CS546 Parallel and Distributing Processing

- Instructor: Professor Xian-He Sun
  - Email: sun@iit.edu, Phone: (312) 567-5260
  - Office hours: 4:30pm-5:30pm Tuesday, Thursday SB235C
- TA: Xiaoyang Lu
  - Email: xlu40@hawk.iit.edu
  - Office hour: SB003
- Blackboard:
  - http://blackboard.iit.edu
- Additional Web site
  - http://www.cs.iit.edu/~sun/cs546.html

What This Course Is About

- Parallelism
  - What is parallelism?
  - What can be parallelized?
  - Inhibitors and degradation of parallelism: dependences
- Different patterns of parallelism
  - Regular data parallelism
  - Irregular data parallelism
  - Task-queue based parallelism
  - Pipeline parallelism
- Application and Algorithm

What This Course Is About

- Parallel architectures
  - Multicore
  - Shared and distributed parallel computers
  - Memory hierarchy and concurrency
- Standard models of parallelism
  - Shared memory (OpenMP, multithreading, multicore)
  - Distributed Memory (MPI)
  - GPU/GPGPU (CUDA)
  - Data parallelism (HPF)
- Performance of parallel programs
  - Metrics
  - Machine-independent optimization
  - Machine-dependent optimization
  - Performance evaluation techniques

Course Outline

- Overview
- Concurrency and dependency
- Performance evaluation
- Parallel algorithm
- Parallel programming
- Data parallelism and data access
- Methodology
- Current trend

Course Materials

- Required:
  - Lecture slides and assigned reading will be put online
- Recommend book:
  - A. Grama, V. Kumar et al., “Introduction to Parallel Computing”, Addison Wesley, 2003. ISBN 0-201-64865-2
  - W. Gropp, et. al., Using *MPI Tow-Volume Set*
- Reference:
  - Kai Hwang,Jack Dongarra, and Geoffrey C. Fox *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*

Prerequisite

- CS430 “Introduction to Algorithms”
- CS450 “Operating Systems”
- Familiar with
  - Programming in C/C++
  - UNIX tools and development environment
    - Command
    - Editors (vi, emacs), compilers (gcc), makefiles (GNU make)
  - Basic concepts of computer architecture
- Computing platforms:
  - Local and remote

Misc. Course Details

- You are expected to attend all lectures and presentations
- Grading
  - Homework and programming assignments (30%)
  - Exam (50%)
  - Term project (20%), see http://www.cs.iit.edu/~sun/html/report.html
  - Important date: Spet. 10, Nov. 26, 2019
- Homework are due in class on assigned date
  - There is a late penalty of 10% per day
- Use the course blackboard
  - Announcements, Lecture notes, Assignments
  - …

Policies

- Collaboration policy
  - Encouraged for high level concepts and understanding the courses materials
  - but …..
- Cheating policy
  - Copying all or part of another student's homework
  - Allowing another student to copy all or part of your homework
  - Copying all or part of code found in a book, magazine, the Internet, or other resource

Policies

- IIT Code of Academic Honesty [link]
- All violations of academic integrity will be reported to academichonesty@iit.edu
- Sanctions for violations of academic integrity
  - **Expulsion from a course.** The student is assigned a punitive failing grade of ‘E’ for the course and can no longer participate in the course or receive evaluation of coursework from the instructor.
  - **Suspension.** Suspension is a status assigned for various periods of time in which a student’s enrollment is interrupted. A suspended student may not attend day or evening classes, participate in student activities, or live in campus housing. A suspended student may apply for reinstatement at the end of the period of suspension. If reinstated, the student may be placed on disciplinary probation for a period of time designated by the Dean for Academic Discipline.
  - **Expulsion**. Expulsion is the complete severance of association with the University. Notation of the violation of the Code is made on the student’s transcript

Any Questions?

Introduction

What Is Computer Science?

Computer science is laying the foundations and developments the real search paradigms and scientific methods for the exploration of the world of information and intellectual processes that are not directly governed by physical laws.

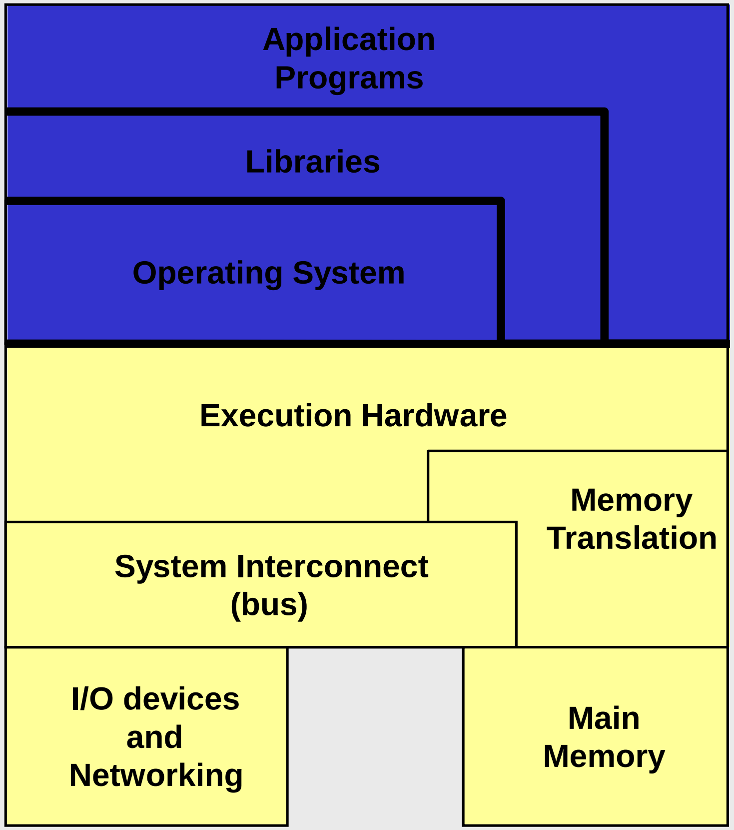
By Juris Hartimanis, Turing Award Lecture

Such people (computer scientists) are especially good at dealing with situations where different rules apply in different cases; they are individuals who can rapidly change levels of abstraction, simultaneously seeing things “in the large” and “in the small”.

By Donald Knuth, Turing Award Lecture

The "Machine"

- Different perspectives on what the *Machine* is:
- OS developer

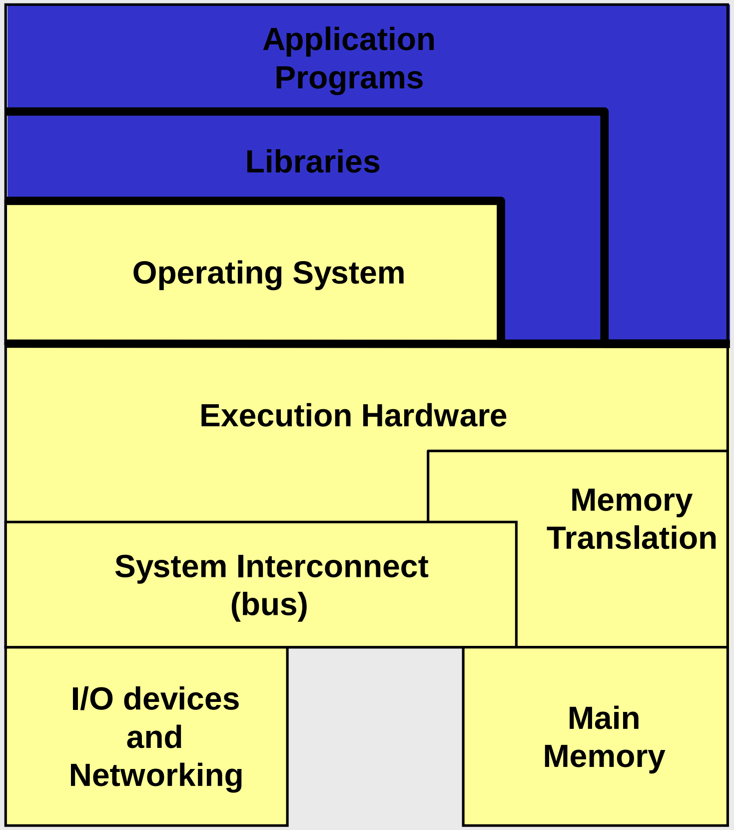


Instruction Set Architecture

- –ISA
- –Major division between hardware and software

The "Machine"

- Different perspectives on what the *Machine* is:
- Compiler developer

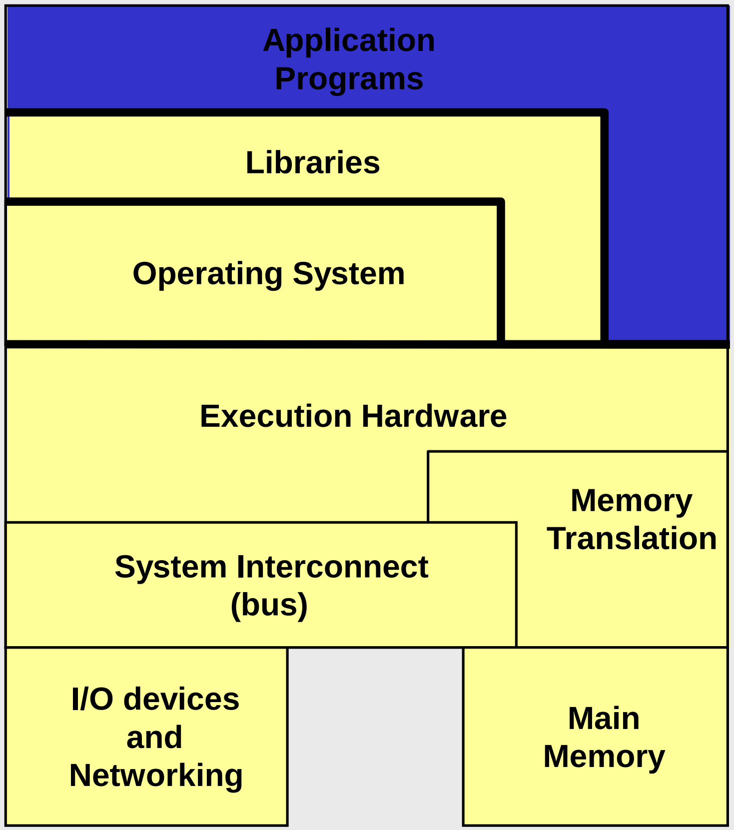


Application Binary Interface

- ABI
- User ISA + OS calls

The "Machine"

