

# Distributed Computing

## An Introduction

# Profile



---

Name: Mahmoud Badry

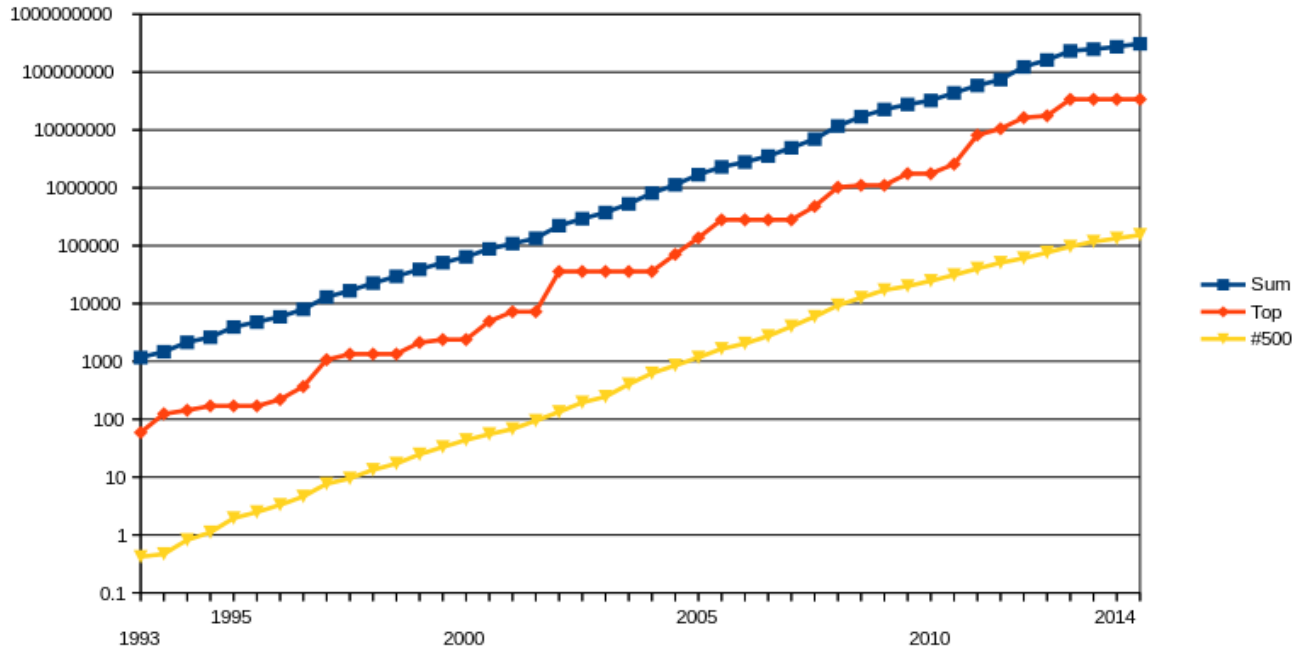
Tel.: 01090138058

E-Mail: [mmal8@fayoum.edu.eg](mailto:mmal8@fayoum.edu.eg)

Office hours: Wednesday and Thursday

Lab Github: [https://github.com/mbadry1/FCIFayoum\\_Distributed-computing\\_2018](https://github.com/mbadry1/FCIFayoum_Distributed-computing_2018)

# Super computers History



The logarithmic y-axis shows performance in GFLOPS.

Maximum is near Exa-bytes.

FLOP: float point operation.

