

- Aula 2 -**FUNDAMENTOS DE SISTEMAS DISTRIBUÍDOS****1. INTRODUÇÃO**

Nos primeiros momentos os computadores eram máquinas grandes e caras. Ao longo do tempo dois avanços tecnológicos mudaram os sistemas informatizados:

- O desenvolvimento do Microprocessador, que levaria ao desenvolvimento dos Computadores Pessoais (PC).
- O desenvolvimento das Redes de Computadores, que permitiriam que centenas de máquinas fossem conectadas possibilitando a transferência de dados entre usuários.

Essas são as bases para o surgimento e crescimento dos sistemas distribuídos atualmente.

2. SISTEMAS CENTRALIZADOS

Os primeiros sistemas funcionavam com uma organização de sistemas centralizados, onde um único computador detinha o controle do hardware e software disponível. Na década de 60, com o desenvolvimento da multiprogramação, sistemas de tempo compartilhado permitiram vários usuários acessarem os recursos de uma máquina central através dos chamados “terminais burros”.

A configuração dos sistemas centralizados de tempo compartilhado define a quantidade de usuários e trabalhos que podem ser realizados de maneira eficiente.

Um maior número de usuários reduz a fatia de tempo (time-slice) disponível para cada um, aumenta o número de interrupções e consumo dos recursos do sistema para tarefas de gerenciamento. Além disso, o barramento pode causar um gargalo no sistema pela limitação de dados que podem circular nas conexões.

Com o desenvolvimento dos microprocessadores, alguns sistemas centralizados, tipicamente associados a mainframes, passam a contar com múltiplos processadores com o objetivo de aumentar a quantidade de trabalhos produzida por unidade de tempo (*throughput*), porém mantendo suas características de segurança.

3. SISTEMAS PARALELOS (Múltiplos processadores)

A partir da década de 70, o desenvolvimento dos microprocessadores permitiu obter computadores de menor porte e de grande desempenho, definindo uma tendência para o *downsizing* dos sistemas computacionais. Esta tendência leva a substituição de sistemas de grande porte por computadores de menor porte interligados e a conseqüente descentralização.

A visão de Sistemas Paralelos está associada à sistemas genéricos de processamento distribuído entre vários nós de processamento (computadores individuais, não necessariamente

homogêneos) de maneira que um processo de grande consumo possa ser executado em um nó disponível ou subdividido por vários nós.



Figura 1 - Mainframe IBM 3083 (32 Mb de armazenamento)

O ganho mais óbvio na distribuição das tarefas está na possibilidade de uma tarefa ser divisível em várias subtarefas que serão executadas simultaneamente em paralelo. Apesar do aumento de produção, foi necessário implementar um Sistema de Controle de Processamento para gerenciar e distribuir as tarefas entre os processadores resultando em um tempo de processamento total maior que a soma das partes.

Com a descentralização, ocorre a distribuição dos dados e processos entre vários ambientes computacionais que mantêm comunicação, podendo estar dispersos geograficamente.

Os sistemas com múltiplos processadores oferecem benefícios como:

- Mais processadores executam mais tarefas em menos tempo, aumentando a produção (*Throughput*);
- Economia, pois o custo do processador é inferior ao custo do computador;
- Maior confiabilidade (Sistema Tolerante a Falha), uma vez que a degradação pode ser controlada, além de que a falha de um processador não paralisa o sistema.

As características específicas dos sistemas paralelos são definidas pela forma de comunicação entre os processadores e o grau de compartilhamento dos recursos:

- **Sistemas Fortemente Acoplados (*Tightly Coupled*):** Processadores compartilham memória (único espaço de endereçamento) e são interligados por um barramento;
- **Sistemas Fracamente Acoplados (*Loosely Coupled*):** A computação é distribuída entre vários processadores físicos que trocam mensagens através de linhas de comunicação e possuem sua própria memória local.

O compartilhamento de recursos nos sistemas fortemente acoplados se dá através de interrupções e a comunicação é feita via memória compartilhada. Nos sistemas fracamente acoplados, os processadores possuem seus próprios recursos individuais e estão conectados por barramento de alta velocidade, linhas telefônicas ou outro meio de comunicação.

3.1. Sistemas Fortemente Acoplados

O sistema fortemente acoplado mais comum é definido como Multiprocessamento Simétrico (*Symmetric Multiprocessing* - SMP), onde não há hierarquia entre os processadores e cada processador executa uma cópia idêntica do Sistema Operacional que é responsável pelo gerenciamento dos processos e do compartilhamento de recursos. A maioria dos sistemas operacionais modernos suporta SMP.

