



UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS – CCEA

PROJETO DE SISTEMAS DISTRIBUÍDOS

Ingrid Morgane Medeiros de Lucena

INTRODUÇÃO

- O projeto de SD tem como objetivo as metas de projeto chaves:
 - Alta Performance
 - Confiabilidade
 - Escalabilidade
 - Consistência
 - Segurança
- Abordar as questões técnicas básicas de projeto que afetam o alcance destas metas.
- Focalizar arquiteturas de sistema e componentes usados para satisfazer os requisitos genéricos.

INTRODUÇÃO

- O **projeto** e a **implementação** de SD deve ser aberto.
- O desenvolvimento de **aplicações distribuídas** frequentemente resulta na necessidade de adicionar **novos serviços**.
- Esses **serviços** devem ser projetados satisfatoriamente em conjunção com os serviços existentes e satisfazer **padrões** similares com respeito a **metas de projeto chaves**.

INTRODUÇÃO

- Neste sentido o desenvolvimento de **cada nova** aplicação distribuída envolve **decisões de projeto** que objetivam as mesmas metas como no projeto inicial do sistema.
- Assim o conhecimento do projeto é indispensável, não somente no projeto inicial de SDs, mas também para o **desenvolvimento de aplicações distribuídas**.

INTRODUÇÃO

- Essas **metas** podem ser expressas na forma de **declarações formais** sobre o **comportamento de um sistema**, tal como:
- *“o serviço continuará a estar disponível após a falha de qualquer processo servidor”,* que pode ser verificada, ou por análise do projeto ou por teste.
- O alcance de tais metas é uma medida do sucesso de um projeto.

INTRODUÇÃO

- Principais problemas técnicos e imprevistos, para o alcance das metas de projeto chaves.
- *“Um bom projeto é mais do que resolver problemas técnicos e evitar imprevistos; ele deve ser orientado para as metas em função dos desenvolvedores de software e dos usuários finais.”*

INTRODUÇÃO

- As necessidades específicas de ambas destas classes (desenvolvedores de software e usuários finais) são notoriamente difíceis de identificar.
- **Requisitos genéricos do usuário** para sistemas distribuídos, é um grande desafio.

QUESTÕES BÁSICAS DE PROJETO



Universidade
ESTADUAL DA PARÁ

- Questões que não são relacionadas à natureza distribuída de um sistema – tais como técnicas de engenharia de software, interação usuário-computador e projeto de algoritmo.
- Discussão de interesse: questões que surgem especificamente da natureza distribuída do sistema.

QUESTÕES BÁSICAS DE PROJETO

De natureza distribuída:

- Nomeação
- Comunicação
- Estrutura do Software
- Alocação de Carga
- Manutenção de Consistência

QUESTÕES BÁSICAS DE PROJETO

- **Nomeação (Naming)**
- SDs são baseados: **compartilhamento de recursos e transparência de distribuição.**
- Os nomes atribuídos a recursos ou objetos devem ter **significados globais** que são **independentes da localização** do objeto, e eles devem ser suportados por um sistema de interpretação de nomes. (Similar ao trabalho do DNS).

NOMEAÇÃO

- O sistema de interpretação de nomes deve traduzir nomes, no sentido de habilitar programas acessarem recursos nomeados de forma otimizada.
- Questão de projeto: projetar esquemas de nomeação com **escalabilidade** para um grau apropriado, no qual nomes são traduzidos eficientemente para satisfazer **metas** apropriadas de **performance**.

NOMEAÇÃO

- A provisão de esquemas de propósito geral para nomeação de recursos é uma consequência necessária do compartilhamento de recursos.
- Esquemas de nomeação devem ser projetados para suportar escalabilidade e transparência de localização.

NOMEAÇÃO

- Serviços de nomes mantém a informação necessária para resolver nomes e prover um serviço de resolução de nomes para clientes.
- Seu projeto tem impacto sobre a performance, escalabilidade e na segurança de sistemas distribuídos.

