Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

intel

# AI and Predictive Analytics in Data-Center Environments

## Performance & Executing Experiments

Josep Ll. Berral @BSC

Intel Academic Education Mindshare Initiative for AI

# Introduction

“We have to choose where/how to run AI algorithms & experiments”

# Introduction

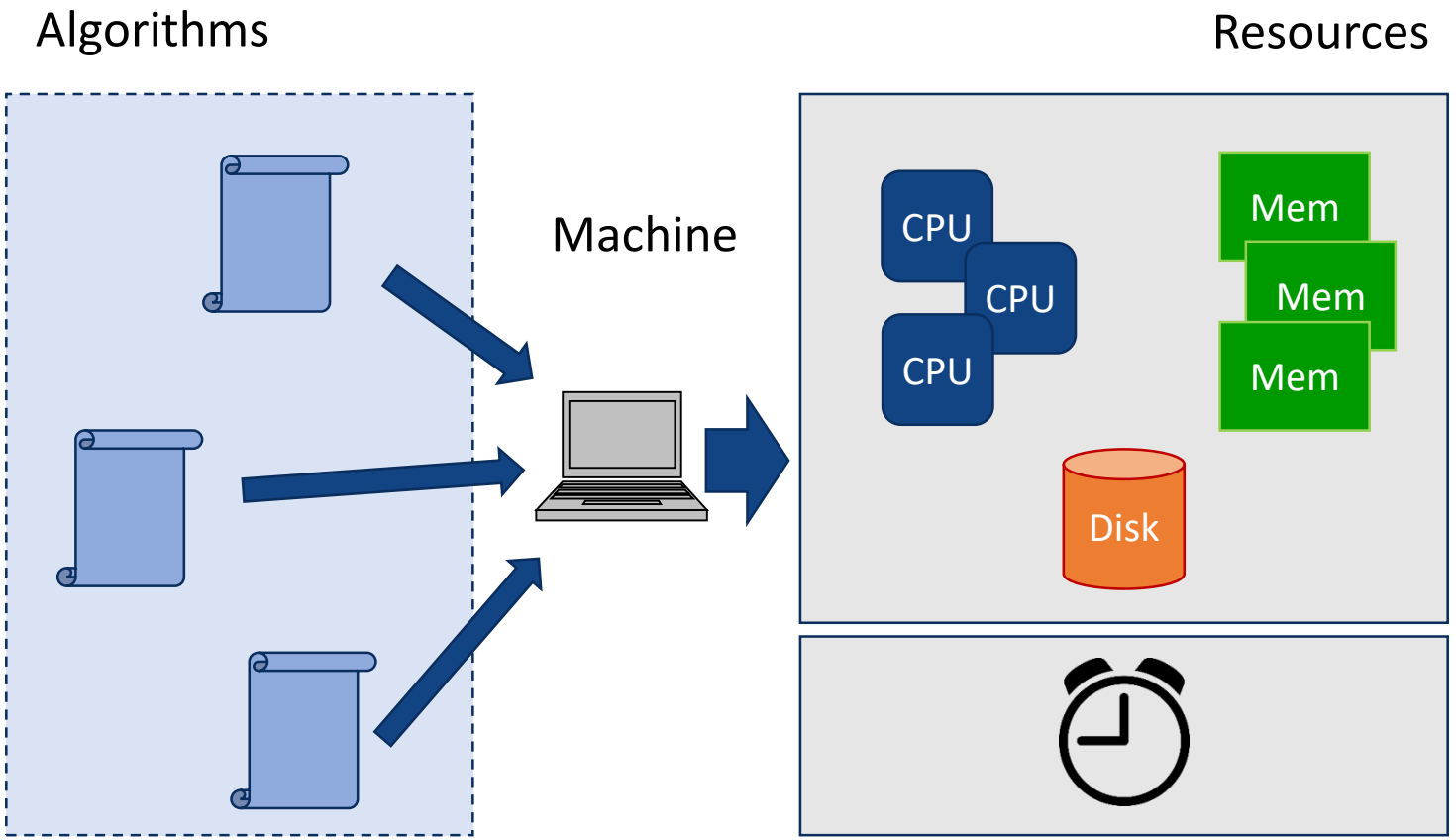
- Algorithms have computing/data requirements
  - Computation Resources
    - CPU, Memory, GPUs, accelerators, storage, ...
  - Time to run
    - Train models, infer new data, …
  - Data
    - What we are modeling and imitating

