Технології графічного процесінгу

(Масивно-паралельні обчислення на графічних прискорювачах
...
Massively Parallel Computing on Graphic Processing Units - GPUs)

Lecture 1. Introduction

Yuri G. Gordienko

(Гордієнко Юрій Григорович)

NTUU-KPI, 2021

yuri.gordienko@gmail.com

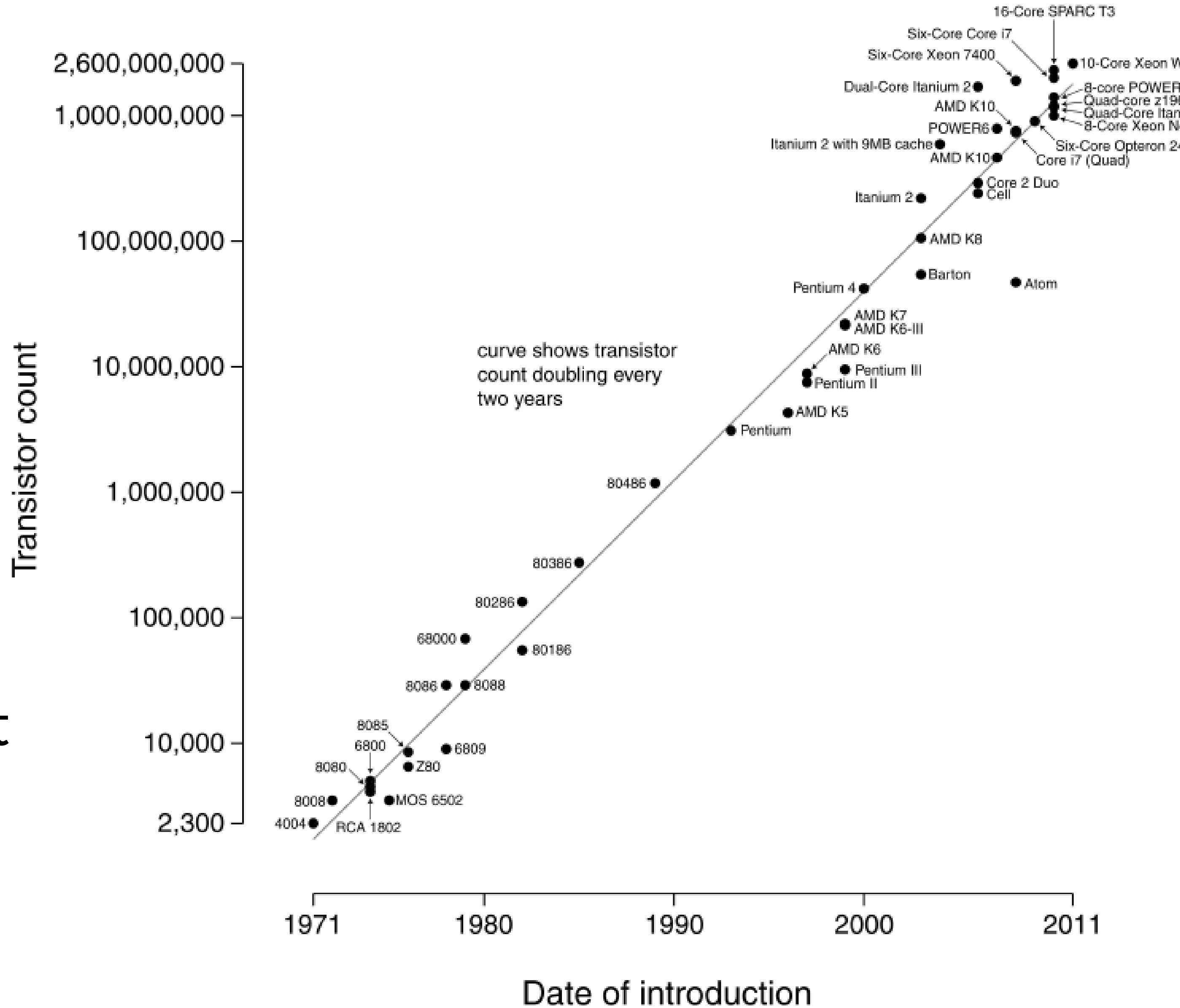
Why we need other computing technologies?

- Multiprocessor PC is not enough
- Example: movie rendering
  - “Disney’s Cars 2” (2011) ~11-90 hours to render each frame;
  - “Monsters University” (2013) ~29 hours/frame
    - Total time: over 100 million CPU hours
    - 3000-5000 AMD processors; 10 Gbps networks
- Example: Google search

~5.1 billion queries per day; index >50 billion web pages; hundreds of thousands of servers to do this

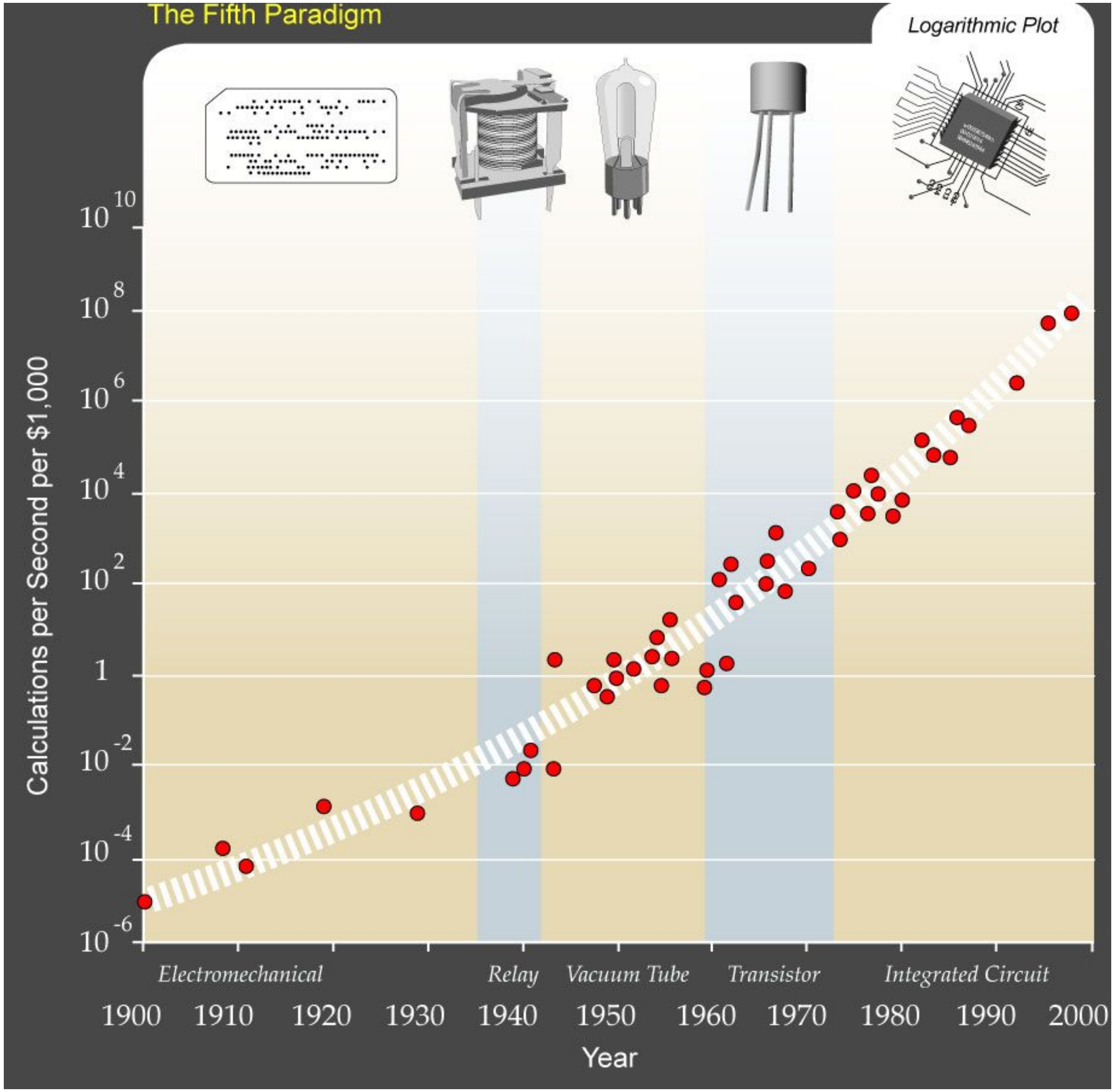
State of the Art – CPU

Moore’ Law: CPU transistors versus dates of introduction. The line corresponds to exponential growth with transistor count **doubling** every two years.

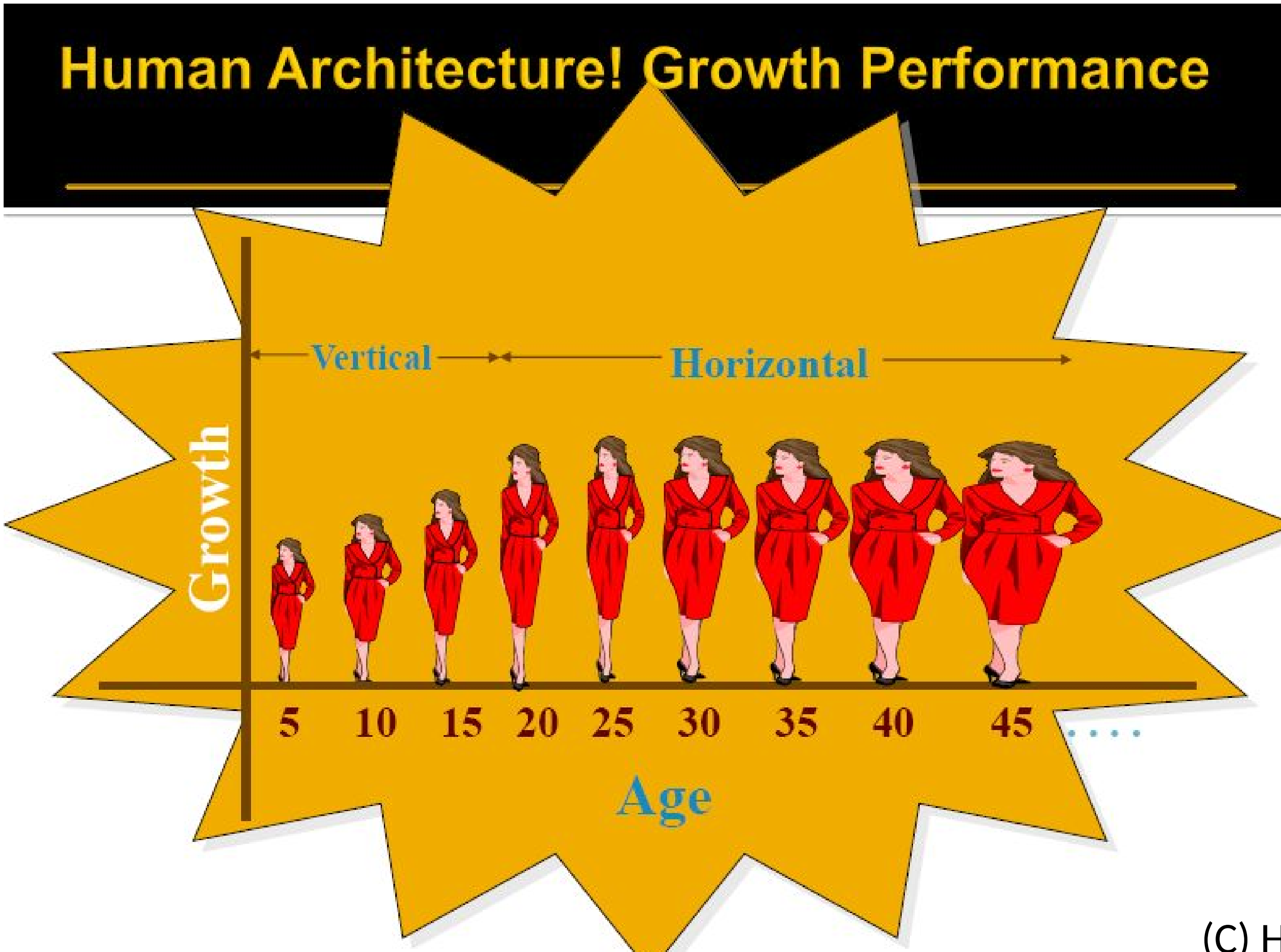


State of the Art – High-tech

Kurzweil's extension of Moore's law: **calculations per second versus time** - from integrated circuits to earlier transistors, vacuum tubes, relays and electromechanical computers.

