

1. Definição de Sistema Distribuído
2. Motivações e Objetivos dos Sistemas Distribuídos
 - Vantagens do Sistemas Distribuídos sobre os Sistemas Centralizados
 - Vantagens do Sistemas Distribuídos sobre PCs independentes
3. Conceitos de Hardware
 - Múltiplos Processadores baseados em Barramento e Chaveados
 - Múltiplos Computadores baseados em Barramento e Chaveados
4. Conceitos de Software
 - Sistemas Operacionais de Rede
 - Sistemas Distribuídos Verdadeiros
 - Sistemas *Timesharing* Multiprocessados
5. Aspectos de Projeto de Sistemas Distribuídos
 - transparência, flexibilidade, confiabilidade, performance e escalabilidade

... Introdução aos Sistemas Operacionais Distribuídos

- ★ A evolução dos Sistemas Computacionais pode ser discutida em 02 etapas:
 - ... a primeira, de 1945 até meados da década de 80, onde computadores grandes e caros compunham os sistemas computacionais;
 - ... a segunda de meados da década de 80 para cá, onde avanços na tecnologia e a invenção de redes de alta velocidade começaram a mudar este cenário.
 - ☞ ... de um custo por máquina de 10 milhões de dolares e execução de 1 instrução por segundo, caminhamos para máquinas que 1000 dolares e 10 milhões de instruções por segundo.
 - ☞ ... com a advento das LANs e WANs, máquinas foram interconectadas umas às outras possibilitando que a informação fosse transferida em milisegundos com taxas variando de 64 Kbps a Gbps para algumas redes experimentais.
- ★ **Resultado:** viabilidade de compor sistemas computacionais compostos de um grande número de CPUs interconectadas por redes de alta velocidade.

- ... surge então os sistemas distribuídos em contraste com os sistemas centralizados ou *single-processor systems* compostos de uma única CPU, memória, periféricos e alguns terminais;
- ... a questão central é, portanto, o **software**.

Sistemas Distribuídos requerem software radicalmente diferente dos Sistemas Centralizados.

1 - O que é um Sistema Distribuído ?

A Distributed System is a collection of independent computers that appear to the users of the system as a single computer.

Andrew S. Tanenbaum

★ Aspectos chaves desta definição:

- ① *hardware*: as máquinas são autônomas;
- ② *software*: os usuários vêem o sistema como um único computador.

... 1 - O que é um Sistema Distribuído ?

★ Antes de continuarmos com mais definições, vejamos alguns exemplos de sistemas distribuídos:

- ❶ considere uma rede de estações em uma universidade, onde nenhuma processador em particular está associado permanentemente a qualquer usuário, mas alocados dinamicamente quando da necessidade.

... o que mais é preciso acrescentar para que este sistema se comporte como um sistema *timesharing* clássico monoprocessado, qualificando-o desta forma como um sistema distribuído ?

- ❷ considere uma fábrica repleta de robôs, com cada um capaz de tratar visão, planejamento, comunicação e outras tarefas.

como os robôs devem atuar para que o sistema seja contabilizado como um sistema distribuído ?

... 1 - O que é um Sistema Distribuído ?

★ Como exemplo final, pense sobre um grande banco com centenas de filiais espalhadas, cada uma com um computador principal para armazenar as contas locais bem como permitir as transações.

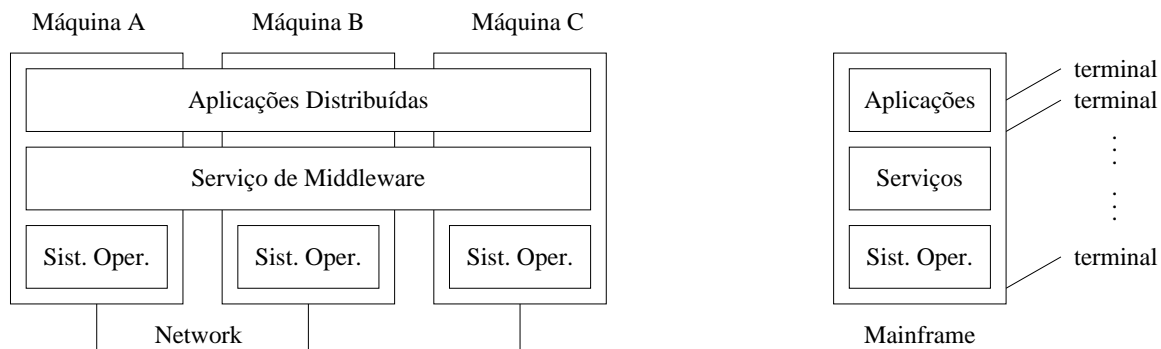
Adicionalmente, assuma que cada computador pode se comunicar com computadores de outras filiais bem como com o computador central.

- ... se as transações são realizadas sem que seja necessário considerar onde o cliente ou conta está;
- ... se os usuários não percebem qualquer diferença entre este sistema e um centralizado que o substituisse;

☞ este também deverá ser considerado como um Sistema Distribuído.

2 - Motivações e Objetivos dos Sistemas Distribuídos

- **Objetivo:** discutir as motivações e objetos dos Sistemas Distribuídos como padrões, discutindo suas vantagens e desvantagens quando comparados com os Sistemas Centralizados Tradicionais.



