

UNIVERSIDADE ESTADUAL DE MATO GROSSO DO SUL
CURSO DE CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA DE ALGORITMOS PARALELOS E DISTRIBUÍDOS

TRABALHO DE SISTEMAS DISTRIBUÍDOS

TEMA: DESENVOLVER UM SISTEMA P2P PARA COMPARTILHAMENTO DE
DOCUMENTOS PDF.

DOURADOS – MS, 20/09/2024

0) INTRODUÇÃO

O compartilhamento de arquivos Peer to Peer (P2P) é uma abordagem diferente para compartilhar arquivos em comparação ao download de arquivos tradicional. Nesta abordagem, estamos tentando alcançar o local do arquivo sem qualquer envolvimento com um servidor central. Portanto, nesta abordagem, a mesma máquina funcionará como um servidor e um cliente, dependendo da função durante um compartilhamento de arquivo. Os recursos de compartilhamento podem ser únicos ou duplicados porque vários servidores podem ter o mesmo conteúdo. O compartilhamento de arquivos P2P está se tornando popular rapidamente em todo o mundo devido à disponibilidade do grande número de aplicativos de compartilhamento de arquivos. No entanto, todos esses aplicativos são baseados em protocolos de compartilhamento de arquivos P2P bem conhecidos.

A abordagem de compartilhamento de arquivos P2P permite que os clientes enviem solicitações para o arquivo que você deseja baixar. Então, ele fornece um procedimento para localizar o arquivo consultando outros computadores (pares) conectados pela internet. Quando o cliente encontra o local, ele começa a baixar o arquivo. Agora, outros computadores (pares) poderão obter o arquivo do computador do cliente. Este conceito é baseado no problema básico de pesquisa.

