

[◀ VOLTAR](#)

Conceitos e Metas de Sistemas Distribuídos

Descrever os conceitos, objetivos e metas em torno dos Sistemas Distribuídos

NESTE TÓPICO

- › Conceitos de Sistemas Distribuídos
- › Objetivos de Sistemas Distribuídos
- › Metas dos Sistemas Distribuídos



Ao fazer a leitura dos tópicos anteriores, certamente notará que alguns acontecimentos foram marcantes para a evolução dos sistemas computacionais. Sobre este tema e de forma sucinta, vale destacar a redução de tamanho e o aumento do desempenho do computador a partir das mudanças na arquitetura de hardware, na maneira como as tarefas foram automatizadas visando o aumento de produtividade do computador e o surgimento dos sistemas operacionais. Todos estes avanços foram importantes, mas ainda as empresas e centros de pesquisa executavam suas tarefas em computadores centralizados e sem comunicação alguma. Diante disso, dois acontecimentos tecnológicos marcaram a era moderna da computação em geral, são eles:

1. **O desenvolvimento dos microprocessadores:** a produção em grande capacidade de processadores de 8, 16, 32 e 64bits.
2. **O desenvolvimento das redes de computadores:** das redes locais (LANs) que transmitem mensagens entre computadores dentro de uma empresa a taxas de 10 bilhões de bits por segundo, para as redes de longa distância (WANs) que conectam computadores no mundo inteiro utilizando velocidades que podem variar entre 64 Kbits a Gigabits por segundo.

Estes dois principais acontecimentos contribuíram para a descentralização dos sistemas até o surgimento dos atuais sistemas distribuídos.

Conceitos de Sistemas Distribuídos

Um sistema distribuído é composto por computadores que possuem autonomia e se mostram aos seus usuários (pessoas ou outros softwares) como um sistema único e coerente. Assim, quando se decide em distribuir os componentes e aplicações, pretende-se atingir um contingente maior de usuários, melhorar a produtividade destes usuários e a disponibilidade do acesso à informação.

Exemplos de sistemas distribuídos

- **Intranet:** É um sistema de informação distribuído acessível pela plataforma web, mas com disponibilidade de acesso somente aos usuários de uma empresa, departamentos ou grupos de trabalho, ou seja, o acesso é feito a partir do interior da empresa (veja a figura 1). O objetivo é o compartilhamento de informações internas da empresa e de recursos como bancos de dados, impressoras e arquivos.
- **Extranet:** Assim como a Intranet, trata-se um sistema de informação distribuído acessível pela plataforma web. Mas o acesso pode ser realizado fora da empresa a partir de uma página de login protegida por protocolos de criptografia que solicita um usuário e senha, e uma rede privada (veja a figura 1). O objetivo é o compartilhamento de informações internas da empresa e de recursos como bancos de dados, impressoras e arquivos, além de facilitar o acesso a estes recursos em casa ou de qualquer lugar (*home office*). Além disso, a Extranet pode ser composta por um grupo de Intranets interconectadas formadas por empresas fazem parcerias de negócios entre si.
- **Internet:** Trata-se de um sistema distribuído com acesso a partir de qualquer computador independentemente da localização geográfica e de qualquer dispositivo que tenha acesso à rede móvel ou cabeada.
- **Computação Móvel:** Trata-se de pequenos dispositivos móveis com menor capacidade de memória, armazenamento e desempenho, mas que oferecem aos seus usuários uma experiência com alta mobilidade e disponibilidade de acesso.
- **Computação Ubíqua:** É um modelo de computação distribuída comumente presente em locais fechados ou com acesso limitado a um grupo de usuários. São formados por dispositivos móveis, computadores, televisores, PDAs (*Personal Digital Assistants* - Assistentes Pessoais Digitais) e POSs (*Point of Sales* ou Ponto de Vendas) e estão presentes em grandes lojas, supermercados, residências e escritórios.



Objetivos de Sistemas Distribuídos

Os sistemas distribuídos possuem objetivos muito claros que dependem do tipo de implementação que podem atingir uma ou mais metas que serão discutidas a seguir:

- **Compartilhamento de Recursos:** O compartilhamento de recursos é um fator importante para o trabalho colaborativo entre os usuários e na disponibilidade de acesso. Assim, os sistemas distribuídos compartilham recursos como arquivos, diretórios, impressoras e bancos de dados a partir de qualquer computador ou dispositivo móvel.

