Introduction to HPC at ECMWF

125,435,900,000,000,000
8,498,600,000,000,000
4,249,300,000,000,000

Iain Miller

Iain.miller@ecmwf.int

Overview

- Aims of the course
- What is Parallel Computing?
- How to build a supercomputer
- Parallel Programming 101
- Common Terms
- Six Levels of Parallelism

Aims of the course

- At the end of the course you should:
  - Have an understanding of what supercomputing facilities are available at ECMWF
    - How to access
    - How to compile
    - How to run a job
    - Where to store data
    - How to debug a programme
  - Have the fundamentals of parallel programming:
    - Know the difference between Distributed and Shared Memory parallelism
    - Be able to use basic MPI and OpenMP commands
  - Have some knowledge of the challenges ahead

Quiz – no answers required

- MPI
- SIMD
- GPU
- Infiniband
- Send Receive
- Thread
- Rank
- All Reduce
- Halo exchange
- Manycore
- Master

What is Parallel Computing?

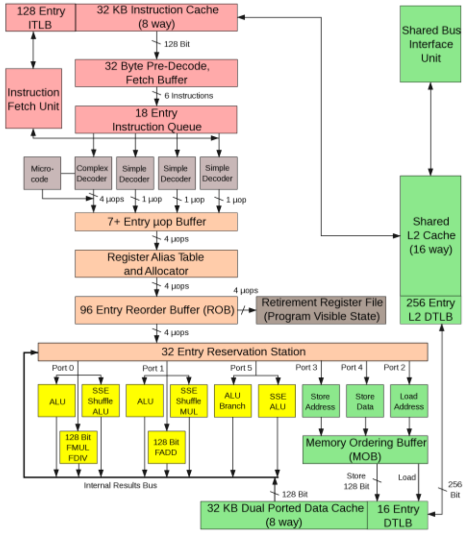
The simultaneous use of more than one processor or computer to solve a problem

Why do we need Parallel Computing?

- Serial computing is too slow
- Need for large amounts of memory not accessible by a single processor

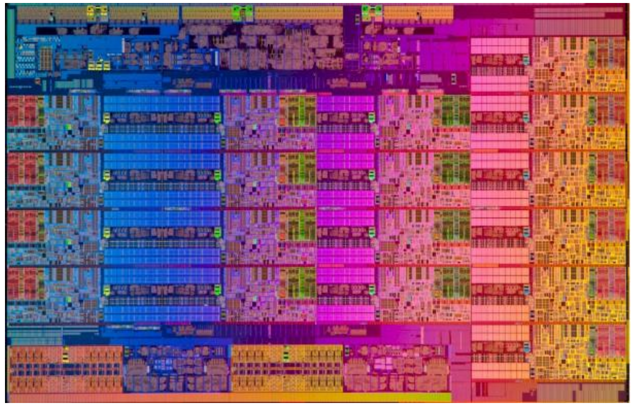
How to build a supercomputer

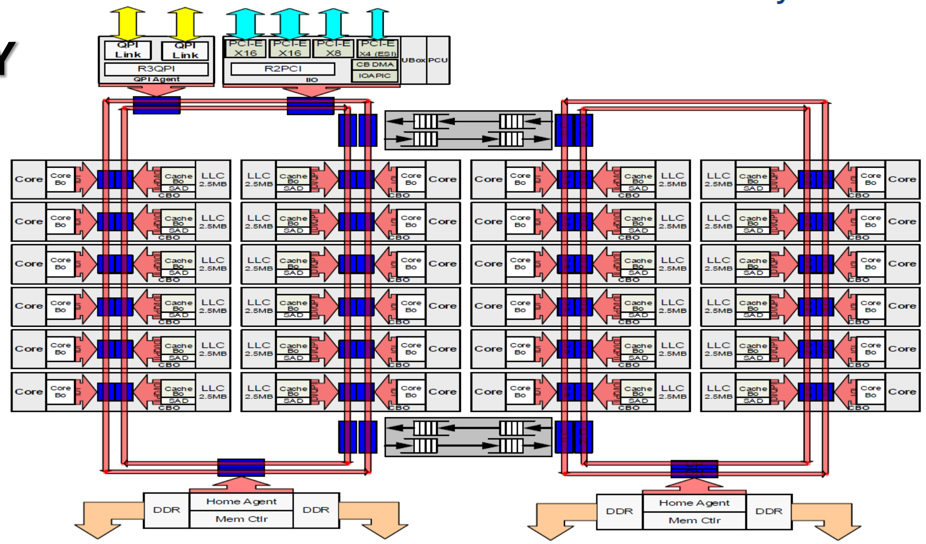
- Lowest component part is the **CORE**
  - Often referred to as a **PROCESSING ELEMENT**
  - Does the actual computation
  - Usually has some independent cache memory to store data and instructions
- A number of cores are laid together on one silicon **CHIP**
  - Current Intel Xeon chips can have up to 24 cores on one chip
  - Cores on a chip may share some cache
- Each chip will be connected to some external **MEMORY**



