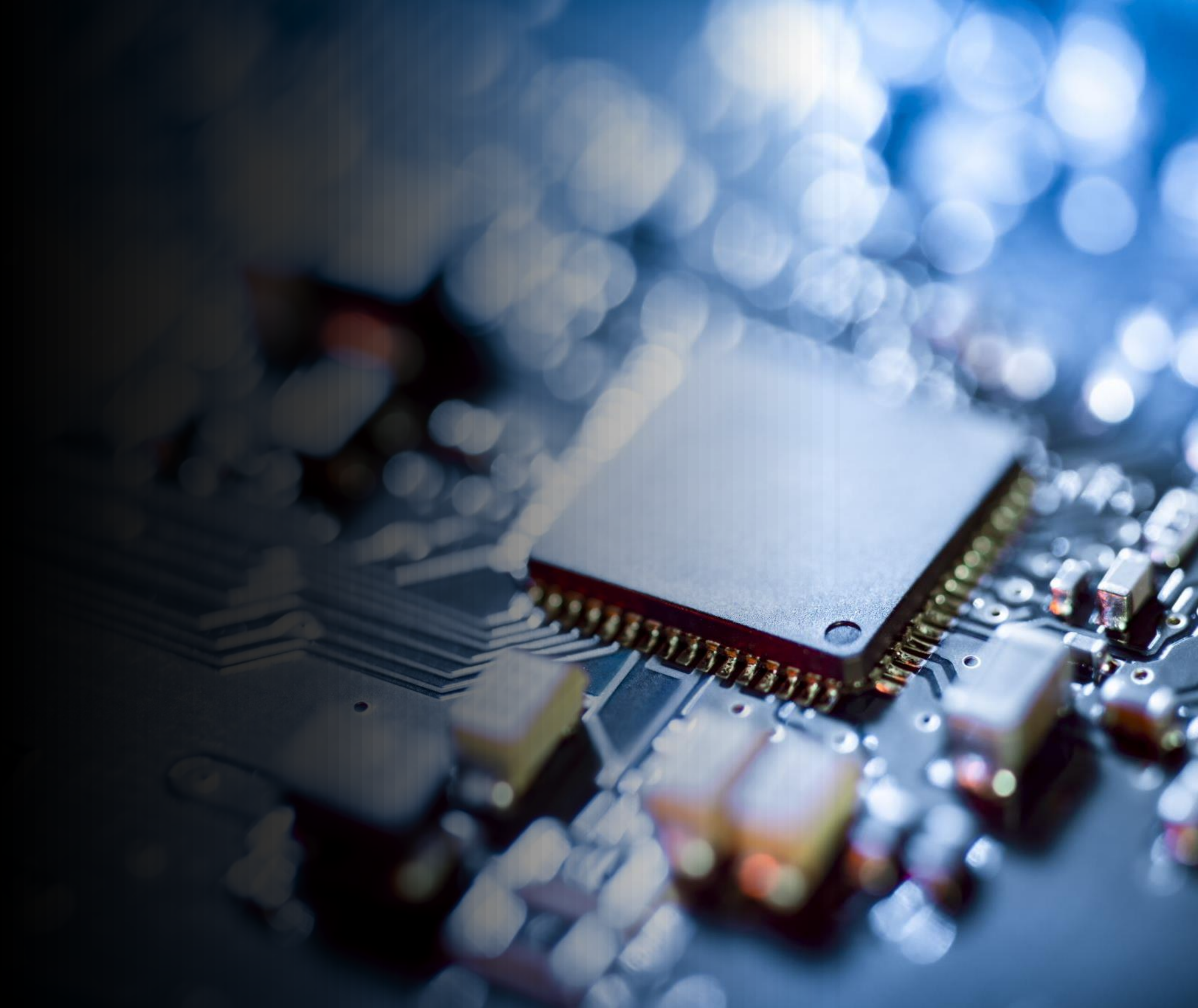
The World of Parallel Programming

From the Transistor to the Cluster
DERFAST - B.3.

By Diego Roa

Overview

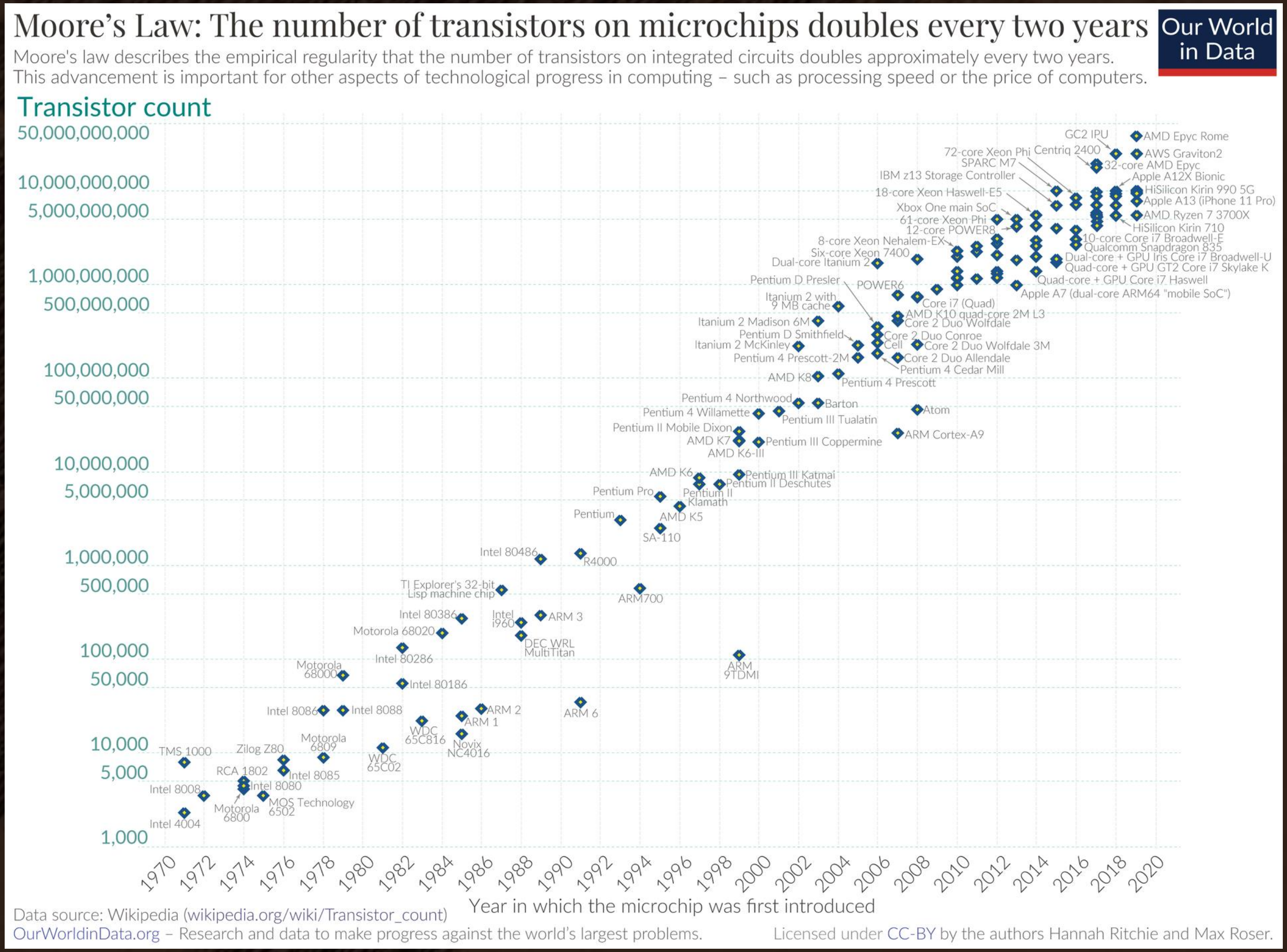
- Parallelism
- Flynn’s Taxonomy
- Programming Models
- Top500



Moore’s Law

Moore's law is the observation that <u>the number of transistors in a dense integrated circuit (IC) doubles about every two years.</u>

Moore's law is an observation and projection of a historical trend. Named after Gordon Moore, the co-founder of Fairchild Semiconductor and Intel.