- **Confiabilidade:** A confiabilidade é um fator importante para as premissas de disponibilidade e segurança da informação em sistemas distribuídos. Para tanto, espera-se que sejam mais confiáveis que os sistemas centralizados ou de tempo compartilhado, tendo em vista que podem conter diversas réplicas e ser mais tolerantes à falha.
- **Desempenho:** Devido ao modelo de arquitetura distribuída voltada para a replicação de elementos dedicados ao processamento (hardware e software), espera-se que, assim como na confiabilidade, os sistemas distribuídos tenham um desempenho superior aos sistemas centralizados.

Metas dos Sistemas Distribuídos

A ideia de um projeto de sistema distribuído depende, obviamente, dos objetivos de cada organização ou instituição, pois nem sempre implementar um projeto de sistema distribuído é a melhor estratégia. Como exemplo, imagine uma empresa de pequeno porte que tenha até 300 funcionários. Para esta empresa é mais importante manter os dados centralizados em um servidor por questões de segurança, pois, pelo porte dela, o desempenho e o acesso externo ao banco de dados não é um fator importante naquele momento.

Assim, o sistema distribuído deve ser projetado para atender os objetivos que vimos anteriormente (compartilhamento de recursos, disponibilidade e desempenho). Para ter êxito na implantação dos sistemas distribuídos, você deve observar as seguintes metas:

- Acesso a recursos
- Heterogeneidade
- Abertura
- Segurança
- Transparência
- Escalabilidade
- Tolerância a Falha
- Concorrência

As metas citadas são importantes para o sucesso de um sistema distribuído e discutiremos cada uma delas a seguir.



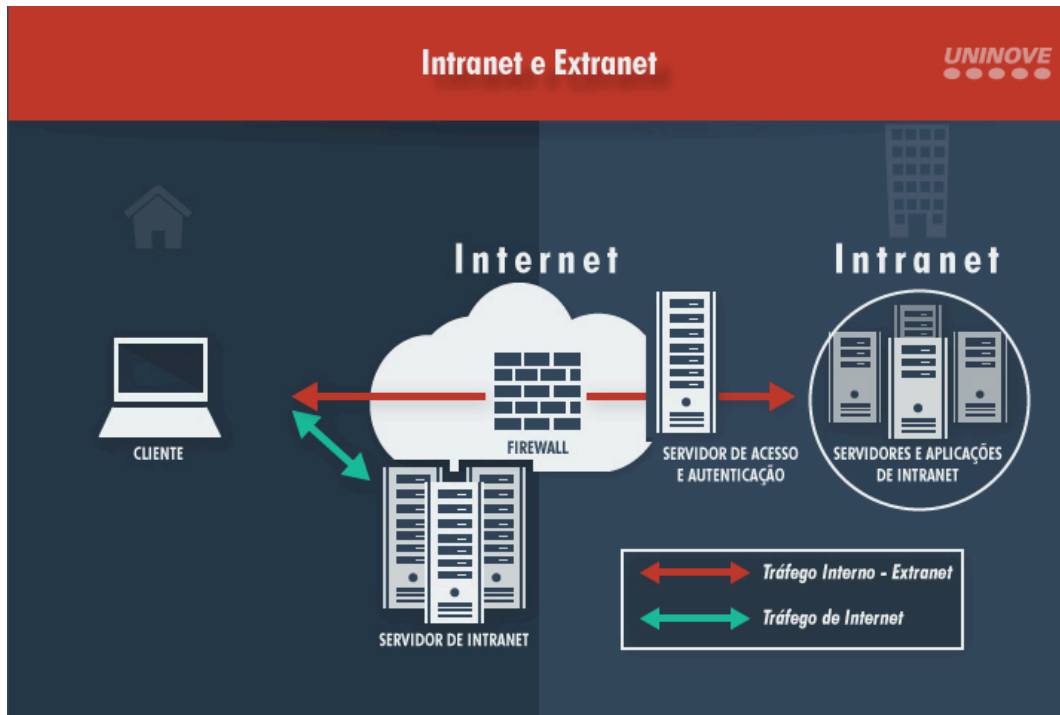


Figura 1 - Estrutura básica da Intranet e Extranet para exemplificar os sistemas distribuídos

Acesso a recursos

O acesso aos recursos computacionais é um fator importante para a economia e o aumento da produtividade dos usuários. Além disso, os usuários podem trocar informações estratégicas fundamentais para a tomada de decisão. Estas questões ajudam as empresas a se tornarem mais eficientes, colaborativas e competitivas. Muitas empresas estão investindo em organizações virtuais formadas por grupos de pessoas que colaboram entre si através de tele e/ou vídeo conferência, de software para colaboração, bancos de dados etc.



