BSC~Microsoft Research Centre

Supercomputing for High Performance Applications

Dr. Tassadaq Hussain

Assistant Professor Riphah International University

Collaborations:

Microsoft Research and Barcelona Supercomputing Center

Barcelona, Spain

UCERD Pvt Ltd Islamabad

IEEE

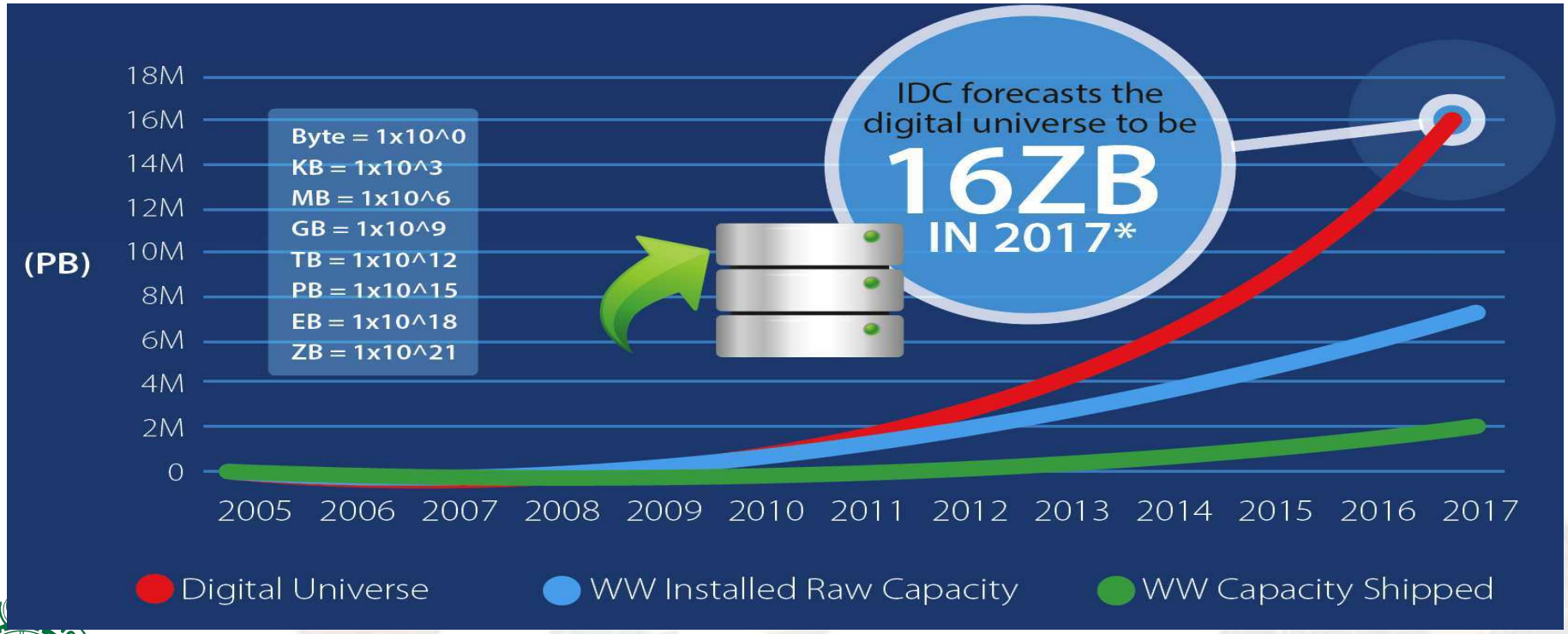
Unal Center of Education Research and Development

www.ucerd.com

- **Problem Statement**
- HPC System
- Supercomputing System
- High Performance Applications
- Supercomputer Design
- Conclusion
- Hands on Experience

Information Future Trend

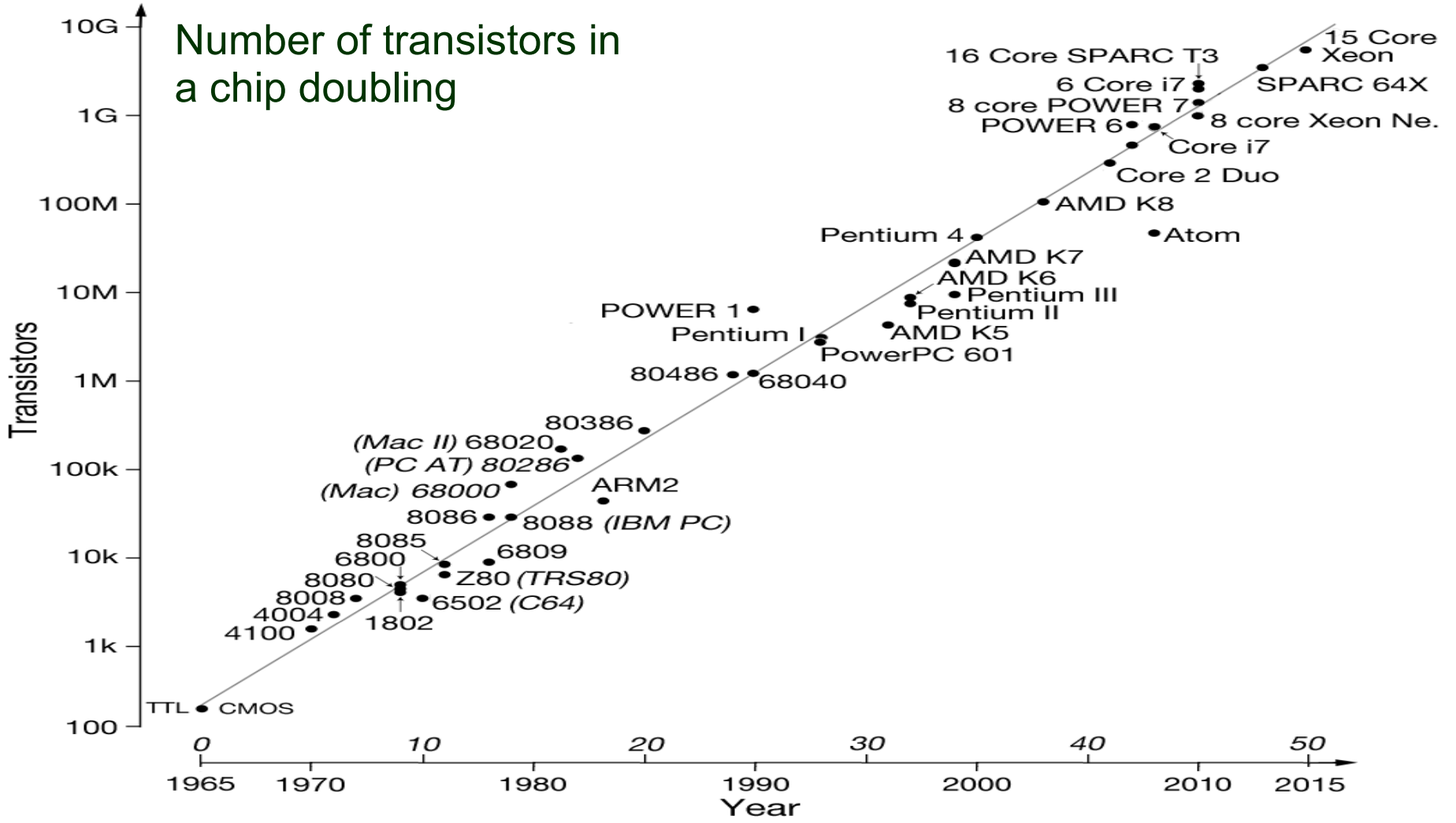
- Information Age
- Information doubling after every 18 months



Technology Research Gartner Inc. states that information data volume doubles after every 18 months.

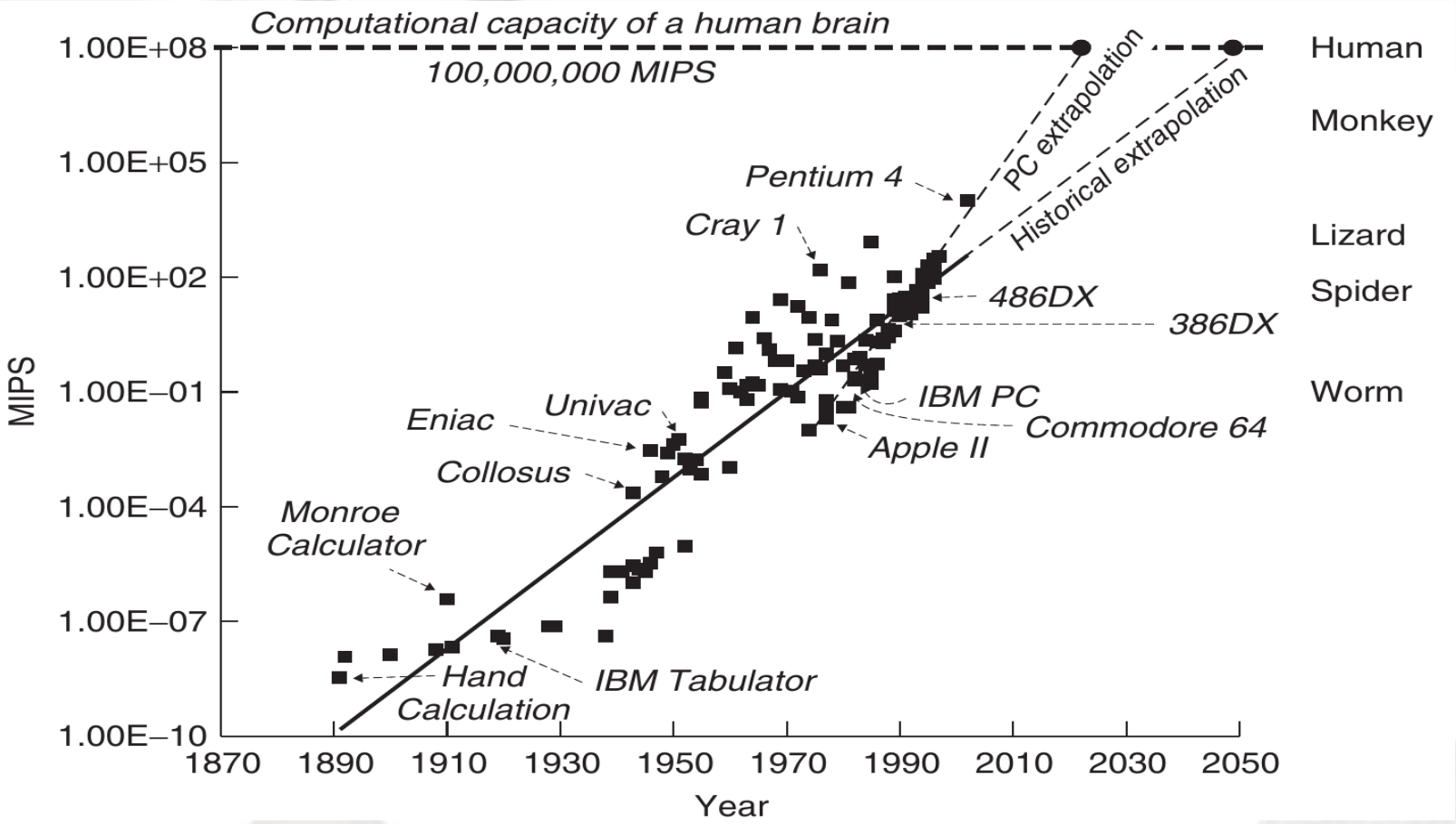
Moore's Law: Transistor Count

Number of transistors in a chip doubling



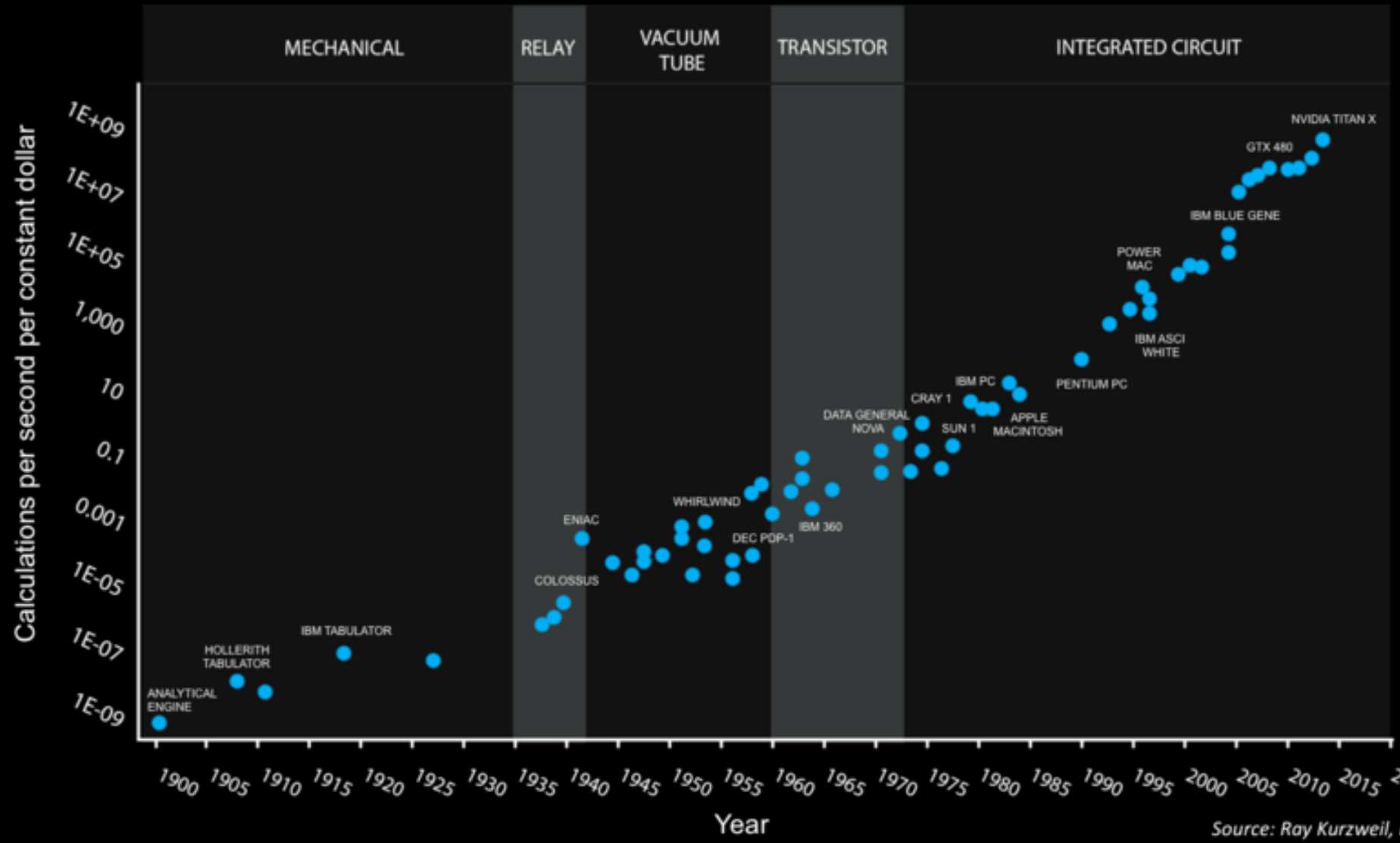
[1] G. E. Moore, "Cramming more components onto integrated circuits," Electronics, vol. 38, no. 8, April 1965.

Performance Improvement



It is estimated that sometime between the years **2025 and 2050**, a **personal computers** will exceed the calculation power of a human brain.

120 Years of Moore's Law

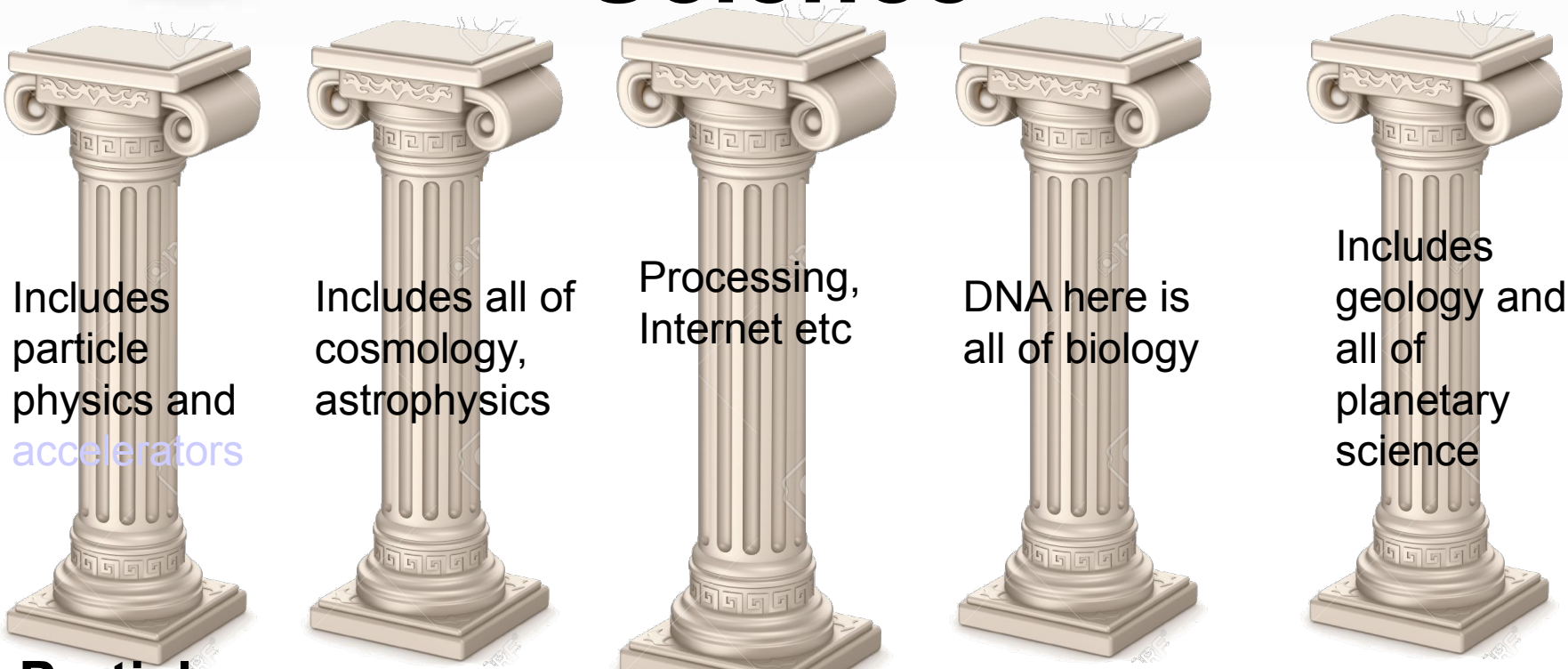


Source: Ray Kurzweil, DFJ

Pillars of Science

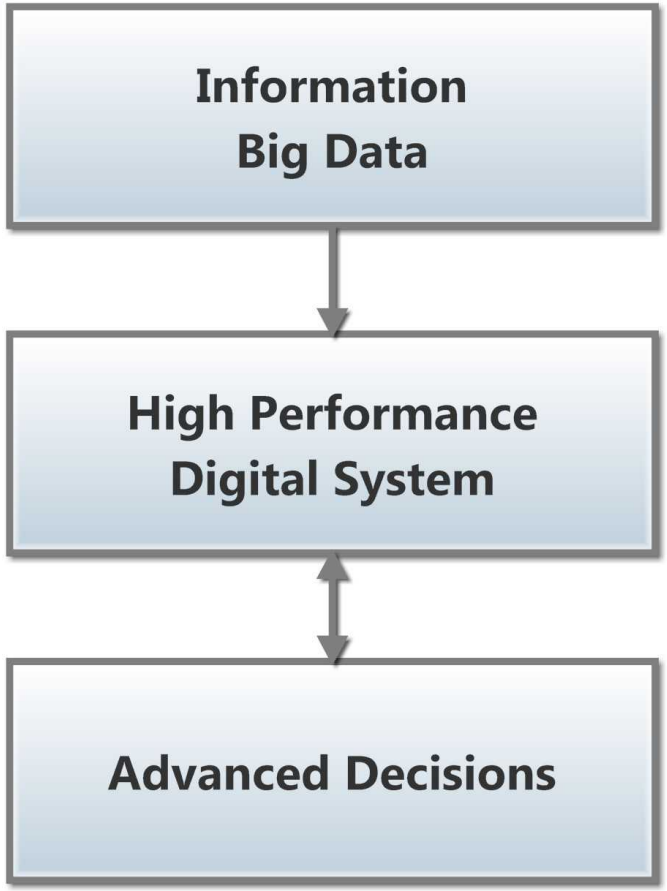
Fermi National Accelerator Laboratory

Science



Importance of HPC System

- The information data volume doubles after every 18 months.
- The performance of digital system get improved after every 18 months.
- High Performance, Low Cost and Low Power Computer Systems.



