Introduction Distributed Systems

Today

- Welcome
- Distributed systems definition, goals and challenges

What is a distributed system?

- Very broad definition
  - Collection of components, located at networked computers, that communicate & coordinate their actions only by passing messages
- Why do you want one?
  - Resource sharing, computation speedup, resilience, people interaction
- Implications
  - Concurrency – concurrent execution is the norm, concurrently executing processes must be coordinated
  - No global clock – coordination often depend on the notion of time but there are limits to clock synchronization
  - Independent failures – multiple and independent failures, sometime even hard to recognize them as such

Example classes of distributed systems

- Distributed computing systems
  - Clusters – set of similar off-the-shell workstations, running their own OS, interconnected by a high-speed LAN
  - Grids - each part potentially under a different administrative domain, hardware/software/network
- Distributing information systems
  - Distributed transaction systems
  - Enterprise application integration
- Mobile and ubiquitous computing
  - Small, networked devices are now part of distributed systems
  - Devices small enough to go unnoticed become part as well

Distributed systems challenges ...

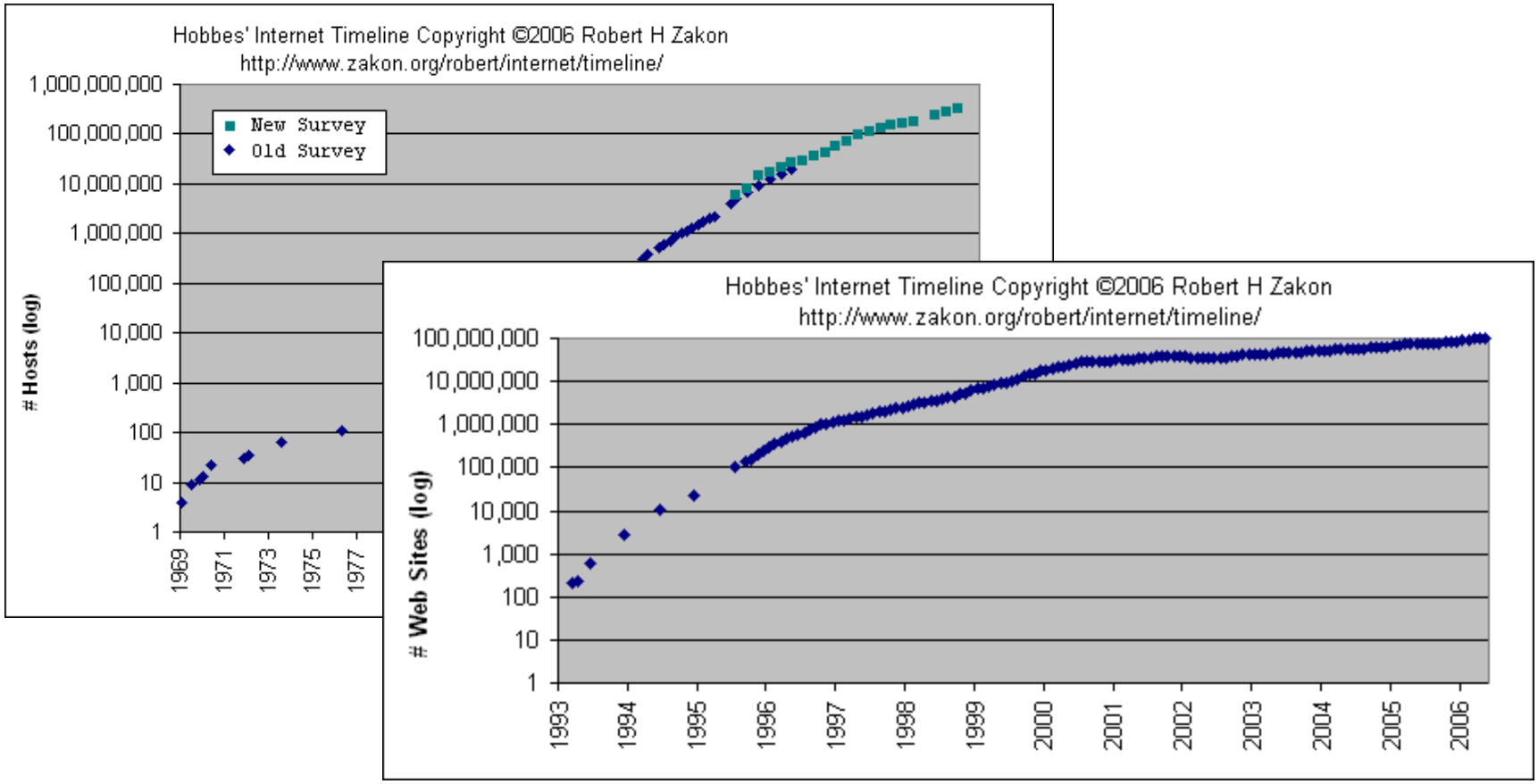
- Heterogeneity
  - Systems are built from a variety of components, mask those differences
- Security
  - Sharing, as always, introduces security issues
- Openness
  - Systems should be extensible – follow agreed-upon rules on component syntax & semantics for interoperability
- Scalability
  - In numbers (users and resources), geographic span and administration complexity