- Different perspectives on what the *Machine* is:
- Application programmer

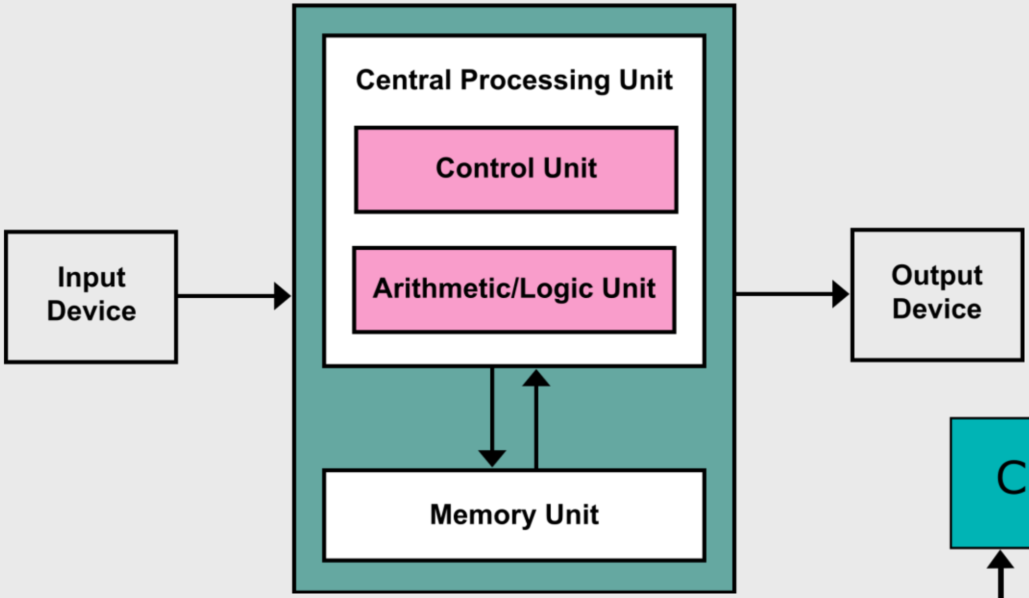


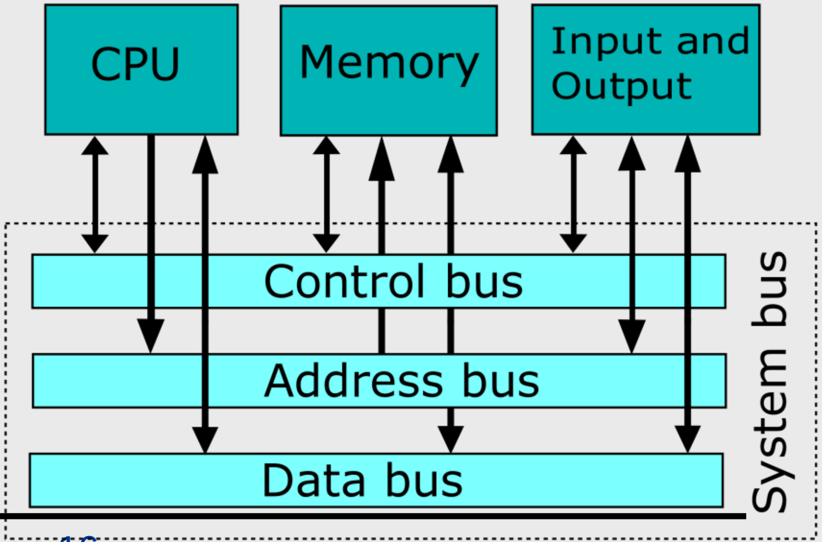
Application Program Interface

- –API
- –User ISA + library calls

The von Neumann Machine

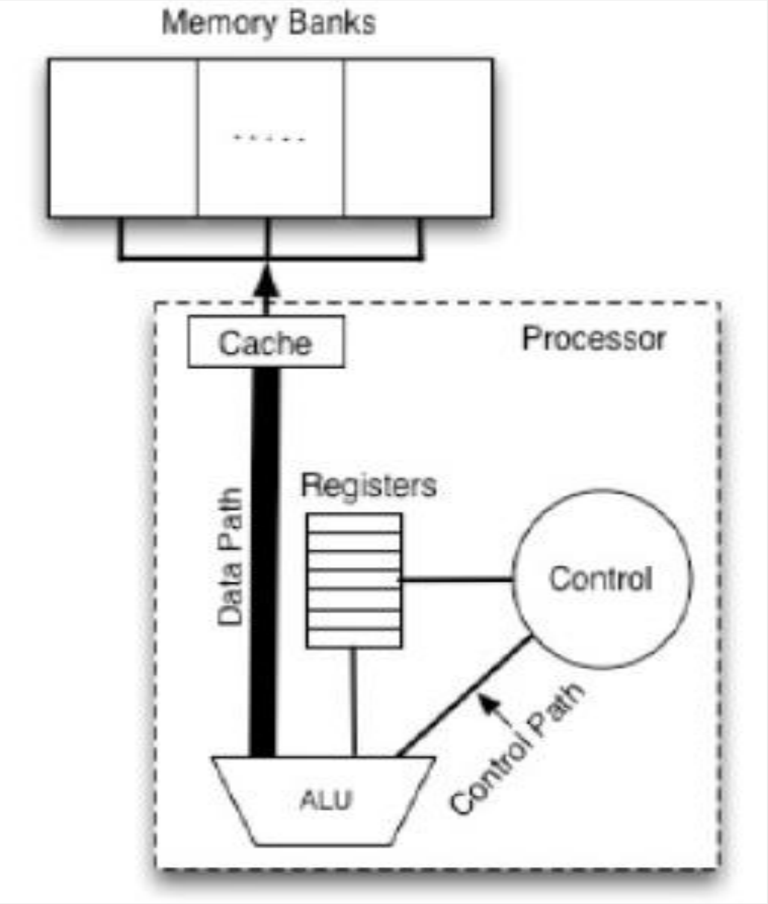
- The basic
- The von Neumann bottleneck



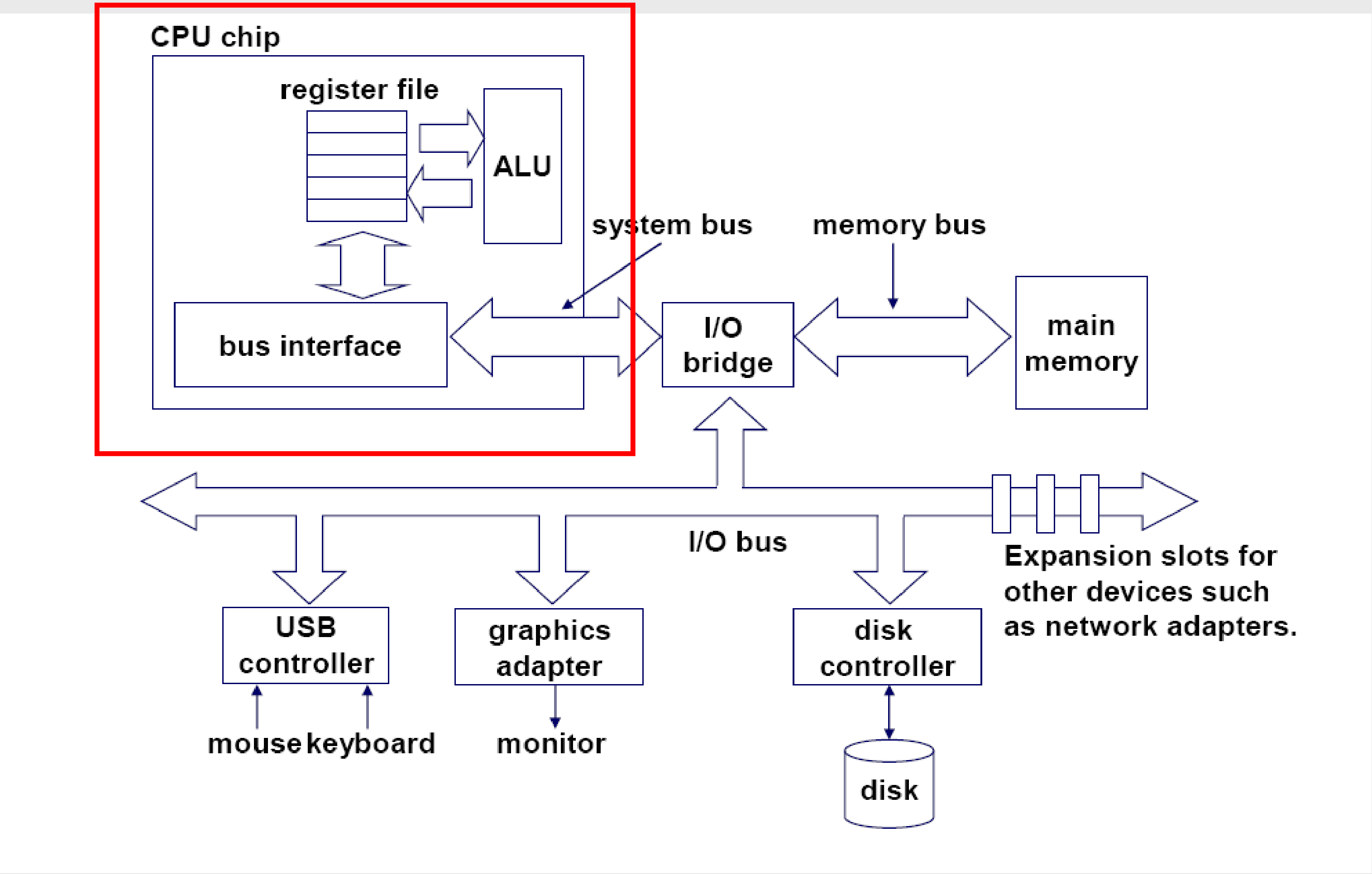


Basic Uni-Processor Architecture elements

- Cache hierarchy
- Arithmetic Logic Units
- Register Sets
- Control
- Memory Interface
- I/O Interface
- Execution pipeline

