An Introduction to High Performance Computing

Ashish P. Kuvelkar
Senior Director (HPC- Tech)
C-DAC, Pune

ACM Winter School 2019 on High Performance Computing (HPC)

Outline

- Introduction to HPC
- Architecting a HPC system
- Approach to Parallelization
- Parallelization Paradigm
- Applications in area of Science and Engineering

Parallel Computing Basics

Principle: Large problems can often be divided into smaller ones, which are then solved concurrently

Parallel Processing: Solving a task by making simultaneous use of multiple processing elements

Distributed Computing: Solving a task by simultaneous use of multiple processing elements, typically isolated and heterogeneous in nature

What is a HPC?

High Performance Computing

- Set of Computing technologies for very fast numeric simulation, modeling and data processing
- Employed for specialised applications that require lot of mathematical calculations
- Using computer power to execute a few applications extremely fast

What is HPC?(continued)

Definition 1

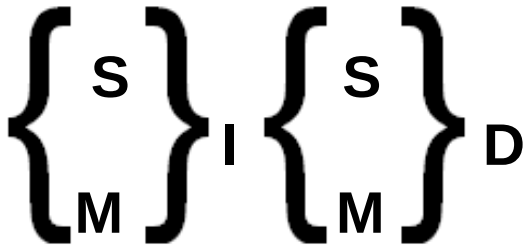
- High Performance Computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly.
- A supercomputer is a system that performs at or near the currently highest operational rate for computers.

Definition 2 (Wikipedia)

- High Performance Computing (HPC) uses Supercomputers and Computer Clusters to solve advanced computation problems.

Flynn's Taxonomy

In 1966, Michael J. Flynn proposed classification of computer architectures



Source: Wikipedia

- SI: Single Instruction –

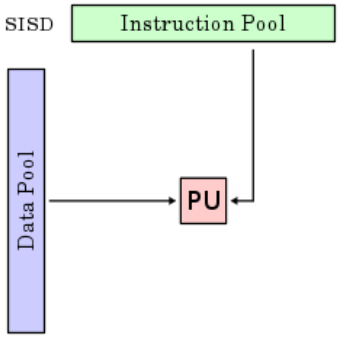
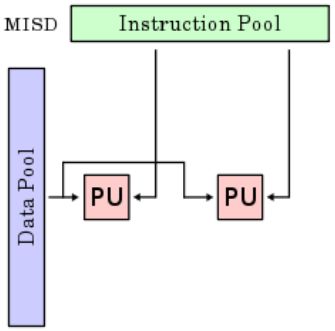
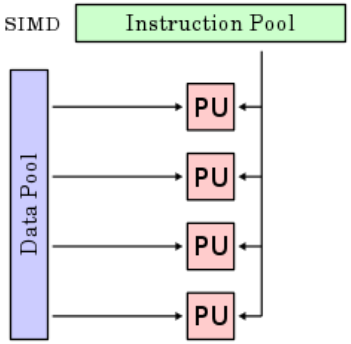
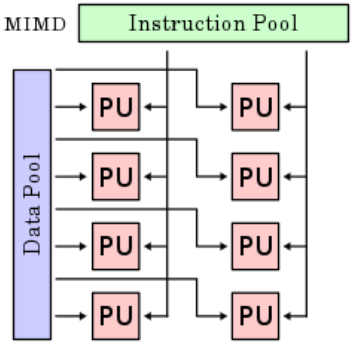
  All processors execute same Instruction

  MI: Multiple Instruction – Different processors may be executing different instructions
- SD: Single Data –

  All processors are operating on the same Data
- MD: Multiple Data-

  Different processors operating on different data

Flynn's Taxonomy

Instruction > Data v	Single	Multiple
Single	SISD	MISD
		
Multiple	SIMD	MIMD
		

Source: Wikipedia

Flynn's Taxonomy

SISD: A sequential computer which does not incorporate any parallelism

- Standard single CPU serial computer and program

SIMD: A computer which operates on multiple data streams using a single instruction stream

- The first use of SIMD was in vector supercomputers like Cray platforms

Flynn's Taxonomy

MISD: Multiple instructions operate on a single data stream.

- Rarely implemented architecture which is typically used for fault tolerance.

MIMD: Multiple independent processors concurrently executing different instructions on their respective data

- Most of the parallel computers are of this type

Evolution of Supercomputers

- Supercomputer in the 1980s and 90s
  - Custom-built computer systems
  - Very expensive

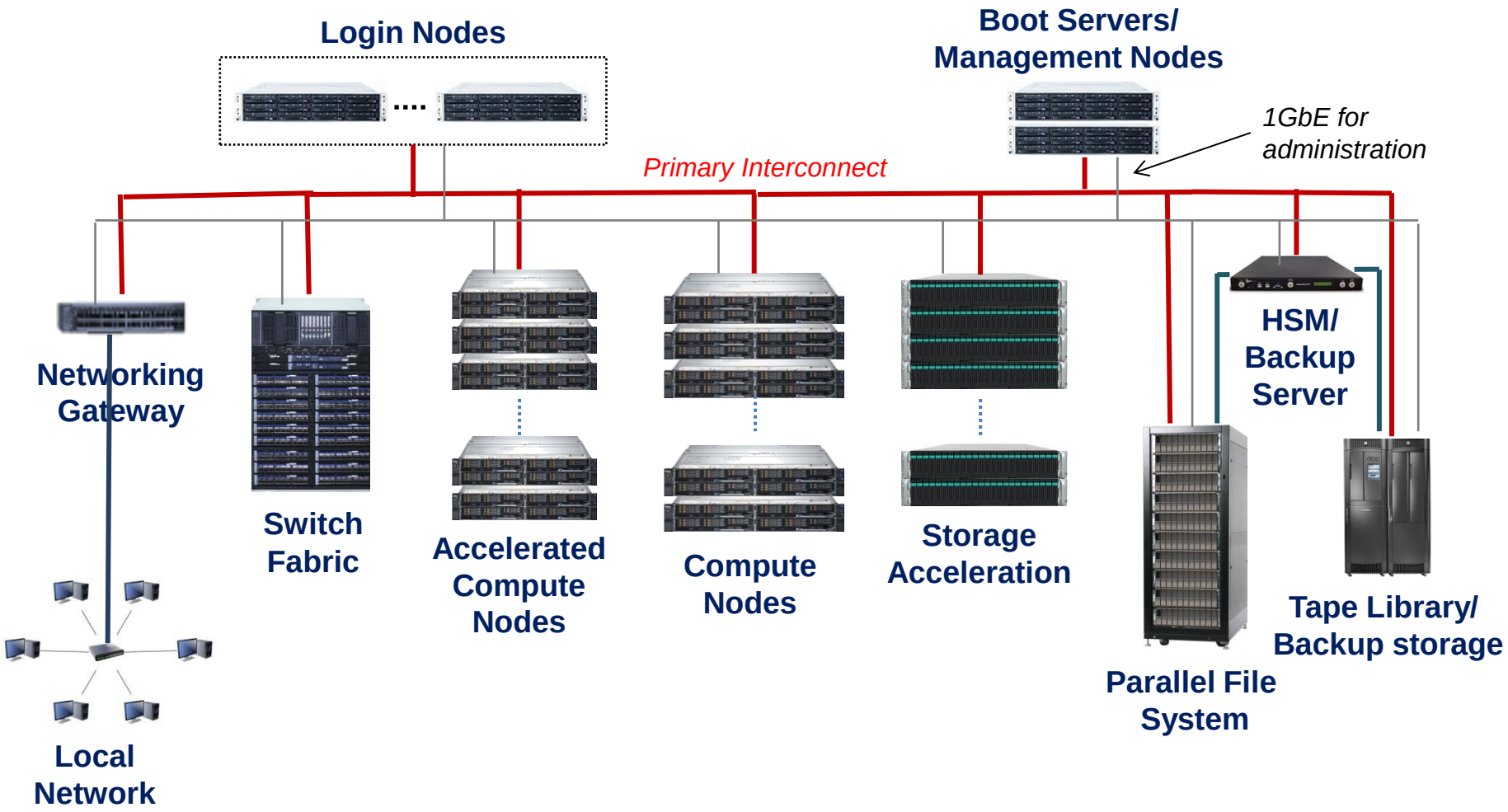
- Supercomputer after 1990s
  - Build using commodity off-the-shelf" components
  - Uses cluster computing techniques

Supercomputers

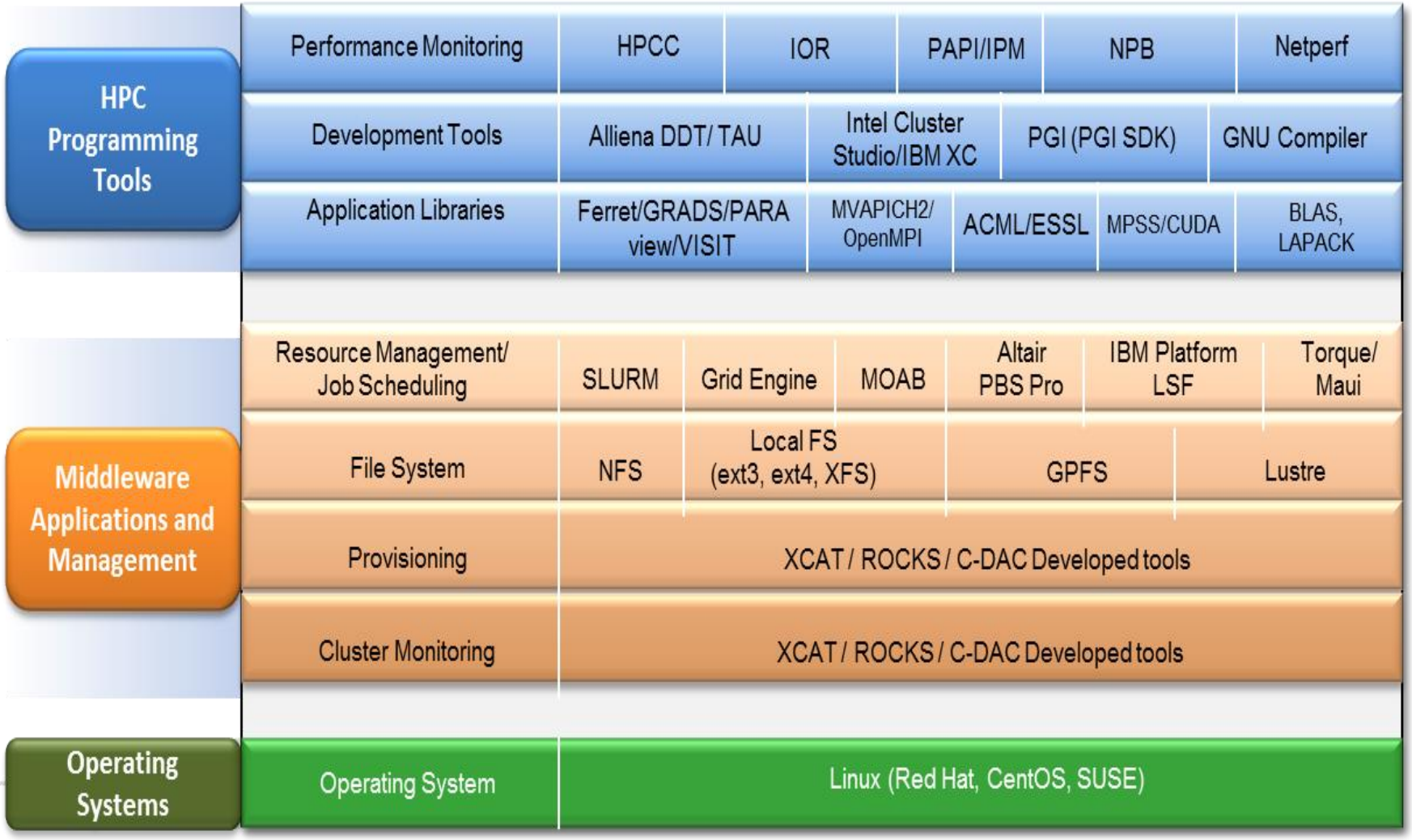
Cray Supercomputer

PARAM Yuva II

Components of Cluster



HPC Software Stack



Single CPU Systems

- Can run a single stream of code
- Performance can be improvement through
    - Increasing ALU width
    - Increasing clock frequency
    - Making use of pipelining
    - Improved compilers
- But still, there is a limit to each of these techniques
- Parallel computing, overcomes the limitations

