Parallel Computing

Before there was Hadoop and Big Data

There was parallel computing

Big data cluster

For Hadoop, Spark, etc.

Cluster

Cluster

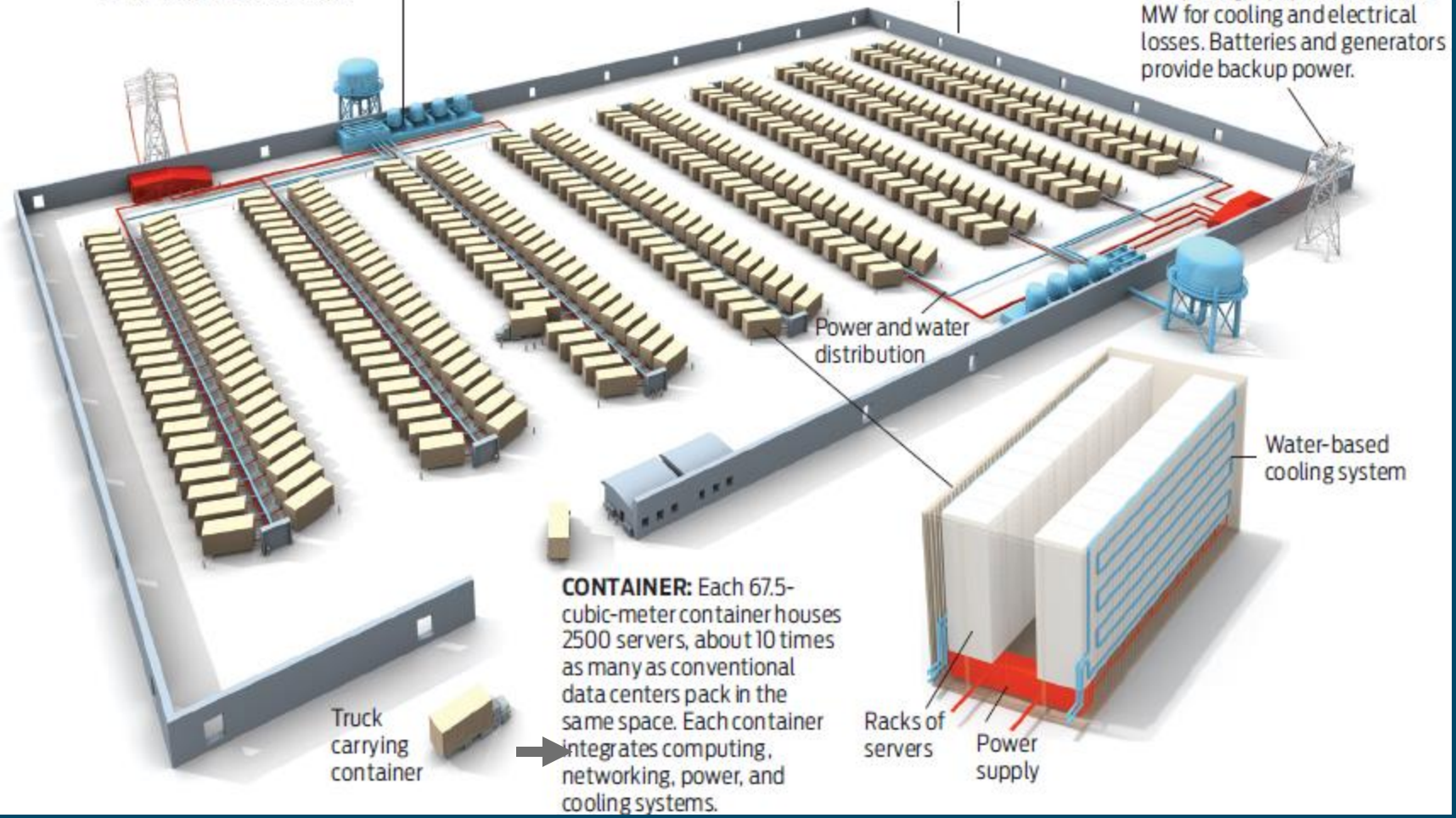
- A computer cluster is a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system
- The Beowulf cluster is a cluster implemented on multiple identical commercial off-the-shelf computers connected with a TCP/IP Ethernet local area network

Wikipedia

COOLING: High-efficiency water-based cooling systems—less energy-intensive than traditional chillers—circulate cold water through the containers to remove heat, eliminating the need for air-conditioned rooms.

