


INTRODUÇÃO

Considerações de Projeto de um SOD

Sistemas Distribuídos

Introdução (considerações de projeto)

- Principal fonte de complexidade: inexistência de informação atualizada consistente sobre o sistema
 - recursos separados fisicamente
 - não existe relógio comum
 - mensagens podem ser entregues com atraso ou perdas
-  maior dificuldade/complexidade para realizar tarefas
 - ex.: como escalonar processadores se não se sabe exatamente quais estão ativos e qual sua carga em determinado momento
- Objetivo
 - usuários devem ver um sistema distribuído como um sistema centralizado virtual que é flexível, eficiente, confiável, seguro e fácil de usar (Pradeep)

Sistemas Distribuídos

Introdução (considerações de projeto)

- Transparência (Sinha)
 - de acesso
 - usuário não distingue entre acesso a recurso local ou remoto
 - interface do usuário - conjunto de chamadas de sistema tem que ser projetadas para fazer sentido em sistemas centralizados ou distribuídos - acessar recursos locais ou distantes
 - facilidade de esquema de nomeação global
 - de localização
 - movimentação de recursos e usuários a vontade
 - recursos mantêm mesmo nome, independentemente de localização
 - usuários acessam mesmos recursos a partir de qualquer nodo

Sistemas Distribuídos

Introdução (considerações de projeto)

- Transparência (Sinha)
 - de replicação
 - replicação: aumento de desempenho e confiabilidade
 - gerência de réplicas: nomeação de réplicas, mapeamento de nome dado pelo usuário para réplica apropriada do recurso, etc.
 - de falhas
 - mascara dos usuários as falhas parciais do sistema
 - continuidade do funcionamento, talvez de maneira degradada, em presença de falhas
 - de migração
 - razões: manter desempenho, confiabilidade e segurança
 - escolha de objeto (processo) a migrar
 - comunicação deve continuar de maneira transparente

Sistemas Distribuídos

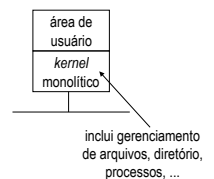
Introdução (considerações de projeto)

- Transparência (Sinha)
 - de concorrência
 - competição por recursos - necessidade de:
 - ordenação de eventos
 - exclusão mútua
 - no-starvation
 - no dead-lock
 - de desempenho
 - reconfiguração da carga do sistema para aumentar desempenho
 - facilidades de alocação de recursos e definição do momento da migração de processos para homogeneizar a carga dos nodos
 - de crescimento/expansão
 - permitir sistema ser estendido sem interromper atividades dos usuários

Sistemas Distribuídos

Introdução (considerações)

- Flexibilidade
 - **Kernel monolítico**
 - gerência de processos, memória, dispositivos, arquivos, nomes, e comunicação entre processos provida pelo kernel
 - kernel grande
 - modificação e adição de serviços dificultadas



Sistemas Distribuídos

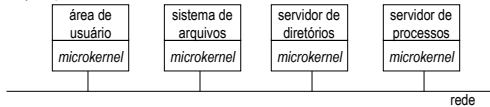
Introdução (considerações)

➤ Flexibilidade

▪ **Microkernel**

- kernel pequeno, facilidades mínimas para implementar demais serviços do sistema
- Funções Básicas em cada nodo:
 - Comunicação entre processos
 - Gerência de memória (básica)
 - Gerência de processos (básica)
 - Operações de E/S de baixo nível

• Funções Comuns: Servidores
gerência de arquivos, nomes,
gerência adicional de processos,
etc.
• Modo usuário.



Sistemas Distribuídos

Faculdade de Informática - PUCRS

Introdução (considerações)

➤ Confiabilidade (*reliability*)

- e.g. 4 servidores 95%
 - probabilidade de os 4 estarem fora do ar = 0,000006
- Outras considerações: perda de informações, segurança, tolerância a falhas, ...

➤ Desempenho

- Aplicação distribuída não deve ser mais lenta que aplicação centralizada
- comunicação:
 - agrupar dados e outras informações para mandar através da rede
 - minimizar tráfego (algoritmos distribuídos - quanto se comunicam ?)
- cache quando possível
- minimizar cópias de dados
- tirar vantagem de multiprocessamento

Sistemas Distribuídos

Faculdade de Informática - PUCRS

Introdução (considerações)

➤ "Escalabilidade" (*scalability*)

- funciona para 200 - funciona para 200.000 ??
- Princípios a serem evitados:
 - componentes centralizados
 - tabelas centralizadas
 - algoritmos centralizados
- Algoritmos distribuídos
 - nenhum computador deve conter todas informações sobre o estado do sistema
 - falha em um computador não deve prejudicar outro
 - não assumir que exista relógio global

Sistemas Distribuídos

Faculdade de Informática - PUCRS