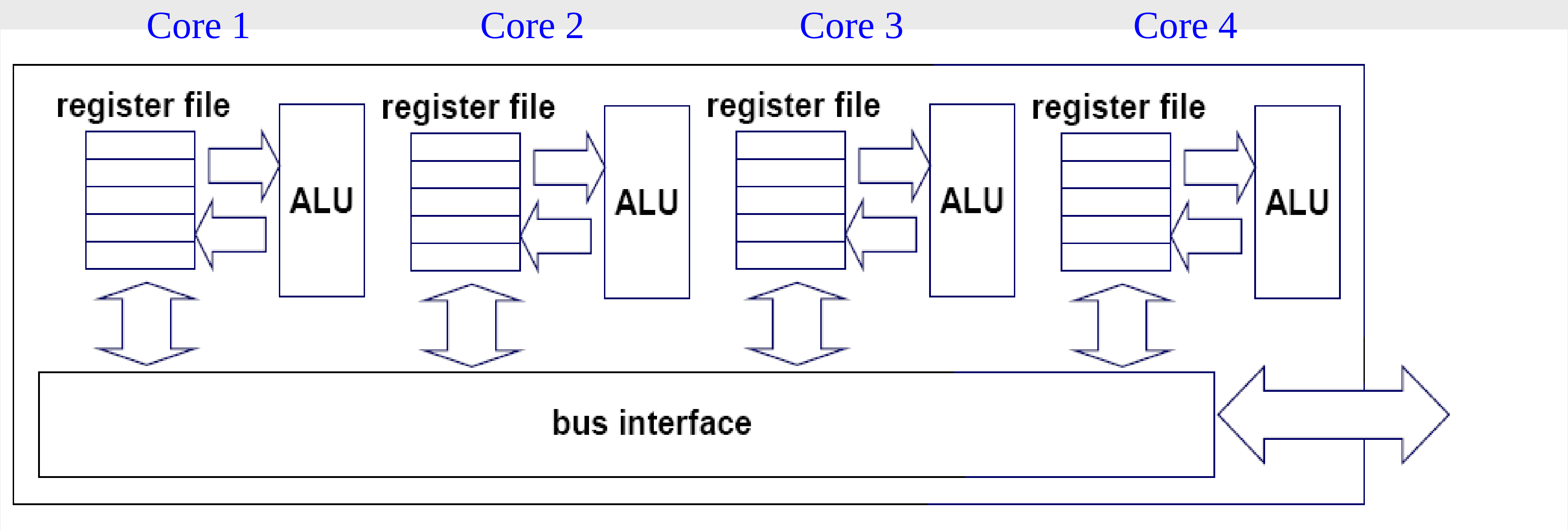


Single Core Computer



Multicore CPUs

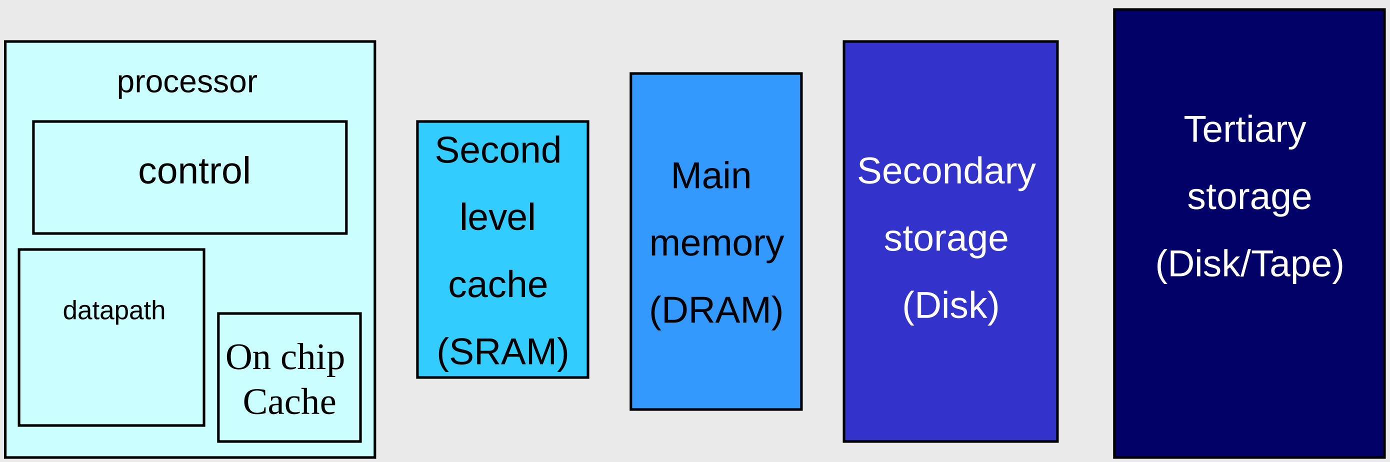
- Most of current Intel and AMD multicore CPUs just put multiple cores



Levels of the Memory Hierarchy

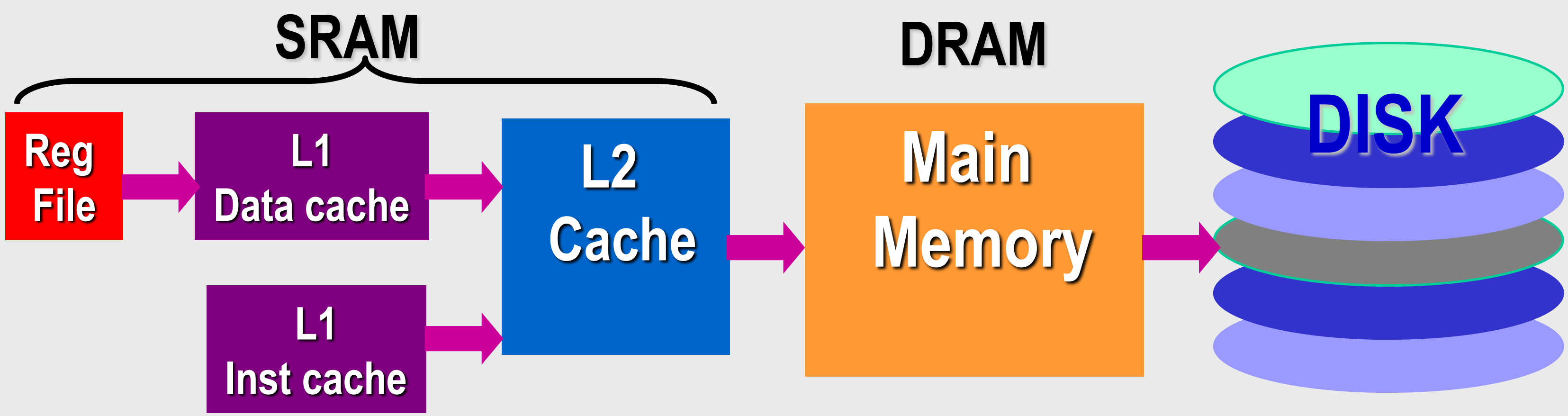
Deeper levels of cache memory

Large memories are slow, fast memories are small and expensive.



Speed (ns): 1-5 ns	5 – 25 ns	80 - 250ns	~10 ms	~ 10 sec
Size (bytes): 32KB	1-2MB	1-8GB	20-200GB	TB

Model of Memory Hierarchy



What is Parallel Processing

- Parallel Processing
  - Several working entities work together toward a common goal
- Parallel Computer
  - A computer designed for parallel processing
- Scalable Computing
  - A parallel computing which can be scaled up to larger size without losing efficiency

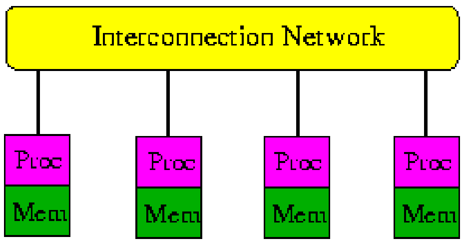
- Supercomputer (high performance computer, high end computer, advanced computer)
  - A general-purpose computer capable of solving individual problems at extremely high computation speed

Parallel Architectures

IBM RS/6000 SP

SGI Power Challenge XL

Distributed Memory Machines



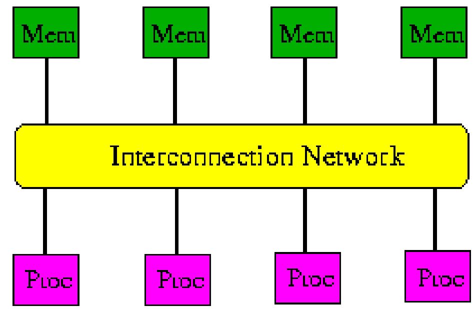
Advantages:

+ scalable
+ latency hiding

Disadvantages:

- harder to program
- program must be replicated

Shared Memory Machines



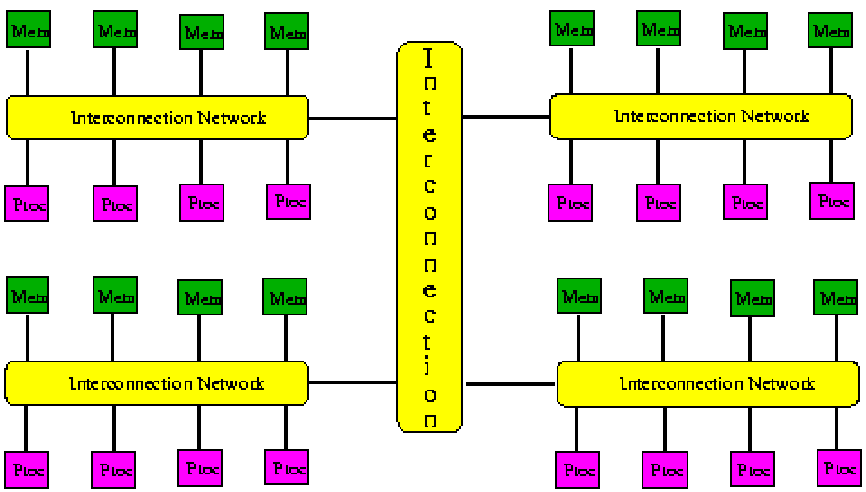
Advantages:

+ ease of programming
+ processors share code and data

Disadvantages:

- scalability problem

Cluster of Symmetric Multiprocessor Systems (SMP)



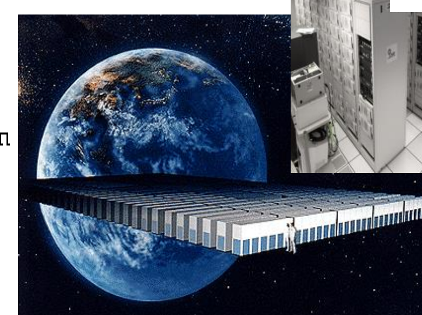
NEC SX-5 multi node (8 CPUs pro Knoten)

Advantages:

+ scalable

Disadvantages:

- programming paradigm unclear



Earth Simulator (540*8 CPUs)

What Is Computer System?

