

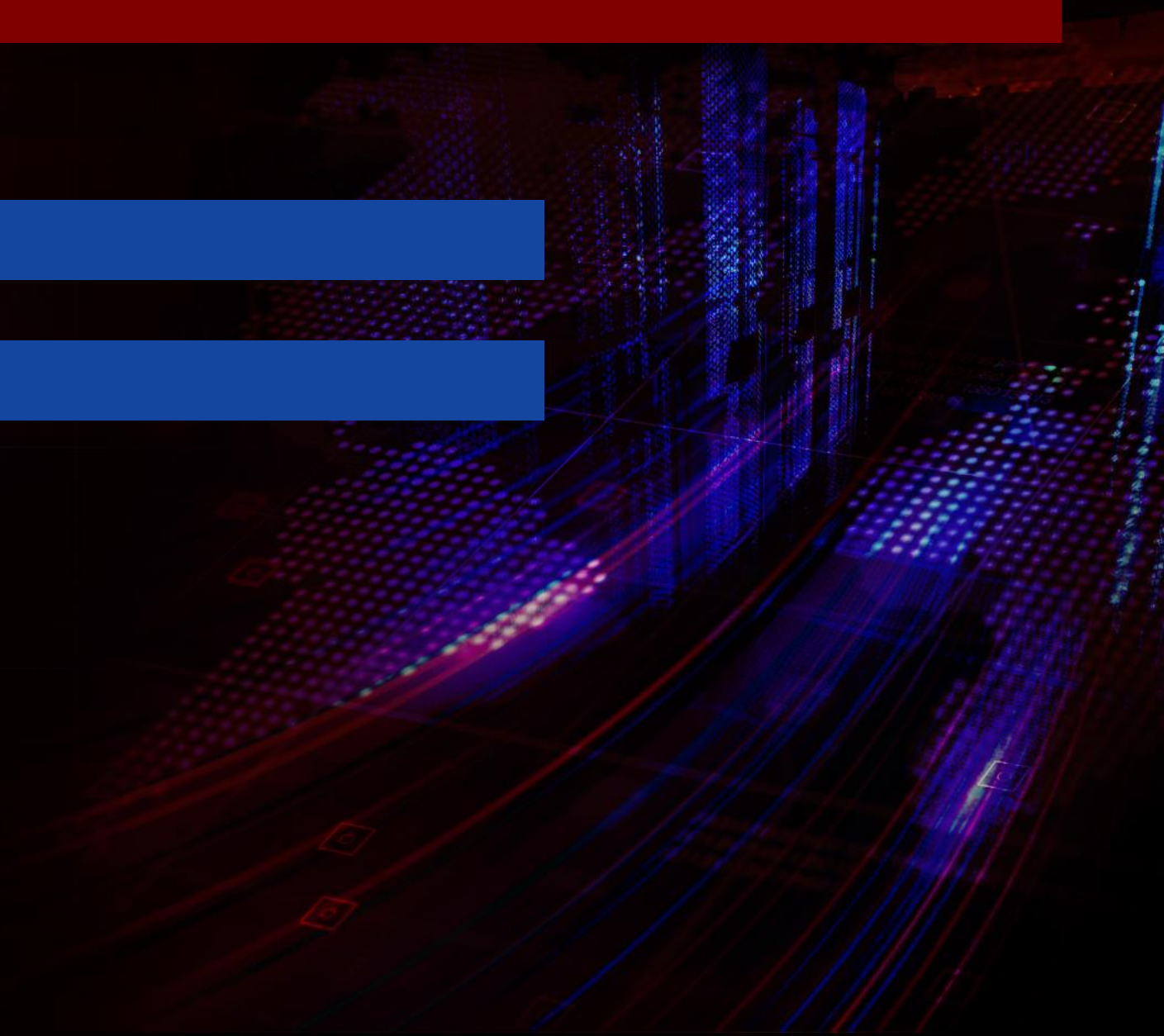
COMPUTAÇÃO ESCALÁVEL

**Paralelismo, Concorrência e
Escalabilidade**



ROTEIRO

- Arquiteturas de Memória
- Modelos de Programação



ARQUITETURAS DE MEMÓRIA

MEMÓRIA COMPARTILHADA

- Múltiplos processadores compartilham os mesmos recursos de memória em um espaço de endereçamento global
- Modificações efetuadas em uma região de memória por um processador são visíveis a todos os outros processadores

