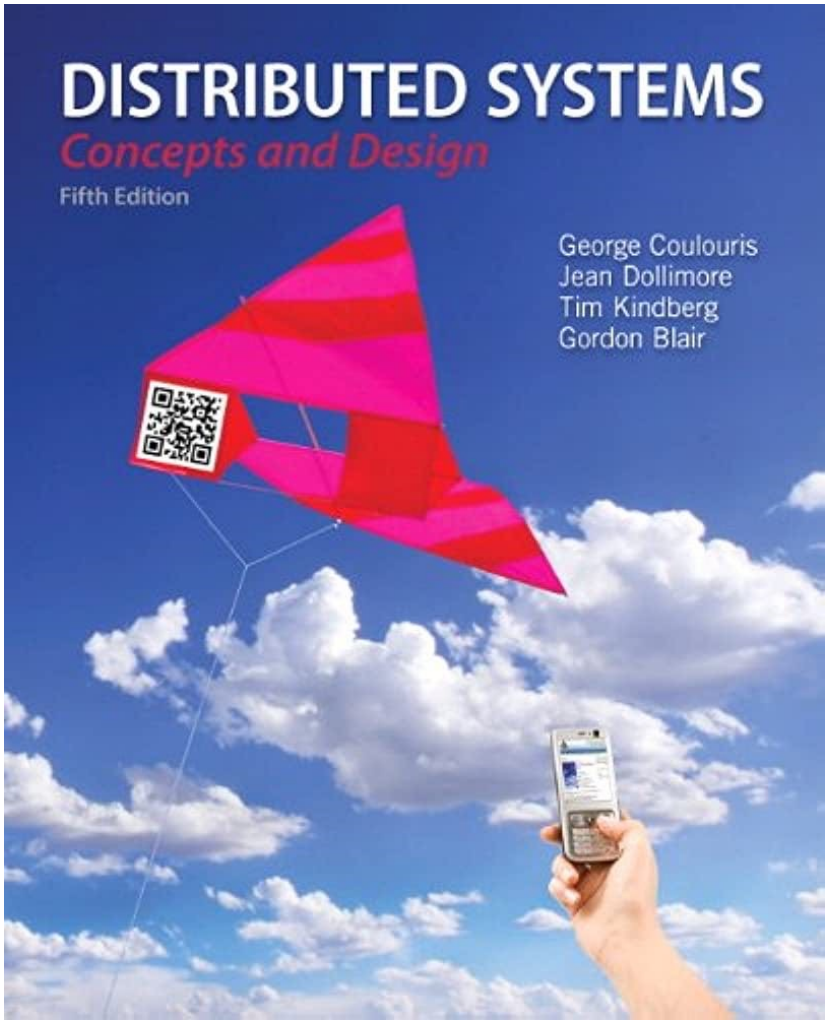
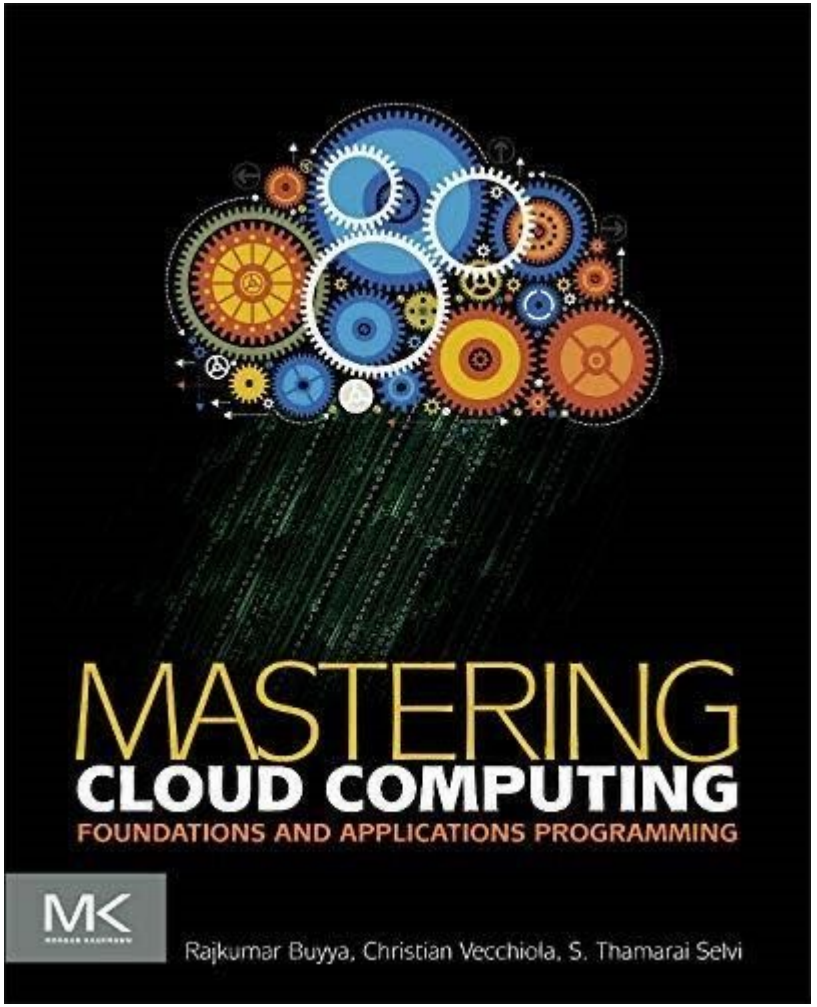
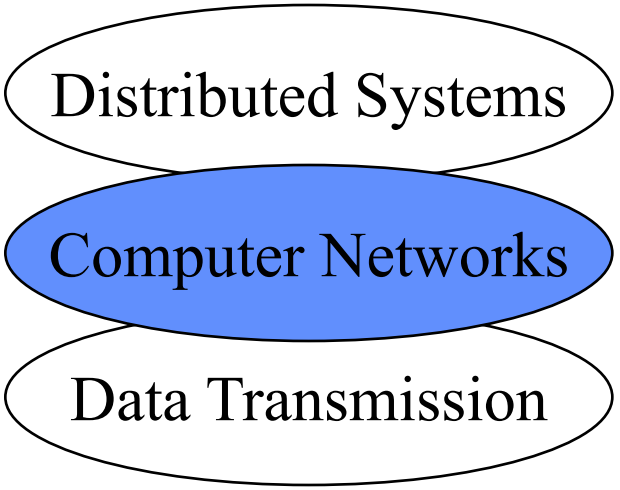
Distributed Systems

Current State of Computing
Characterisation of Distributed Systems

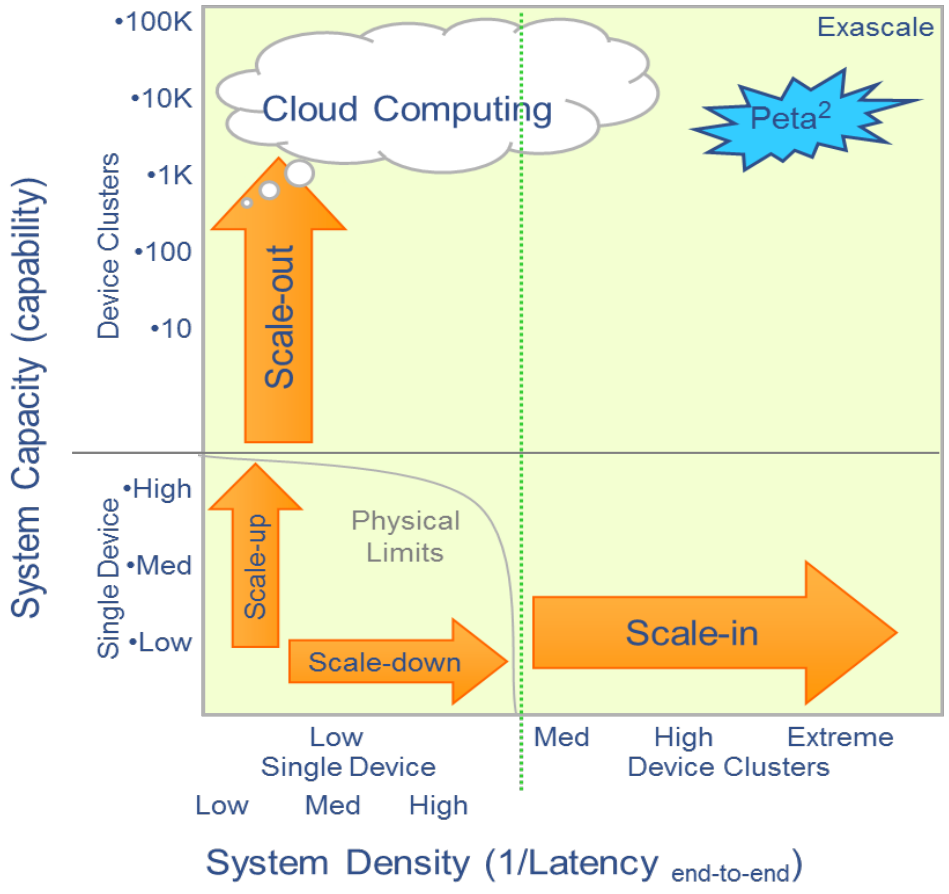

DISTRIBUTED SYSTEMS
Concepts and Design
Fifth Edition
George Coulouris
Jean Dollimore
Tim Kindberg
Gordon Blair

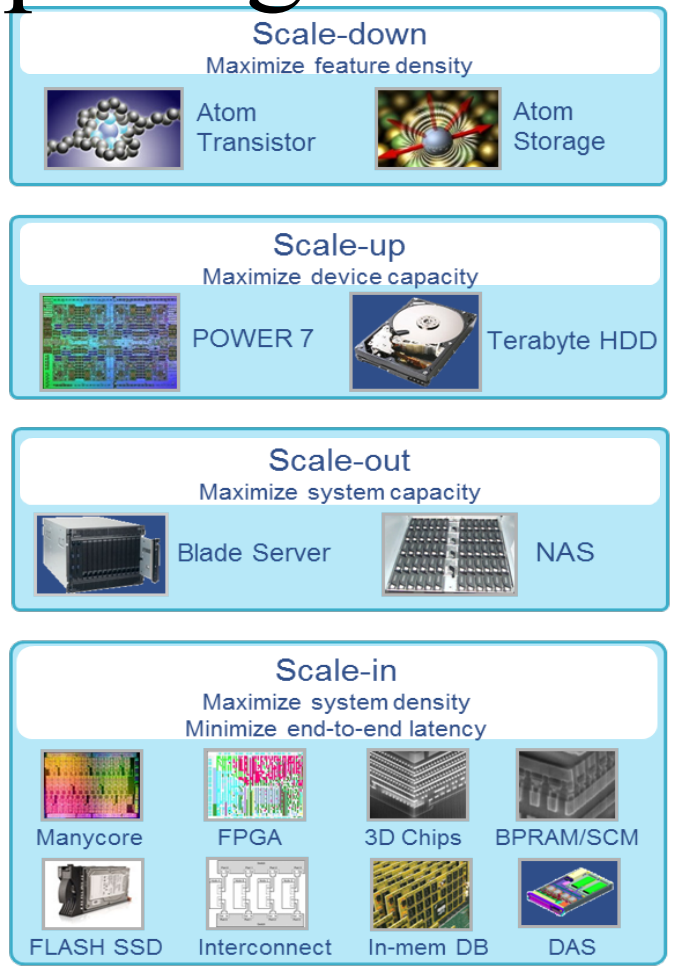

MASTERING
CLOUD COMPUTING
FOUNDATIONS AND APPLICATIONS PROGRAMMING
MK
Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi

- Computer Network: A collection of *interconnected autonomous* computers
  - Generality: Built from general purpose hardware - not optimised for any particular application or data type
- Computer Network vs. Distributed System
  - Transparency
  - DS is a software system that runs on top of CN

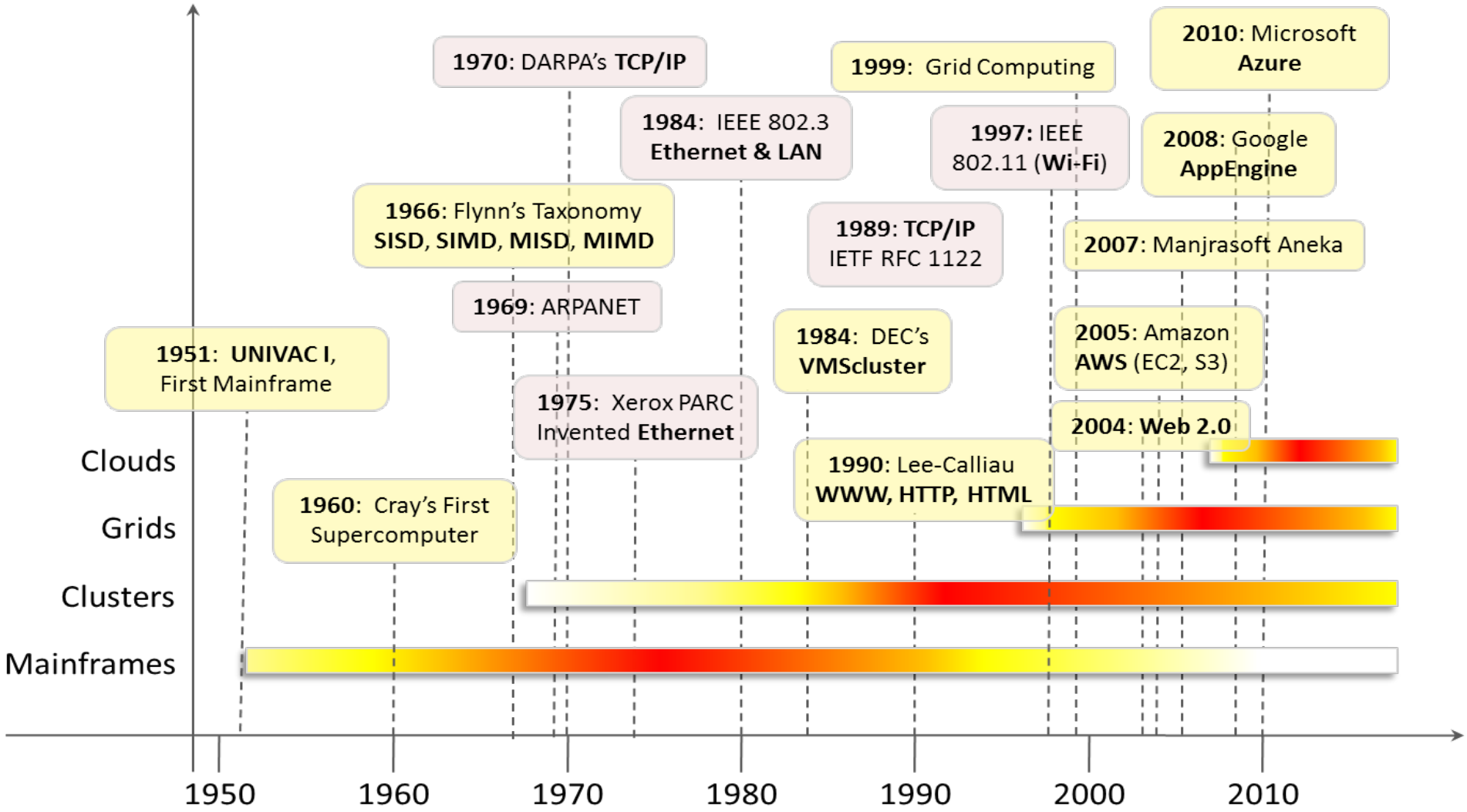


Trends in Computing





Source: IBM

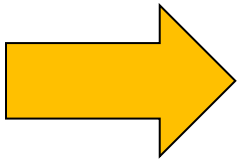

1951: UNIVAC I, First Mainframe
1960: Cray's First Supercomputer
1966: Flynn's Taxonomy SISD, SIMD, MISD, MIMD
1969: ARPANET
1970: DARPA's TCP/IP
1975: Xerox PARC Invented Ethernet
1984: IEEE 802.3 Ethernet & LAN
1984: DEC's VMScluster
1989: TCP/IP IETF RFC 1122
1990: Lee-Calliau WWW, HTTP, HTML
1997: IEEE 802.11 (Wi-Fi)
1999: Grid Computing
2004: Web 2.0
2005: Amazon AWS (EC2, S3)
2007: Manjrasoft Aneka
2008: Google AppEngine
2010: Microsoft Azure
Clouds
Grids
Clusters
Mainframes
1950
1960
1970
1980
1990
2000
2010

Manchester ATLAS 1962

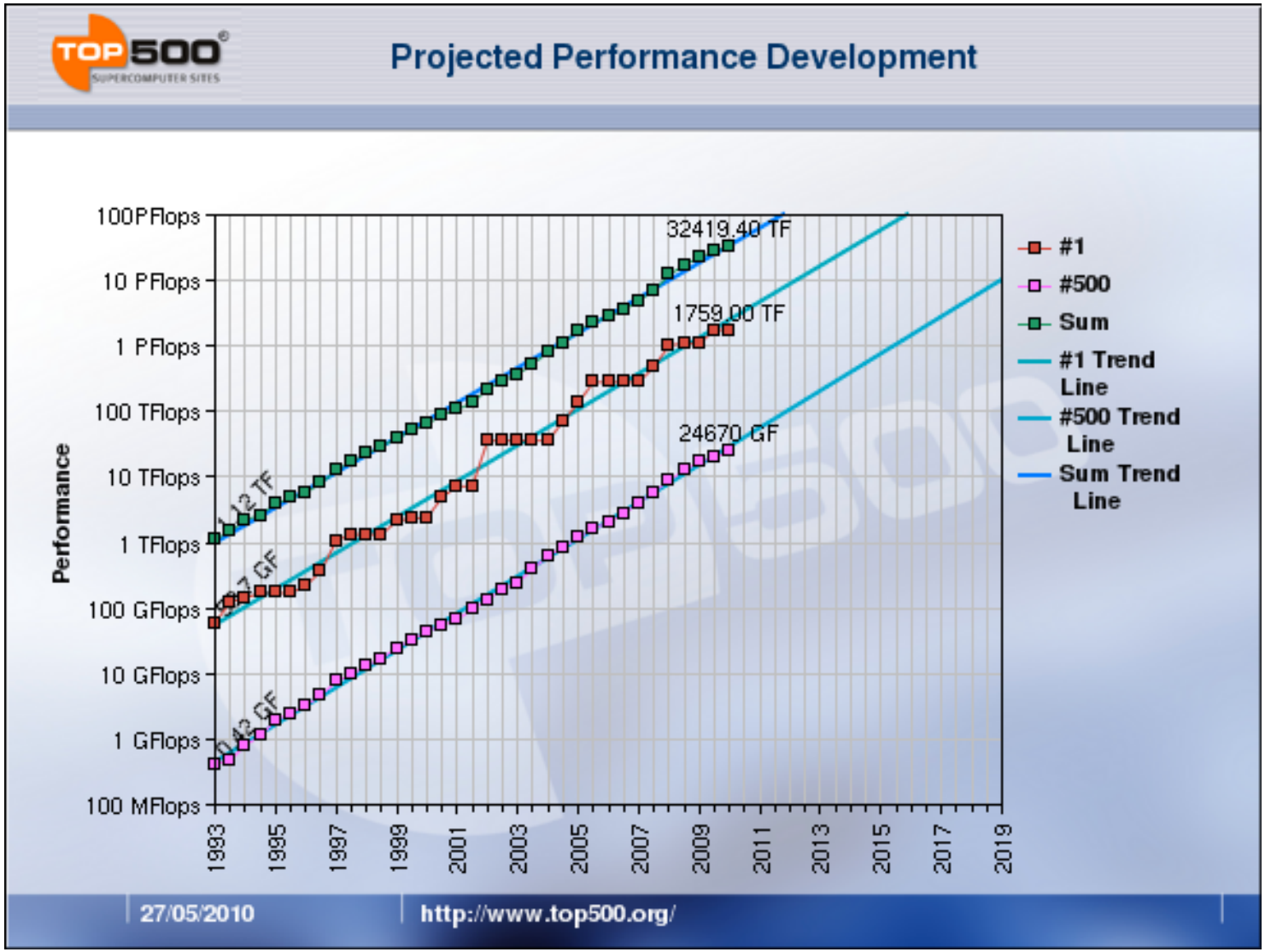
- First Supercomputer in the world
- (Cray CDC 6600 1964)
- Equivalent to four IBM 7094s
- Asynchronous processor
- Associative Memory


