Scientific Computing in practice

Kickstart 2015

Ivan Degtyarenko, Janne Blomqvist, Mikko Hakala, Simo Tuomisto

School of Science, Aalto University

June 1, 2015

Day #1. What is it?

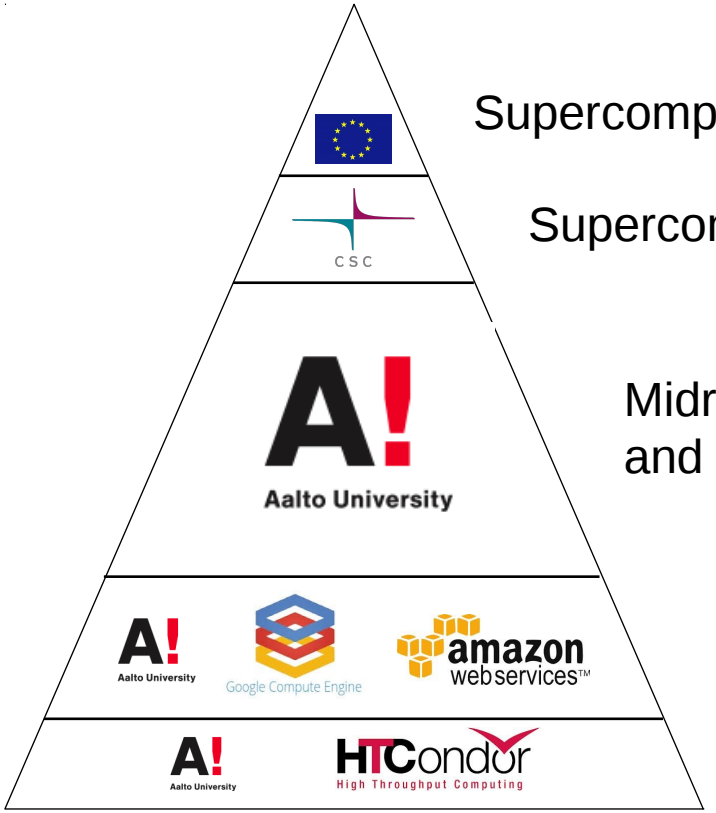
HPC crash course

- Part of Scientific Computing in Practice: kickstart 2015 course *(http://science-it.aalto.fi/scip/)*
- Mon 1.6 at 09:00 till about 17:00, an 45 min lunch break in between and bunch of 10 min breaks in addition
- Overview of the scientific computing concepts and terminology
- Intro into computational clusters architecture
- CSC and Grid resources introduction
- Condor introduction
- Triton hands-on tutorial: common, jobs with SLURM, IOs with Lustre, environment, Matlab

Course purpose: kick-off for Aalto researchers to get started with the available computational resources at Aalto and around

Feel free to interrupt and ask

Computing resources pyramid



Supercomputing at EU: Prace facilities

Supercomputing at CSC: sisu, taito

Midrange Computing: Aalto Triton-cluster and Grid computing by Science-IT

Cloud computing, departments' local servers

on workstations: Condor, local runs

Why supercomputing?

Primarily increase the problem scale, secondarily decrease the job runtime

Make possible serving large number of user applications at one place, homogeneous environment

Other reasons: physical constraints preventing CPU frequency scaling: principal limitation of *serial* computing

Supercomputing is often the only way to achieve specific computational goals at a given time

Computing Thesaurus

- Parallel Computing: means using more than one processing unit to solve a computational problem
- Embarrassingly Parallel: solving many similar, but independent, tasks; e.g. parameter sweeps. Often associated with Grid or Condor computing
- Distributed (Grid, Condor) Computing: solving a task by simultaneous use of multiple isolated, often physically distributed heterogeneous computer systems connected with a specific middleware like Condor or ARC
- High Performance Computing (HPC) or Supercomputing: use of the fastest and the biggest machines with fast interconnects and large storage capabilities to solve large computational problems
- GPU computing (or other accelerators): use of GPU cards (or other accelerators) to solve computational problems
- Cluster: machine for computing comprised of two or more nodes linked by interconnect
- Compute node: computational unit, has CPU, memory, accelerators
- Storage: at HPC a fast and reliable standalone device connected through cluster interconnect that stores data
- Interconnect: communication links between the compute nodes (Ethernet, Infiniband, etc.)

flop/s

FLOPS (or flops or flop/s) – FLoating point OPerationS, a measure of a computer's performance

… a product of number of cores, cycles per second each core runs at, and number of double-precision (64 bit) FLOPS each core performs per cycle (depending on CPU could be a factor of four/eight/sixteen/thirty two)

Theoretical peak-performance of Triton (as of spring 2015) ~ 80 Tflop/s. LINPACK on full cluster never performed.

