

# Parallel computing for chemical simulations or Everything You Always Wanted to Know About how many CPUs do you need\* (\*But Were Afraid to Ask)

Benjamin Schweitzer

June 22, 2018

## Tasks

## Tasks

- $\sum_i^3 x_i^2$

## Tasks

- $\sum_i^3 x_i^2$
- Painting a wall

## Tasks

- $\sum_i^3 x_i^2$
- Painting a wall
- Harvesting crops

## Tasks

- $\sum_i^3 x_i^2$
- Painting a wall
- Harvesting crops

## Mode

- A few neurons

## Tasks

- $\sum_i^3 x_i^2$
- Painting a wall
- Harvesting crops

## Mode

- A few neurons
- Painters

## Tasks

- $\sum_i^3 x_i^2$
- Painting a wall
- Harvesting crops

## Mode

- A few neurons
- Painters
- Workers

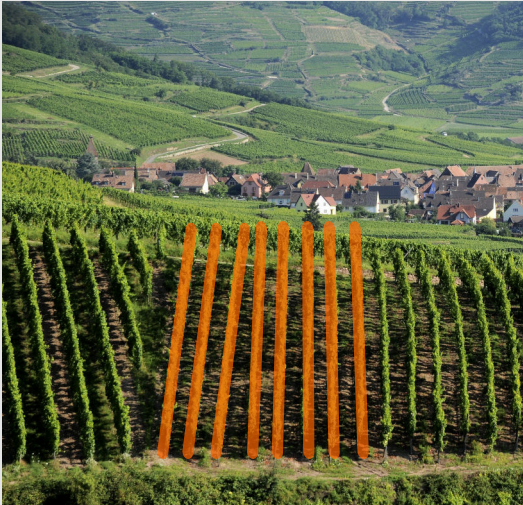




## Macroscopic example



## Macroscopic example



### Task & rules

- Harvest 8 vine lines
- Only one worker per line
- Only one task per worker

## Macroscopic example



## Results

- 1 workers
- 8 hours

## Macroscopic example



## Results

- 2 workers
- 4 hours

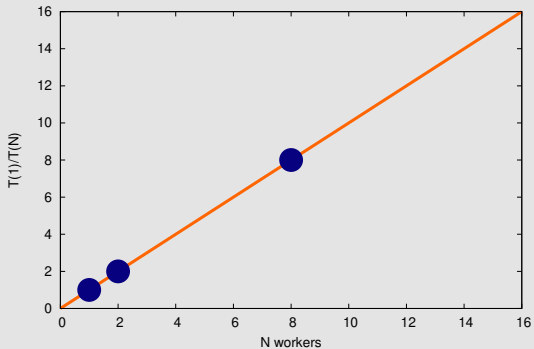
## Macroscopic example



## Results

- 8 workers
- 1 hour

Scaling

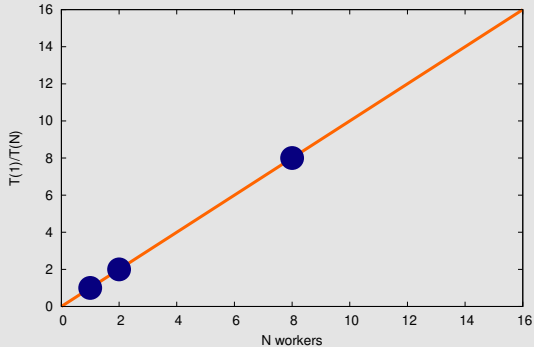


## Results

- 8 workers
- 1 hour

## Macroscopic example

Scaling



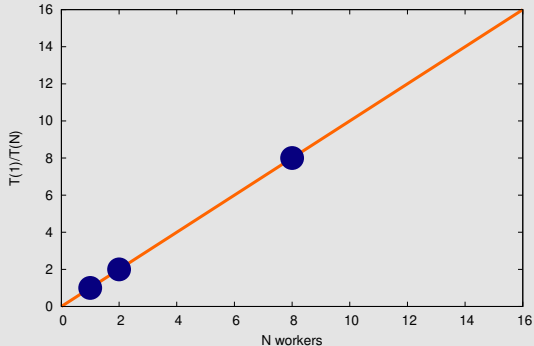
## Results

- 8 workers
  - 1 hour
- 16 workers ?



