

Multiprocessor architectures

Vocabulary

Processor

- May have multiple cores in one integrated circuit

Core

- Central Processing Unit(CPU)
- share e.g. Bus, Memory Controller, Cache(L3 most common)

Distributed Shared Memory

- One shared address space

Multi-Processor

- Systems consisting of at least two Processors

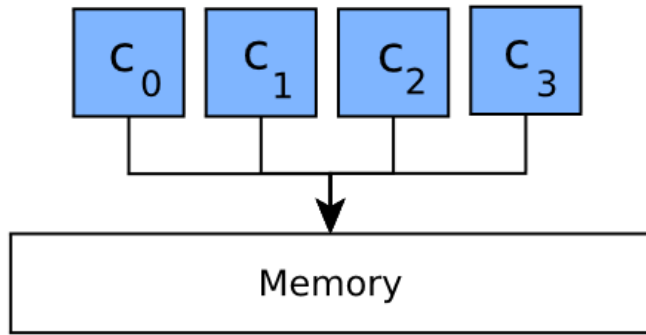
NUMA-Node

- Memory and all processors which are directly connected (can also include other IO-devices)

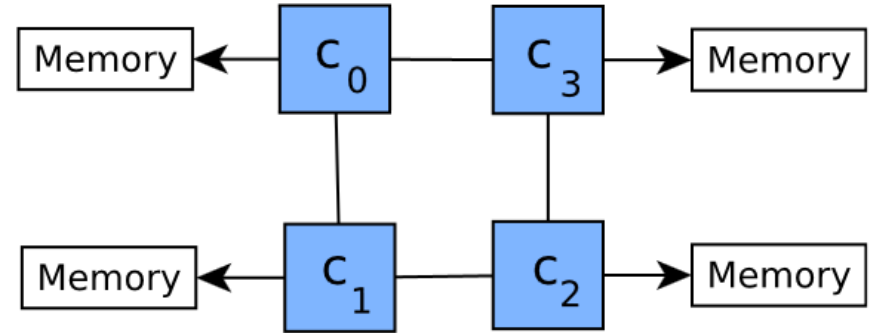
High-Speed Interconnect

- Used especially for connecting processors to each other

NUMA vs UMA



UMA



NUMA

Figure 8: The UMA (*aka* SMP) and NUMA memory architectures.

A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures (http://www.global-sci.com/openaccess/v15_285.pdf)

What is NUMA?

- Non-Uniform Memory Access
- Distributed shared memory
- Local and remote memory
- Increased latency, decreased bandwidth
- Can also affect other devices

Why NUMA?

- In UMA Systems: all processors share a memory controller and connection to memory
- In NUMA Systems: multiple memory controllers and connections can share the load
 - NUMA Systems scale better in settings with many memory accesses by different processors

Classification of NUMA-Systems

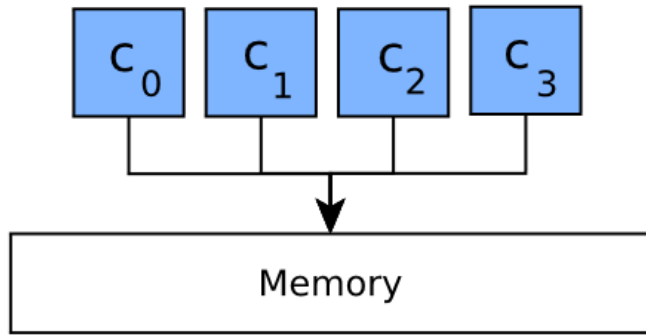
Distance metrics:

- NUMA-Ratio: describes latency
- Number of hops

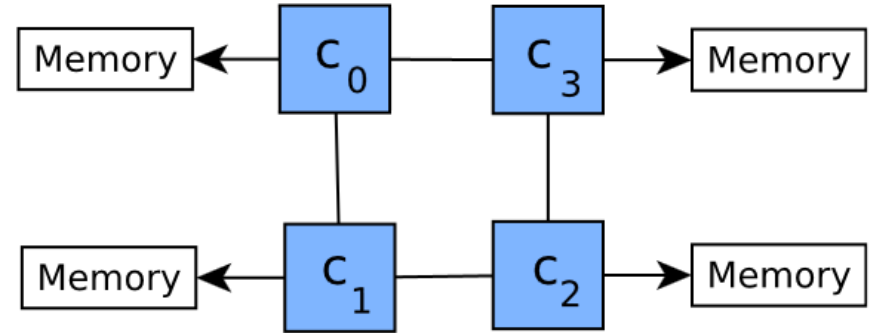
2:1 NUMA-Ratio: it takes twice as long to access remote memory