2.1 - Vantagens do Sist. Distribuídos sobre os Sist. Centralizados

- **econômico:** soluções de custo/performance efetivas estão frequentemente atreladas a um grande número de CPUs baratas agrupadas em um sistema;
- **velocidade:** sistemas distribuídos normalmente oferecem maior suporte computacional que *mainframes*, dada as considerações teóricas e de engenharia;
- **inerência na distribuição:** algumas aplicações requerem a separação espacial de máquinas, ou seja, são inerentemente distribuídas;
- **confiabilidade:** distribuindo a carga em várias máquinas, uma simples falha traria abaixo uma única máquina, deixando o restante do sistema intacto;
- **crescimento incremental:** suporte computacional pode ser adicionado em pequenos incrementos possibilitando expansão gradual.

2.2 - Vantagens do Sistemas Distribuídos sobre PCs independentes

- ★ Dado que microprocessadores constituem a melhor alternativa de custo/performance, porque as pessoas simplesmente não adquirem seus PCs e trabalham de forma independentemente ?
- ... somente por uma coisa, usuários precisam compartilhar dados!

... 2.2 - Vantagens do Sistemas Distribuídos sobre PCs independentes

- Vantagens dos Sistemas Distribuídos sobre PCs Isolados:
- **compartilhamento de dados:** absolutamente essencial para a grande maioria das aplicações, assim as máquinas devem ser interconectadas;
- **compartilhamento de dispositivos:** permitir usuários compartilharem periféricos caros tais como *plotters*, robôs, etc.
- **comunicação:** uma outra razão para conectar grupos isolados de computadores em um sistema distribuído é agregar valor na comunicação pessoa-pessoa.
- **flexibilidade:** por fim, sistemas distribuídos são potencialmente mais flexíveis que conceder a cada usuário um computador pessoal.

2.3 - Desvantagens do Sistemas Distribuídos

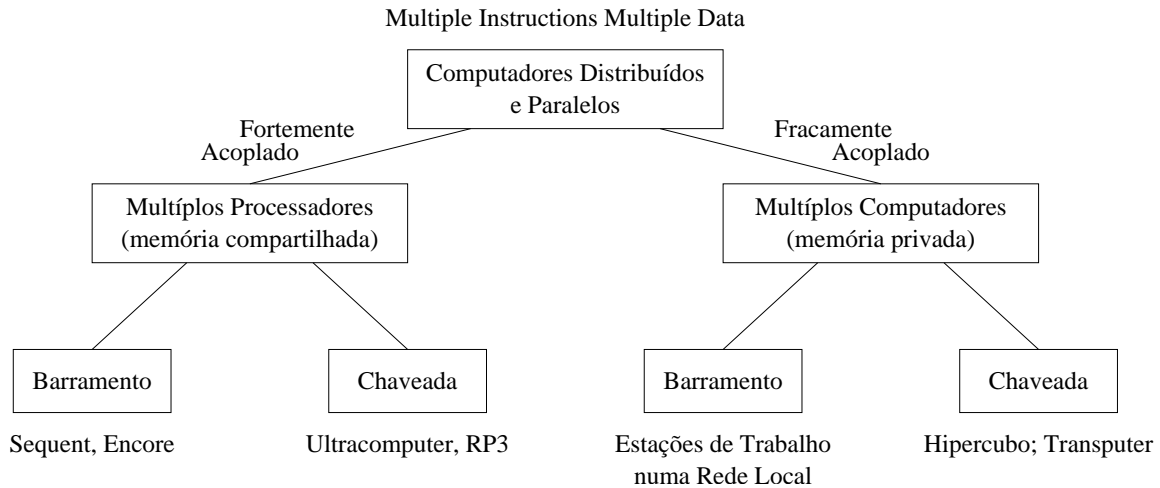
- ❶ **software:** que tipo de sistemas operacionais, linguagens de programação e aplicações são apropriadas para estes sistemas ?
 - ... *softwares* pequenos já constituem uma realidade.
- ❷ **networking:** o sistema de comunicação pode se saturar ou causar outros problemas, trazendo “por terra” todas as vantagens dos sistemas distribuídos decorrentes da comunicação.
- ❸ **segurança:** acesso facilitado propiciado pelos sistemas distribuídos também se aplica aos dados mantidos sob sigilo e/ou com restrição de acesso.
- ★ Apesar dos **problemas potenciais**, o sentimento é de que as vantagens se sobrepõem às desvantagens, ao mesmo tempo que se espera uma maior credibilidade dos sistemas distribuídos como resultado de melhoramentos.

