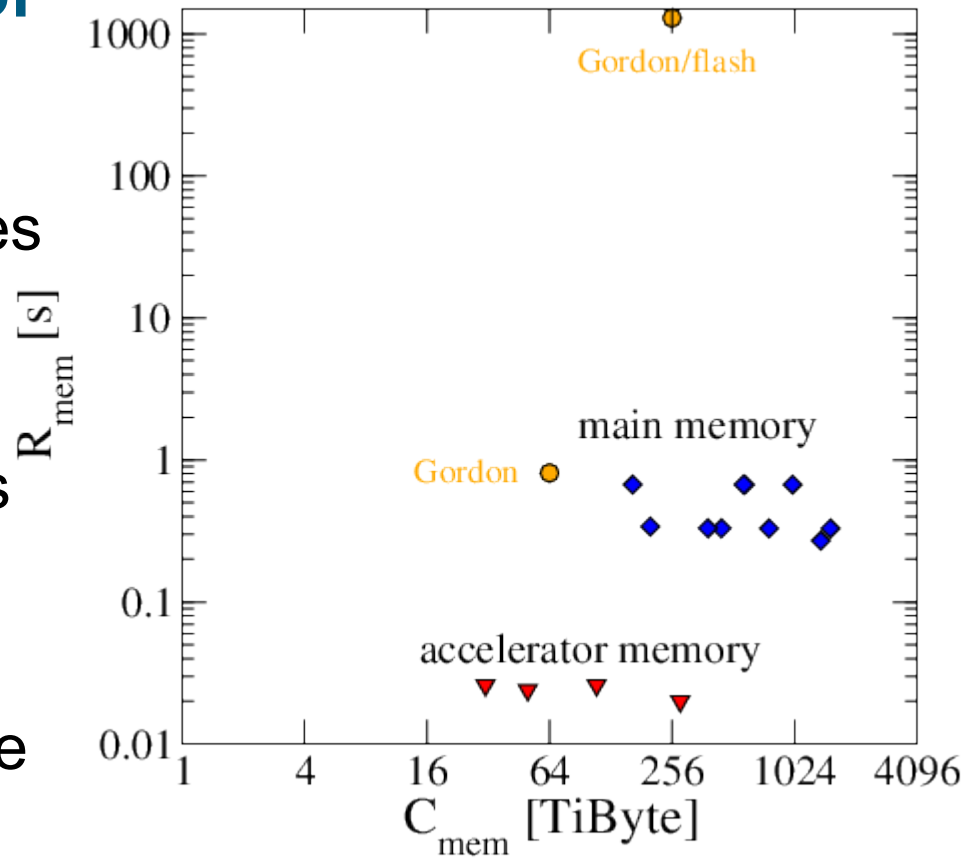
[R. Hazra et al., 2013]



Memory Architectures of Top500 Systems

Top 10 ranks of Top500 as of November 2013

- Maximum memory capacity stagnating at about 1.5 PiBytes
  - But: Top500 provides selective view!
- Increasing number of systems with accelerators



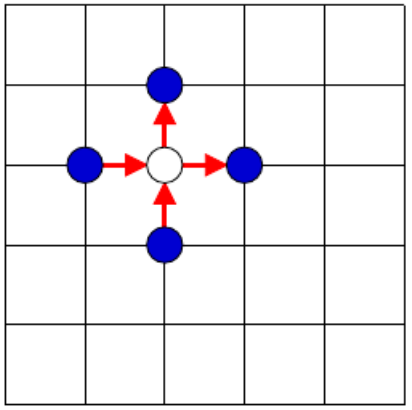
Gordon@SDSC

- Architecture integrating a large number of SSDs
- Rank #129 at Top500 list

LQCD: Memory Performance Requirements

Arithmetic Intensity Wilson Dslash

- AI = Number of Flop / Amount of transferred data
- AI_{dslash} = 1320 Flop / 3 kiByte = 0.4 Flop/Byte



Balanced architecture

- Balance compute and data transport performance
- Relevant parameters:
  - Floating-point operation throughput B_{fp}
  - Memory bandwidth B_{mem}

For most compute devices: $B_{fp}/B_{mem} > 5$

- To mitigate performance bottleneck need to maximize reuse of data → **leverage data locality**