Intel Core 2 Architecture

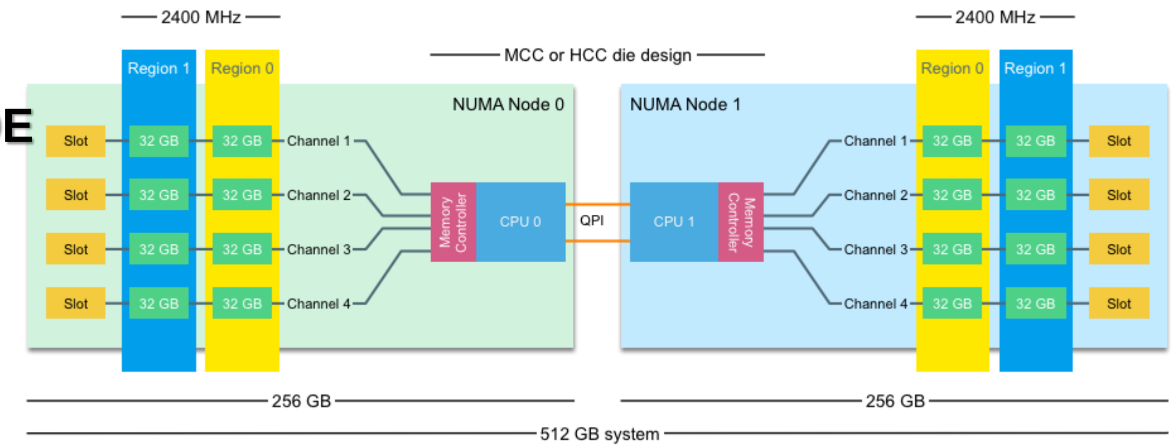
Intel® Xeon® Processor E5 v4 Product Family HCC



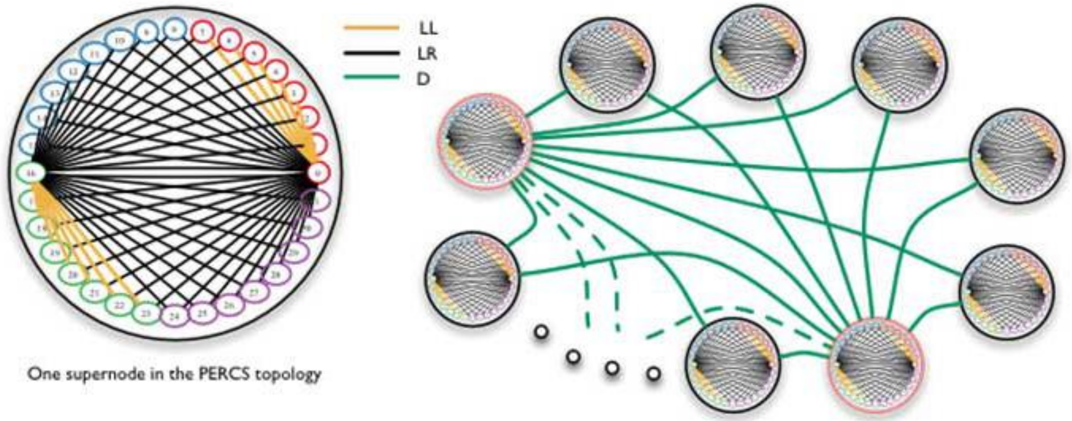


How to build a supercomputer

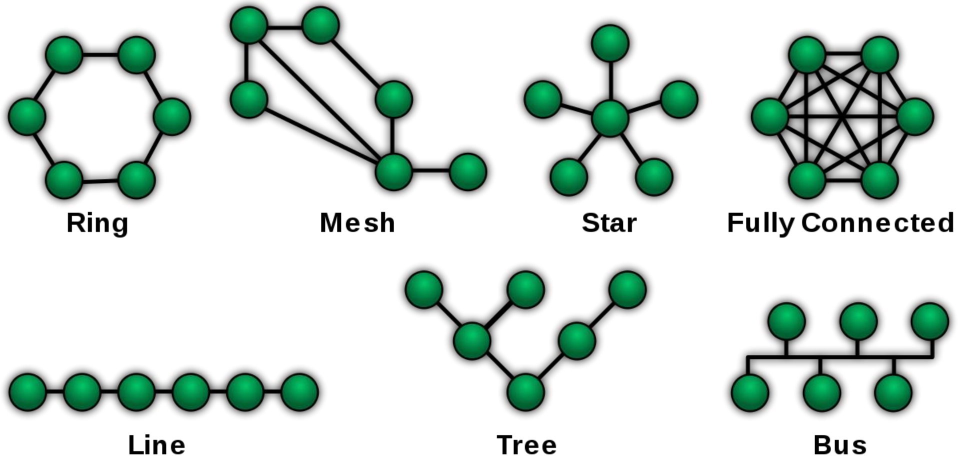
- One or more chips are then put together into a **NODE**
  - Contains memory accessible by all cores on the node
  - Will also have the connections to:
    - Access external storage such as disk and/or tape
    - Network interface controllers



- An **INTERCONNECT** is then used to combine the nodes into a **CLUSTER**
  - Interconnects come in a variety of **PROTOCOLS** and **TOPOLOGIES**
    - Protocols: X GigE, Infiniband, Aries, Tofu, Torrent
    - Topologies: Torus, Mesh, Fat Tree, Hypercube, Ring, Bus, Star, Fully Connected, Dragonfly