Relevance to industry and academia

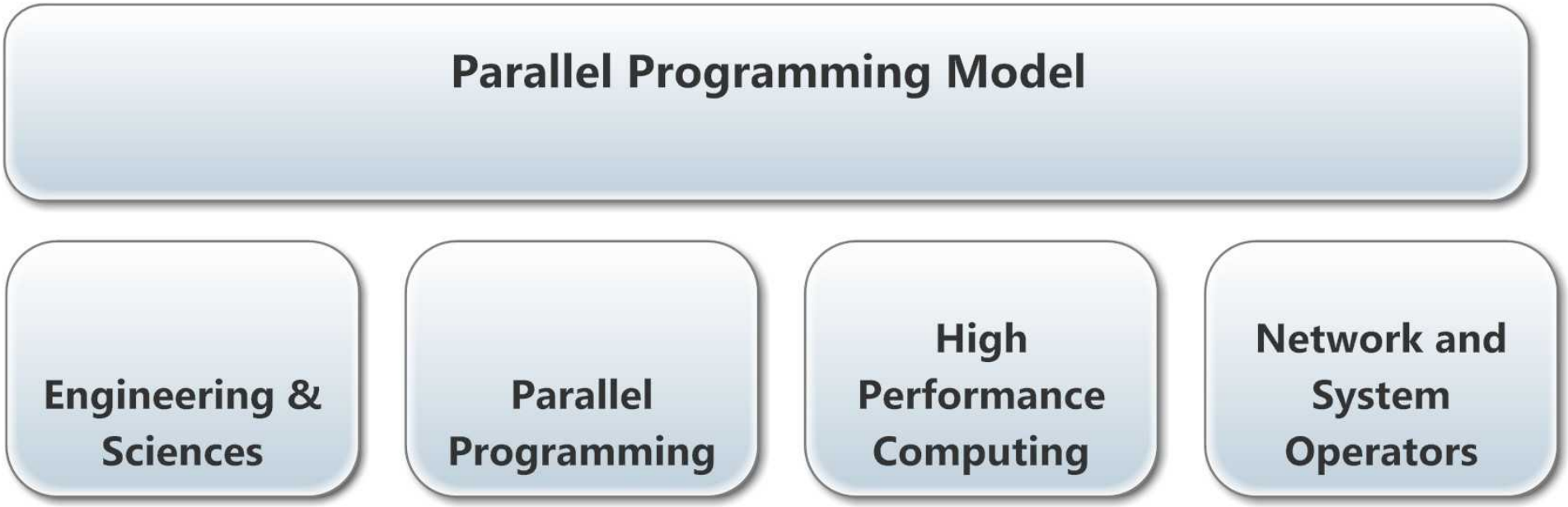
- High performance computing is the need of the day.
- Equally important for high tech industry.
- Optimum resource utilization.
- Less time to solve complex compute intensive problems.

Challenges

- Restrictions on High Performance Target Technologies.
- Limited availability of High Performance Advance hardware.
- Even having high performance hardware does not guaranty its optimum usage.
- High end expertise are required to utilize high performance hardware/software.

- Problem Statement
- **HPC System**
- Supercomputing System
- High Performance Applications
- Supercomputer Design
- Conclusion
- Hands on Experience

Generic Proposal: High Performance Supercomputing



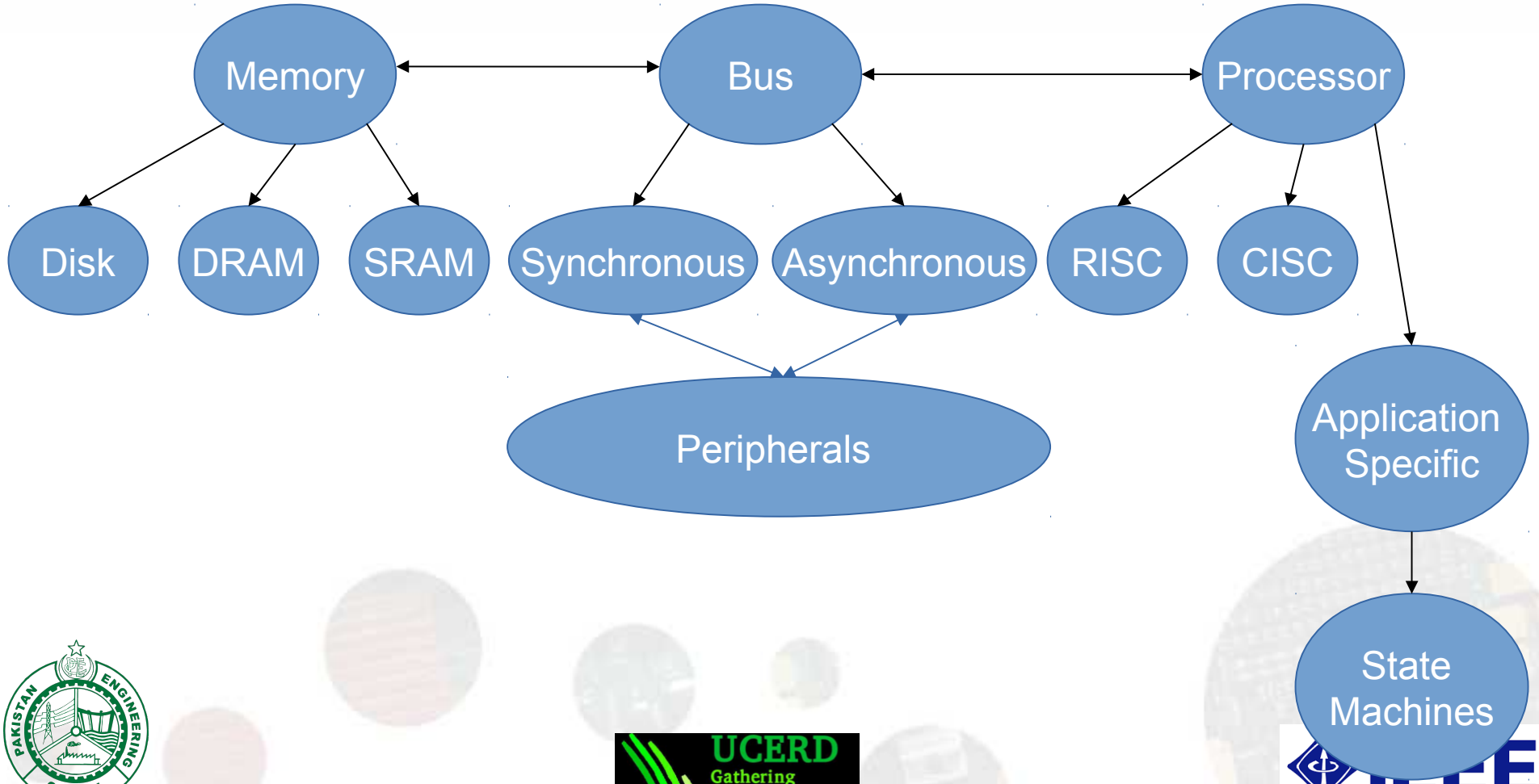
High performance system architectures for Artificial Intelligence, Embedded Real-time Systems etc.

Objectives

- Executes Engineering and Sciences Applications
  - Compute and Data Intensive
  - Complex and irregular data structures
  - e.g. Artificial Intelligence, Fluid Dynamics, Structural Analysis, 3D/4D Imaging.
- Handle Information in Big Data
  - Support local memory, main memory and external memory systems
  - Perform memory read/write operations in parallel with processing unit
- Multiple Heterogeneous Cores
  - RISC (SSP), vector processor (VP) and application specific hardware accelerator (ASHA)
- Provides Programming support
  - Provide standard C/C++ parallel programming languages for real-time and standalone applications.
  - Support Tools for Visual Analysis, Modeling and Simulations e.g Ansys HPC.

- Problem Formulation
- Proposal
- **Supercomputing System**
- High Performance Applications
- Supercomputer Architectures
- Conclusion
- Hands on Experience

Digital System Components



Processor System Architecture

- Hardware
  - Processor
  - Bus
  - Memory
  - Peripherals

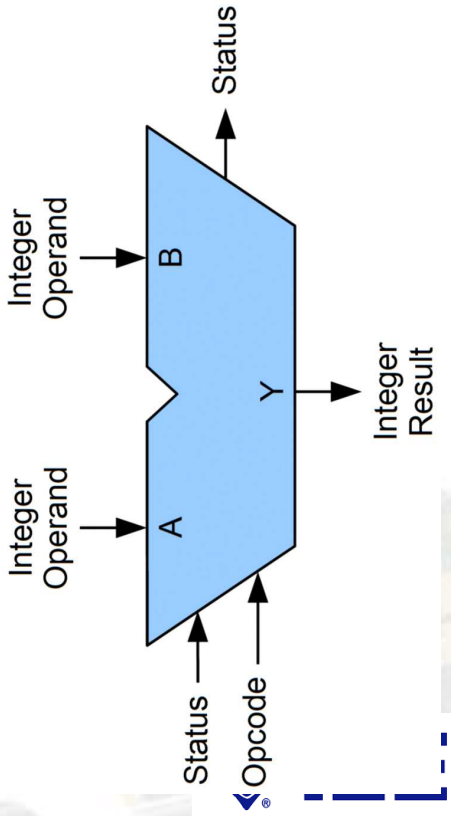
Processor

A simple processor takes a single instruction and generate results in a given time called instruction cycles.

An instruction includes two values (operands) and an arithmetic or a logic operation (operator).

Values (operands) can be from memory

or peripherals.



Important Parameters of a Processor

Clock

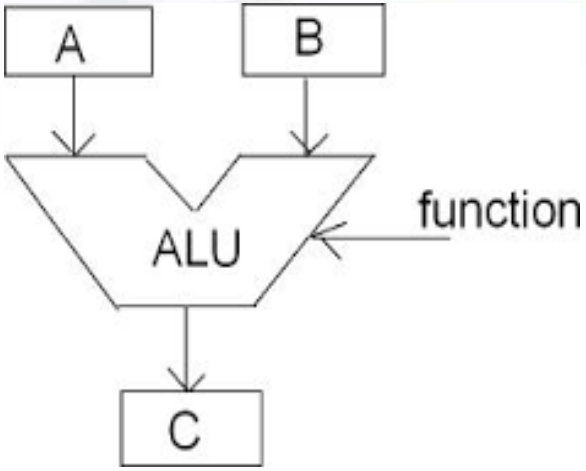
Data Bus

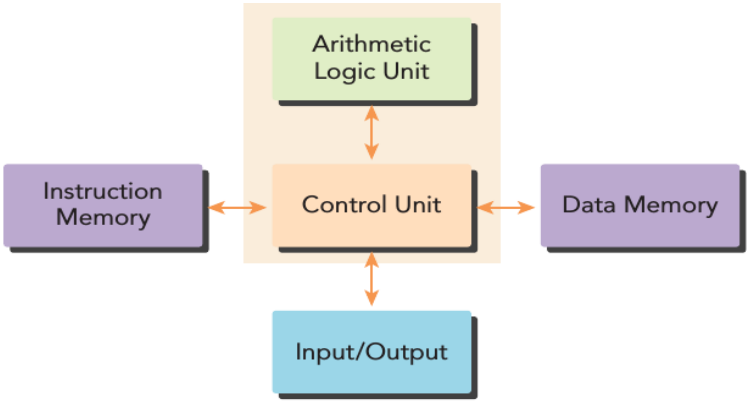
Instruction Bus

Instructions Per Cycles

Pipeline Stage

Basic introduction of Microprocessor





0	A	B	A+B
1	A	B	A-B

Processor Architectures

SISD

- RISC

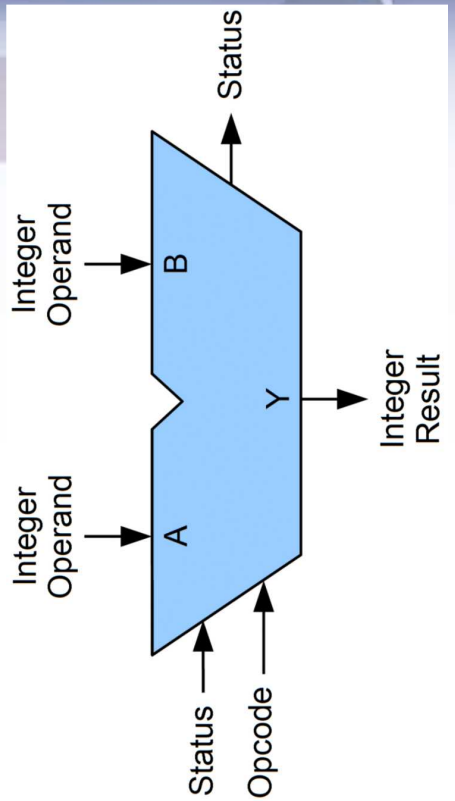
SIMD

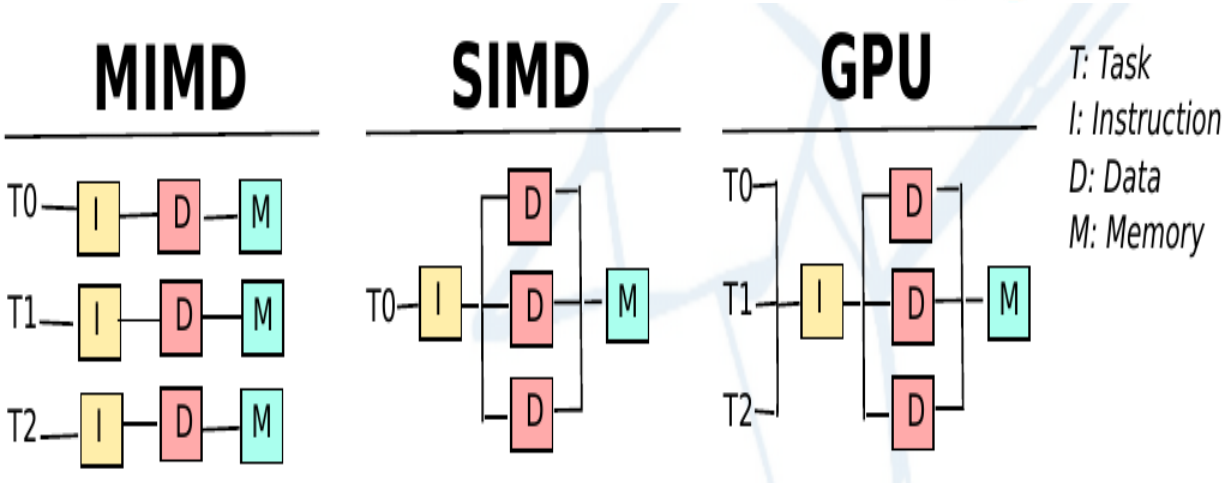
- CISC

MISD

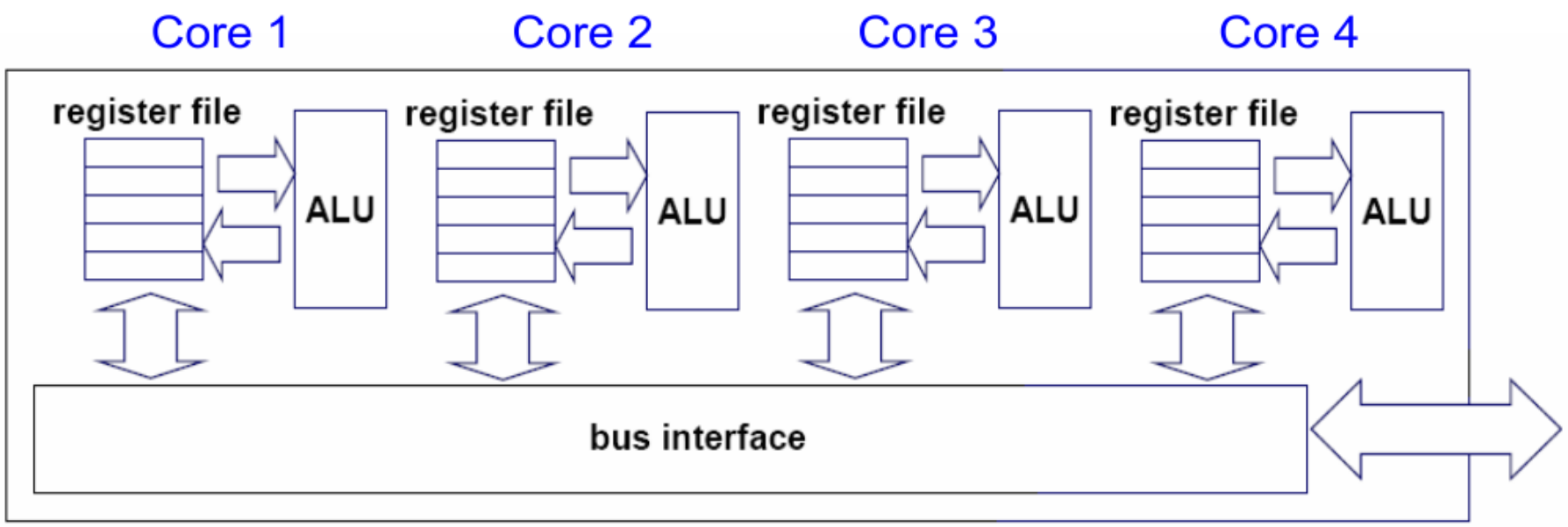
MIMD

- Multi-core





Multi-core Processor



Types of MIMD Architecture

Centralized shared-memory architectures or symmetric shared-memory multiprocrssors (SMP) or uniform memory access (UMA) architectures.

Distributed-memory multiprocessors.

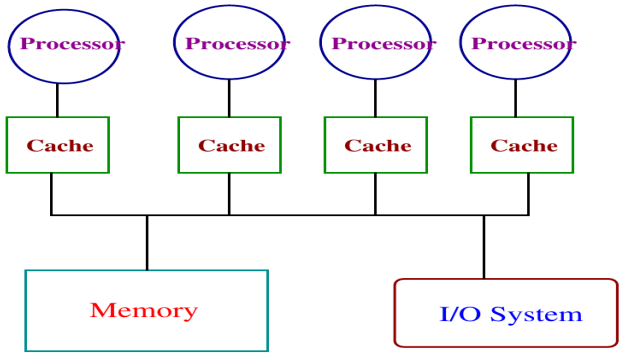
SMP: Shared Memory Processor

Small number of similar processors (at most a few dozen).