## Macroscopic example

Scaling



## Results

- 8 workers

- 1 hour

→ 16 workers ?

not with the one worker per line rule

## Macroscopic example

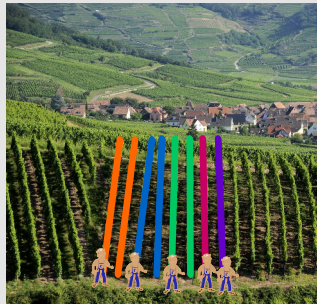
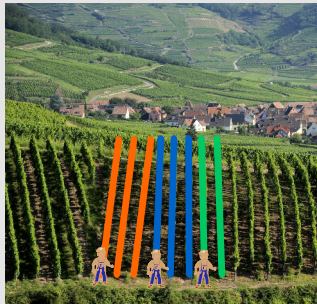


## Macroscopic example



Pay 16, get 8

# Macroscopic example

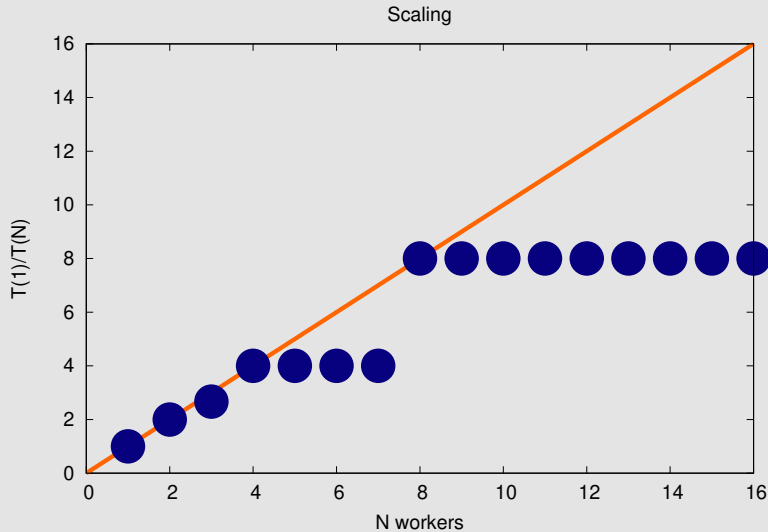


Odd numbers ?