DIVISÃO EM DUAS CLASSES:

- UMA – tempo de acesso uniforme
- NUMA – tempo de acesso não uniforme



ARQUITETURAS DE MEMÓRIA

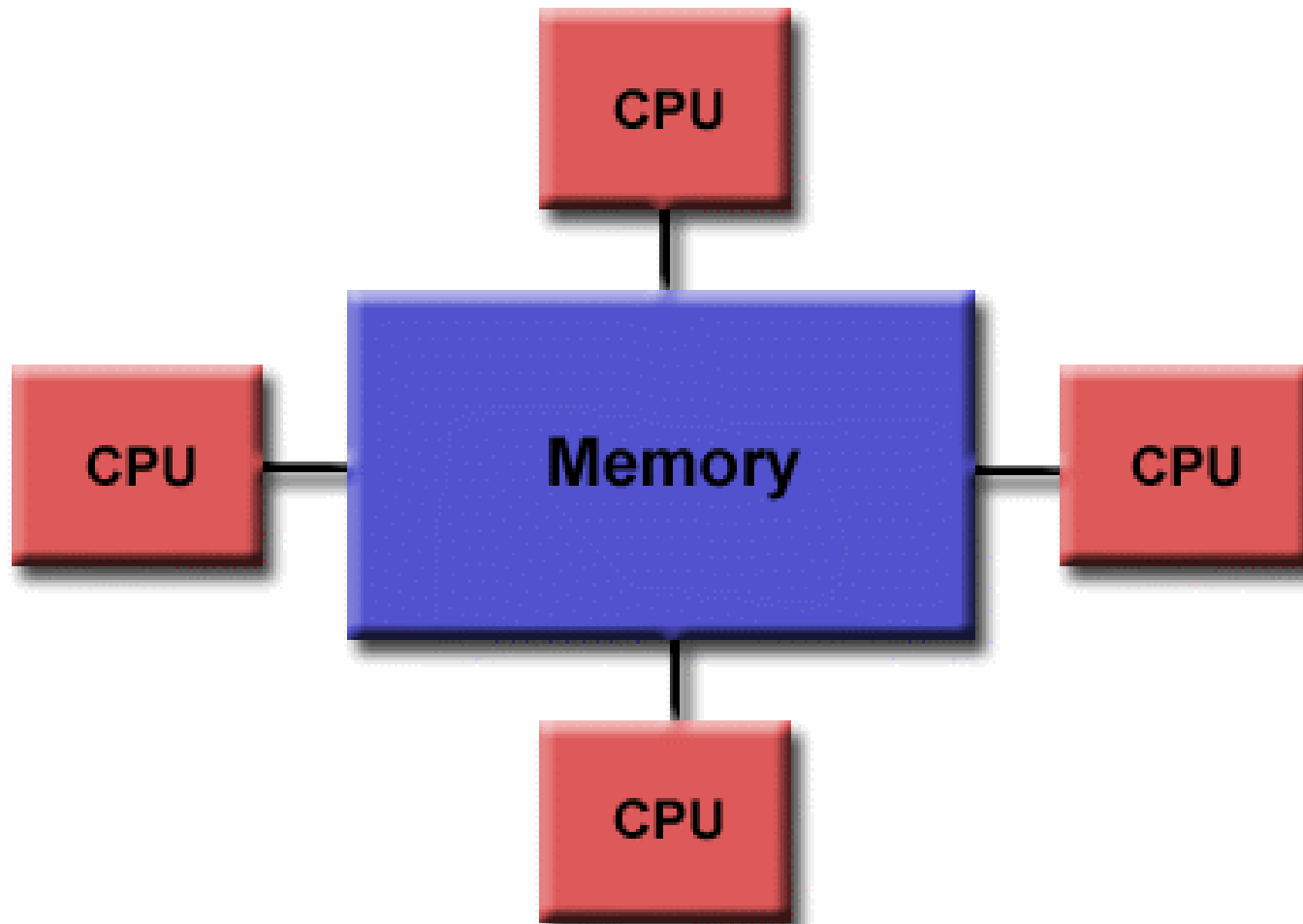
UNIFORM MEMORY ACCESS (UMA)