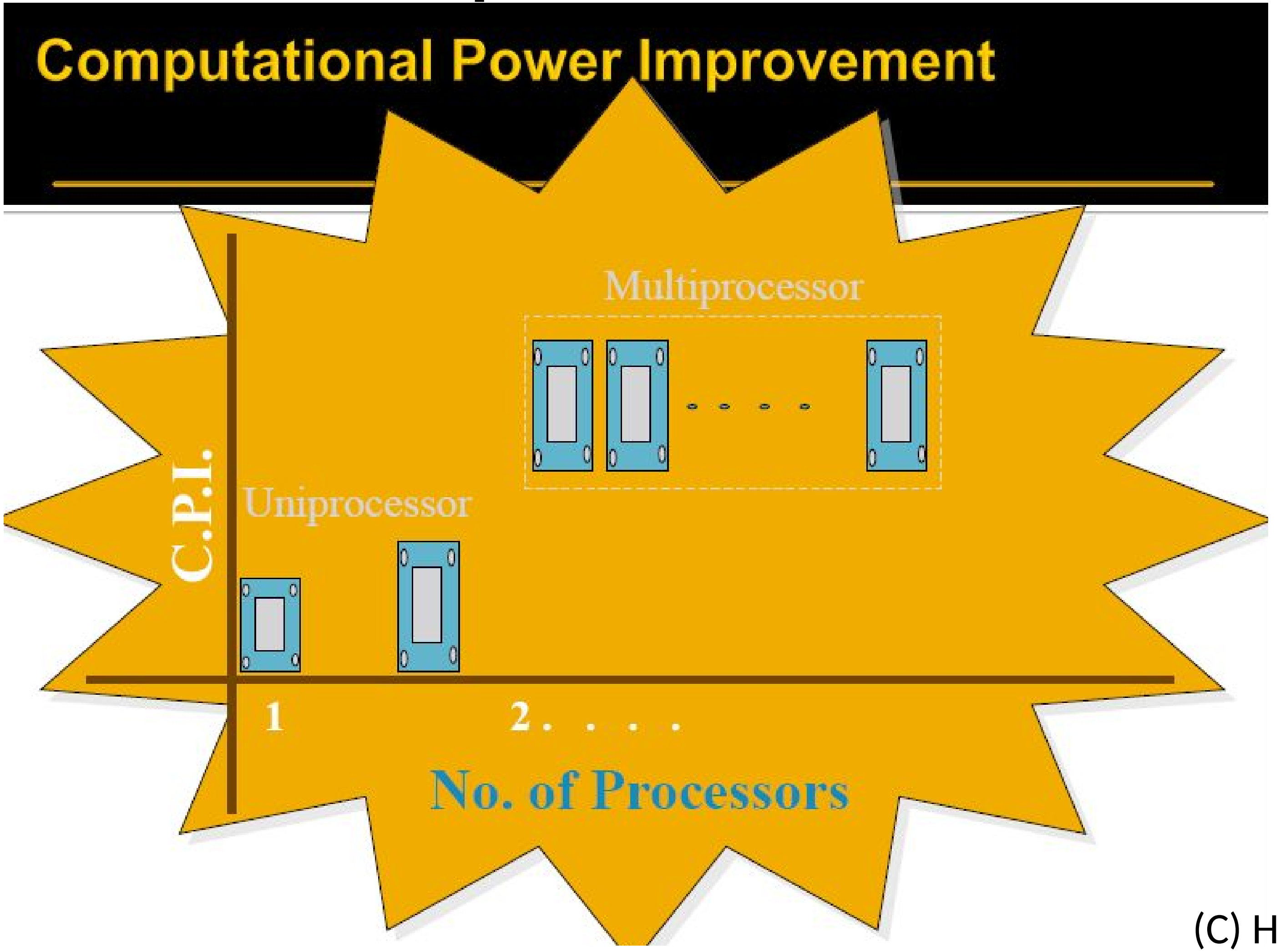


How to increase the computing power?



(C) Hamad

How to increase the computing power?



(C) Hamad

Other Computing Modes - Illustration

Mythbusters:

- Adam
- Jamie

Vivid presentation on GPU-principle at NVIDIA conference (2008)

Distributed Computing - Definition

What is Distributed Computing?

“A collection of independent computers that appears to its users as a single coherent system” (C) Tannenbaum, van Steen

- … are networked together
- … appear to the user as a one computer
- … work together to achieve a common goal

Distributed Computing – "Alternative" Definition

What is Distributed Computing?

You know you have a distributed system when the crash of a computer you've never heard of stops you from getting any work done.

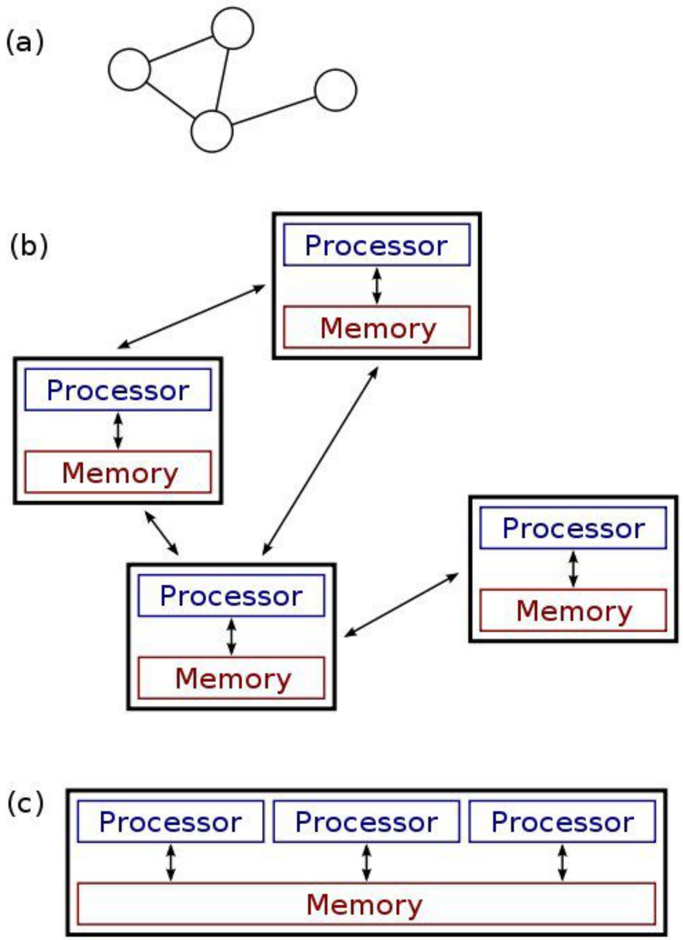
– *Leslie Lamport*

Distributed and Parallel Computing

- What is the difference?

Various aspects (points of view):

- **Connectivity**
- **Memory**
- **Granularity**

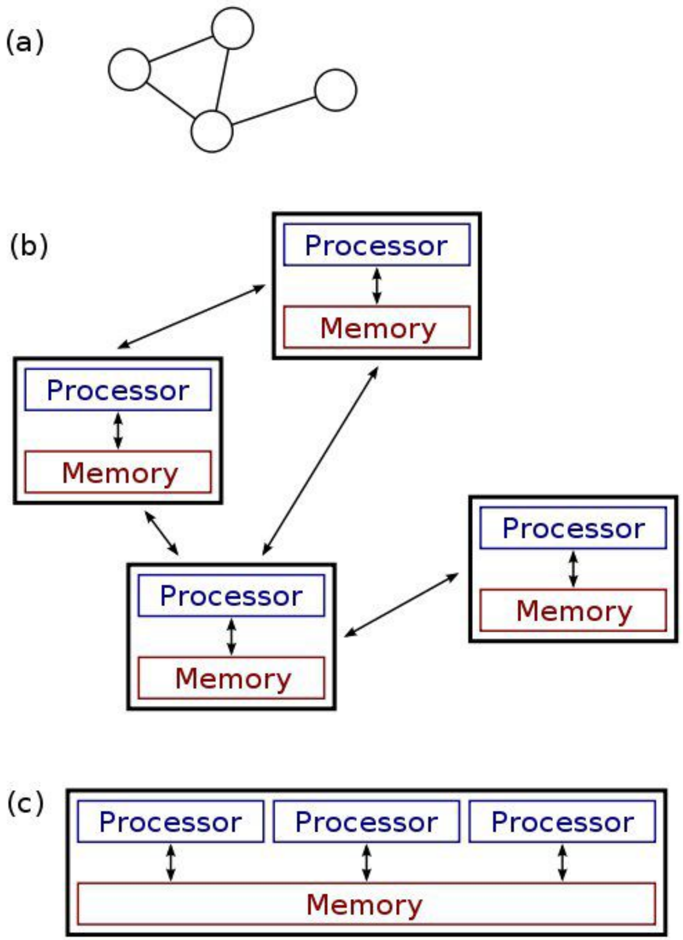


Distributed and Parallel Computing

- What is the difference?

<u>Connectivity:</u>

- Parallel System: a **tightly-coupled form of distributed computing**
- Distributed System: a **loosely-coupled form of parallel computing**

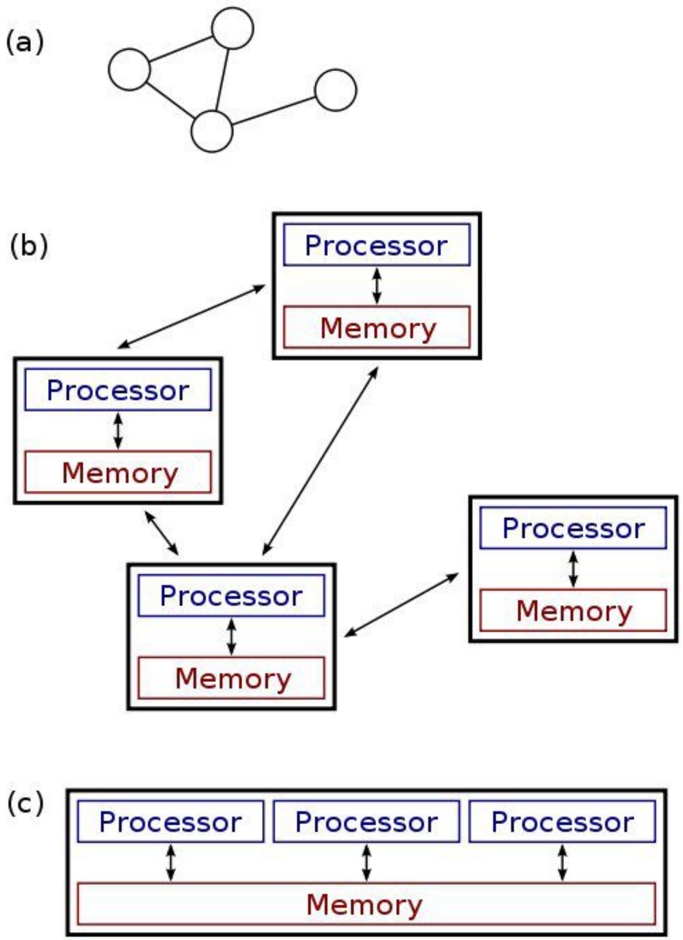


Distributed and Parallel Computing

- What is the difference?