Kbyte:  $10^3$  byte

Mega:  $10^6$  byte

Giga:  $10^9$  byte

Tera:  $10^{12}$  byte

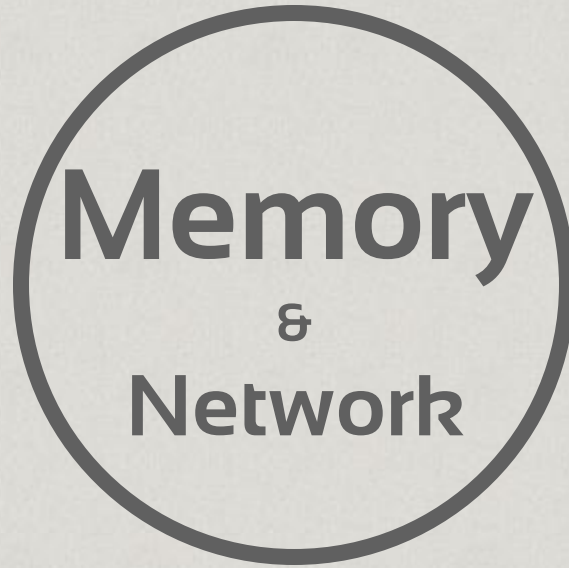
Peta:  $10^{15}$  byte

Exa:  $10^{18}$  byte

Source: Wiki

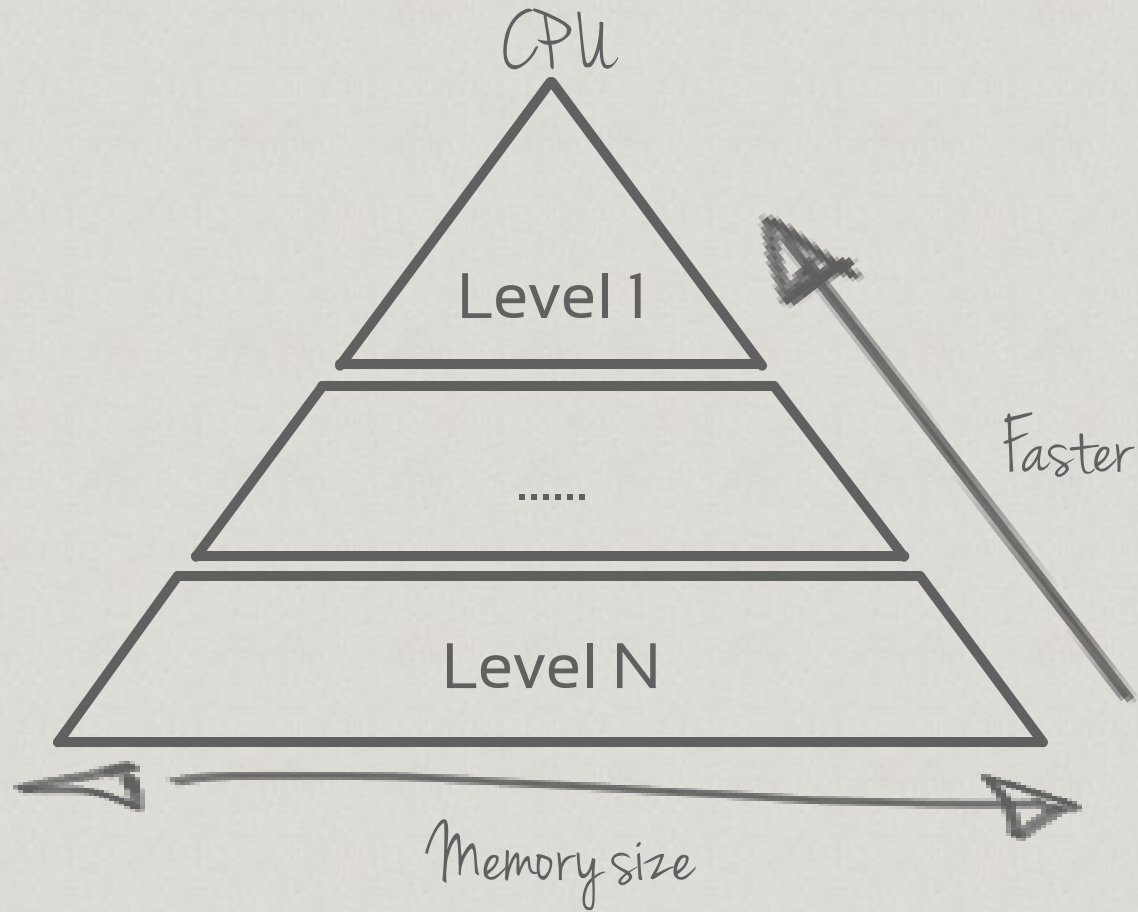
# AGENDA



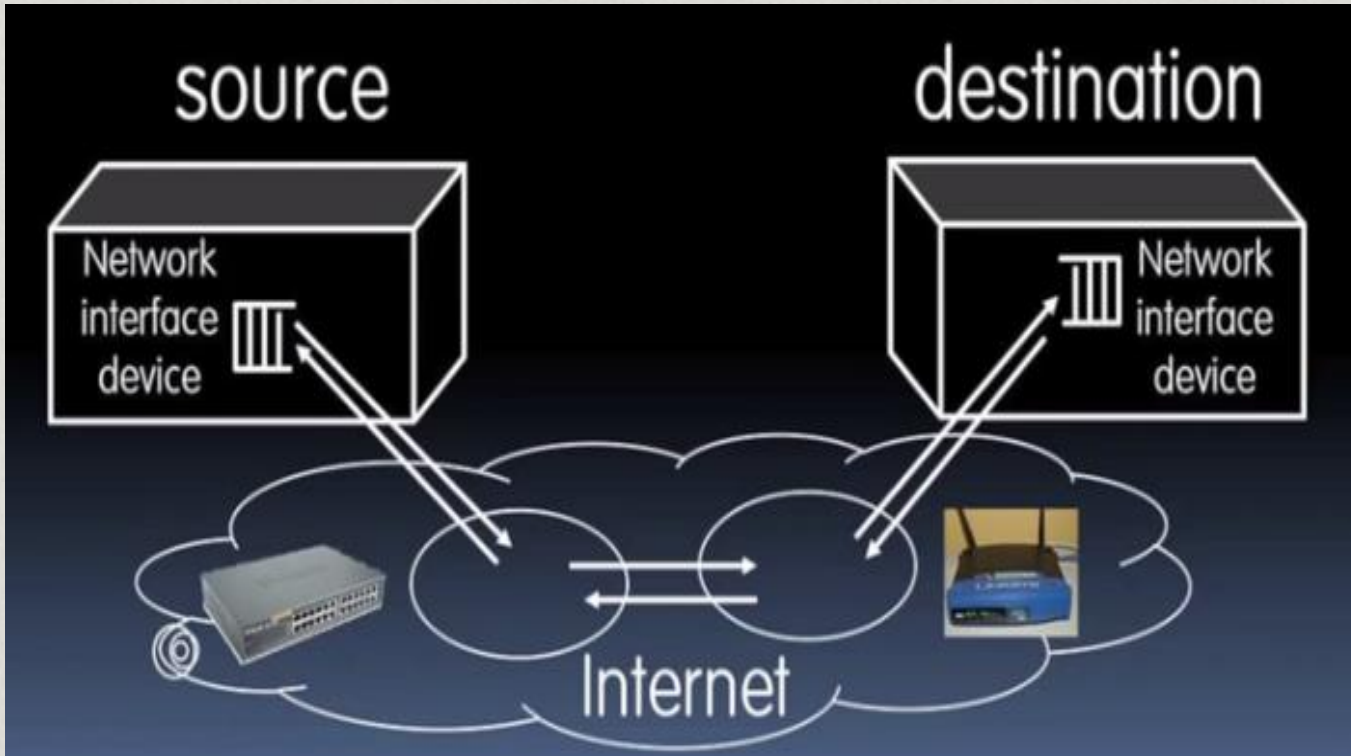


Basics

# Memory



# Computer Networks

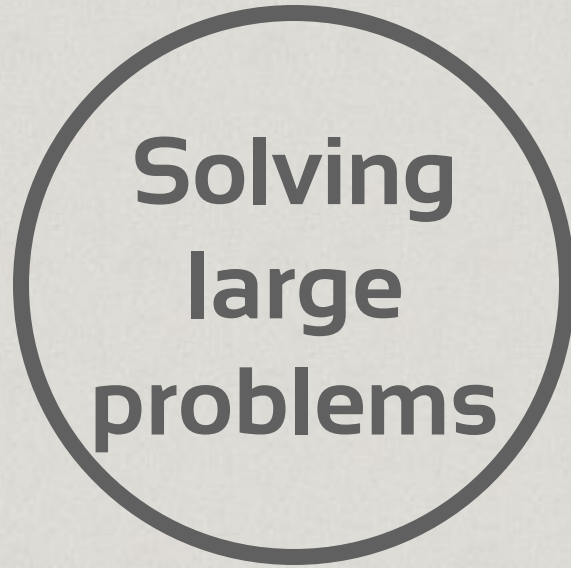


Source: UC Berkeley youtube video