Top 10 of top500 are petaflop computers, i.e. they perform 10^{15} floating point operations per second; your calculator is >10 flop/s, your quad core PC about 25 Gflop/s, single 12 cores Triton compute node about 125 Gflop/s, Nvidia Tesla K20 GPU computing processors around 1.2 Tflop/s

Top5 of top500.org

November 2014

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945

Real HPC example: K computer

RIKEN AICS K computer

- 64-bit Sparc VIIIfx, 2.0 GHz, 8-way SMP
- 11.28 PFlop/s peak, 10.51 PFlop/s Linpack
- 1.41 PByte memory; 705,024 CPU cores
- 864 racks; 88,128 nodes
- 6D interconnect (Tofu)
- Water cooling

RIKEN AICS K computer (cont.)

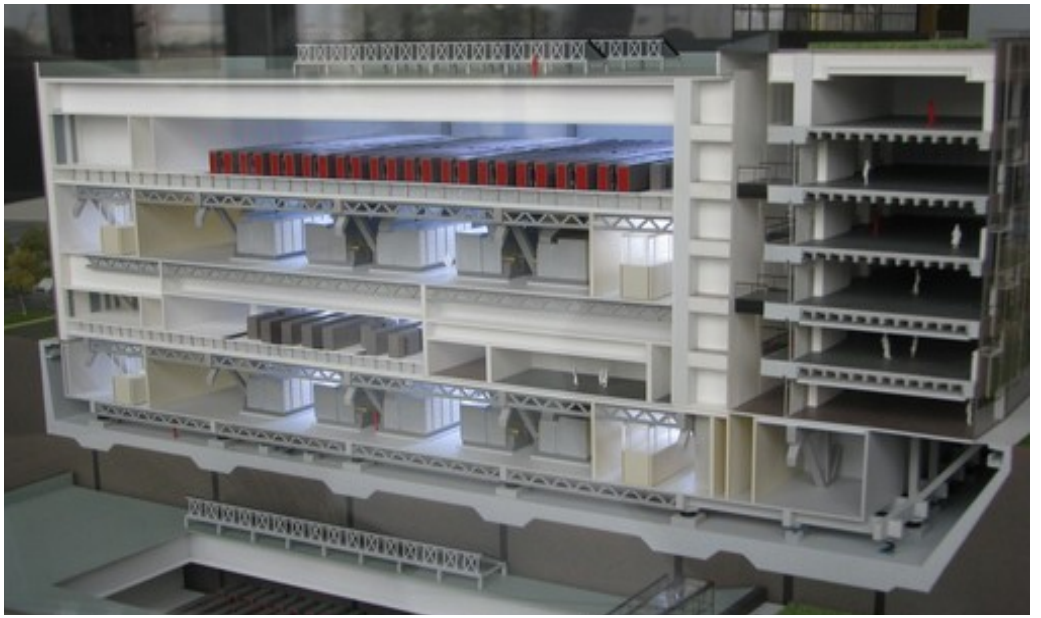
The only computer with its own train station

4th floor: K computer

3rd floor: Computer cooling

2nd floor: Disks

1st floor: Disk cooling



K computer building Earth quake security

horizontal moves damper

Vertical moves damper

“wiggle” moves damper

flexible pipes

Local examples

at Aalto: Triton – HP cluster: SL230s, SL390s and BL465s compute nodes interconnected with InfiniBand

https://wiki.aalto.fi/display/Triton/Triton+User+Guide

at CSC: Sisu – Cray XC30 and Taito – HP cluster. Both are on top500, #37 and #109 / #425

