

# **MCA - Mestrado em Computação Aplicada**

## **SDA - Sistemas Distribuídos Avançados**

### **Módulo 04:**

### **Programação Paralela**

**Prof. Dr. Charles Christian Miers – [charles.miers@udesc.br](mailto:charles.miers@udesc.br)**

Linha - Sistemas Computacionais

GRADIS - Grupo de Pesquisa em Redes de Computadores e Sistemas Distribuídos

LabP2D - Laboratório de Processamento Paralelo e Distribuído

PPGCA - Programa de Pós-Graduação em Computação Aplicada



# Computação Paralela: Definições

*“Um computador paralelo é um conjunto de processadores que estão habilitados a cooperar para resolver um problema.”* Ian Foster

- Motivações:
  - Computadores paralelos são interessantes porque oferecem potência de cálculo com recursos computacionais concentrados
  - O uso do paralelismo é ubíquo e tornou-se vital a empresas e usuários domésticos



# Computação Paralela: Definições

- Tipo de Paralelismos:
  - Paralelismo de dados:
    - O processo atua sobre sequência de funções ou operações similares, não necessariamente idênticas, sendo executadas em elementos de grandes estruturas de dados
    - Dados são particionados entre as tarefas
    - Técnicas de decomposição de domínio devem ser aplicadas sobre os dados que esta operando
  - Paralelismo funcional:
    - Cálculos totalmente diferentes podem ser executados concorrentemente nos mesmos dados ou em conjuntos de dados distintos

# Principais redundâncias

- Hardware:
  - Enlaces
  - Equipamentos de rede
  - Servidores
  - Componentes de servidores e equipamentos
    - *E.g.*, fonte, disco, *etc.*
- Software:
  - Frontend
    - *E.g.*, interfaces, APIs, *etc.*
  - Backend
    - *E.g.*, banco de dados, volumes, objetos, *etc.*

- Duas abordagens:
  - Taxonomia de Flynn
  - Arquitetura



# **Taxonomia de Flynn (Instruções/Dados)**

- Pesquisador Michael J. Flynn propôs em 1966 uma classificação dos computadores baseando-se no fluxo de instruções e no tratamento dos dados
- Taxonomia de Flynn leva em consideração como o fluxo de instruções e como os dados são processados

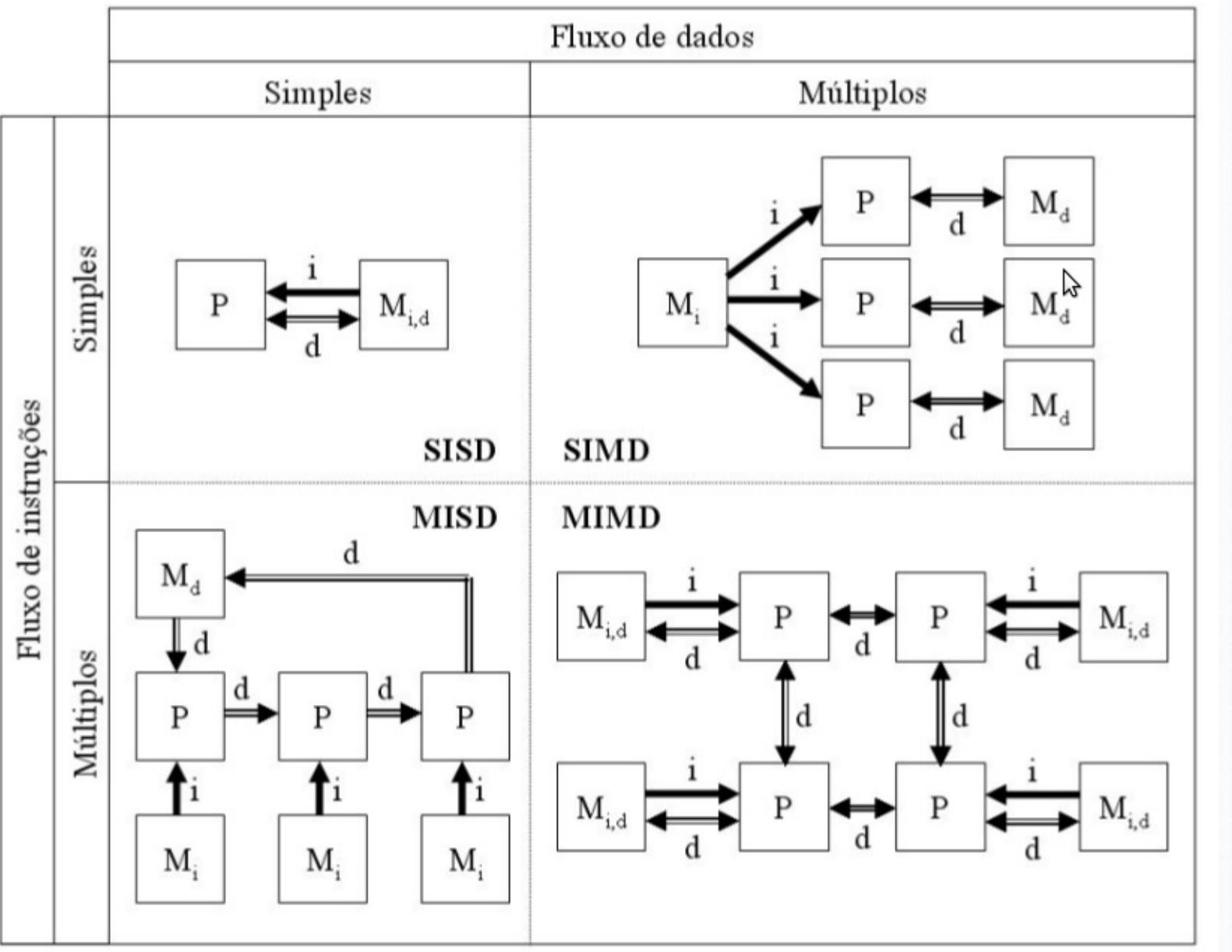


# Taxonomia de Flynn (Instruções/Dados)

	Instruções Simples	Instruções Múltiplas
Dados Simples	SISD	MISD
Dados Múltiplos	SIMD	MIMD

- SISD: *Single Instruction Single Data*
- SIMD: *Single Instruction Multiple Data*
- MISD: *Multiple Instruction Single Data*
- MIMD: *Multiple Instruction Multiple Data*

# Taxonomia de Flynn (Instruções/Dados)



P = processador

M = memória

i = instruções

d = dados



# Taxonomia de Flynn (Instruções/Dados)

## SISD:

load A	time
load B	
$C = A + B$	
store C	
$A = B * 2$	
store A	

## SIMD:

prev instruct	prev instruct	prev instruct
load A(1)	load A(2)	load A(n)
load B(1)	load B(2)	load B(n)
$C(1) = A(1) * B(1)$	$C(2) = A(2) * B(2)$	$C(n) = A(n) * B(n)$
store C(1)	store C(2)	store C(n)
next instruct	next instruct	next instruct

P1

P2

P3

## MISD:

prev instruct
load A(1)
$C(1)=A(1)*1$
store C(1)
next instruct

prev instruct
load A(1)
$C(2)=A(1)*2$
store C(2)
next instruct

prev instruct
load A(1)
$C(n)=A(1)*n$
store C(n)
next instruct

time

## MIMD:

prev instruct	prev instruct	prev instruct
load A(1)	call funcD	do 10 i=1,N
load B(1)	$x = y * z$	$\alpha = w ^\star 3$
$C(1) = A(1) * B(1)$	$sum = x * 2$	$\zeta = C(i)$
store C(1)	call sub1(i,j)	10 continue
next instruct	next instruct	next instruct