QUESTÕES BÁSICAS DE PROJETO

- **Comunicação**
 - A performance e a confiabilidade das técnicas de comunicação usadas para implementação de SDs são críticas para a performance do sistema.
 - Altas velocidades de transmissão podem ser alcançadas nas redes (LAN ou WAN).
- Mesmo com redes de alta velocidade, comunicação pode **consumir tempo**, por causa do **número e complexidade** das **camadas de software** envolvidas.
- Questão de projeto: Otimizar a implementação de comunicação em SDs, considerando um modelo de programação de alto nível para seu uso.

COMUNICAÇÃO

- O projeto de subsistemas de comunicação, particularmente os modelos de comunicação para os quais eles são otimizados têm impacto sobre a **performance** e na **abertura do sistema**.
- Os **modelos Cliente/Servidor e Multicast** são os mais efetivos para a construção de SDs abertos e suas implementações eficientes são requisitos chave de projeto.

QUESTÕES BÁSICAS DE PROJETO



Universidade
ESTADUAL DA PARÁ

- **Estrutura do Software**
- A característica de ser **aberto** é alcançada através do projeto e construção de componentes de software com **interfaces bem-definidas**.
- Abstração de Dados é uma importante técnica de projeto para SDs. Serviços podem ser vistos como os gerenciadores de objetos de um determinado tipo de dados

- Abstração de Dados é uma importante técnica de projeto para SDs.
- Serviços podem ser vistos como gerenciadores de objetos de um determinado tipo de dados.
- A interface para um serviço pode ser vista como um conjunto de operações.
- Questão de projeto:
 - Estruturar um sistema de modo que novos serviços possam ser introduzidos, para trabalharem em conjunção com o serviços já existentes, mas, sem duplicação de elementos de serviços que já existem.
 - Métodos para se estruturar software são requeridos para se alcançar os benefícios de abertura do sistema.

ESTRUTURA DE SOFTWARE

- As escolhas desses métodos se situam no **domínio de projeto de sistemas operacionais**
- Os métodos pelos quais os recursos básicos requeridos por todas as aplicações serão providos e o grau para o qual eles podem ser configurados e otimizados para satisfazer os requisitos do usuário

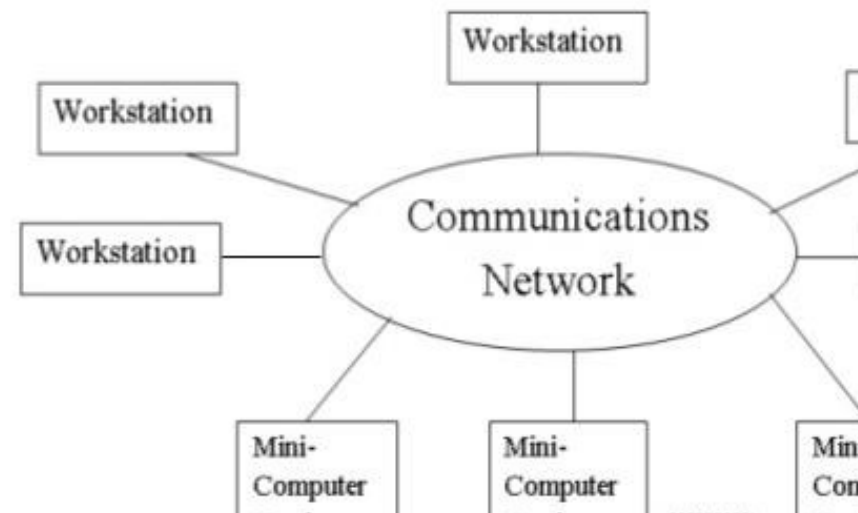
ALOCAÇÃO DA CARGA DE TRABALHO

Sistema de Computação Centralizado:

- Todos os recursos de processador e memória são disponíveis para alocação pelo SO, em uma maneira requerida pela carga de trabalho imposta pelo usuário corrente.

Sistema Distribuído Simples

- A performance do processador e capacidade de memória de uma workstation determina o tamanho da maior tarefa que pode ser realizada por um usuário.



ALOCAÇÃO DA CARGA DE TRABALHO

- O simples modelo **Workstation-Server** não otimiza o uso de recursos de processamento e memória,
- Também não habilita ao seu usuário, que tenha tarefa de computação exigindo grande poder de processamento e requerimento de memória, a obter recursos adicionais.

