Introduction to Computing Systems - from Scientific Computing's Perspective

Le Yan
HPC @ LSU

Goals

- Understand how their components work together
- Learn useful command line tools

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Why We Are Here

- Maximize productivity using computing systems as the (primary) research instruments
- Understand how your instrument works is key to success
- Not just research
  - A driver doesn’t have to be a mechanic, but having mechanic knowledge will be very helpful for drivers to fulfil their goals

Performance Matters in Scientific Computing

- Productivity matters

  Know how long some analysis should run (and know something is wrong when it takes longer)
  Know where to look when something is wrong
  Write Python script to automate lots of pre-processing and post-processing

Performance Matters in Scientific Computing

- Productivity matters

Know how long some analysis should run (and know something is wrong when it takes longer)
Know where to look when something is wrong
Write Python script to automate lots of pre-processing and post-processing

Performance Matters in Scientific Computing

- Productivity matters

Know how long some analysis should run (and know something is wrong when it takes longer)
Know where to look when something is wrong
Write Python script to automate lots of pre-processing and post-processing

Don’t know how long some analysis should run (and don’t know something is wrong when it takes longer)
Don’t know where to look when something is wrong
Don’t write python script to automate lots of pre-processing and post-processing, i.e. do everything by hand

Performance Matters in Scientific Computing

- Productivity matters

Know how long some analysis should run (and know something is wrong when it takes longer)
Know where to look when something is wrong
Write Python script to automate lots of pre-processing and post-processing

Don't know how long some analysis should run (and don't know something is wrong when it takes longer)
Don't know where to look when something is wrong
Don't write python script to automate lots of pre-processing and post-processing, i.e. do everything by hand

Have An Expectation for Performance

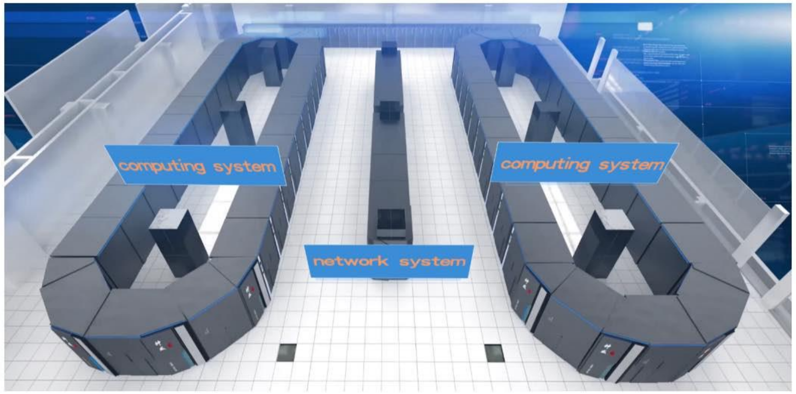
- Would you be suspicious
  - If it takes 5 minutes to open Microsoft Word on your laptop?
  - If it takes 15 minutes to download a song to your phone?
  - It takes 15 hours to assembly and align a 5GB sequence?

You Are Suspicious. Now What?

- Fortnite (or Youtube or whatever app) is laggy. What would you do?
  - Check network?
  - Other programs running on your computer?
  - Fan stopped running?
  - …

You Could Face Similar Problems on Supercomputers

- Biggest supercomputer in the world
  - More than 10 million cores
  - Power consumption: 15 MW (10,000 people)
- Being "super" doesn't make performance issue go away



Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

What Do Computers Do?

- To compute, of course.
- What does it exactly mean?

What Do Computers Do?

- To compute, of course.
- What does it exactly mean?
- To process data, transforming it from one form to another
  - Ex: open a web page on your laptop
  - Ex: play video games
  - Ex: process genome sequence

CENTER FOR COMPUTATION & TECHNOLOGY

What Do Computers Do?

- In each of the examples
  - Raw data is obtained from some places (and stored on a computer)
    - Input devices (mouse and keyboard)
    - Other computers and devices
  - Raw data is processed by the computer
  - Processed data is presented in various forms

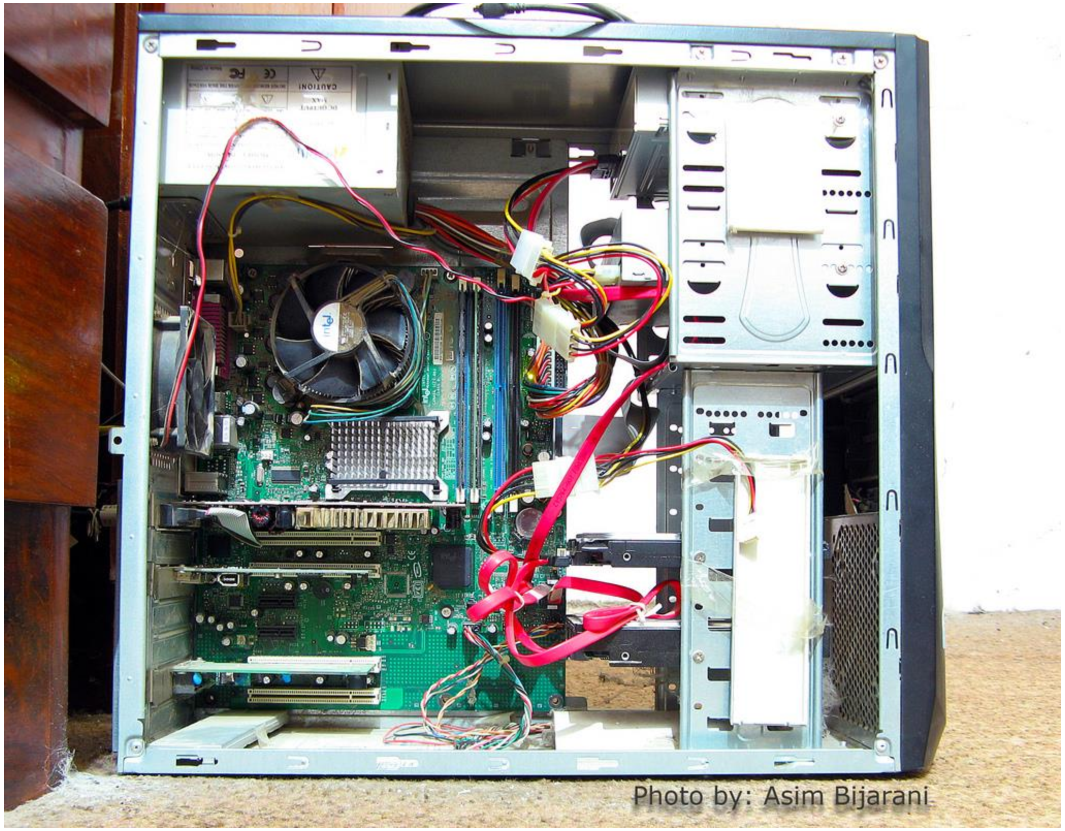
Computers vs. Manufacturing Plants

Components of A Plant

- Workshops
  - Product assembly lines
  - Staging areas
- Warehouses
- Pipelines

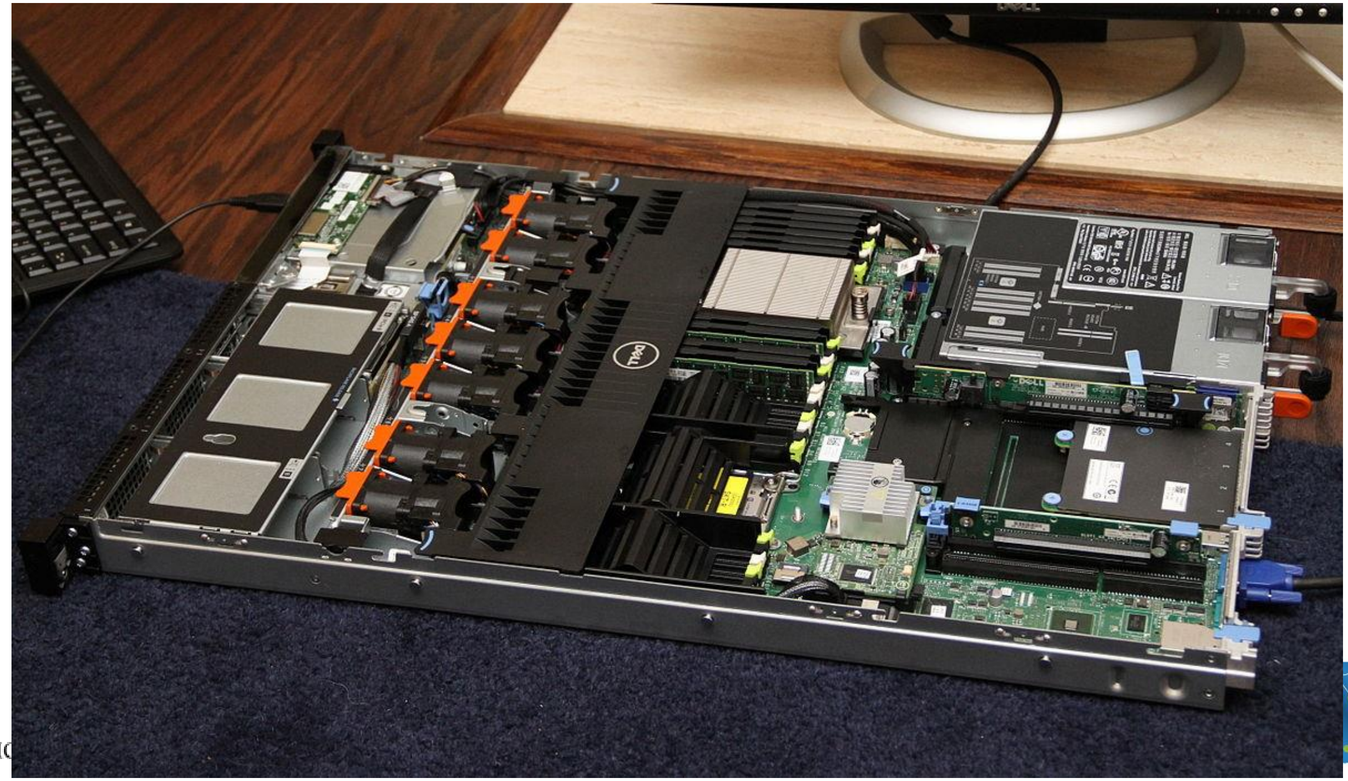
Components of A Computer

- Recognize anything?



Component of A Computer

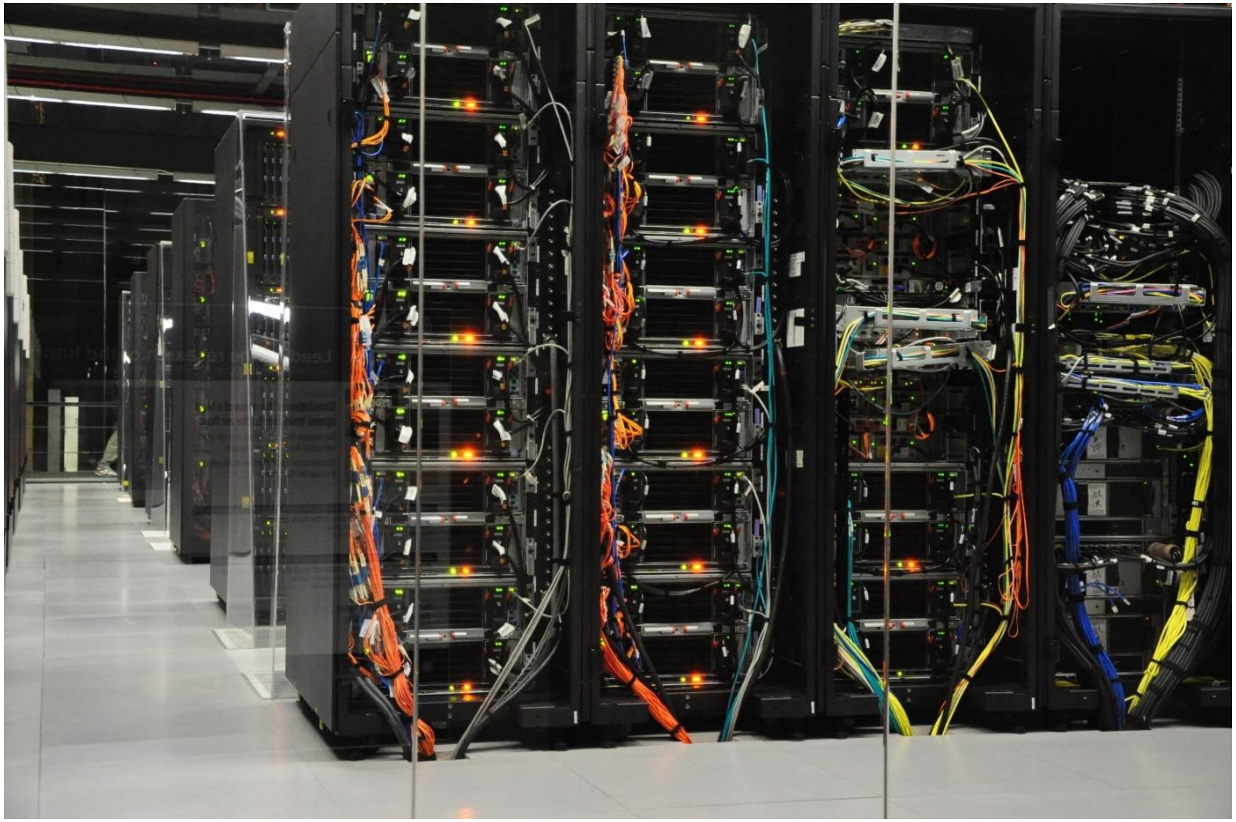
- What about this?



