सुस्वागतम्
நல்வரவு
ସୁସ୍ୱାଗତମ
సుస్వాగతం
સુસ્વાગતમ
সুস্বাগতম
ಸುಸ್ವಾಗತ
സുസ്വാഗതം
সুস্বাগতম
ਸੁਆਗਤਮ
सुस्वागतम्
خوش آمدید

Ministry of Electronics and Information Technology
Government of India

High Performance Computing

Introduction to High-Performance Computing

Sowmya shree N
HPC Tech, C-DAC Pune
(ssowmya@cdac.in)

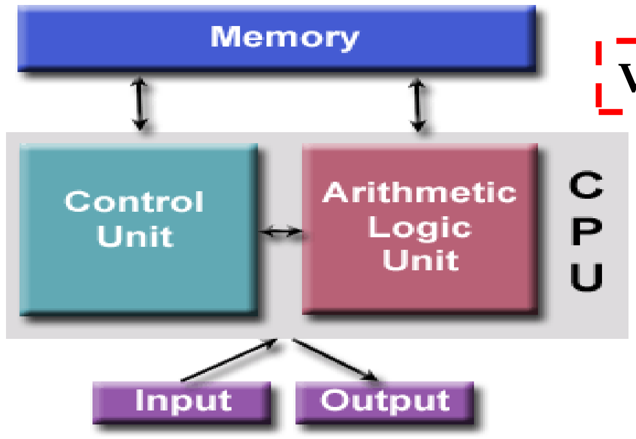
Agenda

- ❑ Introduction to Processor Architecture and Classification
- ❑ Evolution of Processor
- ❑ Evolution of Computing Motherboards
- ❑ Data Transfer Methodology
- ❑ HPC Systems Architecture
- ❑ Parallel Architectures & Programming Models
- ❑ Use cases of HPC

Traditional Computer Architecture

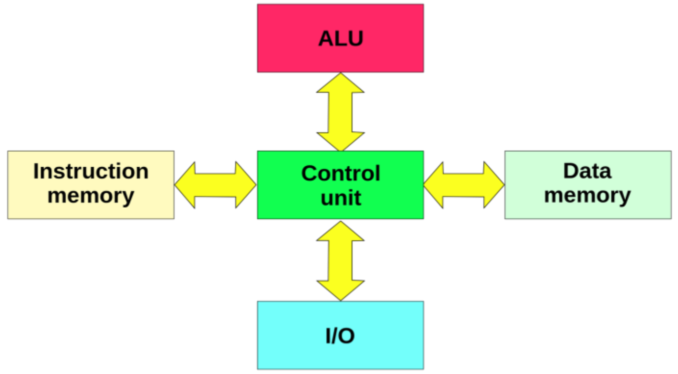
Instruction and Data bus based Classification

- Von-Neumann Architecture
  - ❖ Same bus is used
- Harvard Architecture
  - ❖ Different bus is used



Von-Neumann Architecture

Harvard Architecture

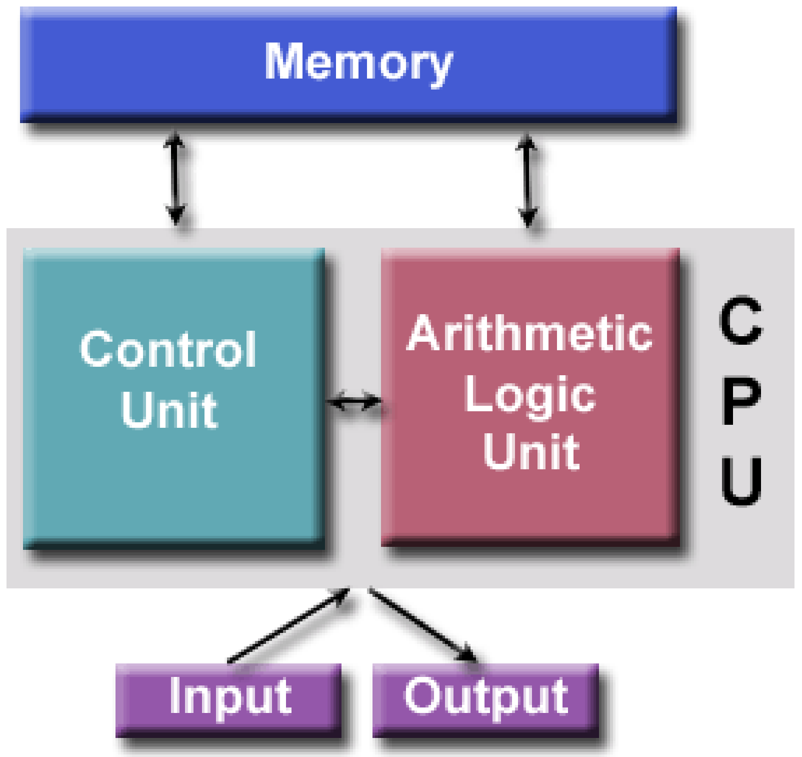


Instruction and Data Stream based Classification

- Single Instruction, Single Data (SISD) → Scalar Processor
- Single Instruction, Multiple Data (SIMD) → Vector Processor
- Multiple Instruction, Single Data (MISD) → Not realistic
- Multiple Instruction, Multiple Data (MIMD) → Super computer (HPC)

Von-Neumann Architecture

von-Neumann Architecture

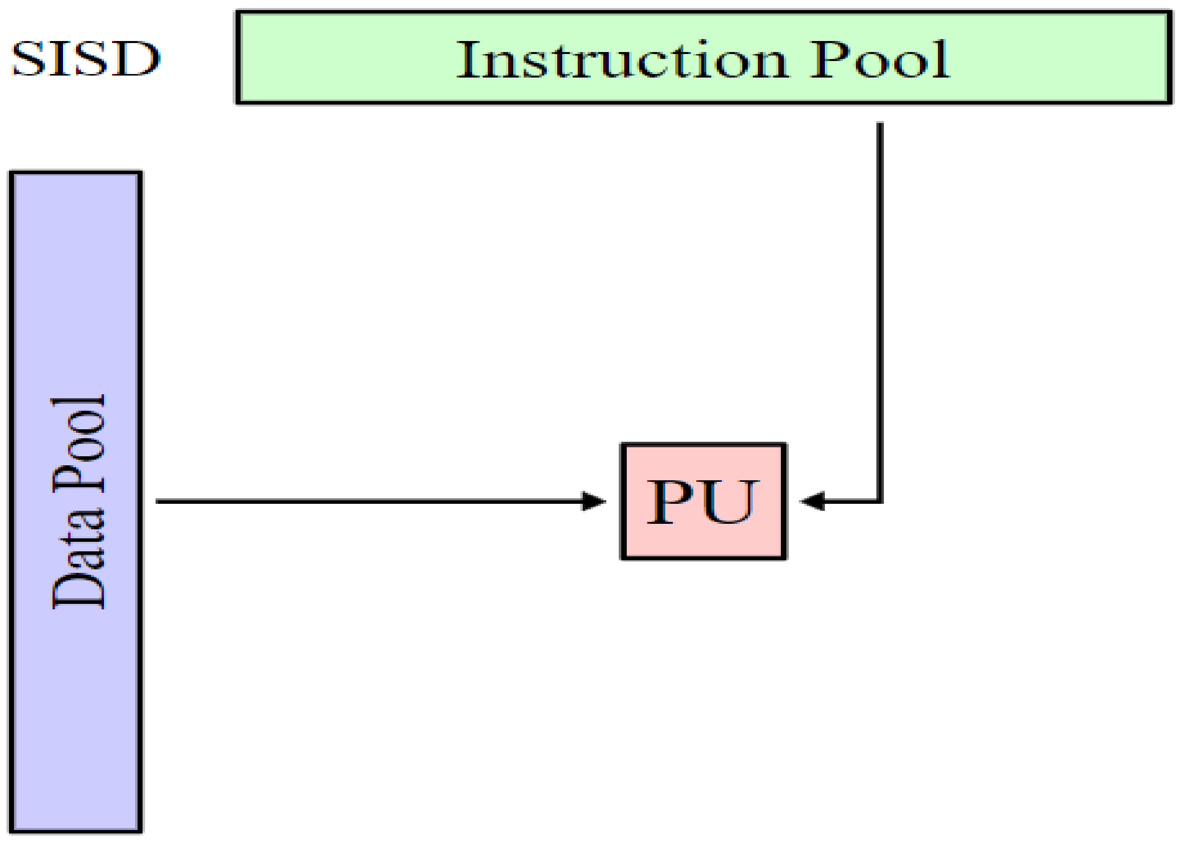


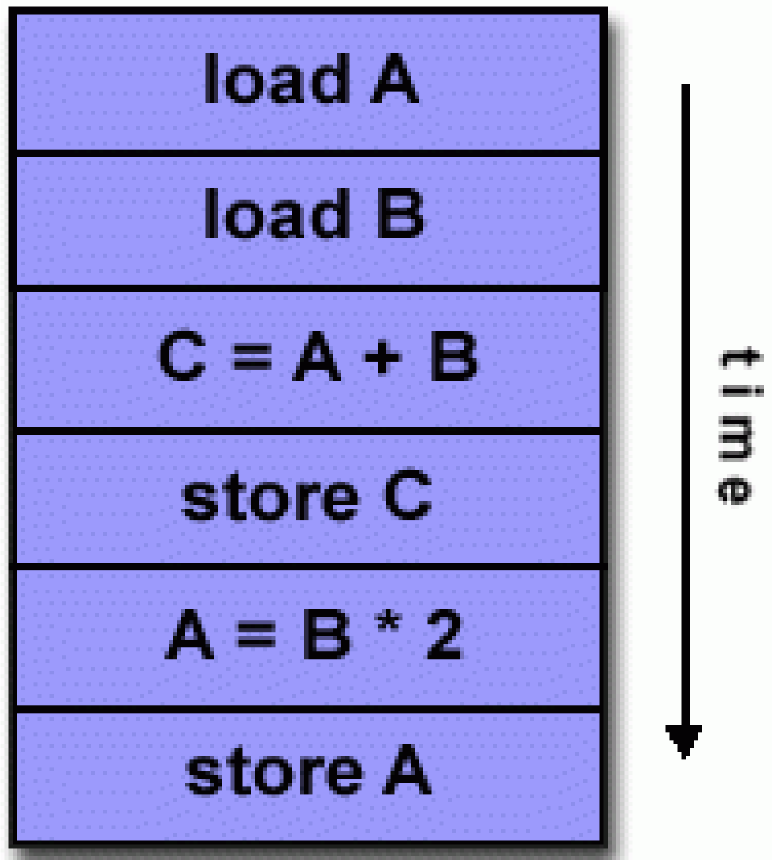
- ❑ Comprised of four main components:
  1. Memory
  2. Control Unit
  3. Arithmetic Logic Unit
  4. Input/Output
- ❑ Read/write, random access memory is used to store both program instructions and data
- ❑ Program instructions are coded data which tell the computer to do something
- ❑ Data is simply information to be used by the program
- ❑ Control unit fetches instructions/data from memory, decodes the instructions and then ***sequentially*** coordinates operations to accomplish the programmed task.
- ❑ Arithmetic Unit performs basic arithmetic operations
- ❑ Input/Output is the interface to the human operator

Parallel computers still follow this basic design, just multiplied in units.

Flynn's Classification

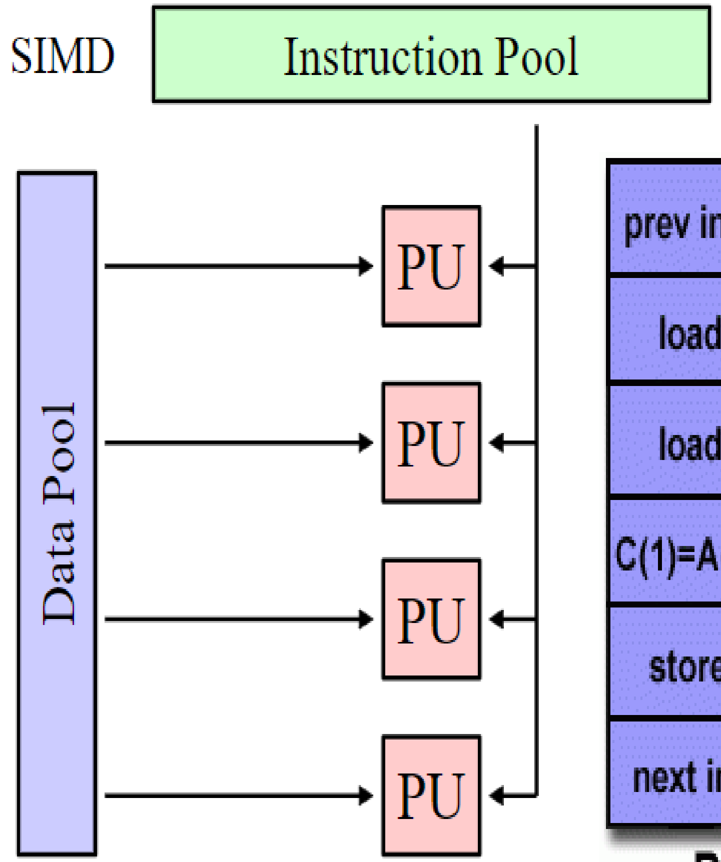
Single Instruction Single Data (SISD)

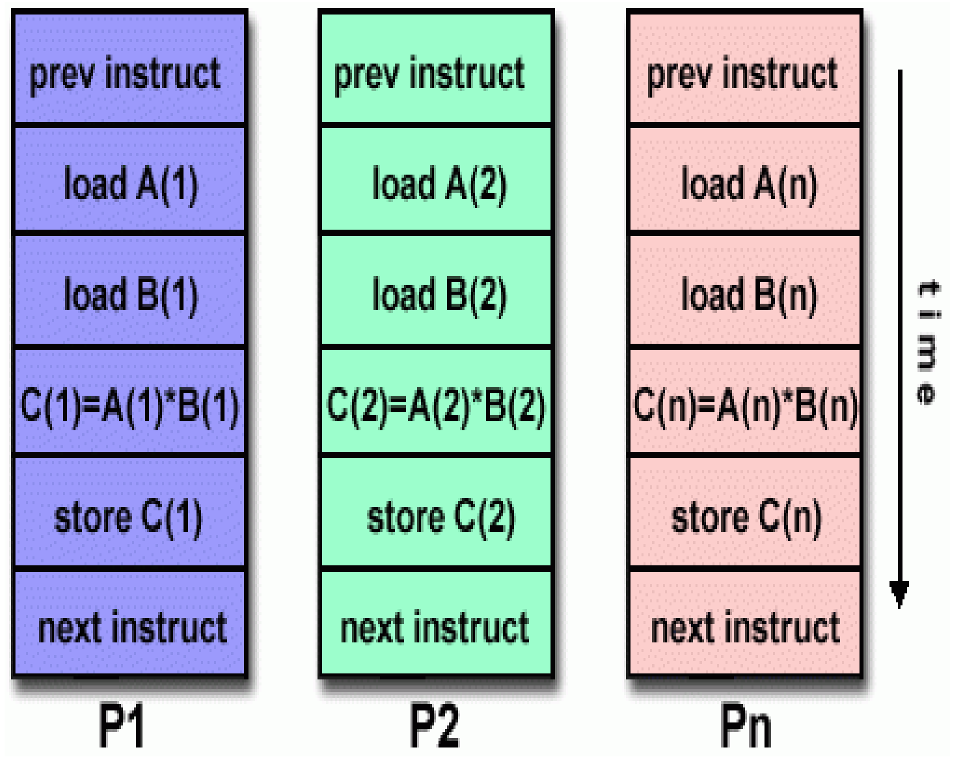


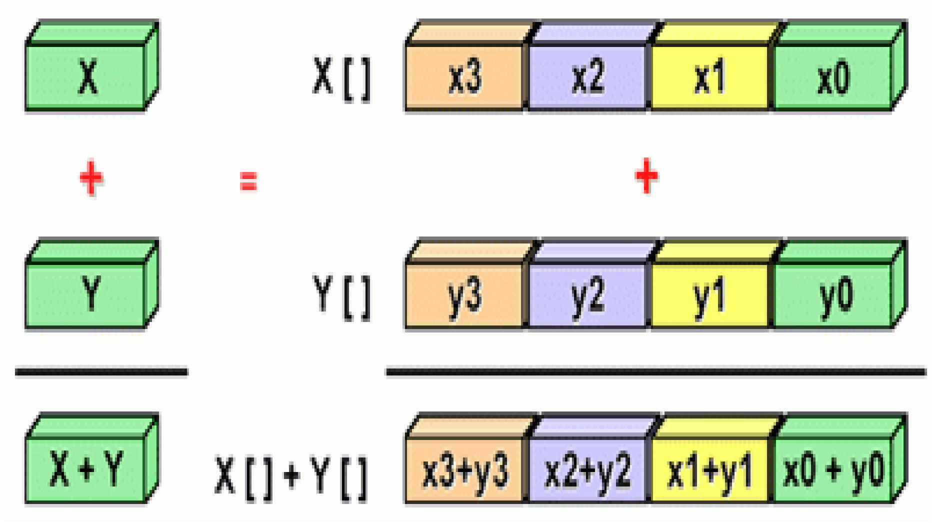


Single Instruction Multiple Data (SIMD)

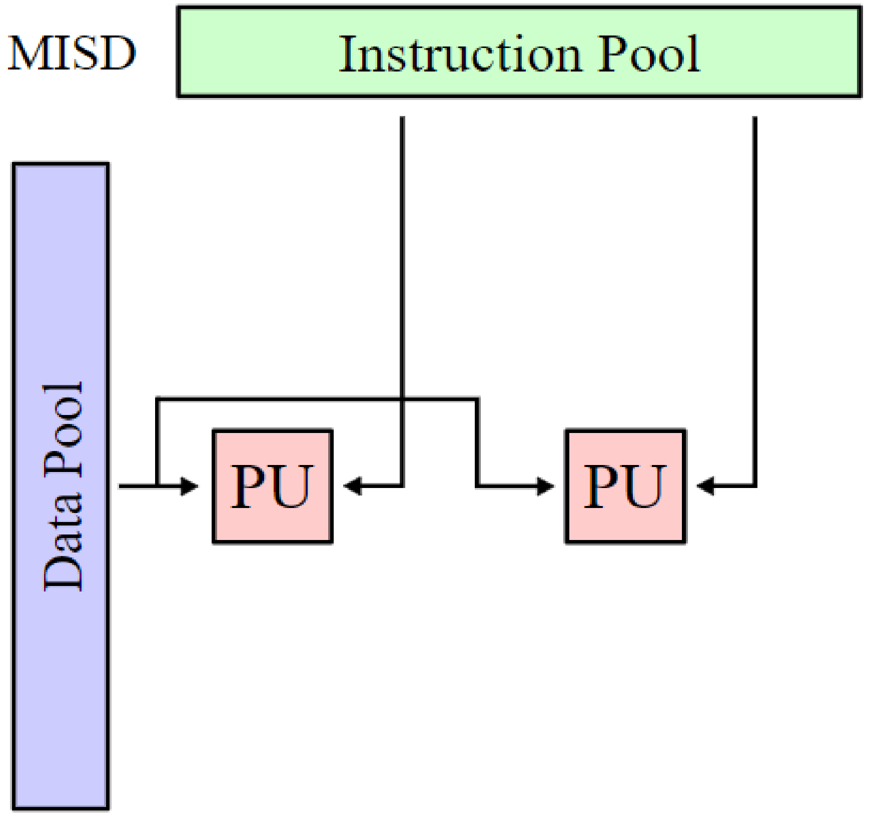
Single Instruction Multiple Data (SIMD)