Moore’s Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

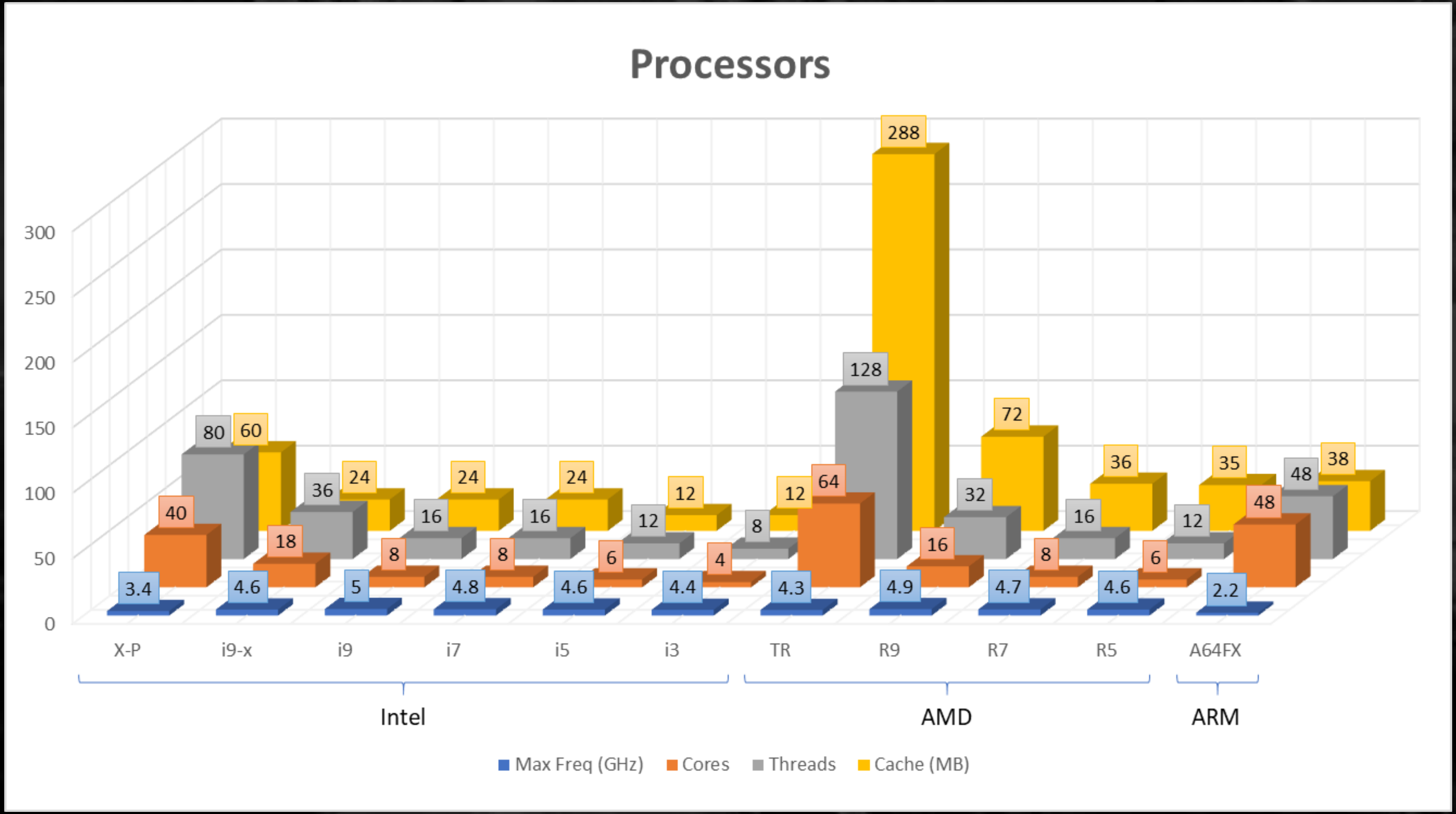
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world’s largest problems.

Parallelism

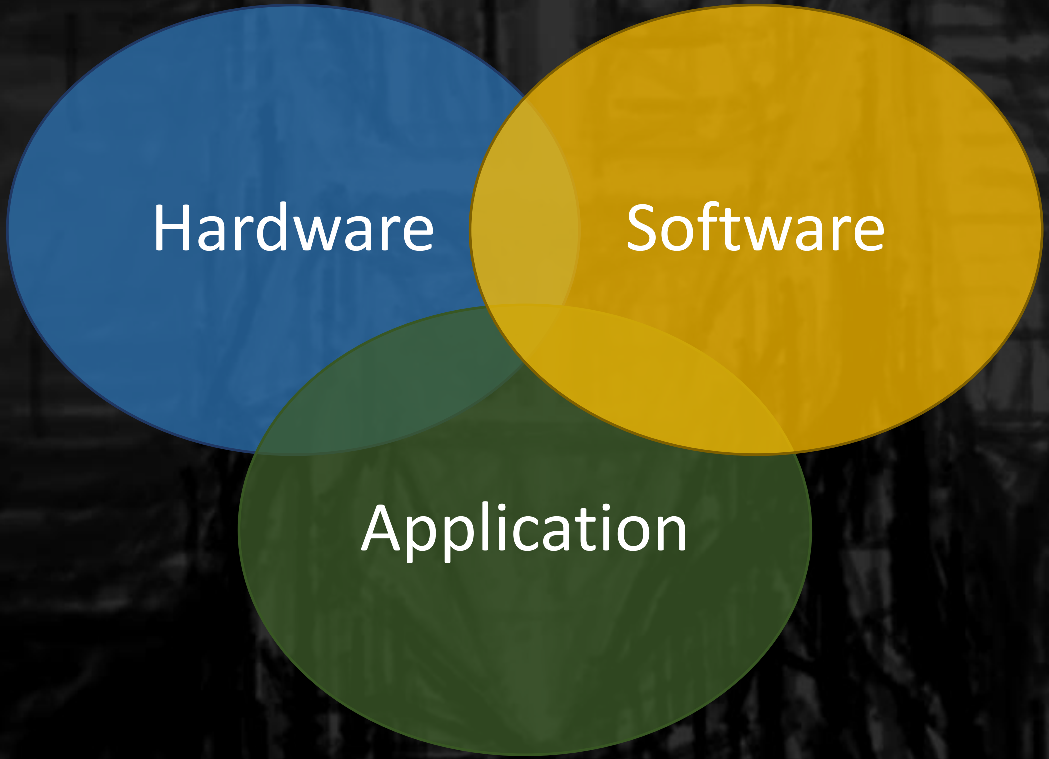
Computer's performance used to be improved by increasing clock frequency

Physical constrains limit the maximum frequency and power usage



Most modern computers are parallel

Parallelism



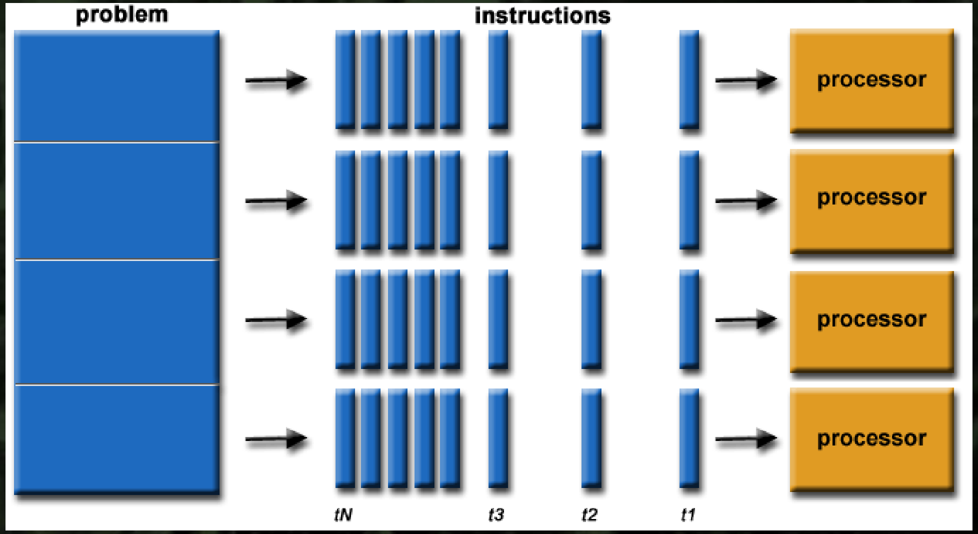
Can be present at different levels

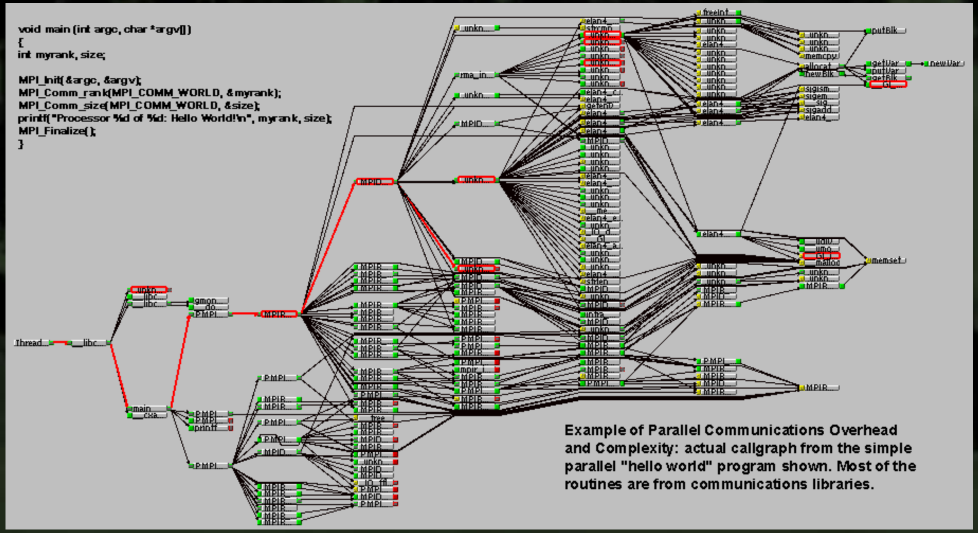
Can be introduced by the user, the compiler, or the system

