

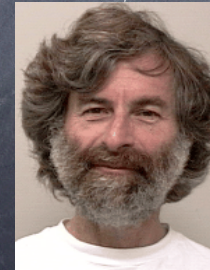
CS 380D Distributed Computing I

Lorenzo Alvisi
Anurag Agarwal

What is a distributed system?

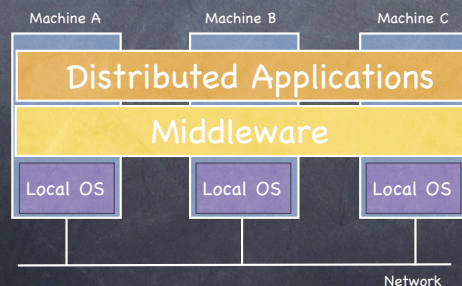
"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."

Leslie Lamport



What is a distributed system?

A distributed system is software through which a collection of independent computers appears to its users as a single, coherent system



Goals (and auto-goals) of a Distributed System

- Connecting Resources and Users
- Transparency
- Openness
- Scalability

Transparency

Transparency	Description
Access	Hides differences in data representation and invocation mechanisms
Location	Hides where an object resides
Migration	Hides from an object that object's location
Relocation	Hides from a client the change of location of an object to which the client is bound
Replication	Hides that an object may be replicated, with replicas at different locations
Concurrency	Hides coordination of activities between objects
Failure	Hides the failure and recovery of object
Persistence	Hides whether a resource is in memory or on disk

Openness

- ④ Easily interact with other open systems
- ④ Conform to well-defined interfaces
- ④ Achieve independence in heterogeneity wrt
 - > Hardware
 - > Platform
 - > Languages
- ④ Support different app/user-specific policies

Openness

- ④ Easily interact with other open systems
- ④ Conform to well-defined interfaces
- ④ Achieve independence in heterogeneity wrt
 - > Hardware
 - > Platform
 - > Languages
- ④ Support different app/user-specific policies

ideally, provide only mechanisms!

Scalability

- ④ **Size scalability**
 - number of users and processes
- ④ **Geographical scalability**
 - maximum distance between nodes
- ④ **Administrative scalability**
 - number of administrative domains

scaling

④ Distribute

- ④ partition data and computation across multiple machines
- ④ Java applets, DNS, WWW

④ Replicate

- ④ make copies of data available at different machines
- ④ mirrored web site, replicated fs, replicated db

④ Cache

- ④ allow client processes to access local copies
- ④ Web caches, file caching

consistency

A first course in Distributed Computing...

④ Two basic approaches

- ④ cover many interesting systems, and distill from them fundamental principles
- ④ focus on a deep understanding of the fundamental principles, and see them instantiated in a few systems

A few intriguing questions

- ④ How do we talk about a distributed execution?
- ④ Can we draw global conclusions from local information?
- ④ Can we coordinate operations without relying on synchrony?
- ④ For the problems we know how to solve, how do we characterize the "goodness" of our solution?
- ④ Are there problems that simply cannot be solved?
- ④ What are useful notions of consistency, and how do we maintain them?
- ④ What if part of the system is down? Can we still do useful work? What if instead part of the system becomes "possessed" and starts behaving arbitrarily: All bets are off?