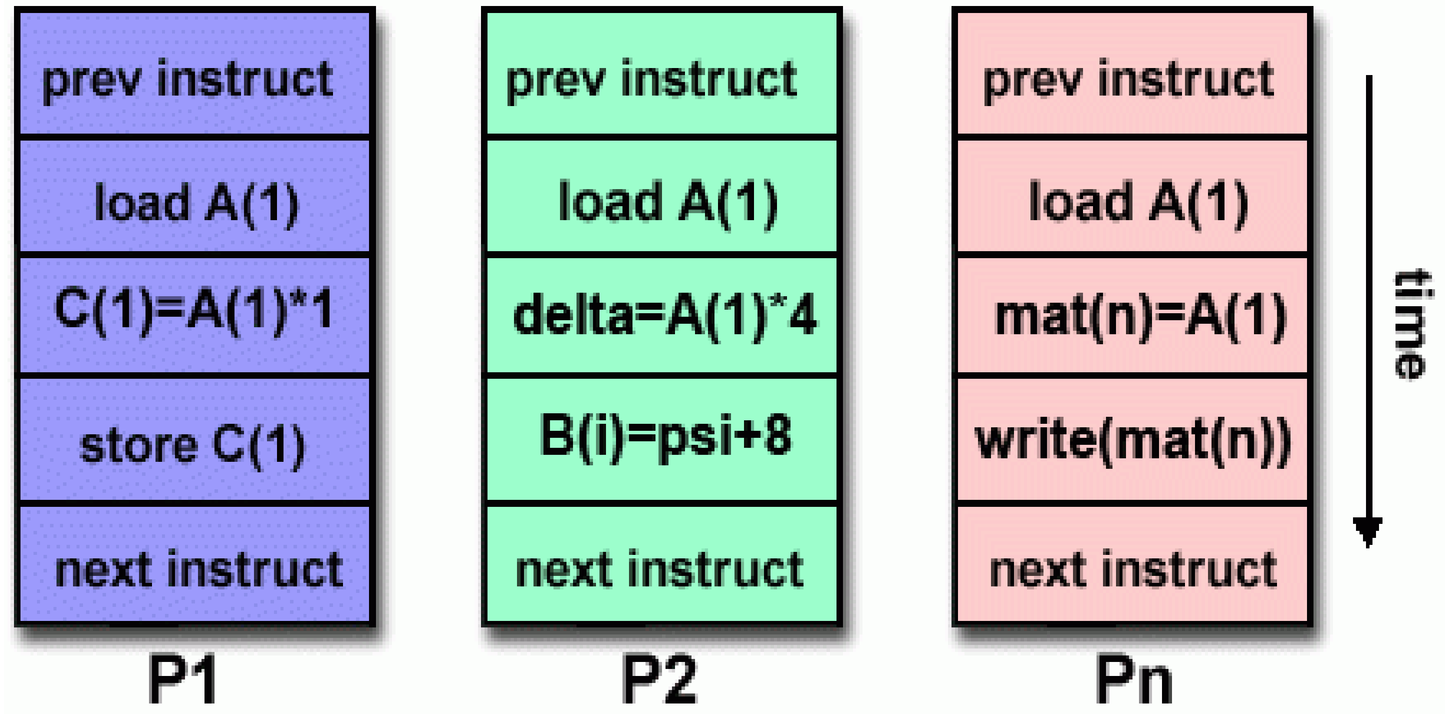




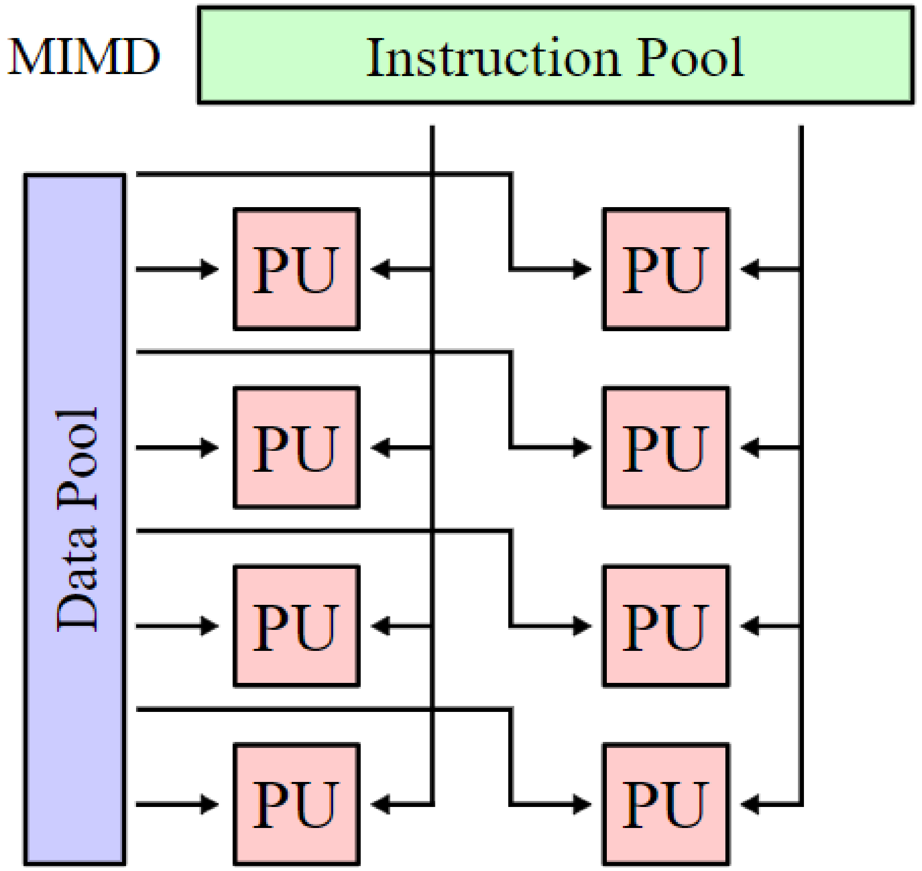


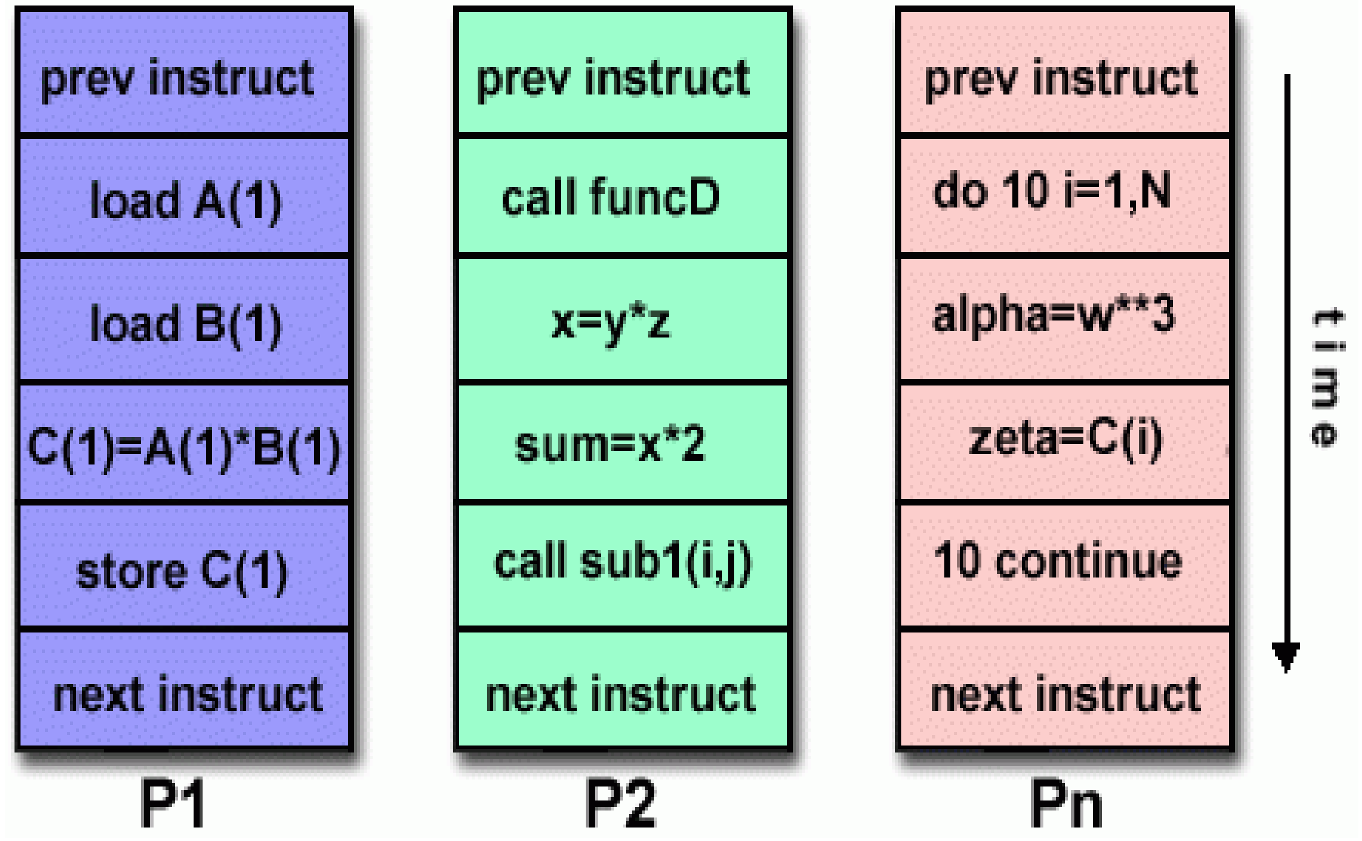
Multiple Instructions Single Data (MISD)





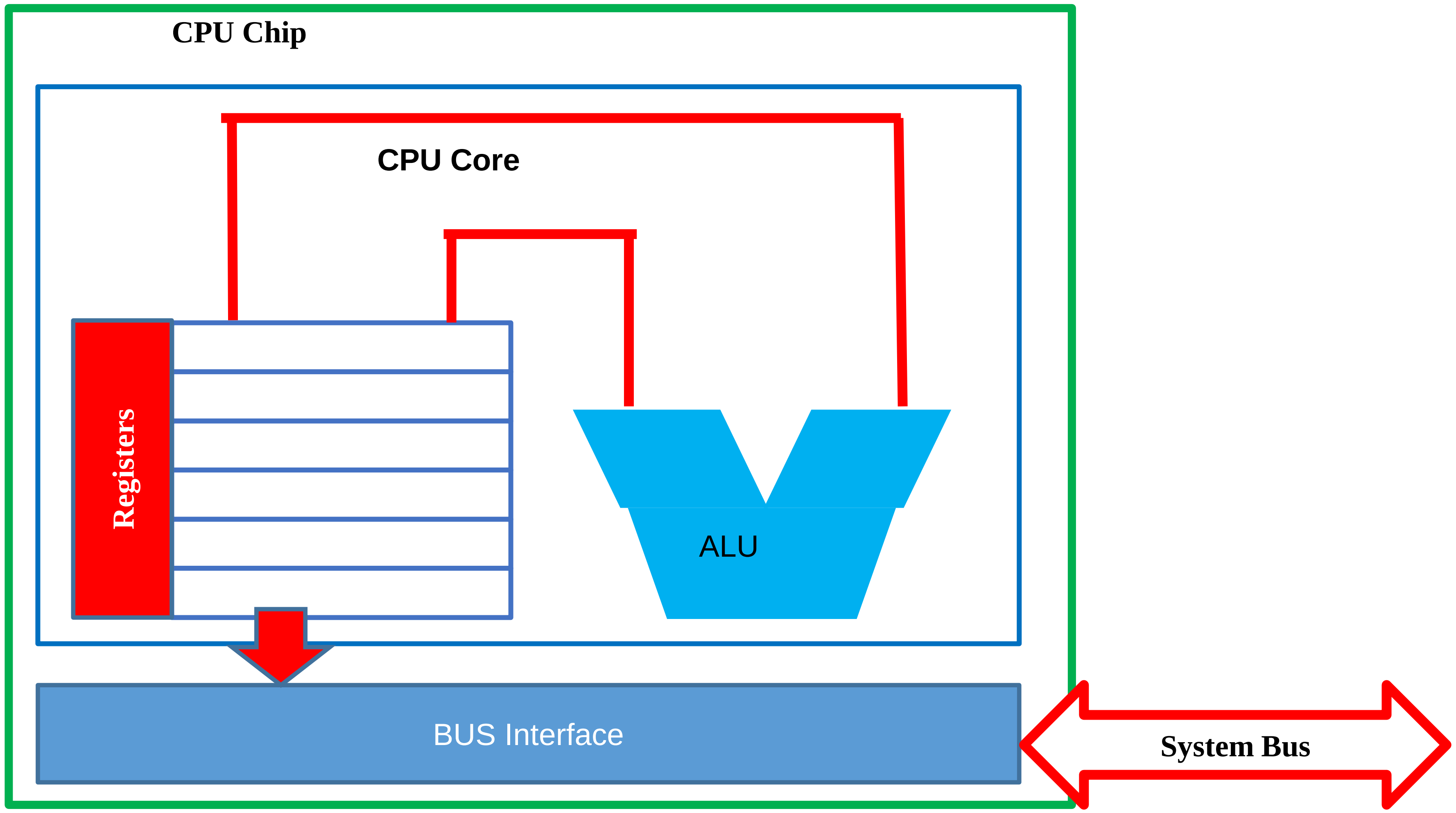
Multiple Instructions Multiple Data (MIMD)