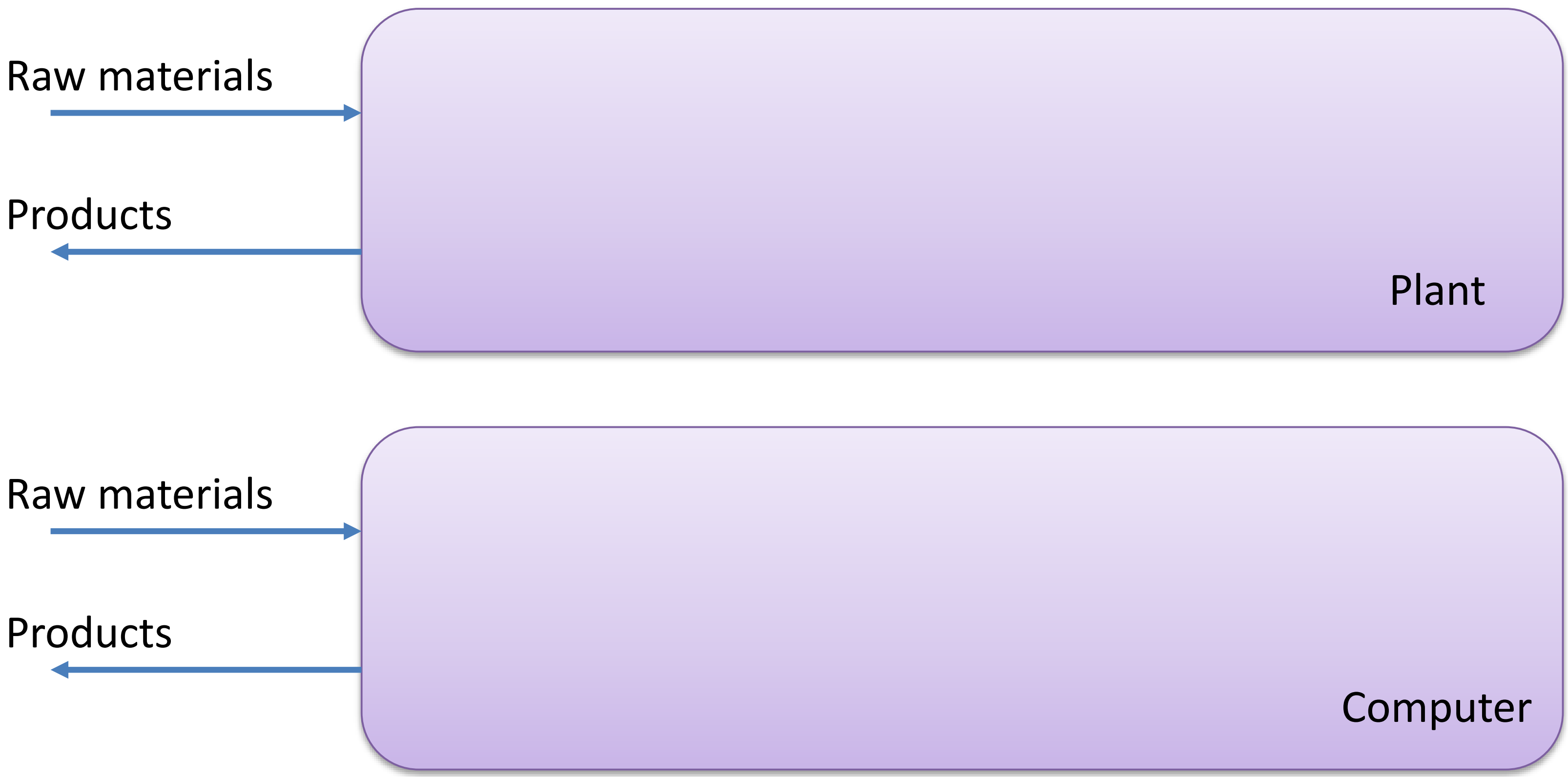
Component of A Computer

- And this?

Components of A Computer



Computers vs. Manufacturing Plants



Computers vs. Manufacturing Plants

Manufacturing plants	Computers
Raw material	Data
Assembly line	CPU
Storage space (warehouse, staging area etc.)	Storage system
Pipeline	Bus

Anything missing?

CENTER FOR COMPUTATION & TECHNOLOGY

Computers vs. Manufacturing Plants

Manufacturing plants	Computers
Raw material	Data
Assembly line	CPU
Storage space (warehouse, staging area etc.)	Storage system
Pipeline	Bus
Operation manual/plan	Software (another form of data)

Goals (translated)

- Trained to be a manager who is capable of running a plant effectively and efficiently
- Contents as practical and simplistic as possible
  - Details that have no immediate effects are likely to be omitted, e.g. how data is stored in the storage devices, details of network protocols

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Data – Know Your Raw Materials

- All information in a computing system is represented by a bunch of bits
- A **bit** is either 0 or 1
  - Denote by lower case “b”
- They are arranged into 8-bit clunks, each of which is called a **byte**
  - Denote by upper case “B”

Know Your Numbers

Prefix	Quantity
K (kilo)	1,000
M (mega)	1,000,000
G (giga)	1,000,000,000
T (tera)	1,000,000,000,000
P (pita)	1,000,000,000,000,000
E (exa)	1,000,000,000,000,000,000

Know Your Numbers

- Memory size (~GB)
- Hard drive capacity (~TB)
- Network bandwidth (~Gb)
- Kindle ebooks (KB ~ MB)

0's and 1's are only meaningful in a context

- 8 bits (1 byte): 01110101
- Integer (10-based): 117
- Text: “u” (lower case)
- 8-bit grayscale:
- Could as well be a part of 4-byte integer, 4-byte real, 4-byte color etc.

Types of Data: Textual

- ASCII (American Standard Code for Information Interchange): a map between characters and one-byte integers.
- Human readable, but not machine readable
  - A human-readable source file needs to be “compiled” to something that the machine can understand

Types of Data: Binary

- Executables: bits that are arranged so that CPU can execute it directly
  - The process of translating human readable source files to binary executables is called "compilation", completed by special software called "compiler".
- Binary data files

Installation vs Compilation

- Compilation
  - Translate human-readable source files to machine-readable binary executables
  - One type of software installation
- Installation
  - Can either be compilation or coping text files to desired location or a hybrid of both

Command Line Interface

- On the Jupyter Notebook page, click on “new”, then select “terminal” in the pull down menu
- At the prompt, type “python hello.py”

Exercise

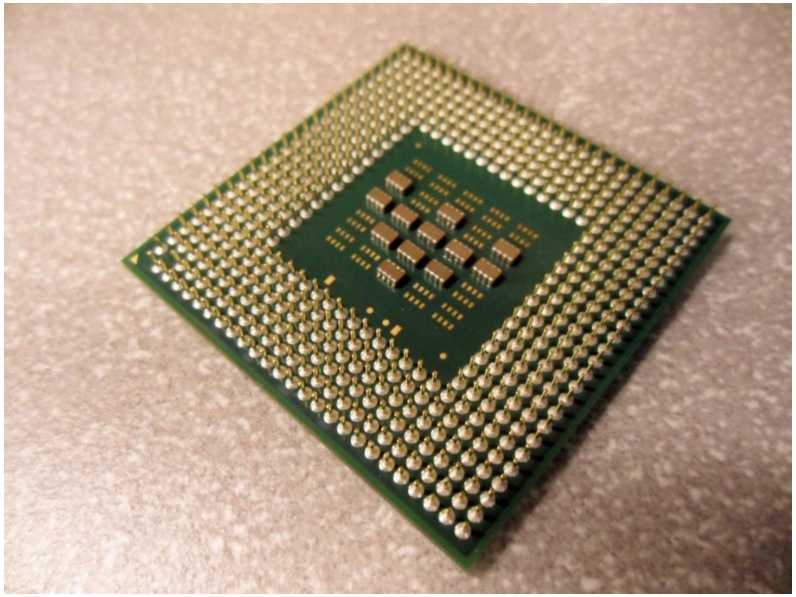
- Find out the size of this file.
- Find out the type of this file (text? binary?).
- Print the content of `hello.py`.

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

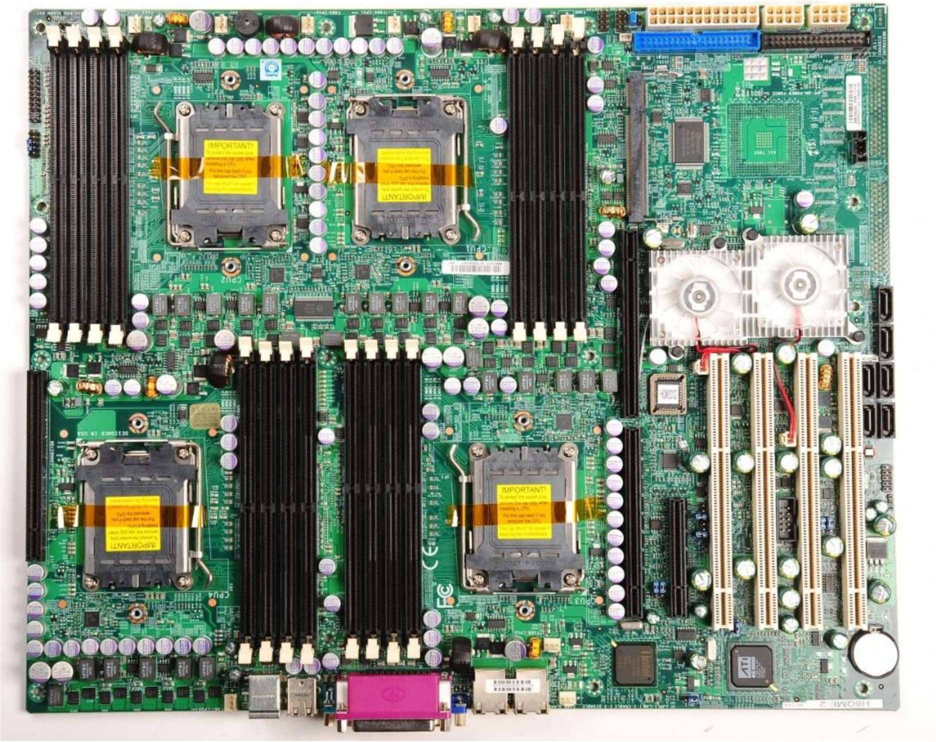
Central Processing Unit (CPU)

- Or simply processors
- This is where the numbers are crunched

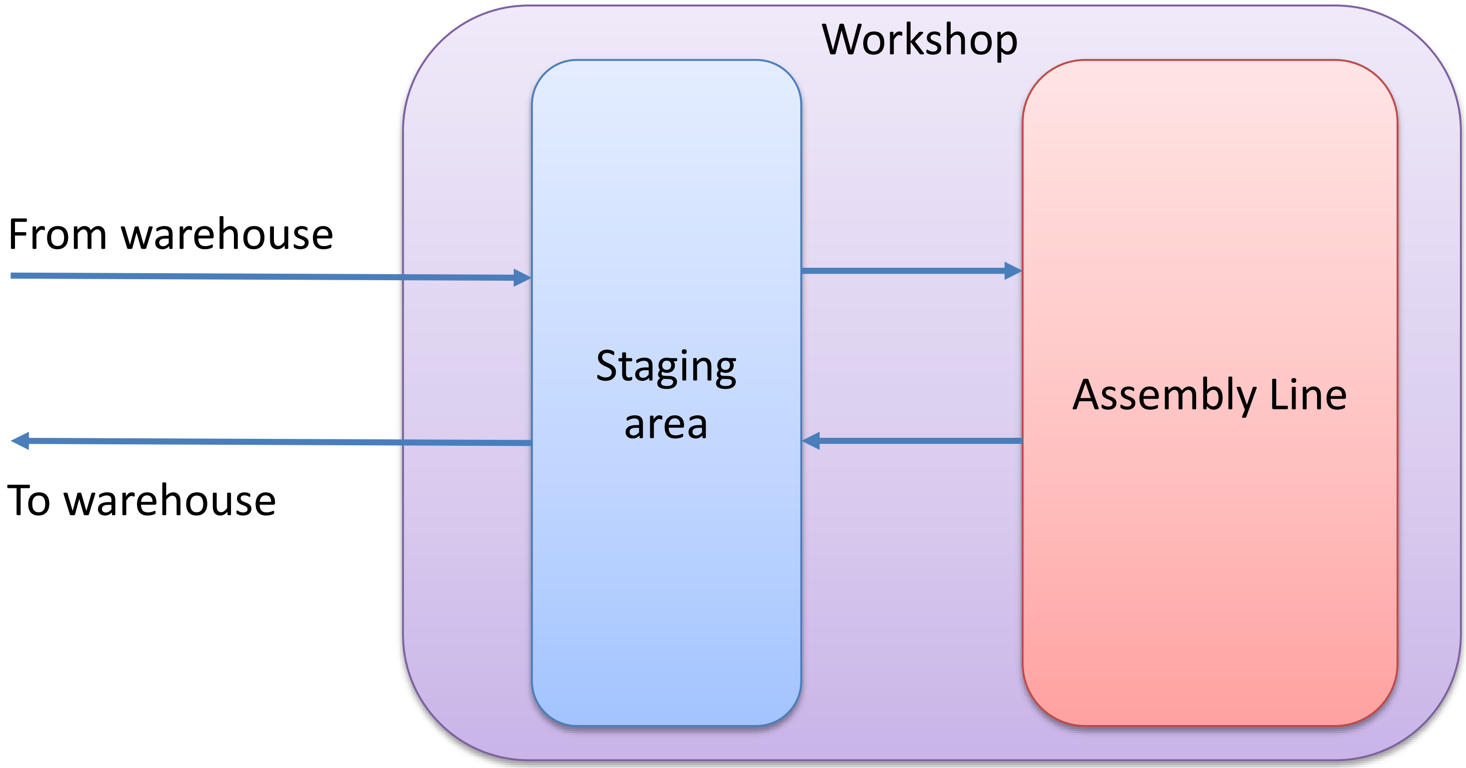


Central Processing Unit (CPU)

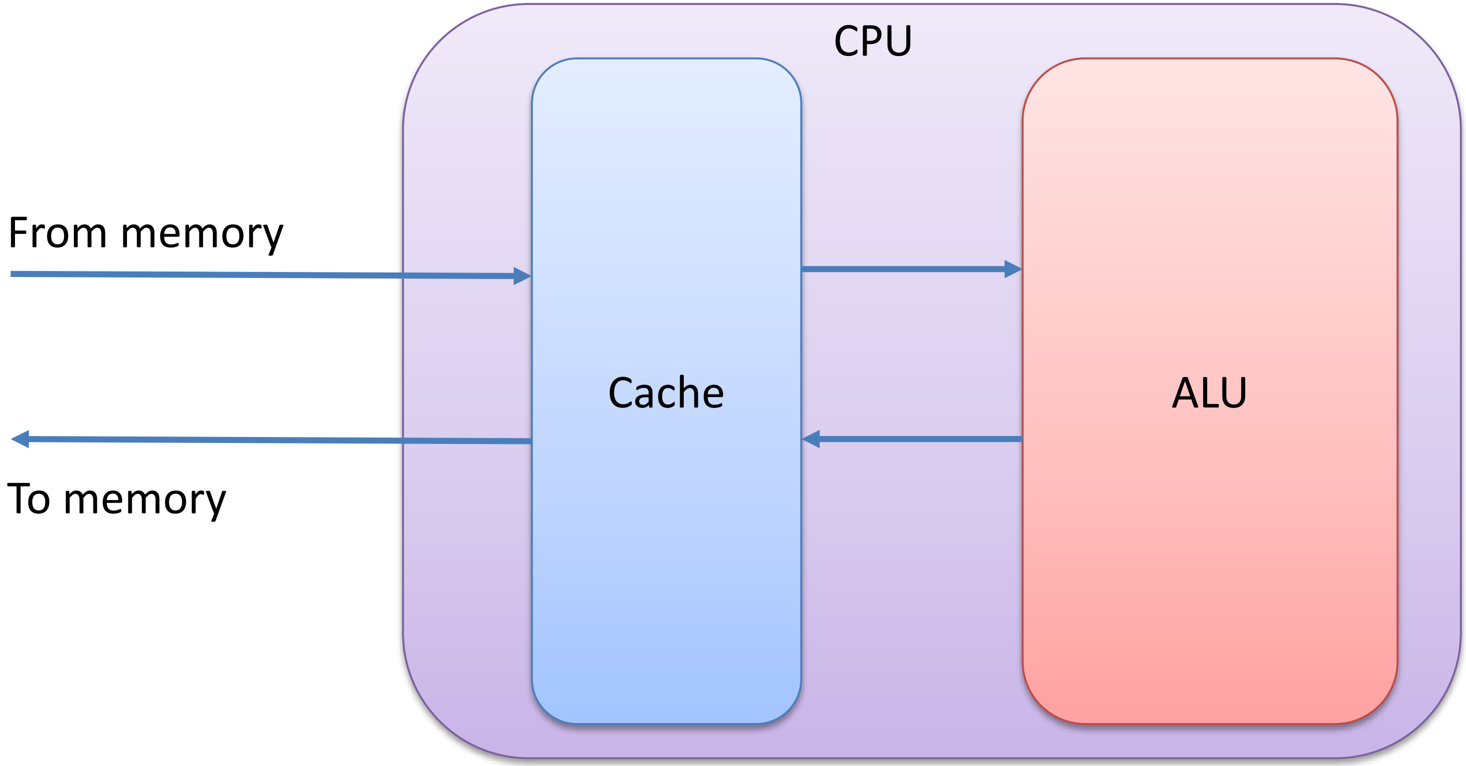
- A computer can have more than one CPU
- How many CPU sockets are shown in this picture?