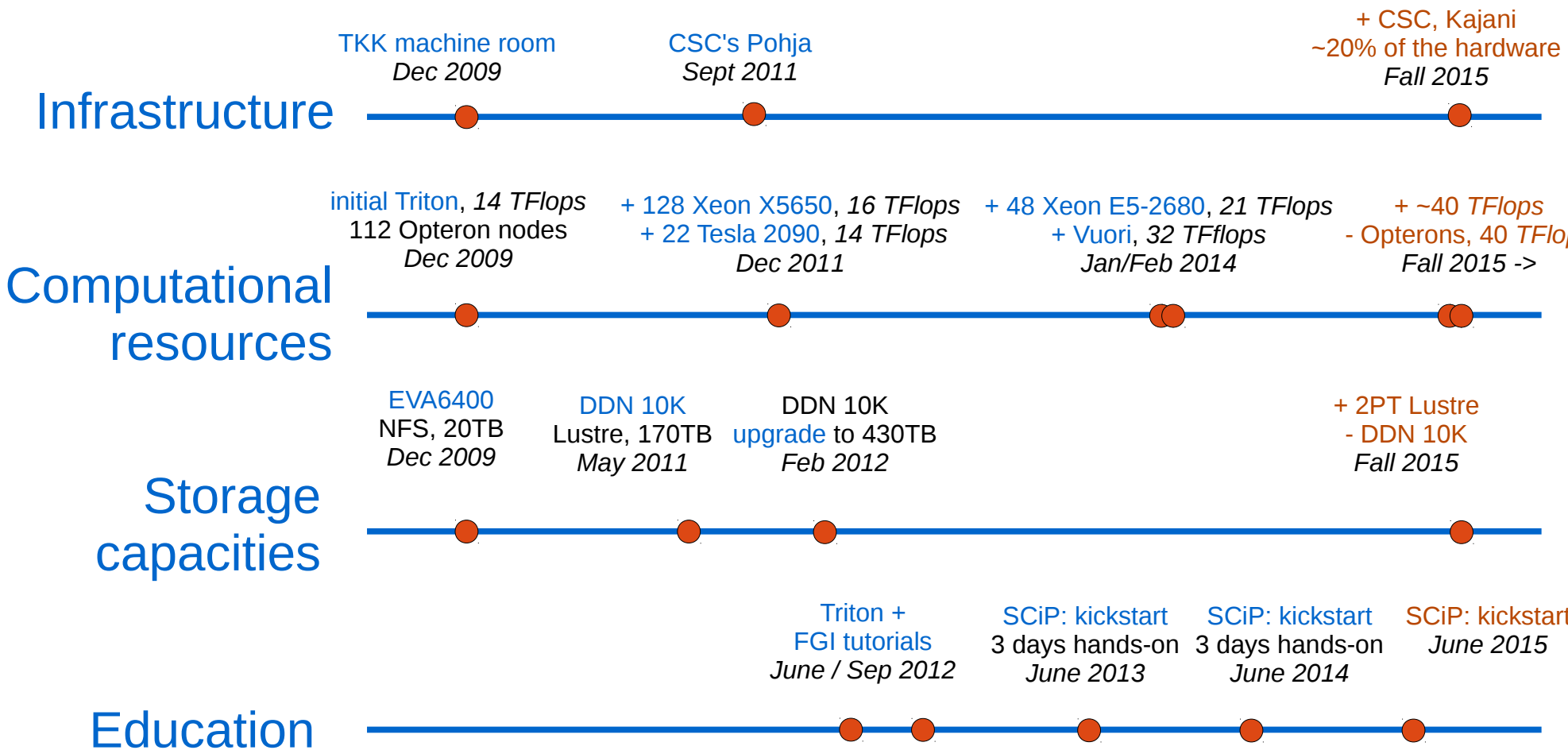
http://www.csc.fi/english/pages/hpc/resources

Scientific Computing@Aalto: Science IT

http://science-it.aalto.fi

- Computational Studies are within top-5 highest priorities at Aalto
- midrange Scientific Computing resources for Aalto researchers: Triton-cluster
- wide collaboration between Finnish Universities and CSC, part of European grid infrastructure
- 2005 collaboration started under M-grid; 2010 -- FGI, a wider consortium; 2014 – FGCI, Finnish Grid and Cloud Infrastructure *(in progress)*
- joint procurement and administration collaboration, focus on serving (special) needs of Aalto's computational science, i.e. raw CPU power, Matlab, big data sets, large memory servers, GPU computing, Hadoop

Science IT timeline



Science IT management

Management board

- Prof. Martti Puska (SCI/PHYS) , Prof. Jari Saramäki (SCI/CS), Prof. Keijo Heljanko (SCI/CS, chair), Prof. Mikko Kurimo (ELEC/SPA) and Doc. Maarit Mantere (SCI/CS)

Stakeholder model

- All costs are distributed to schools (departments) based on the agreed Science-IT share
- Key departments (NBE, PHYS, CS) contribute 50% to administrative personnel responsible for day-to-day tasks. Local support person named at each participating unit.
- 20% of resources for Grid => Freely available for Finnish research community via grid-interface
- Small share 5% for users outside stakeholder model

Science IT support team

The core team

- Mikko Hakala, D. Sc. (Tech) (CS+NBE/SCI)
- Janne Blomqvist, D. Sc. (Tech) (PHYS+NBE/SCI)
- Simo Tuomisto, (CS/SCI)
- Ivan Degtyarenko, D. Sc. (Tech) (PHYS/SCI)

- responsible for ”daily cluster maintenance”
- wiki-area at http://wiki.aalto.fi, look for Triton User Guide, support tracker at http://tracker.triton.aalto.fi
  - easy entry point for new users

Support team members

- Jussi Hynninen (ELEC)
- named by Departments to be local Science-IT support team members
  - provide department user support, being close to researchers
- being contact person between schools (departments) and Science-IT support

Triton: overview

- 550 compute nodes with 12 or 20 CPU cores
  - mix of Xeon IvyBridge, Xeon Westmere and Opteron nodes
- InfiniBand connections
- GPU computing (Tesla M2090/M2070/M2050 cards)
- Fat (large memory) nodes with 1TB of RAM
- High-performance Lustre filesystem, capacity 430 TB
- Operating system: Scientific Linux 6

