

HPC and modeling

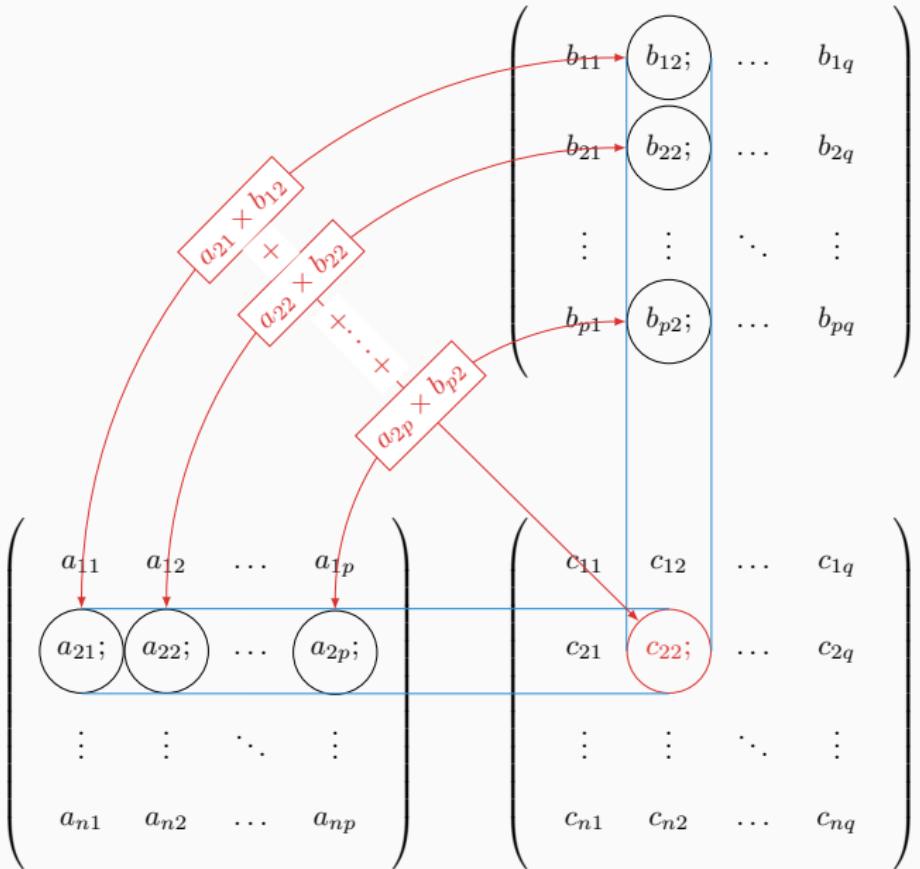
Chapter 1 – A first take on parallelism

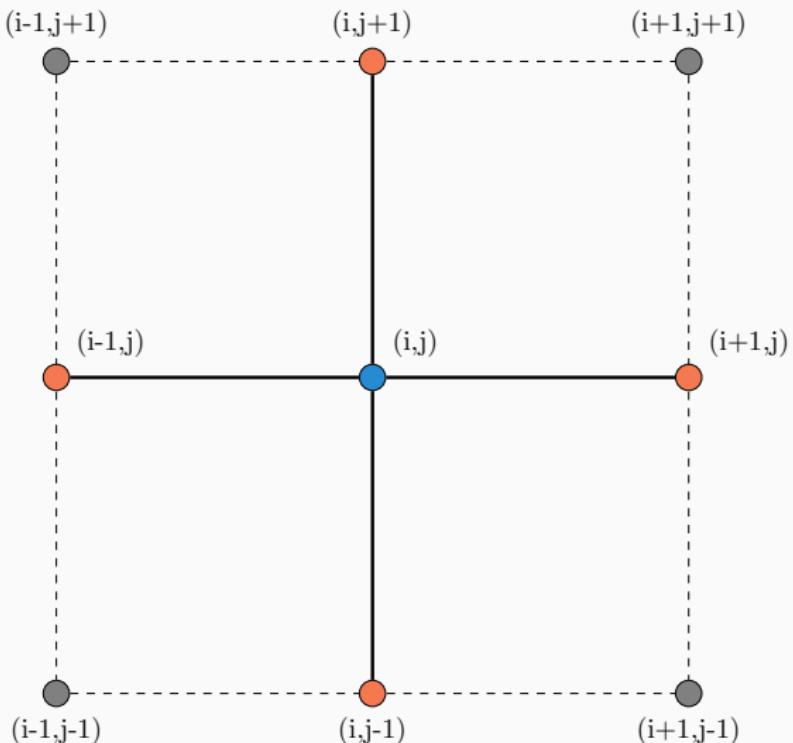
M2 – MSIAM

5 octobre 2016

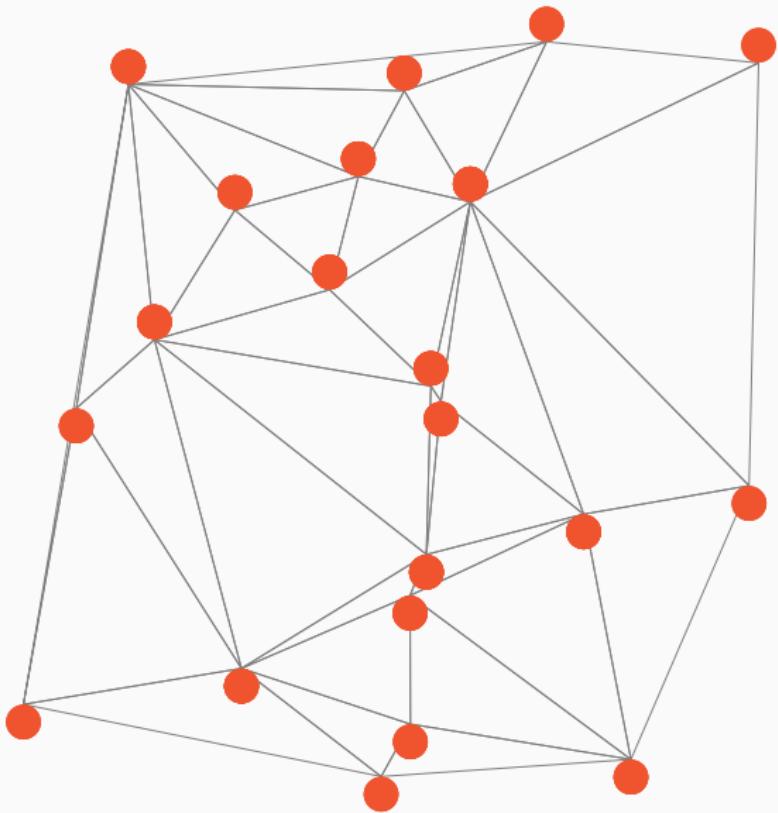
Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	Tianhe-2 [MilkyWay-2] - TH-IVB-FEP 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
3	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
4	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
5	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

MATRIX MULTIPLICATION





GRAPH



AGENDA

Organization

Introduction

Parallel platforms

Contact

Christophe Picard : christophe.picard@imag.fr

Office 174 – Imag Building – 1st floor

Email headers : [MHPC]

Course website :

<http://chamilo2.grenet.fr/inp/courses/ENSIMAGWMM9AM28>

Class organization

Lectures, labs and project.

- Decrease time to solution.
- Solve larger problem.
- Combine resources of several processing units : gain access to more memory and more processing power.
- Harness the processing power of modern architectures.
- Use idle computer to perform embarrassing parallelism computation (SETI@home).
- Improve the precision of computations in a limited time (weather forecast).

- Understand the different level of parallelism.
- Apprehend the concepts required for real-life applications.
- Experiment with different tools of parallel computing.
- Put into practice the theoretical concepts through an application.