<u>Memory:</u>

- Parallel System: processors access a **shared memory to exchange information**
- Distributed System: uses a "**distributed memory**". **Message passing is used to exchange information between the processors as each one has its own private memory.**



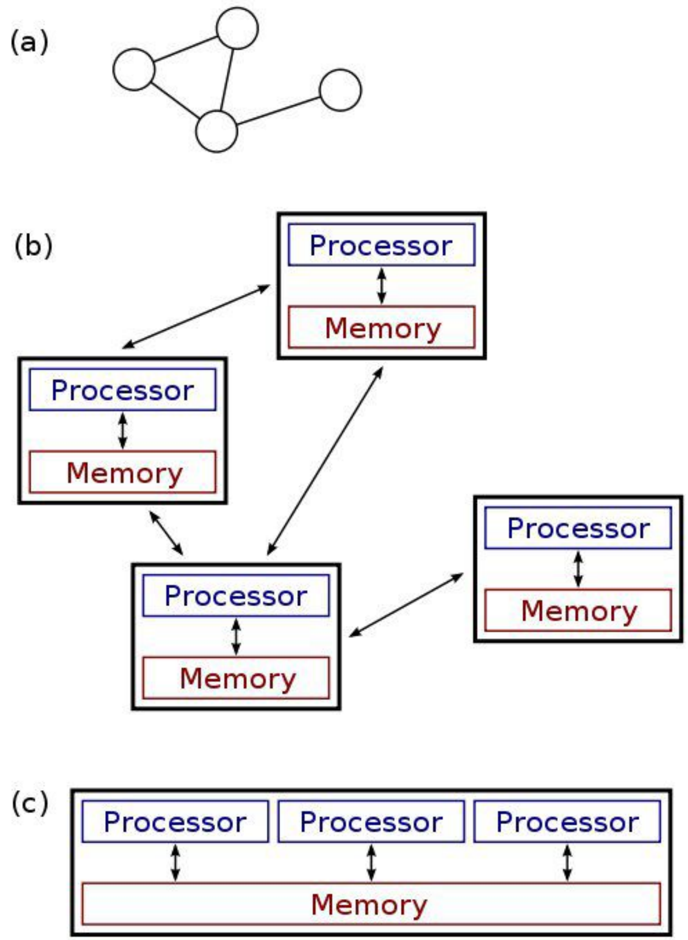
Distributed and Parallel Computing

- What is the difference?

<u>Granularity (Heterogeneity):</u>

- Distributed System: a more **coarse-grained form of parallel computing**

- Parallel System: a **finer-grained form of distributed computing**



Distributed Computing - Applications

- **Strategic Systems (Defence / Intelligence)**
- **Visualization and Graphics**
- **Economics and Finance**
- **Scientific Computing**
  - **Physics (LHC – Higgs boson!)**
  - **Bioinformatics (protein docking)**
  - **Geology (seismography)**
  - **Astronomy (simulation of galaxies)**

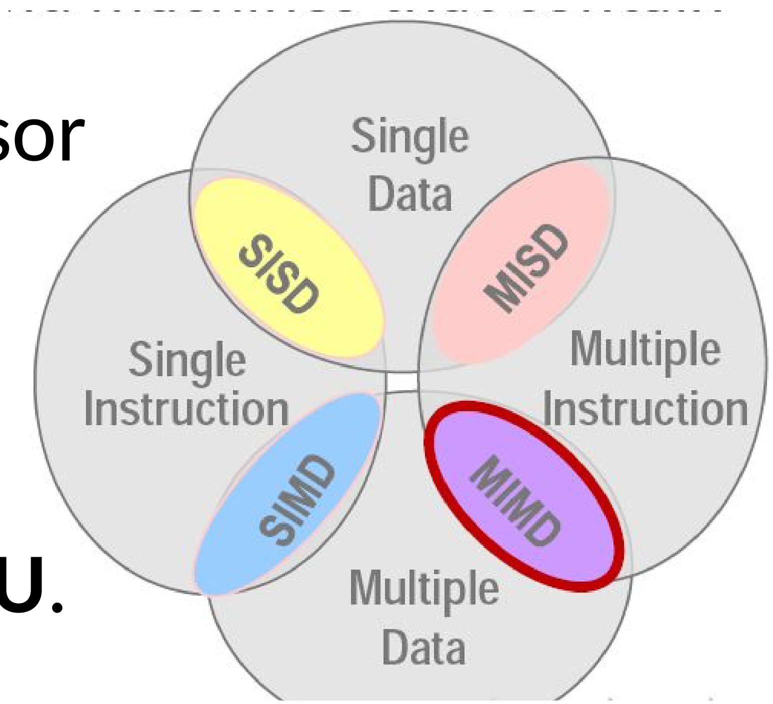
Distributed Computing - Models

- **Architectural Models**
- **Interaction Models**
- **Fault Models**

Distributed Computing – Architectural Models

Flynn's Taxonomy:

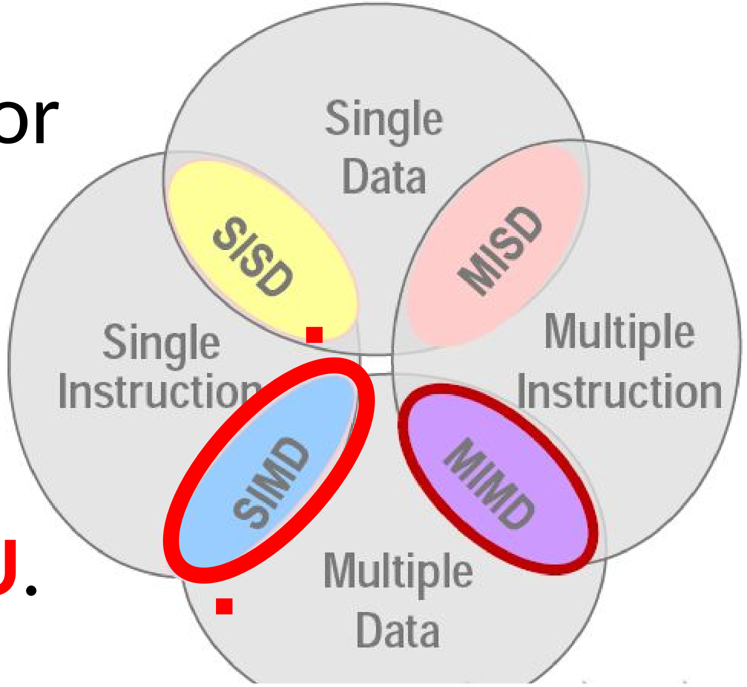
- **SISD:** traditional uniprocessor computers
- **MISD:** Space Shuttle flight control computer
- **SIMD:** array processor, **GPU**.
- **MIMD:** parallel systems, **distributed systems**.



Distributed Computing – Architectural Models

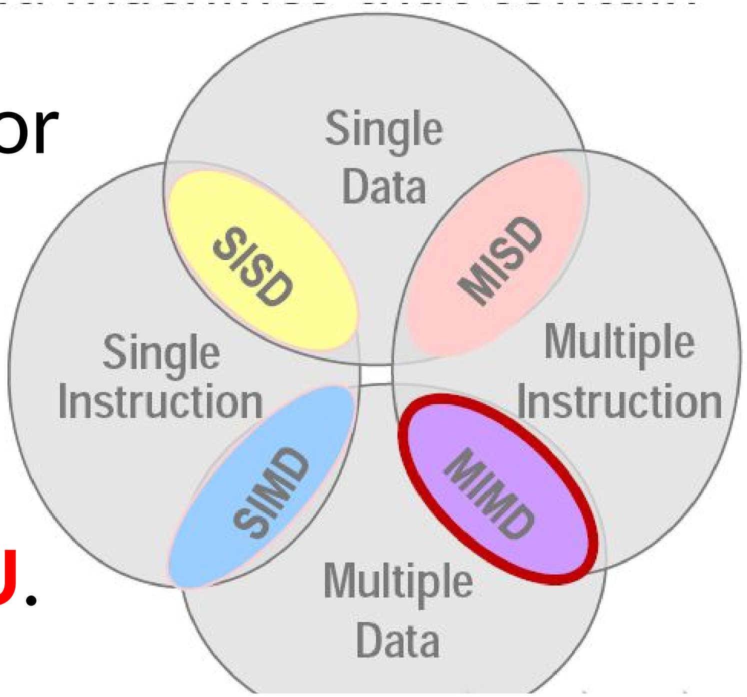
Flynn's Taxonomy:

- **SISD:** traditional uniprocessor computers
- **MISD:** Space Shuttle flight control computer
- **SIMD:** array processor, **GPU**.
- **MIMD:** parallel systems, **distributed systems**.



Distributed Computing – Architectural Models

Flynn's Taxonomy:



- **SISD:** traditional uniprocessor computers
- **MISD:** Space Shuttle flight control computer
- **SIMD:** array processor, **GPU**.
- **MIMD**: parallel systems, **distributed systems,** **multi-GPU systems**.

Distributed Computing – Architectural-Service Models

- **Centralized (highly-coupled, cluster computing):** mainframe, cluster, **GPU**
- **Client-server:** mail, banking, computations
- **Multi-tier :** grid, DNS
- **Peer-to-peer:** file exchange, computations

Distributed Systems – Interaction Models

Crucial questions:

- How do we handle time?
- Are there time limits on process execution, message delivery, and clock drifts?

- **Synchronous** distributed systems

- **Asynchronous** distributed systems

Distributed Systems – Fault Models

Crucial question: what kind of faults can occur?

- Omission faults:

  A processor or communication channel fails to perform actions it is supposed to do.

- Timing faults (in synchronous distributed systems):

  If any of this time limits is exceeded.

- Arbitrary faults (the most general and worst):

  Intended processing steps or communications are omitted or/and unintended ones are executed.

Distributed Computing - Advantages

- Performance
- Distribution – NOT for GPU
- Reliability (fault tolerance) – NOT for GPU
- Incremental growth (scalability) – BUT…
- Sharing (computation/data/resources/) – NOT for GPU
- Communication – NOT for GPU
- Economics (green computing)
- Flexibility – NOT for GPU

Distributed Computing – Disadvantages

- **Heterogeneity (hardware, software, operation, human factor) – NOT for GPU**
- **Software development**
- **Networking – NOT for GPU, except for multi-GPU**
- **Security – NOT for GPU**
- **Incremental growth (scalability)**
- **Price**

Distributed Computing - **Pitfalls**

- The network is NOT reliable **– NOT for GPU**.
- The network is NOT secure **– NOT for GPU**.
- The network is NOT homogeneous **– NOT for GPU**.
- The topology is NOT constant **– NOT for GPU**.
- **Latency is NOT zero.**
- **Bandwidth is NOT infinite.**
- **Transport cost is NOT zero.**
- There is NO single administrator **– NOT for GPU.**

Distributed Computing - Design

The main characteristics:

- **<u>Transparency</u>**
- **<u>Scalability</u>**
- **<u>Performance Predictability</u>**
- **<u>Heterogeneity</u>**
- **Fault-tolerance**
- **High availability**
- **Recoverability**
- **Security**

Distributed Computing - Transparency

How to make impression that the collection of machines is a "simple“ single computer?