In general, Computer System could mean both
computer system hardware (architecture)
and the computer system software, but
mostly means system software

System Software: OS, libraries, runtime environments

Parallel OS

Distributed OS,

Distributed Systems

Personal Introduction

- Research interests
  - High Performance Computing (Big Data Systems)
  - Memory and I/O system
  - Performance Analysis and Modeling

- Research group:
  - Scalable Computing Software Laboratory (SCS)
  - http://www.cs.iit.edu/~scs/
  - Weekly Research seminar

Personal Introduction

- Distinguished Professor of Computer Science
  - The highest academic rank from a university
- IEEE Fellow
  - The highest professional recognition from IEEE
- Associate Editor-in-Chief of IEEE Transactions of Parallel and Distributed Systems
  - The flagship professional journal in computer science
- The Past Chair of the CS Department at Illinois Tech

The Scalable Computing Software Lab

www.cs.iit.edu/~scs/

Specialize in high performance software systems for big data applications

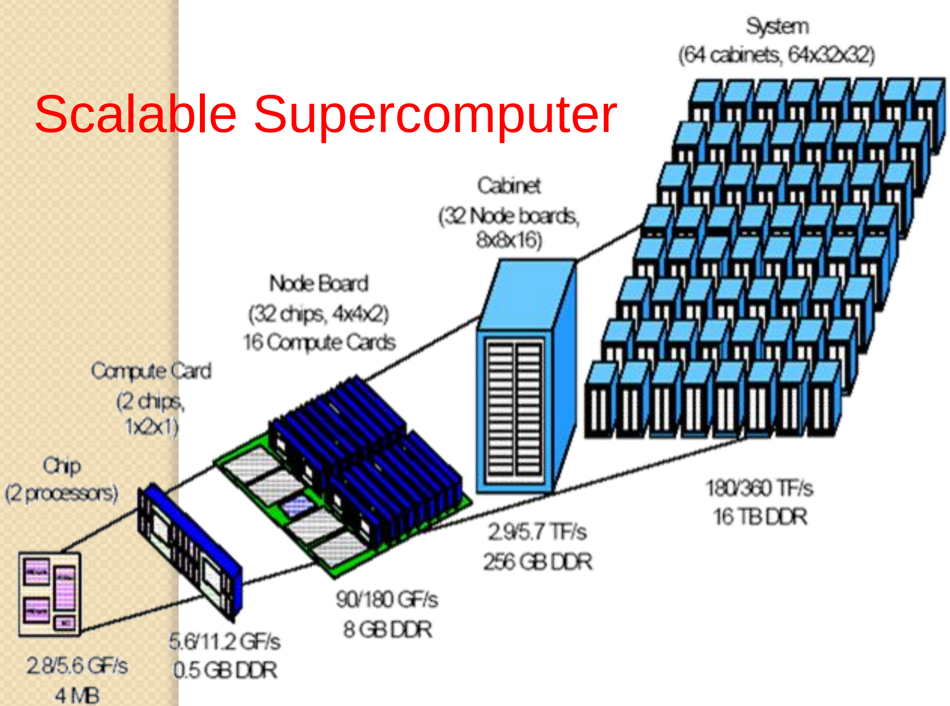
Supported by:

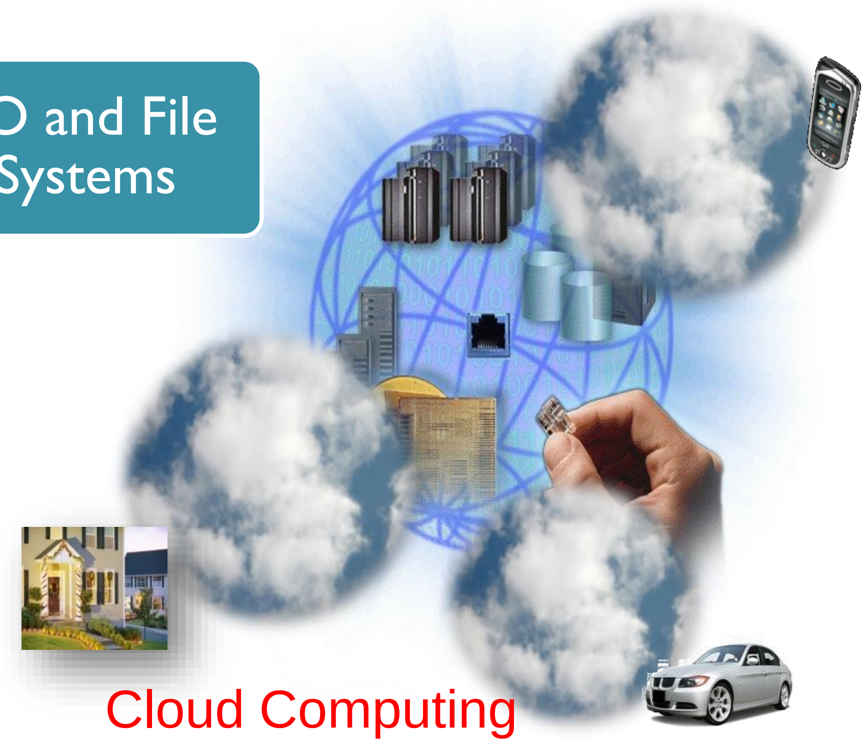
- NSF, DoE, NASA, and industry

Advanced Computing at SCS

- Data intensive computing
  - Big data management
- Memory Systems
- I/O and File Systems

Scalable Supercomputer





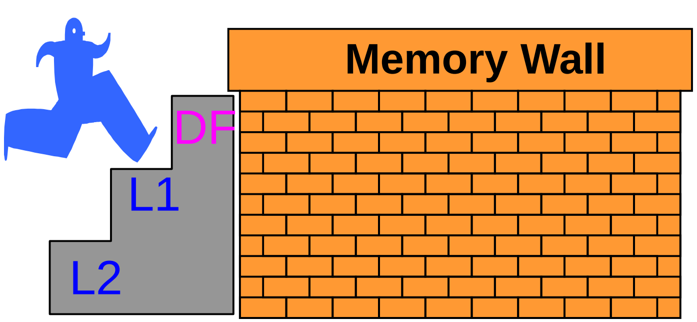
Cloud Computing

System Research

Hot Issues

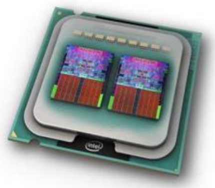
- AI and Deep Learning
- Big Data
- *High Performance and Could Computing*

COMPUTING POWER

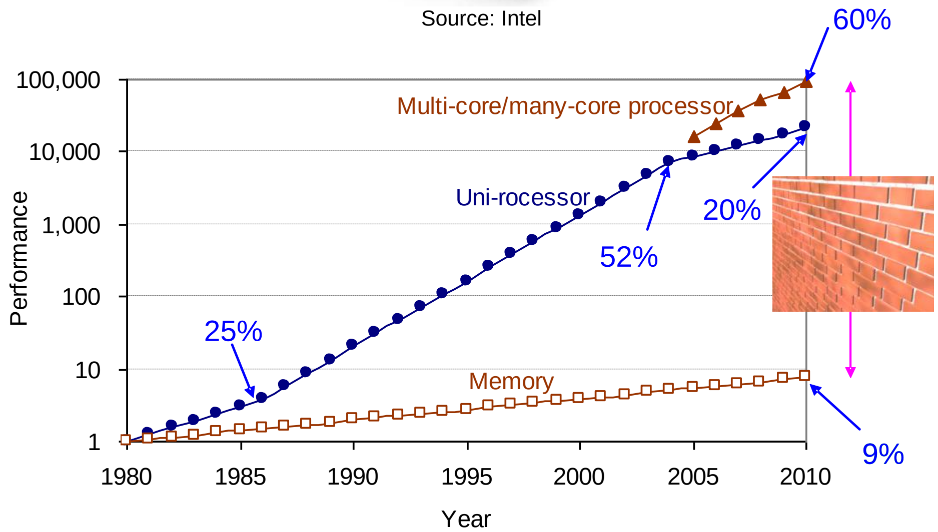


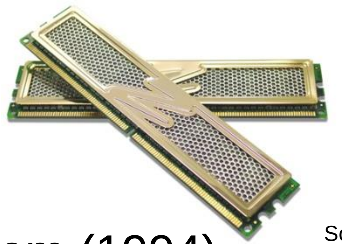
Impact: The Memory-wall Problem

- Processor performance increases rapidly
  - Uni-processor: ~52% until 2004
  - Aggregate multi-core/many-core processor performance even higher since 2004
- Memory: ~9% per year
  - Storage: ~6% per year
- Processor-memory speed gap keeps increasing



Source: Intel





Source: OCZ

Memory-bounded speedup (1990), Memory wall problem (1994)

Memory Hierarchy

Capacity
Access Time, Bandwidth

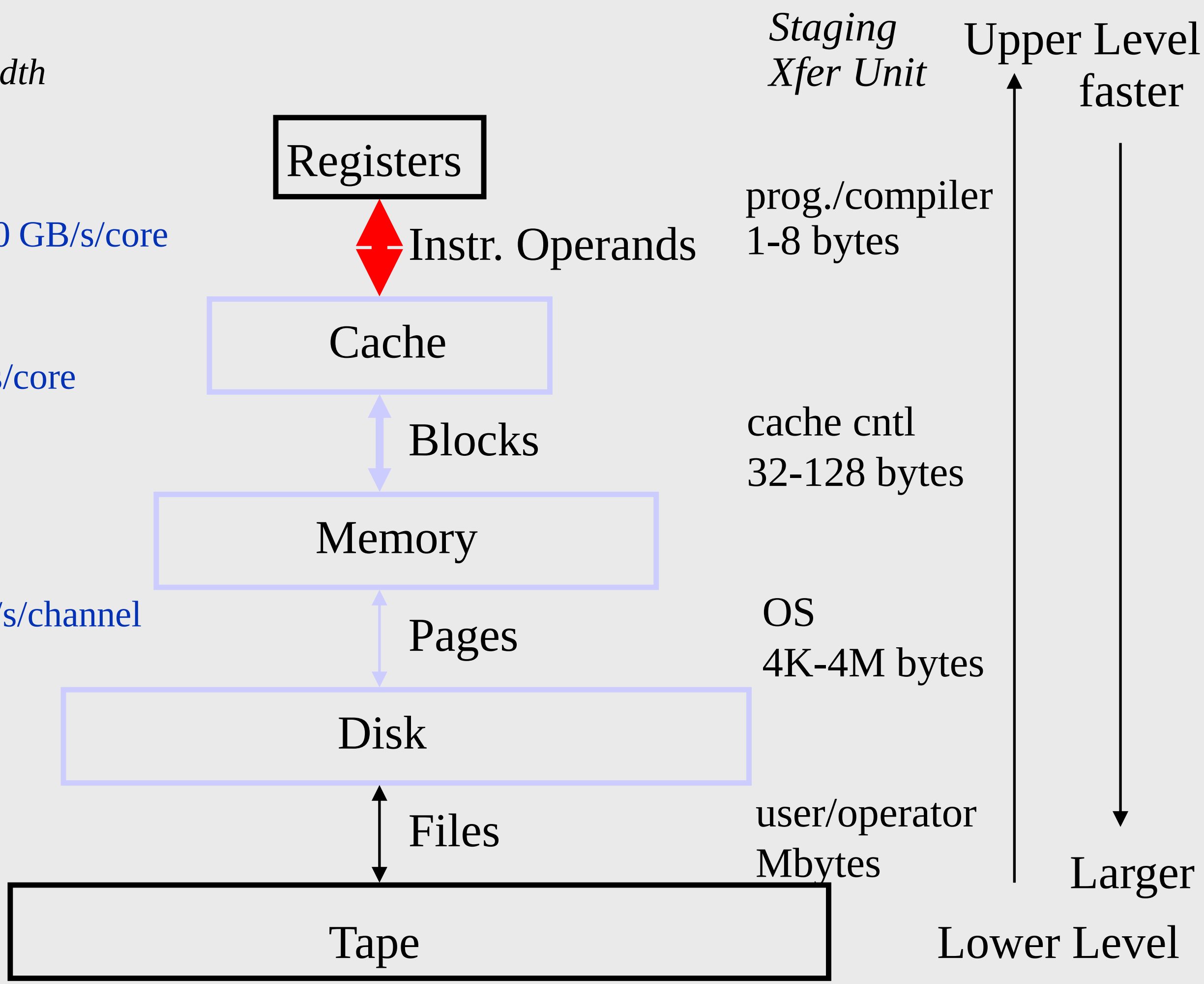
CPU Registers
<8KB
<0.2~0.5 ns, 500~800 GB/s/core

