Back To Module Home

Distributed Systems

0% completed

Introduction to Distributed Systems

System Models

Let's see the distributed system models.

We'll cover the following

- Nature of real-life distributed systems
- Making a generic model
  - Properties each system follows
- Categories of distributed systems
  - Synchronous system
  - Asynchronous system

Nature of real-life distributed systems#

Real-life distributed systems can differ drastically in many dimensions. These differences depend on factors like the network that deploys them, the hardware they run on, etc.

Thus, we need a common framework to solve problems generically. This way, we don't need to repeat the reasoning for the different variations of these systems.

Making a generic model#

To create a model of a distributed system, we must define several properties it must satisfy. If we prove an algorithm is correct for this model, we can be sure that it'll also be correct for all the systems that satisfy these properties.

Properties each system follows#

The main important properties in a distributed system concern the following:

- How the nodes of a distributed system interact with each other
- How the nodes of a distributed system can fail

Categories of distributed systems#

There are two main categories of distributed systems that depend on the nature of communication:

1. Synchronous systems
2. Asynchronous systems

Synchronous system#

A **synchronous system** is one where each node has an accurate clock, and there is a known upper bound on the message transmission delay and processing time. As a result, the execution is split into rounds. This way, every node sends a message to another node, the messages deliver, and every node computes based on the messages it receives During this, all nodes run in lock-step.

Asynchronous system#

An **asynchronous system** is one where there is no fixed upper bound on how long it takes for a node to deliver a message, or how much time elapses between consecutive steps of a node. The system nodes do not have a common notion of time and, thus, run at independent rates.

> The previous lesson discussed the challenges arising from network asynchrony.

So, it should be clear by now that the *synchronous* model is much easier to describe, program, and reason about. However, the *asynchronous* model is closer to real-life distributed systems, such as the Internet, where we cannot control all the components they involve. Also, there are minimal guarantees on the time it takes to send a message between two places.

> As a result, most of the algorithms we look at in this course assume an *asynchronous* system model.

Back

Measures of Correctness in Distribute…

Next

Types of Failures

Mark as Completed

Report an Issue