Heterogeneidade

Os sistemas distribuídos são compostos por diferentes hardwares, sistemas operacionais, linguagens de programação e redes de computadores. A razão disso, é que os sistemas legados (sistemas antigos da organização e que atendem essencialmente as necessidades dos usuários) foram desenvolvidos com linguagens de programação, são executados em hardwares e sistemas operacionais diferentes aos atuais ambientes Web e Móveis. Desta forma, reescrever todos os programas (incluindo os legados), substituir o hardware e os sistemas operacionais em busca de um ambiente homogêneo custará muito mais e levará muito mais tempo para implantar uma solução como essa. Uma boa solução na relação custo/benefício é o desenvolvimento ou aquisição de uma camada de software (*middleware*) que poderá lidar com todas estas diferenças (veja a figura 2). Os seguintes cenários abaixo retratam situações contendo os ambientes heterogêneos:

Cenário 1: Um computador cliente executando o hardware CISC com MS-Windows solicitando um serviço Web Apache que executa em um servidor CISC com Red Hat (GNU/Linux).

Cenário 2: Uma transação bancária que inicia a partir de um dispositivo móvel com Android e consome um recurso de *web service* em Java a partir do servidor Ubuntu *Server*. O servidor Ubuntu encaminha a requisição do usuário para um mainframe RISC executando o sistema operacional IBM-AIX (UNIX) com Cobol.

Cenário 3: Um computador cliente executando Debian solicita uma consulta de clientes para um serviço Web que executa em um servidor CISC com Windows *Server*. O servidor Web retransmite a requisição do cliente para uma grade computacional que executa em diversos sistemas operacionais diferentes (Linux/Windows/Unix). A comunicação entre o cliente e o servidor Web ocorre em rede cliente/servidor, mas a grade computacional executa em rede P2P (ponto a ponto).

Vimos nestes exemplos algumas situações onde temos diversos hardwares (RISC/CISC), sistemas operacionais (Windows/Linux/Unix), linguagens de programação (Java/Cobol) e redes (cliente-servidor/P2P) se comunicam durante todo o processo.

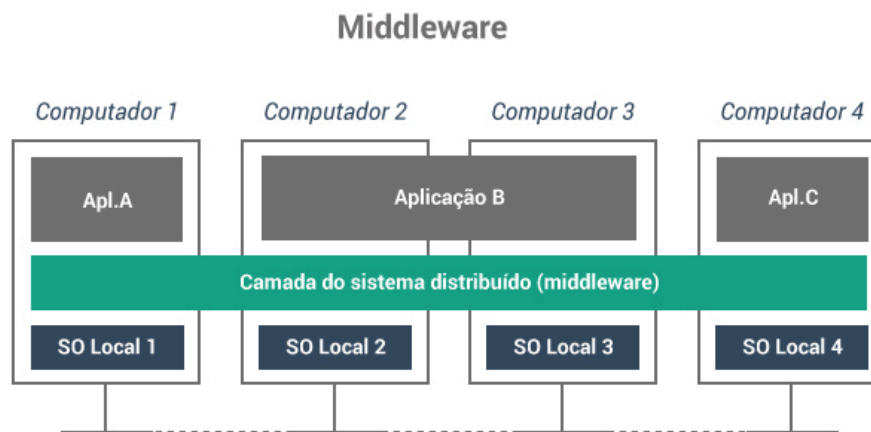


Figura 2 - Camada de Middleware.

Fonte: TANENBAUM, A. S.; STEEN, M. V.; Sistemas Distribuídos; Princípios e Paradigmas; Pearson Prentice Hall; 2ª edição; 2007.