Exploiting Data Locality in LQCD

Micro-architecture level

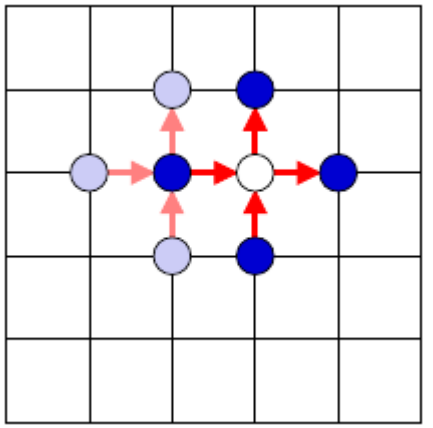
- Complex multiply-add
- Multiplication of small matrices

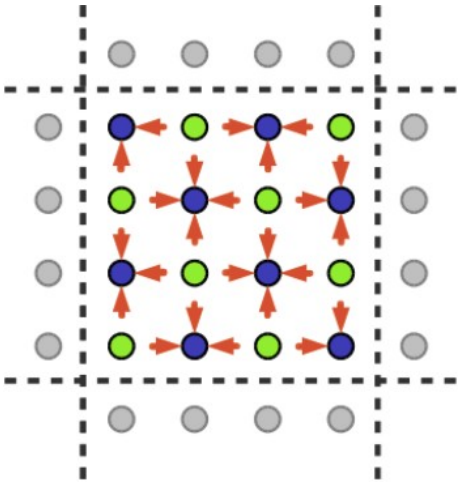
Processor level

- Re-use of stencil elements

Domain-decomposition solvers

- Blocks are processed independently without communication





Network

Link technology

- Performance characterized by
  - Bandwidth (still increasing)
  - Latency (difficult to reduce further)
- Common features of today's high-performance network technologies:
  - Point-to-point
  - Based on serializer/de-serializer architecture (SerDes)

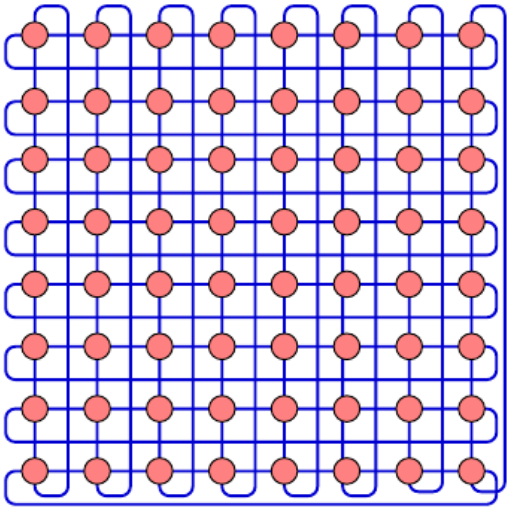
Network topology

- Key parameters:
  - Diameter: Maximum distance between 2 nodes
  - Bi-section bandwidth: minimum bandwidth between two equally sized half networks

Network Topologies

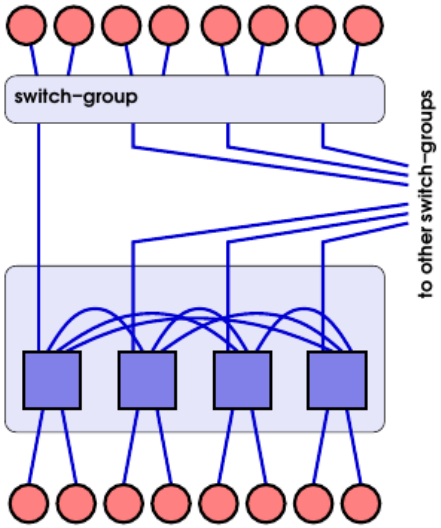
d-dimensional torus

- *d*=3, 5, also toroidal variants
- Stencil-type kernels map typically well
- Large *d* beneficial to keep diameter small and bi-section bandwidth large



Fat tree

- Tree-like topology using multiple levels of high-radix routers
- Fat = number of links the same at each layer



Dragonfly

- Multi-level all-to-all topology
- Small diameter, configurable bi-section bandwidth