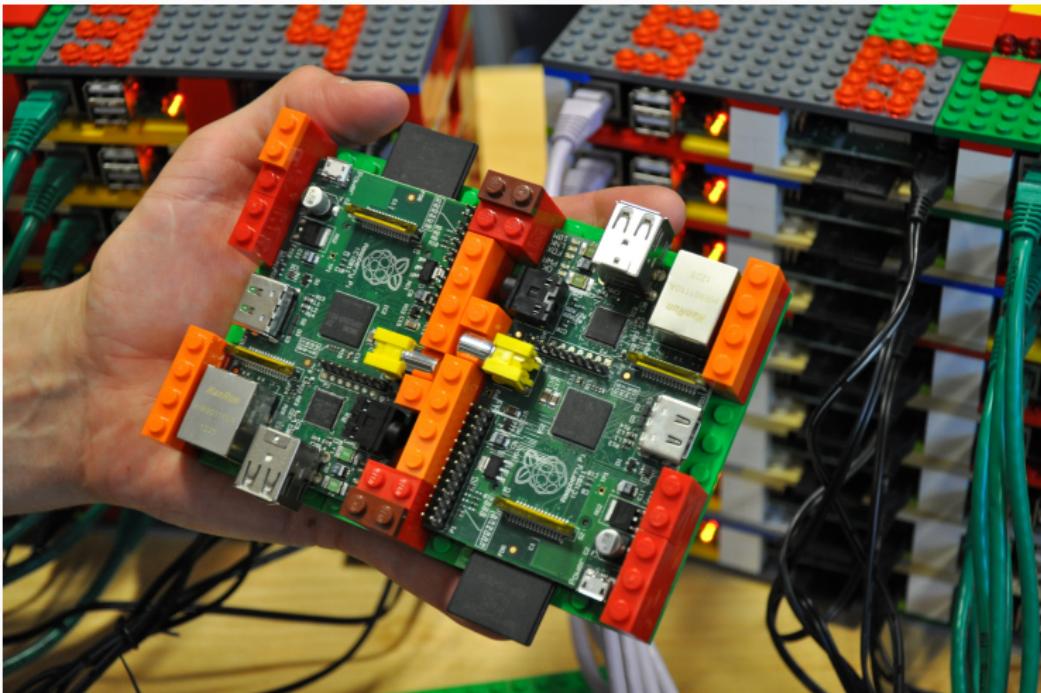
- Introduction – Scale of parallelism (1 class)
- Programming pattern in parallelism – Part 1 (1 class)
- Labs – Introduction to OpenMP (1 lab)
- Programming pattern in parallelism – Part 2 (1 class)
- Labs – Introduction to MPI (1 lab)
- Small scale project (6 classes)
 - Introduce the problem.
 - Analysis of the code.
 - Strategies for parallelism.