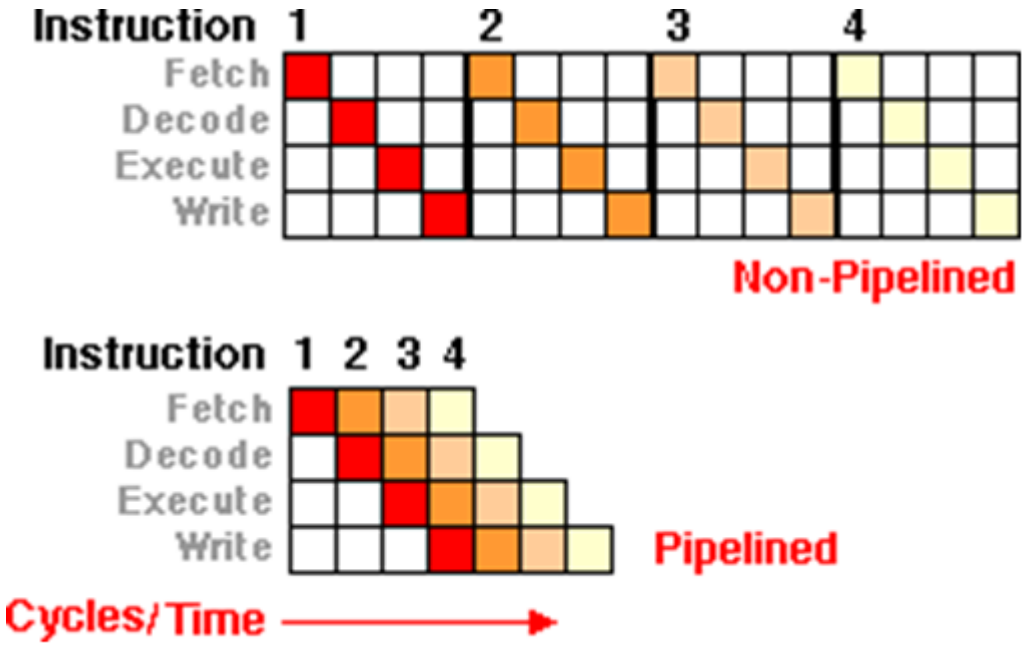
Why use Parallel Computing?

- Overcome limitations of single CPU systems
  - Sequential systems are slow
    - Calculations make take days, weeks, years
    - More CPUs can get job done faster
  - Sequential systems are small
    - Data set may not fit in memory
    - More CPUs can give access to more memory
- So, the advantages are
    - Save time
    - Solve bigger problems

Single Processor Parallelism

- Instruction level Parallelism is achieved through
  - Pipelining
  - Superscaler implementation
  - Multicore architecture
  - Using advanced extensions

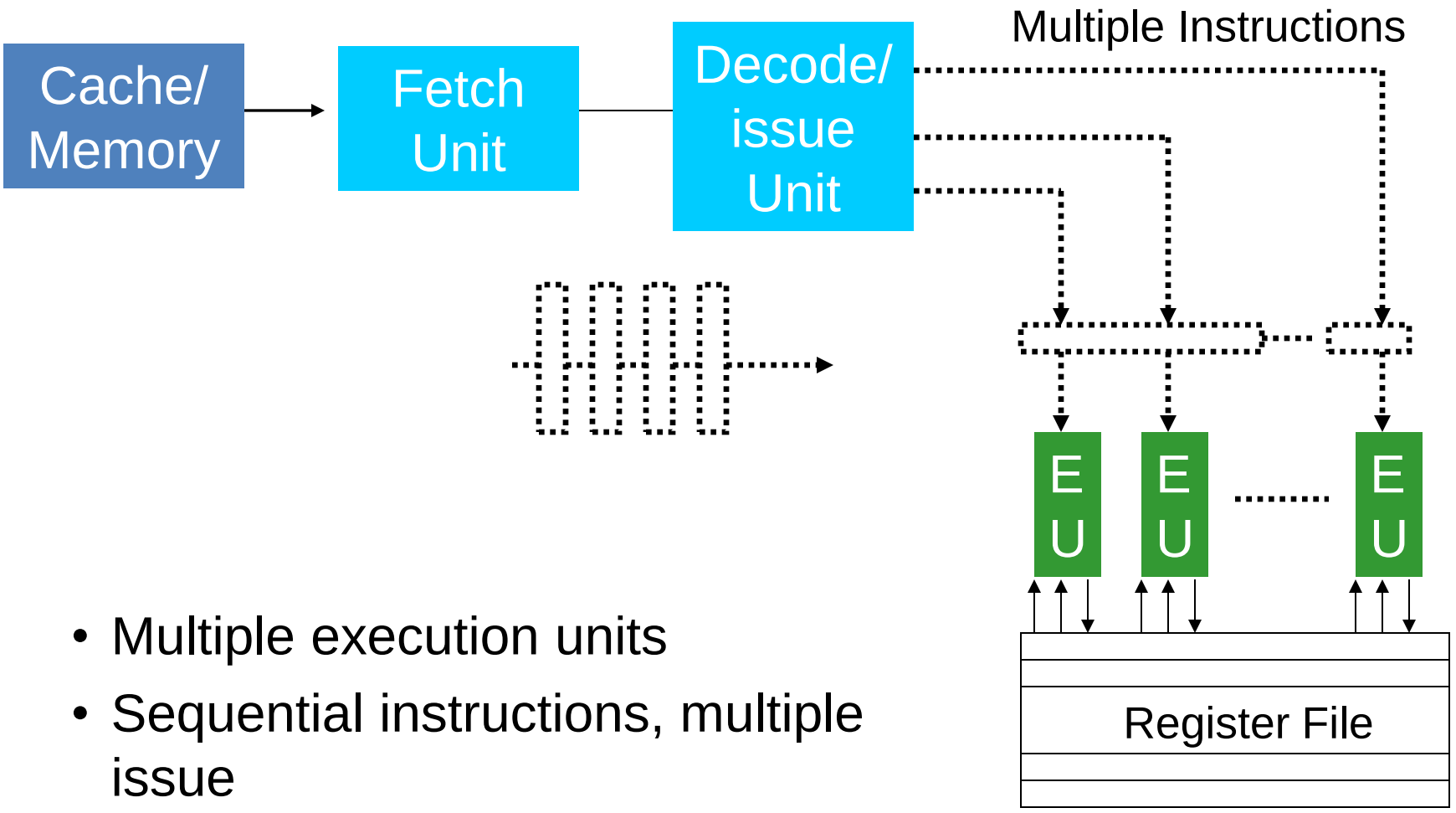
Pipelined Processors



- A new instruction enters every clock
- Instruction parallelism = No. of pipeline stages

Diagram Source: Quora

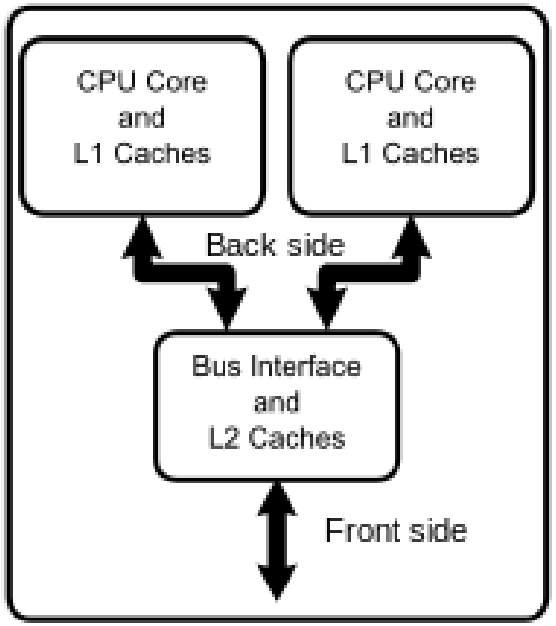
Superscaler



- Multiple execution units
- Sequential instructions, multiple issue

Multicore Processor

- Single computing component with two or more independent processing units
- Each unit is called cores, which read and execute program instructions



Source: Wikipedia.

Advanced Vector eXtensions

- Useful for algorithms that can take advantage of SIMD
- AVX were introduced by Intel and AMD in x86
- Using AVX-512, applications can pack
  - 32 double precision or 64 single precision floating point operations or
  - eight 64-bit and sixteen 32-bit integers
- Accelerates performance for workloads such as
  - Scientific simulations, artificial intelligence (AI)/deep learning, image and audio/video processing

Parallelization Approach

Implicit Parallelism

- Parallelism is exploited implicitly by the compiler and runtime system
  - Automatically detects potential parallelism in the program
  - Assigns the tasks for parallel execution
  - Controls and synchronizes execution

(+) Frees the programmer from the details of parallel execution

(+) it is a more general and flexible solution

(-) very hard to achieve an efficient solution for many applications

Explicit Parallelism

- It is the programmer who has to
  - Annotate the tasks for parallel execution
  - Assign tasks to processors
  - Control the execution and the synchronization points

(+) Experienced programmers achieve very efficient solutions for specific problems

(-) programmers are responsible for all details

(-) programmers must have deep knowledge of the computer architecture to achieve maximum performance.

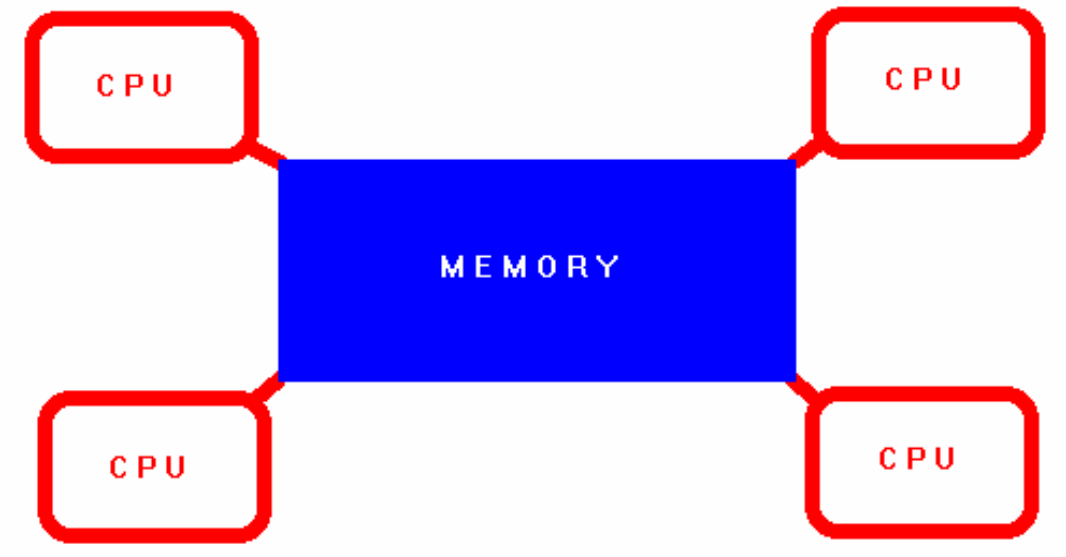
Explicit Parallel Programming Models

Two dominant parallel programming models

- Shared-variable model
- Message-passing model

Shared Memory Model

- Uses the concept of single address space
- Typically SMP architecture is used
  - Scalability is not good



Shared Memory Model

- Multiple threads operate independently but share same memory resources
- Data is not explicitly allocated
- Changes in a memory location effected by one process is visible to all other processes
- Communication is implicit
- Synchronization is explicit

