Decentralized Systems [WIP]

Fundamentals, Modeling, and Applications

Sommersemester 2024

✉ mail@nilslambertz.de
nilslambertz

Alle Angaben ohne Gewähr. Keine Garantie auf Vollständigkeit oder Richtigkeit.

1 Introduction

1.1 What is a distributed system?

1.1.1 Characteristics by van Steen and Tanenbaum

"*A distributed system is a collection of autonomous computing elements that appears to its users as a single coherent system.*" ([van Steen and Tanenbaum, 2016]), this results in two characteristics:

1. "**Collection of autonomous computing elements**": "*In practice, nodes are programmed to achieve common goals, which are realized by exchanging messages with each other*" ([van Steen and Tanenbaum, 2016])
2. "**Appears as a single coherent system**": Appears as a single large system

1.1.2 Aspects of characteristic 1

- "*we cannot assume that there is something like a global clock*" ([van Steen and Tanenbaum, 2016]), therefore the **synchronization and coordination** between participants must be worked out
- "*The fact that we are dealing with a collection of nodes implies that we may also need to manage the membership and organization of that collection*" ([van Steen and Tanenbaum, 2016]), therefore we need to think about identities and possible access-restrictions

1.1.3 Aspects of characteristic 2

- "*To assist the development of distributed applications, distributed systems are often organized to have a separate layer of software that is logically placed on top of the respective operating systems of the computers that are part of the system [...] leading to what is known as middleware*" ([van Steen and Tanenbaum, 2016])

1.1.4 Observations

- Distributing tasks and aggregating a result from them is not easy at all
- The coordination of those tasks is still **centralized**

1.2 What makes a distributed system a decentralized system?

According to **ISO/TC 307**: "*distributed system wherein control is distributed among the persons or organizations participating in the operation of the system*"

1.2.1 Three types of Decentralization (Vitalik Buterin)

Vitalik Buterin defines three types of Decentralization:

- **Architectural (de)centralization**: How many **physical computers** is a system made up of? How many of those computers can it tolerate breaking down at any single time?
- **Political (de)centralization**: How many **individuals or organizations** ultimately control the computers that the system is made up of?
- **Logical (de)centralization**: Does the **interface and data structures** that the system presents and maintains look more like a single monolithic object, or an amorphous swarm? One simple heuristic is: if you cut the system in half, including both providers and users, will both halves continue to fully operate as independent units?

1.2.2 Our definition of decentralized systems

- A decentralized system has political decentralization, where multiples parties are making their own independent decisions (they can still coordinate with each other)
- If a system is architecturally but not politically decentralized, we call it a distributed system
- Decentralized systems can be **logically decentralized or centralized**
- Decentralized systems can be open systems (anybody can participate) or closed systems

1.3 Reasons for decentralization

1.3.1 Reasons for architectural decentralization

- **Latency**
- **Scalability**: Scale number of machines running the system
- Increase **fault tolerance** and **availability**, remove single point of failures
- Increase **attack resistance**, because no central points exist

1.3.2 Reasons for political decentralization

- **Collusion resistance**: It is harder for participants to collude in ways that benefit a small group at the expense of other participants
- **Power** can be distributed "equally"

1.3.3 Reasons for logical decentralization

Logical decentralization is not always possible or even wanted, especially in use cases we cover. An example are **Distributed Ledgers**, where the goal is to have one commonly agreed state of the system at any point in time.

1.4 Challenges of decentralization

Decentralized systems come with risks/challenges to avoid harm to the system or its participants:

- **Time and synchrony**: Do we have a global clock? Is the communication synchronous or asynchronous?
- **Behaviour** of nodes: Can we handle arbitrary behaviour? How many faulty nodes can we tolerate?
- **Identity**: Do we have an open system? Are nodes (identities) known?

References

[van Steen and Tanenbaum, 2016] van Steen, M. and Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, 98(10):967–1009.