LQCD Network Requirements

Wilson Dslash

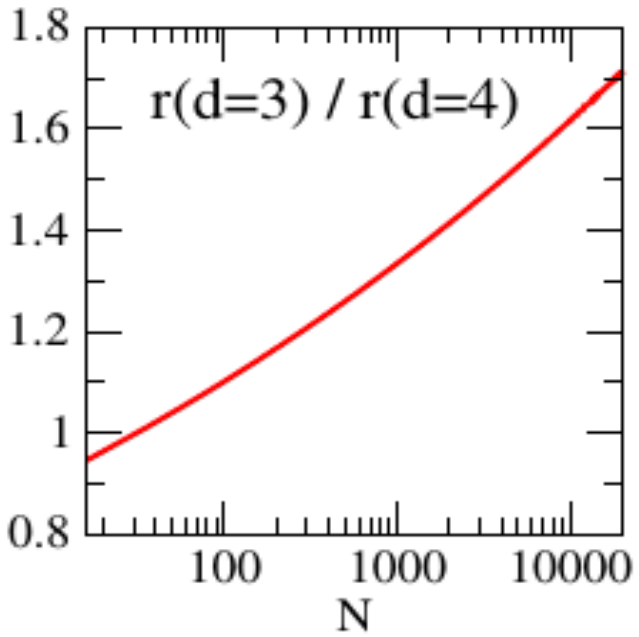
- Well mapped onto torus *d*-dimensional torus network (d ≤ 4)
- Local volume on N nodes:

$$v = L^{4-d}\left(\frac{L}{N^{1/d}}\right)^{d} = L^{4-d}\,l^{d}$$

- Surface to (local) volume ratio r :

$$r = \frac{2d\,L^{4-d}\,l^{d-1}}{v} = 2d\,l^{-1} = 2d\,\frac{N^{1/d}}{L}$$

☞ Higher dimension (potentially) better



Collective communication

- Small diameter networks (potentially) better
- Not performance critical

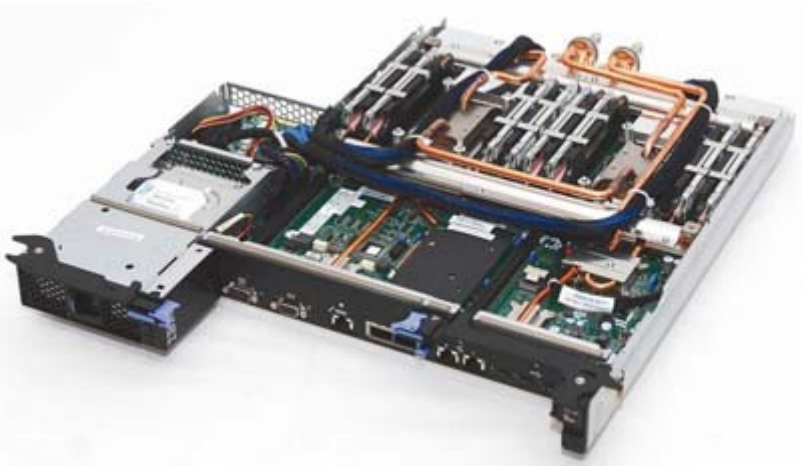
Top Entries in Top500 (November 2013)

Rank	Machine/ Architecture	Location
1	Tianhe-2	NUDT (CH)
2	Titan	ORNL (US)
3, 5, 8, 9, ...	Blue Gene/Q	LLNL (US), ANL (US), JSC (DE), ...
4	K Computer	KEK (JP)
7	Piz Daint	CSCS (CH)
10	SuperMUC	LRZ (DE)

