History

1945-1985: Computers were large & expensive.

mid-1980s: 2 major advances, in technology.

5. Development of microprocessors.
   8-bit: INTEL 8080
   Motorola 6800
   later 16-, 32-, 64-bit

Economic terms:
   a NYC 10 million dollars: 1 instr/sec
   to
   1000 dollars: 1 billion instr/sec

Price/performance: $10^{13}$

if cars had such improvement
   Rolls Royce would cost 1 dollar & give
   1 billion miles/gallon.

II. High-speed computer networks LANS.

allow a large number of machines to be connected & transfer information between these machines in few microseconds.

100 million to 10 billion bits/sec

WANs: 64 Kbps to 1 Gbps.

large no. of interconnected nyc called Distributed/cell systems/NN.
Distributed Systems

is a collection of independent computers that appears to all users as a single coherent system.

Properties
- Autonomous
- Single system
- Collaboration
- No physical clock
- No shared memory
- Geographical separation
- No assumption about the type of the computer
- No assumption as to how they are connected

Characteristics
- Communication occurs in a consistent and uniform way
- User interaction

Desirable goals:
- Easy to expand and scale
- Continuously available, even when some parts are temporarily out-of-order

Diagram:
```
    | App 1 |
    | App 2 |
    | App 3 |
Dist Syst. Layer (middleware)
    | OS    |
    | OS    |
    | OS    |

Network
```
Goals
A dist. system should make resources easily accessible; it should hide the fact that resources are distributed across a network; it should be open; and it should be scalable.

1. Making Resources Available

- Access remote resources
- Share them in a controlled and efficient way.

Examples:
- Printers
- Storage
- Data
- Files

Economy of scale.

Users → resources → collaboration.

2. Distribution Transparency

A distributed system that is able to present itself as if it were a single system is TRASPARENT.

Types:
- Accessibility: hide differences in representation.
- Location: hide where resources are.
- Migration: enabling movement of resources.
- Relocation: migration during use.
- Replication: multiple copies act as one.
Concurrency: hide resource in shared
Failure: hide the failure + recovery of resource

3. Openness
- offers services according to standard rules that describe the syntax + semantics of those services.

Defined through Interface: Interface Definition Language (IDL).

Interoperability

Portability

Usability

4. Scalability
a. A system be scalable with respect to its size. e.g. add more resources + users.
b. Geographically scalable in one in which users and resources may lie far apart.
c. Administratively scalable.

Scalability problems:
- centralized services: implemented as a single server. => bottleneck.
- centralized data:
Centralized algorithms: bad idea.

Distributed / Decentralized algorithms:

1. No. my/c has complete information.
2. My/c make decision based on local info.
3. Failure of one my/c should not be a show-stopper.
4. No assumption about a global clock.

Synchronous vs Asynchronous communication

Common Pitfalls.

1. Network is reliable
2. NN is secure
3. NN is homogeneous
4. Topology is fixed
5. Latency is zero
6. BW is infinite
7. Transport cost is zero
8. One administrator
Challenges

Communication
- Send/Receive
- RPC
- ROI
- Message oriented vs Stream oriented

Processes
- Processes + thread + client/server
- Code migration

Naming:
- Complex

Synchronization
- Mutual exclusion
- Leader election
- Physical clocks vs Logical clocks
- Global state

Data Storage & Access:
- File sharing
- Memory sharing

Consistency & Replication:

Fault-tolerance:
- Dealing with failures
New Challenges

Mobile Systems

Sensor Networks

P2P Computing

Publish-Subscribe, Content Dist., Multimedia

Grid Computing

Cloud Computing