Alocação de Carga de Trabalho:

- Boa **performance** é um requisito para a maior parte dos produtos oriundos da engenharia, e é de interesse primordial para a maioria dos programadores e projetistas de sistemas.

ALOCAÇÃO DA CARGA DE TRABALHO

- Alocação da Carga de Trabalho tem impacto na efetividade com qual os recursos de hardware de um SD são usados, e portanto, na **performance** total do sistema.
- Computadores **multiprocessadores de memória compartilhada** são frequentemente usados para tarefas intensivas de processador em SDs.
- Questão de Projeto para SDs:
 - Como dispor o **processamento**, a **comunicação** e os **recursos**, distribuídos em uma rede, visando obter um efeito ótimo no processamento de uma carga de trabalho que estará sempre mudando.

QUESTÕES BÁSICAS DE PROJETO

- Manutenção de Consistência:
 - Questões de consistência frequentemente surgem da separação dos recursos de processamento e da concorrência em SDs.
- Sua significância para o projeto está no seu impacto sobre a **performance** de SDs e aplicações distribuídas.
- Questão de Projeto: A manutenção de consistência em um custo razoável é talvez o mais difícil problema encontrado no projeto de um SD.
- A necessidade para consistência:
 - Quando dados compartilhados ou recursos são atualizados por diversos processos independentes e existe a necessidade para manter consistência entre cópias de dados replicados.

REQUISITOS DO USUÁRIO

- Consideração pelos projetistas de SDs das necessidades dos usuários potenciais.
- A maior parte dos SDs são projetados para prover facilidades de computação de propósito geral
- Os mais importantes requisitos que surgem em sistemas práticos:
 - Funcionalidade
 - “O que o sistema deve fazer para os usuários.”
 - Qualidade de Serviço
 - “Envolvendo questões sobre performance, confiabilidade e segurança.”
 - Reconfiguração
 - “A necessidade para um sistema acomodar mudanças sem causar desrupção da provisão de serviço existente.”

REQUISITOS DO USUÁRIO

- Habilitam usuários e programas de aplicação a explorarem as características especiais de sistemas distribuído
- Não tem necessidade de comprometer as características-chaves e as metas de projeto já identificadas.

FUNCIONALIDADE

- Quais serviços podem usuários e desenvolvedores de aplicação esperar de um sistema distribuído ?
- Quais aplicações os usuários podem rodar ?

- Benefícios de Sistemas Distribuídos são:
 - A economia e a conveniência proporcionada pelo compartilhamento de recursos de HWe de informação;
 - O melhoramento potencial em performance e confiabilidade oriunda da separação dos componentes de software, de modo que possam ser executados em computadores separados.
- O **requisito mínimo** para um sistema distribuído é a de que **a funcionalidade** provida por um tal sistema, **não deve ser menor** do que o usuário poderia esperar de qualquer um único computador isolado.

- O requisito mais forte é que um sistema distribuído deve proporcionar um melhoramento com relação aos serviços providos por qualquer computador isolado, exceto, compartilhamento de recursos, através de uma ou ambas melhorias:
- Compartilhamento através de uma rede pode trazer uma variedade de recursos, do que poderia ser proporcionado por um único computador.
- Utilização das vantagens de distribuição pode ser possível a nível de API, de modo que aplicações que executam compartilhamento explícito, tolerância à falha ou aplicações paralelas podem ser programadas.

- **A maioria dos usuários estão interessados, principalmente, na questão de quais aplicações eles podem rodar.**
- Por um lado, novas funcionalidades - aquelas que não existem em um sistema centralizado - podem ser providas em um sistema distribuído, considerando a possibilidade de união de funcionalidades já existentes.
- Por outro lado, já existe muito software, que roda em sistemas centralizados.
- **Muito esforço e investimento financeiro foi feito para a produção e manutenção desse software.**
- Assim, de um indivíduo ou organização que tenha investido nesse software, surge a questão:
 - Esse software ainda trabalhará na transição para um novo sistema distribuído ?
 - Algum software tem de ser substituído em uma transição para um sistema que explora as vantagens da distribuição ?