STRUCTURE: A 24 000-square-meter facility houses 400 containers. Delivered by trucks, the containers attach to a spine infrastructure that feeds network connectivity, power, and water. The data center has no conventional raised floors.

POWER: Two power substations feed a total of 300 megawatts to the data center, with 200 MW used for computing equipment and 100 MW for cooling and electrical losses. Batteries and generators provide backup power.



CONTAINER: Each 67.5-cubic-meter container houses 2500 servers, about 10 times as many as conventional data centers pack in the same space. Each container integrates computing, networking, power, and cooling systems.

Container with 2500 servers

IEEE Spectrum, 2009, speculation

ROBERT MCMILLAN BUSINESS 02.01.13 06:30 AM

MICROSOFT JAMS THOUSANDS OF SERVERS INTO BOXES, BUILDING OUT $1 BILLION DATA CENTER

They're are pre-assembled containers that come jammed with as many as 2,500 servers, that suck cool air in on one side and spit it out on the other.

High performance computing cluster

HPC super computers

Scale-up (bigger computers) are expensive

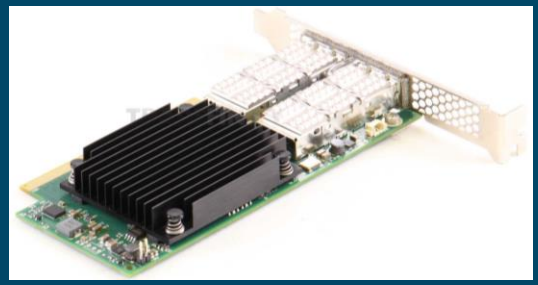
- Early designs had specialized chips and custom OS
- Newer designs are Linux clusters of general-purpose processors with specialized interconnection (hardware network)

The IBM Blue Gene/P supercomputer "Intrepid" at Argonne National Laboratory runs 164,000 processor cores

Cray (now HPE) Trinity Supercomputer

HPC Applications

- The Weather Company
  - Forecasts out to 3 days
  - Provides 13km coverage
  - Updated every 1 – 6 hours
- HPC Computing
  - 2,600 CPU cores
  - QDR/FDR infiniband (~ 50 Gbps link)



https://www.ecmwf.int/sites/default/files/elibrary/2016/16821-forecasts-applications-driven-high-performance-computing-weather-company.pdf

The Quest for Higher Performance: 2012 Update

Top Three Supercomputers in November 2012 (http://www.top500.org)

1. Cray Titan	2. IBM Sequoia	3. Fujitsu K Computer
ORNL, Tennessee	LLNL, California	RIKEN AICS, Japan
XK7 architecture	Blue Gene/Q arch	RIKEN architecture
560,640 cores, 710 TB, Cray Linux	1,572,864 cores, 1573 TB, Linux	705,024 cores, 1410 TB, Linux
Cray Gemini interconn't	Custom interconnect	Tofu interconnect
17.6/27.1 PFLOPS*	**16.3/20.1 PFLOPS***	**10.5/11.3 PFLOPS***
AMD Opteron, 16-core, 2.2 GHz, NVIDIA K20x	Power BQC, 16-core, 1.6 GHz	SPARC64 VIIIfx, 2.0 GHz
8.2 MW power	7.9 MW power	12.7 MW power