Program Parallelism

From a program perspective, there is parallelism at the level of:

- Programming Language
- Vector expressions
- Program blocks, such as loops
- Tasks (program sub-activities, with a limited duration)
- Processes (that may live as long as the program)



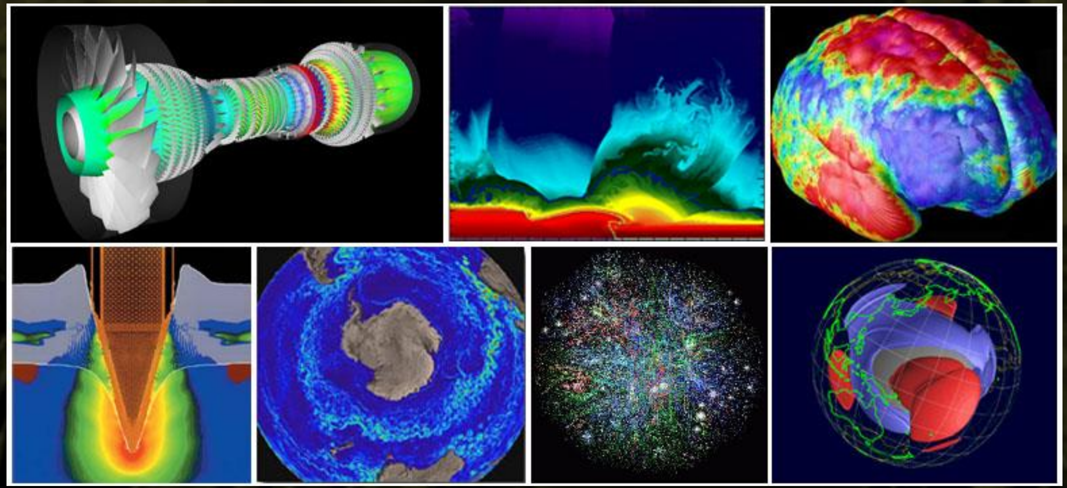


```
void main (int argc, char *argv[])
{
int myrank, size;

MPI_Init( &argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
printf("Processor %d of %d: Hello World!\n", myrank, size);
MPI_Finalize();
}
```

Example of Parallel Communications Overhead and Complexity: actual callgraph from the simple parallel "hello world" program shown. Most of the routines are from communications libraries.

Application Parallelism



Parallelism can also be exploited at the application level. This can be:

- Intrinsically parallel applications
- User-managed: workflows, such as parameter sweeps
- System-managed: run queue

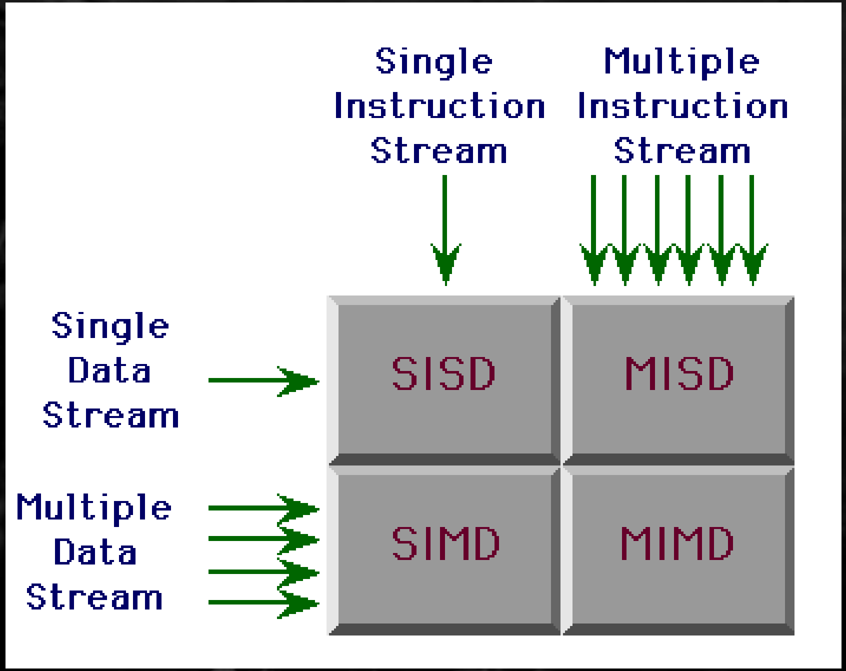
As parallel computers become larger and faster, we are now able to solve problems that had previously taken too long to run.

Hardware Parallelism

From a hardware point of view, there is parallelism at the level of:

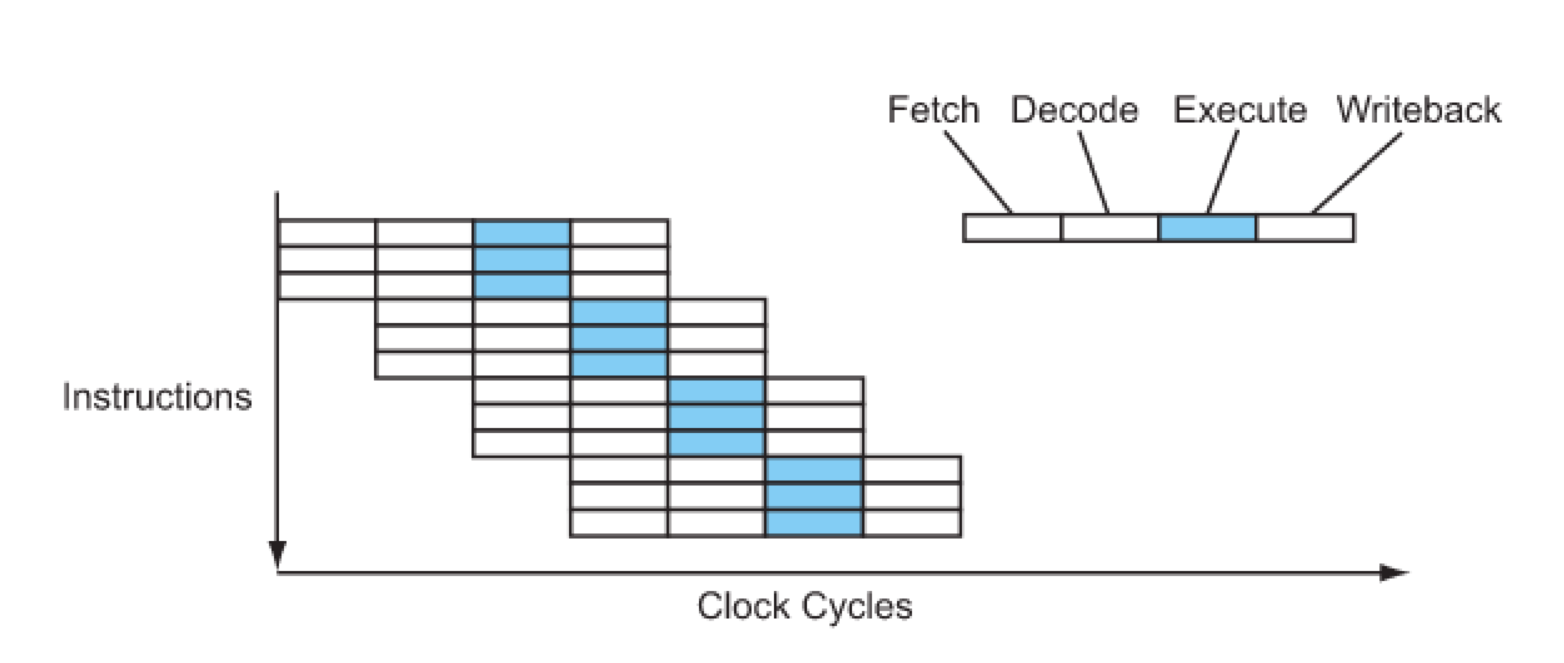
- Instructions
- Data I/O
- Multiple Processors (cores)
- Multiple Nodes (Distributed)
- Clusters
- Grid Computing

