

APLICAÇÕES ADAPTATIVAS EM ARQUITETURAS PARALELAS

Guilherme Galante
Rodrigo da Rosa Righi



PARTE I – Introdução e Conceitos

Arquiteturas Paralelas

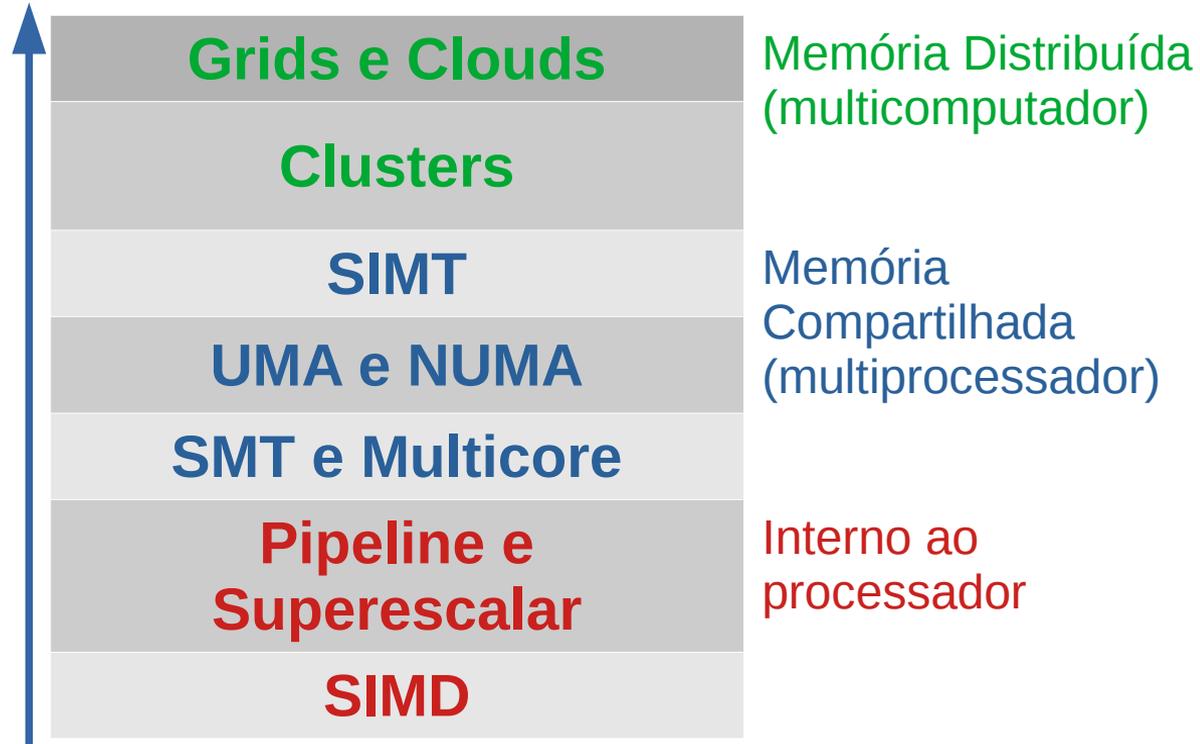
- Memória Compartilhada
- Clusters
- Grids
- Clouds/Fog