Em sistemas de Multiprocessamento Assimétrico, um processador tem o controle de todos os recursos do sistema em uma estrutura mestre/escravo que se assemelha aos sistemas centralizados, possuindo mesmos benefícios e problemas. Estes sistemas utilizam processadores específicos para tarefas específicas podendo aumentar a eficiência para tais tarefas, porém aumentando a complexidade do sistema.

O primeiro sistema a utilizar a arquitetura SMP foi o Burroughs B 5500 em 1961, que podia utilizar 1 ou 2 processadores.



Figura 2 - Burroughs B5500 (University of Virginia Computer Museum)

Como todos os sistemas com múltiplos processadores, o SMP também é tolerante à falhas. Se um processador falhar, outro processador assume a tarefa. Em situações onde há mais de uma tarefa a ser realizada, este sistema possui desempenho melhor que os sistemas monoprocessados, uma vez que as tarefas podem ser realizadas simultaneamente em paralelo.

Similar a esta arquitetura, a tecnologia multi-core (duo-core, quad-core, ...) pode ser considerada como uma arquitetura SMP aplicada aos núcleos que fazem o papel de processadores conectados por barramento e compartilhamento memória e sistema operacional.

3.2. Sistemas Fracamente Acoplados

Sistemas de Rede são sistemas fracamente acoplados que atuam de maneira independente e trocam mensagens através de uma infraestrutura de rede que pode ser local (LANs) ou de longa distância (WANs). É possível trabalhar com arquiteturas do tipo cliente-servidor ou *peer-to-peer*.

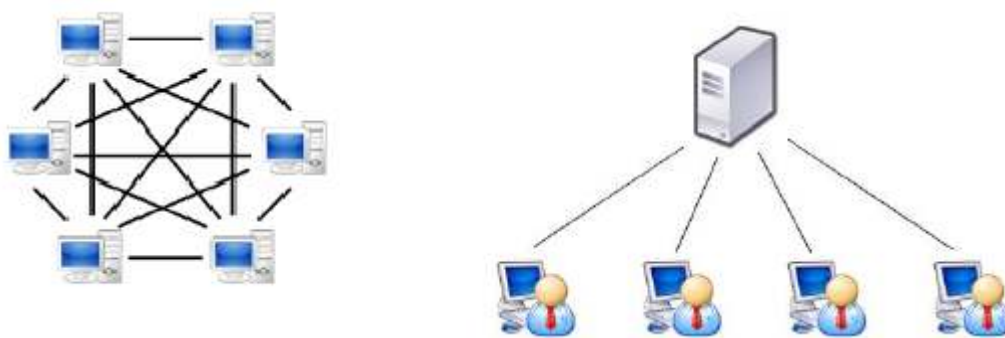


Figura 3 - Arquitetura cliente-servidor e ponto-a-ponto

Em sistemas de rede, fracamente acoplados (*Loosely Coupled*), os computadores estão interligados entre si, mas mantêm sua “individualidade”, isto é, cada computador tem seus próprios recursos de processador e memória local e utilizam um sistema de comunicação como linha telefônica ou barramento de alta velocidade para trocar mensagens com os demais computadores da rede.

Não há necessidade dos computadores envolvidos na rede possuírem configurações similares de hardware ou software podendo, cada computador, ter seu Sistema Operacional particular. Tudo o que é necessário é conhecer o protocolo de comunicação. Estes sistemas são altamente flexíveis, pois são independentes da tecnologia, porém a comunicação através de troca de mensagens reduz o desempenho do processamento global.

3.3. Sistemas Distribuídos

Semelhante aos Sistemas de Rede, estes sistemas são menos autônomos, com seus componentes, interagindo o suficiente para dar a impressão de um único sistema. São vantagens dos Sistemas Distribuídos:

- Eficiente compartilhamento de recursos e da carga de trabalho;
- Distribuição mais eficiente permite computação mais rápida;
- Capacidade de Redundância;
- Sistema de Tolerância a Falhas.

Os sistemas distribuídos precisam que seus componentes possuam a mesma tecnologia para obter a ilusão de um sistema único. Esta tecnologia única pode ser obtida através de hardware ou de software que implemente a portabilidade dos sistemas.

Um Sistema Operacional único para todos os computadores pode permitir o balanceamento da carga de trabalho e dos recursos e controlar a redundância dos dados.