* max/peak performance

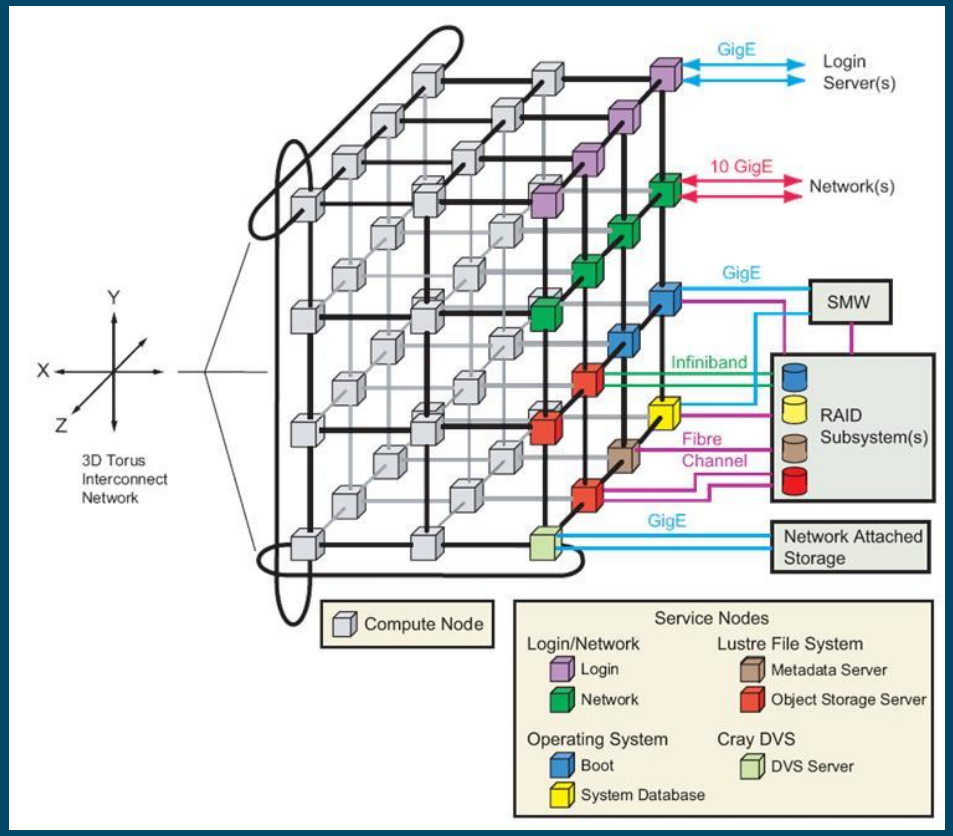
In the top 10, IBM also occupies ranks 4-7 and 9-10. Dell and NUDT (China) hold ranks 7-8.

Top 500 computers

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	706,304	64,590.0	89,794.5	2,528

https://top500.org/lists/top500/2021/06/

Cray Gemini super interconnect (very expensive)

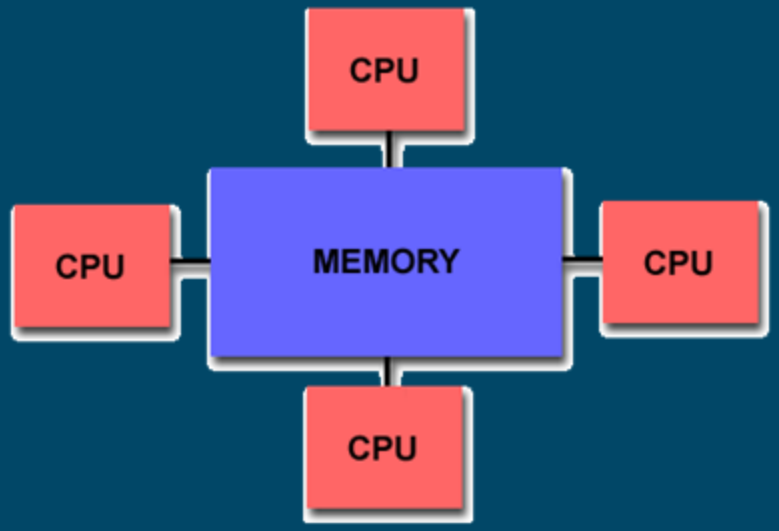


- Interconnect network

- The Cray XE6 blade: Two Gemini interconnects on the left (which is the back of the blade), with four two-socket server nodes and their related memory banks
- https://www.theregister.co.uk/2010/05/25/cray_xe6_baker_gemini/