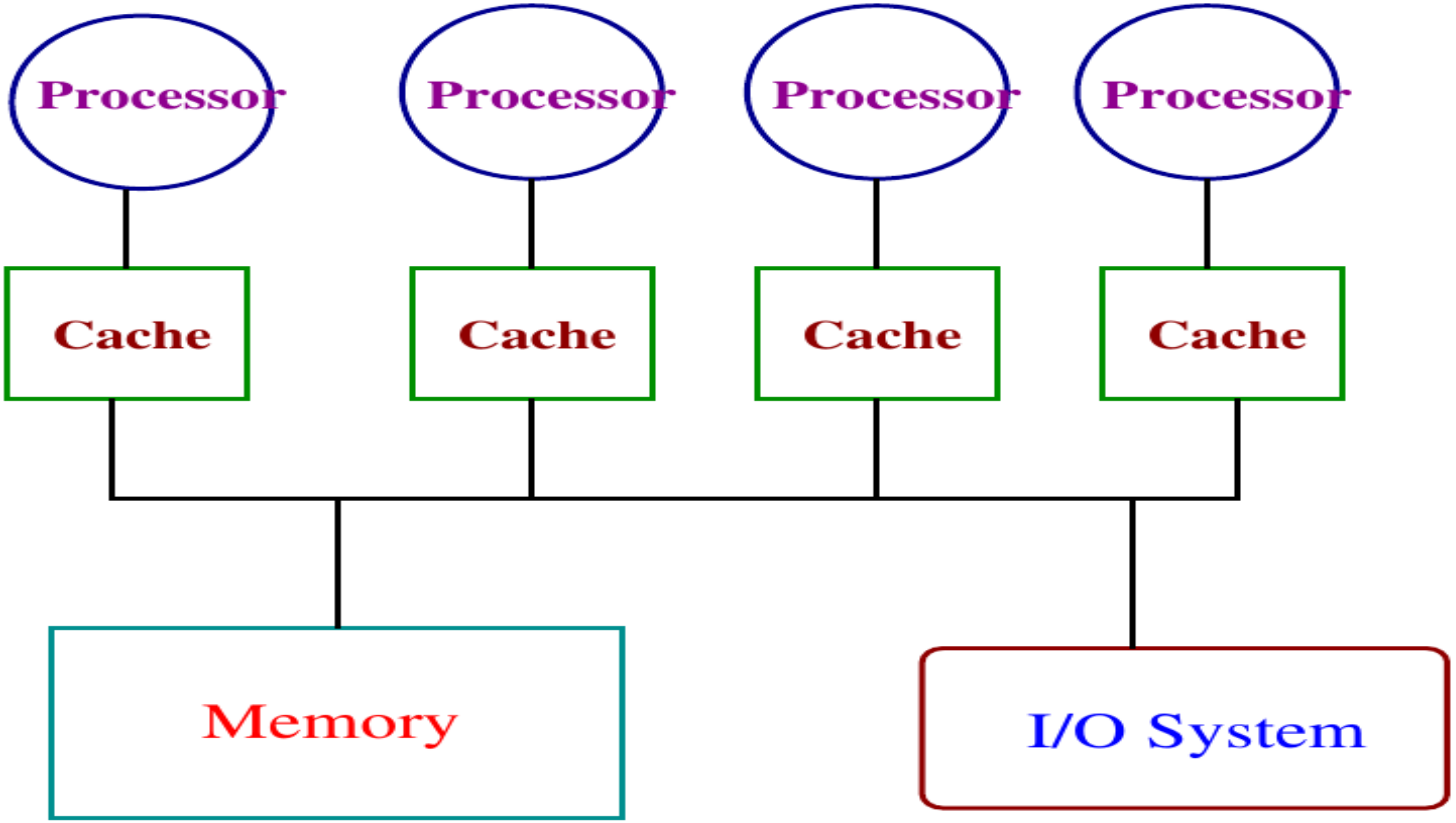
Each processor has a large cache.

A centralized memory (multiple banks) is shared through a memory bus.

Each memory location has identical access time from each processor.



SMP


Processor
Processor
Processor
Processor
Cache
Cache
Cache
Cache
Memory
I/O System

Distributed Memory

Larger processor count.

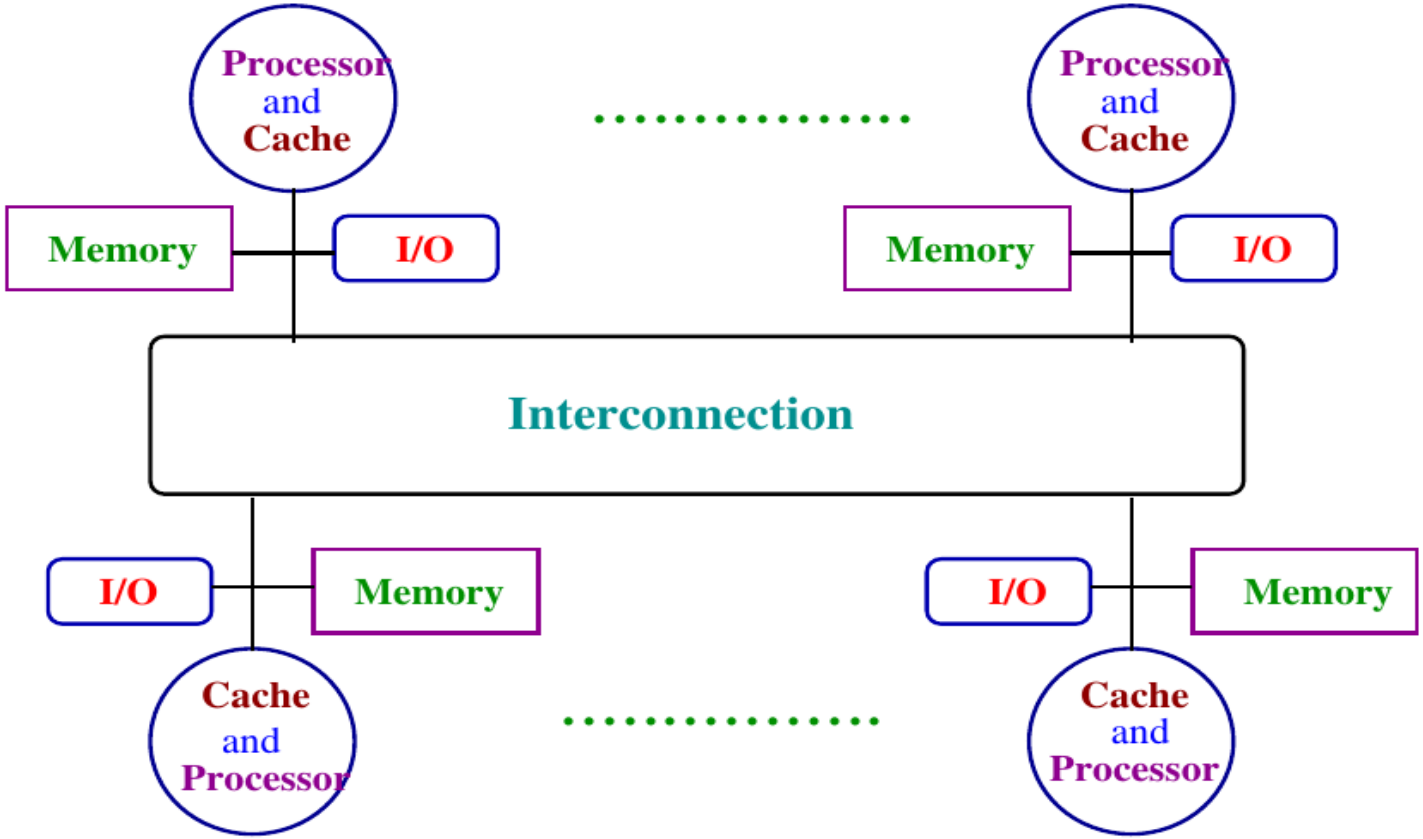
Memory is physically distributed among the processors for better bandwidth.

Connected through high-speed interconnection e.g. switches.

Distributed Memory

The bandwidth for the local memory is high and the latency is low.

But access to data present in the local memory of some other processor is complex and of high latency.


Processor and Cache
Processor and Cache
Memory
I/O
Memory
I/O
Interconnection
I/O
Memory
I/O
Memory
Cache and Processor
Cache and Processor

Distributed Shared Memory Architecture

Large-scale multiprocessors have physically distributed memory with the processors.

There are essentially two different models of memory architectures and the corresponding models of communication.

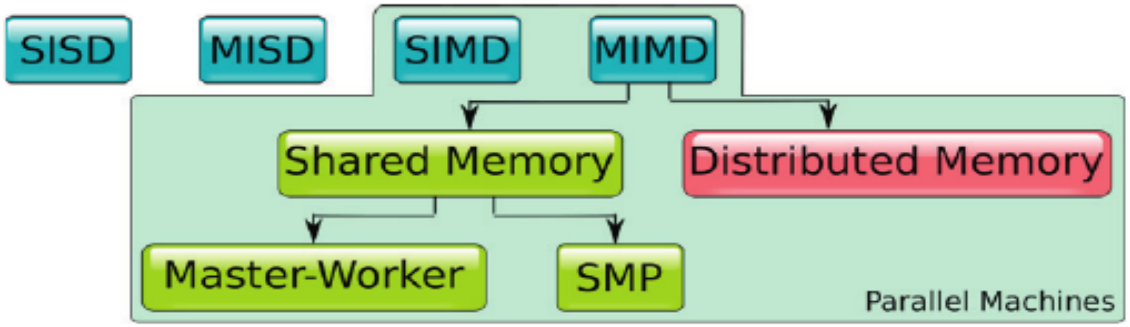
DSM Architecture

- Memory is distributed with different processors to support higher bandwidth demand of larger number of processors.
- Any processor can access a location of physically distributed memory (with proper access permission).
- This is called distributed shared-memory architecture (DSM) also known as NUMA (nonuniform memory access).

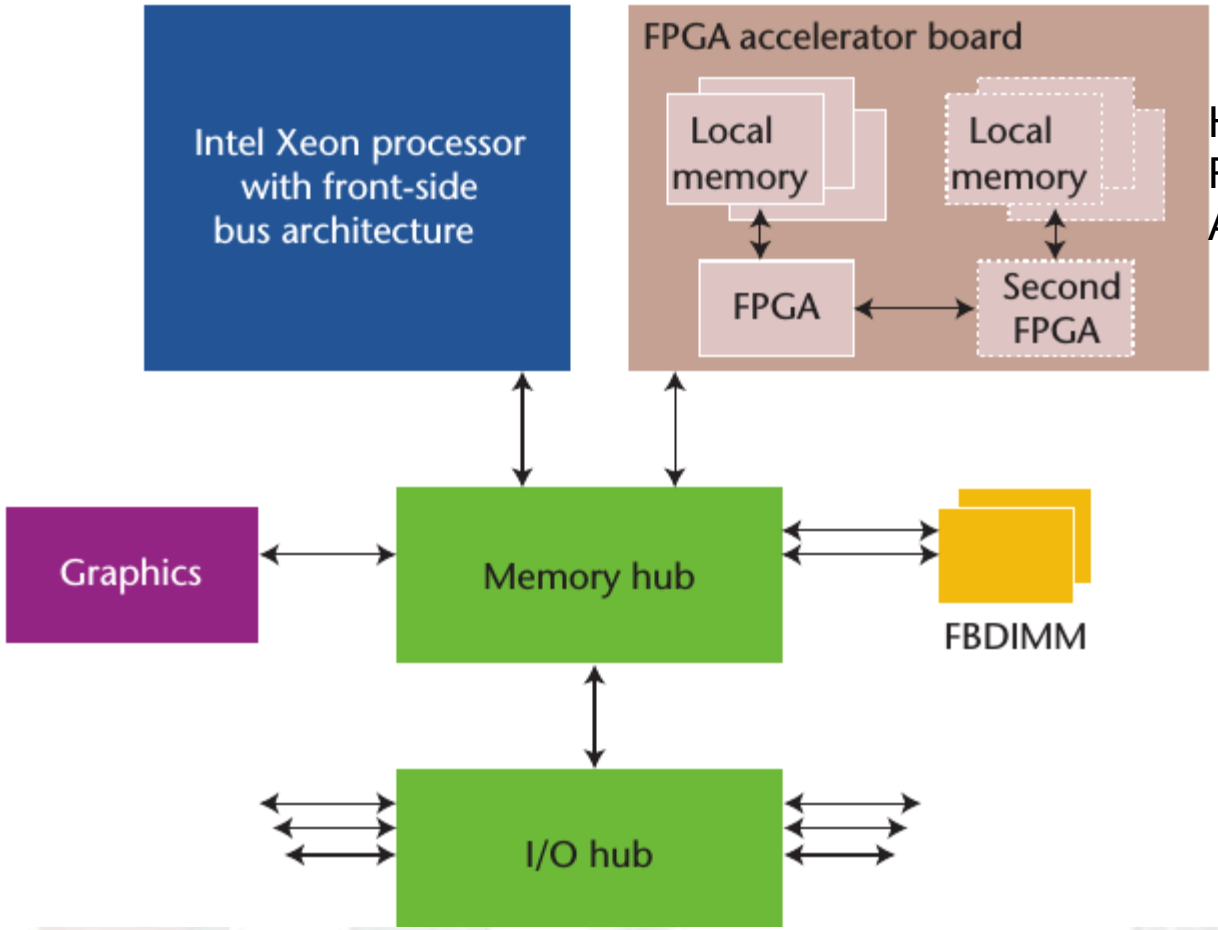
Available Computer Architectures

There are currently two trends in utilizing the increased transistor count afforded by miniaturization and advancements in semiconductor materials:

- Increase the **on-chip core count**,
  - Combined with augmented specialized SIMD instruction sets (e.g., SSE and its subsequent versions, MMX, AESNI, etc.) and larger caches.
  - This is best exemplified by Intel's x86 line of CPUs and the Intel Xeon Phi coprocessor.
- **Combine heterogeneous cores** in the same package,
  - Typically CPU and GPU ones, each optimized for a different type of task.
  - This is best exemplified by AMD's line of Accelerated Processing Unit (APU) chips. Intel is also offering OpenCL-based computing on its line of CPUs with integrated graphics chips.



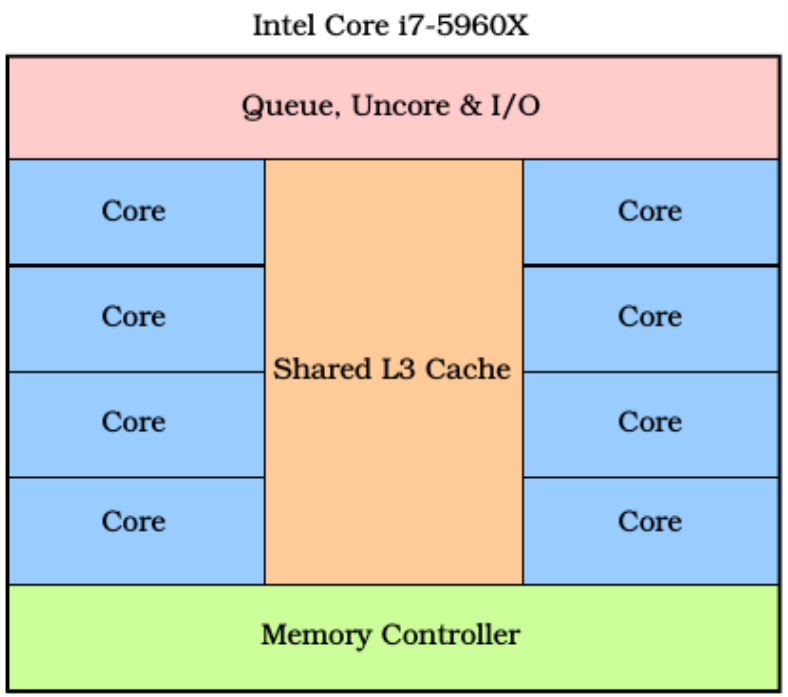
Reconfigurable Accelerators



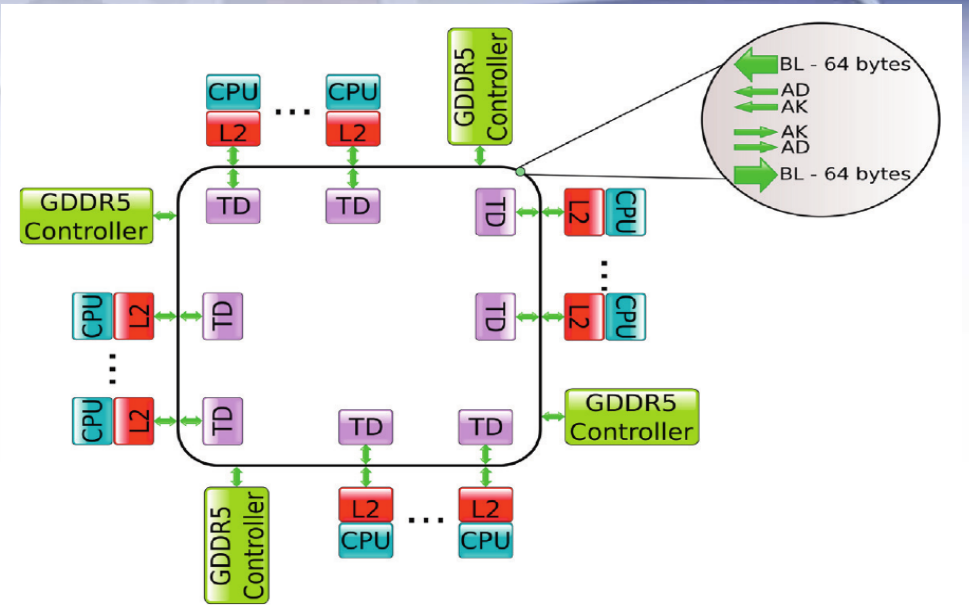
HPC Reconfigurable Architecture

CPU: Intel Processor

CPUs employ large on-chip (and sometimes multiple) memory caches, few complex (e.g., pipelined) arithmetic and logical processing units (ALUs), and complex instruction decoding and prediction hardware to avoid stalling while waiting for data to arrive from the main memory.