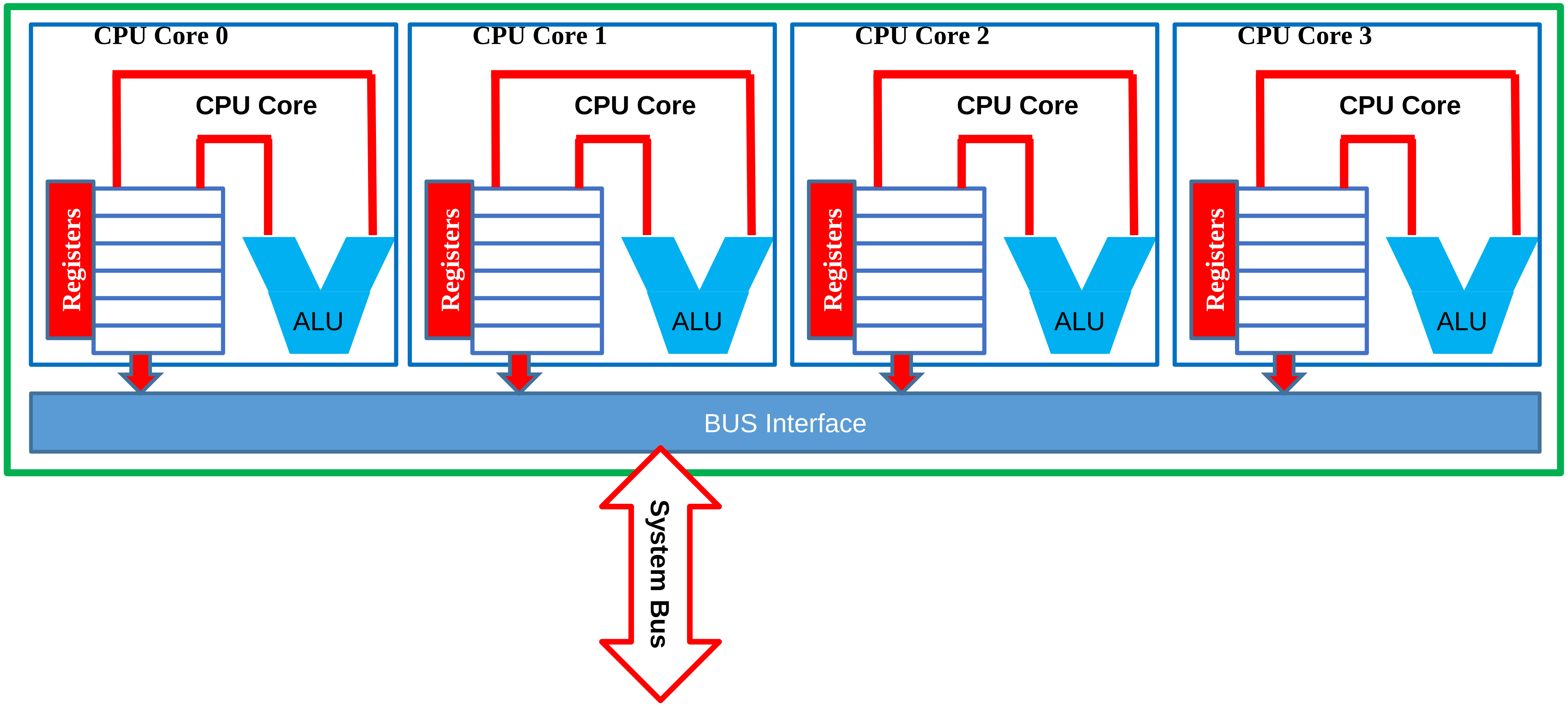


Core Computing Unit

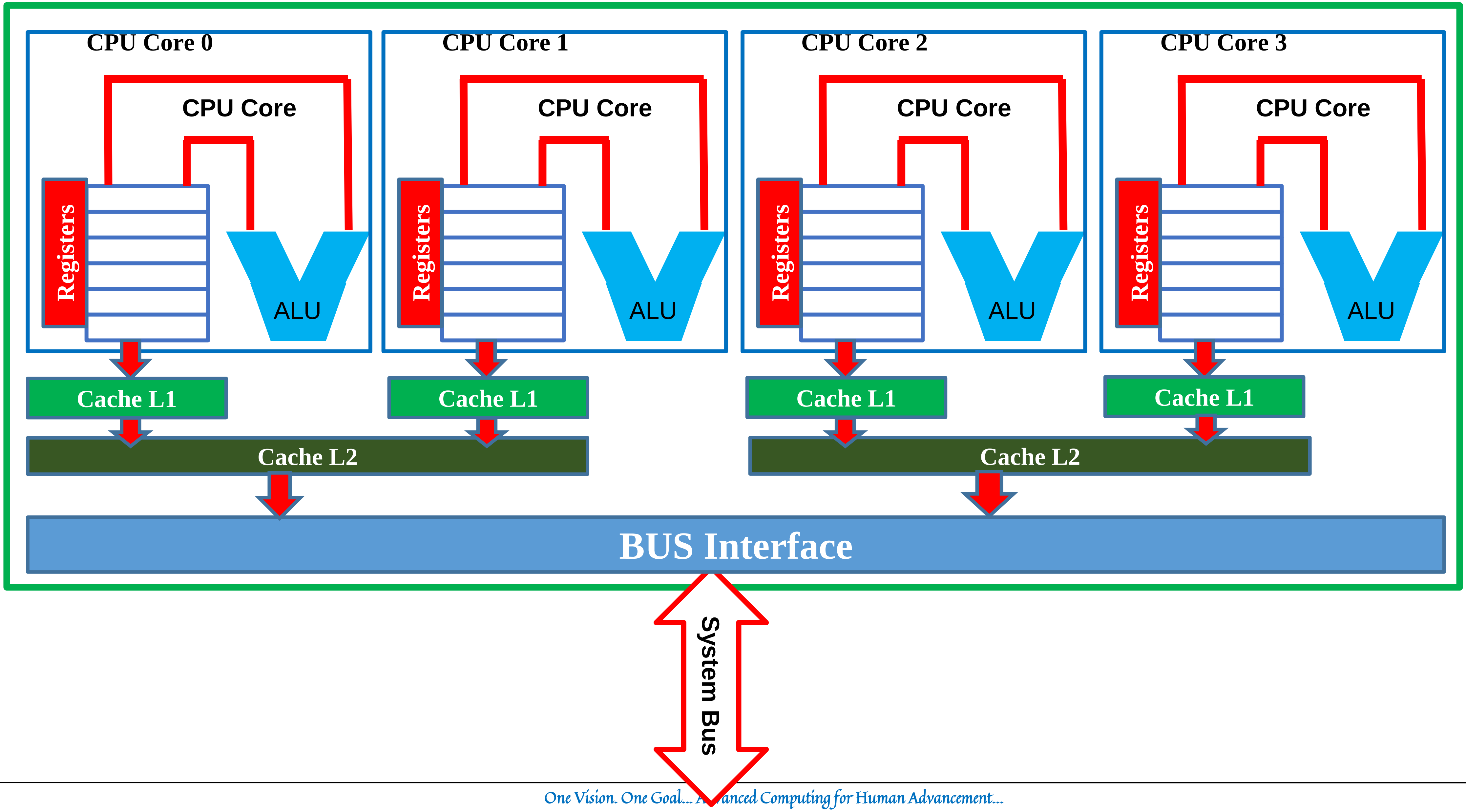
Core Computing Unit



Multi Core Processor

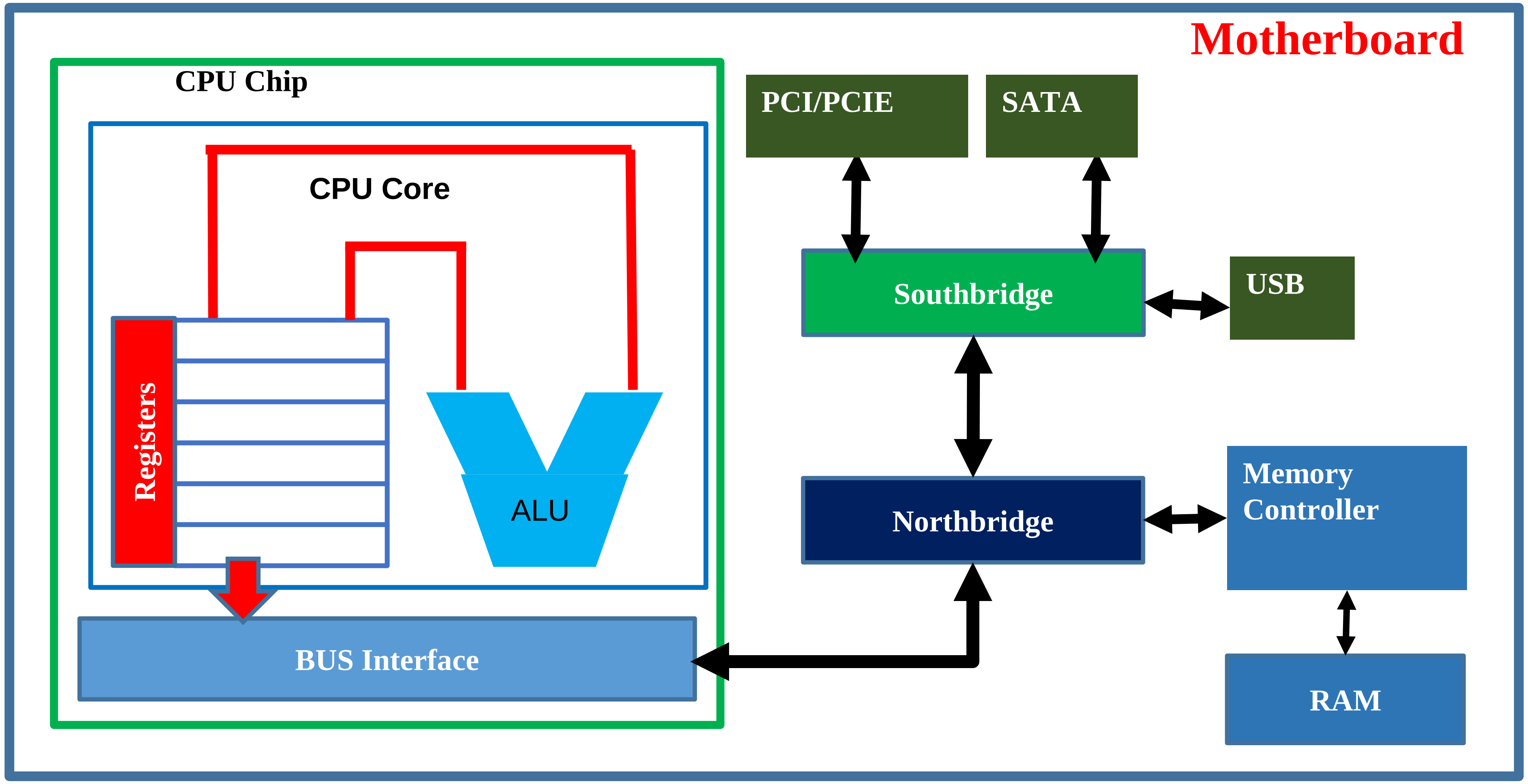


Multi Core Cluster Processor

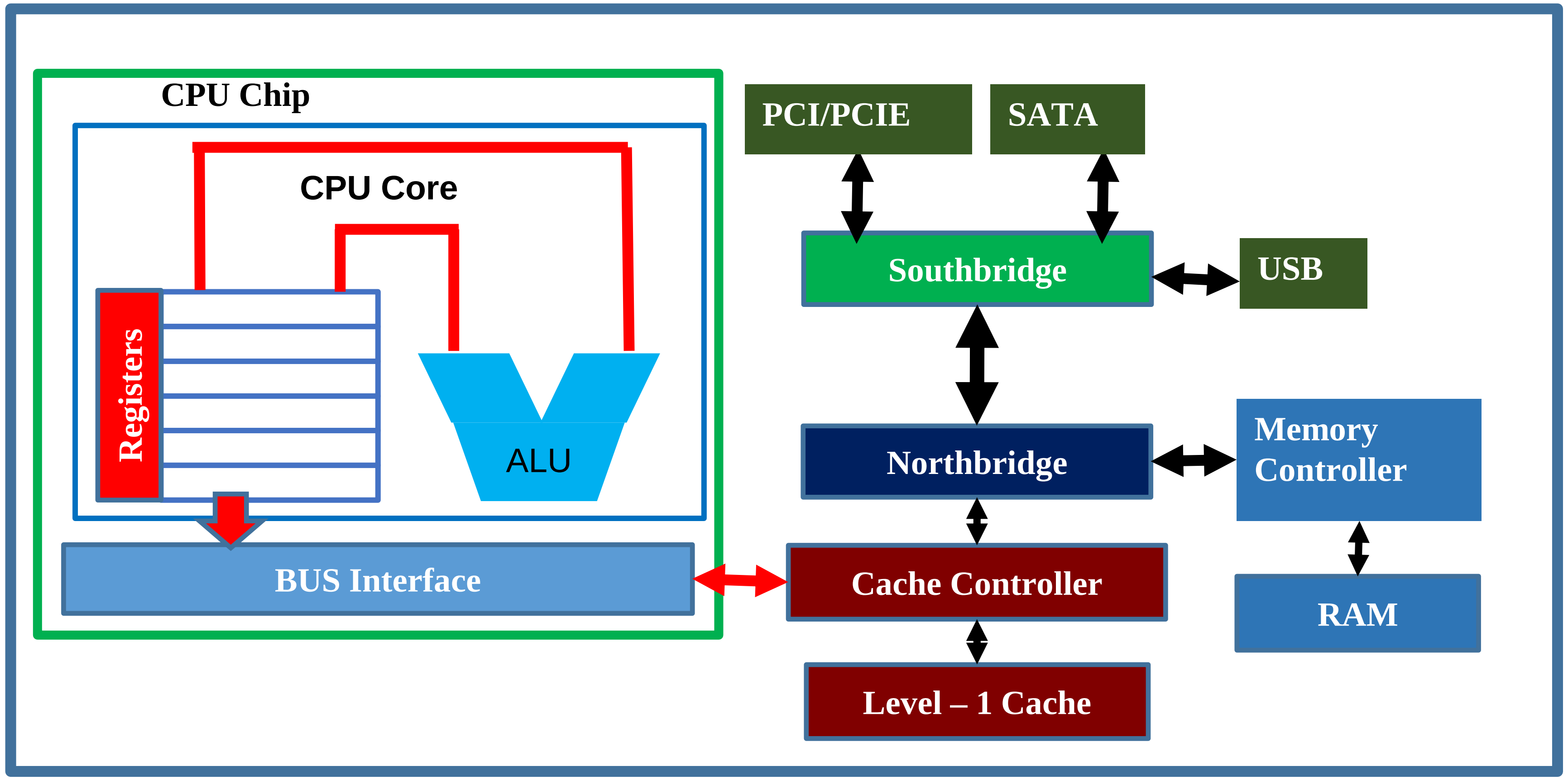


Evolution of Motherboard

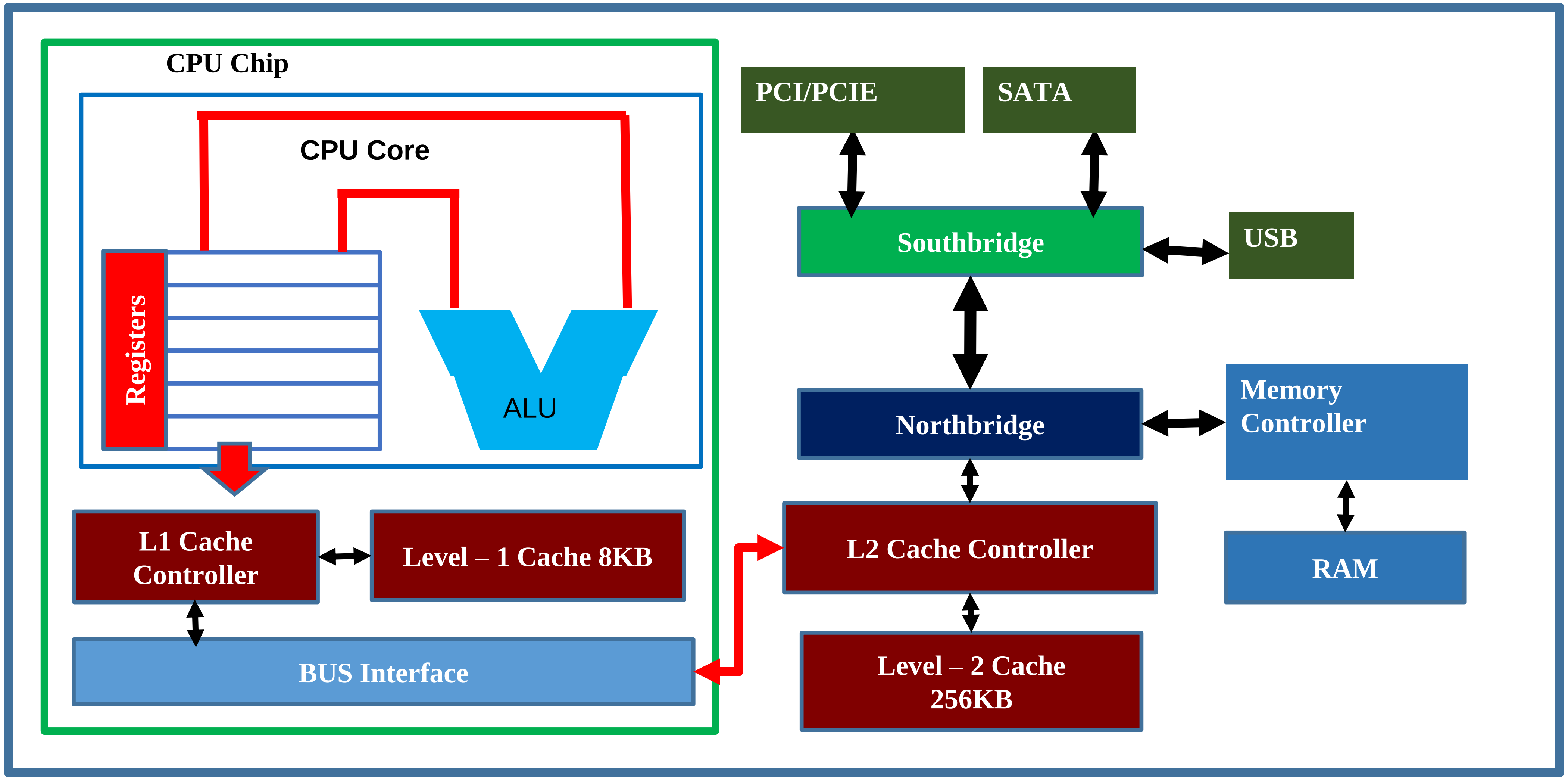
Motherboard Architecture



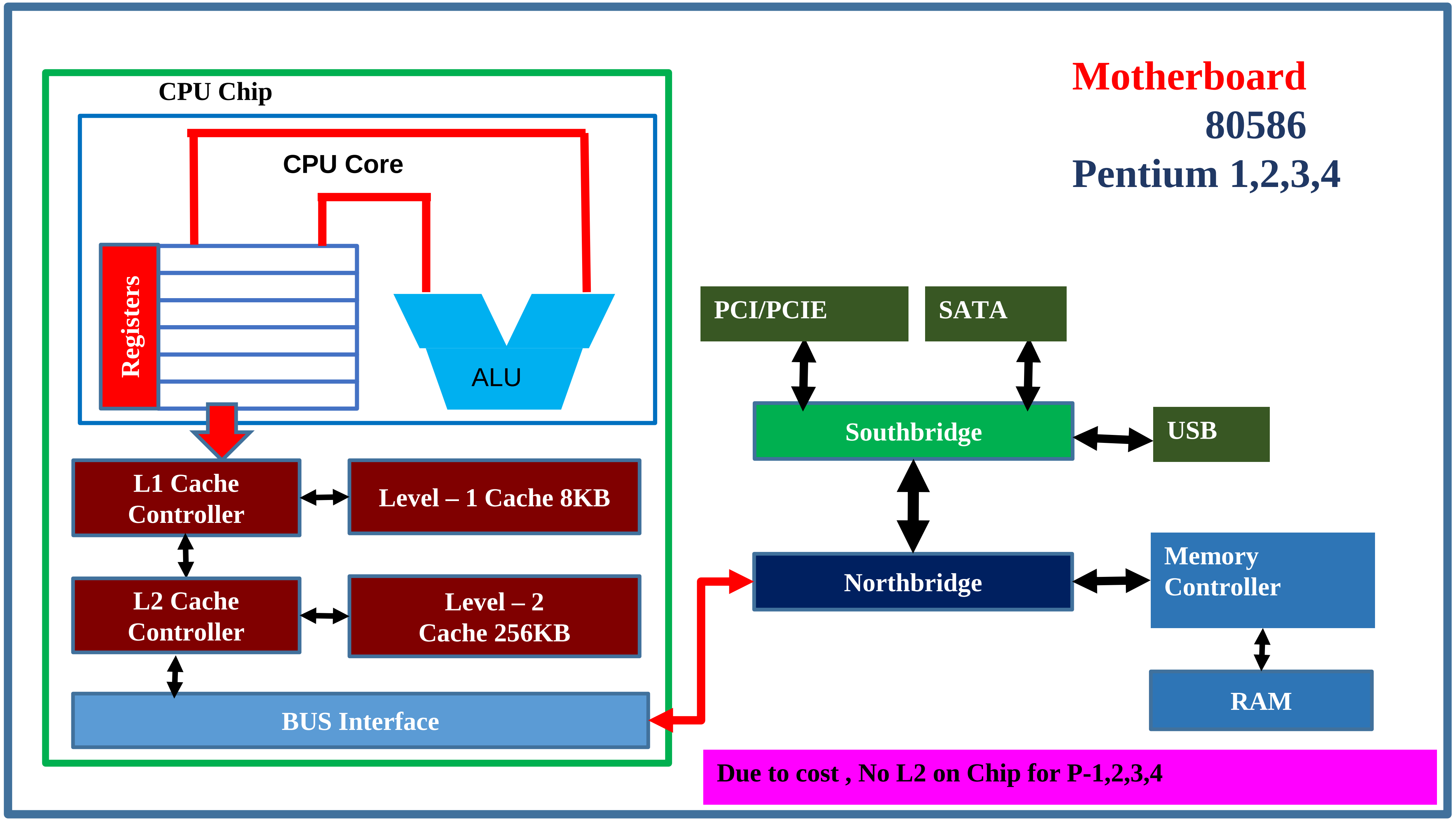
L1 - Cache Memory



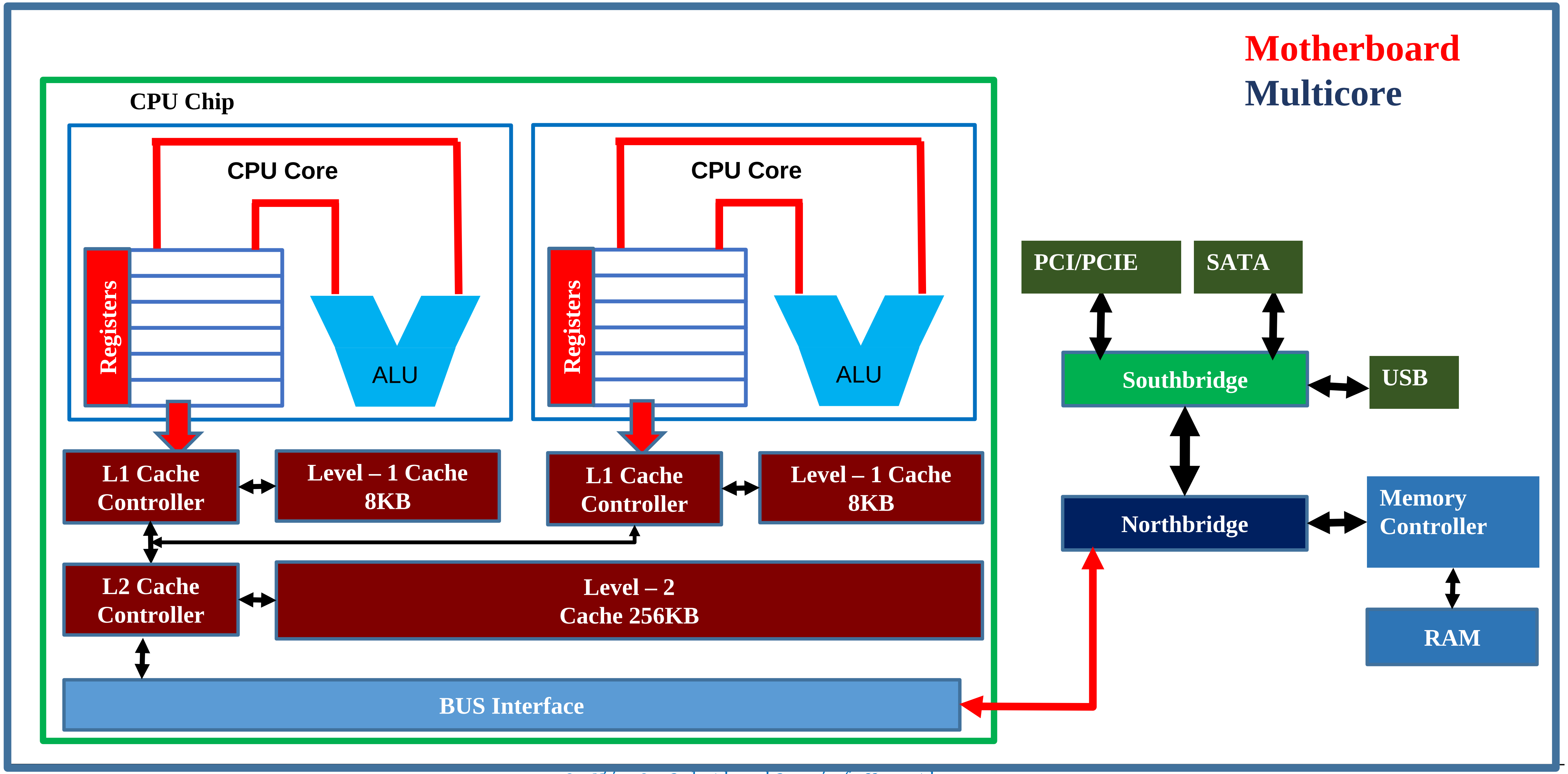
L2 - Cache Memory



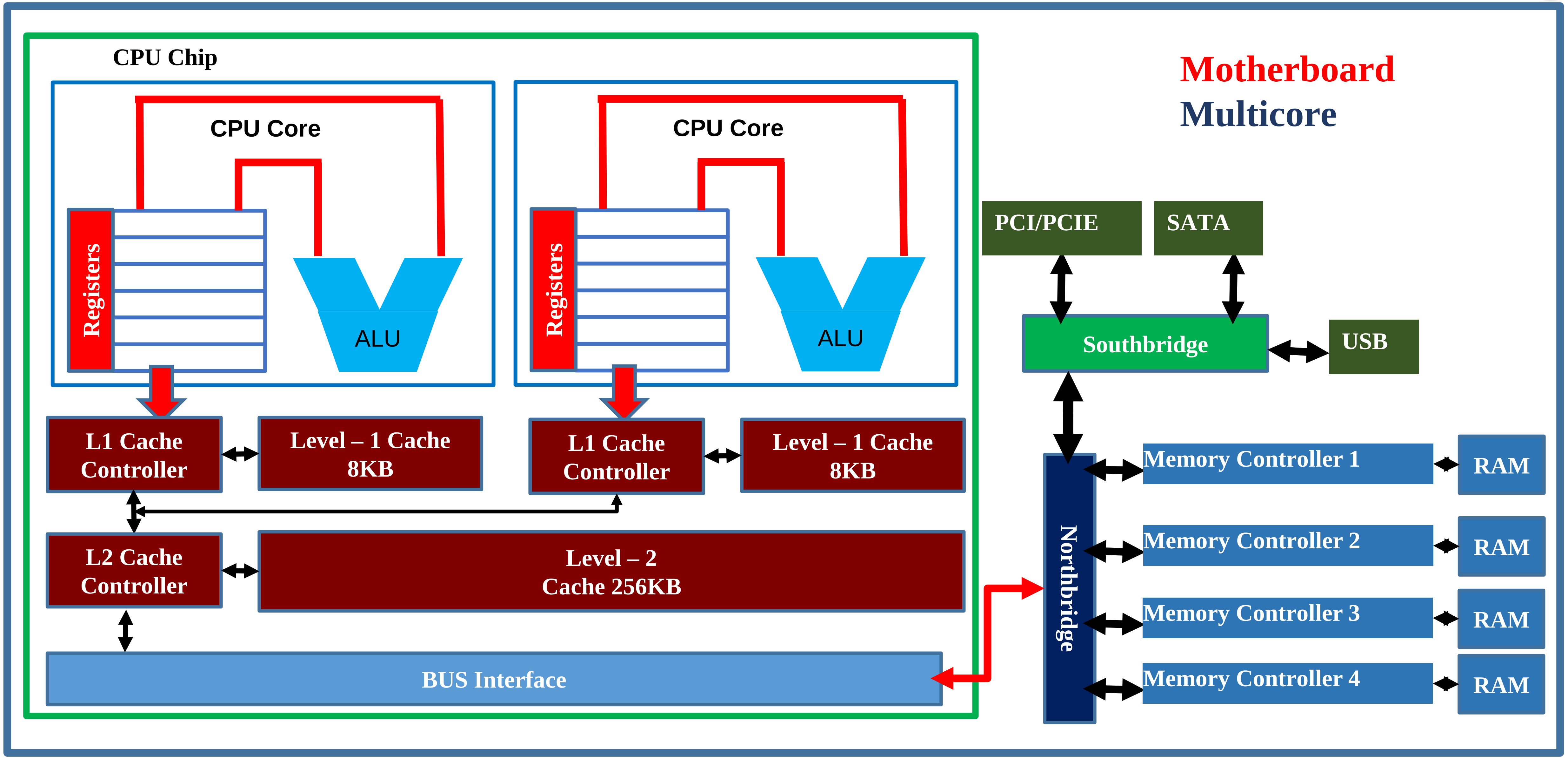
On Chip L1 & L2 - Cache Memory



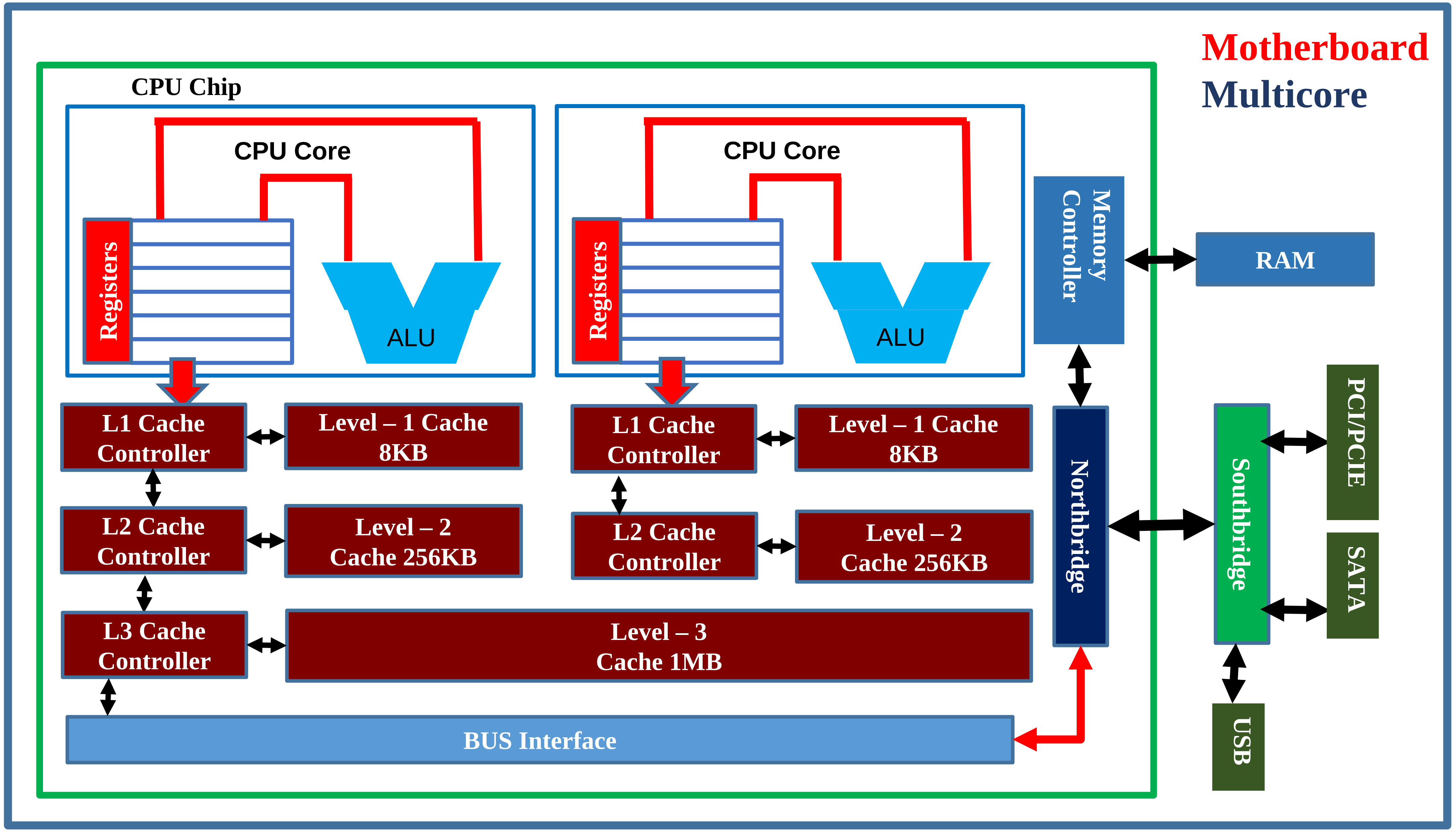