Advantages & Disadvantages of Shared Memory Model

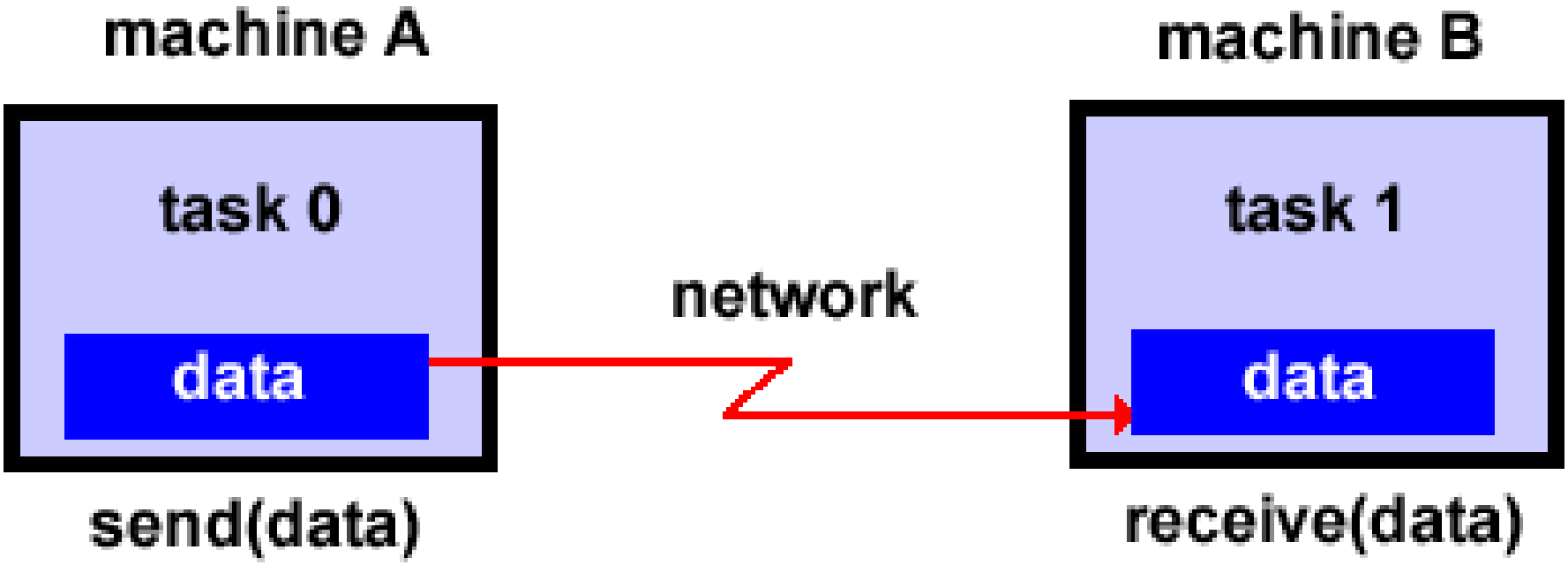
Advantages :

- Data sharing between threads is fast and uniform
- Global address space provides user friendly programming

Disadvantages :

- Lack of scalability between memory and CPUs
- Programmer is responsible for specifying synchronization, e.g. locks
- Expensive

Message Passing Model



Characteristics of Message Passing Model

- Asynchronous parallelism
- Separate address spaces
- Explicit interaction
- Explicit allocation by user

How Message Passing Model Works

- A parallel computation consists of a number of processes
- Each process has purely local variables
- No mechanism for any process to directly access memory of another
- Sharing of data among processes is done by explicitly message passing
- Data transfer requires cooperative operations by each process

Usefulness of Message Passing Model

- Extremely general model
- Essentially, any type of parallel computation can be cast in the message passing form
- Can be implemented on wide variety of platforms, from networks of workstations to even single processor machines
- Generally allows more control over data location and flow within a parallel application than in, for example the shared memory model
- Good scalability

Parallelization Paradigms

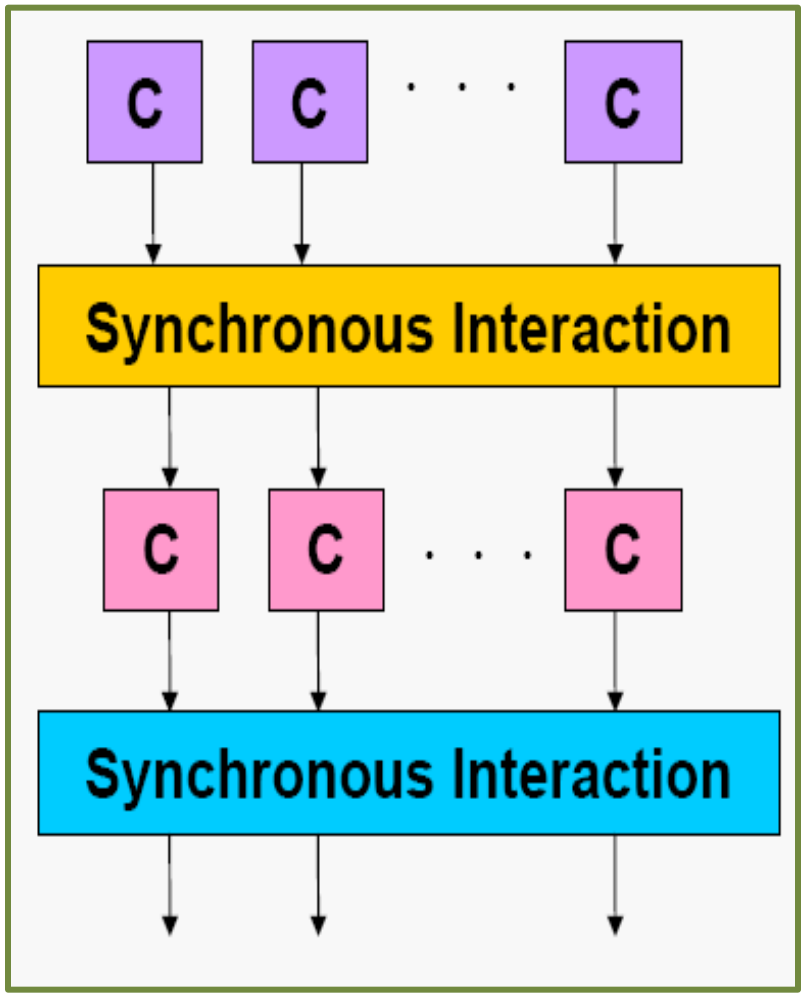
Ideal Situation !!!

- Each Processor has a Unique work to do
- Communication among processes is largely unnecessary
- All processes do equal work

Parallel Algorithm Paradigms

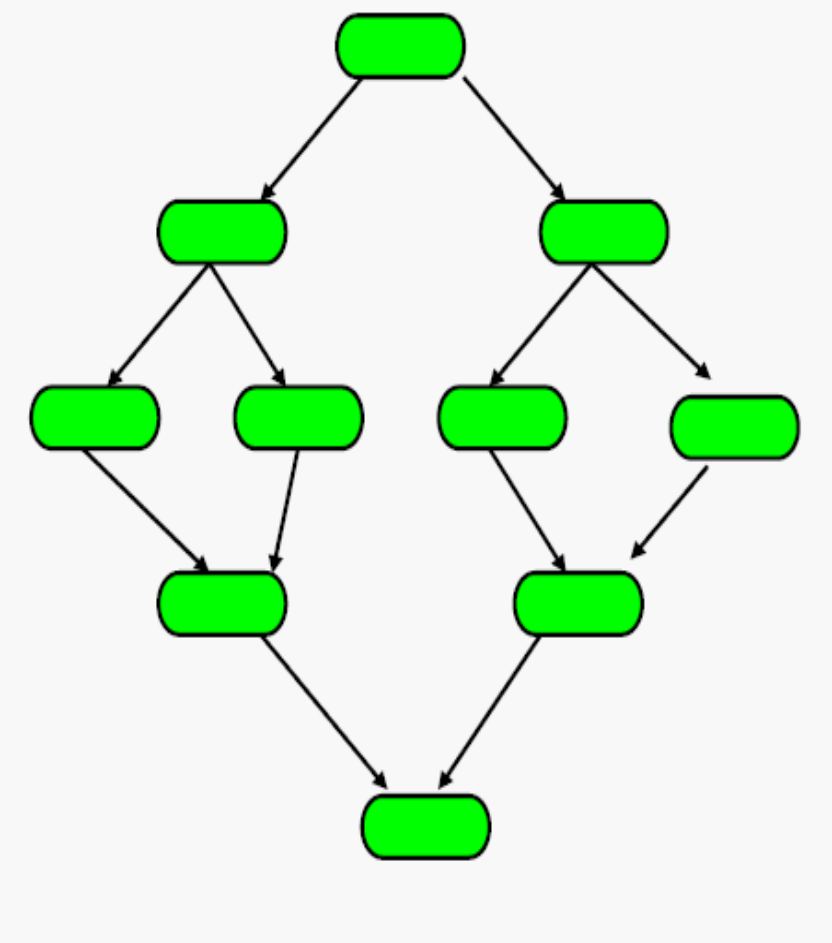
- Phase parallel
- Divide and conquer
- Pipeline
- Process farm
- Domain Decomposition

Phase Parallel Model



- The parallel program consists of a number of super steps, and each has two phases.
- In a computation phase, multiple processes each perform an independent computation.
- In interaction phase, the processes perform one or more synchronous interaction operations, such as a barrier or a blocking communication.

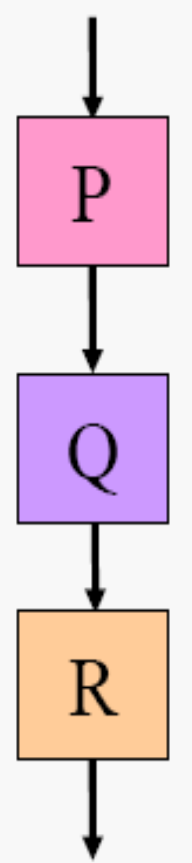
Divide and Conquer model



- A parent process divides its workload into several smaller pieces and assigns them to a number of child processes.
- The child processes then compute their workload in parallel and the results are merged by the parent.
- This paradigm is very natural for computations such as quick sort.

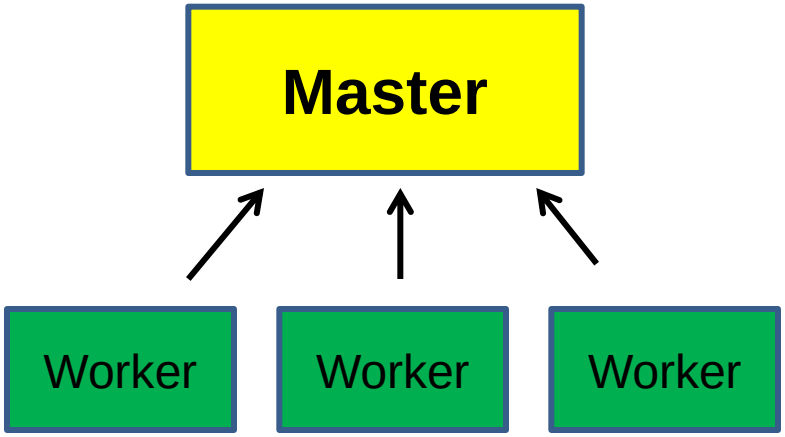
Pipeline Model

Data Stream



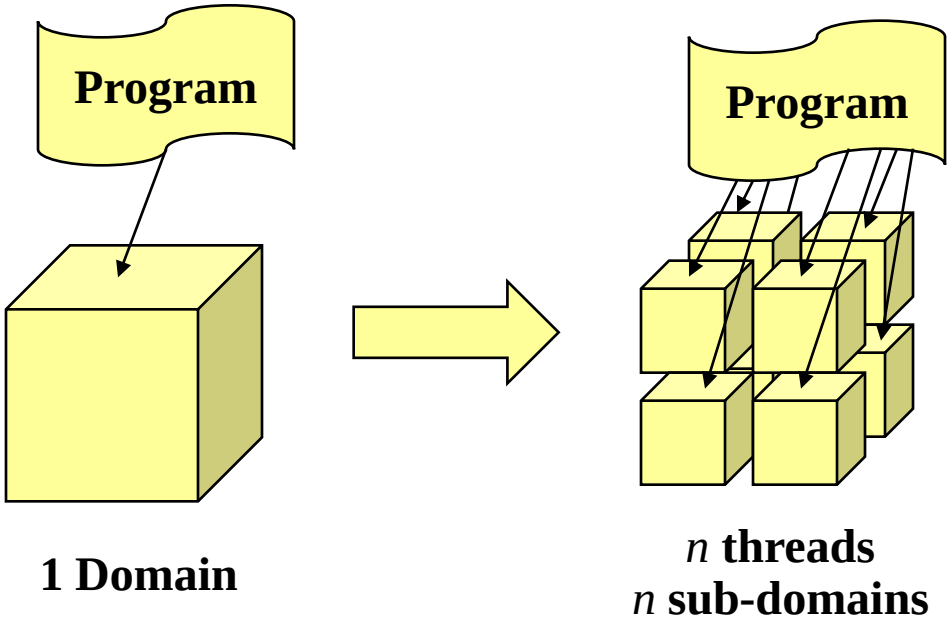
- In pipeline paradigm, a number of processes form a virtual pipeline.
- A continuous data stream is fed into the pipeline, and the processes execute at different pipeline stages simultaneously.

