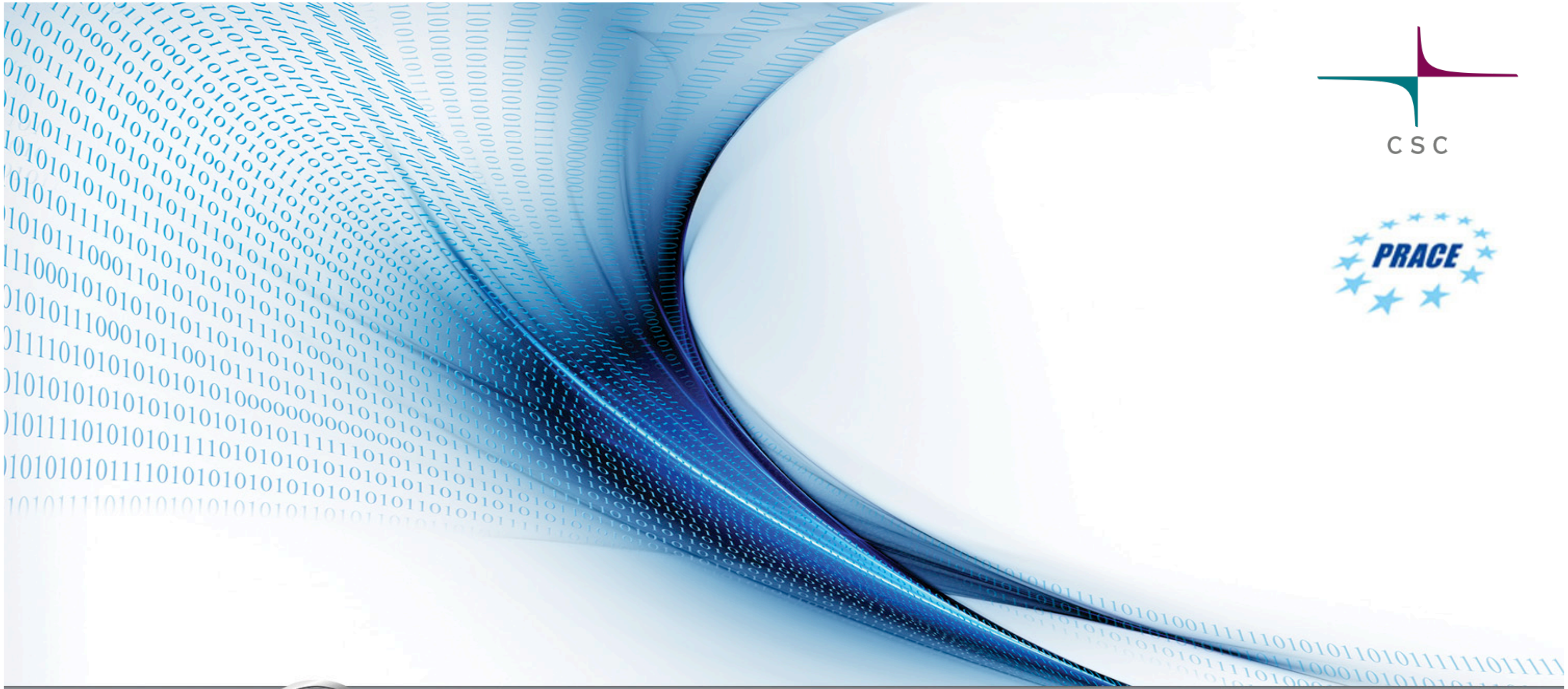


CSC

PRACE

State of the art in accelerated computing

Goal

- Provide a overview of the most prominent hardware accelerators
  - NVidia Tesla, Intel Xeon Phi
  - Touch on some others as well
- Discuss most recent and future developments
- Showcase work by some of our pioneering accelerator pilot users

Agenda

- Intel Xeon Phi (Knight’s Corner)
  - What is it?
  - Programming models
- Nvidia Tesla (Kepler)
  - What is it?
  - Programming models
- Others
- Comparison of Phi and GPGPU
- Looking into the future

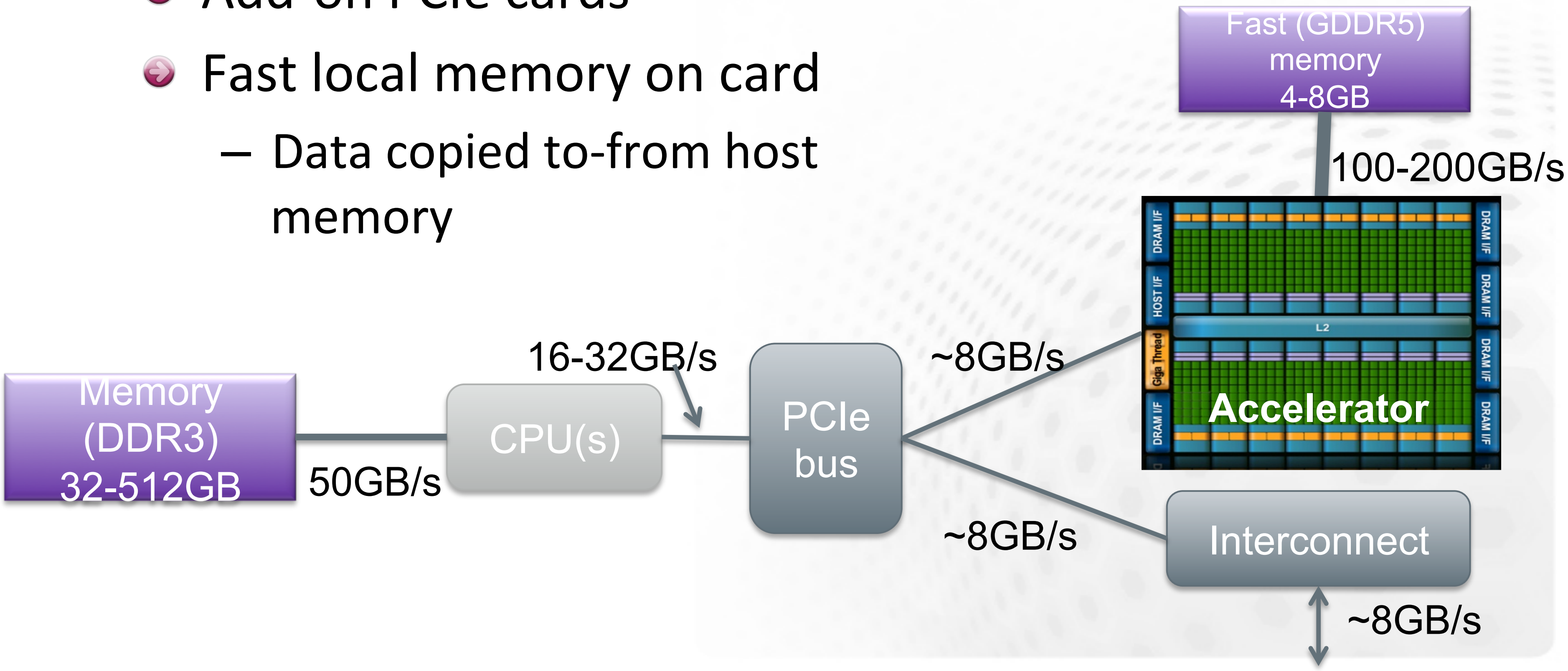
INTRODUCTION

Accelerators & Coprocessors

- Dedicated logic for specific workloads
  - In HPC this means: Flop/s, mem BW, high parallelism
- Tradeoffs
  - Limitations in general-purpose compute capabilities
  - Programs must typically be adapted to some extent
- Different families of technologies
  - GPGPU (**Nvidia Tesla**, AMD)
  - Manycores (**Intel MIC**, Adapteva)
  - FPGA (Convey etc.)

Accelerators and Coprocessor model today

- Add-on PCIe cards
- Fast local memory on card
  - Data copied to-from host memory



Fast (GDDR5) memory 4-8GB

100-200GB/s

Memory (DDR3) 32-512GB

50GB/s

CPU(s)