THE
EXASCALE CHALLENGE

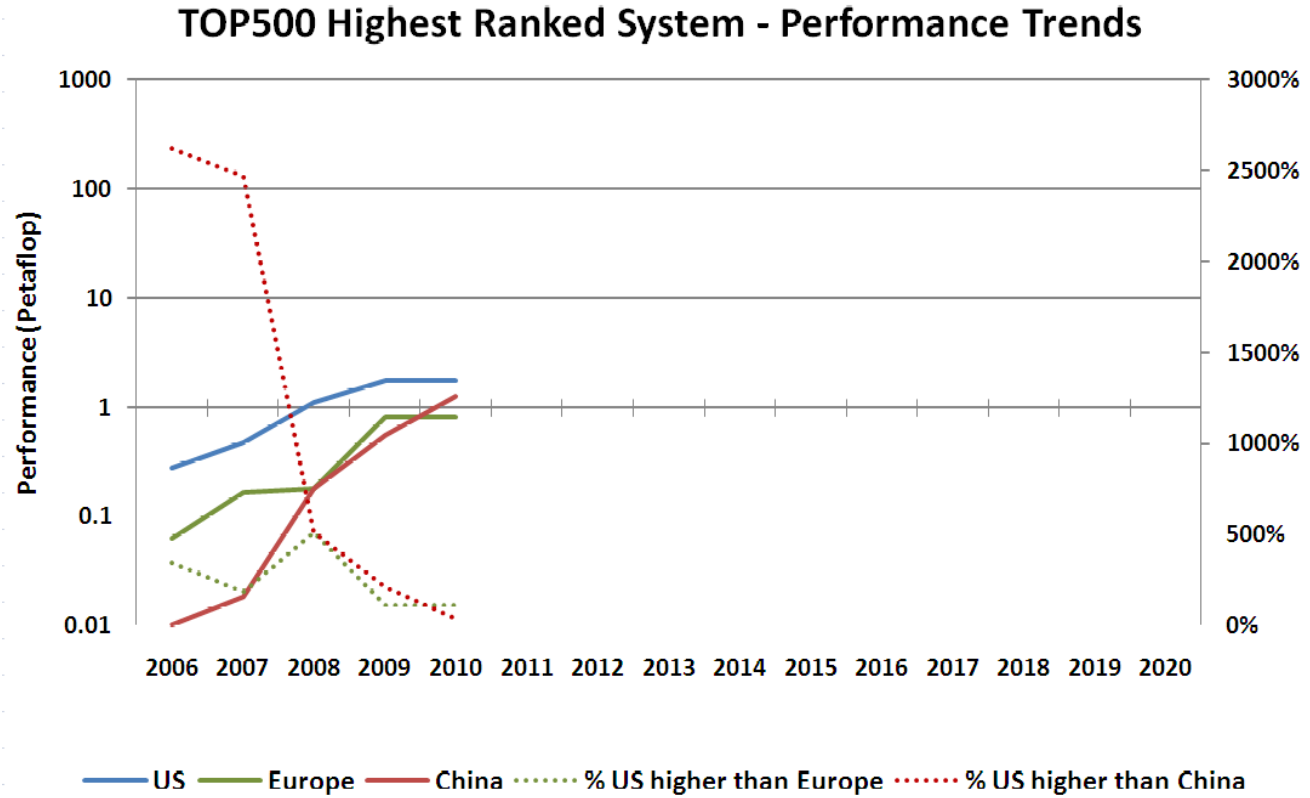
- Petascale Computing (10^{15})
- Multicore computing
  - 1-24 cores commodity architectures
  - 100+cores proprietary architectures
  - 400+ GPU cores



- Exascale Computing (10^{18})
- Manycore computing
  - ~1000-core commodity architectures (heterogeneous, merged with GPUs etc)
- 1M nodes
- 1B processor cores


TOP500
SUPERCOMPUTER SITES
Projected Performance Development
100 PFlops
10 PFlops
1 PFlops
100 TFlops
10 TFlops
1 TFlops
100 GFlops
10 GFlops
1 GFlops
100 MFlops
Performance
1993
1995
1997
1999
2001
2003
2005
2007
2009
2011
2013
2015
2017
2019
32419.40 TF
1759.00 TF
24670 GF
1.12 TF
59.7 GF
0.42 GF
#1
#500
Sum
#1 Trend Line
#500 Trend Line
Sum Trend Line
27/05/2010
http://www.top500.org/

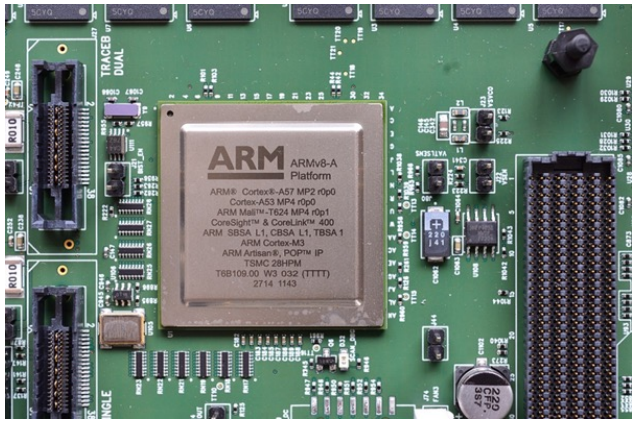
Towards Exascale Computing



- In 2006, the performance leadership of the US's highest performance system was more than 2500% higher than the best system in China, in the first half of 2010, the gap decreased to less than 40% (**Source: US HPC Advisory Council**)

TOP 500
The List.

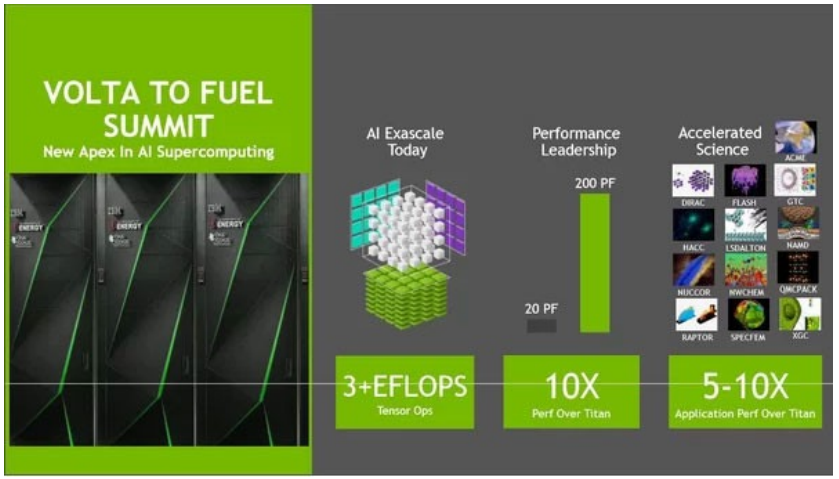
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646

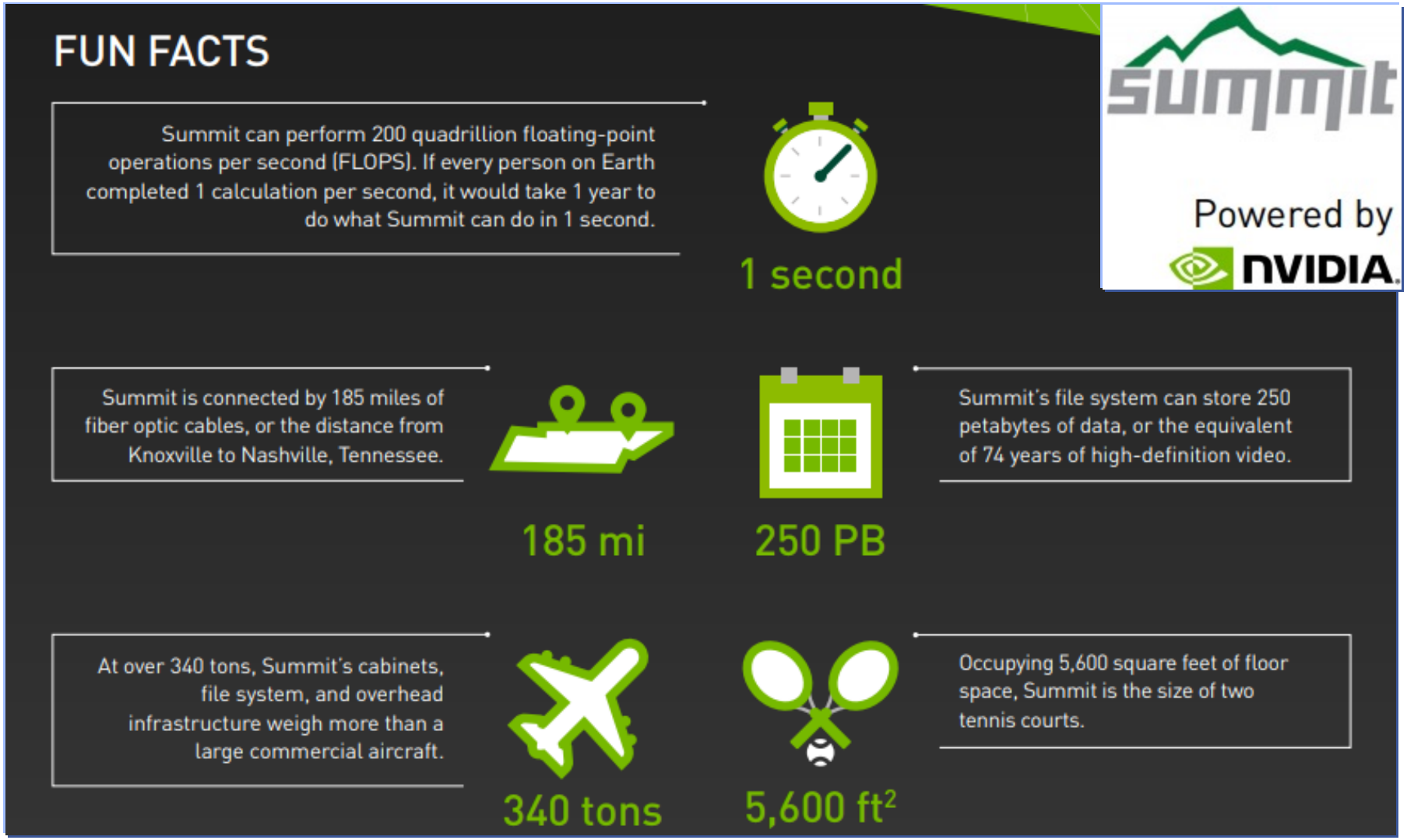


- Supercomputer Fugaku
- RIKEN Center for Computational Science (R-CCS)
- Fujitsu A64FX: 64-bit ARM, SIMD instruction set with 512-bit vector implementation
- 158,976 nodes
- 7,299,072 cores
- 415 PetaFLOPS

- Oak Ridge National Laboratory in Tennessee
- As big as two tennis courts
- **9,216 IBM Power9** processor chips running at 3.1GHz, and each of those has 22 processing core
- Connected to each pair of Power9 chips are six Nvidia Tesla V100 graphics chips
- **27,648 V100**
- 200 PetaFLOPS

- 4,608 interconnected computer nodes housed in refrigerator-sized cabinets
- liquid-cooled by pumping 4,000 gallons of water per minute through the system.




FUN FACTS
Summit can perform 200 quadrillion floating-point operations per second (FLOPS). If every person on Earth completed 1 calculation per second, it would take 1 year to do what Summit can do in 1 second.
1 second
summit
Powered by
NVIDIA
Summit is connected by 185 miles of fiber optic cables, or the distance from Knoxville to Nashville, Tennessee.
185 mi
250 PB
Summit's file system can store 250 petabytes of data, or the equivalent of 74 years of high-definition video.
At over 340 tons, Summit's cabinets, file system, and overhead infrastructure weigh more than a large commercial aircraft.
340 tons
5,600 ft²
Occupying 5,600 square feet of floor space, Summit is the size of two tennis courts.

- SIERRA
- Lawrence Livermore National Laboratory
- IBM POWER9 CPUs in conjunction with Nvidia Tesla V100 GPUs
- 1,572,480 cores
- 125 PetaFLOPS

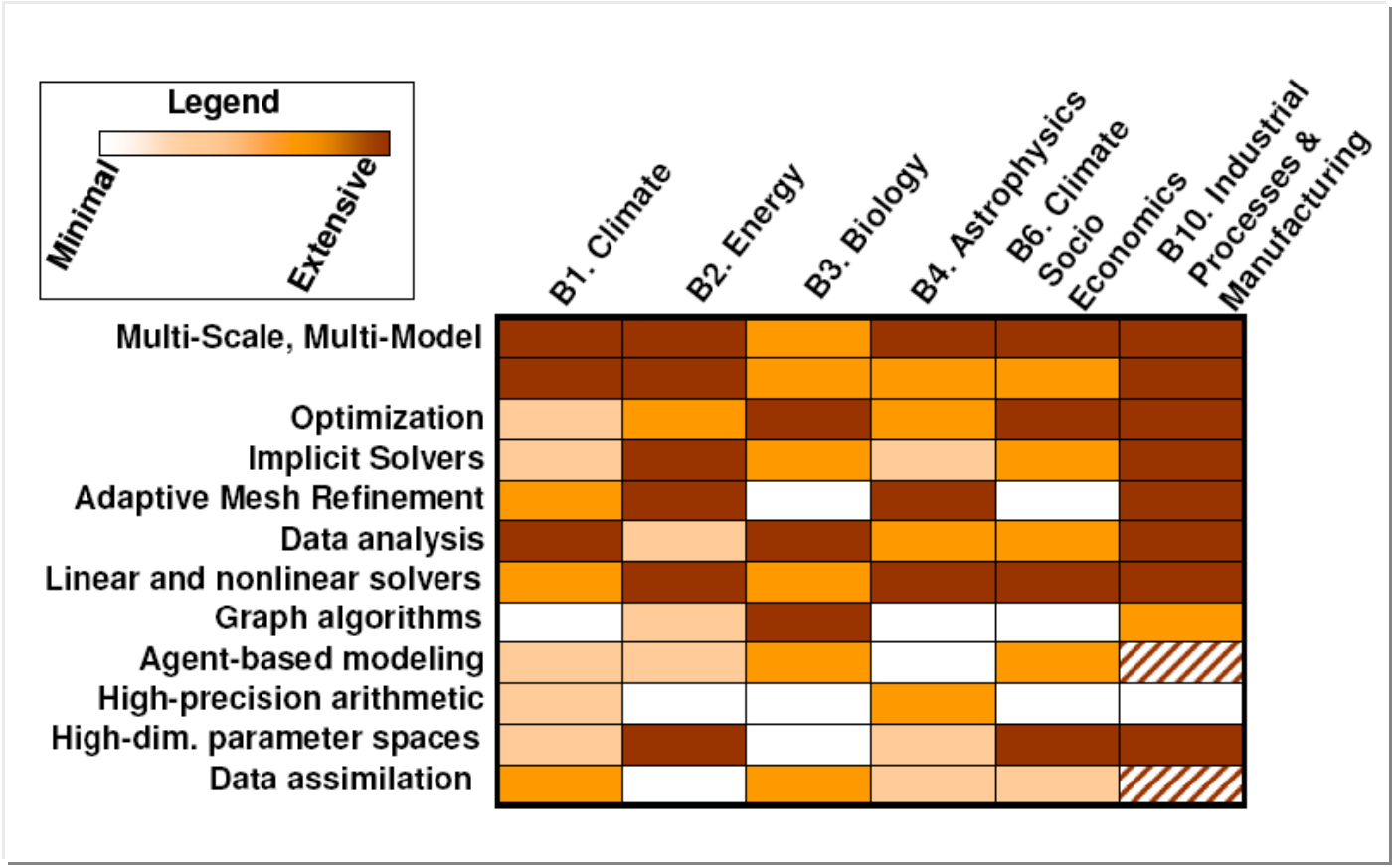
Sunway TaihuLight - 神威·太湖之光

- 40,960 Chinese-designed Sunway SW26010 manycore 64-bit RISC processors based on the architecture
- Each processor chip contains 256 processing cores, and an additional four auxiliary cores for system management
- A total of 10,649,600 CPU cores across the entire system
- 93 PetaFLOPS

Sunway TaihuLight - 神威·太湖之光

- 40,960 Chinese-designed Sunway SW26010 manycore 64-bit RISC processors based on the architecture
- Each processor chip contains 256 processing cores, and an additional four auxiliary cores for system management
- A total of 10,649,600 CPU cores across the entire system
- 93 PetaFLOPS

Drivers: Traditional Workloads



Source: ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, DARPA IPTO

Drivers: Data

- Data volume, velocity, and variety is growing at an astounding rate with a full 90% of the world's data less than two years old.
- "Big Data is big. It's 2.5 quintillion bytes of data every day big."
- Almost 90% of this data is unstructured

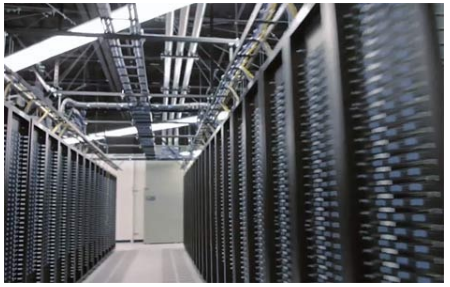
Homeland Security

•600,000 records/sec

Telco Promotions

•100,000 records/sec

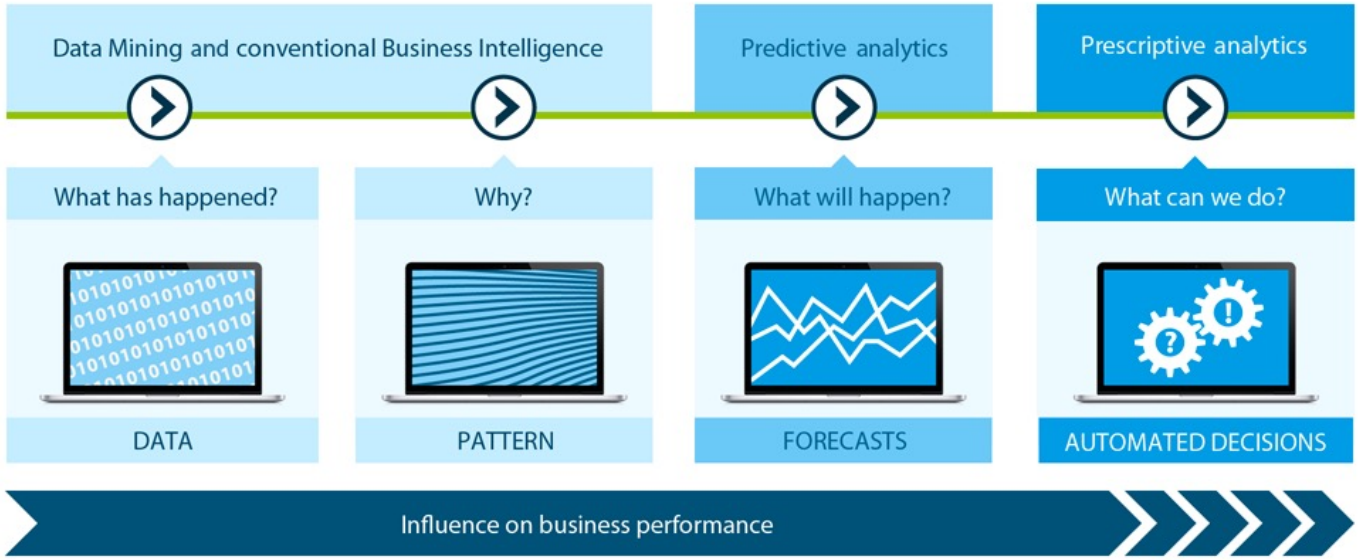
- 300M users
- 10000 data centres
- 30000 servers
- 25 Terabytes of Log Data - daily



Traffic

•250000 GPS probes/sec

Analytics


Data Mining and conventional Business Intelligence
Predictive analytics
Prescriptive analytics
What has happened?
Why?
What will happen?
What can we do?
DATA
PATTERN
FORECASTS
AUTOMATED DECISIONS
Influence on business performance


Regular Analytics
Big Data
Descriptive Analytics
What happened?
Diagnostic Analytics
Why did it happen?
Predictive Analytics
What will happen?
Prescriptive Analytics
How can we make it happen?
Information
Hindsight
Insight
Foresight
Optimization

Technological Advances: Moore's Law

- Moore's law predicts a 60% annual increase in the number of transistors that can be put on a chip

Chip	Date	MHz	Transistors	Memory	Notes
4004	4/1971	0.108	2,300	640	First microprocessor on a chip
8008	4/1972	0.108	3,500	16 KB	First 8-bit microprocessor
8080	4/1974	2	6,000	64 KB	First general-purpose CPU on a chip
8086	6/1978	5-10	29,000	1 MB	First 16-bit CPU on a chip
8088	6/1979	5-8	29,000	1 MB	Used in IBM PC
80286	2/1982	8-12	134,000	16 MB	Memory protection present
80386	10/1985	16-33	275,000	4 GB	First 32-bit CPU
80486	4/1989	25-100	1.2M	4 GB	Built-in 8K cache memory
Pentium	3/1993	60-233	3.1M	4 GB	Two pipelines; later models had MMX
Pentium Pro	3/1995	150-200	5.5M	4 GB	Two levels of cache built in
Pentium II	5/1997	233-400	7.5M	4 GB	Pentium Pro plus MMX

Figure 1-10. The Intel CPU family. Clock speeds are measured in MHz (megahertz) where 1 MHz is 1 million cycles/sec.