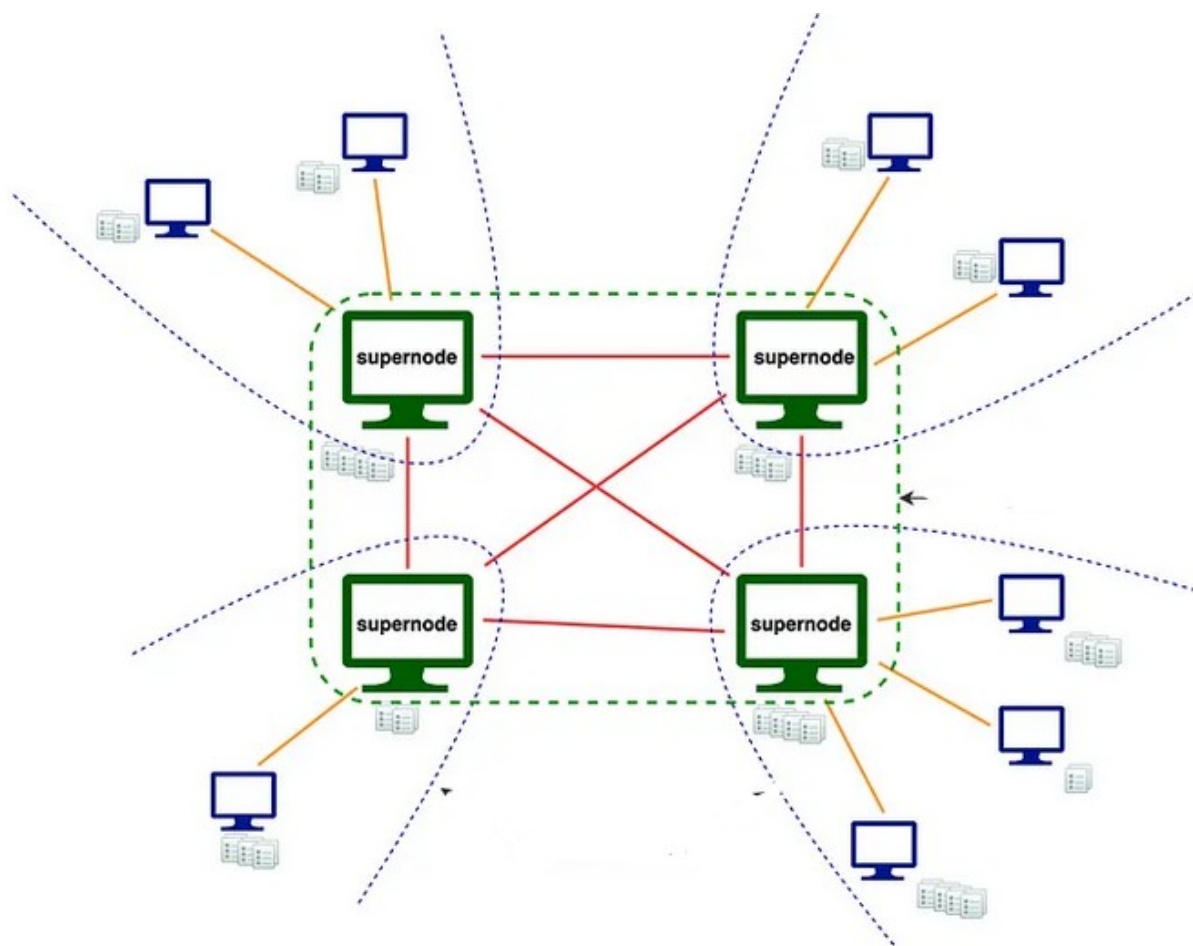
1) DESCRIÇÃO DO MODELO P2P

O nosso modelo usará uma abordagem híbrida que combina abordagens centralizadas e descentralizadas.

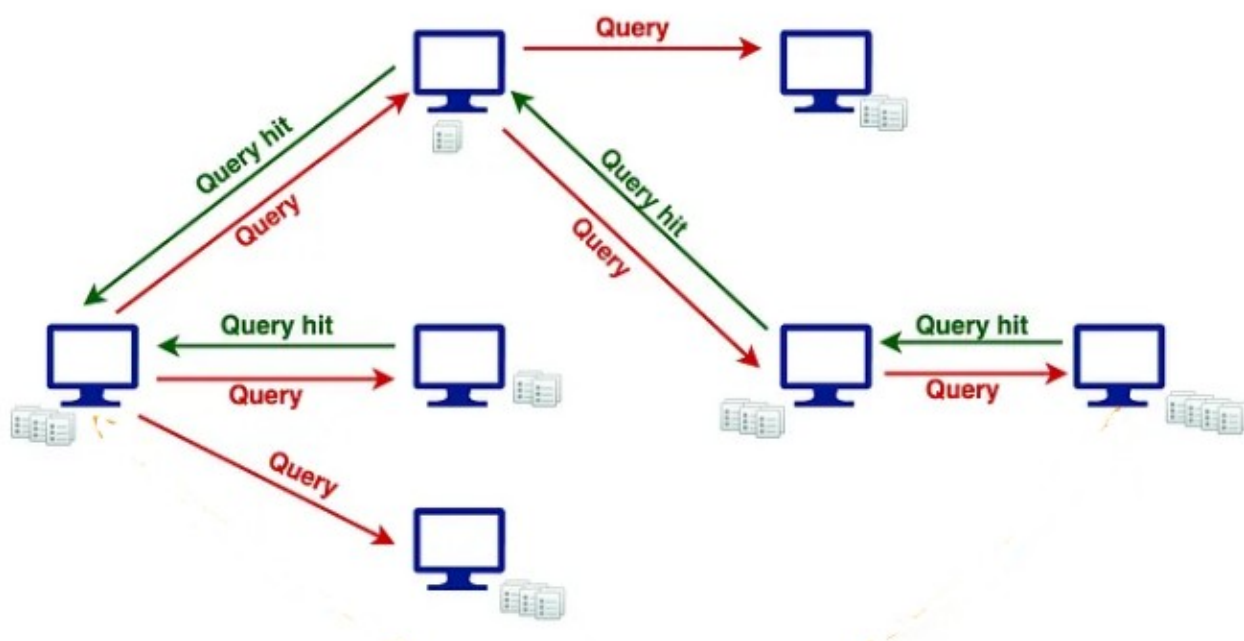
Quando um cliente em particular está procurando um arquivo, primeiro essa solicitação é enviada para um super nó. Agora, o super nó verifica seu arquivo de índice. Se o arquivo existir em seu índice, o super nó envia o local do arquivo para o cliente e a pesquisa é concluída a partir desse ponto. Senão, os super nós inundarão a consulta para outros super nós. Se o arquivo existir em um dos índices dos outros super nós, esse super nó enviará o local para o super nó originado (aquele que gerou a primeira solicitação). Finalmente, o super nó envia o local para o cliente.

Mais detalhadamente, a comunicação entre os super nós se dará por meio de inundação (broadcast) de consultas que não tem um servidor centralizado para manter o controle dos arquivos. Ou seja, ninguém sabe onde o arquivo está e por isso a consulta é necessária. Quando alguém quer um arquivo específico, ele consultará seu vizinho. Se o arquivo não estiver disponível com eles, os vizinhos enviarão a solicitação para seus vizinhos. Da mesma forma, as mensagens serão inundadas entre os pares. Quando o arquivo for encontrado, o query hit o enviará pelo caminho reverso. Finalmente, o cliente obterá o arquivo do local recebido. Como você pode ver no diagrama acima, as mensagens são inundadas pela rede.