16-32GB/s

PCIe bus

~8GB/s

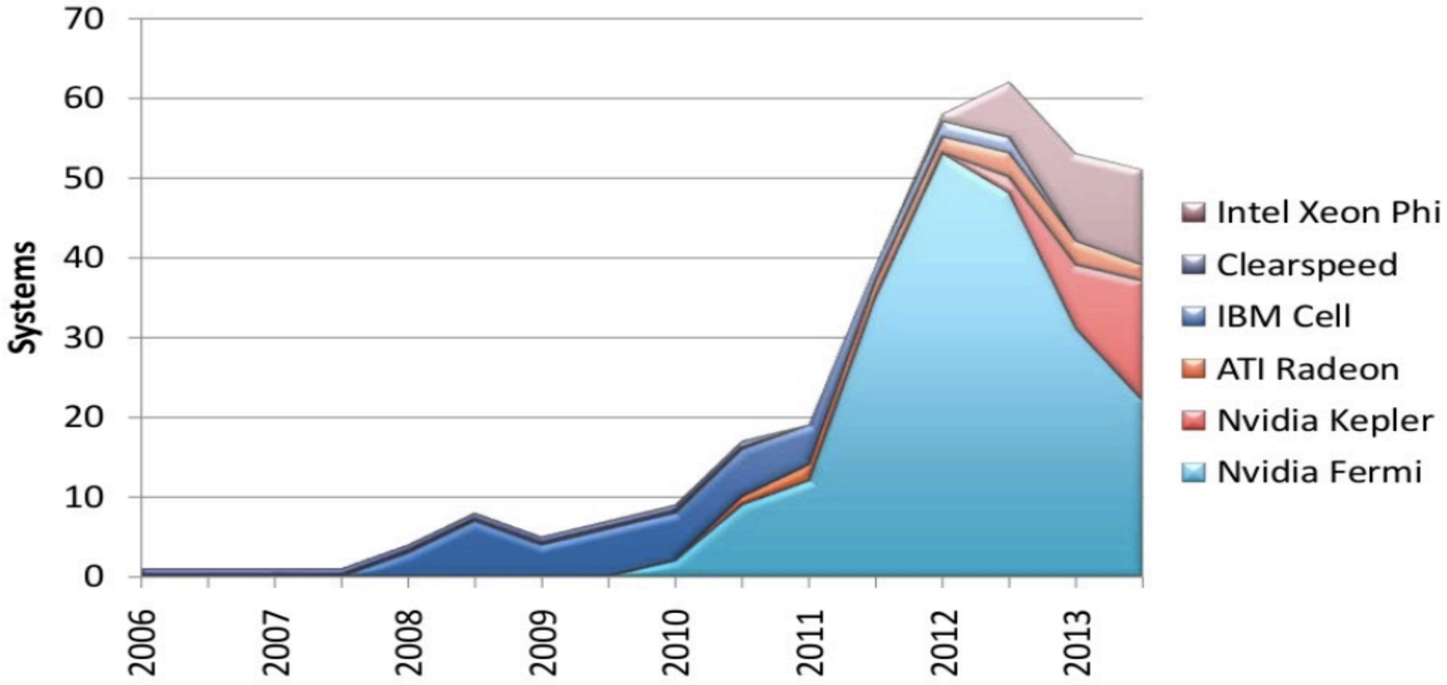
Accelerator

~8GB/s

Interconnect

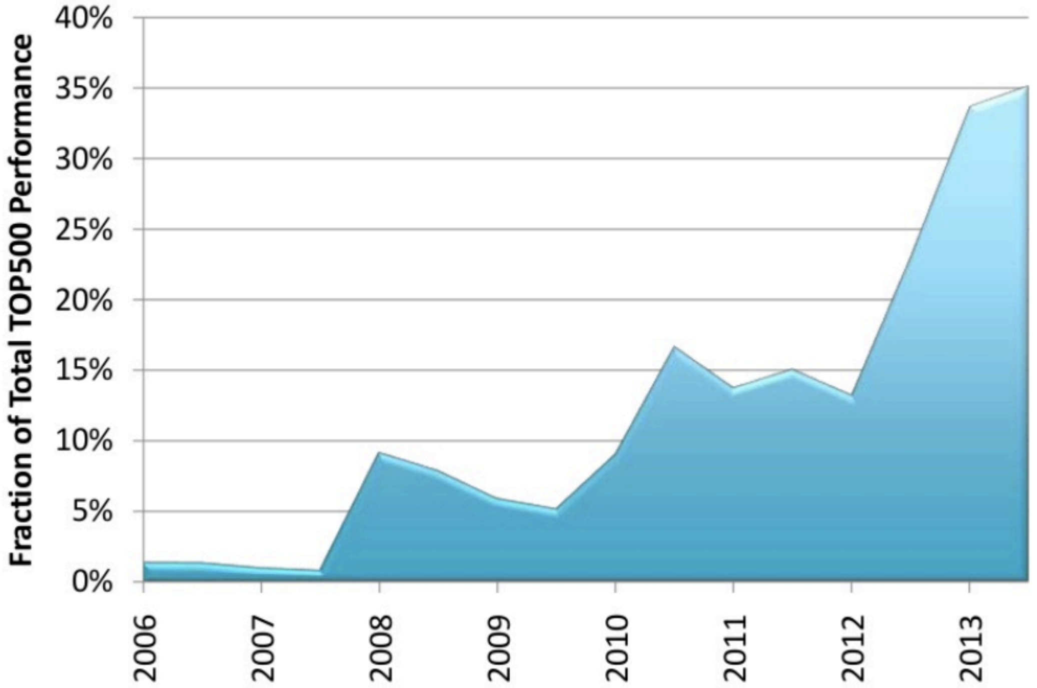
~8GB/s

Accelerated system count in Top500



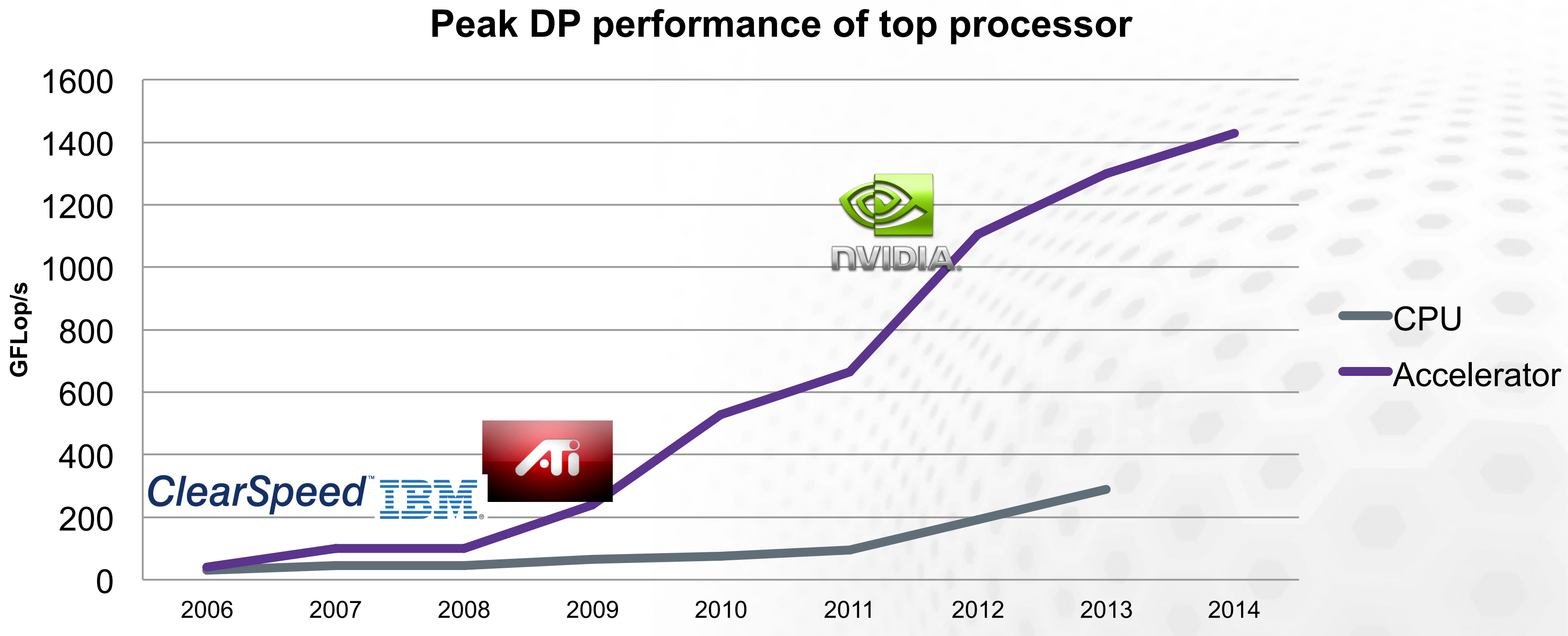
Source: http://www.top500.org

Performance Share of Accelerated Systems in Top500

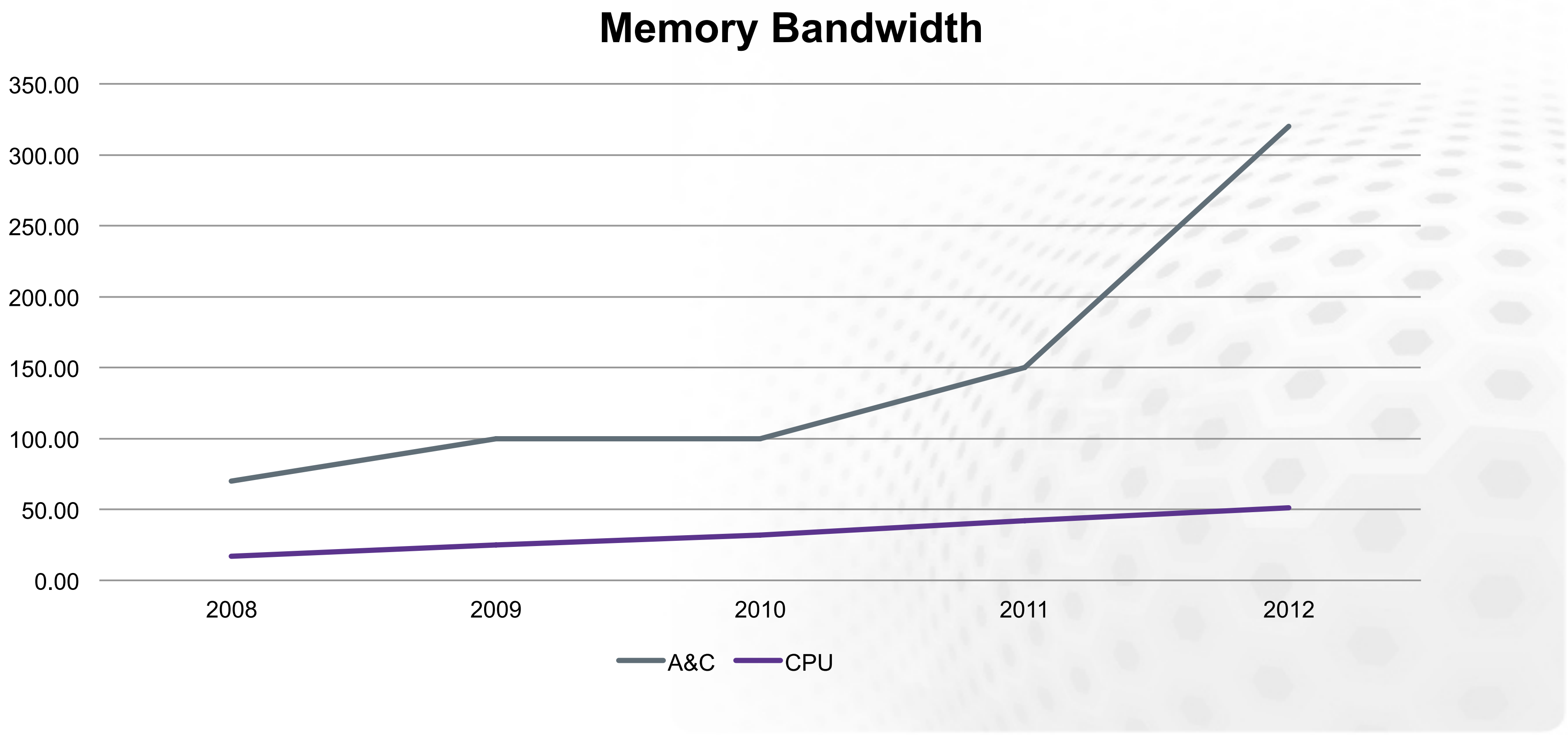


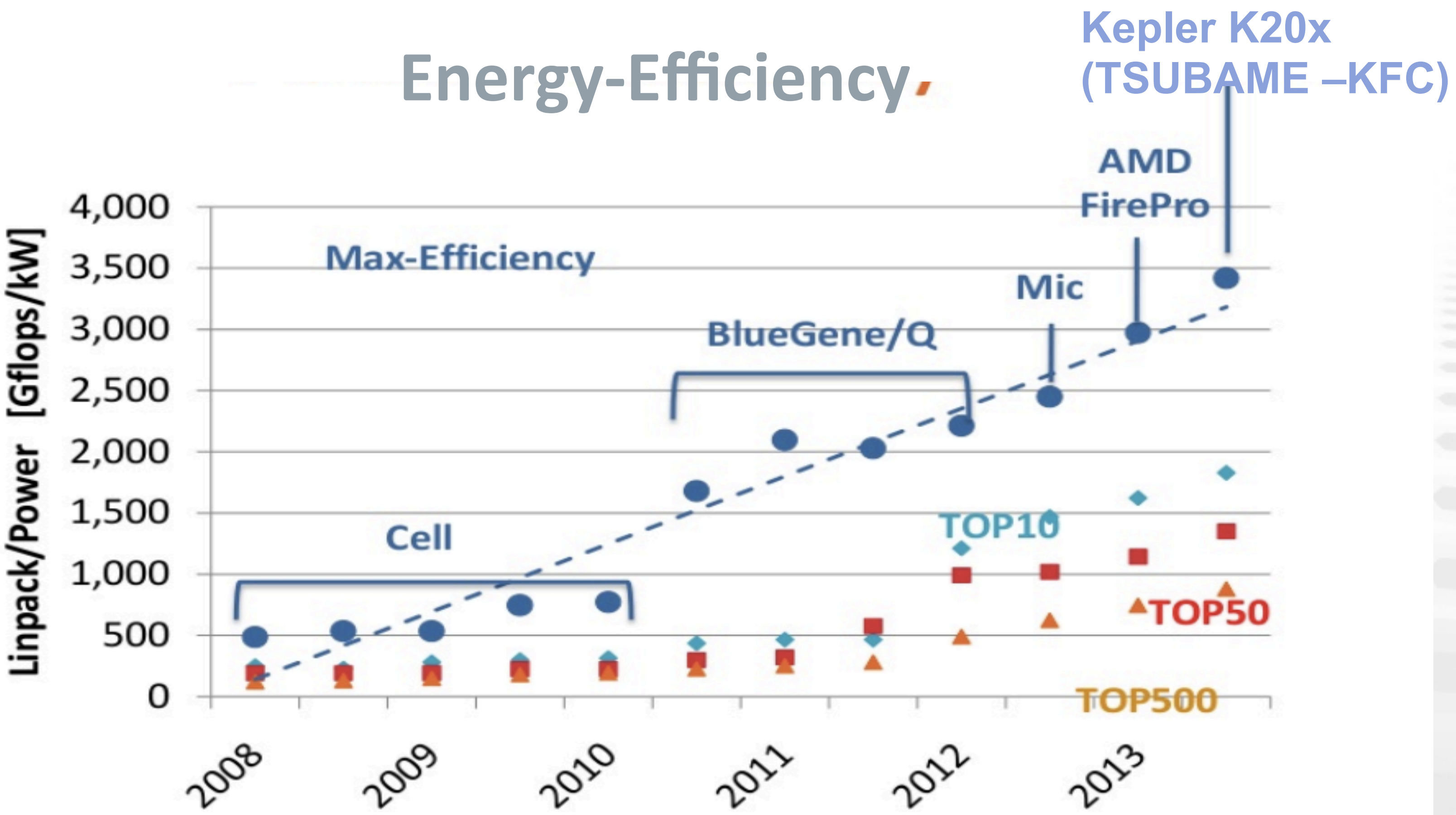
Source: http://www.top500.org

Evolution of Performance



Evolution of Performance

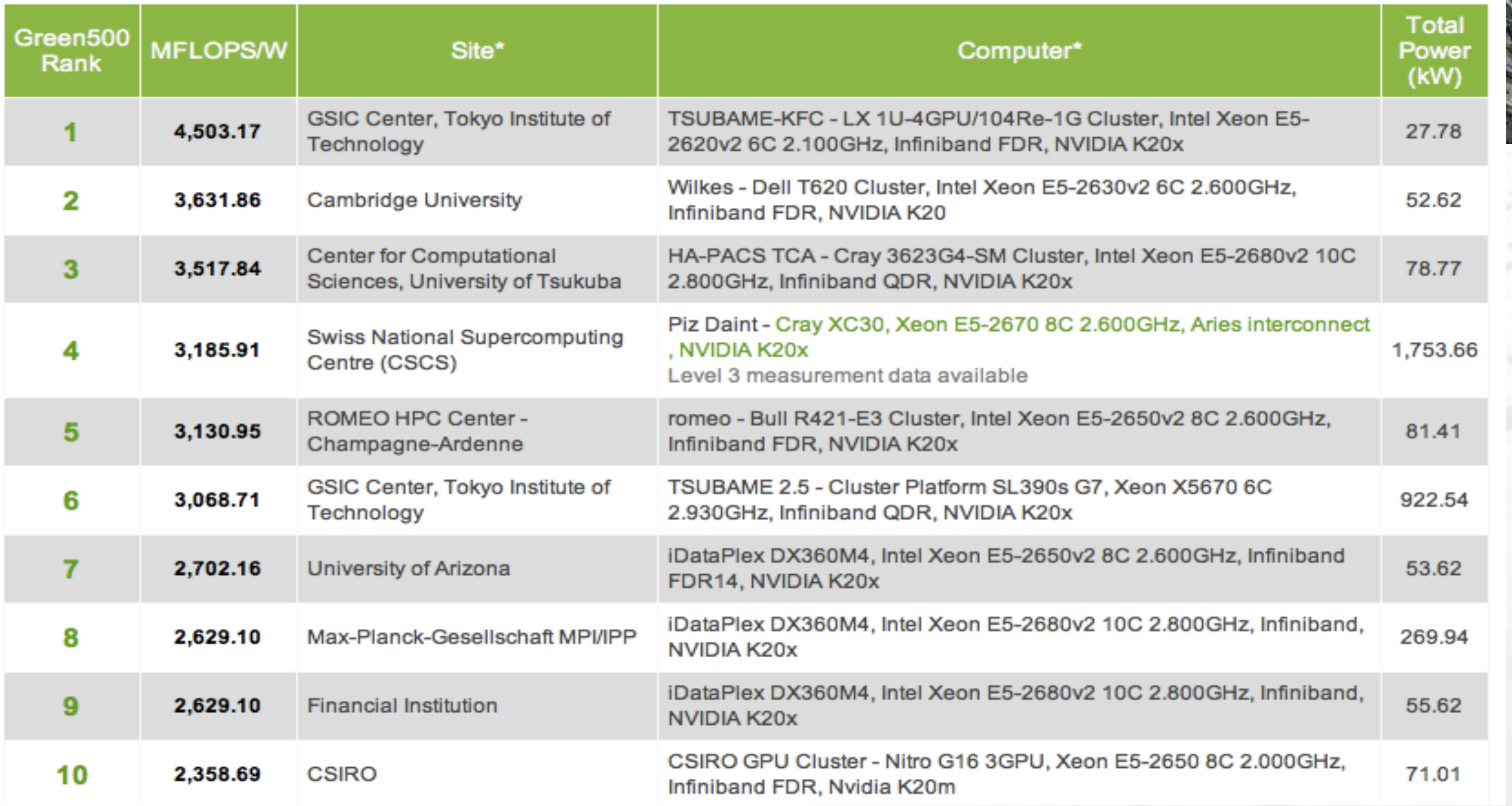




Source: http://www.top500.org

Source: green500.org

Green500 11/2013



Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	4,503.17	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	27.78
2	3,631.86	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
3	3,517.84	Center for Computational Sciences, University of Tsukuba	HA-PACS TCA - Cray 3623G4-SM Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x	78.77
4	3,185.91	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Level 3 measurement data available	1,753.66
5	3,130.95	ROMEO HPC Center - Champagne-Ardenne	romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x	81.41
6	3,068.71	GSIC Center, Tokyo Institute of Technology	TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.930GHz, Infiniband QDR, NVIDIA K20x	922.54
7	2,702.16	University of Arizona	iDataPlex DX360M4, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR14, NVIDIA K20x	53.62
8	2,629.10	Max-Planck-Gesellschaft MPI/IPP	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	269.94
9	2,629.10	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	55.62
