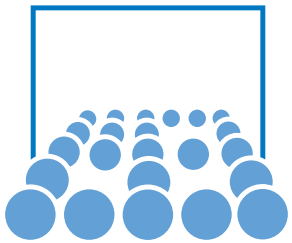
Parallel Architectures

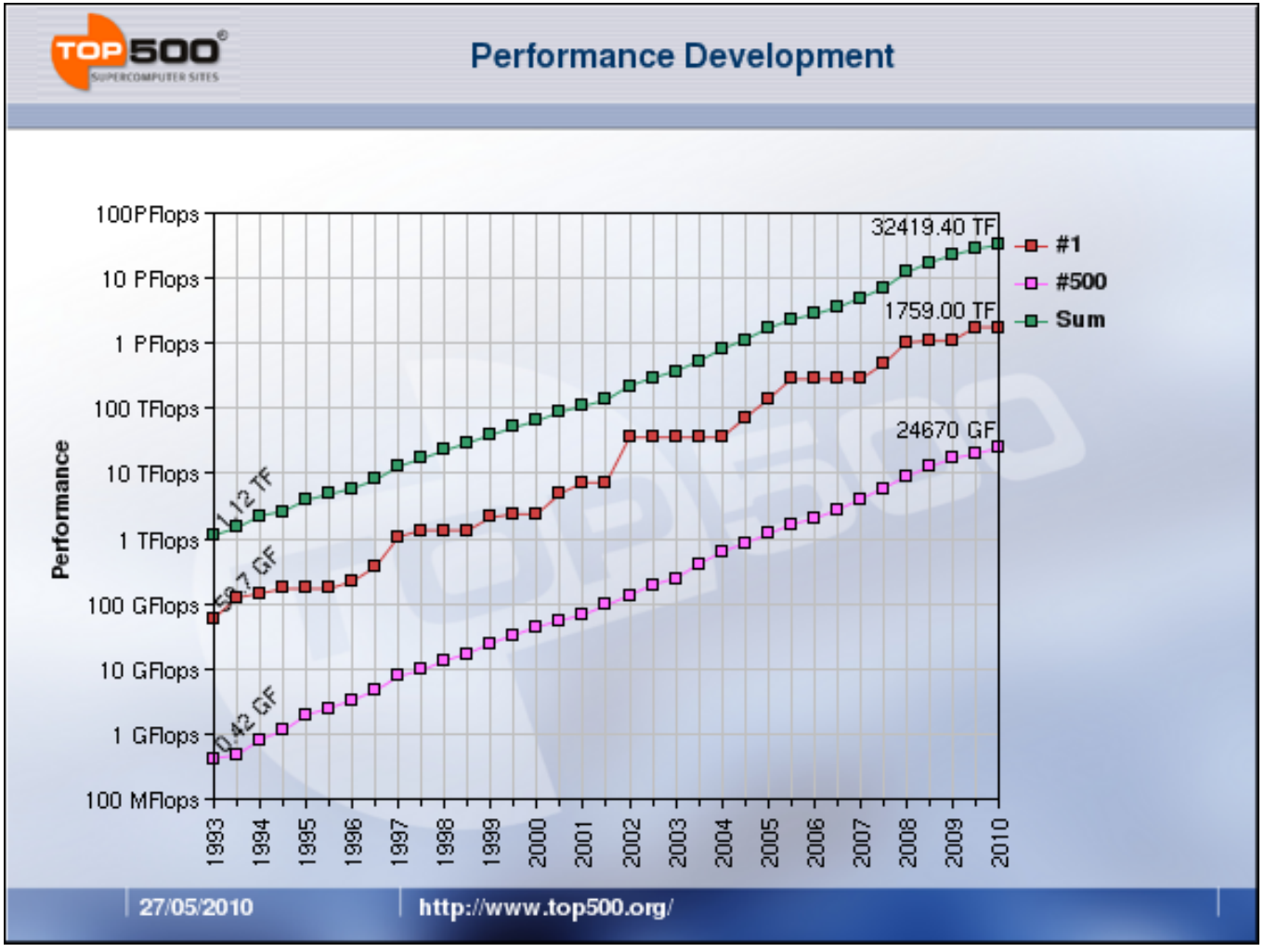
Masterpraktikum - High Performance Computing