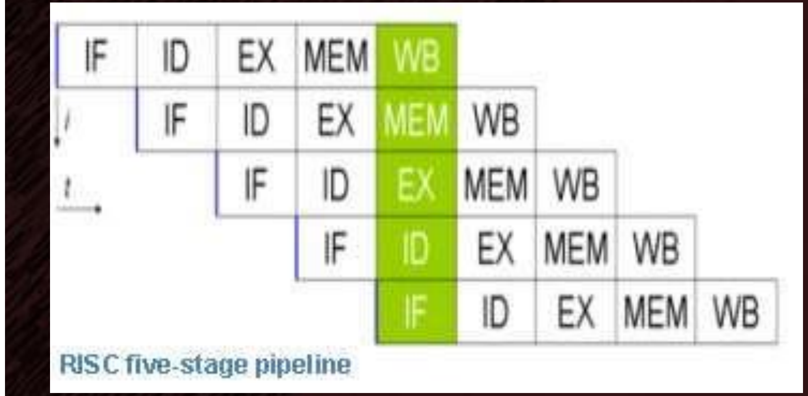
Flynn’s Taxonomy



- **SISD** (Single instruction, single data): Sequential Computer
- **SIMD** (Single instruction, multiple data): A single instruction is simultaneously applied to multiple different data streams.
- **MISD** (Multiple instruction, single data): Heterogeneous systems operate on the same data stream and must agree on the result.
- **MIMD** (Multiple instruction, multiple data): Multiple autonomous processors simultaneously executing different instructions on different data.

Instruction Level Parallelism (ILP)





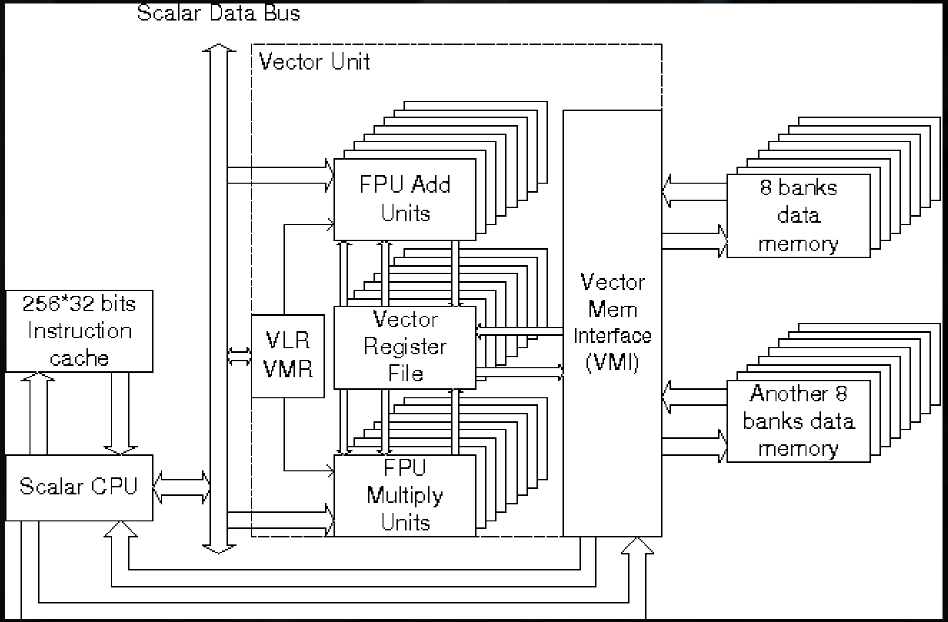
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

RISC five-stage pipeline

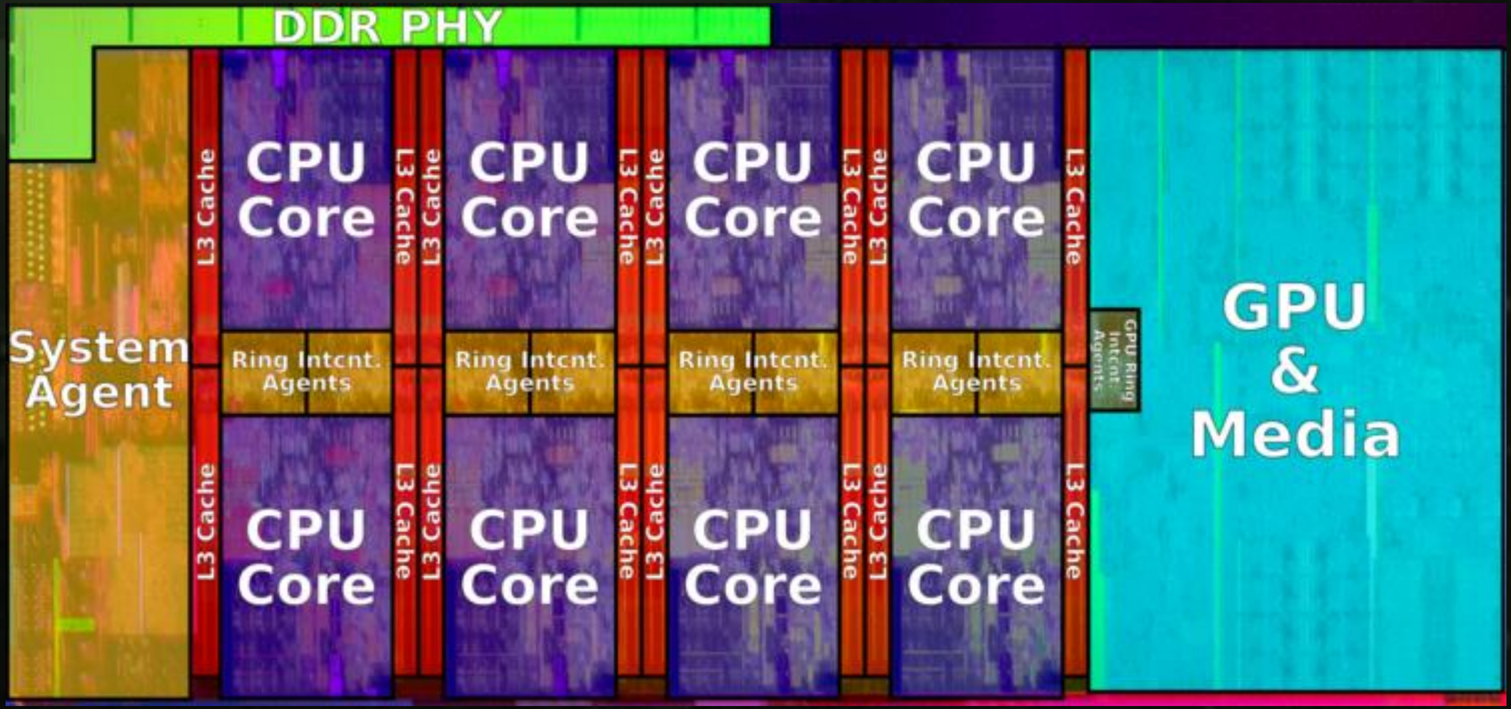
Parallel or simultaneous execution of a sequence of instructions in a computer program. ILP refers to the average number of instructions run per step of this parallel execution. System mechanisms identify and enforce dependencies