10	2,358.69	CSIRO	CSIRO GPU Cluster - Nitro G16 3GPU, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, Nvidia K20m	71.01

What About Programming?

- Explicit kernels
  - Code in a specific language or language extension
  - I.e. CUDA, OpenCL, PyCUDA, Cilk
- Directives
  - Hints to the compiler embedded in regular code
  - I.e. OpenACC, OpenMP accelerator extensions
- Libraries
  - Calls to accelerator-enabled libraries
  - I.e. CUBLAS, MKL

Recent Developments

- Improvements in programmer productivity
  - Improvements in compilers, profilers, debuggers
  - Directive-based languages (OpenACC, OpenMP 4.x)
  - New hardware features
  - Increasing library support and application ecosystem
- Major deployments in US and EU
  - Oak Ridge Titan, CEA Curie, TACC Ranger, CSCS Piz Daint
  - More commonly used applications ported

Things evolve at a very rapid pace!

Conventional wisdom may be misleading!

INTEL XEON PHI

Xeon Phi Nomenclature 101

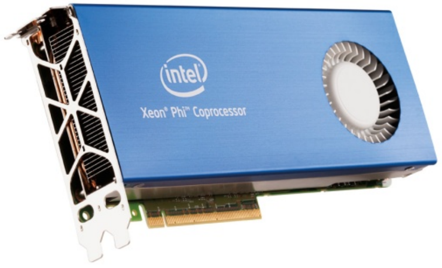
- It is a **coprocessor**, not a GPU or accelerator
- **MIC** is the name of the architecture
  - Comparable to Intel64 on CPU side
- **Xeon Phi** is the brand name of the product
- Architecture generation named as **Knight's …**
  - Comparable to ”Nehalem”, ”Sandy Bridge” etc. on CPU side
- Different models have number designations
  - i.e. 5110P, SE10, 7120

Timeline

- 2008 - **Larrabee** GPGPU announced
  - Was not productized
- 2010 - **MIC** and the **Knight's** series announced
  - Re-architected for pure computing
  - **Knight's Ferry** (**KNF**) development kit