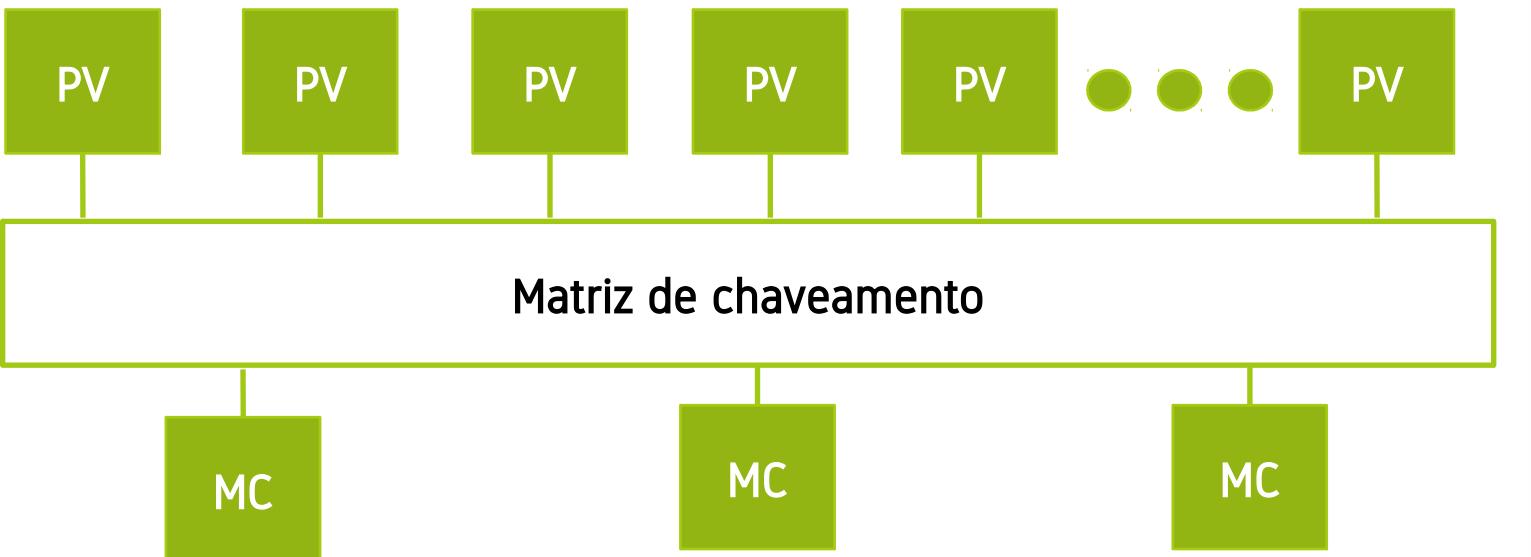
P1

P2

Pn

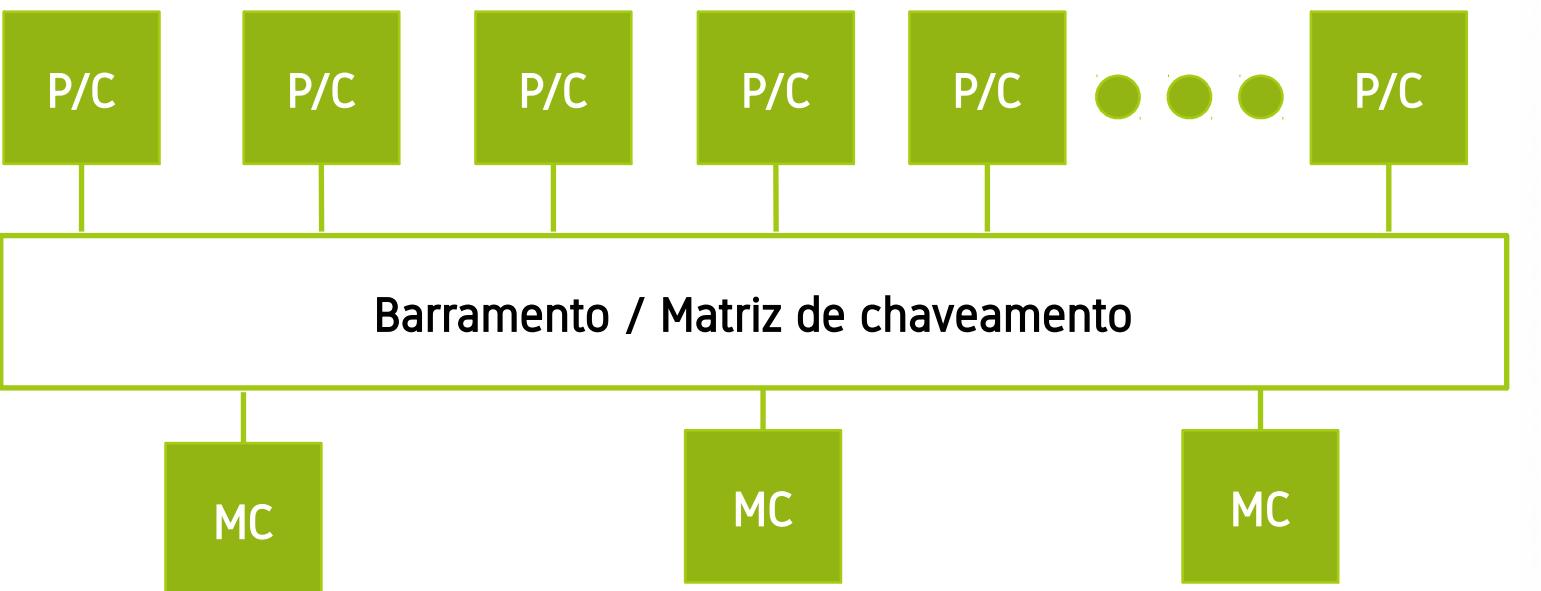
# Arquitetura: Processadores Vetoriais Paralelos (PVP)

- Processadores otimizados para processamento vetorial
- PV: processador vetorial
- MC: memória compartilhada
- Classificação: UMA
- Exemplos: Cray C-90, Cray T-90, Fujitsu VP, NEC Sx



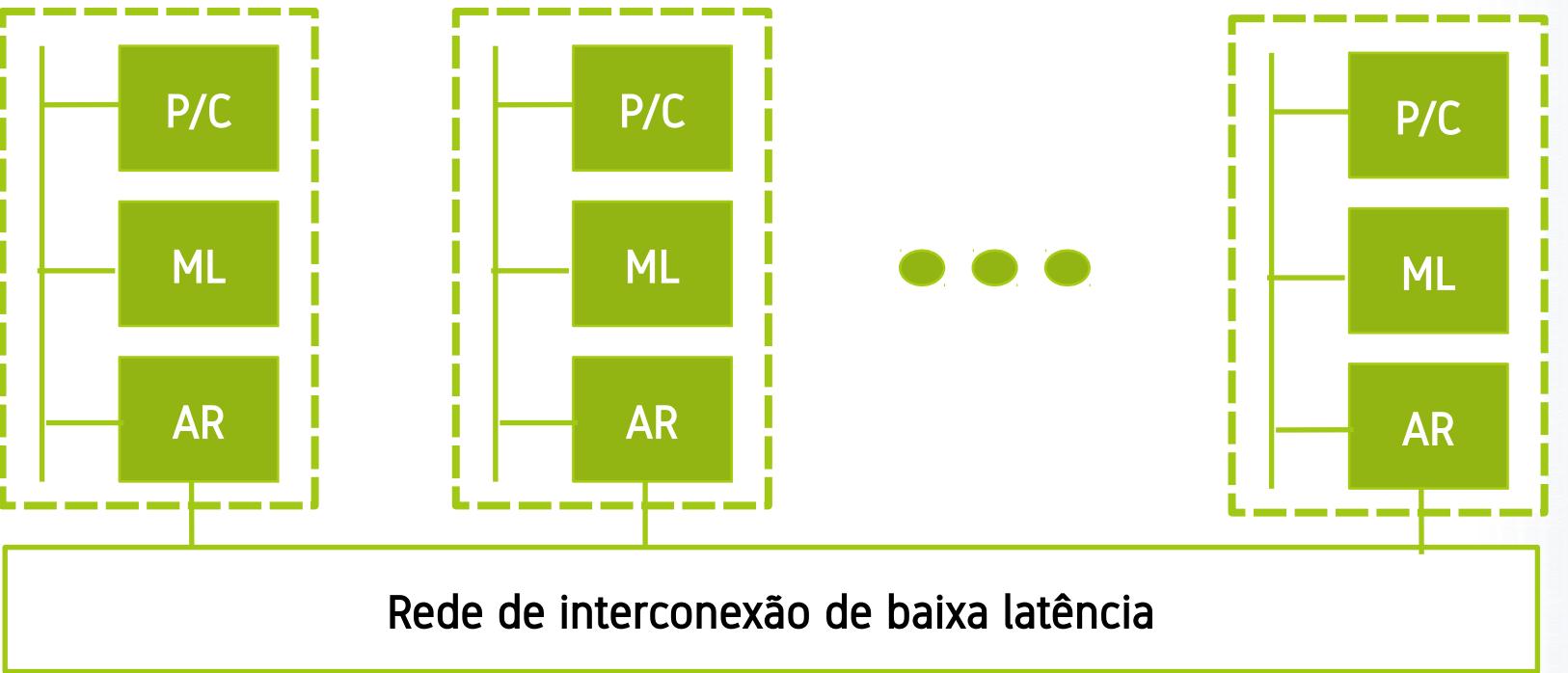
# Arquitetura: Multiprocessadores simétricos (SMP)

- Processadores comerciais
- Todos os processadores tem igual acesso ao barramento
- P/C: processador + cache
- MC: memória compartilhada
- Classificação: UMA
- Exemplos: SGI Power Challange, Sun Ultra Enterprise, DEC Alpha Server



