

JORGE CHAM © 2016

WWW.PHDCOMICS.COM

Predictive Modelling & Analytics

Rosalind Wang

Spring 2022 (week 10)

What we will be doing

Week	Lecture	Assessment due
Week 1 28-07-2022	Introduction to the unit. Using Python	
Week 2 04-08-2022	Maths revision	
Week 3 11-08-2022	Gradient Descent	
Week 4 18-08-2022	SGD and variants, linear and log regression	Quiz 1
Week 5 25-08-2022	Regularisation and motivation, Ridge Regression.	
Week 6 01-09-2022	Feature selection and regularisation. Lasso regression.	Quiz 2. Assignment part 1
Week 7 08-09-2022	Neural Networks (I)	
Week 8 15-09-2022		Quiz 3
Week 9 22-09-2022	<i>public holiday</i>	
Week 10 29-09-2022	Practical machine learning	
Week 11 06-10-2022	Neural Networks (II)	
Week 12 13-10-2022	Support Vector Machines	Quiz 4
Week 13 20-10-2022	Naive bayes	
Week 14 27-10-2022	Semi-supervised learning	Quiz 5, Assignment part 2.

Today

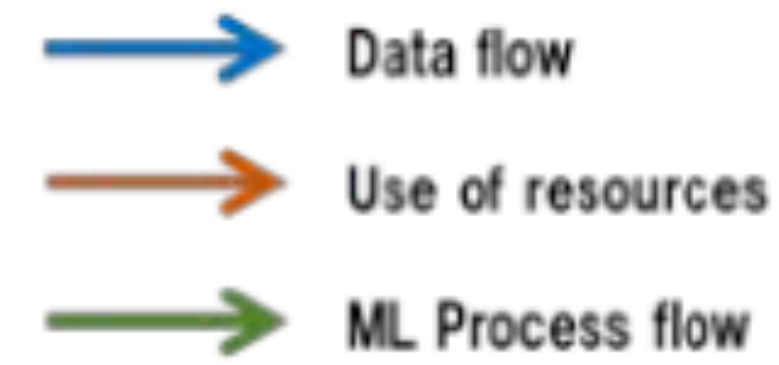
Parallel programming

Accelerated programming on GPU

Running jobs on HPC

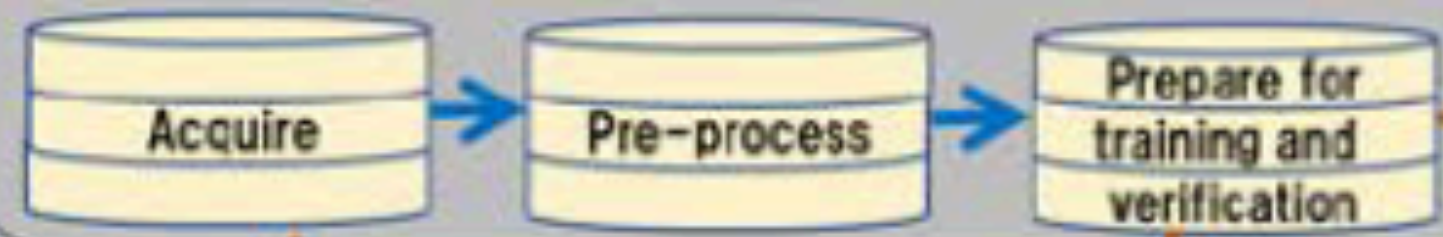
Some slides today based on material from ADACS and NVIDIA courses.

Data Science Workflow

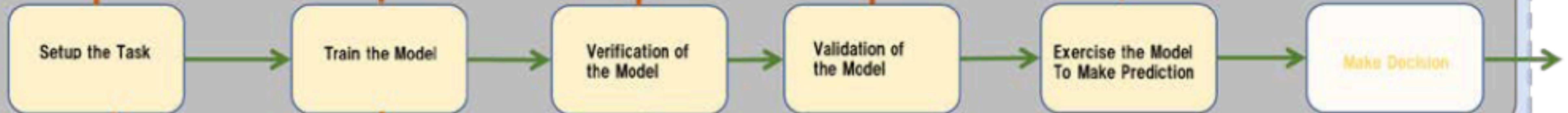


Model Creation

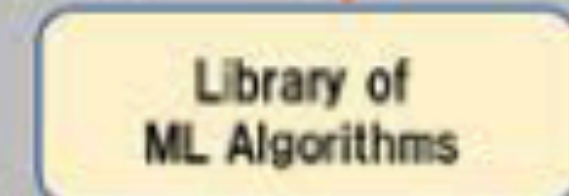
Data



Knowledge acquisition and application



Tools



Model Evaluation

Evaluation metrics

Model Use

input Data

Make Decision

Parallel Programming

Modern computers have multiple cores, which can be used at the same time for computing jobs