Manufacturing Plant



In CPU



Cache

- Memory on chip
  – Can read and write data at a higher speed than the main memory
- More on this later

Floating Point Operation

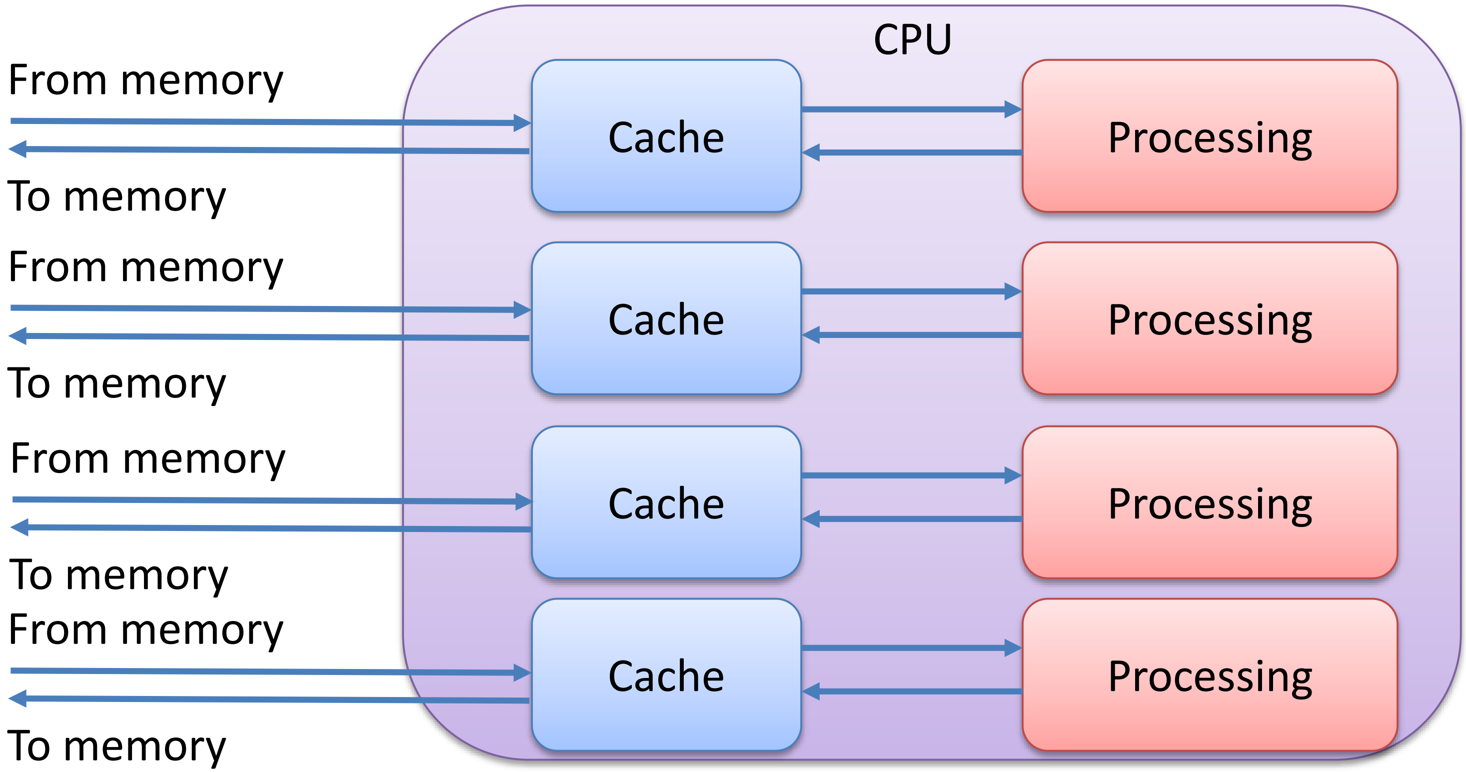
Measuring CPU’s Speed

- Cycles per second
  - Ex: 2.6 GHz = 2.6 x 10^9 cycles per second
- CPU process data with instructions
  - One instruction = operate on a few bytes of data
    - Ex: addition, multiplication etc.
- Modern CPU’s are capable of executing more than one instruction per cycle

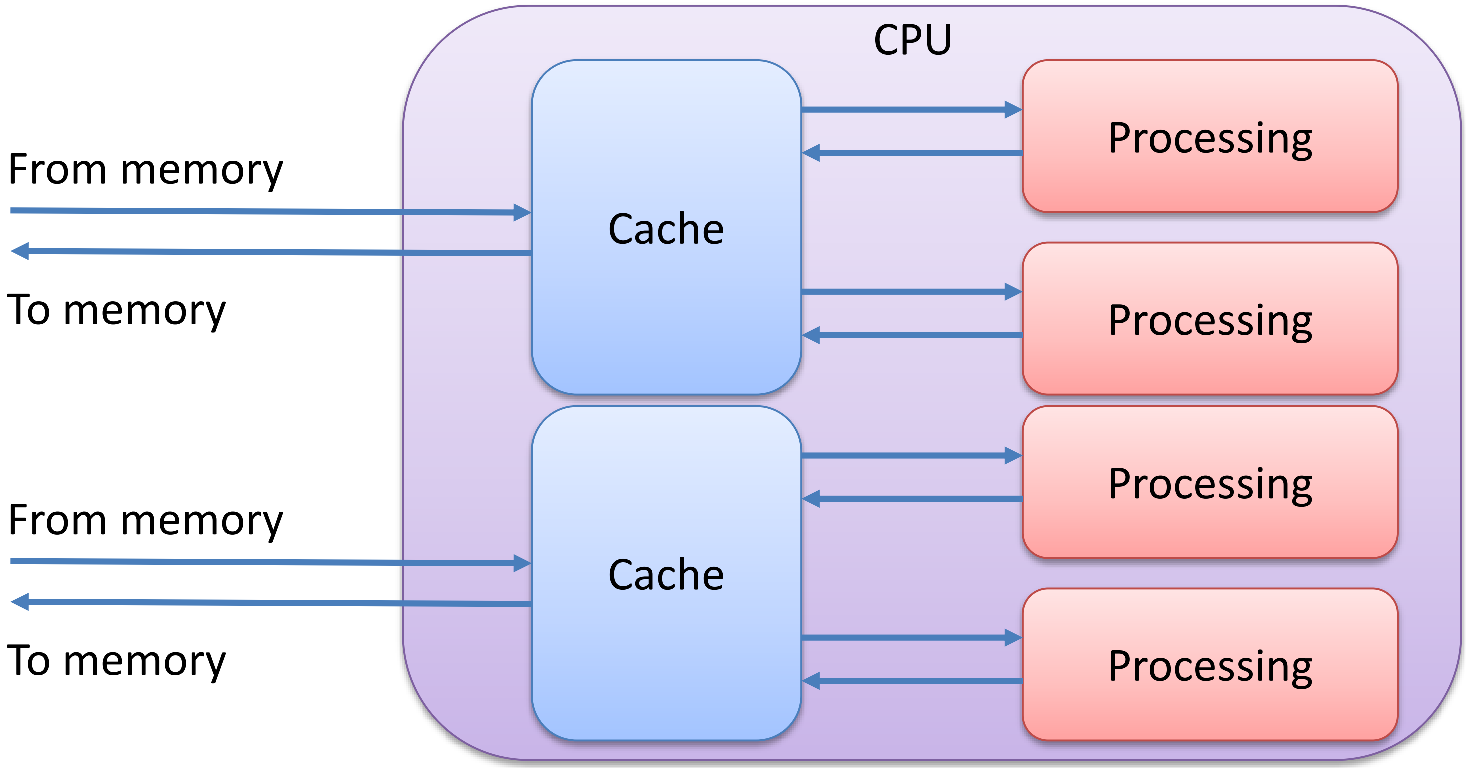
Multicore Architeture

- In modern CPU's there are usually more than one processing unit
  - Multiple assembly lines in a workshop
  - Current generation of Intel CPU's (Skylake family) could have anywhere between 2 to 28 cores
- Each core can performance different tasks simultaneously.

Different Design with MultiCores



Different Design with MultiCores



Measuring CPU's Speed - Again

- With multicore architecture and multi-instruction per cycle, the actual speed of CPU, as measured by number of floating point operations, is:

  (Frequency) x (Number of instructions per cycle) x (Number of cores)

- Ex: Intel Skylake 6148 CPU @ 2.40 GHz

  $2.4 \times 10^9 \times 16 \times 20 = 768$ GFLOPS

Exercise

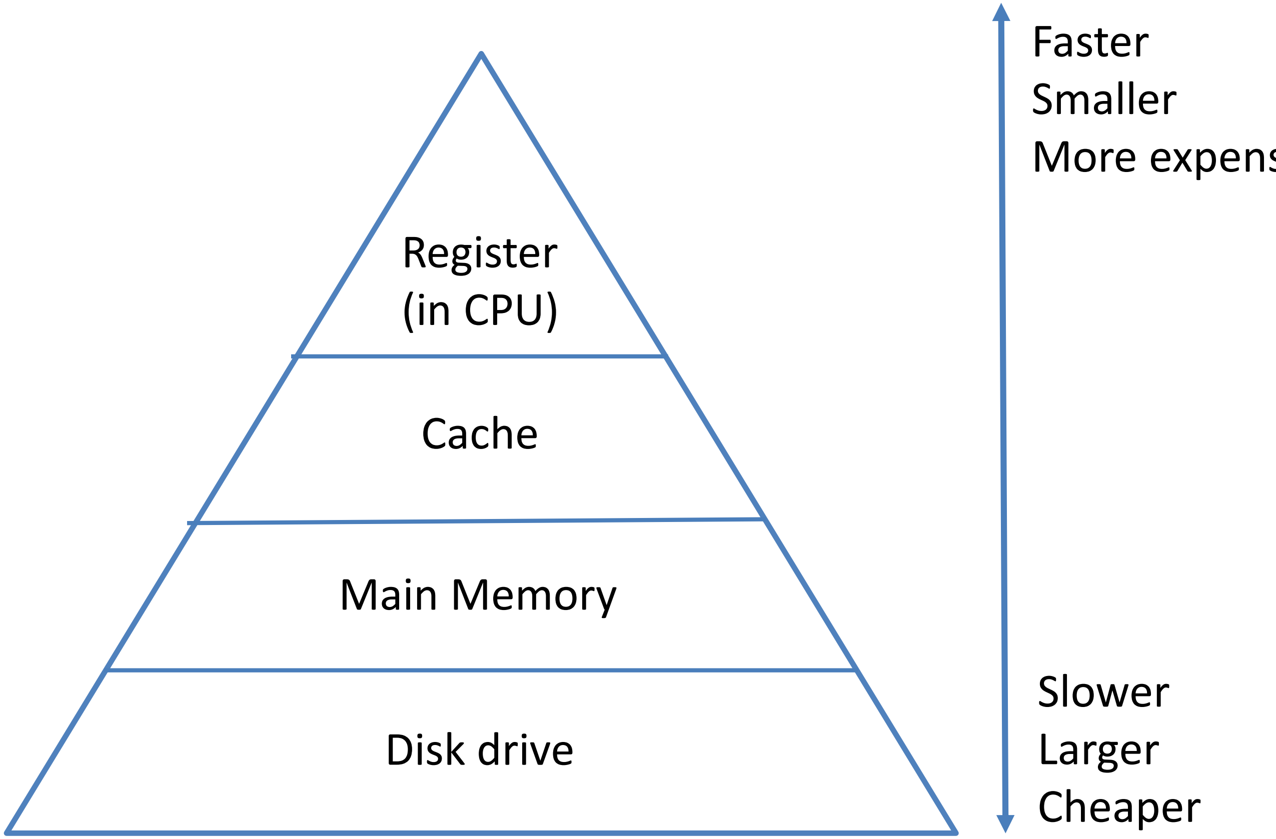
- Find out how many sockets and cores there are on the node where Jupyter is running.
- Explain this text: “Each server is equipped with two sockets Intel Sandy Bridge processor, each with 10 cores @ 2.60 GHz.”

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Storage Hierarchy

- In computers storage is organized as a hierarchy
  - Each level serves as a cache for the next level

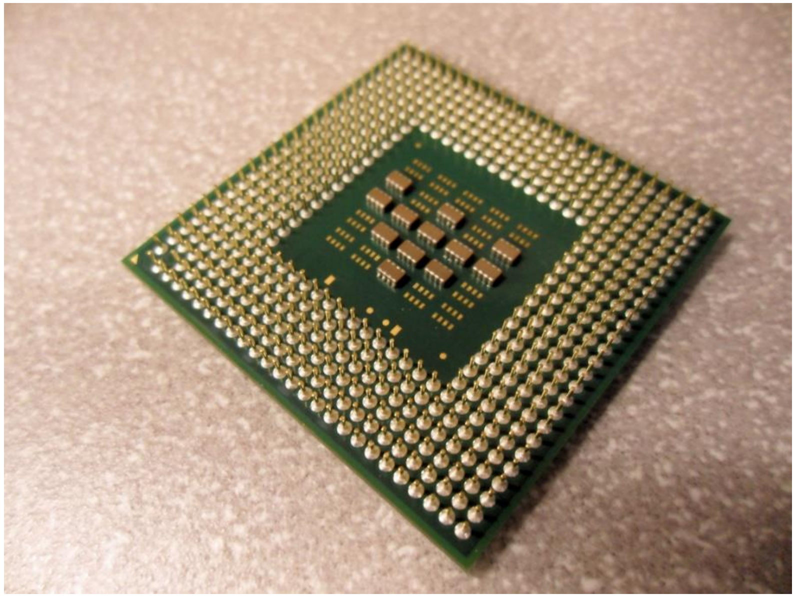


Why Use Cache?

- Processor-memory gap – CPU's are capable of processing data at a much higher speed that the storage devices can supply
  - CPU's are almost always hungry
- Ultra-fast storage device is very expensive to build
- Cache serves as temporary staging areas for information that the processor is likely to need in the near future.

