NT FILE chapter2.tex

Related Work cha:users$_{m}anual$

In this chapter we further detail the issues related to the Centralized Federated Learning Applications and explain why we should use Decentralized Federated Learning Applications as well as how are we expecting to deal with some issues related to delegating helpers to improve systems performance and avoid adverse effects on the functionality of the devices (the network nodes), but first some fundamental subjects will be introduced for a better general understanding of our proposal.

Decentralized Systems sec:decentralised$_{s}ystems$

Decentralized systems are fundamental in distributed computing, enabling tasks to be executed without a central authority.This architectural approach enhances scalability, fault tolerance, and resilience by distributing responsibilities across multiple nodes, allowing them to function autonomously while maintaining coordination. Such systems are particularly valuable in environments that require adaptive and robust communication between a multitude of participants, including P2P networks, blockchain technologies, and decentralized applications.

In decentralized systems, nodes can dynamically join, leave, or fail without significantly disrupting overall functionality.This is made possible by the use of adaptive protocols that ensure that the system remains operational. Data sharing, connectivity maintenance, and task distribution are facilitated by the deployment of algorithms and protocols. For example, membership protocols guarantee that nodes are aware of each other's presence, whereas peer-to-peer communication models facilitate direct interaction between nodes. Membership protocols serve to ensure that nodes are aware of each other's presence, while P2P communication models facilitate direct interaction between nodes.

Decentralized systems are a foundational element of modern technologies, including blockchain and decentralized P2P file-sharing platforms. These systems demonstrate the potential of decentralization to address scalability and reliability challenges in diverse domains.

Membership Protocols sub:membership$_{p}rotocols$

Membership protocols are vital for keeping decentralized swarm systems organized. These systems rely on many individual nodes that interact to accomplish tasks, and each node needs to know which other nodes are active at any moment.

enumerate

Epidemic and Scalable Global Membership Service:

The Epidemic Membership Service works much like spreading a rumor: each node shares information about its presence with a few nearby nodes. These nodes then pass along the update, which eventually reaches the whole network. This approach enables each node to keep a partial view of the network, meaning it stays updated on some nodes but not all. This setup keeps the system scalable and ready to handle changes in swarm size. Since updates move organically throughout the network, nodes can join or leave without burdening any single part of the system.

Decentralized Membership Protocols:

In larger swarms, keeping track of every single node isnt practical. Decentralized Membership Protocols simplify this by letting each node manage enough information to stay connected without needing to know all nodes. For example, the HyParView protocol organizes nodes into two lists: an active view, which contains frequently updated links to a few critical nodes, and a passive view, a backup list of other nodes. The active view keeps each node actively connected, while the passive view acts as a fallback in case of node failure. This two-layered approach makes HyParView resilient and allows quick recovery if nodes unexpectedly drop out. The protocol is often used within frameworks like Babel, which supports secure and automatic connections.