Parallel programming — making use of resources

Optimisation — making better use of the resources

Task based parallelism

Single core task, exclusive node use

	CPU cores							
Socket 1	1	2	3	4	5	6	7	8
	9	10	11	12	13	14	15	16
Socket 2	17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31	32

Single core task, shared node use

	CPU cores							
Socket 1	1	2	3	4	5	6	7	8
	9	10	11	12	13	14	15	16
Socket 2	17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31	32

Single core task, with job packing

	CPU cores							
Socket 1	1	2	3	4	5	6	7	8
	9	10	11	12	13	14	15	16
Socket 2	17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31	32

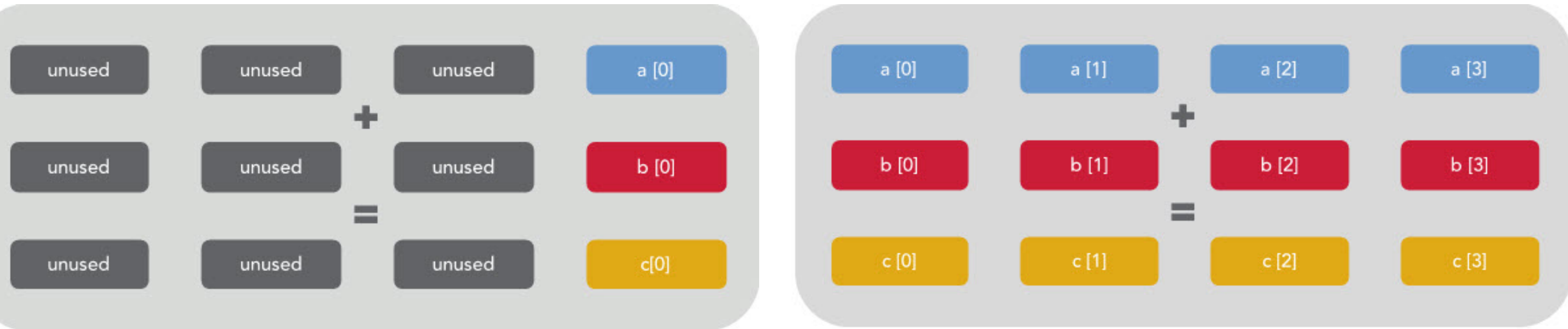
Core status

	In use
	Not available
	Idle

demo: job packing with xargs

Vectorised operations

Given the task $\mathbf{a} + \mathbf{b} = \mathbf{c}$, we can do it sequentially, or in vectorised form.



Many python libraries uses this to optimise the operations:

- numpy, scipy, scikit-learn, scikit-image

When possible, use ready-built libraries.

Domain or data based parallelism

Data	0	1	2	3	4	5	6	7	8
CPU0	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)	f(8)
Time	1	2	3	4	5	6	7	8	9

Data	0	1	2	3	4	5	6	7	8
CPU0	f(0)	f(1)	f(2)	Done!					
CPU1	f(3)	f(4)	f(5)						
CPU2	f(6)	f(7)	f(8)						
Time	1	2	3	4	5	6	7	8	9

Parallel the task over multiple cores.