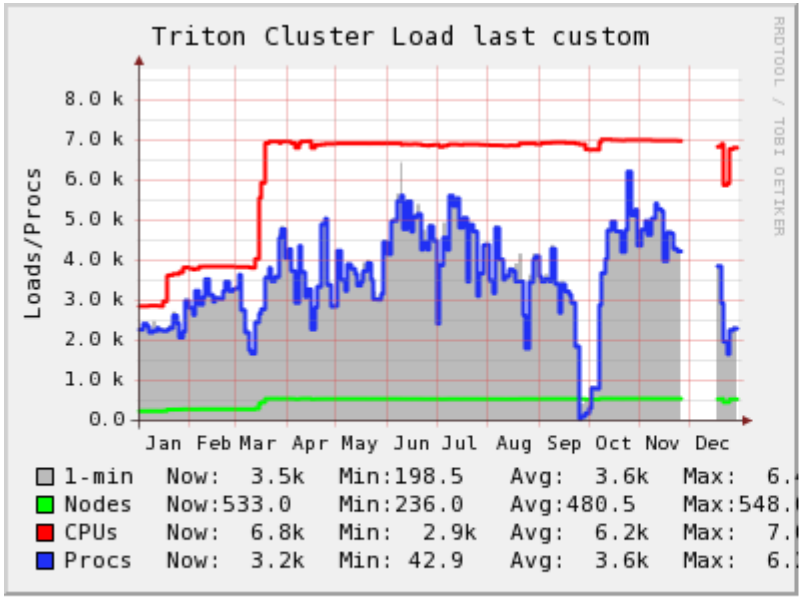
Triton tech specs in details

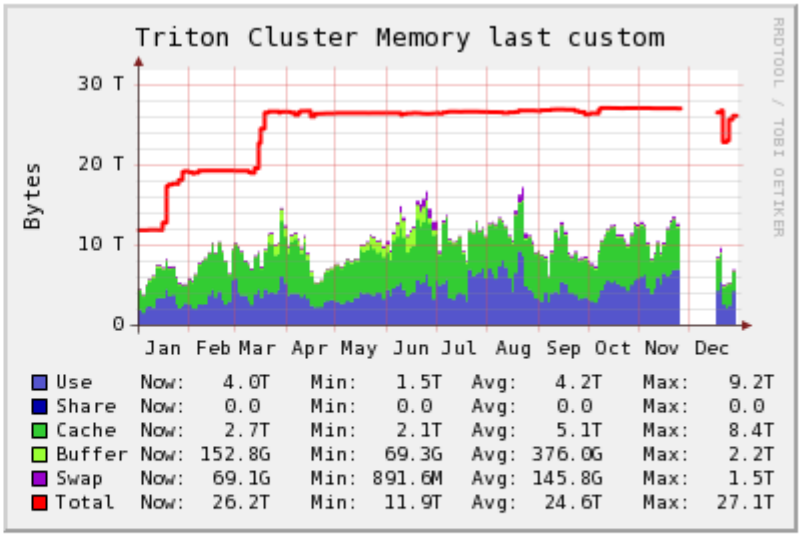
- **HP SL390s G7 nodes**
  - 2x Xeon X5650 @ 2.67GHz (6core) / 48 GB RAM
  - 2x local SATA disks (7.2k) -> 800GB of local storage
  - 2x Tesla M20[5|7|9]0 GPU (Fermi, 3/6 GB mem) on 19 servers / 24 GB RAM
- **HP BL465c G6 nodes**
  - 2x AMD Opteron 2435 @ 2.6GHz (6core)
  - 200GB local storage
  - 16/32/64 GB RAM
- **HP DL580 G7 nodes**
  - 4x X7542 @ 2.67 GHz (6core)
  - 1.3TB local storage
  - 1024 GB RAM
- **HP SL230s G8 nodes**
  - 2x E5 2680 v2 @ 2.8 GHz (10 cores)
  - 1.8TB local storage
  - 64/256 GB RAM

Triton tech specs in details (cont.)

- **Lustre storage system**
  - 600 TB raw capacity
  - 20 000 IOPS, 3-4 GB/s Stream (read/write)
  - DDN SFA10k system
- **Interconnect**
  - SL230s: 4xFDR Infiniband (56 Gbps)
  - SL390s, DL580: 4xQDR Infiniband (40 Gbps)
  - BL465: 4xDDR Infiband (20 Gbps)
  - 10 GB Ethernet in/out
- **Operating system & Software**
  - Scientific Linux 6.x (RHEL based)
  - CVMFS (apps over the grid sites)
  - SLURM as a batch system
  - MDCS – Matlab distributed computing server with 128 seats
  - Modules + lots of software