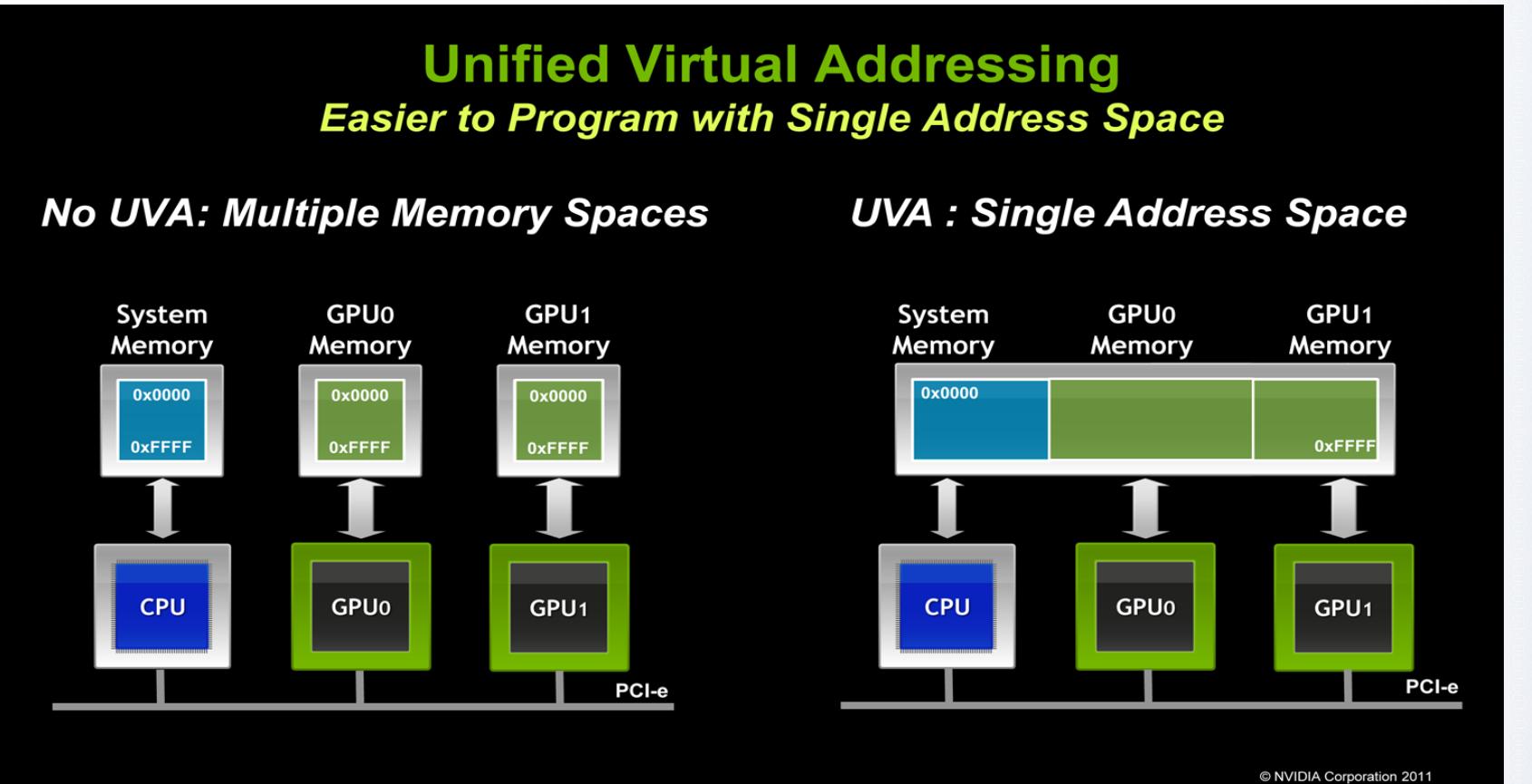
# Arquitetura: Máquinas Massivamente Paralelas (MPP)

- Milhares de processadores comerciais (P/C)
- Rede proprietária de alta velocidade
  - Obtenção de alto desempenho através da utilização de diversos processadores
- ML: memória local
- AR: adaptador de rede
- Classificação: NORMA
- Exemplos: Intel Paragon, IBM SP2



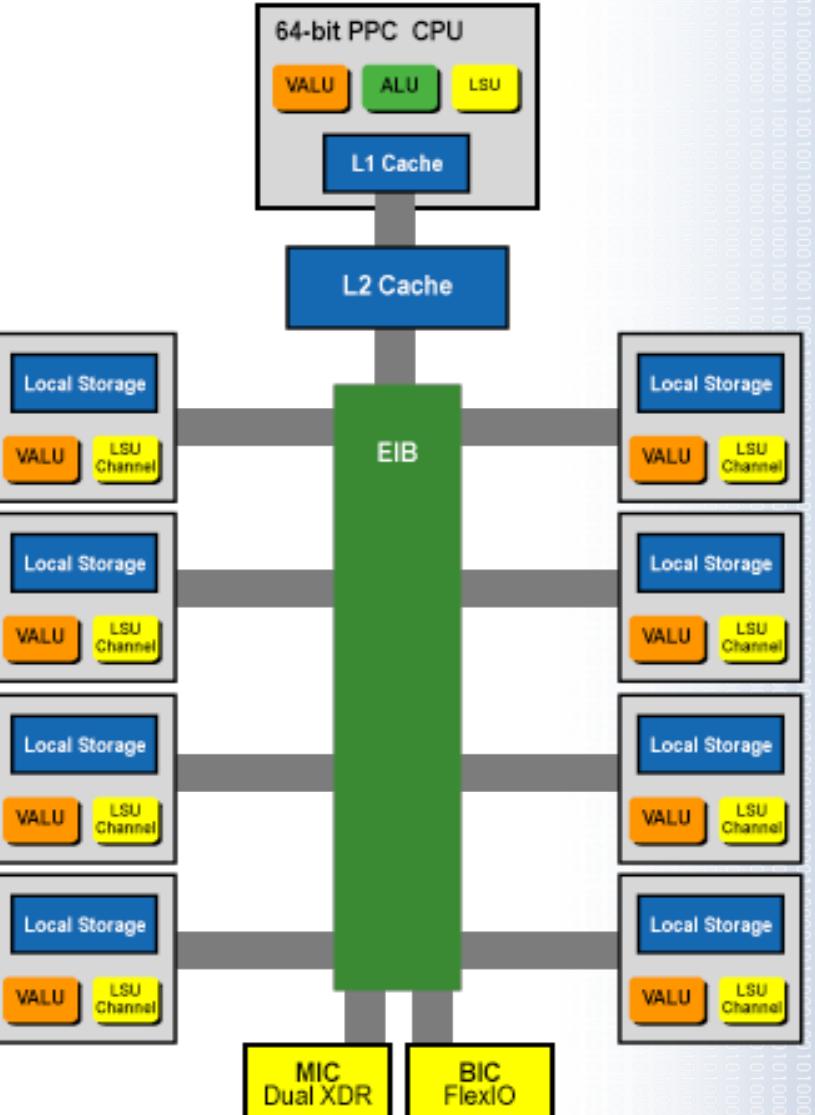
# NVIDIA CUDA

- Projeto CUDA: Computer Unified Device Architecture
  - <https://developer.nvidia.com/cuda-zone>



# Arquitetura: Cell Processor

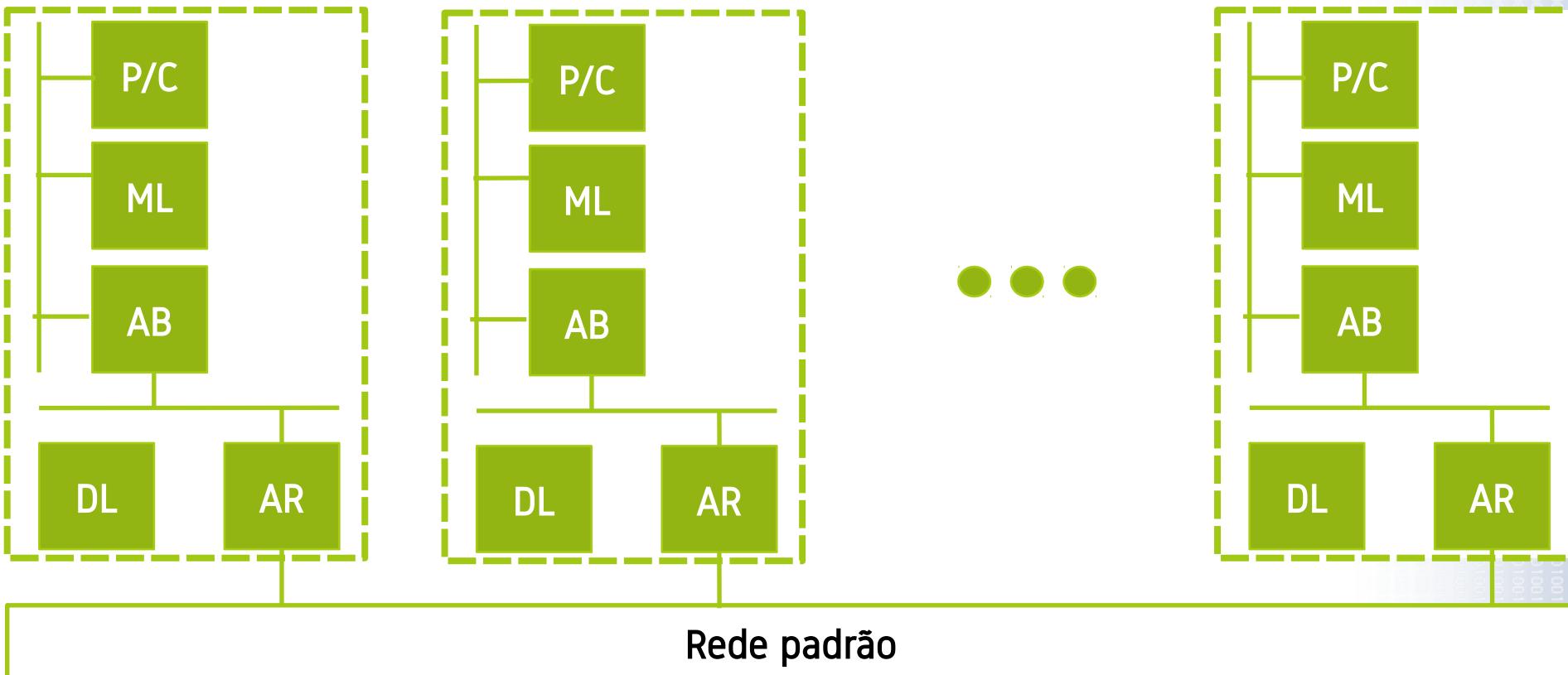
- Cell Broadband Engine Architecture (CBEA)
- Arquitetura semelhante a GPU NVIDIA G80
- Arquitetura utilizada em consoles Sony PS3