3 workers :  $3+3+2$

5 workers :  $2+2+2+1+1$

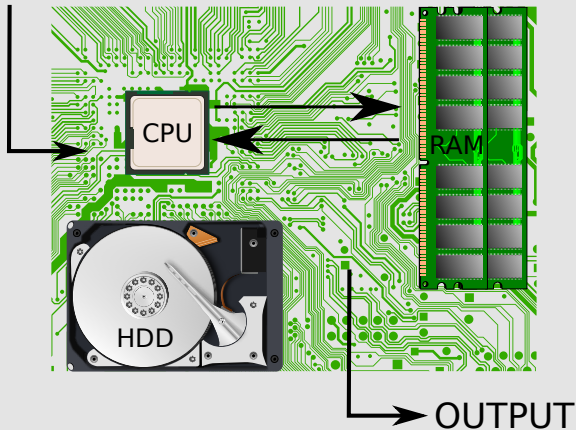
## Macroscopic example



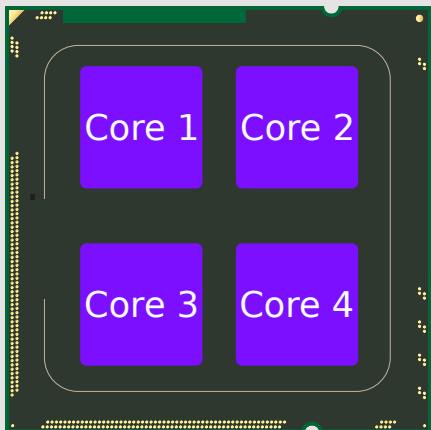
## Workers in computers

## Von Neumann architecture

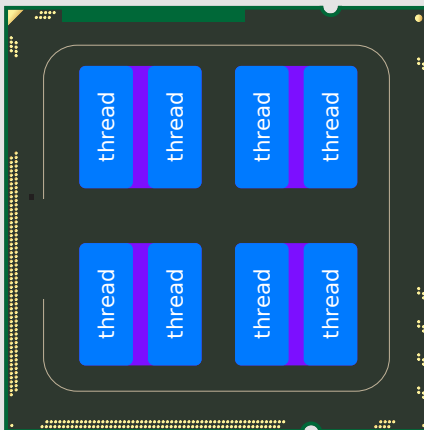
INPUT



## Cores

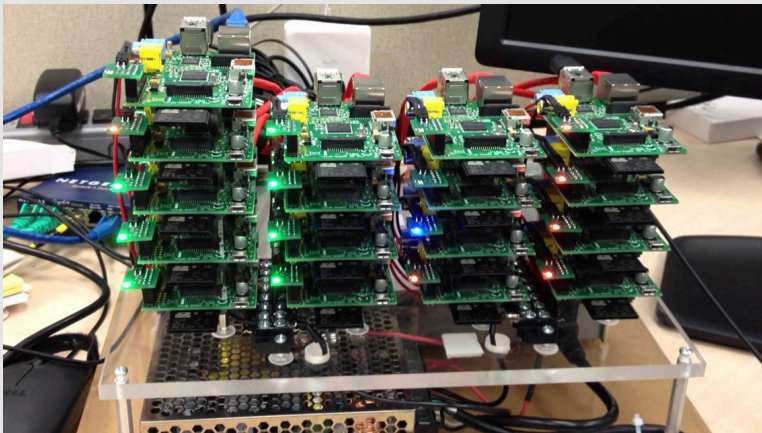


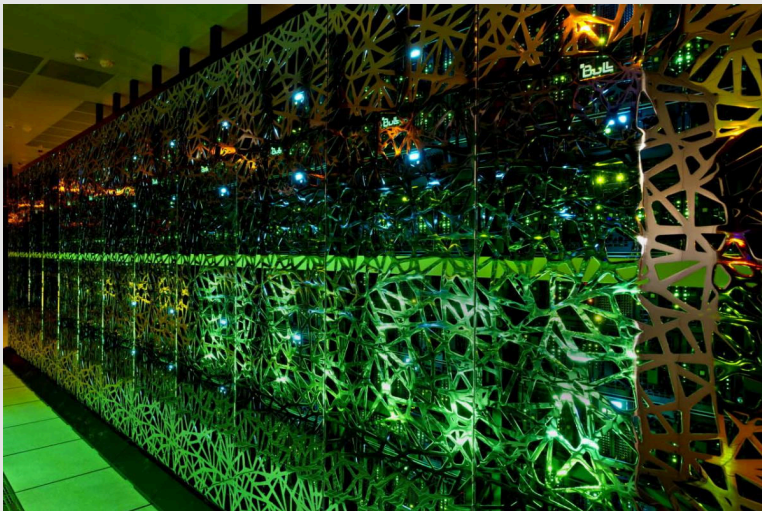
## Threads





## Supercomputers !!!





## “Nodes”

A node is just a computer

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
- Connect several nodes

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
  - Connect several nodes
- There is your cluster !

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
  - Connect several nodes
- There is your cluster !

## Know you gear



## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
  - Connect several nodes
- There is your cluster !

## Know you gear

- **READ THE DOCUMENTATION**

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
- Connect several nodes

→ There is your cluster !

## Know you gear

- **READ THE DOCUMENTATION**

- In doubt, double check (*i.e* CPU brand website)

## “Nodes”

A node is just a computer

- Usually 2 CPUs ( *i.e* 16 “cores”)
- Connect several nodes

→ There is your cluster !

