



Introdução Programação

Anthony Ferreira La Marca

anthony@computacao.cua.ufmt.br




Programação Concorrente

- Um **programa concorrente** é um **conjunto de programas sequenciais** os quais são **executados em uma *abstração* de paralelismo**.



Exemplos de Concorrência

- Sobreposição de I/O e Processamento
(Overlapped I/O and Computation)
- Multiprogramação (Multi-programming)
- Multi-tarefação (Multi-Tasking)
- MultiThread




Sobreposição de I/O e Computação

- No início dos tempos dos primeiros SOs, controlar I/O não podia ser feito concorrentemente com outra computação **sobre um único processador**.
- Mas a evolução do SOs, fez surgir a **concorrência**, retirando da computação principal, alguns microsegundos necessários para controlar I/O



Multiprogramação

- É a execução concorrente de diversos **processos** independentes sobre um único processador.
- Desta forma, dá a impressão da execução de vários programas ao mesmo tempo



Time-Slicing

- Para dar a oportunidade para todos os processos serem executados
- Compartilhar o processador entre diversas computações de processos.
- Um programa do SO chamado **Escalonador** é executado para determinar qual processo deve ser permitido executar no próximo intervalo.



Multi-Tarefa

- Semelhante a Multi-programação
- Aumento na comutação de processos



Multi-Thread

- Estende o conceito de multiprogramação
- Decompõe um problema (processo) em subproblemas (threads)
- Feito entre todos os processos concorrentes



Problemas

- Devido a diversas **interações entre os processos** que compreendem um programa concorrente é difícil escrever um programa concorrente correto.
- Para interagirem, processos precisam se **sincronizar** e se **comunicar** diretamente ou não.



Programação

- Semáforos
- Monitores
- Looks
- Threads



Programação Paralela

- ▶ “É uma forma de computação em que vários cálculos são realizados simultaneamente, operando sob o princípio de que um grande problema, geralmente, pode ser dividido em problemas menores, que então são resolvidos em paralelo”.
- ▶ Tornou-se possível, a partir de máquinas de arquitetura paralela / sistemas operacionais distribuídos



Diferenças



➤ Acoplamento

- Sistemas paralelos são **fortemente acoplados**:
- Sistemas distribuídos são **fracamente acoplados**:



Diferenças

- Previsibilidade

- O comportamento de sistemas paralelos é mais previsível.
- Já os sistemas distribuídos são mais imprevisíveis



Diferenças

- Influência do Tempo
 - Sistemas distribuídos são bastante influenciados pelo tempo de comunicação pela rede;
 - Em sistemas distribuídos, o tempo de troca de mensagens pode ser desconsiderado.



Diferenças



Controle

- Em geral em sistemas paralelos se tem o controle de todos os recursos computacionais;
- Já os sistemas distribuídos tendem a empregar também recursos de terceiros.



Vantagens

- Usam melhor o poder de processamento
- Apresentam um melhor desempenho
- Permitem compartilhar dados e recursos
- Podem apresentar maior confiabilidade
- Permitem reutilizar serviços já disponíveis
- Atendem um maior número de usuários
- Escalabilidade



Dificuldades

- Desenvolver, gerenciar e manter o sistema
- Controlar o acesso concorrente a dados e a recursos compartilhados.
- Evitar que falhas de máquinas ou da rede comprometam o funcionamento do sistema.
- Garantir a segurança do sistema e o sigilo dos dados trocados entre máquinas
- Lidar com a heterogeneidade do ambiente.