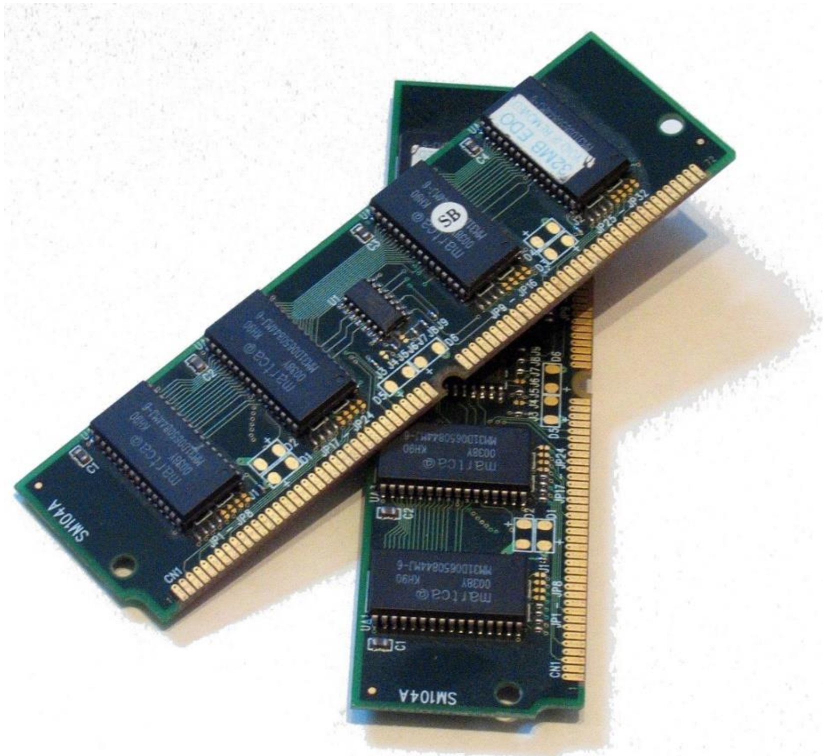
Cache - SRAM

- Dynamic Random Access Memory
- Volatile
  - Data is lost when power is turned off



Primary Storage - DRAM

- Dynamic Random Access Memory
- Volatile
- 100x cheaper than SRAM



Secondary Storage – Spinning Drives

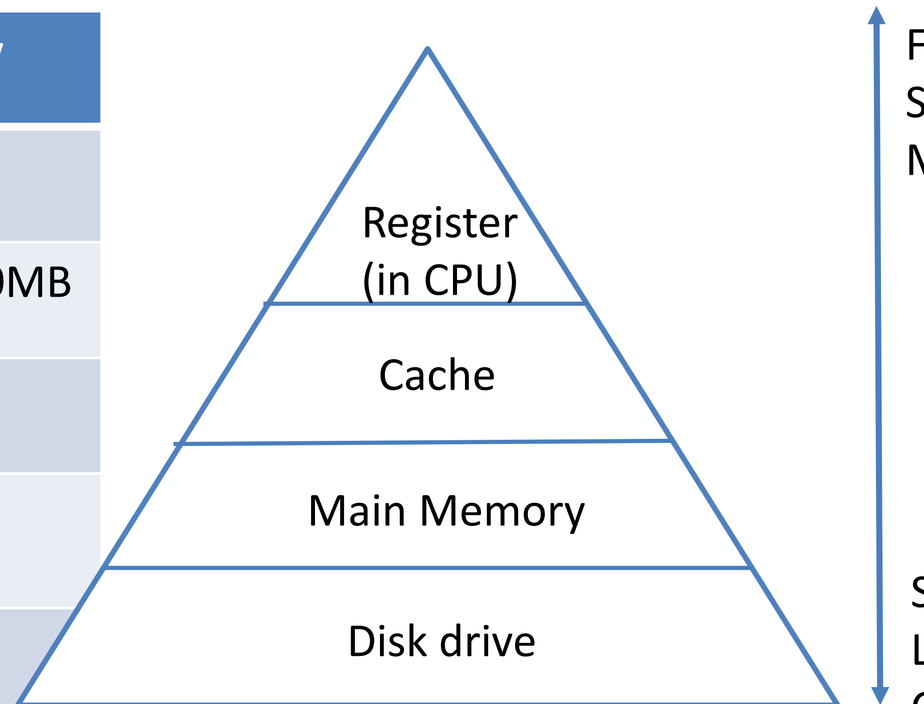
- Use electromagnetic media to store data
- Non-Volatile

Secondary Storage – Solid State Drives

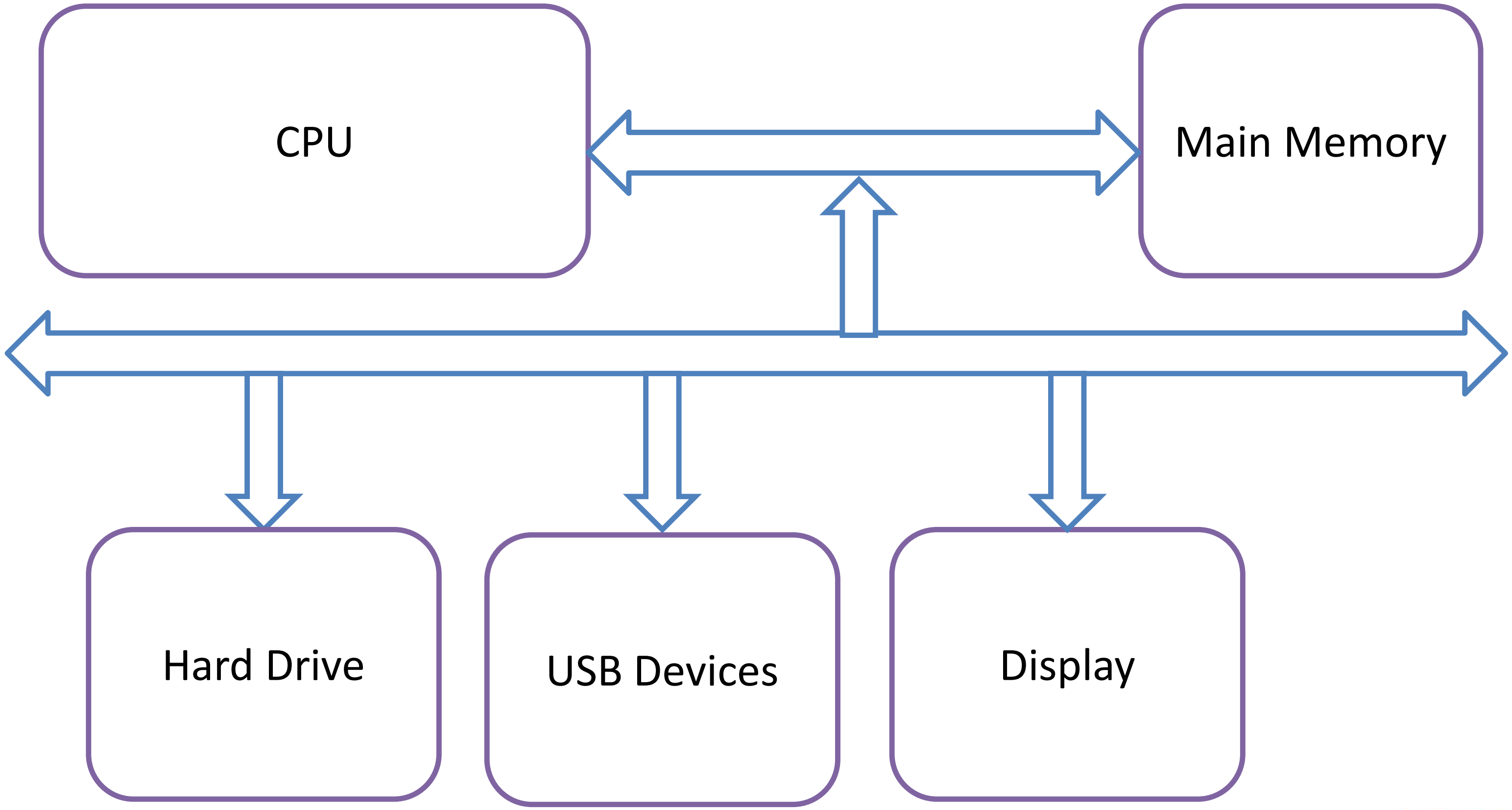
- Flash memory
- Non-Volatile

Performance and Capacity of Each Level of Storage

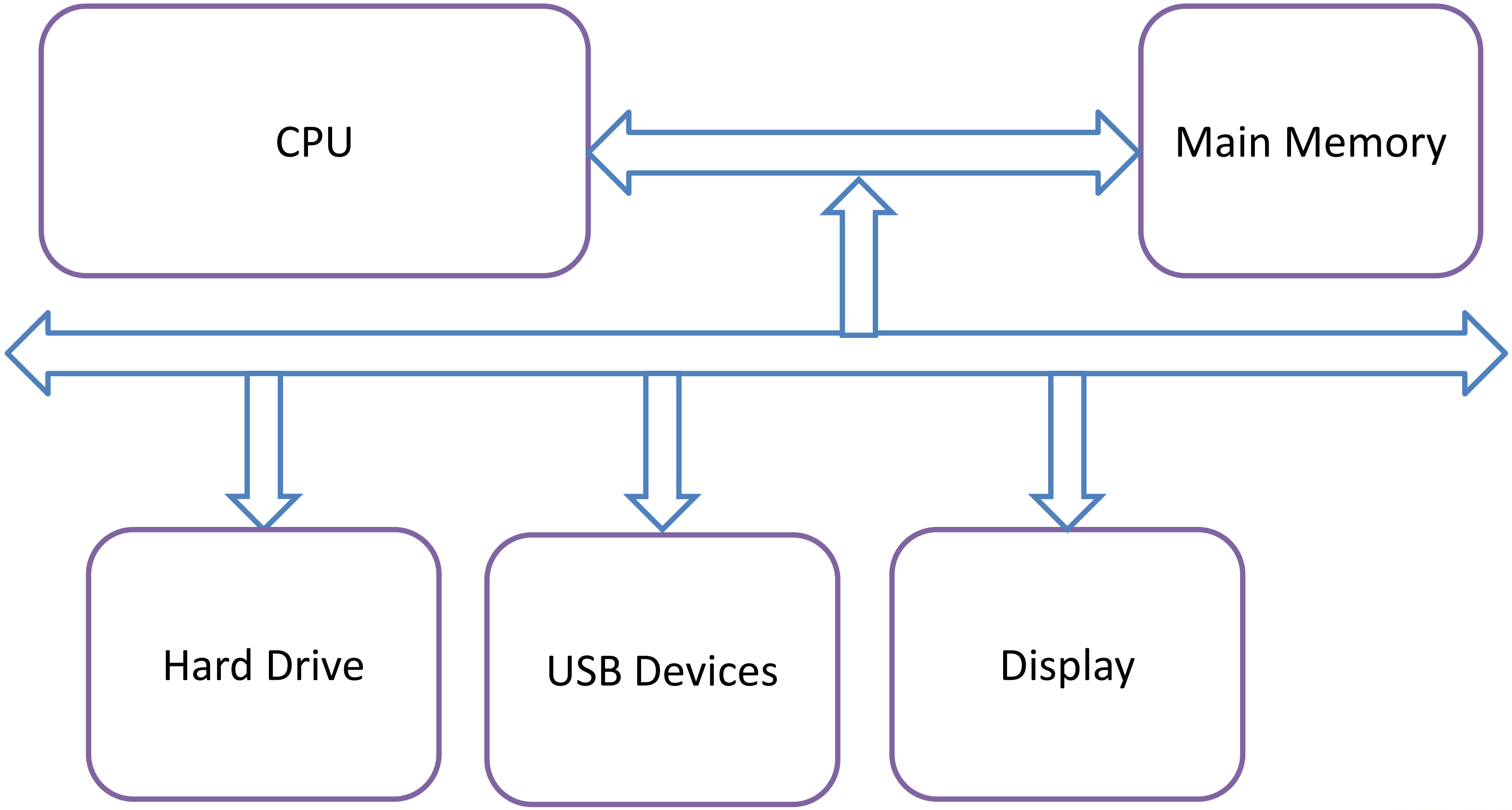
Level	Latency (cycles)	Capacity
Register	1	~ KB
Cache	~ O(1) – O(10)	~ KB - 10MB
DRAM	~ O(10) – O(100)	~ 100GB
Secondary - Spinning Drive	~ O(1,000,000)	~ 10TB
Secondary - SSD		~ TB