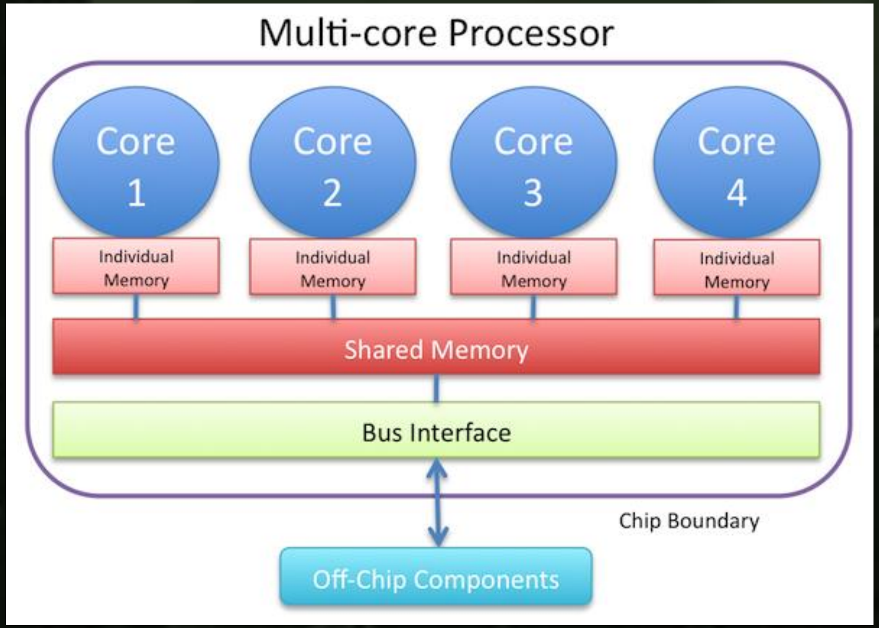
Vector Machines

A central processing unit (CPU) that implements an instruction set designed to **operate** efficiently and effectively **on large one-dimensional arrays of data called vectors**



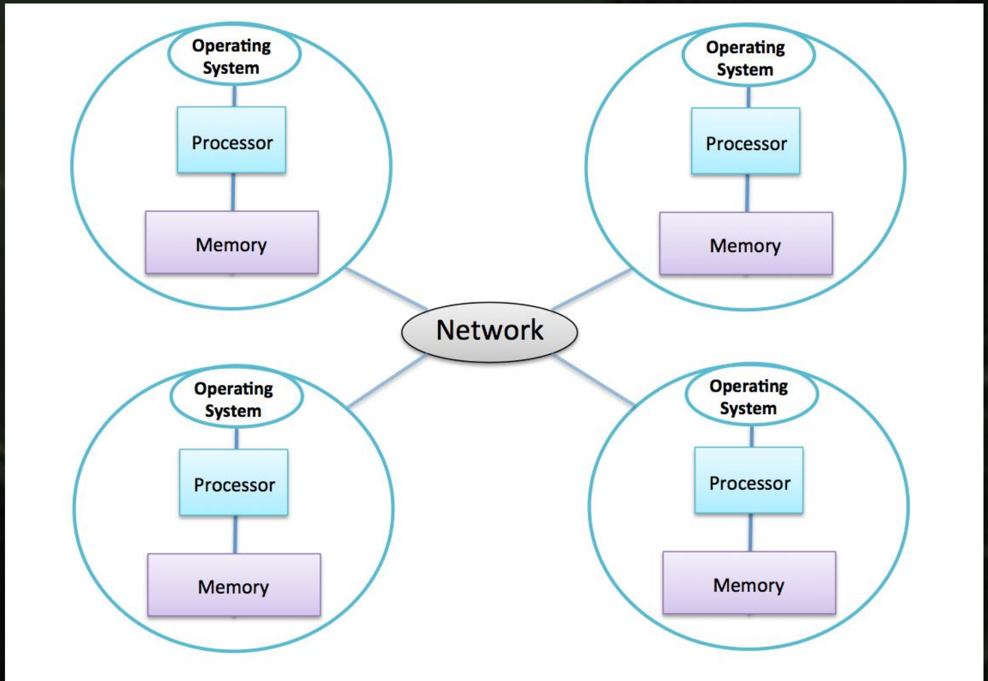
Multi-core / Shared-Memory

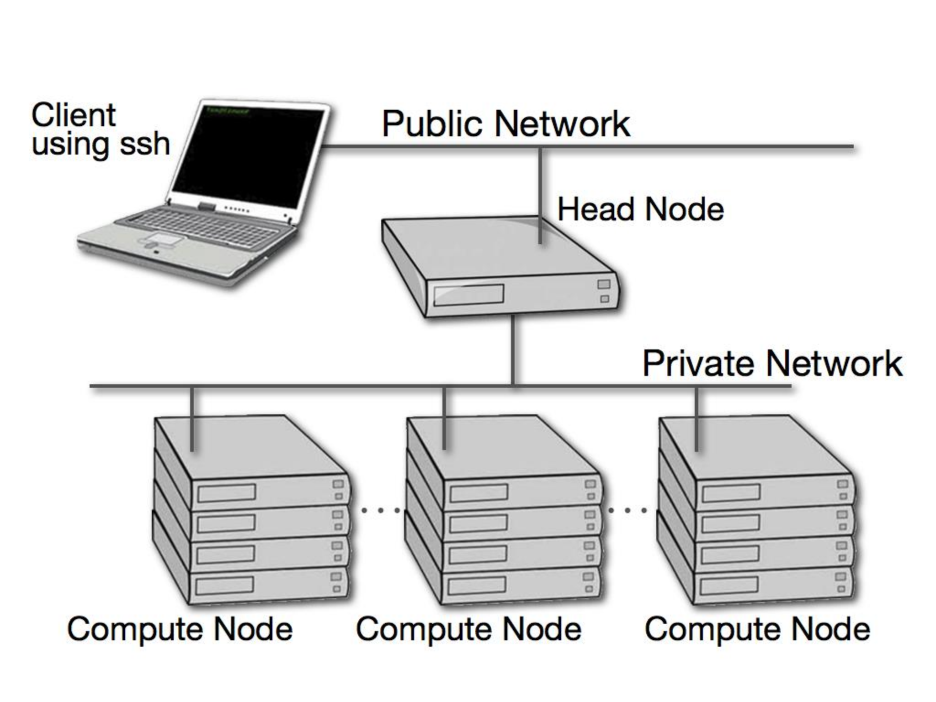




A computer processor on a single integrated circuit with two or more separate processing units, called cores, each of which reads and executes program instructions. A shared memory multiprocessor offers a single memory space used by all processors

Multi-node / Distributed-Memory

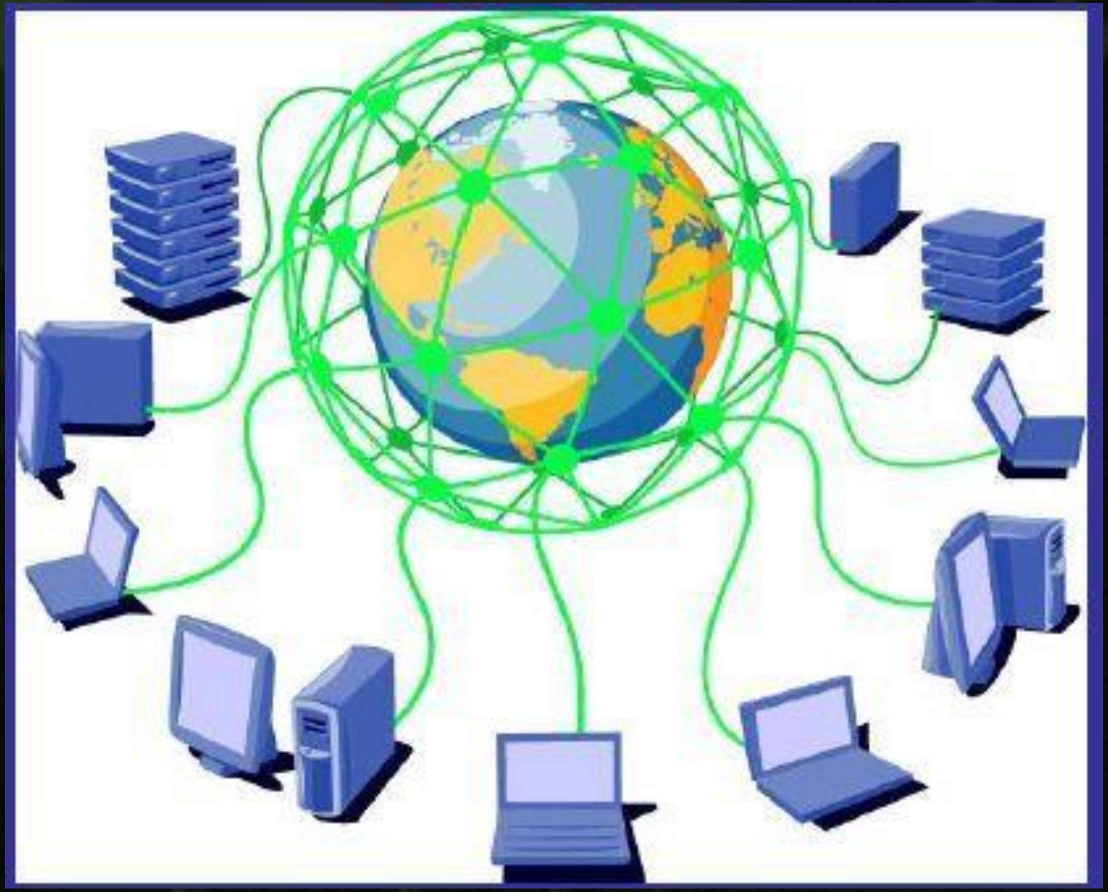




Each processor has its own private memory.