The CELL Architecture

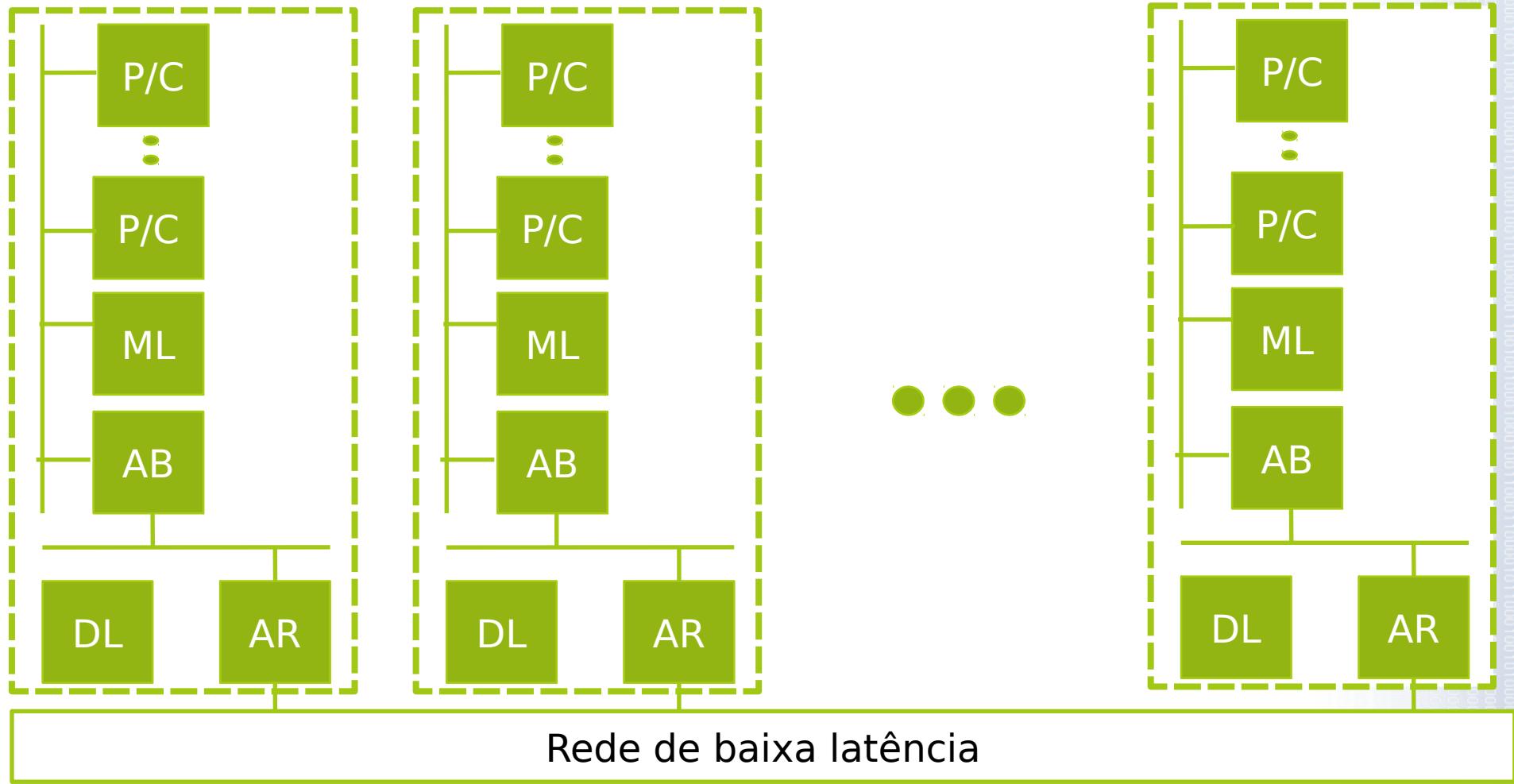
# Arquitetura: Redes de estações de trabalho (NOW)

- Várias estações de trabalho interligadas por tecnologia tradicional de rede
  - *E.g.*, Ethernet e ATM
- Máquina NORMA



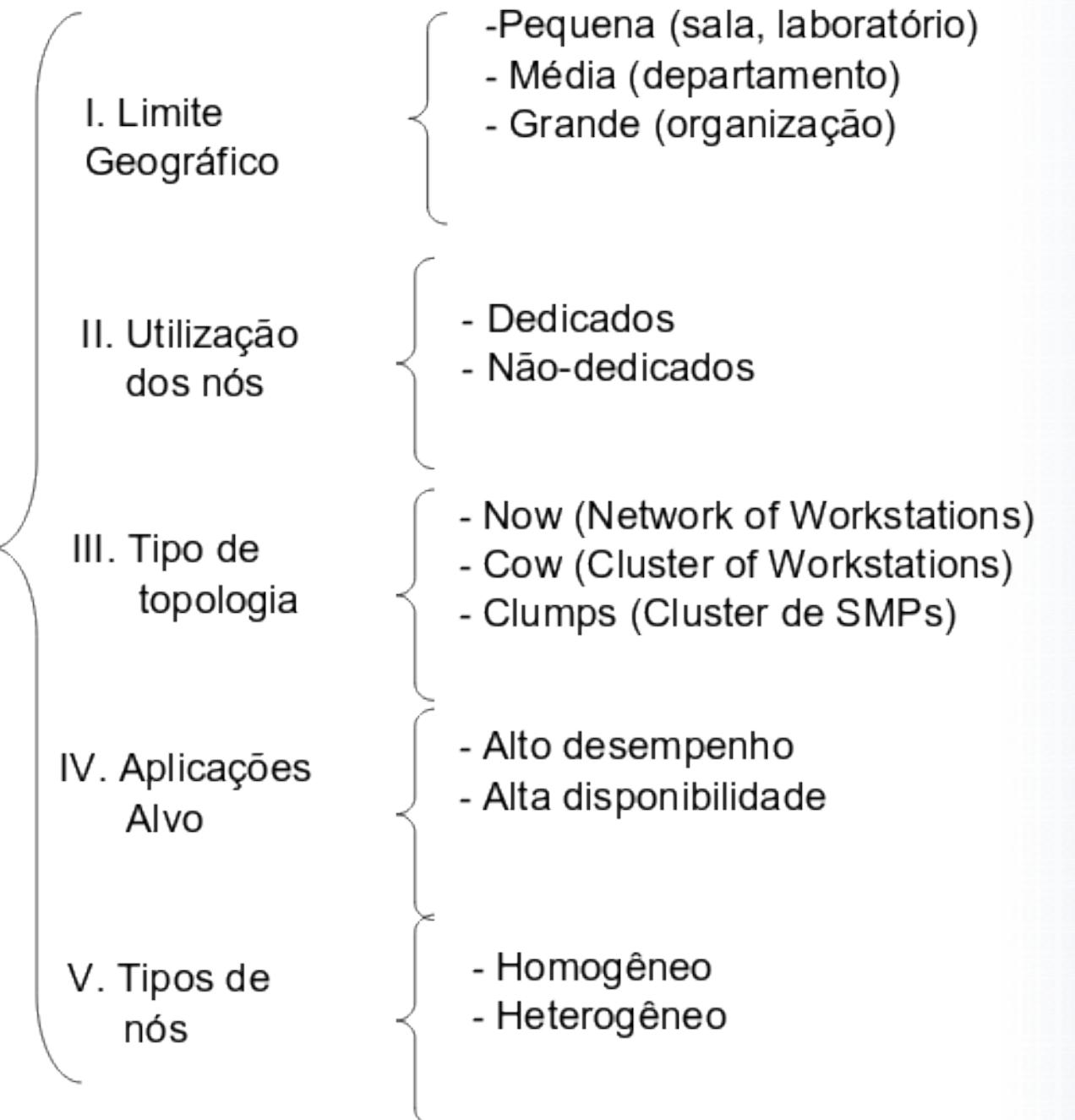
# Máquinas agregadas (clusters)