Machines to run our algorithms

Data to feed our algorithms

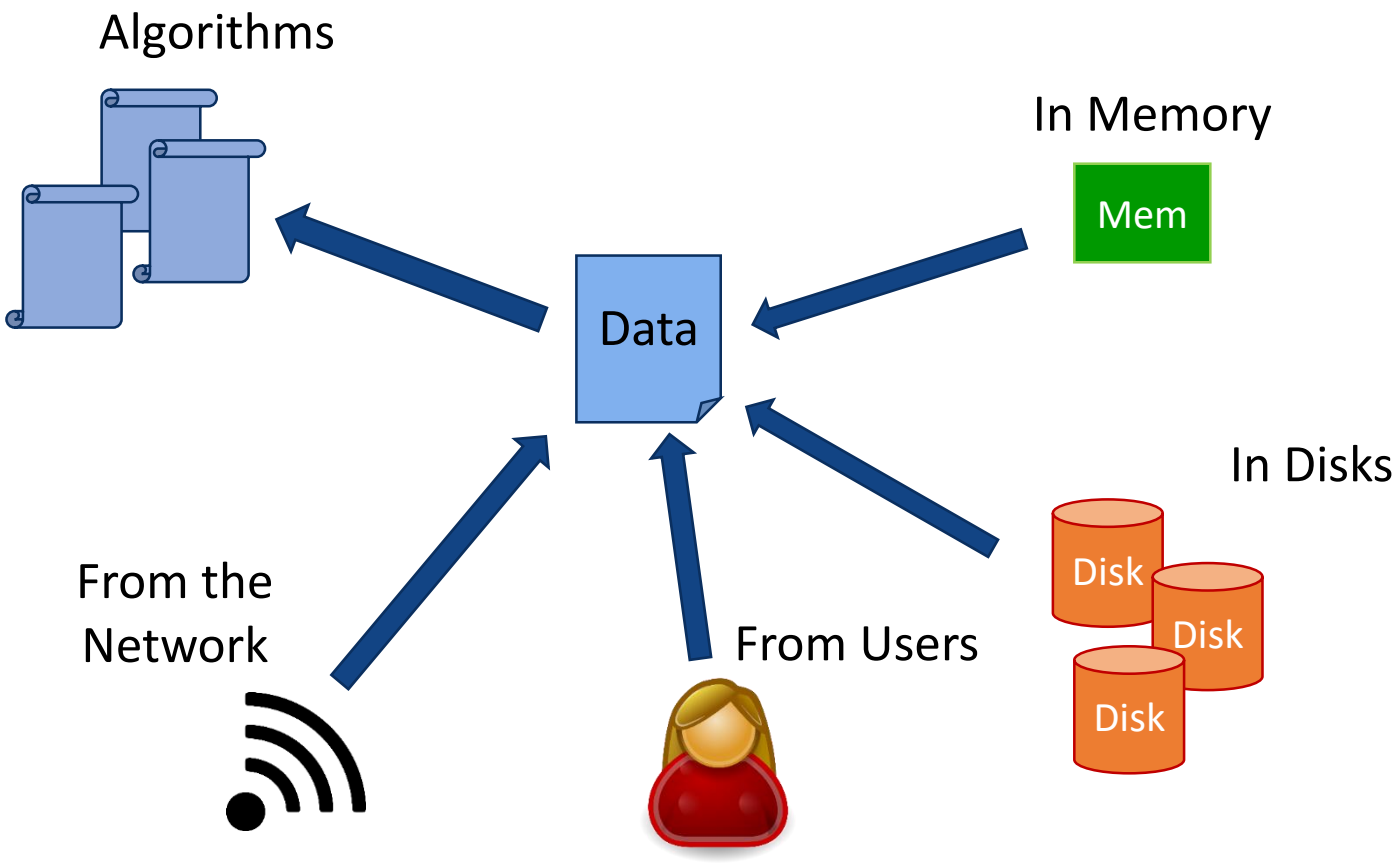
# Resources

- Algorithms have computing requirements



# Resources

- Algorithms also have data requirements

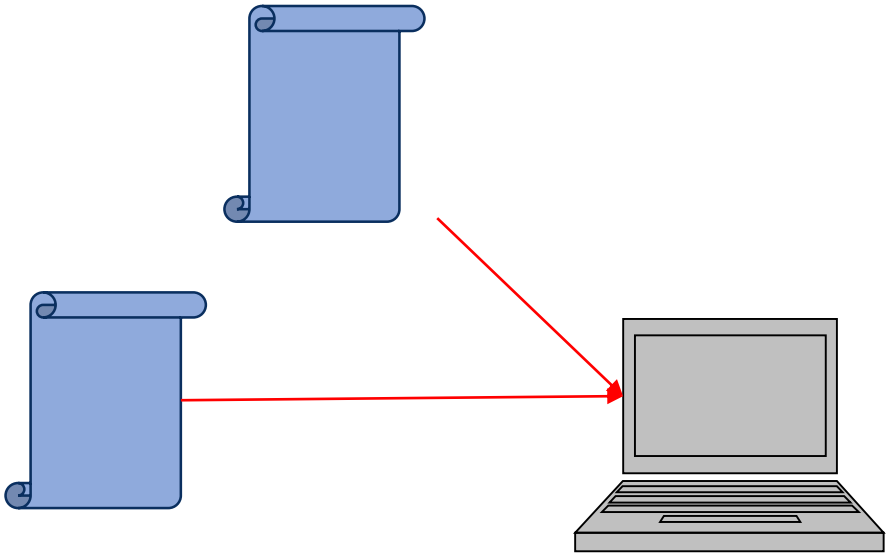


# Environment

We need a COMPUTING ENVIRONMENT!

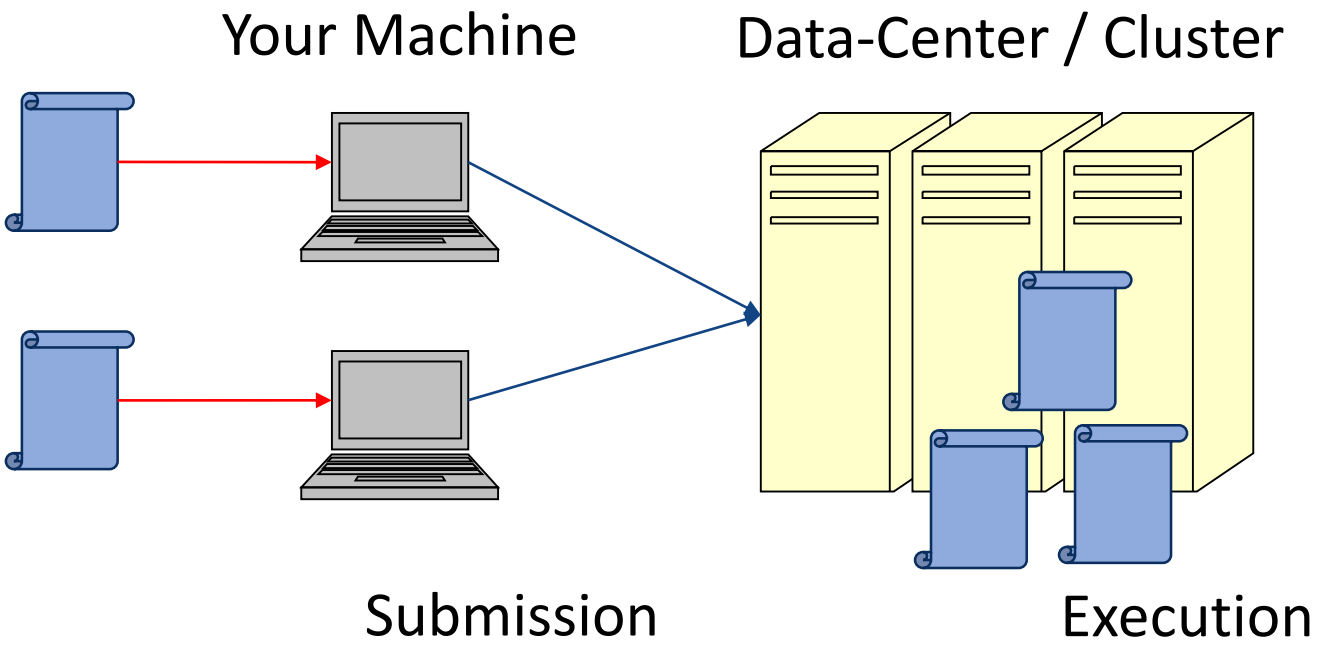
# Environment

- Local Machines
  - Own computer
  - Workstations at work
  - ...