Triton stats 2014





- **Users**
  - 606 user accounts of them 207 active
  - 33 departments
  - 6 schools
- **Jobs**
  - Over 10'000'000 jobs calculated
  - 6,4 M cpu-hours computed
  - Utilization ~75%

Run times (cpuh)

Memory (Gb)	0-0,5	0,5-1	0,5-4	4-12	12-24	24-48	48+		
0-0,1	1866524	99714	515324	52077	63056	17636	15061	2629392	47 %
0,1-0,5	879414	283087	218123	35138	14443	5500	5506	1441211	26 %
0,5-1	246050	112533	88228	23010	6855	2364	6451	485491	9 %
1-2	145293	50007	105176	34695	9256	5787	7816	358030	6 %
2-4	48769	23889	78951	40014	12219	5229	5667	214738	4 %
4-8	146513	73827	45992	23918	3879	1784	3614	299527	5 %
8-16	22023	10519	97049	10000	3497	3491	3966	150545	3 %
16-32	2472	654	2364	920	2668	2046	589	11713	0 %
32-64	4563	395	1093	591	339	359	1935	9275	0 %
64-128	1516	433	880	327	66	47	229	3498	0 %
128-256	0	0	2	16	18	11	40	87	0 %
256+	0	0	1	3	0	2	32	38	0 %
	3363137	655058	1153183	220709	116296	44256	50906	5603545	
	60 %	12 %	21 %	4 %	2 %	1 %	1 %		

10^6
10^5
10^4
10^3
10^2
0-99

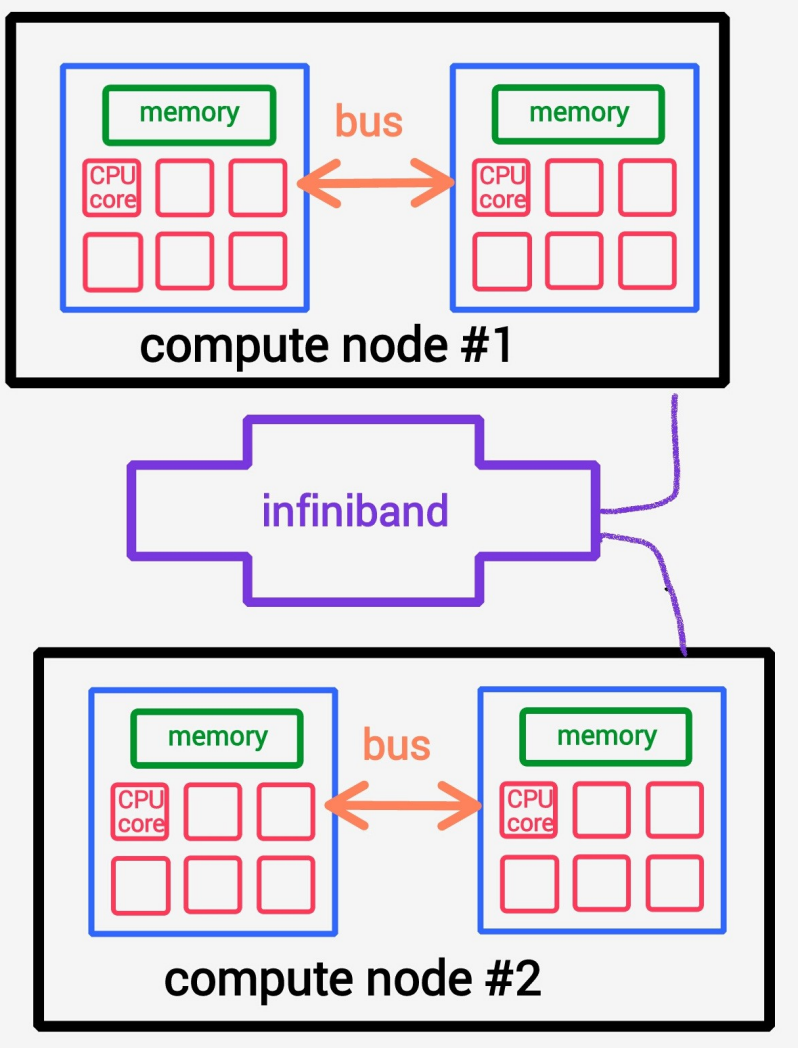
Triton as a parallel computer

In principle, all the computers today are parallel from a hardware perspective (!)