SuperMUC

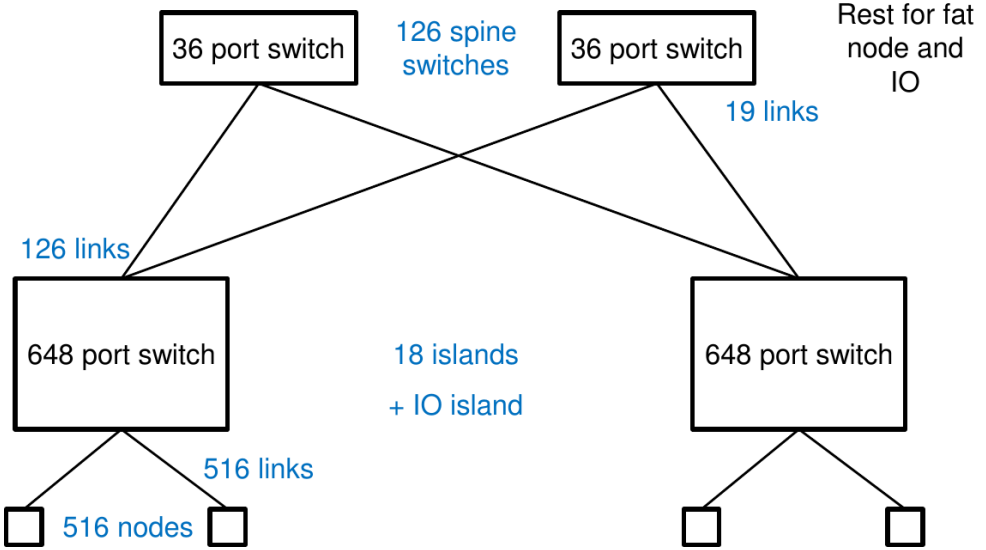
Node architecture

- 2 Intel Sandy Bridge-EP octo-core processor
- 32 GByte host memory



Network architecture

- Infiniband network
- Pruned fat-tree topology



Peak performance: 3.2 PFlops/s

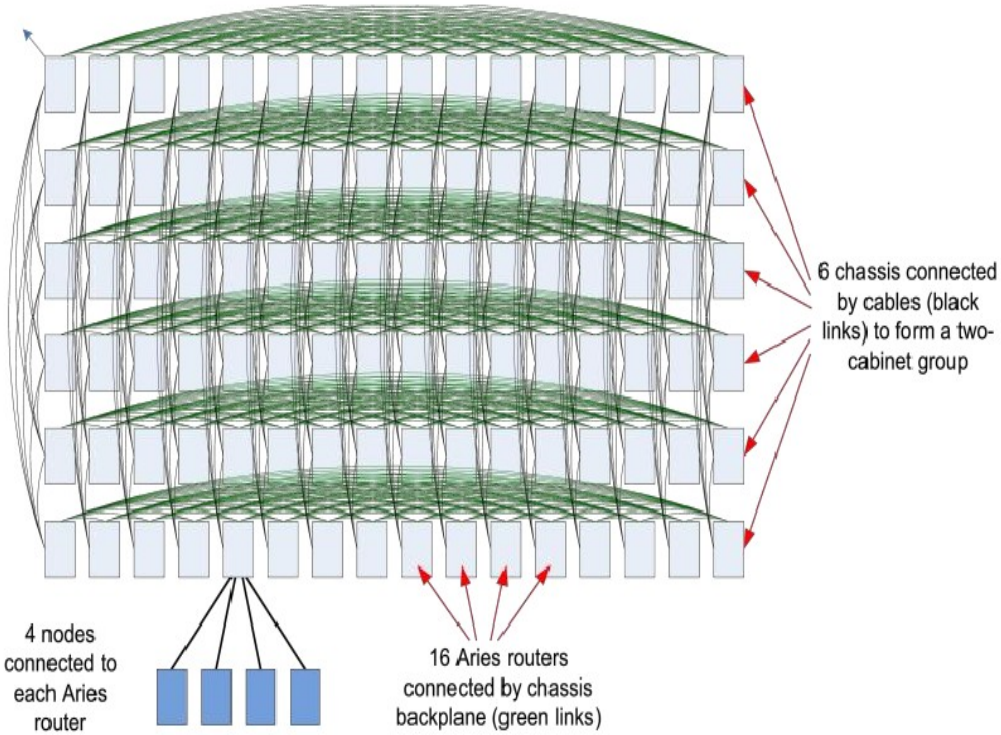
Cray XC30

Node architecture

- 1 Intel Sandy Bridge-EP octo-core processor
- 32 GByte host memory
- 1 NVIDIA K20x
- 6 GByte device memory