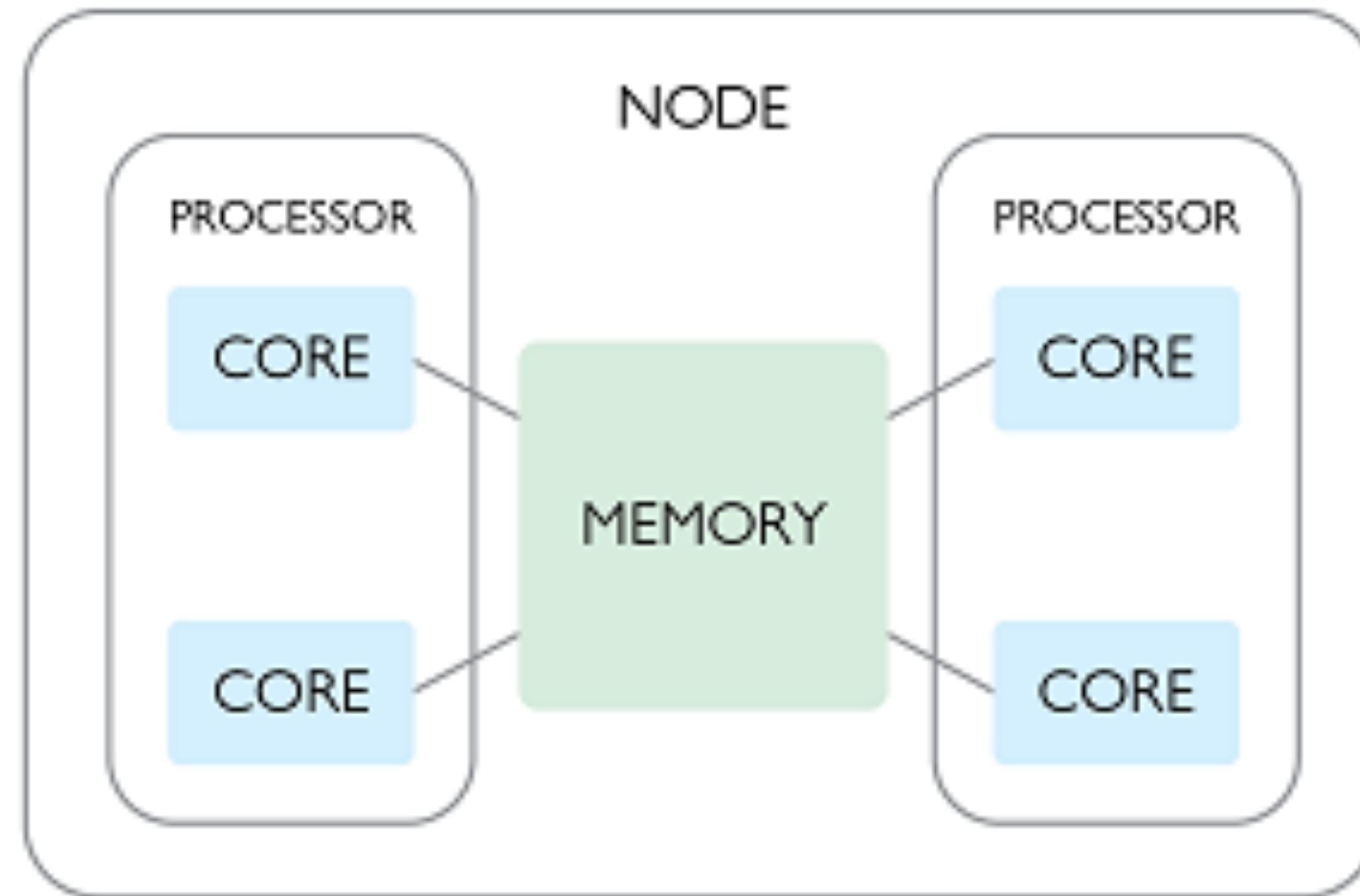
Domain or data based parallelism

Data	0	1	2	3	4	5	6	7	8
CPU0	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)	f(8)
CPU1	g(0)	g(1)	g(2)	g(3)	g(4)	g(5)	g(6)	g(7)	g(8)
CPU2	h(0)	h(1)	h(2)	h(3)	h(4)	h(5)	h(6)	h(7)	h(8)
Time	1	2	3	4	5	6	7	8	9

Parallel different tasks over multiple cores.

