

02 – Sistemas Distribuídos

- Cenário anterior: computadores independentes;
- Surgimento de Microprocessadores mais poderosos;
- LANs e WANs;
- Internet.

02 – Sistemas Distribuídos

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

02 – Sistemas Distribuídos

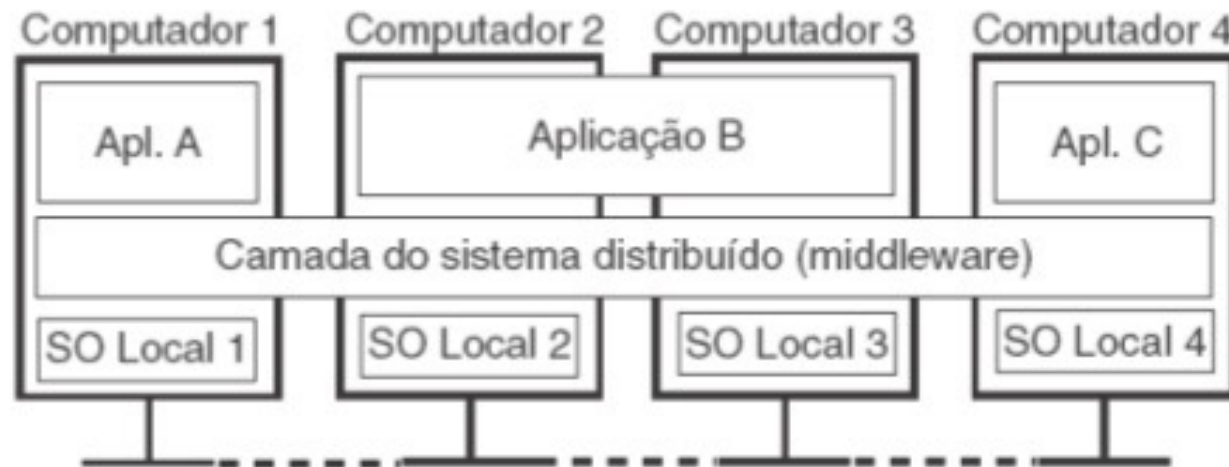
Características Importantes:

- Independência do tipo de dispositivo e plataforma utilizada;
- A forma de conexão deve ser oculta aos usuários;
- Escalabilidade;
- Transparência na manutenção/reconfiguração do sistema.

02 – Sistemas Distribuídos

Middleware:

- Camada de software localizada entre a aplicação final e o sistema operacional;



02 – Sistemas Distribuídos

Metas:

- Acesso aos recursos
 - Proporciona Economia;
 - A segurança ainda é um problema.
- Transparência;
- Abertura;
- Escalabilidade.

02 – Sistemas Distribuídos

Transparência:

- 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.

02 – Sistemas Distribuídos

Transparência:

Transparência	Descrição
Acesso	Ocultar diferenças na representação de dados e no modo de acesso a um recurso
Localização	Ocultar o lugar em que um recurso está localizado
Migração	Ocultar que um recurso pode ser movido para outra localização
Relocação	Ocultar que um recurso pode ser movido para uma outra localização enquanto em uso
Replicação	Ocultar que um recurso é replicado
Concorrência	Ocultar que um recurso pode ser compartilhado por diversos usuários concorrentes
Falha	Ocultar a falha e a recuperação de um recurso

02 – Sistemas Distribuídos

Transparência:

- A total transparência de distribuição é quase impossível;
- O usuário deve estar ciente das limitações de uma rede e da localização de um recurso.

02 – Sistemas Distribuídos

Abertura:

- Uma outra meta importante de sistemas distribuídos é a abertura. Um sistema distribuído aberto é um sistema que oferece serviços de acordo com regras padronizadas que descrevem a sintaxe e a semântica desses serviços.

Exemplo:

Em casos de redes de computadores há regras padronizadas que governam o formato, o conteúdo e o significado de mensagens enviadas e recebidas. Tais regras são formalizadas em protocolos. Em Sistemas Distribuídos , em geral os serviços são especificados por meio de interfaces.

02 – Sistemas Distribuídos

Abertura:

- Descrição da sintaxe e da semântica dos serviços;
- Utilização de interfaces;
- Interoperabilidade;
- Portabilidade;
- Extensível.

02 – Sistemas Distribuídos

Escalabilidade:

- Pode ser medida em 3 dimensões: tamanho, geográfica, administrativa
- Problemas referentes a falta de escalabilidade em tamanho:

Conceito	Exemplo
Serviços centralizados	Um único servidor para todos os usuários
Dados centralizados	Uma única lista telefônica on-line
Algoritmos centralizados	Fazer roteamento com base em informações completas

02 – Sistemas Distribuídos

Escalabilidade:

- Algoritmos Descentralizados
 - Nenhuma máquina tem informações completas sobre o estado da rede;
 - As máquinas decidem com base em informações locais;
 - A falha de uma máquina não arruína o algoritmo;
 - Não utiliza relógio global.