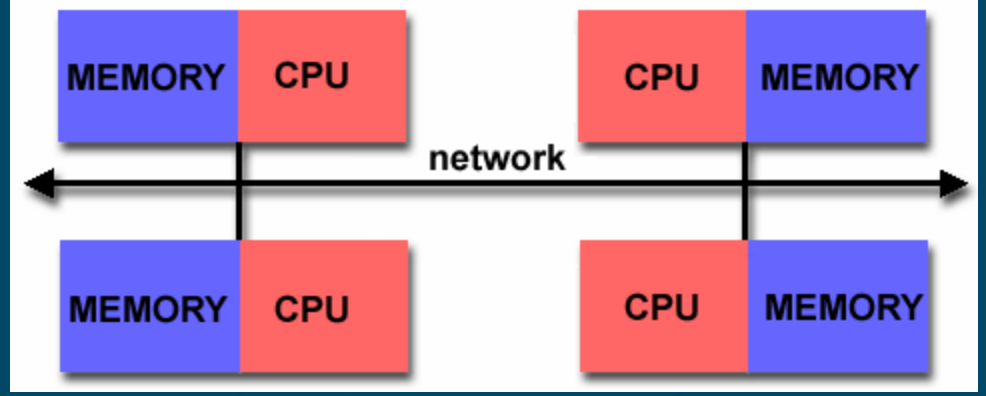
Shared Memory

- All processors to access all memory as global address space
  - e.g., multi-core



- Multiple processors can operate independently but share the same memory resources.
- Changes in a memory location effected by one processor are visible to all other processors.

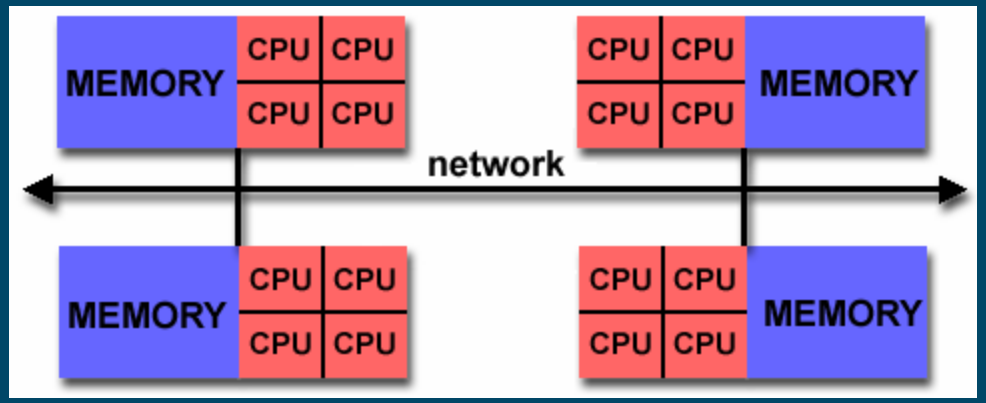
Distributed Memory



- Distributed memory systems require a communication network to connect inter-processor memory.
- Processors have their own local memory.
  - no concept of global address space across all processors.
  - Changes it makes to its local memory have no effect on the memory of other processors.
- When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. Synchronization between tasks is likewise the programmer's responsibility.
- The network "fabric" used for data transfer varies widely, though it can be as simple as Ethernet.

Hybrid Distributed-Shared Memory

- The largest and fastest computers in the world today employ both shared and distributed memory architectures.