Computational tasks can only operate on local data, and if remote data are required, the computational task must communicate with one or more remote processors.

Grid Computing

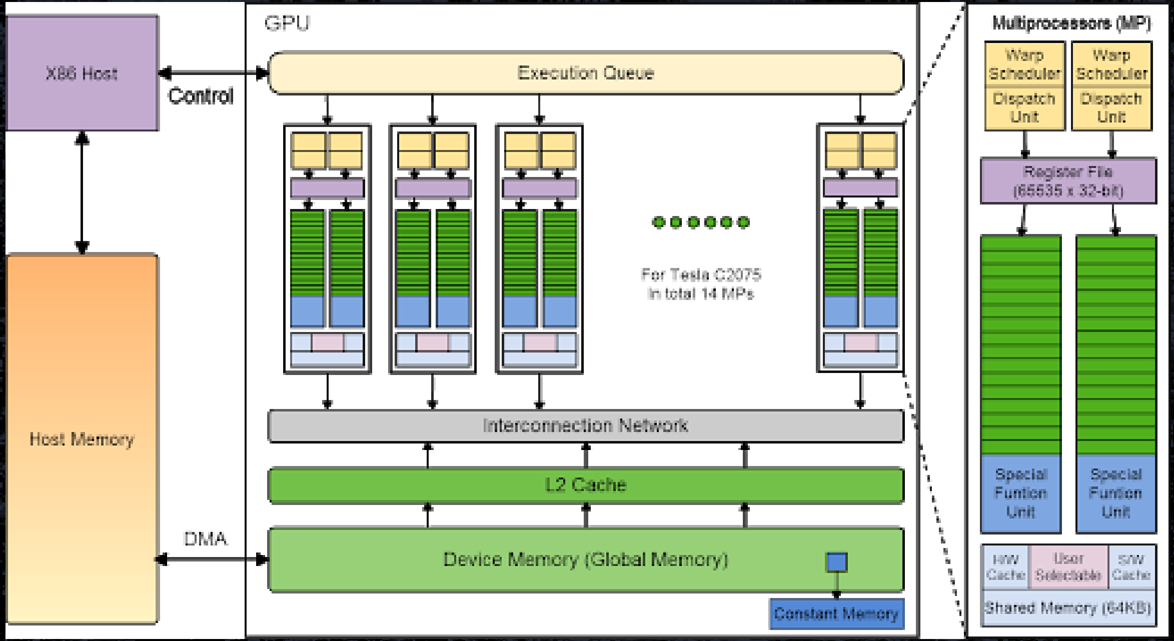


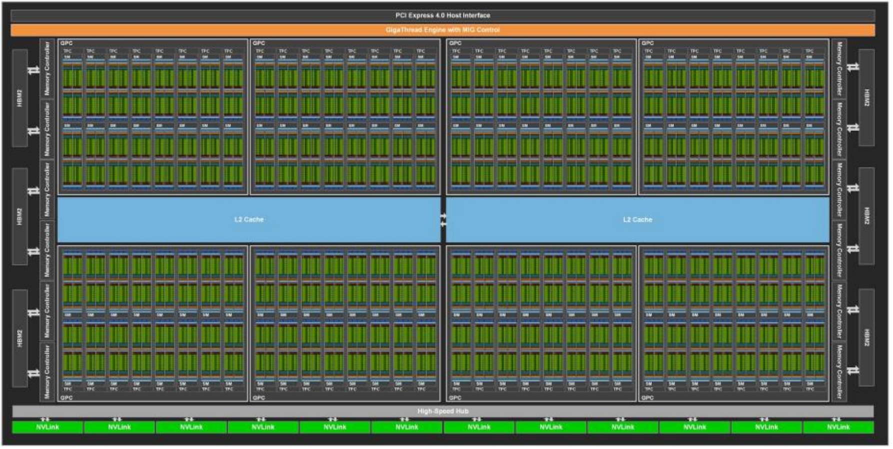
The use of widely distributed computer resources to reach a common goal.

Grid computers tend to be more heterogeneous and geographically dispersed than cluster computers.

CPU scavenging and volunteer computing were popularized beginning in 1997 by distributed.net and later in 1999 by SETI@home to harness the power of networked PCs worldwide, in order to solve CPU-intensive research problems.

Graphics Processing Unit (GPU - GPGPU)

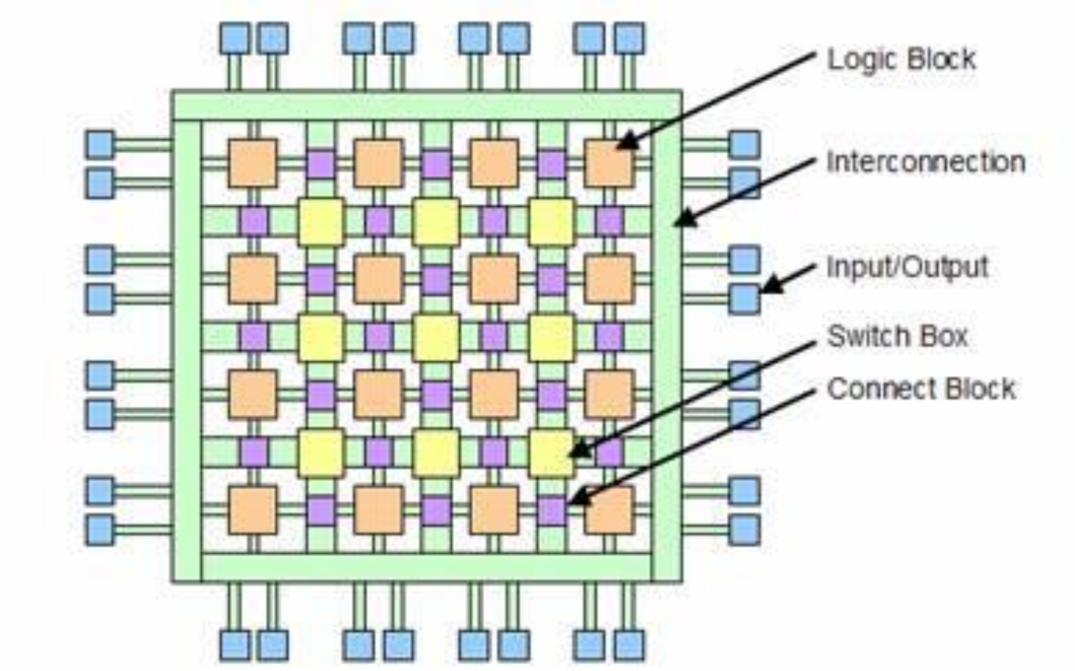




Is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.

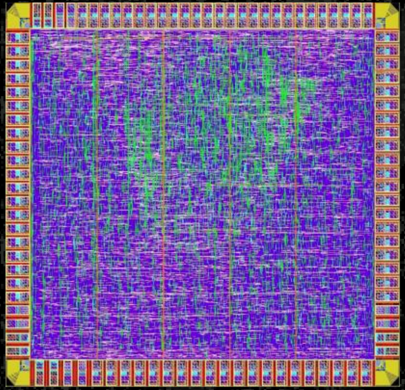
GPGPU pipeline is a kind of parallel processing between one or more GPUs and CPUs that analyzes data organized in large arrays in the same way than images or graphics.

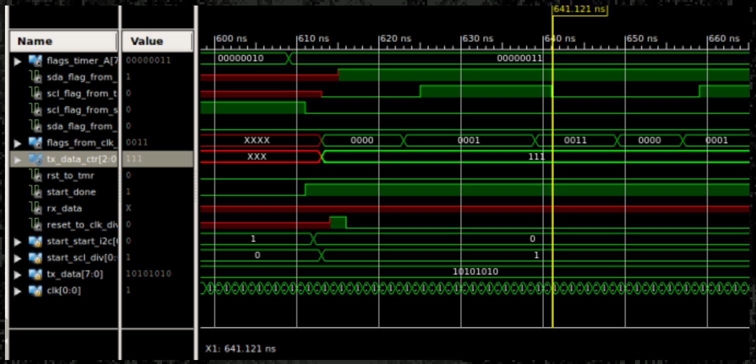
FPGA



A field-programmable gate array (FPGA) is an **integrated circuit designed to be configured by a customer or a designer after manufacturing**.