Network architecture

- High-speed links with ~5 GByte/s
- Dragon-fly topology



Cray XC30 @ CSCS

- 5272 nodes
- 7.8 PFlop/s peak

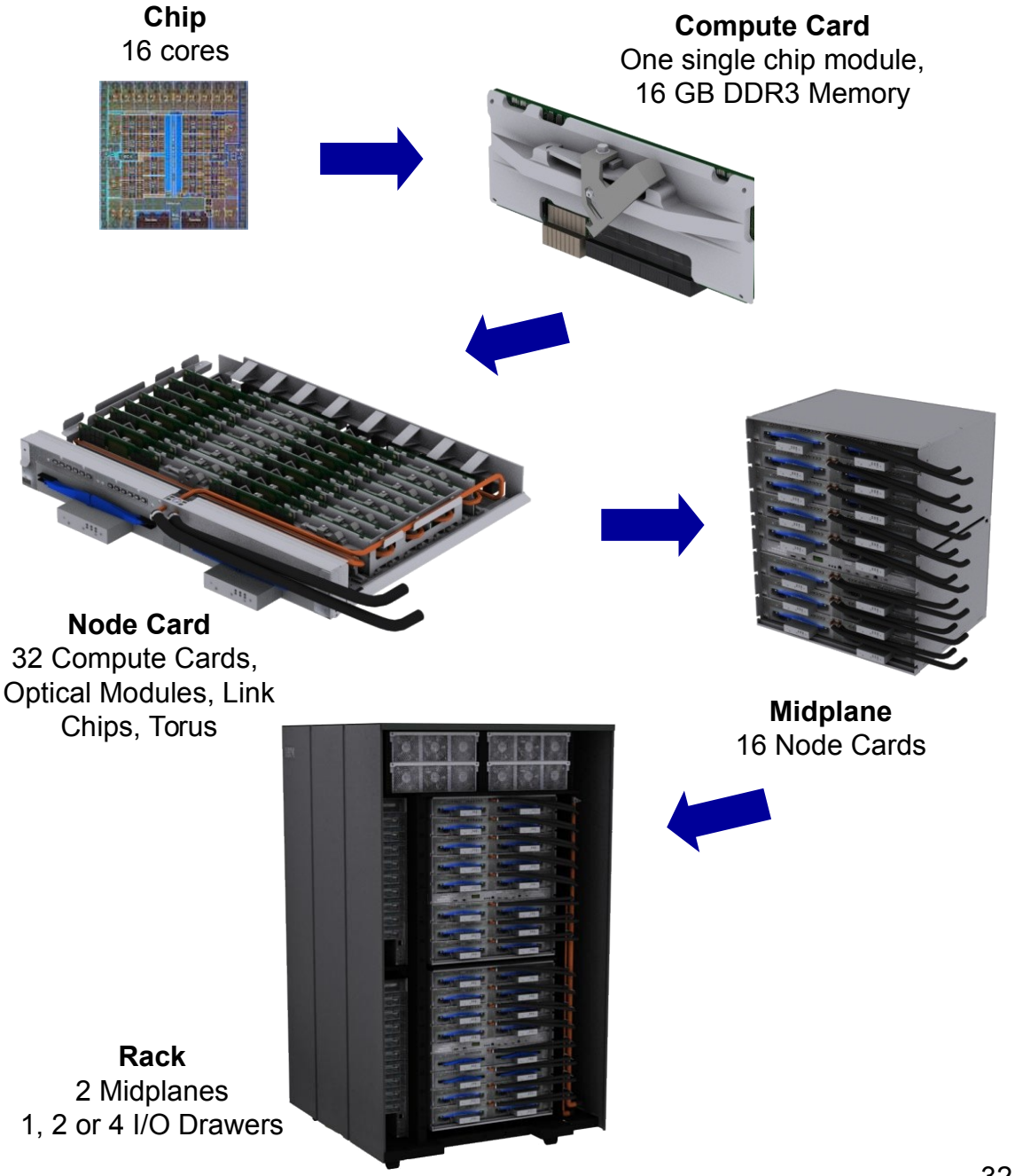
Blue Gene/Q

Rack parameters

- 1,024 nodes = 16,384 cores
- 210 TFlop/s
- Typically: 68 kW (JSC 2012 average)

Features

- Direct liquid cooling
- Network cabling:
  - Copper inside midplane
  - Otherwise optical



Blue Gene/Q (cont.)