## Know you gear

- **READ THE DOCUMENTATION**

- In doubt, double check (*i.e* CPU brand website)

- Be sure of what you ask for

## TGCC-Curie, thin nodes

## TGCC-Curie, thin nodes

- Intel Xeon E5-2680  
( FYI, > 1500 € at release  
date)  
8 cores 16 threads

## TGCC-Curie, thin nodes

- Intel Xeon E5-2680  
( FYI, > 1500 € at release date)  
8 cores 16 threads
- 2 E5-2680 per node → “16 workers”, 32 virtual workers

## TGCC-Curie, thin nodes

- Intel Xeon E5-2680  
( FYI, > 1500 € at release date)  
8 cores 16 threads
- 2 E5-2680 per node → “16 workers”, 32 virtual workers

## Submission

#MSUB -N 1 Number of nodes  
#MSUB -n 16 Number of tasks to run  
#MSUB -c 1 Number of cores per task

## TGCC-Curie, thin nodes

- Intel Xeon E5-2680  
( FYI, > 1500 € at release date)  
8 cores 16 threads
- 2 E5-2680 per node → “16 workers”, 32 virtual workers

## Submission

#MSUB -N 1 Number of nodes  
#MSUB -n 16 Number of tasks to run  
#MSUB -c 1 Number of cores per task

- What am I using ?



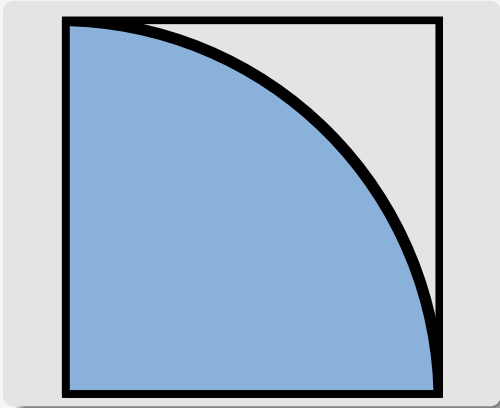
## TGCC-Curie, thin nodes

- Intel Xeon E5-2680  
( FYI, > 1500 € at release date)  
8 cores 16 threads
- 2 E5-2680 per node → “16 workers”, 32 virtual workers

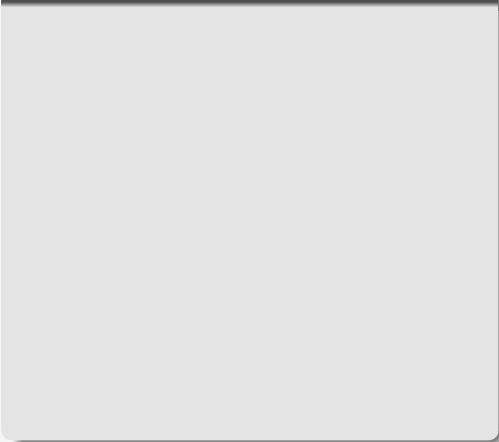
## Submission

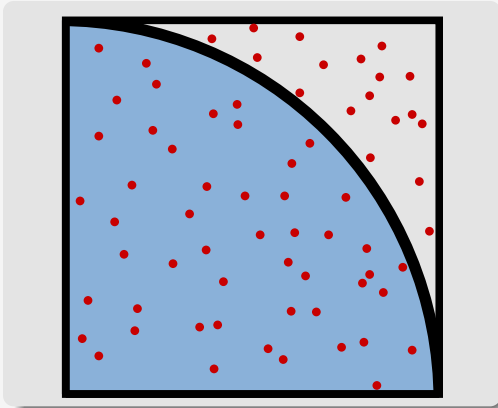
- #MSUB -N 1 Number of nodes
- #MSUB -n 16 Number of tasks to run
- #MSUB -c 1 Number of cores per task
- What am I using ?
- 16 cores from 2 CPUs in 1 node

Vine harvesting is obvious, what about programs ?