- Principal diferença para NOW:
  - Rede de baixa latência
- Classificação: NORMA



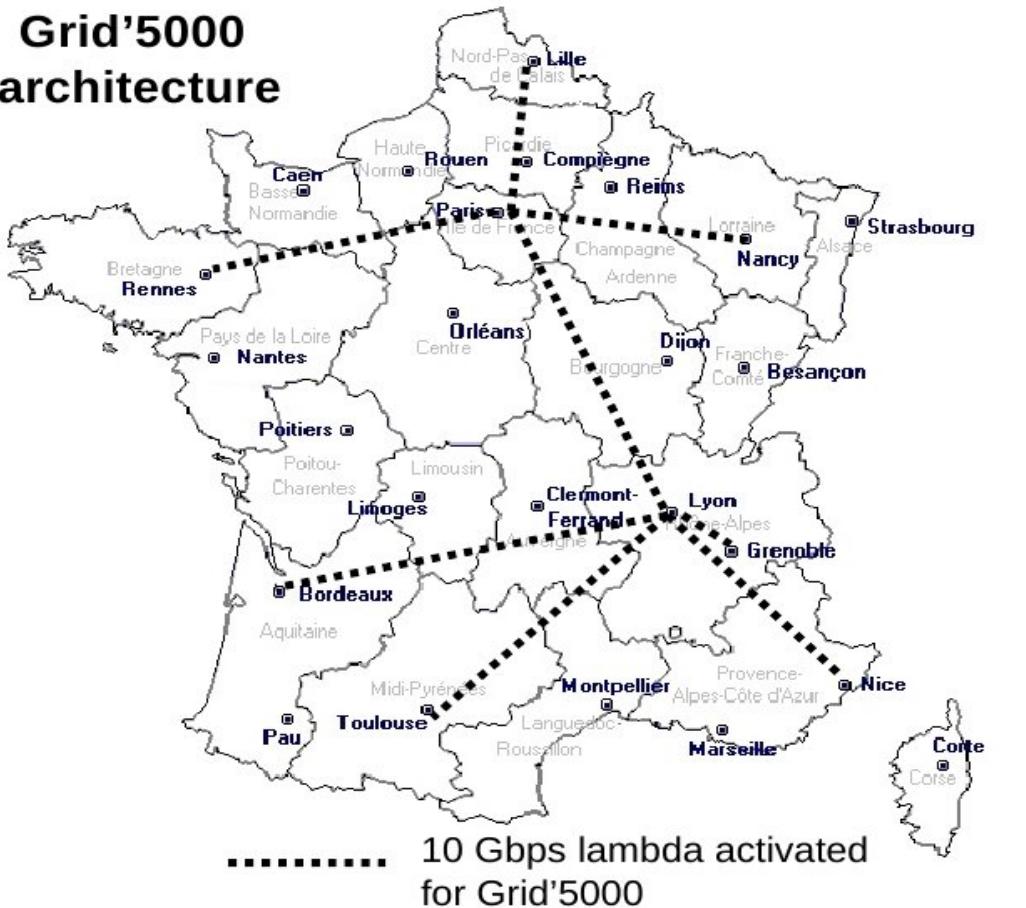
## Métricas para Classificação dos Clusters

# Classificação de clusters



# Grades computacionais (Grids)

## Grid'5000 architecture



- Grid5000
  - [www.grid5000.fr](http://www.grid5000.fr)
  - [www.futuregrid.org](http://www.futuregrid.org)
- Mais de 9 sites interconectados por uma rede dedicada
- +/- 8000 processadores
- Rede?
- Ponto no Brasil: UFRGS / POA

# Grades computacionais (Grids)



## Processor families

Processors \ Sites	Bordeaux	Grenoble	Lille	Luxembourg	Lyon	Nancy	Reims	Rennes	Sophia	Toulouse	Processors total
AMD Opteron	226		52		158		88	80	212	280	1096
Intel Xeon	102	232	148	44		328		178	90		1122
<b>Sites total</b>	<b>328</b>	<b>232</b>	<b>200</b>	<b>44</b>	<b>158</b>	<b>328</b>	<b>88</b>	<b>258</b>	<b>302</b>	<b>280</b>	<b>2218</b>

## Processor details

Processors \ Sites	Bordeaux	Grenoble	Lille	Luxembourg	Lyon	Nancy	Reims	Rennes	Sophia	Toulouse	Processors total
AMD Opteron 2218	186								100	280	566
AMD Opteron 250					158						158
AMD Opteron 275									112		112
AMD Opteron 285			52								52
AMD Opteron 6164 HE							88	80			168
AMD Opteron 8218	40										40
Intel Xeon E5420 QC		68									68
Intel Xeon E5440 QC			92								92
Intel Xeon E5520		164							90		254
Intel Xeon E5620			56								56
Intel Xeon EM64T	102										102
Intel Xeon L5335			44								44
Intel Xeon L5420						184		128			312
Intel Xeon X3440						144					144
Intel Xeon X5570							50				50
<b>Sites total</b>	<b>328</b>	<b>232</b>	<b>200</b>	<b>44</b>	<b>158</b>	<b>328</b>	<b>88</b>	<b>258</b>	<b>302</b>	<b>280</b>	<b>2218</b>

# Grades computacionais (Grids)

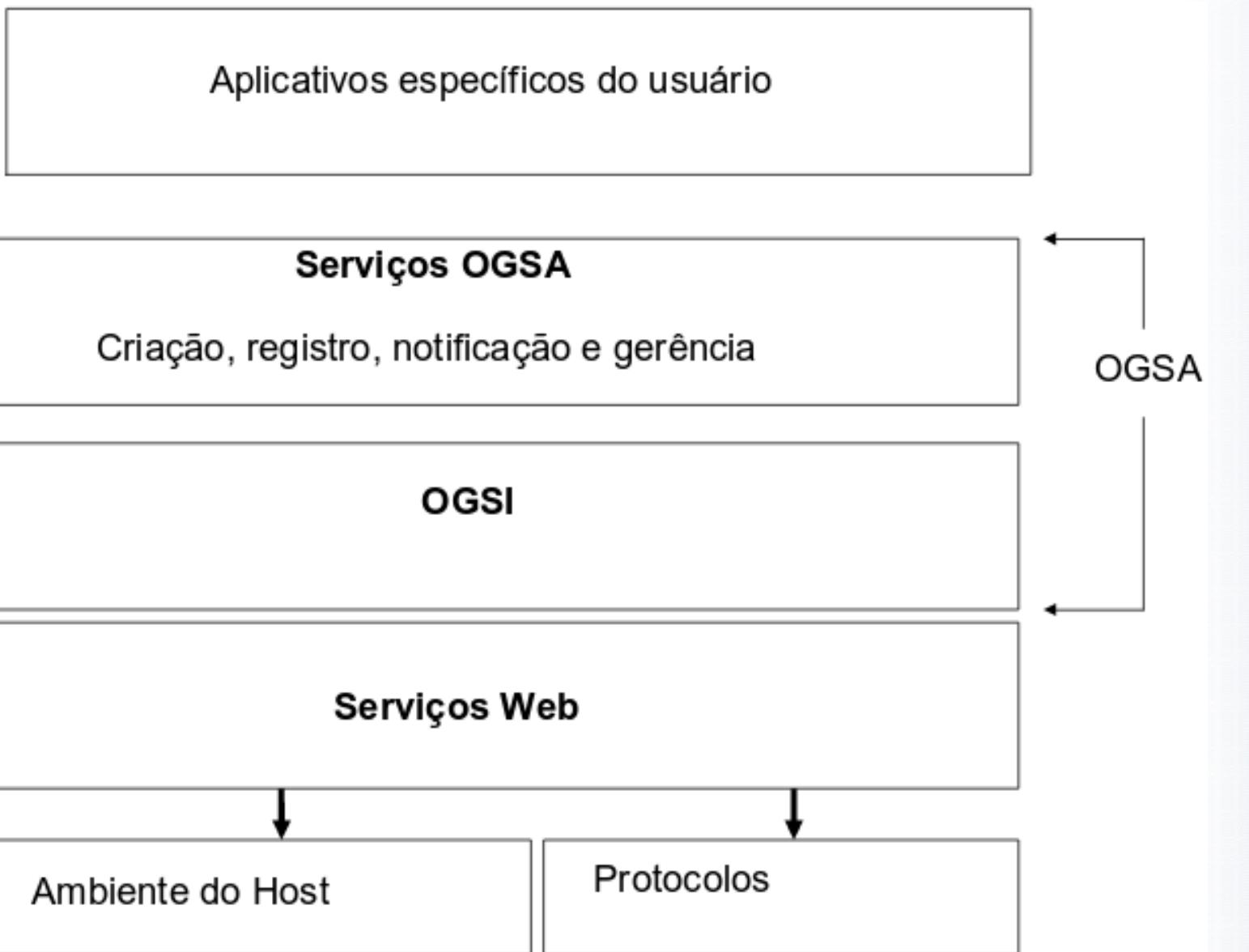
## Open Grid Services Architecture (OGSA)



