Distributed Systems CH1 Notes

Keh-Harng Feng

August 23, 2017

1 Characterization of Distributed Systems

1.1 Introduction

Distributed System: A system in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages. Can be spatially separated by any distance. Why? **To share resources.**

Resources:

Hardware: disk drives, printers etc.

Software: files, databases, data objects etc.

Consequences from the definition:

- *Concurrency*: work can be done simultaneously by different components, sharing resources as necssary.
- *No global clock*: when programs need to cooperate the coordination is done by exchanging messages (no single clock synchronized for all components of the system).
- *Independent failures*: any of the component can fail but still leave the other components running. The failure of a component or unexpected termination of a program is not immediately made known to the other components with which it communicates.

1.2 Examples of Distributed Systems:

- Web Search: need to search and index an enormous amount of data. Google's solution involves very large numbers of networked computers in data centers, a distributed file and storage system, lock service (?) and a programming model that supports the management large parallel and distributed computations.
- Massive Multiplayer Online Games (MMOGs): very large numbers of users interact through the Internet in a persistent world. Some approaches are:

1. Client/Server: A central server with a single copy of game state distributed to clients. May use cluster of nodes to increase reliability.
2. Partition the universe to a number of servers (ie: based on geographical location). Easy to extend by adding more servers.
3. P2P technology (each client contributes resources such as storage and processing to acoomodate the game). See Chapters 2 and 10.

- Financial Trading: need to have real-time access to a wide range of data (ie: current share prices, trends, economic/political developments etc).The emphasis is to deliver events such as a drop in a share price reliably and quickly to very large numbers of clients. Client/Server doesn't work! Solution: *distributed event-based systems* (see Chapter 6).