Intel Xeon Phi



- A Super-scalar Architecture
- Xeon Phi comes equipped upto 72 x86 cores that are heavily customized Pentium cores.
- The customizations include the ability to handle four threads at the same time.
- The coherency is managed by distributed tag directories (TDs)

Intel Super Scalar: A Many Core Architecture

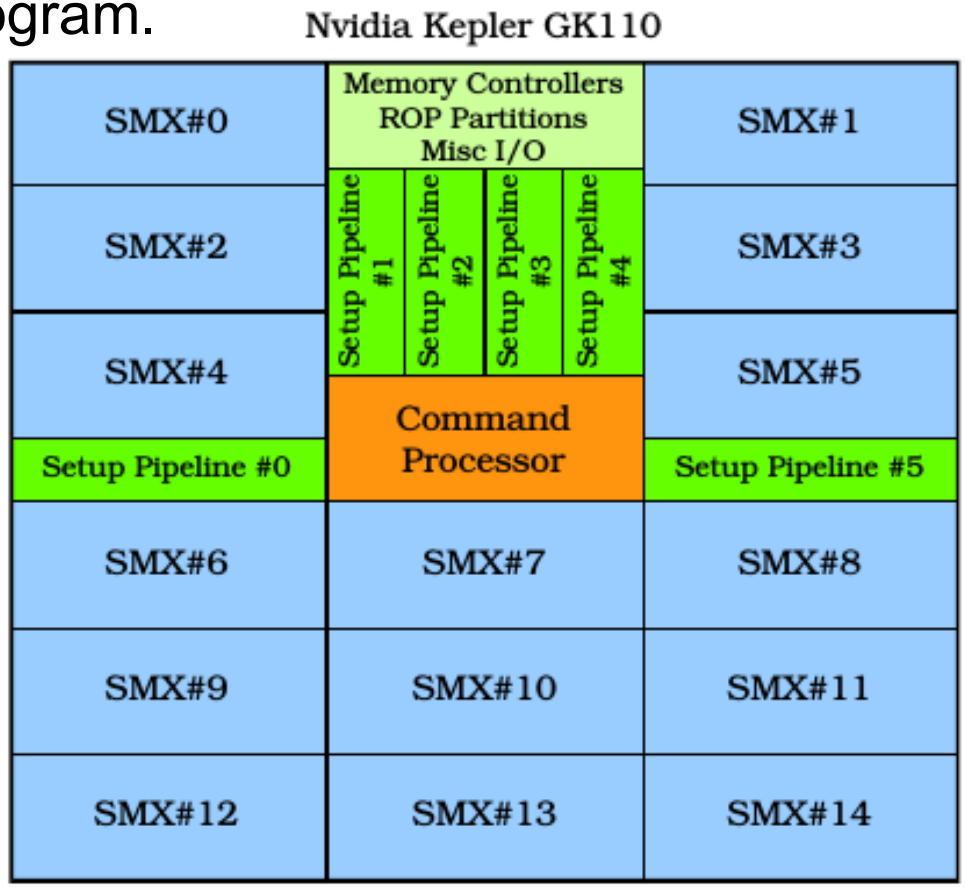
Processor Number	Availability	# of Cores/# of Threads	Clock Speed	Max TDP/Power	Memory Types	Fabric	L2 Cache
Intel® Xeon Phi™ Processor 7250 (16GB, 1.40 GHz, 68 core)	Now	68/272	1.4 GHz	215 W	DDR4-2400	No	34 MB
Intel® Xeon Phi™ Processor 7230 (16GB, 1.30 GHz, 64 core)	Now	64/256	1.3 GHz	215 W	DDR4-2400	No	32 MB
Intel® Xeon Phi™ Processor 7210 (16GB, 1.30 GHz, 64 core)	Now	64/256	1.3 GHz	215 W	DDR4-2133	No	32 MB
Intel® Xeon Phi™ Processor 7290 (16GB, 1.50 GHz, 72 core)	Sept. 2016	72/288	1.5 GHz	245 W	DDR4-2400	No	36 MB
Intel® Xeon Phi™ Processor 7290F (16GB, 1.50 GHz, 72 core)	Oct. 2016	72/288	1.5 GHz	260 W	DDR4-2400	Yes	36 MB
Intel® Xeon Phi™ Processor 7250F (16GB, 1.40 GHz, 68 core)	Oct. 2016	68/272	1.4 GHz	230 W	DDR4-2400	Yes	34 MB
Intel® Xeon Phi™ Processor 7230F (16GB, 1.30 GHz, 64 core)	Oct. 2016	64/256	1.3 GHz	230 W	DDR4-2400	Yes	32 MB
Intel® Xeon Phi™ Processor 7210F (16GB, 1.30 GHz, 64 core)	Oct. 2016	64/256	1.3 GHz	230 W	DDR4-2133	Yes	32 MB

Graphics Processing Unit (GPU) and CPU

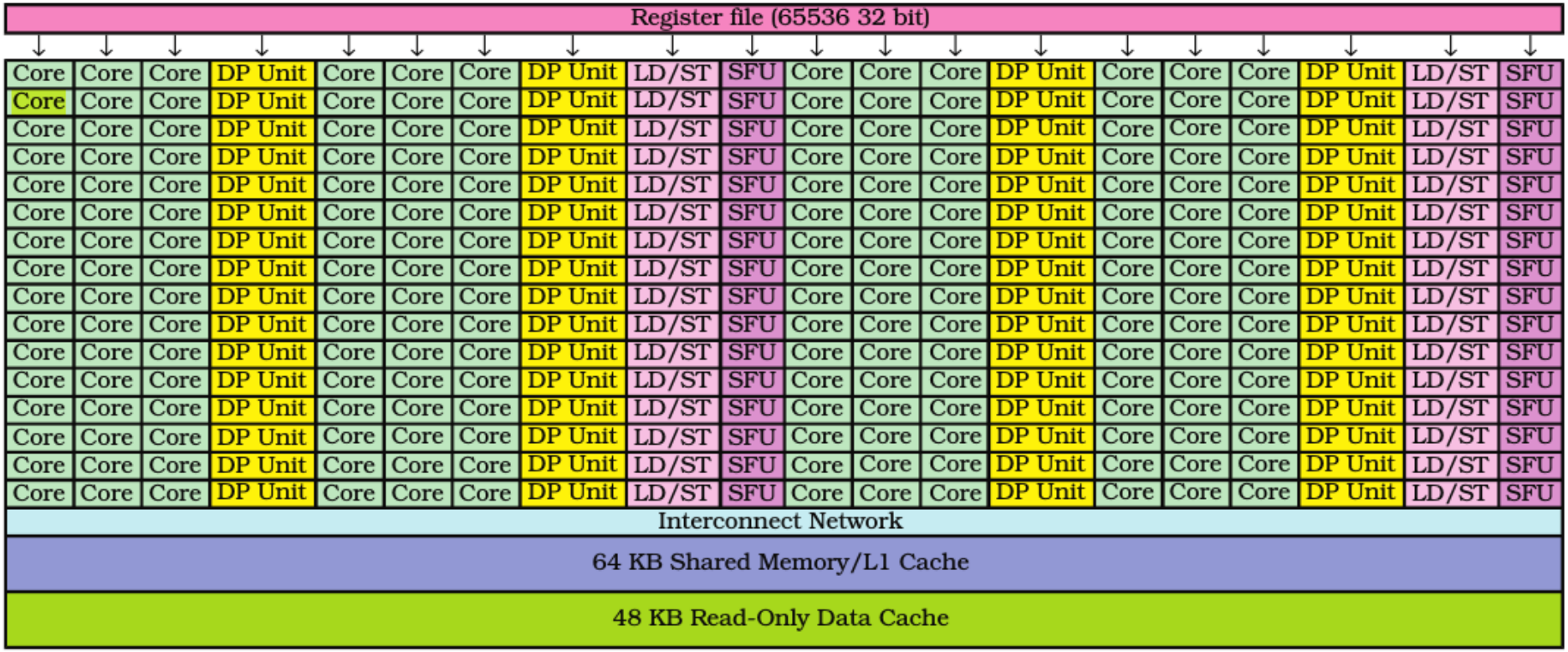
- **GPUs** have been developed as a means of processing massive amount of graphics data very quickly, before they are placed in the card's display buffer.
- Their design envelope dictated a layout that departed from the one traditionally used by conventional CPUs.
- **GPU** uses small on-chip caches with a big collection of simple ALUs capable of parallel operation, since data reuse is typically small for graphics processing and programs are relatively simple. In order to feed the multiple cores on a GPU, designers also dedicated very wide, fast memory buses for fetching data from the GPU's main memory.

Nvidia Graphics Processing Unit (GPU)

- SM, SMM SMX (Streaming Multiprocessors): Single SMX contains 192 cores executes in SIMD fashion
- Each SMX can run its own program.
- CUDA and OpenACC Programming Models



GPU SMX Internal Architecture



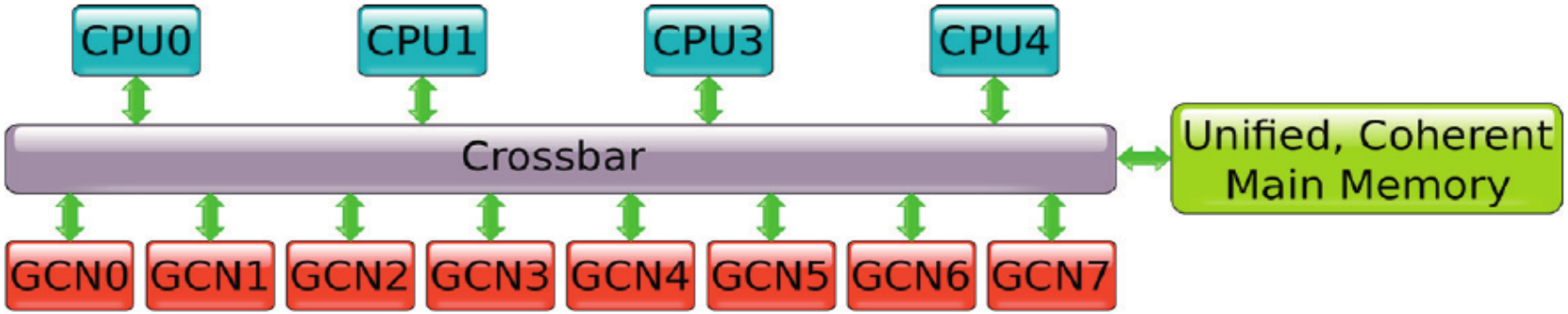
192 **Core** : single-precision cores
64 **DP Unit** : double -precision cores
32 **LD/ST** : load/store units
32 **SFU** : Special Function Units

High Performance Accelerators

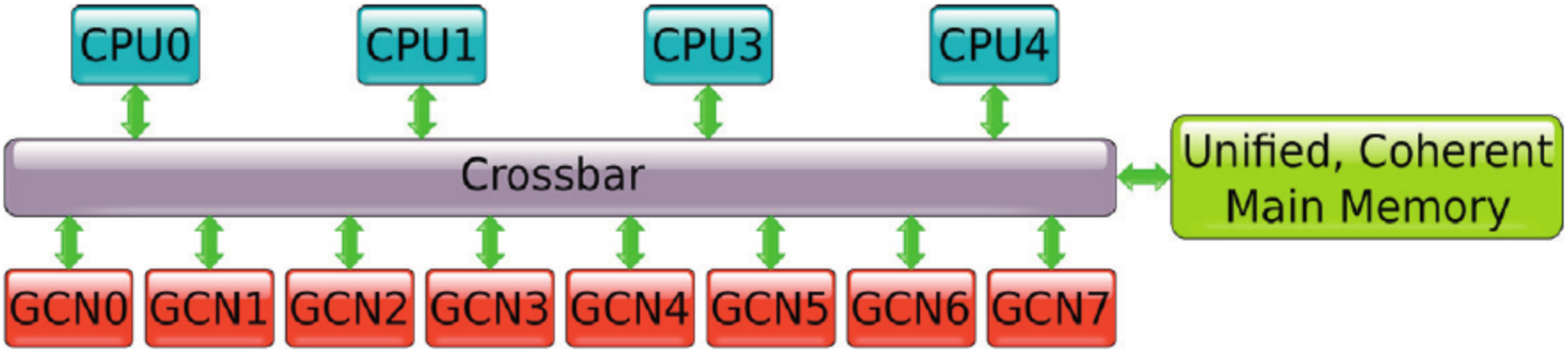
NVIDIA Tesla Family Specification Comparison				
	Tesla P100	Tesla K80	Tesla K40	Tesla M40
Stream Processors	3584	2 x 2496	2880	3072
Core Clock	1328MHz	562MHz	745MHz	948MHz
Boost Clock(s)	1480MHz	875MHz	810MHz, 875MHz	1114MHz
Memory Clock	1.4Gbps HBM2	5Gbps GDDR5	6Gbps GDDR5	6Gbps GDDR5
Memory Bus Width	4096-bit	2 x 384-bit	384-bit	384-bit
Memory Bandwidth	720GB/sec	2 x 240GB/sec	288GB/sec	288GB/sec
VRAM	16GB	2 x 12GB	12GB	12GB
Half Precision	21.2 TFLOPS	8.74 TFLOPS	4.29 TFLOPS	6.8 TFLOPS
Single Precision	10.6 TFLOPS	8.74 TFLOPS	4.29 TFLOPS	6.8 TFLOPS
Double Precision	5.3 TFLOPS (1/2 rate)	2.91 TFLOPS (1/3 rate)	1.43 TFLOPS (1/3 rate)	213 GFLOPS (1/32 rate)
GPU	GP100 (610mm2)	GK210	GK110B	GM200
Transistor Count	15.3B	2 x 7.1B(?)	7.1B	8B
TDP	300W	300W	235W	250W
Cooling	N/A	Passive	Active/Passive	Passive

AMD GPU

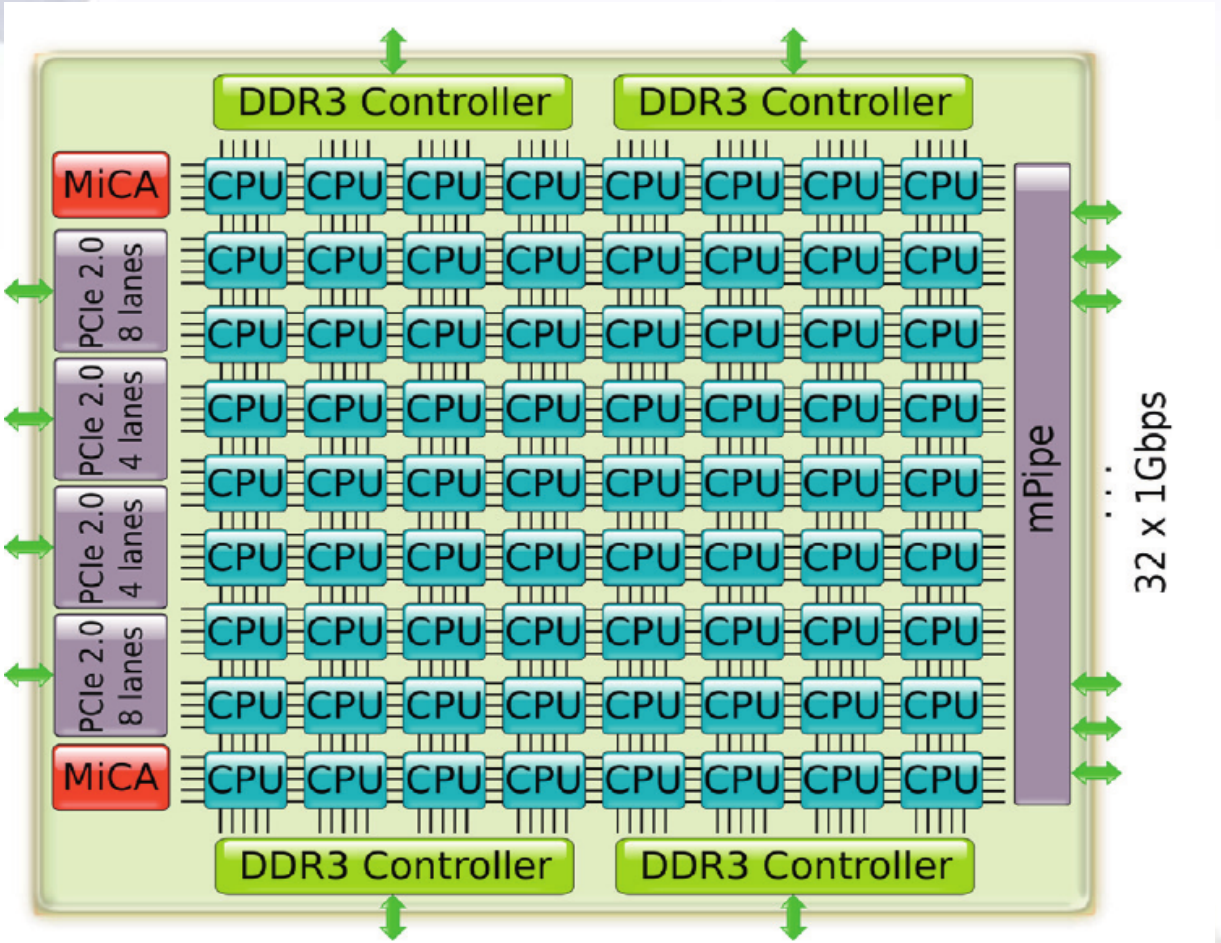
- AMD's APU chips implement the Heterogeneous System Architecture (HSA).
- The significant of AMD GPU is the unification of the memory spaces of the CPU and GPU cores. This means that there is no communication overhead associated with assigning workload to the GPU cores, nor any delay in getting the results back.
- This also removes one of the major hassles in GPU programming, which is the explicit (or implicit, based on the middleware available) data transfers that need to take place.



- The HSA architecture identifies two core types:
- The Latency Compute Unit (LCU), which is a generalization of a CPU. A LCU supports both its native CPU instruction set and the HSA intermediate language (HSAIL) instruction set.
- The Throughput Compute Unit (TCU), which is a generalization of a GPU. A TCU supports only the HSAIL instruction set. TCUs target efficient parallel execution.

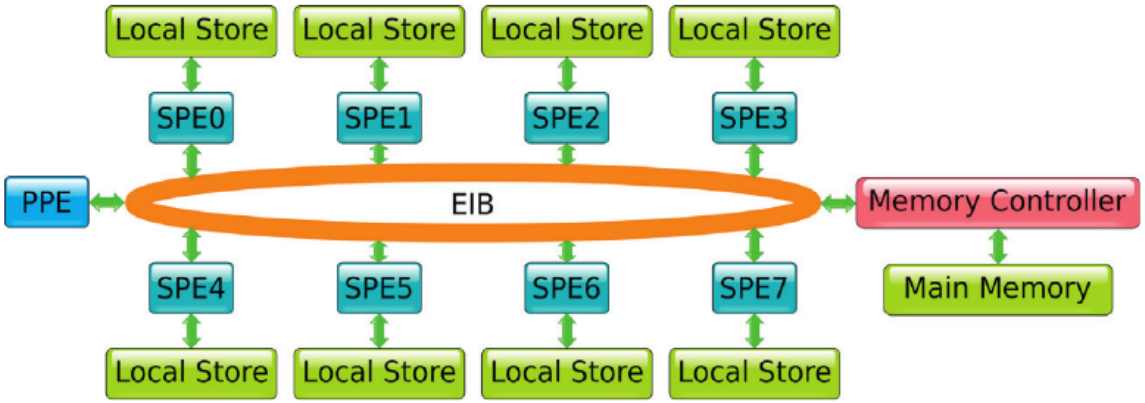


TILERA'S TILE-GX8072



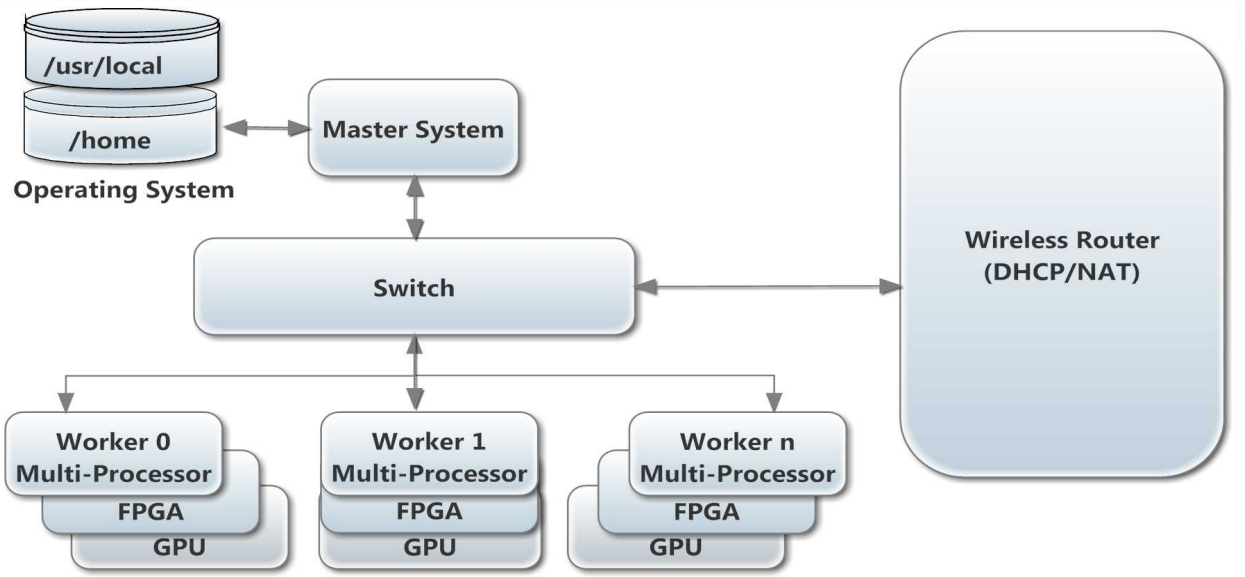
Power PC

- Master Core: 64-bit PowerPC core also called the Power Processing Element.
- Worker Core: Synergistic Processing Element SPE having 128-bit vector processors.
- Own SIMD instruction set.