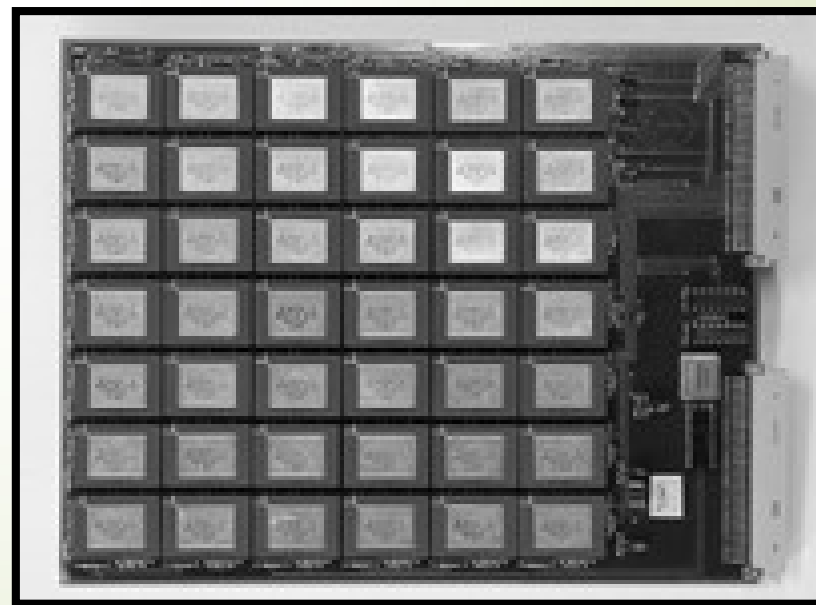
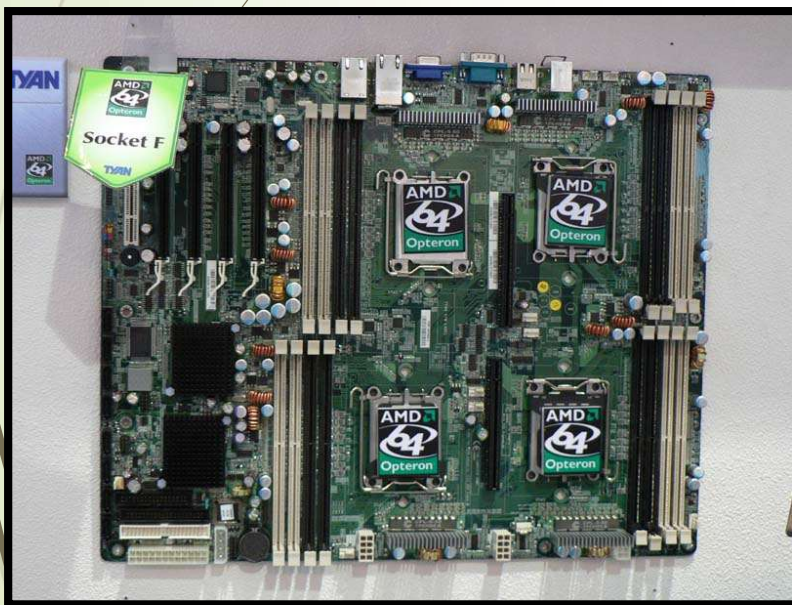
Plataformas de Execução

- Um S.O. multitarefa permite simular o paralelismo em um único processador, alternando a execução de processos.
- Um processador com núcleo múltiplo permite paralelismo real entre processos, executando múltiplas instruções por ciclo.
- Início com os processadores
- Core 2 duo e core 2 quad



Plataformas de execução

- Uma Placa-Mãe Multiprocessador permite que cada processador execute um processo.





Plataformas de execução

- **Cluster** é o nome dado a um sistema montado com mais de um computador, cujo objetivo é fazer com que todo o processamento da aplicação seja distribuído aos computadores
- Com isso, é possível realizar processamentos que até então somente computadores de alta performance seriam capazes de fazer.

Plataformas de execução

- Um **cluster** é uma solução de baixo custo para processamento paralelo de alto desempenho.