FPGAs contain an **array of programmable logic blocks**, and a hierarchy of "reconfigurable interconnects" allowing blocks to be "wired together", like many logic gates that can be inter-wired in different configurations.

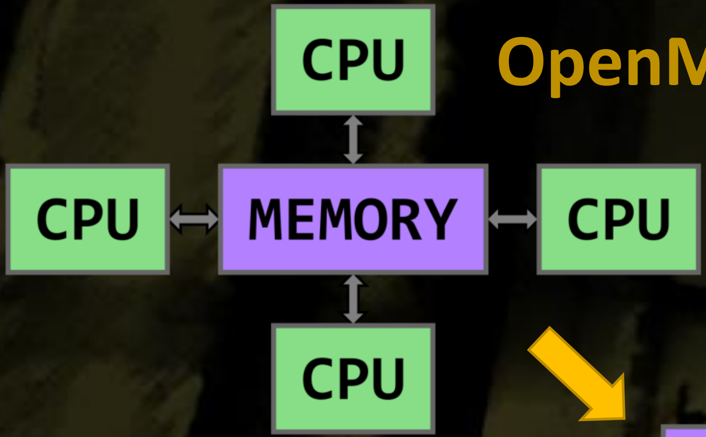




Name	Value
flags_timer_A[7	00000011
sda_flag_from_	1
scl_flag_from_t	0
scl_flag_from_s	0
sda_flag_from_	0
flags_from_clk_	0011
tx_data_ctr[2:0	111
rst_to_tmr	0
start_done	1
rx_data	X
reset_to_clk_div	0
start_start_i2c[(	0
start_scl_div[0:	1
tx_data[7:0]	10101010
clk[0:0]	1