Process Farm Model



- Also known as the master-worker paradigm.
- A master process executes the essentially sequential part of the parallel program
- It spawns a number of worker processes to execute the parallel workload.
- When a worker finishes its workload, it informs the master which assigns a new workload to the slave.
- The coordination is done by the master.

Domain Decomposition



This methods solve a boundary value problem by splitting it into smaller boundary value problems on subdomains and iterating to coordinate the solution between adjacent subdomains.

Desirable Attributes for Parallel Algorithms

- Concurrency
  - Ability to perform many actions simultaneously
- Scalability
  - Resilience to increasing processor counts
- Data Locality
  - High ratio of local memory accesses to remote memory accesses (through communication)
- Modularity:
  - Decomposition of complex entities into simpler components

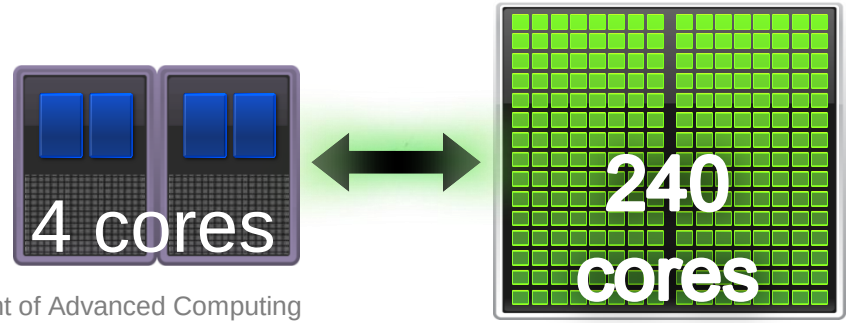
Writing parallel codes

- Distribute the data to memories
- Distribute the code to processors
- Organize and synchronize the workflow
- Optimize the resource requirements by means of efficient algorithms and coding techniques

Heterogeneous Computing: GPUs + CPUs

Massive processing power introduces I/O challenge

- Getting data to and from the processing units can take as long as the processing itself
- Requires careful software design and deep understanding of algorithms and architecture of
  - Processors (Cache effects, memory bandwidth)
  - GPU accelerators
  - Interconnects (Ethernet, IB, 10 Gigabit Ethernet),
  - Storage (local disks, NFS, parallel file systems)

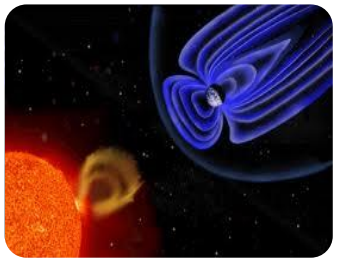


Application Areas of HPC in Science & Engineering

HPC in Science

Space Science

- Applications in Astrophysics and Astronomy



Earth Science

- Applications in understanding Physical Properties of Geological Structures, Water Resource Modelling, Seismic Exploration

Atmospheric Science

- Applications in Climate and Weather Forecasting, Air Quality

HPC in Science

Life Science

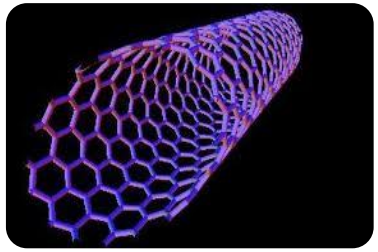
- Applications in Drug Designing, Genome Sequencing, Protein Folding

Nuclear Science

- Applications in Nuclear Power, Nuclear Medicine (cancer etc.), Defence

Nano Science

- Applications in Semiconductor Physics, Microfabrication, Molecular Biology, Exploration of New Materials



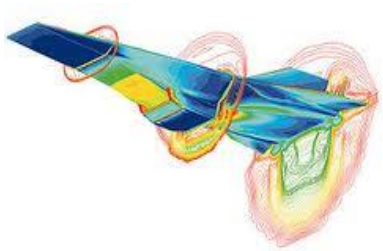
HPC in Engineering

Crash Simulation

- Applications in Automobile and Mechanical Engineering

Aerodynamics Simulation & Aircraft Designing

- Applications in Aeronautics and Mechanical Engineering



Structural Analysis

- Applications in Civil Engineering and Architecture

Multimedia and Animation

DreamWorks Animation
SKG produces all its animated movies using HPC graphic technology

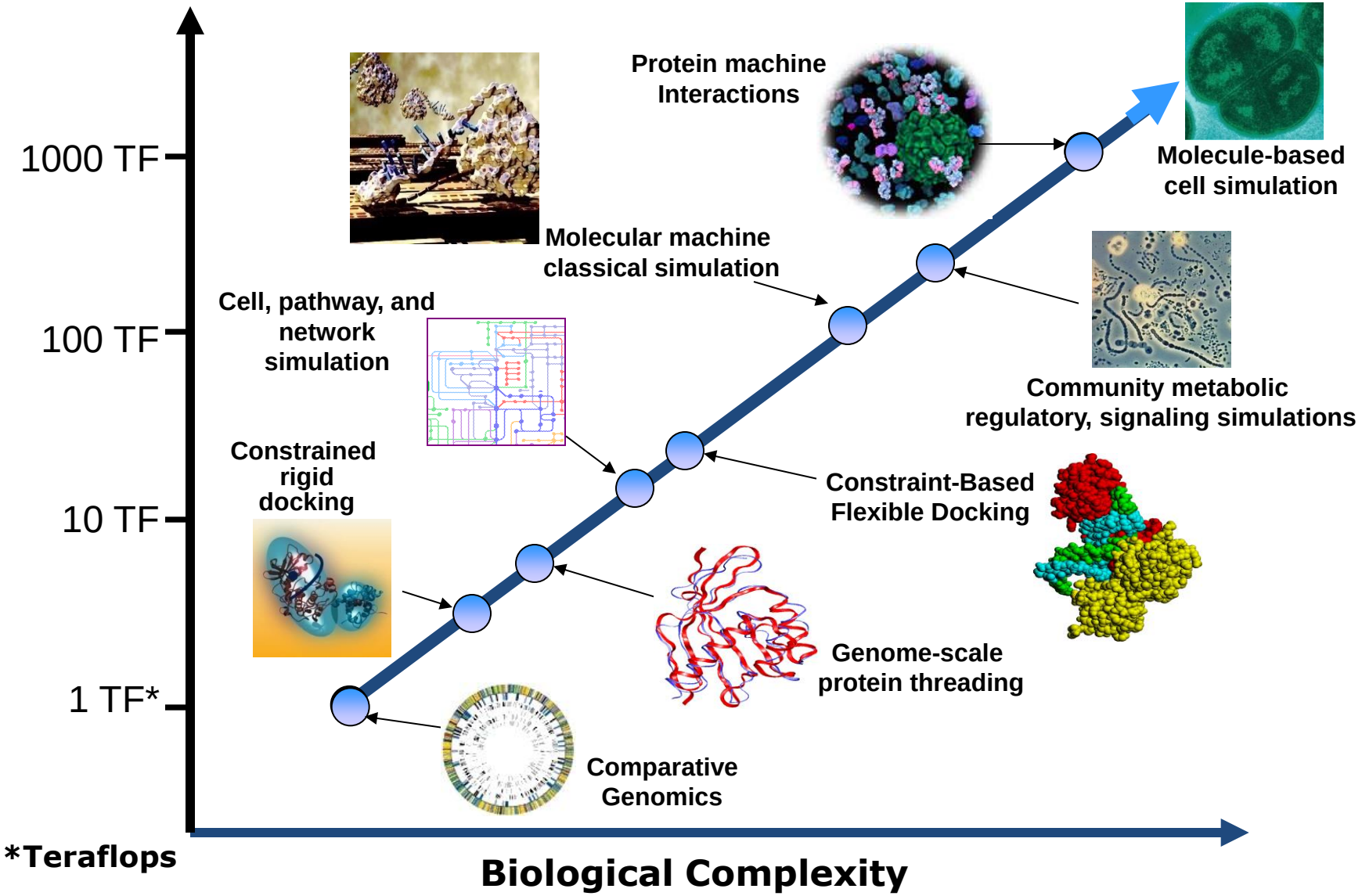
Graphical Animation Application in Multimedia and Animation

HPC in Bioinformatics

Genome: The Book of Life

- We are “programmed”
- The “code of life” is hidden in the genome
- Genome is a molecule which control all functions of an organism
- Every living being has a genome
- A genome is a complete “parts list”
- Genomes are sequenced to understand the “code” and ultimately understand “life”
- Biology is complex
- Even today we do not know much about how the genome functions

HPC for Bioinformatics



HPC in weather and climate forecasting

What is a Numerical Weather Model?

- Fluid Mechanics and Thermodynamic equations. Usually PDEs in time and 3-D space.
- Numerical methods available to solve the equations (given the boundary conditions)

Governing equations

1. Conservation of momentum (Newton's second law)
   Accelerations of 3-d wind ($F = Ma$)

2. Conservation of mass
   Conservation of air (mass continuity)
   Conservation of water (moisture)

3. Conservation of energy
   First law of thermodynamics

4. Relationship among p, V, and T
   ideal gas law?

Governing equations of Atmosphere

1. Equation of momentum:

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-f\left(u-u_g\right)-\frac{1}{\rho}\frac{\partial p}{\partial y}+\frac{\partial}{\partial z}\left(\overline{-v'w'}\right)$$

2. Continuity equation:

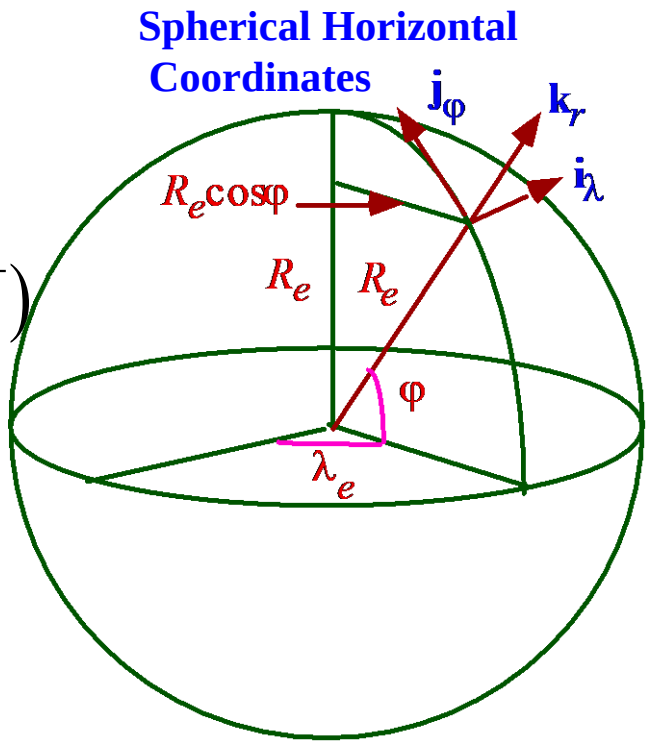
$$\frac{\partial \rho}{\partial t}+\rho\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)=0$$

3. Equation of energy:

$$\frac{\partial \theta}{\partial t}+u\frac{\partial \theta}{\partial x}+v\frac{\partial \theta}{\partial y}+w\frac{\partial \theta}{\partial z}=\frac{\partial}{\partial z}\left(\overline{-w'\,\theta}\right)$$

4. Equation of moisture:

$$\frac{\partial q}{\partial t}+u\frac{\partial q}{\partial x}+v\frac{\partial q}{\partial y}+w\frac{\partial q}{\partial z}=\frac{\partial}{\partial z}\left(\overline{-w'\,q'}\right)$$



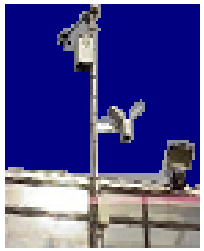
Observations data for Weather Modeling

Boundary values in terms of weather parameters have to be collected first before applying the model (~10 billion observations/day).