One supernode in the PERCS topology

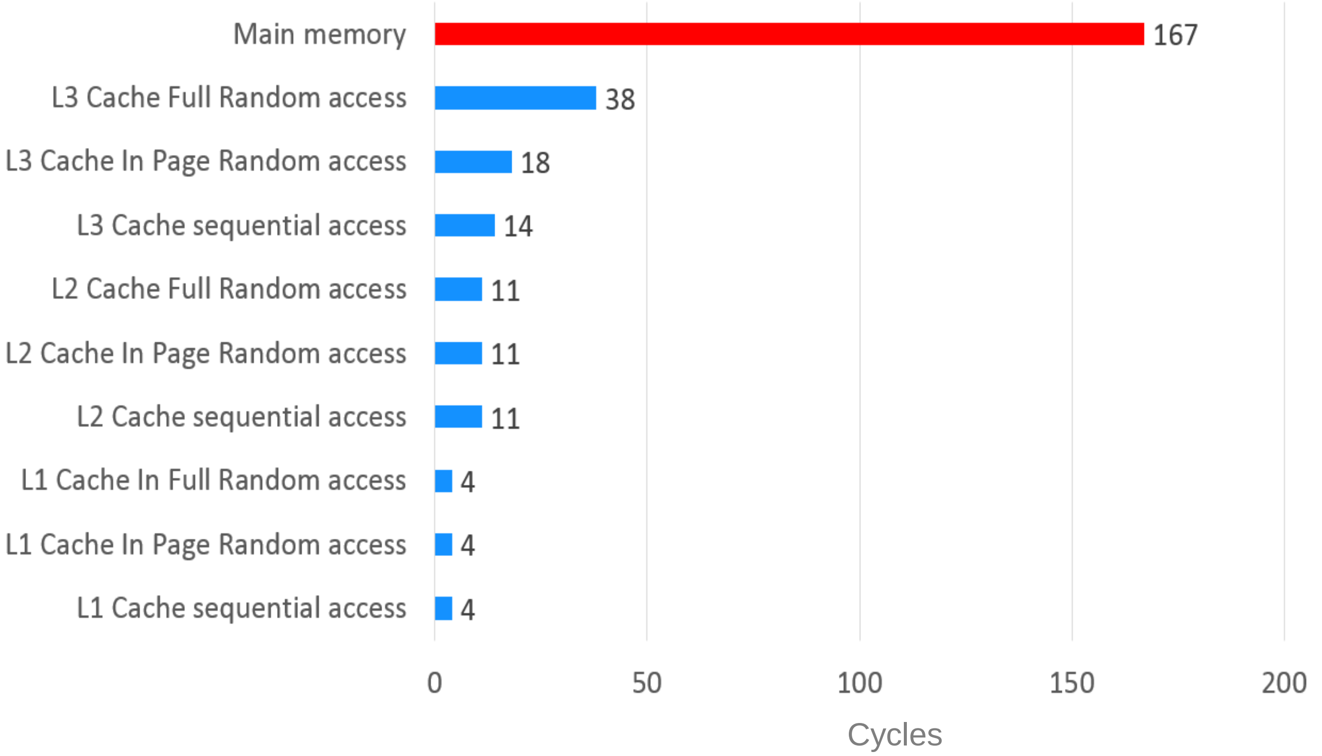


How powerful are Supercomputers

- Base unit for comparing supercomputers is **FLOPS** or **FLOATING POINT OPERATIONS PER SECOND**
  - We are currently in the **PETASCALE** era where the top computers have in excess of 1 **PetaFlops** of computing power
  - The first **EXASCALE** machine is likely to come online in the early 2020s (more of this on Friday)
- Remember the red numbers on the title page:
  - 125.4359 Pflops is the theoretical peak power of the Sunway TaihuLight in China, the current number one machine on the Top 500 list
    - It would take **196 days 4 hours 33 minutes 17s** for every person in the world doing one calculation a second to match what TaihuLight can do in **1s**!
  - 8.4986 Pflops is the theoretical peak figure for both clusters at ECMWF added together
    - 12 days 7 hours 58s
  - 4.2493 Pflops is the figure for CCA
    - 6 days 15 hours 30 minutes 29s