1:1 = UMA

NUMA vs UMA



UMA



NUMA

Figure 8: The UMA (*aka* SMP) and NUMA memory architectures.

A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures (http://www.global-sci.com/openaccess/v15_285.pdf)

Intel Xeon (E7 88xx)

Frequency	Cores/ Processor	Threads/ Core	#MCs /Processor
2.2GHz<=x<=3.4GHz	6,10,12,15	2	1 or 2

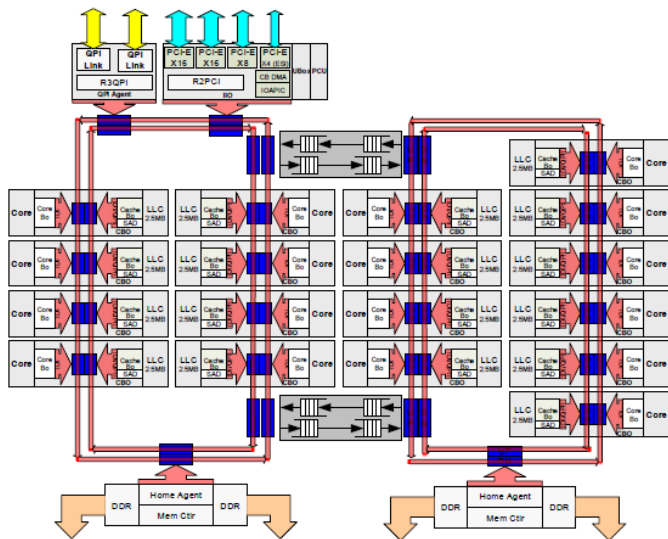
- NUMA since Nehalem microarchitecture (2007)
- High-Speed Interconnections to up to 3 other processors

Cluster-On-Die

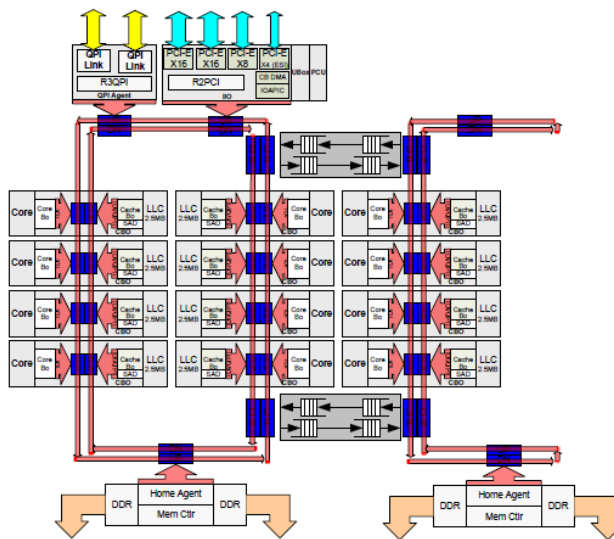
- Mode for Haswell microarchitecture(Intel)
- Enables 2 NUMA-nodes for one processor

Haswell EP Die Configurations

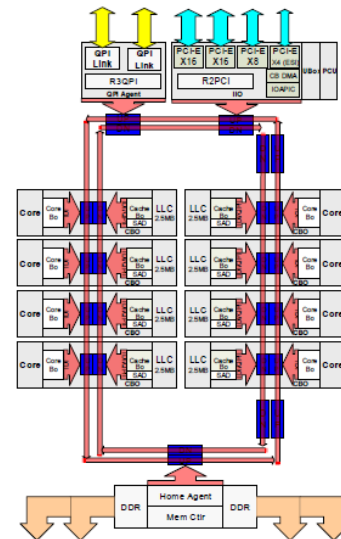
14-18 Core (HCC)