Muitas vezes utilizado como sinônimo de Sistemas Distribuídos, o Cluster é o principal exemplo de sistema distribuído e oferece um compartilhamento de armazenamento, sendo uma arquitetura intermediária entre sistemas de multiprocessamento utilizada para computação de alto desempenho com alta confiabilidade.

- **Clustering assimétrico:** Um servidor executa a(s) aplicação(ções) enquanto os outros ficam em espera.

- **Clustering simétrico:** Todos os *hosts* estão executando a(s) aplicação(ções) .



Figura 4 – Cluster Beowulf

3.4. Sistemas Distribuídos x Redes

Os termos Sistemas em Rede e Sistemas Distribuídos são utilizados para designar sistemas fracamente acoplados que possuem as características de compartilhamento de recursos e confiabilidade (tolerância a falhas), porém possuem características próprias que os diferenciam:

- Em Sistema de Redes temos computadores interligados em que o usuário (requisitante do serviço) deve explicitamente especificar em qual máquina irá executar seus programas ou obter seus recursos.

- Em sistemas distribuídos temos a Transparência do Sistema: o usuário não “vê” os demais computadores interligados permitindo que o sistema procure os recursos.

Apesar de todos os sistemas fracamente acoplados possuírem importantes características comuns e serem referenciados quase como sinônimos, é necessário reconhecer as diferenças entre os sistemas de Rede e Distribuídos, pois suas características próprias determinam seu grau de desempenho e confiabilidade, sendo uma informação importante para a configuração de um sistema que proverá a solução de algum problema.

4. SISTEMAS DISTRIBUÍDOS

Os primeiros conceitos de Sistemas Distribuídos surgiram na década de 70 quando as redes de computadores começaram a se tornar viáveis técnica e economicamente. Assim, de modo geral a disciplina de Sistemas Distribuídos faz estudos sobre todos os tipos de redes em que há compartilhamento de recursos. Apesar da visão inicial deste compartilhamento estar ligada a recursos como impressora, arquivos, banco de dados, etc... Sistema Distribuído também está associado ao conjunto de processos concorrentes acessando recursos distribuídos, os quais podem ser compartilhados através de passagem de mensagens em ambiente de rede.

“Um Sistema Distribuído consiste de um conjunto de processos distintos, espacialmente separados que se comunicam por troca de mensagens”

Lamport, 1978

Pela abordagem inicial dada por Lamport em 1978, qualquer rede de computadores interligada é considerado um sistema distribuído. Este autor ainda considera que um sistema é distribuído se o atraso da transmissão da mensagem não é desprezível comparada ao tempo entre eventos de um processo único. O conceito de tempo associado à transmissão de uma mensagem é importante em sistemas distribuídos e é derivado do conceito de ordem dos eventos e é de particular interesse na solução de problemas de sincronização.

“Sistema no qual os componentes de hardware e software localizados em uma rede de computadores se comunicam e coordenam suas ações somente por troca de mensagens”

Coulouris et al, 2001

Nesta definição, computadores conectados em rede devem estar espacialmente separados por alguma distância entre si, resultando como conseqüências:

- Concorrência
- Não existência de relógio global
- Falhas Independentes

A concorrência (e seus problemas) existe porque no ambiente de redes, cada computador é independente permitindo o trabalho simultâneo em várias máquinas.

A inexistência de relógio global é conseqüência do fato do tempo necessário para a troca de mensagens não ser desprezível em sistemas geograficamente dispersos. Esta consideração é motivada diretamente pelo ambiente de rede e leva as mesmas idéias de sincronização consideradas por Lamport.

Considerando que todos os sistemas computacionais falham, Coulouris considera que os sistemas distribuídos falham de várias maneiras, isolando uma parte do sistema sem acarretar

SISTEMAS DISTRIBUÍDOS

Prof. Ricardo Rodrigues Barcelar
<http://www.ricardobarcelar.com.br>

falha global. Algumas destas falhas são associadas a problemas diversos de comunicação e algumas vezes são referidas como falhas Bizantinas.

“Sistema Distribuído é um conjunto de processadores que não compartilham memória nem relógio. Os processadores se comunicam por meio de redes de comunicação”

Silberschatz, 2000

Silberschatz considera que os sistemas distribuídos são mais flexíveis e seus componentes possuem maior autonomia se assemelhando muito a sistemas de rede, porém com o controle de um único Sistema Operacional, conforme figura 5.