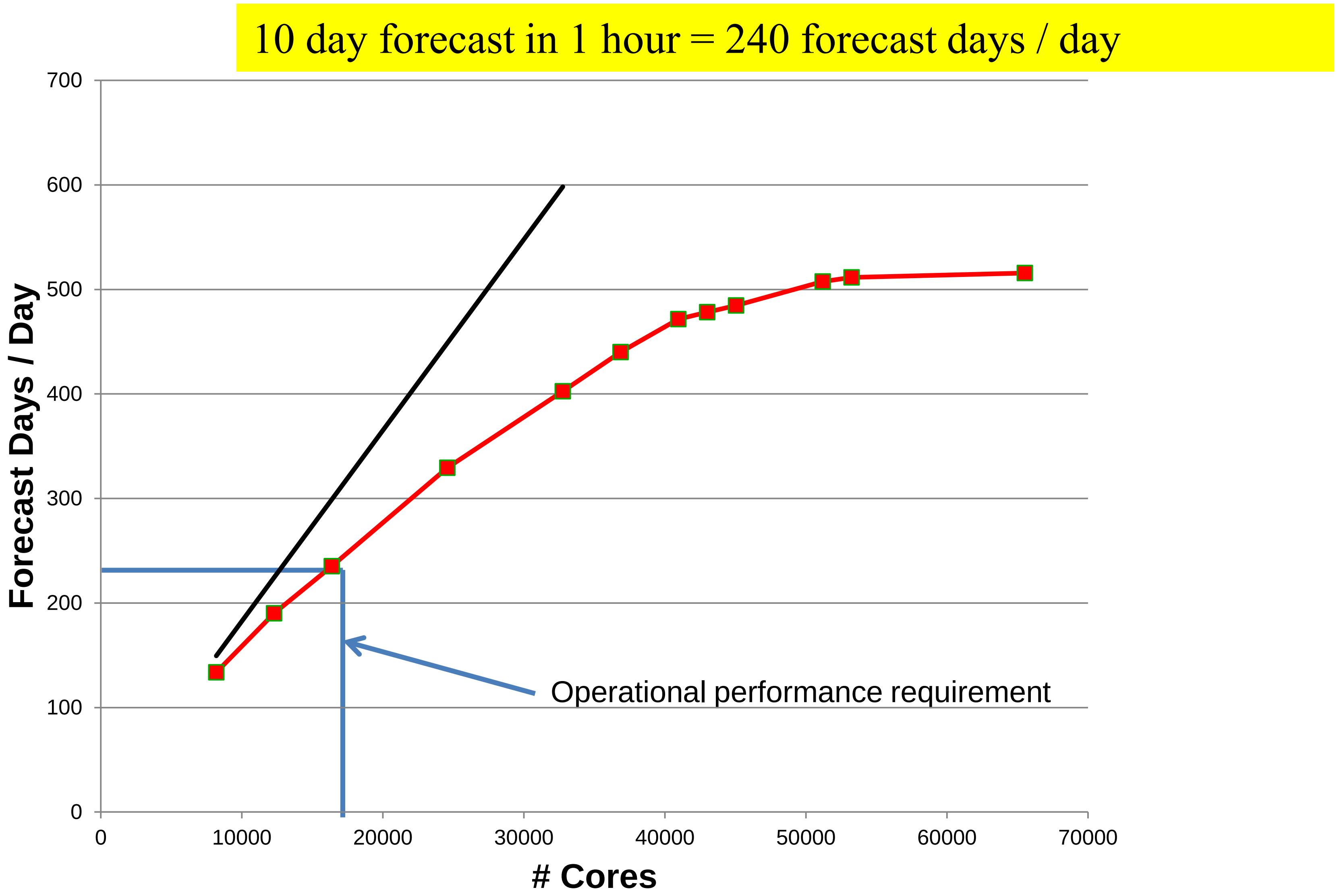
How powerful are Supercomputers

- However, you will never use anywhere near the theoretical peak of a machine
- Moving data around from memory or disk or other nodes into the registers of a core mean that there will always be lots of cycles not being used to calculate