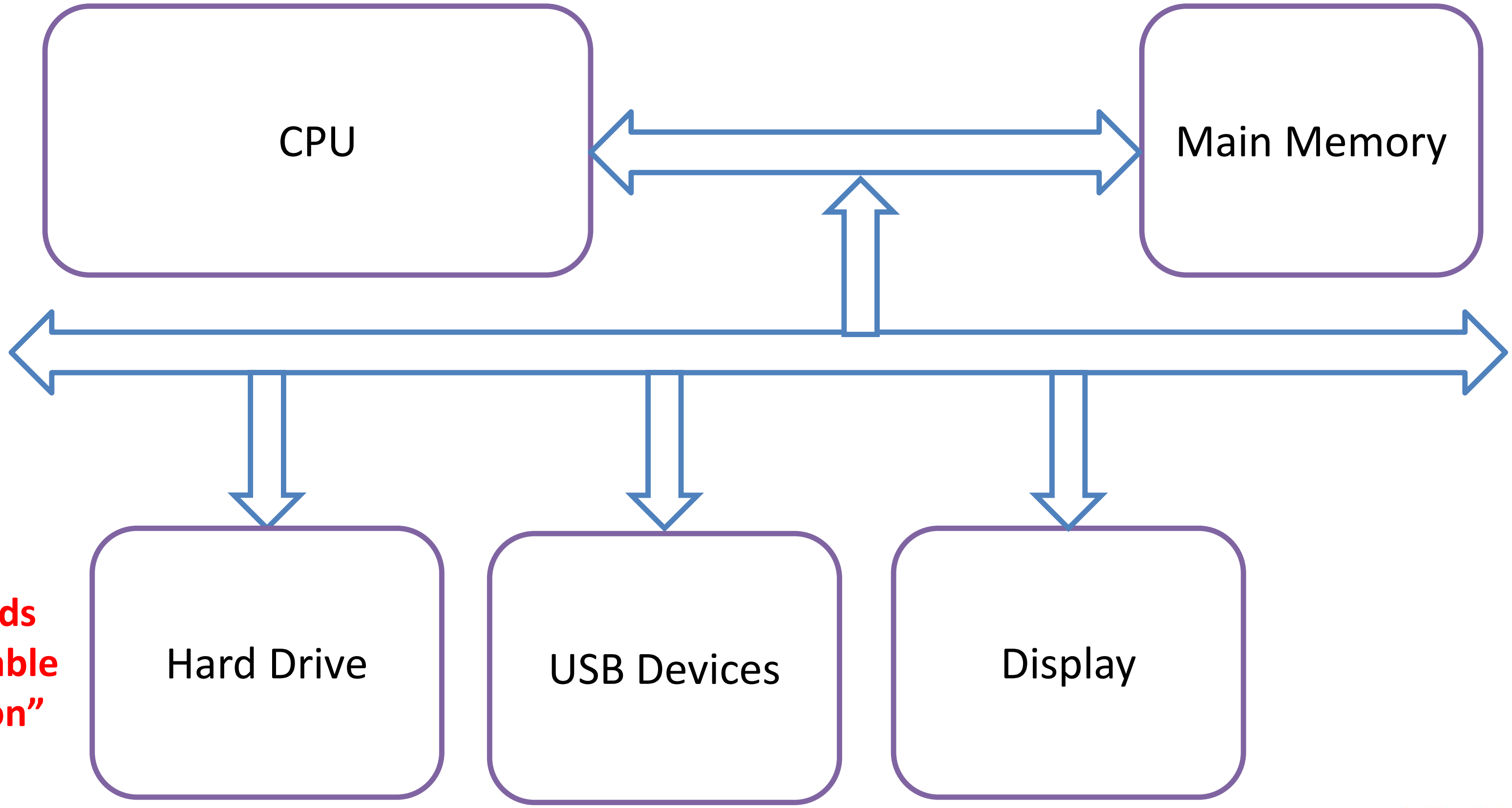
Computing Systems



Computing Systems

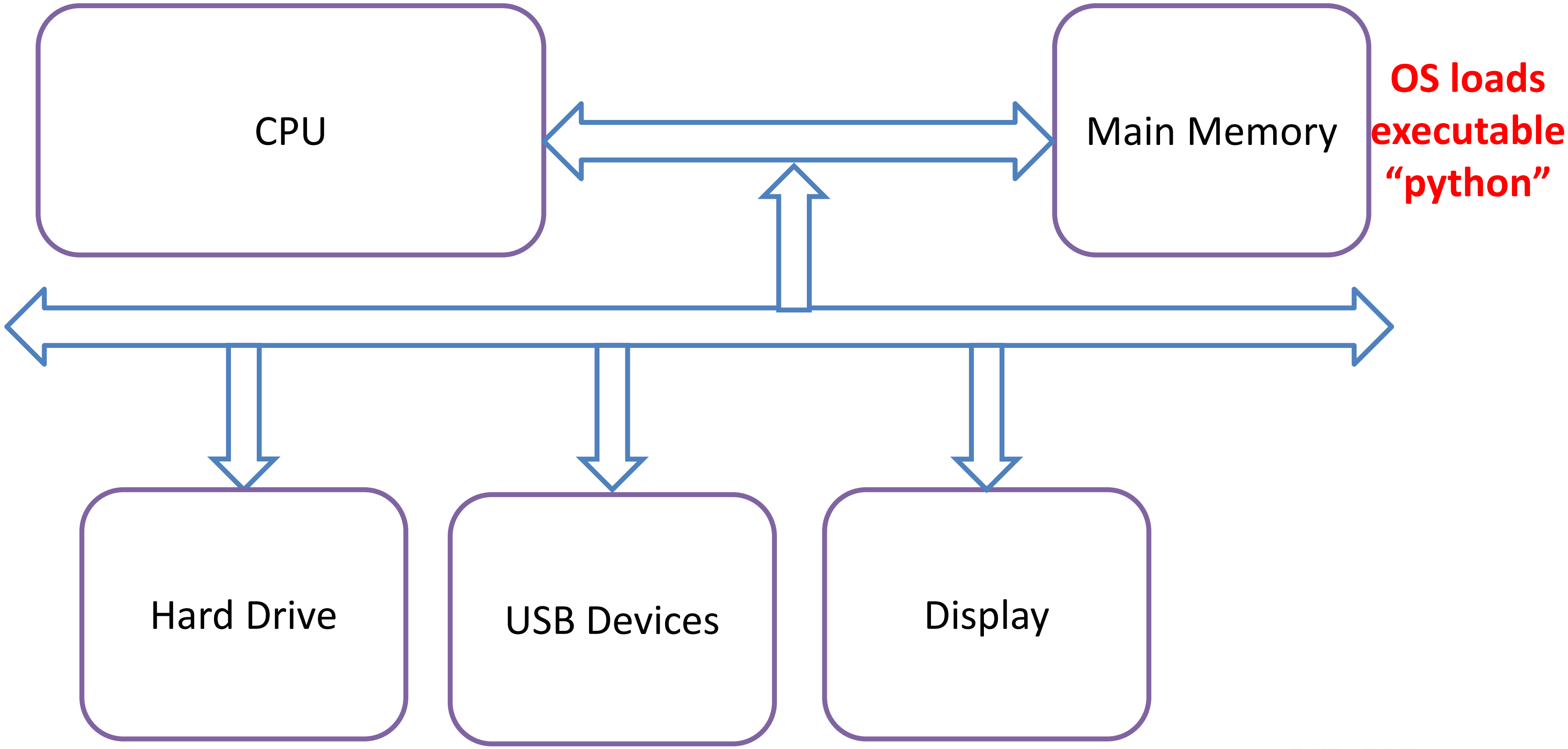


Hello World Step 1

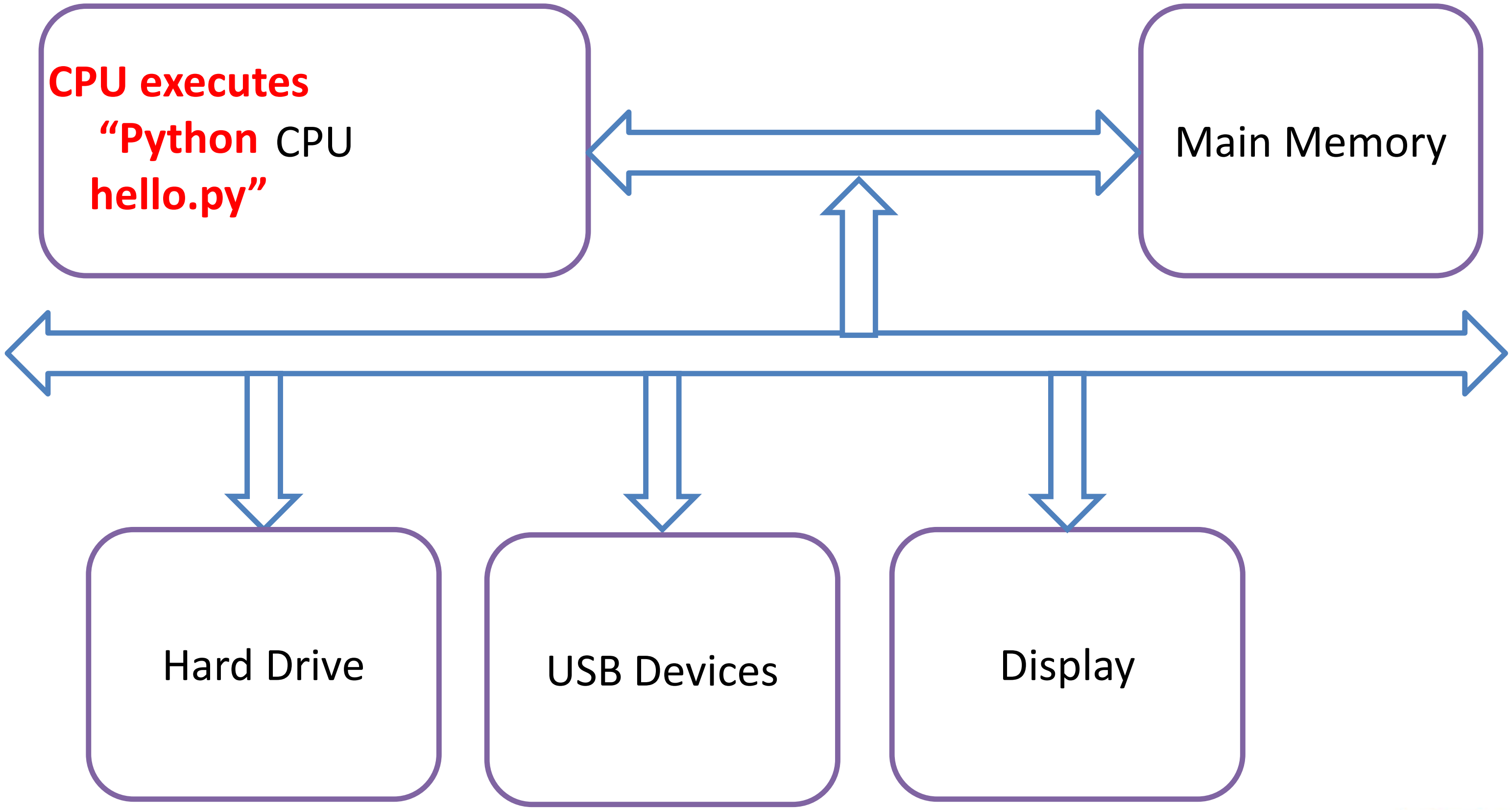


OS finds executable "python"

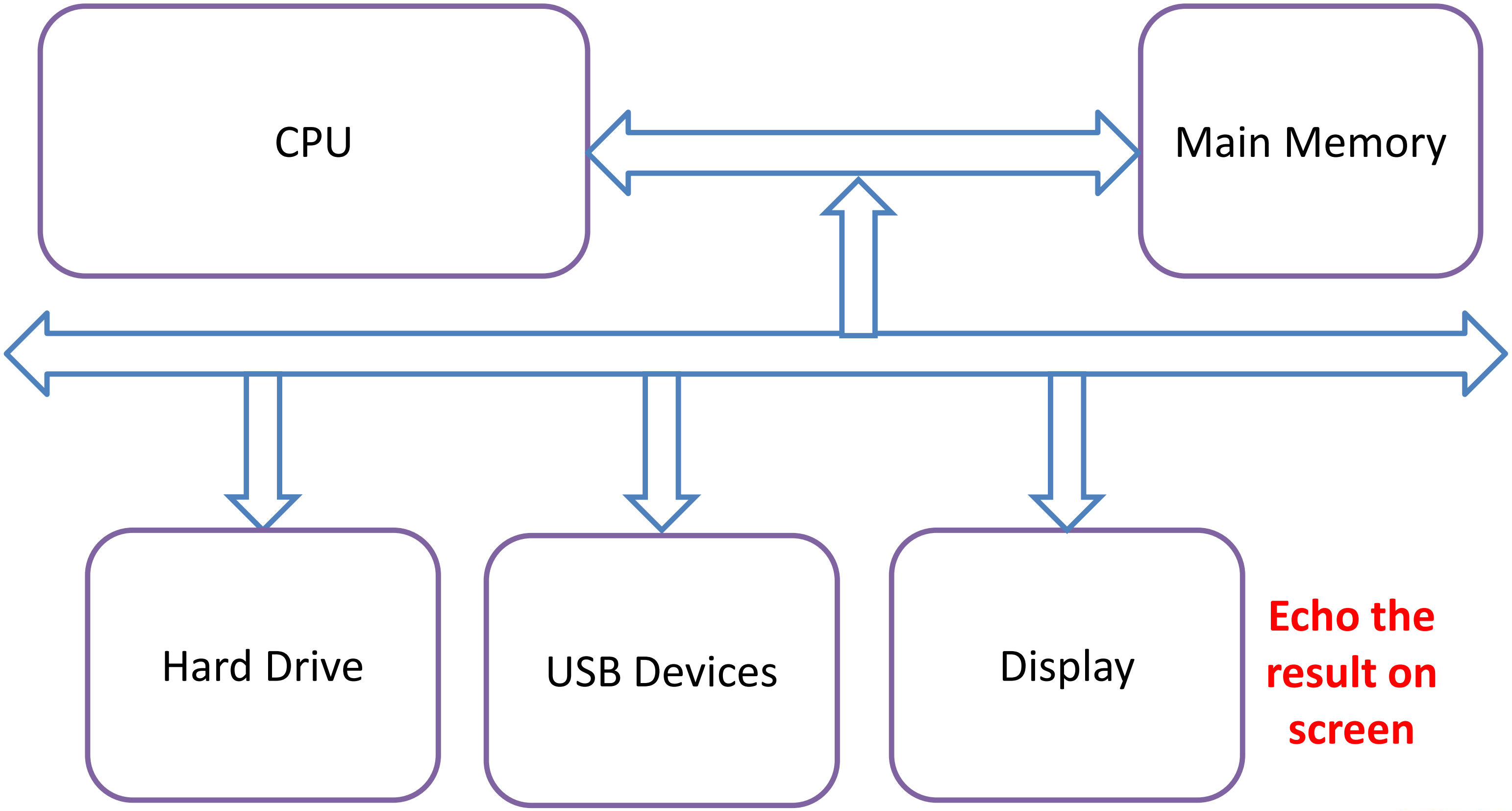
Hello World Step 2



Hello World Step 3



Hello World Step 3



Echo the result on screen

What We Learn

- A lot of time is spend on moving data around
  - Is this good or bad (for performance)?

Command

- Free
- Numactl

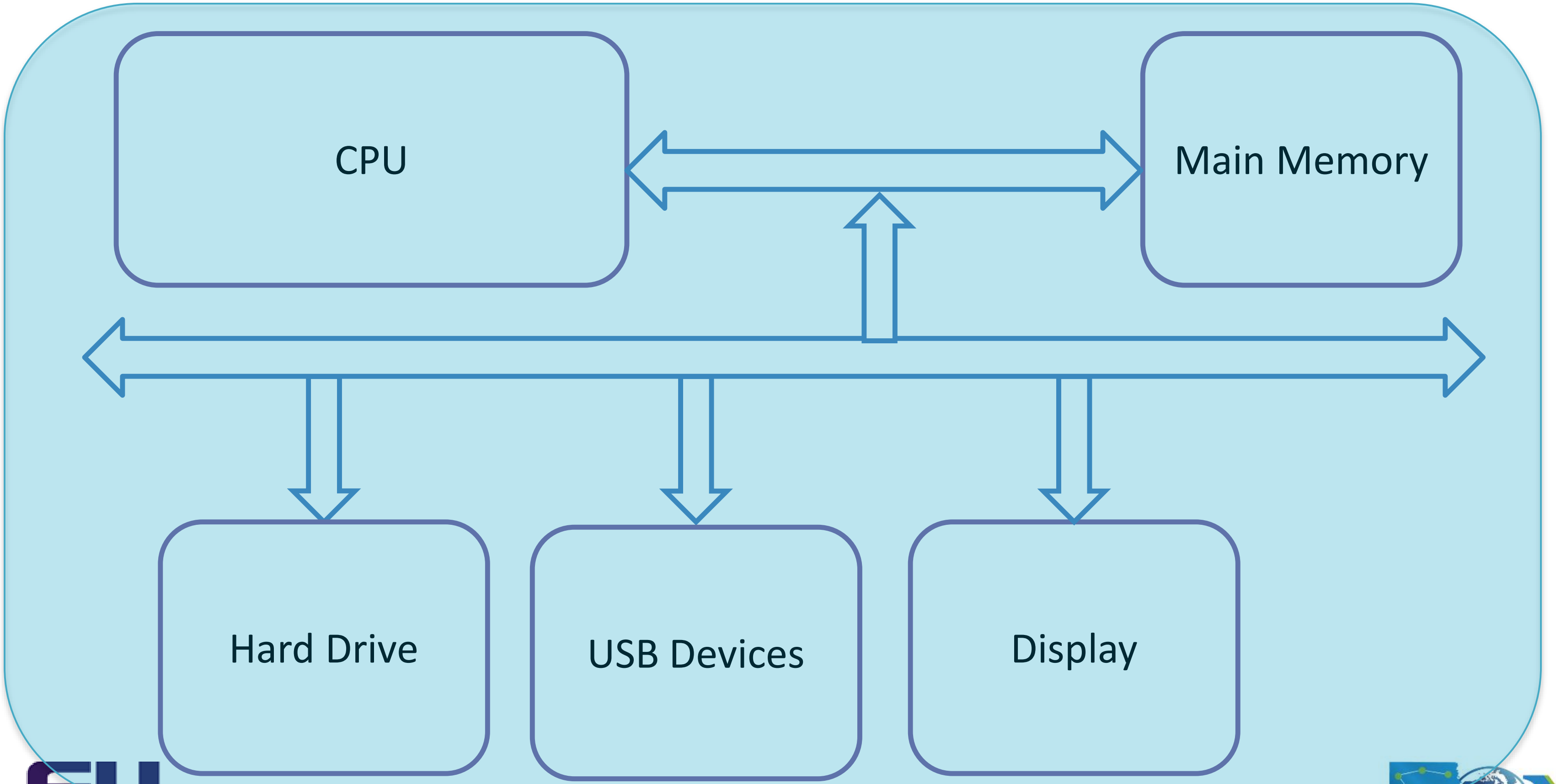
Exercise

- Find out the size of L1, L2, L3 cache as well as the main memory

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Operating System Orchestrate Program Execution