- Comum em processadores simétricos (SMP)
- Tempo de acesso igual para toda a memória
- Coerência de Cache
 - Se um processador atualiza um dado na memória compartilhada, isso tem que ser propagado para os outros processadores (cache interna à CPU)
 - Normalmente feita em hardware



ARQUITETURAS DE MEMÓRIA

UMA



ARQUITETURAS DE MEMÓRIA

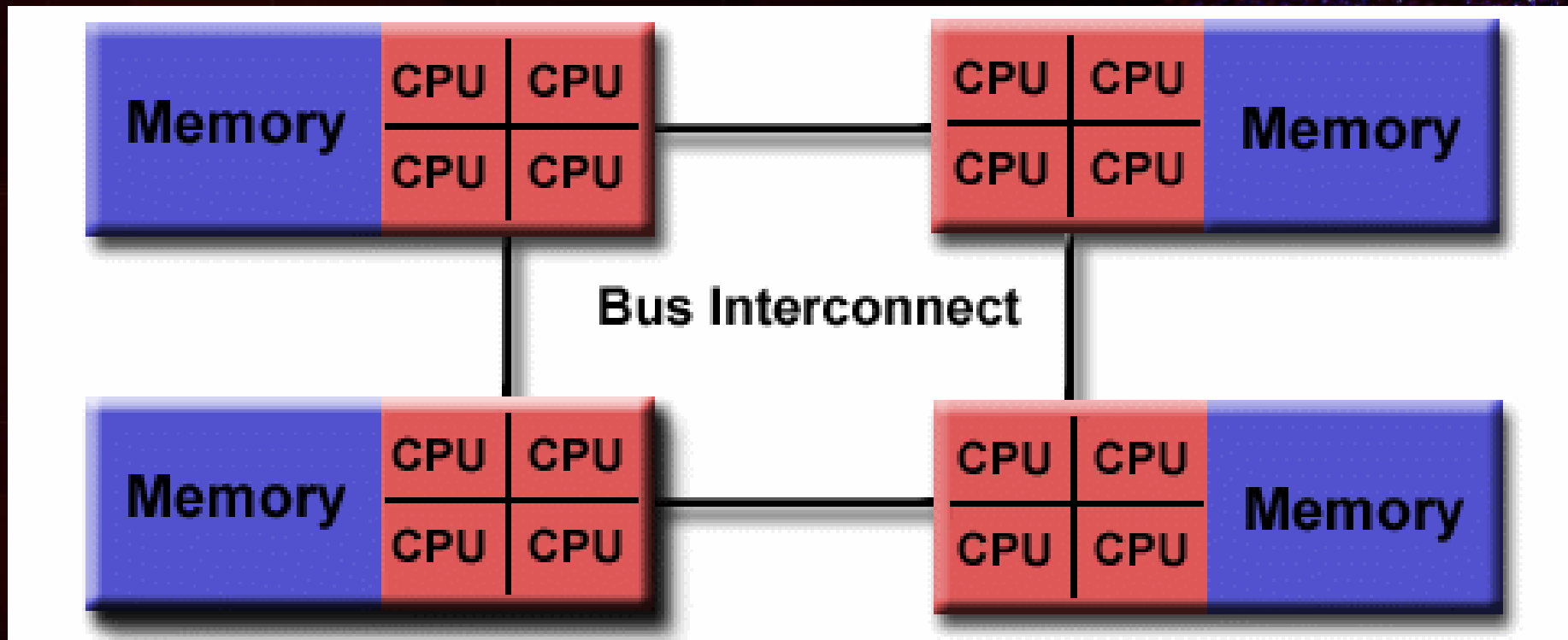
NON-UNIFORM MEMORY ACCESS

- **SMPs conectados por canais de comunicação**
- **O espaço de endereçamento ainda é global e um SMP pode acessar diretamente a memória do outro**
- **O tempo de acesso a uma posição de memória varia se ela pode ser acessada localmente ou em um processador remoto através dos canais de comunicação**
- **Se existe Coerência de Cache, a arquitetura é chamada de CC-NUMA (Cache Coherent NUMA)**



ARQUITETURAS DE MEMÓRIA

NUMA



ARQUITETURAS DE MEMÓRIA

MEMÓRIA COMPARTILHADA

VANTAGENS

- O espaço de endereçamento global permite um modelo de programação mais amigável (conhecido)
- O compartilhamento de dados entre as tarefas é rápido

DESVANTAGENS

- Falta de escalabilidade
- Aumentando o número de CPUs, aumenta o tráfego entre memória e CPU (bottleneck de von Neumann)
- É de responsabilidade do programador a sincronização dos acessos à memória global compartilhada



ARQUITETURAS DE MEMÓRIA

MEMÓRIA DISTRIBUÍDA

- **Sistemas de memória distribuída possuem uma rede de comunicação para interconectar os processadores**
- **Cada processador tem sua própria memória local**