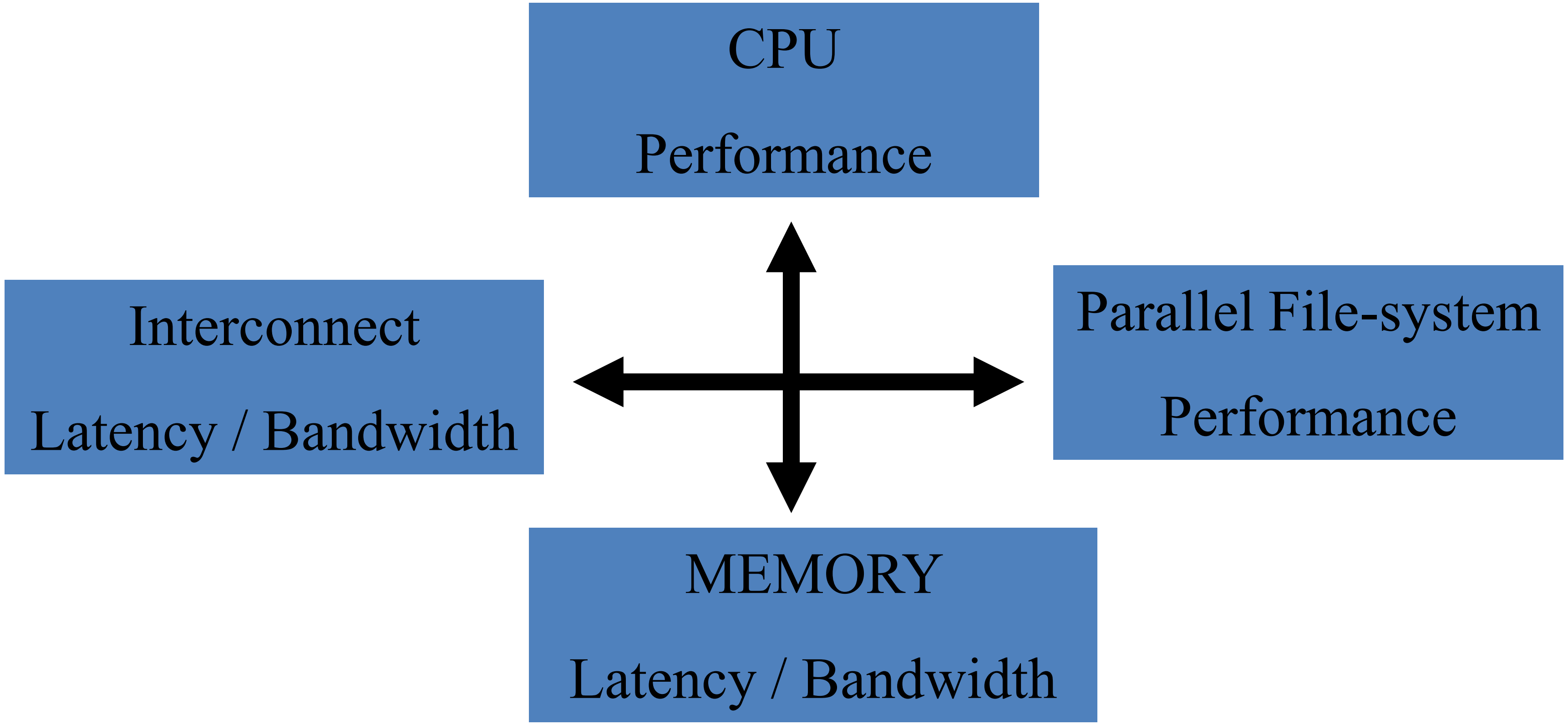


- Network latency:
  100ns per hop
  500ns normally
  2µs max
- MPI latency
  ~1.2µs small messages

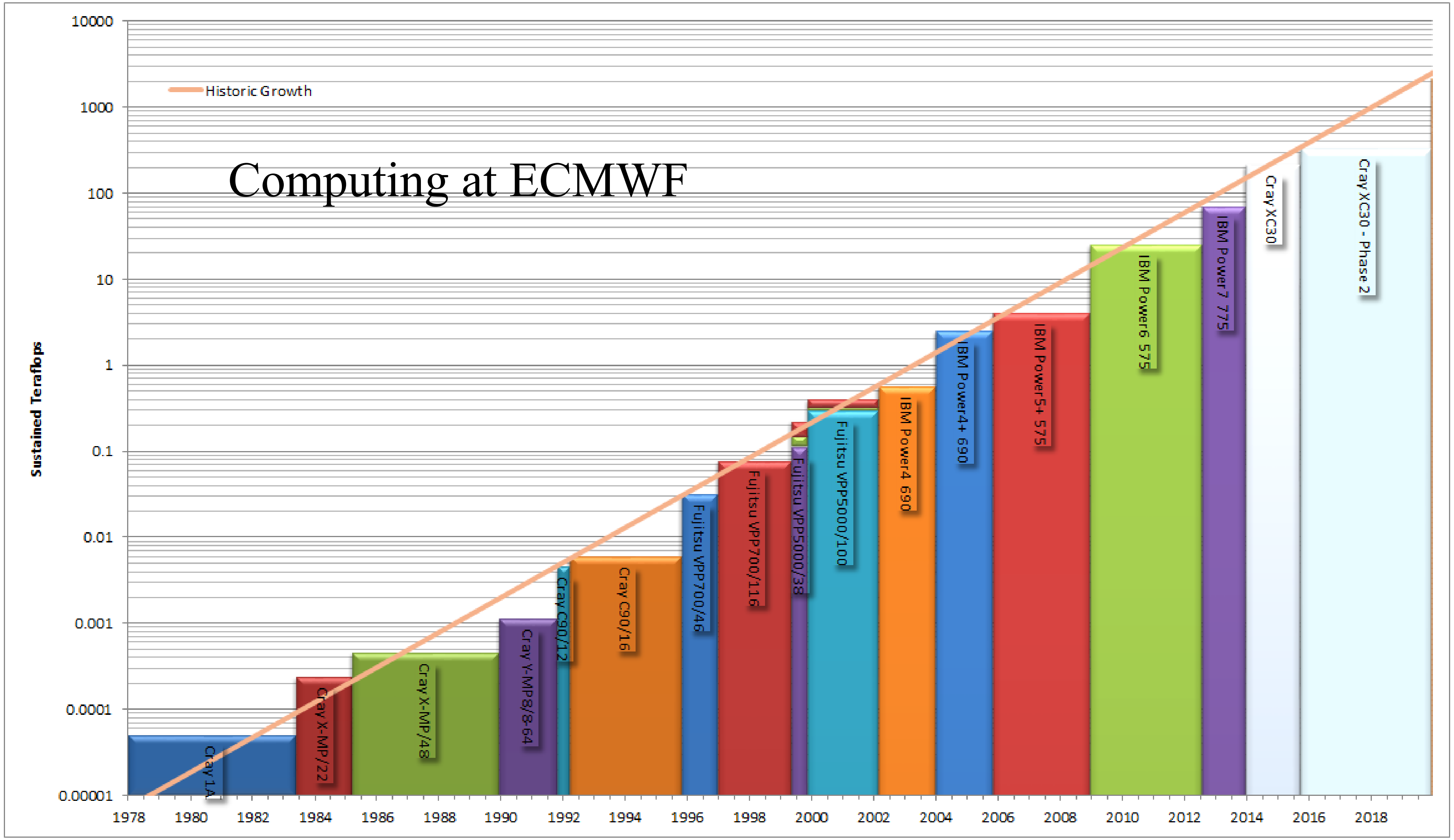
T2047 IFS global model (10 km) performance on CRAY XE6, 2012