Radar

weather buoys

Weather stations



10 billion observations/day,

Satellites

Air craft Data

Ship weather observations

Thank You

ashishk@cdac.in

Thank You

ashishk@cdac.in

National Supercomputing Mission

About the NSM

- Government of India initiated 7 year project with an outlay of Rs. 4500 Crores
- Implementing agencies : C-DAC and IISc
- Activities
  - Infrastructure Development
  - Application Development
  - R&D for Exascale Computing
  - HR Development

Objectives of NSM

- Enhance the national capability in solving grand challenge problems of national and global relevance
- Empower scientists & researchers with state-of-the-art compute facilities for their cutting-edge research in respective Domains
- Reduce redundancies and avoiding duplication of efforts and investments
- Consolidate & build synergy in various ongoing efforts in supercomputing across the country

Scope of the Mission

- Number of supercomputing systems of different sizes and scales with total computer power to the tune of 50-60 Peta Flops
- Building National Supercomputing Grid by interconnecting various HPC systems over NKN
- Developing supercomputing applications through collaborations
- HPC manpower development
- HPC Research and Development leading to Exascale computing readiness

NSM Application Development

At least 5 applications of national relevance to be developed and deployed

Areas include but not limited to

- Computational biology
- Climate modelling, weather prediction
- Engineering including CFD, CSM, CEM
- Disaster simulations and management
- Computational chemistry and material science
- Discoveries beyond Earth (Astrophysics)
- Big data Analytics

HR Development Activities

- Generate enough human resources that can take-up and spearhead supercomputing activities in the country
- Development of highly professional HPC-aware human resource pool
  - For HPC applications development
  - For managing, monitoring and running complex HPC systems
- Manpower needs to be trained at all levels
  - Undergraduate
  - PhD and Masters
  - PG Diploma
- 20,000 strong manpower to be developed in 7 years

Needs to be addressed

- A large number of Universities with good representation and distribution across the country must adopt Parallel Computing as a core course.
- A good number of the students should also obtain working experience by working in the NSM projects
- NSM should also help facilitate job opportunities in India for parallel and supercomputing.
- Manpower needs to trained in
  - HPC related concepts
  - Languages, Tools
  - Hardware and network development

Target Candidates

- Students (UG and PG)
  - Computer Science
  - Non-CS like Mechanical, Civil etc.
- Research Scholars
- Domain experts in R & D institutes
  - Bio-Informatics, CFD etc.
- Working professionals
  - Space, Defense, Private companies

How NSM-EG-HRD can help

- Sharing model curriculum
  - Institutes can adopt to suit their needs
- Conducting Faculty Development Programs
  - Institutes can nominate faculties
- Institutes can conduct programs

Thank You

ashishk@cdac.in

Thank You

ashishk@cdac.in

Compute Node Architectures

CPU only

Accelerators based

- Architectures specific to some applications using GPGPUs, MIC, FPGA accelerators

CPU only Compute Node

CPU

- Typically 2 CPU system with total CPU core count ranging from 32 to 64 cores
- Current commercial options – Skylake from Intel, Power 9 from IBM, EPYC from AMD, Thunder X2 (ARM) from Cavium

Memory

- Typical memory size 4-8 GB/ core i.e 128 GB- 512GB per node – DDR4 2666 Mhz
- Some applications require high memory ranging from 512TB to 4TB per node

CPU node compute Power – 1 TF to 3 TF peak

Memory technologies

Requirement: Large, high b/w and low latency memory

CPUs are using more DRAM channels to increase the bandwidth – Typically 6-8 channels giving 180 – 250 GB/sec memory Bandwidth

Memory Closer to CPU on chip

- HBM (3D stack) – Typical size 8GB-32GB

Use of NVRAM to increase capacity instead of DRAM for cost effectiveness

HPC Interconnect/Network

High bandwidth (100Gbps – 200 Gbps)

Low latency (<1 us)

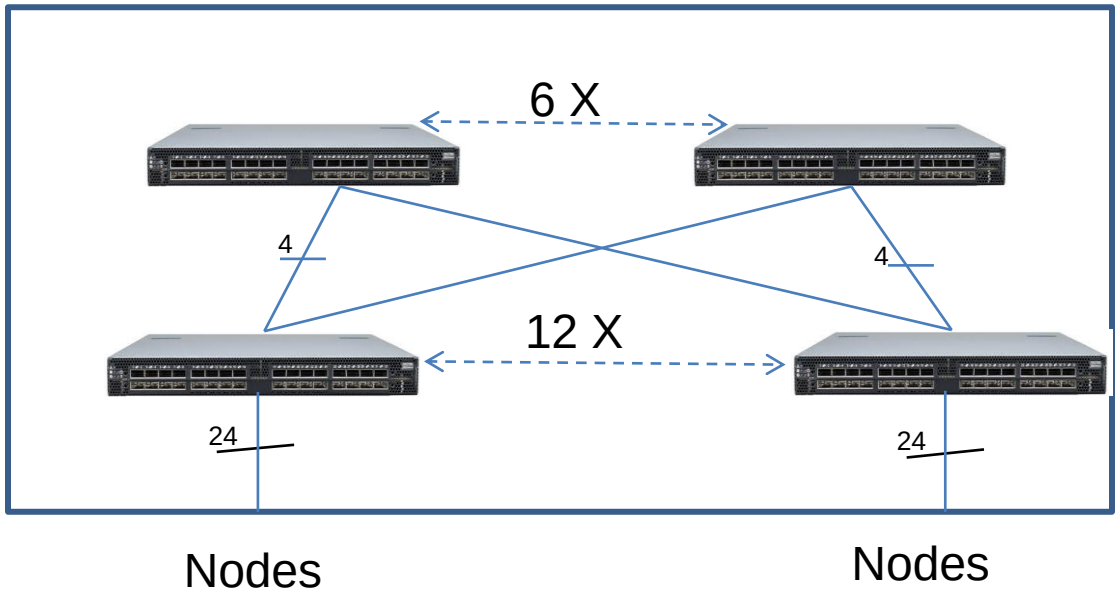
Support for more CPU cores at user level communication

High scalability

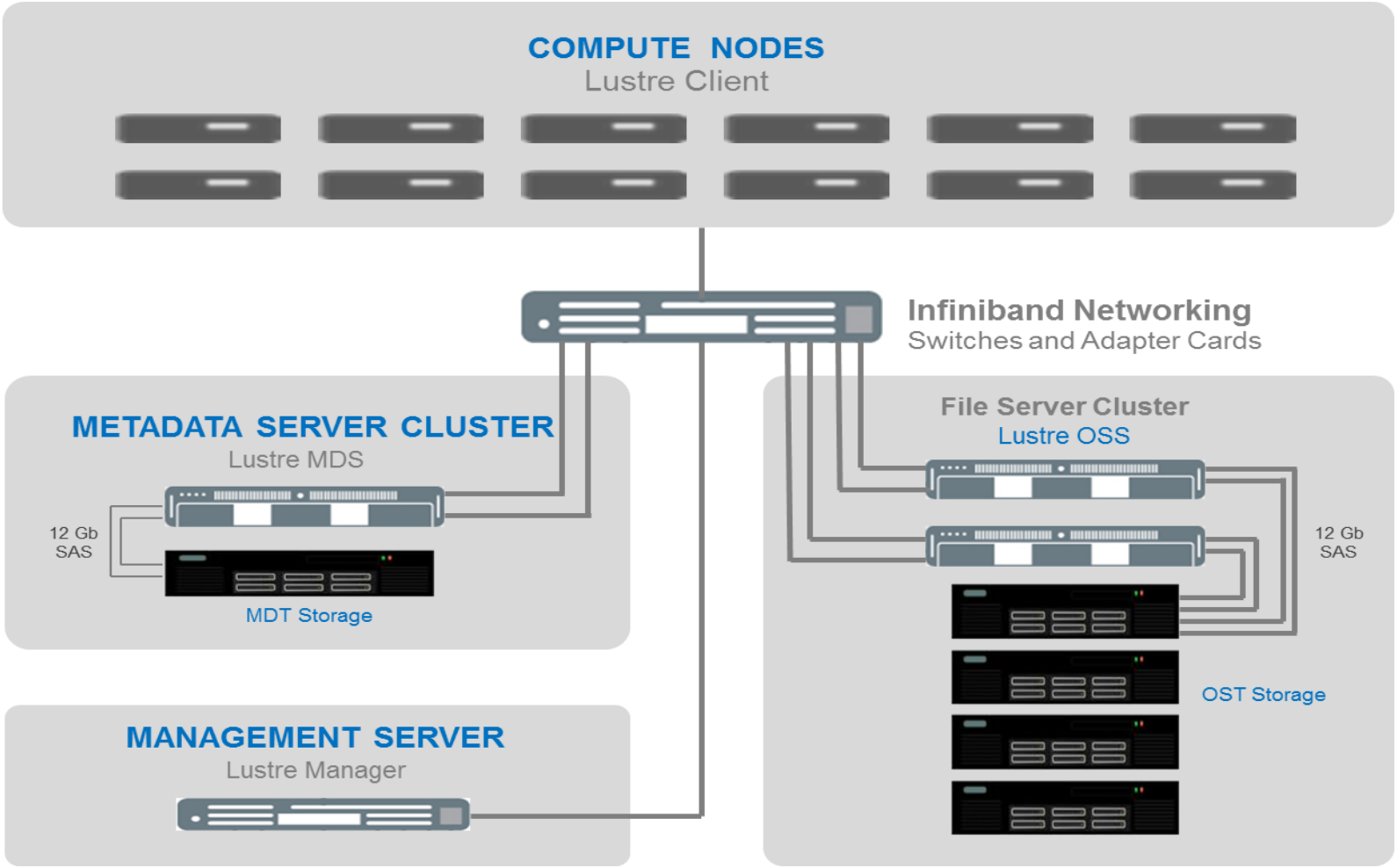
- Interconnect for 10,000+ nodes
- Low hop, low latency topology
- Reliable and intelligent routing