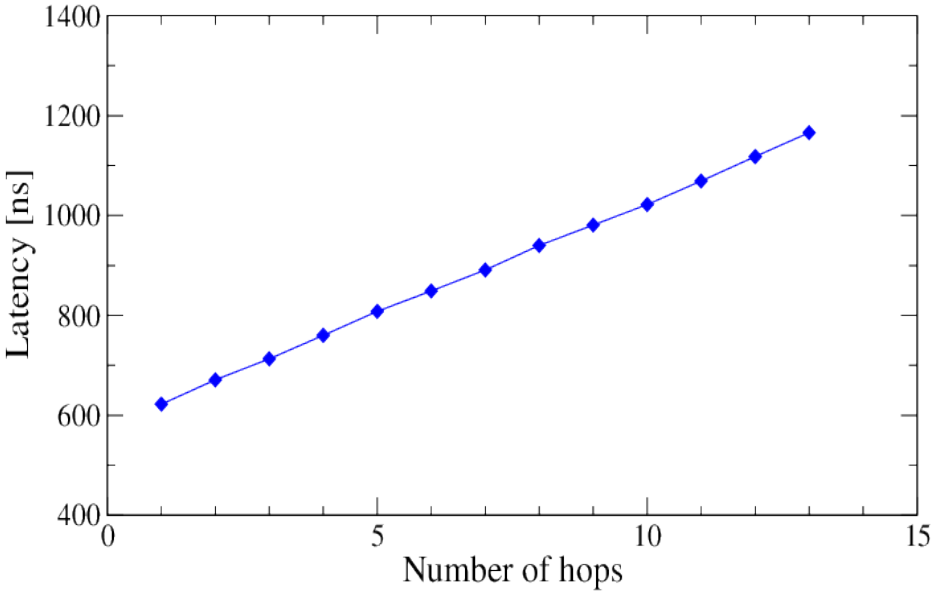
Network features

- On-chip network
- High-speed link technology
- Ultra-low
- 5-dimensional torus

Large installations

- “Sequoia” at LLNL
  - 96 racks
  - 1.6 10^6 cores
  - 6.3 10^6 FMA/cyle → 20 PFlop/s @ 1.6 GHz
- “JUQUEEN” at JSC: 28 racks

Ping-pong latency:



[Dong Chen et al., SC'11]

Comparison of leading Top500 systems

	Sequoia	K Computer	Piz Daint
System architecture	Blue Gene/Q	K Computer	XC30
Vendor	IBM	Fujitsu	CRAY
Top500 (11/2013)	#3	#4	#6
Processor type	Blue Gene/Q A2	Sparc64 VIIIfx	Xeon E5-2650
N_{core}	1 572 864	663 552	42 176
Accelerator type	–	–	K20 GPU
N_{acc}	–	–	5 272
Network topology	5d torus	3d toroidal	dragonfly
Link bandwidth B_{link} [GByte/s]	2	5	4.7-5.25
Floating-point peak B_{fp} [PFlop/s]	20.1	10.6	7.8
Memory capacity C_{mem} [PiByte]	1.5	1.3	0.2
Power [MWatt]	7.9	12.7	2.3

Energy Efficiency

Energy-to-solution vs. power consumption

$$\Delta E = \int_{t_{\text{start}}}^{t_{\text{end}}} P(t)\,dt \leq P_{\max}\Delta t$$

Energy efficiency equals power efficiency if power consumption is roughly constant

$$\epsilon_{\text{E}} = \frac{N_{\text{fp}}}{\Delta E} \simeq \frac{b_{\text{fp}}\,\Delta t}{P\,\Delta t} = \epsilon_{\text{P}}$$

Popular power efficiency ranking: Green500

- Flops/Watt while executing HPL benchmark
- Current #1: TSUBAME-KFC-LX with 4.5 GFlop/s/W

Goal for ~2020: 100 GFlop/s/W

Outlook

Increasing parallelism

- Development driven by power constraints
- May lead to heterogeneous architectures comprising light- and heavy-weight cores

Richer memory hierarchies

- Need for faster, 3-d integrated memory technologies
- Providing both high bandwidth and large capacity requires hierarchy

Faster networks

- Higher bandwidth, but similar latencies as today
- Topologies with less global links
- Better integration of “accelerators”