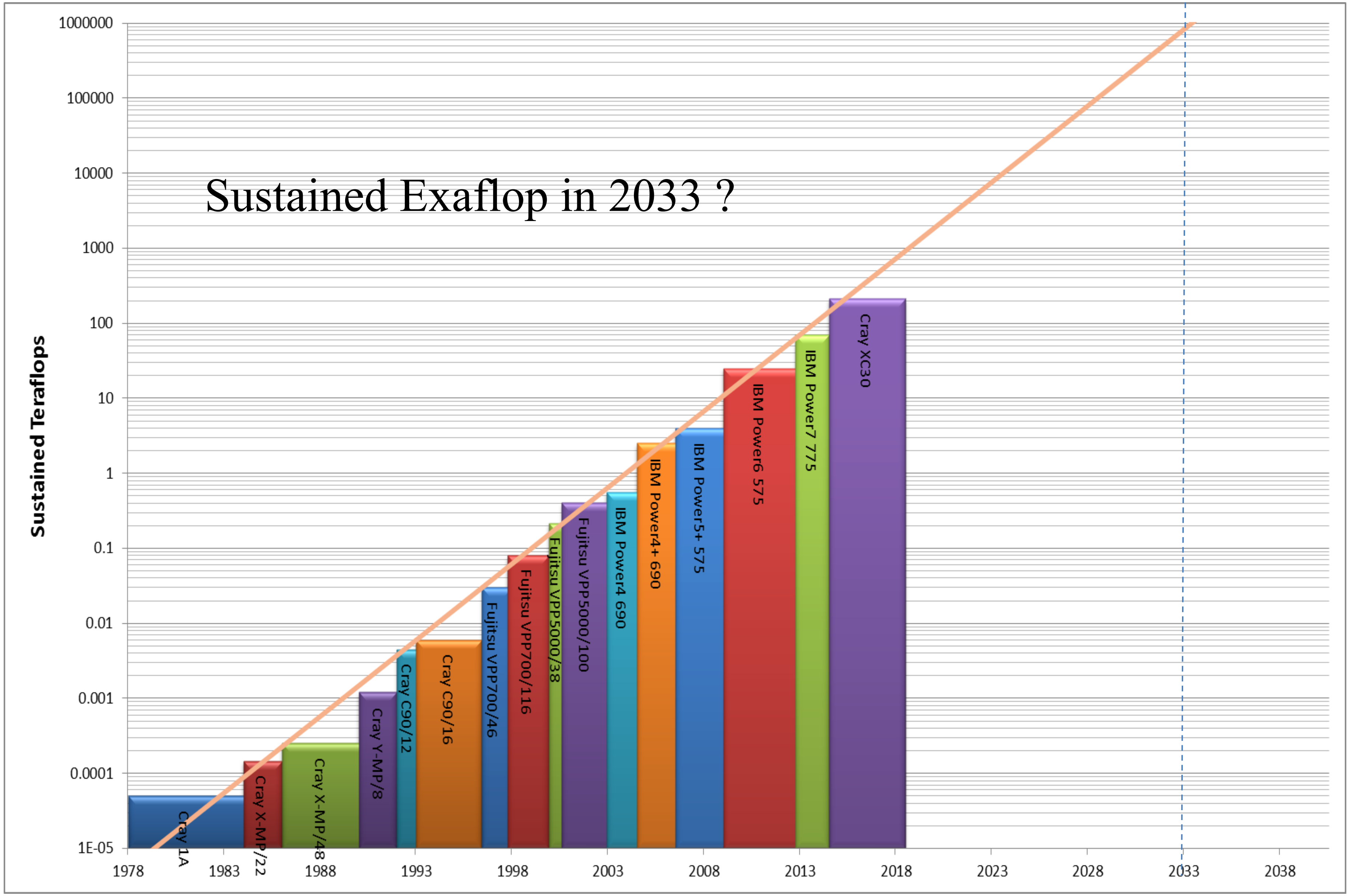


Key Architectural Features of a Supercomputer



“a balancing act to achieve good sustained performance”


Computing at ECMWF
Historic Growth
Sustained Teraflops
10000
1000
100
10
1
0.1
0.01
0.001
0.0001
0.00001
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
Cray 1A
Cray X-MP/22
Cray X-MP/48
Cray Y-MP8/8-64
Cray C90/12
Cray C90/16
Fujitsu VPP700/46
Fujitsu VPP700/116
Fujitsu VPP5000/38
Fujitsu VPP5000/100
IBM Power4 690
IBM Power4+ 690
IBM Power5+ 575
IBM Power6 575
IBM Power7 775
Cray XC30
Cray XC30 - Phase 2