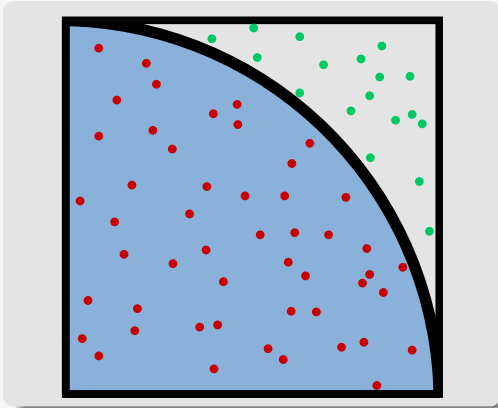
Estimate  $\pi$  Monte Carlo style





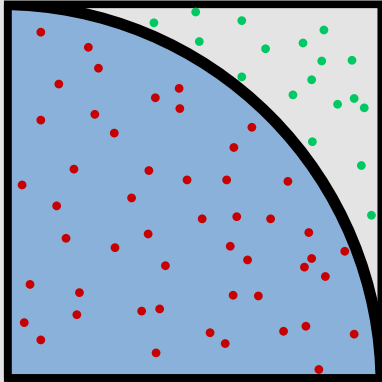
## Estimate $\pi$ Monte Carlo style

- Random “shots” (with uniform distribution)



## Estimate $\pi$ Monte Carlo style

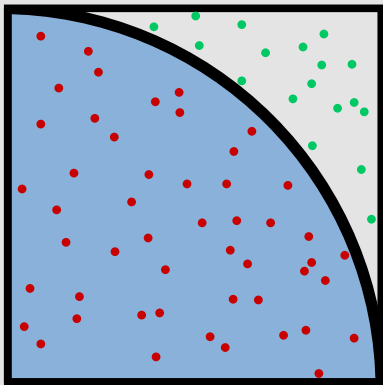
- Random “shots” (with uniform distribution)
- $a$  red points  $b$  green points



## Estimate $\pi$ Monte Carlo style

- Random “shots” (with uniform distribution)
- $a$  red points  $b$  green points

Quadrant & square surface ratio is  $\frac{\pi}{4}$

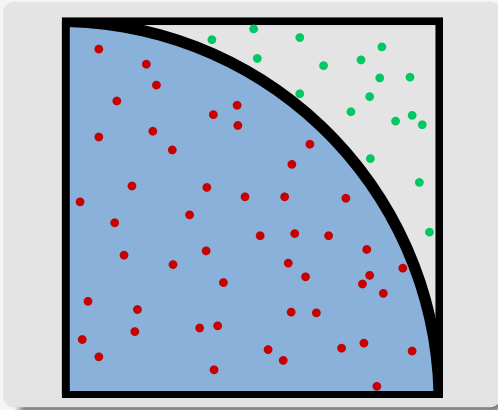


## Estimate $\pi$ Monte Carlo style

- Random “shots” (with uniform distribution)
- $a$  red points  $b$  green points

Quadrant & square surface  
ratio is  $\frac{\pi}{4}$

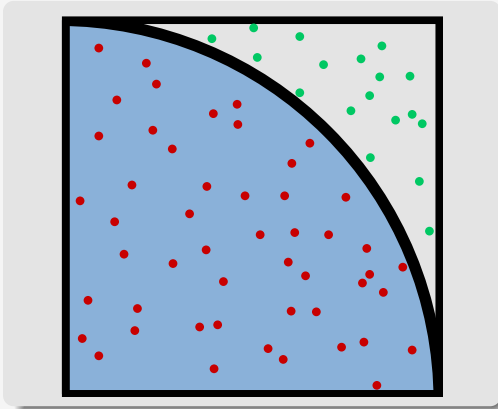
$$\rightarrow \frac{\pi}{4} \approx \frac{a}{a+b}$$



## Tasks

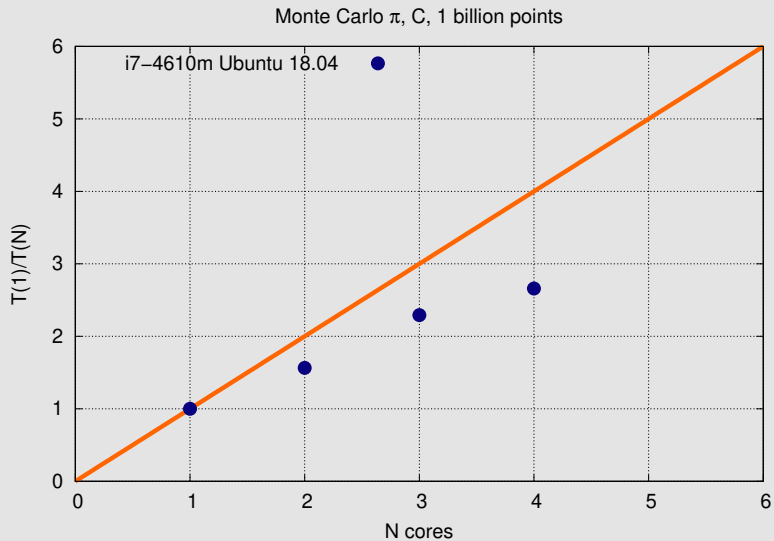
- Generate points coordinates
- Determine if in or out