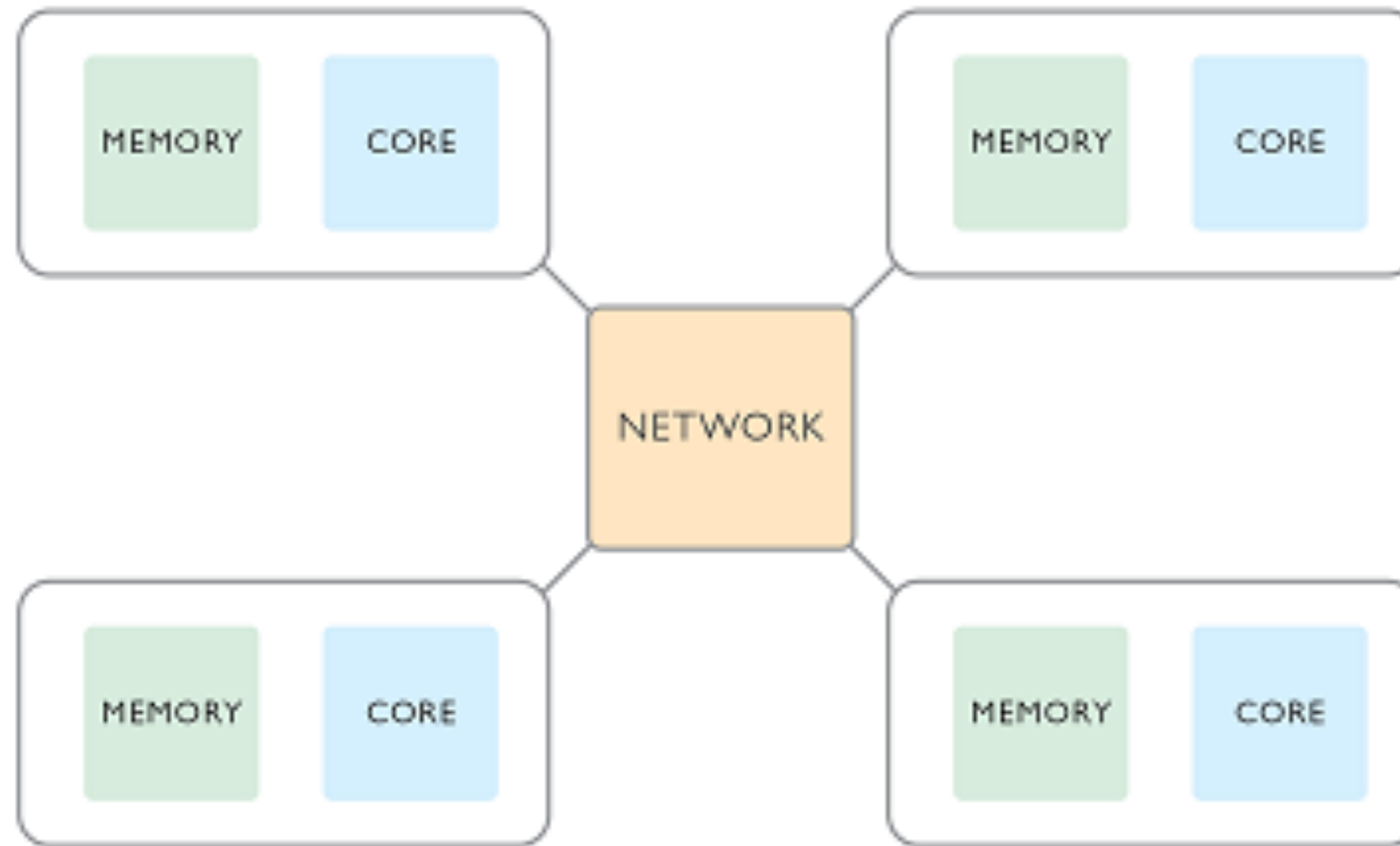
Parallel processing with shared memory

Shared memory computer