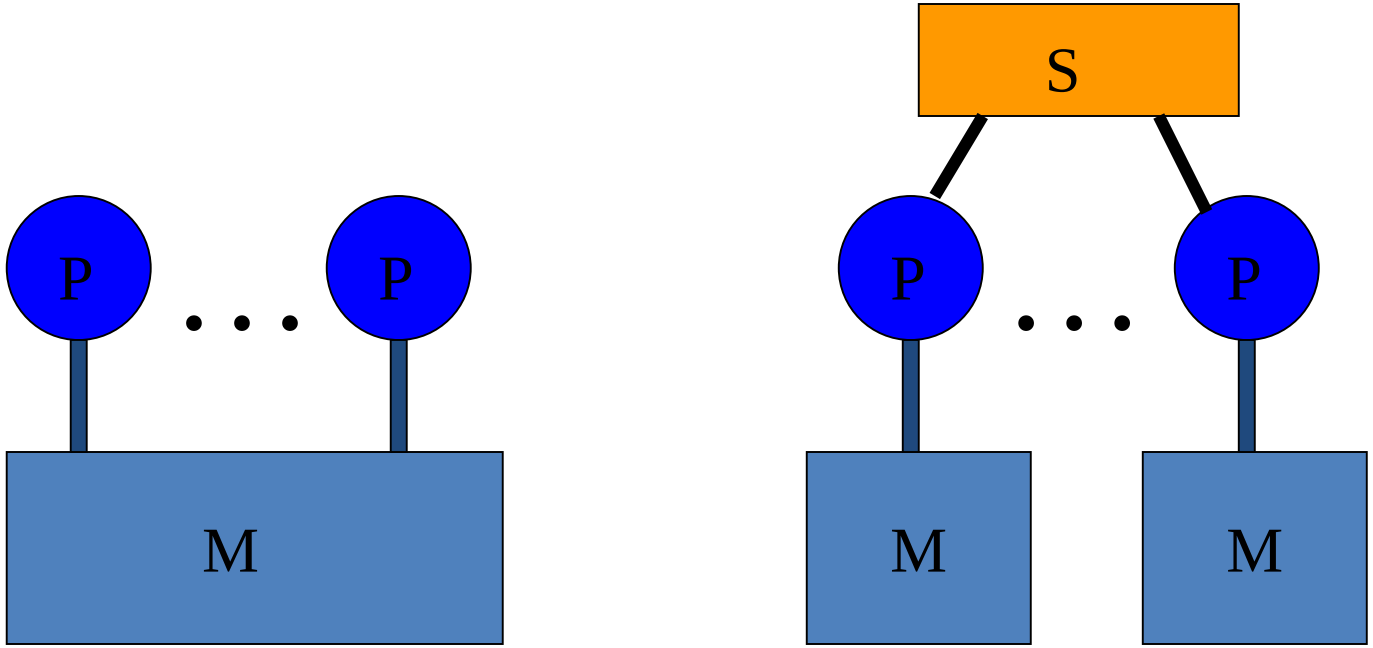

Sustained Exaflop in 2033 ?
Sustained Teraflops
1000000
100000
10000
1000
100
10
1
0.1
0.01
0.001
0.0001
1E-05
1978
1983
1988
1993
1998
2003
2008
2013
2018
2023
2028
2033
2038
Cray 1A
Cray X-MP/22
Cray X-MP/48
Cray Y-MP/8
Cray C90/12
Cray C90/16
Fujitsu VPP700/46
Fujitsu VPP700/116
Fujitsu VPP5000/38
Fujitsu VPP5000/100
IBM Power4 690
IBM Power4+ 690
IBM Power5+ 575
IBM Power6 575
IBM Power7 775
Cray XC30

Parallel Computing 101

- Most parallel computing can be putting into one of two catagories:
  - Shared memory computing
    - All cores in the job can see all the memory available
    - Data is passed around by writing into the same array/variables
    - Entire domain sits on one memory system
    - Main programming language OpenMP (Thursday)
  - Distributed memory computing
    - Each core has dedicated memory that only it can access
    - Data is passed around by sending messages
    - Domain needs to be split between different memory systems
    - Main programming language MPI (Wednesday)
- Hybrid Distributed Shared Memory Computing now becoming the norm
- More details on Tuesday