- 2011 - **Knight's Corner** (**KNC**) development kit
  - Alpha/beta versions of the final products
- 2012 **- Intel Xeon Phi** brand introduced
  - First products based on the KNC architecture
- 201x - **Knight's Landing** (**KNL**)
  - Both coprocessor and in-socket versions

Xeon Phi (Knight’s Corner, KNC)

- First commercially available generation of Xeon Phi
- Officially introduced at ISC12, released at SC12
- Many Integrated Core (MIC) architecture
- Coprocessor on a PCI express card
- 10s of x86-based cores with
  - hardware multithreading
  - instruction set extensions for HPC
- Very high-bandwidth local GDDR5 memory
- Runs a stripped-down version of Linux
  - You can ssh in!



Intel MIC Philosophy

- Design the hardware for HPC
  - Strip out ”general purpose” CPU features
    - Branch prediction, out-of-order execution etc.
  - Can pack in many more ”dumb” cores
- Use x86 architecture and programming models
  - Use common code base with CPU
    - Same tools and libraries as on CPU
    - Same parallel paradigms (MPI, OpenMP, Cilk etc.)
  - Optimization strategies similar to CPU
    - **Optimizations for Phi tend to improve CPU performance**

KNC Xeon Phi models

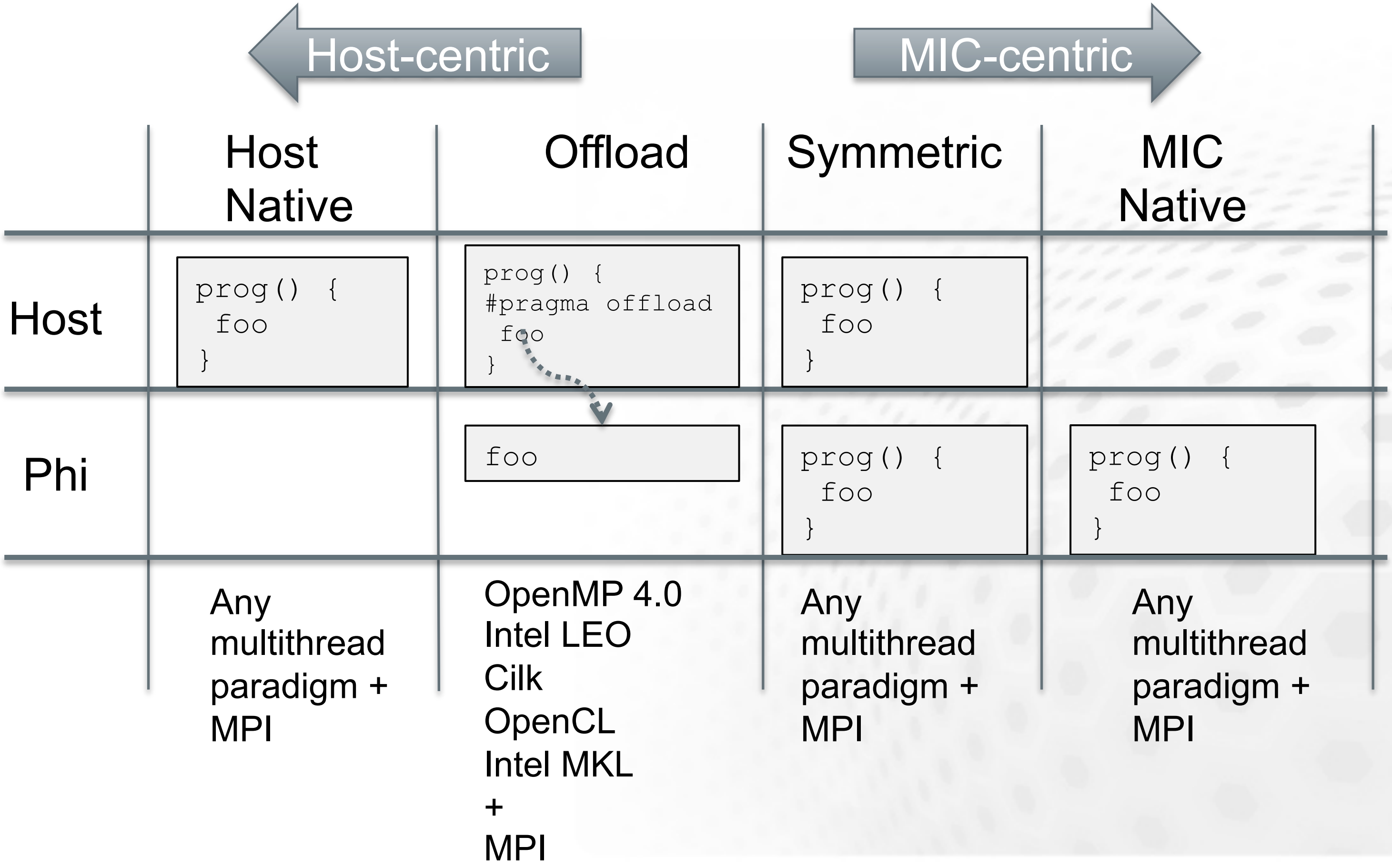
	7120	SE10 (special edition)	*5110*	3120
Cores	61	61	60	57
Clock Rate (GHz)	1.238 (1.33)	1.1	1.053	1.1
Raw perf (Gflops)	1200	1073	1011	1002
Memory Size (GB)	16	8	8	6
L2 Size (MB)	30.5	30.5	30	28.5
Memory BW (GB/s)	352	352	320	240
TDP Power (W)	300	300	225	240
Cooling	Passive (7110P) Vendor defined (7110X)	Passive (SE10P) Vendor defined (SE10X)	Passive	Passive, Active