Intel Omnipath and Mellanox Infiniband

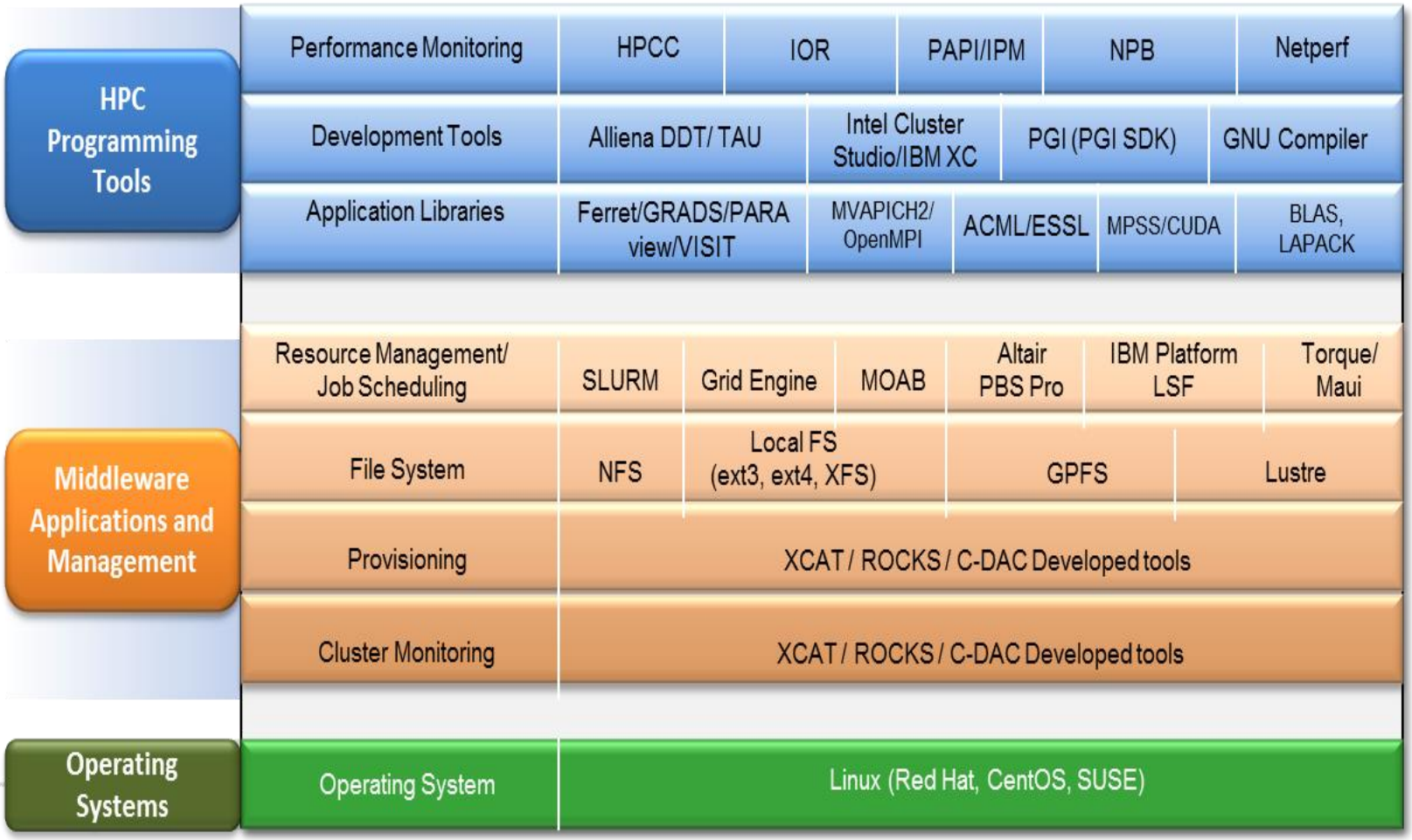
288 ports HPC network using 48 port ASIC



Storage Architecture- Lustre



HPC Software Stack



Compute Node Architectures for AI

Training – Computationally expensive problem, which can run in days/months

Inferencing – A Real time problem

- CPUs like Intel Xeon processors are used to perform volume inferencing in the data center or cloud. Some are using FPGAs for extremely low-latency volume inferencing jobs.

Platforms for AI

- CPUs, GPGPUs
- For low power and speed
  - Special purpose-built silicon for AI training such as the Intel Neural Network Nervana Processor family and Google's TPU
  - FPGAs, which can serve as programmable accelerators for inference.
  - Neuromorphic chips such as Loihi by Intel
- Quantum computing

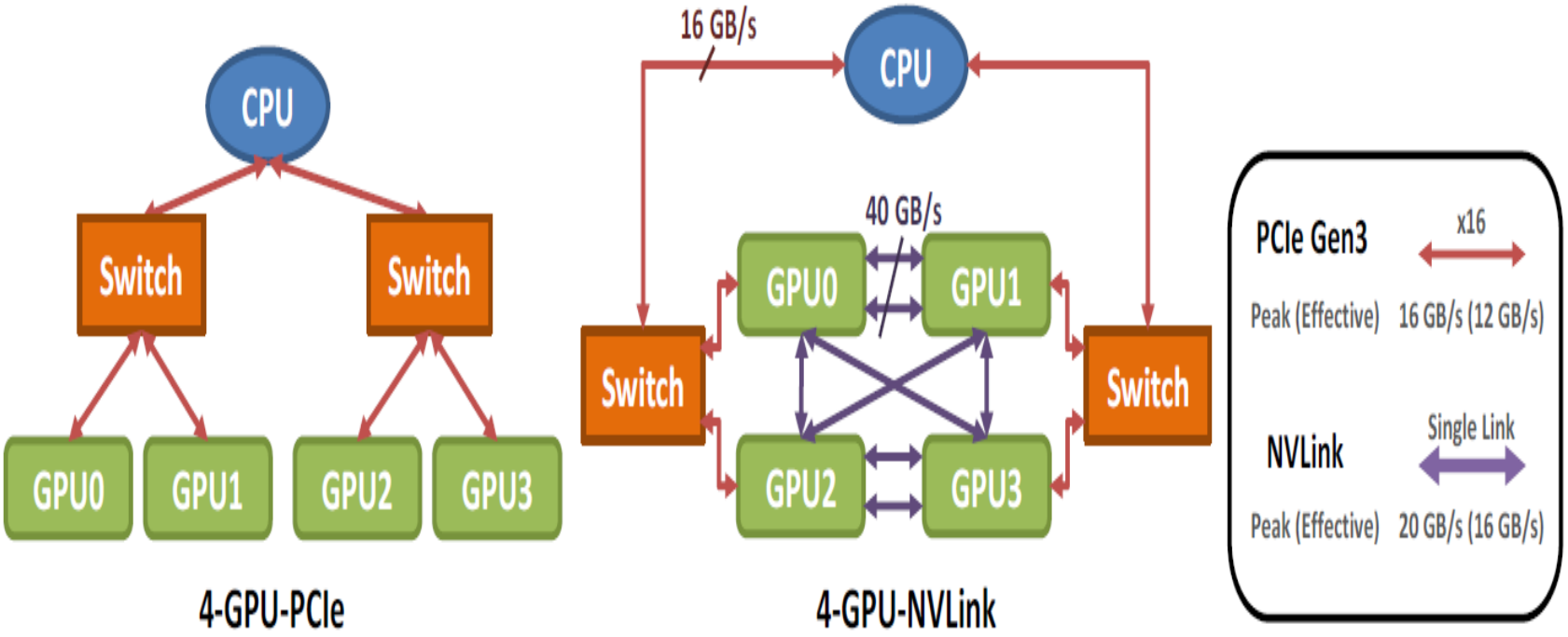
NVIDIA V100 GPU

- 80 Streaming processors
- Each SM has:
  - 64 FP32 cores
  - 64 INT32 cores
  - 32 FP64 cores
  - 8 Tensor Cores
  - Four texture units
- 7.8 FP64 TF
- 12.5 Tensor TF
- 16 GB memory

V100: Streaming Processor

Source: NVIDIA

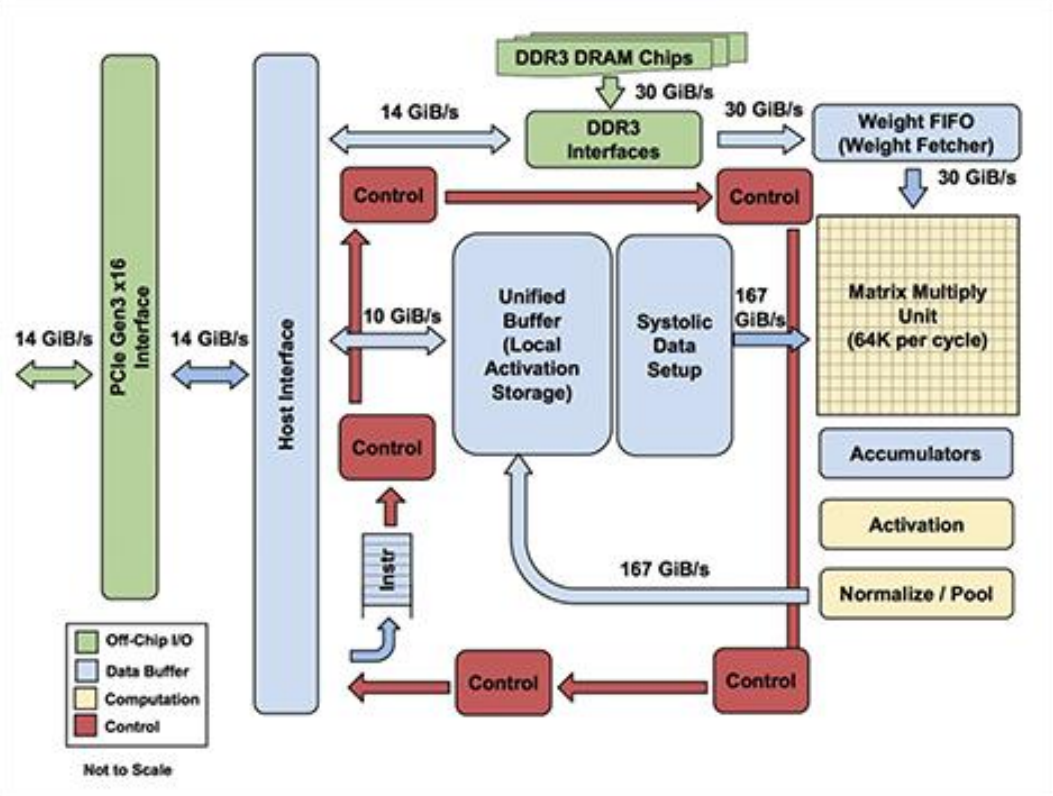
Accelerator compute node: CPU-GPU Connectivity



Source: NVIDIA

Google's Tensor Processing Unit (TPU)

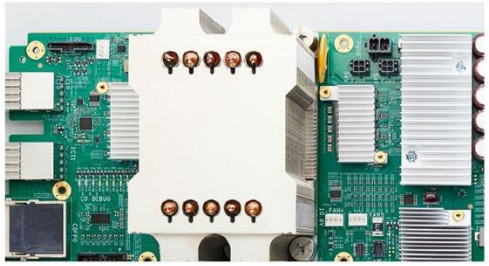
- The Matrix Unit: 65536 (256x256) 8 bit multiply-accumulate (MAC) units
- 700 Mhz clock rate
- Peak: 92 T operations/sec
- >25X as many MACs vs GPU
- 4 MiB of onchip accumulator memory
- 24 MiB of on-ship Unified buffer
- 8 GiB of off-chip weight DRAM memory