3 - Conceitos de Hardware

- ★ Ainda que todos os Sistemas Distribuídos sejam constituídos de múltiplas CPUs, há diferentes maneiras do *hardware* ser organizado, especialmente no que se refere interconexão e comunicação.
 - ... vários esquemas de classificação tem sido propostos nos últimos anos, mas nenhum ainda é largamente aceito;
 - ... provavelmente o mais citado é o esquema proposto na *Taxonomia de Flynn* (1972), que embora rudimentar, tem por essência a classificação segundo o número de fluxos de intruções e de fluxos de dados.
 - ✗ *Single Instruction Single Data*
 - ✗ *Single Instruction Multiple Data*
 - ✗ *Multiple Instruction Single Data*
 - 👉 *Multiple Instruction Multiple Data* - **Sistemas Distribuídos**

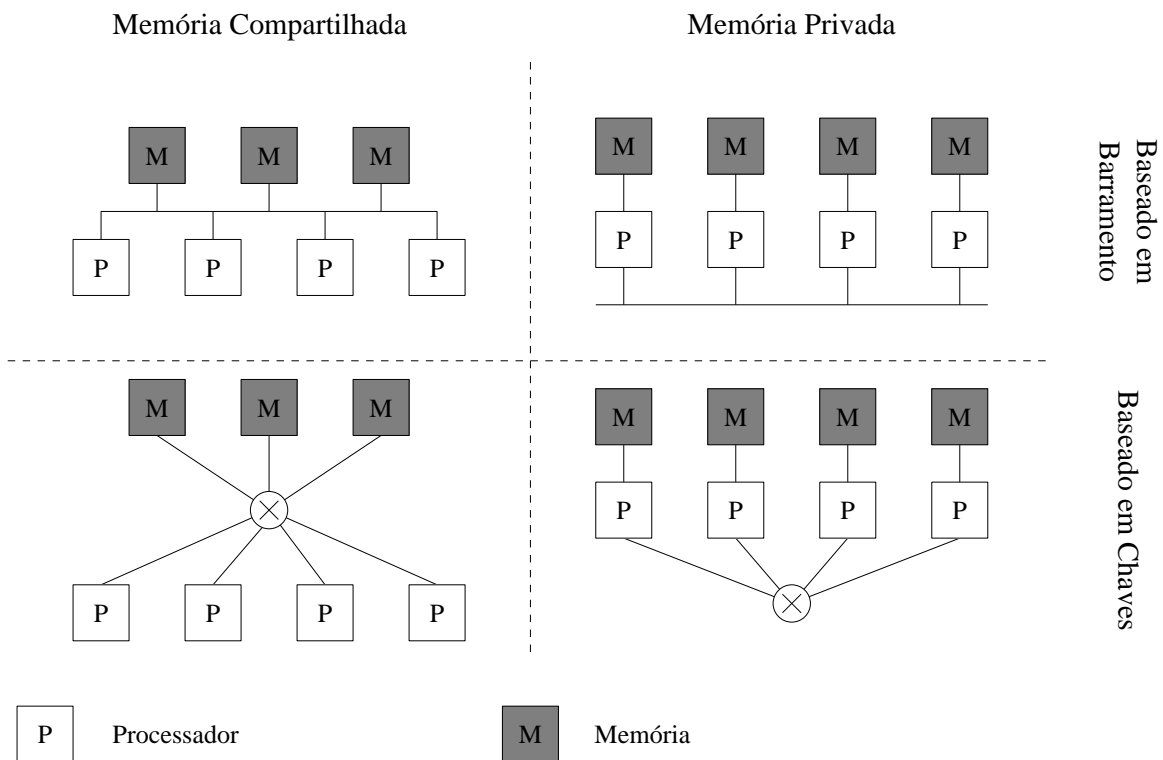
... 3 - Conceitos de Hardware

- Taxonomia de Sistemas Computacionais Distribuídos e Paralelos:



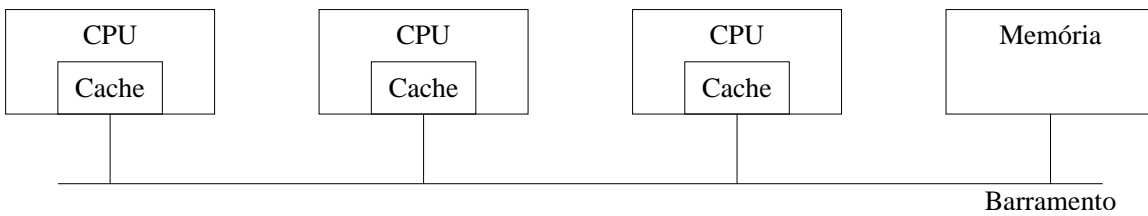
- ... a importância desse tópico reside no fato de poder verter informações adicionais, dado que diferentes máquinas requerem diferentes sist. oper.