AGENDA

Organization

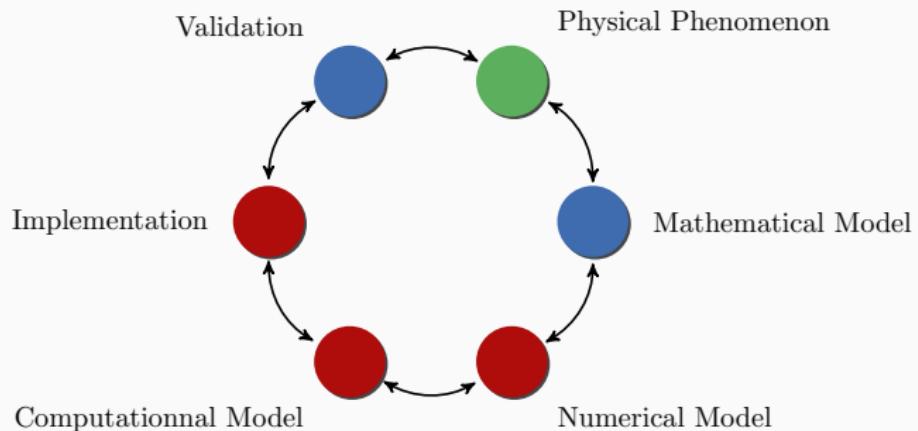
Introduction

Parallel platforms





- Sequential programming is limited
 - applications are parallel.
 - access to memory is limited.
 - engineering/cost limitations : it is easier to increase the number of unit than the frequency of a processor.
- Expectations for the applications are increasing
 - weather forecast
 - fluid-structure
 - CAD
 - big-data
 - cryptology
- Performances are evolving
 - Hardware is faster.
 - Algorithms are more efficient.



Intensive computation

- Solve a problem faster.
- Models are more sophisticated.
- Increase resolution of models.
- Increase interactivity.

Example

- Improve the rate : compute N problems simultaneously
 - ⇒ Run the same sequential program N times using different inputs.
- Decrease response time : solve a problem with N times faster.
 - ⇒ The program is executed only once using N processes.
- Increase the size of the problem : compute a problem N times larger.
 - ⇒ The program is executed only once combining N memory resources.

HOW TO INCREASE PERFORMANCES?

HPC attempts to speed solution by dividing task into sub-tasks and executing simultaneously on different processing units.

- Identify where parallelism will be the most effective.
- Know the set of technological constraints.
- Design solution adapted to the problem and the constraints.

LIMITATIONS OF SEQUENTIAL SOLUTIONS

Clock speed limitation

- Current leakage.
 - Power consumption.
 - Heat dissipation.
- }
- Not compatible with mobile devices

Standard optimisations

- Instruction prefetching
 - Instruction reordering
 - Pipelined functions units
 - Branch prediction
 - Functional unit allocation
 - Hyperthreading
- }
- No control of the programmer

- Processors : multicore, memory, network, accelerators, instructions.
- Compilers : dedicated library, automatic parallelism.
- Algorithms : tailored algorithms.
- Mathematics : adapted numerical methods, evolutionary methods.

