

Computação Distribuída: exemplos de uso

Sistema distribuído é uma porção de computadores, todos conectados em rede, coordenados por uma, ou mais, máquina(s) administradora(s) e que utilizam softwares que permitam o compartilhamento de seus recursos, como memória, processamento, hardware (como impressora etc) para um único fim, como quebrar um código, criptografar ou descobrir a melhor solução de um problema em matemática.

Ou seja, são vários computadores, que se comportam como um só, para fazer uma coisa de uma maneira rápida e eficiente, ou ter muito espaço de armazenamento.

São sistemas muito usados em supercomputadores, que fazem pesquisas científicas, previsões climáticas, descobrir novas partículas, controlar epidemias e outras doenças, fazer simulações e tudo...foi-se o tempo que ciência era feita no papel, lápis e laboratório.

O problema do caixeiro viajante:

Imagine que você se formou em computação e vai dar palestras em 'n' cidades. Você recebeu uma grana fixa pra viajar. Como é um cara esperto, resolveu criar um algoritmo para otimizar isso, traçar um plano para economizar o máximo possível nas viagens. Como? Em vez de sair viajando 'na louca', bolou um esquema (em C), pra calcular a rota mínima.

Por exemplo, se você for em 4 cidades, A, B, C e D, sabendo que pode ir de uma pra qualquer outra, que você mora na cidade A e tem que voltar pra ela, e o preço da passagem aérea depende somente da distância (quanto mais longe, mais caro), óbvio que existe um percurso melhor, dentre todos: ABCDA, ABDCA, ACBDA, ACDBA, ADBCA, ADCBA.

Pra resolver isso em C é só saber um pouco de Análise Combinatória.

Como você começa e termina em A, e não repete as outras cidades, é tudo uma combinação, sem repetição, das outras 'n-1' cidades. Na linguagem matemática, isso quer dizer: $(n-1)!$ Fatorial.

O tempo vai crescendo assustadoramente, levando minutos, horas e acima de 20 cidades, levaria ANOS pra você obter o melhor caminho.

Agora imagine pra estudar o genoma humano, que tem bilhões de partes, quanto tempo levaria até estudarmos e catalogarmos isso tudo, para descobrirmos curas de doenças e nos prevenirmos?

Porém, aqui que vem a beleza da Computação Distribuída: se unirmos a capacidade de processamento de 10% dos computadores residenciais (a maioria só usa para Youtube + Facebook), já teríamos descoberto uma porção de tratamentos, conheceríamos mais nosso corpo, planeta, teríamos previsto catástrofes etc.

Exemplo de Sistema distribuído: Clientes e Servidores

Em um jogo online, Worms Armageddon, por exemplo, um jogador atira com uma bazuca em uma direção e com uma determinada força, sabendo a força do vento, essa informação é enviada ao servidor, que checa e calcula onde esse tiro vai. O servidor envia essa informação para um outro jogador na mesma direção, assim são geradas novas informações sobre o cenário e distribuídas a todos os jogadores conectados.

Outro exemplo: P2P

P2P, ou peer-to-peer, é um ótimo exemplo de sistema distribuído.

São os famosos programas de torrent. Neles, não há uma central de rede, ninguém é de ninguém e é dando que se recebe, ou seja, quanto mais você compartilha, mais compartilham você, e mais rápido você recebe dados. E recebe de todos os lugares e pessoas do mundo.

Se está baixando um filme, por exemplo, pode está recebendo o começo dele de um pc na Argentina, 3 minutos do meio de um Canadense e os 5 minutos finais de um Chinês. O importante é receber.

Nas redes do tipo P2P, todos fazem papel de Servidor e Cliente

Diferente de sua faculdade ou empresa, não há uma administração ou sala-chefe. Todos são clientes e todos são servidores.

Em um servidor normal, você recebe seu arquivo, com banda limitada, do início ao fim. Em P2P, recebe aos poucos, com banda ilimitada (o máximo que você aguentar), aos pedaços...quem já usou torrent sabe o quanto é rápido e prático.

Claro, se houverem muitos usuários. Pois se não houver, fica horrível...aí seria melhor ter a rede normal, pelo menos teria sua banda garantida.

Isso tudo, de jogo, P2P, descobrir doenças, epidemias, previsões climáticas etc, é feito de forma extremamente rápida e eficiente.

É uma simplificação, mas isso envolve complexos algoritmos, que administram, muitas vezes, milhões de máquinas, com processadores diferentes, seres diferentes controlando a máquina, conexão diferente, em lugares diferentes do planeta.

De um certo modo, a computação distribuída, é uma Internet (Rede, na verdade) melhorada, em suas entranhas há diversas definições, propósitos e aspectos diferentes, diferenças na arquitetura, no funcionamento, na programação, nas 'camadas', além de, por exemplo, utilizar alguns aspectos positivos da rede convencional e da P2P.

Computação distribuída, devido sua otimização, também está relacionada com Supercomputadores.



IBM Sequoia, um dos computadores mais rápidos do mundo. Usado para pesquisas na área de energia nuclear, astronomia, previsões climáticas, genética e oceanografia.

Com base nos fundamentos apresentados em aula sobre os sistemas distribuídos, responda:

Resolução em grupo de 2 ou 3 alunos (mesmo grupo dos trabalhos anteriores)

Entrega: 10/09 (quinta-feira)

1. Sabemos que sistemas distribuídos possui várias vantagens sobre os sistemas centralizados. Indique e explique quais as vantagens.
2. O que é programação concorrente e quais são as principais vantagens?
3. Uma das desvantagens dos sistemas distribuídos é a falta de segurança. Discuta a afirmação acima e dê exemplos.
4. O que significa dizer que um Sistema Distribuído é transparente? Dê exemplos de diferentes tipos de transparências.
5. Qual(is) é (são) a(s) motivação(ões) para projetar Sistemas Distribuídos Abertos?
6. Cite características que possam determinar a diferença entre um Sistema Operacional de Rede e um Sistema Operacional Distribuído.
7. Quais características permitem definir uma arquitetura cliente/servidor de três camadas?
8. Qual é a diferença entre distribuição vertical e distribuição horizontal?
9. Qual a utilidade dos semáforos na programação paralela. Explique seu funcionamento.
10. Imagine uma situação em que os assinantes de uma lista de e-mail possuem replicas locais dos dados enviados (escritos) para essa lista. Como administrador do software que coordena a entrega de mensagens da lista, você deve garantir duas coisas:
A) Mensagens enviadas por um usuário devem ser lidas na ordem em que elas foram enviadas. Exemplo: se o usuário 1 envia as mensagens A e B, todos os outros usuários devem ler as mensagens nessa ordem: primeiro a mensagem A e depois a B;
B) Mensagens enviadas por usuários diferentes podem ser lidas em ordem diferente, no entanto, quando dois usuários tentarem ler, eles deverão obter o mesmo resultado. Exemplo: Se o usuário 1 enviar a mensagem P e o usuário 2 enviar a mensagem Q, os usuários 3 e 4 podem "ler" Q antes de P. No entanto, em um determinado intervalo de tempo, esses usuários irão "ler" apenas Q. Após isso, eles irão "ler" apenas P.
Diante do exposto, qual(is) modelo(s) de consistência você deve implementar no software de controle de mensagens dessa lista? Argumente a favor da sua escolha.