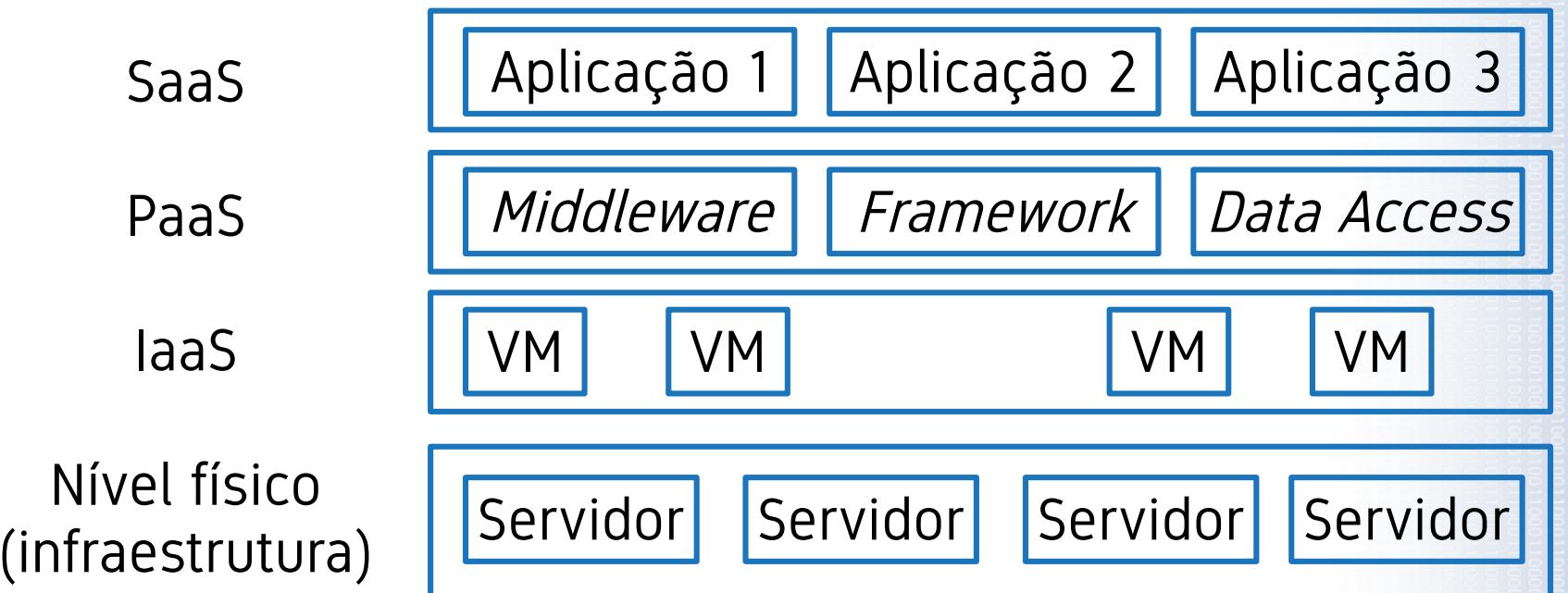
***Web Services***

***Protocolos de Grid***

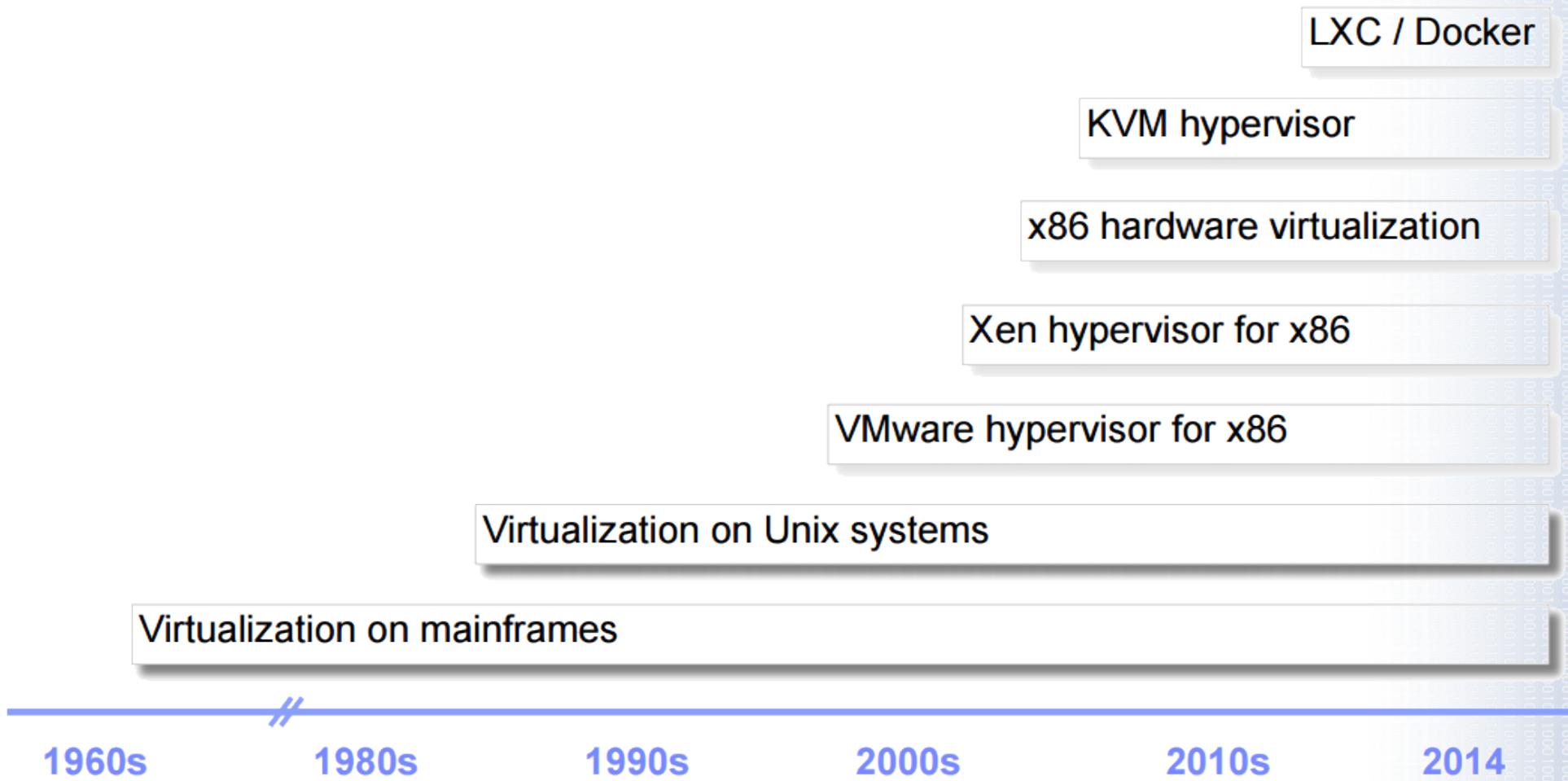
# Grades computacionais (Grids)



