

Sistemas Distribuídos

Prof.viniciusedu@gmail.com

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”

Essa definição tem vários aspectos importantes. O primeiro é que um sistema distribuído consiste em componentes (isto é, computadores) autônomos. Um segundo aspecto é que os usuários, sejam pessoas ou programas, acham que estão tratando com único sistema. Isso significa que, de um modo ou de outro, os componentes autônomos precisam colaborar. Como estabelecer essa colaboração é o cerne do desenvolvimento de sistemas distribuídos. Observe que nenhuma premissa é adotada em relação ao tipo de computadores. Em princípio, até mesmo dentro de um único sistema, eles poderiam variar desde computadores centrais (mainframes) de alto desempenho até pequenos nós em redes de sensores. Da mesma maneira, nenhuma premissa é adotada quanto ao modo como os computadores são interconectados.

Uma característica importante é que as diferenças entre os vários computadores e o modo como eles se comunicam estão, em grande parte, ocultas aos usuários e também deveria ser relativamente fácil expandir ou aumentar a escala de sistemas distribuídos. Essa característica é uma consequência direta de ter computadores independentes, porém, ao mesmo tempo, de ocultar como esses computadores realmente fazem parte do sistema como um todo.

CURSO: Ciência da Computação / Sistemas de Informação

SÉRIE: 7º. semestre

TURNO: Diurno/Noturno

DISCIPLINA: Sistemas Distribuídos – 541E

CARGA HORÁRIA SEMANAL: 04 horas-aula

I – EMENTA

Definição e histórico de sistemas distribuídos. Arquiteturas. Processos. Comunicação. Nomeação. Sincronização. Tolerância a falha. Segurança. Cluster de Alto-desempenho Beowulf.Grids.

II – OBJETIVOS GERAIS

Apresentar ao aluno os principais aspectos que envolvem o projeto e a implementação de um sistema distribuído, com ênfase no Cluster Beowulf.

III – OBJETIVOS ESPECÍFICOS

Mostrar ao aluno: as principais características que um sistema operacional deve possuir, formas de organização, algoritmos envolvidos e formas de implementação.

IV – CONTEÚDO PROGRAMÁTICO

Módulo	Assunto
1	Definição de um sistema distribuído; Tipos de Sistemas Distribuídos;
2	Arquiteturas; Introdução a Supercomputação;
3	Processos;
4	Comunicação; Nomeação;
5	Sincronização; Tolerância a falha;
6	Segurança;
7	Clusters de Computadores
8	Considerações de Projeto – Cluster Beowulf; Configuração;
9	Gerenciamento de um cluster;
10	Programação Paralela;
11	Ferramentas de Análise de Desempenho e Otimizações;
12	Cluster X Grid.

V – ESTRATÉGIA DE TRABALHO

Aulas expositivas. Listas de exercícios para serem resolvidos fora da sala de aula para fixação dos assuntos abordados nas aulas expositivas. Proposta de projetos extracurriculares.

Sugestão de Projetos: Construção de clusters utilizando (MPICH, PVM e OpenMosix) para testar a performance utilizar o POV RAY para renderizar imagens.

VI – AVALIAÇÃO

Provas bimestrais e trabalhos.

VII – BIBLIOGRAFIA BÁSICA

COULOURIS, George. DOLLIMORE, Jean. KINDBERG, Tim. Sistemas Distribuídos - Conceitos e Projeto. Bookman Companhia Ed., 2007.

PITANGA, Marcos. Construindo Supercomputadores com Linux. Rio de Janeiro: Brasport, 2008.

TANENBAUM, Andrew S. Steen, Maarten Van. Sistemas Distribuídos. 2a.ed. São Paulo: Person Prentice Hall, 2007.

-

VIII - BIBLIOGRAFIA COMPLEMENTAR

MARQUES, José Alves. GUEDES, Paulo. Tecnologia de Sistemas Distribuídos. FCA, 1998.

RIBEIRO, Uira. Sistemas Distribuídos. Axcel Books, 2005.

SILBERSCHATZ, Abraham. GALVIN, Peter Baer. Gagne, Greg. Sistemas Operacionais com Java. Rio de Janeiro: Elsevier, 2008.

TANENBAUM, Andrew S. Sistemas Operacionais Modernos. Livros Técnicos e Científicos Editora S.A., 1999.