- Access
- Location
- Migration
- Replication
- Concurrency
- Failure
- Performance

Distributed Computing - Scalability

The system should remain efficient even with a significant increase in the number of users and resources connected:

- cost of adding resources should be reasonable;
- performance loss with increased number of users and resources should be controlled;
- software resources should not run out (number of bits allocated to addresses, number of entries in tables, etc.)

Distributed Computing - Performance

How to predict/control performance?

- The performance of individual workstations.
- The speed of the communication infrastructure.
- Extent of reliability (fault tolerance) (replication and preservation of coherence imply large overheads).
- Flexibility in workload allocation: for example, idle processors (workstations) could be allocated automatically to a user's task.

Distributed Computing - Heterogeneity

- **different hardware**: mainframes, workstations, PCs, servers, etc. – **multi-GPU**;
- different software: UNIX, Windows, OS/2, iOS, Android, Tizen, Real-time OSs, etc. – **NOT for GPU**;
- various devices: PCs, mobiles, ATM-machines, telephone switches, robots, sensors, etc. – **NOT for GPU**;
- diverse networks and protocols: Ethernet, FDDI, ATM, TCP/IP, Novell Netware, etc. – **NOT for GPU**

Types of Distributed Computing Systems

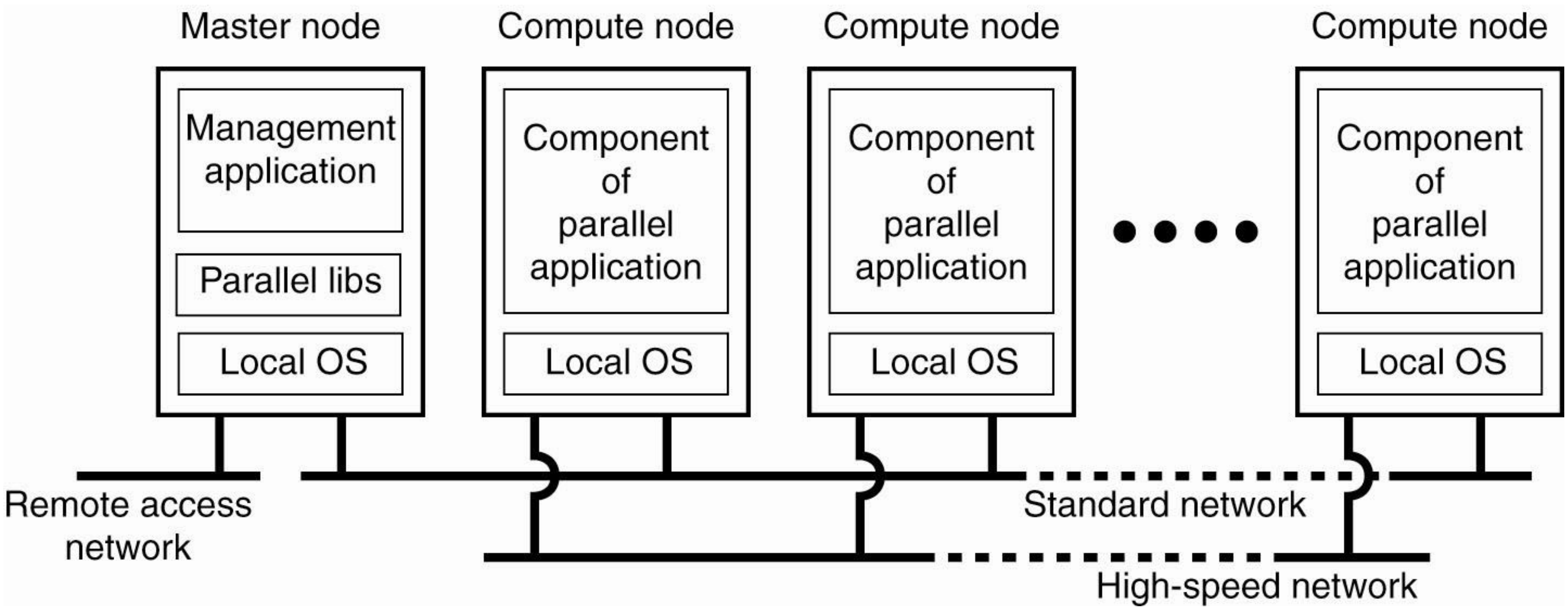
Cluster Computing - Definition

What is Cluster Computing?

Collection of high-end computers usually closely connected through a LAN

- **Homogeneous:** OS, hardware
- **Work:** together like a single computer
- **Applications are hosted on one machine and user machines connect to it. Clients connect via terminals**
- **Applications:** storage, calculations.

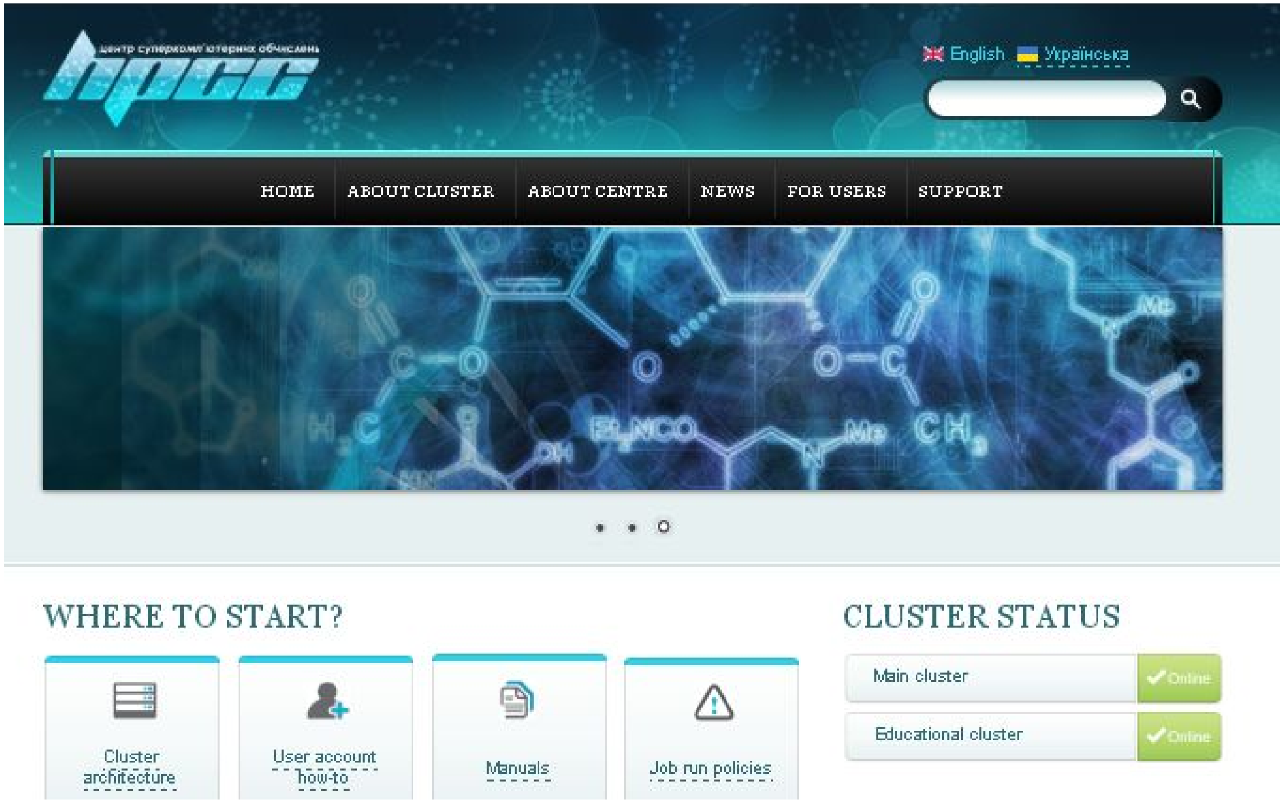
Cluster Computing - Scheme



Note: Scaling is not easy. Multiple entities competing for the same resource

(C) Tannenbaum, van Steen

Cluster Computing - Examples



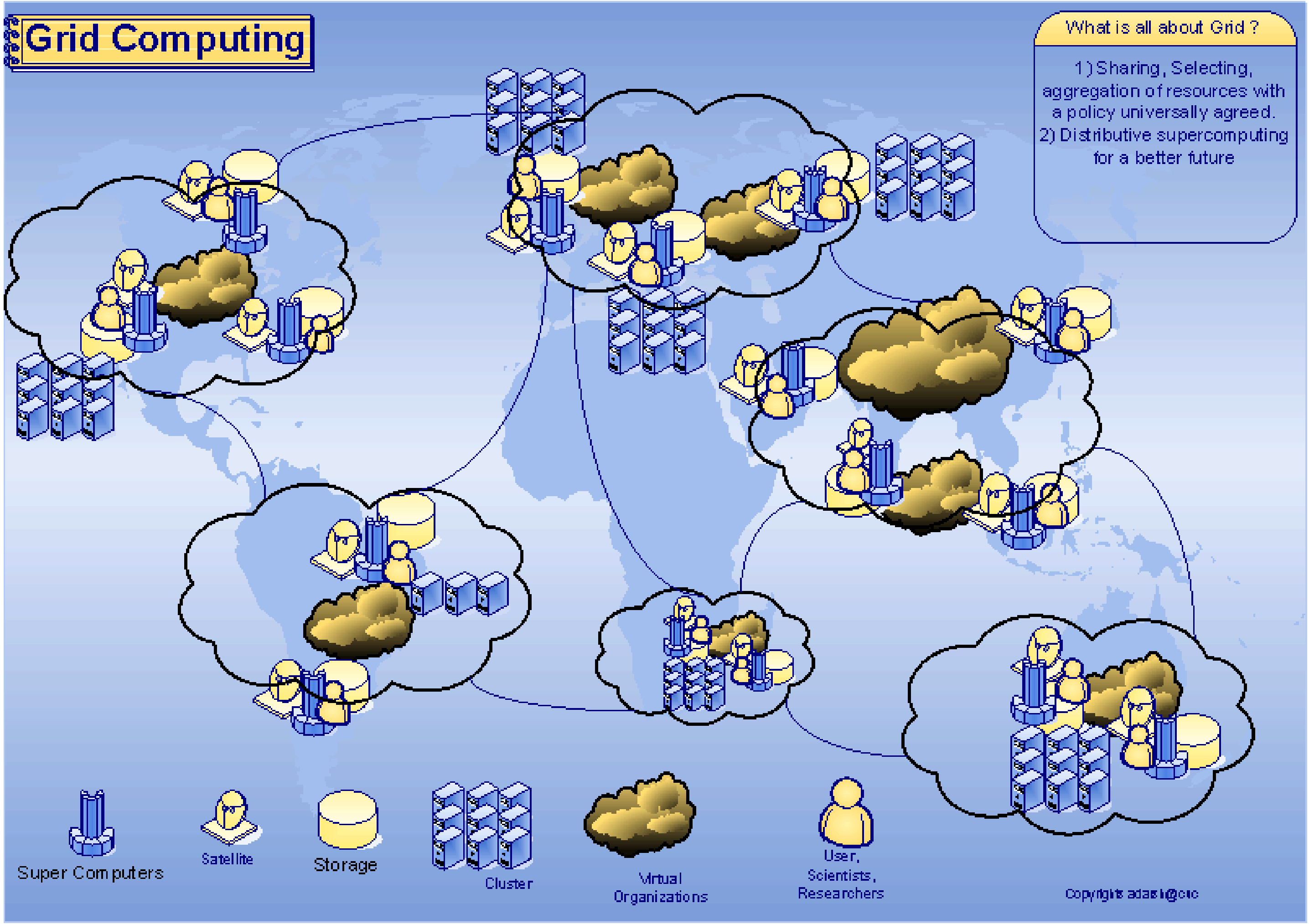
Cluster of the High Performance Computing Centre (http://http://hpcc.kpi.ua/)

Grid Computing - Definition

Collection of clusters, which may be combined in a "Grid“ of a massive computing power.