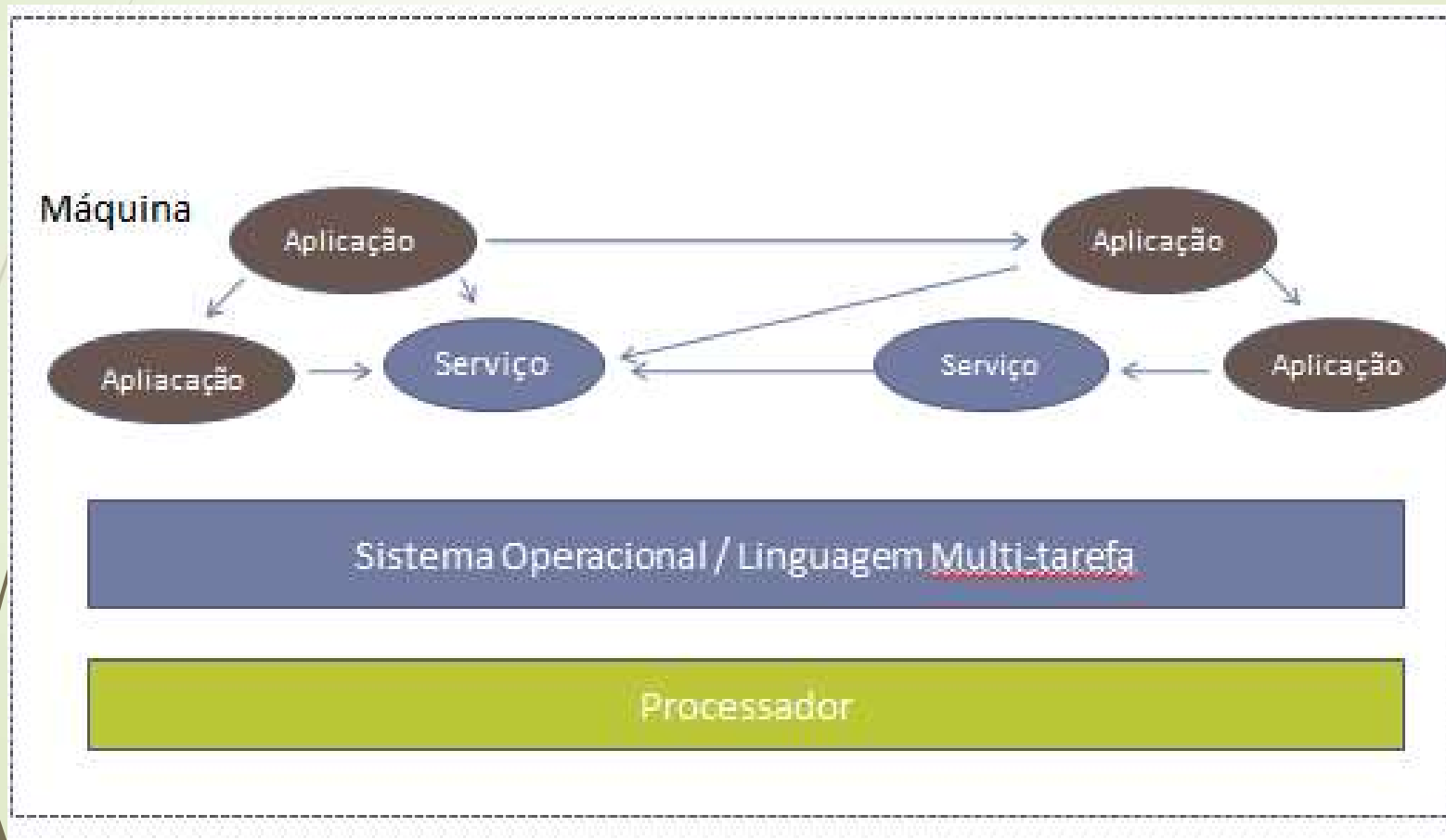
Suporte Computacional

- Suportes para Computação Paralela e Distribuída devem fornecer:
 - Mecanismos para execução paralela ou distribuída de programas.
 - Mecanismos para controle de concorrência.
 - Mecanismos para comunicação/sincronização entre processos / *threads* em paralelo / distribuídos
 - Ferramentas e mecanismos para desenvolvimento, testes, gerenciamento, controle dos programas
 - Mecanismos para segurança e tolerância a falhas, ...