1. Introdução

1.1 Definição de um Sistema Distribuído

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”

Essa definição tem vários aspectos importantes. O primeiro é que um sistema distribuído consiste em componentes (isto é, computadores) autônomos. Um segundo aspecto é que os usuários, sejam pessoas ou programas, acham que estão tratando com único sistema. Isso significa que, de um modo ou de outro, os componentes autônomos precisam colaborar. Como estabelecer essa colaboração é o cerne do desenvolvimento de sistemas distribuídos. Observe que nenhuma premissa é adotada em relação ao tipo de computadores. Em princípio, até mesmo dentro de um único sistema, eles poderiam variar desde computadores centrais (mainframes) de alto desempenho até pequenos nós em redes de sensores. Da mesma maneira, nenhuma premissa é adotada quanto ao modo como os computadores são interconectados.

Uma característica importante é que as diferenças entre os vários computadores e o modo como eles se comunicam estão, em grande parte, ocultas aos usuários e também deveria ser relativamente fácil expandir ou aumentar a escala de sistemas distribuídos. Essa característica é uma consequência direta de ter computadores independentes, porém, ao mesmo tempo, de ocultar como esses computadores realmente fazem parte do sistema como um todo.

1.2 Transparência da distribuição

Uma meta importante de um sistema distribuído é ocultar o fato de que seus processos e recursos estão fisicamente distribuídos por vários computadores. Um sistema distribuído que é capaz de se apresentar a usuários e aplicações como se fosse apenas um único sistema de computador é denominado transparente.

1.2.1 Tipos de transparência

- Acesso
 - Oculta diferenças na representação de dados e no modo de acesso a um recurso.
- Localização
 - Oculta o lugar em que um recurso está localizado.
- Migração
 - Oculta que um recurso pode ser movido para outra localização.
- Relocação
 - Oculta que um recurso pode ser movido para outra localização enquanto em uso.
- Replicação
 - Oculta que um recurso é replicado.
- Concorrência

- Oculta que um recurso pode ser compartilhado por diversos usuários concorrentes.
- Falha
 - Oculta a falha e a recuperação de um recurso.

1.3 Ciladas

Sistemas Distribuídos são diferentes do software tradicional porque os componentes estão dispersos por uma rede. Não levar essa dispersão em conta durante o projeto é o que torna tantos sistemas desnecessariamente complexos e resulta em erros que precisam ser consertados mais tarde. Peter Deutsch, que, na época, trabalhava na Sun Microsystems, formulou esses erros como as seguintes premissas falsas que todos adotam ao desenvolver uma aplicação distribuída pela primeira vez:

1. A rede é confiável.
2. A rede é segura.
3. A rede é homogênea.
4. A topologia não muda.
5. A latência é zero.
6. A largura de banda é infinita.
7. O custo de transporte é zero.
8. Há só um administrador.

Observe como essas premissas se referem a propriedades exclusivas de sistemas distribuídos: confiabilidade, segurança, heterogeneidade e topologia da rede; latência e largura de banda; custos de transporte e, por fim, domínios administrativos. No desenvolvimento de aplicações não distribuídas, é provável que a maioria dessas questões nem apareça.

1.4 Sistemas de informação distribuídos

Encontrado em organizações que se defrontaram com uma profusão de aplicações em rede para as quais a interoperabilidade se mostrou uma experiência dolorosa. Muitas das soluções de middleware existentes são resultado do trabalho com uma infra-estrutura na qual era mais fácil integrar aplicações a um sistema de informações de âmbito empresarial (Bernstein, 1996; Alonso et al., 2004).

À medida que as aplicações se tornavam mais sofisticadas e eram gradualmente separadas em componentes independentes, notavelmente distinguindo componentes de banco de dados de componentes de processamento, ficou claro que a integração também deveria ocorrer de modo que permitisse às aplicações se comunicar diretamente umas com as

outras. Isso resultou, atualmente, em uma enorme indústria dedicada à integração de aplicações empresariais (Enterprise Application Integration – EAI).

1.5 Sistemas distribuídos pervasivos

Caracterizados por sua estabilidade: os nós são fixos e têm uma conexão mais ou menos permanente e de alta qualidade com uma rede.