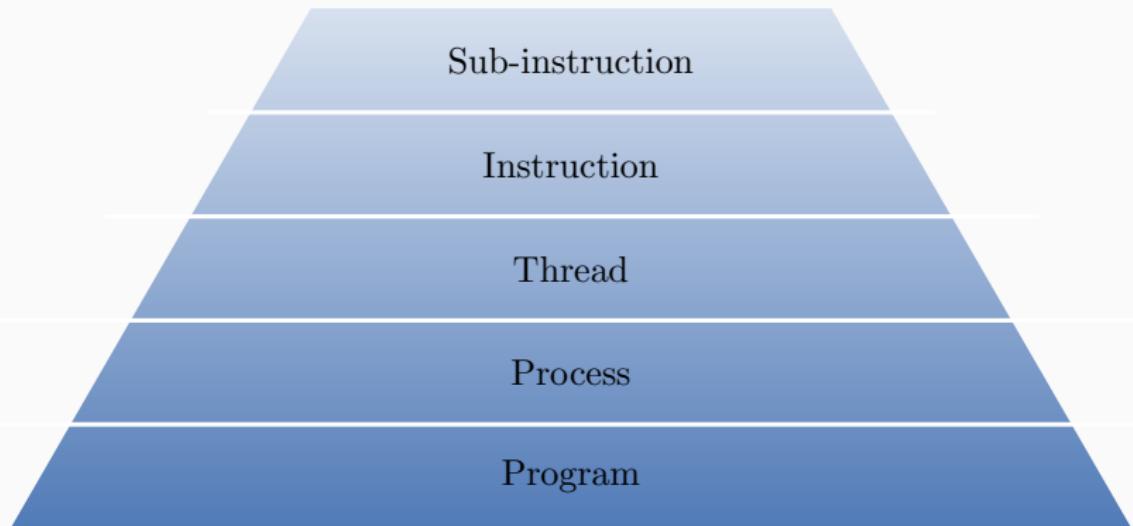
AGENDA

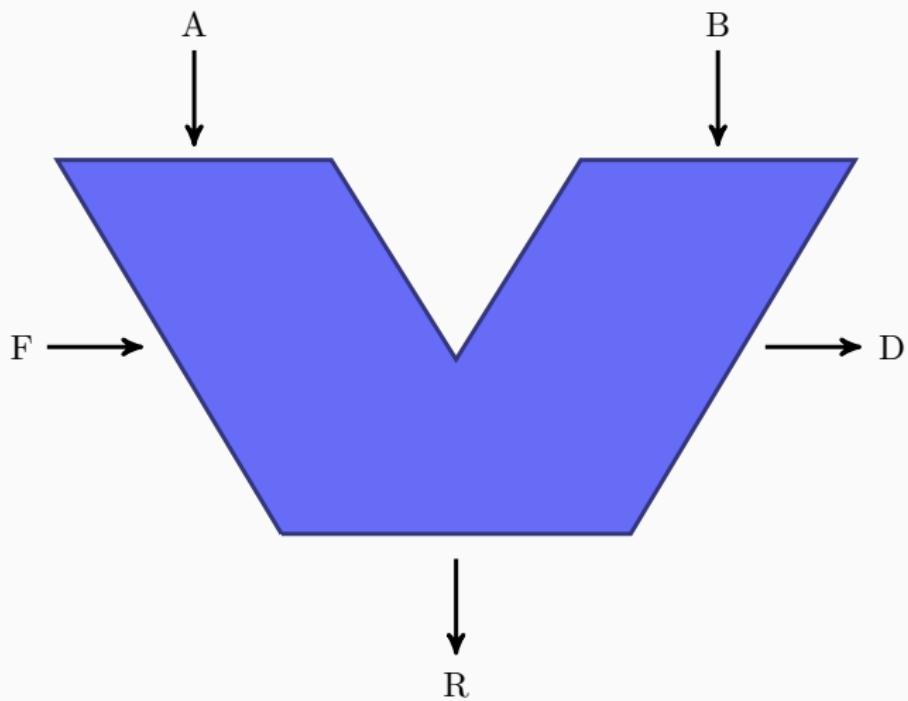
Organization

Introduction

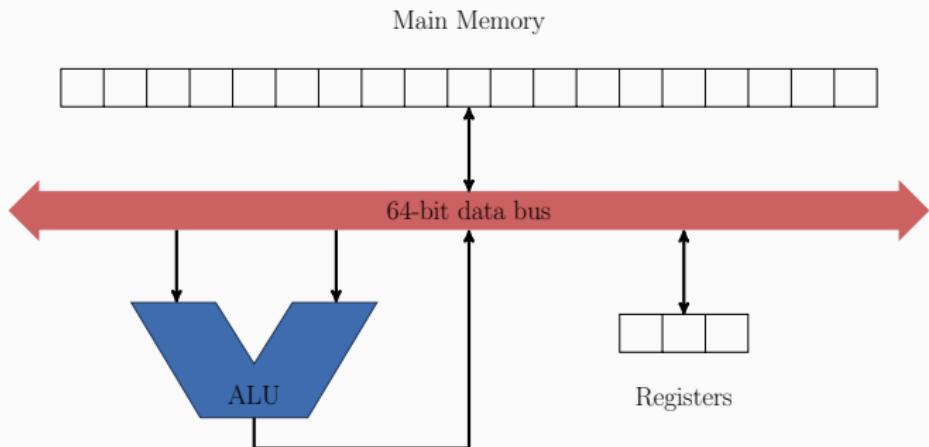
Parallel platforms

PARALLEL HIERARCHY

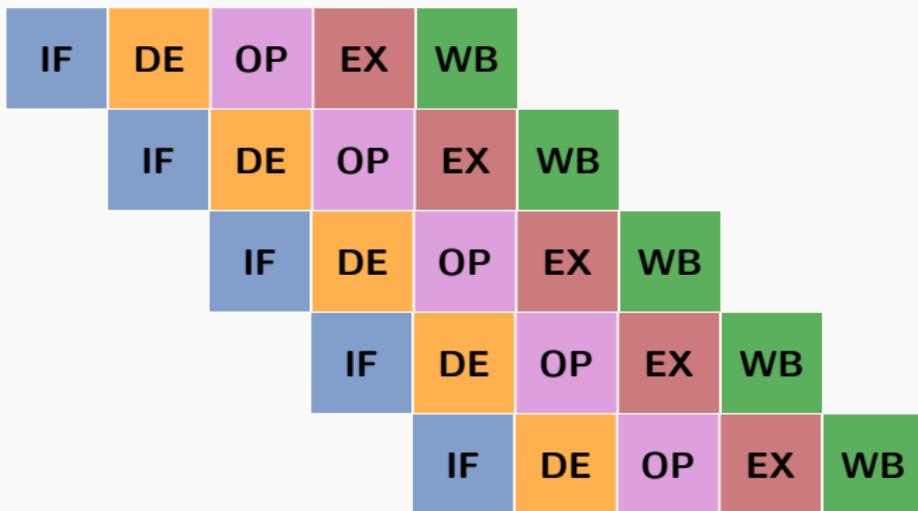




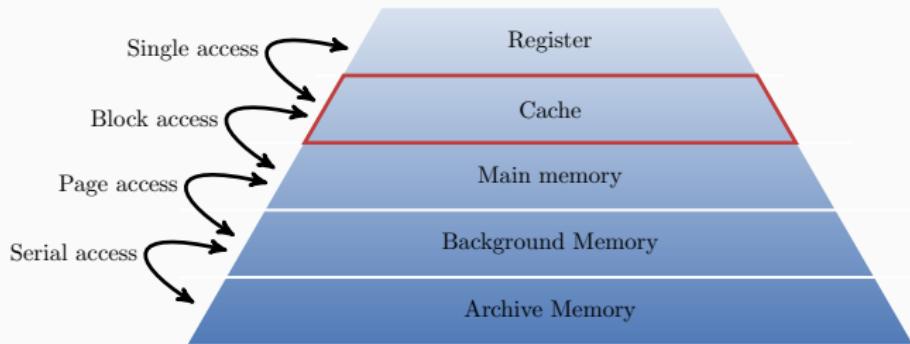
ABSTRACT PROCESSOR



PIPELINE ARCHITECTURE



MEMORY HIERARCHY



- SISD – Single Instruction, Single Data
- SIMD – Single Instruction, Multiple Data
- MISD – Multiple Instruction, Single Data
- MIMD – Multiple Instruction, Multiple Data

Most modern architectures are based on MIMD principles.

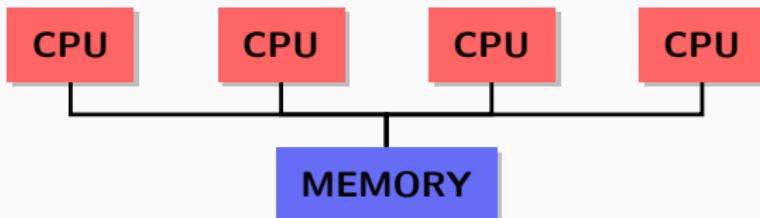
- Multiple processing units : all the processing unit shares the same global memory.
 - Scaling is complex from the algorithm point of view but also from the technical point of view.
 - Intuitive programming since most of modern programming tools manage memory accesses automatically.
 - Local programming
- Cluster : aggregation of processing unit link through a high speed network. Memories are locals and each unit has its own memory. There is no global memory access.
 - Scaling only depends on the number of resources that are allocated to the cluster.
 - Specific communication protocol must be used for the processors to interact.
 - Optimization of the ratio computation/communication requires careful design.