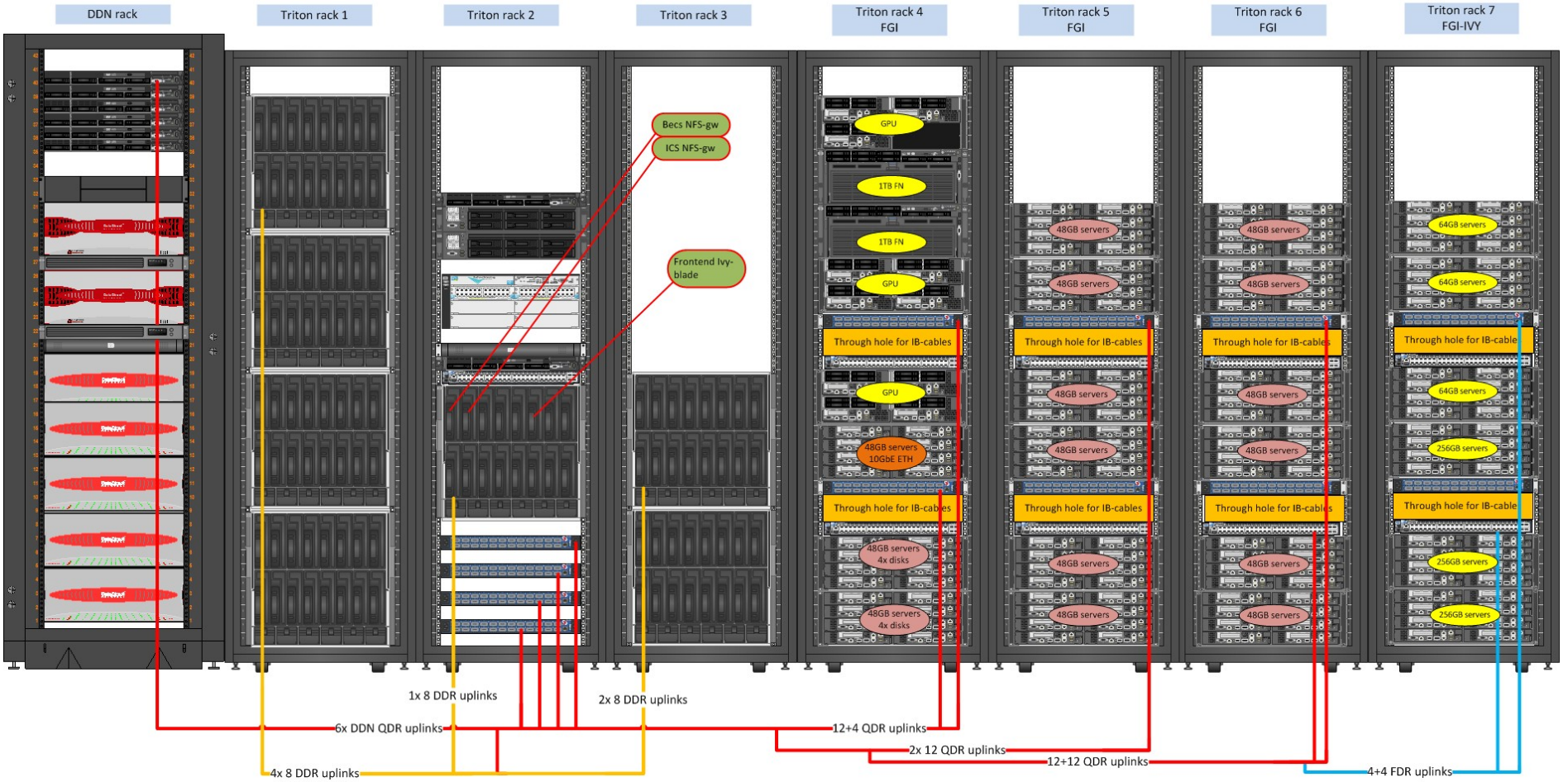
Triton's compute nodes are multi-processor parallel computers in itself. On top of that they are connected through network to make larger computer: a cluster.

Hybrid architecture: it is both and shared memory and distributed memory

- Shared: within one node
- Distributed: over the cluster



Triton's schema (no Vuori part)



Aalto University
School of Science

Distributed + Shared

From end-user perspective:

- Potentially run over whole cluster
- System requires a communication network to connect inter-process memory. The network "fabric" used for data transfer varies widely, though it can be as simple as Ethernet. On Triton we use Infiniband.

From application developer perspective:

- Processors have their own local memory. Memory addresses in one processor do not map to another processor: no global address space across all processors
- Each processor operates independently. Changes to its local memory have no effect on the memory of other processors. The concept of cache coherency does not apply.
- When needed, a processor needs access to data in another processor, it requests it through communication network
- It is the task of the programmer to explicitly define how and when data is communicated: Message Passing Interface – MPI
- MPI implementations: OpenMPI, MVAPICH, etc

From end-user perspective:

- On Triton run within the single node only, i.e. 20 CPU cores at most, though can be run on any modern PC as well
- Running on 12 CPU cores of the same node is way more efficient than running on 12 nodes through interconnect

From developer perspective:

- Way easier to program with OpenMP than with MPI, code remains a serial code as such
- Ability for all processors to access all memory as global address space
- Multiple processors can operate independently but share the same memory resources: changes in a memory location affected by one processor are visible to all other processors
- If not specifically stated otherwise, 'shared memory' means ccNUMA – cache coherent Non-Uniform Memory Access; cache coherent means if one processor updates a location in shared memory, all the other processors know about the update. Cache coherency is accomplished at the hardware level
- With NUMA not all processors have equal access time to all memories; one CPU can directly access memory of another CPU but access across link is slower

Distributed + Shared (cont.)