Multi-Core with On Chip L1 & L2 Cache Memory



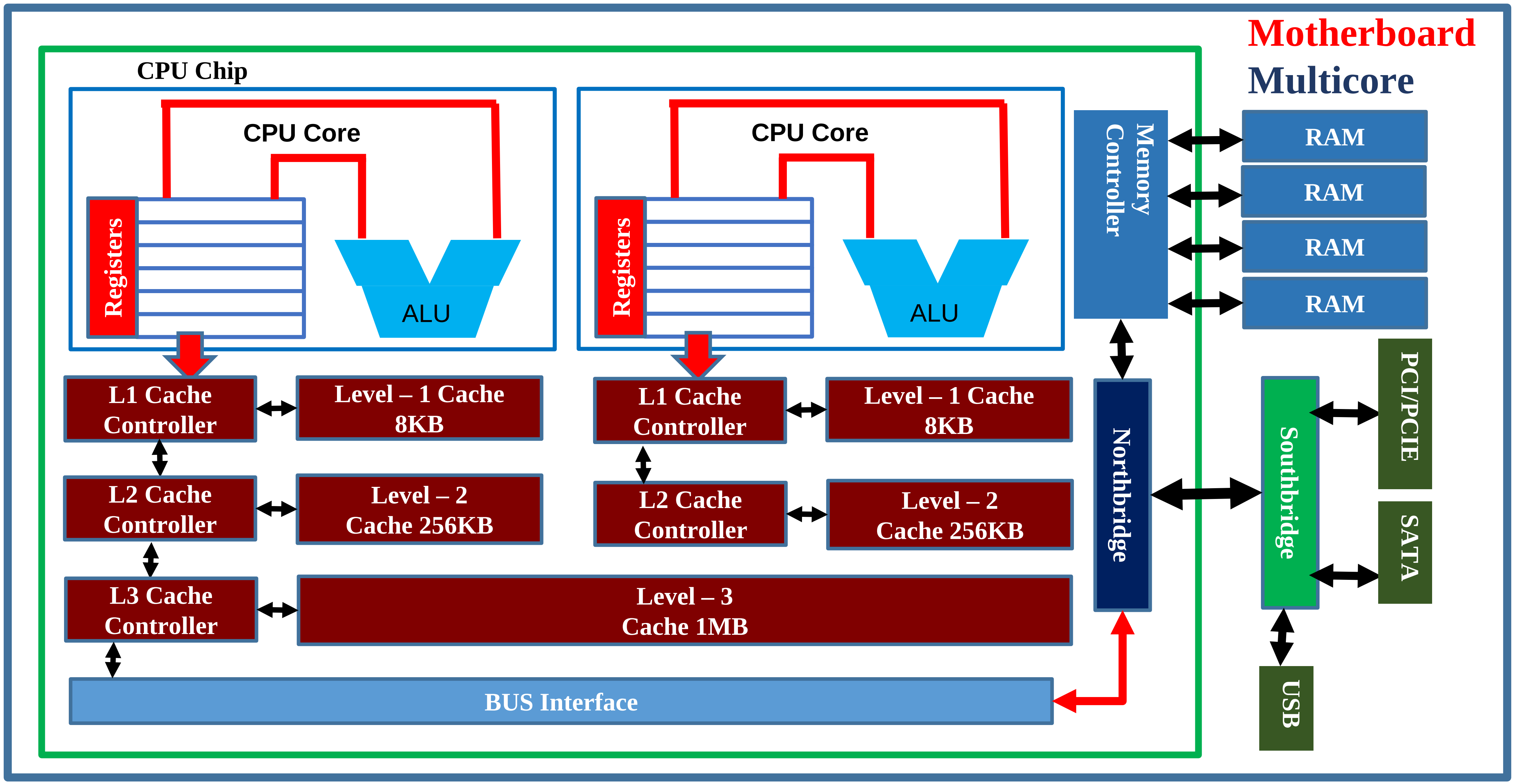
Multiple Memory Controller



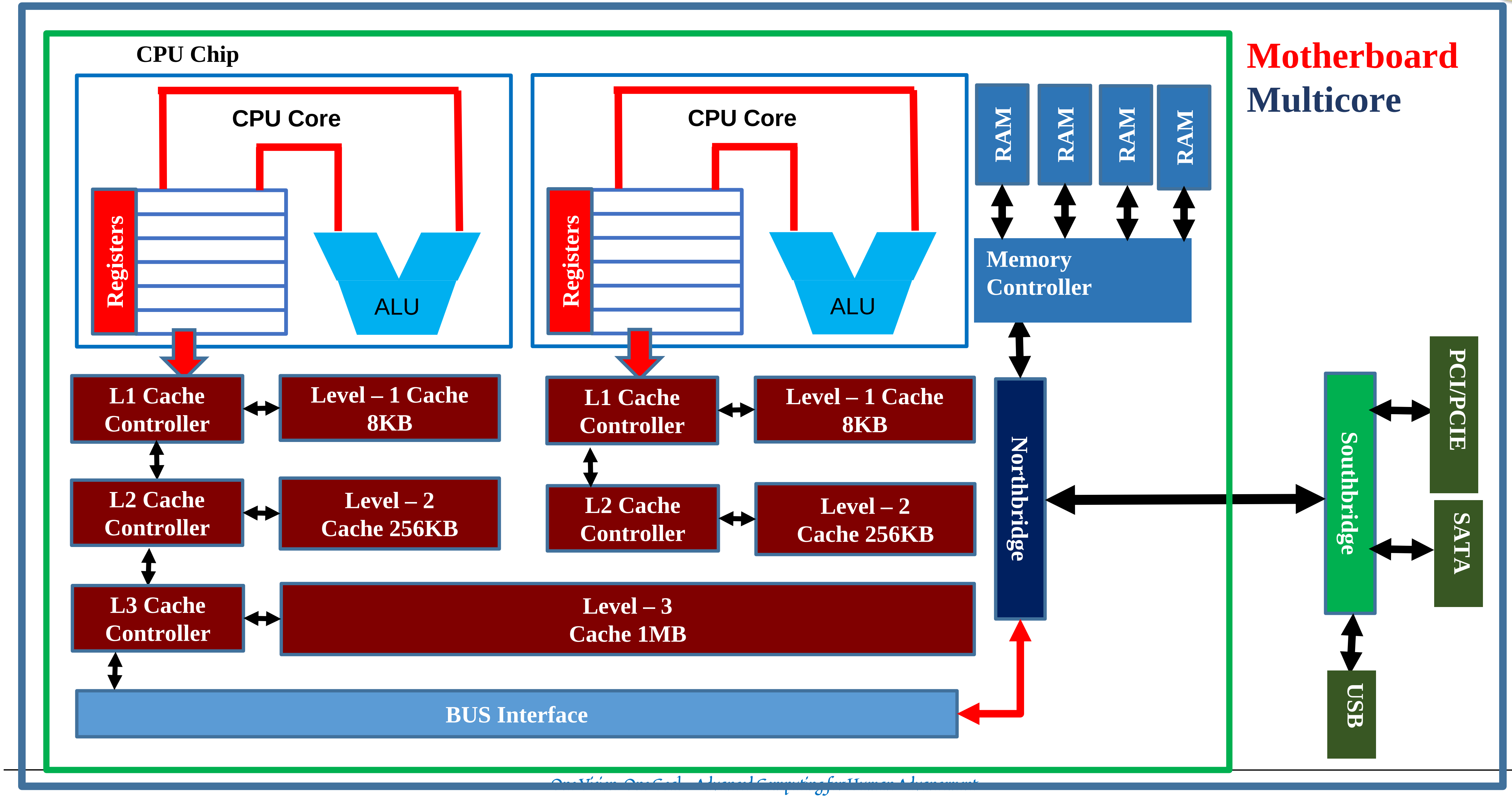
On Chip Memory Controller



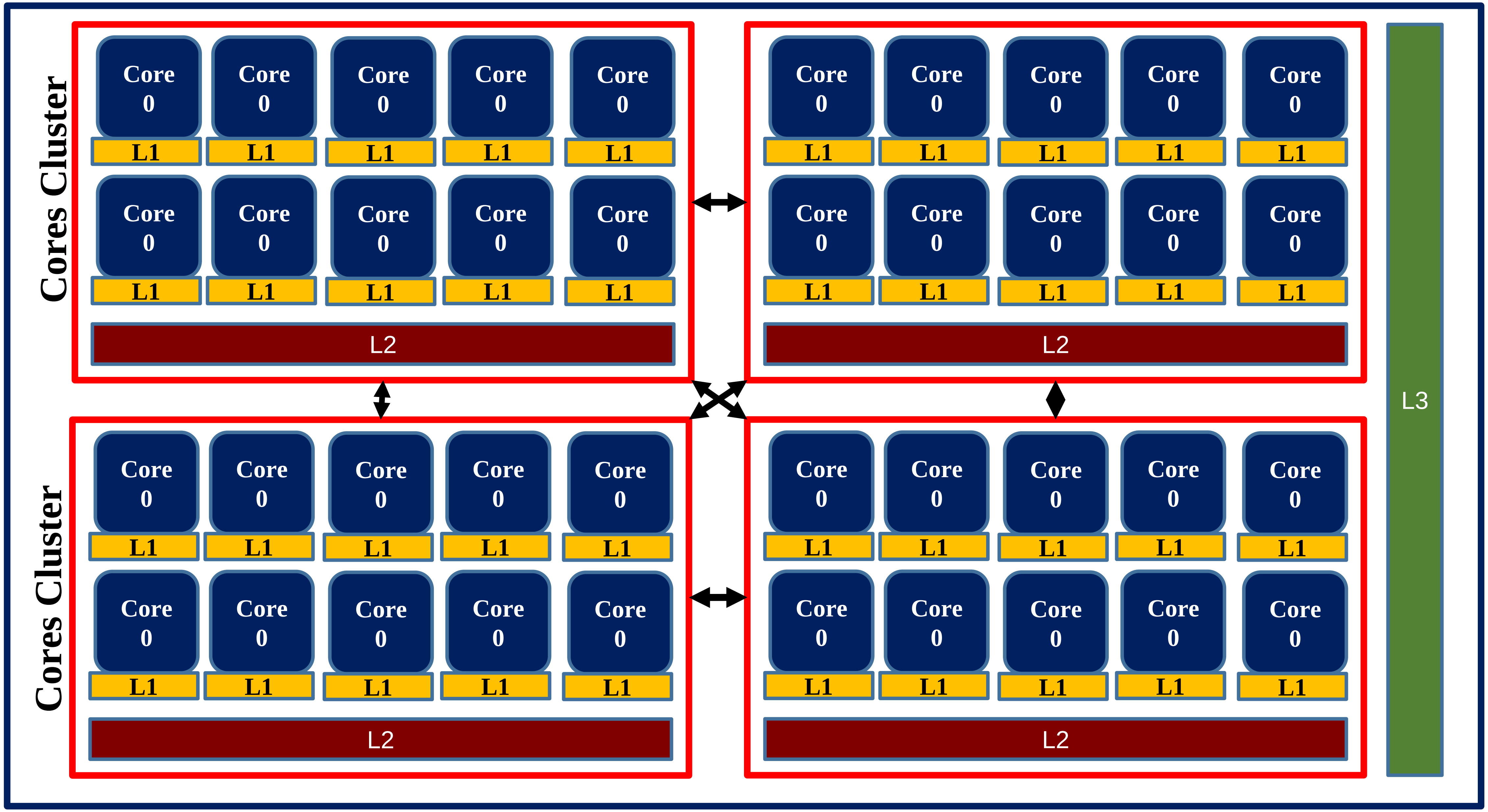
Multiple On Chip Memory Controller



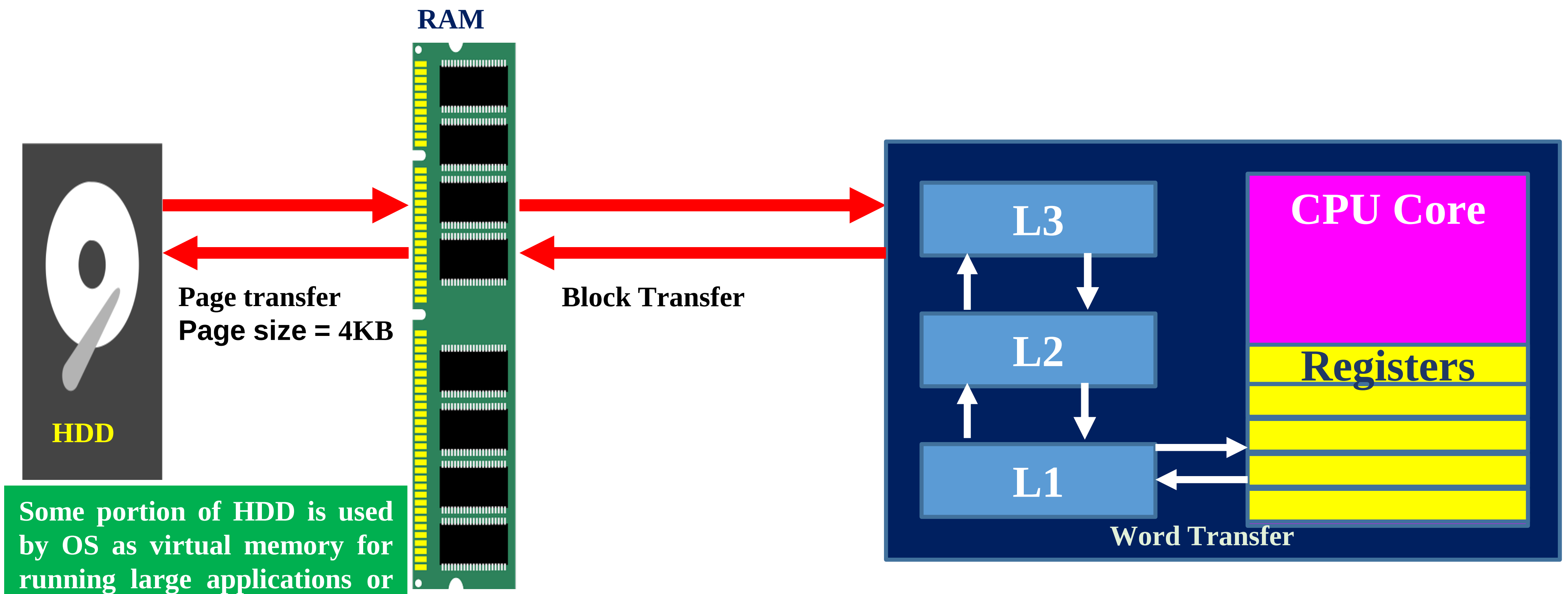
On Chip Main Memory



High Computing Processor Architecture



Data Transfer Generalized Method

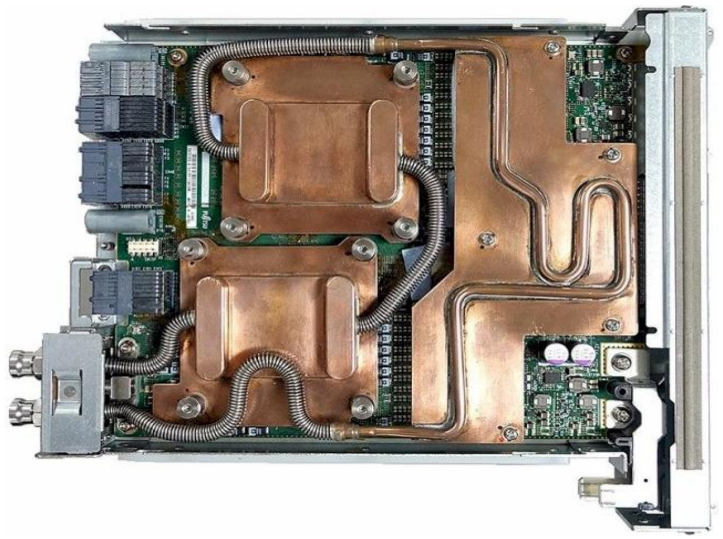


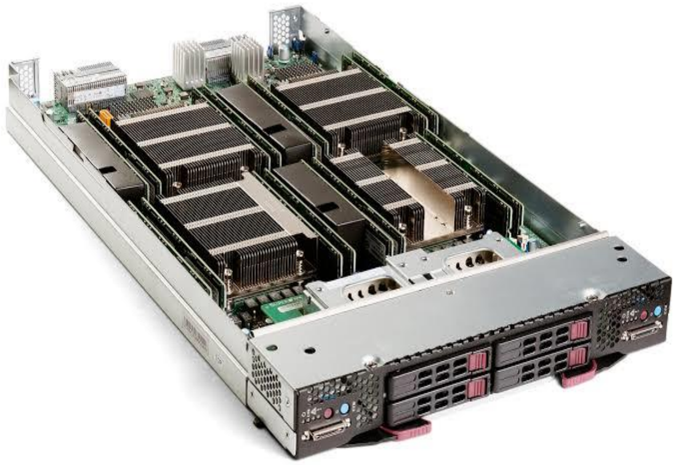
Some portion of HDD is used by OS as virtual memory for running large applications or number of applications.