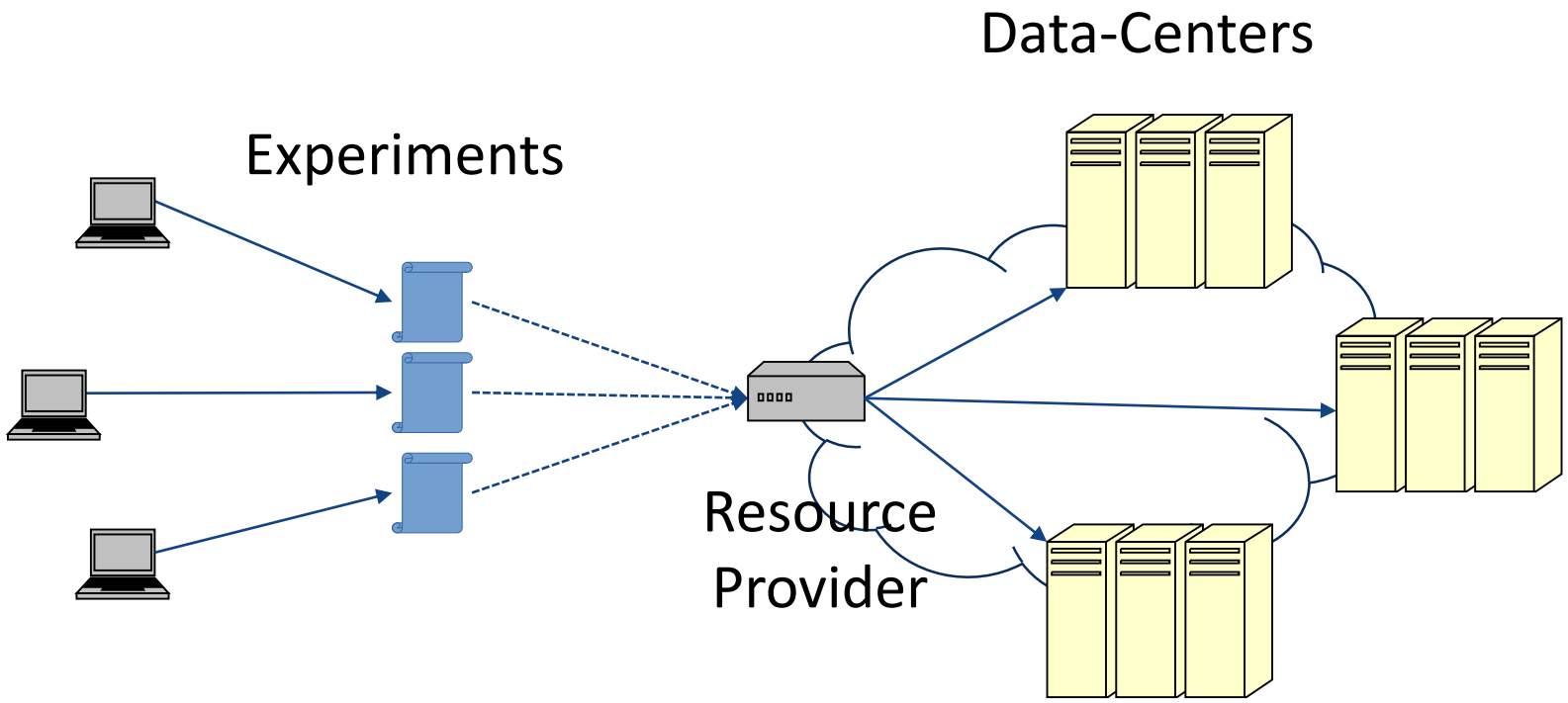
# Environment

- Cluster Machines
  - DataCenter at work
  - DataCener at labs
  - Scientific grids
  - ...