Cache
<50MB
1-10 ns, 50~150GB/s/core

Main Memory
Giga Bytes
50ns-100ns 5~10GB/s/channel

Disk
Tera Bytes, 5 ms
100~300MB/s

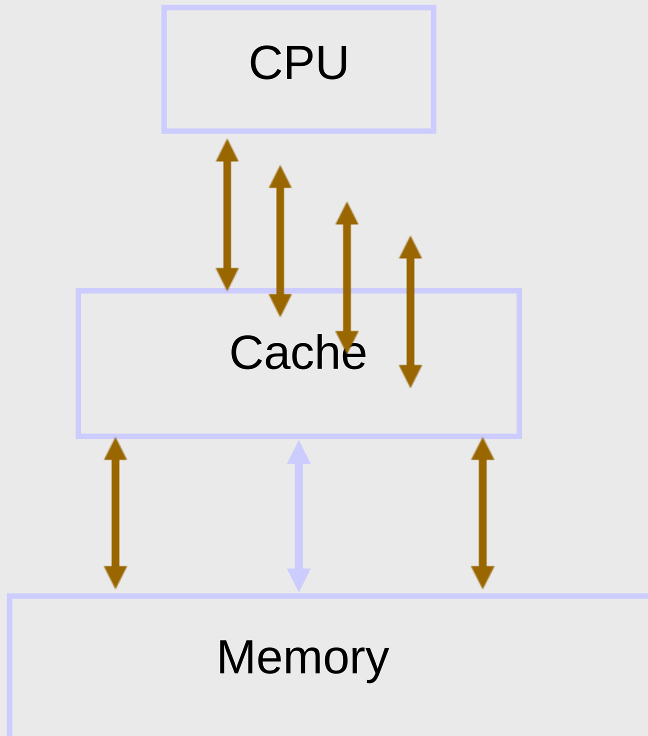
Tape
Peta Bytes or
infinite
sec-min



Solution: Memory Hierarchy and Concurrency

Multi-core
Multi-threading
Multi-issue

Out-of-order Execution
Speculative Execution
Runahead Execution



Multi-banked Cache
Multi-level Cache

Pipelined Cache
Non-blocking Cache
Data Prefetching
Write buffer

Multi-channel
Multi-rank
Multi-bank

Pipeline
Non-blocking
Prefetching
Write buffer

Input-Output (I/O)

Parallel File System

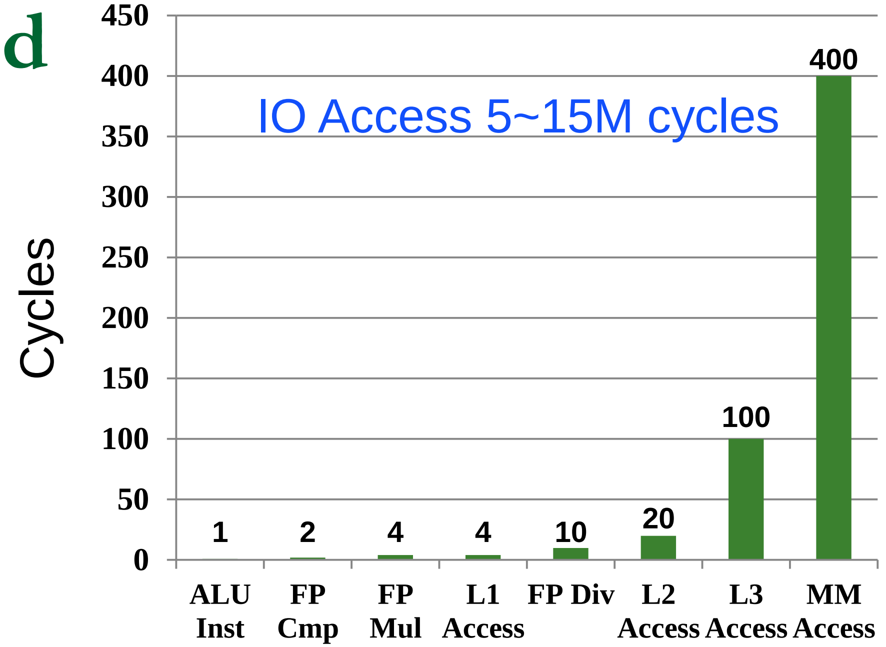
Disks

Assumption of Current Solutions

- Memory Hierarchy: Locality
- Concurrence: Data access pattern
  - Data stream

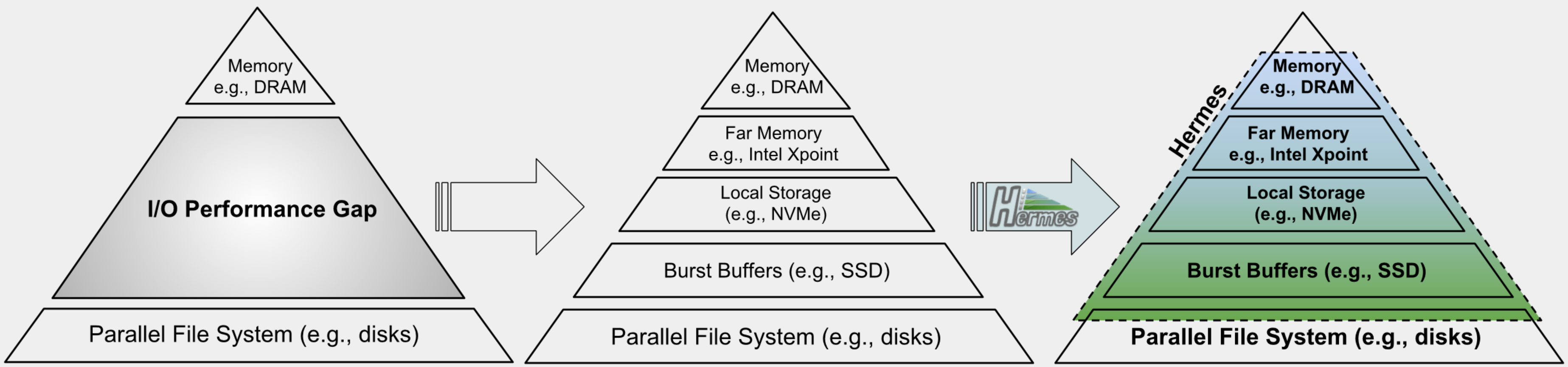
Extremely Unbalanced Operation Latency

Performances vary largely

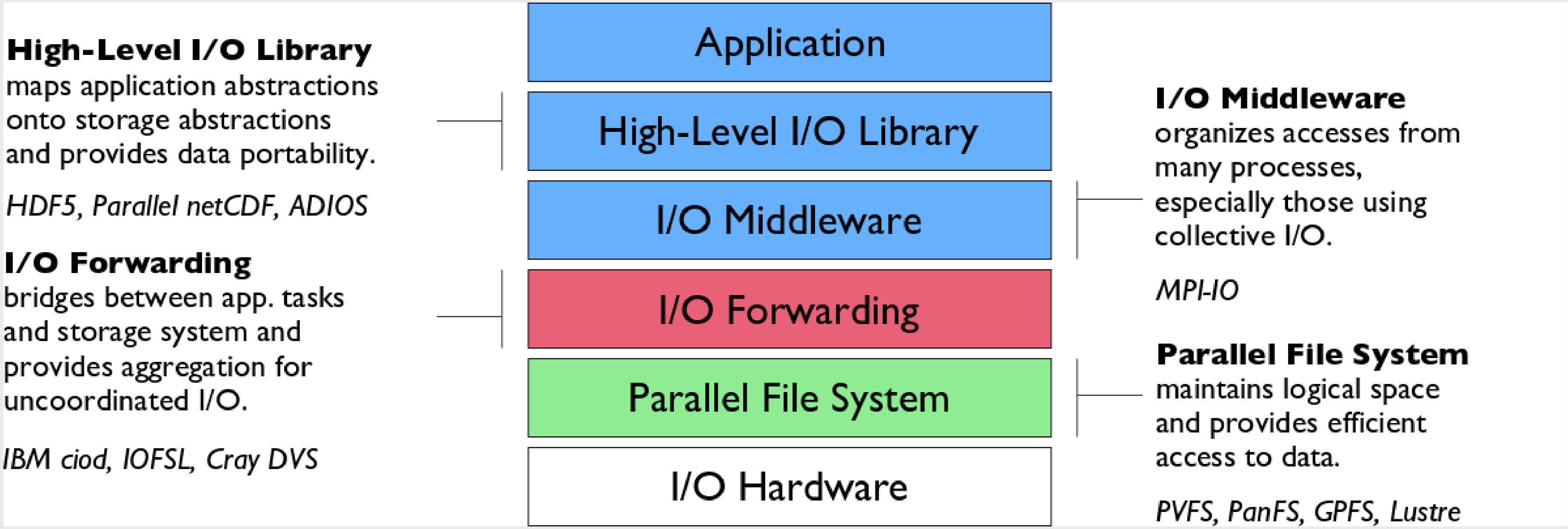


Current Project: Hermes

- A new, multi-tiered, distributed caching platform that:
  - Enables, manages, and supervises I/O operations in the Deep Memory and Storage Hierarchy (DMSH).
  - Offers selective and dynamic layered data placement/replacement
  - Is modular, extensible, and performance-oriented.
  - Supports a wide variety of applications (scientific, BigData, etc.,).