- **Não existe um espaço de endereçamento global, um processador tem acesso somente a sua memória**
- **É tarefa do programador determinar quando e como um dado é propagado de um processador para o outro**
- **A sincronização das tarefas é também função do programador**

ARQUITETURAS DE MEMÓRIA

MEMÓRIA DISTRIBUÍDA

VANTAGENS

- A memória é escalável com o número de processadores
- Mais Processadores → Maior Quantidade de Memória
- Acesso rápido à memória do próprio processador sem interferência e sobrecarga da coerência da cache

DESVANTAGENS

- O programador precisa cuidar de muitos detalhes associados à comunicação de dados entre os processadores
- Dificuldade de mapear estruturas de dados projetadas para uma memória global com esta organização de memória



MODELOS DE PROGRAMAÇÃO

- Modelos de programação paralela existem como uma abstração acima da arquitetura de hardware e de memória
- Apesar de não ser evidente, os modelos não são específicos para um determinado tipo de arquitetura
- Qualquer modelo pode (teoricamente) ser implementado em qualquer hardware

MODELOS DE PROGRAMAÇÃO

MODELO DE MEMÓRIA COMPARTILHADA

- Neste modelo as tarefas compartilham um espaço de endereço comum, onde elas podem ler e escrever simultaneamente
- VANTAGEM → Para o programador
 - Não existe a necessidade de comunicação explícita de dados entre as tarefas
- Mecanismos como semáforos podem ser utilizados para controlar o acesso à memória compartilhada