Page size = 4KB, Depends on OS

Block Size = Cache Line Size = Cache Block

Physical View of HPC Elements


ARM FX Compute Node


Intel Compute Blade

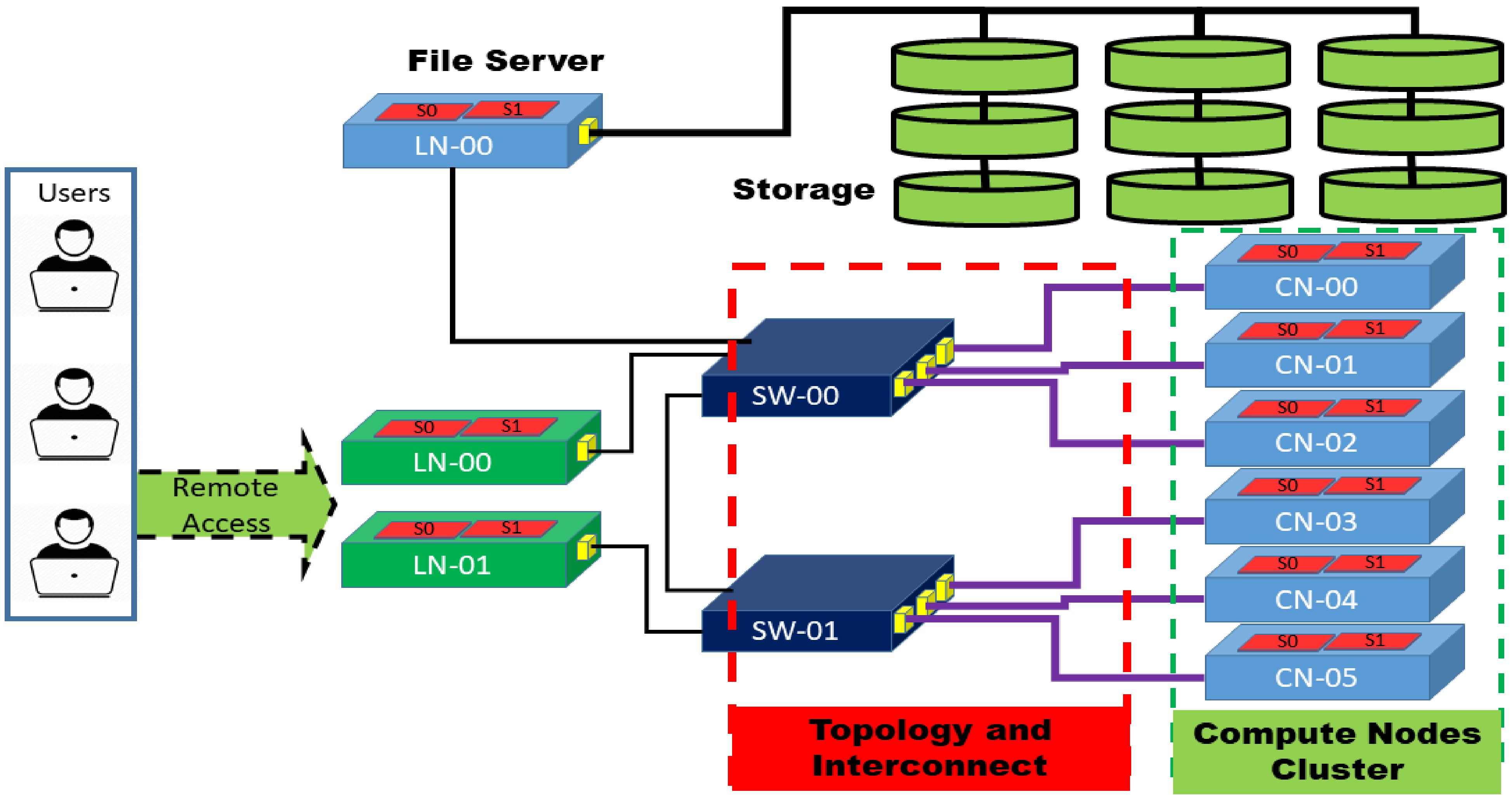
Compute Rack

Nvidia Co-processor

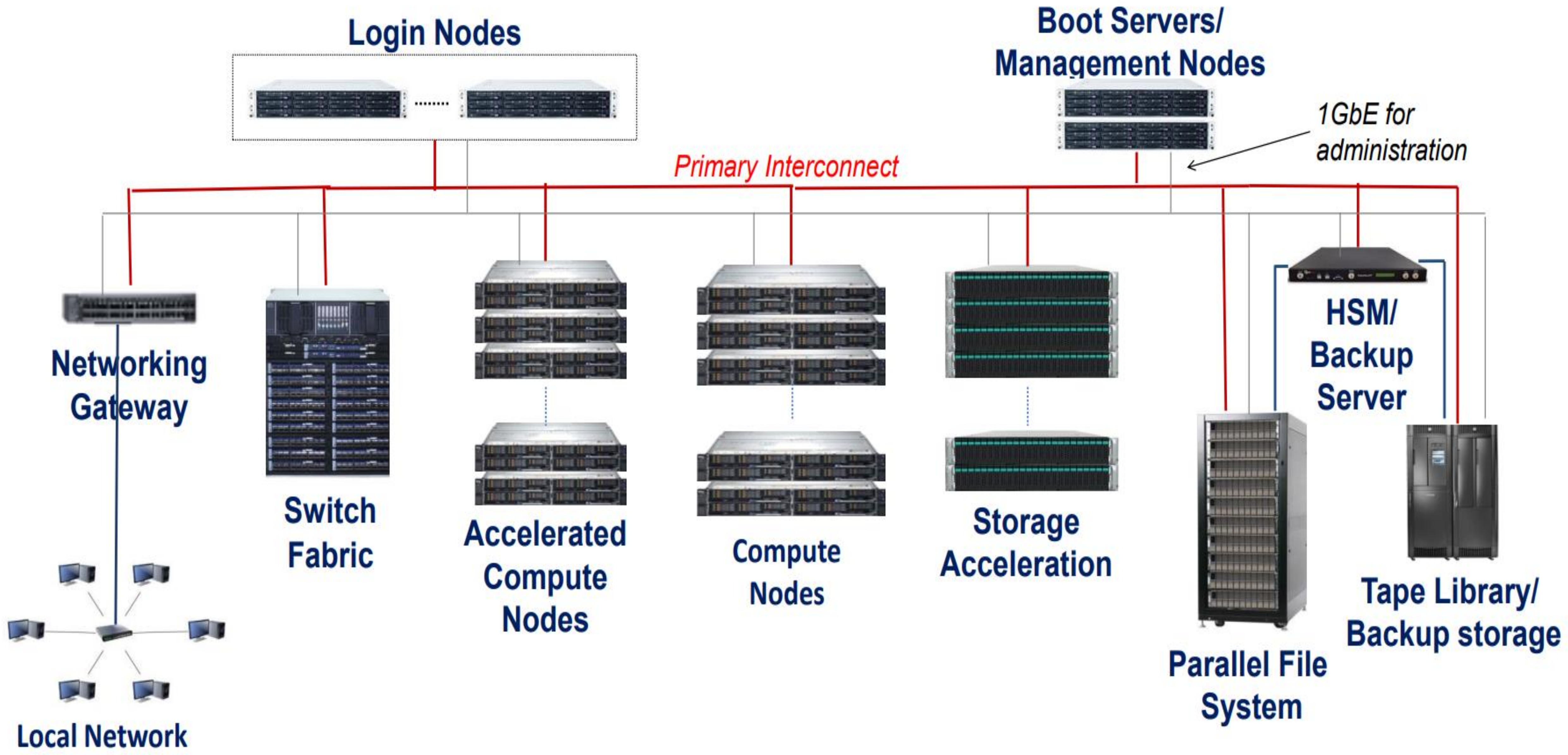
Intel Co-processor

AMD Co-processor

Simplified View of HPC Architecture

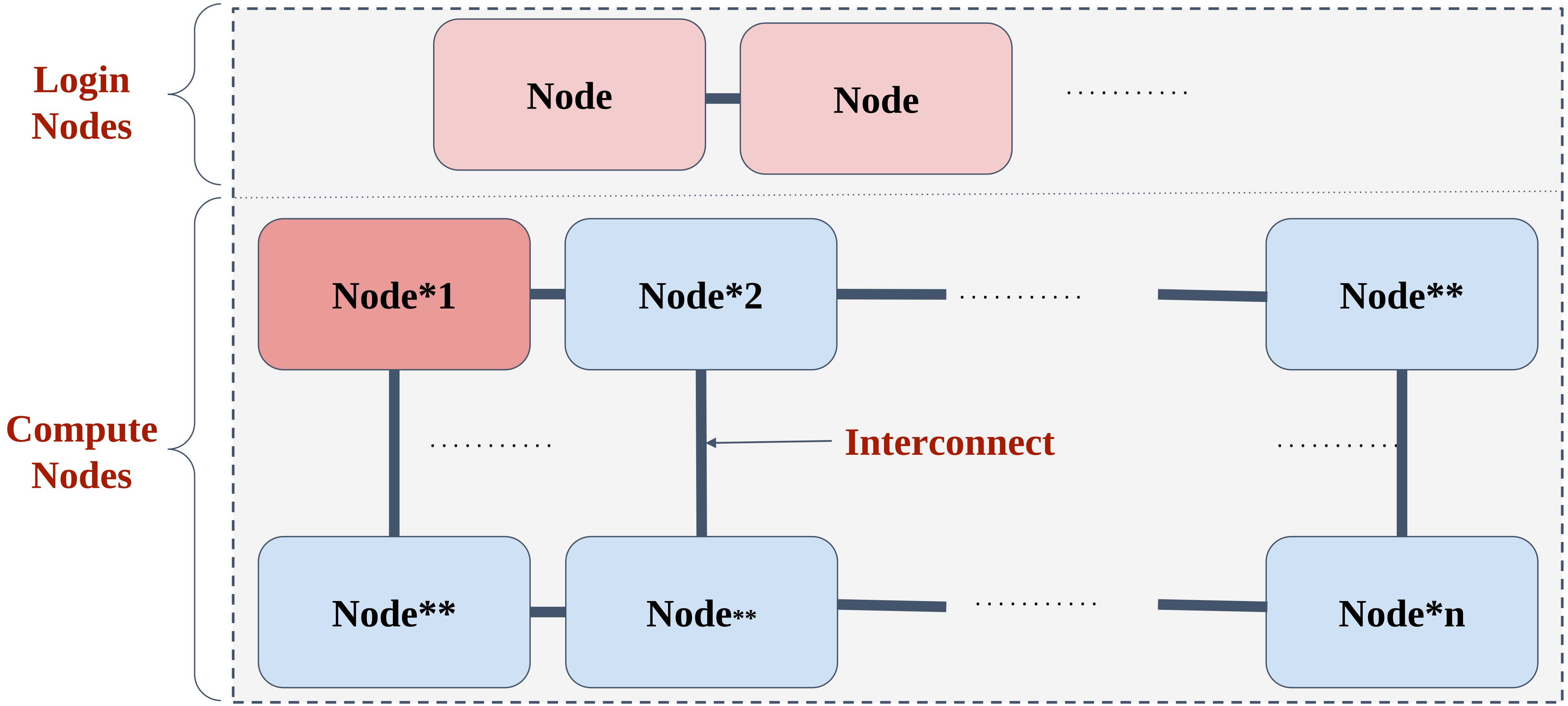


Advanced View of HPC Architecture

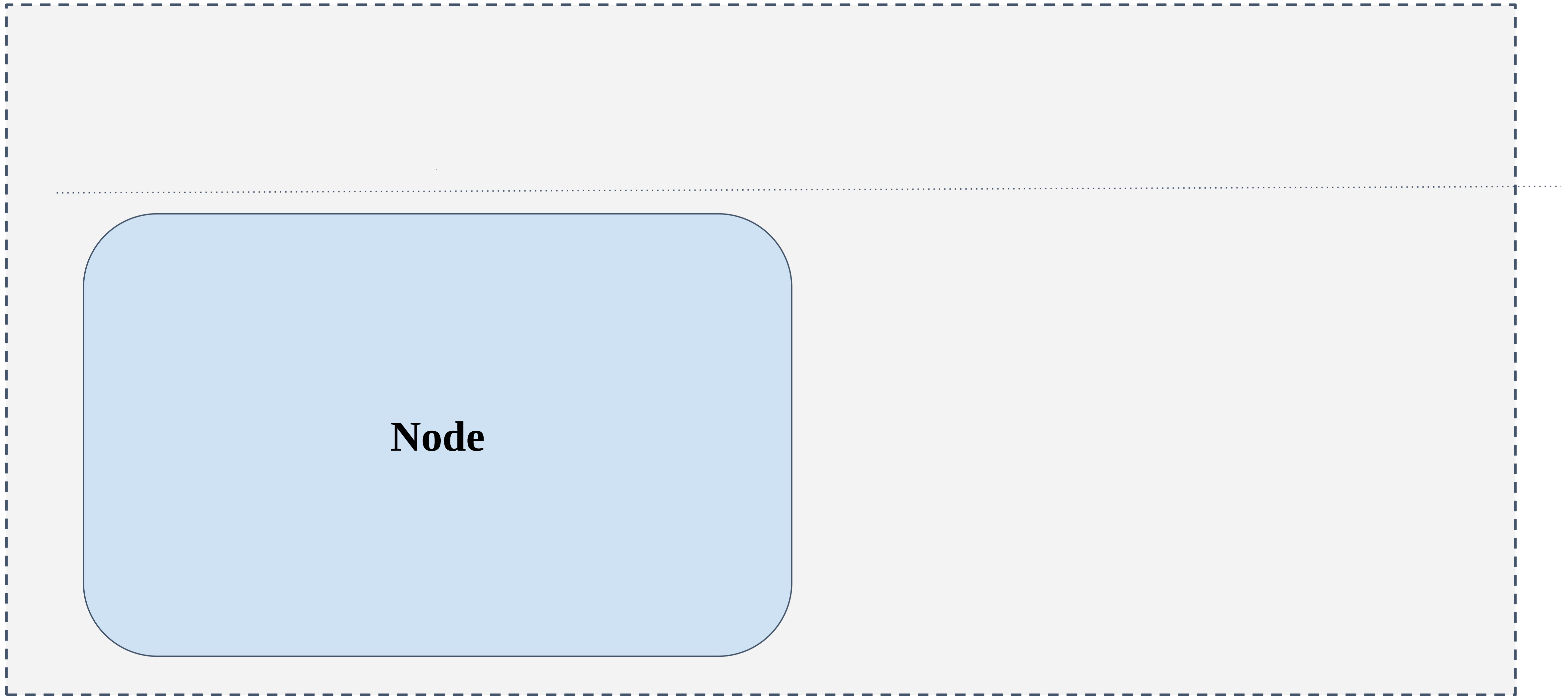


Typical HPC System Architecture

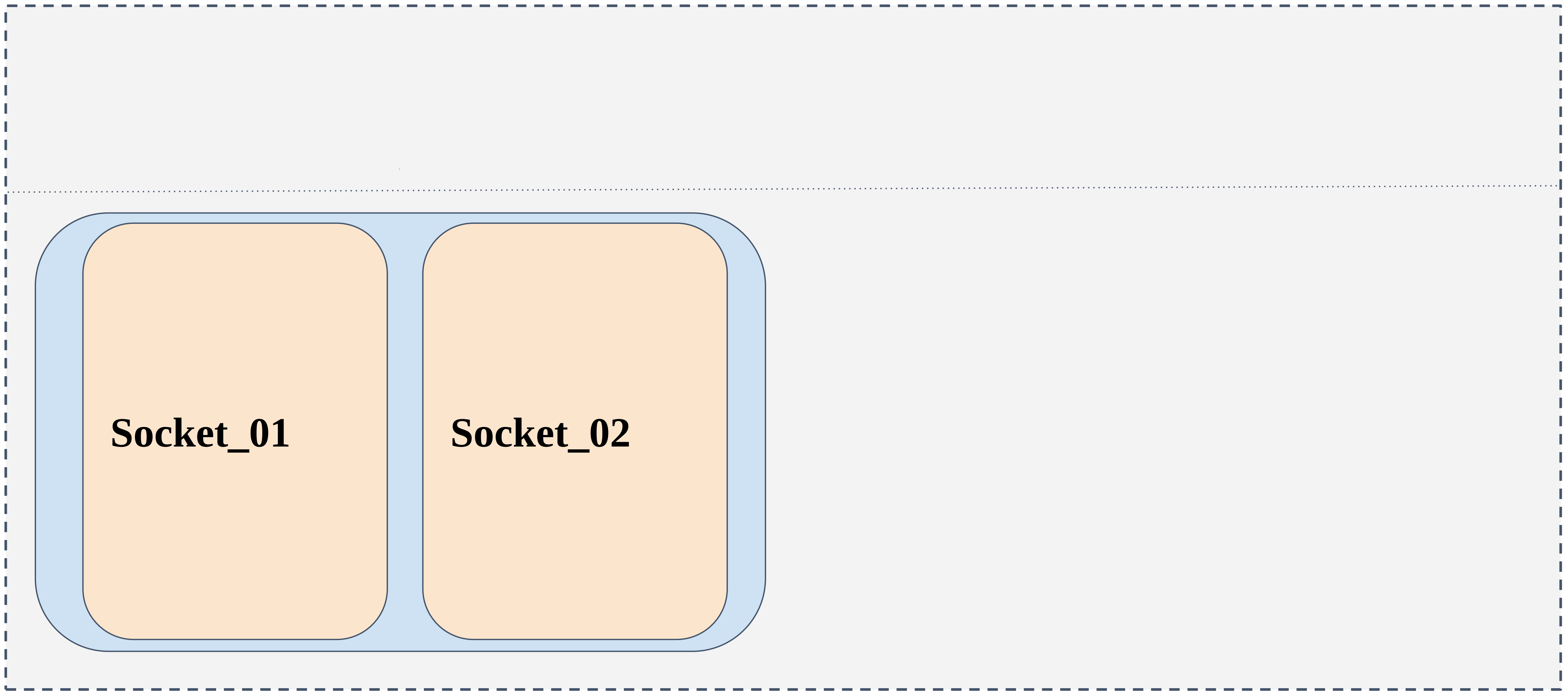
Typical HPC System



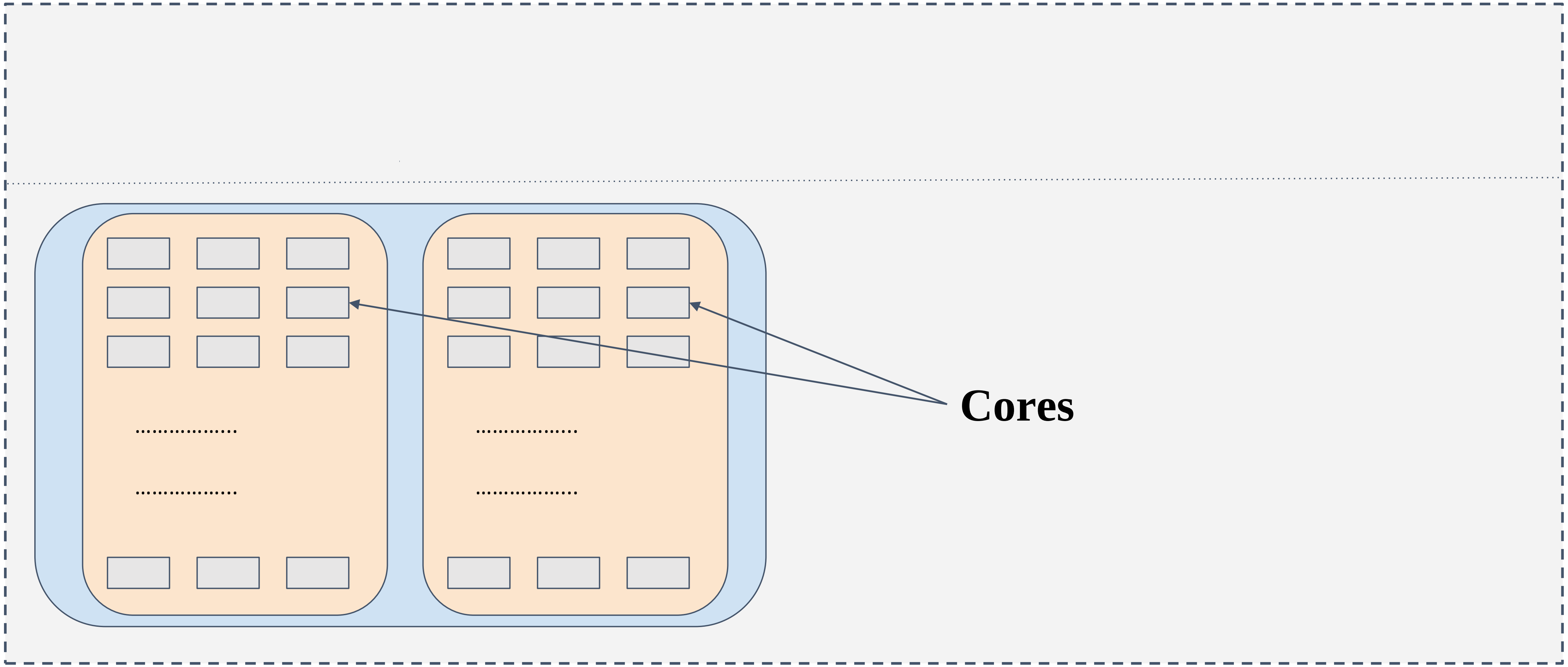
Simplified View of HPC Architecture



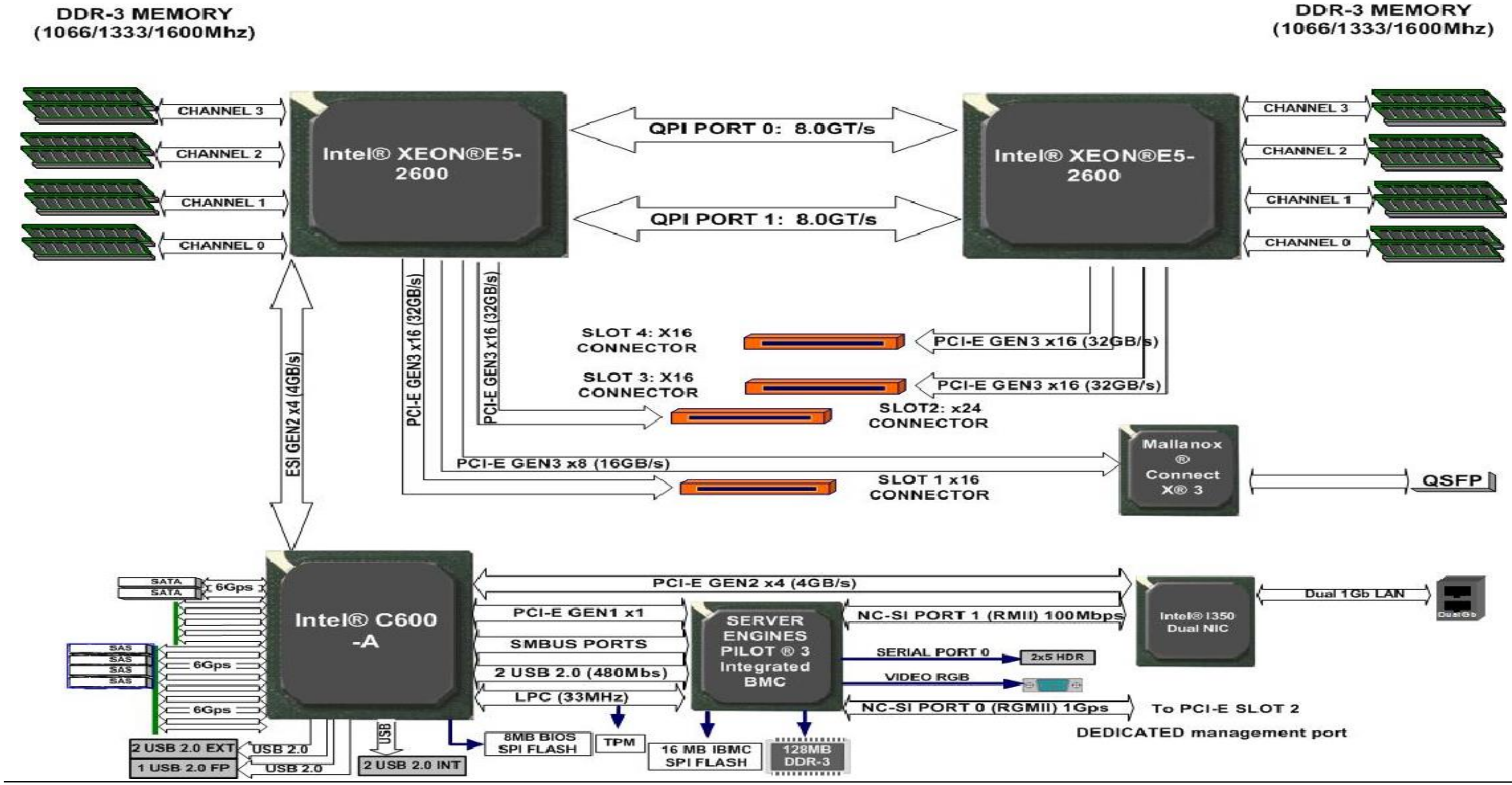
Simplified View of HPC Architecture



Simplified View of HPC Architecture



Detailed View of Compute Node Architecture



Instruction vs Micro-operation (μOps)

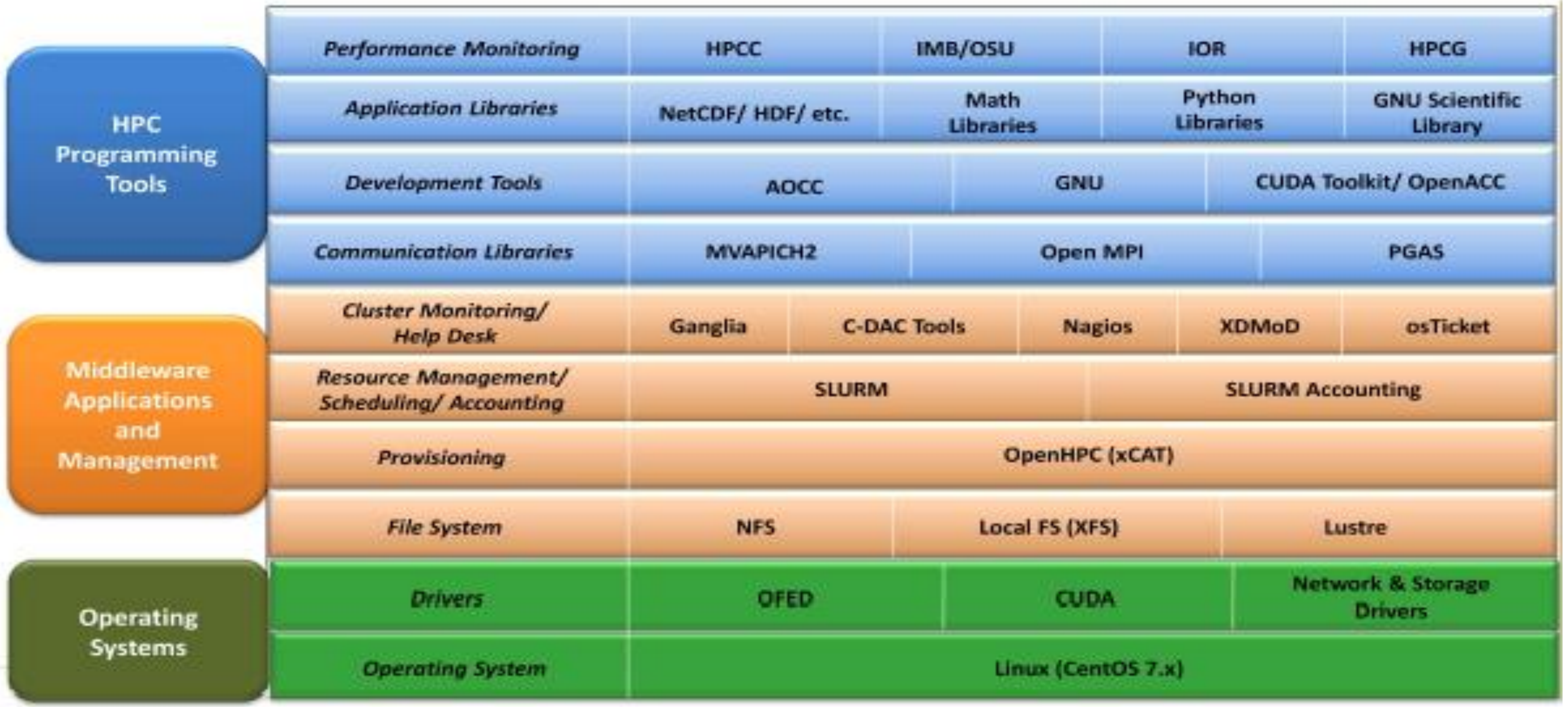
Instruction	μ Ops	Explanation
ADD EAX , EBX	1	EAX ←EAX + EBX

ADD EAX , [MEM]	2	Reg ← [MEM]
		EAX ← Reg + EAX

ADD [MEM] , EAX	3	Reg ← [MEM]
		Reg ← Reg + [MEM]
		[MEM] ← Reg

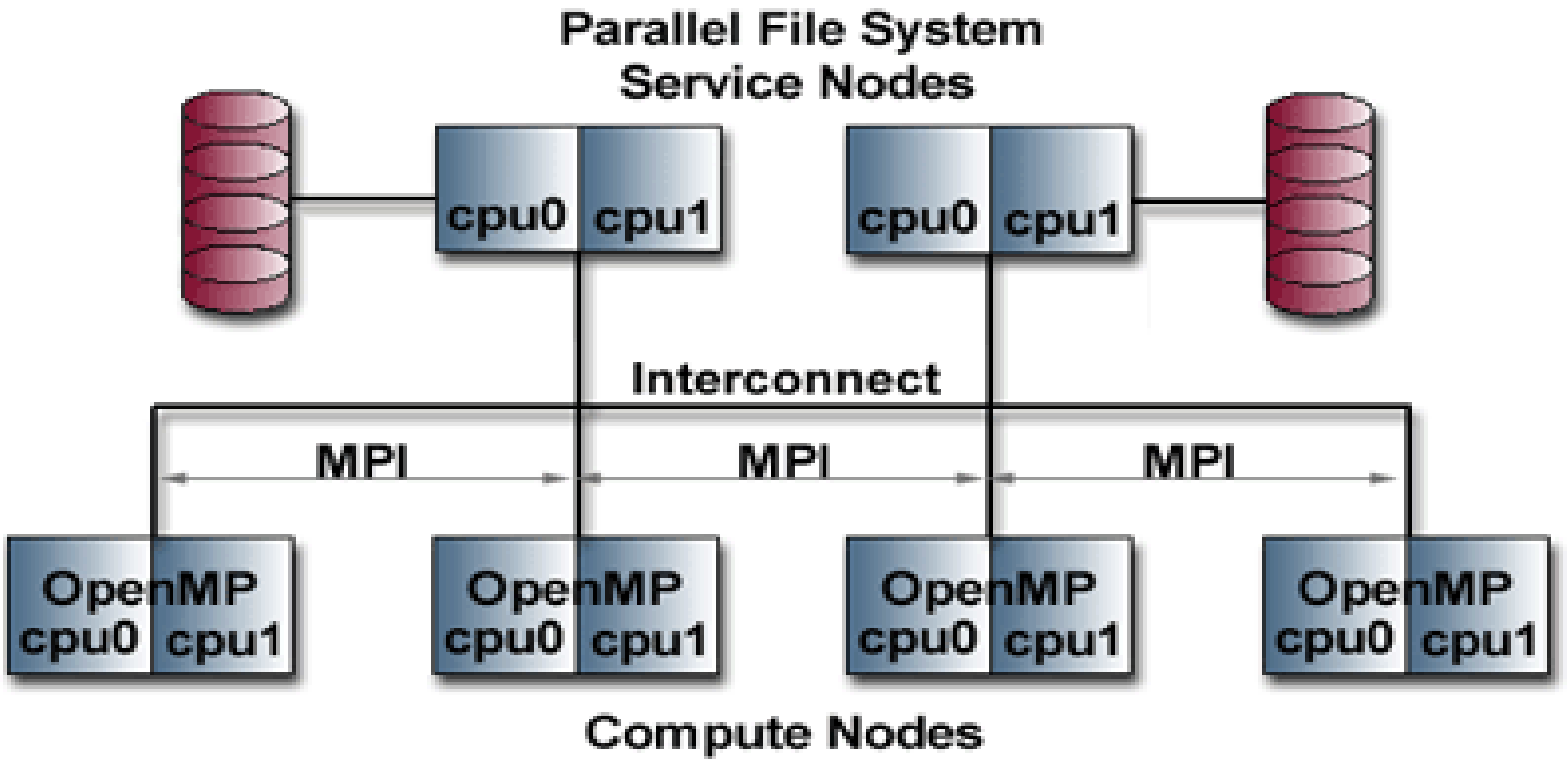
Note: μOps can be executed out-of-order

HPC Software Stack



HPC Programming Tools	*Performance Monitoring*	HPCC	IMB/OSU	IOR	HPCG	
	Application Libraries	NetCDF/ HDF/ etc.	Math Libraries	Python Libraries	GNU Scientific Library	
	Development Tools	AOCC	GNU	CUDA Toolkit/ OpenACC		
	Communication Libraries	MVAPICH2	Open MPI	PGAS		