Operating Systems Manage Hardware

- Two purposes for operating systems
  - Protect the hardware from misuse by runaway applications
  - Provide applications **with simple and uniform mechanisms** for manipulating complicated and often wildly different low-level hardware devices.
- Hardware includes:
  - CPU
  - Primary storage (memory)
  - Secondary storage (disks)
  - And everything else
- Ex: Shell is part of the operating system
  - Provide a interface for us to interact with the computing system

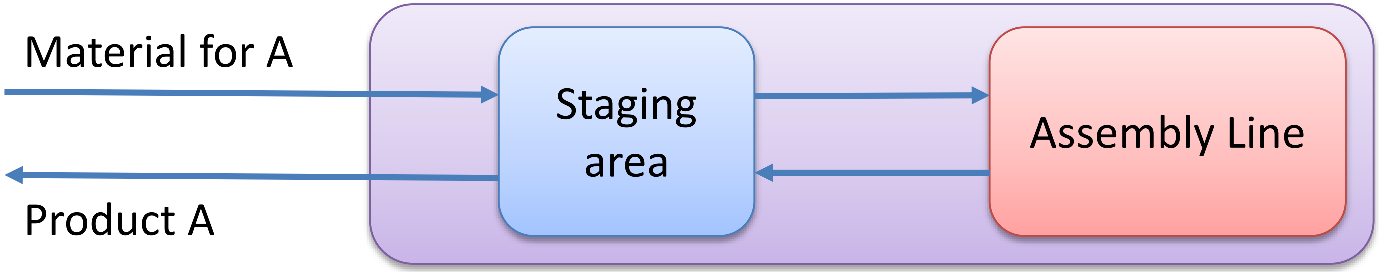
CENTER FOR COMPUTATION & TECHNOLOGY

Process

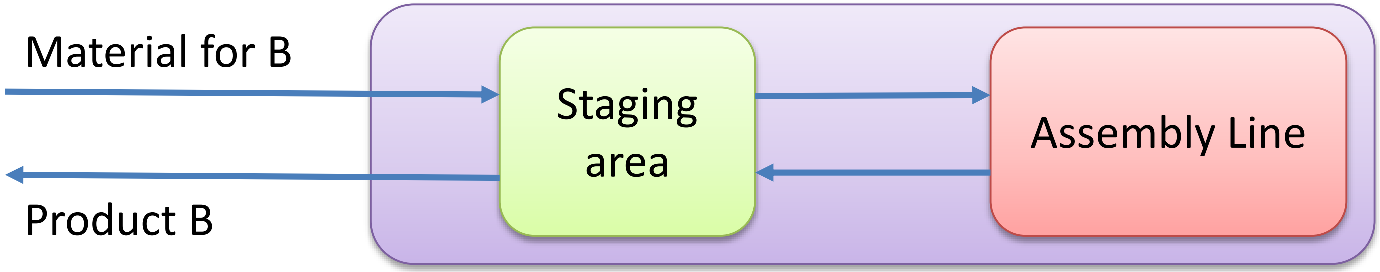
- The illusion/abstraction provided by OS that a program is the only one running on the system
- Using this model, multiple programs can run concurrently on the same system

Why Use Processes?

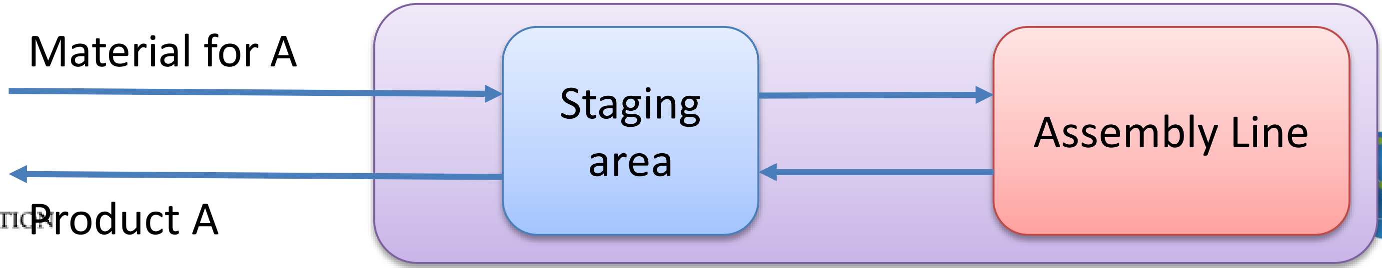
Imagine the plant produces product A, B and C with the same assembly line.



Suppose raw material for A is delayed. Switch to product B.



Switch back to product A when the raw material becomes avalable.



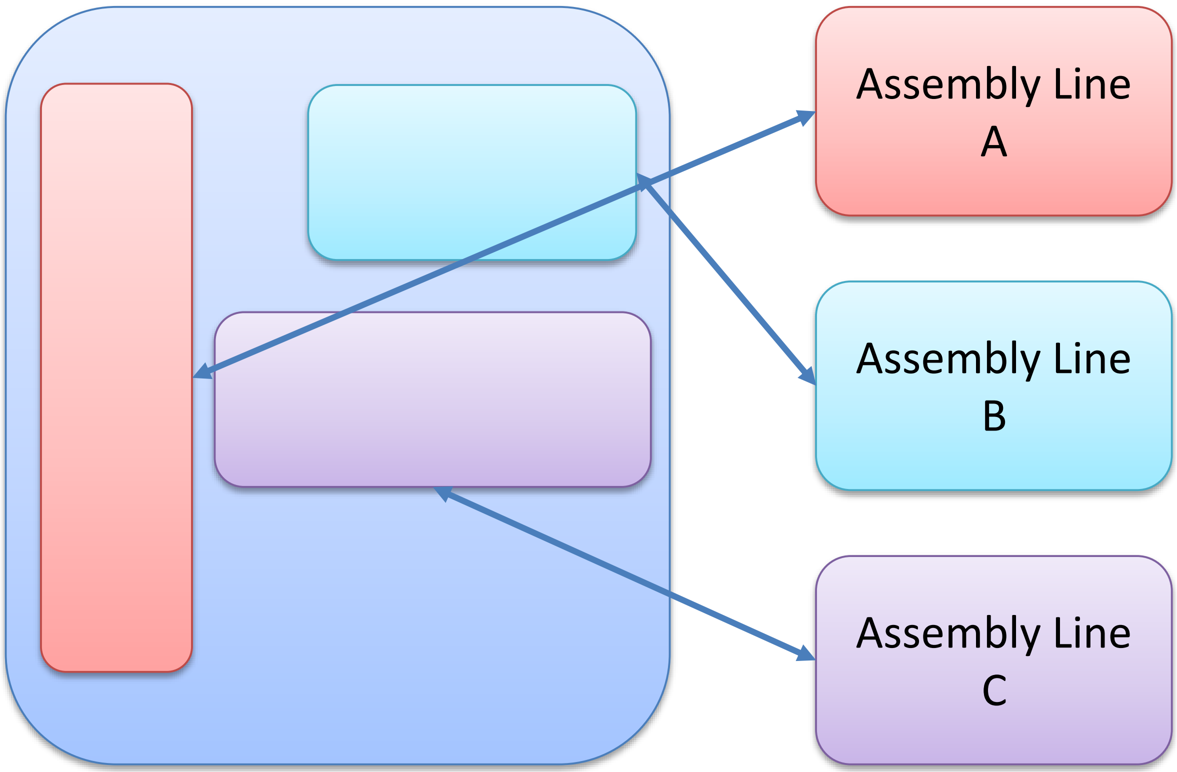
Threads

- A "light weight" process
- Multiple threads run concurrently within the same process
- Compared to processes, threads are less flexible, but it takes less efforts to switch contexts between threads
  - Tradeoff: performance vs flexibility

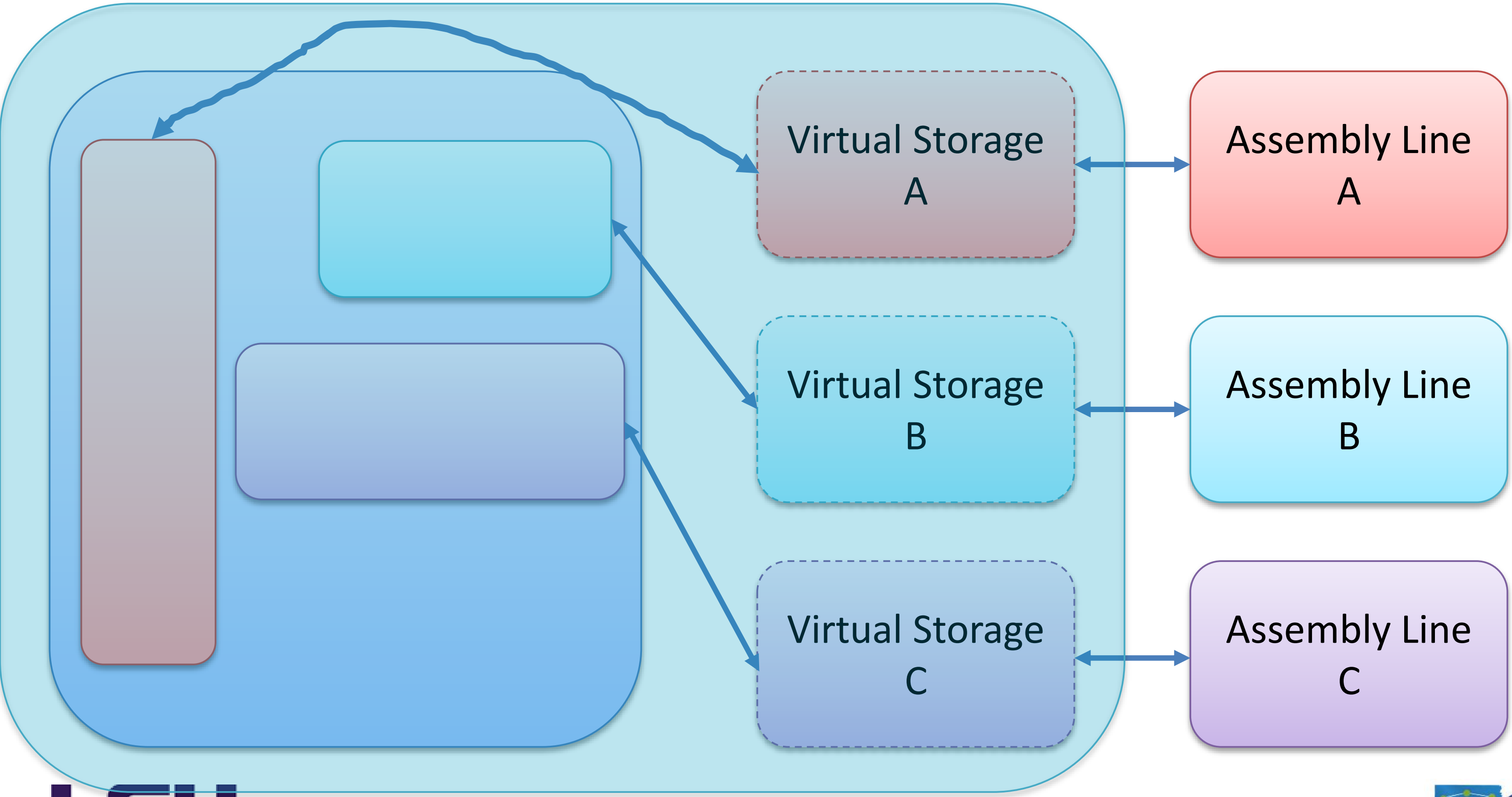
Virtual Memory

- The abstraction used by the OS to manage memory
- Virtual memory gives all programs an illusion that they have exclusive use of memory
- OS maps virtual memory to physical memory

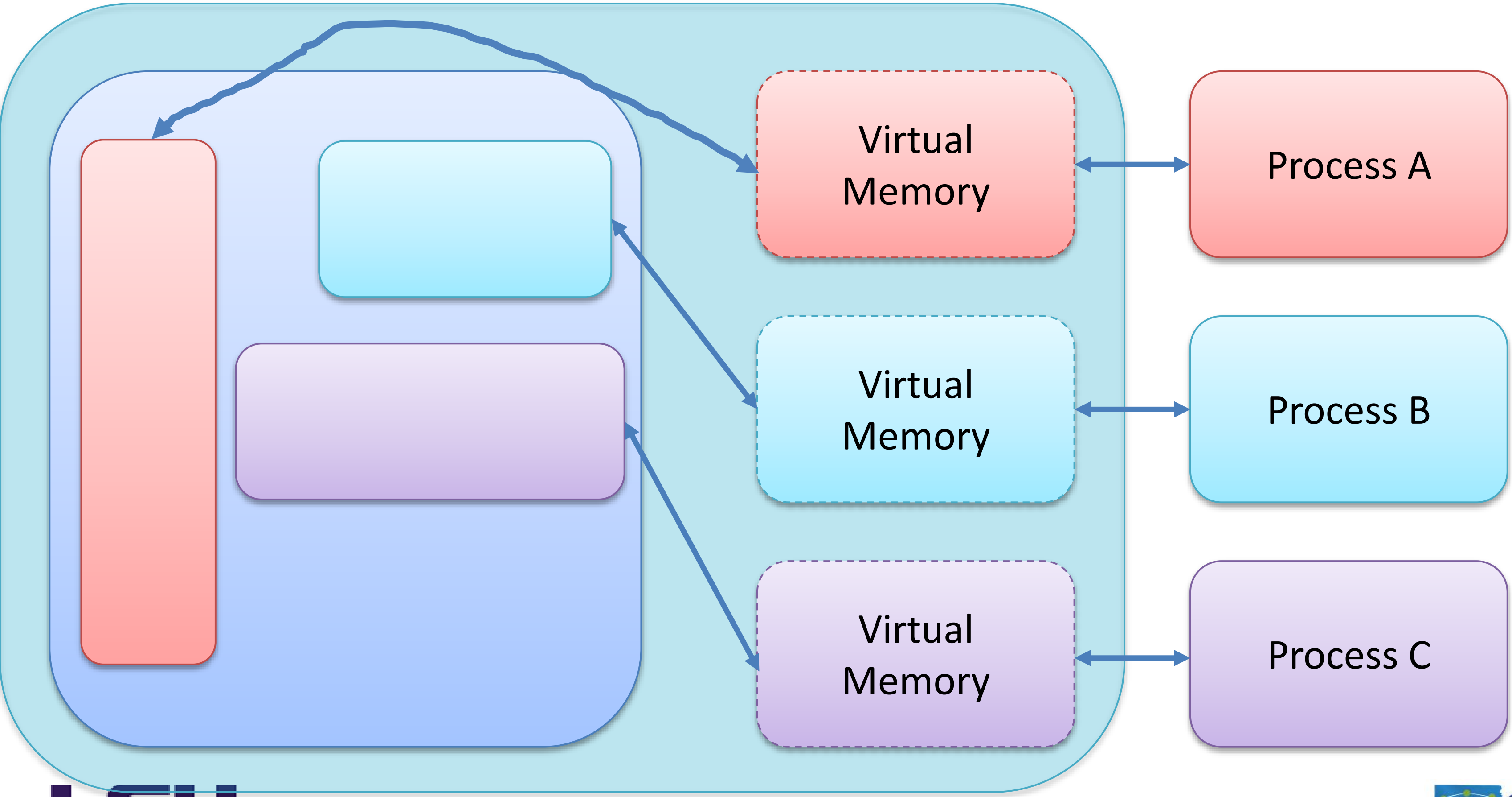
Virtual Memory



Virtual Memory



Virtual Memory



Files

- Files are a sequence of bytes
- From the OS point of view, files include
  - Normal files on disks
  - Keyboard
  - Display
  - All input/output devices
- More on this in the next section

Monitoring Processes

- The "ps" and "top" commands reveal information about running processes

```
top - 09:08:39 up 66 days, 23:14,  0 users,  load average: 0.00, 0.00, 0.00
Tasks: 672 total,   1 running, 671 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.1%sy,  0.0%ni, 99.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65945060k total,  9701748k used, 56243312k free,   163136k buffers
Swap: 134217724k total,     9824k used, 134207900k free,  6827404k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
18125 lyan1     24   4 27896 1856 1056 R  0.3  0.0   0:00.07 top
17357 lyan1     20   0  119m 3296 1568 S  0.0  0.0   0:00.07 bash
17450 lyan1     24   4  311m  61m 8700 S  0.0  0.1   0:02.57 jupyter-noteboo
18045 lyan1     24   4  119m 3328 1544 S  0.0  0.0   0:00.06 bash
```

Commands

- Ps
- Top

Exercise

- Find out the version of OS we are using.
- Find out the process ID of Jupyter Notebook.
- Find out how much virtual and physical memory Jupyter Notebook uses.