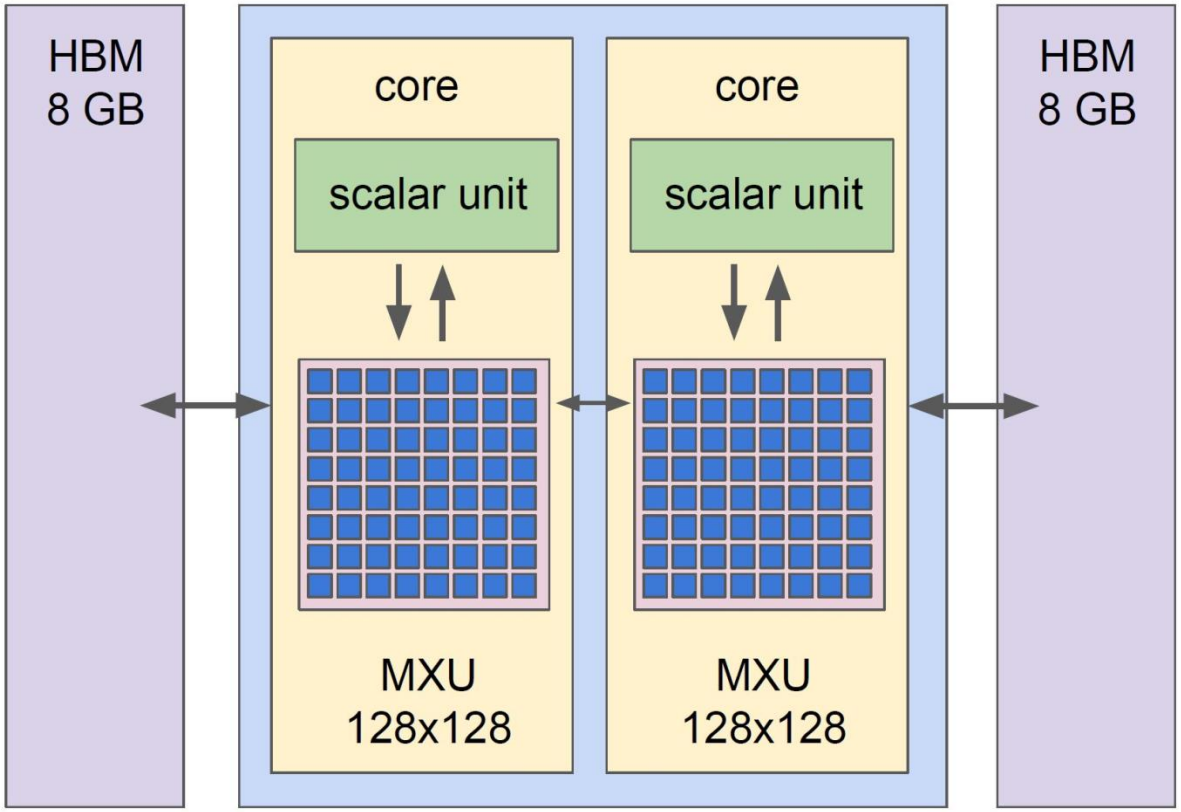
Source: Google

TPU 2

TPUv2 Chip

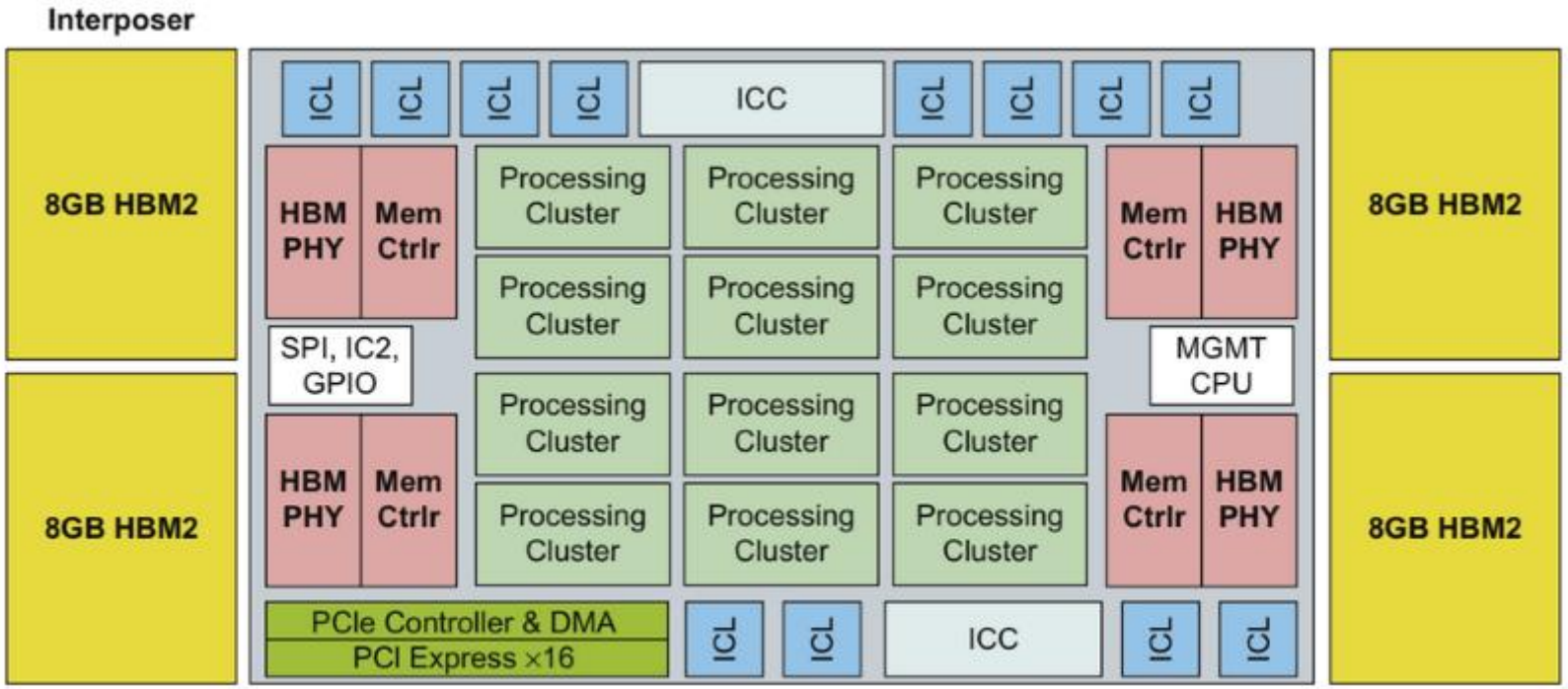


- 16 GB of HBM
- 600 GB/s mem BW
- Scalar unit: 32b float
- MXU: 32b float accumulation but reduced precision for multipliers
- 45 TFLOPS



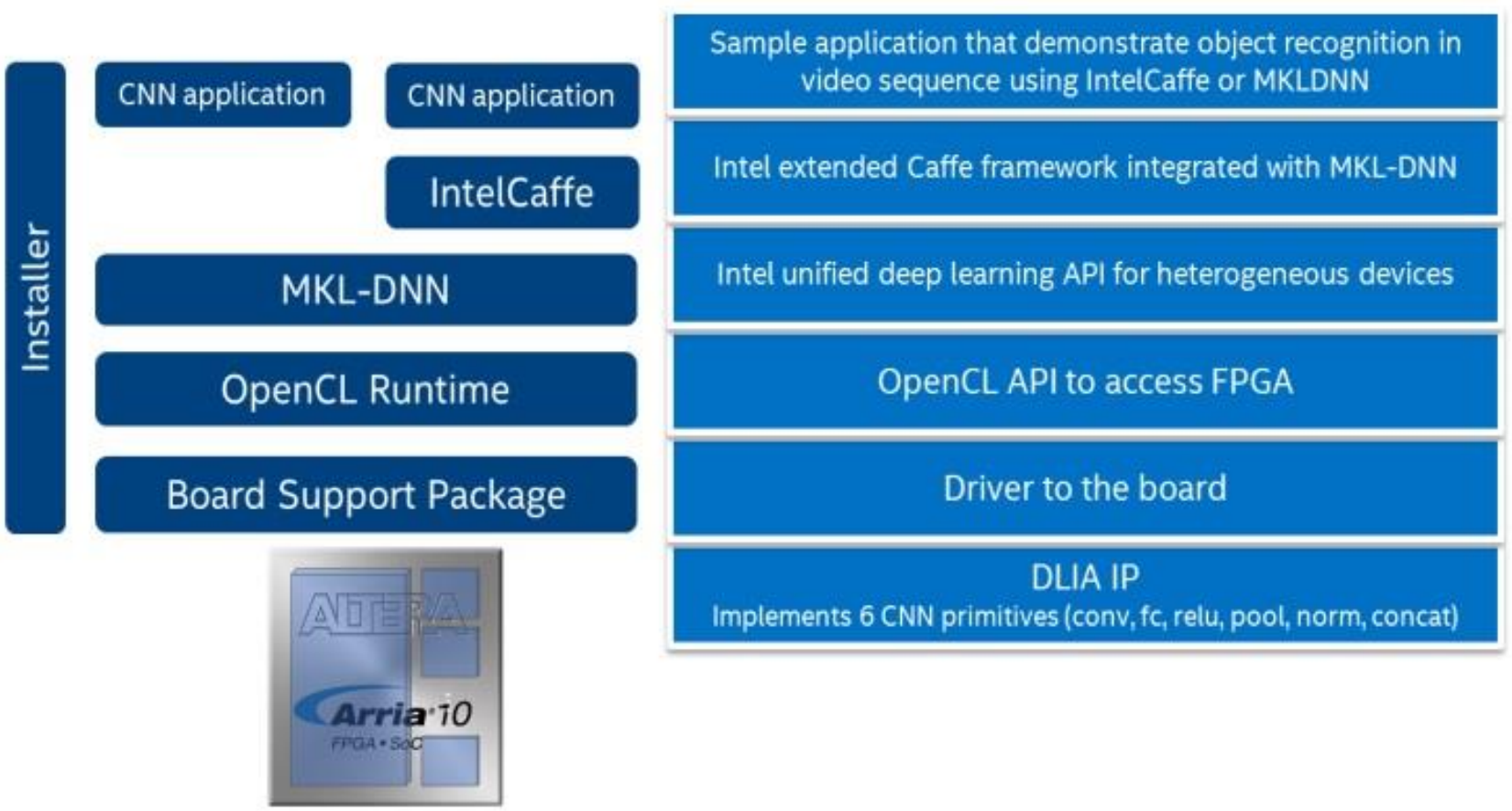
Source: Google

Intel's Nervana, called Lake Crest



Source: Hennessy and Patterson, "Computer Architecture: A Quantitative Approach"

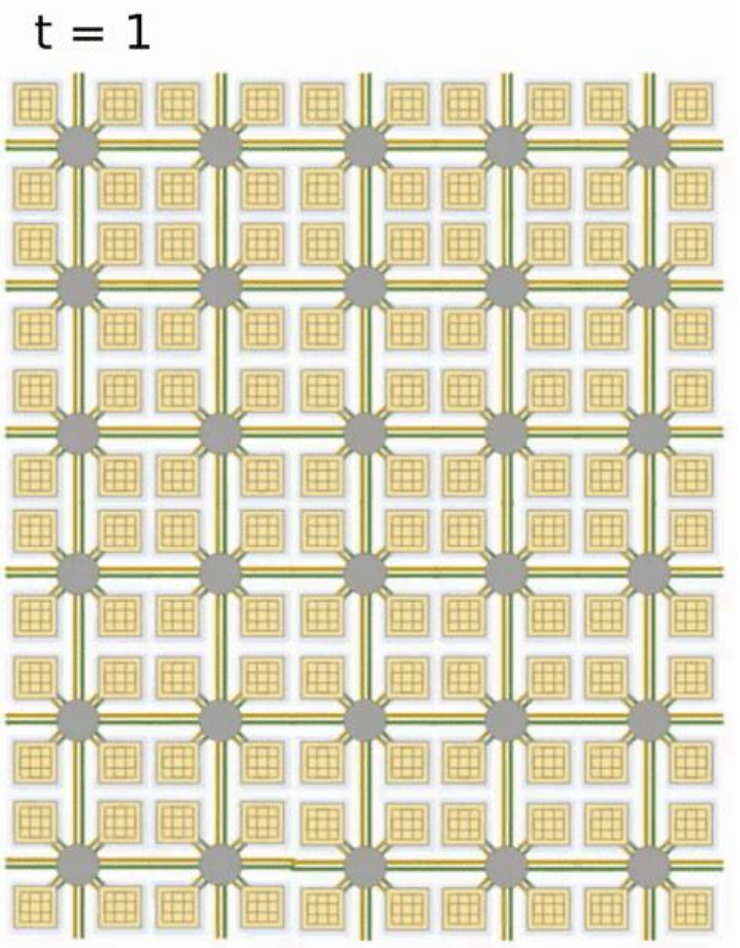
Intel Deep Learning Inference FPGA Accelerator