Middleware Applications and Management	*Cluster Monitoring/ Help Desk*	Ganglia	C-DAC Tools	Nagios	XDMoD	osTicket
	Resource Management/ Scheduling/ Accounting	SLURM	SLURM Accounting			
	Provisioning	OpenHPC (xCAT)				
	File System	NFS	Local FS (XFS)	Lustre		
Operating Systems	*Drivers*	OFED	CUDA	Network & Storage Drivers		
	Operating System	Linux (CentOS 7.x)				

Parallel File System

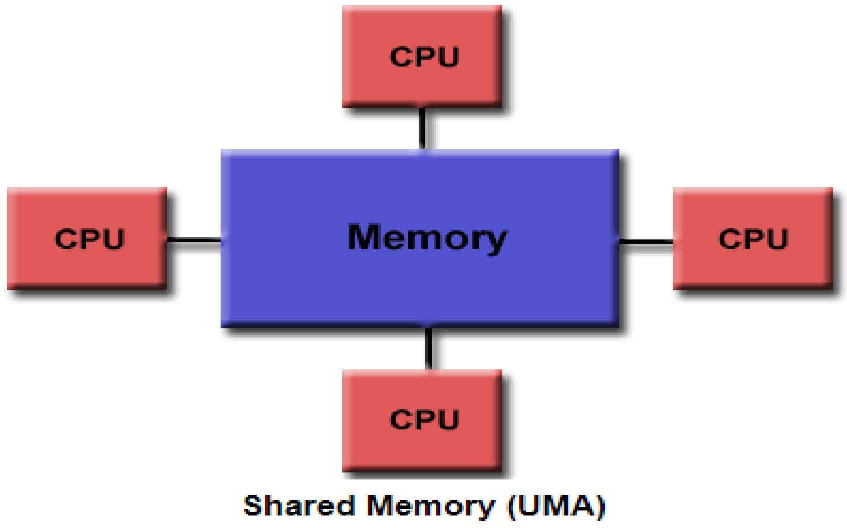


Parallel Programming Models

Parallel Programming Models

Shared Memory Model

- OpenMP

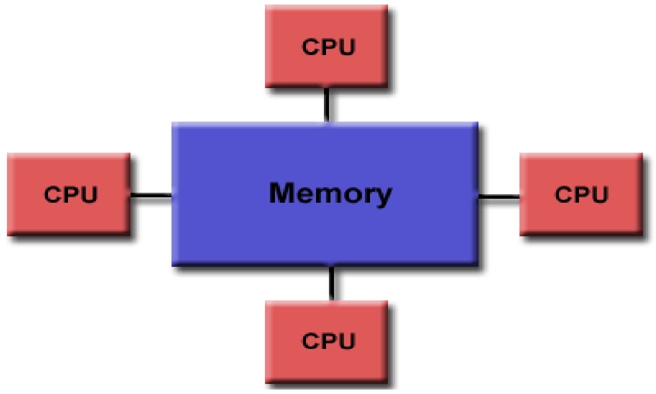


Shared Memory (UMA)

Parallel Programming Models

Shared Memory Model

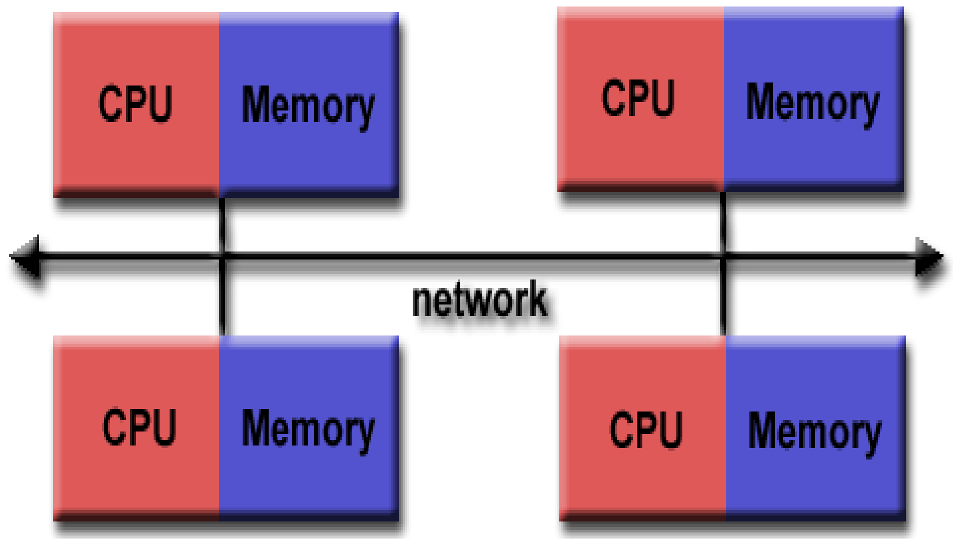
- OpenMP



Shared Memory (UMA)

Distributed Memory Model

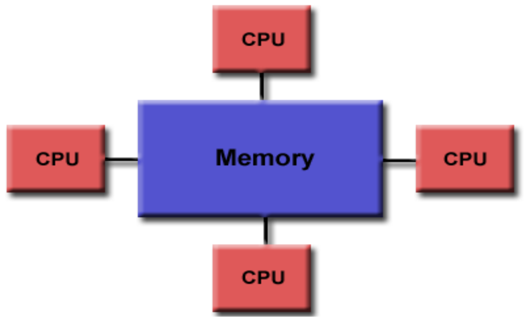
- Message Passing Interface (MPI)



Parallel Programming Models

Shared Memory Model

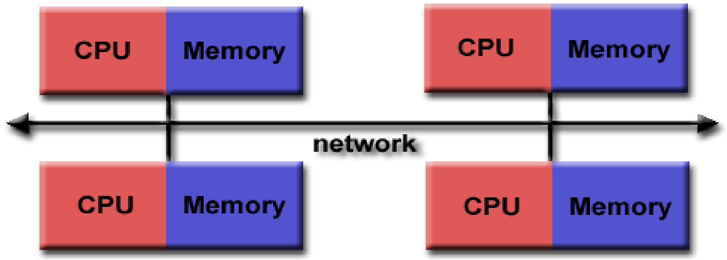
- OpenMP



Shared Memory (UMA)

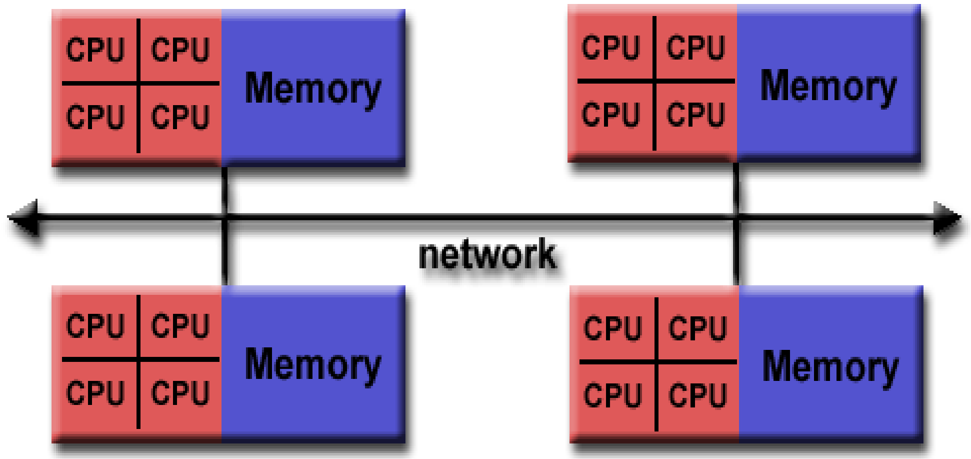
Distributed Memory Model

- Message Passing Interface (MPI)



Hybrid Memory Model

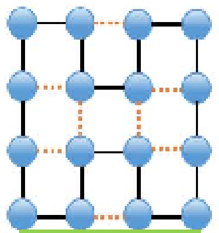
- OpenMP + MPI

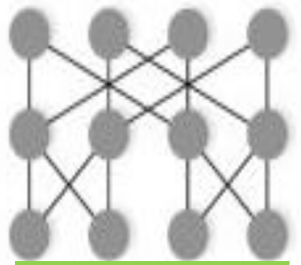


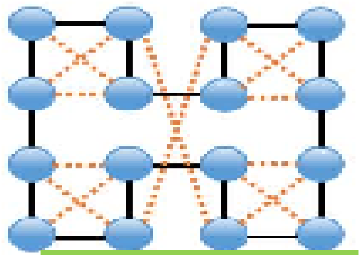
HPC Interconnection Network (Topology)

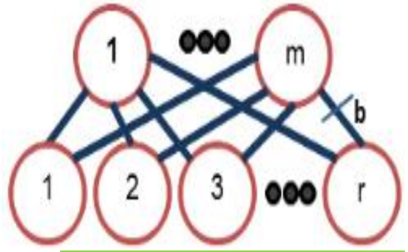
High-speed interconnection networks determine the performance of many communication-heavy HPC applications

HPC Interconnection Network (Topology)