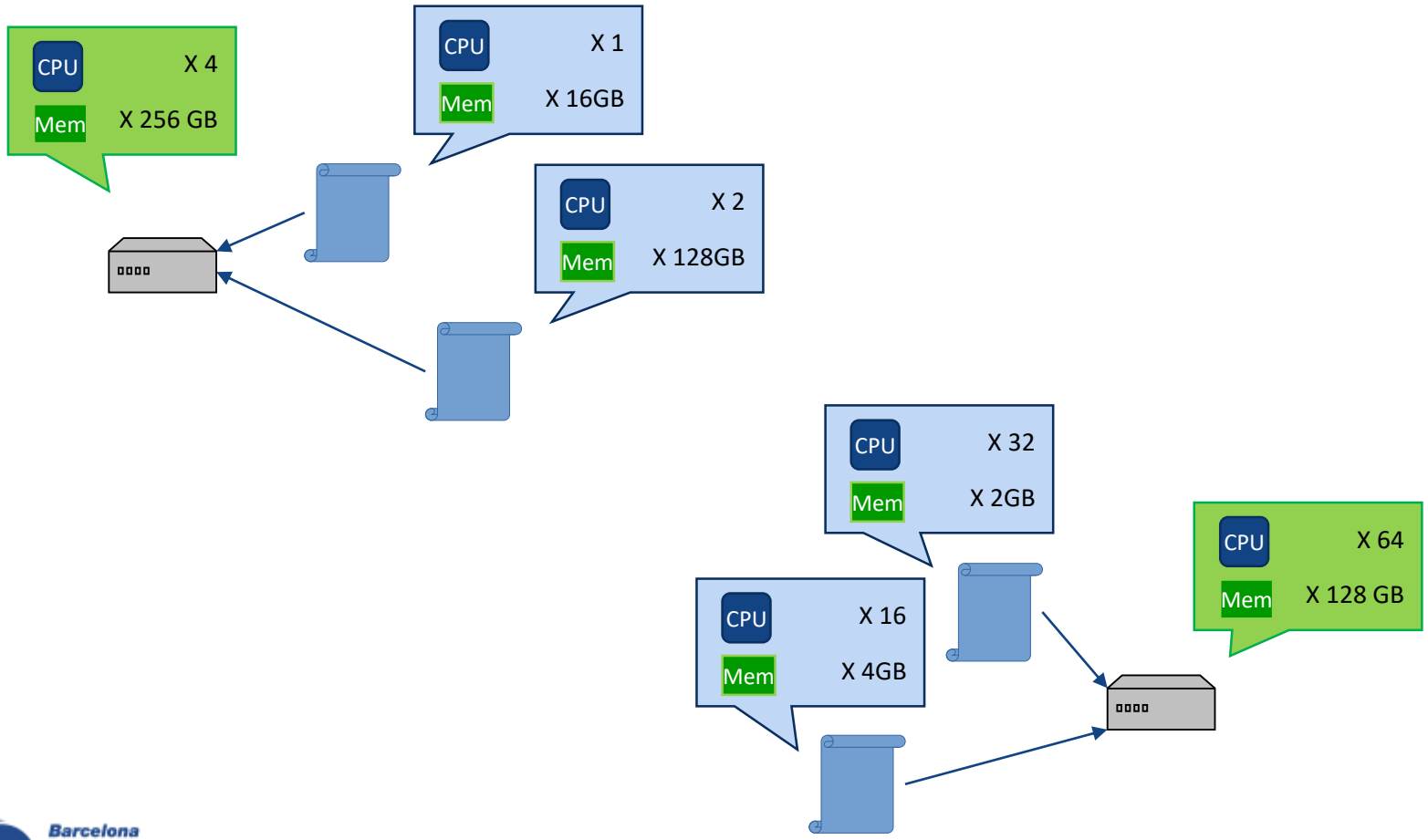
# Environment

- The Cloud
  - Data-Centers from Resource Providers



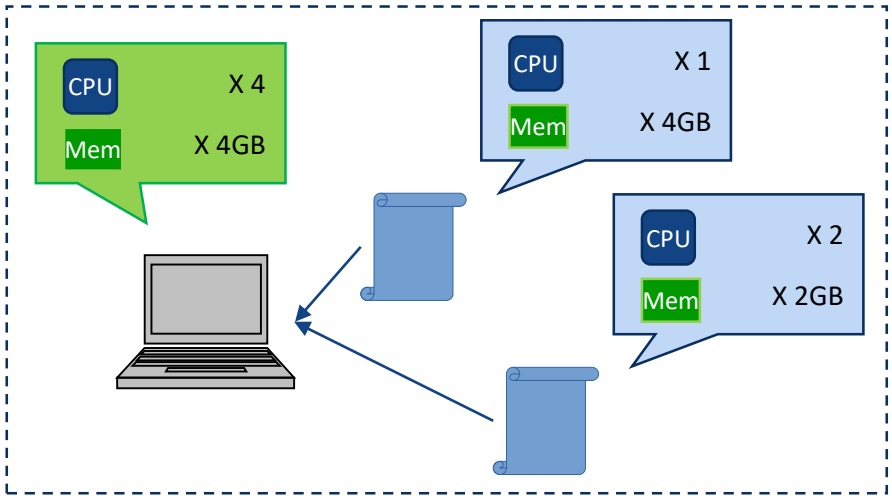
# Environment

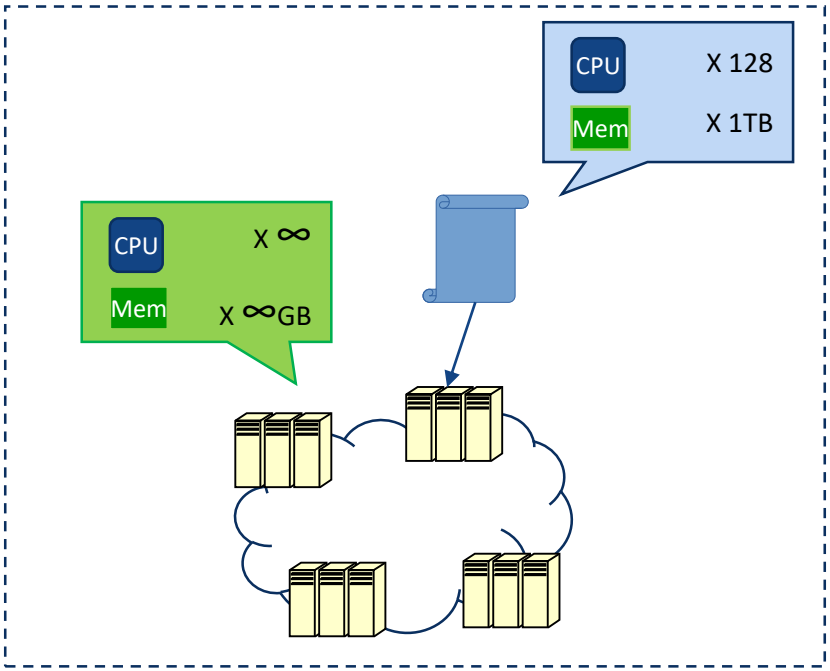
- Choosing the environment

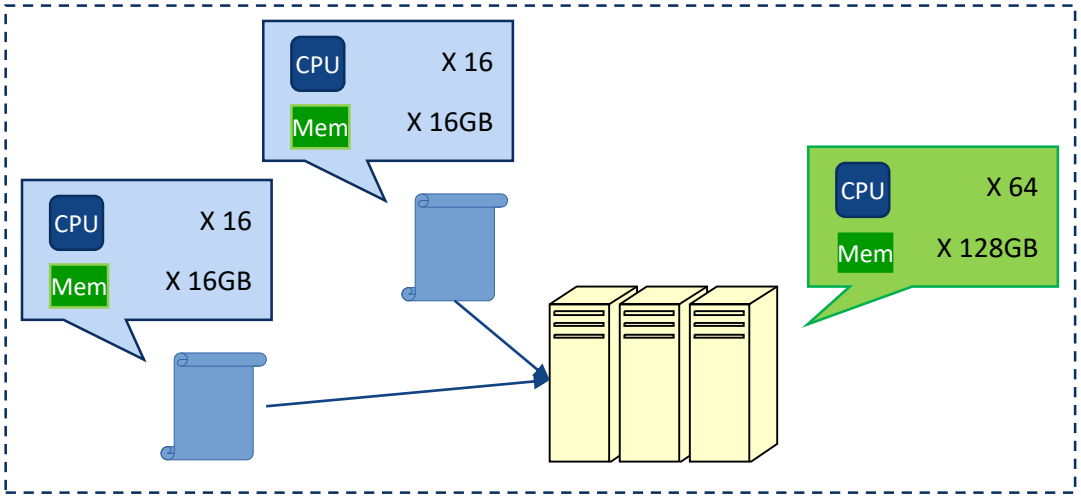


# Environment

- Choosing the environment



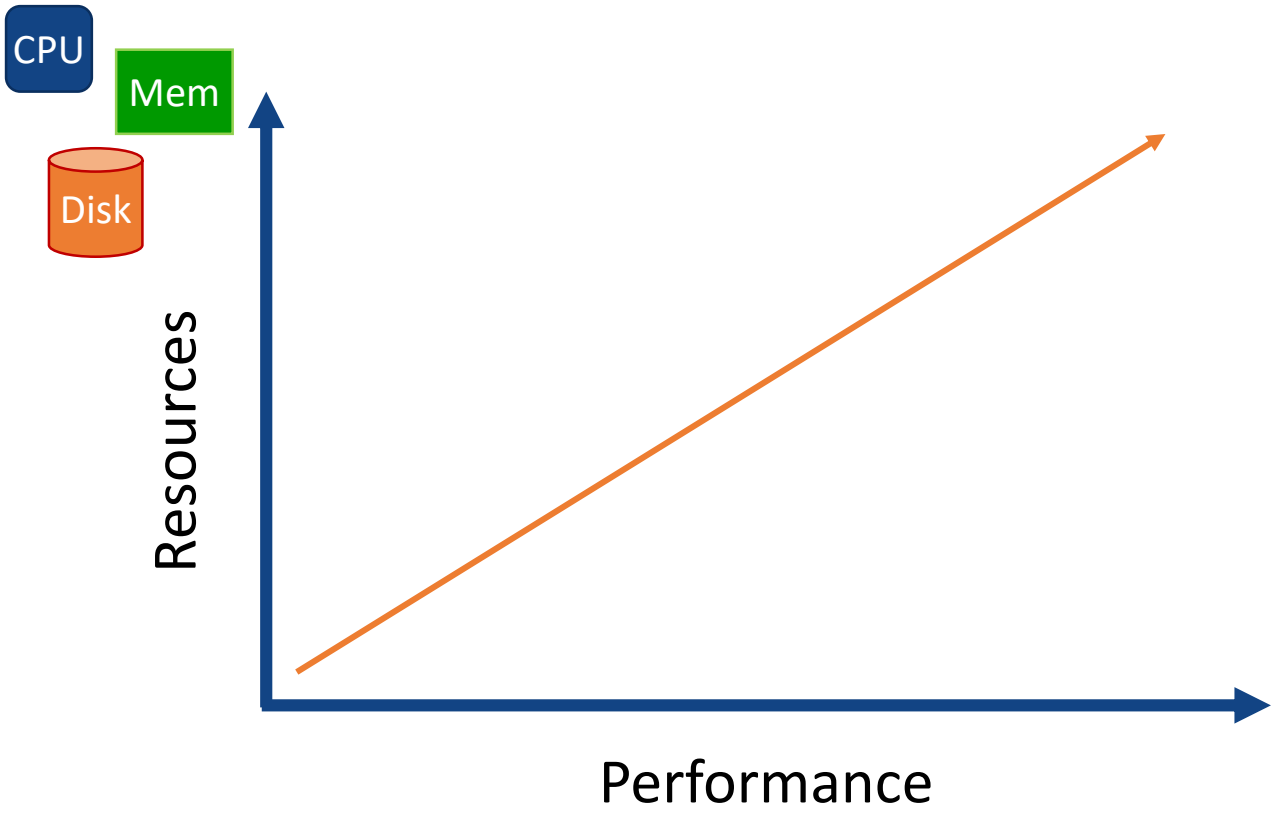
# Performance

"Work x Time"