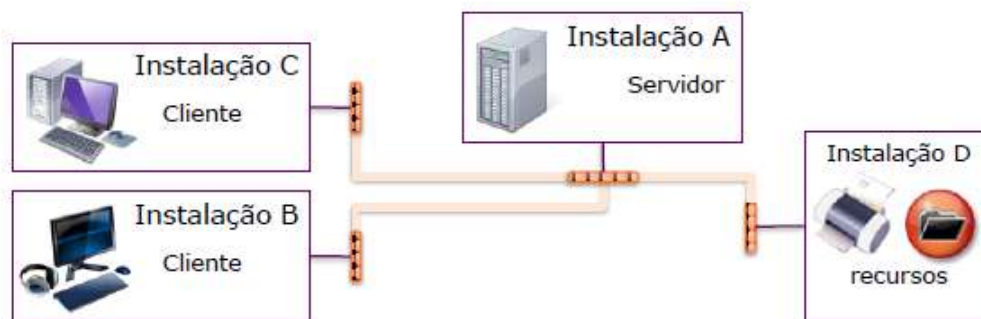


Figura 5 - Rede de Comunicação

Como visto, os sistemas distribuídos não são diferentes dos sistemas em rede, são sistemas fracamente acoplados interconectados por uma rede de comunicação. Do ponto de vista de um processador específico, os demais processadores e respectivos recursos são remotos, enquanto seus recursos são locais. O propósito de um sistema distribuído é fornecer um ambiente eficiente para o compartilhamento de recursos remotos.

Novamente o conceito de relógio global é considerado. A questão de sincronização de sistemas cooperativos que trabalham de maneira concorrente apresenta um dos principais desafios para sistemas geograficamente dispersos.

A necessidade de um único sistema operacional se traduz em um ambiente homogêneo. Esta opção pode não ser a mais apreciada, uma vez que o ambiente heterogêneo é um atrativo em sistemas de compartilhamento de recursos dispersos, mas facilitaria a configuração e gerência destes sistemas.

“Sistema Distribuído é um conjunto de computadores independentes (autônomos) que se apresenta a seus usuários (pessoas ou programas) como um sistema único e coerente”

Tanenbaum, 2007

Nesta visão, sistemas distribuídos são compostos por sistemas com autonomia reduzida, tipicamente unidos por uma camada de software e comumente chamados de *Middleware* permitindo que possuam seu sistema operacional local.

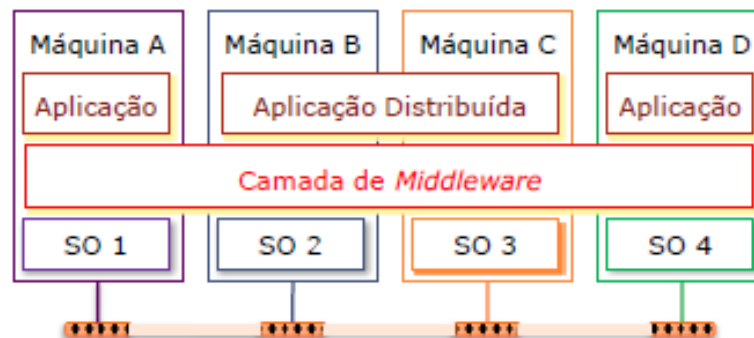


Figura 6 - *Middleware*

Para um conjunto de componentes autônomos aparentarem ser um único sistema deve ter componentes que colaboram entre si. Isto significa que precisarão executar as tarefas de maneira cooperativa, estando sujeitos aos mesmos problemas de sincronização referidos anteriormente.

Tanenbaum inicialmente desconsidera o tipo de computador utilizado (mainframe, PC, etc...) e a maneira de conexão entre estes computadores, assumindo o grau de liberdade e flexibilidade dos sistemas em rede. Para permitir um ambiente heterogêneo de rede e computadores, uma camada de *Middleware* pode se estender por várias máquinas para oferecer uma interface para as aplicações.

Apesar dos diferentes conceitos apresentados o título “Distribuído” usualmente é reservado para sistemas formados por processadores dispersos geograficamente compartilhando recursos e cooperando para a realização de alguma tarefa.

Em suma, podemos finalmente conceituar **Sistema Distribuído** como sendo um **conjunto de sistemas autônomos (independentes em recursos) interligados por uma rede de comunicação que se comporta como um sistema fortemente acoplado**.