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

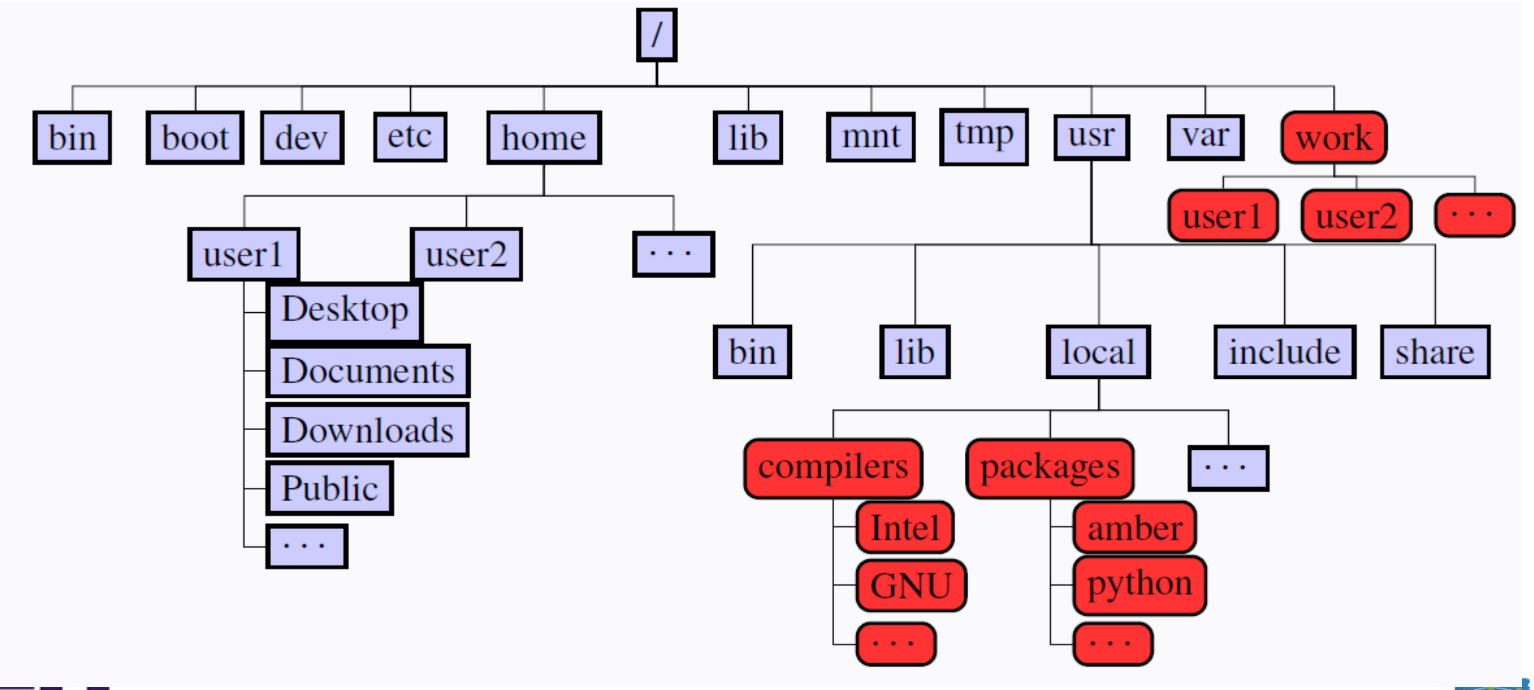
Files

- A sequence of bytes
- An abstraction used by OS to represent all I/O devices
- So that the programmers can develop portable application without knowing details of underlying technology
- Our focus here is the

CENTER FOR COMPUTATION & TECHNOLOGY

File Layout

- Files on disks are organized in an inverted tree structure

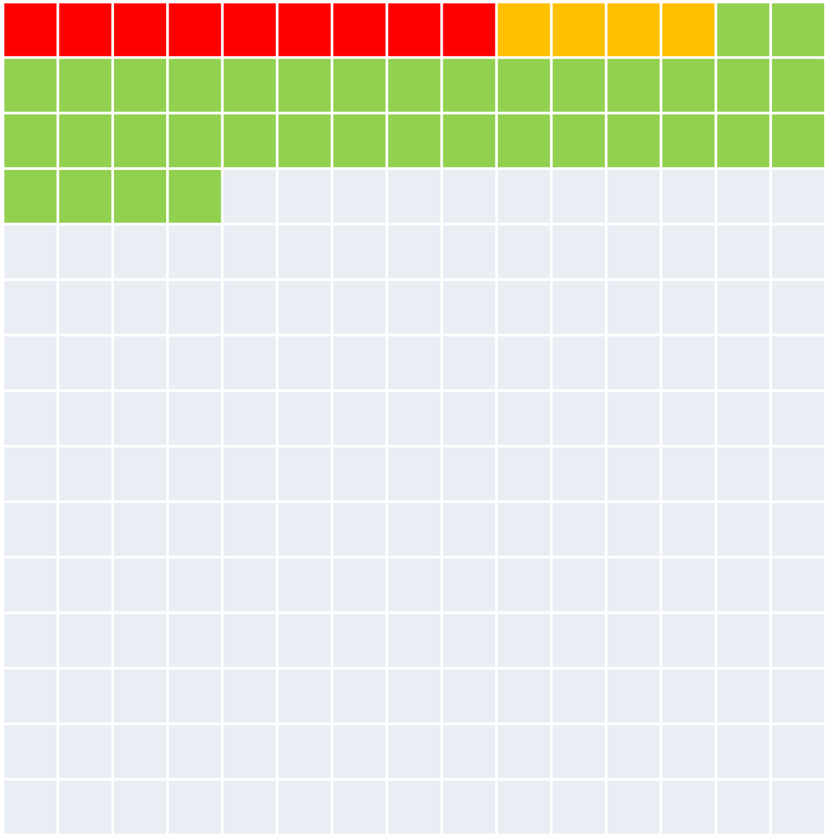


How Files Are Stores on Disks

- Imagine you disk as a huge warehouse

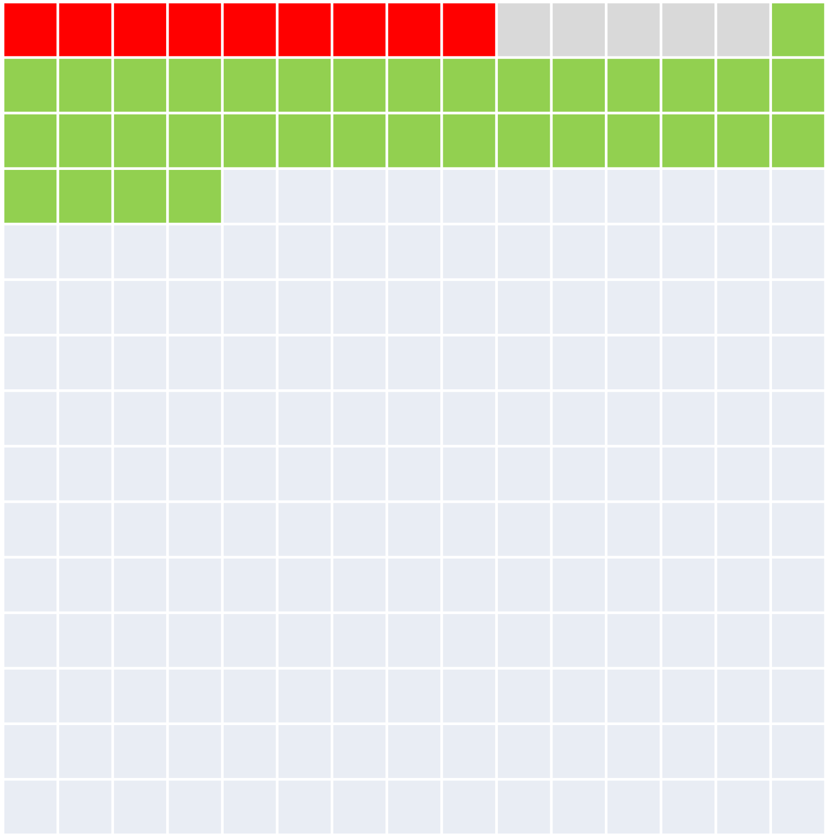
How Files Are Stores on Disks

- Now three files (i.e. sequences of bytes) are written to it



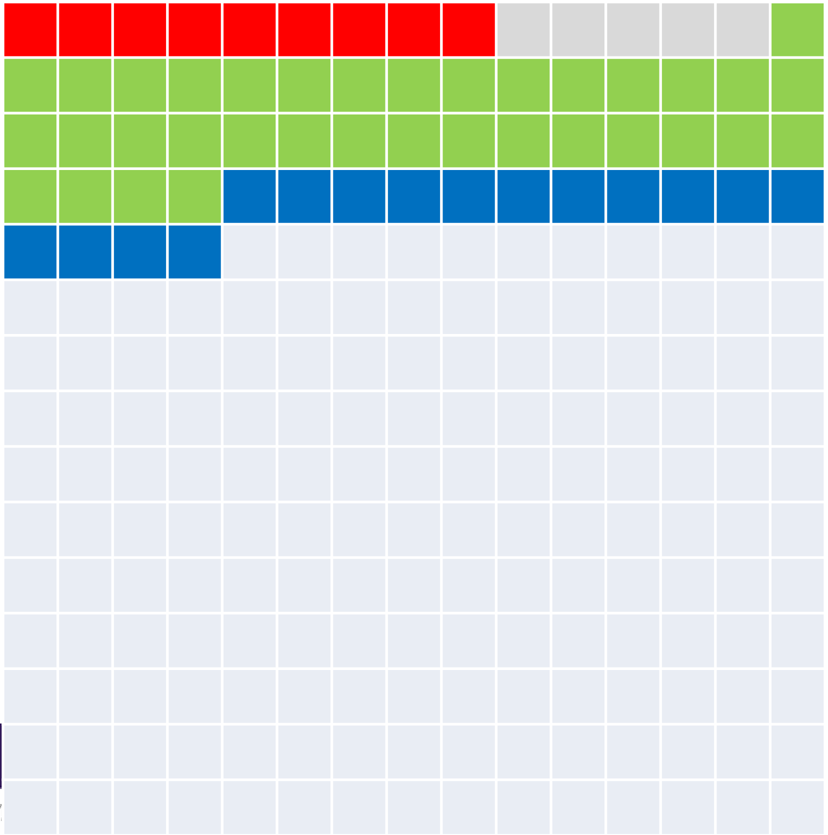
How Files Are Stores on Disks

- Delete the second file



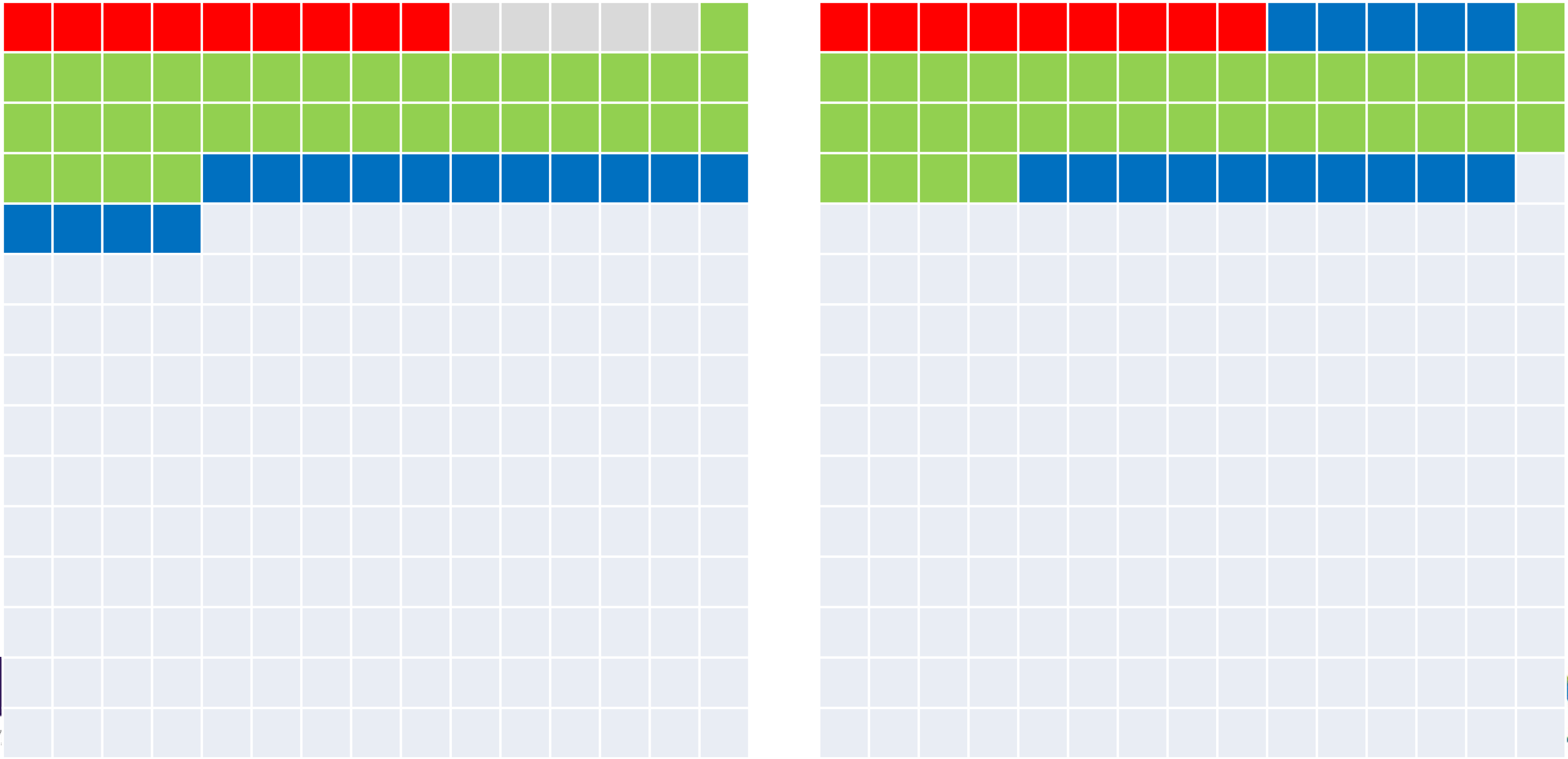
How Files Are Stores on Disks

- Now, what should we do if another sequence of 15 bytes need to be written?