10-12 Core (MCC)



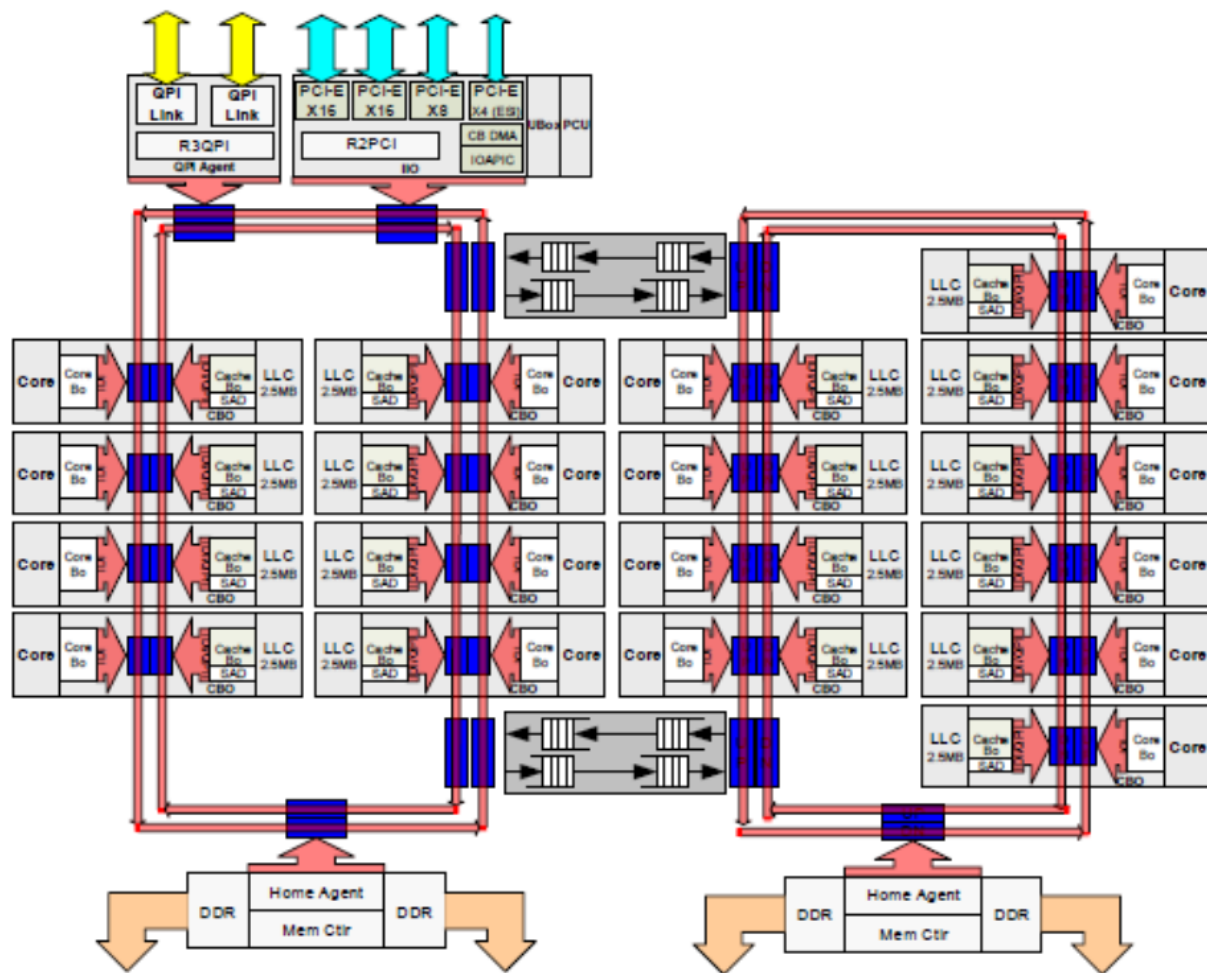
4-8 Core (LCC)