De modo geral, sistemas com múltiplos processadores ou com múltiplos fluxos de controle (*Threads*) são tratados como Sistemas Distribuídos. O critério básico para esta consideração é a possibilidade de compartilhar recursos entre componentes separados que trabalham de maneira cooperativa.

4.1. Características de Sistemas Distribuídos

Ter um Sistema Distribuído não significa que ele é a melhor solução. É preciso observar as características desejadas, como ter um fácil acesso aos seus recursos, ocultar o fato dos recursos estarem distribuídos em uma rede, ser flexível e poder ser expandido. Os principais objetivos de um sistema distribuído são:

- Compartilhamento de Recursos
- Confiabilidade
- Desempenho

4.1.1. COMPARTILHAMENTO DE RECURSOS

A principal meta dos Sistemas Distribuídos é facilitar o acesso a recursos remotos (hardware, dados, páginas Web, etc...) e seu compartilhamento. A razão óbvia para o compartilhamento dos recursos é a economia. O compartilhamento também facilita a colaboração e a troca de informações entre usuários separados geograficamente, permitindo que trabalhem em conjunto em organizações virtuais (*groupware*¹).

É mais barato compartilhar uma impressora (ou outro componente) por diversos usuários do que ter uma impressora para cada usuário. Também é economicamente justificável compartilhar equipamentos de mais alto custo como supercomputadores ou mesmo estações de tratamento de imagens e unidades de armazenamento de alto desempenho e capacidade.

4.1.2. CONFIABILIDADE

De modo geral espera-se que os sistemas distribuídos sejam mais confiáveis do que sistemas centralizados, uma vez que existe redundância de recursos. Podemos definir a confiabilidade como a probabilidade de um sistema estar operando ou funcionando em certo intervalo de tempo. Para ser confiável, um sistema deve possuir alta disponibilidade e segurança. É comum utilizar mecanismos de redundância (de hardware ou software) para melhorar essa confiabilidade do sistema.

Algumas tarefas devem ser executadas de maneira contínua não podendo sofrer interrupções como aquelas associadas ao sistema financeiro (Bancos, bolsa de valores,...), telemedicina e sistemas críticos (tempo real), entre outras.

Um sistema para ser confiável não significa que ele é seguro. A segurança do sistema está ligada a integridade dos dados e a confiabilidade está associada à operação correta.

Sendo um defeito mecânico ou algorítmico, uma falha ocorre quando o comportamento do processador não é consistente com sua definição semântica o que gera um erro que pode causar a paralisação do sistema.

Há dois tipos de falhas dependendo de seu comportamento:

- **Fail-stop:** Em uma falha de parada, o sistema para de funcionar após mudar para o estado no qual a falha pode ser detectada. É um modo de falha muito simples cujos danos visíveis são o sistema parar de funcionar na qual o estado interno e os dados contidos neste estado são perdidos. Como estes problemas são visíveis e facilmente detectados, a sua prevenção é facilitada.

- **Falha Bizantina:** No caso das falhas bizantinas o sistema continua trabalhando, mas não produz os resultados corretos. Esta falha é tipicamente associada à comunicação entre os componentes do sistema e normalmente são causadas por erros não detectados em algum

¹ *Groupware*: Termo usado para descrever softwares que facilitam a colaboração e o trabalho em grupo através do compartilhamento de informações. A maior parte dos produtos *Groupware* fornece uma interface para ambiente colaborativo permitindo aos usuários compartilhar itens como calendários e contatos, garantindo maior produtividade dos usuários. Por exemplo, uma secretária pode alocar um compromisso no calendário do chefe, de forma que este consiga visualizá-lo de qualquer lugar via Internet.

software do sistema e seus danos nem sempre são facilmente visíveis. Conseqüentemente são muito mais difíceis de serem detectadas, dificultando sua prevenção e correção.

Outro aspecto da confiabilidade é a disponibilidade, que pode ser vista como a probabilidade do sistema estar operando em um determinado instante do tempo e é essencial para que um sistema seja confiável. Tanto a confiabilidade quanto a disponibilidade para sistemas críticos são representadas na forma 99,9999% do tempo. Na prática, esta característica indica o tempo que o sistema pode funcionar ininterruptamente sem ser afetado por falhas, manutenção preventiva ou corretiva, etc.