Source: Intel

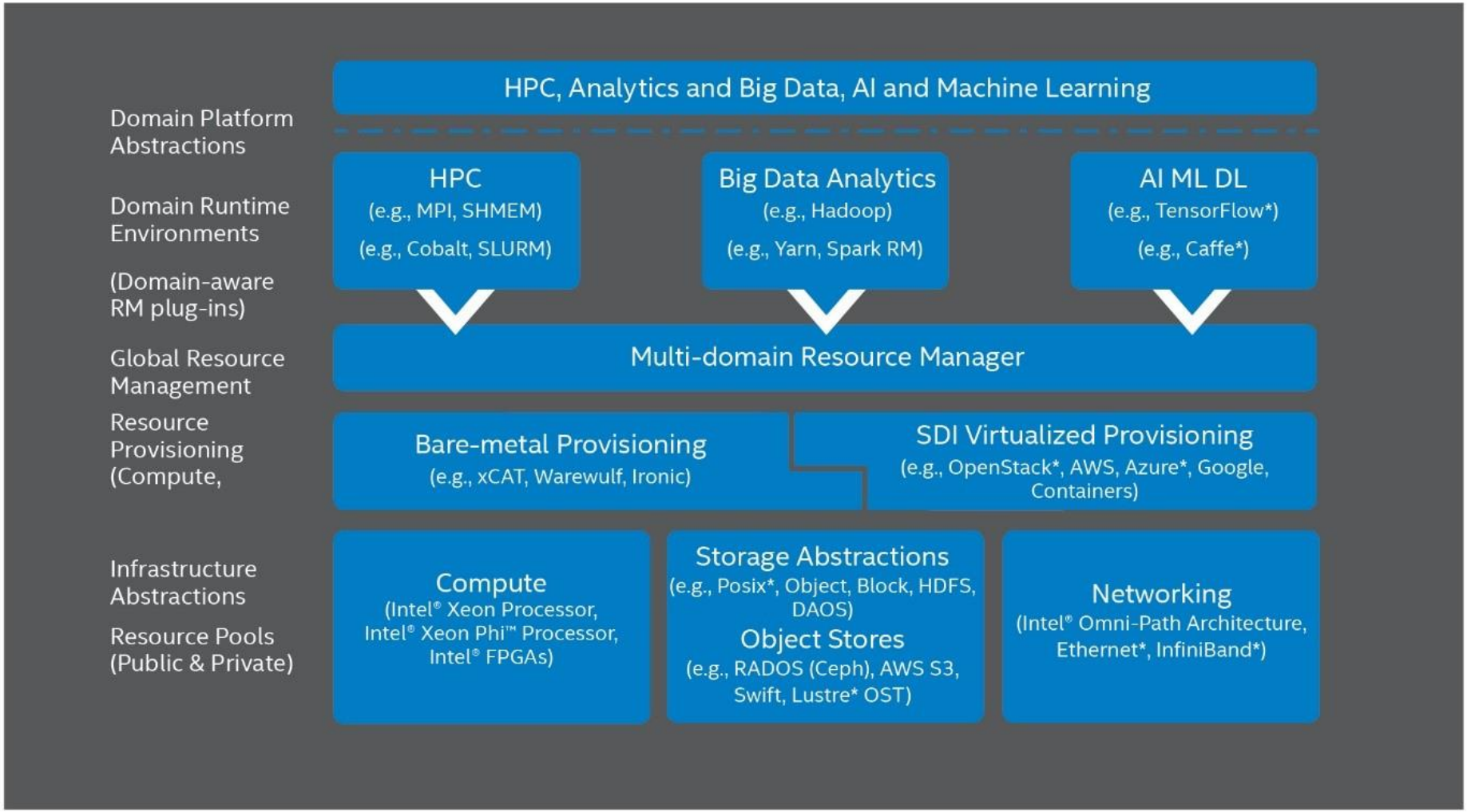
Intel’s neuromorphic chip Loihi

- 128 neuromorphic cores + 3 X86 cores
- More accurately mimics human brain than DL
- > 100X power efficient than current architectures



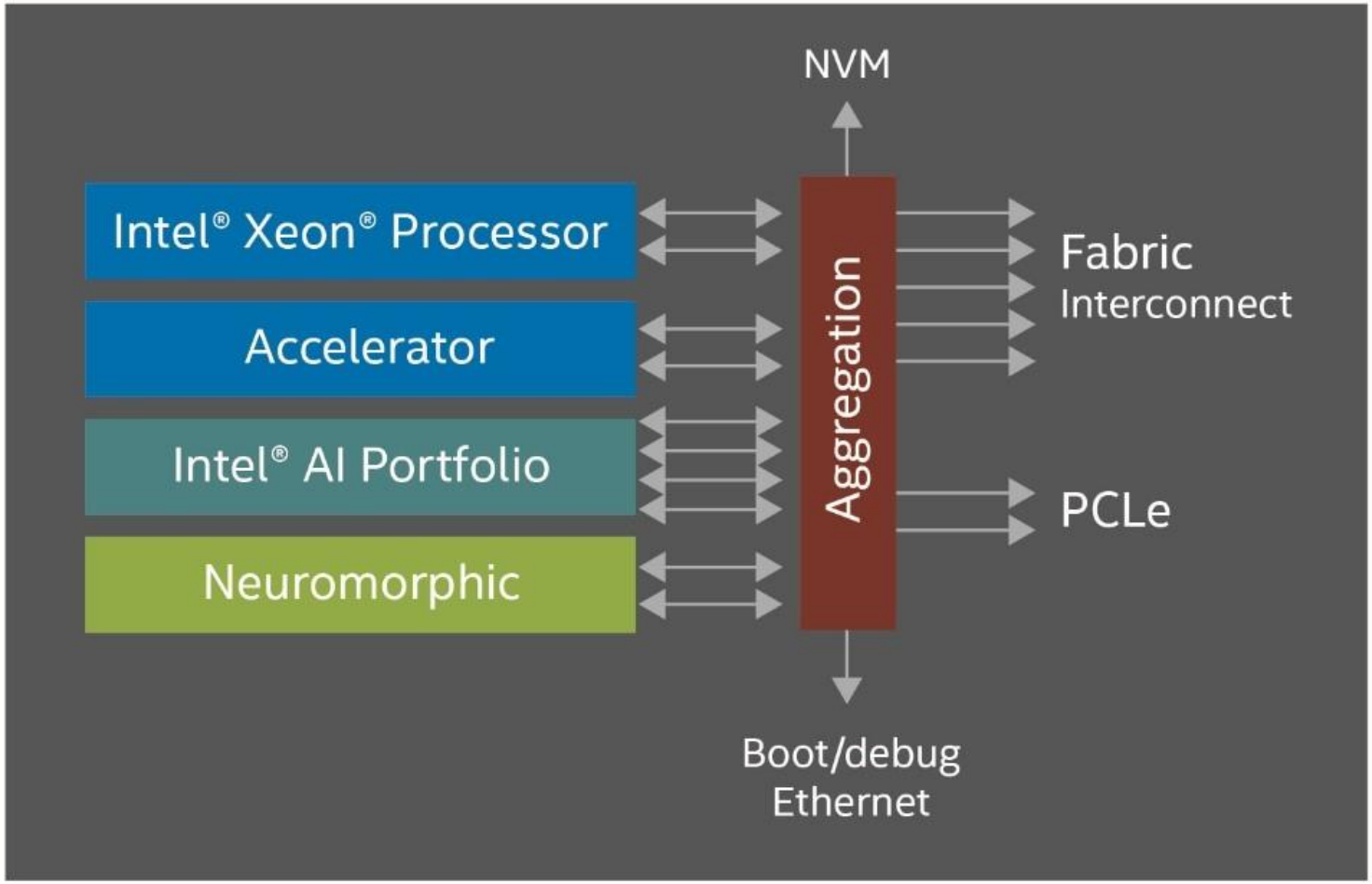
Source wkichip.org

Unifying the “3 Pillars”



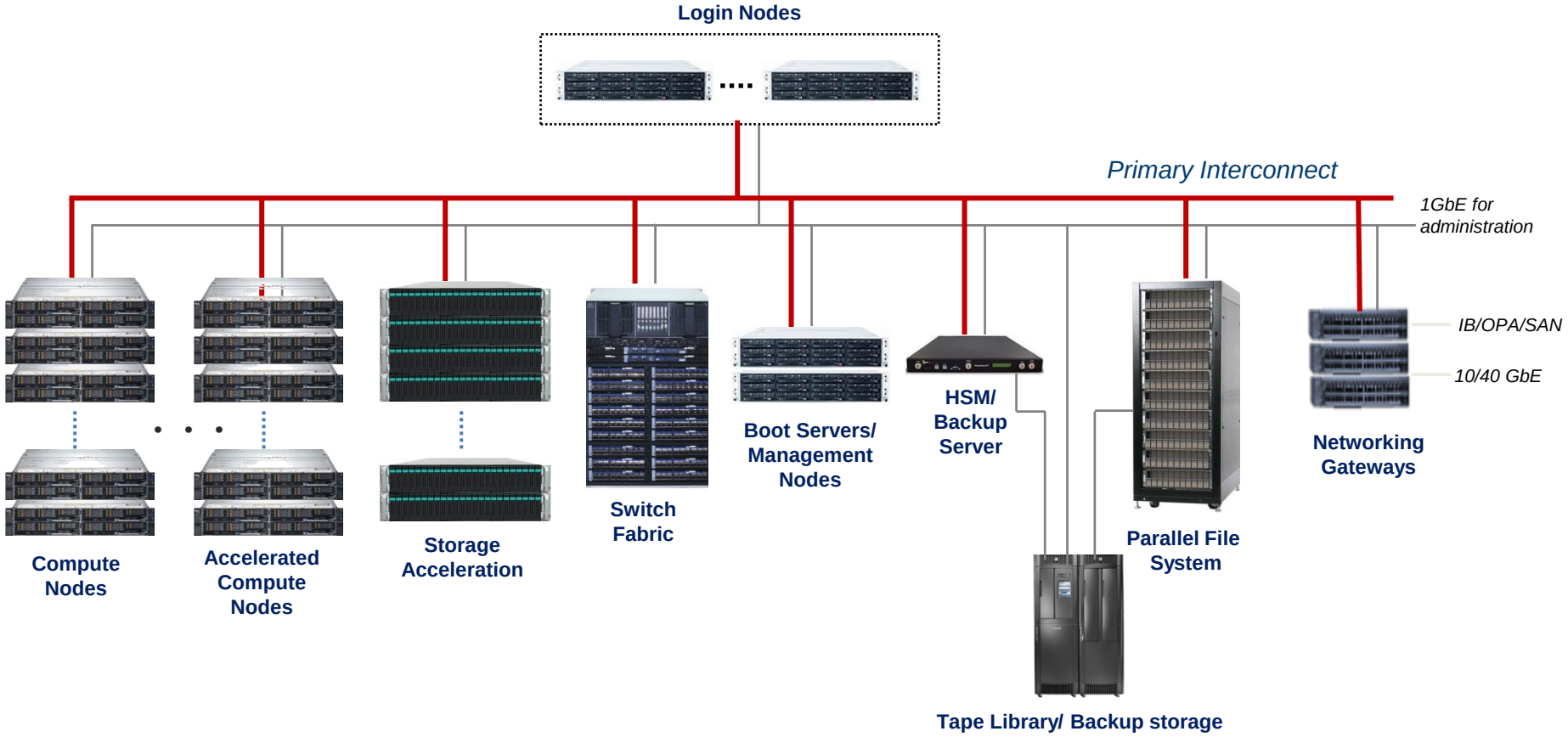
Source: Al Gara, Intel

Intel's proposed configurable Future Platform

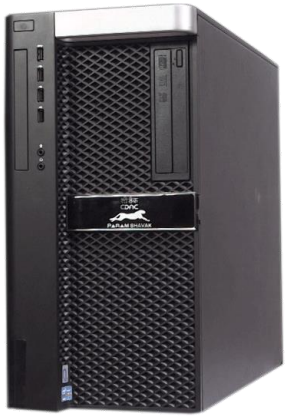


Source: Al Gara Intel

HPC System Architecture



PARAM Shavak



Affordable, Ready to use Appliance for HPC, Deep Learning and Virtual Reality

Hardware choice, Software stack, Scheduling, Frameworks, ready to use Applications, Examples and Tutorials

Deployed More than 70 systems are in the field

C-DAC is extending the concept to 100TF cluster

Summary

System Architectures for HPC, AI and Big Data are unifying

Specialized hardware are emerging for AI ML DL applications for power efficiency and speed