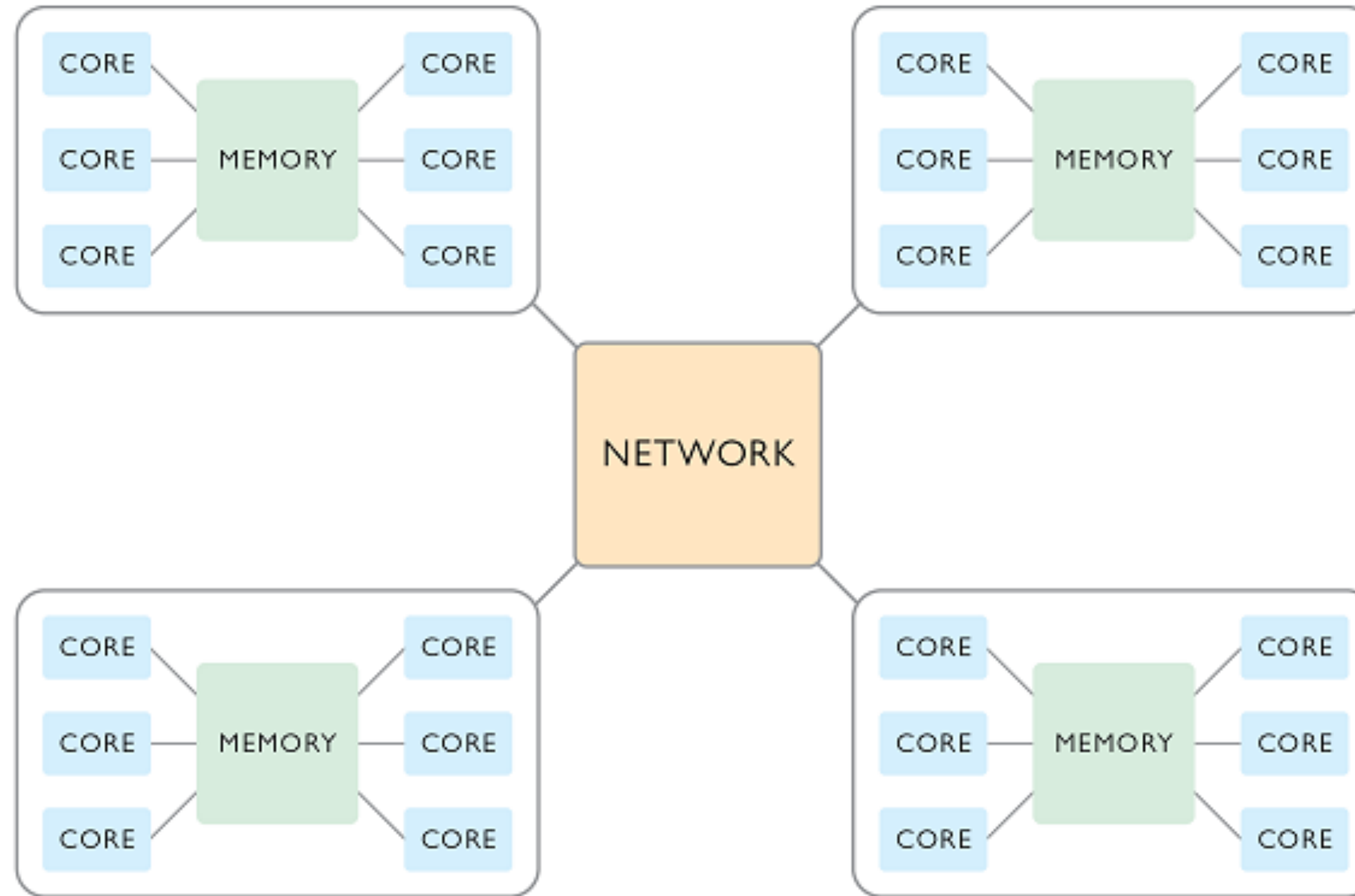
Parallel processing with distributed memory