# Nuvens Computacionais



# Virtualização: perspectiva histórica



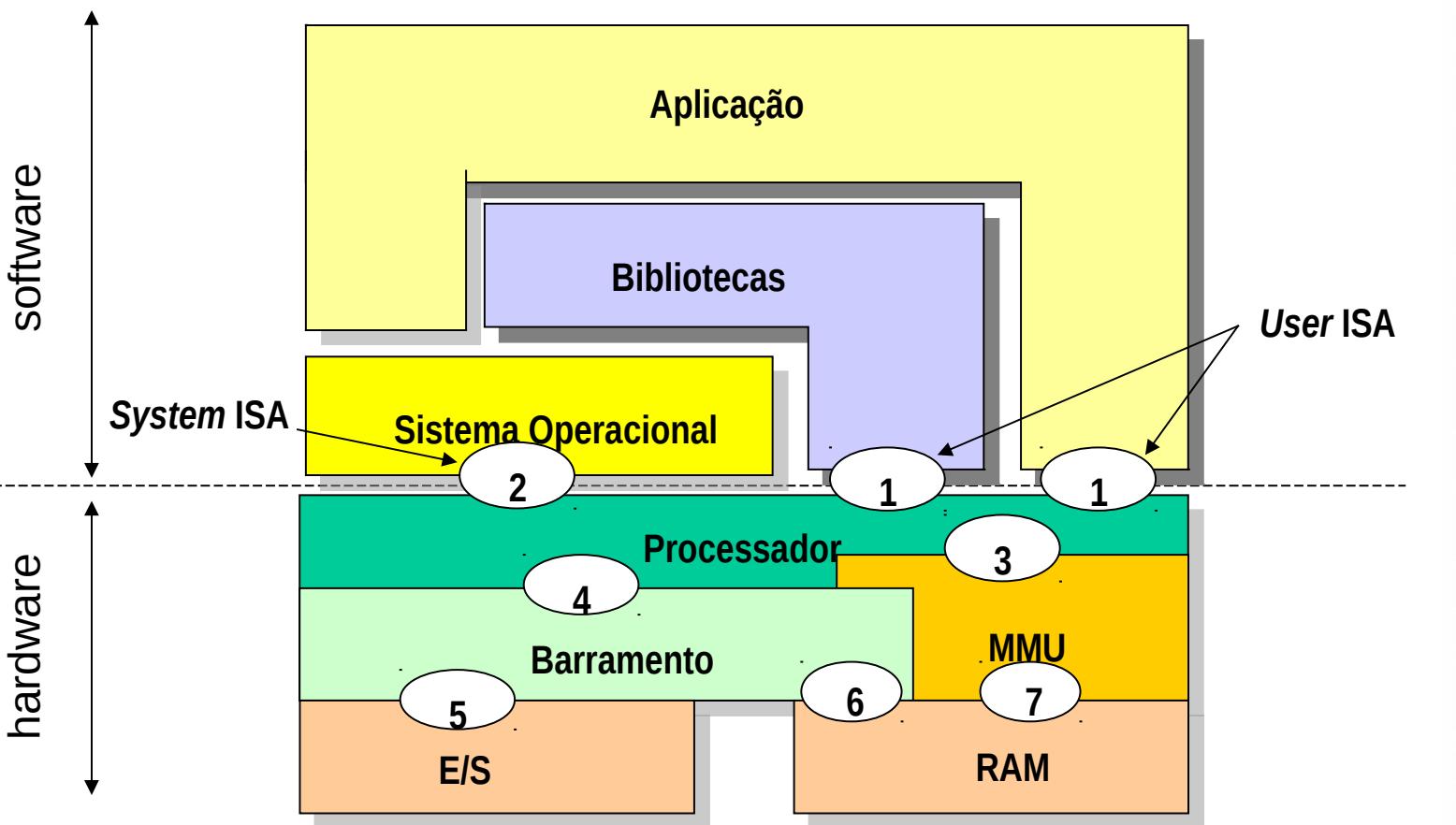


# Virtualização: Conceito

- Método para dividir recursos de um computador em múltiplos ambientes de execução usando várias técnicas
  - Particionamento de hardware e software
  - Compartilhamento de tempo
  - Simulação completa ou parcial da máquina (modelo)
  - Emulação (comportamento)
- Pragmaticamente:
  - Técnica que permite particionar um único sistema computacional em vários outros sistemas totalmente separados
  - É um ambiente que estende/substitui uma interface existente por outra

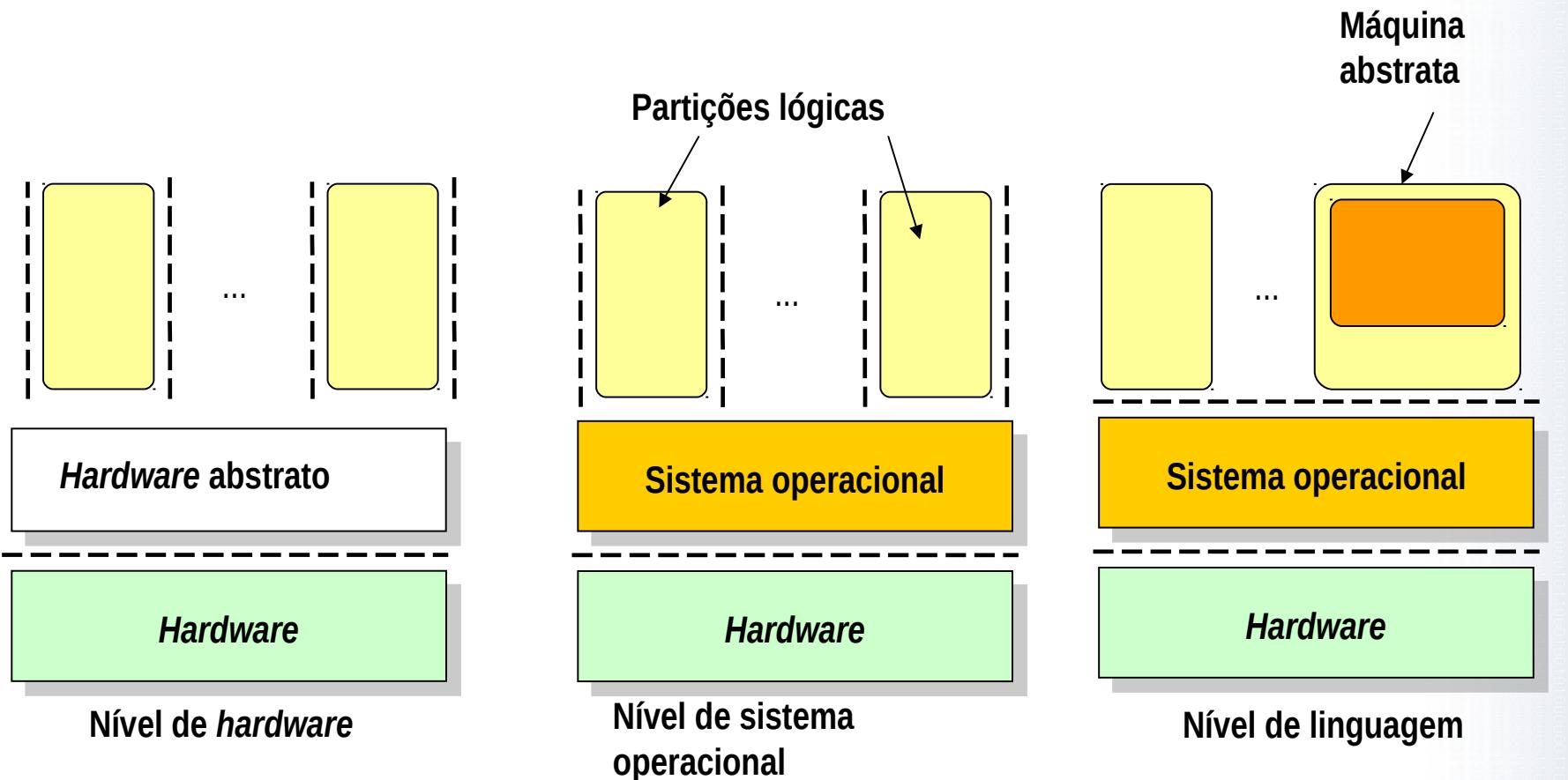
# Virtualização: Arquitetura abstrata

- ISA: *Instruction Set Architecture*
  - Interface limítrofe entre o nível de abstração de hardware e o de software
  - Retratam (pelo menos) dois modos de operação da CPU
    - Privilegiado → *System ISA*
    - Não privilegiado → *User ISA*