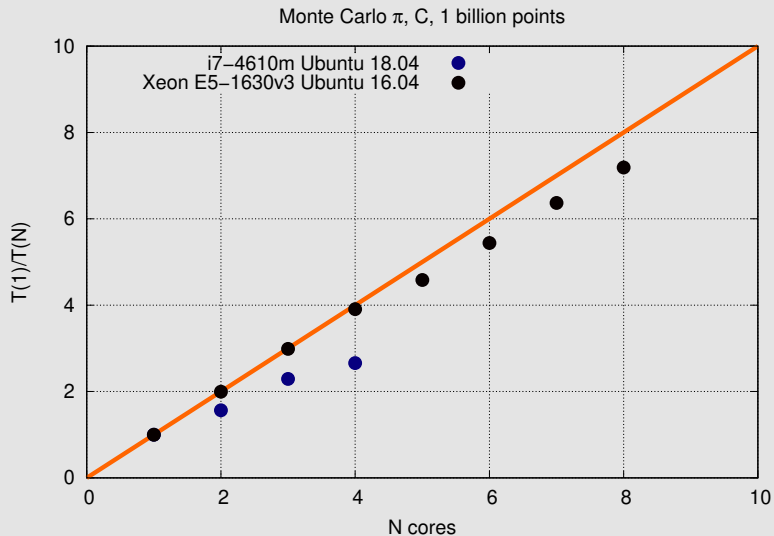


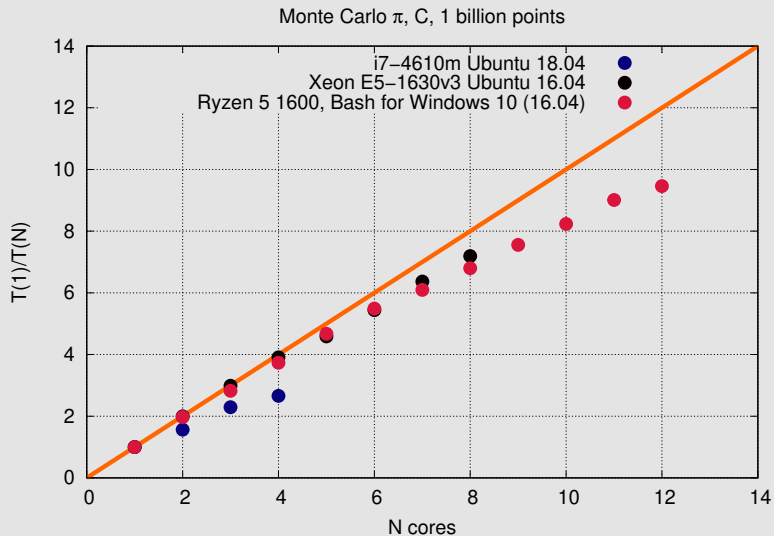


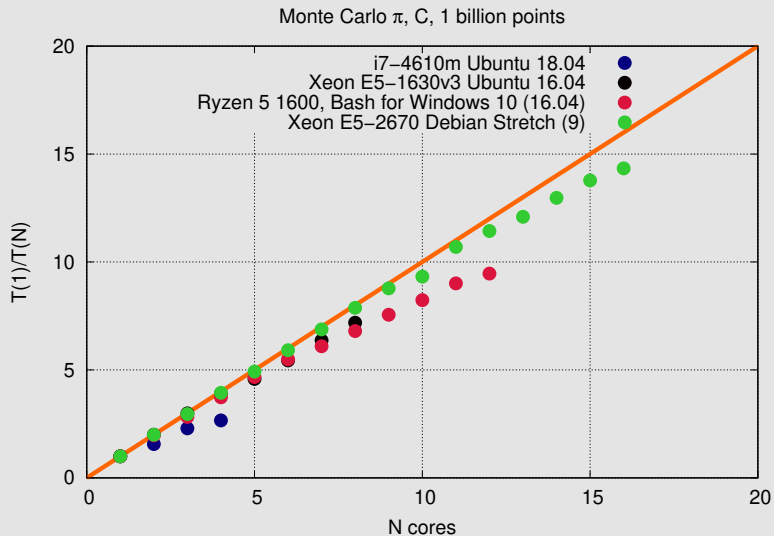
## Tasks

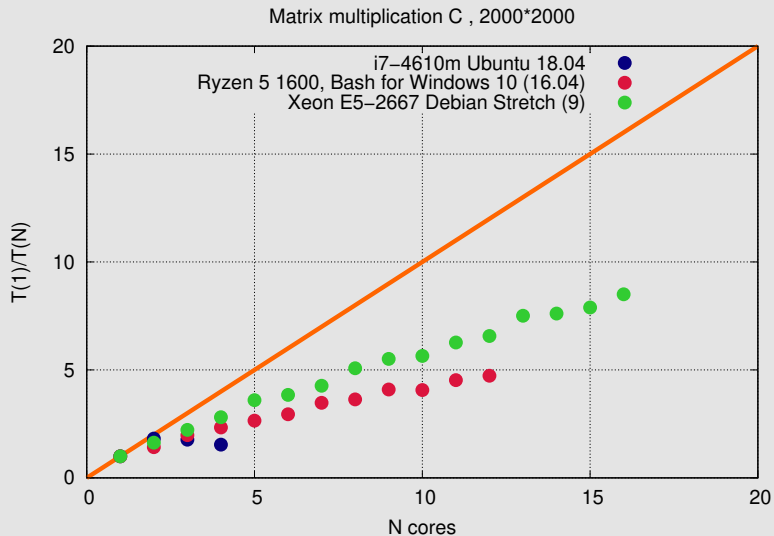
- Generate points coordinates
- Determine if in or out
- **Gather the results and compute  $\pi$**