02 – Sistemas Distribuídos

Escalabilidade:

- Problemas referentes à escalabilidade geográfica:
 - Comunicação Síncrona;
 - Comunicação ponto a ponto.
- A centralização de recursos também impede a escalabilidade geográfica;
- Técnicas: Ocultar latências de:
 - Comunicação (assíncronismo, threads, processamento local);
 - Distribuição (divisão da tarefa em vários servidores);
 - Replicação (cache, consistência fraca ou forte).

02 – Tipos de Sistemas Distribuídos

- Sistemas de Computação Distribuídos;
- Sistemas de Informação Distribuídos;
- Sistemas Distribuídos Pervasivos.

02 – Sistemas de Computação Distribuídos

- Utilizada para tarefas de computação de alto desempenho;
- Pode ser dividida em dois subgrupos:
 - Computação de Cluster;
 - Computação em Grade;

02 – Sistemas de Computação Distribuídos

Sistemas de Computação de Cluster:

- Populares pela razão preço/desempenho;
- Rede Local;
- Mesmo S.O.;
- Apenas o nó mestre executa o middleware;
- Programação paralela, um único programa, intensivo, é executado em paralelo em várias máquinas.
- Cada Cluster consiste em um conjunto de nós de computação controlados e acessados por meio de um único nó mestre;
- Exemplo: Clusters de Beowulf (baseado em Linux);

02 – Sistemas de Computação Distribuídos

Sistemas de Computação em Grade:

- Cluster em sua maioria os pc's são parecidos (mesmo hardware, mesmo S.O., mesma rede);
- Sistemas em Grade possuem alta heterogeneidade;
- Recursos de diferentes organizações são reunidos para permitir colaboração de um grupo de pessoas ou instituições;
- Organização Virtual;
- Prover acesso a recursos de diferentes domínios administrativos;

02 – Sistemas de Computação Distribuídos

Sistemas de Computação em Grade:

- Arquitetura em camadas:
 - Camada-base;
 - Camada de Conectividade;
 - Camada de Recursos;
 - Camada Coletiva;
 - Camada de Aplicação.

02 – Sistemas de Informação Distribuídos

- Transações
 - Atômicas: tudo ou nada;
 - Consistentes;
 - Isoladas;
 - Duráveis;
 - Aninhadas;
 - Uso de primitivas;

02 – Sistemas de Informação Distribuídos

- RPC e RMI;
- MOM (Message-Oriented Middleware);
 - Publish/Subscribe;

02 – Sistemas Pervasivos

- A instabilidade dos sistemas é esperada;
- Mobilidade;
- Não existe transparência;
- Ausência de controle administrativo humano;
- Aplicações: sistemas domésticos, tratamento de saúde, sensores.

02 – Sistemas Pervasivos

- “Cria ambientes com computação e comunicação, de maneira integrada aos seres humanos, onde a percepção de se estar lidando com computadores seria mínima.”;
- Um sistema pervasivo deve tratar cada dispositivo que seja dotado de alguma “inteligência”;
- Sistemas ubíquos/pervasivos são caracterizados pela capacidade de estarem em vários lugares simultaneamente (a ubiquidade) e por estarem disseminados (pervasivos) no ambiente de maneira não obstrusiva, quase invisível para o usuário comum.

02 – Estilos Arquiteturais

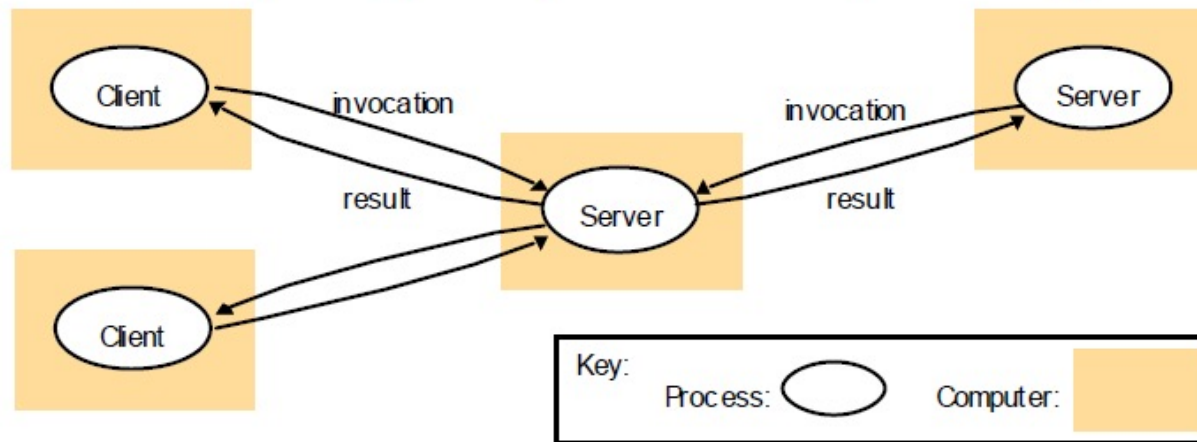
- Em Camadas;
 - O componente em uma camada X pode chamar componentes na camada subjacente X-1;
- Baseado em Objetos;
 - Chamada de procedimentos remotos;
- Centradas em Dados;
 - Comunicação por meio de um repositório comum. Ex: Web;
- Baseado em Eventos;
 - Publicar/Subscrever.

