

Distributed Systems

Mangat Rai Modi

Applift India

23rd August 2016

Overview

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

① PART I: Introduction

Terminology

Exercises

② PART II: Consensus Algorithms

References

Some Terms

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

To start, we will try to clear the concepts on some jargon.

- Distributed Systems
- Grid Computing
- Cloud computing

Distribution Systems

Distributed Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Definition:

A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages

Architectures: Client-Server, 3-Tier, N-Tier and Peer to Peer

Grid Computing

Distributed Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Definition:

Grid computing is the collection of computer resources from multiple locations to reach a common goal

How is it different than Supercomputer & Cluster?

Cloud Computing

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Definition:

A system where application makes requests to the resources through some service. Service maps these requests to physical resources which can be dynamically allocated when needed.

Questions:-

Distributed Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

- Is WWW Distributed System?
- Where does these three systems lie together?
- What is Applift RTB system like?

Consensus:-

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Consensus is the task of getting all processes in a group to agree on some specific value based on the votes of each processes. Consensus is a fundamental problem in fault tolerant distributed computing.

- synchronizing replicated state machines and making sure all replicas have the same (consistent) view of system state.
- electing a leader (e.g., for mutual exclusion)
- distributed, fault-tolerant logging with globally consistent sequencing
- managing group membership
- deciding to commit or abort for distributed transactions

Algorithm-I

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Two Phase Commit:

Either every node agrees or transaction will be aborted

Voting - A coordinator suggests a value to all nodes and gathers their responses (whether they agree to the value or not). For our scenario, a transaction coordinator asks whether all resource managers (database server instances) can commit to a transaction or not. The RMs reply with a yes or no.

Commit - If everyone agrees, the coordinator contacts all nodes to let them know the value is final. If even one node does not agree, inform all nodes that the value is not final.

Production Uses: MongoDB, Oracle-11g

Algorithm-II

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Three Phase Commit:

Adds a third phase in 2pc called "Prepare to Commit".

on receiving a yes from all nodes in voting step, the coordinator sends a "prepare to commit" message. The expectation is that nodes can perform work that they can undo, but nothing which cannot be undone. Each node acknowledges to the coordinator that it has received a "prepare to commit" message.

Algorithm-III

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Paxos:

Majority driven decision. Every node is a leader.

- **Assigning an order to the Leaders.** This allows each node to distinguish between the current Leader and the older Leader, which prevents an older Leader (which may have recovered from failure) from disrupting consensus once it is reached.
- **Restricting a Leader's choice in selecting a value.** Once consensus has been reached on a value, Paxos forces future Leaders to select the same value to ensure that consensus continues. This is achieved by having acceptors send the most recent value they have agreed to, along with the sequence number of the Leader from whom it was received. The new Leader can choose from one of the values received from the Acceptors, and in case no one sends any value, the Leader can choose its own value.

Production Uses: Aerospike, AWS, Apache Mesos, Oracle Nosql

Algorithm-IV

Distributed
Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

Raft

Elects a leader first using Timeouts, Majority driven Decision

- Each node have a random timeout and becomes candidate when timeout occurs.
- Each candidate node waits till random timeout to apply for leadership.
- Majority votes ensures that there is only one leader, otherwise process repeats.
- Leader keeps sending heartbeats to follower nodes to prevent them from becoming candidate.
- Leader sends data along with heartbeats.

Production Uses: Zookeeper, Redis Cluster

References

Distributed Systems

Mangat Rai
Modi

PART I:
Introduction
Terminology
Exercises

PART II:
Consensus
Algorithms
References

- ① <http://thesecretlivesofdata.com/raft/>
- ② <http://www.cs.utexas.edu/users/lorenzo/corsi/cs380d/papers/paper2-1.pdf>
- ③ <http://www.podc.org/influential/2001-influential-paper/>
- ④ <http://antirez.com/news/62>