Advantages:

- Memory is scalable with number of processors. Increase the number of processors and the size of memory increases proportionally
- Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherency.
- Cost effectiveness: can use commodity, off-the-shelf processors and networking.

Disadvantages:

- The programmer is responsible for many of the details associated with data communication between processors.
- It may be difficult to map existing data structures, based on global memory, to this memory organization.
- Interconnect is way slower as comparing to memory bus even in case of NUMA

Advantages:

- Global address space provides a user-friendly programming perspective to memory
- Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs
- Memory latency is a way lower that interconnect

Disadvantages:

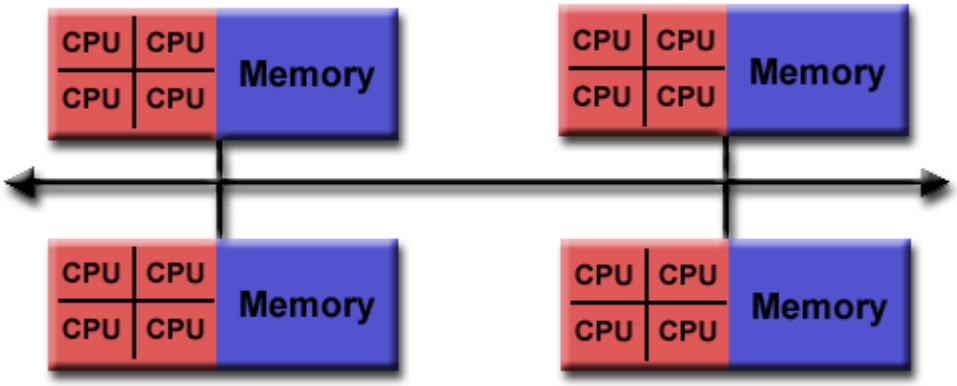
- Primary disadvantage is the lack of scalability between memory and CPUs. Adding more CPUs can geometrically increases traffic on the shared memory-CPU path, and for cache coherent systems, geometrically increase traffic associated with cache/memory management.
- Programmer responsibility for synchronization constructs that ensure "correct" access of global memory.
- Expense: it becomes increasingly difficult and expensive to design and produce shared memory machines with ever increasing numbers of processors

Hybrid Distributed-Shared Memory

Today's HPC installations employ both shared and distributed memory architectures

- The shared memory component is usually a cache coherent NUMA machine. Processors on a given SMP can address that machine's memory as global, though it will take longer to access some regions of memory than others.
- The distributed memory component is the networking of multiple cc-NUMA machines. A particular compute node knows only about their own memory - not the memory on another nodes. Therefore, network communications are required to move data from one node to another.

Current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.



Interconnect

- Interconnect: link in between the compute nodes
- Distributed memory model, main characteristics latency and bandwidth
- Infiniband fabrics on Triton: fast & expensive, very high throughput and very low latency
  - for MPI, Lustre [aka $WRKDIR] on Triton
  - Xeon Westmere nodes: SL390, DL580: 4xQDR Infiniband (40GT/s)
  - Xeon IvyBridge nodes: SL230: 4xFDR Infiniband (56GT/s)
  - Opteron nodes: BL465: 4xDDR Infiband (20GT/s)
  - 4x spine switches are QDR: Voltaire 4036
  - Mostly fat-tree configuration with 2:1 subscription, ivy[01-48] and cn[225-488] differs
    - See full Inifiniband map at https://wiki.aalto.fi/display/Triton/Cluster+overview#Clusteroverview-Networking
- Ethernet: cheap & slow
  - 1G internal net for SSH, NFS [aka /home] on Triton
  - 10G uplink towards Aalto net

Accelerating: GPU computing

GPGPU stands for *General-Purpose computation on Graphics Processing Units*.