Abertura

Com a expansão das redes de computadores, da Internet e dos sistemas móveis nos últimos 10 ou 15 anos, o acesso descentralizado às informações vitais se tornou imprescindível para os negócios empresariais. A necessidade de informação em tempo real e a competitividade no mundo dos negócios fizeram com que os sistemas computacionais fossem alinhados a esta nova realidade. Assim, um sistema distribuído aberto é aquele que oferece serviços aos seus usuários conforme as regras especificadas. Tais regras determinam o formato, o conteúdo e o significado destes serviços. Um exemplo disso são os protocolos de rede. Os protocolos governam o formato,

o conteúdo e o significado das mensagens que são enviadas e recebidas na rede. Seguindo esta concepção, os sistemas computacionais devem atender, ao menos, uma das características de abertura abaixo listadas:

- **Portabilidade:** Também conhecido como multiplataforma e trata-se do sistema distribuído que implementa as mesmas interfaces, mas que as executa em plataformas computacionais diferentes. Um sistema operacional com esta capacidade é o Unix, pois as mesmas interfaces implementadas para a plataforma de processadores CISC são as mesmas em RISC. Para tanto, o IEEE (*Institute of Electrical and Electronics Engineers* ou Instituto de Engenheiros Eletricistas e Eletrônicos) desenvolveu um padrão para as demais implementações Unix denominado por POSIX (*Portable Operations Systems for Interface Unix* ou Sistemas Operacionais Portáteis com Interface Unix). A linguagem Java também é portátil e, assim, viabiliza a implementação do mesmo código-fonte em qualquer plataforma de sistema operacional (Unix, GNU/Linux, Mac-OSx e Windows) a partir da utilização da JVM (Java *Virtual Machine* ou Máquina Virtual Java).
- **Interoperabilidade:** Trata-se do sistema distribuído desenvolvido por um fornecedor de software que se comunica com outro sistema distribuído de outro fornecedor a partir de uma linguagem de definição de interface (IDL - *Interface Definition Language*). A IDL permitirá a comunicação entre sistemas ou aplicações distribuídas a partir de uma confiança recíproca de serviços, ou seja, a partir de um método ou especificação comum de comunicação adotado entre as duas partes.

Segurança



A partir das premissas básicas para a adoção de um sistema distribuídos, tais como o compartilhamento de recursos e a disponibilidade, a segurança é uma preocupação cada vez mais importante neste contexto. Isso tanto é verdade que, a partir do momento que um diretório é compartilhado ou uma porta de serviço de rede é disponibilizado para acesso de usuários numa rede local, estamos indiretamente abrindo alguns pontos de vulnerabilidade que podem permitir o acesso de pessoas não autorizadas. Além disso, alguns sistemas transmitem sensíveis pela rede como texto puro, tais como informações confidenciais, dados de usuários e senhas. Estas informações sensíveis podem ser lidas por qualquer intruso. Assim, esse tipo de informação sensível não deve ser transmitida pela rede sem nenhum mecanismo de criptografia.

Outro problema está no rastreamento da comunicação para elaborar um perfil específico do usuário e assim enviar uma mala direta ou *spam* sem uma prévia autorização do usuário. Além disso, muitos perfis são montados para descobrir os dados de usuários e senhas ou dados de conta corrente. Isso se configuram em uma violação explícita da privacidade e produz sentimentos de desconfiança e descontentamento dos usuários. Desta forma, a segurança deve adotar uma política que alinhada com as três preocupações a seguir:

- **Integridade:** Trata-se de uma política de segurança que visa a proteção da informação contra corrupção ou alteração indevida.

- **Disponibilidade:** Trata-se de ações de segurança para manter a disponibilidade da informação. Caso estas ações não sejam efetivas, um invasor poderá tornar o acesso ao recurso, consequentemente à informação, indisponíveis.
- **Confidencialidade:** Trata-se de uma política de segurança que visa a proteção da informação e o controle de acesso para que pessoas não autorizadas tenham acesso às informações sigilosas.

Transparência

A transparência é uma meta importante para os sistemas distribuídos, pois esconde dos usuários o fato de muitos recursos computacionais e serviços estarem distribuídos entre diversos computadores pela rede. Um sistema distribuído que tem a capacidade de apresentar ao usuário como um sistema único e coerente é definido por transparente.