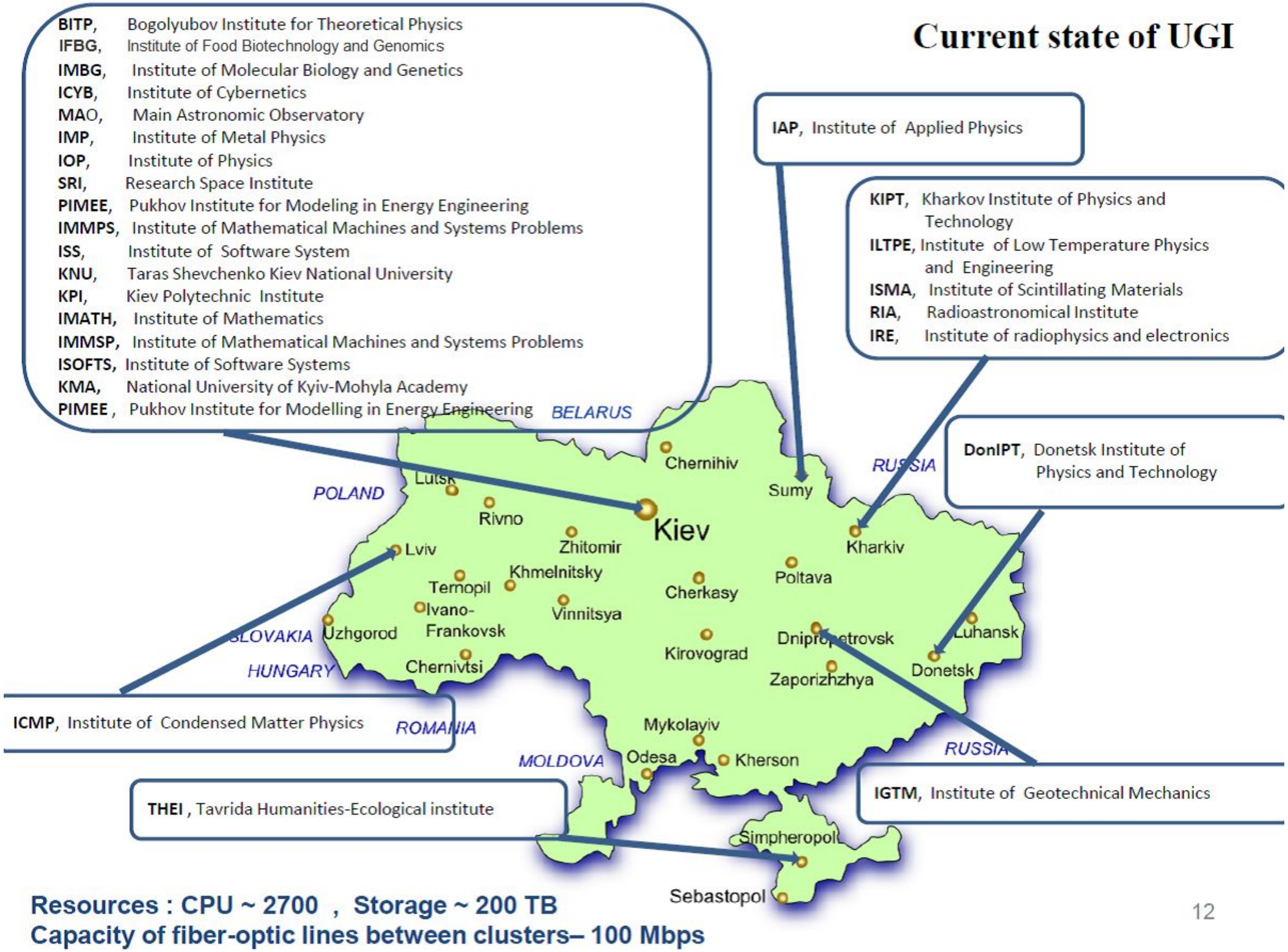
- **Heterogeneous**: systems differ in hardware/software/ administrative domains and deployed network technologies
- **Work**: for collaborations grids use virtual organizations.
- **Applications:** storage and calculations in science, finance government, manugfacture.

Grid Computing - Scheme



Purpose: Computational Grid, Data Grid, Collaboration Grid

Grid Computing - Examples



Ukrainian National grid (http://ung.in.ua)

GPU Computing - Definition

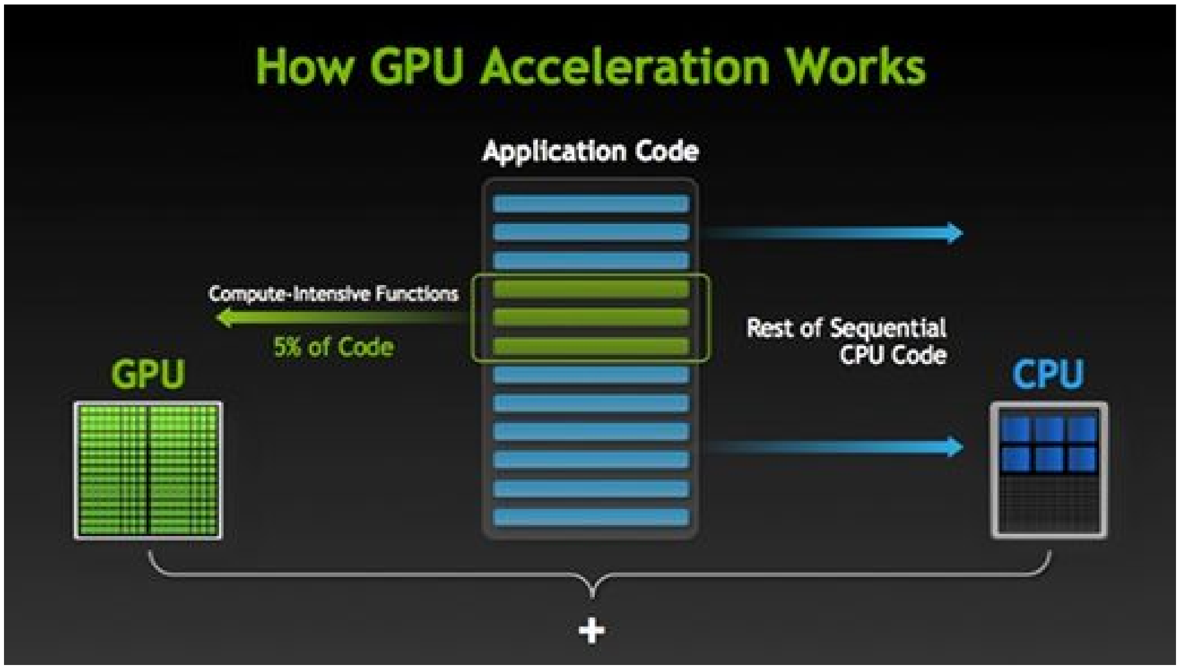
What is GPU Computing?

General-purpose computing on graphics processing units (**GPGPU** or **GPU**)

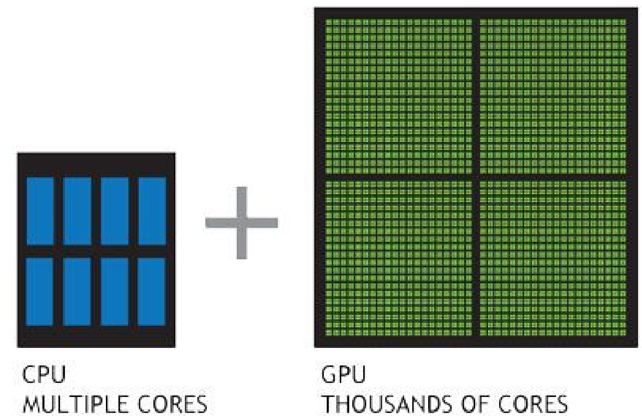
Work: vector instructions (**SIMD**), only effective for problems that can be solved using stream processing (data for similar computation) **SIMD - why it is distributed? – independent from CPU, several graphic cards can be integrated in PC, clusters, etc.**

Applications: calculations, gaming, multimedia.

GPU Computing - Scheme



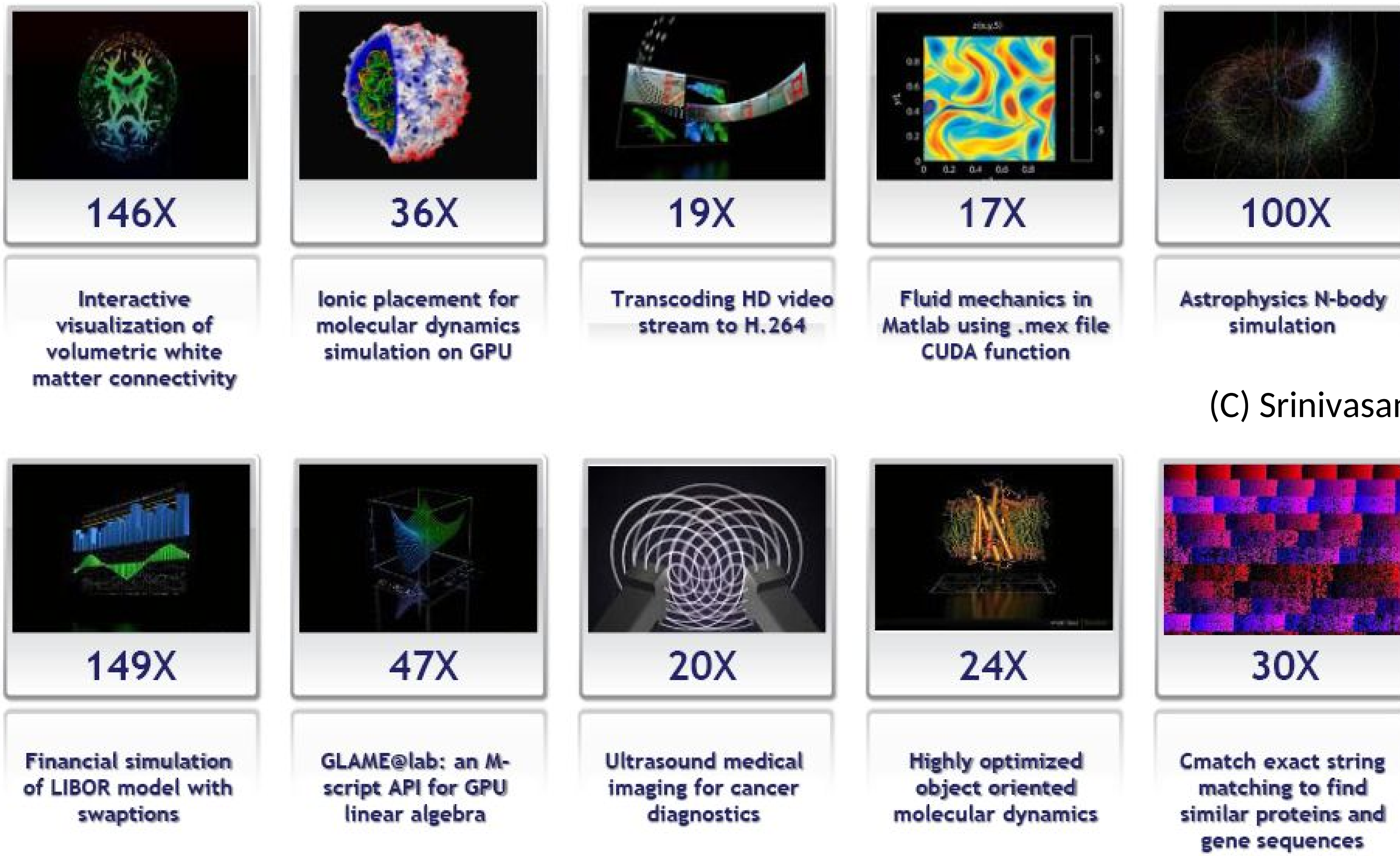
(C) NVIDIA



CPU versus GPU

NVIDIA “Tesla K40” card: **2880** parallel processing cores. Compare: **1.3 TFLOPs <-> 2-8 GFLOPs in PC**!