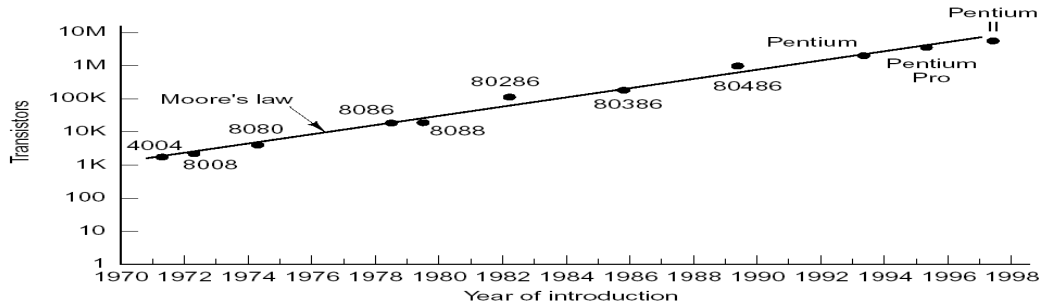


Figure 1-11. Moore's law for CPU chips.

MIT Technology Review

Computing / Microchips

Moore's Law Is Dead. Now What?

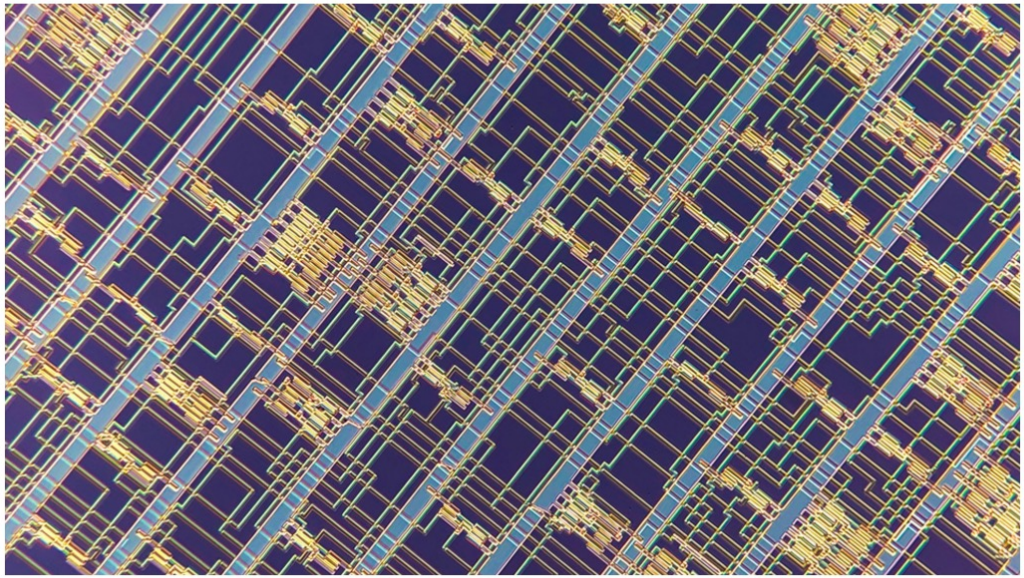
Shrinking transistors have powered 50 years of advances in computing—but now other ways must be found to make computers more capable.

by **Tom Simonite** May 13, 2016

MIT Technology Review

Sign in Subscribe

Topics Magazine Newsletters Events



FELICE FRANKEL

Computing / Microchips

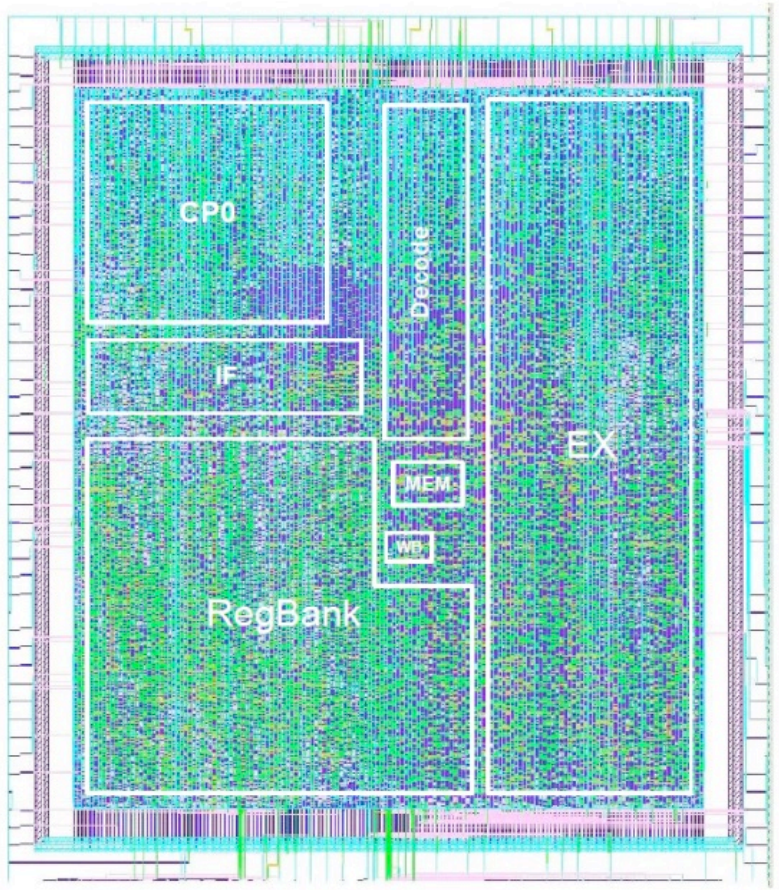
The world's most advanced nanotube computer may keep Moore's Law alive

MIT researchers have found new ways to cure headaches in manufacturing carbon nanotube processors, which are faster and less power hungry than silicon chips.

by **Martin Giles** Aug 30, 2019

Energy: Cooling

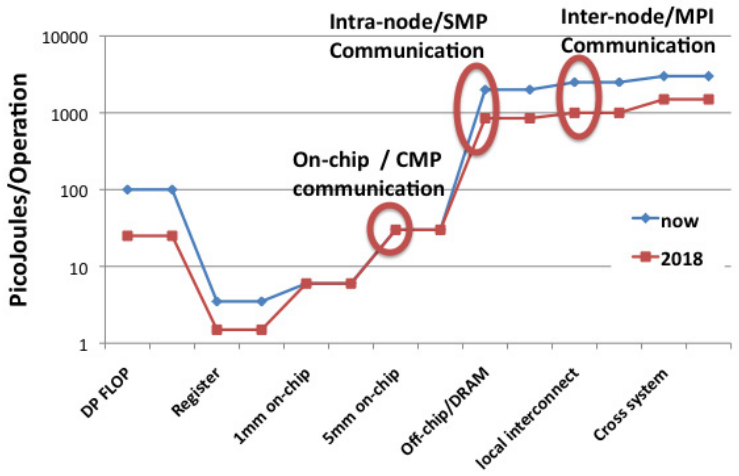
Energy: Processing and Clocks



- New VLSI Paradigms
- Asynchronous Hardware
- GALS: Globally Asynchronous Locally Synchronous

Source: Zhang and Theodoropoulos, SAMIPS, A Synthesisable MIPS Processor

Energy: Data



- Energy cost of data movement relative to the cost of a flop

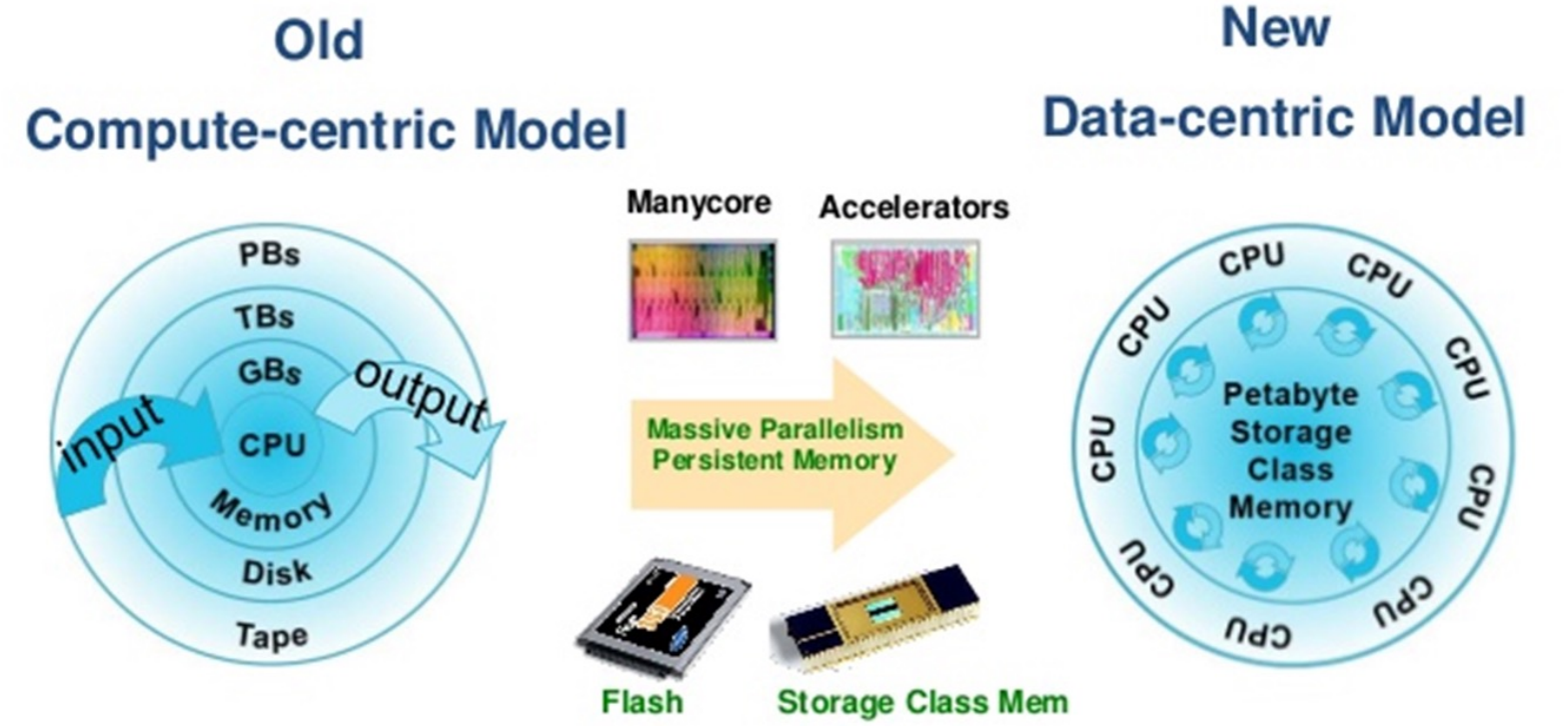
- (Source: Shalf et al., Exascale Computing Technology Challenges, 2011)

$$Power \approx B \times l^2 / A$$

- Power consumed increases proportionally to the bit-rate, so as we move to ultrahigh-bandwidth links, the power requirements are crucial factor.
- Improvements in chip lithography (making smaller wires) will not improve the energy efficiency or data carrying capacity of electrical wires.
- Without major breakthroughs in packaging technology or photonics, it will not be feasible to support globally flat bandwidth across a system
- Optical technology not a generic solution

Energy: Data

- Power consumption is highly distance-dependent, so bandwidth is likely to become increasingly localised as power becomes a more difficult problem.



Source: IBM

SpiNNaker: ‘Human brain' supercomputer

- 57,600 **ARM9** processors
- each with 18 cores
- Total: 1,036,800 cores and over 7 TB of RAM

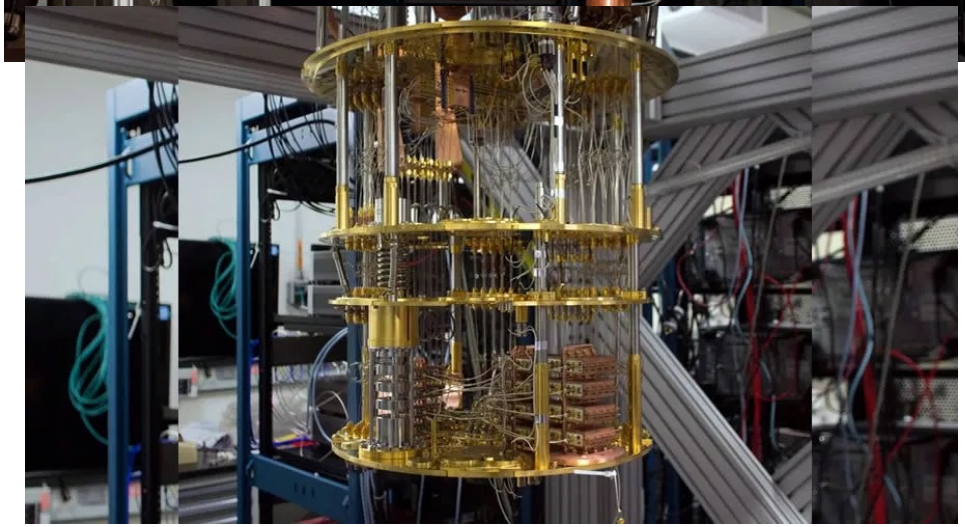
“SpiNNaker completely re-thinks the way conventional computers work. We’ve essentially created a machine that works more like a brain than a traditional computer”

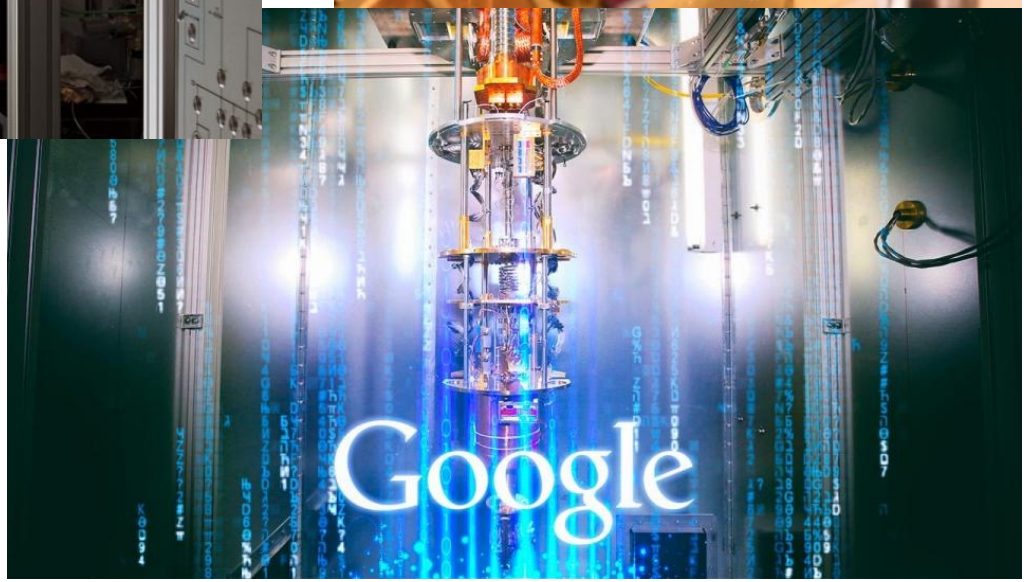
Steve Furber, ICL Professor of Computer Engineering, Manchester University