A transparência pode ser analisada por diversos aspectos:

- **Acesso:** A transparência do acesso esconde do usuário as diferenças de exibição dos dados e informações, além do modo de acesso ao recurso computacional. Isso se aplica para atender o objetivo de compartilhamento de recursos.
- **Localização:** A transparência da localização esconde do usuário a localização do recurso computacional. Isso se aplica para atender os objetivos de disponibilidade e compartilhamento de recursos.
- **Migração:** A transparência da migração esconde do usuário o fato do recurso computacional ter sido movido para outra localização. Isso se aplica para atender o objetivo de disponibilidade.
- **Relocação:** A transparência da relocação esconde do usuário a possibilidade de um recurso computacional ter sido movido para outra localização, inclusive no exato instante que está em operação. Isso se aplica para atender o objetivo de disponibilidade.
- **Replicação:** A transparência da replicação esconde do usuário o fato do recurso computacional possuir uma ou mais réplicas. Isso se aplica para atender os objetivos de confiabilidade e desempenho.
- **Concorrência:** A transparência da concorrência esconde do usuário a possibilidade do recurso computacional ser utilizado por diversos usuários simultaneamente. Isso se aplica para atender os objetivos de disponibilidade e compartilhamento de recursos.
- **Falha:** A transparência da falha esconde do usuário a falha de algum componente do sistema distribuído e a sua auto recuperação. Isso se aplica para atender o objetivo de confiabilidade.

A partir dos aspectos discutidos acima cabe uma reflexão: será que o caminho deve tender à transparência do sistema? A resposta é: Cada caso deve ser analisado de maneira ímpar. Um exemplo disso é a impressora. A impressora não deve estar oculta quanto ao seu estado (ocupada, livre, erro etc) e quanto à sua localização dentro da organização. Imagine um diretor-executivo que acabou de imprimir um documento confidencial e não sabe a



localização da impressora. Se o documento cair em mãos erradas, com certeza alguns potenciais problemas irão ocorrer. Outra situação é de um aplicativo móvel que está tentando a conexão de um servidor repetidas vezes antes de tentar em outro servidor. Muitos aplicativos somente desistem após inúmeras tentativas em vários servidores. O fato de tornar transparente as tentativas em diferentes servidores, pode tornar a aplicação lenta e gerar insatisfação do usuário. Desta forma, uma ideia melhor que tornar o processo totalmente transparente é deixar que o usuário decida em continuar ou não as tentativas de conexão em outros servidores.

Assim, a transparência é uma característica importante e que deve ser aplicada com cuidado para não interferir em questões como o desempenho e facilidade de compreensão dos procedimentos executados pela aplicação por parte dos usuários.

Escalabilidade

A quantidade de dispositivos conectados à Internet tem aumentado de forma significativa nos últimos anos. Isso permite que qualquer pessoa no continente sul-americano possa enviar uma mensagem para outra pessoa no continente asiático em fração de segundos. Tendo em vista a dimensão da Internet, os recursos que ela pode ou poderá oferecer e o número expressivo de usuários, estamos discutindo uma questão muito importante neste contexto: a ampliação do sistema.

Diante deste cenário de crescimento que vimos acima, a escalabilidade é a propriedade de um sistema distribuído qualquer que permite aumentar o seu desempenho com o acréscimo de hardware (computadores) de maneira proporcional a capacidade de recursos que foi acrescida. Em outras palavras, à medida que a concorrência de recursos aumenta (CPU, memória, disco etc) com o aumento de usuários, a quantidade de computadores dedicados ao processamento aumenta proporcionalmente, tornando-se um ambiente escalável. De certa forma, a escalabilidade de um sistema distribuído pode ser medida a partir de três perspectivas de escalabilidade listadas a seguir:

1. **Escalável quanto ao tamanho:** Permite adicionar novos recursos e usuários ao sistema.
2. **Escalável quanto à localização geográfica:** Permite que usuários e recursos do sistema estejam longe uns dos outros.
3. **Escalável quanto à gestão administrativa:** Permite a facilidade na gestão do sistema mesmo quando administrado por organizações administrativas diferentes.