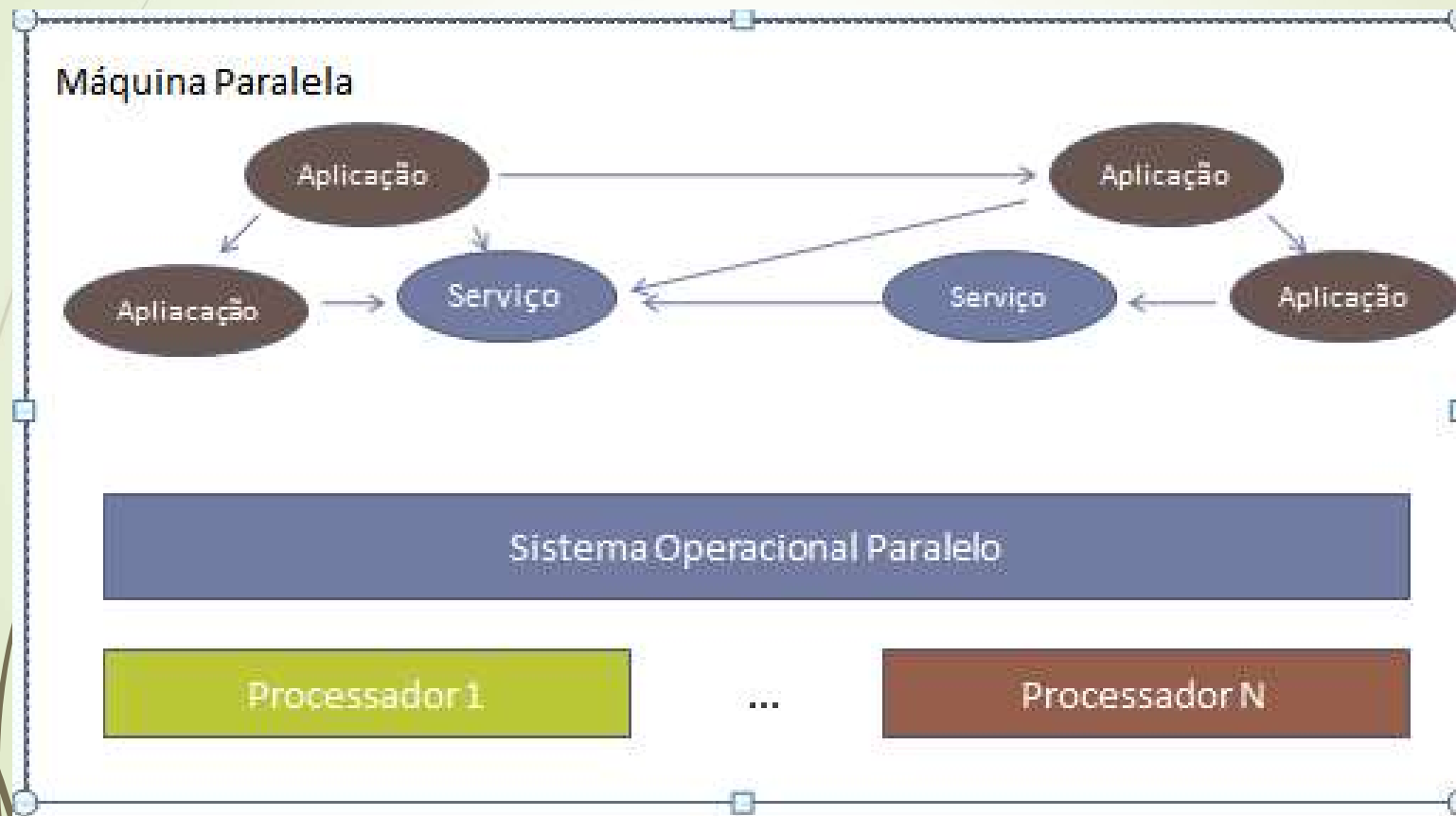
Suporte Computacional para Programação Paralela

- ▶ Sistemas Operacionais Multi-Tarefa: permitem a troca de contexto entre processos / threads.
 - ▶ Ex.: Windows, Linux, Solaris, HP-UX, AIX, etc.
- ▶ Linguagens Multi-Tarefa: permitem escrever programas pseudos-paralelos ou paralelos, usando um único processador.
 - ▶ Ex.: Java, C++
- ▶ Sistemas Operacionais Paralelos: permitem usar vários processadores em uma máquina.
 - ▶ Ex.: Linux, Solaris, Windows, etc.
- ▶ Suportes para Programação Paralela permitem criar máquinas virtuais paralelas.
 - ▶ Ex.: PVM
- ▶ Ferramentas e mecanismos
 - ▶ Threads, OpenMP e MPI