Wolfgang Eckhardt
Thomas Auckenthaler
Technische Universität München

9. November 2010



Top500 List


TOP500
SUPERCOMPUTER SITES
Performance Development
Performance
100 PFlops
10 PFlops
1 PFlops
100 TFlops
10 TFlops
1 TFlops
100 GFlops
10 GFlops
1 GFlops
100 MFlops
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
1.12 TF
59.7 GF
0.42 GF
32419.40 TF
1759.00 TF
24670 GF
#1
#500
Sum
27/05/2010
http://www.top500.org/

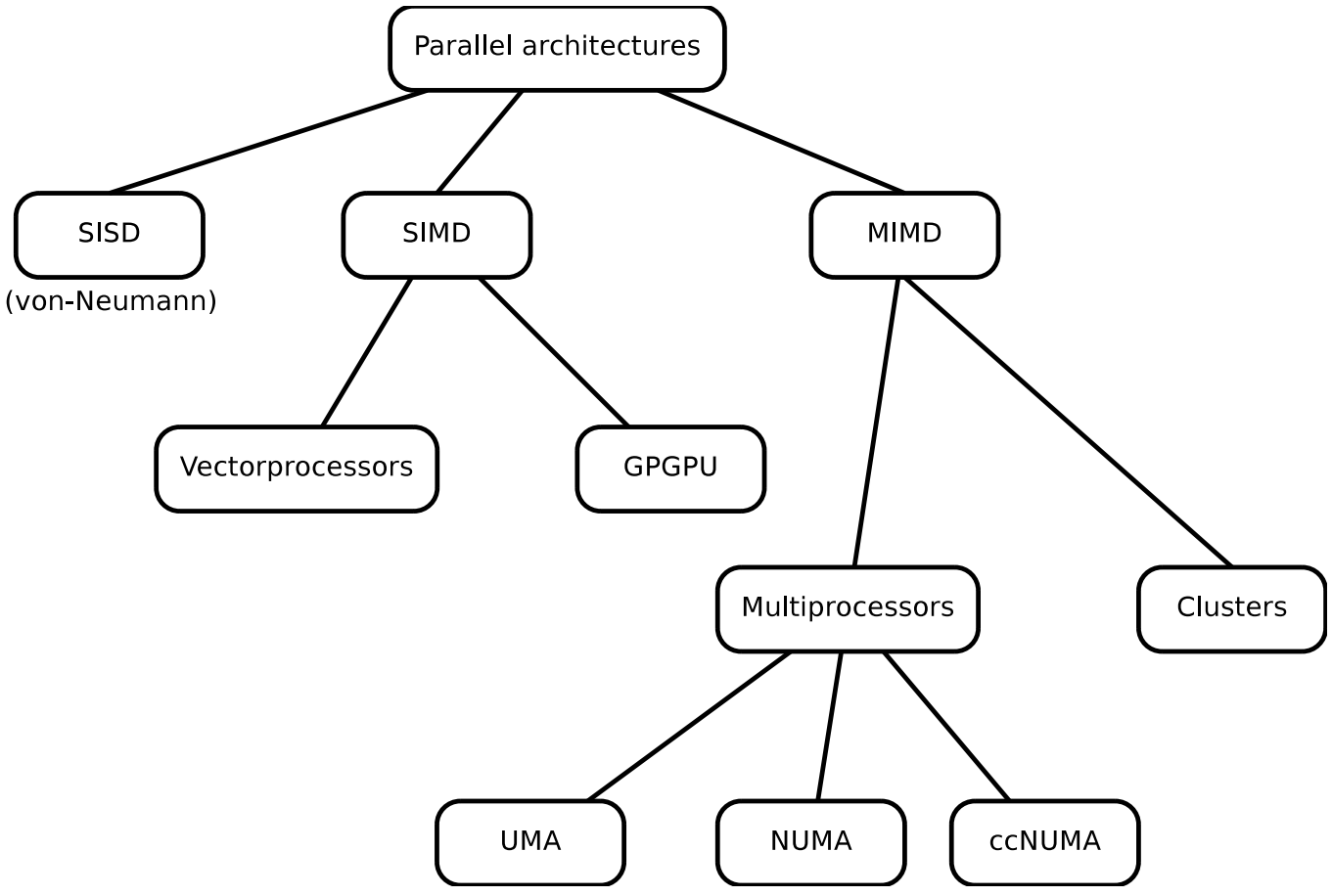
Top500 List

#		Location	R_{max}	R_{peak}
1	Jaguar - Cray XT5, Six core Opteron	USA	1759	2331
2	Nebulae - Nvidia Tesla	China	1271	2984
3	Roadrunner - Cell	USA	1042	1375