Em uma visão mais ampla, temos o seguinte cenário do super nó:



O modelo de pesquisa entre os super nós para descobrir onde está o arquivo procurado:



O modelo p2p consiste dos seguintes elementos:

- Cliente: Máquinas que possuem arquivos pdf e que estão na rede.
- Super nós: são responsáveis pelas buscas de recursos (documentos) armazenados em uma estrutura de arquivos. O super nó também elabora as informações de roteamento do nó servidor para o nó coordenador e, faz um broadcast das informações recebidas do nó coordenador para os nós folhas do nó servidor. O super nó deve possuir uma estrutura de arquivo contendo uma lista das máquinas que estão em seu domínio e, dos arquivos que cada máquina possui.
- Um nó coordenador (nó mestre) que coordena os recursos administrados. Cada super nó da rede se registra no nó mestre. Tem como objetivo prevenir cópias duplicatas de super nós e, gerenciar a entrada e saída de máquinas super nós.

2) PASSOS PARA ESTABELECIMENTOS DA REDE P2P

1) Configuração da Rede Super Nó.

O nó coordenador é inicializado primeiramente. Então o super nó é inicializado se registra no nó coordenador o qual gera uma chave (identificador) única que é repassada ao super nó. Após receber a resposta, o super nó envia um ACK ao nó coordenador avisando que recebeu a chave única. Quando todos os super nós se registrarem, o nó coordenador envia um broadcast informando que os registros estão finalizados a todos os super nós.

2) Configuração da rede.

Somente após todos os super nós serem registrados, a construção da rede começa. Nesta etapa, os clientes se registram no super nós. Estes enviam a cada nó servidor uma chave única hash com o seu endereço de ip associado (pode ser usado o SHA-1). Após receber a resposta de registro, o super nó envia um ACK a cada cliente registrado. Após todos os super nós registrarem seus clientes, cada super nó envia um pacote finish para todos os super nós e para o nó coordenador indicando o término da operação.

3) Armazenamento dos documentos pdf.

Quando os super nós receberem a solicitação de download/upload de documentos dos clientes (vamos trabalhar apenas com 3 clientes), ele (o super nó) atualizará a lista na estrutura de arquivos, extraíndo destes os metadados e as chaves hash usando o SHA-1.

4) Busca pelos documentos.

Uma vez que haja uma requisição de documentos, o super nó extrai os metadados para obter as chaves. Depois verifica sua estrutura de arquivos de índices. Se o arquivo existir em seu índice, o super nó envia o local para o cliente requisitante. Caso contrário, se o arquivo não estiver em seu índice, o super nó inunda a rede com consulta a outros super nós.

5) Inclusão de novos nós.

Depois que a rede estiver totalmente construída, se um novo nó (cliente ou super nó) desejar se juntar à rede, ele deverá proceder do mesmo modo que os passos executados no item 2. Vamos testar com a inclusão de um novo nó quando a rede já estiver em funcionamento. Lembre-se de utilizar o protocolo de inclusão de membros do Multicast. Este protocolo ajudará na hora do super nó acessar os clientes associados a ele.

6) Saída de um nó.

Quando um nó desejar sair, ele deverá enviar uma mensagem para o super nó (se for um cliente) ou para o nó coordenador (se for um super nó). Para o super nó, a tabela de roteamento é então atualizada e os registros do nó que saiu são deletados. Vamos testar a saída de um nó cliente e de um super nó. Lembre-se de utilizar o protocolo de saída de membros do Multicast.

3) TESTES E EXPERIMENTOS

Basicamente, o teste se propõe a testar o funcionamento da rede, gerando relatórios sobre cada etapa do funcionamento.

4) OBSERVAÇÃO IMPORTANTE 1

Durante o processo de eleição, desligaremos o nó coordenador (mestre). Conforme visto em sala de aula, um dos super nós identificará a ausência do nó mestre e chamará a eleição. Para essa etapa basta implementar o algoritmo visto em sala (alg. do valentão). Deixe uma estrutura condicional no seu código de super nó para o caso de um super nó ser eleito coordenador (mestre). Neste caso, essa estrutura condicional passará a funcionar assumindo as atividades da máquina mestre.