Suporte Computacional para SO/ Linguagens Multi-Tarefas



Sistemas Computacionais para SOs Paralelos

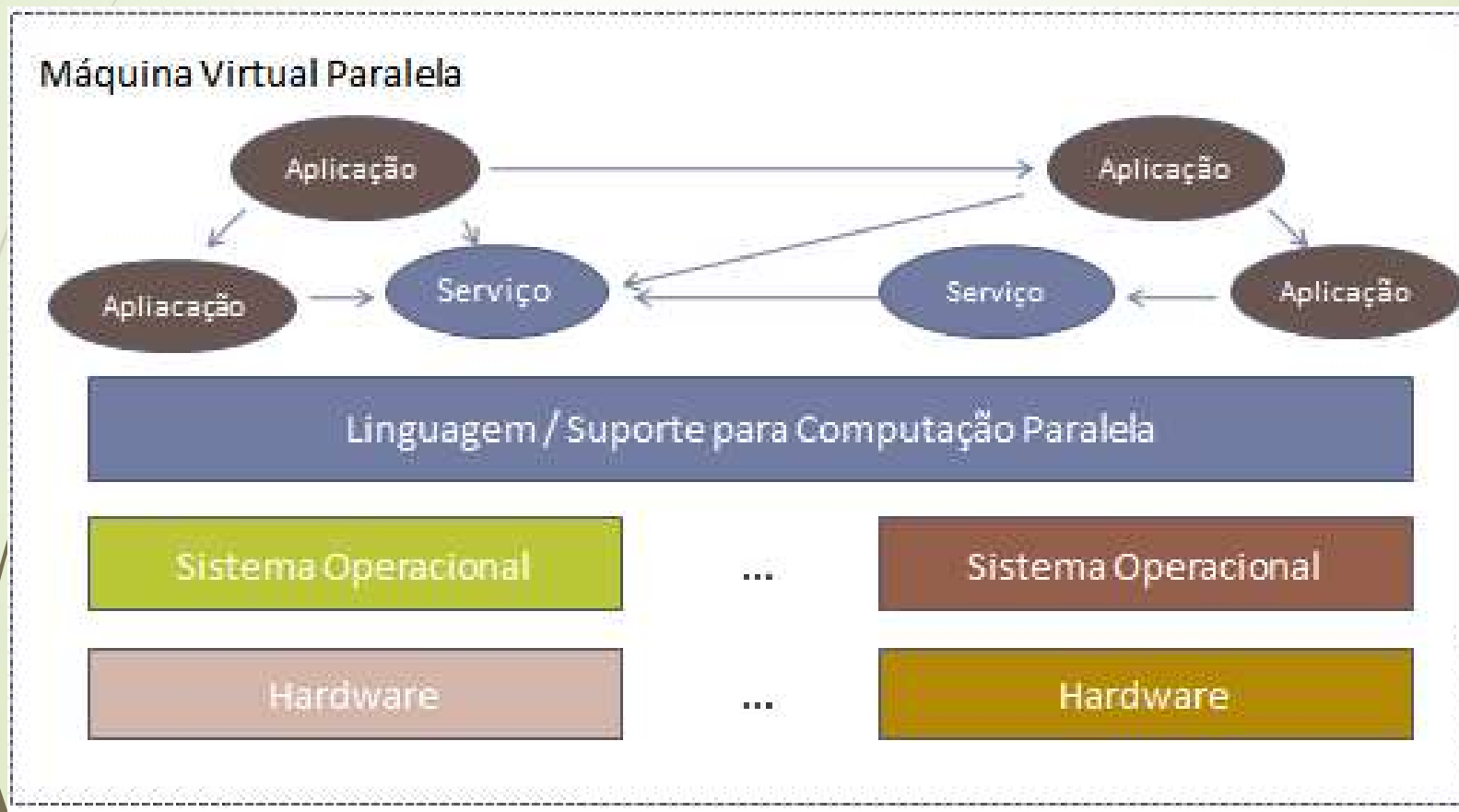




PVM

- *Parallel Virtual Machine*
- Lida transparentemente com o roteamento de mensagens, conversão de dados e o escalonamento de processos em uma rede formada por arquiteturas antes incompatíveis.
- Segue o paradigma da programação de aplicações com multiprocessadores e memória distribuída
- Possui bibliotecas para C e Fortran