"Capacity to Progress"

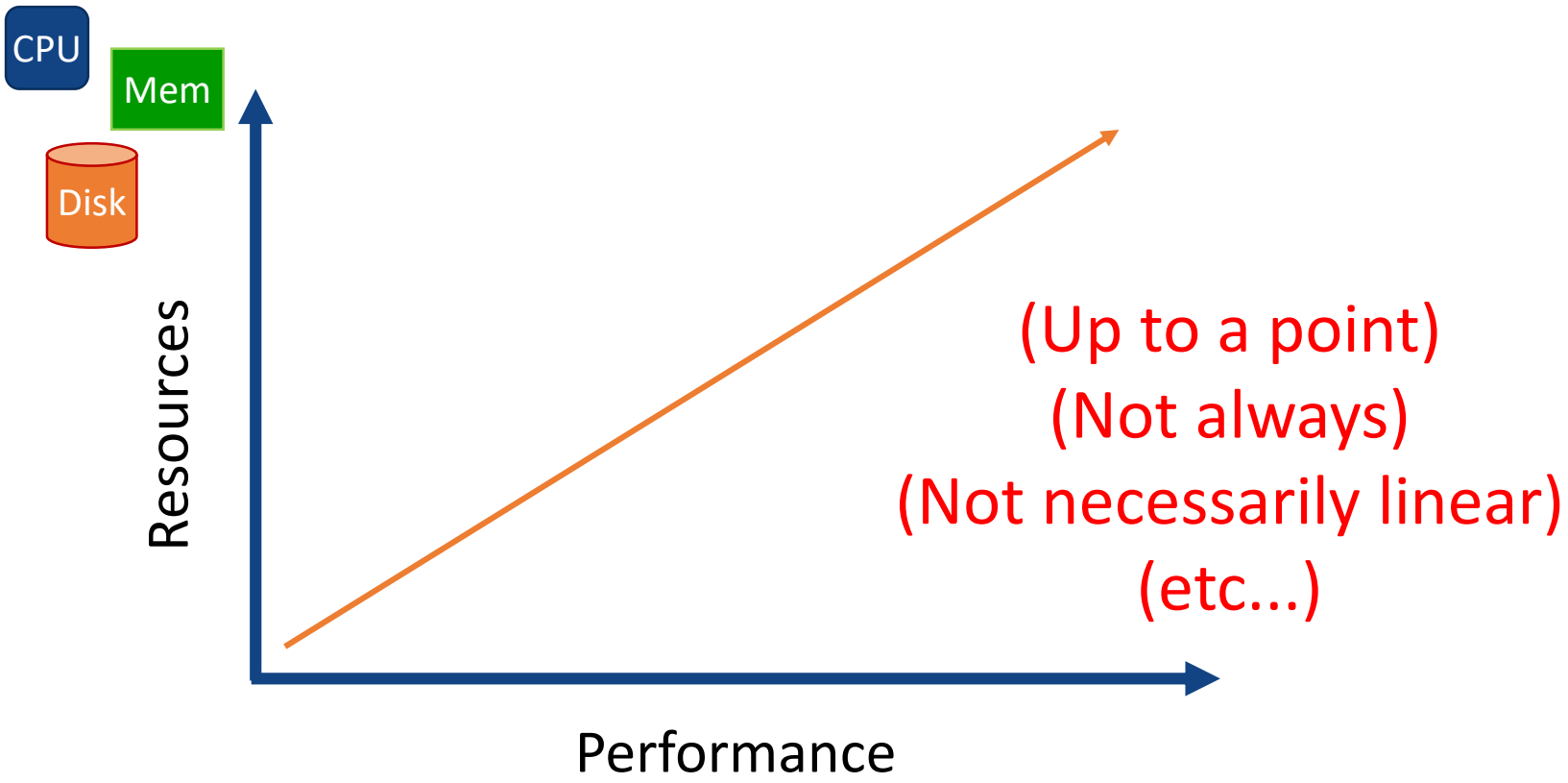
# Performance

- Performance is (usually) linked to Resources



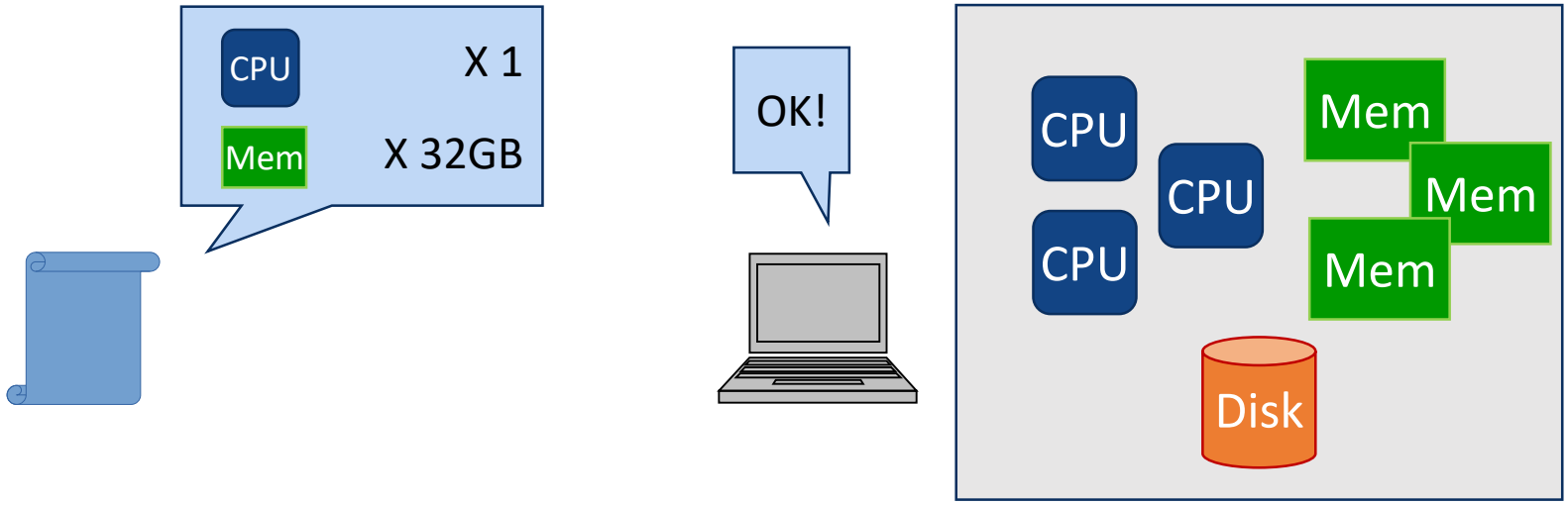
# Performance

- Performance is (usually) linked to Resources



# Performance

- Computing Environment
  - Pool of Resources
- Algorithms/Apps/Experiments
  - Require resources