Message is been delivered using some network devices from source to destination using some internet protocols.

Internet Speed is growing rabidly!

Ethernet can now transfer 10 gb per second compared to dial up that was 56 kb per second



## Two Approaches

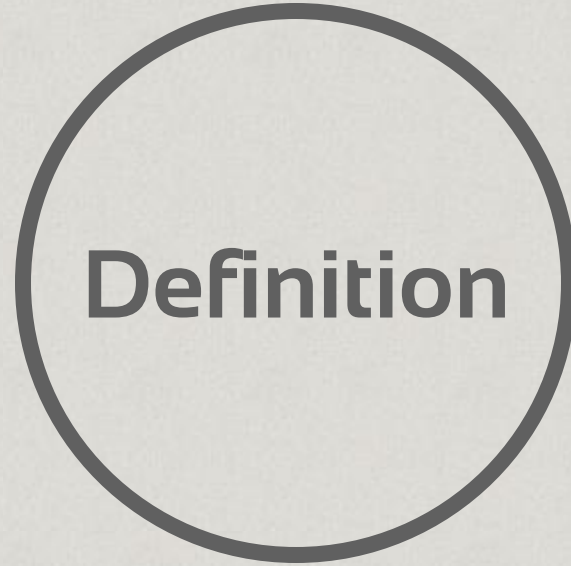


# Large Problems Solving

- Using super computers.
  - Multiple processors in one box/room from one vendor that has a shared memory.