... 3 - Conceitos de Hardware



3.1 - Múltiplos Processadores baseados em Barramento

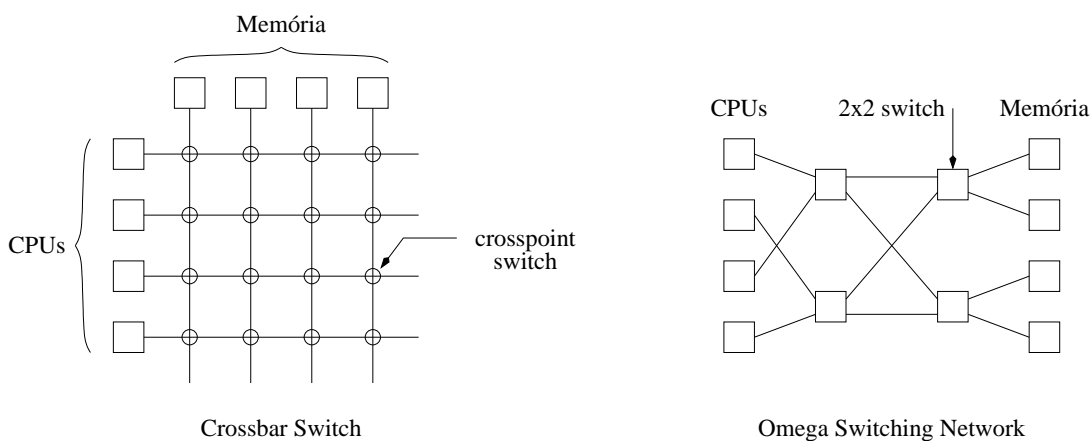
- ... consiste de CPUs interconectadas entre si e a um módulo de memória através de um barramento comum.



- ... tais sistemas pressupõem a **coerência** da memória, que por sua vez desempenha um importante papel nos sistemas operacionais distribuídos;
- ... a *cache* oferece performance quando o número de CPUs é grande, o que por consequência gera aumento de tráfego no barramento;
- ... por outro lado, a introdução da *cache* pressupõe mecanismos para manter a coerência na memória, *write-through* ou *snoopy cache*.

3.2 - Múltiplos Processadores Chaveados

- ... para multiprocessadores com mais de 64 processadores, diferentes métodos se fazem necessários para conectar as CPUs à memória;
- ... uma possibilidade é dividir a memória em módulos e conectá-los às CPUs através chaves (*crossbar switch* ou *n x n switch*).



... 3.2 - Múltiplos Processadores Chaveados

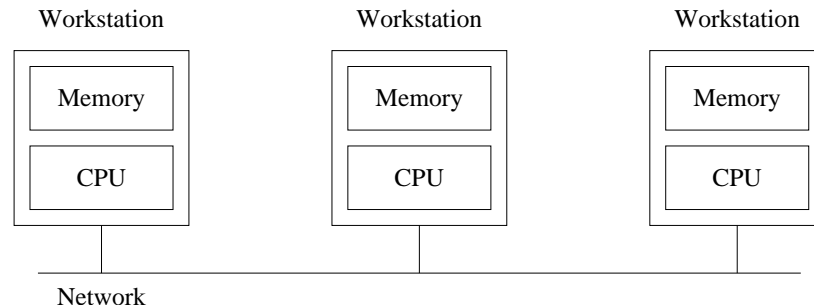
- ... um aspecto positivo da chave *crossbar* é que muitas CPUs podem acessar a memória simultaneamente, exceto no caso de acesso ao mesmo módulo;
- ... para n CPUs e n módulos de memória, n^2 chaves fazem-se necessárias;
- ... para valores grandes de n a rede *omega* constitui-se numa alternativa para redes chaveadas, dado que requer um menor número de chaves;
- ... no caso geral, com n CPUs e n memórias, a rede *omega* requer \log_2^n estágios, cada um contendo $n/2$ chaves;
- ... as chaves podem ser gatilhadas em *nanosegundos* ou menos, mas este atraso também é um grande problema;
- **Exemplo:** Considere uma rede *omega* onde $n = 1024$ com cada CPU executando a uma taxa de 100 MIPS. Qual é o custo de chaves ?

... 3.2 - Múltiplos Processadores Chaveados

- ... para o problema proposto, uma alternativa é associar algumas memória a cada uma das CPUs, ou seja, *Non Uniform Memory Access* ou NUMA;
- Como visto, Múltiplos Processadores baseados em Barramentos, mesmo usando *shoopy caches*, estão limitados a não mais que 64 CPUs;
- ... para conectar mais que 64 CPUs, redes chaveadas contornam o problema mas a um custo senão proibitivo, muito alto;
- ... por outro lado, a abordagem do acesso não uniforme da memória, exige algoritmos complexos para não se gerar perda de performance;
- ★ ... **construir grandes sistemas com múltiplos processadores com memória compartilhada e fortemente acoplados é difícil e caro!**

3.3 - Múltiplos Computadores baseados em Barramento

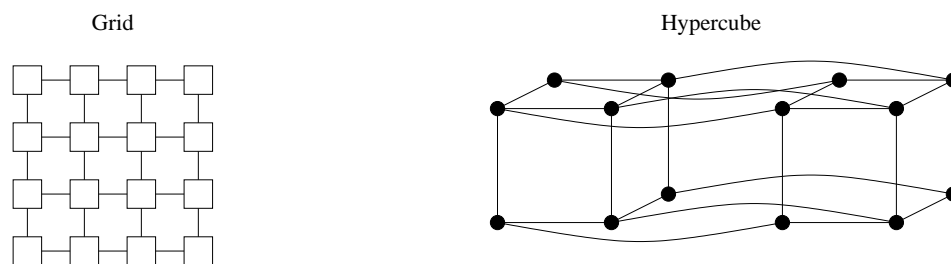
- Por outro lado, construir Múltiplos Computadores não exige memória compartilhada, ou seja, cada CPU tem a sua própria memória.
- ... resta então o problema, como as CPUs se comunicam, ou seja, espera-se um volume de tráfego muitas vezes menor que nos casos anteriores;



- **Exemplo:** ... uma rede LAN de 10-100 Mbps suportando múltiplos computadores e um barramento de 300 Mbps suportando múltiplos processadores.