GPU Computing - Examples



(C) Srinivasan

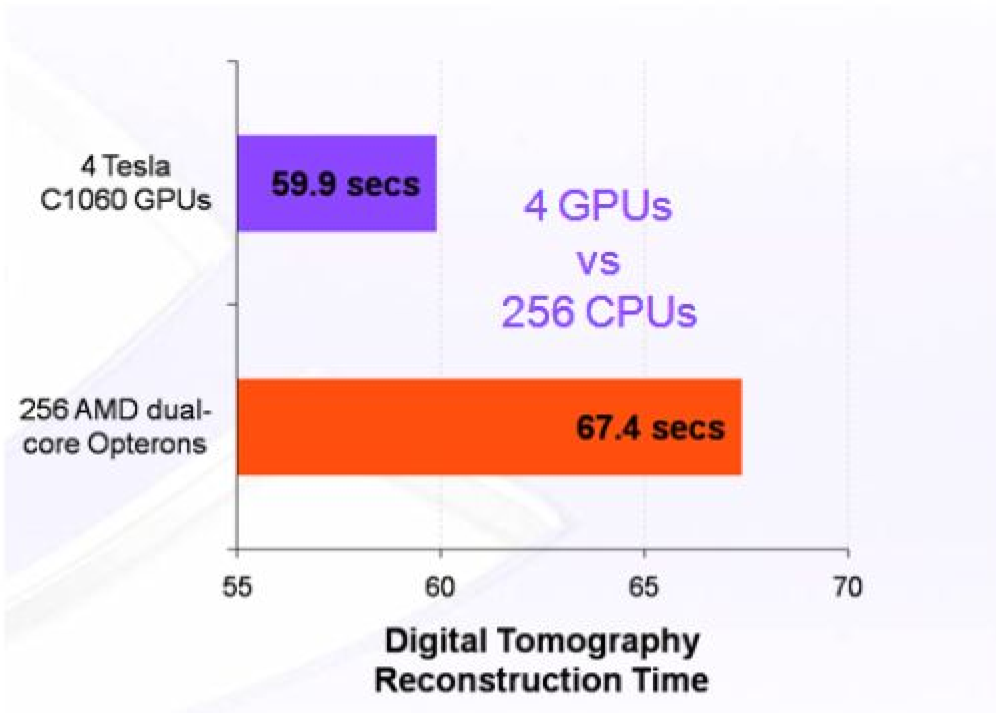
Science (above), gaming, multimedia

GPU Computing - Examples

Desktop beats Cluster

CalcUA

$5 Million



(C) Srinivasan

Tesla Personal Supercomputer

$10,000

Again: SIMD - why it is distributed? – independent from CPU, several graphic cards can be integrated in PC, clusters, etc.

Other Computing Modes - Illustration

Mythbusters:

- Adam
- Jamie

Vivid presentation on GPU-principle at NVIDIA conference (2008)

Other Computing Modes - Illustration

GPU

Mythbusters:

- Adam
- Jamie

Vivid presentation on GPU-principle at NVIDIA conference (2008)

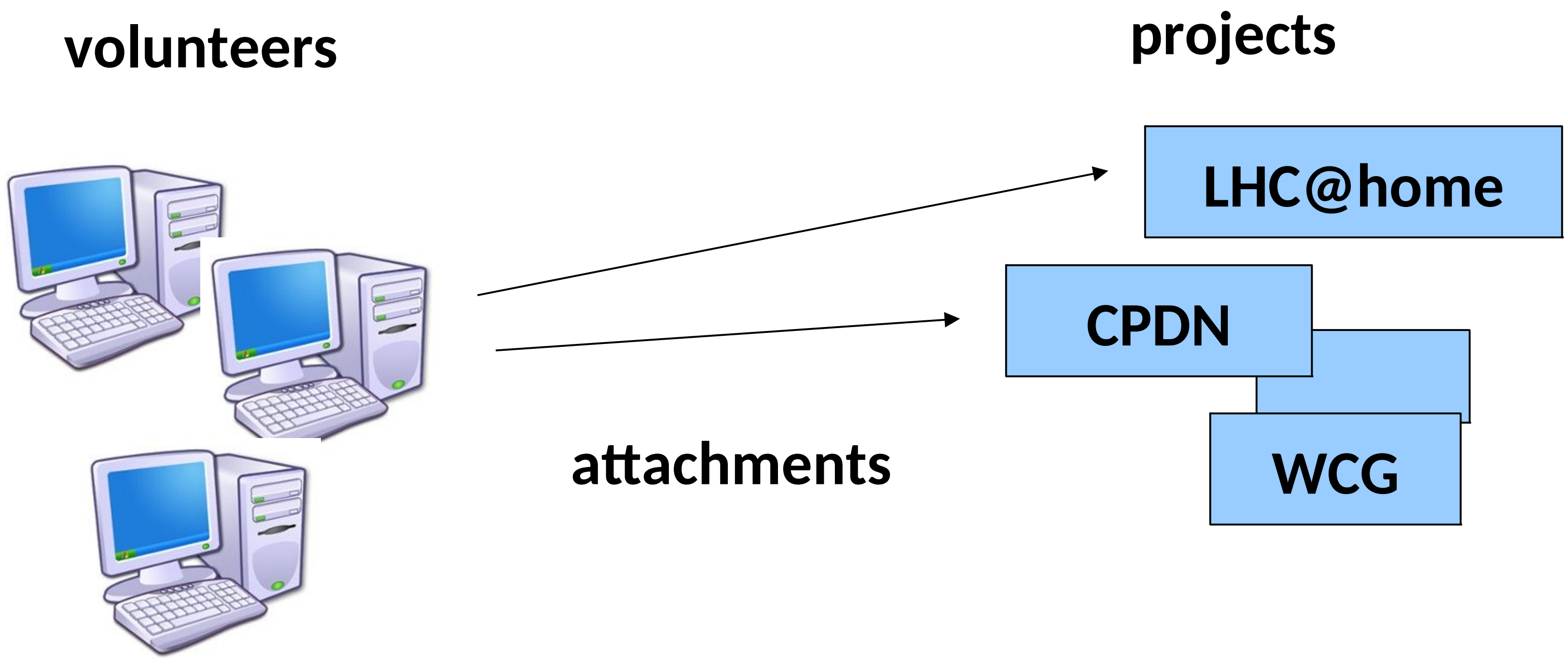
Volunteer Computing - Definition

What is Volunteer Computing?

Computer owners donate their computing resources (such as processing power and storage) to one or more "projects".

- **Why it is important (motivation):**
  - costs
  - performance
- **Applications:** science, multimedia.

Volunteer Computing - Scheme



- Volunteers select project and get “job”
- Volunteers get feedback on their contribution
- Projects compete for volunteers

Volunteer Computing – cost of 1 TFLOPS-year

- Cluster: $145K
  - Computing hardware; power/AC infrastructure; network hardware; storage; power; sysadmin
- **GPU Tesla Cluster: $10K**
- Cloud: $1.75M
- Volunteer: $1K - $10K
  - Server hardware; sysadmin; web development

Volunteer Computing – Performance

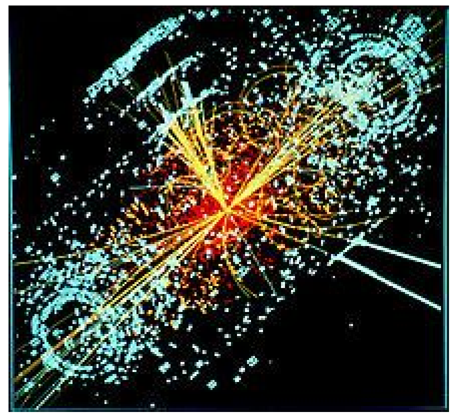
- Current
  - 500K people, 1M computers
  - 6.5 PetaFLOPS (3 from GPUs, 1.4 from PS3s)
- Potential
  - 1 billion PCs today, 2 billion in 2015
  - GPU: approaching 1 TFLOPS
  - How to get 1 ExaFLOPS:
    - 4M GPUs * 0.25 availability
  - How to get 1 Exabyte:
    - 10M PC disks * 100 GB

Volunteer Computing - Examples

- SETI@home

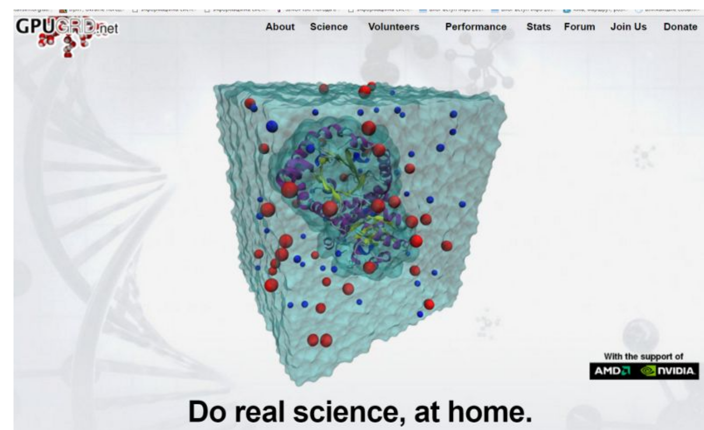
Arecibo radio telescope

- LHC@home



Higgs boson

- **gpugrid.net**



- SLinCA (Ukraine) – materials science – Monte Carlo and molecular dynamics simulations

Desktop Grid Computing - Definition

What is Desktop Grid Computing?

A form of distributed computing in which an organization (business, university, etc.) uses its existing computers (desktop and/or cluster nodes) to handle its own long-running computational tasks.

- **Applications:** calculations, multimedia.