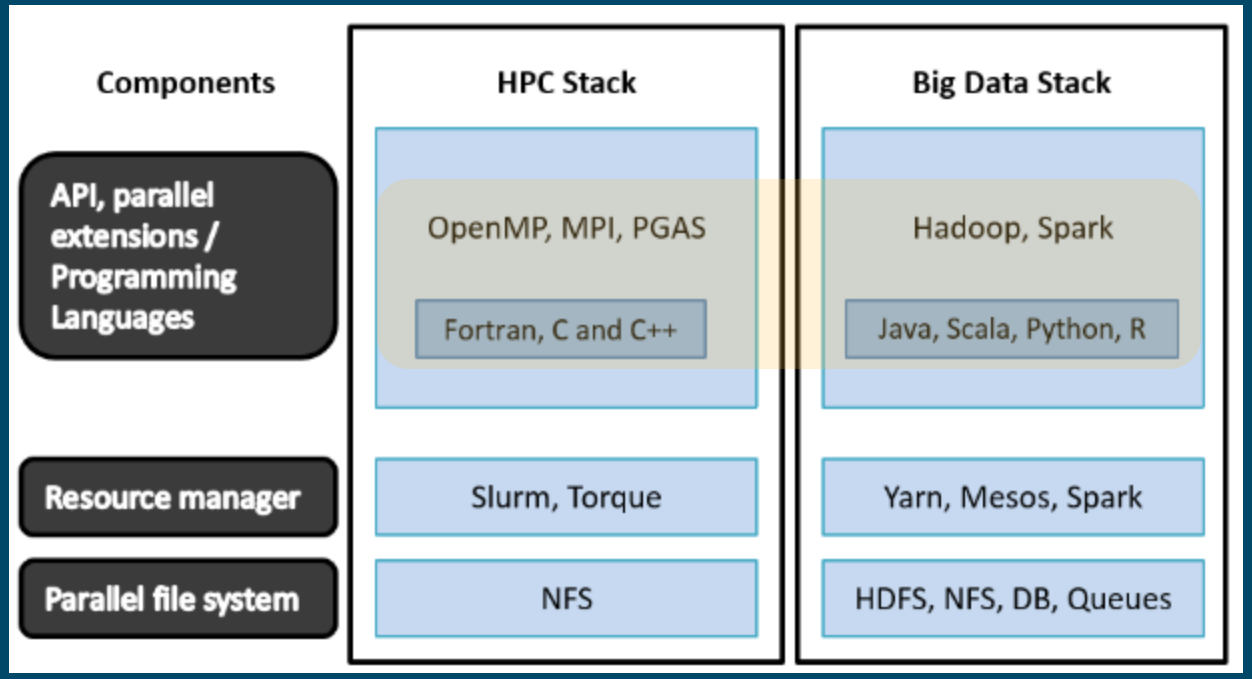
- The shared memory component is usually a cache on a machine.
- Processors can address that machine's memory as global.
- Network communications are required to move data from one computer to another.

Symmetric multiprocessing (SMP) , https://en.wikipedia.org/wiki/Symmetric_multiprocessing

HPC Parallel Programming

HPC vs Big Data

- HPC relies on the programmer more
- Hadoop/Spark provides an API



Feature	Hadoop	HPC
Scheduling	Custom scheduler FIFO/Capacity/Fairshare	System wide schedulers LSF/MOAB/Torque
Storage	Architected to work with DAS. HDFS[21][6] is the primary file system	Large NFS and Parallel file stores with very little DAS
Compute	Compute is moved to the nodes with stationary data, data moved across in stages	Custom to the application implementation
Interconnect	High speed interconnects are not mandated; architected to work well with Ethernet	Infiniband is the primary interconnect to ensure fast data movement
Services	Relies on services that control the execution of the flow and execution	Services are central to environment, typical applications do not expect custom daemons

Kashyap, S. N., Fewings, A. J., Davies, J., Morris, I., Green, A. T. T., & Guest, M. F. (2015). Big data athpc wales. *arXiv preprint arXiv:1506.08907*.