Notable systems with Xeon Phi

- NUDT Tianhe-2
  - June 2013
  - Guangzhou, China
  - 16k nodes: 48k Phis, 32k CPUs
  - 33.8 Pflops (#1 in Top500)
  - Fully customized design & interconnect by Inspur

- TACC Stampede
  - January 2013 (First big Xeon Phi system)
  - Austin, Texas
  - 6.4k nodes: 12.8k Phis, 12.8k CPUs
  - 8.5 Pflops (#7 in Top500)

Programming Models



Host-centric

MIC-centric

	Host Native	Offload	Symmetric	MIC Native

Host	prog() {
	  foo
	}
		prog() {
		#pragma offload
		  foo
		}
			prog() {
			  foo
			}	

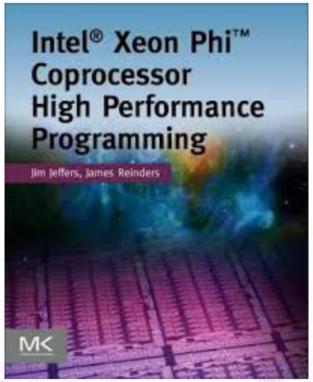
Phi		foo	prog() {
			  foo
			}
				prog() {
				  foo
				}

	Any multithread paradigm + MPI	OpenMP 4.0 Intel LEO Cilk OpenCL Intel MKL + MPI	Any multithread paradigm + MPI	Any multithread paradigm + MPI

KNC versus a regular Xeon

- CPU
  - Lower clock rate (~3x lower clock rate)
  - Larger amount of threads (~15x more available threads)
  - In-order execution, with thread stalls on a L1 cache miss (Xeon is superscalar out-of-order)
  - Both Xeon Phi and Xeon have hardware prefetching
- Memory
  - Less main memory per thread (~20x less memory)
  - Less main memory bandwidth per thread (~2x less main memory bandwidth)
  - Higher main memory latency per thread (~2x higher main memory latency)
- Vector instructions
  - 512-bit vs 256-bit AVX/AVX2 (~2x wider vector units)
  - Richer vector instruction set (FMA, masks, gather scatter, etc.)

Useful Links


Intel® Xeon Phi™ Coprocessor High Performance Programming
Jim Jeffers, James Reinders
MK

- James Jeffers, James Reinders, *"Intel Xeon Phi Coprocessor High Performance Programming"*, Morgan Kaufmann, 2013.
- PRACE Xeon Phi Best Practices Guide
  - http://www.prace-ri.eu/Best-Practice-Guides
- Dr. Dobb's Xeon Phi Guide
  - http://www.drdobbs.com/parallel/programming-intels-xeon-phi-a-jumpstart/240144160
- Phi programming for CUDA developers
  - http://www.drdobbs.com/parallel/cuda-vs-phi-phi-programming-for-cuda-dev/240144545

NVIDIA TESLA GPU

Timeline

- **~2003** First experiments to use programmable shaders
- **2006** CUDA introduced
- **2007** Tesla architecture
  - 1st dedicated GPGPUs
  - Compute capability: 1.x
- **2009** Fermi architecture
  - ECC memory
  - Compute capability: 2.x
- **2012** Kepler architecture
  - Various HPC enhancements, lower clock, more cores
  - Compute capability: 3.x
- **2013** Kepler K40
  - More memory, PCI Express 3

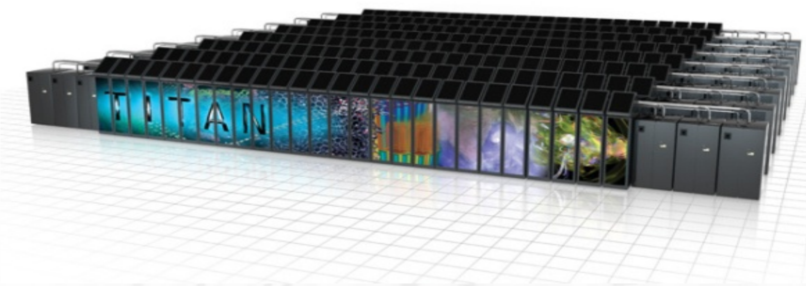
	SP	**DP**
2007 / Tesla	622	77
2009 / Fermi	1288	515
2012 / Kepler	3950	1310
2013 / K40	4290	1430

Evolution of Nvidia Tesla

	K40	**K20X**	**K20**	**M2090**
Stream Processors	**2880**	**2688**	**2496**	**512**
Core Clock	745MHz	732MHz	706MHz	650MHz
Boost Clock(s)	**810MHz, 875MHz**	N/A	N/A	N/A
Shader Clock	N/A	N/A	N/A	1300MHz
Memory Clock	**6GHz** GDDR5	5.2GHz GDDR5	**5.2GHz** GDDR5	3.7GHz GDDR5
Memory Bus Width	384-bit	384-bit	320-bit	384-bit
VRAM	**12GB**	6GB	5GB	6GB
Single Precision	4.29 TFLOPS	3.95 TFLOPS	3.52 TFLOPS	1.33 TFLOPS
Double Precision	**1.43** TFLOPS (1/3)	**1.31** TFLOPS (1/3)	1.17 TFLOPS (**1/3**)	655 GFLOPS (1/2)
Transistor Count	7.1B	7.1B	**7.1B**	3B
TDP	235W	235W	225W	250W
PCI Express	**Gen 3**	Gen 2	Gen 2	Gen 2

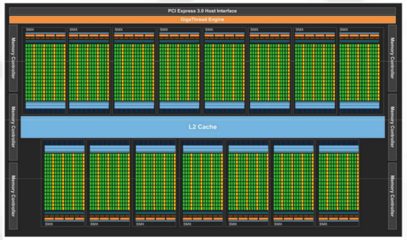
Notable systems with Nvidia Tesla

- ORNL Titan
  - Knoxville, Tennesee, USA
  - Cray XK7 with Kepler K20x
  - 18688 CPUs, 18688 GPUs
  - #2 on the Top500 list
- CSCS Piz Daint
  - Lugano, Switzerland
  - Cray XC30 with K20x
  - #6 on the Top500 list
    - Most powerful in Europe
  - #4 on Green500



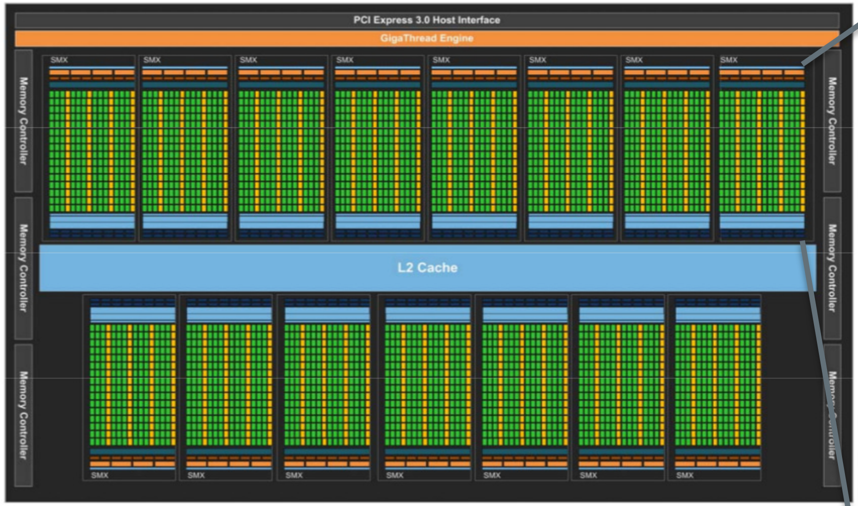
Nvidia Tesla architecture

- Tesla is the name of the product line
  - As well as the 1st generation of cards
- PCI Express connected coprocessor card
  - Very fast local GDDR5 memory
    - 5-16GB, ~200 GB/s
  - 1000s of ”CUDA cores”
    - Grouped into symmetric multiprocessors (SMX)
      - Kind of like a CPU: All threads running on SMX execute same instruction on all the CUDA cores
      - Local memory and register pool on each SMX
      - Synchronization between SMXs is expensive