Types of Parallel Computer

P=Processor
M=Memory
S=Switch

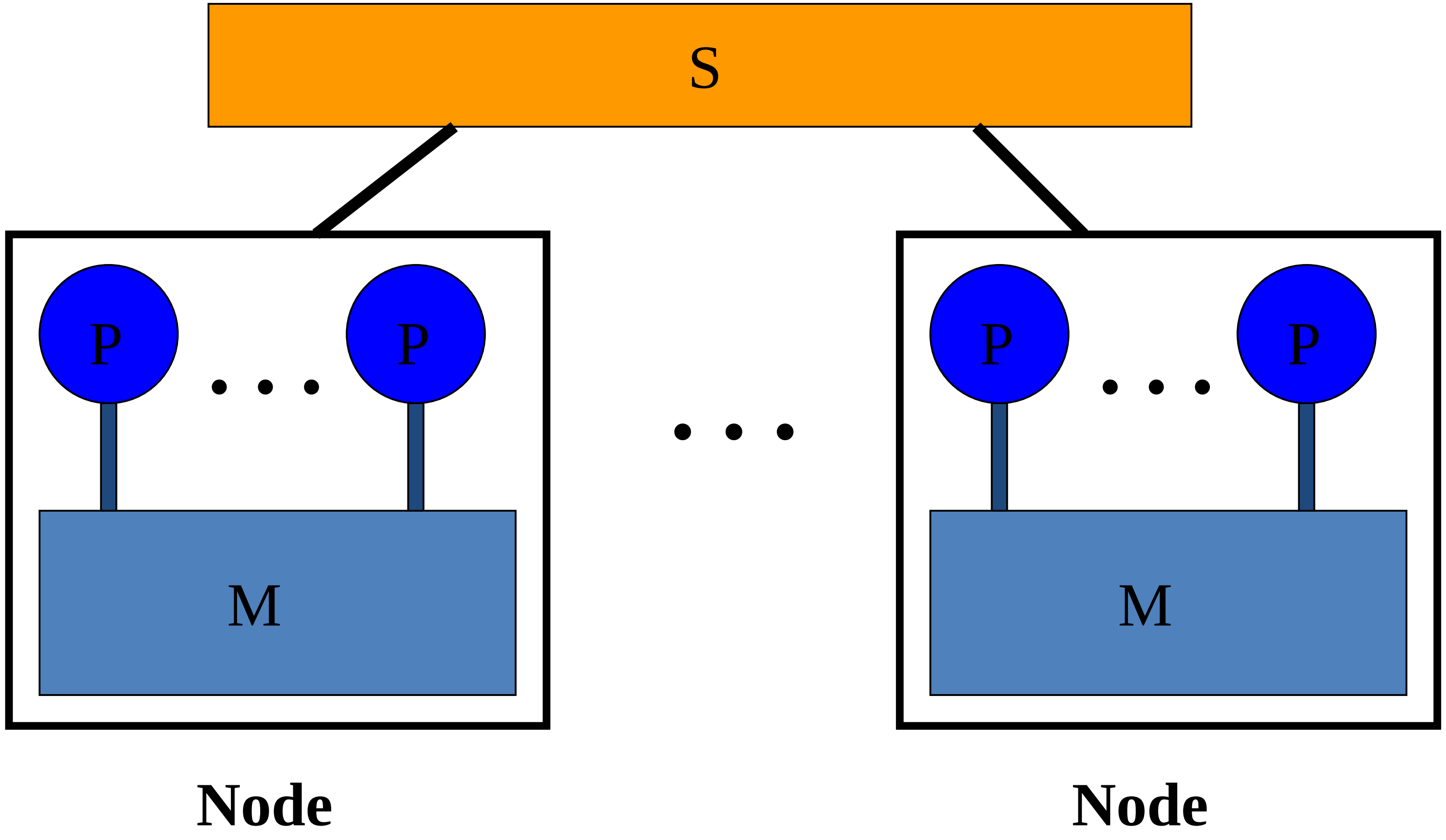


Shared Memory

Distributed Memory

CRAY Hybrid Cluster
(Distributed + Shared memory)

P=Processor
M=Memory
S=Switch



Common Terms

- SIMD
  - Single Instruction Multiple Data
    - Vector calculations on a core
- MIMD
  - Multiple Instruction Multiple Data
    - Can be used to describe modern parallel codes
- SPMD
  - Single Program Multiple Data
    - This is where most parallel programs that use MPI sit
- MPMD
  - Multiple Program Multiple Data
    - Generally a MASTER/SLAVE concept
- SISD
  - Single Instruction Single Data
    - Serial applications run in this mode
- NUMA
  - Non-Uniform Memory Access
    - Describes systems with different access times to different physical memory locations
- Latency
  - Time to send data from one point to another
- Bandwidth
  - “Width” of pipe between two points

Common Terms II

- Tasks
  - MPI
- Threads
  - OpenMP
- Hyperthreads
  - Multiple virtual cores running on a single core
- Register
  - Where data for “immediate” computation is stored
- Vector
  - Array of common data
- FMA
  - Fused Multiply Add
    - 3 pieces of data can be summed and multiplied together in one cycle
    - Can give different results compared to doing each part separately

Six Levels of Parallelism

- Node Level
  - Split your domain across a number of identical nodes or MPI tasks
  - How many depends on how quick and how much memory is needed
- Socket Level
  - Different sockets have different access times to different banks of memory
  - Need to think how to split up memory and control in OpenMP
- Core Level
  - The number of cores on a socket will determine how to split tasks v threads
- Vector Level
  - How many pieces of data can each instruction operate on at the same time
- Pipeline Level
  - Function of the number of hyperthreads available
  - Measure of how many different instruction streams are available
- Instruction Level
  - How many instructions can be combined into one cycle
  - E.g. FMA
- Seventh level of Job level
  - E.g. Ensembles and multiple runs
  - Good for ensuring correctness, reproducibility and fault tolerance

Further Reading

- https://www.hpcwire.com/2011/03/08/compilers_and_more_programming_at_exascale/
- https://www.hpcwire.com/2011/03/28/compilers_and_more_expose_express_exploit/
- https://www.hpcwire.com/2011/04/14/compilers_and_more_exascale_programming_requirements/
- https://en.wikipedia.org/wiki/Supercomputer
- https://en.wikipedia.org/wiki/FLOPS
- https://en.wikipedia.org/wiki/Central_processing_unit
- https://en.wikipedia.org/wiki/Network_topology
- http://www.cray.com/sites/default/files/resources/CrayXCNetwork.pdf