# Some Theory

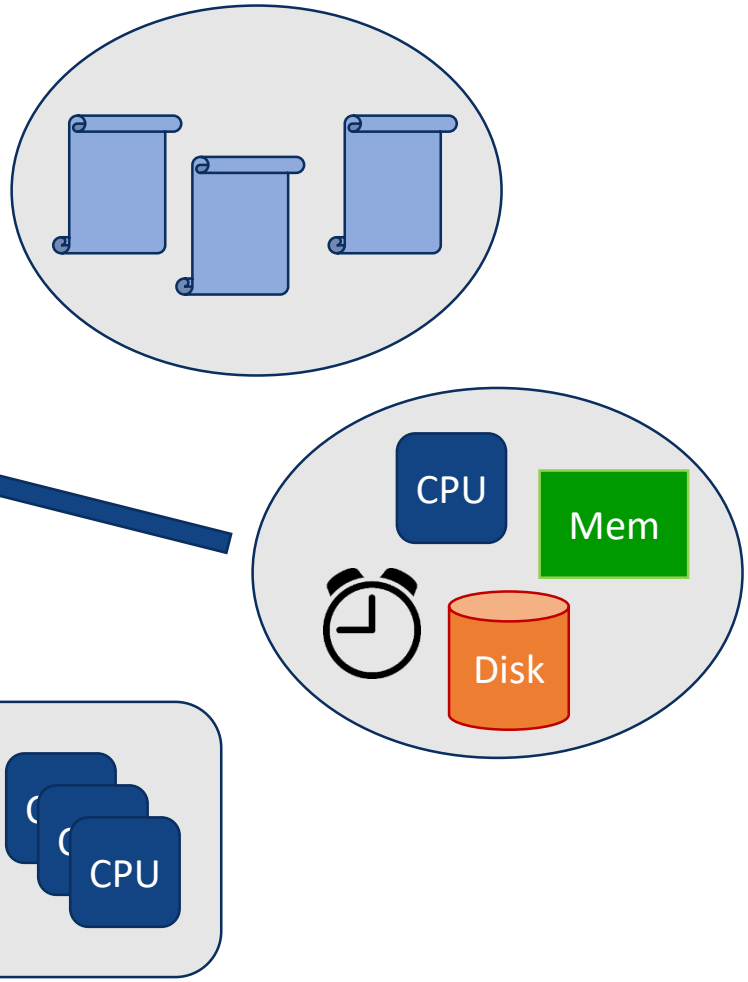
## Little's Law

- Little's Law: $L = \lambda W$
  - "Arrival rate x Dedicated time per input = Average load"

# Some Theory

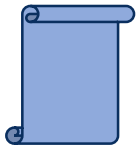
## Little's Law

- Relation between
  - Received load
  - Resources/Time required
  - Average load // Required resources



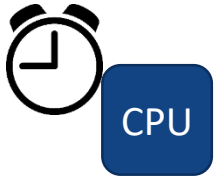
# Little's Law: L = λW

- Example
  - we submit λ = 100 experiments / hour



100 Exp / hour

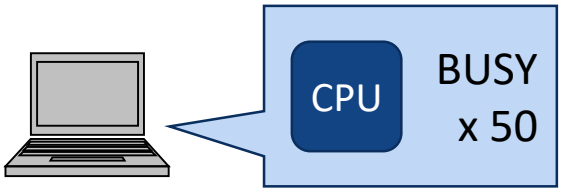
  - Exps. take an average of W = 0.5 hours
    - [with 1 CPU per exp.]




½ hour
1 CPU

  - Average number of exps. on our system: L = 50 exps
    - [avg. 50 CPUs in use]

What we expect
(what we need)



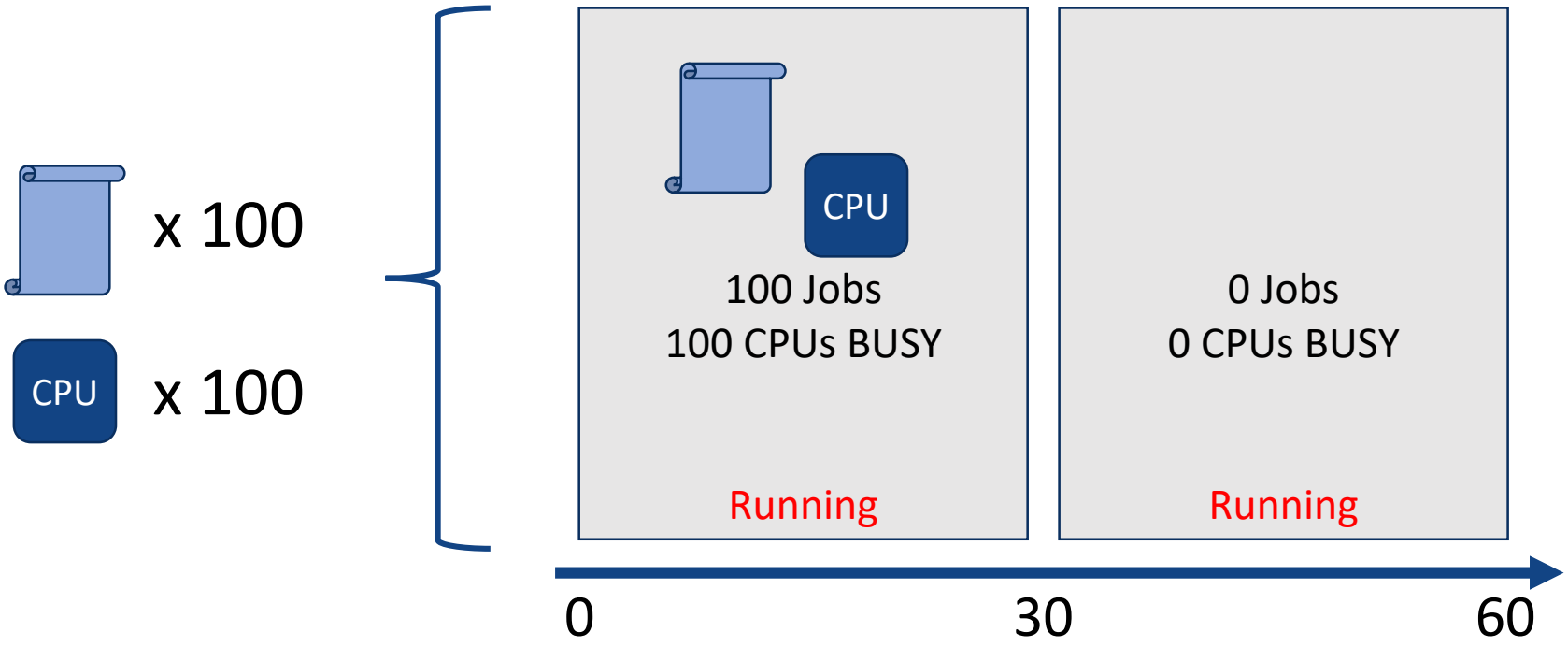
# Little's Law

Demo!