4	Kraken - Cray XT5, Six core Opteron	USA	831	1028
5	Jugene - BlueGene/P	Germany	825	1002
6	Pleiades - SGI Altix, Xenon	USA	772	973
7	Tianhe - ATI Radeon	China	563	1206
8	BlueGene/L	USA	478	596
9	Intrepid - BlueGene/P	USA	458	557
10	Red Sky - Sun, Xenon	USA	433	497
...				
92	HLRB2 - Altix 4700	Germany	56	62

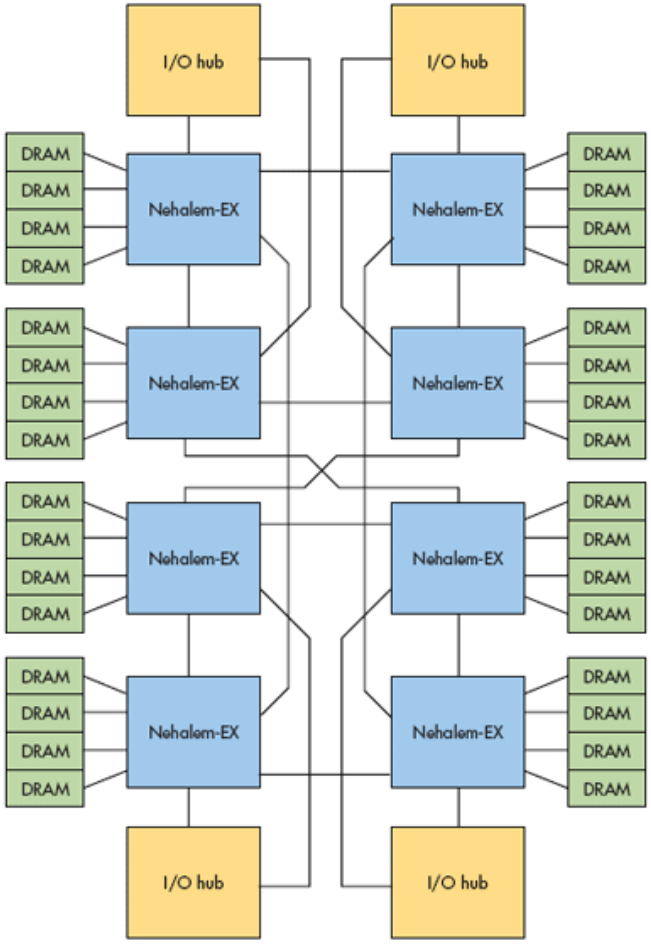
Theoretical Peak vs. sustained Performance

	Device Count	Proj. Runtime NAS-LU [s]	Max. Power [kW]	Theo. Peak [TFLOP/s]
Tesla C2050	256	233.33	59.5	131.9
BlueGene/P	8192	217.26	32.8	27.9
Tesla C2050	4096	143.26	974.8	2110.3
BlueGene/P	16384	118.29	65.6	55.7