4.1.3. DESEMPENHO

Em geral, espera-se que uma aplicação distribuída apresente um desempenho melhor que aquele atribuído a um sistema centralizado. O desempenho é uma grandeza que pode ser avaliada de diferentes formas dependendo do tipo de aplicação, uma vez que este fator desempenho pode estar associado a diferentes conceitos ou aspectos. Possíveis variáveis para medidas de desempenho são:

- Tempo de resposta do servidor
- *Throughput* (número de *jobs* por unidade de tempo)
- Quantidade de recursos consumidos pela rede
- Resultados de *benchmarks*
- Tempo de transmissão de dados

Um sistema com múltiplos processadores fracamente acoplados pode obter um desempenho que seria impossível em um sistema com um processador, uma vez que podem reunir uma quantidade muito superior de processadores do que um sistema fortemente acoplado, mas raramente N processadores tem um desempenho N vezes maior que um processador (escalabilidade), uma vez que depende de troca de mensagens.

Por outro lado, um sistema com N processadores custa menos do que um processador com N vezes mais velocidade, especialmente quando é possível aproveitar máquinas potencialmente ociosas e realizar tarefas em paralelo, aumentando o *throughput* do sistema. Deste ponto de vista, os sistemas distribuídos apresentam uma melhor relação de preço/desempenho.

Uma solução distribuída também pode fazer uso de paralelismo de execução para melhorar o desempenho da aplicação, implicando que um determinado problema pode ser parcialmente resolvido em diferentes máquinas do sistema simultaneamente. Neste sentido, existem diferentes níveis de paralelismo:

- Paralelismo de Granularidade Fina (*fine-grained*)
- Paralelismo de Granularidade Grossa (*coarse-grained*)

A granularidade consiste na avaliação do nível de detalhamento do comportamento de um elemento que é exposto. Quanto mais bem detalhado é o comportamento do seu objeto é exposto, mais fina é sua granularidade.

Atualmente, a granularidade tem recebido particular atenção devido à sua importância na Arquitetura Orientada à Serviços (SOA) por ser um fator decisivo na definição de uma abordagem por componentes do sistema que permite maior flexibilidade (Orientado a Reusabilidade) ou na padronização que permite maior produtividade (Orientado a Negócios).

4.1.3.1. Desempenho e Granularidade

Paralelismo de Granularidade Fina implicam em mais flexibilidade, porque há mais, pequenos incrementos, pequenas unidades funcionais. O Paralelismo de Granularidade Grossa possui grandes pedaços de capacidade funcional como em grandes processos computacionais onde há poucas interações e poucos dados.

Assim, serviços com pouco detalhamento têm uma estrutura grande e, portanto, uma granularidade grossa (*coarse-grained*) também chamada de baixa granularidade. Os serviços com muito detalhamento são serviços de pequena estrutura e possuem uma granularidade fina (*fine-grained*) ou alta granularidade.

4.2. Outros aspectos dos Sistemas Distribuídos

Em uma visão mais ampla podemos considerar que os sistemas distribuídos fazem parte do nosso dia a dia e possuem um projeto bastante simples. Mesmo dessa forma é necessário considerar alguns tópicos que podem ser tornar um desafio no projeto de um sistema distribuído:

- Heterogeneidade
- Abertura
- Segurança
- Transparência
- Escalabilidade
- Tolerância a Falhas
- Concorrência

4.2.1. HETEROGENEIDADE

Sistemas Distribuídos precisam ser construídos a partir de uma variedade de diferentes redes, sistemas operacionais, hardware e linguagens de programação. Os protocolos de comunicação, usados tipicamente na internet e em outras redes de computadores, mascaram as diferenças entre as redes e o *Middleware* pode lidar com as outras diferenças.

Um *Middleware* é um termo aplicado a camadas de software que fornecem uma abstração de programação, bem como uma máscara de heterogeneidade das camadas de redes, sistemas operacionais, hardware e linguagens de programação.

O *Middleware* fornece um modelo computacional uniforme para programadores. Alguns exemplos de *Middleware*:

- CORBA: Provê “Invocação remota de objetos” o que permite que um objeto executando em um computador invocar um método de outro objeto executando em outro computador. Esta implementação esconde o fato de que mensagens são transmitidas por uma rede a fim de enviar uma requisição de invocação e sua réplica.

- Java RMI: Semelhante ao modelo CORBA, a “Invocação Remota de Métodos” da linguagem provê um mecanismo de *middleware* baseado em uma linguagem de programação única e utiliza uma máquina virtual (JVM) para obter um código portátil. Apesar de esta solução permitir a comunicação entre diferentes arquiteturas, geralmente não pode ser aplicada a outras linguagens de programação.