I/O stack



Additional I/O software provides improved performance and usability over directly accessing the parallel file system. Reduces or (ideally) eliminates need for optimization in application codes.

Work Opportunities

- Research opportunities for graduate students:
  - Always look for self-motivated and hard-working grad students
  - Ph.D. students: CS597 and CS691
  - MS students: CS591 “Research and Thesis for MS Degree”
  - Take CS546 & CS550, check my research projects, send me your CV
- Research opportunities for undergrad students:
  - NSF REU (Research Experiences for Undergraduates) with Prof. Xian-He Sun
    - Various project topics, including development of scheduling simulator, analysis of system logs, ….
    - $13-$15 per hour
    - If interested, contact Prof. Sun (sun@iit.edu)

Question:

- With the data-centric view, what is the machine, and what is the machine layers?

Introduction

Historical Point of View

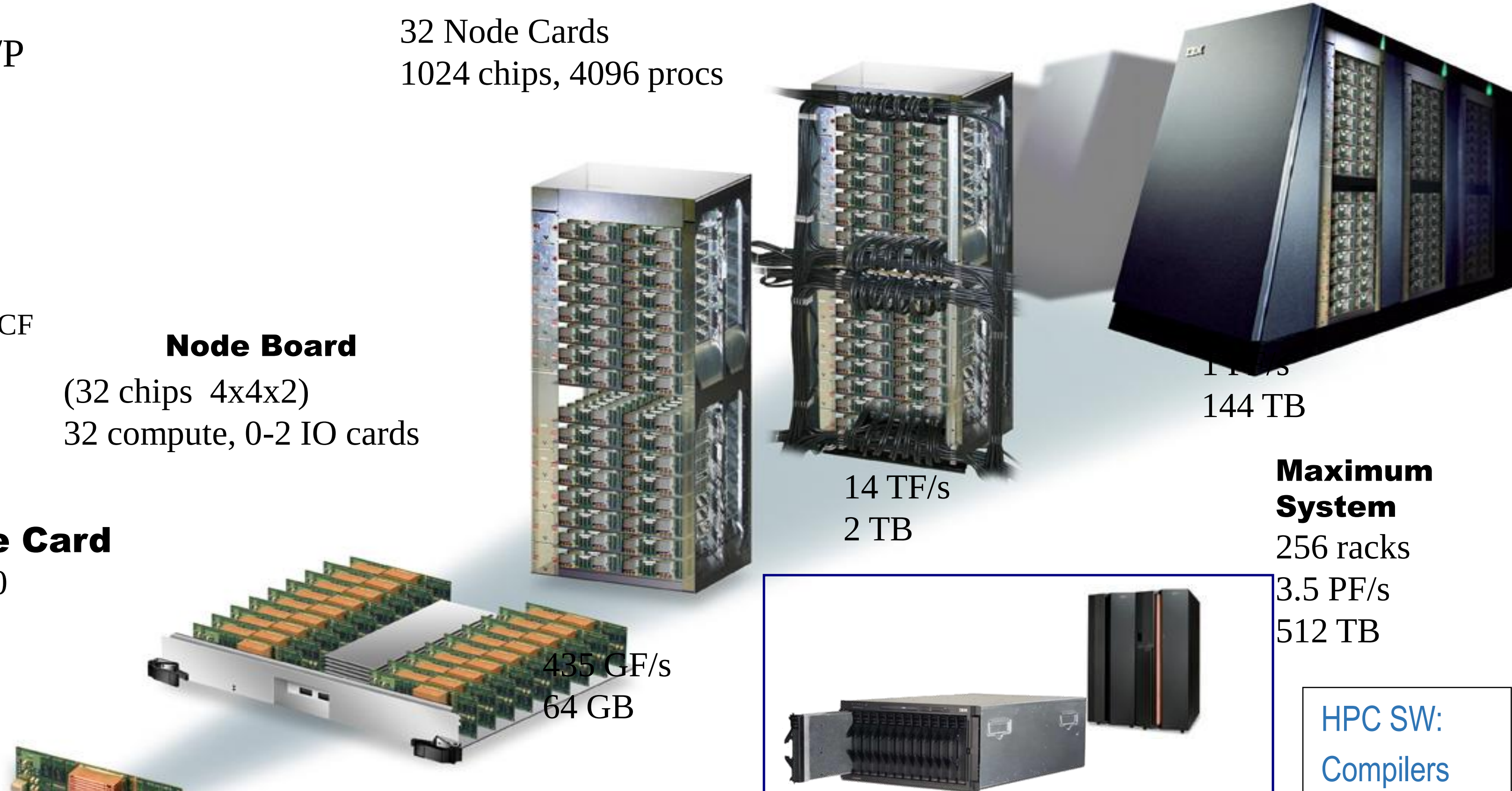
The Third Wave of Computing Revolutions

- Network, communication, and interconnectivity
- Begin in the late 90s until now
- Machine/machine, software/software, people/people
- Anytime, anywhere, WWW
- The communications landscape is shifting
  - Could computing，Big Data, Network of things
  - Edge Computing, Pervasive computing

Evolution of Computing:
The biggest machine becomes even bigger

IBM BG/P

Source: ANL ALCF



Petaflops System

72 Racks

1 PF/s
144 TB

Rack Cabled 8x8x16

32 Node Cards
1024 chips, 4096 procs

14 TF/s
2 TB

Node Board
(32 chips 4x4x2)
32 compute, 0-2 IO cards

435 GF/s
64 GB

Compute Card
1 chip, 20 DRAMs

13.6 GF/s
2.0 GB DDR
Supports 4-way SMP

Chip
4 cores

850 MHz
8 MB EDRAM

Maximum System
256 racks
3.5 PF/s
512 TB

Front End Node / Service Node
System p Servers
Linux SLES10

HPC SW:
Compilers
GPFS
ESSL
Loadleveler

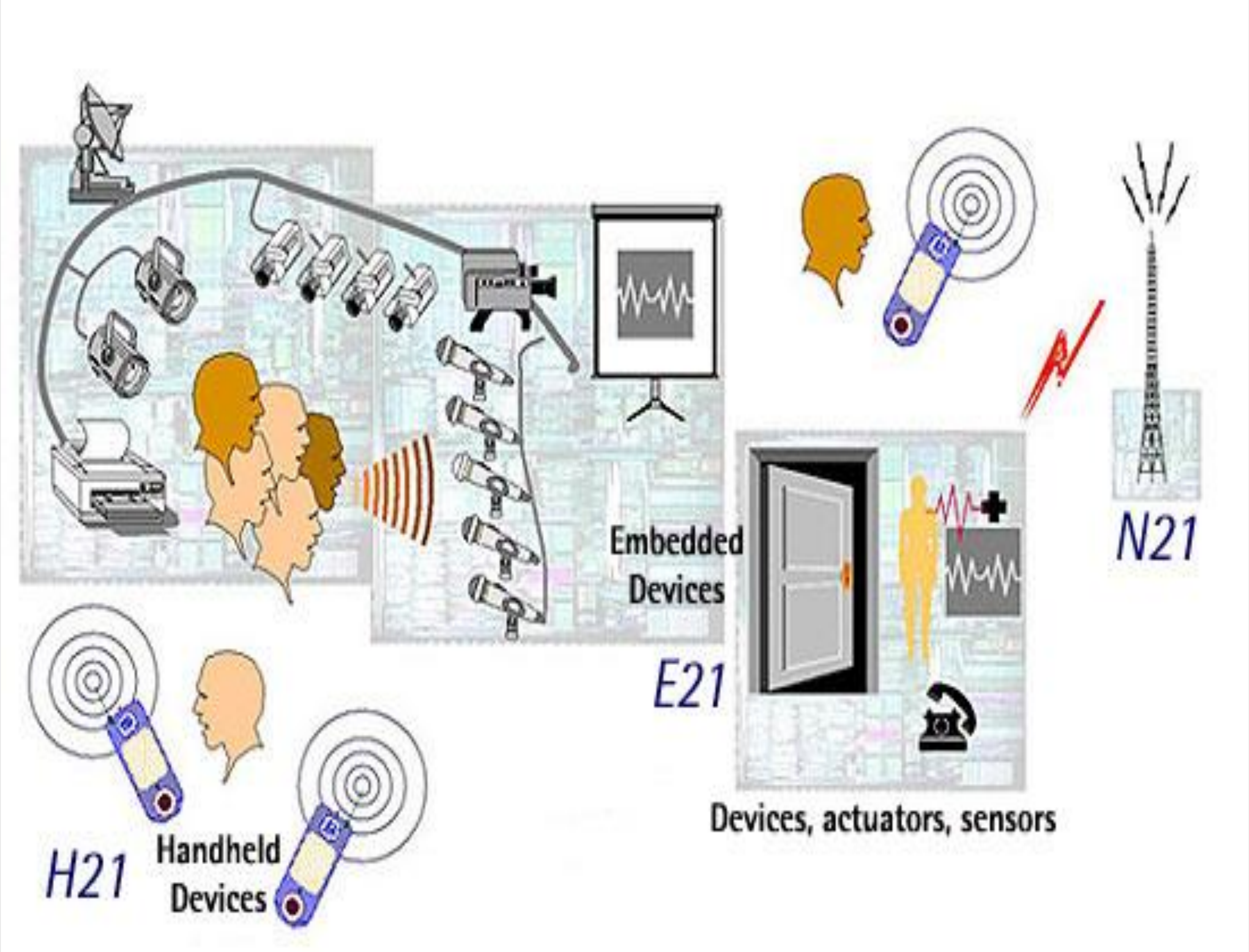
Summit: the World Fastest Computer

- 148.6 petaflops (187.66 petaflop theoretical peak)
- 2,282,544 IBM Power 9 core
- 2,090,880 Nvidia Volta GV100 core
- Power efficiency 11.324gigaflop/watt