Distributed systems challenges

- Failure handling
  - Partial failures are hard to manage – each component may fail and must be aware of and handle other components' failures
- Concurrency
  - Sharing also brings in inconsistency from concurrency
- Transparency
  - Hide the fact that the system **is** distributed

Scalability problems

- Scalability in numbers of users and resources, geographic span and administration complexity
- Scalability in numbers



Scalability problems

- Scalability in numbers - limiting features
  - Centralized services – a single server for all users
  - Centralized data – a single HOSTS file for the Internet
  - Centralized algorithms – routing with complete information
- Characteristics of decentralized algorithms
  - No machine has complete information about the system state
  - Machines make decisions based only on local information
  - Failure of one machine does not ruin the algorithm
  - There is no implicit assumption that a global clock exists

Scalability problems

- Geographic scalability – from LANs to WANs
  - Synchronous communication, where requesting client blocks until it gets a response, makes it hard to scale
  - Communication is unreliable and nearly always point-to-point – e.g. broadcast for locating a service doesn't work
  - Centralized solutions are clearly an issue as well
- Administration complexity
  - Conflicting policies, with respect to resource usage, management and security, must be handle
  - E.g. Can you trust your sys admin? Can you trust users from another domain?

Scalability techniques

Three general classes of techniques for scaling

- Hiding communication latencies
  - Key for geographic scalability
  - How? Asynchronous communication, shipping code
- Distribution
  - Break up and distribute the system – e.g. Domain Name System, World Wide Web
- Replication/Caching
  - To increase availability, balance load and avoid communication latencies
  - The drawback – consistency

Challenges – Transparency

- Types of transparency
  - Access – What's data representation? Access local and remote resources using identical operations
  - Location – Where's the resource located?
  - Migration – Have the resource moved?
  - Relocation – Is the resource being move?
  - Replication – Are there multiple copies?
  - Concurrency – Is anybody else accessing the resource now?
  - Failure – Has it been working all along?
- Do we **really** want transparency?
  - Impossible – remote controlling a space ship
  - A bad idea – creating false expectations
  - Against application's goals – pervasive computing and location awareness

Challenges & pitfalls

Adding to the challenges, common false assumptions

- *The network is reliable*
- *… secure*
- *… homogenous*
- *The topology does not change*
- *Latency is zero*
- *Bandwidth is infinite*
- *Transport cost is zero*
- *There is one administrator*

Question 1

- *Use the world wide web as an example to illustrate the concept of resource sharing. Discuss two of the main challenges that must be dealt with in this context.*

This class – topics

- Introduction to distributed systems
- Peer-to-peer – definition, classes, uses
  - Dealing with an imperfect Internet
  - Content distribution networks for the people
- Challenges they may help us address
  - Censor resistance
  - Preserving anonymity
- Challenges they face
  - Peers and their service providers
  - That legal issue

Course outcomes

You should be able to …

- Explain the basic technological concepts behind P2P
- Understand the potential of P2P approaches to resource sharing
- Recognize the different approaches to P2P
- Explain some of the potential implications of P2P to the business enterprise
- Discuss some short and long-term future trends as seen by industry and academic researchers in the field

Class organization - grading

- Class participation 50%
- Assignments 50%
  - Readings 15% - you will select 3 of the paper listed for the week and write a summary report, following the guideline provided in the class website
  - Short in-class questions 15% - will gather in group of 2-3 to answer a given question, come up with a common answer for the group, but any dissenting opinion will be noted properly.
  - Essay 20% - you will write a 5-page essay examining the juxtaposition of 2-3 of a list of given topics (e.g. unstructured P2P, self-adaptation, overlay multicast, censorship, …)