Existem **três opções abertas**, quando considerando a **migração** de computação centralizada multi-usuário, ou computação baseada em PC de um único usuário, para a **computação distribuída**:

1. Adaptar sistemas operacionais existentes:

- Continua-se a usar sistemas operacionais existentes, que tenham sido adaptados para rede, tal como o UNIX;
- Adiciona-se serviços, como aplicações no topo do UNIX, sempre que possível, mas modifica-se o kernel, em alguns casos importantes, para prover transparentemente, uma implementação distribuída de uma facilidade existente.

2. Migrar para um inteiramente novo sistema operacional projetado para sistemas distribuídos:

- Para se obter as vantagens totais da distribuição, os projetistas não devem estar restritos pelas características de sistemas antigos.
- Todos os serviços de sistemas operacionais podem ser abertos e distribuídos nesta opção. O software existente torna-se em desuso.

3. Emulação:

- Migra-se para um novo sistema operacional projetado para sistemas distribuídos, mas um que possa emular um Ou mais sistemas operacionais existentes.
- Assim, o software existente continuará a rodar, mas um novo software pode ser escrito usando a interface de chamadas de sistema do novo sistema operacional, para se obter as vantagens da distribuição.
- Ambos tipos de software podem rodar lado-a-lado.

FUNCIONALIDADE

- A opção 1 é um pouco limitada nas possibilidades para novas funcionalidades que ela oferece.
- A opção 2 representa um ideal do ponto de vista do projetista, mas pode deixar muitos usuários insatisfeitos.
- A opção 3 parece preferível em relação a qualquer das outras duas.

RECONFIGURAÇÃO

- A **escalabilidade** de um projeto de sistema distribuído e sua habilidade para acomodar **heterogeneidade**, são relevantes quando o projeto é para ser implementado em instalações separadas de diferentes tamanhos e máquinas, e quando uma instalação particular é expandida.
- No último caso, a mudança deve ocorrer sem parar o sistema.

Existem duas **escalas de tempo** em sistemas distribuídos, sobre as quais, mudanças são **causas potenciais para parar o sistema**, e que devem ser consideradas no projeto do software do sistema:

Mudanças a Curto-Prazo:

- O software de sistema distribuído deve ser projetado para tratar as seguintes mudanças em condições de tempo de execução:
 - Um processo com falha, computador ou componente da rede é substituído por um outro, trabalhando replicando o outro.
 - Carga computacional é deslocada de máquinas sobrecarregadas para máquinas menos sobrecarregadas, de forma a aumentar o **throughput** do sistema distribuído.

RECONFIGURAÇÃO

Evolução a Médio ou Longo-Prazo:

- Em adição a mudanças devido a expansão em escala e a necessidade para acomodar componentes heterogêneos que ocorrem como parte de um desenvolvimento normal de sistema distribuído, **máquinas existentes são, algumas vezes, atribuídas com novos papéis, ou aquisições de novas máquinas são feitas.**
- A habilidade para acomodar mudanças sobre qualquer destas escalas de tempo é chamada de reconfiguração do sistema (reconfigurability), e as próprias mudanças individuais são chamadas **reconfigurações.**
- O requisito para efetuar reconfigurações sem parar o sistema importa em torná-las **transparentes.**

QUALIDADE DE SERVIÇO

- Uma vez que usuários são providos com a funcionalidade que eles requerem de um serviço em um SD, podemos ir indagar sobre a provisão da qualidade de serviço, e se existe qualquer valor adicionado surgido da distribuição.
- Discutimos qualidade de serviço sobre os seguintes aspectos:
 - **Performance** (tempo de resposta experimentado pelos seus usuários).
 - **Confiabilidade** (medida de quão pequena é a probabilidade do sistema desviar do comportamento como ele foi projetado, em relação à especificação de seu comportamento correto).
 - **Disponibilidade** (medida da proporção de tempo que o sistema está disponível para uso).
 - **Segurança** (o problema tem dois aspectos), **privacidade** e **integridade** dos dados que viajam na rede;
 - abertura à **interferência** com software de sistema.

Referências

Notas de aula Computação Distribuída - UNIRONDON - Cuiabá - MT - João Bosco Manguiera Sobral/Daniela Barreiro Claro

COULOURIS, George. Sistemas distribuídos: conceitos e projeto. 5. ed. Porto Alegre: Bookman, 2013. 1023 p.