# Resource Limits

- 100 CPUs

| | |
|---|---|
| Arrival rate: | λ = 100 experiments / hour |
| Exps. take: | W = 0.5 hours [1 CPU per exp.] |
| Average exps. in: | L = 50 exps |

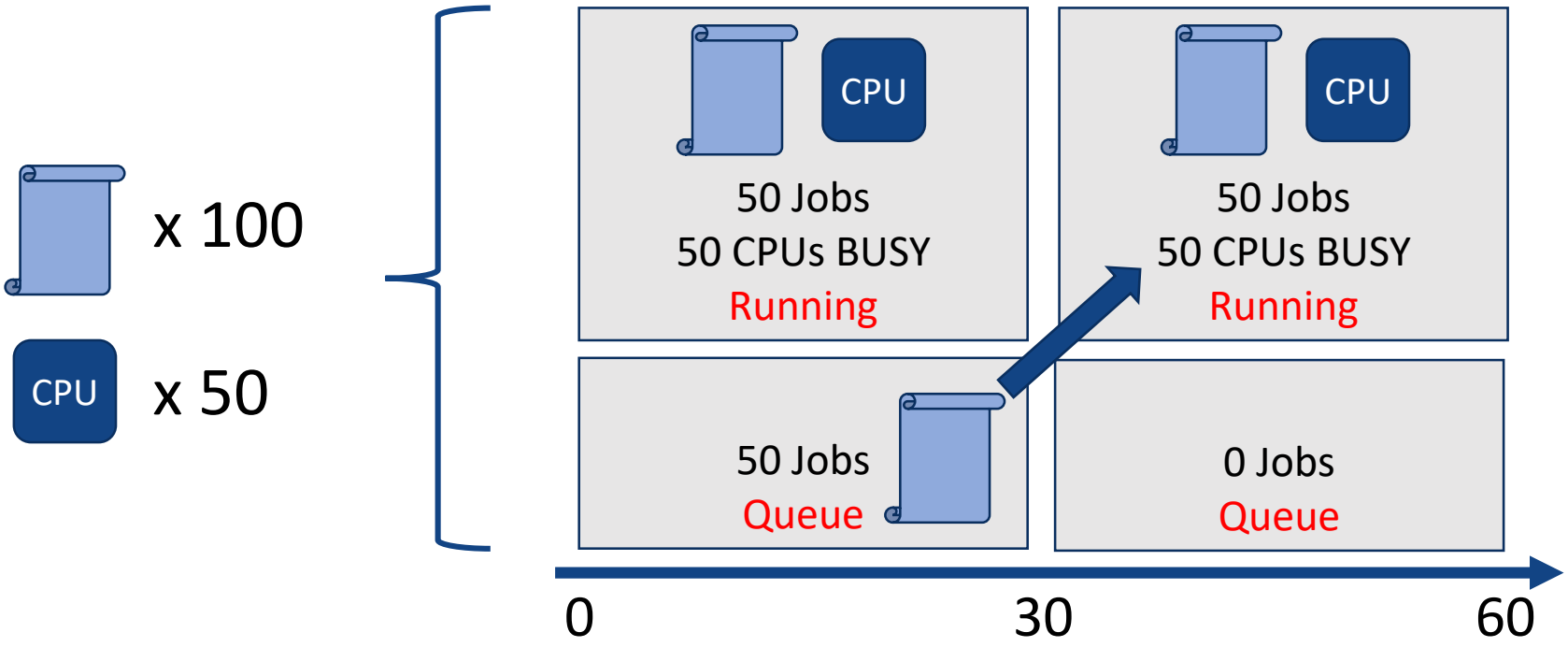


# Resource Limits

- 50 CPUs

  (What Little's Law indicated)

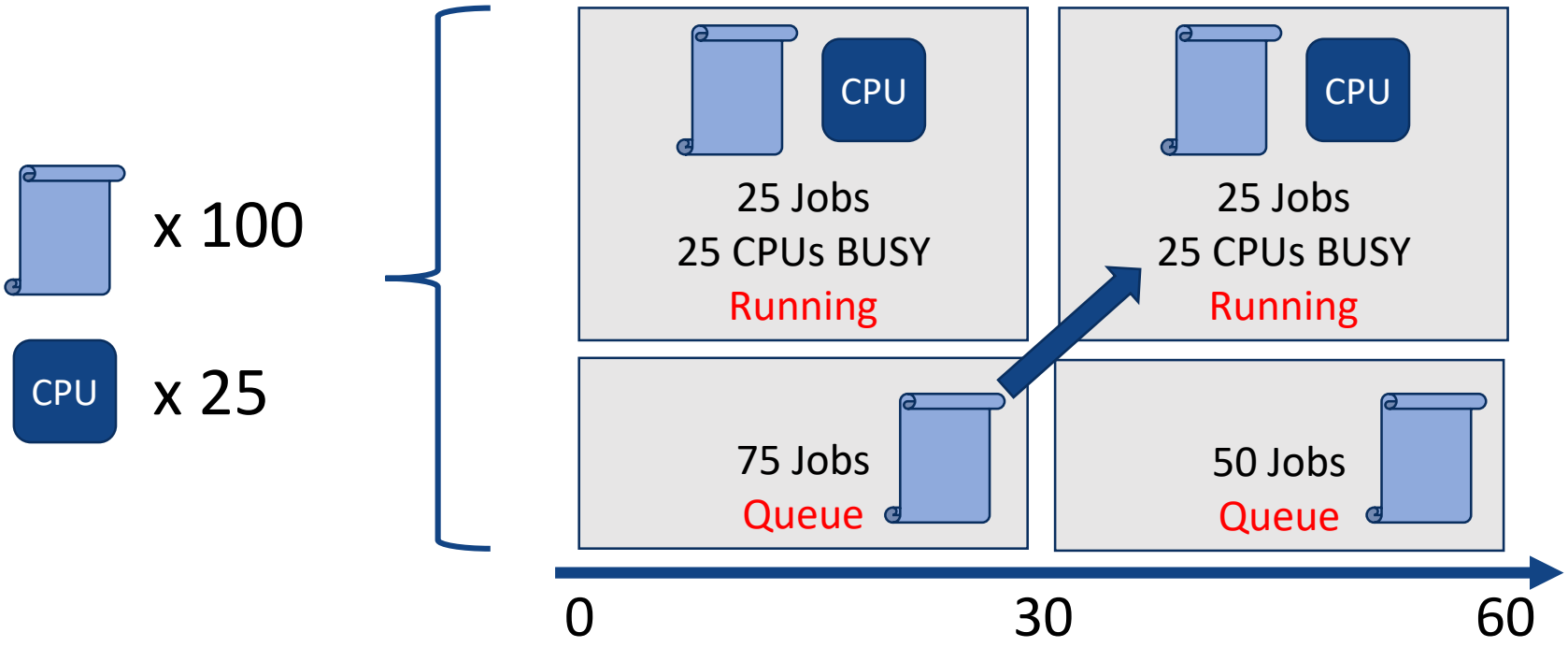
| | |
|---|---|
| Arrival rate: | λ = 100 experiments / hour |
| Exps. take: | W = 0.5 hours [1 CPU per exp.] |
| Average exps. in: | L = 50 exps |



# Resource Limits

- 25 CPUs
  - Less than needed!