Programming using OpenMP

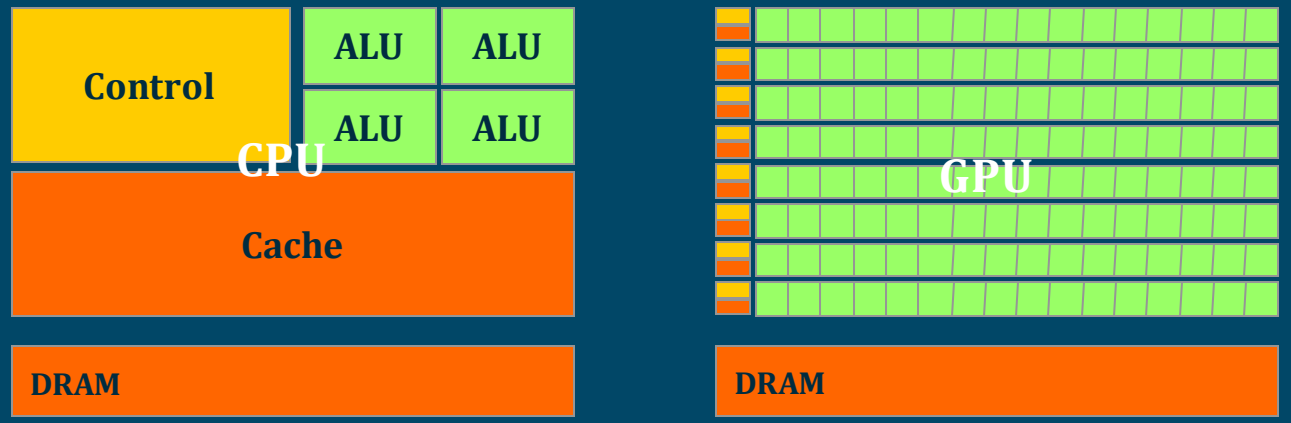
explicit thread programming C++

```
// Run one OpenMP thread per device per MPI node
#pragma omp parallel num_threads(devCount) if (initDevice())
 {
        // Block and grid dimensions
        dim3 dimBlock(12,12);
        kernel<<<1,dimBlock>>>();
        cudaThreadExit();
}
else
{
        printf("Device error on %s\n",processor_name);
}
        MPI_Finalize();
        return 0;
}
```

Very difficult to program, coordinating among threads

GPU programming adds further complications

- The GPU is specialized for compute-intensive, highly data parallel computation (exactly what graphics rendering is about)
  - transistors can be devoted to data processing rather than data caching and flow control



- The fast-growing video game industry exerts strong economic pressure that forces constant innovation

GPU programming example

Simple example (Matrx addition):

cpu C program:

```
void addVector (fload *a, float *b,
                fload *c, int N)
{
    int i, index,
    for (i = 0; i<N, i++) {
        c[index] = a[index] + b[index];
    }
}

void main()
{
   .....
   addVector(a, b, c, N);
   .....
}
```

cuda program:

```
__global__ void addVector (fload *a, float *b,
                           float *c)
{
    int i = threadIdx.x + blockDim.x*blockIdx.x;
    c[i] = a[i] + b[i];
}

Void main()
{
   ...
   // allocation & transfer data to GPU
   //Excuate on N/256 blocks of 256 threads each
   addVector << N/256, 256 >> ( d_A, d_B, d_C);
   .....
}
```

