



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Tecnologia em Análise e Desenvolvimento de Sistemas

Trabalho Prático:
Implementação do Protocolo Two Phase Commit
(2PC)

ITALLO OTAVIO
MANUELA DE SOUZA

MAIO – 2025
CONTAGEM - MG

1. Arquitetura do Sistema

O sistema foi desenvolvido com uma arquitetura baseada no modelo cliente-servidor distribuído. O processo central é o coordenador, que atua como cliente iniciando conexões com os servidores participantes. Cada participante escuta conexões via socket TCP e responde conforme o protocolo. O coordenador toma decisões com base nas respostas dos participantes. Cada participante simula uma instância distribuída com armazenamento próprio em arquivo de texto.

No desenvolvimento deste sistema distribuído que implementa o protocolo Two Phase Commit (2PC), optamos inicialmente por utilizar o endereço **localhost** em todos os participantes e no coordenador. Essa escolha se deveu ao fato de estarmos realizando os testes em uma única máquina, porém em terminais diferentes. Dessa forma, conseguimos simular a comunicação entre múltiplos nós sem a necessidade de infraestrutura física distribuída, facilitando o desenvolvimento e depuração do sistema.

Entretanto, para que o sistema funcione em um ambiente verdadeiramente distribuído, com participantes rodando em máquinas diferentes dentro de uma mesma rede local, é necessário substituir o endereço **localhost** pelos **endereços IP reais** dessas máquinas.

2. Funcionamento do Protocolo Implementado

Foi implementado o protocolo Two-Phase Commit (2PC), que funciona em duas etapas:

- *Fase 1 - Votação (Prepare)*: o coordenador envia uma mensagem "prepare" para todos os participantes. Cada um deve responder "sim" se estiver pronto para realizar a transação ou "não" se não puder.
- *Fase 2 - Decisão (Commit ou Abort)*: se todos os participantes responderem "sim", o coordenador envia "commit" para todos. Caso algum responda "não" ou falhe, o coordenador envia "abort" para cancelar a transação.

3. Metodologia Utilizada para Escrita em Arquivos Distribuídos

Cada participante possui um arquivo de texto próprio. Quando recebem a mensagem "commit", eles escrevem a transação no arquivo. A escrita é feita em modo append, adicionando uma nova linha para cada transação aprovada. Caso recebam "abort" ou falhem, nada é escrito. Isso garante que somente transações aprovadas por todos sejam persistidas.

4. Estratégias para Simulação de Falhas e Como Elas Afetam a Decisão

Foi implementada uma opção para simular falhas nos participantes. Antes de votar, o participante pode "falhar" intencionalmente, simulando quedas de rede ou travamentos. Nessa situação, o participante não responde ao coordenador. O coordenador, ao perceber a ausência de resposta, interpreta como falha e automaticamente envia "abort" para os demais, garantindo que nenhum participante finalize a transação de forma isolada.

5. Desafios Enfrentados e Decisões de Projeto

Um dos principais desafios foi gerenciar a comunicação via sockets entre múltiplos processos e lidar com exceções, especialmente durante simulações de falha. Foi decidido manter a implementação simples e didática, sem uso de bibliotecas complexas, priorizando clareza e

controle manual das falhas. Além disso, foi adicionado um sistema de log para registrar as etapas do protocolo e facilitar a verificação dos resultados.

6. Exemplos de Execução com Resultados Esperados

Para executar o sistema, é necessário abrir 4 terminais:

- Terminal 1: Coordenador
- Terminal 2-4: Participantes

Antes de qualquer execução, os arquivos de participantes e o log estarão vazios ou não existirão:

participante_Participante1.txt: (não existe ou vazio)
participante_Participante2.txt: (não existe ou vazio)
participante_Participante3.txt: (não existe ou vazio)
log_2pc.txt: (não existe ou vazio)

Cenário 1: Transação Bem-Sucedida (Todos Votam "sim")

Comandos de Execução

Terminal 1 (Participante 1)

```
python participante.py Participante1 localhost 5001
```

Terminal 2 (Participante 2)

```
python participante.py Participante2 localhost 5002
```

Terminal 3 (Participante 3)

```
python participante.py Participante3 localhost 5003
```