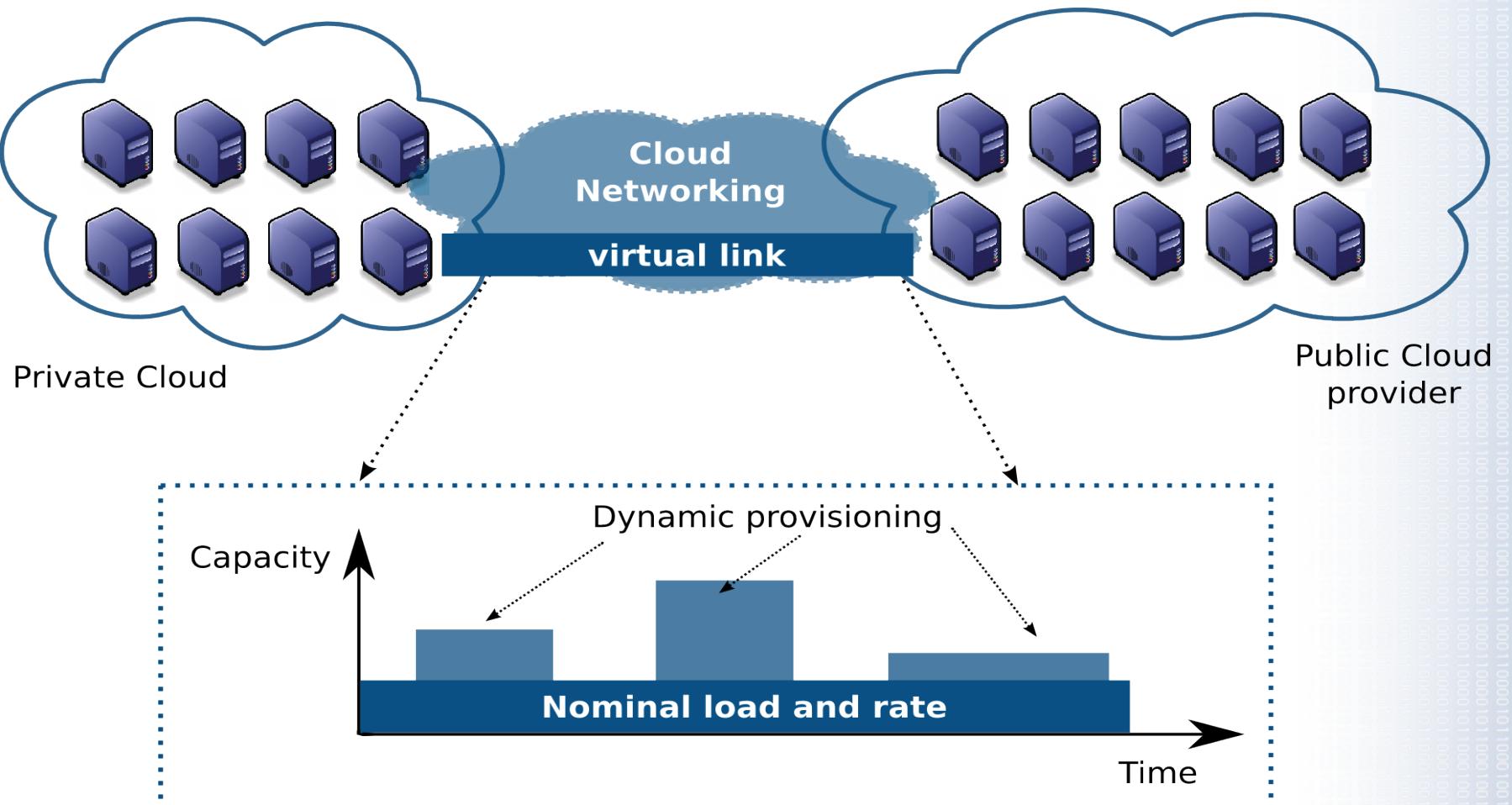


# Virtualização: Tipos

- Consiste em estender ou substituir um recurso, ou interface, por um outro de modo a imitar um comportamento
- Objetivos comuns: compatibilidade, isolamento e desempenho



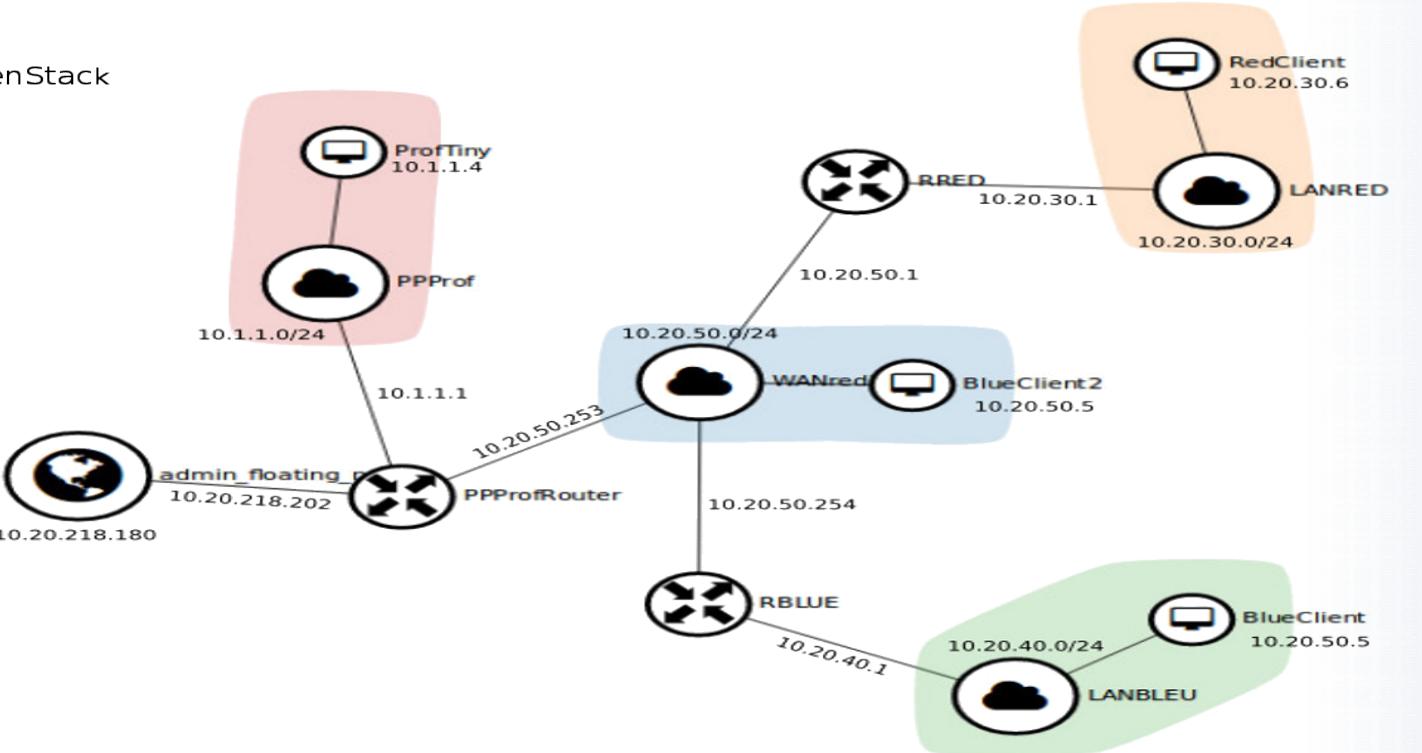
# Nuvens de comunicação





# Nuvens de comunicação

Rede Virtual com OpenStack



## Routers / RRED

DESTINATION CIDR	NEXT HOP
0.0.0.0/0	10.20.50.253
10.1.1.0/24	10.20.50.253
10.20.40.0/24	10.20.50.254

```
root@blueclient2:~# route -n
Kernel IP routing table
Destination      Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0          10.20.50.253   0.0.0.0       UG    0      0    0 ens3
10.1.1.0         10.20.50.253   255.255.255.0  UG    0      0    0 ens3
10.20.30.0       10.20.50.1     255.255.255.0  UG    0      0    0 ens3
10.20.40.0       10.20.50.254   255.255.255.0  UG    0      0    0 ens3
10.20.50.0       0.0.0.0        255.255.255.0  U      0      0    0 ens3
169.254.169.254 10.20.50.3    255.255.255.255 UGH   0      0    0 ens3
```

## Routers / RBLUE

DESTINATION CIDR	NEXT HOP
0.0.0.0/0	10.20.50.253
10.1.1.0/24	10.20.50.253
10.20.30.0/24	10.20.50.1

## Routers / PPPProfRouter

DESTINATION CIDR	NEXT HOP
0.0.0.0/0	10.20.218.254
10.20.30.0/24	10.20.50.1
10.20.40.0/24	10.20.50.254

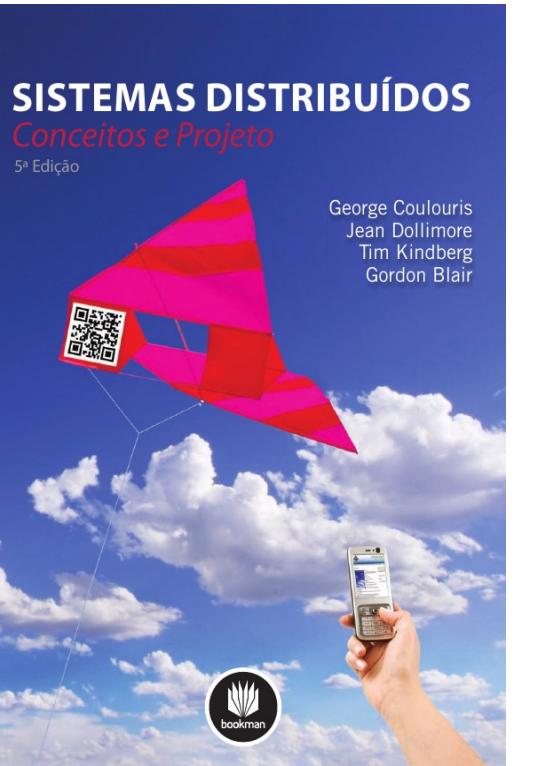


UDESC  
JOINVILLE



# Leitura recomendada

- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G.  
Sistemas Distribuídos: Conceitos e Projeto.  
5<sup>a</sup> Edição. Bookman, 2013.  
– Capítulos 4 e 7



# Direitos autorais

- As transparências apresentadas são adaptadas do material original desenvolvido por:
  - COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. *Sistemas Distribuídos: Conceitos e Projeto*, 5<sup>a</sup> Edição. Bookman, 2013
  - Material de aula do Prof. Dr. Maurício Aronne Pillon
  - Material de aula do Prof. Charles Christian Miers