3.4 - Múltiplos Computadores Chaveados

- Várias interconexões foram propostas, mas todas contemplando a propriedade de que cada CPU tenha acesso direto e exclusivo à sua própria memória.



- ... para problemas com natureza inerentemente bi-dimensional, tais como teoria dos grafos ou visão, *grids* são mais apropriados;
- ... já o **hipercubo** é um cubo n dimensional, ou seja, cada CPU tem n conexões para as outras CPUs (escala logarítmica).

... 3.4 - Múltiplos Computadores Chaveados

- ... no hipercubo, apenas os vizinhos são conectados, o que sugere roteamento da mensagem até o seu destino, ou seja, vários saltos;
- ... no hipercubo o maior caminho que uma mensagem percorre cresce logaritmicamente com o seu tamanho, enquanto que no *grid* o crescimento se dá com a raiz quadrada do número de CPUs;
- Hipercubos com 1024 CPUs já são comercialmente disponíveis há alguns anos, mas projetos com 16.384 CPUs já são disponíveis.

4 - Conceitos de Software

- ✧ **Aspecto Chave:** a forma como o sistema se apresenta aos usuários bem como a imagem que os usuários fazem do mesmo **é fortemente determinada pelo software e não pelo hardware** do sistema.

... embora *software*, diferentemente do *hardware*, tenha natureza imprecisa e amorfa, ainda assim é possível distinguir dois tipos de sistemas operacionais para múltiplas CPUs: fracamente e fortemente acoplados.

- ✧ Exemplo: considere um grupo de computadores pessoais dotados de processador, memória, disco e o seu próprio sist. oper. compartilhando recursos como impressora e base de dados através da rede.

... quais avaliações acerca do acoplamento no *software* podemos extrair deste exemplo ? ... quais os tipos de sistemas distribuídos ?

4.1 - Sistemas Operacionais de Rede

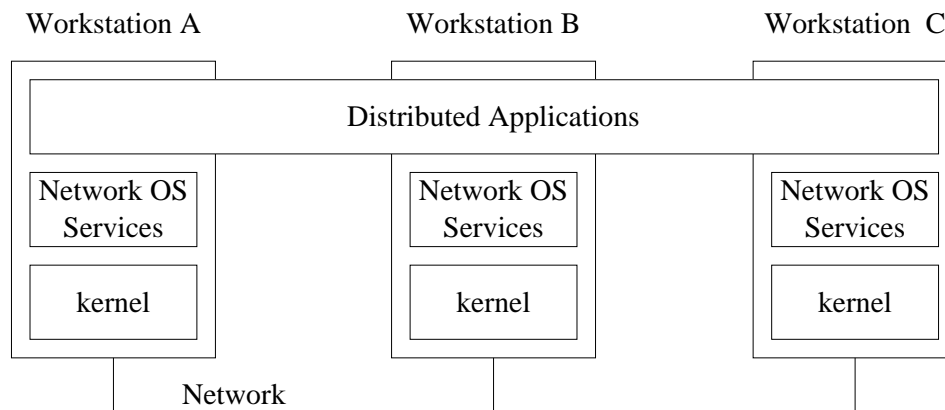
- ✧ Consideremos inicialmente os sistemas de *software* e *hardware* fracamente acoplados, dada a maior probabilidade de ocorrência nas organizações.

... para tanto, considere algumas estações de trabalho conectadas a uma rede local, cada qual com o seu próprio sistema operacional, mas sem necessariamente acomodar disco rígido;

... muito provavelmente o sistema operacional utilizado neste tipo de ambiente deverá gerenciar as estações e os servidores de arquivo individualmente bem como a comunicação entre as mesmas.

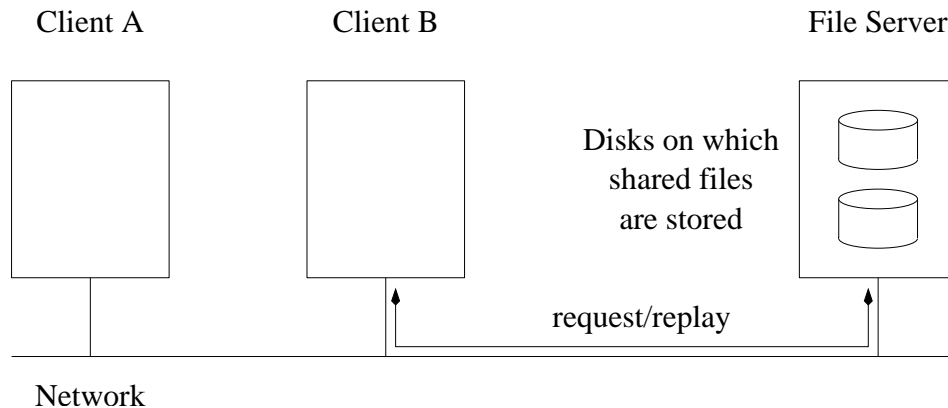
... 4.1 - Sistemas Operacionais de Rede

- ✧ Neste cenário onde máquinas tem um alto grau de autonomia e onde não encontramos poucos requisitos por parte do sistema, usualmente o denominamos de **Sistema Operacional de Rede**.