Um exemplo: a profusão de técnicas para mascarar falhas e recuperação dará a impressão de que as coisas podem dar errado apenas raramente. Da mesma maneira, conseguimos ocultar aspectos relacionados com a real localização de um nó na rede, o que permite, efetivamente, que usuários e aplicações acreditem que os nós continuam onde estão.

Contudo, a questão ficou muito diferente com a introdução de dispositivos de computação móveis e embutidos. Atualmente encontramos sistemas distribuídos nos quais a instabilidade é o comportamento esperado. Nesses sistemas, que denominamos sistemas distribuídos pervasivos, os equipamentos costumam ser caracterizados por seu pequeno tamanho, pela alimentação por bateria, por sua mobilidade e por terem somente uma conexão sem fio, se bem que nem todas essas características se aplicam a todos os dispositivos. Além do mais, tais características não precisam ser necessariamente interpretadas como restritivas, como é ilustrado pelos smart phones. (Roussos et al., 2005).

Como seu nome sugere, um sistema distribuído pervasivo é parte de nosso entorno; por isso, é, em geral, inerentemente distribuído. Um aspecto importante é a ausência geral de controle administrativo humano. Na melhor das hipóteses, os dispositivos podem ser configurados por seus proprietários; porém, quanto ao mais, eles precisam descobrir automaticamente seu ambiente e se encaixar o melhor que puderem. Grimm et al. (2004) tornaram esse encaixar mais exato pela formulação dos três requisitos para aplicações pervasivas apresentados a seguir:

1. Adotar mudanças contextuais;
2. Incentivar composição ad hoc;
3. Reconhecer compartilhamento como padrão;

1.6 Sistemas domésticos

Um tipo cada vez mais popular de sistema pervasivo, mas que talvez seja o menos restrito, são sistemas montados ao redor de redes domésticas. Em geral, esses sistemas são compostos de um ou mais computadores pessoais. Porém, o mais importante é que integram eletrônicos de consumo típicos como aparelhos de TV, equipamentos de áudio e vídeo, dispositivos para jogos, smart phones, PDAs e outros equipamentos de uso pessoal em um único sistema. Além disso, podemos esperar que todos os tipos de dispositivos, como eletrodomésticos de cozinha, câmaras de vigilância, relógios, controladores de iluminação e assim por diante, serão conectados a um único sistema distribuído.

Da perspectiva de sistema, há vários desafios que precisam ser enfrentados antes que os sistemas pervasivos domésticos se tornem realidade. Um desafio importante é que tal sistema deve ser completamente autoconfigurável e autogerenciável. Não se pode esperar que usuários finais estejam dispostos ou sejam capazes de manter um sistema distribuído doméstico ligado e em funcionamento se seus componentes forem propensos a erros, como acontece com muitos dos dispositivos existentes hoje.

Muito já foi conseguido por meio dos padrões Universal Plug and Play (UPnP), pelos quais dispositivos obtêm automaticamente endereços IP, podem descobrir uns aos outros e assim por diante.

1.7 Sistemas eletrônicos para tratamento de saúde

Uma outra classe de sistemas pervasivos importante e que está começando a fazer sucesso é a relacionada ao tratamento eletrônico (pessoal) de saúde. Com o aumento do custo do tratamento médico, estão sendo desenvolvidos novos dispositivos para monitorar o bem-estar de indivíduos e entrar automaticamente em contato com médicos quando necessário. Em muitos desses sistemas, uma meta importante é evitar que as pessoas sejam hospitalizadas.

Sistemas para tratamentos de saúde costumam ser equipados com vários sensores organizados em uma rede de área corporal, de preferência sem fio. Uma questão importante é que, na pior das hipóteses, tal rede deve incomodar uma pessoa o mínimo possível. Com essa finalidade em vista, a rede deve ser capaz de funcionar quando a pessoa estiver em movimento, sem que esta precise estar presa por fios elétricos a dispositivos imóveis.

Da perspectiva do sistema distribuído, deparamos imediatamente com questões como:

1. Onde e como os dados monitorados deverão ser armazenados?
2. Como podemos evitar a perda de dados cruciais?
3. Qual é a infra-estrutura necessária para gerar e transmitir sinais de alerta?
4. Como os médicos podem dar retorno on-line?
5. Como pode ser alcançada a extrema robustez do sistema de monitoração?
6. Quais são as questões de segurança e como as políticas adequadas podem ser impostas?