Distributed memory computer



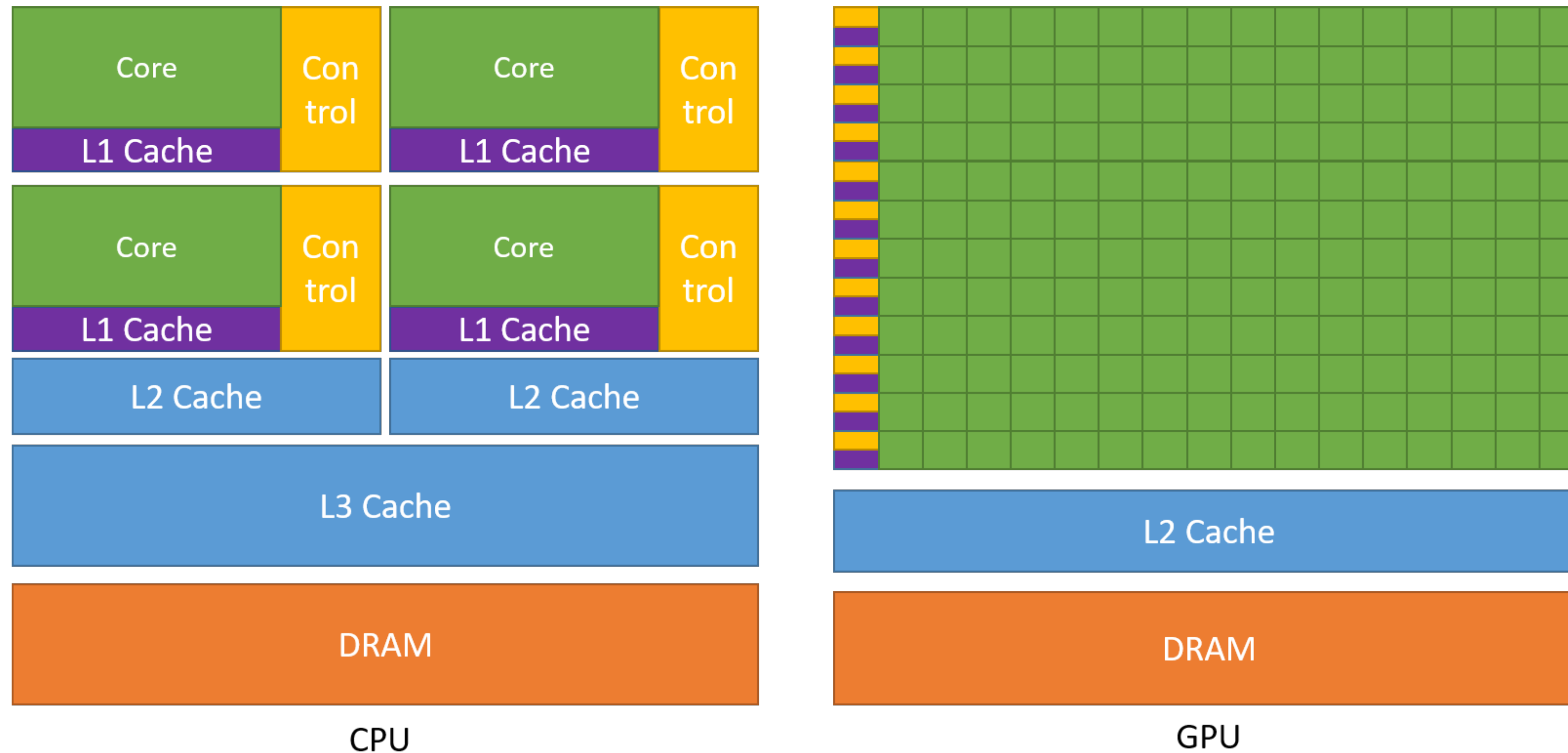
Hybrid parallel processing

Hybrid parallel computer



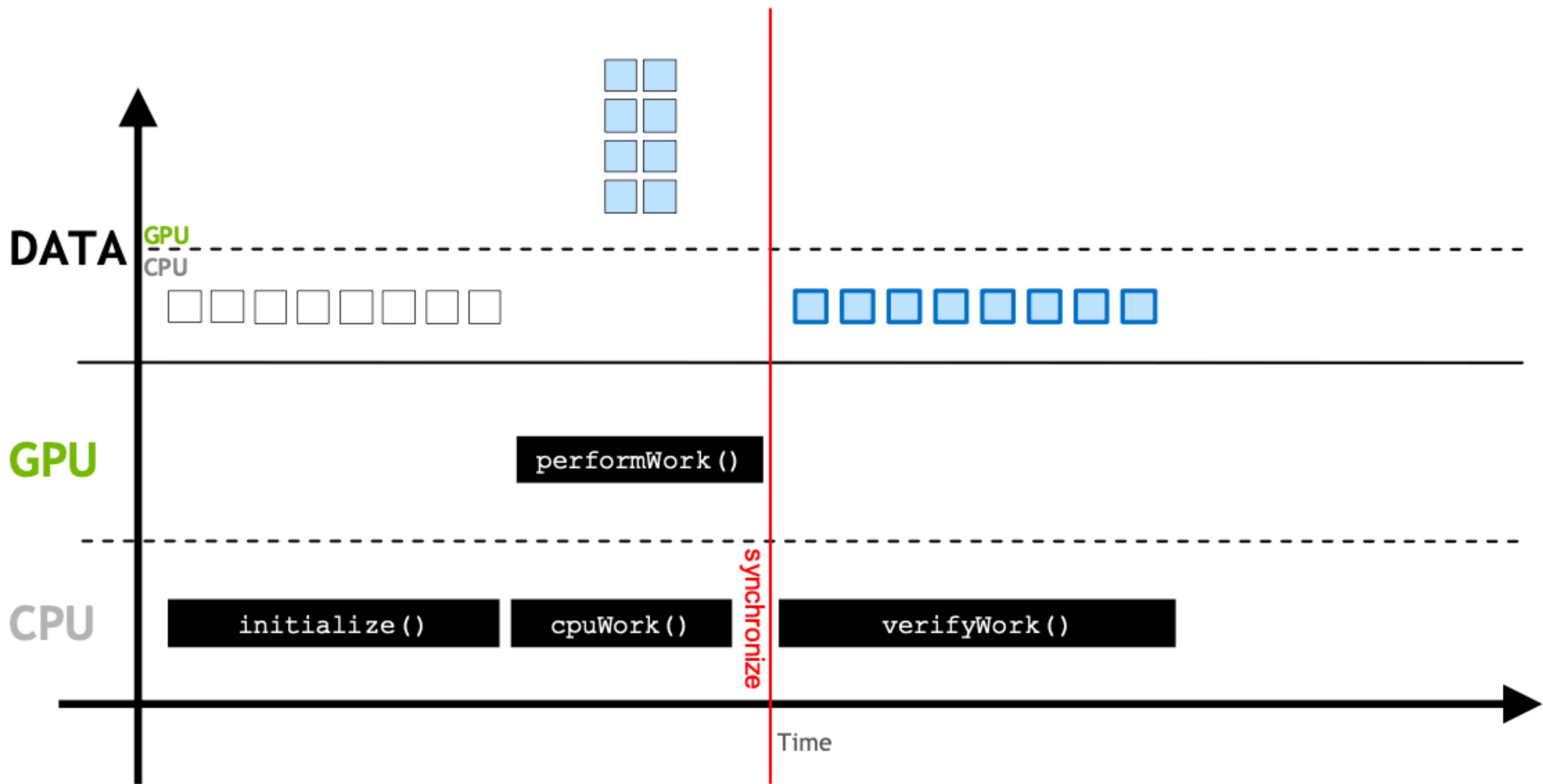
Accelerated programming on GPU

Difference between CPU and GPU



CPU — serial processing

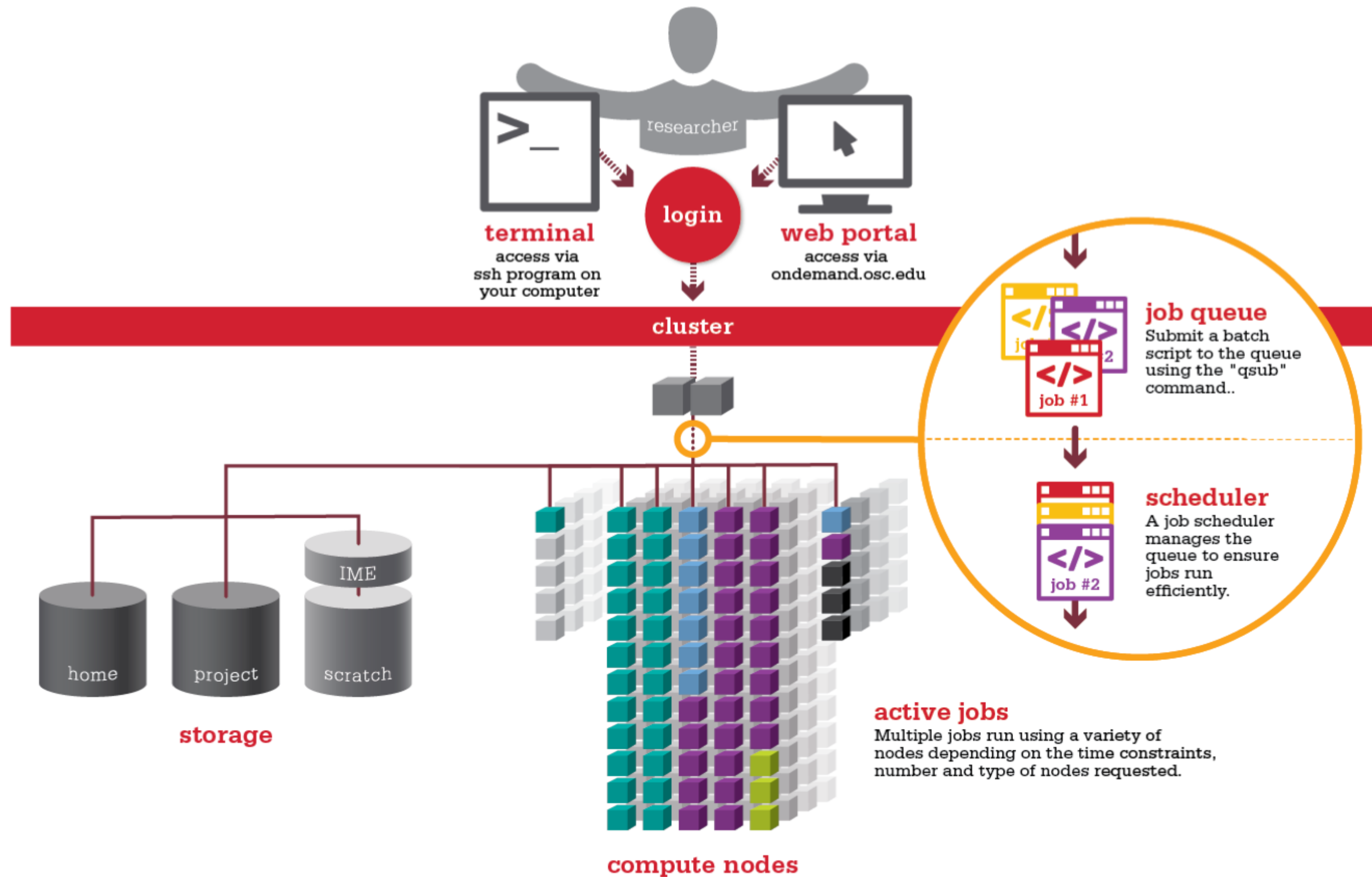
GPU — parallel processing



CUDA code

```
void CPUFunction()  
{  
    printf("This function is defined to run on the CPU.\n");  
}  
  
__global__ void GPUFunction()  
{  
    printf("This function is defined to run on the GPU.\n");  
}  
  
int main()  
{  
    CPUFunction();  
  
    GPUFunction<<<1, 1>>>();  
    cudaDeviceSynchronize();  
}
```

High Performance Computing



Batch job

Most common type of job on HPC

A script that describes all the processing required, plus estimate of resources

SLURM scheduler then assign resources and execute the work

Software on an HPC

The LMOD system is designed to allow users on HPCs to manage software they are using.