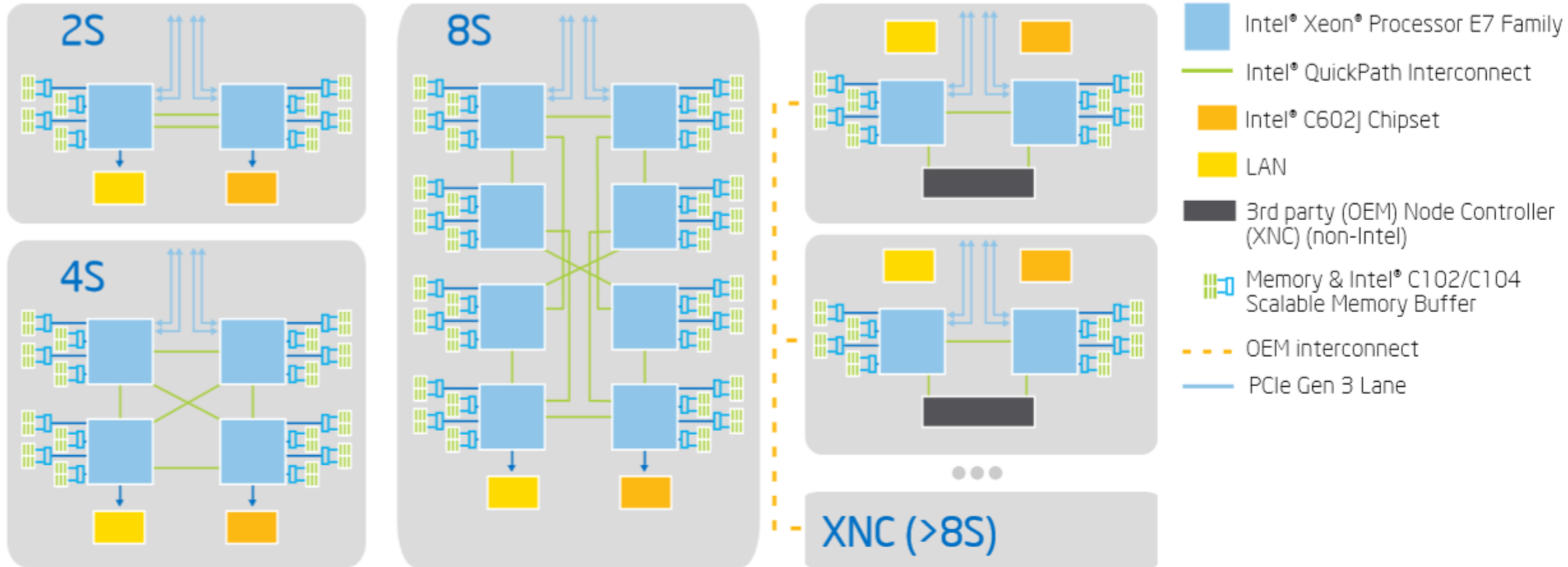
Not representative of actual die-sizes, orientation and layouts – for informational use only.

Chop	Columns	Home Agents	Cores	Power (W)	Transistors (B)	Die Area (mm ²)
HCC	4	2	14-18	110-145	5.69	662
MCC	3	2	6-12	65-160	3.84	492
LCC	2	1	4-8	55-140	2.60	354

14-18 Core (HCC)