... 4.1 - Sistemas Operacionais de Rede

- ... diferentes clientes podem montar o servidor de arquivos em diferentes pontos do sistema de arquivo local.



4.2 - Sistemas Distribuídos Verdadeiros

- ... no sistema operacional de rede, a menos do sistema de arquivo compartilhado, é aparente aos usuários que o sistema é constituído de vários computadores interligados pela rede local;
 - ... na seqüência, iremos tratar dos sistemas fortemente acoplados no *software* e fracamente acoplados no *hardware* (e.g., múltiplos computadores);
 - ... o objetivo por trás desses sistemas é criar a ilusão para os usuários que toda a rede de computadores constitui um único sistema *timesharing*;
 - Alguns autores referem-se a esta propriedade como *Single System Image*, outros com uma pequena diferença a denominam de *Virtual Uniprocessor*.
- 👉 **Idéia:** os usuários não estão cientes da existência de múltiplas CPUs no sistema, daí a propriedade de *Single System Image*.

... 4.2 - Sistemas Distribuídos Verdadeiros

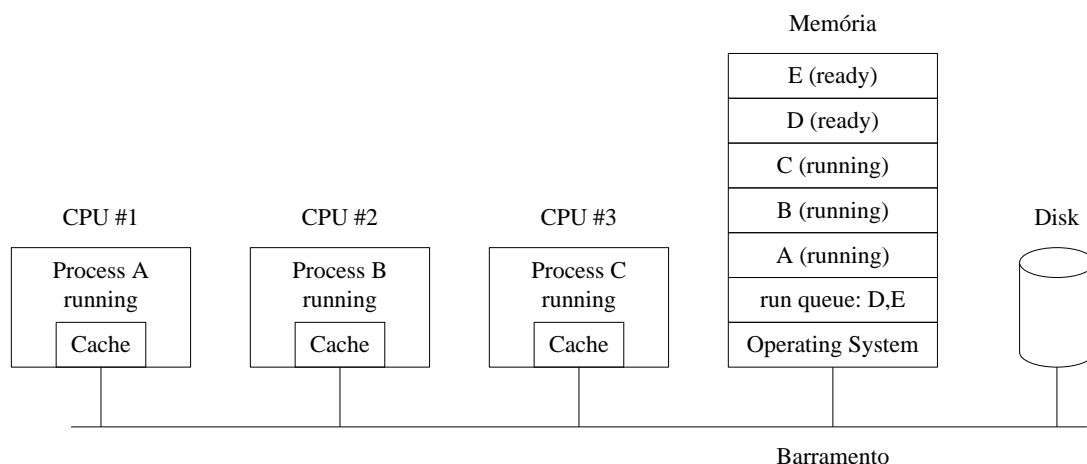
✧ Vejamos então algumas características dos Sistemas Distribuídos:

- ... **mecanismo de comunicação entre processos:** deve ser único e global, ou seja, qualquer processo pode se comunicar com qualquer outro processo;
- ... **esquema de proteção global:** bits de proteção do UNIX, junção de listas de controle de acesso não garante *single system image*;
- ... **gerenciamento de processos:** como processos são criados, destruídos, iniciados e finalizados não pode variar de máquina para máquina;
- ... **sistema de arquivos:** deve ser visível em qualquer máquina do sistema, naturalmente que sob os mesmos mecanismos de proteção e segurança;
- ... **chamadas de sistema:** deve apresentar a mesma interface, como consequência, *kernels* idênticos serão executados em todas as CPUs do sistema;

4.3 - Sistemas Timesharing Multiprocessados

✧ ... a última combinação que desejamos discutir é sistemas com *software* fortemente acoplados e *hardware* fortemente acoplados.

- ... a característica chave desta classe de sistemas é a existência de uma fila única, ou seja, a lista de todos os processos do sistema que estão desbloqueados e prontos para serem executados é única.



... 4.3 - Sistemas Timesharing Multiprocessados

- ... um aspecto no qual múltiplos processadores diferem apreciavelmente dos sistemas distribuídos e de rede é a organização dos sistema de arquivos;
 - ... quando algum processo executa uma chamada de sistema, uma interrupção é gerada e mantida pelo sistema operacional utilizando semáforos, monitores ou algo equivalente, de modo a bloquear a CPU enquanto a seção crítica é tratada ou tabelas gerais são acessadas;
 - ... processo semelhante se dá com a *cache* global.
- ✧ Deve ficar claro que os métodos utilizados em Múltiplos Processadores para dar a idéia de sistema único não são aplicáveis a máquinas que não tem memória compartilhada.