Sistemas Computacionais para Computação paralela





Computação com Objetos Distribuídos



- Cliente Servidor

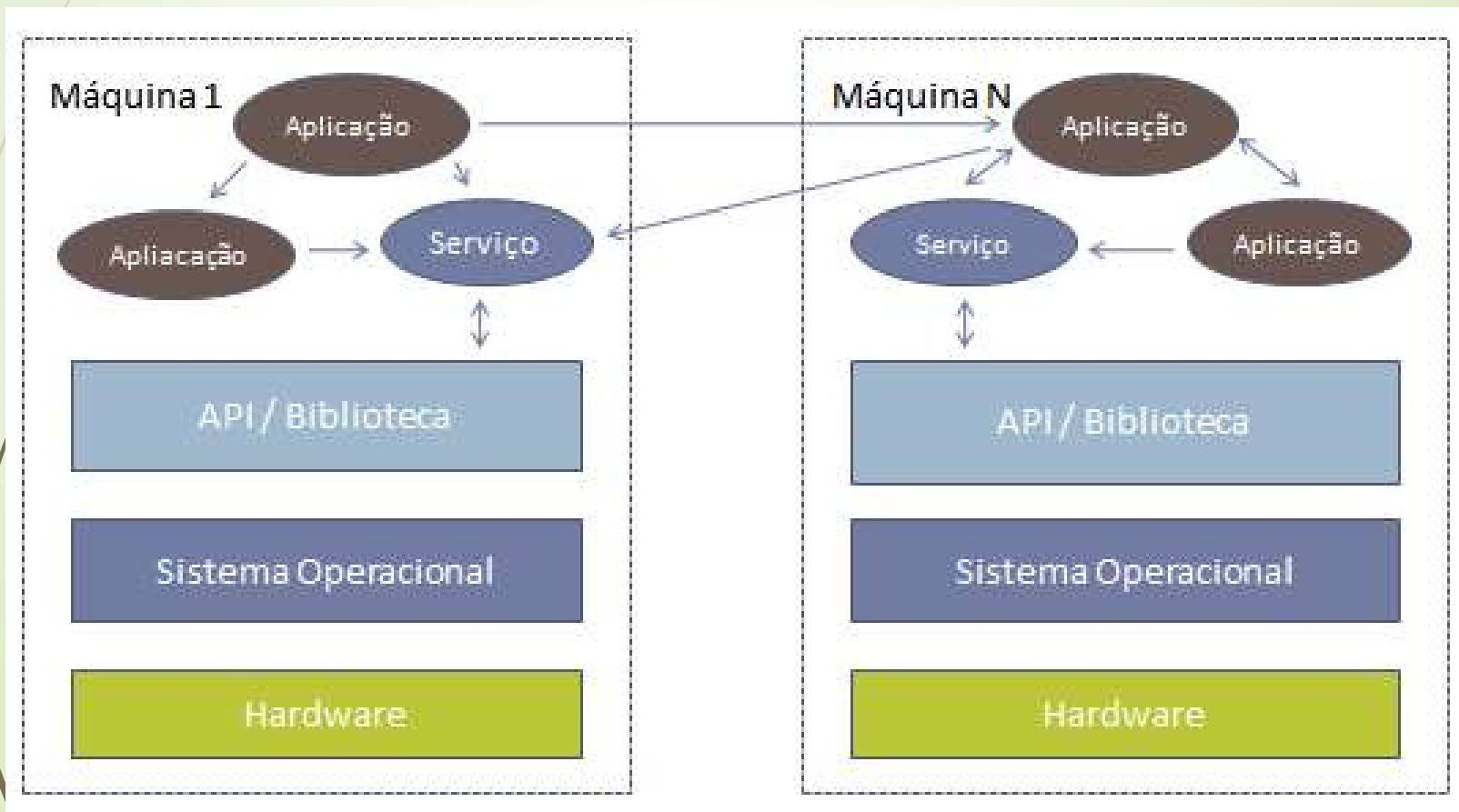
- P2P



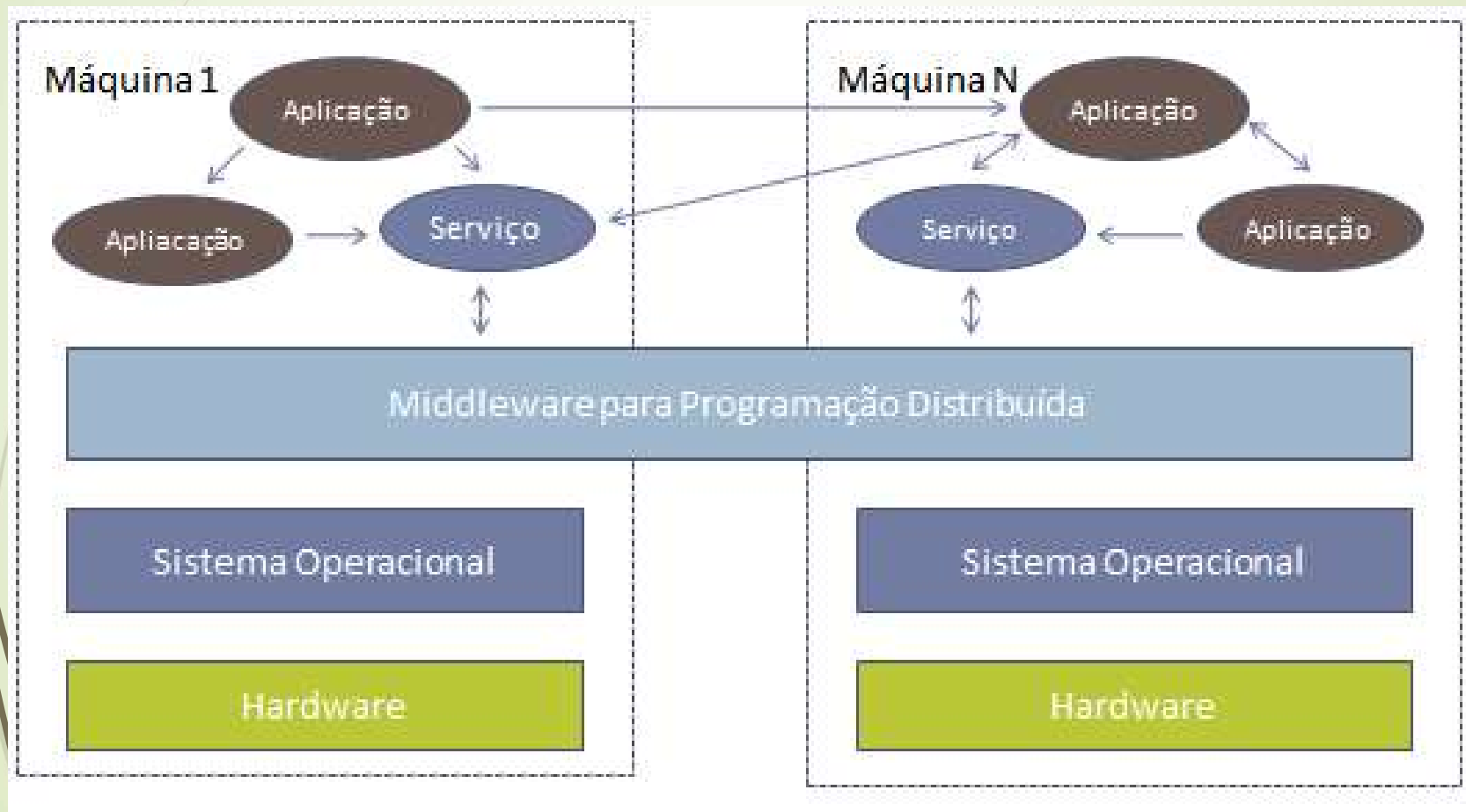
Suporte Computacional para Programação Distribuída