IBM Q

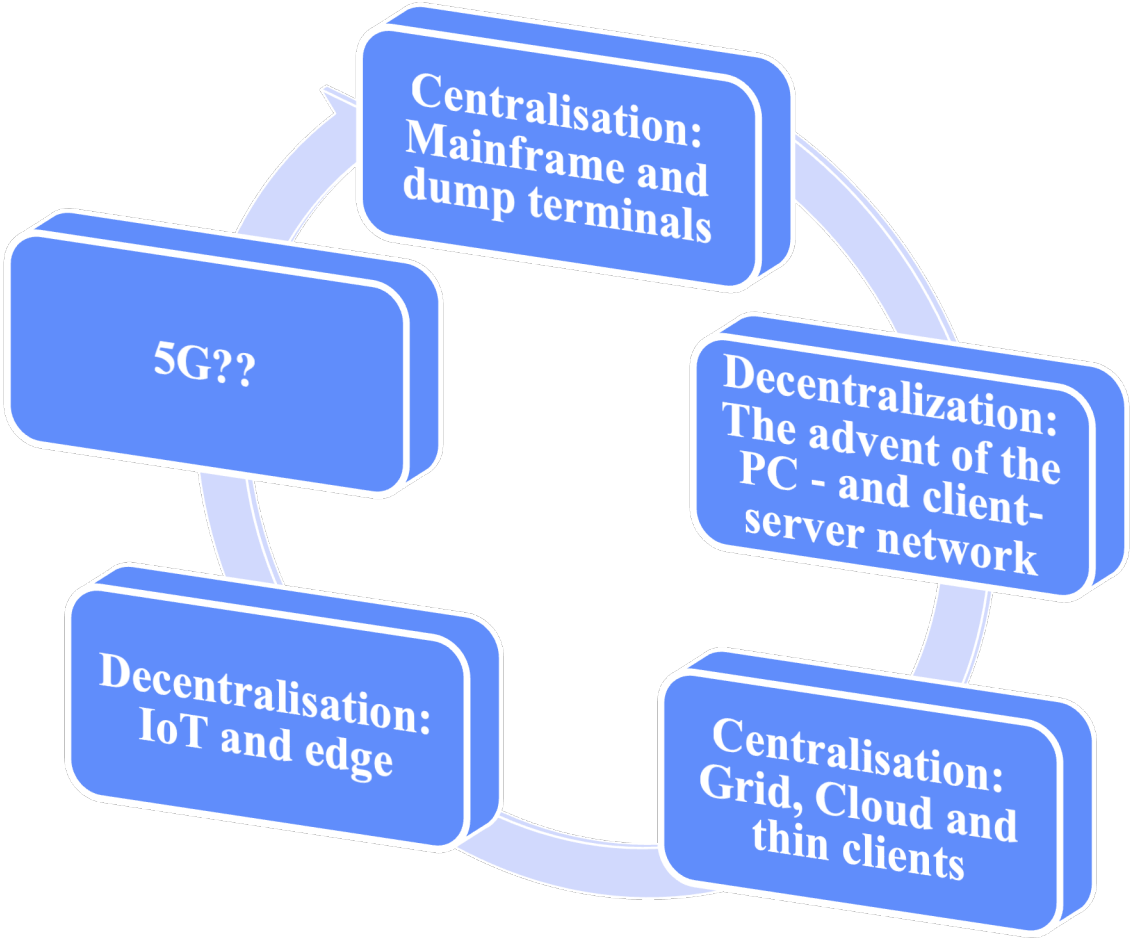

D-Wave 2000Q
D-Wave
D-Wave 2000Q


Google
Bristlecone



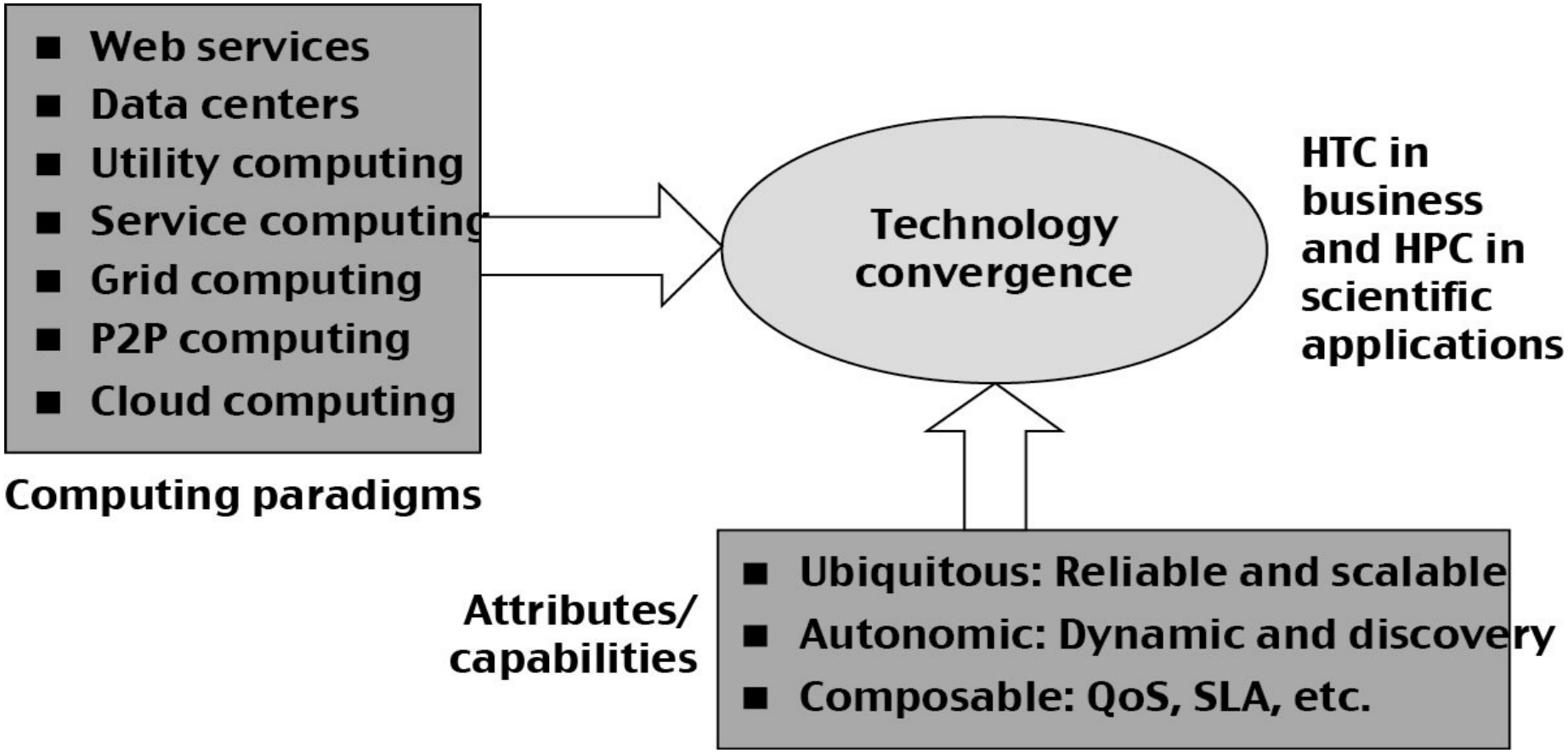

Google

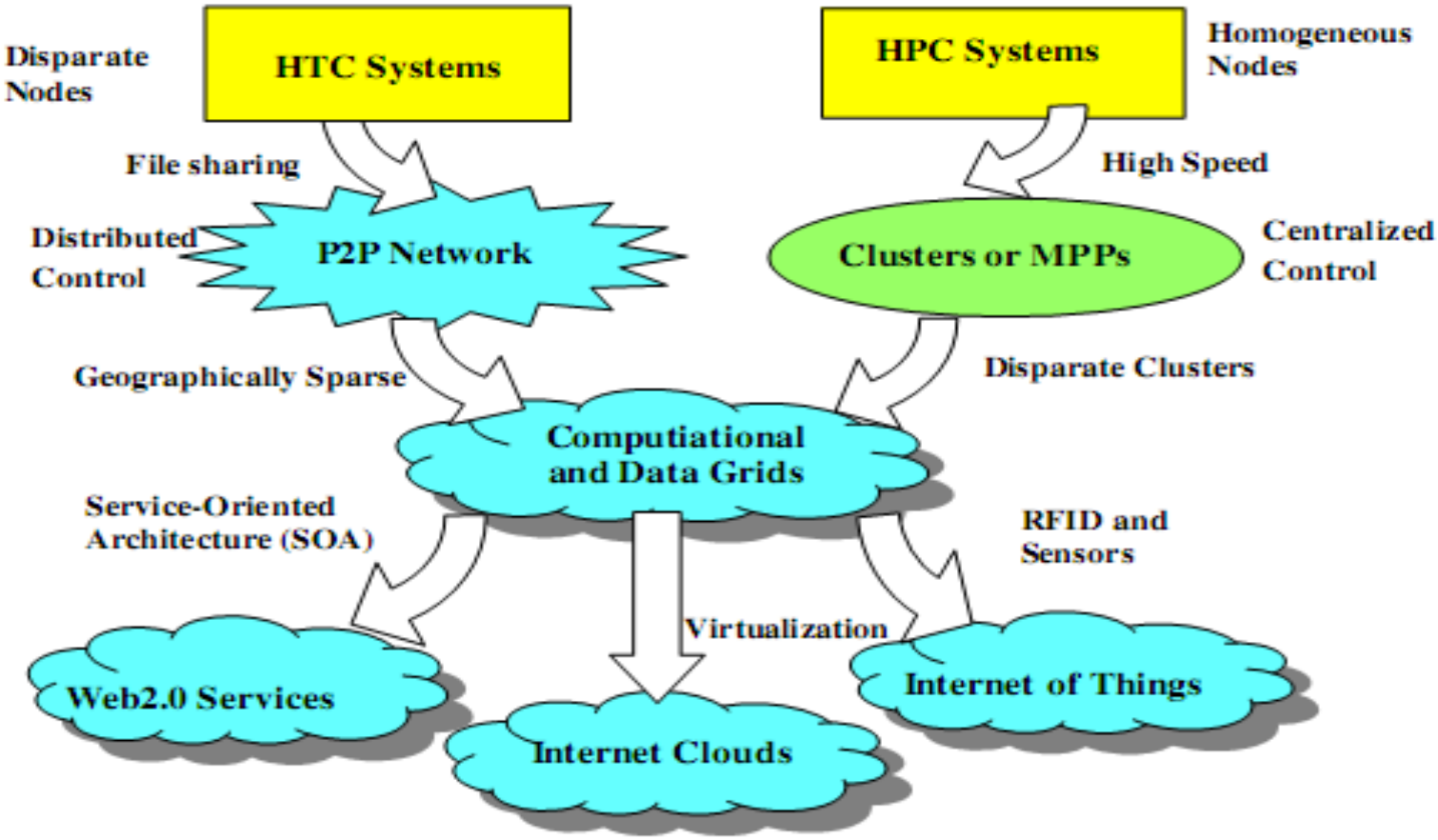
Systems Cycle



From Desktop/HPC/Grids to Internet Clouds in 30 Years

- HPC moving from centralized supercomputers to geographically distributed desktops, desksides, clusters, and grids to clouds over last 30 years
- R/D efforts on HPC, clusters, Grids, P2P, and virtual machines has laid the foundation of cloud computing that has been greatly advocated since 2007
- Location of computing infrastructure in areas with lower costs in hardware, software, datasets, space, and power requirements – moving from desktop computing to datacenter-based clouds


Web services
Data centers
Utility computing
Service computing
Grid computing
P2P computing
Cloud computing
Computing paradigms
Technology convergence
HTC in business and HPC in scientific applications
Attributes/ capabilities
Ubiquitous: Reliable and scalable
Autonomic: Dynamic and discovery
Composable: QoS, SLA, etc.



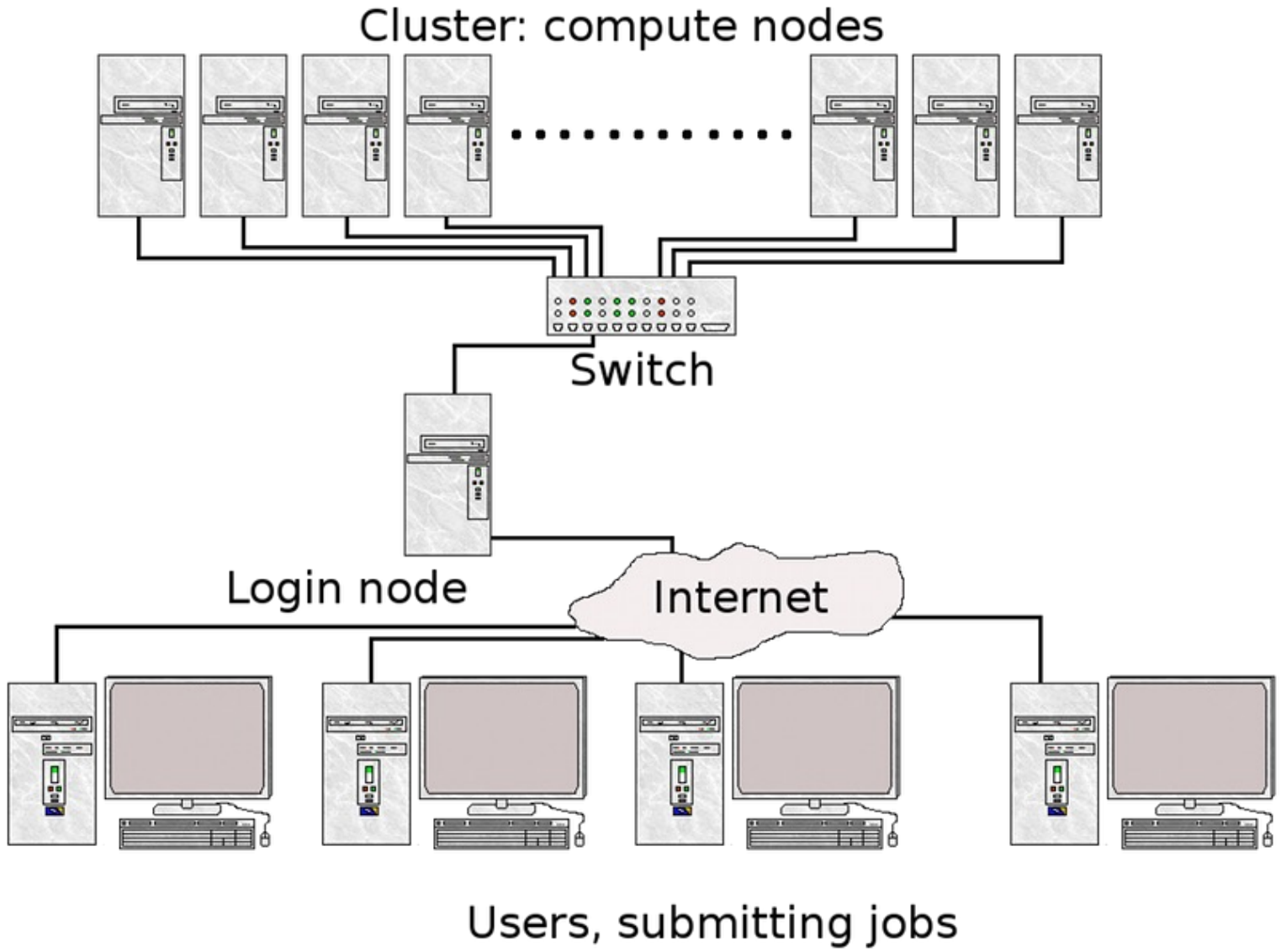
HPC: High-Performance Computing

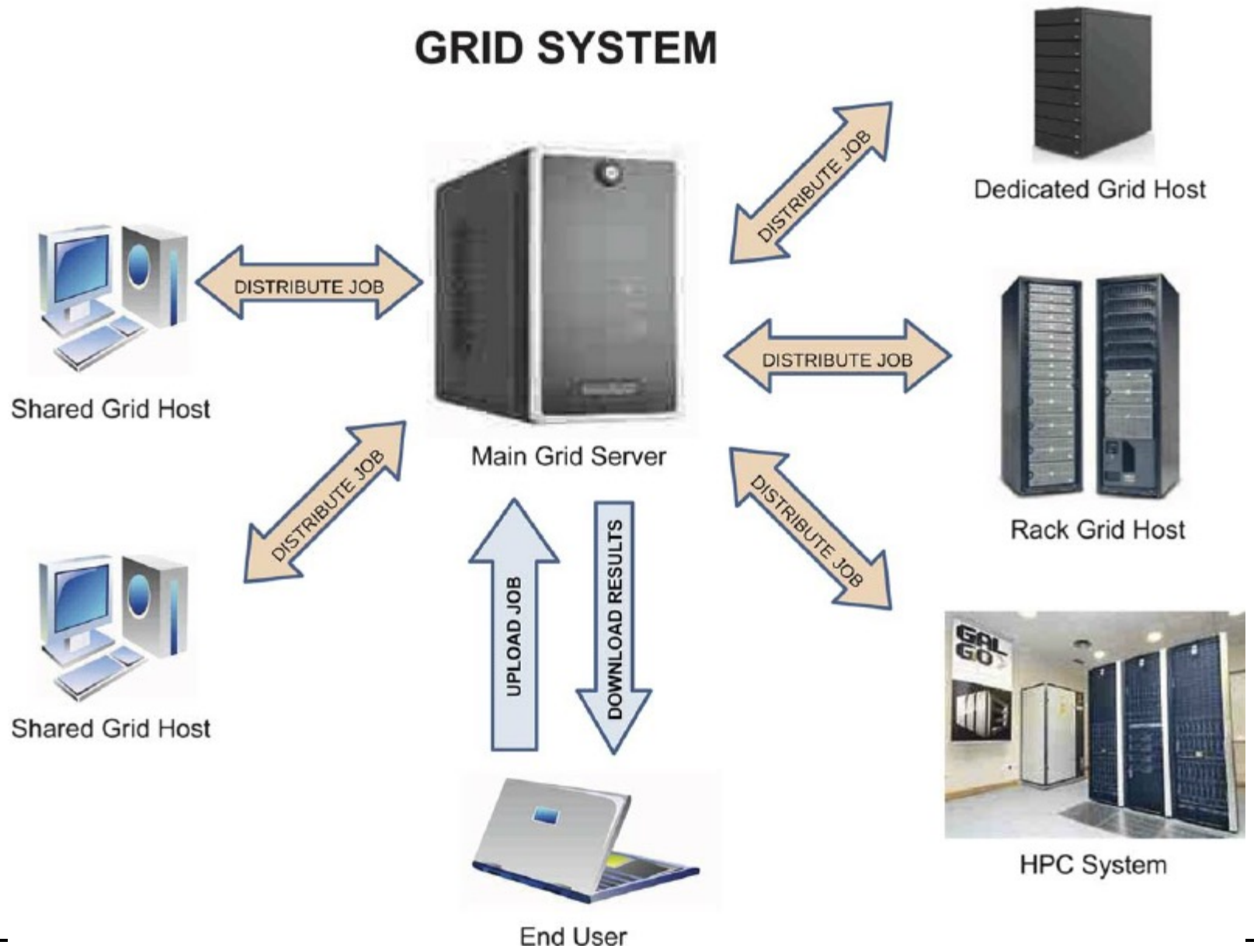
HTC: High-Throughput Computing

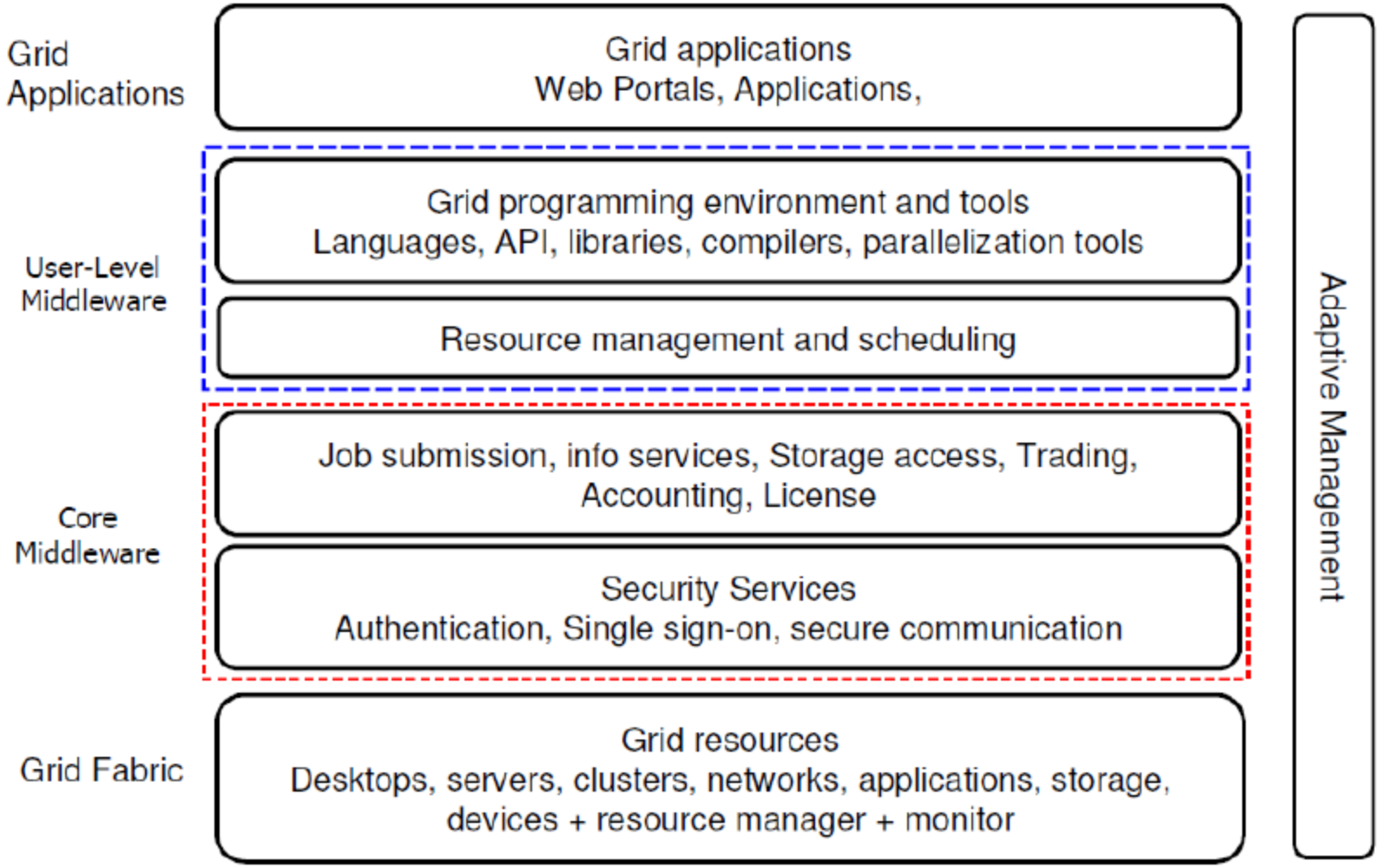
P2P: Peer to Peer

MPP: Massively Parallel Processors

Source: K. Hwang, G. Fox, and J. Dongarra, ***Distributed and Cloud Computing,*** **Morgan Kaufmann, 2012.**