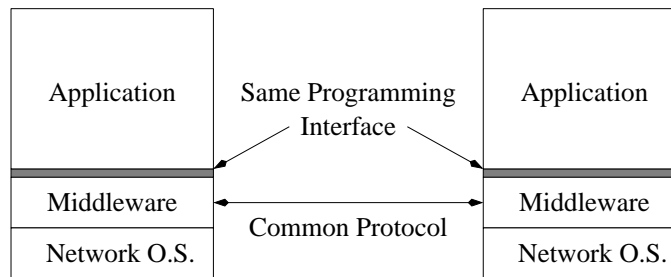
... 4.3 - Sistemas Timesharing Multiprocessados

- ✧ Vejamos uma breve comparação entre os sistemas anteriormente discutidos:

Item	Network Operating System	Distributed Operating System	Multiprocessor Operating System
Does it look like a virtual uniprocessor?	No	Yes	Yes
Do all have to run the same operating system?	No	Yes	Yes
How many copies of operating system are there?	N	N	1
How is communication achieved?	shared files	messages	shared memory
Are agreed upon network protocols required?	Yes	Yes	No
Is there a single run queue?	No	No	Yes
Does file sharing have well-defined semantics?	Usually No	Yes	Yes

... 4.3 - Sistemas Timesharing Multiprocessados

- ... *open middleware-based distributed system* requer que os protocolos na camada de *middleware* sejam os mesmos, assim como suas interfaces.



Item	Distributed O.S.		Network O.S.	Middleware based O.S.
	Multiprocessors	Multicomputers		
degree of transparency	very high	high	low	high
same O.S. on all nodes	yes	yes	no	no
numbers of copies of O.S.	1	N	N	N
basis for communication	shared memory	messages	files	specific model
resource management	global, central	global, distributed	per node	per node
scalability	no	moderately	yes	varies
openness	closed	closed	open	open

5 - Aspectos de Projeto dos Sistemas Distribuídos

- ✧ No restante deste tópico, iremos descrever alguns dos aspectos chaves de projeto que devem ser contemplados quando do projeto de Sistemas Distribuídos.

5.1 - Transparência em Sistemas Distribuídos

- ✧ Provavelmente, o aspecto mais importante seja obter uma imagem única do sistema (*single system image*), ou seja, apresentar-se a seus usuários como um sistema *timesharing* simples e tradicional.
 - ... o sistema que atende este objetivo é dito ser **transparente**.
- ✧ Por outro lado, o conceito de Transparência envolve mais detalhes e sutilezas que os apresentados a primeira vista, com por exemplo no cenário abaixo:
 - ... considere um sistema distribuído consistindo de estações de trabalho, cada qual executando um sistema operacional padrão;
 - ... como pode ser tratada a transparência de serviços do sistema, com por exemplo a leitura de arquivos?

... 5.1 - Transparência em Sistemas Distribuídos

- ✧ Este conceito se aplica a vários aspectos dos Sistemas Distribuídos:
 - **localização:** usuários não conseguem informar onde recursos de *hardware* e *software*, tais como CPUs, impressoras, arquivos estão localizados;
 - **migração:** recursos estão livres para moverem de um lugar para outro (e.g., nó do sistema) sem a necessidade de mudar seus nomes;
 - **replicação:** o sistema operacional é livre para efetuar cópias adicionais de arquivos e outros recursos sem que os usuários percebam;
 - **concorrência:** múltiplos usuários podem compartilhar recursos automaticamente, ou seja, eles não se dão conta da existência de outros usuários;
 - **paralelismo:** atividades em paralelo podem ocorrer, sem que para isso os usuários tomem conhecimento de que o sistema como um todo tirou vantagem do potencial paralelismo do sistema.

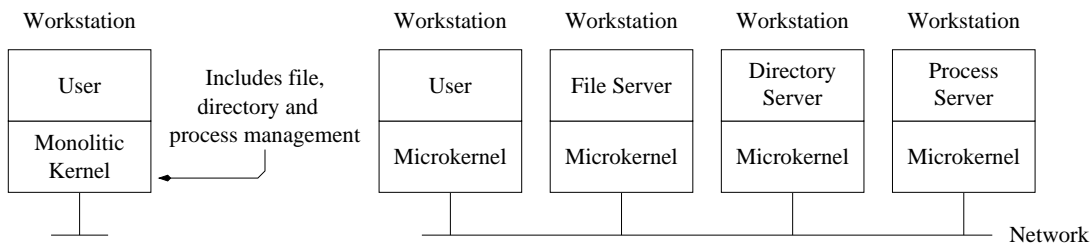
5.2 - Flexibilidade em Sistemas Distribuídos

- ✧ A importância do termo **flexibilidade** reside no fato de que decisões de projeto que possam parecer razoáveis, podem mais tarde estarem erradas.

Design decisions that now seem reasonable may later prove to be wrong. The best way to avoid problems is thus to keep one's options open.

Andrew S. Tanenbaum

- ✧ Considere duas abordagens acerca da estrutura dos Sistemas Distribuídos.



... 5.2 - Flexibilidade em Sistemas Distribuídos

- ... o *kernel* monolítico é a base dos sistemas operacionais centralizados atuais, com o acréscimo da funcionalidade de rede e integração de serviços remotos;
- ... muitos sistemas distribuídos, que são extensões do UNIX, utilizam-se desta abordagem, dado que o próprio UNIX tem por base um kernel monolítico.

If the Monolithic Kernel is the reigning champion, the Microkernel is the up-and-coming challenger.