Terminal 4 (Coordenador)

```
python principal.py
```

Interação Esperada

No Terminal 4 (Coordenador):

```
==== SIMULADOR 2PC DISTRIBUÍDO ====
```

```
Digite a mensagem da transação que deseja realizar: Primeira transação de
```

teste

==== INÍCIO DA VOTAÇÃO ====

[Coordenador] Enviando pedido de voto para Participante1 em localhost:5001

[Coordenador] Recebeu voto 'sim' de Participante1

[Coordenador] Enviando pedido de voto para Participante2 em localhost:5002

[Coordenador] Recebeu voto 'sim' de Participante2

[Coordenador] Enviando pedido de voto para Participante3 em localhost:5003

[Coordenador] Recebeu voto 'sim' de Participante3

===== RESUMO DOS VOTOS =====

Participante1 votou: SIM

Participante2 votou: SIM

Participante3 votou: SIM

[Fim da Fase de Votação] Iniciando Fase de Decisão...

[Coordenador] Todos votaram 'sim'. Enviando COMMIT...

Resultado da transação: COMMIT - Todos os arquivos foram atualizados com a transação.

Em cada Terminal de Participante, quando solicitado:

[ParticipanteX] Deseja simular falha antes de votar? (sim/nao): nao

ParticipanteX, deseja realizar a transação? (sim/nao): sim

[ParticipanteX] Commit recebido. Atualizando arquivo...

[ParticipanteX] Commit finalizado.

Conteúdo dos Arquivos Após a Transação

participante_Participante1.txt:

Primeira transação de teste

participante_Participante2.txt:

Primeira transação de teste

participante_Participante3.txt:

Primeira transação de teste

log_2pc.txt:

[2025-05-18 12:45:00] [Coordenador] Iniciando votação para a transação:

Primeira transação de teste

[2025-05-18 12:45:05] Participante1 votou: sim

[2025-05-18 12:45:10] [Coordenador] Recebeu voto 'sim' de Participante1

[2025-05-18 12:45:15] Participante2 votou: sim

[2025-05-18 12:45:20] [Coordenador] Recebeu voto 'sim' de Participante2

[2025-05-18 12:45:25] Participante3 votou: sim

[2025-05-18 12:45:30] [Coordenador] Recebeu voto 'sim' de Participante3

[2025-05-18 12:45:35] [Coordenador] Todos votaram 'sim'. Commit sendo enviado.

[2025-05-18 12:45:40] Participante1 escreveu no arquivo: Primeira transação de teste

[2025-05-18 12:45:45] Participante2 escreveu no arquivo: Primeira transação de teste

[2025-05-18 12:45:50] Participante3 escreveu no arquivo: Primeira transação de teste

[2025-05-18 12:45:55] Resultado da transação: COMMIT - Todos os arquivos foram atualizados com a transação.

Cenário 2: Transação Abortada (Um Participante Vota "nao")

Interação Esperada

No Terminal 4 (Coordenador):

==== SIMULADOR 2PC DISTRIBUÍDO ====

Digite a mensagem da transação que deseja realizar: Segunda transação de teste

==== INÍCIO DA VOTAÇÃO ====

```
[Coordenador] Enviando pedido de voto para Participante1 em localhost:5001
[Coordenador] Recebeu voto 'sim' de Participante1
[Coordenador] Enviando pedido de voto para Participante2 em localhost:5002
[Coordenador] Recebeu voto 'nao' de Participante2
[Coordenador] Enviando pedido de voto para Participante3 em localhost:5003
[Coordenador] Recebeu voto 'sim' de Participante3

===== RESUMO DOS VOTOS =====
Participante1 votou: SIM
Participante2 votou: NAO
Participante3 votou: SIM

[Fim da Fase de Votação] Iniciando Fase de Decisão...
[Coordenador] Pelo menos um voto foi 'não' ou houve falha. Enviando ABORT...

Resultado da transação: ABORT - Transação cancelada, nenhum arquivo foi
alterado.
```

Em cada Terminal de Participante, quando solicitado:

Participante1 e Participante3

```
[ParticipanteX] Deseja simular falha antes de votar? (sim/nao): nao
ParticipanteX, deseja realizar a transação? (sim/nao): sim [ParticipanteX]
Abort recebido. Transação cancelada.
```

Participante2

```
[Participante2] Deseja simular falha antes de votar? (sim/nao): nao
Participante2, deseja realizar a transação? (sim/nao): nao [Participante2]
Abort recebido. Transação cancelada.
```

Conteúdo dos Arquivos Após a Transação

```
participante_Participante1.txt:
```

Primeira transação de teste

participante_Participante2.txt:

Primeira transação de teste

participante_Participante3.txt:

Primeira transação de teste

log_2pc.txt:

[Entradas anteriores...]

[2025-05-18 12:46:00] [Coordenador] Iniciando votação para a transação:
Segunda transação de teste

[2025-05-18 12:46:05] Participante1 votou: sim

[2025-05-18 12:46:10] [Coordenador] Recebeu voto 'sim' de Participante1

[2025-05-18 12:46:15] Participante2 votou: nao

[2025-05-18 12:46:20] [Coordenador] Recebeu voto 'nao' de Participante2

[2025-05-18 12:46:25] Participante3 votou: sim

[2025-05-18 12:46:30] [Coordenador] Recebeu voto 'sim' de Participante3

[2025-05-18 12:46:35] [Coordenador] Abortando transação devido a voto 'não'
ou falha.

[2025-05-18 12:46:40] Resultado da transação: ABORT - Transação cancelada,
nenhum arquivo foi alterado.

Cenário 3: Transação Abortada (Falha em um Participante)

Interação Esperada

No Terminal 4 (Coordenador):

==== SIMULADOR 2PC DISTRIBUÍDO ====

Digite a mensagem da transação que deseja realizar: Terceira transação de teste

==== INÍCIO DA VOTAÇÃO ====

[Coordenador] Enviando pedido de voto para Participante1 em localhost:5001

[Coordenador] Recebeu voto 'sim' de Participante1

[Coordenador] Enviando pedido de voto para Participante2 em localhost:5002

[Coordenador] Falha na comunicação com Participante2: [Erro de conexão]

[Coordenador] Enviando pedido de voto para Participante3 em localhost:5003

[Coordenador] Recebeu voto 'sim' de Participante3

===== RESUMO DOS VOTOS =====

Participante1 votou: SIM

Participante2 votou: FALHA

Participante3 votou: SIM

[Fim da Fase de Votação] Iniciando Fase de Decisão...

[Coordenador] Pelo menos um voto foi 'não' ou houve falha. Enviando ABORT...

Resultado da transação: ABORT - Transação cancelada, nenhum arquivo foi alterado.

Em cada Terminal de Participante, quando solicitado:

Participante1 e Participante3

[ParticipanteX] Deseja simular falha antes de votar? (sim/nao): nao
ParticipanteX, deseja realizar a transação? (sim/nao): sim [ParticipanteX]
Abort recebido. Transação cancelada.

Participante2

[Participante2] Deseja simular falha antes de votar? (sim/nao): sim
[Participante2] Falhou antes de votar!

Conteúdo dos Arquivos Após a Transação

participante_Participante1.txt:

Primeira transação de teste

participante_Participante2.txt:

Primeira transação de teste

participante_Participante3.txt:

Primeira transação de teste

log_2pc.txt:

[Entradas anteriores...]

[2025-05-18 12:47:00] [Coordenador] Iniciando votação para a transação:
Terceira transação de teste

[2025-05-18 12:47:05] Participante1 votou: sim

[2025-05-18 12:47:10] [Coordenador] Recebeu voto 'sim' de Participante1

[2025-05-18 12:47:15] Participante2 falhou antes de votar!

[2025-05-18 12:47:20] [Coordenador] Falha na comunicação com Participante2:
[Erro de conexão]

[2025-05-18 12:47:25] Participante3 votou: sim

[2025-05-18 12:47:30] [Coordenador] Recebeu voto 'sim' de Participante3

[2025-05-18 12:47:35] [Coordenador] Abortando transação devido a voto 'não'
ou falha.

[2025-05-18 12:47:40] Resultado da transação: ABORT - Transação cancelada,
nenhum arquivo foi alterado.