Uma técnica que não pode faltar em um SD é um algoritmo de trate a consistência do sistema abordado. Há três algoritmos muito interessantes que tratam de consistência: 2pc, 3pc e Paxos. Eu proponho a implementação do 2pc mas, caso queiram experimentar técnicas mais avançadas fiquem a vontade para as outras duas.

Durante o processo de eleição do novo coordenador, realizaremos a réplica da estrutura de arquivos do super nó que irá se tornar Coordenador (nó mestre) para um novo super nó que entrará na nossa rede. O processo de replicação passará pela técnica de consenso. O processo de aceite (consenso) pela máquina escolhida se dará da seguinte forma: O nó responsável pela estrutura de arquivos possui a estrutura (chave, valor) para cada documento. Cada máquina cliente pertencente ao domínio daquele super nó deverá fazer a soma dos campos valor de todos os pares da estrutura. Se o valor resultante da soma de cada máquina cliente for igual ao valor presente na estrutura de arquivos, a máquina vota sim; Caso contrário, a máquina vota não.

5) OBSERVAÇÃO IMPORTANTE 2

A linguagem de programação utilizada na implementação poderá ser: C, C++, Go, Elixir e Python. A API para comunicação em rede poderá ser da escolha do grupo (socket, mpi, rpc,...).

6) O QUE ENTREGAR

1. O código e a documentação devem ser entregues em um arquivo Zip. Dentro deste arquivo Zip devem conter um readme.txt com o nome do(s) autor(es) e o roteiro para a execução do código e, um arquivo PDF da documentação. Inclua todos os arquivos fontes (.c, .h, makefile (caso opte por fazer), não incluam executáveis ou arquivos objetos), em um único diretório. Você pode optar por fazer um Makefile para cada etapa do desenvolvimento. Embora não seja obrigatório, pode otimizar bastante o desenvolvimento do trabalho.
2. O nome dos arquivos devem seguir o seguinte da etapa de desenvolvimento. Por exemplo, busca_pdf.c
3. Submissões onde os programas não sigam as especificações de parâmetro e nome, makefiles não funcionando ou arquivos necessários faltando NÃO SERÃO CORRIGIDOS. Programas que não compilem também não serão corrigidos.

7) DOCUMENTAÇÃO

O texto da documentação deve ser breve, de forma que o professor possa entender o que foi feito no código sem ter que entender linha a linha dos arquivos. Implementações modularizadas deverão mencionar quais funções são implementadas em cada módulo ou classe. A documentação deverá conter os seguintes itens:

- O sumário do problema a ser tratado;
- Uma descrição dos algoritmos e dos tipos abstratos de dados, das principais funções, procedimentos e das decisões de implementação;
- Decisões de implementação que por ventura estejam omissos na especificação;
- Como foi tratada a comunicação entre as máquinas;
- Testes mostrando que o programa está funcionando de acordo com as especificações, seguidos de sua análise;
- Print Screens mostrando o correto funcionamento do sistema e exemplos de testes executados;
- Conclusão e referências bibliográficas.

8) AVALIAÇÃO

A avaliação do trabalho será composta pela execução dos programas desenvolvidos e pela

análise da documentação. Todos os alunos deverão apresentar e explicar em laboratório o trabalho.

Os seguintes itens serão avaliados:

- A qualidade do código (código bem organizado, estruturado, com comentários explicativos, variáveis com nomes intuitivos, modularidade, etc);
- Execução correta do código com entrada de testes a serem definidas no momento da avaliação. As entradas de testes irão exercitar a funcionalidade completa do código e, testar casos especiais ou de maior dificuldade de implementação, mas que devem ser tratados por um programa correto;
- Conteúdo da documentação que deve conter os itens mencionados anteriormente;
- Coerência e coesão da documentação (apresentação visual e organização, uso correto da língua portuguesa, qualidade textual e facilidade de compreensão);
- CASOS DE CÓPIA NÃO SERÃO TOLERADOS;
- Qualquer elemento que não esteja especificado neste documento, mas que tenha que ser inserido para que o simulador funcione, deve ser descrito na documentação de forma explícita.
- Pontuação:
 - Rede do sistema P2P configurada e transferindo arquivos: 2,0 pontos;
 - Inclusão de um novo membro no sistema: 2,0 pontos;
 - Saída de um membro do sistema: 2,0 pontos;
 - Funcionamento correto do algoritmo de eleição: 2,0 pontos;
 - Funcionamento correto do algoritmo de consenso: 2,0 pontos.

8) DATA DA ENTREGA

20/11/2024 – Essa é a data limite, sem possibilidade de adiamento.

O trabalho está configurado para ser feito em dupla. Qualquer formato diferente do definido deve ser conversado com o professor.