- ▶ Suporte para Computação Distribuída
 - ▶ **APIs e Bibliotecas:**
Ex.: UNIX Sockets, WinSock, java.net(Sockets),,RPC
 - ▶ **Middleware para Programação Distribuída:**
Ex: Java RMI, CORBA RMI.
 - ▶ **Servidores de Aplicação:**
Ex.: Tomcat, JBoss,
- ▶ Sistemas operacionais distribuídos caíram em desuso por não suportarem heterogeneidade de ambiente.

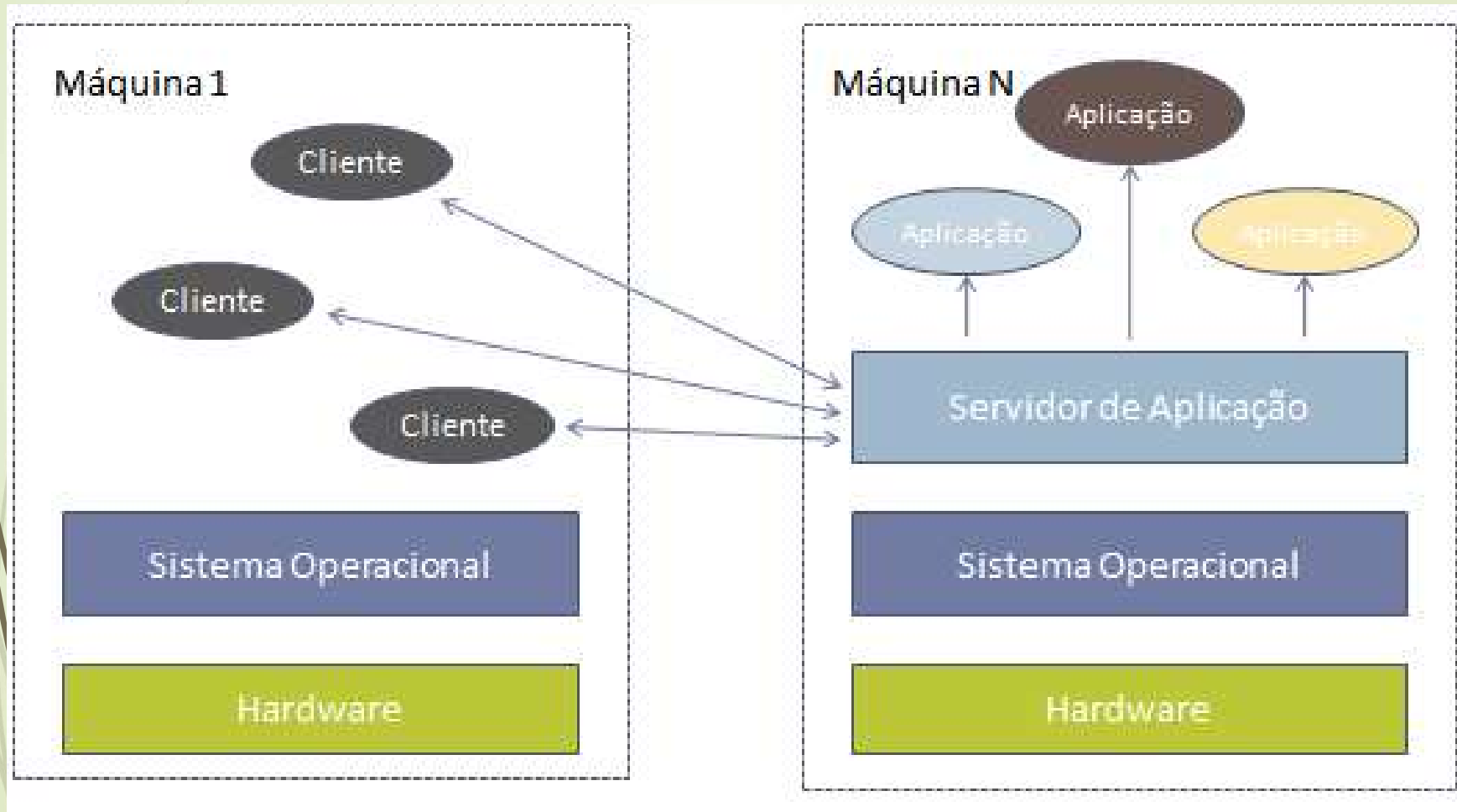
API / Biblioteca - RPC (Remote Procedure Call)