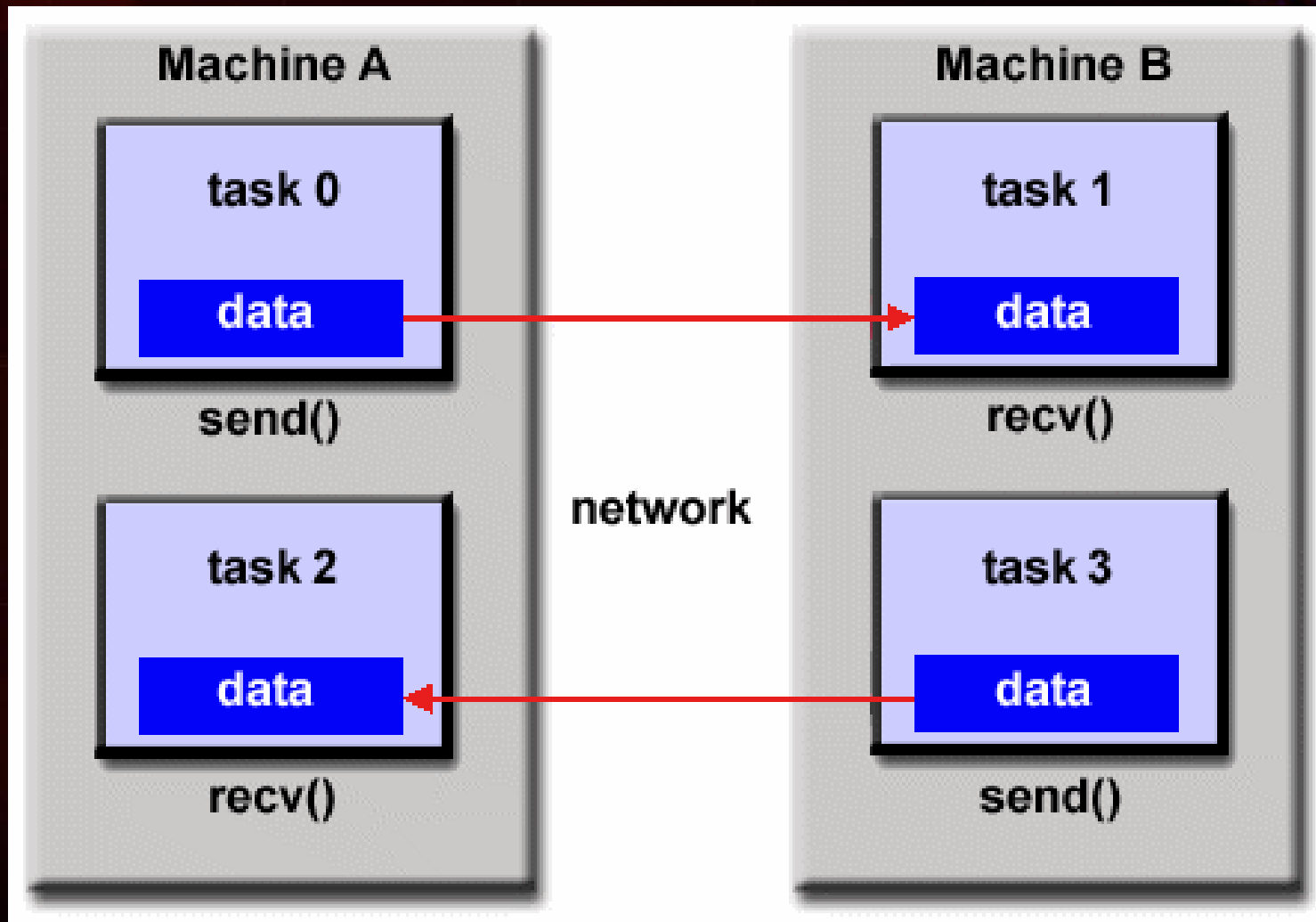


MODELOS DE PROGRAMAÇÃO

MODELO DE MEMÓRIA DISTRIBUÍDA

- Conjunto de tarefas que utilizam a memória local durante a computação
- Várias tarefas podem residir na mesma máquina
- Tarefas trocam dados através de comunicação, enviando e recebendo mensagens
- A transferência de dados em geral requer operações de cooperação a serem executadas por cada processo
Exemplo:
- Operação de envio deve ter uma operação de recepção correspondente

MODELOS DE PROGRAMAÇÃO



**MODELO DE
MEMÓRIA
DISTRIBUÍDA**

BIBLIOGRAFIA

1. [Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar - 2^a ed., Addison Wesley](#)
2. https://computing.llnl.gov/tutorials/parallel_comp/