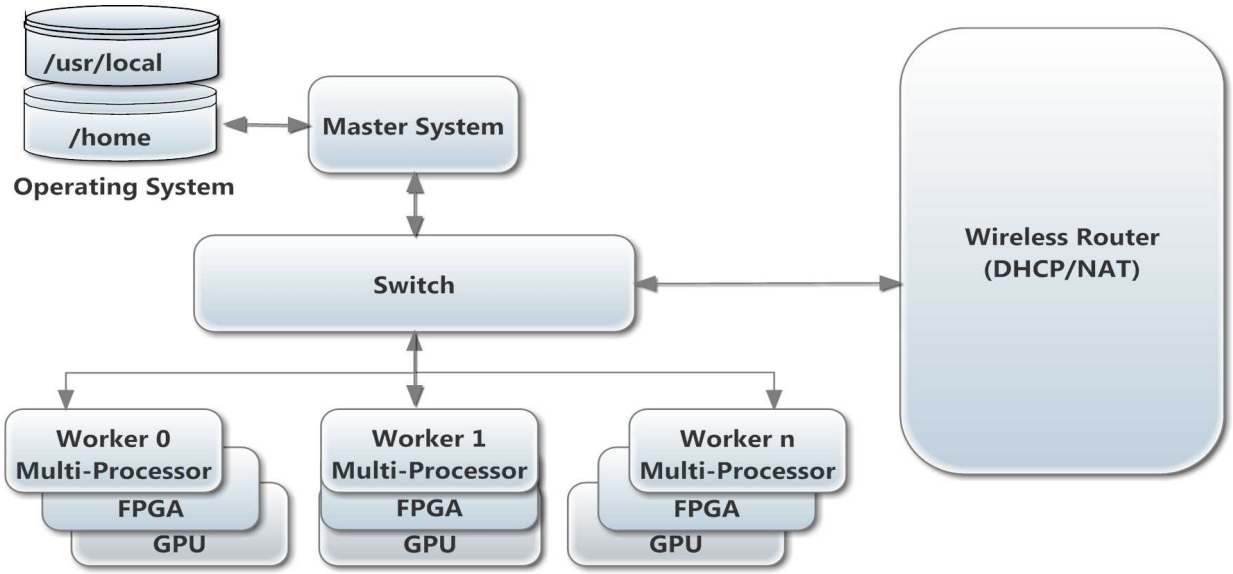


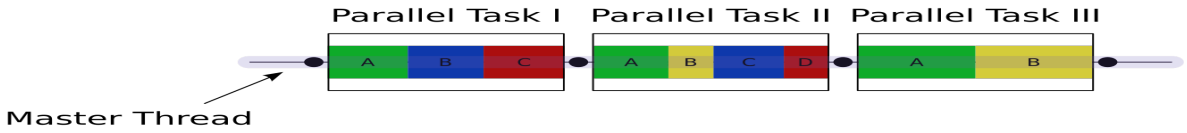
- Problem Statement
- HPC System
- **Supercomputing System**
- High Performance Applications
- Supercomputer Design
- Conclusion
- Hands on Experience

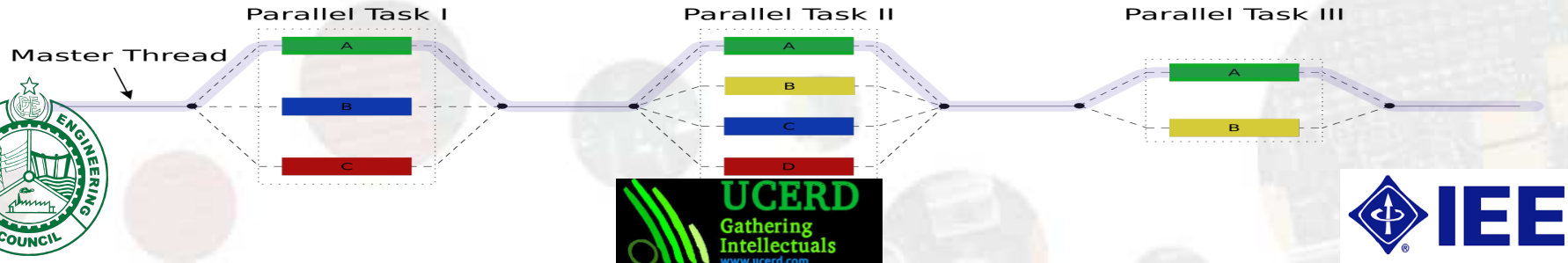
Basic Introduction of Supercomputing



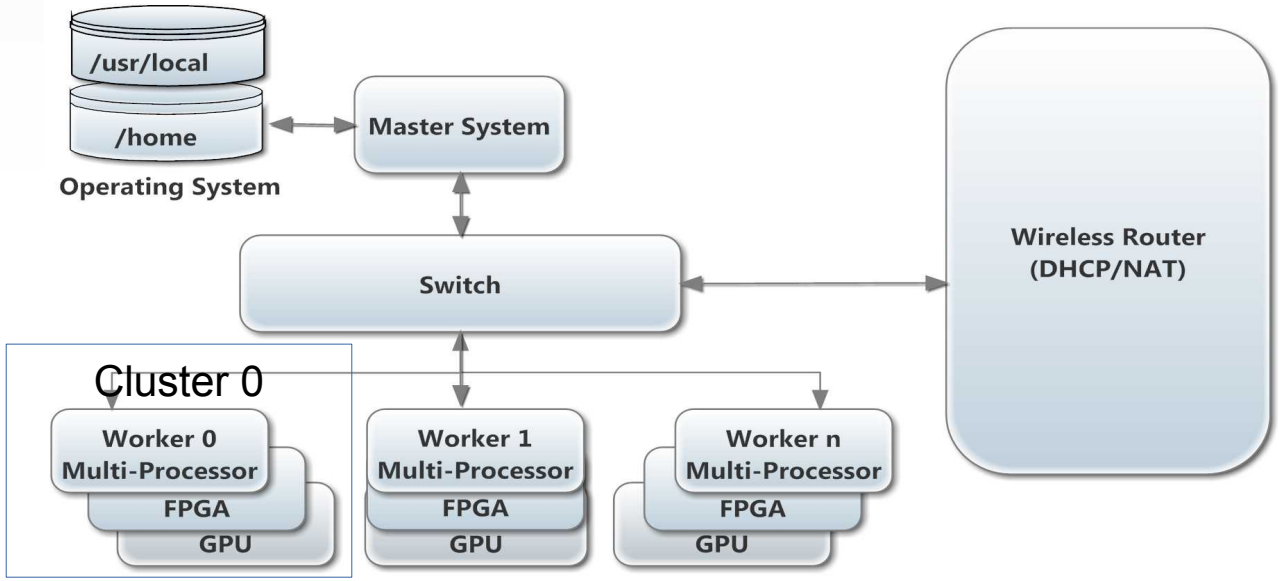
Basic Introduction of Supercomputing

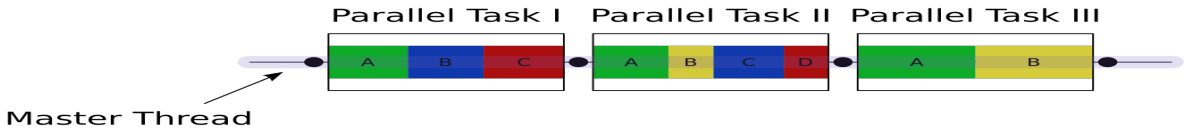


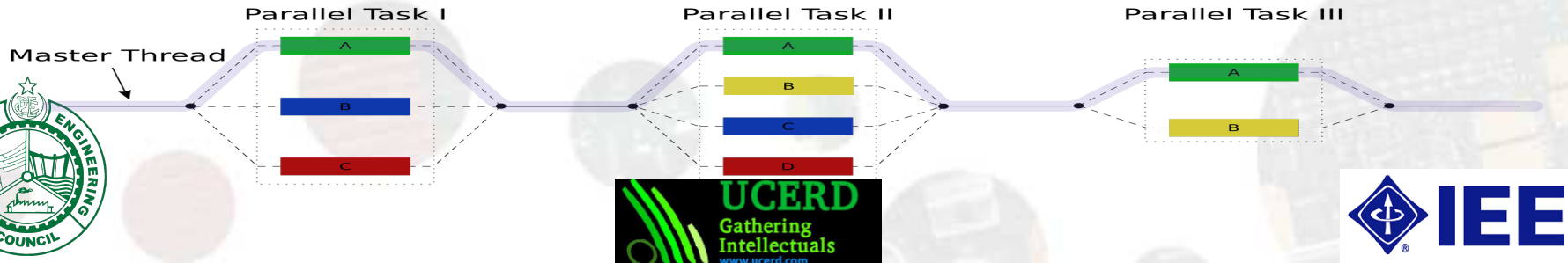




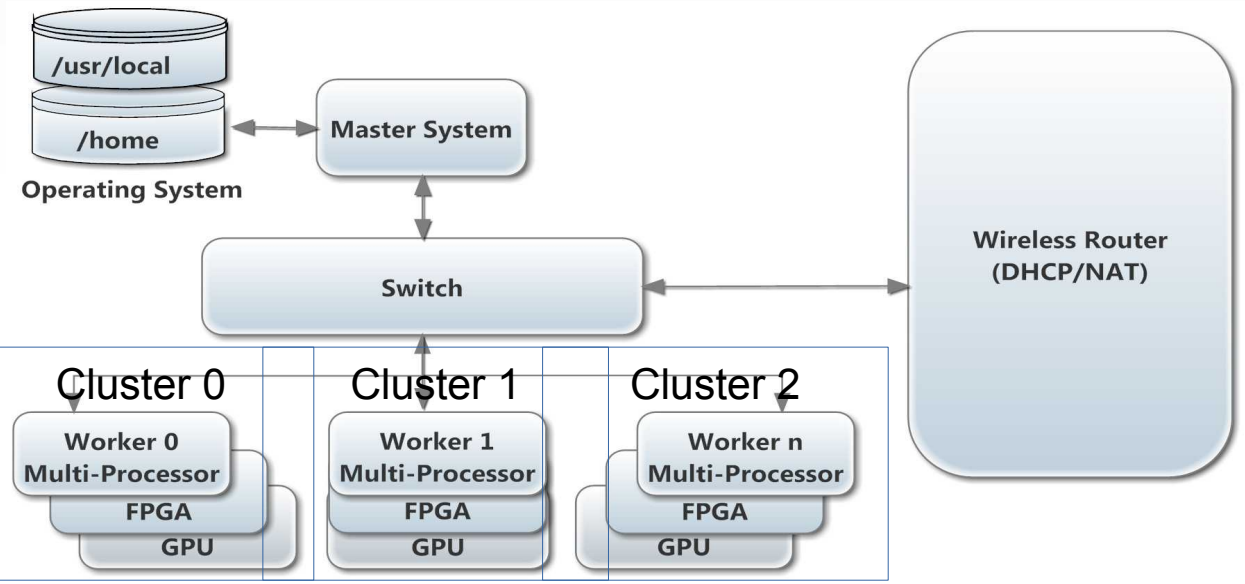
Basic Introduction of Supercomputing

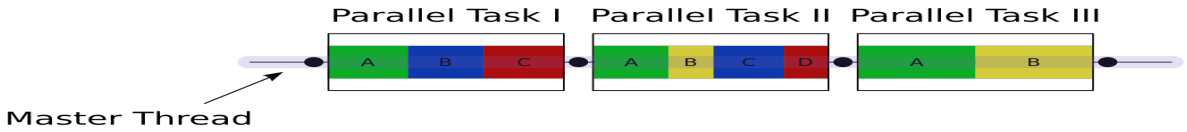


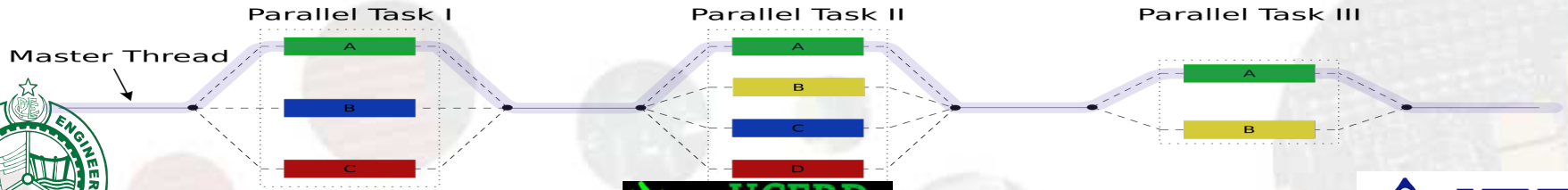




Basic Introduction of Supercomputing

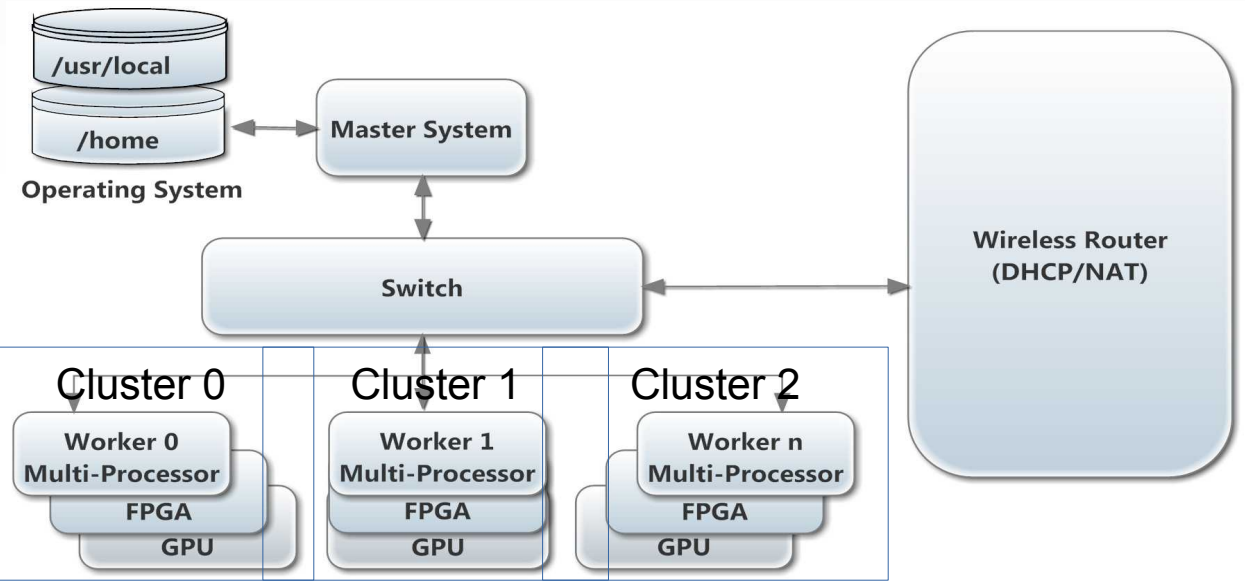


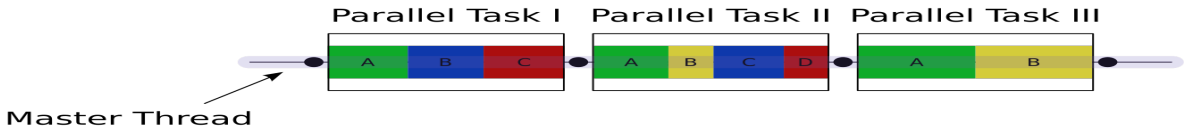


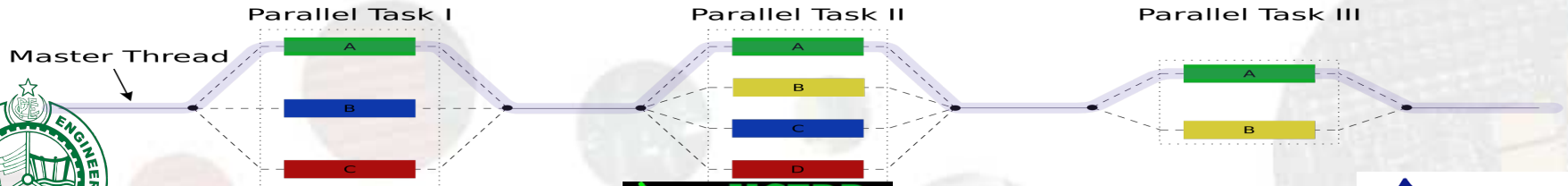


Basic Introduction of Supercomputing

- Performance
- Programmability
- Portability
- Scalability
- Accessibility
- Power
- Cost

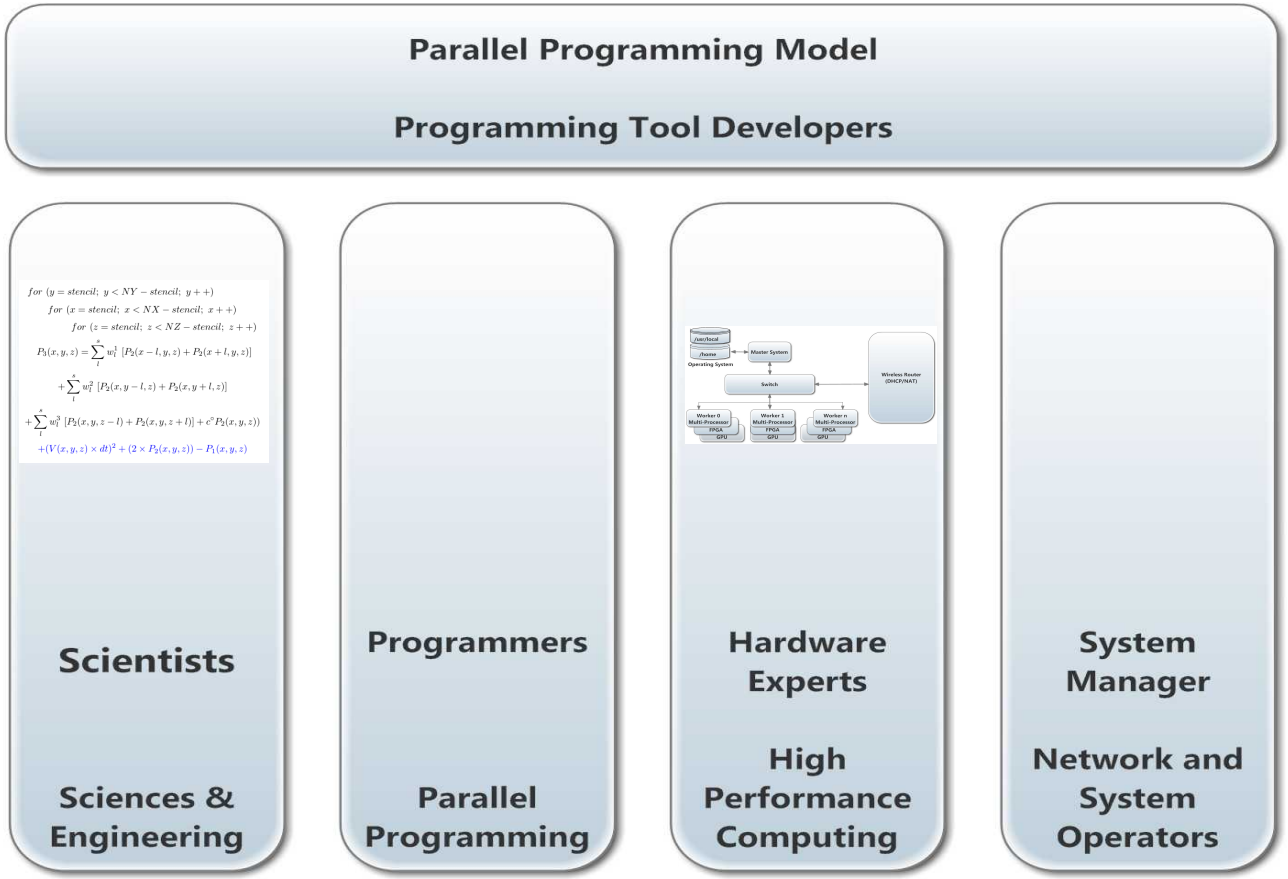






Supercomputing System

- Scientist Community
- Hardware Expert
- Tool Developer
- Programmer
- System Manager



Scientist Community

Life Science

- e.g Medicine, Biology etc.