Middleware para Programação Distribuída



Servidor de Aplicação






Computação com Web Services

- Modelo Cliente-Servidor
- Mas, com servidor que visualiza serviços em outros servidores, usando uma linguagem única de comunicação entre as partes (XML).
- Permite a comunicação entre aplicações diferentes estando em computadores diferentes



Computação em Grid

- Utiliza recursos de diferentes computadores com o intuito de resolver problemas de grande complexidade e/ou volume
- 



Computação em Grid - Analogia

- Imagine a ida a um quiosque com os amigos
- Para não sobrecarregar ninguém, cada fica responsável por levar uma parte de todas as coisas
- Um leva as bebidas, outro a carne, outro a churrasqueira, outra leva os acompanhamentos e o outro mini-geladeira
- Agora, quando chegarem ao destino, dividem as tarefas para o preparo dos alimentos
- Por fim, todos usufruem de todo o alimento produzido a partir do que cada um trouxe
- Computação em grid funciona assim



Computação em Grid - Analogia

- Imagine 10 computadores Cada computador disponibiliza os seus recursos tais como CPU (Processador), Memória RAM, HD para o armazenamento. Pronto!
- Agora você precisa de um Sistema de Computação em Grid, que vai permitir ao usuário, através de um único computador, acessar todos os recursos disponíveis em todos os outros computadores. Assim você cria um SUPER COMPUTADOR! Legal!



Computação em Nuvem

- O conceito de **computação em nuvem** (em inglês, **cloud computing**) refere-se à utilização da memória e das capacidades de armazenamento e cálculo de computadores e servidores compartilhados e interligados por meio da Internet, seguindo o princípio da computação em grid.
- Você acessa recursos(memória, armazenamento, processamento) e realiza tarefas em servidores disponibilizados por terceiros, acessados via internet
- Sua máquina local fica praticamente responsável pela Entrada/Saída



Computação em Nuvem - Analogia

- Imagine a mesma situação anterior. Você vai viajar, só que ninguém levou nada! Após chegarem lá você precisa de comida! Óbvio! Só que neste caso vocês foram avisados que lá onde vocês iam, existe um galpão no meio do mato e lá tem tudo que vocês precisam.
- Ninguém sabe onde fica fisicamente este galpão e nem como ele é abastecido, mas isso não é tão importante. Basta ligar para um número que foi disponibilizado para vocês, passando o número do teu quarto e o que você deseja, que tudo será entregue na tua mesa. Legal hein!



Computação em Nuvem

- Bom, na prática, para utilizar o conceito de Computação em Nuvem, você ainda precisa seguir o princípio de computação em Grid, pois você continua utilizando recursos disponibilizados por outros.
- Só que desta vez os recursos que os outros disponibilizam não está na 'terra' onde você se encontra, mas está nas 'nuvens' onde está a INTERNET! Assim você precisa apenas de dispositivos de entrada e saída pois os recursos de armazenamento, memória e processamento estarão todos disponíveis na 'nuvem'. O acesso é realizado de forma remota.