Intel Xeon



Intel® Xeon® Processor E7 v2 Family Product Brief

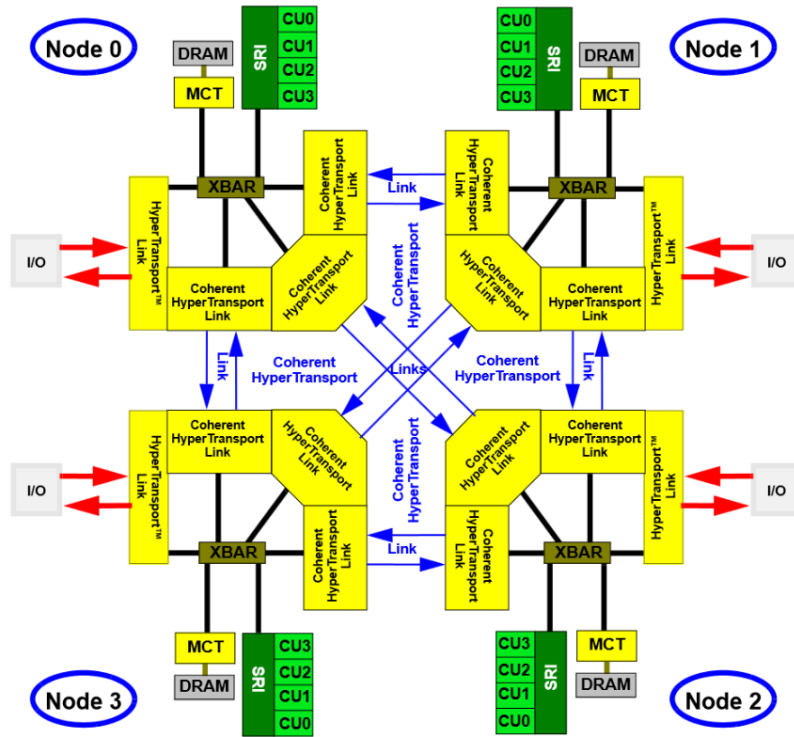
19.11.14 - Kirstin Heidler - NUMA Seminar

AMD Opteron (Bulldozer)

Frequency	Cores/ Processor	Threads/ Core	#MCs/ Processor
1.8GHz<=x<=3.6GHz	4,8,12,16	2	1

- NUMA since 2003
- first system which supported 64bit and 32bit without performance penalties for 32bit-mode
- High-Speed Interconnections to up to 3 other processors

AMD Opteron



Software Optimization Guide for AMD Family 15h Processors (http://support.amd.com/TechDocs/47414_15h_sw_opt_guide.pdf)

Figure 13. Block Diagram of a cNUMA AMD Family 15h 4P Multiprocessor System

Oracle Sparc T5

Frequency	Cores/ Processor	Threads/ Core	#MCs/ Processor
3.6GHz	16	8	4

- Can execute 2 threads per core at same time
- High-Speed Interconnections to up to 4 other processors
- NUMA since SPARC T3(2010)

