CSCI 5673
Distributed Systems

Lecture Set Zero

Introduction

Lecture Notes by
Shivakant Mishra
Computer Science, CU Boulder
Last update: January 17, 2017

Introduction

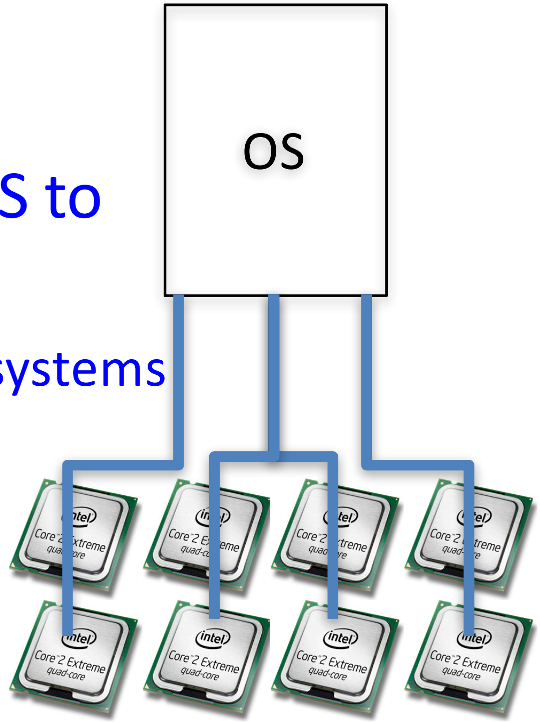
- What is a distributed system?
  - A distributed system is a collection of independent computing components, interconnected via a network, that are capable of collaborating on a task
  - Independent computing components
    - Each computing component has its own set of resources such as CPU, memory, disk, etc and can execute on its own
  - Interconnected: computing components can exchange information using the underlying network

Distributed Systems: Key Features

- No centralized control
- An integration of system services, presenting a transparent view of a multiple computer system with distributed control and resources

Introduction

- Distributed Systems is NOT…
  - Advanced Networking
    - Hardware routing architectures
    - Optical networking
    - TCP enhancements
  - Advanced Operating Systems
    - Multi-core systems – scaling an OS to 128 processors
      - tightly coupled vs. loosely coupled systems
    - Microkernels, etc.



Introduction

- Distributed Systems is NOT…
  - Network Security
    - Distributed Denial of Service (DDOS) attacks
    - Stuxnet virus
    - Rootkits
    - Advanced cryptography, ETC…
  - Software Engineering
    - Thread-based implementations of servers

Scope

- Assumptions about what you know:
  - Operating Systems
    - Processes, threads, synchronization, mem mgmt/virtual memory, file systems,
    - Distributed file systems, VMs
  - Network Systems
    - Ethernet/WiFi/MAC protocols, VLANs
    - IP routing, RIP, OSPF, BGP, IP multicast, MPLS
    - TCP/UDP transport layers
    - SSL/SSH, public/symm key, certificates, VPNs
    - HTTP, DNS, SMTP application layer protocols
    - Socket API and programming

Why distributed systems?

- Price/performance ratio
  - Don’t get twice computing for twice the price with multiprocessors
- Distributing applications may make sense
  - ATMs, graphics, remote monitoring
- Interactive communication & entertainment
  - work and play together:

    email, gaming, telephony, instant messaging
- Remote content
  - web browsing, music & video downloads, IPTV, file servers

Why distributed systems?

- Mobility
- Increased reliability
- Increased scalability
- Incremental growth

Problems

Designing distributed software is difficult

- Operating systems handling distribution
- Programming languages?
- Efficiency?
- Reliability?
- Administration?

Network

- disconnect, loss of data, latency

Security

- want easy and convenient access

Limitations of Distributed Systems

- Absence of a common global clock
- Absence of shared memory

- Absence of a common global clock
  - Notion of a common global time does not exist
  - Each component has its own local clock
    - Local clocks may drift from one another
    - It is not possible to perfectly synchronize all local clocks because of non-deterministic communication and scheduling delays

- Absence of a shared memory
  - Notion of a common memory (RAM or disk) does not exist
  - Each component may have its own memory
    - No temporal relation among read/write operations in different memory
    - It is not possible to perfectly synchronize all read/write operations

Limitations of Distributed Systems

- Due to the absence of a common global time and a common shared memory
  - It is not always possible to reason about the temporal order of events in a distributed system
  - It is not always possible to collect an up-to-date state of the entire system