- Graphics Processing Units (GPUs) are massively parallel many-core processors with fast memories
- Triton has 22x Tesla M2090 cards on gpu[001-011] nodes, 6x Tesla M2070 on gpu[017-019] and 10x Tesla M2050 on gpu[012-016]
- Card name as a feature: --constraint=tesla2090
- Can be used by some apps out-of-the-box

More accelerators

- Tesla K40, K80: new family of Nvidia GPUs
- Xeon Phi: Intel's 60 cores co-processor by Intel; PCIe card with standard x86 processors; OpenMP programming with standard Intel development tools
  - as of 2015 seems that Tesla dominates on the co-processors market
- Neither of them available on Triton yet; Tesla K40 card available for testing purpose at PHYS
- Common usage reason – more compute performance; the barrier – complexity of programming, need for much wider adoption on software level

Accelerators: way to use

- existing applications tuned for particular accelerator. Examples for Tesla: Gromacs, MATLAB, LAMMPS, NAMD, Amber, Quantum Espresso, etc.; Intel: Accelyrs, Altair, CD-Adapco, etc.
- libraries CUBLAS, cuFFT, Thrust for GPU; Intel's MKL for Phi
- OpenACC: compiler directives that specify regions of code in C/C++ and Fortran to be offloaded from main CPU; hopefully the standard portable approach for the future for programming accelerators; possibly will be incorporated into OpenMP standard
- native programming with CUDA, OpenMP + Intel dev tools

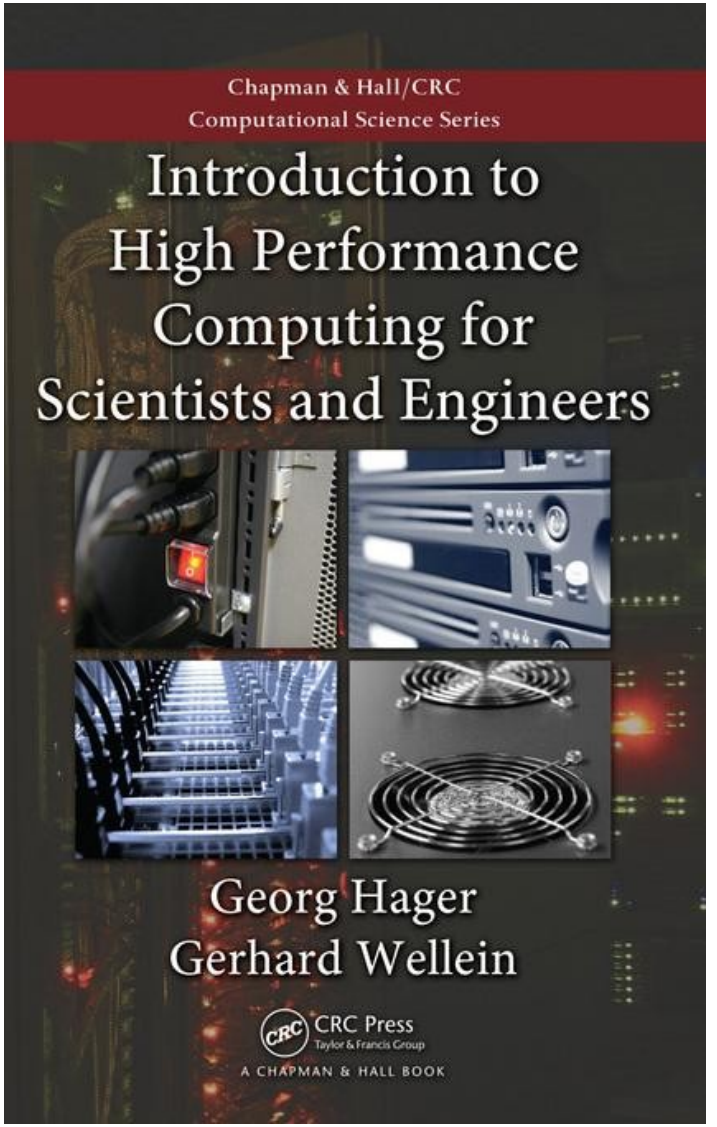
Storage

- “must be” part of any HPC installation, one of the key components
- at HPC often a standalone rack (or number of them) connected to cluster's network (DDN/Infiniband in case of Triton)
- Provides space for work directories, can provide space for any other need
- Reliability, speed, high throughput for hundreds of network clients, massive IOPs, ability to handle large volumes of data quickly
- Parallel filesystems
- In general could be as easy as a pizza-box with a dozen of SATA drives connected to server's, though often a complicated solution with its own hardware controllers, servers and in-house software to manage hundreds of harddrives as a single volume
- Concrete setup depends on needs and finances

Further reading

George Hager, Gerhard Wellein

Introduction to High Performance Computing for Scientists and Engineers



Sources

- *"Crash Course on HPC"* lectures by Bernd Mohr (JSC) http://wwwgsb.fz-juelich.de/SharedDocs/Personen/IAS/JSC/EN/staff/mohr_b
- *"Introduction to Parallel Computing"*, tutorial by Blaise Barney, Lawrence Livermore National Lab: https://computing.llnl.gov/tutorials/parallel_comp/
- *"Introduction to Parallel Computing"* in NIC series by Bernd Mohr: http://www2.fz-juelich.de/nic-series/volume42/mohr.pdf
- Materials to CSC courses "Andvanced MPI", "Parallel I/O", "GPU tutorial", "Grid & Clouds introduction", see http://www.csc.fi/english/csc/courses/archive
- others