# Large Problems Solving

- Using Distributed computing.
  - Many separate computers (each with independent CPU, RAM, HD, NIC) that communicate through a network.
    - Grids (heterogenous computers across Internet)
    - Clusters (mostly homogeneous computers all in one room)



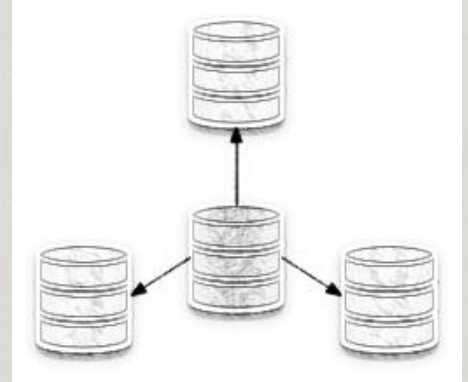
DC definition

# What is Distributed computing?

- Distributed computing is a field of computer science that studies **distributed** systems. A **distributed** system is a software system in which components located on networked **computers** communicate and coordinate their actions by passing messages.

# DC Agent types

- Dispatcher "Master" : gives jobs & collects results.
- Worker "Slave" : (get, process, return) until done





# DC Challenges

# Heterogeneity

- A system consisting of multiple distinct components.
- Usually the solution is a middleware.

# Scalability

- As the system, number of resources, or users increase the performance of the system is not lost and remains effective in accomplishing its goals.



# Fault handling

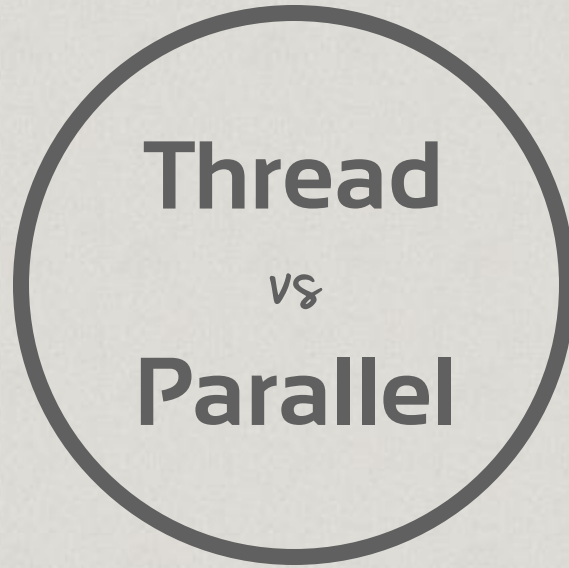
- Detect Failures.
- Recover from failures.
- Build Redundancy.

# Need to parallelize algorithms, data structures

- Must look at problems from parallel standpoint
- Best for problems whose compute times  $\gg$  overhead.