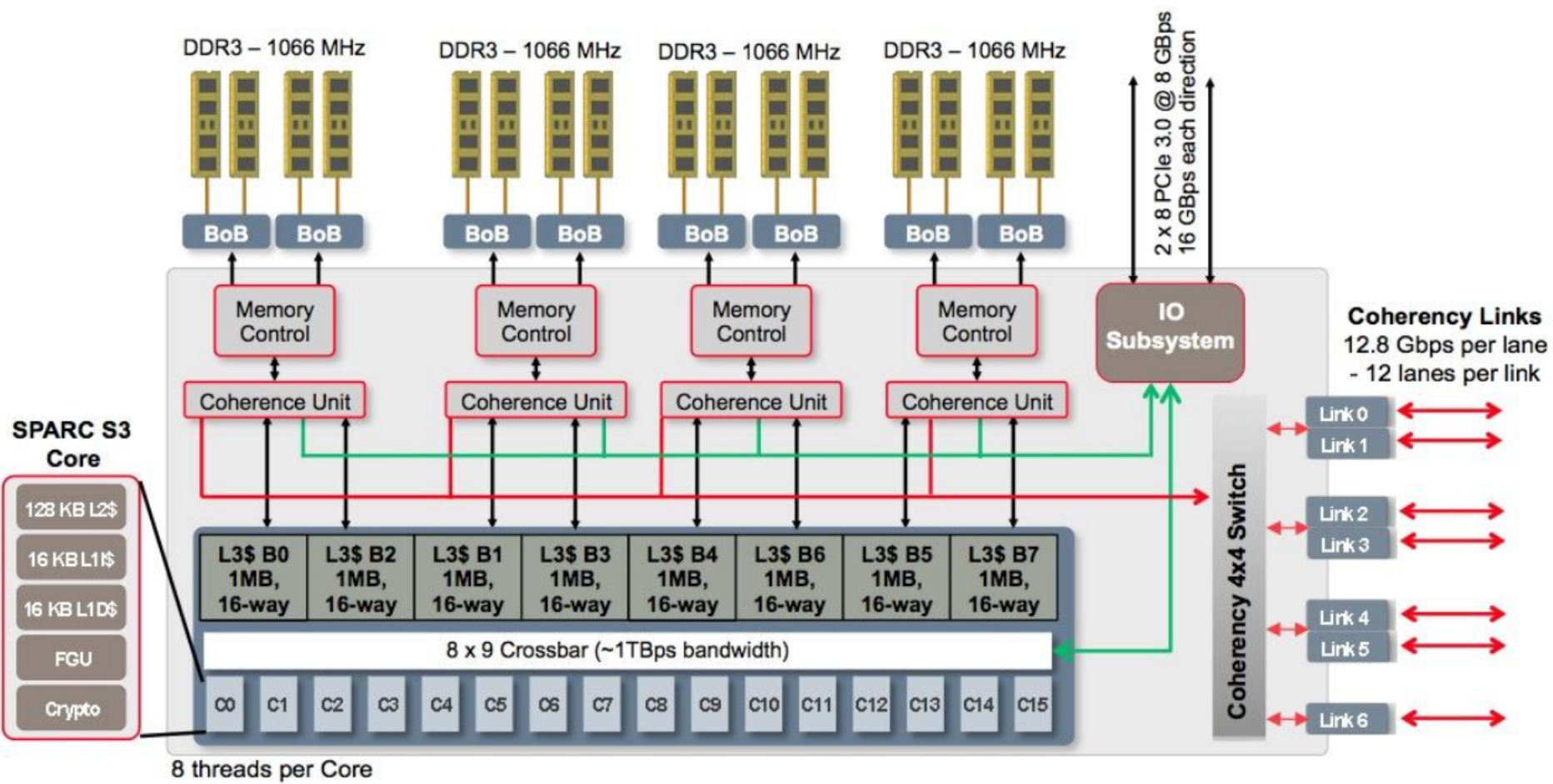


Figure 4. The SPARC T5 processor provides seven coherence links to connect to up to four other processors.

Intel Xeon Phi (5110P)

Frequency	Cores/ Processor	Threads/ Core	#MCs /Processor
1.053GHz	>=50	2	

- 4 threads per core to fully utilize the hardware
- no shared L3 cache
- Distributed Tag Directories(DTDs) for cache coherence
- Access to remote L2 cache almost as slow as off-chip memory access

Intel Xeon Phi

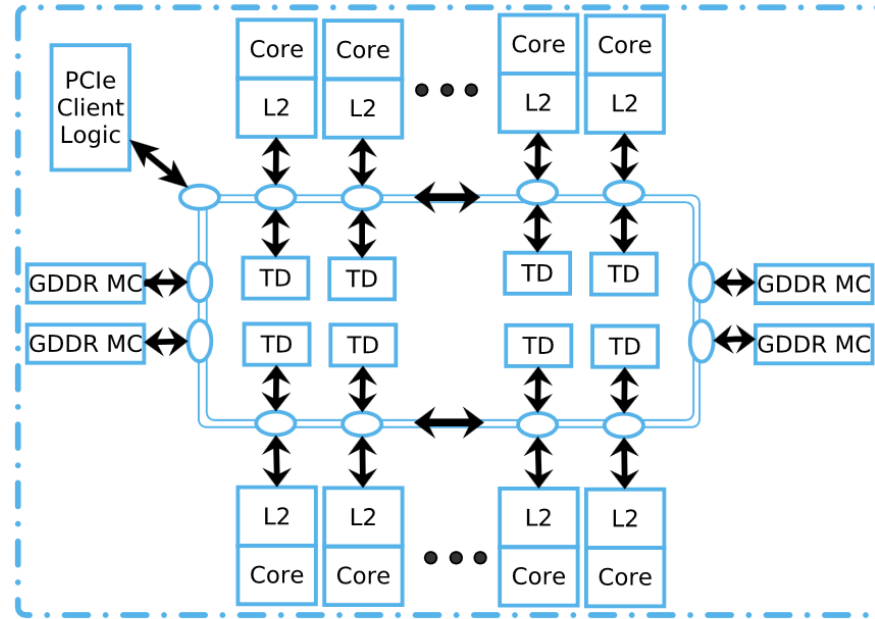


Figure 1: The Intel Xeon Phi Architecture.