Before running any code, load the appropriate module(s)

Batch job

1. Create the code script that does the work
2. Create a bash script which tell SLURM about resources etc.
3. Submit the job
4. Inspect the output

Building a workflow

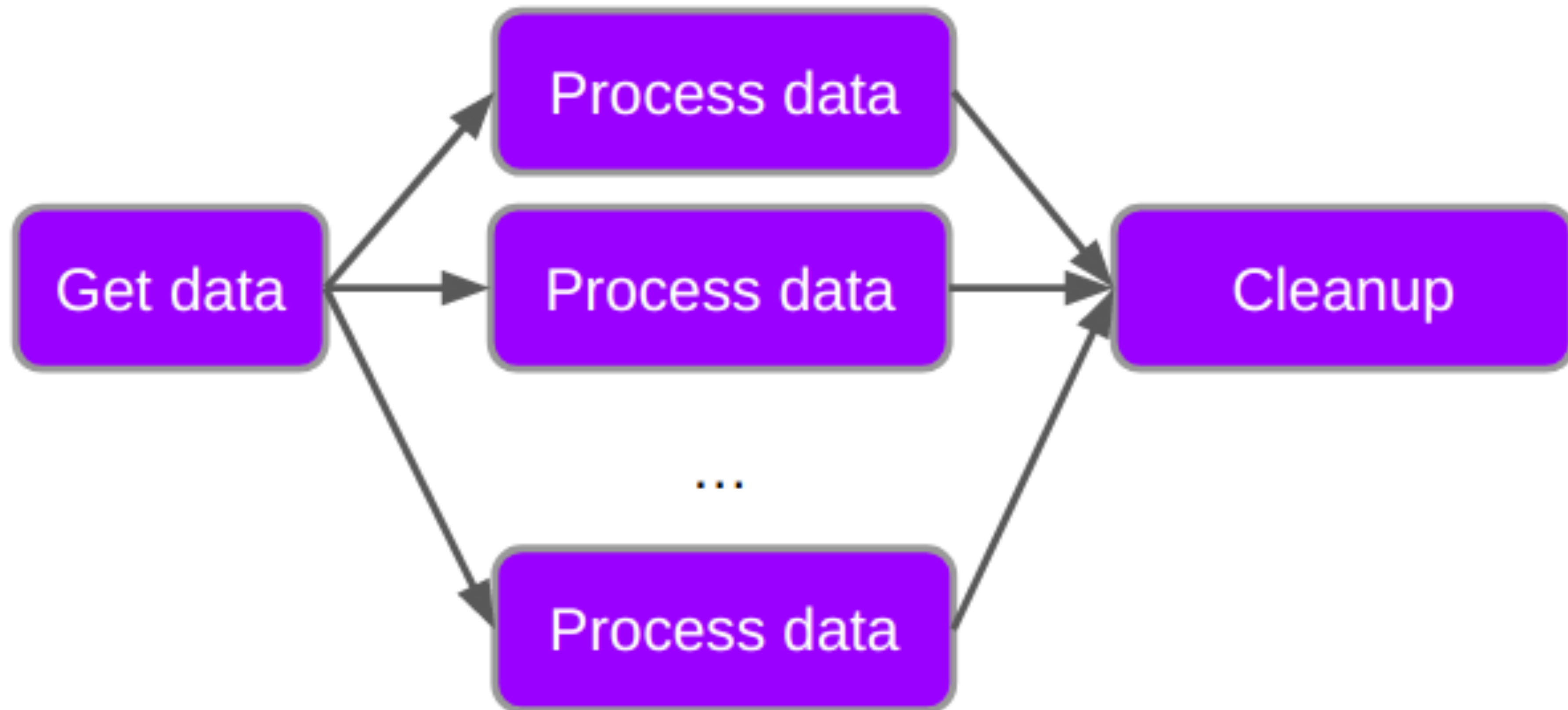


Each box represent work needs to be done

Arrows represent dependency of tasks

We can add ***dependency*** to job submission.

Parallel workflows



Can use *array jobs* to submit the multiple processing jobs.