Programming Models

OpenMP



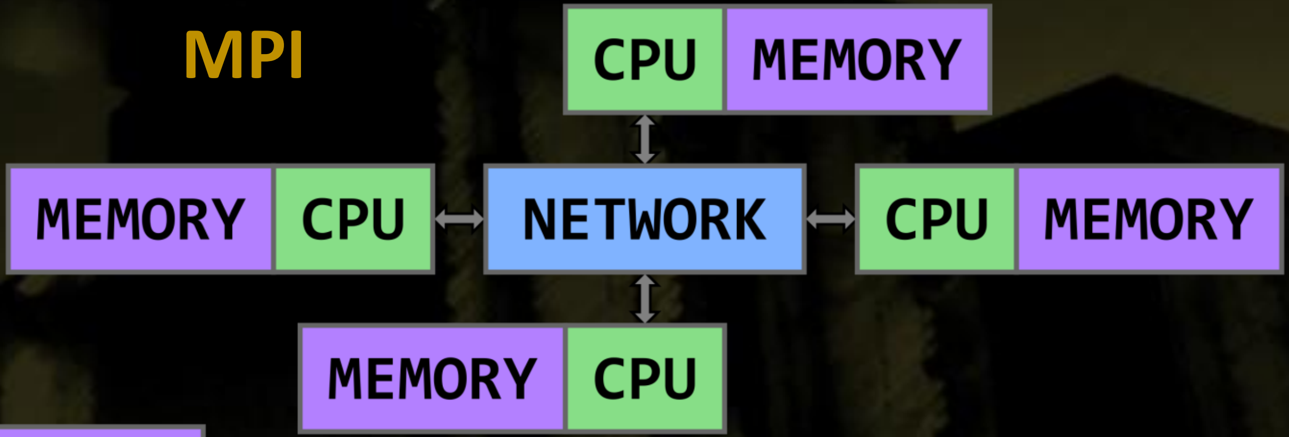
CPU

CPU MEMORY CPU

CPU

Shared Memory

MPI

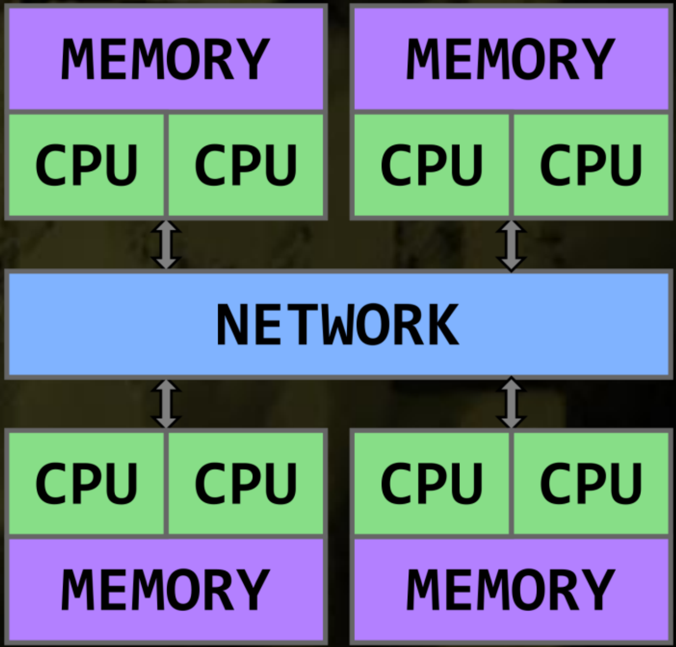


CPU MEMORY

MEMORY CPU NETWORK CPU MEMORY

MEMORY CPU

Distributed Memory



MEMORY MEMORY

CPU CPU CPU CPU

NETWORK

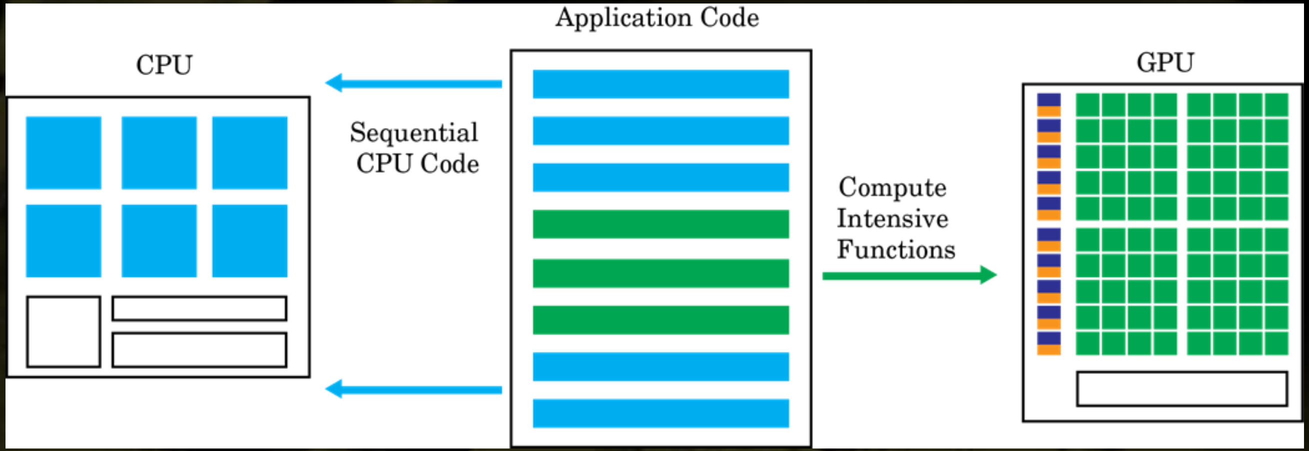
CPU CPU CPU CPU

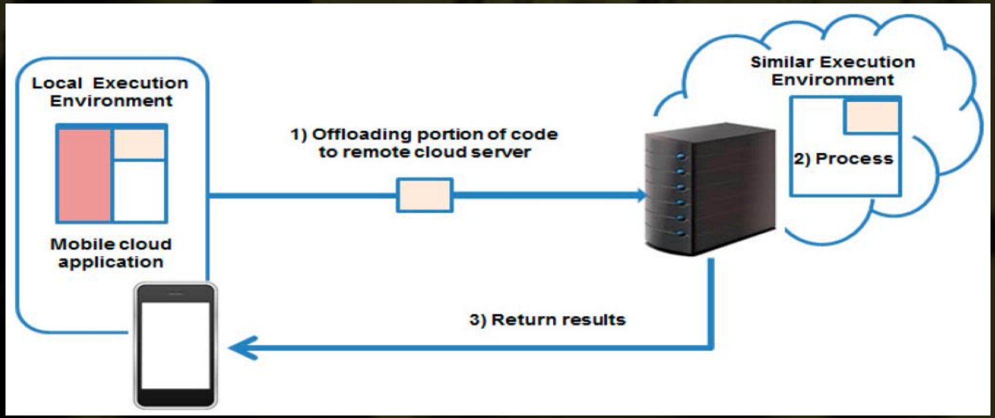
MEMORY MEMORY

Distributed-Shared Memory

Programming Models

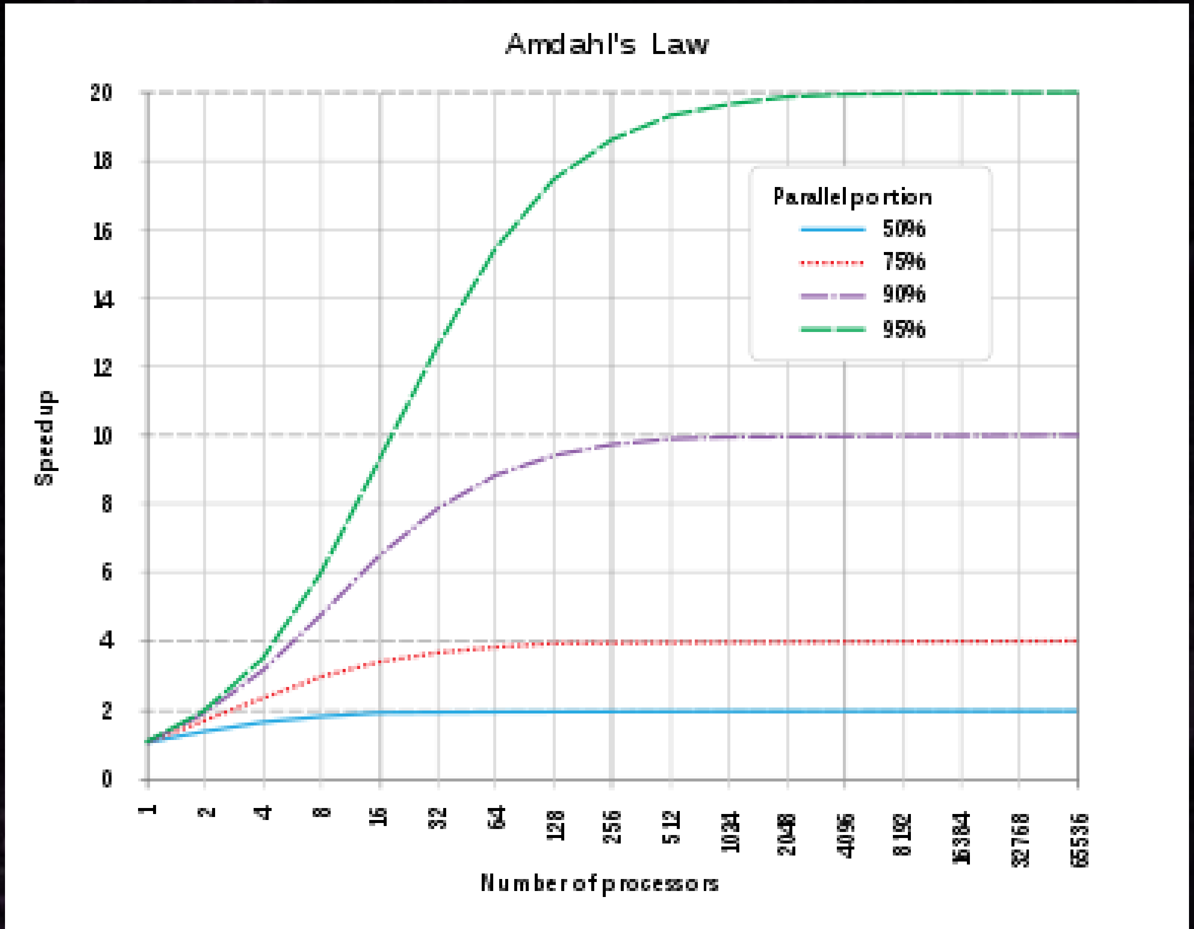
Offloading




Local Execution Environment
Mobile cloud application
1) Offloading portion of code to remote cloud server
Similar Execution Environment
2) Process
3) Return results

Computation offloading is the **<u>transfer of resource intensive computational tasks to a separate processor</u>**, such as a hardware **<u>accelerator</u>**, or an external platform, such as a cluster, grid, or a cloud.

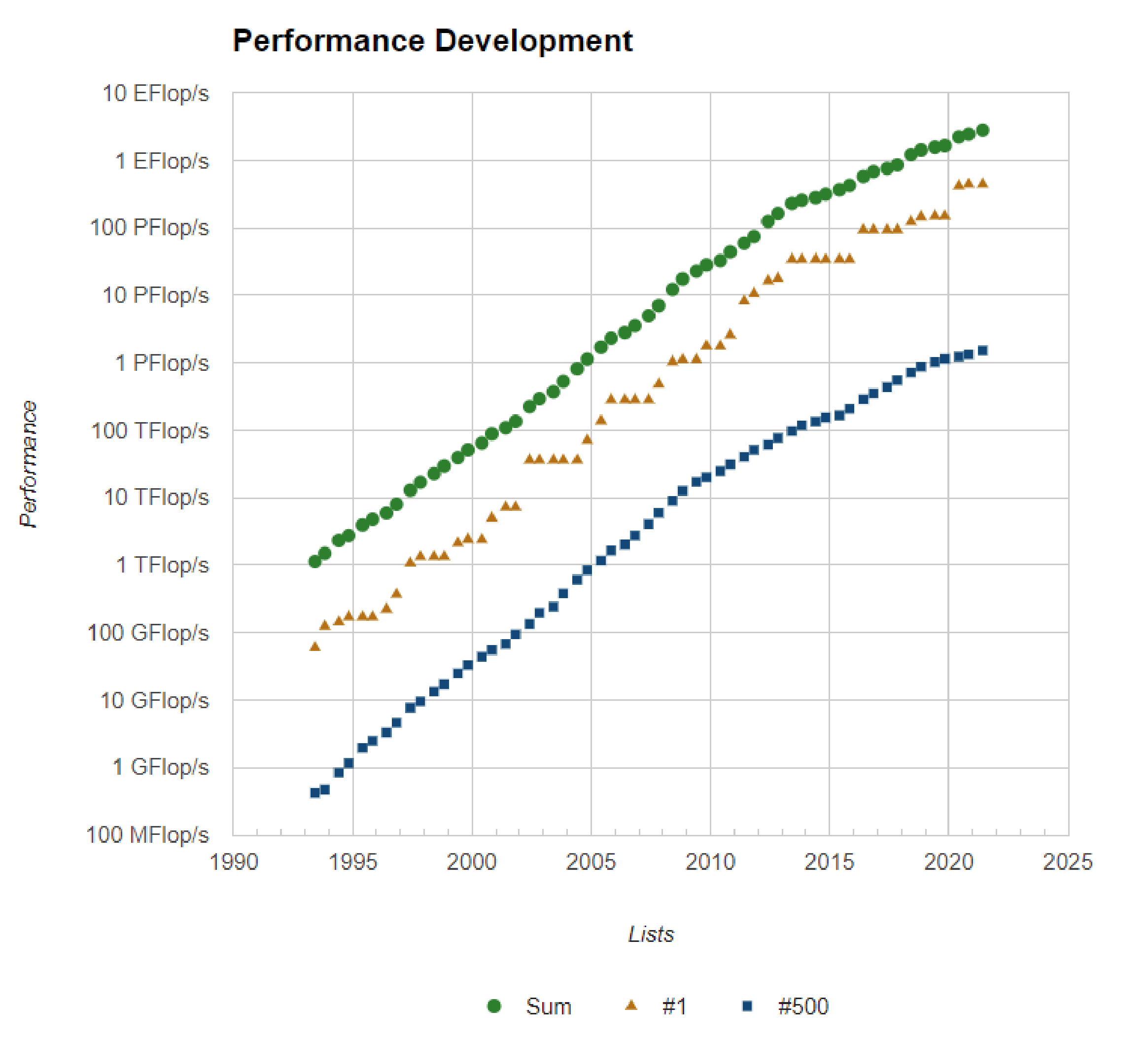
Amdahl's Law



Is a formula which gives the **theoretical speedup** in latency of the execution of a task at fixed workload that can be expected of a system when using multiple processors.

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

Top 500



The main objective of the TOP500 list is to provide a ranked list of general-purpose systems that are in common use for high end applications.

General purpose system means that the computer system must be able to be used to solve a range of scientific problems.

Top 500