Diferentemente dos sistemas domésticos, não podemos esperar que a arquitetura de sistemas pervasivos de tratamento de saúde tenda a passar para sistemas de um único servidor e que seus dispositivos de monitoração operem com funcionalidade mínima. Ao contrário: por razões de eficiência, os dispositivos e redes de áreas corporais terão de suportar processamento de dados na rede, o que significa que os dados de monitoração terão de ser agregados antes de ser armazenados permanentemente ou enviados a um médico. Diferentemente do caso de sistemas de informação distribuídos, ainda não há uma resposta clara para essas questões.

2. Arquiteturas

2.1 Introdução

Sistemas distribuídos podem ser organizados de modos diferentes. Podemos fazer uma distinção entre arquitetura de software e arquitetura de sistema. A última considera onde os componentes que constituem um sistema distribuído estão colocados nas várias máquinas. A primeira se preocupa mais com a organização lógica do software: como os componentes interagem, de que modos eles podem ser estruturados, como podem ficar independentes e assim por diante.

Uma idéia fundamental quando falamos sobre arquiteturas é o estilo arquitetônico. Um estilo reflete o princípio básico que é sugerido na organização da interação entre os componentes de software que compreendem um sistema distribuído. Entre os estilos importantes estão disposição em camadas, orientação a objetos, orientação a eventos e orientação a espaço de dados.

Há variadas organizações de sistemas distribuídos. Uma classe importante é aquela em que as máquinas são divididas em clientes e servidores. Um cliente envia uma requisição a um servidor, que então produzirá um resultado que é retornado ao cliente. A arquitetura cliente-servidor reflete o modo tradicional de modularização de software pelo qual um módulo chama as funções disponíveis em um outro módulo. Colocando componentes diferentes, obtemos uma distribuição física natural de funções por um conjunto de máquinas.

Arquiteturas cliente-servidor costumam apresentar alto grau de centralização. Em arquiteturas descentralizadas, freqüentemente vemos um papel igual desempenhado pelos processos que constituem um sistema distribuído, também conhecidos como sistemas peer-to-peer. Em sistemas peer-to-peer, os processos são organizados em uma rede de sobreposição, que é uma rede lógica na qual todo processo tem uma lista local de outros pares com os quais ele pode se comunicar. A rede de sobreposição pode ser estruturada, caso em que são disponibilizados esquemas determinísticos para rotear mensagens entre processos. Em redes não estruturadas, a lista de pares é mais ou menos aleatória, o que implica que é preciso disponibilizar algoritmos de busca para localizar dados os outros processos.

2.2 Estilos

Os mais importantes para sistemas distribuídos:

1. Arquiteturas em camadas
2. Arquiteturas baseadas em objetos
3. Arquiteturas centradas em dados
4. Arquiteturas baseadas em eventos

As arquiteturas multidividas sugerem várias possibilidades para a distribuição física de uma aplicação cliente-servidor por várias máquinas. A organização mais simples é ter só dois tipos de máquinas:

1. Uma máquina cliente que contém apenas os programas que implementam o nível de interface de usuário.
2. Uma máquina servidor que contém o resto, ou seja, os programas que implementam o nível de processamento de dados.

Nesta organização, tudo é manipulado pelo servidor, ao passo que, em essência, o cliente nada mais é do que um terminal burro, possivelmente com uma interface gráfica bonitinha. Uma abordagem para organizar clientes e servidores é distribuir os programas presentes nas camadas de aplicação.

2.3 Sistemas distribuídos colaborativos

Estruturas híbridas são disponibilizadas notavelmente em sistemas distribuídos colaborativos. A questão principal em muitos desses sistemas é conseguir dar a partida, para o que muitas vezes é disponibilizado um esquema cliente-servidor tradicional. Tão logo um nó se junte ao sistema, ele pode usar um esquema totalmente descentralizado para colaboração.

Para ficarmos no terreno concreto, em primeiro lugar vamos considerar o sistema de compartilhamento de arquivos BitTorrent. O BitTorrent é um sistema peer-to-peer de transferência (downloads) de arquivos. A idéia básica é que, quando um usuário final estiver procurando um arquivo, ele transfira porções do arquivo de outros usuários até que as porções transferidas possam ser montadas em conjunto, resultando no arquivo completo.