Evolution of Computing: The smallest machine becomes even smaller

- Devices become smaller and more powerful
- Devices are coordinated via network
- From "autonomous computing" to coordinated "human-center computing"

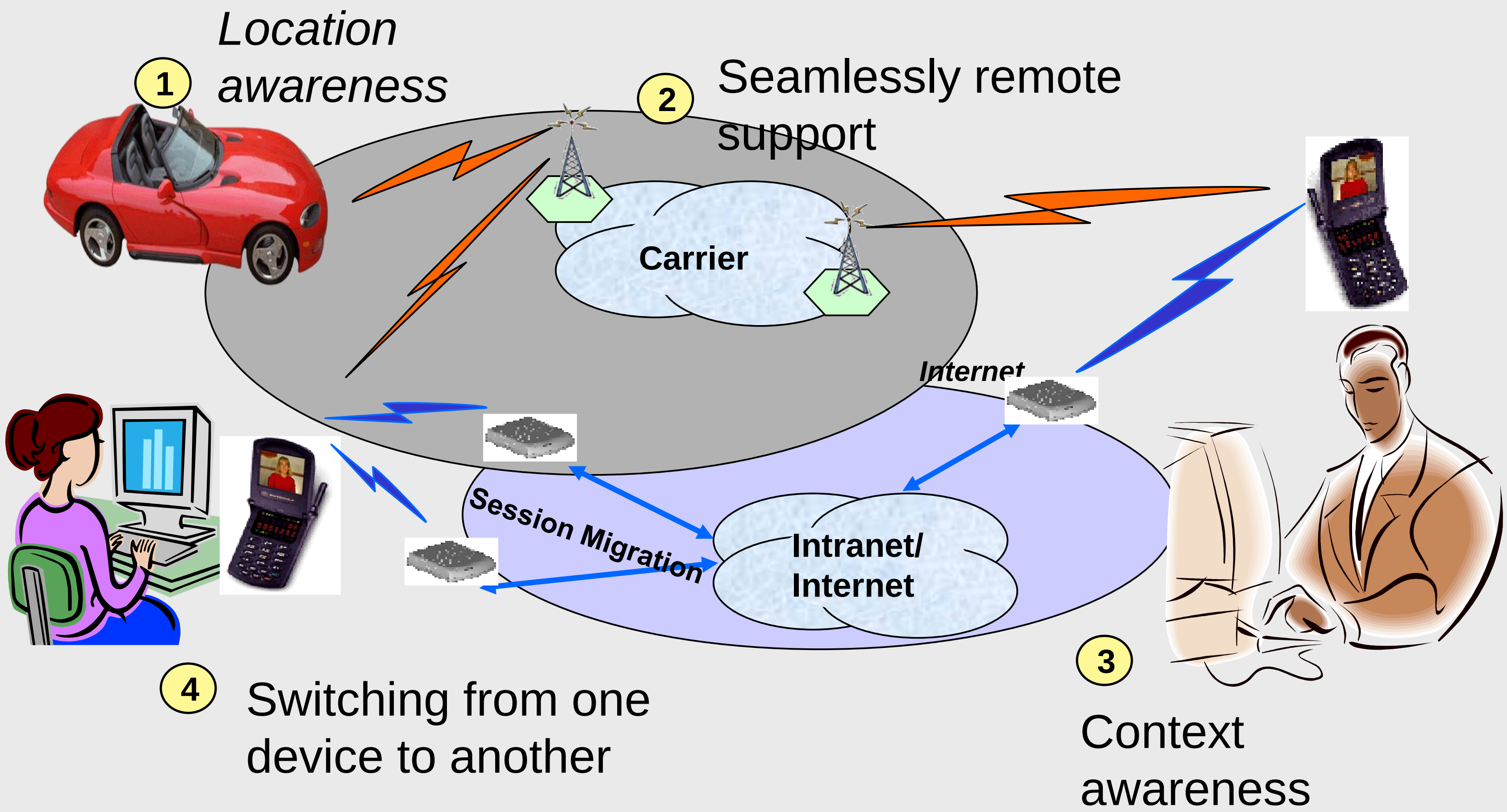
Coordinated Embedded System – Smart Space



Pervasive Computing

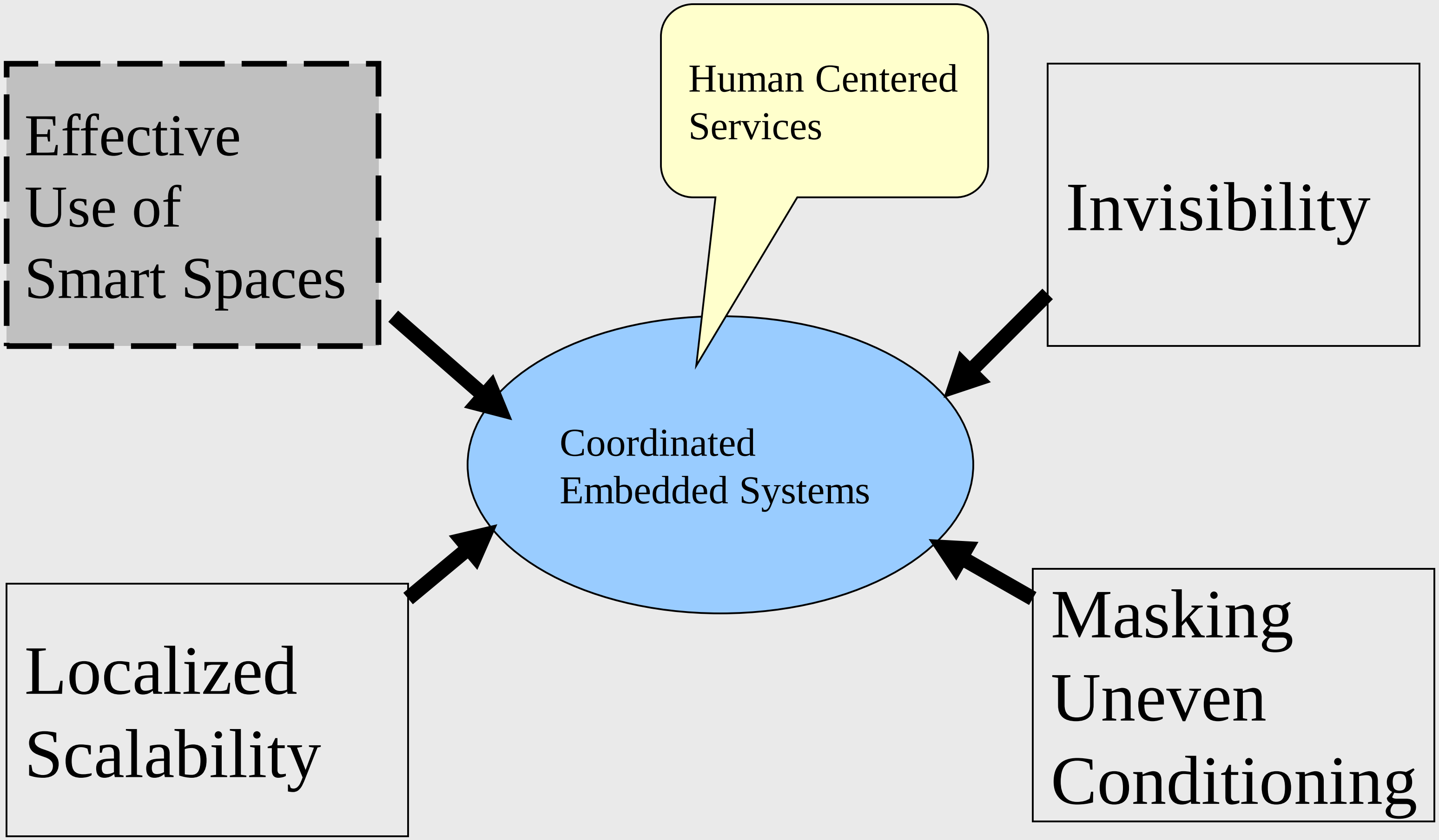
- Computers have become an embed intrinsic part of a sophisticated, networked, pervasive and ubiquitous computing environments around humans.
- Pervasive Computing: create a ubiquitous environment that combines processors and sensors with network technologies (wireless and otherwise) and intelligent software to create an immerse environment to improve life.