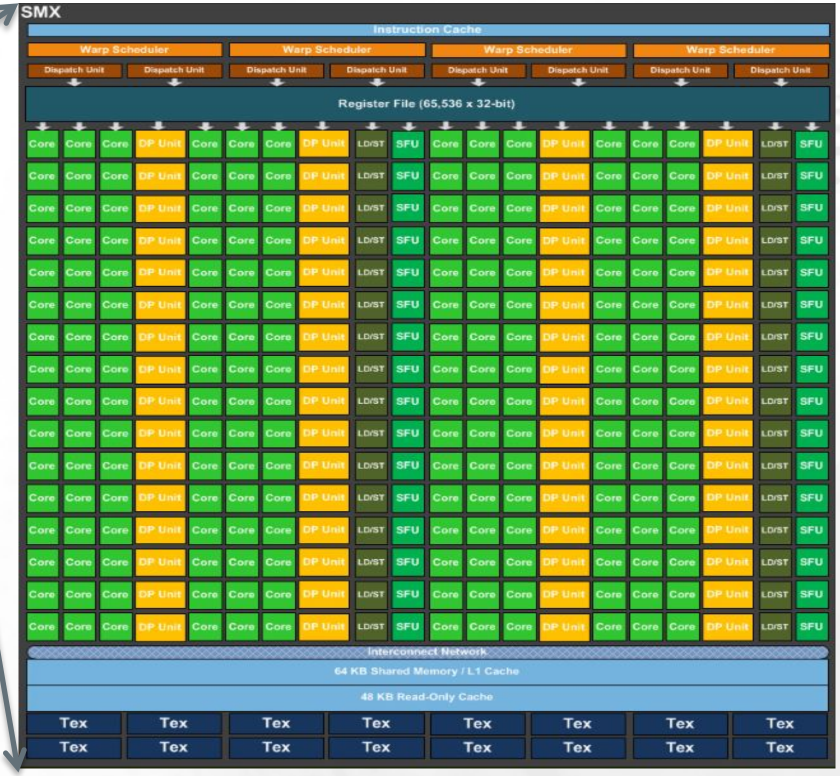


Kepler GK110 Full chip block diagram

Kepler Architecture



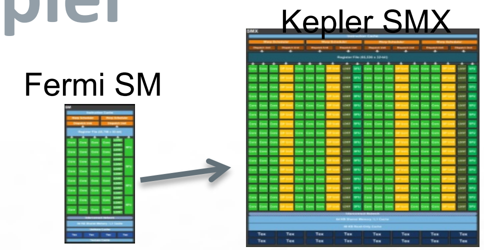
Kepler GK110 Full chip block diagram



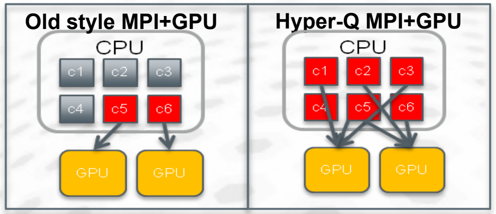
CUDA

- Developed by Nvidia for GPU programming
  - Extensions to C/C++
  - Also x86 and FORTRAN compilers from PGI
- Regions (kernels) that are executed on GPU coded in the special CUDA language
  - C/C++/Fortran with some special extensions
  - Kernels written from a single thread's point of view
    - Each thread has unique ids (kind of like an MPI rank)
      - Thread id and block id

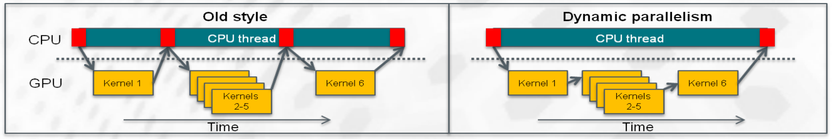
Developments in Kepler



- More cores per multiprocessor (lower freq)
- ECC memory (already in Fermi)
- Larger caches, more registers
- Hyper-Q
  - Multiple processes can run kernels on GPUs



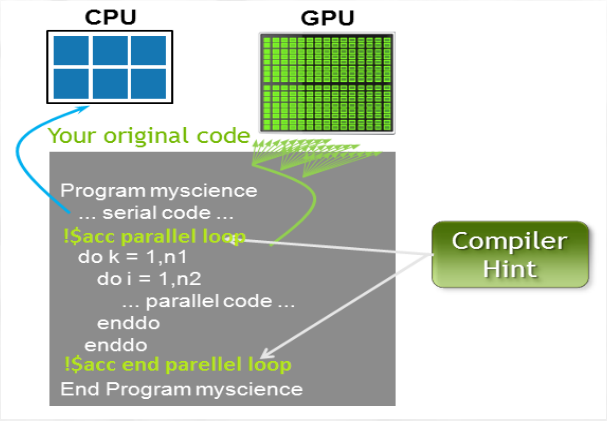
- Dynamic Parallelism
  - Launch kernels inside a kernel



- GPUDirect
  - P2P: GPU-to-GPU communication inside the node
  - RDMA: GPU-to-GPU between nodes over InfiniBand
- Power management features

OpenACC

- Directive-based programming for accelerators
  - Formed in 2011 by Cray, PGI, NVidia, CAPS
  - Recently released version 2.0 of the standard
- Focus on productivity
  - Reasonable performance with little effort
- Compilers
  - CAPS
  - PGI >= v12.6
  - Cray (XK6, XK7 and XC30 w/GPU)



```
Program myscience
  ... serial code ...
!$acc parallel loop
  do k = 1,n1
    do i = 1,n2
        ... parallel code ...
    enddo
   enddo
!$acc end parellel loop
End Program myscience
```

www.openacc-standard.org/
http://www.caps-entreprise.com
www.pgroup.com/resources/accel.htm

Useful Links

- PRACE GPU Best Practices Guide
  - http://www.prace-ri.eu/Best-Practice-Guides
- Nvidia Developer Guide
  - http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf
- Dr. Dobb's Introduction to OpenACC
  - http://www.drdobbs.com/parallel/easy-gpu-parallelism-with-openacc/240001776

COMPARING TESLA AND PHI

MIC – Tesla Translation guide

Tesla	MIC
CUDA core (1 FP op / cycle)	~= MIC (CPU) SIMD lane
CUDA symmetric multiprocessor	~= MIC (CPU) core
CUDA thread block	~= MIC (CPU) threads in a core
One operation on a CUDA Warp	~= One MIC (CPU) SIMD operation
Large oversubscription of work (>4 x resident warps per SM optimal)	Moderate oversubscription of work (2-4x threads per core optimal)
Automatic and manual local caching	Coherent, automatic L2 cache and hardware prefetching
CUDA, OpenCL, OpenACC offloads, Libraries (CUBLAS, CUFFT etc.)	Legacy programming models (OpenMP etc.), LEO offloads, OpenCL , Libraries (MKL etc.)
Host CPU needed for execution	Possible to independently execute native code

Looking at the numbers

- Raw double precision FP performance is similar
  - ~1-1.4 Tflop/s, depending on model
- Single precision (SP) faster on Tesla
  - 3 x DP performance (2 x DP on Phi) on K20
  - Tesla line has K10 cards with even higher SP
- Sustained memory bandwidth is similar
  - ~200GB/s
  - Memory control more automated in MIC
    - Hides complexity but limits advanced optimization

Common challenge: Host complexity

- Systems comprise of multiple islands
  - CPU sockets, memory banks, PCIe IO Hubs
  - Perfomance varies depending on this
- Direct ”peer-to-peer” communication of acclerators not possible between CPU sockets
  - Code still works
  - Degraded performance
    - CPU needs to be bothered

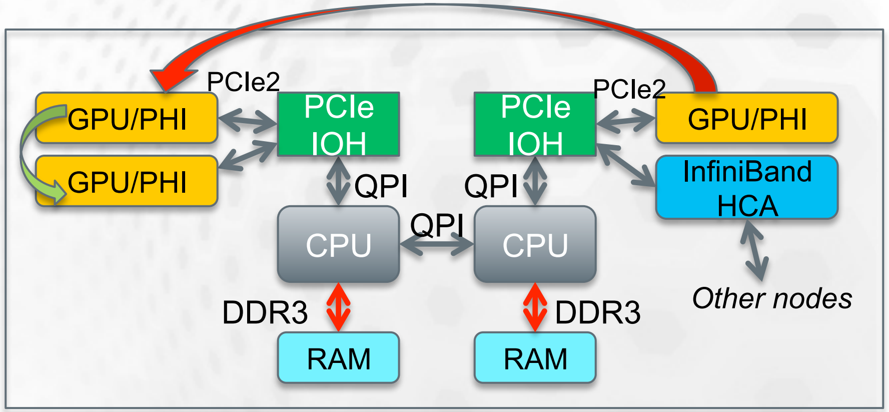
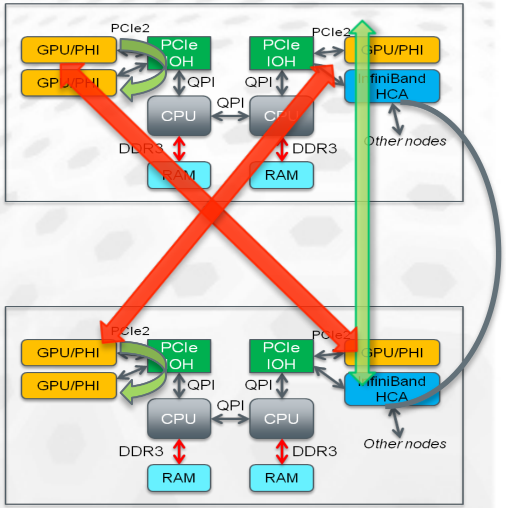


Diagram of a system with 3 accelerators and 2 CPUs

Common Challenge: MPI

- Practical considerations
  - Mapping MPI ranks to specific cards
- Communication efficiency
  - Doing direct communication from accelerator to accelerator possible
    - Much mode efficient than going via CPU and main memory
    - Specific InfiniBand card models, drivers, MPI library version
    - Does not currently work between IO Hubs