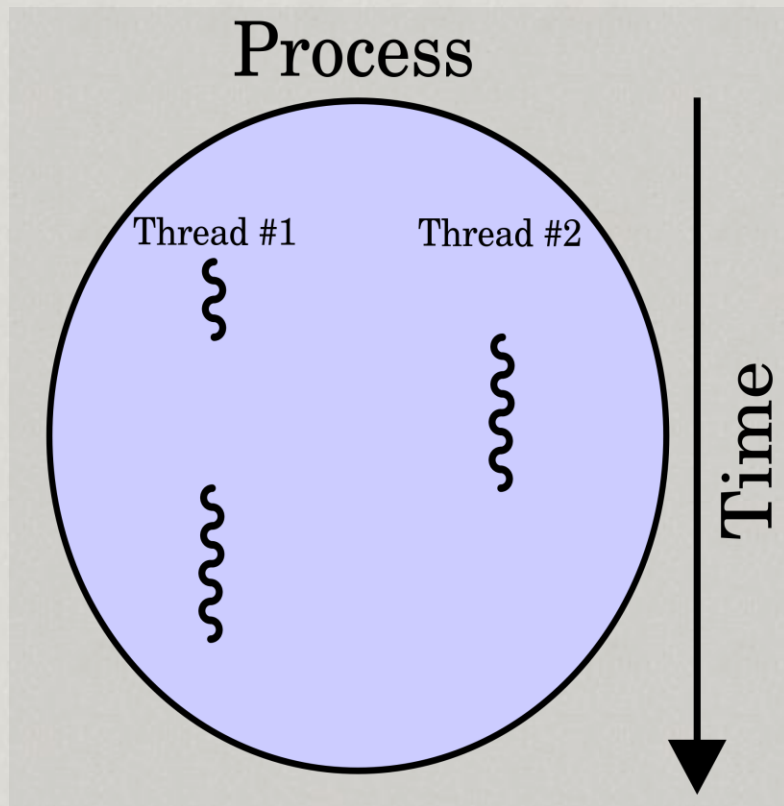
# Inefficiencies

- The inefficiencies resulted from waiting, overhead, etc.



On Shared memory

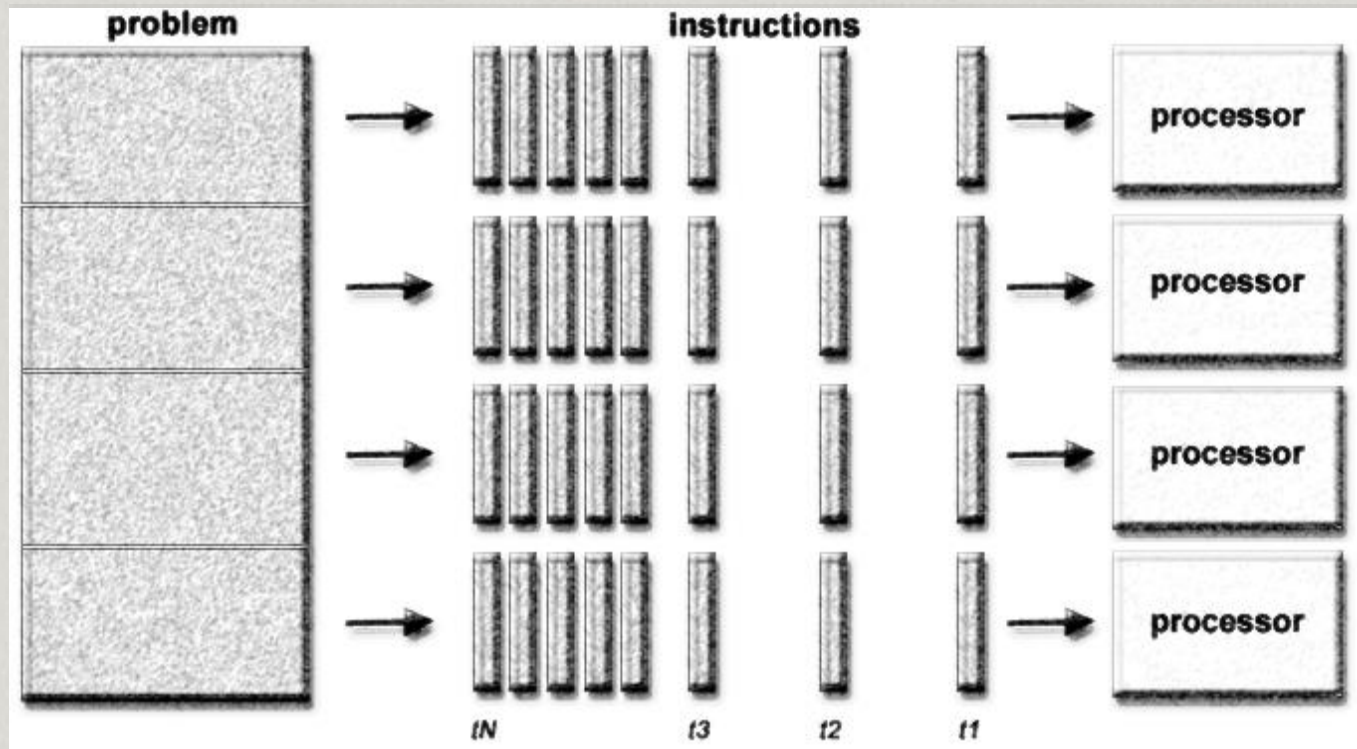
# Multi threading



The traditional multi-threading used to do time-slicing or take advantage of the CPU idle time, which is that while one of the threads in your program was waiting another thread could execute.

Hint: Java Thread object does not guarantee that you will optimally use all the cores available in your computer.

# Parallel Programming



Parallel programming explicitly breaks the task down into smallest unit of execution, where each unit can be executed in parallel on a single CPU core.

Parallel programming can be done automatically or manually.



QUESTIONS

# THANK YOU!