- Hybrids : heterogeneous collection of processing unit with a level of distributed memory. Each node can itself be a shared memory or a distributed memory architecture.
 - Programming is hard : at least two parallel paradigms must be used
 - Optimisation is complex.
 - Performance gain is better.
 - Some resources must be virtualized.

- Grid : heterogeneous processing unit link through low speed network (LAN/WAN).
 - Low network
 - No administration required.
 - Only for applications having low network requirements (SETI@HOME)
- Cloud : virtualization of hardware resources.
 - Low performances compared to standard architectures.
 - Cost effective for low usage.

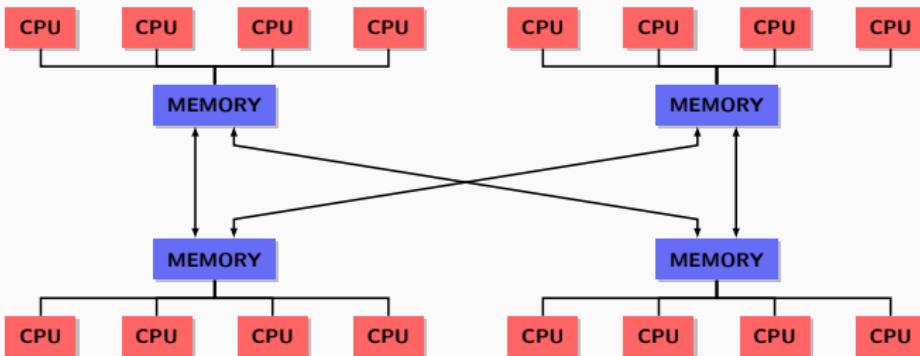
SHARED MEMORY (1)

- All the processors share the same memory space. They communicate using reading and writing shared variables.
- Each processing unit carry out its task independently but modification of shared variables are instantaneous.
- Two kind of shared memories
 - SMP (Symmetric MultiProcessor) – All the processors share a link to the memory. Access to the memory is uniform.



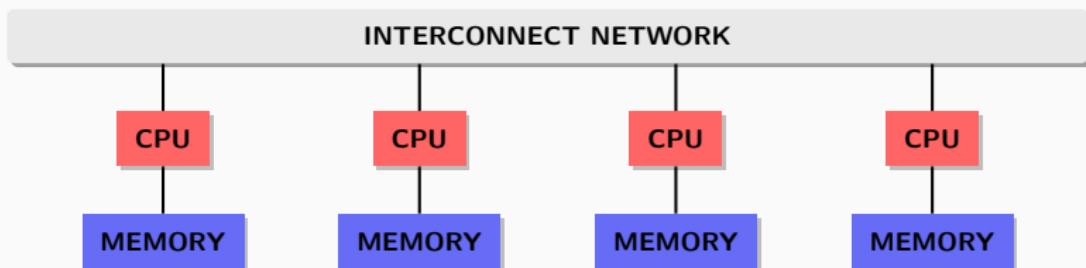
SHARED MEMORY (2)

- NUMA (NonUniform Memory Access) – All the processors can access to the memory but not uniformly. Each processor has a preferred access to some memory part.



- Decrease the risk of bottleneck to memory access.
- Local memory cache on each processor to mitigate the effect of non-uniform access.

- Each processor has its own memory. There is no global memory space.
- Each processor communicate with the others using messages.
 - Modification of variables are local and only the processor managing the memory can access it.
 - Each processor work independently on its own set of variables.
 - The speed of the resolution depends on the architecture : network, topology, processors.
 - Can scale easily.



Rank	Site	System	Cores	Rmax [TFlop/s]	Rpeak [TFlop/s]	Power (kW)
1	National Supercomputing Center in Wuxi China	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	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
3	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
4	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
5	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

NEXT TIME...

Introduction to shared memory computing