Device code

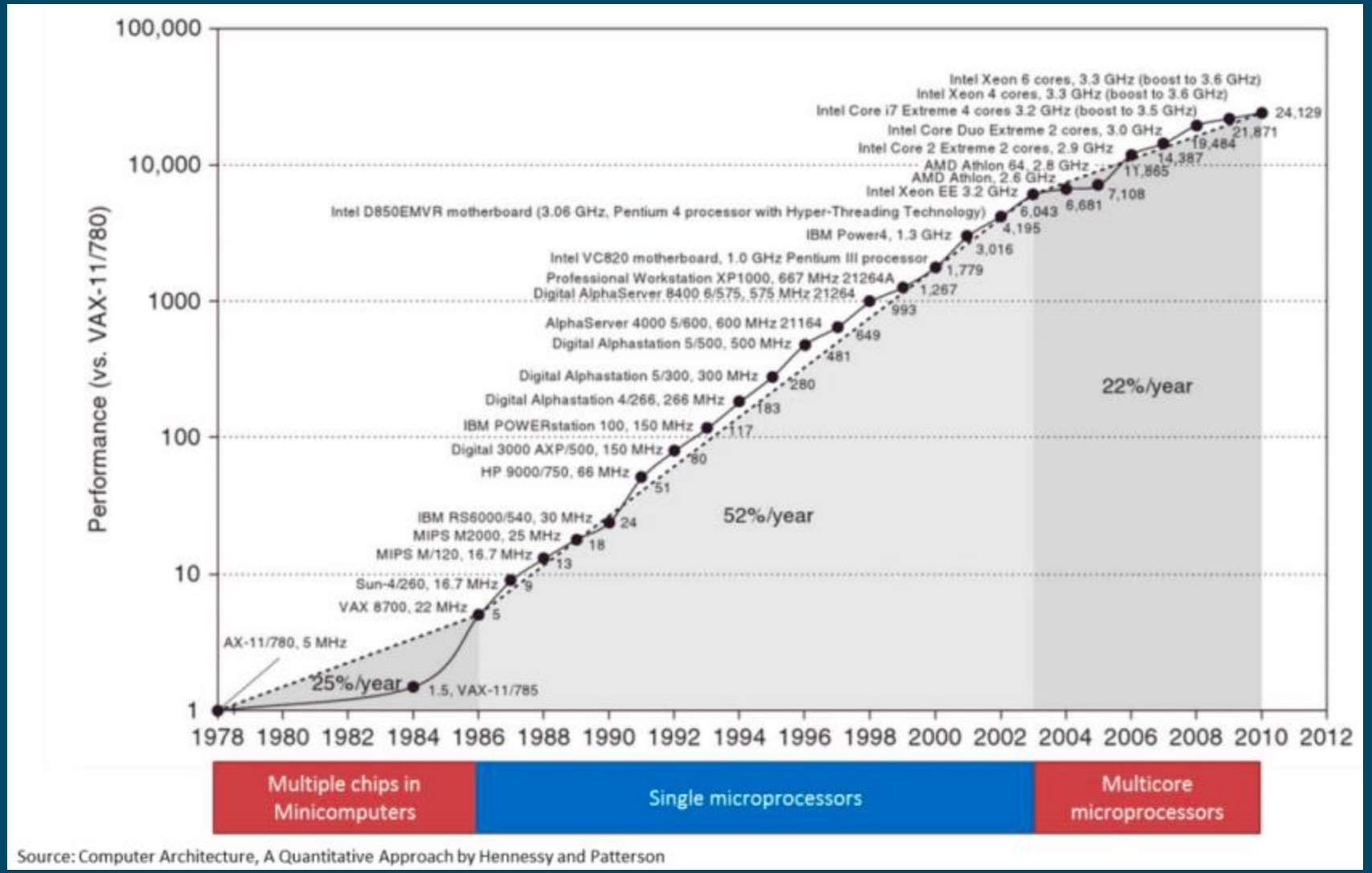
Host code

Big data clusters

Commodity microprocessors are like supercomputers processors (1995)

(instructions per second, not total parallel performance)

Commodity CPUs are faster



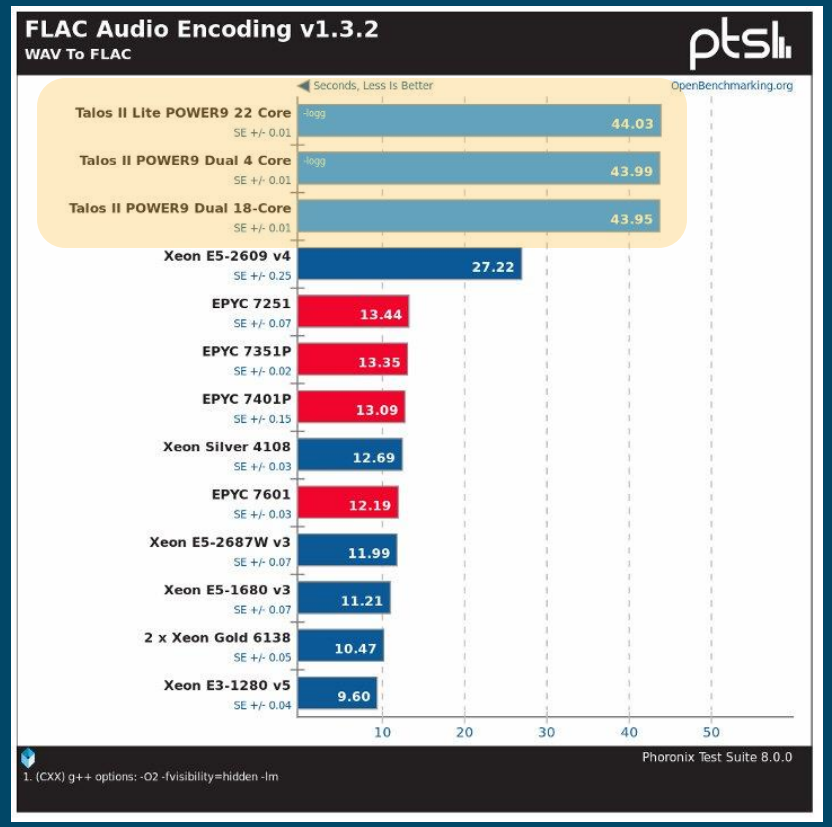
Source: Computer Architecture, A Quantitative Approach by Hennessy and Patterson