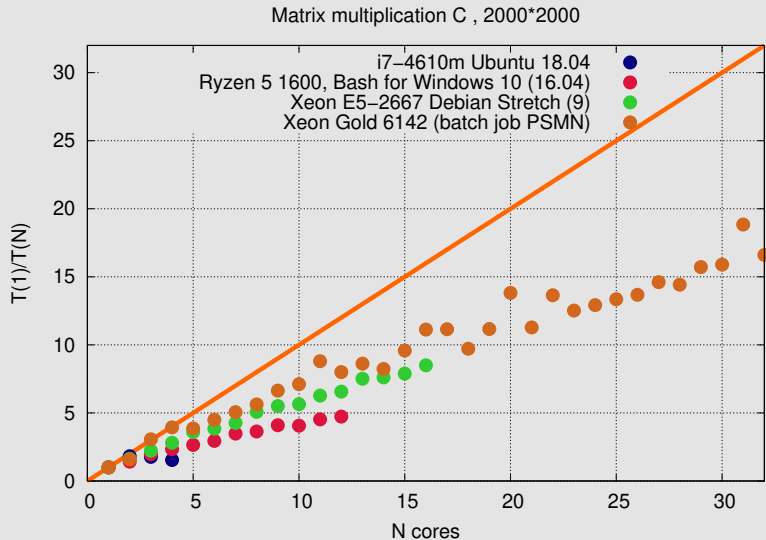




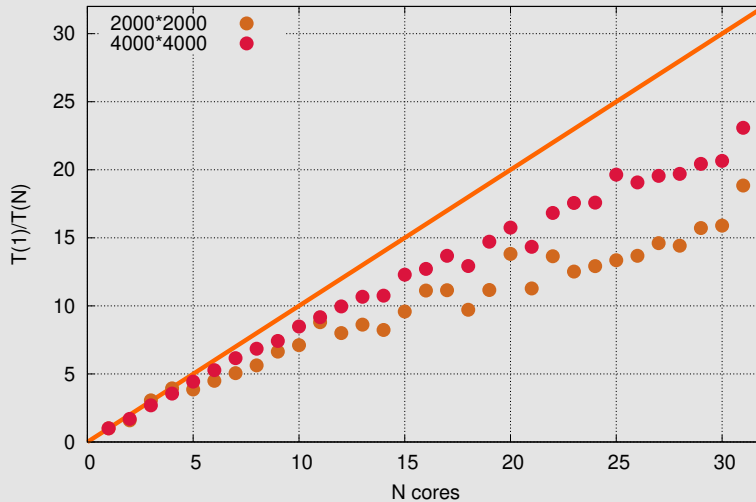




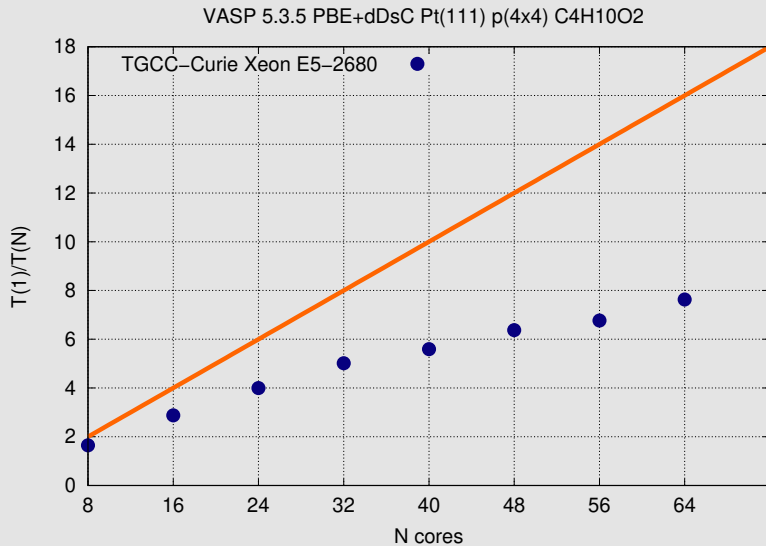


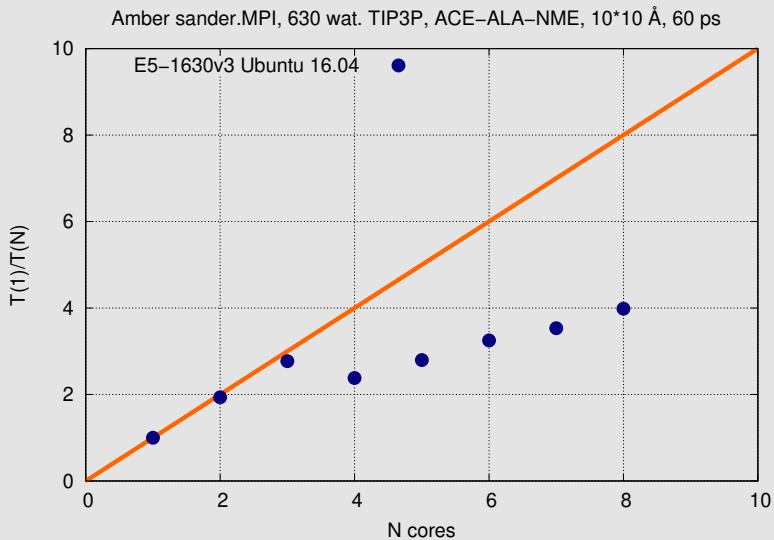


Matrix multiplication C, Xeon Gold 6142

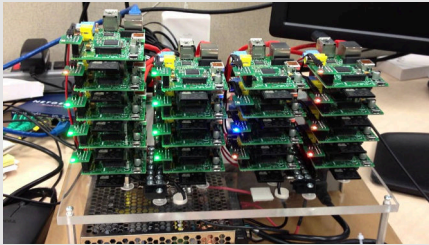








## Cores



## Threads

- Transferring informations
- MPI, openMP

- Cost
- Impact

- Cost
  - Impact
- Optimize