02 – Características Básicas de um Sistema Distribuído



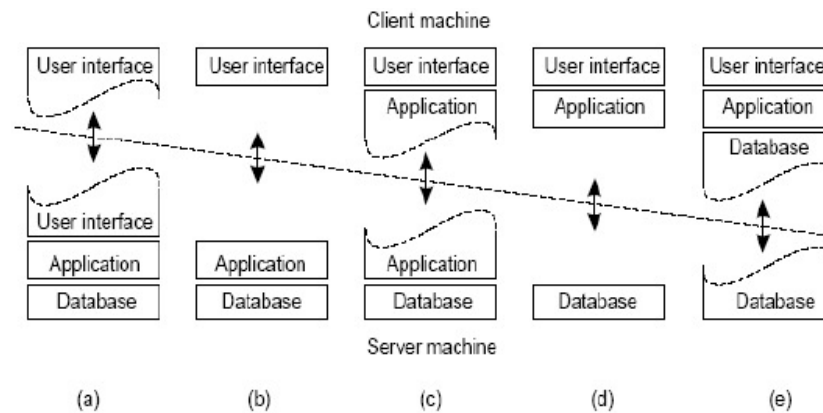
02 – Arquiteturas Centralizadas

- Cliente x Servidor;
- Protocolos com conexão;
- Protocolos sem conexão (Problemas com operações que não-Idempotentes).



02 – Cliente Servidor

- Estilo Arquitetônico em Camadas:
 - Nível de Interface de Usuário;
 - Nível de Processamento;
 - Nível de Dados.



02 – Arquiteturas Descentralizadas

- Peer-To-Peer (P2P);
- Todas as Estações são Clientes e Servidoras;
- Redes de Sobreposição;
- Arquiteturas P2P Estruturadas;
 - Tabelas Hash Distribuídas Determinísticas (DHT);
- Arquiteturas P2P Não Estruturadas;
 - Lista de Alguns Vizinhos;
 - Busca por Inundação na Rede.

02 – Arquiteturas Descentralizadas

- Topologia por localização, por item de dado;
- Superpares:
 - Manutenção do índice (intermediador);
 - Os nós são seus Clientes;
- Arquitetura híbrida
 - Servidor de borda;
 - Sistemas colaborativos;
- Inicialização cliente-servidor;
- Rastreador Servidor;
- BitTorrent.

02 – Middleware

- Middleware
 - Fornece transparência;
 - Diversos estilos arquiteturais (objetos, eventos);
 - Suporte a requisitos da aplicação;
 - Generalização/Complexidade X Especificação/Simplicidade;
- Transparência oferecida por middlewares clientes:
 - Migração;
 - Replicação;
 - Tolerância a falhas.

02 – Middleware Interceptador

- Uma chamada remota a objeto é realizada em três etapas:
 - Um objeto A utiliza localmente uma interface implementada por um objeto B remoto;
 - Essa requisição é enviada a um objeto genérico local (interceptador);
 - Esse objeto envia via rede a mensagem para B;
- O interceptador pode atuar em nível de requisição (definição da réplica contatada) e de mensagem (fragmentação dos dados);
- As condições do meio podem mudar durante a execução de um sistema, sendo ideal sua adaptação em tempo de execução;
- As políticas podem ser trocadas dinamicamente;

02 – Middleware Interceptador

- Técnicas utilizadas para implementar a adaptação em middlewares:
 - Separação de interesses (modularização);
 - Reflexão computacional;
 - Projeto baseado em componentes;
- A flexibilidade desejada em um middleware pode acarretar em grande complexibilidade;

02 – Multithread

- Evita bloqueio do processo principal;
- Mantém múltiplas conexões lógicas;
- Ex: conexões para descarregar páginas multimídia, browser com abas;
- Thread despachante;
- Thread operário;
- Máquina de estado finito;

02 – Virtualização

- Substituição de uma interface de modo a imitar o comportamento de outro sistema;
- Ex: sistema operacional, recurso de hardware;
- Integra a diversidade entre hardware e software;
- Formas de virtualização
 - Máquina virtual de processo: conjunto de instruções abstratas para executar aplicações. Ex: JVM, Emuladores;
 - Monitor de máquina virtual:
 - Conjunto completo de instruções do hardware em utilização ou de outro hardware é fornecido;
 - Vários SO em execução paralela e independente;

02 – Servidores

- Servidor Iterativo x Servidor Concorrente;
- Acesso às Portas:
 - Portas designadas. Ex: padronizadas globalmente;
 - Daemon (informa a porta do serviço);
 - Supervisor (inicia o serviço apenas quando requisitado);
- Transmissão de dados de controle fora da banda, em outra porta, com maior prioridade;
- Servidor com estado: mantém informações sobre clientes, usadas em caso de falha;
- Servidor sem estado: pode manter informações dos clientes, porém sem preocupação com falha;

02 – Modelos Arquiteturais



Referencias Bibliográficas

- Material do Prof. André Gustavo da UERN (andregustavo@uern.br) utilizado como base para composição da apresentação.