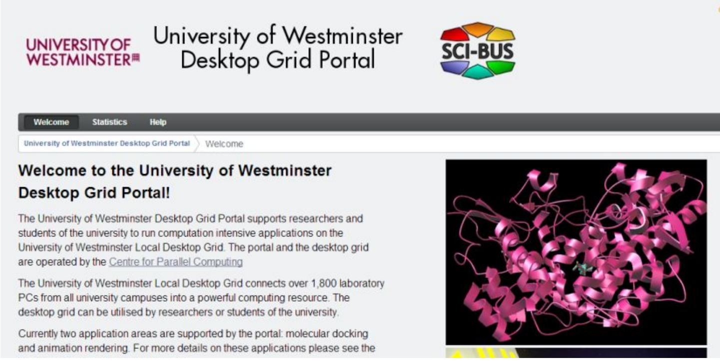
Desktop Grid Computing - Scheme

It is similar to Volunteer Computing, but… differs:

- **The computing resources can be trusted**; i.e. one can assume that the PCs don't return results that are intentionally wrong or falsified
- There is **no need for screensaver graphics**; in fact it may be desirable to have the computation be completely invisible and out of the control of the PC user
- Client **deployment is typically automated**.

Desktop Grid Computing - Examples

- SZTAKI Desktop Grid:
  - How to easily **set up and maintain** your own desktop grid
  - How to easily **develop applications** to be run on the desktop grid
- Westminster University Desktop Grid
  - protein docking, 3D rendering



Peer-to-Peer Computing - Definition

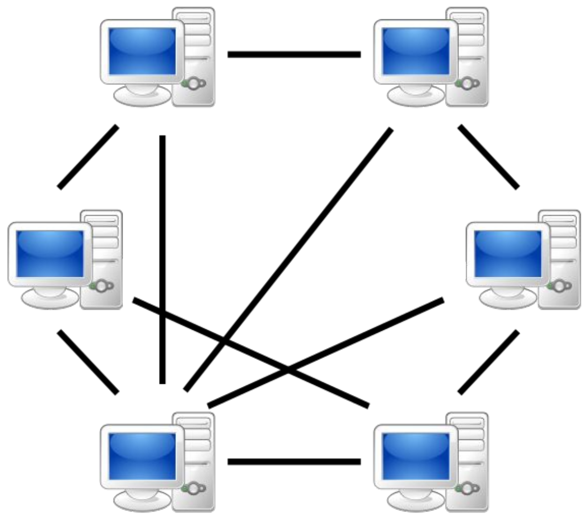
What is Peer-to-Peer Computing?

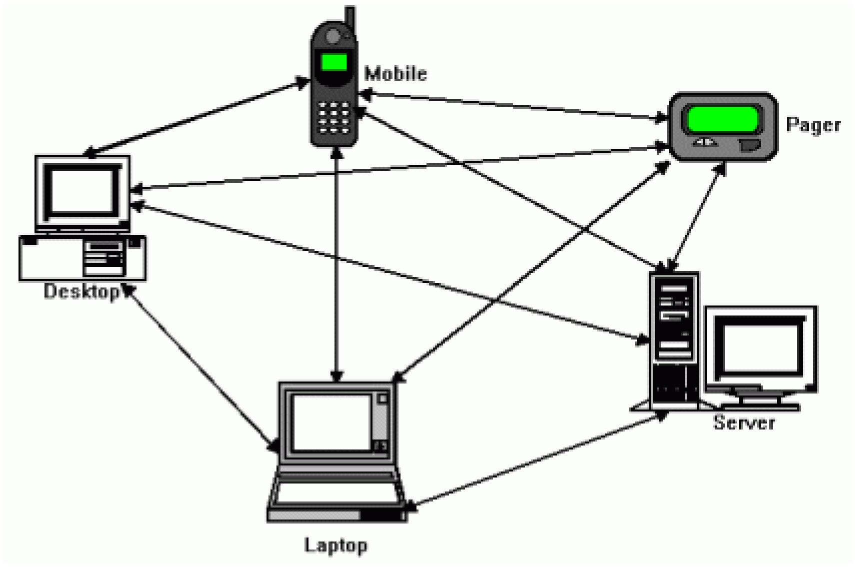
A distributed application architecture that partitions tasks or work loads between peers. Peers are equally privileged, equipotent participants in the application.

- **Work**: No one machine is dedicated to provide special services for others (but sometimes some machine play role of server)

- **Applications:** file sharing, storage, calculations, collaboration.

Peer-to-Peer Computing - Scheme

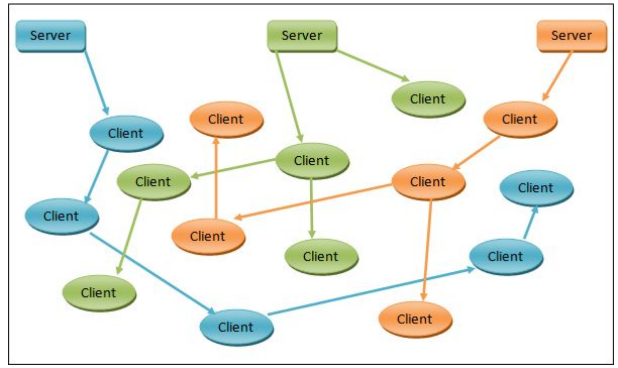




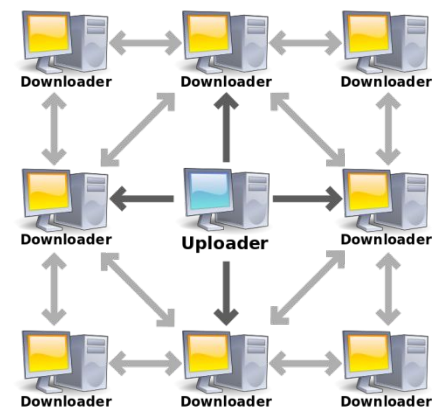
- Unstructured
- Structured
- Hybrid

Peer-to-Peer Computing - Examples

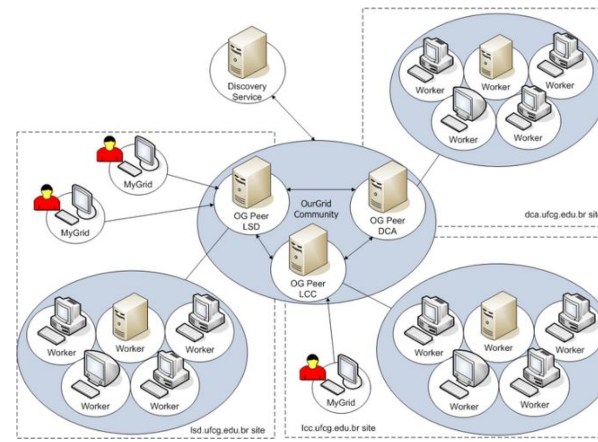
- **Multimedia:** P2PTV -> SopCast



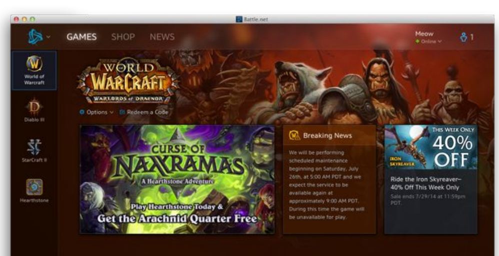
- **File Sharing:** BitTorrent



- **Calculations:** -> OurGrid, XtremWeb



- **Collaborations:** MMORPG -> EveOnline (+WarCraft)



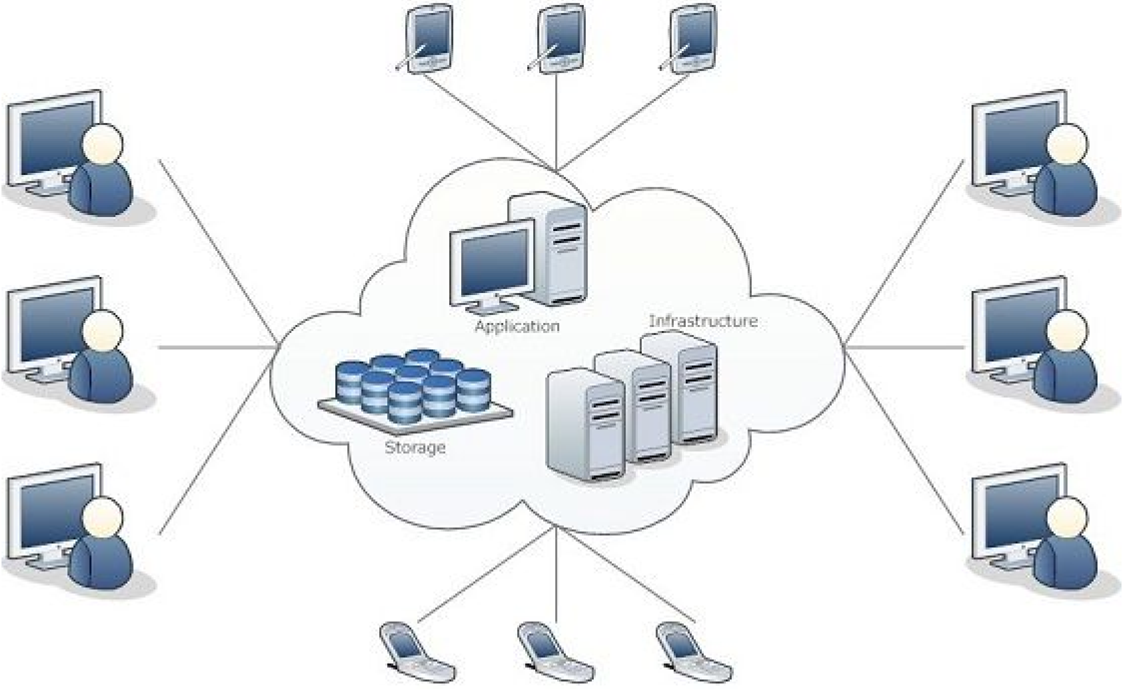
Cloud Computing - Definition

What is Cloud Computing? (jargon)

The delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet).

Applications: e-mail, web conferencing, customer relationship management (CRM)

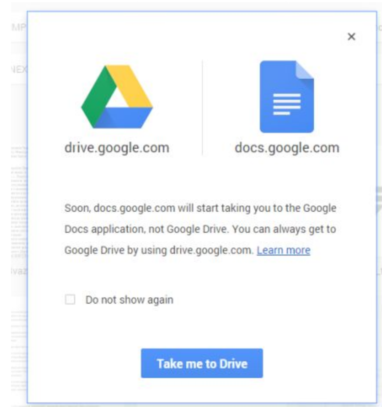
Cloud Computing - Scheme



We need not to install a piece of software on our local PC and this is how, the cloud computing overcomes **platform dependency issues**. Hence, the Cloud Computing is making business application **mobile** and **collaborative**.