Pervasive Computing Applications



Design Challenges

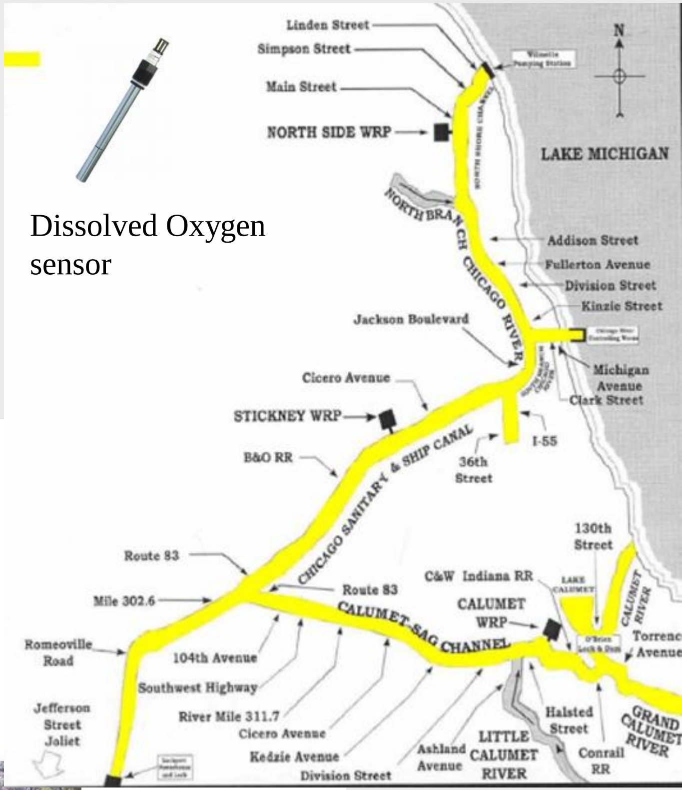
Context awareness and Mobility

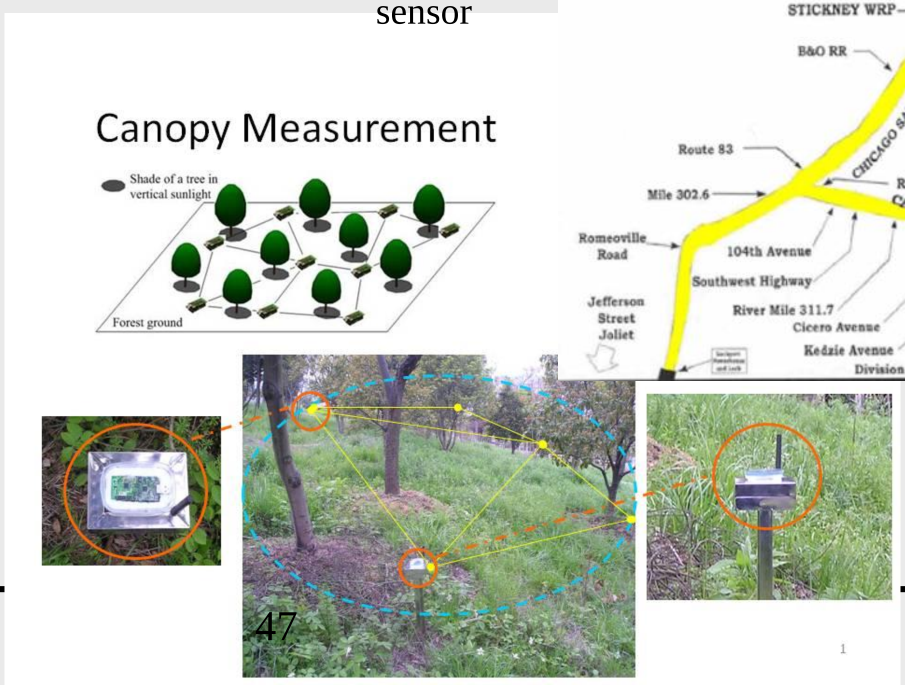


Sensor Network: *Environment monitoring*

- Environment monitoring
  - Chicago Waterway System
  - Ocean Sense
  - GreenObs
- Tracking objects: iLight

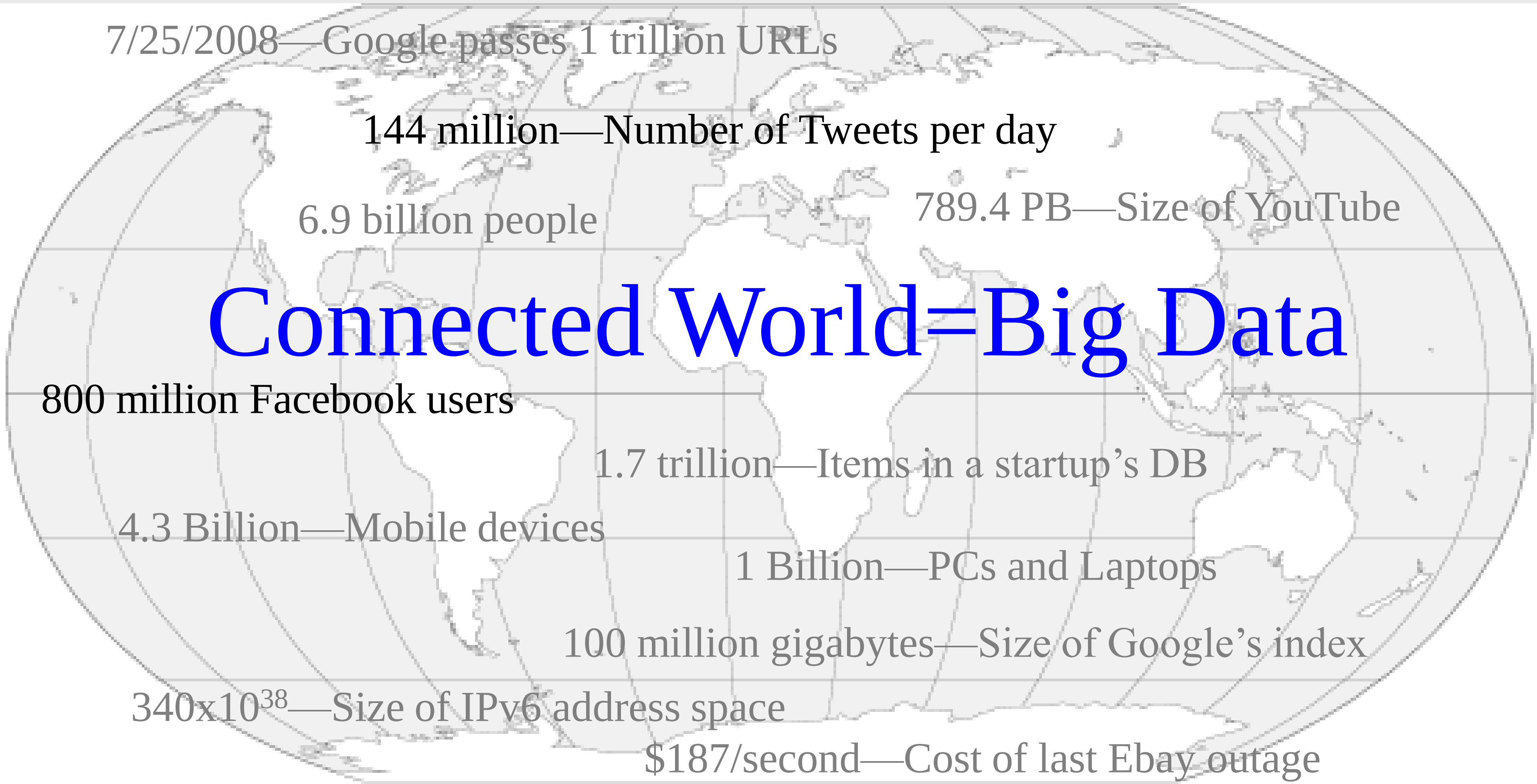
Ammonia sensor





2/4/2011—IPv4 address space exhausted

90PB—Facebook data holdings



7/25/2008—Google passes 1 trillion URLs

144 million—Number of Tweets per day

6.9 billion people

789.4 PB—Size of YouTube

Connected World=Big Data

800 million Facebook users

1.7 trillion—Items in a startup’s DB

4.3 Billion—Mobile devices

1 Billion—PCs and Laptops

100 million gigabytes—Size of Google’s index

$340x10^{38}$—Size of IPv6 address space

$187/second—Cost of last Ebay outage

Courtesy of Ira Hunt

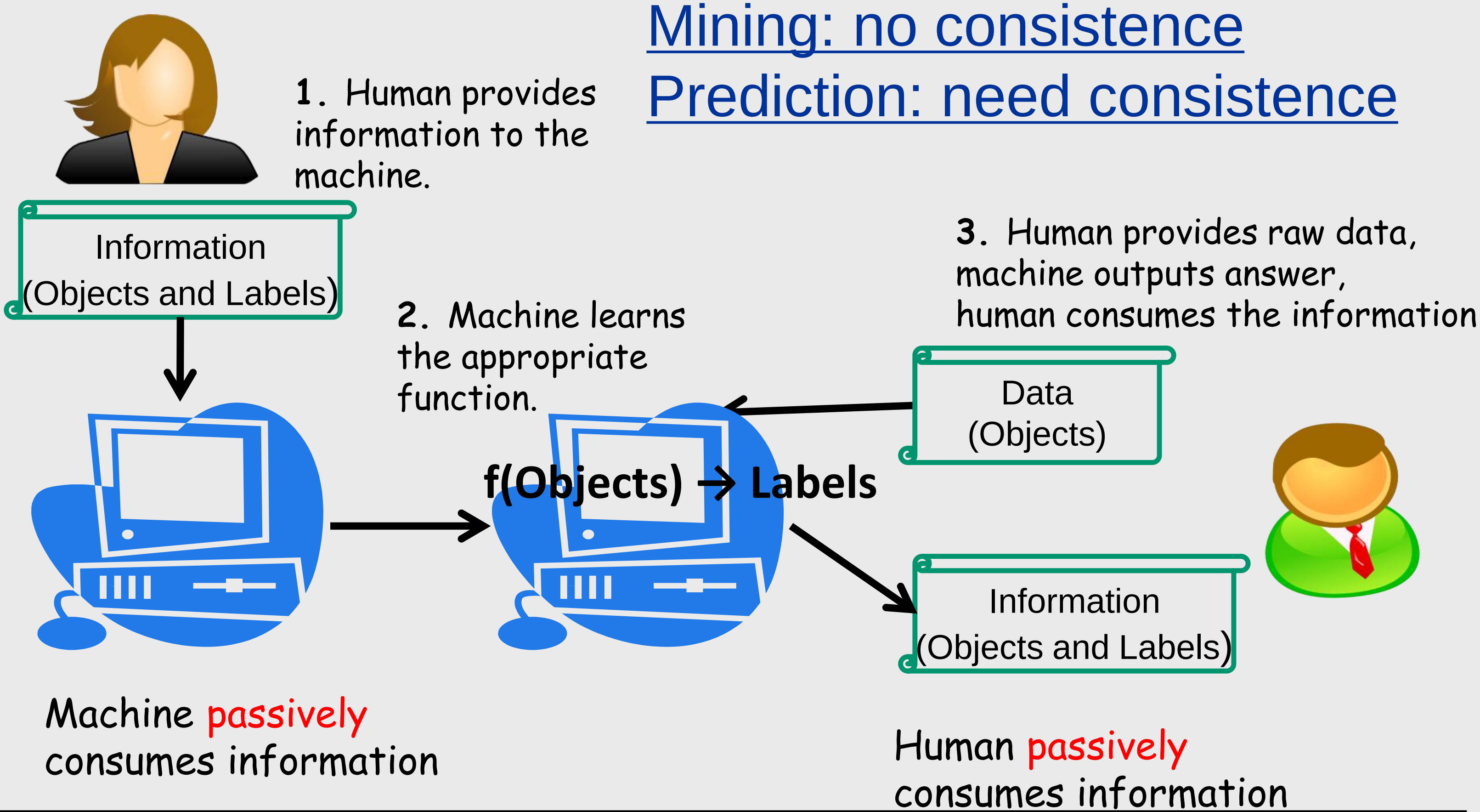
The Surge of Cloud & Big Data

Mimic the electrical power grid

Big Data : discover information/knowledge from data

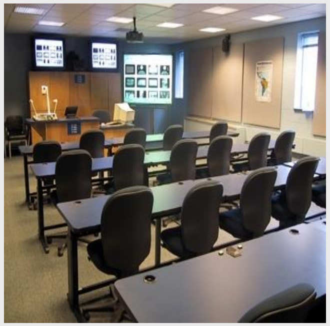
Mining: no consistence

Prediction: need consistence



The View of Future Computing

Human-centered



They are connected to form \`smart space'

Devices become smaller and powerful

Cloud link \`smart spaces' to support \`global smartness'

A device is an entry of the cyber world