

UNIVERSIDADE DE SÃO PAULO

Sistemas de Informação - EACH

DAVI FERNANDES MENESES DA SILVA, N. USP: 14760241

GABRIEL GONÇALVES DE SOUZA RIBEIRO, N. USP: 14691679

Desenvolvimento de Sistemas de Informação Distribuídos

Relatório da parte III do trabalho semestral

SÃO PAULO, SP

2025

1. Introdução

Nesta parte III do exercício, aprimoramos o sistema de compartilhamento de arquivos entre peers para permitir o **download fragmentado por chunks paralelizados**, o **agrupamento de arquivos por nome e tamanho**, além da **coleta e exibição de estatísticas de desempenho** com base em parâmetros como tamanho do chunk e número de peers.

2. Comando Alterar tamanho de chunk

O sistema foi inicializado com o tamanho de chunk padrão de **256 bytes**, conforme especificado no enunciado.

O comando [6] Alterar tamanho de chunk permite que o usuário altere esse valor dinamicamente durante a execução. O novo valor é armazenado localmente e influencia diretamente o particionamento de arquivos durante o download.

3. Alterações no funcionamento do comando Buscar

O comando [4] Buscar arquivos foi modificado para:

- Agrupar arquivos com mesmo nome e tamanho, mesmo que provenientes de peers diferentes;
- Exibir todos os peers que possuem cópias válidas do arquivo;
- Dividir o arquivo em chunks, com base no tamanho atual configurado;
- Distribuir os downloads dos chunks entre múltiplos peers em paralelo;
- Reunir os chunks e reconstruir o arquivo localmente, após decodificação base64.

Essa abordagem melhora a taxa de transferência e simula paralelismo real de redes distribuídas.

4. Comando Exibir estatísticas

Foi implementado o comando [5] Exibir estatísticas], que exibe:

Tam. chunk	N. peers	Tam. arquivo	Tempo [s]	Desvio
------------	----------	-----------------	-----------	--------

O sistema coleta os seguintes dados automaticamente a cada download:

- Tamanho do chunk usado;
- Número de peers que participaram do download;
- Tamanho do arquivo em bytes;
- Tempo total gasto para o download (excluindo tempo de escrita);
- Cálculo do tempo médio e desvio padrão para cada tripla de parâmetros.

5. Decisões de Projeto e Refatorações

- A função `buscar_arquivos()` foi refatorada para lidar com agrupamento, distribuição de chunks e contagem de tempo;
- Foi criada uma estrutura global `estatisticas = {}` com a tripla (chunk, n_peers, tamanho) como chave;
- Os downloads foram paralelizados com `threading.Thread`, distribuindo os chunks por round-robin;
- No tratamento da mensagem DL, o arquivo é lido em binário e o chunk correspondente é extraído e enviado;
- No tratamento da mensagem FILE, os chunks são montados e armazenados até a reconstrução final do arquivo;

- Foi adicionada a função `alterar_tamanho_chunk()` e `exibir_estatisticas()` ao menu principal.

6. Testes realizados

- Criamos arquivos de 1 KB, 10 KB e 100 KB para simular downloads reais;
- Testamos com múltiplos peers com cópias de arquivos idênticos;
- Variamos o tamanho dos chunks (1, 256, 512 bytes) para medir impacto na velocidade;
- Simulamos ambiente com 2, 3 e 4 peers ativos para validar ganho com paralelismo;
- Realizamos múltiplos downloads para obter amostras para estatísticas.

7. Dificuldades enfrentadas

- Garantir que todos os chunks fossem recebidos antes de montar o arquivo final;
- Ajustar a lógica de round-robin para funcionar corretamente com qualquer número de chunks e peers;
- Evitar duplicação ou perda de dados em downloads simultâneos;
- Garantir que os peers respondessem corretamente a mensagens DL com leitura precisa dos arquivos.

8. Instruções de execução

Terminal 1:

```
python main.py 127.0.0.1:9001 vizinhos_peer1.txt diretorio
```

Terminal 2:

```
python main.py 127.0.0.1:9002 vizinhos_peer2.txt diretorio2
```

Terminal 3:

```
python main.py 127.0.0.1:9003 vizinhos_peer3.txt diretorio3
```

Terminal 4:

```
python main.py 127.0.0.1:9004 vizinhos_peer4.txt diretorio4
```