Cloud Computing - Examples

- **<u>Collaborations:</u>**
  Google Docs,
  Microsoft Office 365



- **<u>Storage:</u>**
  Amazon Web Services
- **<u>Calculations:</u>**
  Google App,
  Amazon Web Services
  (including GPU-machines and GPU-clusters)

Google Apps

amazon web services™

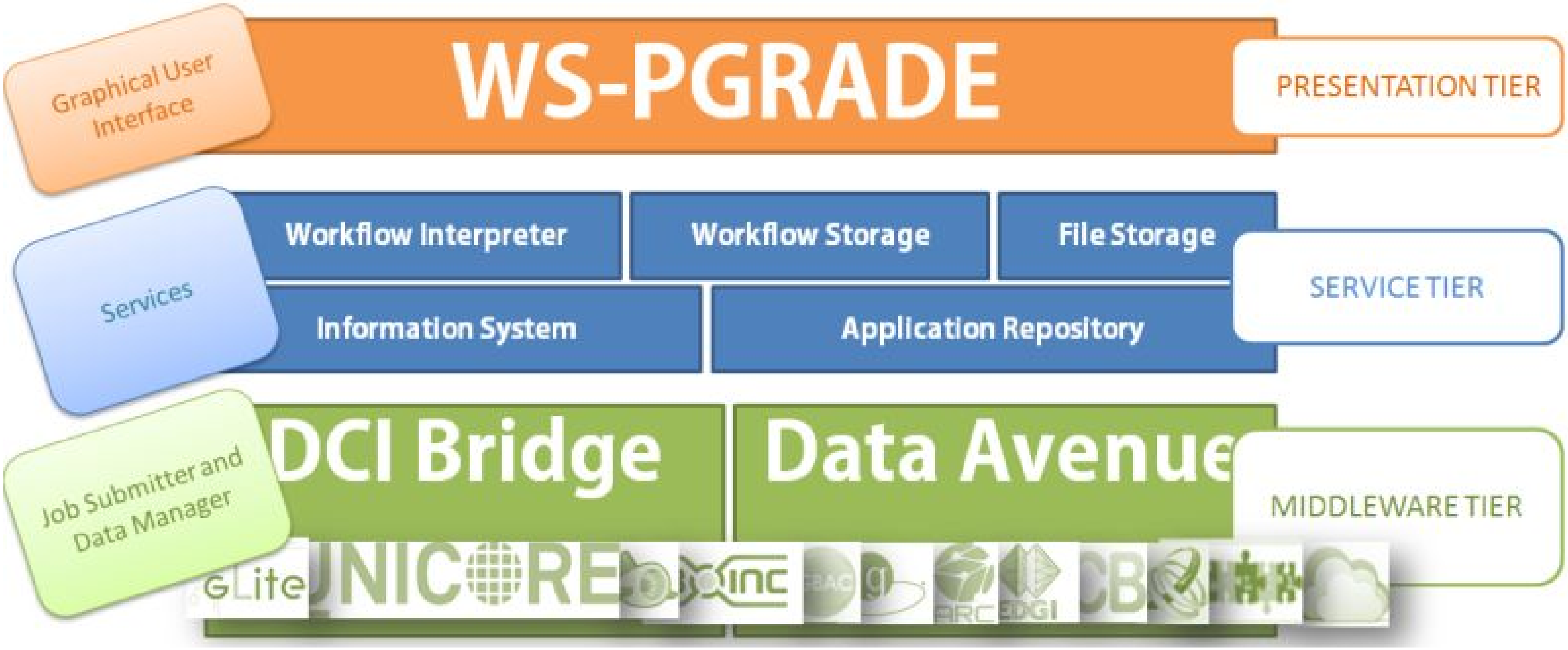
Integration of Distributed Computing Technologies - Definition

What is Science Gateway (SG)?

SG is an interface between a user (or user community) and **MANY VARIOUS** distributed computing infrastructures (DCIs), like grids, clouds, clusters.

- **<u>Applications:</u>** science, multimedia, finance.

Science Gateway Computing - Scheme



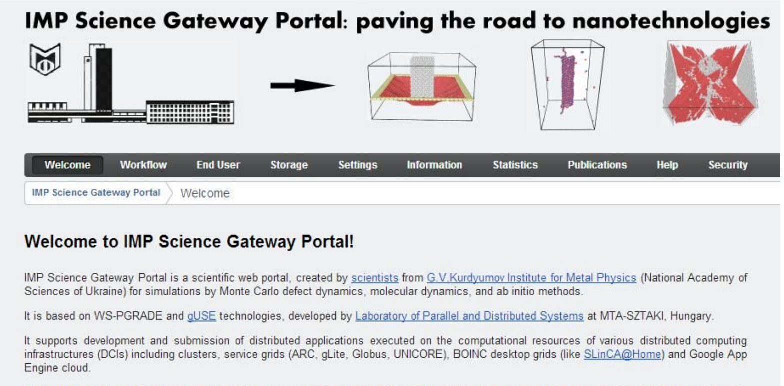
gUSE – Grid and Cloud User Support Environment

Science Gateway Computing - Examples

- Astronomy (VisIVO):

- Chemistry (MoSGrid):

- Materials Science (Ukraine):



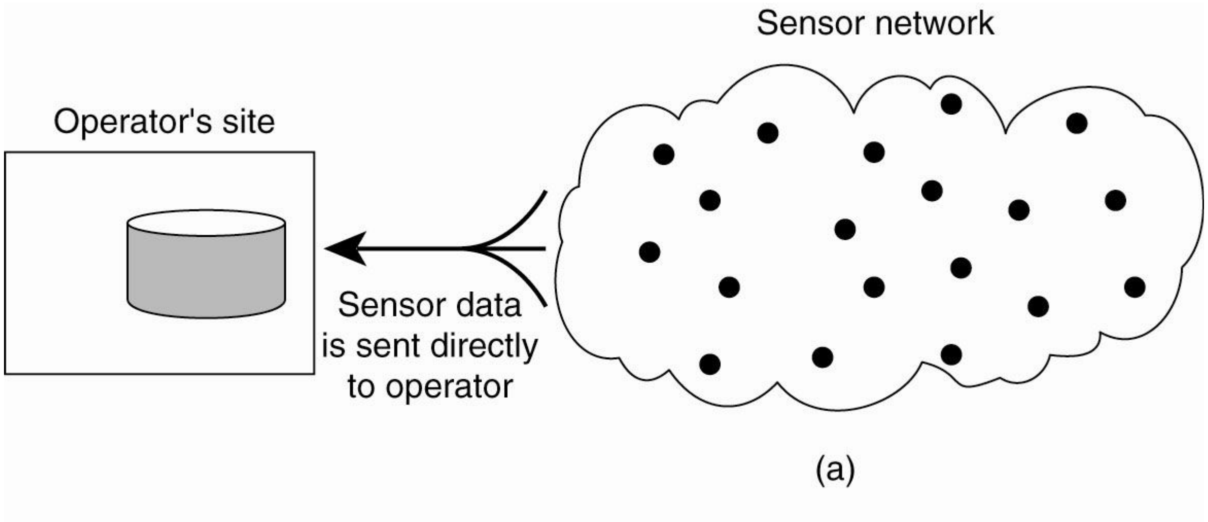
Ubiquitous Computing - Definition

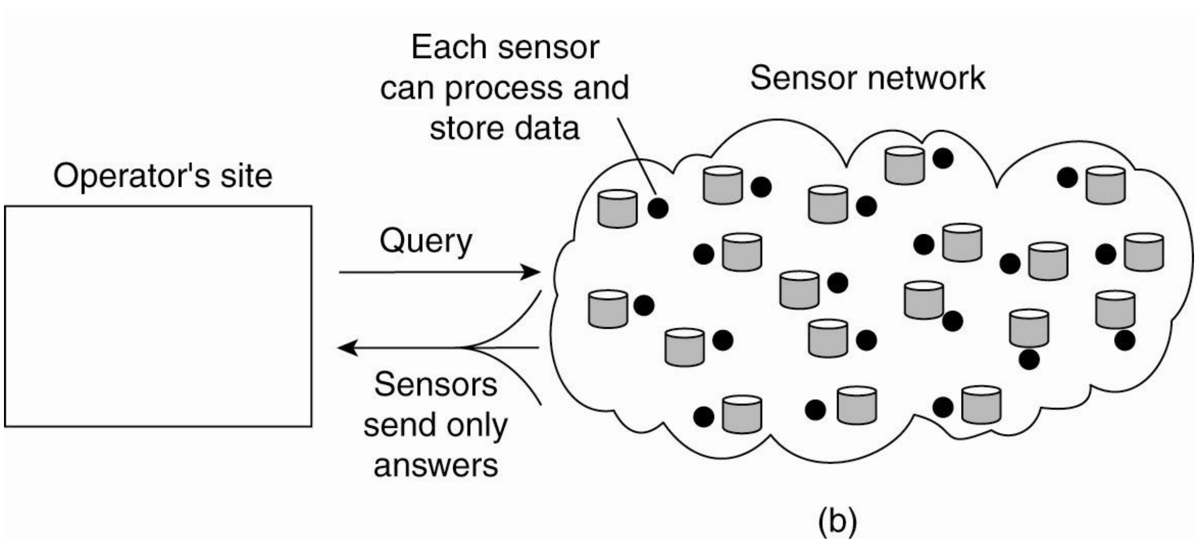
What is Ubiquitous Computing?

In contrast to desktop computing, **Ubiquitous Computing can occur everywhere and anywhere**, using any device, in any location, and in any format, including laptop computers, tablets and terminals in everyday objects such as a fridge or a pair of glasses.

- **Applications:** health care, smart home.

Ubiquitous Computing - Scheme

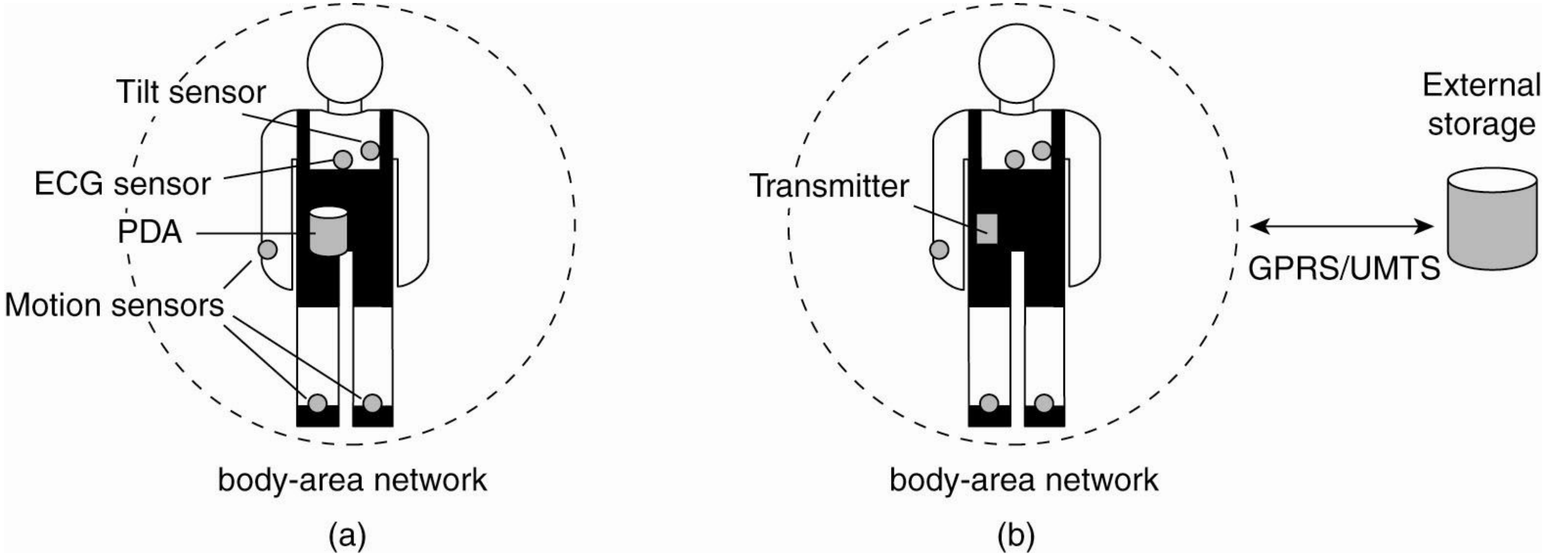




(C) Tannenbaum, van Steen

Ubiquitous Computing - Examples

- Health Care:



- “Smart Home”: home automation

Contacts

Any course-related information
(notifications, reports) from you:

send your message to my e-mail
yuri.gordienko@gmail.com
with the word **GPU2021** in the “Subject” field
(if not, your message will be filtered out to
Spam).