As três dimensões da escalabilidade podem encontrar barreiras que surgem a partir de projetos e modelos computacionais antigos (mais aderentes às redes locais, por exemplo). Nestas barreiras, encontraremos problemas com algoritmos que utilizam técnicas de comunicação síncrona e bancos de dados centralizados. A comunicação síncrona é bem adequada para ambientes de rede local, pois cliente aguarda até centenas de microssegundos para obter uma resposta do servidor. A mesma arquitetura em um ambiente de rede de longa distância, irá produzir uma espera de centenas de milissegundos. Assim, o usuário terá longas esperas até poder continuar com a sua aplicação. Diante destes problemas, sugestões como



adotar arquiteturas descentralizadas para o sistema e os bancos de dados a partir de redes de longa distância, além dos algoritmos de comunicação assíncrona são muito recomendáveis para estes casos. A comunicação assíncrona permite que uma solicitação do cliente seja enviada para um servidor, mas o cliente não fica travado esperando a resposta. Assim, o cliente continua executando outras operações enquanto a requisição não retorna do servidor. Ao chegar a resposta do servidor, a aplicação finaliza a execução do que está fazendo e trata da apresentação da resposta ao usuário. Neste contexto, uma alternativa é a programação paralela com o uso de *threads*. Assim, ao invés de desenvolver um único processo, pode-se desenvolver a tarefa em processos menores (*threads*).

Tolerância à Falha

A tolerância à falha é uma das metas mais difíceis do projeto de um sistema distribuído, pois neste contexto, espera-se que determinadas falhas em um computador que o usuário nunca ouviu falar não impeça que o mesmo continue a sua tarefa. O principal desafio é avaliar quando uma falha tornou o computador totalmente indisponível (morto) ou que a falha tornou o computador muito lento ao ponto de deixá-lo indisponível ao usuário. Um exemplo fácil de entender a diferença entre um computador indisponível (morto) e um computador muito lento é um servidor Web. Se o servidor Web estiver sobrecarregado de processos e/ou sofrendo por um período de concorrência extrema dos usuários, o navegador (*browser*) um outro usuário terá que esperar o servidor Web responder diante da tamanha concorrência. O tempo de espera do navegador esgotará e exibirá que a página requisitada pelo usuário está indisponível. Neste exemplo, o usuário jamais saberá se o servidor está danificado ou se está lento.

Assim, mascarar falhas é uma questão difícil, pois envolverá, certamente, um projeto de algoritmo e de arquitetura distribuída muito bem modelado pensando nestas questões de tolerância e transparência da falha.



Concorrência

A concorrência é uma característica dos sistemas distribuídos quando atingimos os objetivos de compartilhamento de recursos e disponibilidade. Estes objetivos permitem que usuários em diferentes computadores espalhados geograficamente tenham acesso simultâneo aos recursos como bancos de dados e arquivos, por exemplo. O aumento desta concorrência pode gerar indisponibilidade do sistema e o recurso utilizado não deve permitir que dois usuários alterem o mesmo arquivo ao mesmo tempo. Para isso, é importante manter um *thread* que faça o controle criando uma trava no arquivo. Desta forma, cada usuário fará a sua alteração no seu tempo e o outro aguardará ou fará o acesso somente-leitura (*read-only*). Esta estratégia resolve a questão da concorrência, mas ao limitar o acesso ao recurso a um usuário por vez, irá produzir muita lentidão aos usuários. O mais correto neste caso, é a utilização de um algoritmo que crie um mecanismo de transações através de objetos no servidor. Assim, um *thread* poderá ser instanciado para cada usuário onde ficará responsável em executar cada transação paralelamente (inclusive em uma CPU diferente), liberando o usuário de uma fila de espera interminável.

Quiz

Exercício Final

Conceitos e Metas de Sistemas Distribuídos

INICIAR ➤

Referências

TANENBAUM, A. S.; STEEN, M. V.; Sistemas Distribuídos; Princípios e Paradigmas; Pearson Prentice Hall; 2ª edição; 2007.

TANENBAUM, A.S, Sistemas Operacionais Modernos, 3ª edição, Pearson Education do Brasil, 2010.



Avalie este tópico



ANTERIOR

A descentralização e os Sistemas Paralelos

Biblioteca

(<https://www.uninove.br/conhec-a->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Ajuda?

(<https://ava.un>

idCurso=)

Tipos de Sistemas Distribuídos

® Todos os direitos reservados