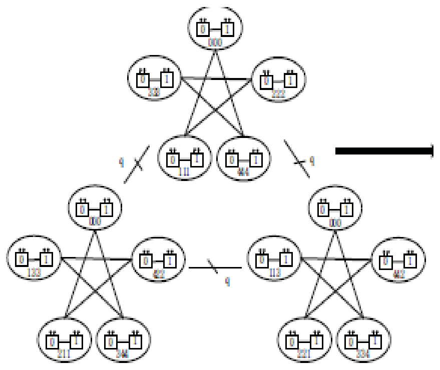

Mesh


Butterfly


Dragonfly



Dragonfly +


Galaxyfly

Spectralfly

1980

1990

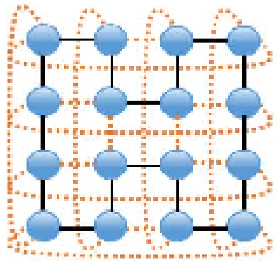
2000

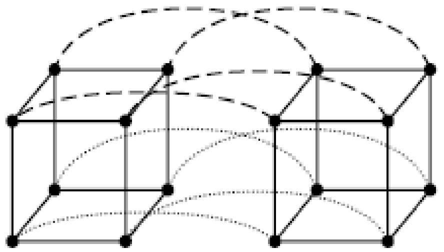
2010

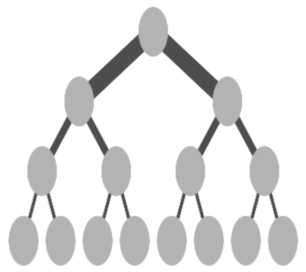
2020

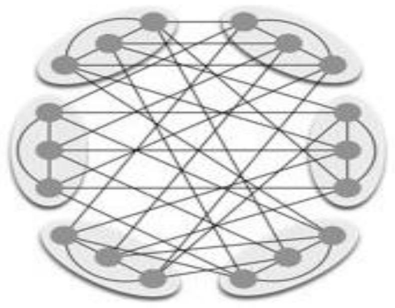
2021

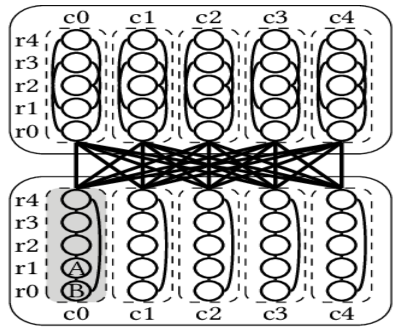
Software Defined Networking (SDN)


Torus


Hypercube

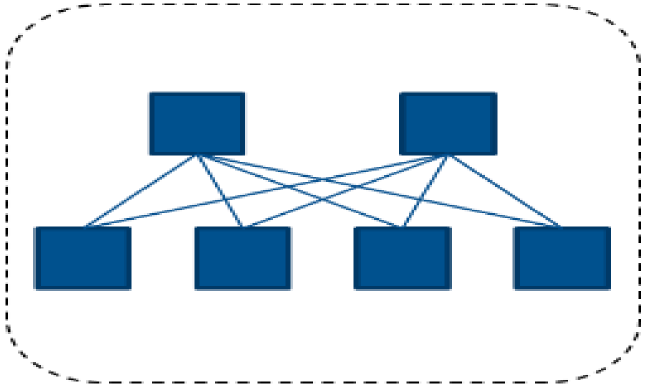

Fat-Tree

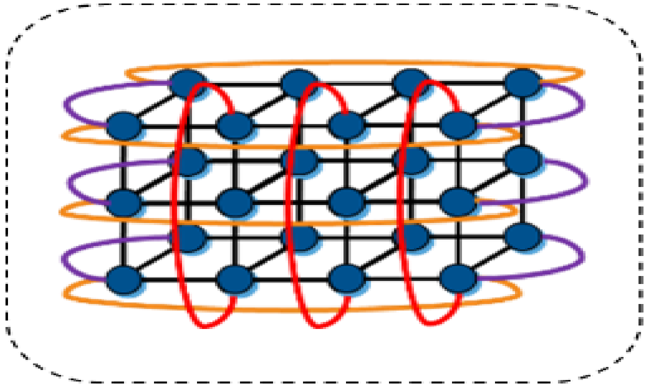

Slimfly

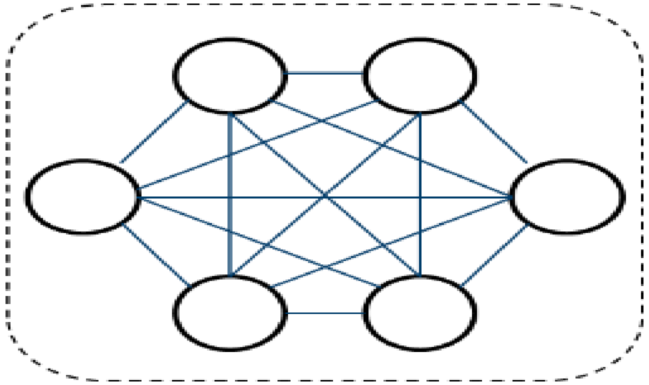


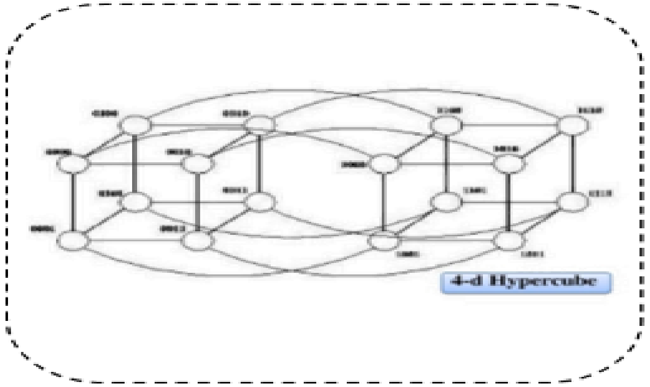
Bundlefly

Popular Topologies (Last Two decades)

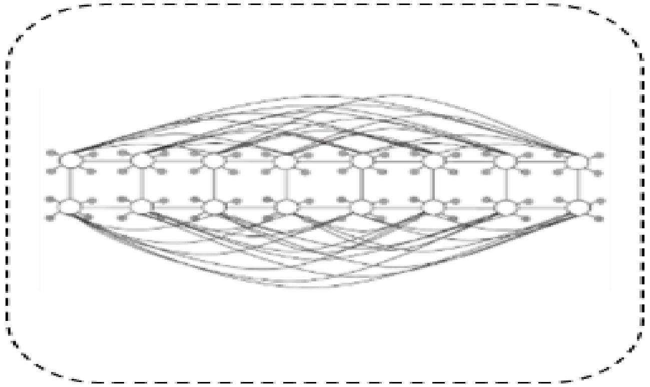

Fat Tree


Torus


Dragonfly



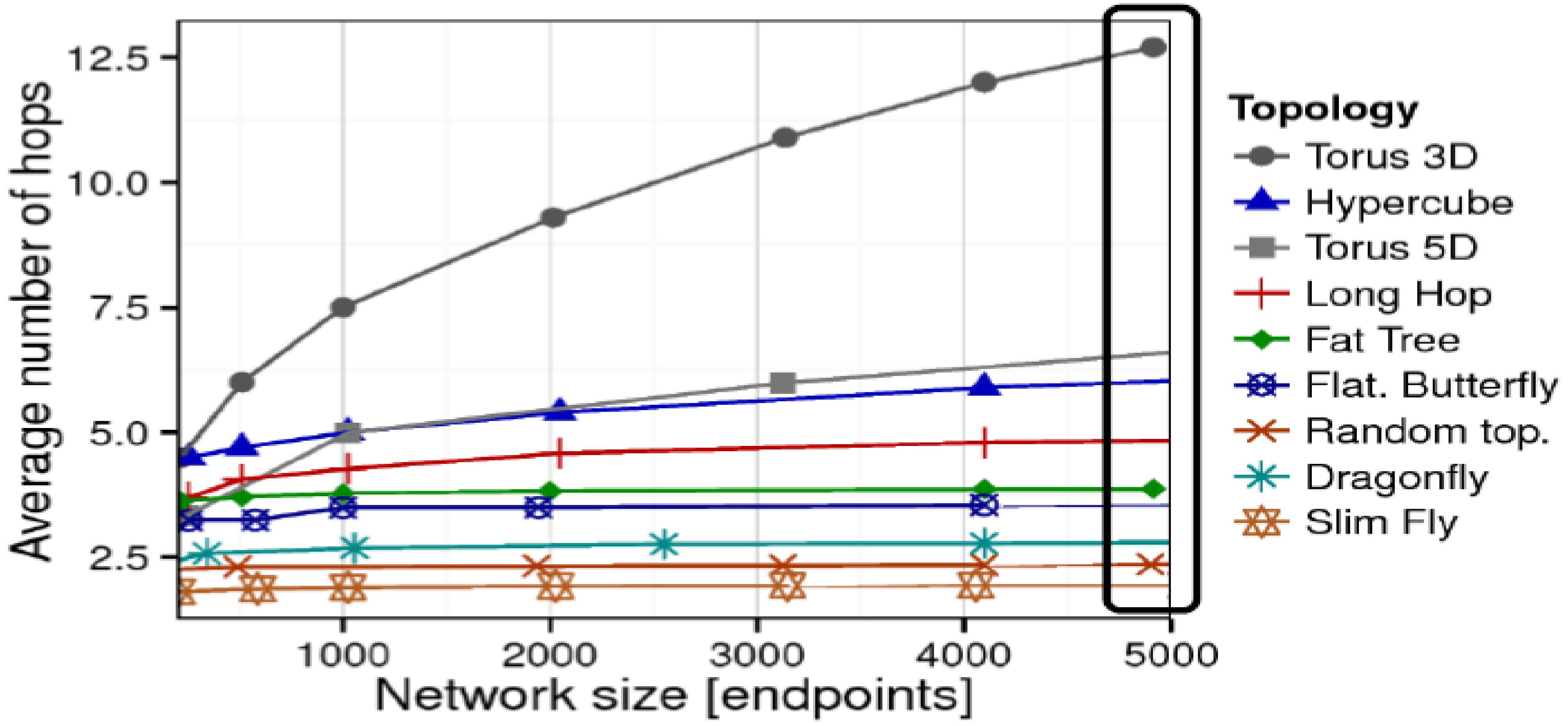
Hypercube


HyperX

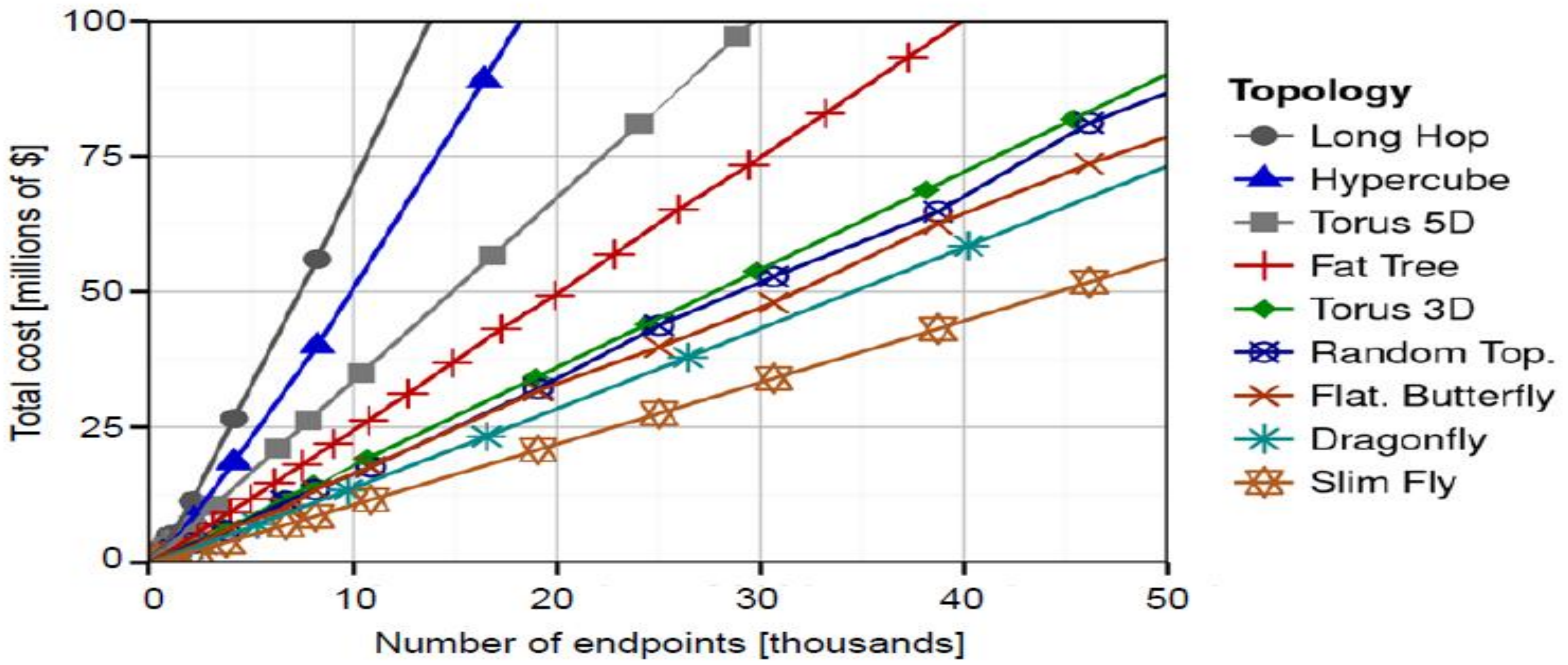
Topologies Used in Top 10 Supercomputers (2021)

S. No	Supercomputer name	Year	Topology	Interconnect
1	Fugaku(Japan)	2020	6D Torus	Tofu
2	Summit(U.S)	2018	**Fat-tree**	Infiniband
3	Sierra(U.S)	2017	**Fat-tree**	Infiniband
4	Sunway TaihuLight	2016	**Fat-tree**	Sunway
5	Perlmutter	2021	Dragonfly +	Infiniband
6	Selene(U.S)	2020	**Fat-tree**	Infiniband
7	JUWELS	2020	Dragonfly +	Infiniband
8	HPC5	2020	**Fat-tree**	Infiniband
9	Frontera(U.S)	2019	**Fat-tree**	Infiniband
10	Tianhe-2A	2017	2D-Tree	Infiniband

Topology vs Scalability



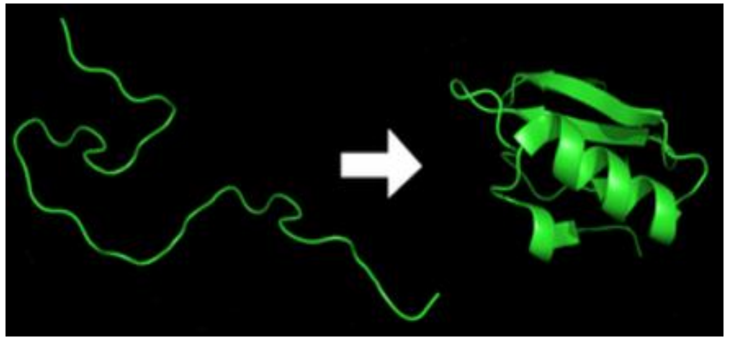
Cost vs Scalability vs Topology



Why High Performance Computing ?

Different Scientific & Engineering simulation and modeling drive the need for greater computing power.

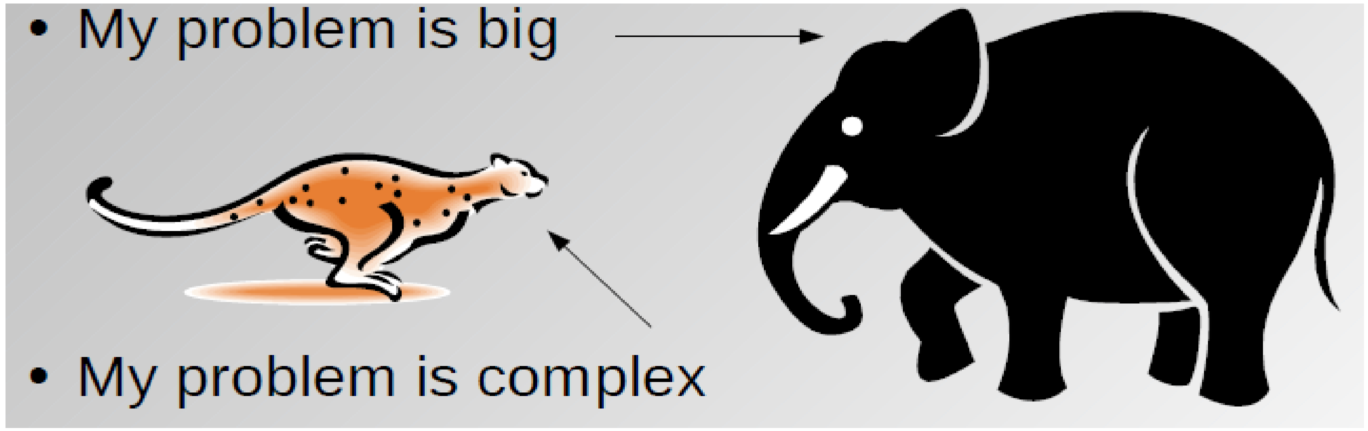
- ❑ Accurate medical imaging
- ❑ Fast and accurate web searches
- ❑ Realistic computer games, Entertainment
- ❑ Climate modeling
- ❑ Protein folding
- ❑ Artificial Intelligence
- ❑ Energy research
- ❑ Data analysis …..
- ❑ ……



Why Parallel Computing ?

Shortens Completion Time

Large Data Size Problems

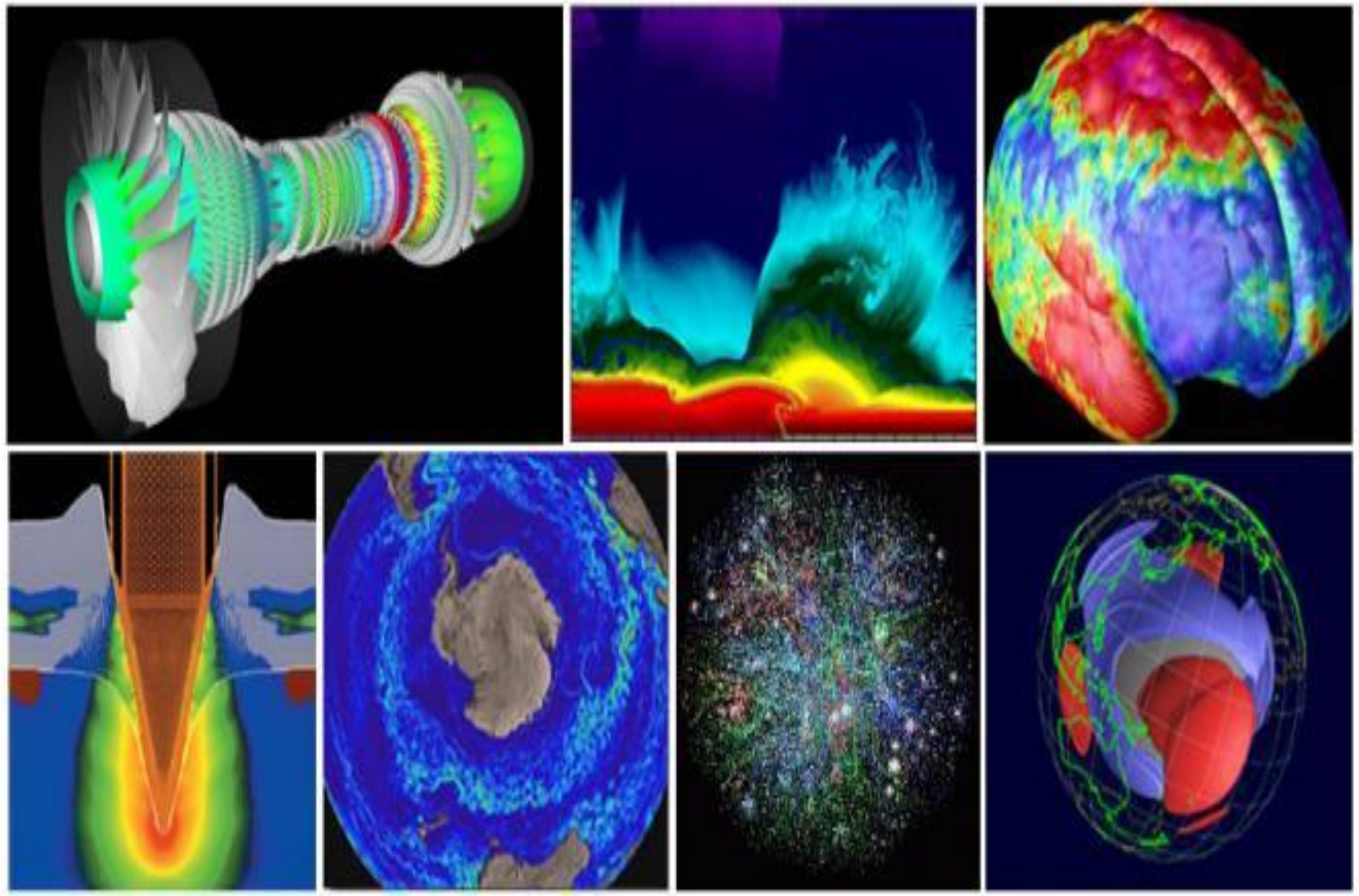


Massively Complex Phenomena

Who are Using Parallel Computing?