Earth Science

- e.g. Environment Studies etc.

Engineering

- e.g. Aerospace, Automotive industry, Biomechanics
- and biomedical research etc.

Targeted Algorithms

- Compute Intensive
- Data Intensive
  - Linear, Differential Equations and Filters for Quick Analysis, Observation and Decision.

Scientist Community

$$for\ (y = stencil;\ y < NY - stencil;\ y++)$$
$$for\ (x = stencil;\ x < NX - stencil;\ x++)$$
$$for\ (z = stencil;\ z < NZ - stencil;\ z++)$$
$$P_3(x,y,z) = \sum_{l}^{s} w_l^1\ [P_2(x-l,y,z) + P_2(x+l,y,z)]$$
$$+ \sum_{l}^{s} w_l^2\ [P_2(x,y-l,z) + P_2(x,y+l,z)]$$
$$+ \sum_{l}^{s} w_l^3\ [P_2(x,y,z-l) + P_2(x,y,z+l)] + c^{\circ} P_2(x,y,z))$$
$$+(V(x,y,z) \times dt)^2 + (2 \times P_2(x,y,z)) - P_1(x,y,z)$$

3D Differential Equation

Life Science

- e.g Medicine, Biology etc.

Earth Science

- e.g. Environment Studies etc.

Engineering

- e.g. Aerospace, Automotive industry, Biomechanics
- and biomedical research etc.

Targeted Algorithms

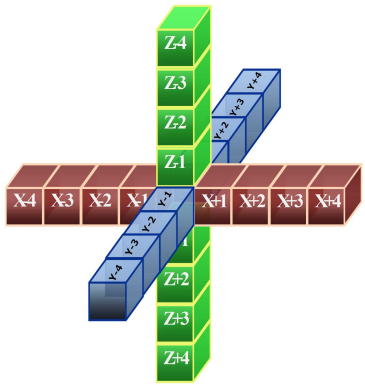
- Compute Intensive
- Data Intensive

- Linear, Differential Equations and Filters for Quick Analysis, Observation and Decision.

Scientist Community

$$for\ (y = stencil;\ y < NY - stencil;\ y++)$$
$$for\ (x = stencil;\ x < NX - stencil;\ x++)$$
$$for\ (z = stencil;\ z < NZ - stencil;\ z++)$$
$$P_3(x,y,z) = \sum_l^s w_l^1\ [P_2(x-l,y,z) + P_2(x+l,y,z)]$$
$$+ \sum_l^s w_l^2\ [P_2(x,y-l,z) + P_2(x,y+l,z)]$$
$$+ \sum_l^s w_l^3\ [P_2(x,y,z-l) + P_2(x,y,z+l)] + c^{\circ} P_2(x,y,z))$$
$$+(V(x,y,z) \times dt)^2 + (2 \times P_2(x,y,z)) - P_1(x,y,z)$$

3D Differential Equation



Life Science

- e.g Medicine, Biology etc.

Earth Science

- e.g. Environment Studies etc.

Engineering

- e.g. Aerospace, Automotive industry, Biomechanics
- and biomedical research etc.

Targeted Algorithms

- Compute Intensive
- Data Intensive

- Linear, Differential Equations and Filters for Quick Analysis, Observation and Decision.

PAKISTAN ENGINEERING COUNCIL

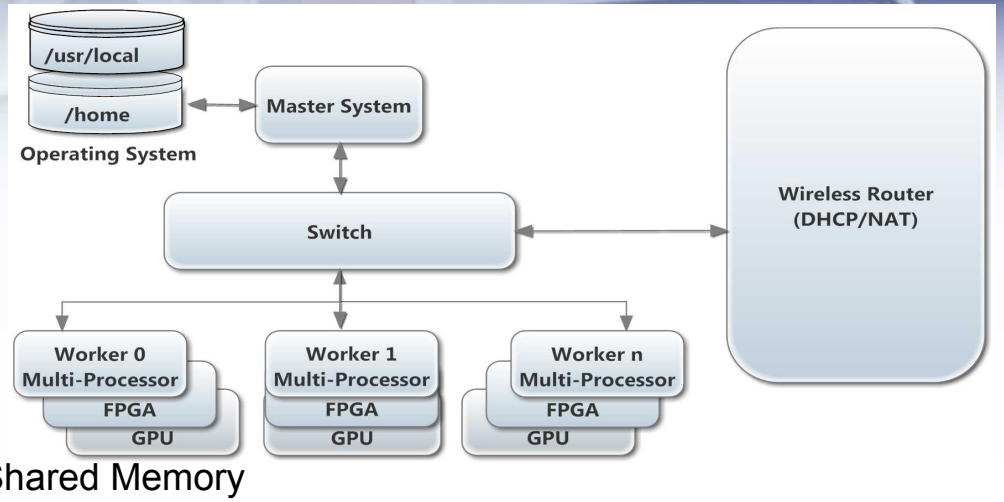
UCERD Gathering Intellectuals www.ucerd.com

IEEE

Hardware Expert

- **SISD**
  - **CPU (RISC)**
    - **Intel, AMD, ARM**
- **SIMD**
  - **GPU – Vector Processor**
    - **Nvidia, AMD, Intel**
- **MIMD**
  - **FPGA (HDL, HLS)**
    - **Xilinx, Altera**
  - **Multi-Core (MTMD)**

Hardware Expert



- **SISD**
  - **CPU (RISC)**
    - Intel, AMD, ARM
- **SIMD**
  - **GPU – Vector Processor**
    - Nvidia, AMD, Intel
- **MIMD**
  - **FPGA (HDL, HLS)**
    - Xilinx, Altera
  - **Multi-Core (MTMD)**

Performance Measurement

FLOPS = **Fl**oating **P**oint **O**peration Per **S**econd

KFLOPS = 10^3

= One Thousand Computation Per Second

= 12.00 * 1212.222 * 21212 + 232323 .....

MFLOPS= 10^6 Million Computation Per Second

GFLOPS= 10^9 Billion

TFLOPS= 10^{12} Trillion

PFLOPS= 10^{15} Quadrillion

EFLOPS= 10^{18} Quintillion

ZFLOPS = 10^{21} Sextillion

Tool Developer

- Software solutions for hardware architectures and application requirements.
- Operating System supports, in a scalable manner, hundreds of heterogeneous processor cores.
- Parallel Programming Models and Software Packages.

Tool Developer: **Programming Models**

- The applications can target **SISD, SIMD, and MIMD** architectures, executing the sequential parts on the SISD and compute intensive parts of the SIMD or MIMD cores.
- Supercomputing system architecture requires a practical programming model that facilitates parallel implementation and supports proper management of data delivery to the processing nodes.
- The demand for supporting joint CPU-GPU-FPGA execution is reflected in more recent programming models such as **OpenCL, OpenACC, and C++ AMP.**

Programmers

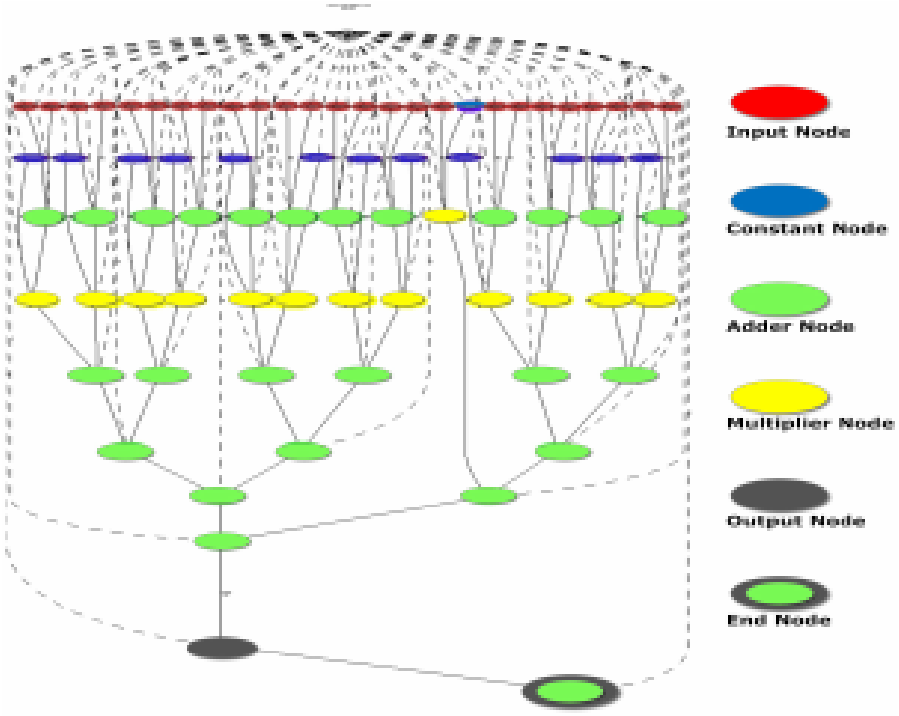
- Sequential application program and converts into parallel program.
- Understand Algorithm/Application data access, data structure, data dependencies and CFG.

$$for\ (y = stencil;\ y < NY - stencil;\ y++)$$
$$for\ (x = stencil;\ x < NX - stencil;\ x++)$$
$$for\ (z = stencil;\ z < NZ - stencil;\ z++)$$
$$P_3(x,y,z) = \sum_{l}^{s} w_l^1\ [P_2(x-l,y,z) + P_2(x+l,y,z)]$$
$$+ \sum_{l}^{s} w_l^2\ [P_2(x,y-l,z) + P_2(x,y+l,z)]$$
$$+ \sum_{l}^{s} w_l^3\ [P_2(x,y,z-l) + P_2(x,y,z+l)] + c^{\circ} P_2(x,y,z))$$
$$+(V(x,y,z) \times dt)^2 + (2 \times P_2(x,y,z)) - P_1(x,y,z)$$

Mathematical Model

Programmers

- Sequential application program and converts into parallel program.
- Understand Algorithm/Application data access, data structure, data dependencies and CFG.



CDFG

Programmers

- Sequential application program and converts into parallel program.
- Understand Algorithm/Application data access, data structure, data dependencies and CFG.

```
#define MX 64
#define MY 64
#define MZ 64
for ( k = Stencil ; k < MY - Stencil ; k++ )
for ( j = Stencil ; j < MZ - Stencil ; j++ )
for ( i = Stencil ; i < MX - Stencil ; i++ )
{
iter = k*(MX*MZ) + (j*MX) + i;
tmp =
Y1*(P2_linear[i+j*iter_j+(k-1)*iter_k] + P2_linear[i+j*iter_j+(k+1)*iter_k]) +
Y2*(P2_linear[i+j*iter_j+(k-2)*iter_k] + P2_linear[i+j*iter_j+(k+2)*iter_k]) +
Y3*(P2_linear[i+j*iter_j+(k-3)*iter_k] + P2_linear[i+j*iter_j+(k+3)*iter_k]) +
Y4*(P2_linear[i+j*iter_j+(k-4)*iter_k] + P2_linear[i+j*iter_j+(k+4)*iter_k]) +
c00 * P2_linear[iter] +
X4*(P2_linear[i+(j-4)*iter_j+k*iter_k] + P2_linear[i+(j+4)*iter_j+k*iter_k]) +
X3*(P2_linear[i+(j-3)*iter_j+k*iter_k] + P2_linear[i+(j+3)*iter_j+k*iter_k]) +
X2*(P2_linear[i+(j-2)*iter_j+k*iter_k] + P2_linear[i+(j+2)*iter_j+k*iter_k]) +
X1*(P2_linear[i+(j-1)*iter_j+k*iter_k] + P2_linear[i+(j+1)*iter_j+k*iter_k]) +
Z4*(P2_linear[(i-4)+j*iter_j+k*iter_k] + P2_linear[(i+4)+j*iter_j+k*iter_k]) +
Z3*(P2_linear[(i-3)+j*iter_j+k*iter_k] + P2_linear[(i+3)+j*iter_j+k*iter_k]) +
Z2*(P2_linear[(i-2)+j*iter_j+k*iter_k] + P2_linear[(i+2)+j*iter_j+k*iter_k]) +
Z1*(P2_linear[(i-1)+j*iter_j+k*iter_k] + P2_linear[(i+1)+j*iter_j+k*iter_k]);
P3_linear[iter] = tmp ;
}
```

C/C++ Program

System Manager

Following teams and experts are required to deal and manage the HPC system.

- Network Configuration
- Power Management
- Software Installation
- Hardware Maintenances

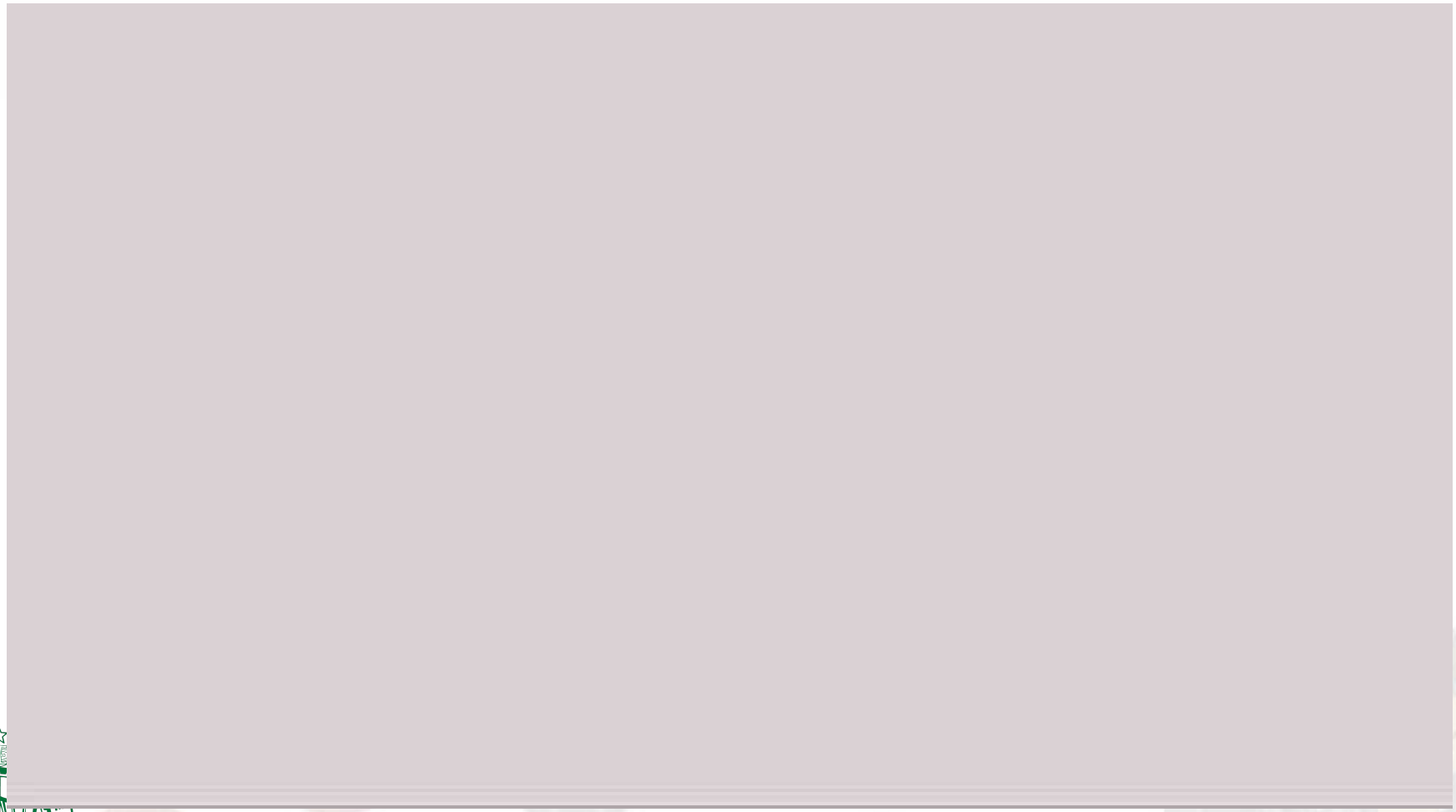
- Problem Statement
- HPC System
- Supercomputing System
- **High Performance Applications**
- Supercomputer Design
- Conclusion
- Hands on Experience

HPC Applications

- Compute Intensive
- Data Intensive
- Complex

- Simulation
  - Analysis
- Modeling
- Artificial Intelligence

An HPC Simulation tool for Scientists and Engineers



- Problem Statement
- HPC System
- Supercomputing System
- High Performance Applications
- **Supercomputer Design**
- Conclusion
- Hands on Experience

Supercomputer Architectures and Artificial Intelligence Frameworks

- Supercomputer Architectures
  - CPU based
  - CPU-GPU based
  - Embedded CPU-GPU-FPGA based
- Software and Package Installation
- Supercomputer configuration
- Application programming models
- Deep Learning Frameworks

CPU Based

MareNostrum III 1.1 PetaFLOPS @ 1,015 kW

37 iDataPlex compute racks.

