

Relatório Técnico: Laboratório III

Comunicação Indireta Pub/Sub e Eleição de Coordenador

Marcel Sant’Ana Claudiano
Arthur Manenti

Turma: Computação Noturno / 8

01 de Dezembro de 2025

1 Introdução

Este relatório descreve a implementação de um sistema distribuído para mineração de criptomoedas simulada, utilizando o paradigma de comunicação indireta *Publish/Subscribe*. O objetivo principal foi desenvolver um sistema onde múltiplos nós (participantes) cooperam para eleger um coordenador e competem para resolver desafios criptográficos baseados em *Proof of Work* (PoW), garantindo consistência e sincronia sem acoplamento direto entre os processos.

2 Metodologia de Implementação

O sistema foi desenvolvido na linguagem **Python**, utilizando a biblioteca **Paho-MQTT** para comunicação assíncrona. A arquitetura baseia-se em uma Máquina de Estados Finita com os estados: *Init*, *Election*, *Challenge* (Líder) e *Running* (Minerador).

2.1 Comunicação e Middleware

Utilizou-se o protocolo MQTT através do *broker* público `broker.emqx.io`. A troca de mensagens ocorre via tópicos hierárquicos JSON:

- `sd/init`: Descoberta de nós e sincronização inicial.
- `sd/voting`: Envio de votos para eleição do líder.
- `sd/challenge`: Publicação dos desafios pelo líder.
- `sd/solution`: Envio de propostas de solução pelos mineradores.
- `sd/result`: Confirmação do vencedor da rodada.

2.2 Algoritmo de Eleição

Para a definição do coordenador, implementou-se uma variação do algoritmo *Bully*. Cada nó gera um ID de votação aleatório. Após a fase de sincronização, todos compartilham seus votos. O nó que possuir o maior *VoteID* é eleito líder. Em caso de empate, utiliza-se o identificador do cliente (*ClientID*) como critério de desempate, garantindo que sempre haja um único líder.

2.3 Mineração e Controle de Concorrência

O desafio criptográfico consiste em encontrar uma *string* (nonce) cujo hash SHA-1 inicie com uma quantidade N de zeros (dificuldade).

Tratamento de Condição de Corrida: Durante os testes, identificou-se que múltiplos mineradores poderiam encontrar a solução quase simultaneamente. Para evitar o problema de “gasto duplo” (múltiplos vencedores para a mesma transação), implementou-se uma trava lógica no Líder. Assim que a primeira solução válida é processada, o estado da transação é marcado como encerrado, e soluções subsequentes para o mesmo ID são descartadas até o início do próximo ciclo.

3 Testes e Execução

Os testes foram realizados instanciando 3 processos distintos em ambiente Windows, conectados via TCP/IP ao *broker* remoto.

3.1 Cenário de Teste

1. Inicialização simultânea de 3 nós.
2. Verificação de descoberta (Broadcast de Init).
3. Eleição automática após detecção de quórum.
4. Execução contínua de ciclos de mineração.

4 Resultados e Observações

O sistema comportou-se de maneira estável. O mecanismo de retransmissão no estado *Init* garantiu que, mesmo com atrasos na abertura dos terminais, todos os nós conseguissem se sincronizar antes da eleição.

Nota sobre a Dificuldade (Vídeo vs. Código): A especificação do projeto exige que a dificuldade do desafio varie entre $[1..20]$ zeros. O código fonte entregue respeita rigorosamente este requisito (`randint(1, 20)`). Entretanto, para a gravação do vídeo de demonstração, o intervalo foi temporariamente ajustado para $[1..5]$. Esta alteração foi necessária apenas para fins didáticos, permitindo que a mineração ocorresse em tempo real durante a gravação, demonstrando múltiplos ciclos de validação e troca de mensagens sem longos períodos de espera computacional.

5 Conclusão

O trabalho cumpriu todos os requisitos propostos. A utilização de MQTT facilitou o desacoplamento dos nós, e a implementação de mecanismos de bloqueio no controlador assegurou a integridade das transações distribuídas, evitando inconsistências comuns em sistemas concorrentes.