Cluster: compute nodes
Switch
Login node
Internet
Users, submitting jobs


GRID SYSTEM
Dedicated Grid Host
DISTRIBUTE JOB
DISTRIBUTE JOB
Shared Grid Host
DISTRIBUTE JOB
Main Grid Server
DISTRIBUTE JOB
Rack Grid Host
DISTRIBUTE JOB
UPLOAD JOB
DOWNLOAD RESULTS
Shared Grid Host
HPC System
End User


Grid Applications
Grid applications
Web Portals, Applications,
User-Level Middleware
Grid programming environment and tools
Languages, API, libraries, compilers, parallelization tools
Resource management and scheduling
Core Middleware
Job submission, info services, Storage access, Trading,
Accounting, License
Security Services
Authentication, Single sign-on, secure communication
Grid Fabric
Grid resources
Desktops, servers, clusters, networks, applications, storage,
devices + resource manager + monitor
Adaptive Management

Grid Standards and Middleware :

Table 1.9 Grid Standards and Toolkits for scientific and Engineering Applications

Grid Standards	Major Grid Service Functionalities	Key Features and Security Infrastructure
OGSA Standard	Open Grid Service Architecture offers common grid service standards for general public use	Support heterogeneous distributed environment, bridging CA, multiple trusted intermediaries, dynamic policies, multiple security mechanisms, etc.
Globus Toolkits	Resource allocation, Globus security infrastructure (GSI), and generic security service API	Sign-in multi-site authentication with PKI, Kerberos, SSL, Proxy, delegation, and GSS API for message integrity and confidentiality
IBM Grid Toolbox	AIX and Linux grids built on top of Globus Toolkit, autonomic computing, Replica services	Using simple CA, granting access, grid service (ReGS), supporting Grid application for Java (GAF4J), GridMap in IntraGrid for security update.

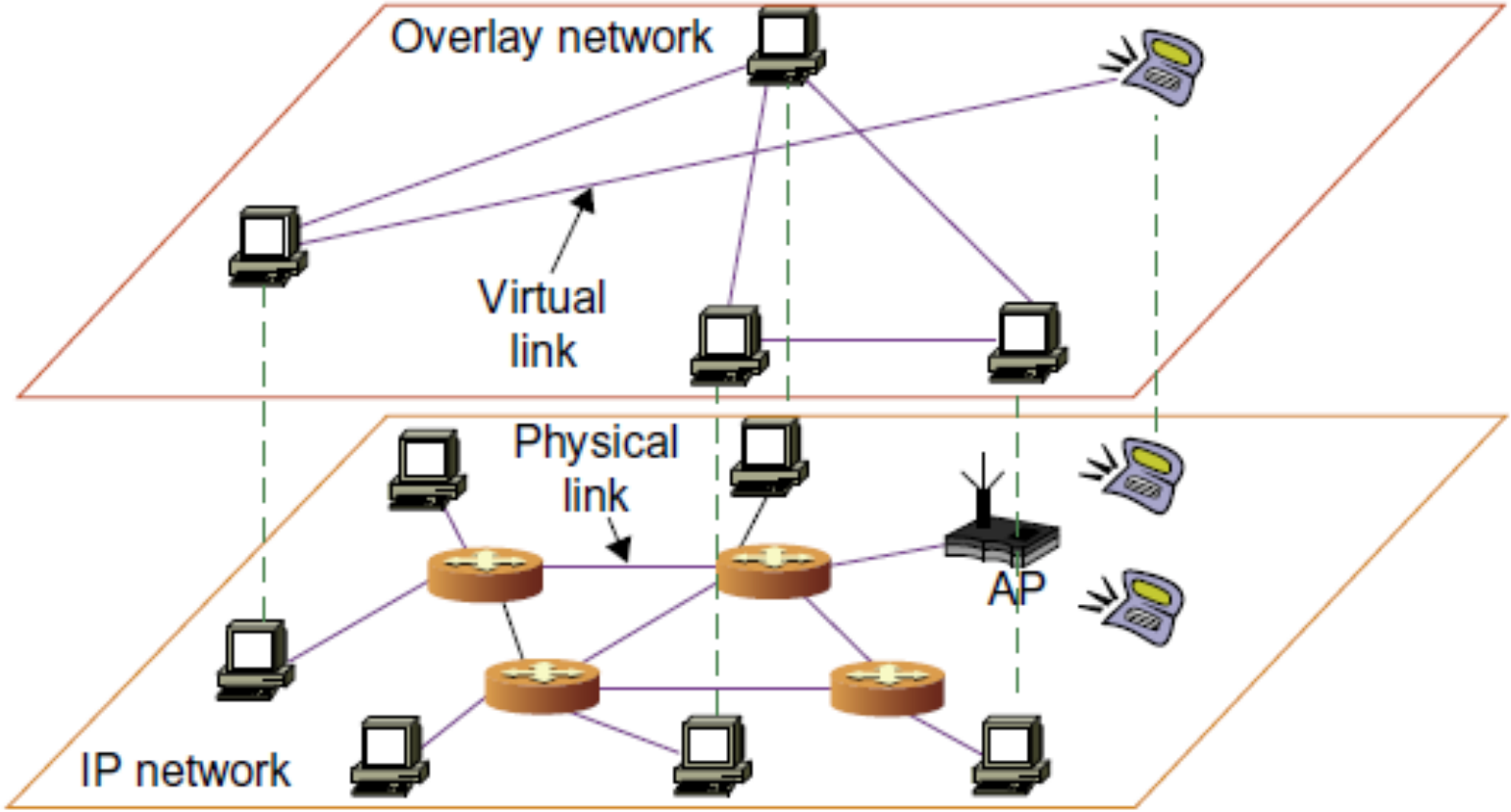
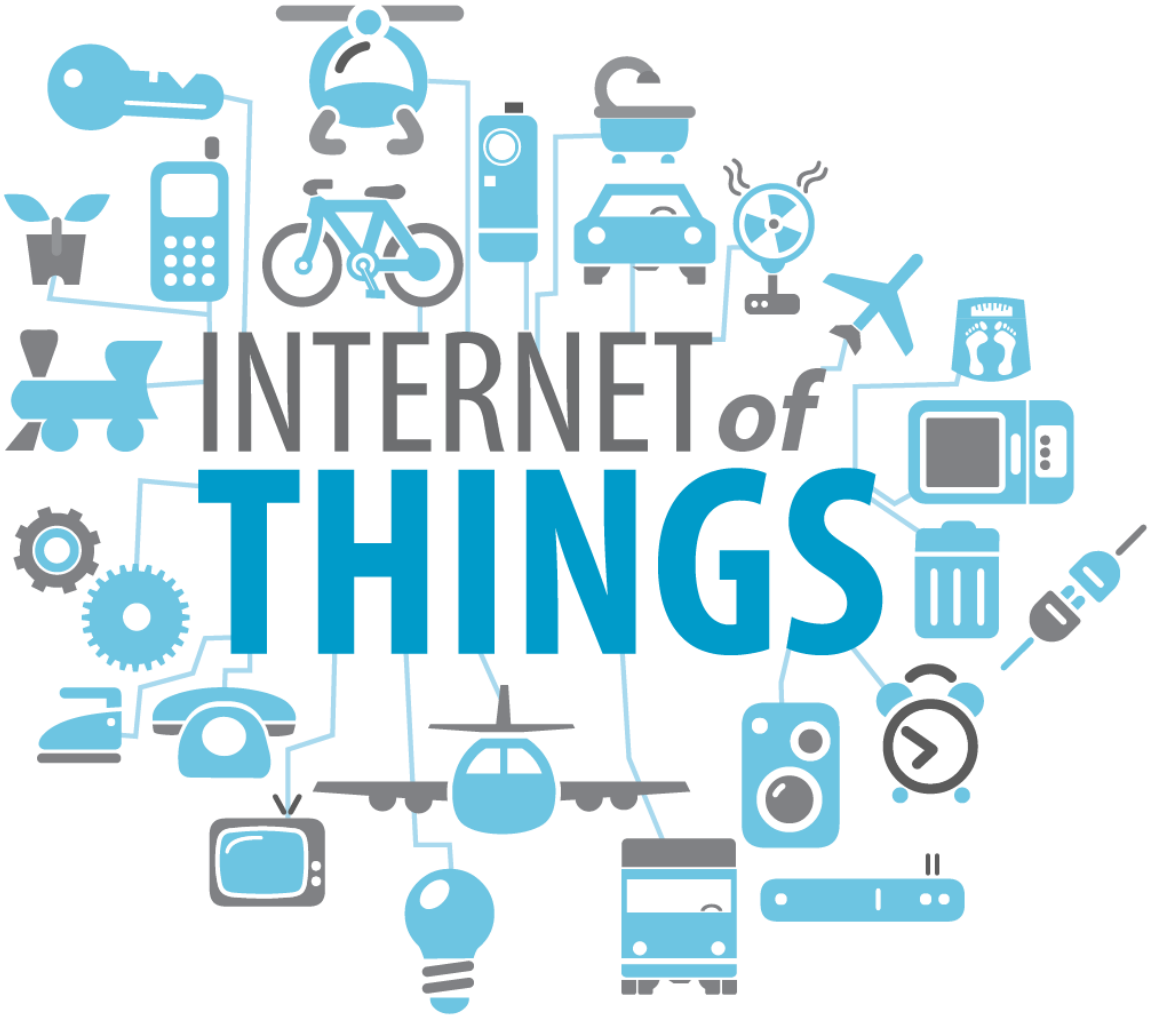
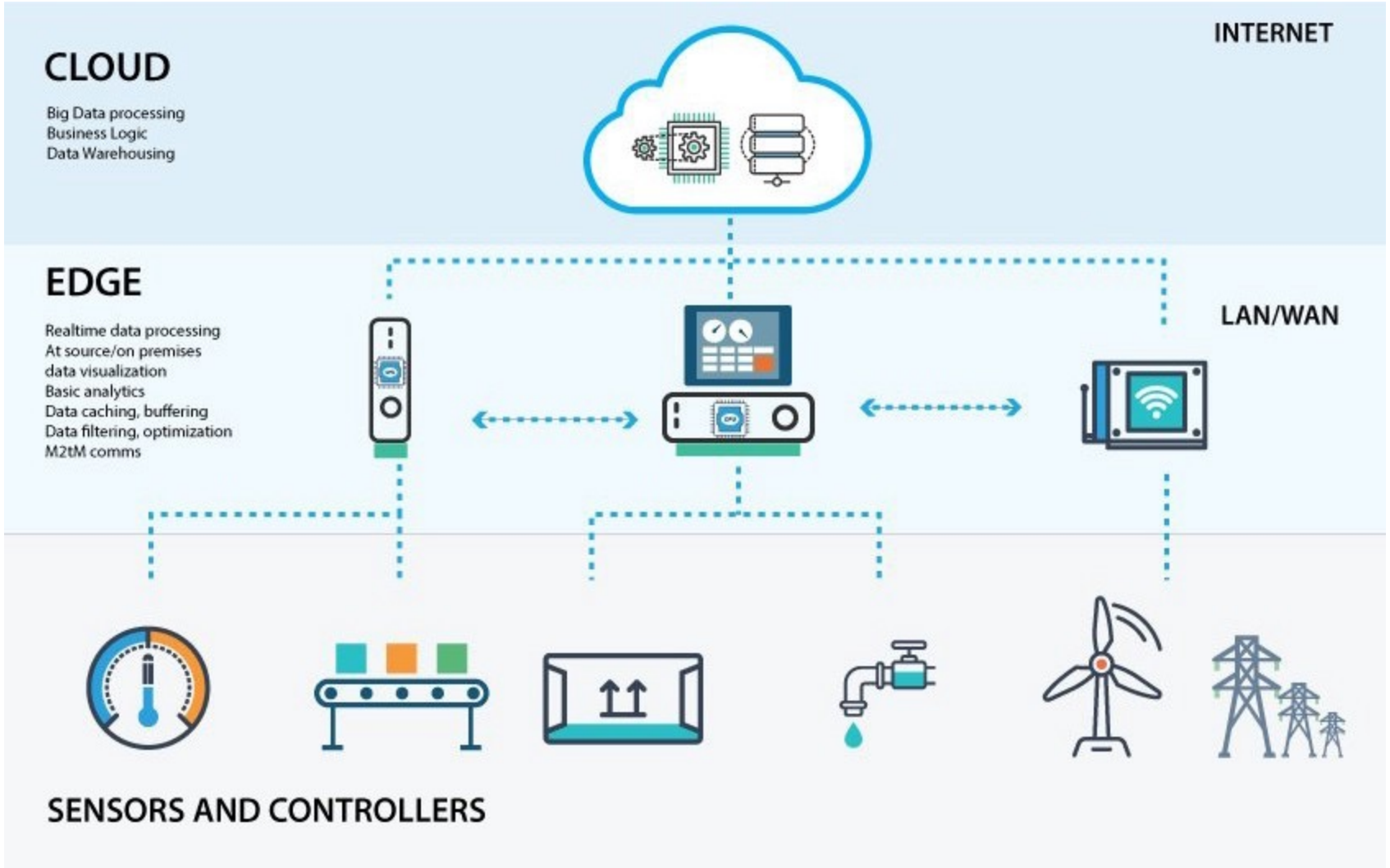

Overlay network
Virtual
link
Physical
link
AP
IP network

Table 1.5 Major Categories of P2P Network Families [42]

System Features	Distributed File Sharing	Collaborative Platform	Distributed P2P Computing	P2P Platform
Attractive Applications	Content distribution of MP3 music, video, open software, etc.	Instant messaging, collaborative design and gaming	Scientific exploration and social networking	Open networks for public resources
Operational Problems	Loose security and serious online copyright violations	Lack of trust, disturbed by spam, privacy, and peer collusion	Security holes, selfish partners, and peer collusion	Lack of standards or protection protocols
Example Systems	Gnutella, Napster, eMule, BitTorrent, Aimster, KaZaA, etc.	ICQ, AIM, Groove, Magi, Multiplayer Games, Skype, etc.	SETI@home, Geonome@home, etc.	JXTA, .NET, FightingAid@home, etc.

The Internet of Things (IoT)




INTERNET
CLOUD
Big Data processing
Business Logic
Data Warehousing
EDGE
Realtime data processing
At source/on premises
data visualization
Basic analytics
Data caching, buffering
Data filtering, optimization
M2tM comms
LAN/WAN
SENSORS AND CONTROLLERS

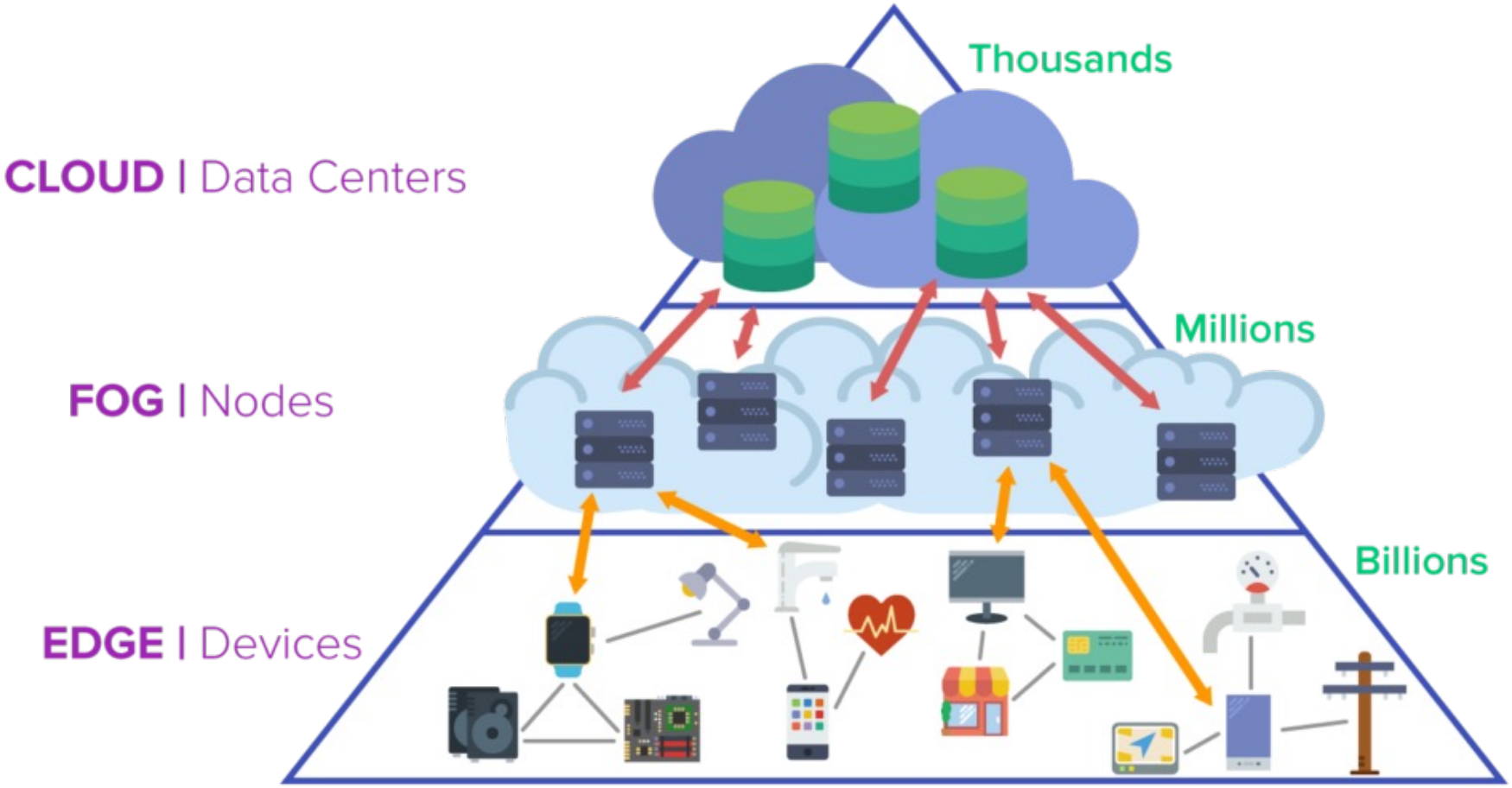
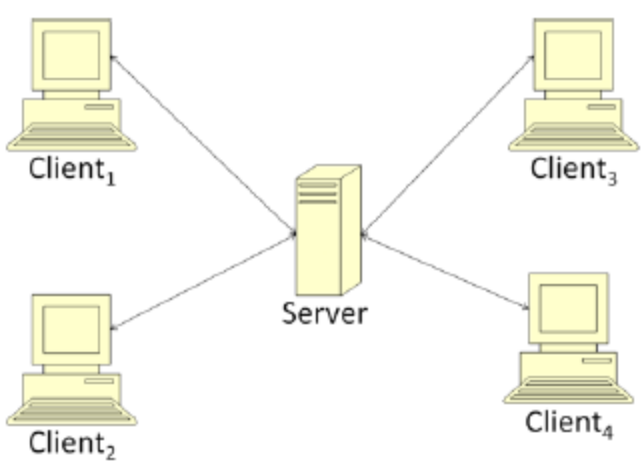
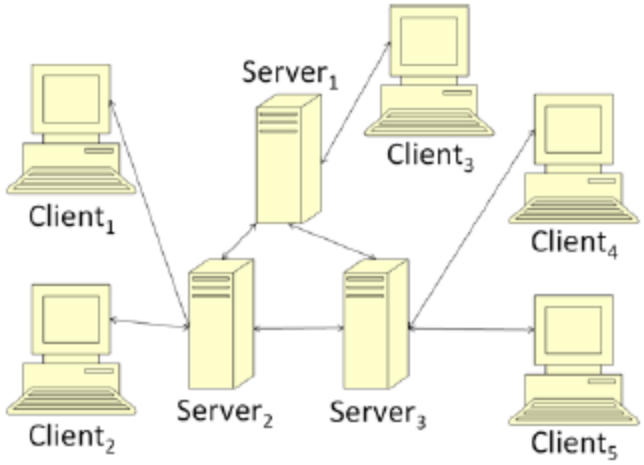