4.2.2. 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. Em geral, os serviços são especificados por meio de interfaces que costumam ser descritas em uma Linguagem de Definição de Interface (IDL – *Interface Definition Language*). Especificações adequadas de interfaces devem ser completas e neutras, características importantes para interoperabilidade e portabilidade.

A interoperabilidade caracteriza até que ponto duas implementações de sistemas ou componentes de fornecedores diferentes devem coexistir e trabalhar em conjunto, com base na mera confiança mútua nos serviços de cada um, especificados por um padrão comum.

A portabilidade caracteriza até que ponto uma aplicação desenvolvida para um sistema distribuído pode ser executada em outro sistema distribuído diferente que implementa as mesmas interfaces.

Junto com estas características da abertura, é desejável que o sistema seja extensível, permitindo maior flexibilidade do sistema.

A abertura é a característica que determina se um sistema pode ser estendido (ampliado) e reimplementado de várias maneiras. Em redes de computadores de modo geral, ela é expressa pela flexibilidade obtida com a utilização dos protocolos de comunicação que permitem diferentes arquiteturas interagir trocando mensagens. Nos sistemas Distribuídos esta abertura é determinada pelo grau pelos quais novos serviços de compartilhamento de recursos podem ser adicionados e ficarem disponíveis para uso por uma variedade de programas clientes.

A abertura de um sistema só é obtida se a especificação e a documentação dos softwares que definem as interfaces dos componentes do sistema forem conhecidas e padronizadas. Assim, esta especificações devem ser completas significando que tudo que é necessário para uma implementação foi, de fato, especificado. O fato de ser neutra significa que a especificação não define como deve ser a aparência da implementação.

4.2.3. SEGURANÇA

Muitas das informações disponíveis nos sistemas distribuídos têm um alto valor intrínseco para seus usuários, portanto sua segurança é consideravelmente importante. A segurança para recursos de informação possui três componentes:

- Confidencialidade: Proteção contra revelação para pessoas não autorizadas;
- Integridade: Proteção contra alteração ou corrupção;
- Disponibilidade: Proteção contra interferência em relação ao acesso ao recurso.

A proteção dos recursos compartilhados é um dos grandes problemas dos sistemas distribuídos, uma vez que o acesso a esses recursos deve ser facilitado para atingir seu objetivo de compartilhamento.

A codificação dos dados (criptografia) pode prover proteção adequada na transmissão dos dados, mas ainda é difícil evitar ataques como o de Recusa de Serviço (*Denial of Service* - DoS).

Em ataques de Recusa de Serviço (DoS) grandes quantidades de requisições sem sentido são feitas a um serviço de modo que um usuário que realmente necessita do serviço não consegue obtê-lo.

4.2.4. TRANSPARÊNCIA

Uma das principais necessidades dos Sistemas Distribuídos é tornar o ambiente de rede invisível provendo a visão de um ambiente único para o usuário. Um Sistema Distribuído que é capaz de se apresentar a usuários e aplicações como se fosse um único sistema de computador é denominado transparente (Tanenbaum). O objetivo é tornar certos aspectos da distribuição invisíveis ao programador de forma que ele precise se preocupar apenas com o projeto de sua aplicação particular.

O usuário não deve ter conhecimento da distribuição de recursos do sistema, portanto é importante ocultar o fato de que os processos e demais recursos estão fisicamente distribuídos. Desta forma não é necessário se preocupar com o local onde uma aplicação está instalada ou os detalhes de como suas operações serão acessadas por outros componentes.

Assim, os recursos que não são diretamente relevantes para a execução de uma tarefa de interesse num determinado instante são escondidos e passam anônimos através da transparência. É fácil perceber que, apesar das vantagens, a transparência nem sempre é desejada. Veja o caso em que o usuário necessita, por exemplo, um documento impresso. É usual que o trabalho seja executado em uma impressora próxima ao usuário, sendo mais conveniente deixar que o usuário faça a escolha dessa impressora. Na prática, um sistema transparente oculta alguma informação do usuário ou dos processos.

Tabela 1- Tipos de transparência

Transparência	Descrição
Acesso	Ocultar diferenças na representação de dados e no modo de acesso
Localização	O local do recurso é desconhecido
Migração e relocação	O recurso pode ser movido, inclusive enquanto está em uso
Replicação	Múltiplas instâncias de um recurso podem ser utilizadas para aumentar confiabilidade e desempenho sem o conhecimento destas réplicas
Concorrência	Vários usuários podem compartilhar o mesmo recurso
Falha	A falha e a recuperação de um recurso não podem afetar o sistema

Os tipos de transparência são reconhecidos pela *International Organization for Standardization* (ISO) de 1995, porém alguns outros tipos podem ser considerados como a transparência de Desempenho ou Mobilidade dada por Coulouris.