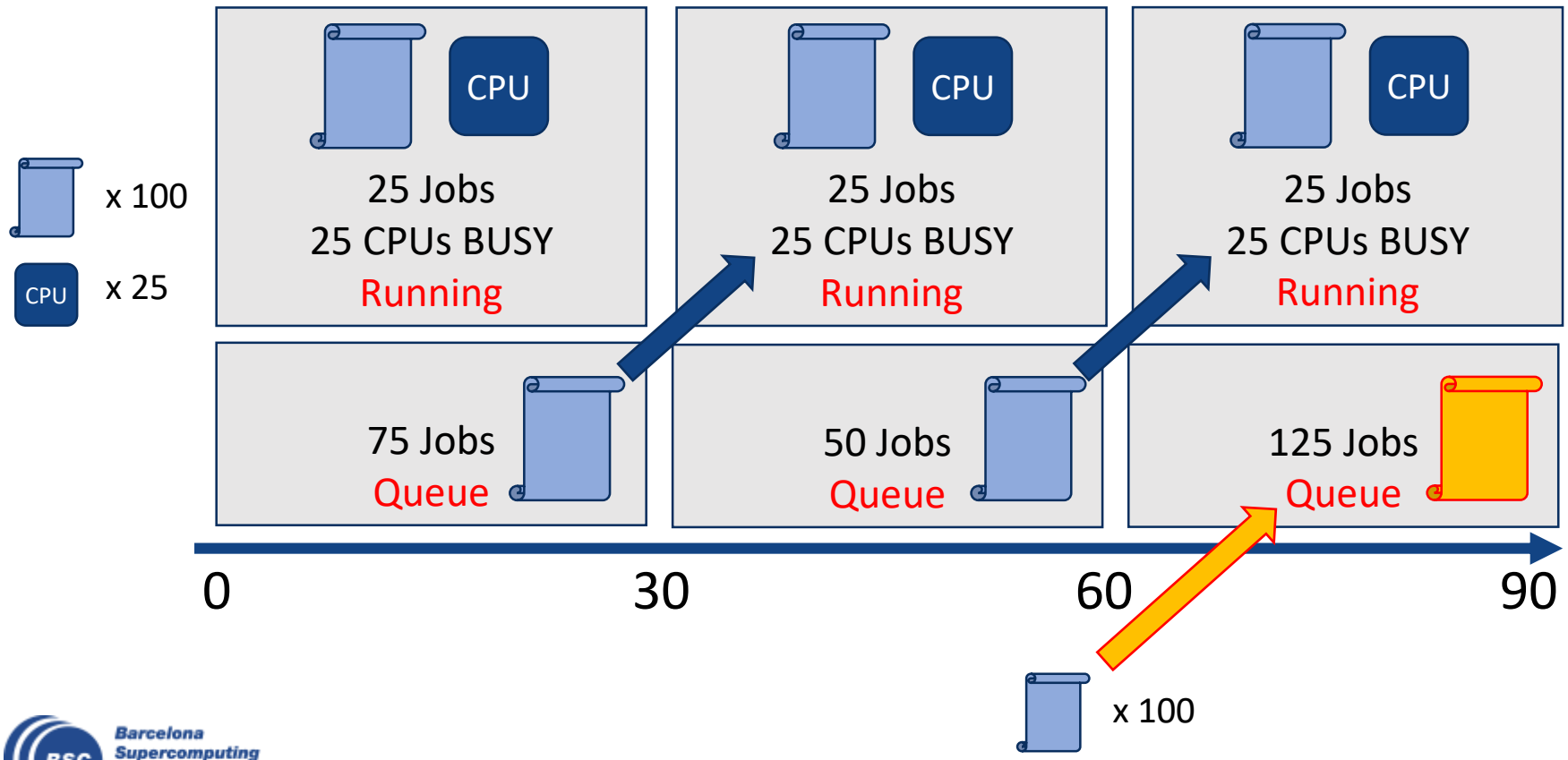
| | |
|---|---|
| Arrival rate: | λ = 100 experiments / hour |
| Exps. take: | W = 0.5 hours [1 CPU per exp.] |
| Average exps. in: | L = 50 exps |



# Resource Limits

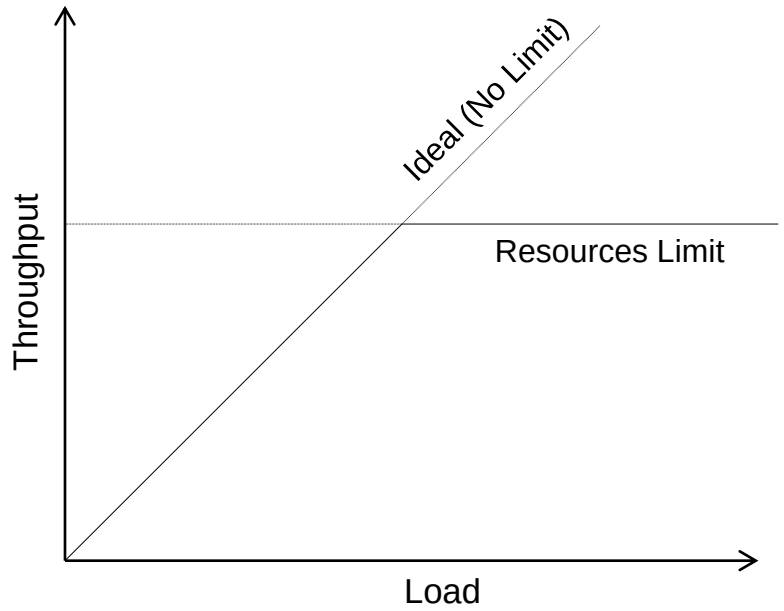
- 25 CPUs
  - Less than needed!

| | |
|---|---|
| Arrival rate: | λ = 100 experiments / hour |
| Exps. take: | W = 0.5 hours [1 CPU per exp.] |
| Average exps. in: | L = 50 exps |



# Throughput

- Throughput: outcome per time unit
  - E.g. experiments finished per hour
  - E.g. data-sets processed per minute
  - E.g. data points trained per second



# Resource Competition

- Systems do not always have “queues”
- Processes (applications) compete in the system
  - For using the CPU
  - For getting some memory
  - For accessing the disk and network (I/O)