CLOUD | Data Centers
Thousands
FOG | Nodes
Millions
EDGE | Devices
Billions

Table 1.2 Classification of Distributed Parallel Computing Systems

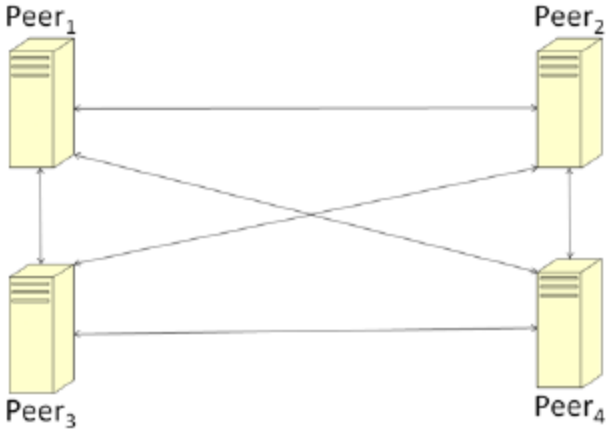
Functionality, Applications	Multicomputer Clusters [27, 33]	Peer-to-Peer Networks [40]	Data/Computational Grids [6, 42]	Cloud Platforms [1, 9, 12, 17, 29]
Architecture, Network Connectivity and Size	Network of compute nodes interconnected by SAN, LAN, or WAN, hierarchically	Flexible network of client machines logically connected by an overlay network	Heterogeneous clusters interconnected by high-speed network links over selected resource sites.	Virtualized cluster of servers over datacenters via service-level agreement
Control and Resources Management	Homogeneous nodes with distributed control, running Unix or Linux	Autonomous client nodes, free in and out, with distributed self-organization	Centralized control, server oriented with authenticated security, and static resources	Dynamic resource provisioning of servers, storage, and networks over massive datasets
Applications and network-centric services	High-performance computing, search engines, and web services, etc.	Most appealing to business file sharing, content delivery, and social networking	Distributed super-computing, global problem solving, and datacenter services	Upgraded web search, utility computing, and outsourced computing services
Representative Operational Systems	Google search engine, SunBlade, IBM Road Runner, Cray XT4, etc.	Gnutella, eMule, BitTorrent, Napster, KaZaA, Skype, JXTA, and .NET	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, Amazon Web Service(AWS), and Microsoft Azure,



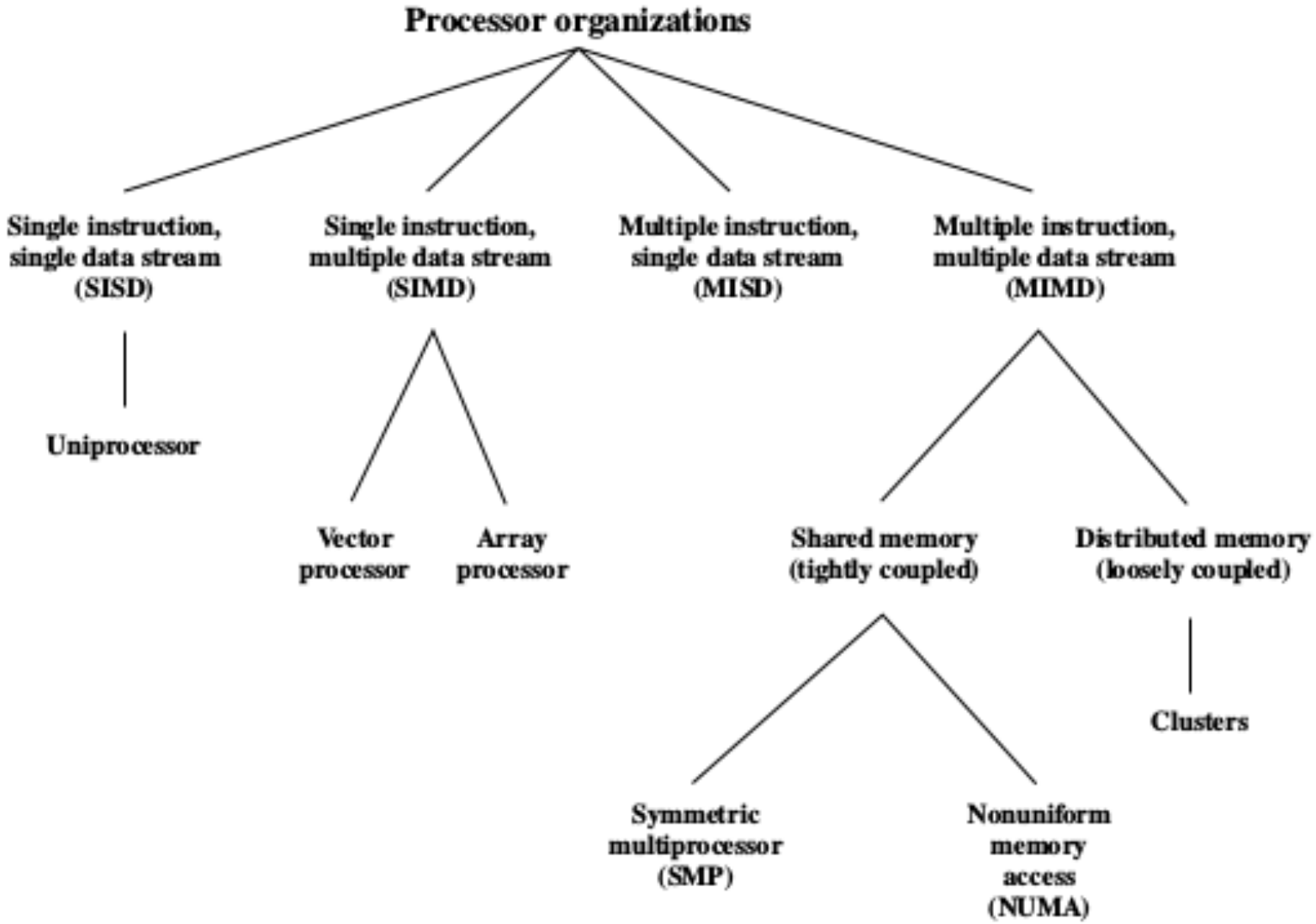
(a) Client-Server

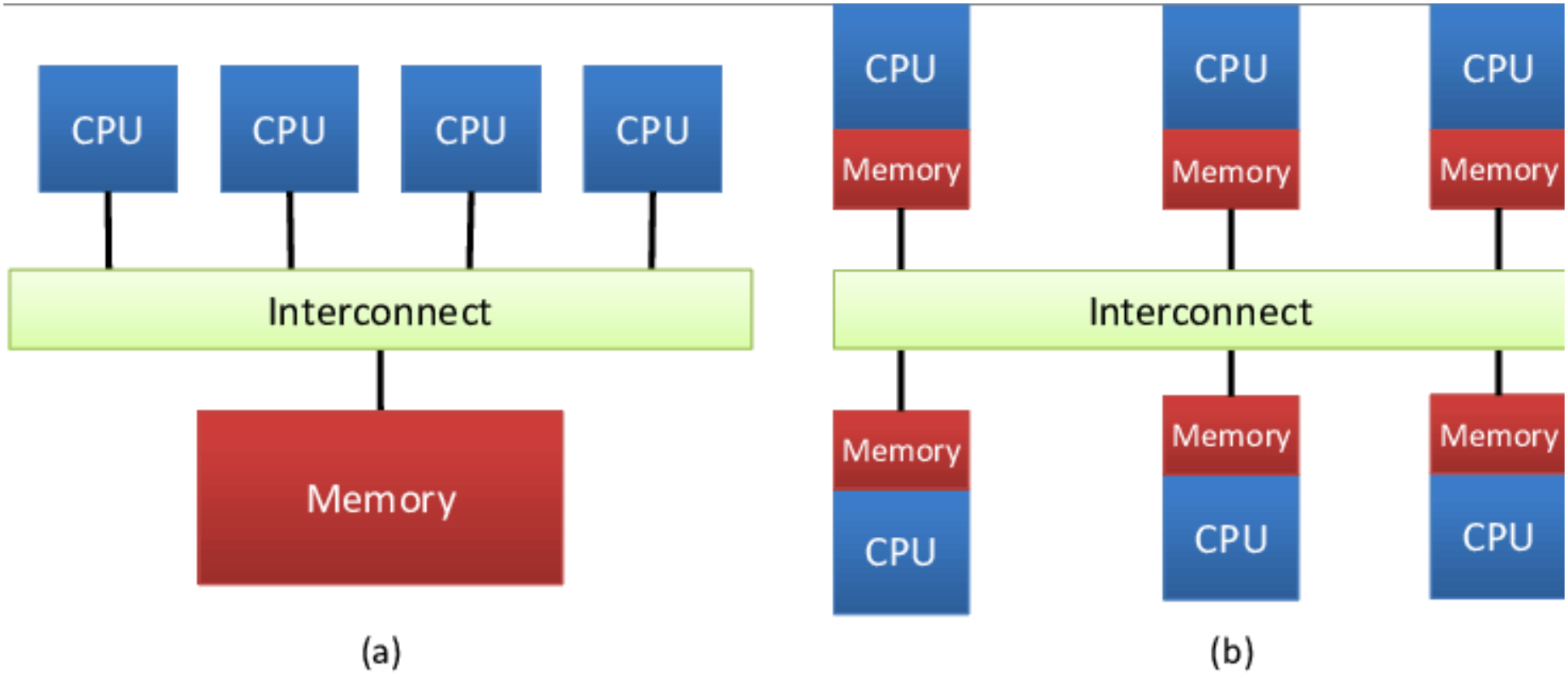


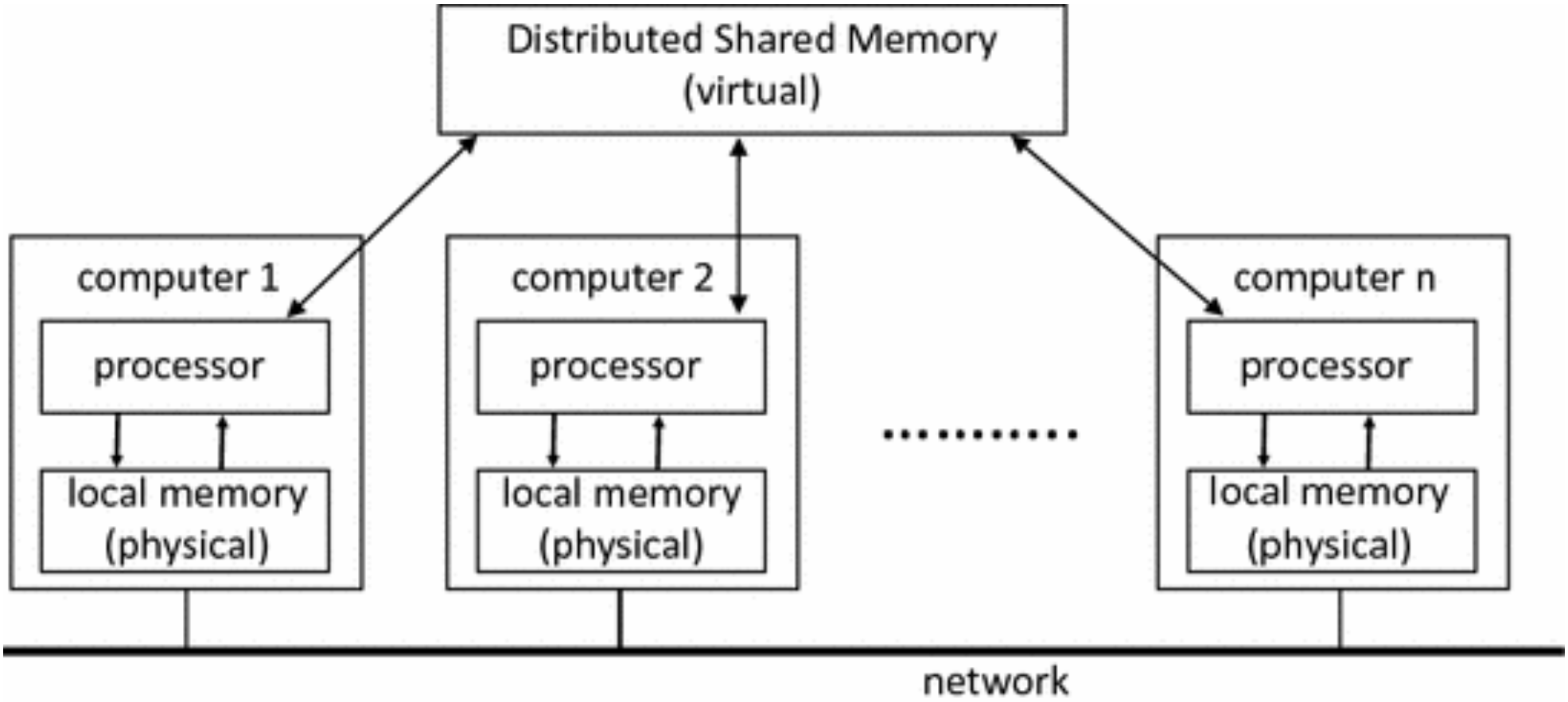
(b) Multiple-Server

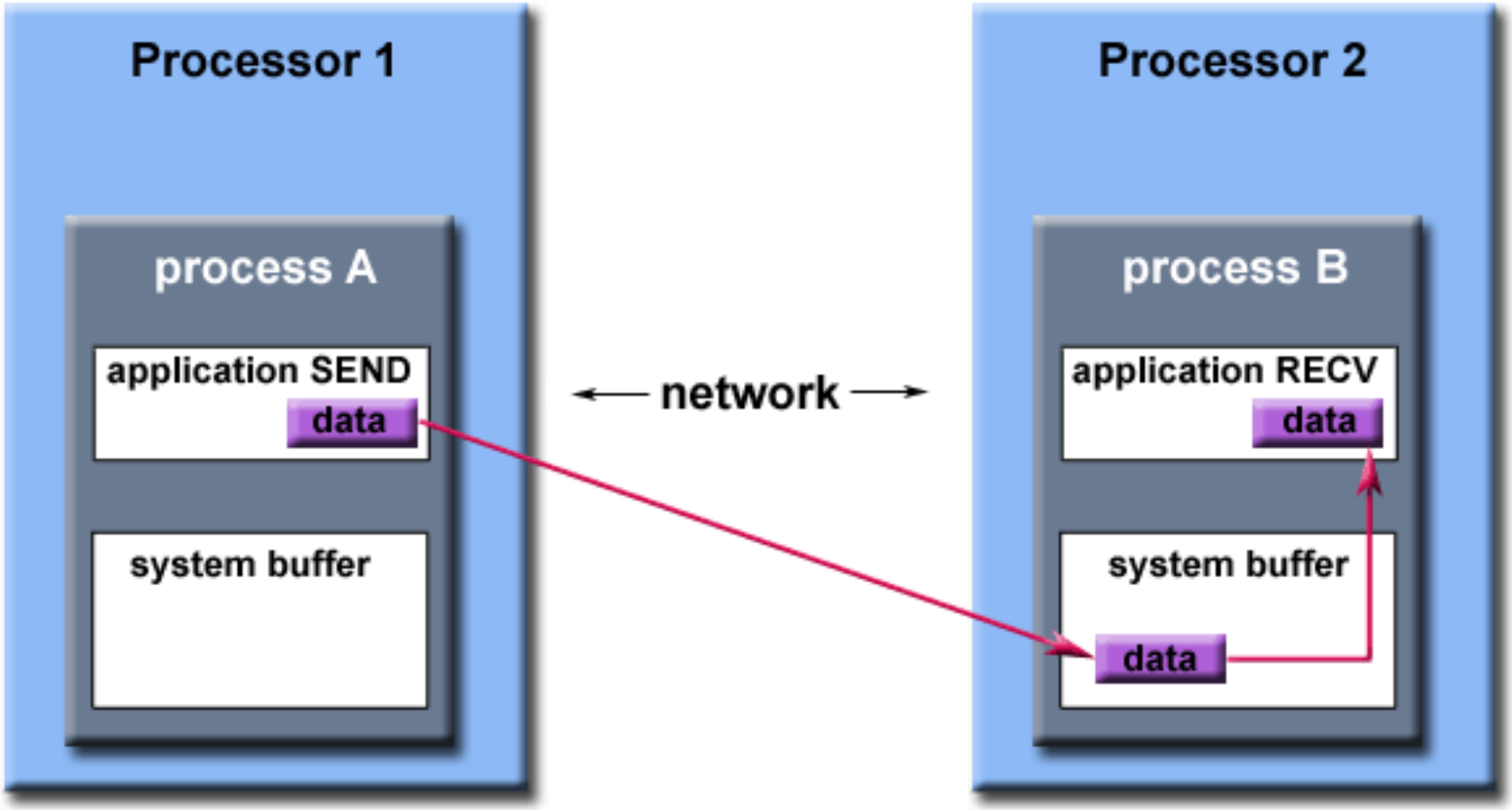


(c) Peer-to-Peer


Processor organizations
Single instruction, single data stream (SISD)
Single instruction, multiple data stream (SIMD)
Multiple instruction, single data stream (MISD)
Multiple instruction, multiple data stream (MIMD)
Uniprocessor
Vector processor
Array processor
Shared memory (tightly coupled)
Distributed memory (loosely coupled)
Clusters
Symmetric multiprocessor (SMP)
Nonuniform memory access (NUMA)


CPU
CPU
CPU
CPU
Interconnect
Memory
(a)
CPU
Memory
CPU
Memory
CPU
Memory
Interconnect
Memory
CPU
Memory
CPU
Memory
CPU
(b)


Distributed Shared Memory
(virtual)
computer 1
processor
local memory
(physical)
computer 2
processor
local memory
(physical)
computer n
processor
local memory
(physical)
network