Common languages: OpenCL

- Somewhat similar to CUDA
  - Designed to be more platform-agnostic
- Compilers available for multiple platforms
  - AMD GPU, Nvidia GPU, Phi, x86 CPU
- Creating truly portable code can be challenging
  - Vendor-specific support libraries
  - Creating performance portable code uncertain
- At least the code base is the same

OpenMP

Common languages: OpenMP 4 accelerator directives

- Ratified a few months ago
  - Support still limited (Latest Intel compilers)
  - Will replace Intel's own LEO offload model on Phi
- Offload code regions with the *target* pragma

```
double A[N];
…
#pragma omp target device(0) map(tofrom:A)
#pragma omp parallel for
for (i=0;i<N;i++)
    A[i]=i;
```

OpenMP 4 accelerator directives

- Topology can be defined with *teams* and *distribute* pragmas

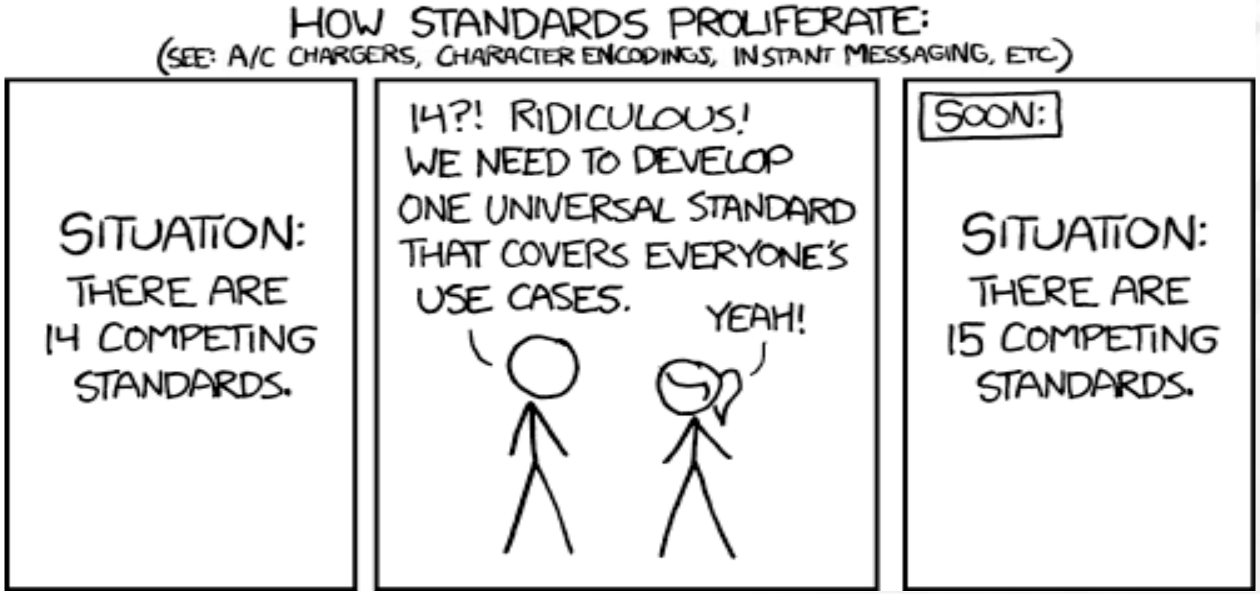
```
#pragma omp target device(0)
#pragma omp teams num_teams(60) num_threads(4) // 60 physical cores, 4 h/w threads each
#pragma omp distribute  // following loop is distributed across the 60 physical cores
for (int i = 0; i < 256 ; i++) {
#pragma omp parallel for  // following loop is distributed across the 4 threads
     for (int j =0; j < 512; j++) {
  …
     }
}
```

- Asynchronous execution using the *task* pragma
- SIMDization of loops with the *simd* pragma

```
#pragma omp simd
for (i=0;i<N;i++)
  a=(j+0.5)/N;
```

One language to rule them all?

- OpenCL, OpenACC, OpenMP 4, OpenMD…
  - No ”one-size-fits-all” solution yet
  - Vendors do not seem to agree about what it should be


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

So, which one is better?

Kepler

- Kepler is a mature product
- Proven performance
- CUDA provides great control for advanced programmers
- Good software ecosystem
- OpenACC seems promising

Phi

- Programmability argument is compelling
  - Simple to maintain a single code tree
  - Optimization still needed
- Variety of programming models to choose from
- Good tools from Intel
- Next generation will be very interesting

ACCELERATORS AT CSC

Existing systems at CSC

- 8 Nvidia Fermi (M2050/2070) GPU nodes on Vuori
  - Will be retired early next year
- 2 Nvidia Quadro GPUs on Hippu
  - Primarily for visualization, will be retired early next year
- T-Platforms PRACE prototype (hybrid.csc.fi)
  - 5 nodes with Xeon Phi 5110
  - 4 nodes with Kepler K20
  - 1 node with Kepler K20x
  - For very small-scale testing and training

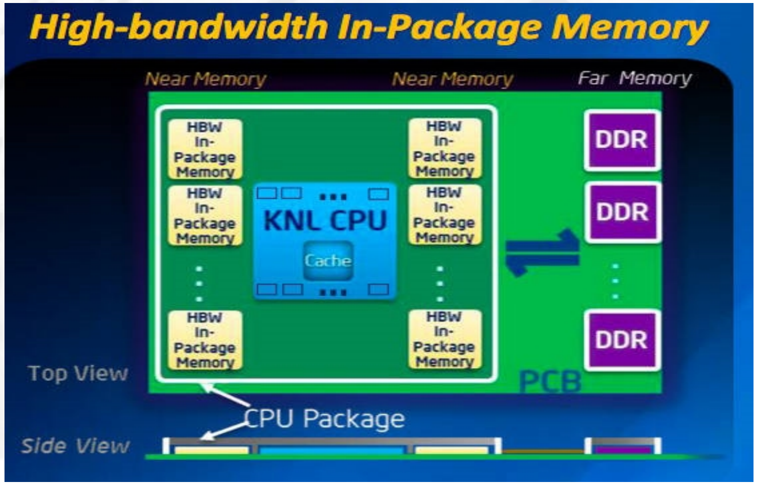
New accelerated supercomputer

- Supplied by Bull SA
- **Phase 1:** 44 Xeon Phi –nodes (+ 1 spare)
  - 2 CPU + 2 Xeon Phi 7120X on each node
  - Currently in acceptance testing
- **Phase 2:** Similar amount of Nvidia -nodes
  - 2 CPU + 2 Nvidia K40
  - 1Q 2014
- FDR InfiniBand
- Extreme energy efficiency
  - Top of the line accelerators
  - Direct liquid cooling
  - Kajaani datacenter

LOOKING INTO THE FUTURE

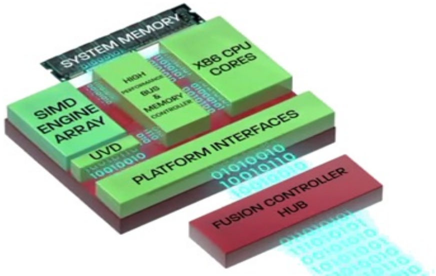
Intel Knight’s Landing

- Next generation of Xeon Phi
- Available as a PCIe card and a standalone CPU
- High-speed ”near memory” on CPU package
- Larger regular (DDR4) ”far memory”
- 14nm process technology



Nvidia Plans

- Tesla Maxwell
  - Unified Virtual Memory
- Tesla Volta
  - Stacked DRAM on top of GPU (~1TB/s)
- IBM Collaboration
  - Combining POWER8 with GPUs
- ARM Collaboration
  - ”Project Denver”
  - 64bit ARM CPU + GPU