Each one composed of:

– 84 IBM dx360 M4 compute nodes

  – 2x E5–2670 SandyBridge-EP 2.6GHz cache 20MB 8-core

  – 500GB 7200 rpm SATA II local HDD

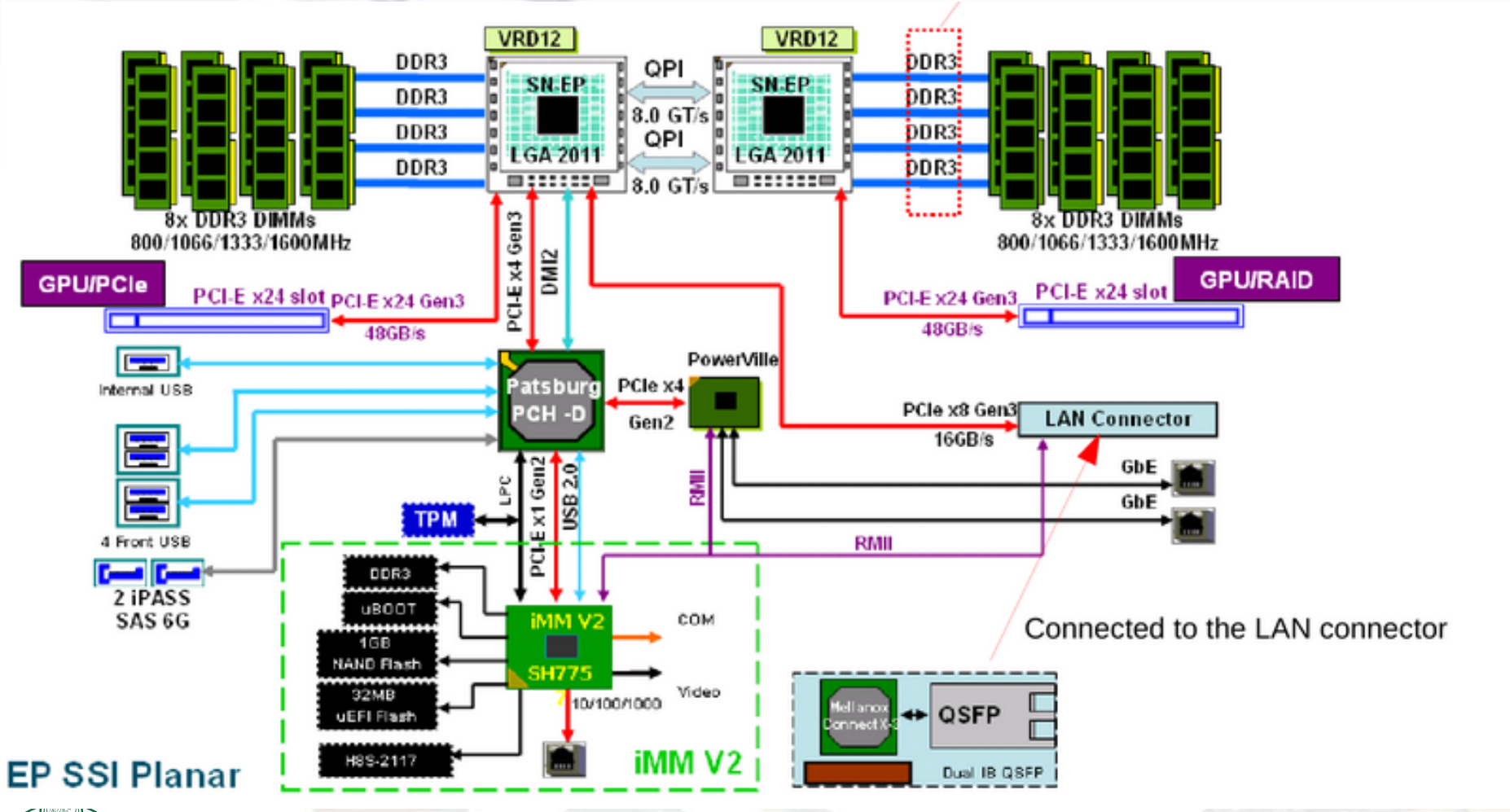
– 4 Mellanox 36-port Managed FDR10 IB Switches

– 2 BNT RackSwitch G8052F (Management Network)

– 2 BNT RackSwitch G8052F (GPFS Network)

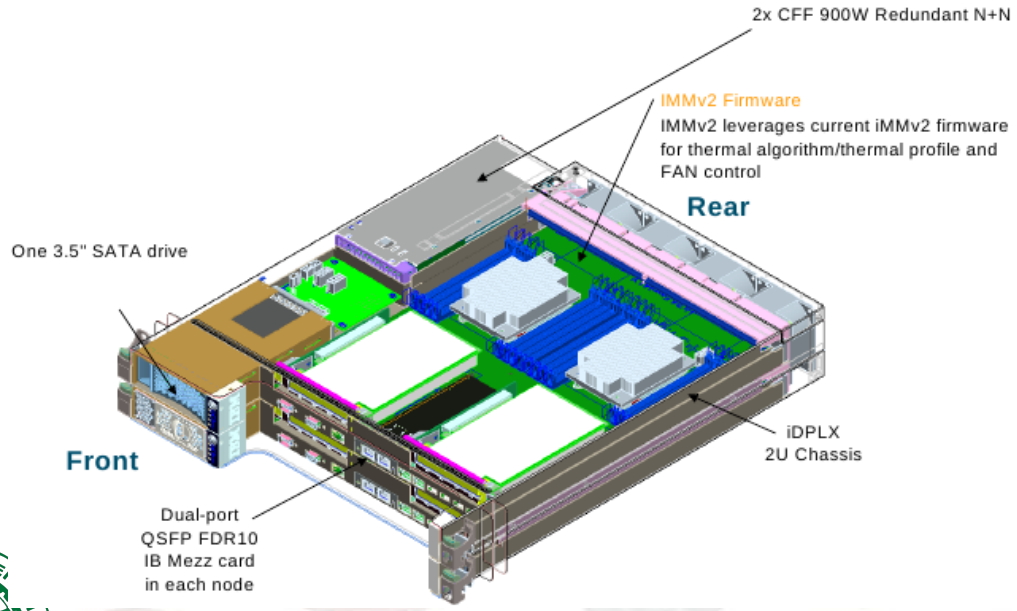
– 4 Power Distribution Units

Bottom-up Design Approach



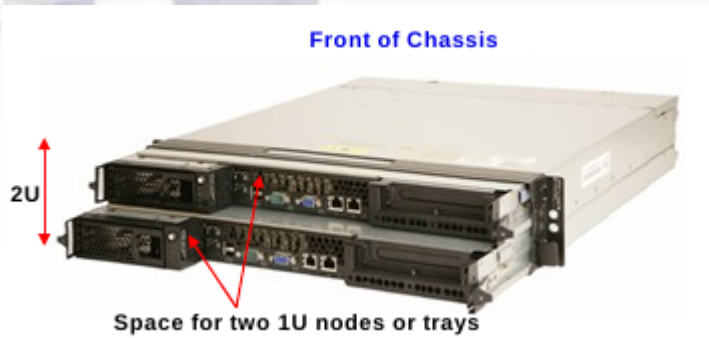
https://www.bsc.es/marenostrum-support-services/marenostrum-system-architecture/compute-rac

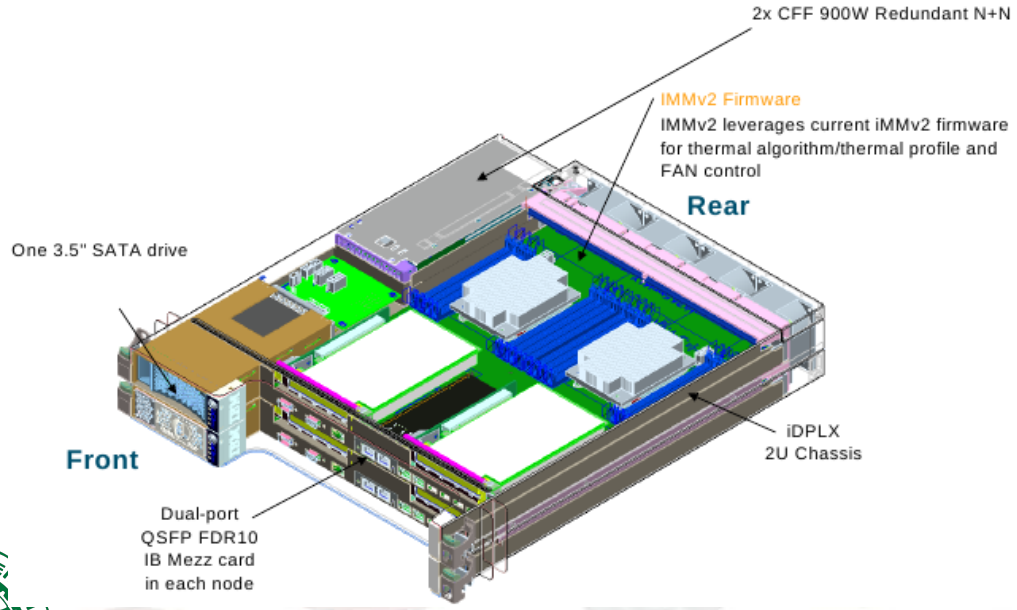
Bottom-up Design Approach



https://www.bsc.es/ma[illegible]t-services/marenostrum-system-architecture/compute-[illegible]

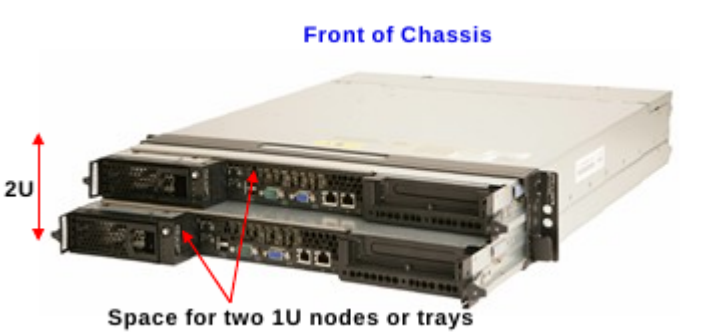
Bottom-up Design Approach

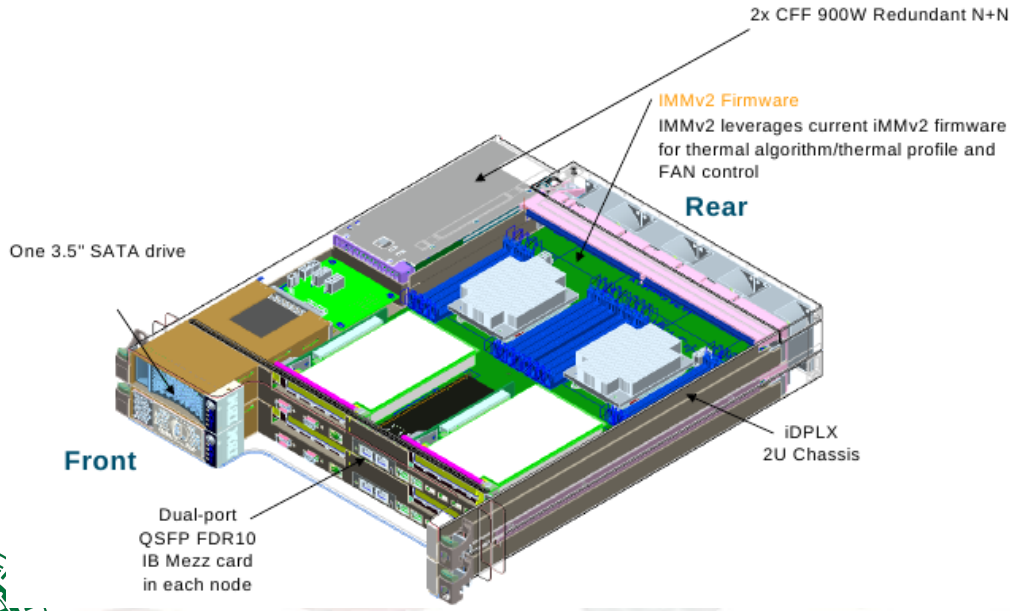


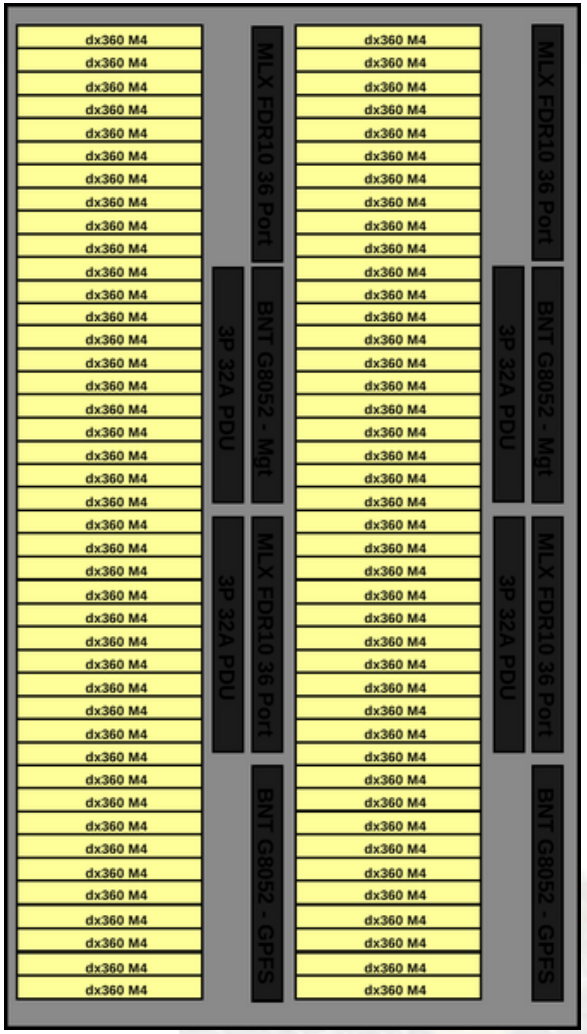


https://www.bsc.es/ma[illegible]t-services/marenostrum-system-architecture/compute-[illegible]

Bottom-up Design Approach

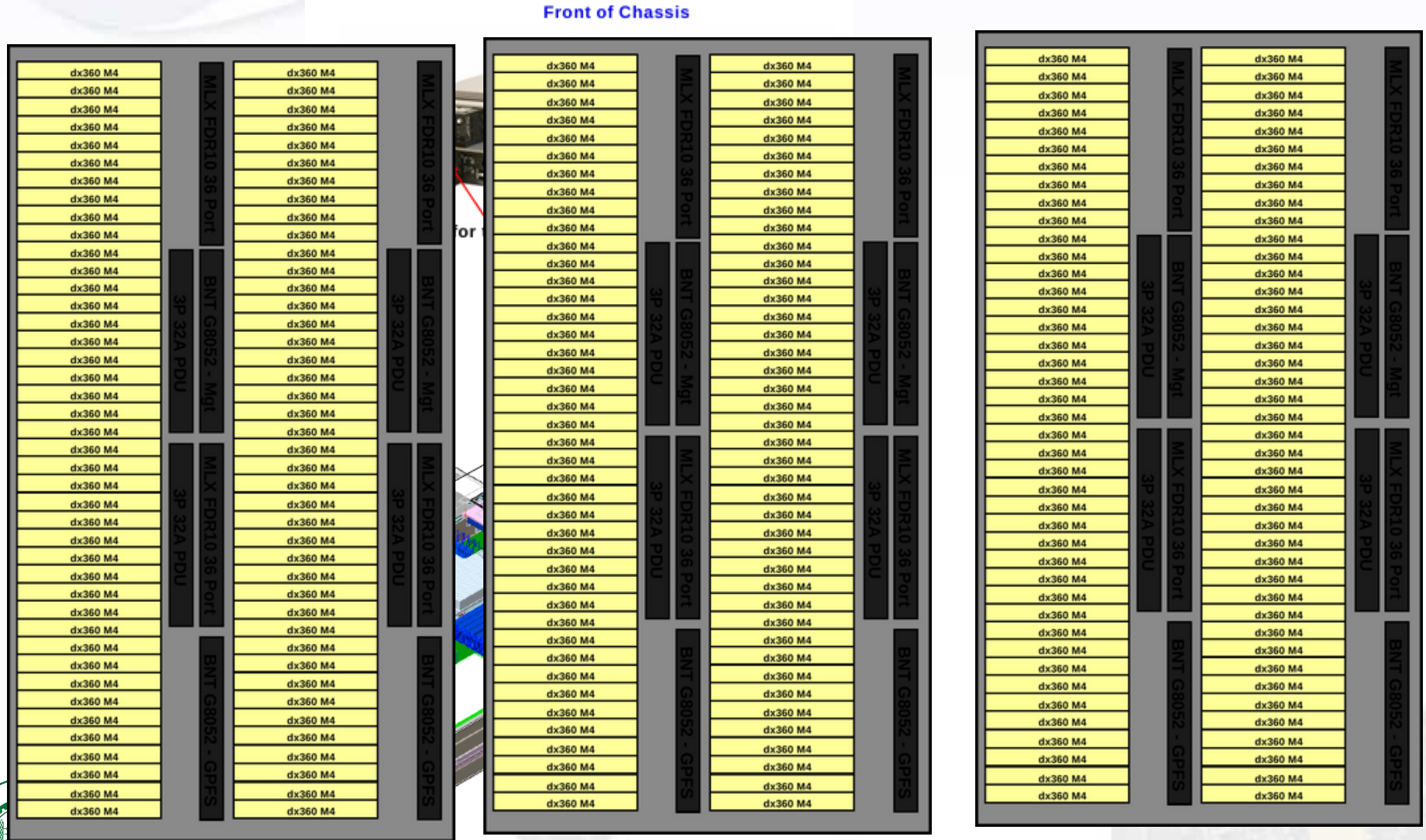






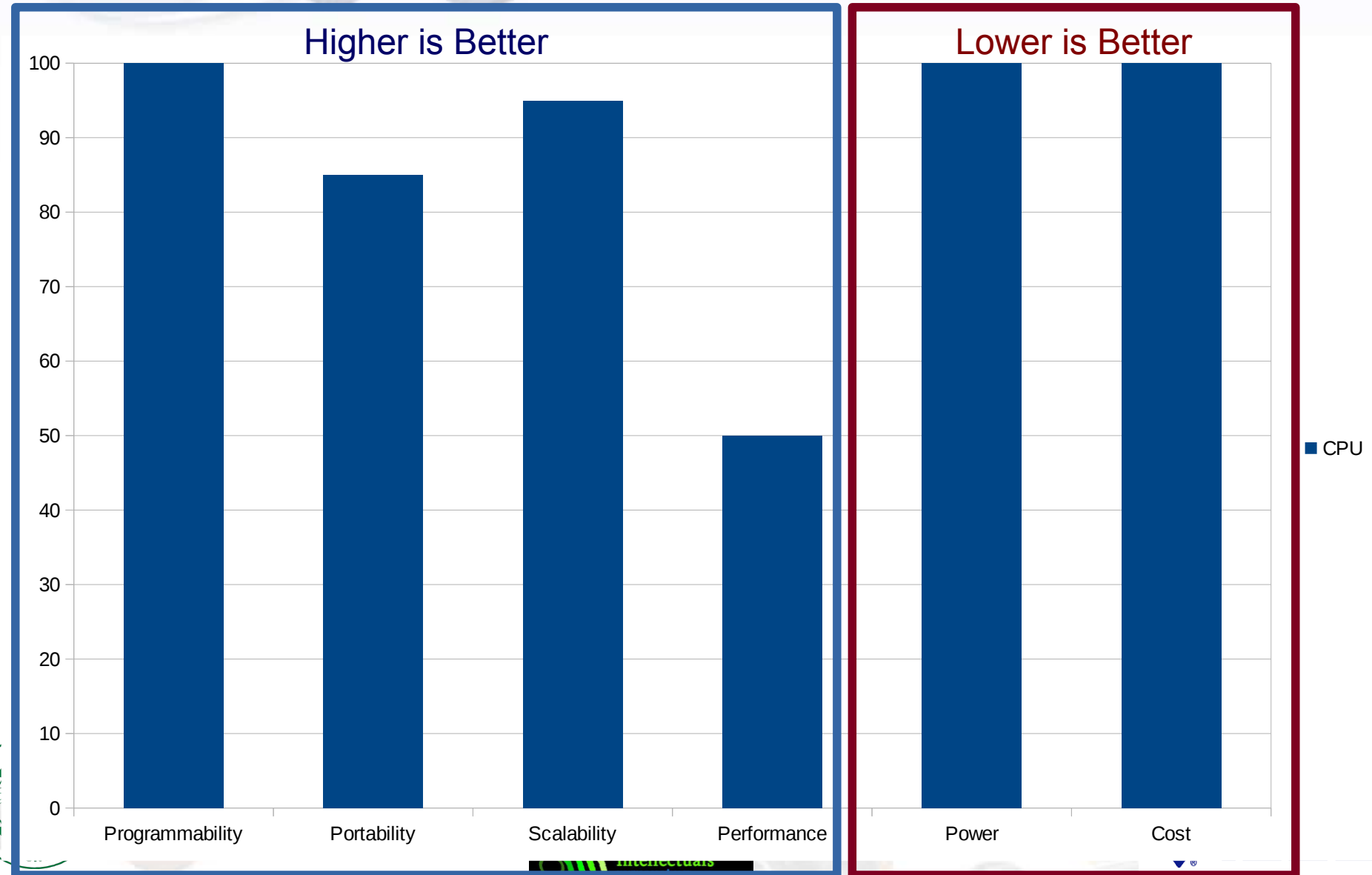
https://www.bsc.es/ma[illegible]t-services/marenostrum-system-architecture/compute-[illegible]

Bottom-up Design Approach



https://www.bsc.es/ma[illegible]t-services/marenostrum-system-architecture/compute-[illegible]

Features Comparison: CPU based



CPU-GPU based

MinoTauro – Peak Performance: 250.94 TFlops

39 bullx R421-E4 servers, each server with:

– 2 Intel Xeon E5–2630 v3 (Haswell) 8-core processors, (each core at 2.4 GHz,and with 20 MB L3 cache)