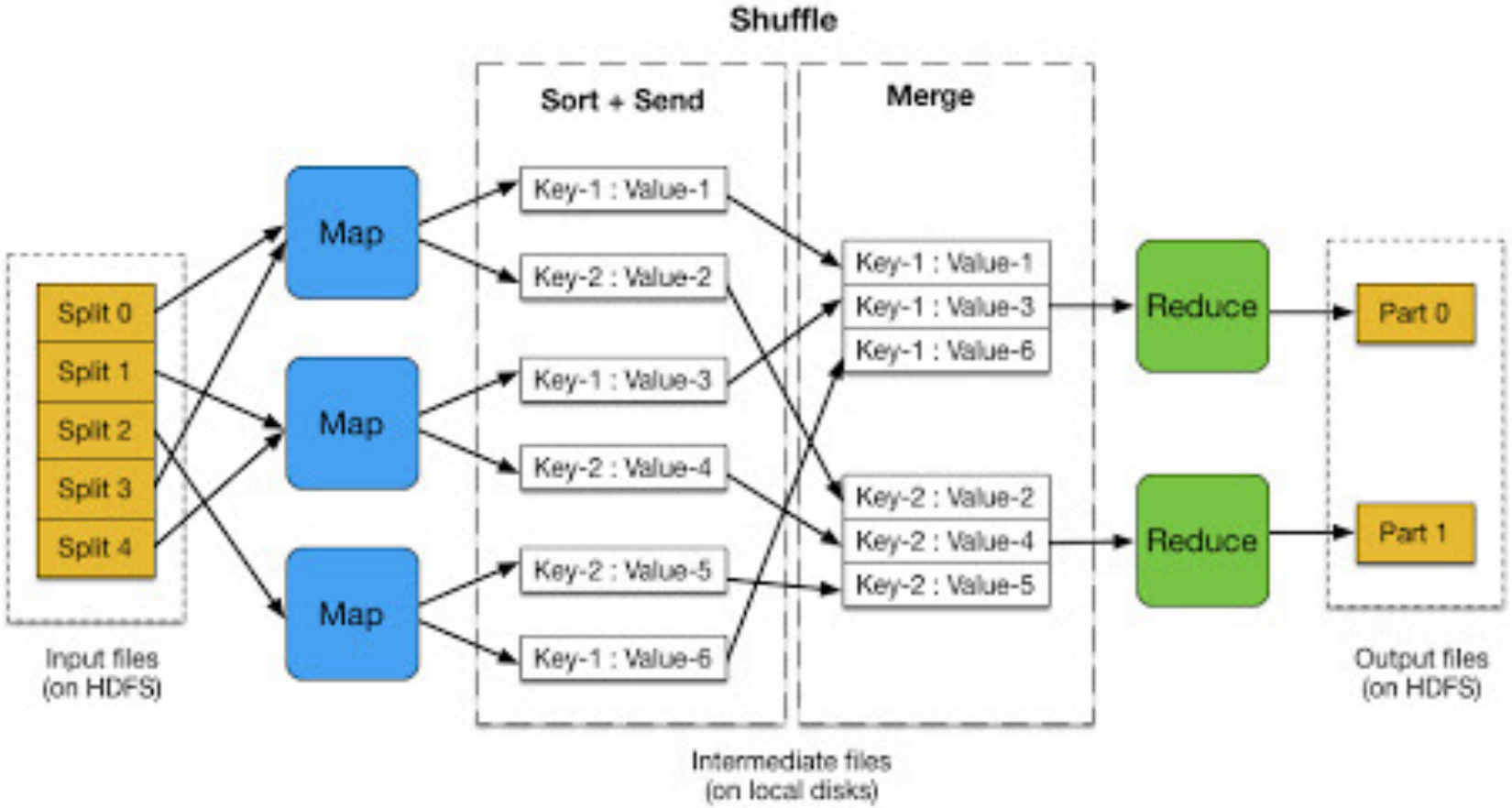


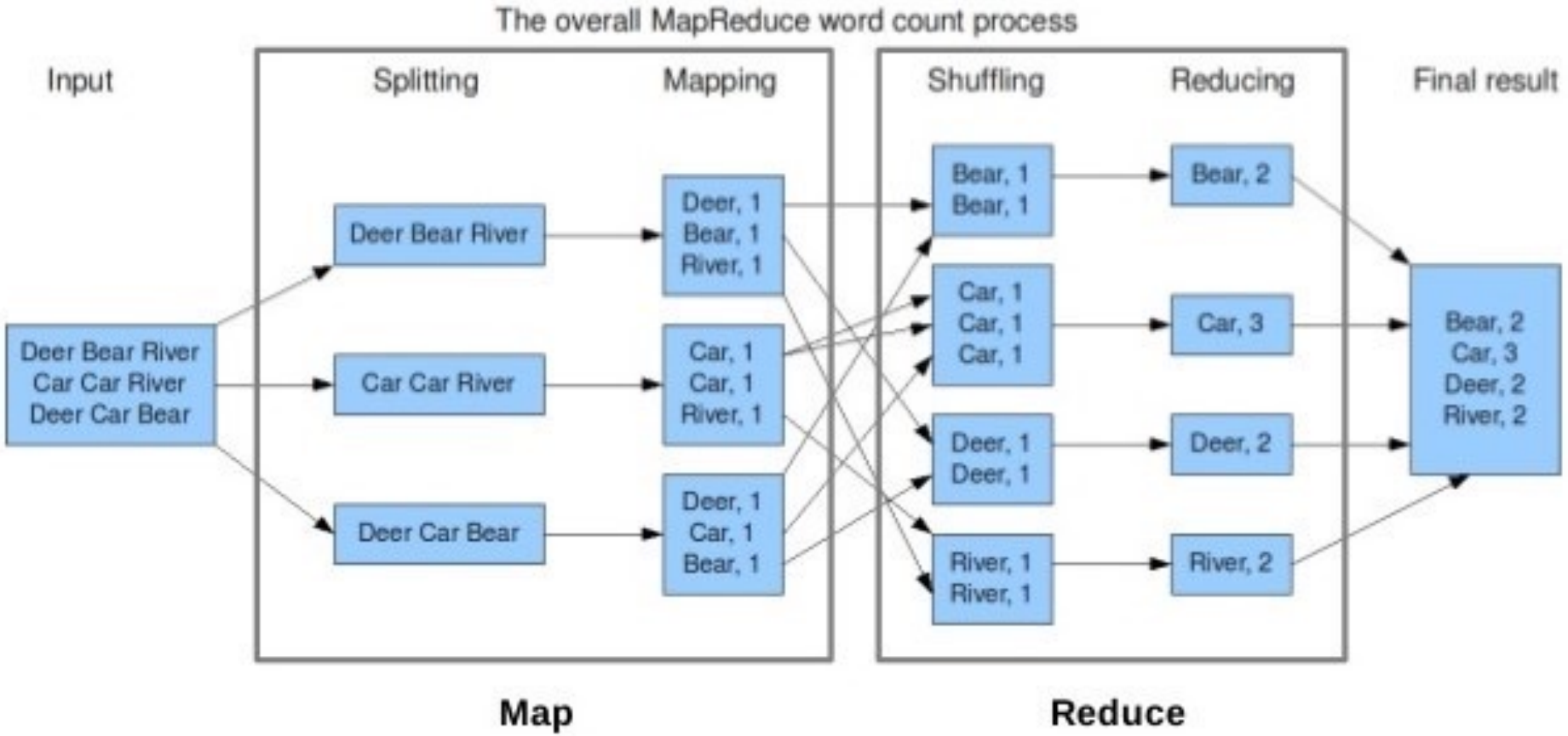
Path of a message buffered at the receiving process

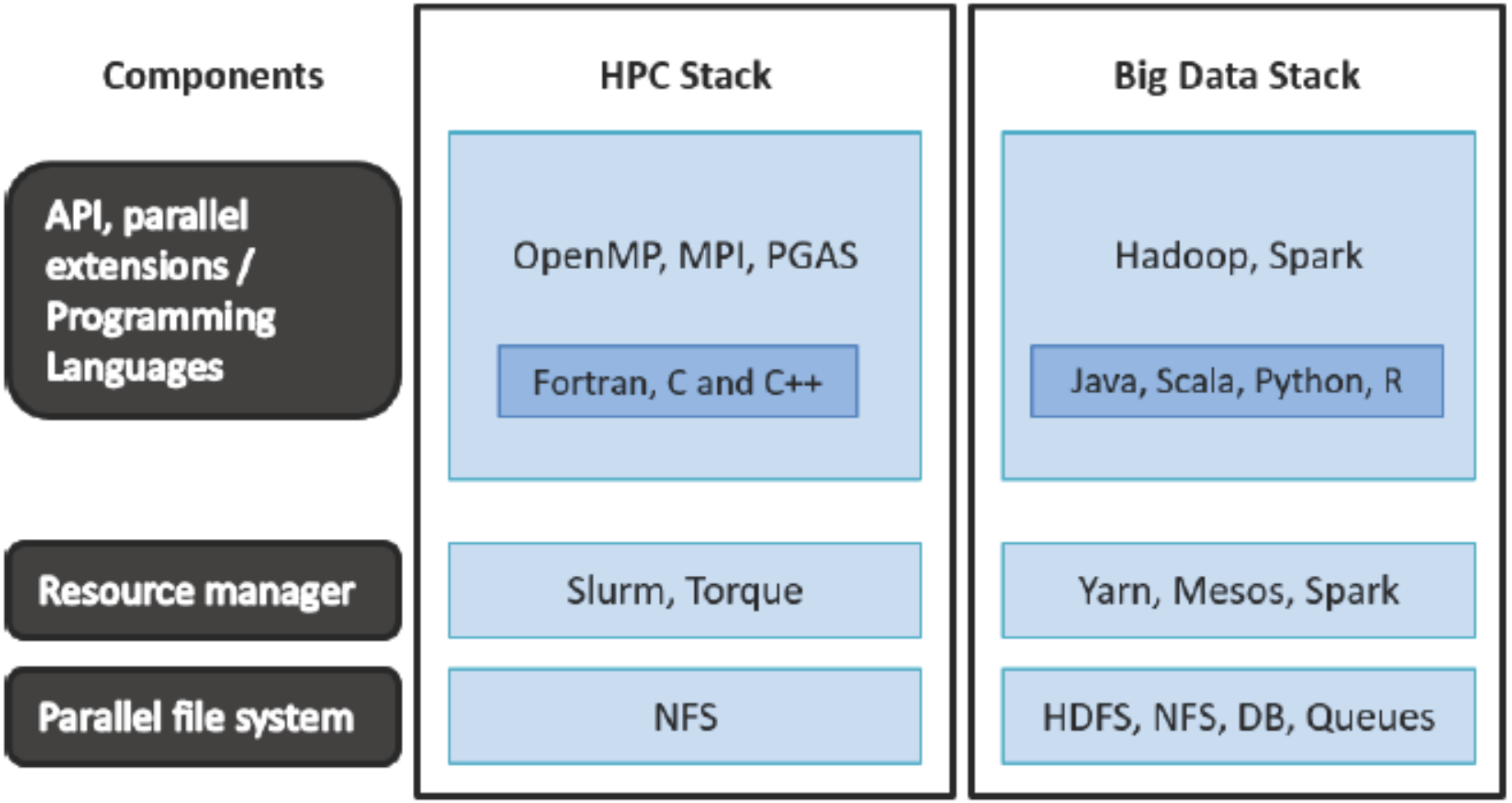
Parallel and Distributed Programming

Table 1.7 Parallel and Distributed Programming Models and Tool Sets

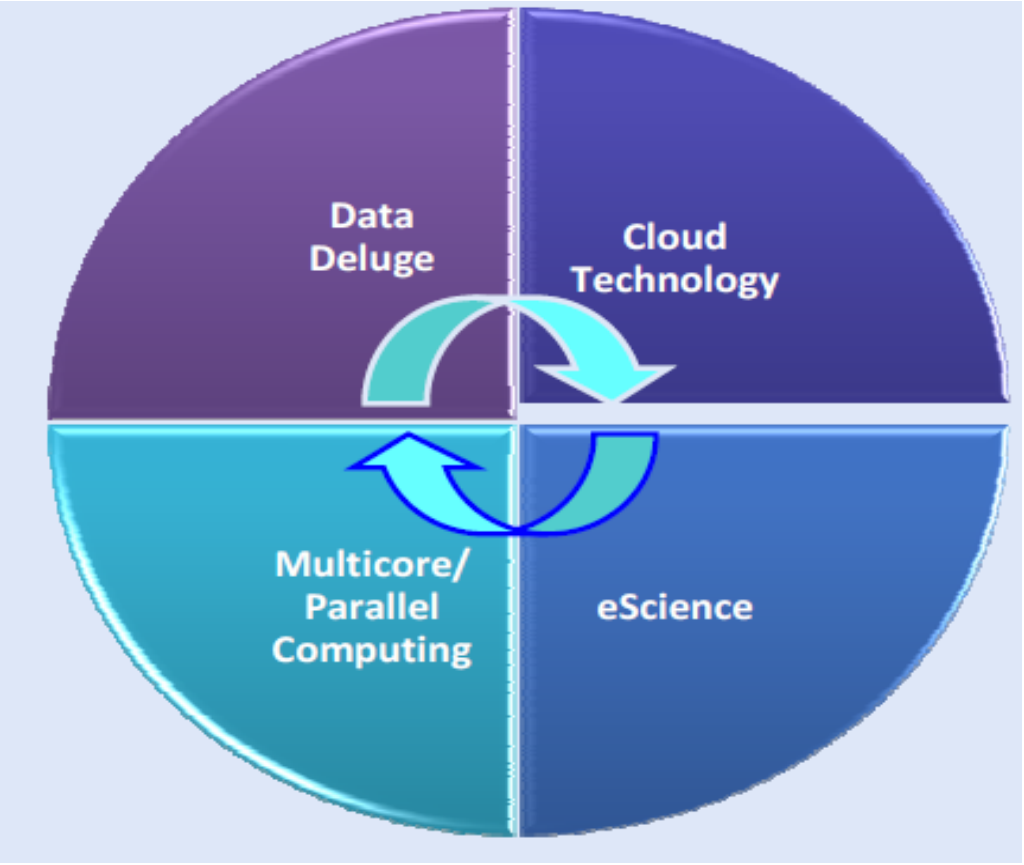
Model	Description	Features
MPI	A library of subprograms that can be called from C or FORTRAN to write parallel programs running on distributed computer systems [6,28,42]	Specify synchronous or asynchronous point-to-point and collective communication commands and I/O operations in user programs for message-passing execution
MapReduce	A Web programming model for scalable data processing on large clusters over large data sets, or in Web search operations [16]	*Map* function generates a set of intermediate key/value pairs; *Reduce* function merges all intermediate values with the same key
Hadoop	A software library to write and run large user applications on vast data sets in business applications (http://hadoop.apache.org/core)	A scalable, economical, efficient, and reliable tool for providing users with easy access of commercial clusters


Shuffle
Sort + Send
Merge
Split 0
Split 1
Split 2
Split 3
Split 4
Map
Map
Map
Key-1 : Value-1
Key-2 : Value-2
Key-1 : Value-3
Key-2 : Value-4
Key-2 : Value-5
Key-1 : Value-6
Key-1 : Value-1
Key-1 : Value-3
Key-1 : Value-6
Key-2 : Value-2
Key-2 : Value-4
Key-2 : Value-5
Reduce
Reduce
Part 0
Part 1
Input files
(on HDFS)
Intermediate files
(on local disks)
Output files
(on HDFS)


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2
Map
Reduce


Components
HPC Stack
Big Data Stack
API, parallel extensions / Programming Languages
OpenMP, MPI, PGAS
Fortran, C and C++
Hadoop, Spark
Java, Scala, Python, R
Resource manager
Slurm, Torque
Yarn, Mesos, Spark
Parallel file system
NFS
HDFS, NFS, DB, Queues

Interactions among 4 technical challenges : Data Deluge, Cloud Technology, eScience, and Multicore/Parallel Computing

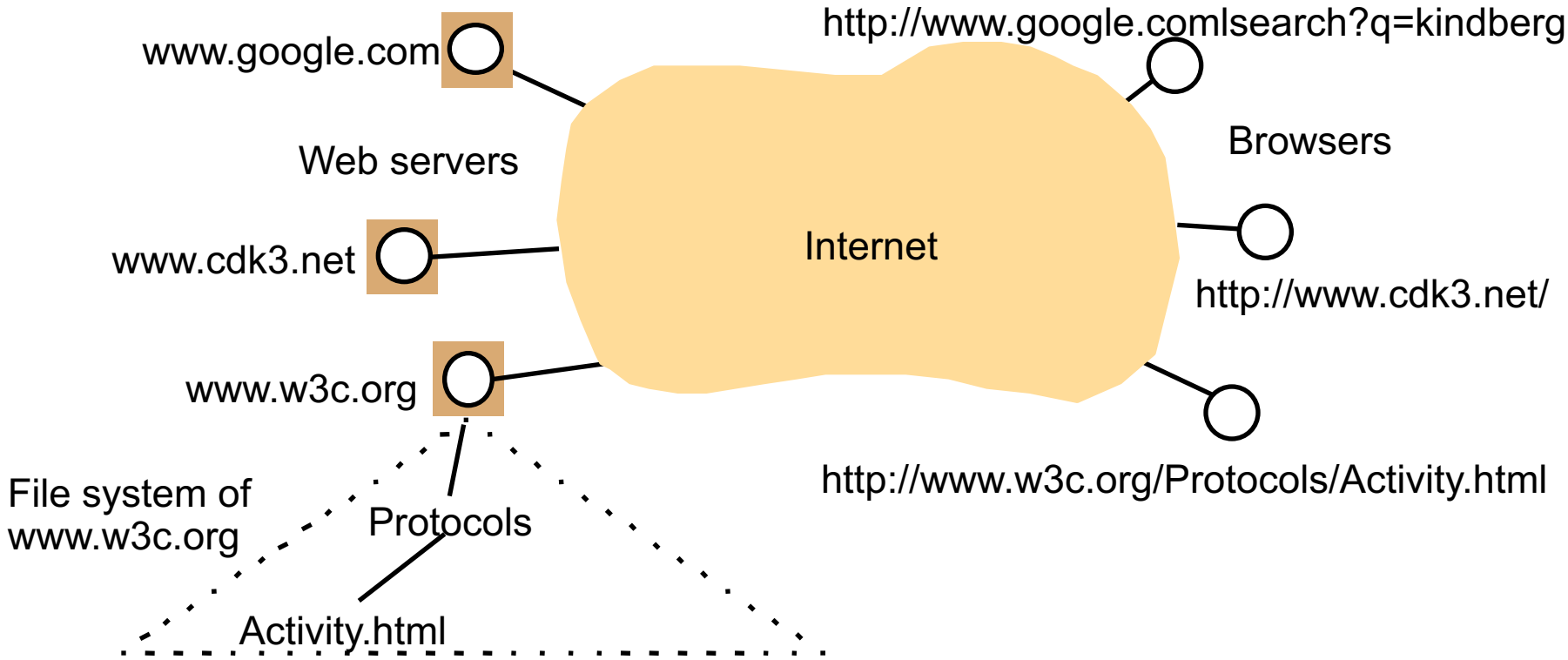


(Courtesy of Judy Qiu, Indiana University, 2011)