Andrew S. Tanenbaum

- ✧ **Idéia:** manter no *kernel* a configuração mínima de serviços, dentre eles:
 - mecanismo de comunicação entre processos;
 - algum gerenciamento de memória;
 - gerenciamento e escalonamento de processos de baixo nível;
 - entrada e saída de baixo nível.

- ✧ **Kernel Monolítico:** vantagem potencial de performance;
- ... interromper o *kernel* e fazer o que tem que ser feito naquele lugar é mais rápido que trocar mensagens com servidores remotos.
- ✧ Não obstante, na prática outros fatores tendem a dominar e o tempo requerido no envio/recepção de mensagens é usualmente negligenciado.

As a consequence, it is likely the microkernel systems will gradually come to dominate the distributed systems scheme, and monolithic kernels will eventually vanish or evolve into microkernels.

Andrew S. Tanenbaum

5.3 - Confiabilidade em Sistemas Distribuídos

- ✧ Um aspecto original do desenvolvimento de Sistemas Distribuídos é a necessidade de torná-los mais confiáveis que Sistemas de Processador Único.
- ... nesse sentido, considere a confiabilidade como a função **OR** da confiabilidade dos seus componentes, então, para um sistema onde cada componente apresenta 0,95 de estar funcionando corretamente, qual é a probabilidade de ao menos um estar funcionando?
- ... no mesmo cenário, considere agora que a confiabilidade seja a função **AND** da confiabilidade dos seus componentes, então, qual é a probabilidade de ao menos um estar funcionando?

Distributed System is one on which I cannot get any work done because some machine I have never heard of has crashed.

Leslie Lamport

✧ Aspectos que devem ser distinguidos no conceito de confiabilidade:

- **availability:** refere-se à fração de tempo em que o sistema possa ser utilizado, ou seja, está relacionado com a disponibilidade dos componentes críticos do mesmo bem como a redundância de *hardware* e *software*;
- **security:** arquivos e recursos devem estar protegidos contra o uso não autorizado, com a ressalva de que aqui este problema é mais severo;
- **fault tolerance:** em geral, sistemas distribuídos podem ser projetados para mascarar as falhas, ou seja, escondê-las dos usuários.

5.4 - Performance em Sistemas Distribuídos

- ✧ **Performance:** conjunto de métricas, tais como: tempo de resposta, vazão de tarefas executadas no tempo, utilização do sistema, consumo de recursos de rede que ao final informam quão rápido ou vagaroso é o sistema.
- ... o problema da *performance* é fundamentada no fato da comunicação, que é essencial nos sistemas distribuídos, estar atrás do suporte computacional das máquinas que compõem o sistema;
 - ... assim, para otimizar a performance, faz-se necessário minimizar o número de mensagens, o que de fato tem outras implicações;
 - ... por exemplo, executar em paralelo atividades em diferentes CPUs, contribui para aumento da performance, mas também exige envio de muitas mensagens.

One possible way out is to pay considerable attention to the grain size of all computations.

Andrew S. Tanenbaum

- **fine-grained parallelism:** tarefas que envolvem uma grande quantidade de cálculos pequenos, com muitas interações entre as tarefas.
- **coarse-grained parallelism:** tarefas que envolvem uma grande quantidade de cálculos grandes, com poucas interações entre as tarefas.

5.5 - Escalabilidade em Sistemas Distribuídos

- ✧ **escalabilidade:** propriedade relacionada aos aspectos de projeto que contemplem diretamente ou indiretamente a capacidade de atendimento às requisições de usuários e *jobs* do sistema como um sistema único.
- ... embora pouco se saiba acerca dos grandes sistemas distribuídos, o princípio que os guia é: evite componentes, tabelas e algoritmos centralizados;
- ✧ **componentes centralizados:** considere um servidor de mail para 50.000.000 de usuários com suporte computacional suficiente, mesmo assim, a capacidade da rede para o tráfego em questão constitui um problema sério;
- ... quão robusto é o sistema na tolerância a falhas?
- ... mesmo considerando que os mails são locais, o envio de mensagem de um usuário a um outro, exige a intermediação do servidor.

5.5 - Escalabilidade em Sistemas Distribuídos

✧ **dados centralizados** também constituem um idéia ruim:

- ... por exemplo, como alguém deveria manter os dados referentes aos números de telefones e endereços de 50.000.000 de pessoas?
- ... assuma que cada registro seja armazenado em 50 caracteres;
- ... onde está o problema, no tamanho do espaço em disco (2,5 GB), na saturação das linhas de comunicação (10-100 Mbps) ou no uso inadequado das linhas de comunicação?

5.5 - Escalabilidade em Sistemas Distribuídos

✧ **algoritmos centralizados** exigem que um grande número de mensagens sejam trocadas no sistema de comunicação, exigindo aumento de carga no roteamento por questões de performance.

✧ **algoritmos descentralizados** são caracterizados por:

- nenhuma máquina tem informação completa do estado do sistema;
- máquinas tomam decisão com base nas informações locais;
- falhas em uma máquina não aruam o algoritmo;
- nenhuma suposição implícita de que exista relógio global.

Algorithms should take into account the lack of exact clock synchronization. The larger the system, the larger the uncertainty.

Andrew S. Tanenbaum