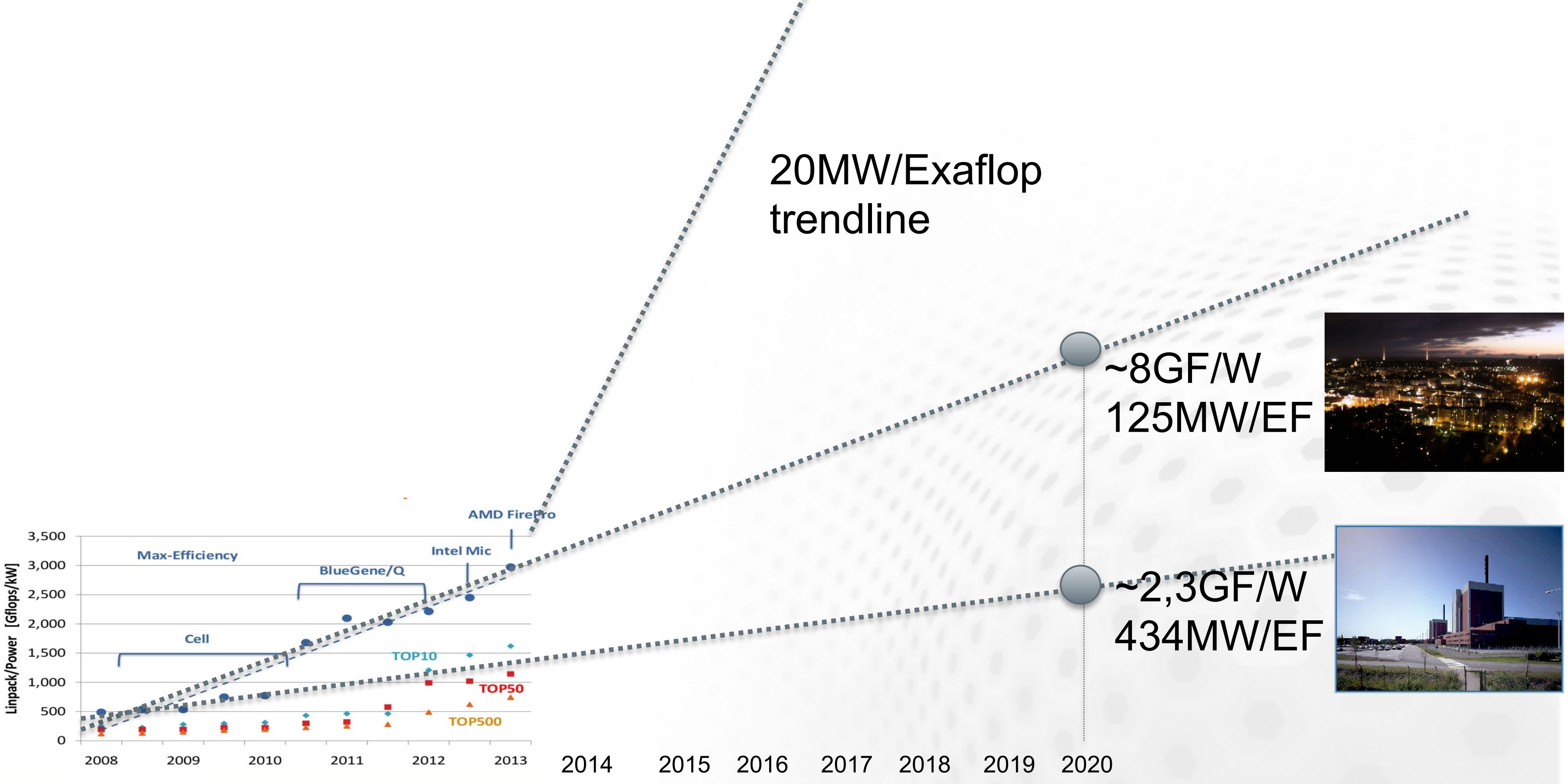
Dark Horse: AMD APU

- AMD has kept a low profile in GPGPU market
  - Promising performance but software stack is lacking
- New AMD APUs are very interesting
  - CPU & GPU on the same die sharing same memory
    - The memory can be GDDR5!
- Deals for Xbox One and PS4
  - Should guarantee that it will be realized
- No idea about HPC versions yet
  - I’d love to get me one!

Exascale power challenge

- *”Exaflop system by ~~2018 2019~~ **2020?**”*
- Setting the bar high
  - – 20MW power consumption (2x)
  - – 1EF Performance (30x)
  - – 1B cores
- Needs a whole-system approach
  - – Infrastructure, hardware, programming models, algorithms
  - – Disruptive technologies are needed

Revisiting the Power Efficiency Slide



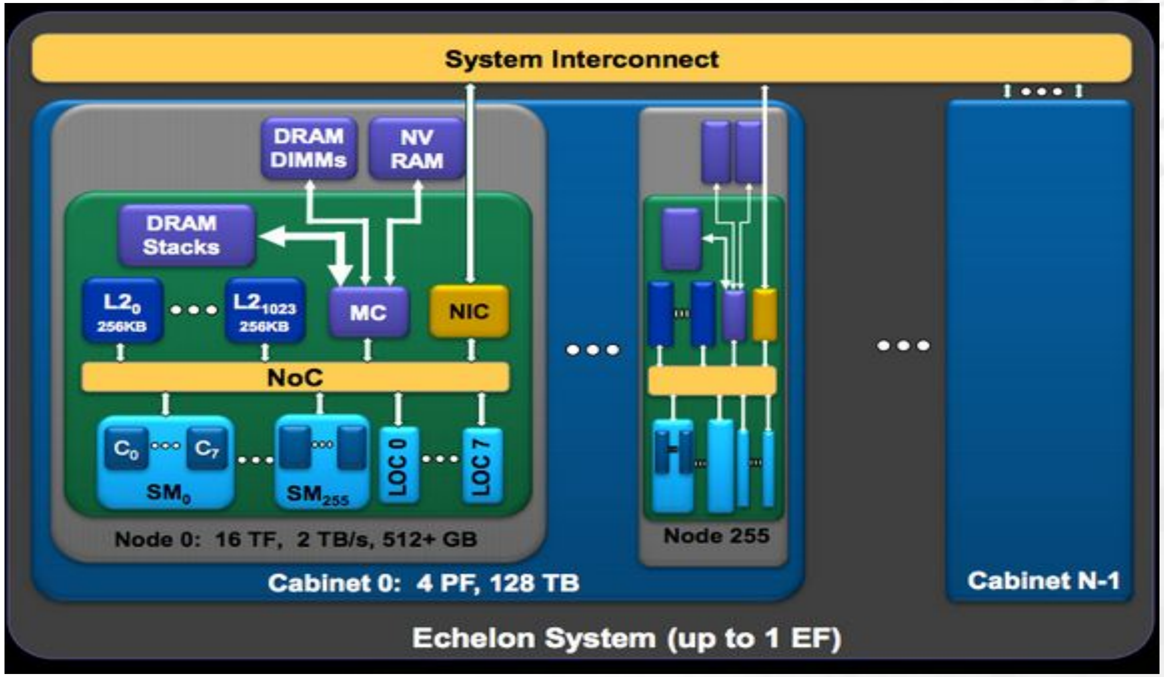
20MW/Exaflop trendline

~8GF/W
125MW/EF

~2,3GF/W
434MW/EF

Project Echelon

- Long-term project by Nvidia & partners for a fully integrated system on a chip (SoC)

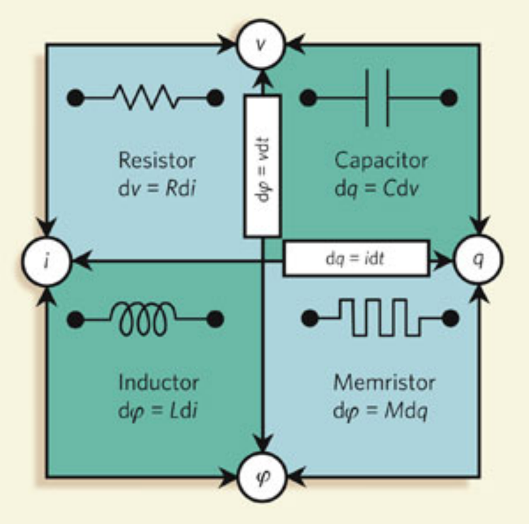


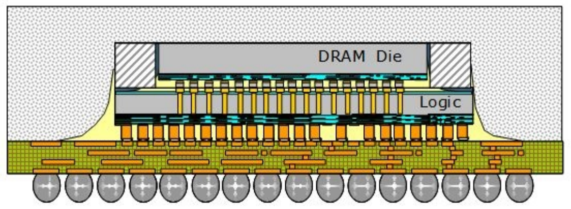
Other Interesting Projects

- PRACE Prototypes
  - Several novel architectures
  - http://www.prace-ri.eu/PRACE-Prototypes
- Mont Blanc
  - Cluster of ARM CPUs (Samsung)
  - http://www.montblanc-project.eu
- DEEP
  - Standalone Xeon Phis using the EXTOLL interconnect
  - http://www.deep-project.eu/ http://www.extoll.de

Emerging disruptive technologies

- Novel memory technologies
  - SSDs
  - Transactional memory
  - Memristors
  - Phase-change memory
- Integrated optics
  - Both on the chip and on circuit board
- 3D stacking
  - E.g. DRAM on top of logic
  - **Micron Hybrid Memory Cube & Automata processor**





KNC 7110 versus Kepler K20x

- Floating point performance
  - DP performance roughly similar (1.2Gflops vs. 1.31 Gflops)
  - SP performance lower (~1.6x lower, 2.4 Gflops vs. 3.95 Gflops)
- Memory
  - Memory bandwidth similar (352Gb/sec vs. 250Gb/sec)
  - Larger memory size (~2.6x larger, 16Gb vs. 6 Gb)
  - Larger L1 cache size (~2.0x larger, 2Mb vs 1Mb in total)
  - Larger L2 cache size (~20x larger, 30Mb vs 1.5Mb)
- Programming environment
  - Legacy code support better on a Xeon Phi (often only in theory)
  - Toolchain support roughly equivalent
  - Programming roughly equivalent (Pthreads/Cilk/OpenMP/TBB vs CUDA/OpenACC)

Conclusions

- Accelerators / coprocessors are becoming a standard commodity in HPC
- GPUs, Xeon Phis and CPUs are very different beasts
  - Different strengths and weaknesses for different use cases
  - Need to look at both producticity and performance
- Directive-based languages look very promising
  - A common standard would be great, however
- The landscape of accelerators is constantly evolving
  - Improvements to the technologies
  - Unexpected business decisions (HPC is not the core business of the vendors)
  - Constant technology tracking needed
- CSC is staying on top of these developments and providing resources
  - Need to engage the users and promote these resources..