Science and Engineering

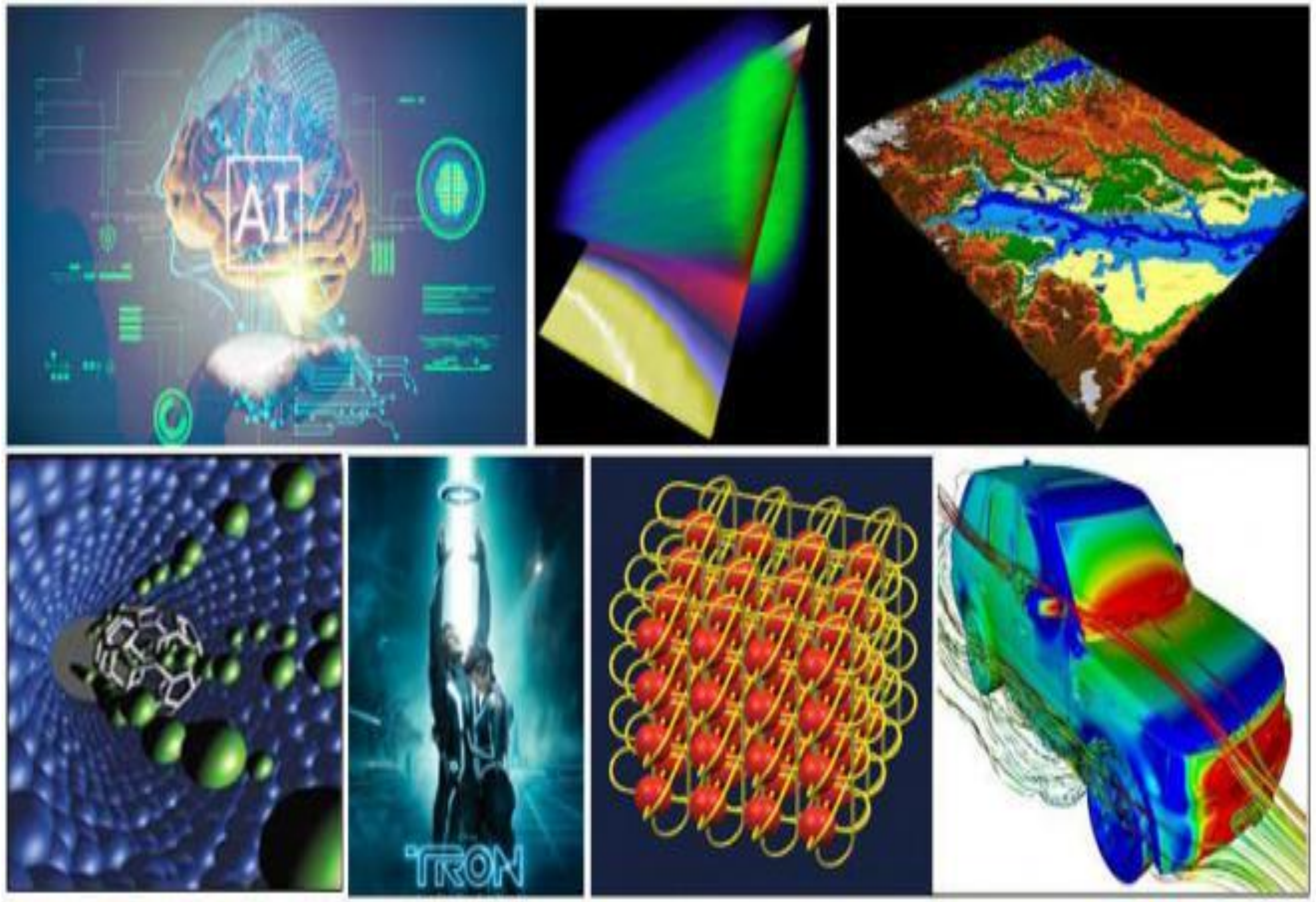
- **Atmosphere**, Earth, Environment
- **Physics** - applied, nuclear, particle, condensed matter, high pressure, fusion, photonics
- **Bioscience**, Biotechnology, Genetics
- **Chemistry**, Molecular Sciences
- **Geology**, Seismology
- **Mechanical Engineering** - from prosthetics to spacecraft
- **Electrical Engineering**, Circuit Design, Microelectronics
- **Computer Science**, Mathematics
- **Defense**, Weapons



Who are Using Parallel Computing?

Industrial and Commercial

- Big Data, databases, data mining
- Artificial Intelligence (AI)
- Oil exploration
- Web search engines, web based business services
- Medical imaging and diagnosis
- Pharmaceutical design
- Financial and economic modeling
- Management of national and multi-national corporations
- Advanced graphics and virtual reality, particularly in the entertainment industry
- Networked video and multi-media technologies
- Collaborative work environments



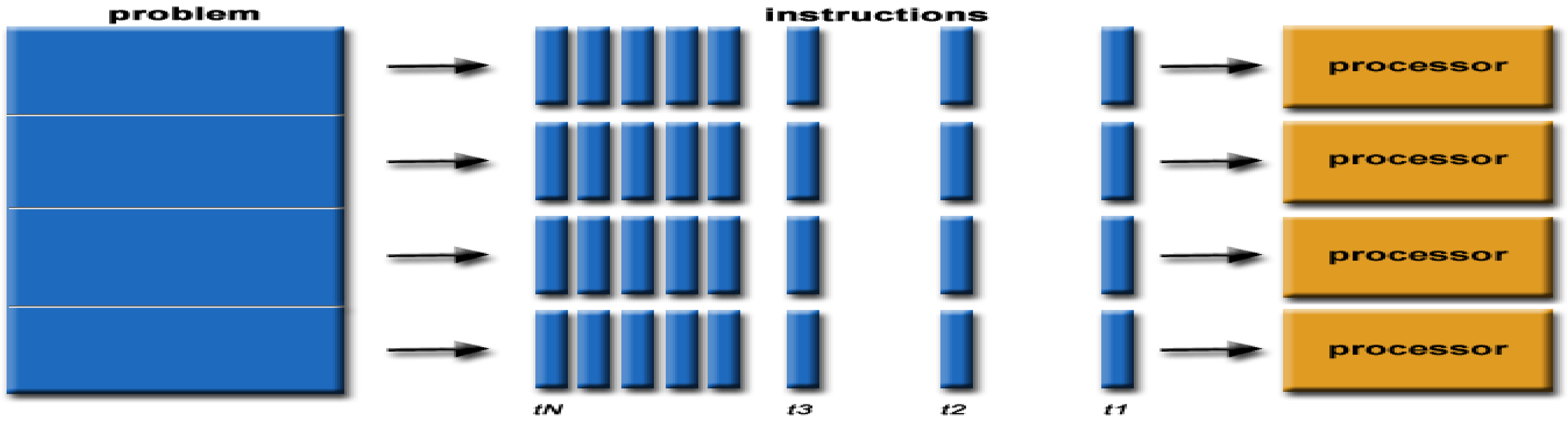
Challenges in Single Computing System?

- ❑ **Limited Computing capacity, not suitable for complex scientific problems**
- ❑ **Faster clock speed leads to cost and power/heat limitations**
- ❑ **Maximum memory size constraint for large size problems**

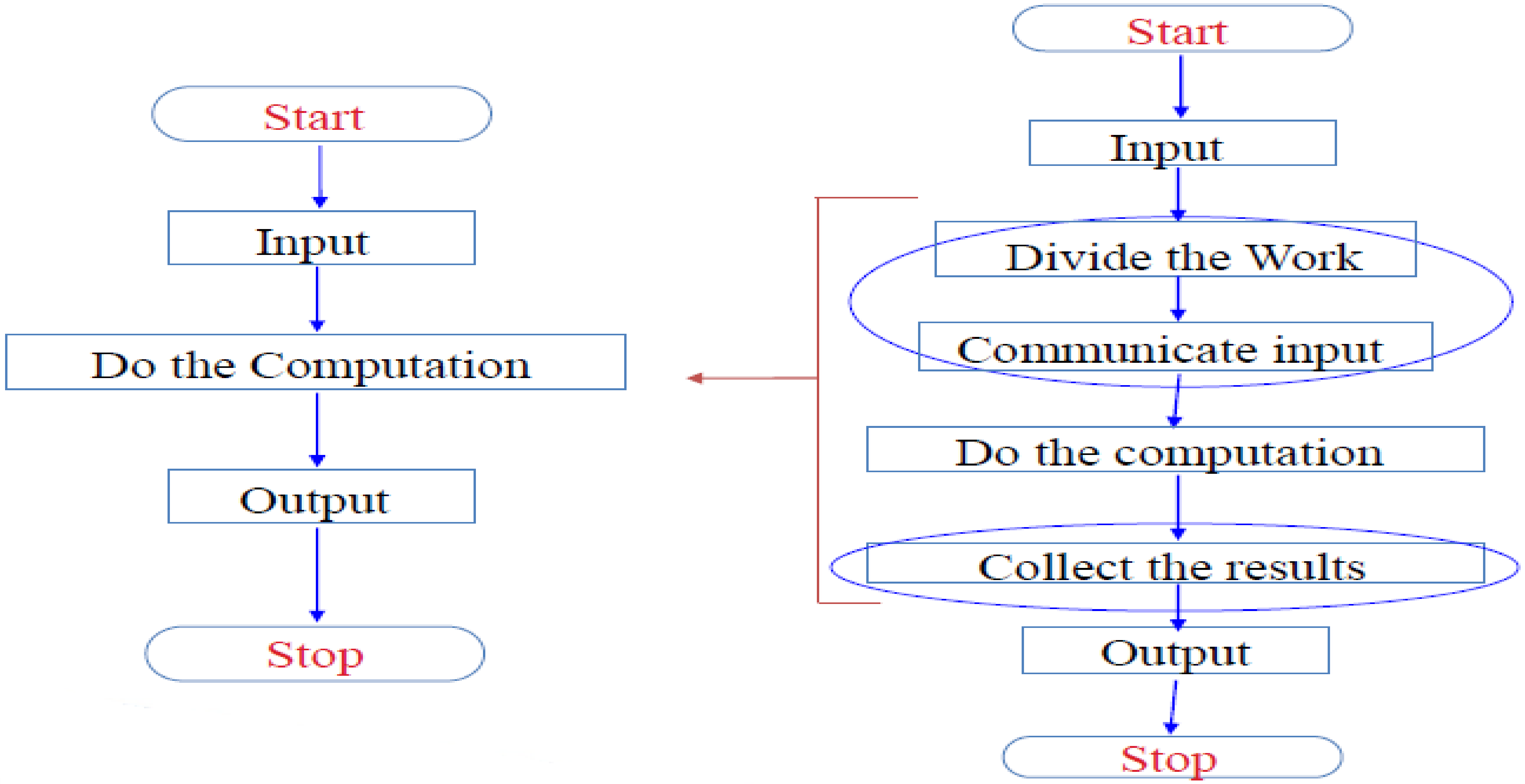
➢ **Solution:** Parallel computing – divide up the work among numerous linked systems**.**

Parallel Computing

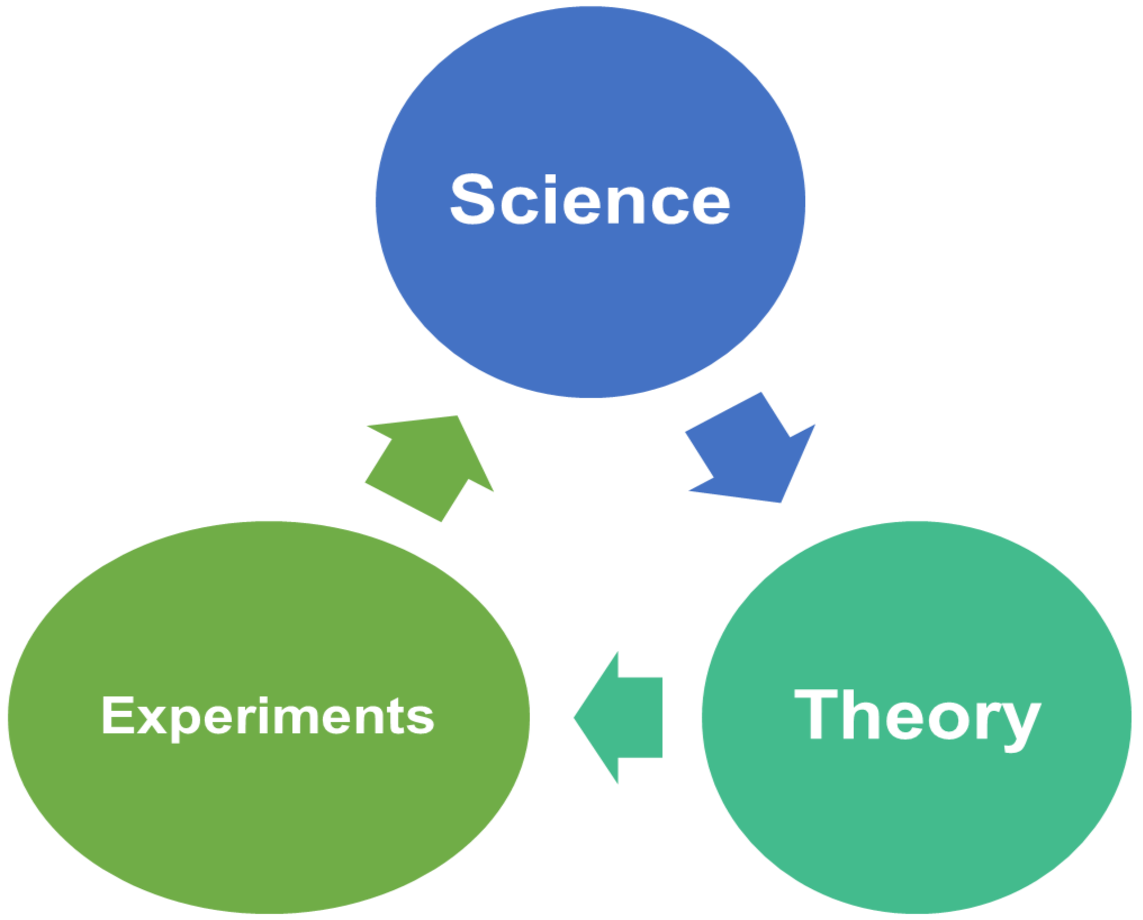
- **Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem**
  - A problem is broken into discrete parts that can be solved concurrently
  - Each part is further broken down to a series of instructions
  - Instructions from each part execute simultaneously on different processors
  - An overall control/coordination mechanism is employed



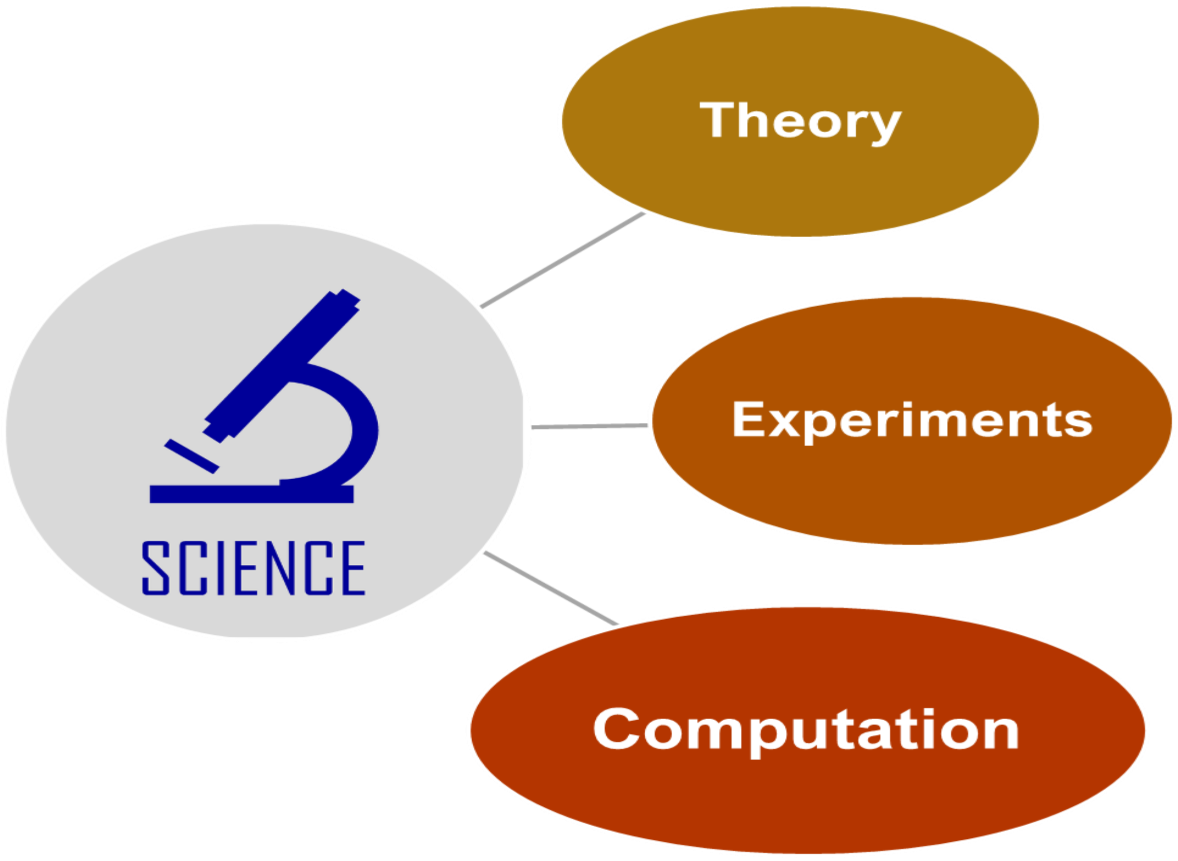
Execution flow of Parallel Systems



Evolution of Computation Techniques

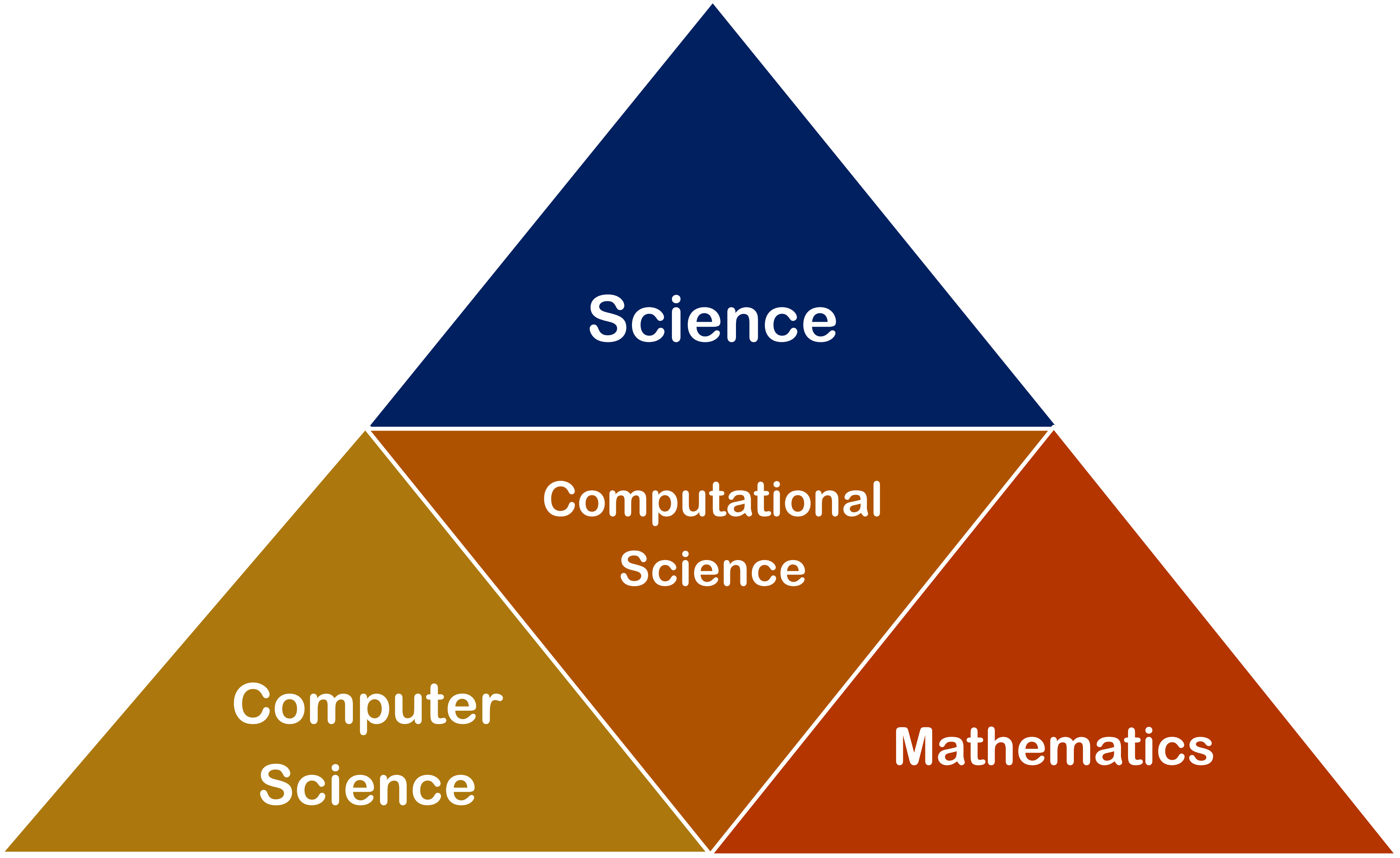


Traditional Way



Modern Way

Evolution of Computation Techniques



Computing Terminologies

Parallel Computing

Distributed Computing

Cluster Computing

Grid Computing

Thank
You