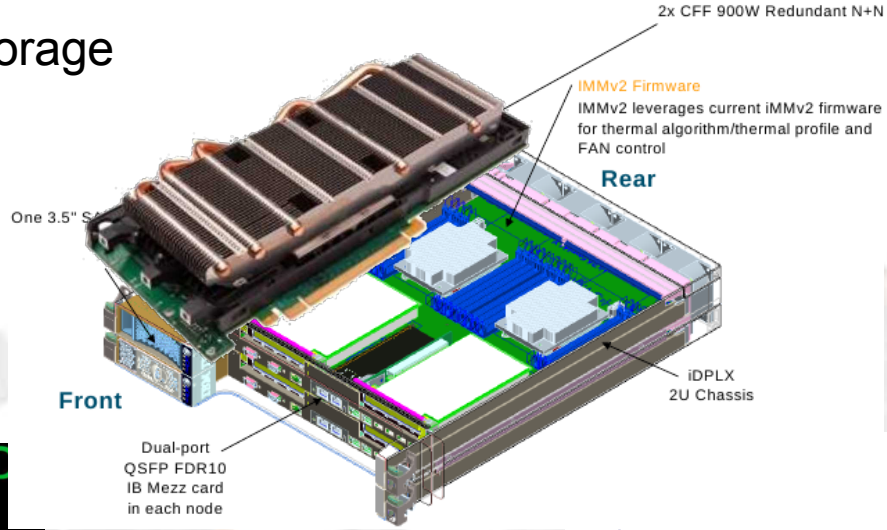
– 2 K80 NVIDIA GPU Cards

– 128 GB of Main memory, distributed in 8 DIMMs of 16 GB – DDR4 @ 2133 MHz – ECC SDRAM

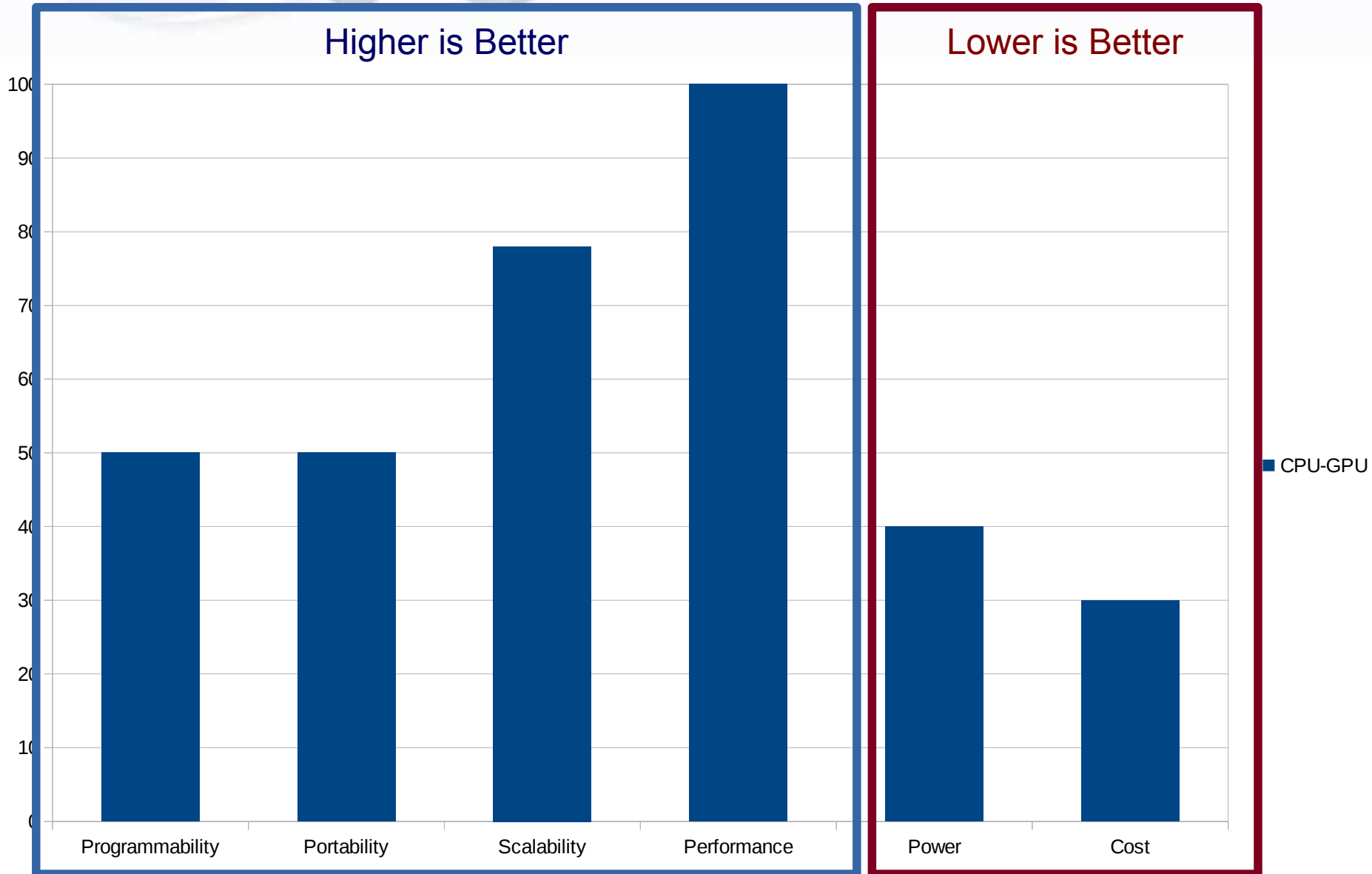
– 120 GB SSD (Solid State Disk) as local storage

– 1 PCIe 3.0 x8 8GT/s, Mellanox ConnectX

– 4 Gigabit Ethernet ports.



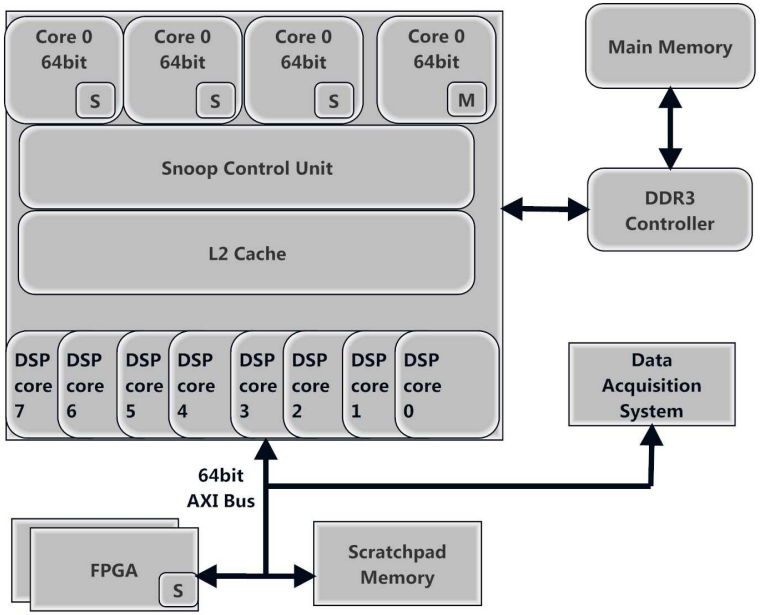
Features Comparison



Embedded CPU-GPU-FPGA based

ARM Multi-Core, Parallela

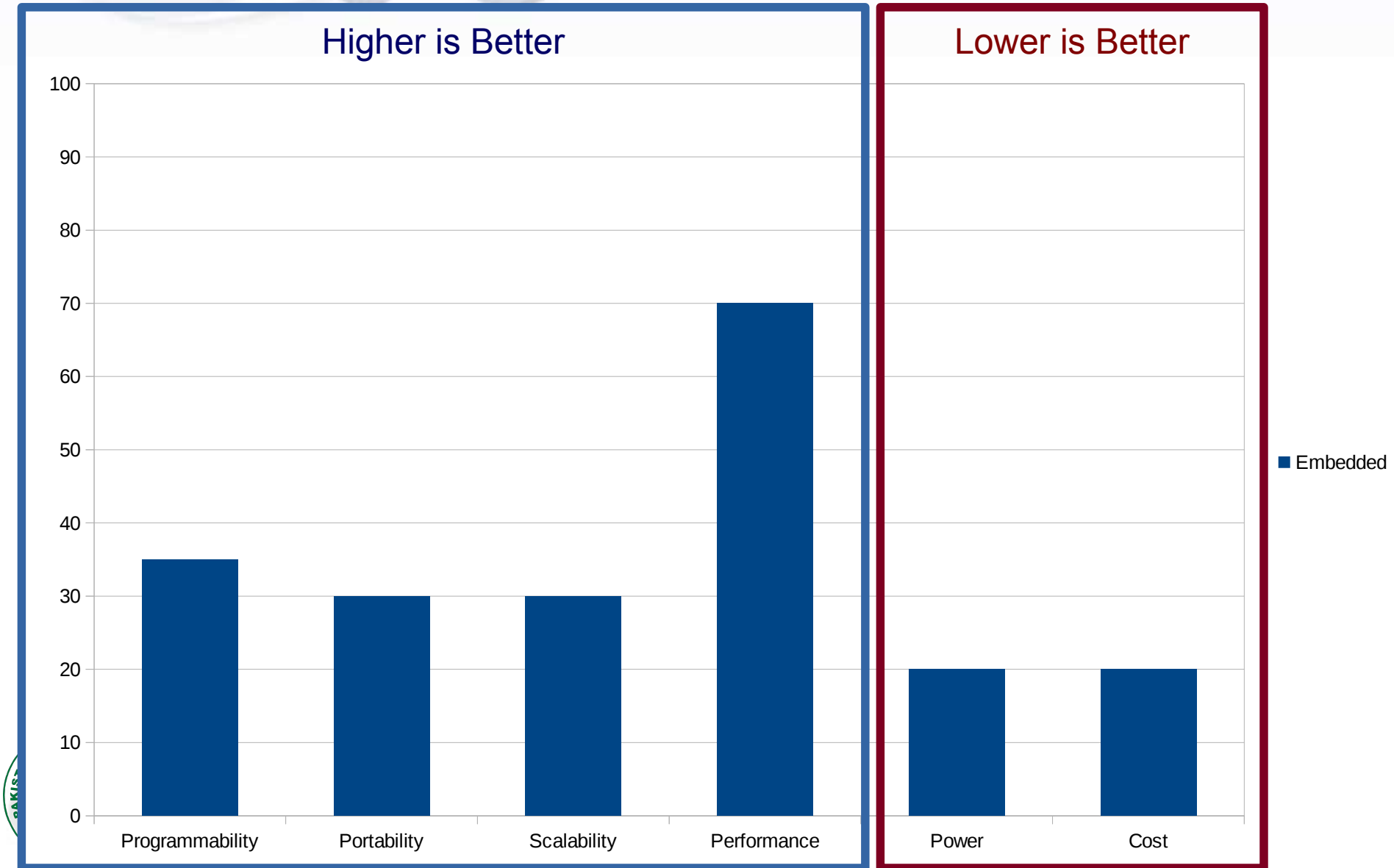
Peak-performance 100 Giga FLOPS to 1.0 Tera FLOPS



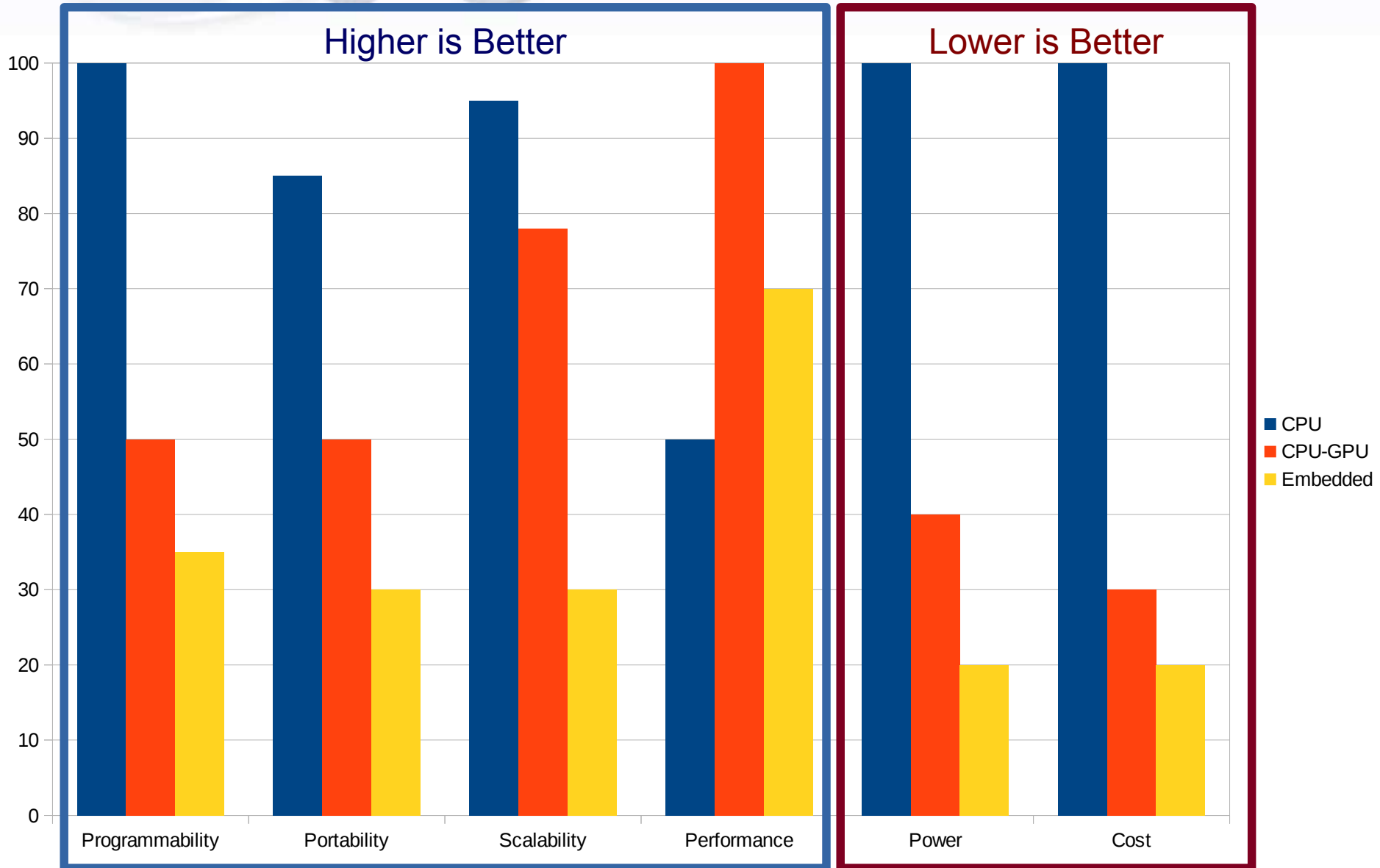
Single Board HPC Computer

IEEE

Features Comparison

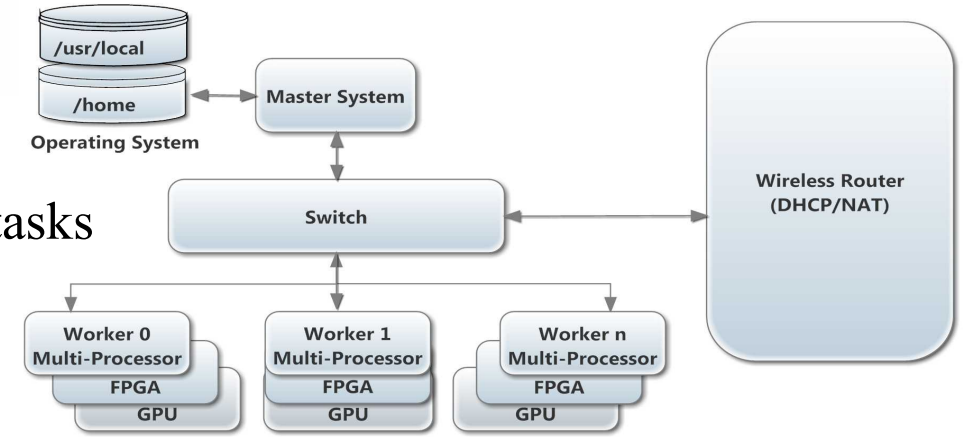


Comparison


Higher is Better
Lower is Better
100
90
80
70
60
50
40
30
20
10
0
Programmability
Portability
Scalability
Performance
Power
Cost
CPU
CPU-GPU
Embedded

Supercomputer Hardware

- Master Node
  - Intel Xeon Core-i7 CPU
  - Control and communicate, divide tasks among compute-nodes
- Compute-nodes
  - Multiple CPUs, GPUs, FPGAs
  - Execute compute task of an application
- Ethernet Switch
  - Master and Compute-nodes connected through Ethernet cable or Infiniband cable

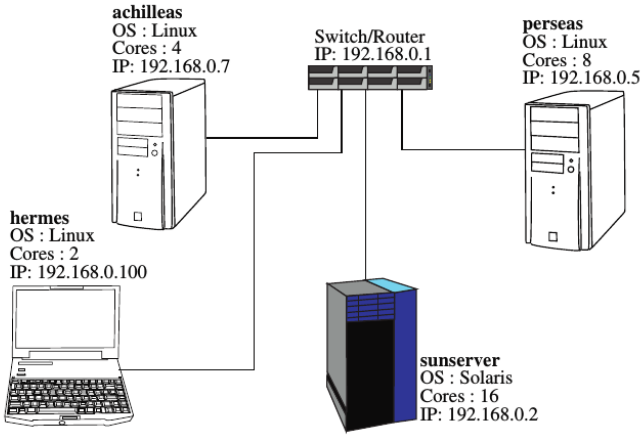


Software and Packages Installation

- Operating System
  - Linux based OS installed on master as well compute-nodes
  - Resource management
  - Scheduling of tasks running on compute nodes
  - Load and balancing
  - I/Os
- SSH Server
  - Install SSH server(secure shell) on master node as well compute nodes
  - It is a protocol used to securely log onto remote systems using TCP/UDP
  - Distribute SSH keys on all compute nodes to connect with master with out using password
- NFS (Network File System)
  - Install NFS server on master and NFS client on compute nodes
  - NFS is using to run a parallel program across the compute nodes
  - NFS operate using TCP/UDP internet protocols

Configuration

- All nodes are assigned an IP address
- IP addresses are assigned using DHCP with static IP configuration
- Together with a DNS server, DDNS service is also enable to allow dynamic host name resolution for all compute nodes
- Example
  - 192.168.1.0/24 subnet
  - Specific IP addresses with in subnet are summarize in Table



Subnet 192.168.1.0 **Netmask** 255.255.255.0

IP range 192.168.1.1 to 192.168.1.255

IP Range	Purpose
192.168.1.1	Gateway
192.168.1.10	Switch
192.168.1.99	Master node
192.168.1.100 to 192.168.1.200	Reserved for compute nodes

Application programming models

- Used to overcome the complexity that is between hardware architecture and application software.
- MPI and MPICH
  - Install MPI (message passing interface) and MPICH libraries on both master and compute nodes
  - Utilizes TCP/IP along with some libraries
  - Create a C or C++ parallel program that can run on multiple nodes simultaneously
  - It provide fast node to node messaging passing protocols and daemon-based process startup/control for supercomputing functioning.

Conclusion

- HPC can ensure a **country's strength, improve its national security system, defense ability,** and promote the timely development of highly modern weaponry. **It is one of the most important measures of a country's overall prowess and economic strength.**
- In this presentation, **heterogeneous supercomputer systems are discussed for artificial intelligence applications.**
- The system architectures are equipped with state of the art **parallel programming models and heterogeneous hardware architecture** which facilitate **application programmer** to write **applications for life and earth sciences efficiently and easily.**

- Problem Formulation
- Proposal
- Supercomputing System
- High Performance Applications
- Supercomputer Architectures
- Conclusion
- **Hands on Experience**

Hands on Experience and Training

Supercomputer System Architectures

- CPU based 1.1 Peta FLOPS
- GPU based 250 Tera FLOPS
- Embedded 300 Giga FLOPS

Applications

- Artificial Intelligence using Deep Learning
- Computer Vision
- Analysis and Modeling using HPC Simulation Tools e.g. Ansys HPC.
- High Performance Embedded Systemats
- Single Board Computer

Supercomputing for High Performance Applications

Dr. Tassadaq Hussain

Assistant Professor Riphah International University

Collaborations:

Microsoft Research and Barcelona Supercomputing Center

Barcelona, Spain

UCERD Pvt Ltd Islamabad