Supercomputers now use commodity CPUs

- IBM Summit

summit

Sponsors	U.S. Department of Energy
Operators	IBM
Architecture	9,216 POWER9 22-core CPUs 27,648 NVIDIA Tesla V100 GPUs[1]
Power	13 MW[2]
Operating system	Red Hat Enterprise Linux (RHEL)[3][4]
Storage	250 PB
Speed	200 petaFLOPS (peak)
Purpose	Scientific research
Web site	www.olcf.ornl.gov/olcf-resources/compute-systems/summit/



	hp intel	IBM POWER8	
Server list price -3-year warranty	$11,496	$13,730	$20,615
Linux OS list price	$2,968	$2,968	$2,986
Total List Price	$14,464	$16,698	$23,583
TCA Street Price*	$11,571	$14,045	$18,866
SPECint_Rate	475	642	708
Price/Perf	$24.36	$21.88	$26.65
Server model	HP Proliant DL180	IBM Power S812LC	IBM Power S812L
Processor / cores	Haswell 10-core 2.6GHz processors	10-core 2.9GHz POWER8	10-core 3.4GHz POWER8
Configuration	256 GB memory, 2x1TB HDD	Same memory, HDD,	Same memory, HDD

New!

HPC vs commodity cluster

- HPC
  - Faster, but much is custom, and thus take much time and money
    - Many more lines of code for programming (means more cost)
    - Much more expensive hardware (e.g., network interconnect)
- Commodity cluster
  - Better ROI and yet appropriate for most applications
  - Exceptions may be numerical simulations, like weather and nuclear reactions using Monte Carlo simulations

HPC and now commodity cluster

- Before Hadoop (commodity cluster)
  - Parallel computing on specialized (expensive) hardware
  - Programming via complex libraries (e.g., C++)
    - Programmer directly manages parallel processes
- After Hadoop (which simplifies computing)
  - Parallel computing on commodity (inexpensive) hardware
  - Programming via common patterns (map and reduce)
    - Programmer declares parallel data and code, which is managed by infrastructure
      - Language constructs for parallelism (reduceByKey)
      - Process controls for managing tasks, including memory & process allocation

Embarrassingly parallel computation

- Independently process data partitions
- Hadoop (Spark) supports processing of very large datasets to exploit (mostly) embarrassingly parallel computation
  - Little or no manipulation is needed to separate the problem into parallel tasks
  - Occurs where there is little or no dependency or need for communication between parallel tasks

https://en.wikipedia.org/wiki/Embarrassingly_parallel

Partitioning for parallelism

- Break the problem into discrete "chunks" of work that can be distributed to multiple tasks.
  - Also known as decomposition or partitioning.
- There are two basic ways to partition computational work among parallel tasks:
  - Data decomposition
    - Run the same program on multiple data partitions
  - Functional (process) decomposition
    - Run a different program on the data, often in a pipeline/stream where outputs of the first process and passed as inputs to the second

Basic Hadoop view

Pipeline view

Commodity computing example

Important to remember

- HPC is faster at numerical simulations, but requires more effort and time
- Commodity clusters have good ROI and are appropriate for embarrassingly parallel computations
  - Like web search indexing & machine learning algorithms