4.2.5. ESCALABILIDADE

O aumento do desempenho em sistemas informatizados encontra sua limitação na capacidade tecnológica de conectividade entre um grande conjunto de processadores. À medida que os sistemas são ampliados podem apresentar perda de desempenho devido às sobrecargas que são geradas no gerenciamento do ambiente e da solução, mas seu comprometimento pode ser impedido com uma boa arquitetura. Um sistema cujo desempenho aumenta com o acréscimo de hardware proporcionalmente à capacidade acrescida é chamado escalável.

Um dos argumentos que levou à descentralização dos sistemas está vinculado ao fato de que sistemas fortemente acoplados possuem limitação prática no aumento do desempenho, pois necessitam uma infra-estrutura especial de barramento e memória para aumentar o número de processadores de forma que não há uma boa relação de custo e benefício.

A escalabilidade é uma das mais importantes metas e um dos principais argumentos em favor dos Sistemas Distribuídos, podendo ser abordada por diferentes pontos de vista:

- **Escalável quanto ao tamanho:** Podem-se acrescentar mais recursos e usuários ao sistema.
- **Escalável geograficamente:** Usuários e recursos podem estar distantes uns dos outros.
- **Escalável Administrativamente:** O sistema ainda pode ser fácil de gerenciar, mesmo abrangendo organizações administrativas diferentes.

Projetar um sistema distribuído escalável apresentará diversos desafios:

- **Custo dos recursos físicos:** à medida que a demanda do sistema cresce, deve ser possível crescer os recursos a um custo razoável.
- **Perda de desempenho:** com o crescimento do sistema a quantidade e distribuição dos recursos devem aumentar, porém isso não pode gerar uma maior complexidade no gerenciamento dos mesmos ou causar considerável redução do desempenho.

- Prevenção da falta de recursos: com o crescimento dos sistemas informatizados é muito difícil fazer uma previsão das necessidades de recursos em longo prazo.

- Gargalos no sistema: se um determinado recurso for muito requisitado, a tentativa de utilização poderá causar um gargalo de desempenho no sistema.

4.2.6. TOLERÂNCIA A FALHAS

Tolerância a Falhas é a capacidade de o sistema sobreviver à falha de alguns dos seus elementos e está associado intimamente à transparência de falhas, confiabilidade e disponibilidade do sistema.

Em sistemas distribuídos, espera-se que a falha de um componente do sistema não afete de maneira significativa a tarefa sendo realizada, isto é, pode haver perda de desempenho, mas a tarefa é realizada. Questões sobre o efeito de uma falha no sistema são as mais difíceis de responder e sanar e dependem dos objetivos do sistema, bem como de sua estrutura de recursos.

Tipicamente obtém-se a tolerância a falhas com a redundância de hardware e software. Se as características de transparência discutidas anteriormente forem atingidas, é possível migrar um recurso de hardware e software de um sistema para outro que esteja disponível sem o conhecimento do usuário.

Questões importantes:

- Como o sistema vai responder à falha de alguns de seus componentes?
- Parte dos seus serviços ficará indisponível?
- Todos os serviços?
- Os demais componentes vão compensar as tarefas daquele que falhou?
- O sistema vai se tornar totalmente inútil?
- Se o sistema for um avião, que acontecerá se um dos computadores de bordo falhar?
- O que deveria acontecer?
- Como detectar a falha?

4.2.7. CONCORRÊNCIA

Um dos grandes atrativos dos sistemas distribuídos é a possibilidade de compartilhar recursos de serviços e aplicativos. Este compartilhamento abre a possibilidade de que um determinado recurso tente ser acessado por vários clientes ao mesmo tempo como, por exemplo, o acesso aos lances em um leilão (mercado livre, e-bay, etc...) que chaga ao limite de tempo. Limitar o recurso compartilhado a um cliente por vez resolve o problema de concorrência, mas também limita a produção do sistema o que não é desejado. Assim, usualmente sistemas distribuídos permitem o acesso concorrente dos recursos, porém é comum utilizar objetos encapsulados que assumam a responsabilidade de alguma estrutura de sincronização.