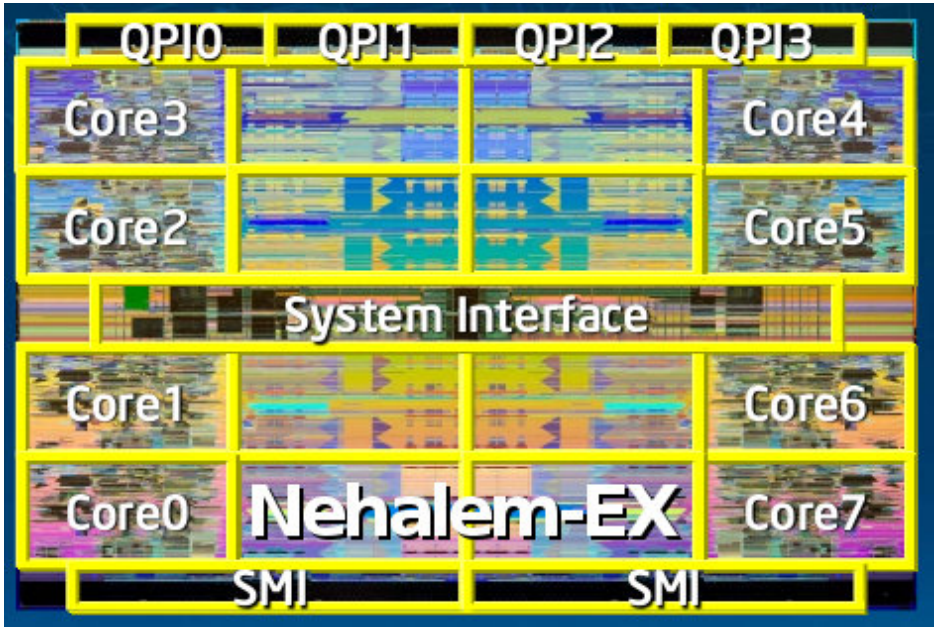
Flynn's Taxonomy



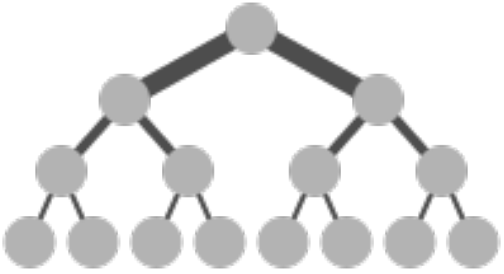
Example: Nehalem-Cluster

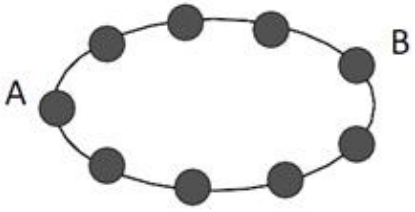


1. Intel's Xeon Nehalem-EX sports eight cores that run two threads each. Each chip has four QuickPath links. An eight-chip system like this one runs 128 threads.

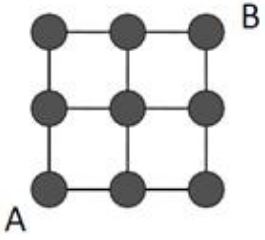


Network Topologies

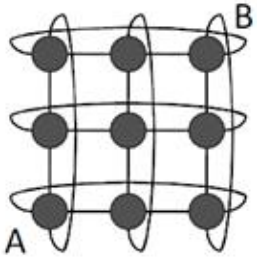




(a) A ring



(b) A mesh



(c) A torus

Parallel Programmin Models

- **Shared memory parallelization**
  - Within nodes
  - Work sharing constructs
  - OpenMP, e.g.
- **Distributed memory parallelization**
  - Between nodes
  - Message passing
  - MPI (Message Passing Interface), e.g.

Performance, Scalability,...

- **Runtime**: $T(p, N)$ = time to solve problem of size N on p processors
- **Parallel speedup (strong scaling)**: $S(p, N) = T(1, N)/T(p, N)$
- **Parallel efficiency**: $E(p, N) = S(p, N)/p$
- **Weak scaling**: $T(p, p \times N)/T(1, N)$, problem size per process is fixed
- **Scaleup**: $Sc(p, N) = N/n$ with $T(1, n) = T(p, N)$