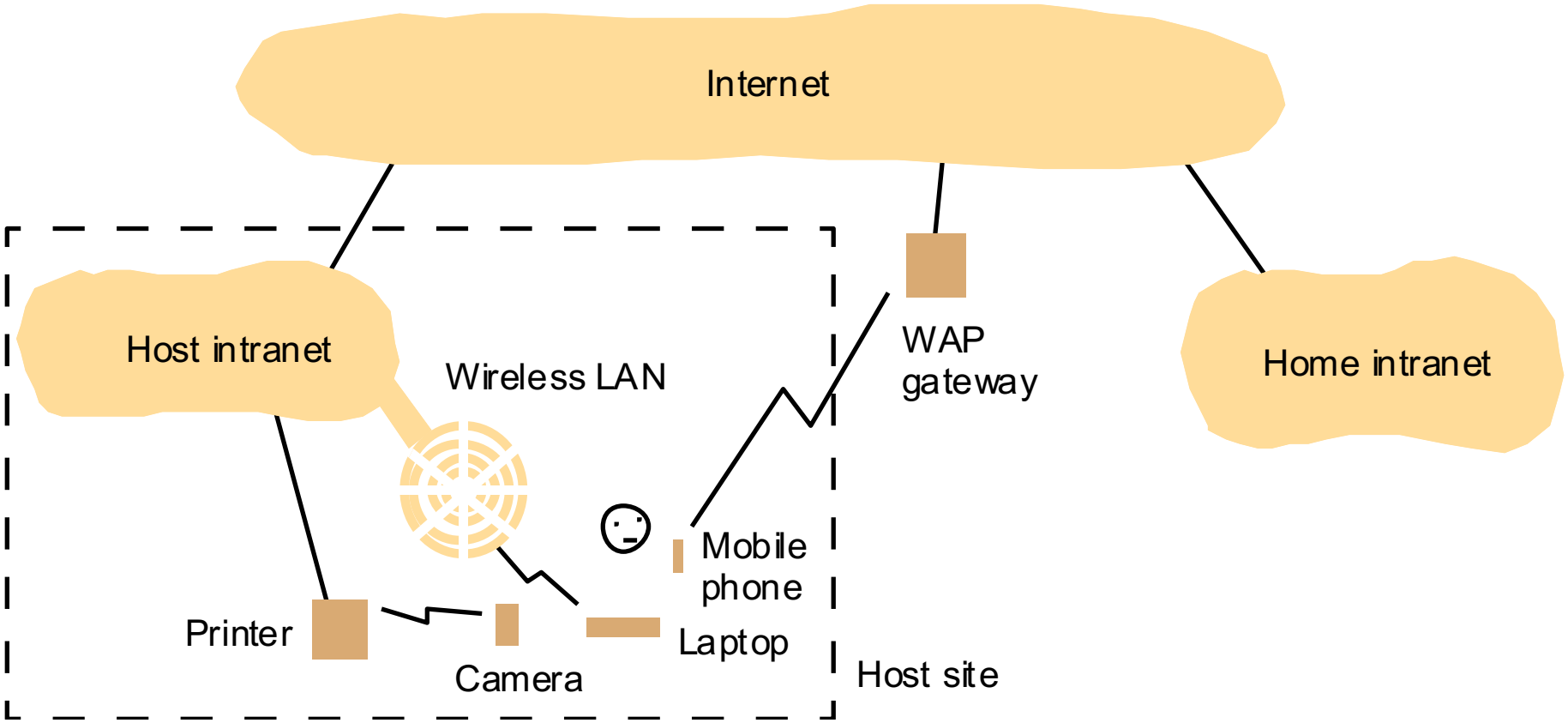
Distributed Systems: Definitions

- What is a distributed system: a system in which components located in networked computers communicate and coordinate their actions only by passing messages.
- Motivation: sharing of resources

Resource sharing: Web servers and web browsers

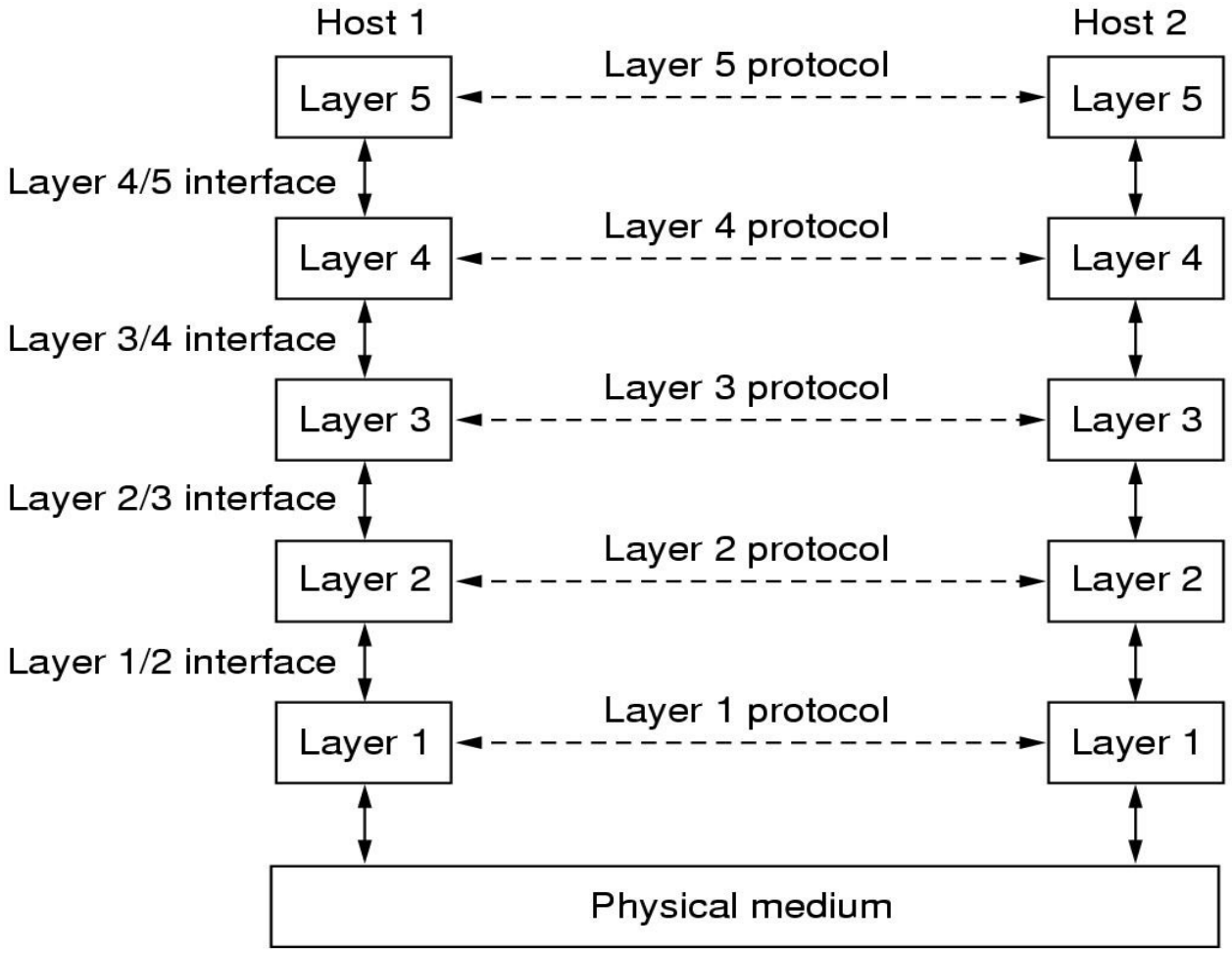


Portable and handheld devices in a distributed system



CN Layered Architecture

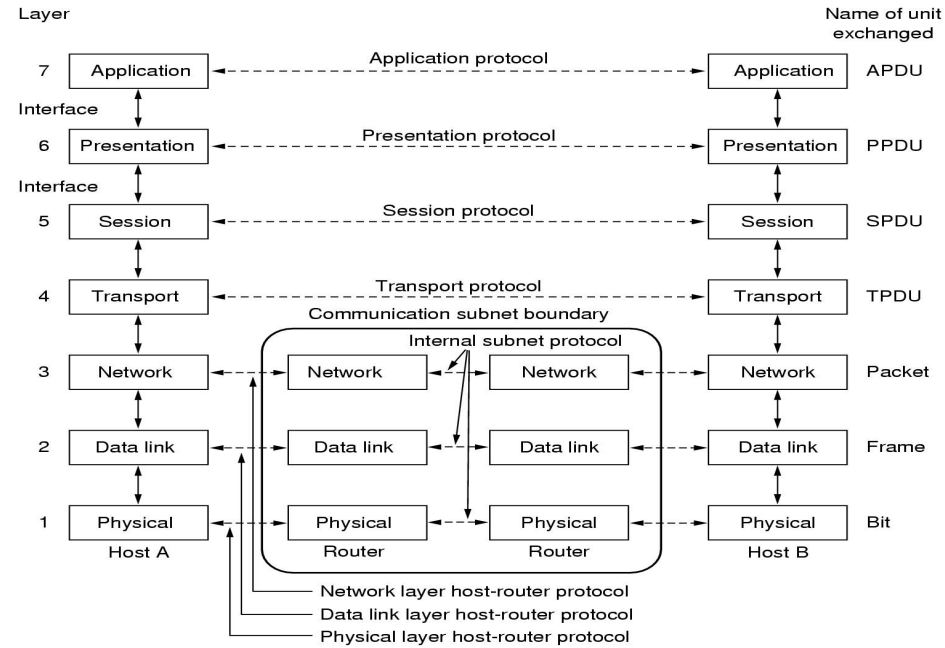
- The purpose of each layer is to offer a communication services to higher level layers
- Each layer has two interfaces
  - peer-to-peer interface
    - defines the form and types of messages exchanged between peers (indirect communication)
  - service interface
    - defines the primitives (operations) that a layer provides to the layer above it
- Layering is non-linear

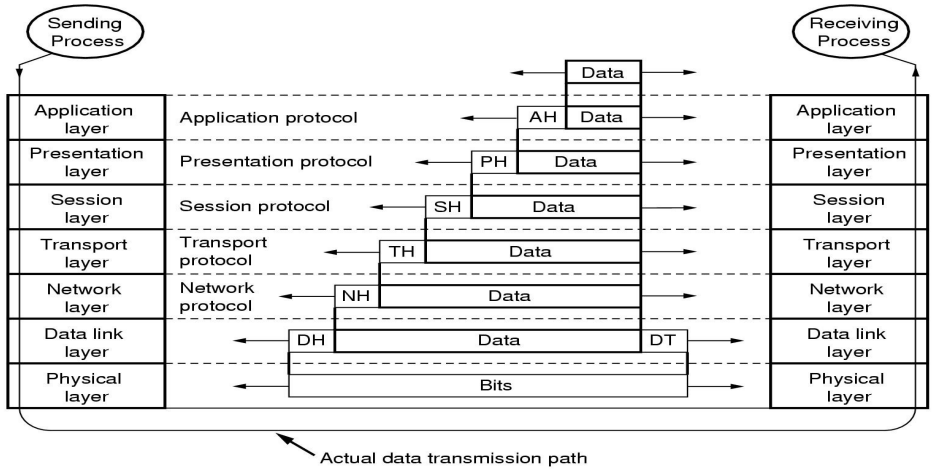


ISO OSI Architecture

- International Standards Organisation (ISO)
- *Physical*: transmission of raw bits onto the communications medium
- *Data link*: reliable transmission of frames, flow control, arbitration
- *Network*: packet switching, routing congestion control
- *Transport*: process-to-process channel, node-to-node connection, provides user services, flow control, multiplexing
- *Session*
- *Presentation*
- *Application*

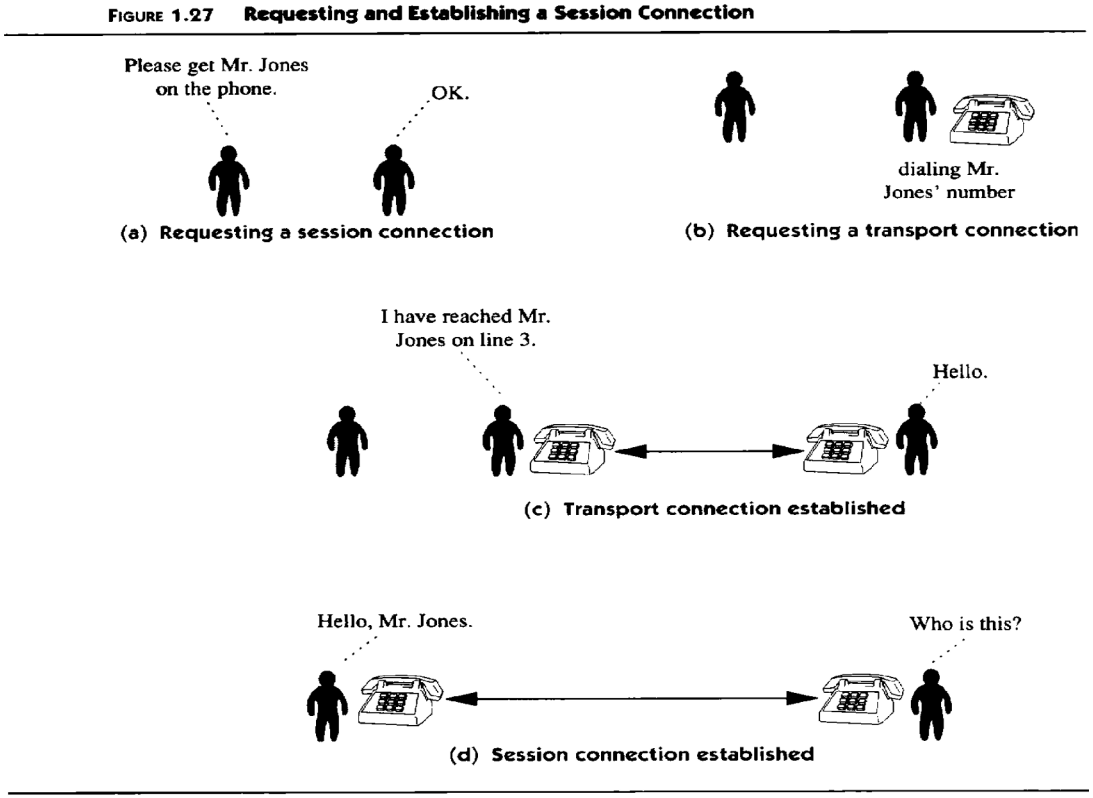
ISO OSI Architecture





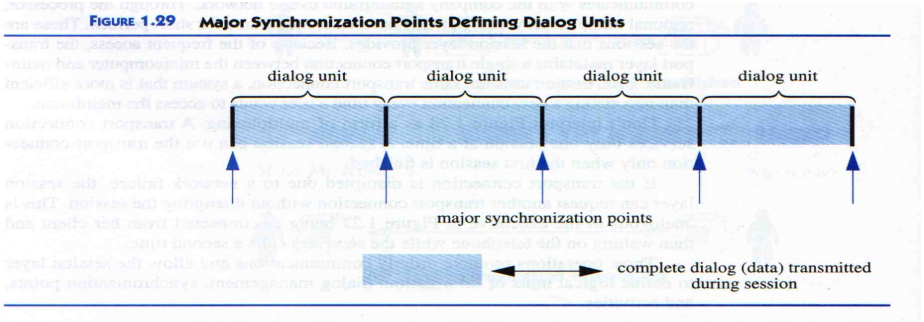
OSI Session Layer

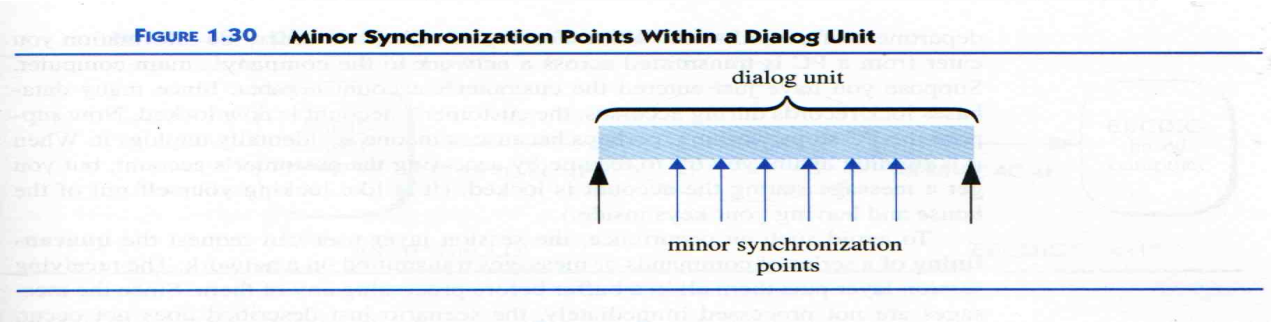
- The protocols necessary to establish and maintain a connection or session between 2 end-users
- Transport vs session

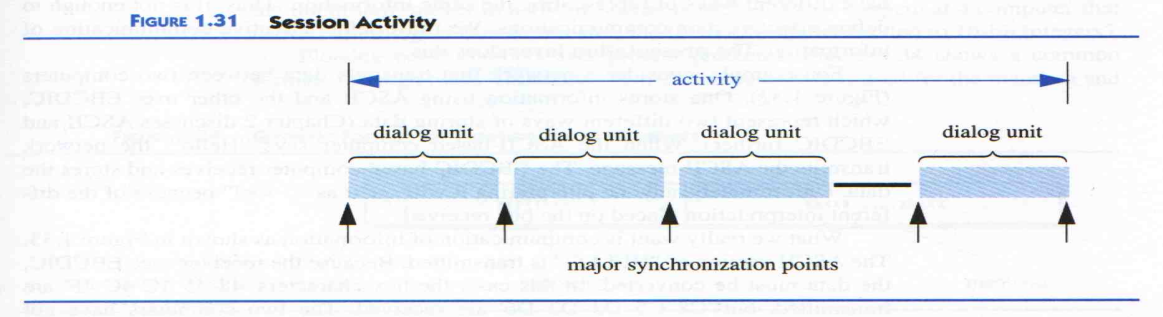


OSI Session Layer

- Orderly communication
  - dialog management
  - synchronisation points (major and minor)
  - activities



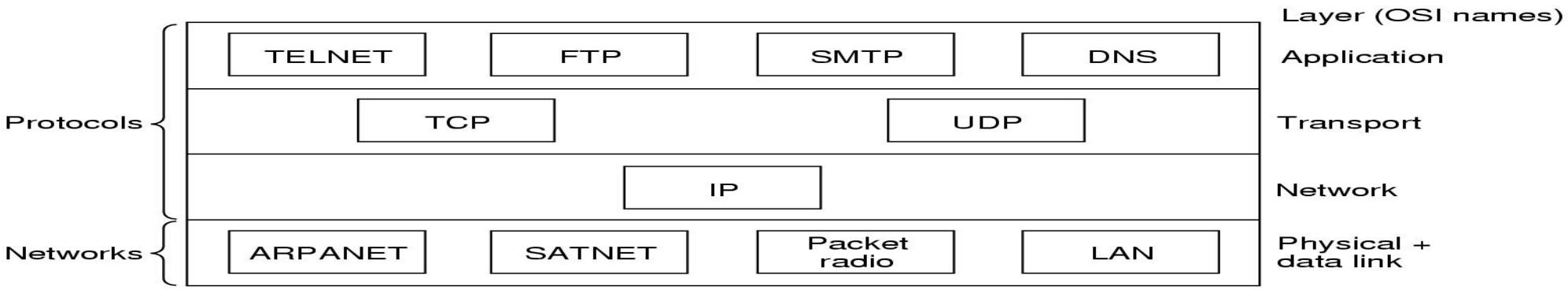




OSI Presentation & Application Layers

- Presentation
  - Effective communication of information rather than of data
  - Code and number conversion
  - Transmission of sophisticated data structures
  - Data compression
- Application
  - Electronic mail
  - File transfer protocols (ftp)
  - virtual terminal protocols (telnet)
  - Distributed system (distributed database)
  - Client-server

Internet Architecture (TCP/IP)



- Host-to-Network Layer (OSI Physical and Data link layers)
- Internet Layer (OSI Network layer - Internet Protocol/IP)
- Transport Layer (Transmission Control Protocol/TCP & User Datagram Protocol/UDP)
- Application Layer

Challenges

- Heterogeneity (mobile code)
- Openness
- Security (denial of service, security of mobile code etc.)
- Scalability (cost of physical resources, performance loss, availability, bottlenecks)
- Failure (detection, correct/hide, tolerate, recover, redundancy)
- Concurrency (consistency)
- Transparency

Transparencies

- *Access transparency*: enables local and remote resources to be accessed using identical operations.
- *Location transparency*: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).
- *Concurrency transparency*: enables several processes to operate concurrently using shared resources without interference between them.
- *Replication transparency*: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.
- *Failure transparency*: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.
- *Mobility transparency*: allows the movement of resources and clients within a system without affecting the operation of users or programs.
- *Performance transparency*: allows the system to be reconfigured to improve performance as loads vary.
- *Scaling transparency*: allows the system and applications to expand in scale without change to the system structure or the application algorithms.