How Files Are Stores on Disks

- Now, what should we do if another sequence of 15 bytes need to be written?



File Systems

- File systems do bookkeeping for files on disks
  - Location, size etc.
- When a file is deleted, the space it occupies does not reset to a “empty” state
  - The file system simply erase the record
- Tradeoff: bookkeeping takes time and resources.
  - Don’t overwhelm the bookkeeper, e.g not a good idea to put 1 million files in the same directory.

CENTER FOR COMPUTATION & TECHNOLOGY

Commands

- Pwd
- Ls
- Rm
- Mv
- Df
- Mount
- du

Exercise

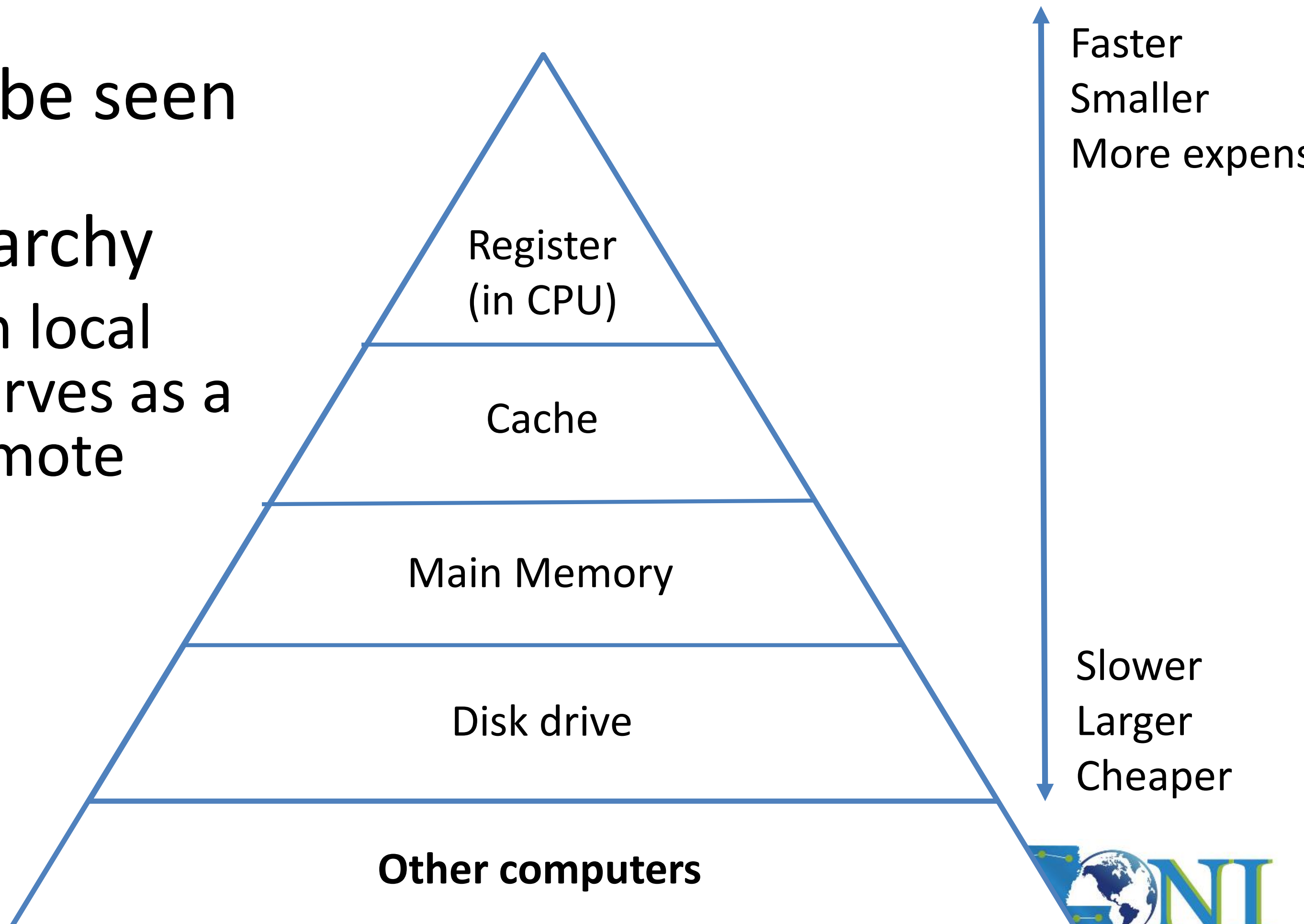
- Find out how much storage is being used by the “Bootcamp2018” directory

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Memory Hierarchy (Updated)

- Network can be seen as part of the memory hierarchy
  - Disk drive on local computer serves as a cache for remote systems

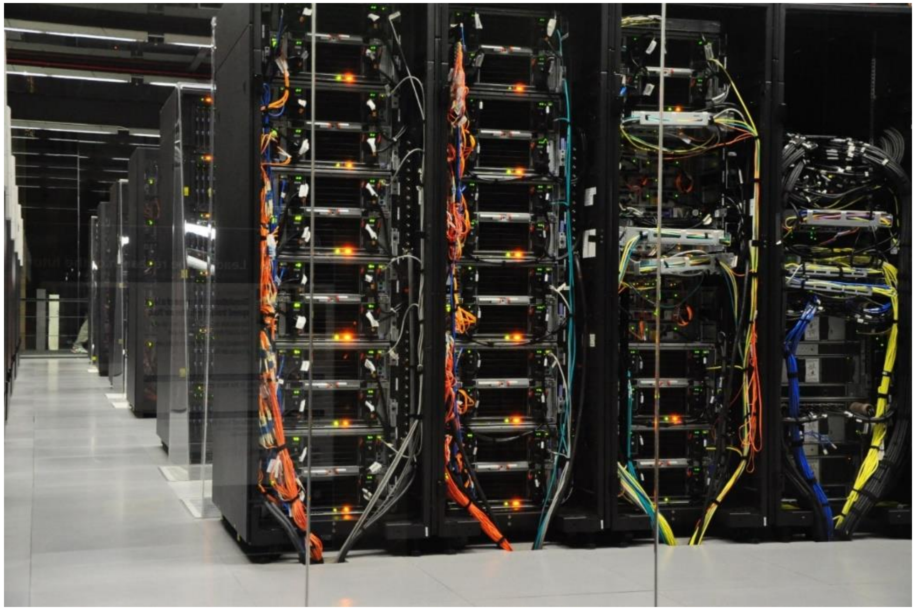


Communication Requirement

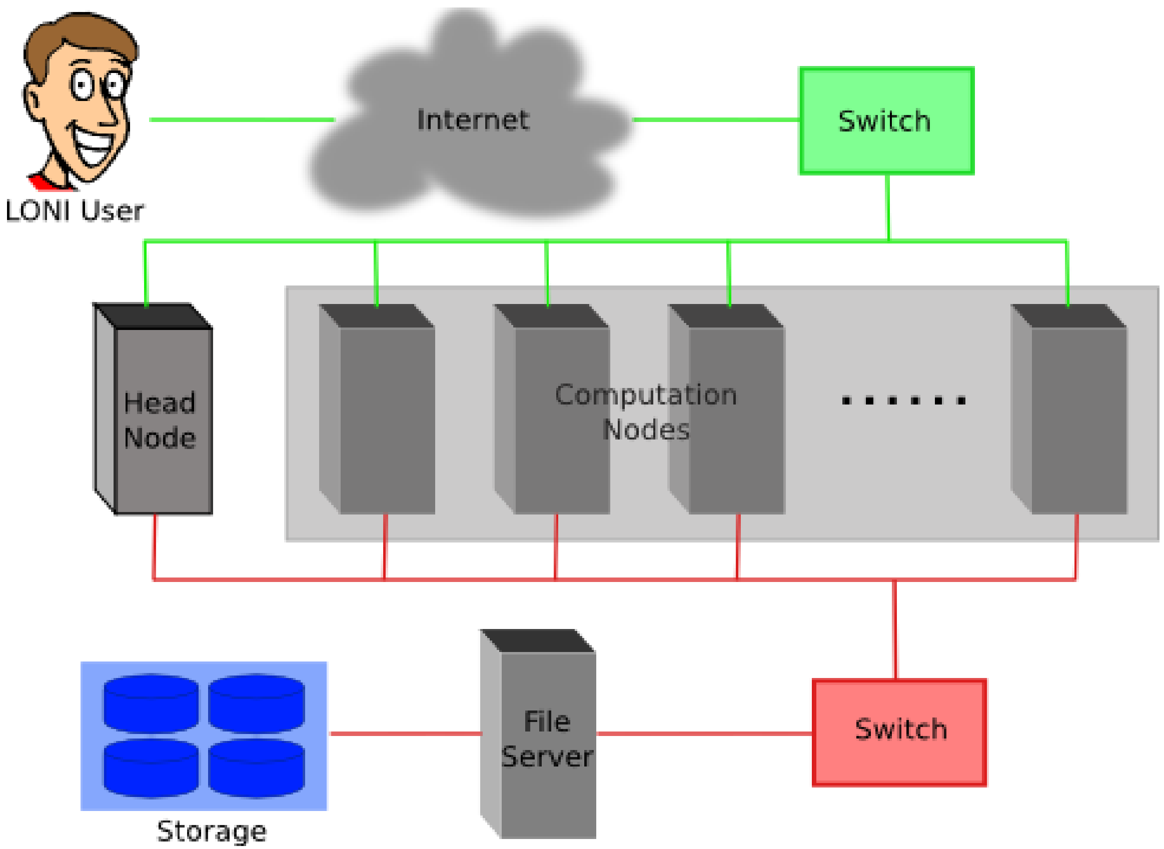
- To deliver a mail to a person, one needs:
  – Building address
  – Room number
  – Name of recipient
- To deliver data to a program, one needs:
  – IP address
  – Port number
  – Name of program

Clusters

- The majority of supercomputers are clusters
  - Individual servers connected by high speed network
  - Servers are known as “nodes” in a cluster
  - They server different purposes
    - Login node
    - Compute node
    - I/O node



"Super Mike 2" Cluster



Commands

- Hostname
- Ipconfig
- Netstat
- nslookup

Exercise

- Find out the host name of the node where your Jupyter Notebook is running and its IP address
- Find out which port Jupyter Notebook is listening to

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

How Do We Improve Performance?

- Better performance = produce more product in unit time
- How to get better performance?

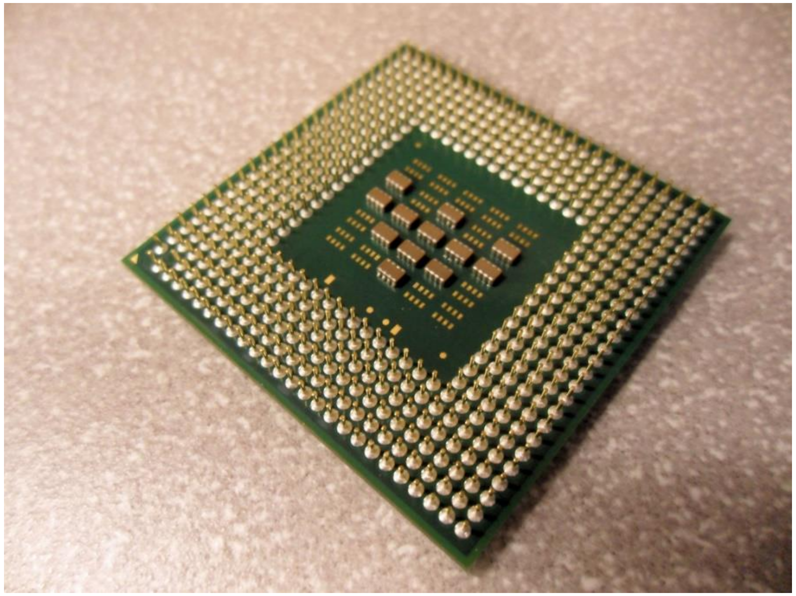
CENTER FOR COMPUTATION & TECHNOLOGY

How Do We Improve Performance?

- Better performance = produce more product in unit time
- How to get better performance?
  - Faster processing
  - More processing units

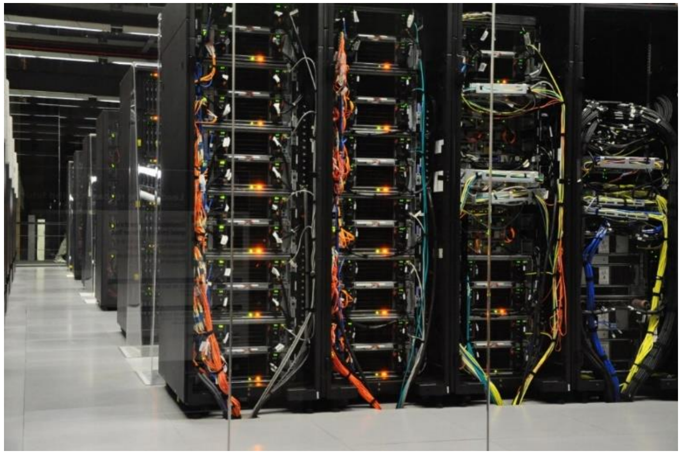
Moore's Law

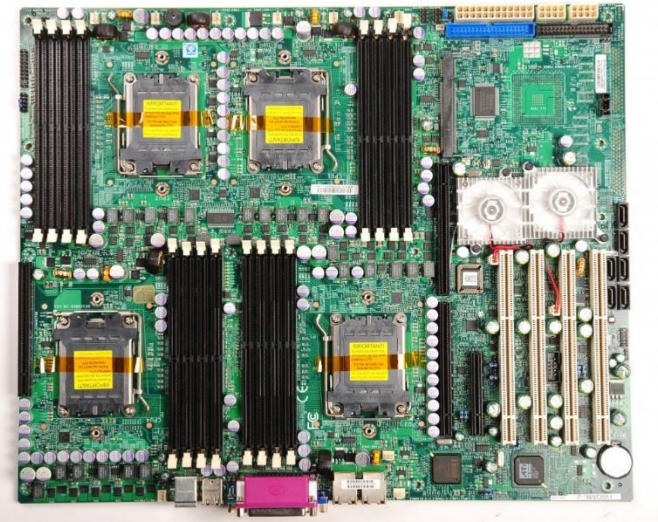
- The fast processing route: given a workshop of fixed size, shrink the size of tools etc. so that the processing power increases.
- Moore's Law
  - Number of transistors double every 18 months
  - Not any more: heat dissipation becomes an inhibitive problem



Parallel Processing

- The more processing units route: build more workshops/plants
- Parallelism/Concurrency
  - Multiple cores
  - Multiple CPU's
  - Multiple nodes





Commands

- Top
- Time
- Pstree

Exercise

- Run the bowtie alignment workflow with 1, 2, 4, 8 and 16 threads and record the timing information

Outline

- Why we are here
- What computers do
- Computer components
  - Data
  - CPU
  - Storage Hierarchy
  - Operating system
  - Files
  - Network
  - Parallel processing
- Putting it together

Putting It Together

- Performance expectation
- Troubleshooting