On June 17, 2013, the [Tianhe-2](#) supercomputer was announced by TOP500 as the world's fastest. It uses Intel Ivy Bridge Xeon and Xeon Phi processors to achieve 33.86 PetaFLOPS.

http://en.wikipedia.org/wiki/Xeon_Phi

Future SOC Lab(Excerpt)

1000 Core Cluster:

25 x 4 x Intel Xeon E7- 4870(Sandy Bridge microarchitecture)

Coprocessors:

- 2 x NVIDIA Tesla K20X
- 2 x Intel Xeon Phi 5110p

How to make use of NUMA Systems

- OpenMP and MPI
- Do-It-Yourself(pthreads + assembler)
- NUMA-aware OS
- maybe: Actor Systems (Scala, Erlang)

Sources

Intel Xeon

<http://www.intel.com/content/www/us/en/processors/xeon/xeon-e7-v2-family-brief.html>

http://www.intel.com/pressroom/archive/reference/whitepaper_QuickPath.pdf

<http://www.intel.de/content/www/de/de/processors/xeon/xeon->

AMD Opteron

http://www.amd.com/Documents/6000_Series_product_brief.pdf

<http://www.amd.com/en-us/products/server/opteron/6000/6300#>

http://support.amd.com/TechDocs/47414_15h_sw_opt_guide.pdf

SPARC T5

http://en.wikipedia.org/wiki/SPARC_T5

[http://www.oracle.com/technetwork/server-
storage/sun-sparc-](http://www.oracle.com/technetwork/server-storage/sun-sparc-
enterprise/documentation/o13-066-sparc-m6-
32-architecture-2016053.pdf)

[enterprise/documentation/o13-066-sparc-m6-
32-architecture-2016053.pdf](http://www.oracle.com/technetwork/server-storage/sun-sparc-
enterprise/documentation/o13-066-sparc-m6-
32-architecture-2016053.pdf)

[http://www.oracle.com/technetwork/server-
storage/sun-sparc-](http://www.oracle.com/technetwork/server-storage/sun-sparc-
enterprise/documentation/o13-024-sparc-t5-
191111-14-Kilsmith-Endler-NUVA-Seminar)

[enterprise/documentation/o13-024-sparc-t5-](http://www.oracle.com/technetwork/server-storage/sun-sparc-
enterprise/documentation/o13-024-sparc-t5-
191111-14-Kilsmith-Endler-NUVA-Seminar)

Intel Xeon Phi

<http://www.pds.ewi.tudelft.nl/fileadmin/pds/homepages/fang/papers/icpe2k14a22.pdf>

Test-Driving Intel Xeon Phi

<https://www-ssl.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-datasheet.html>

<http://www.hpcociety.org/Resources/Documents/6-9NOV2011-AMD-Best%20practices%20for%20programming%20with%20openMP%20on%20NUMA%20systems.pdf>

<http://heteropar2014.bordeaux.inria.fr/slides/slides-8.pdf> Scalable SIFT for NUMA
with Actors

http://sites.amd.com/us/Documents/PID52355A_NUMA_Performance_Considerations_in_VMware_vSphere_FINAL.pdf

http://www.global-sci.com/openaccess/v15_285.pdf Section 6
(Architectures)

A Survey on Parallel Computing and its Applications
in Data-Parallel Problems Using GPU Architectures

Multi-Core to Many-Core

<http://www.altera.com/technology/system-design/articles/2012/multicore-many-core.html>

<http://dl.acm.org/citation.cfm?id=2337210>

Can traditional programming bridge the Ninja performance gap for parallel computing applications?

<http://dl.acm.org/citation.cfm?id=2259046>

Matching memory access patterns and data placement for NUMA systems

<http://mspiegel.github.io/publications/ijhpca11.pdf>

OpenMP Task Scheduling Strategies for Multicore NUMA Systems

<http://dl.acm.org/citation.cfm?id=1993481>

**Memory management in NUMA multicore systems:
trapped between cache contention and interconnect
overhead**

<http://dl.acm.org/citation.cfm?id=2188342>

Automatic NUMA characterization using Cbench