Modelos de Programação Paralela

- OpenMP
- MPI

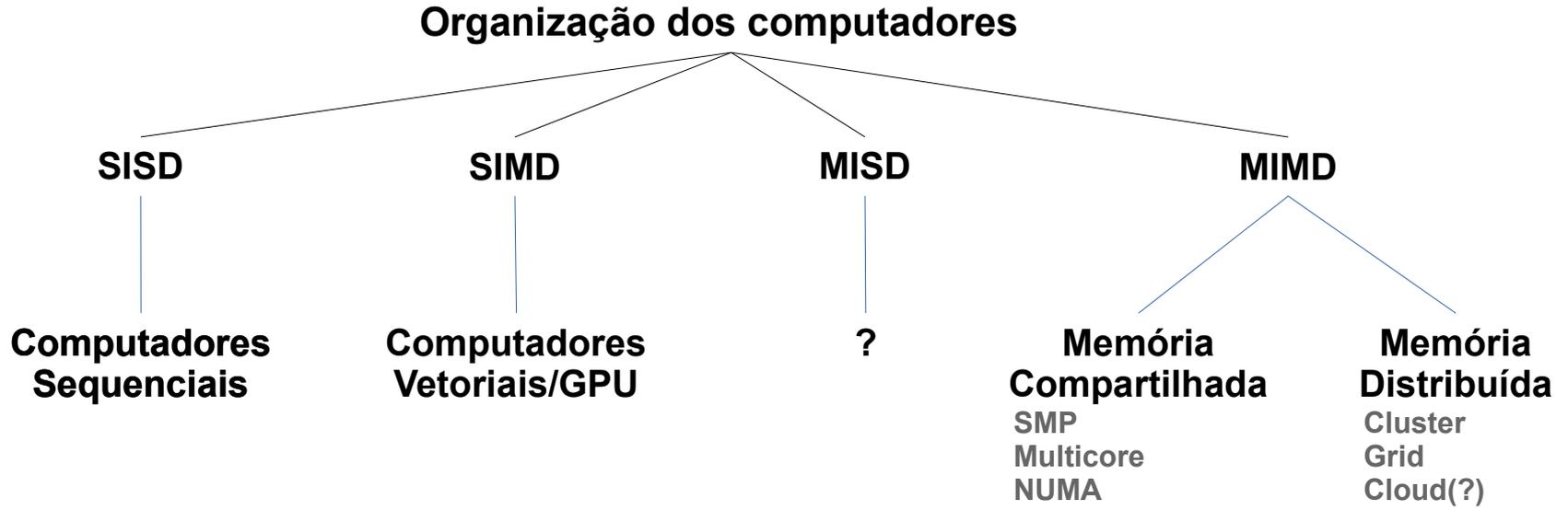
Arquiteturas Paralelas

Níveis de Paralelismo

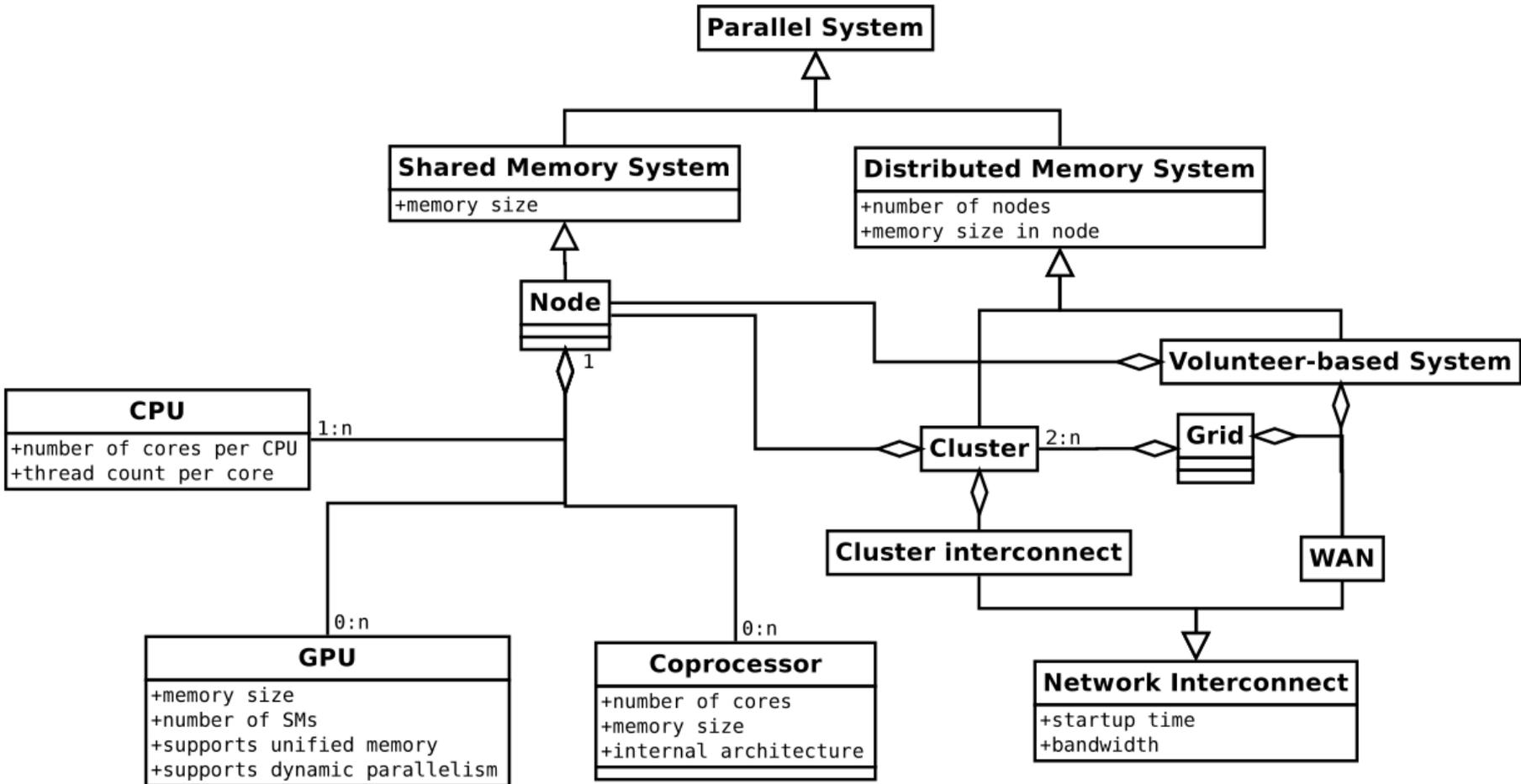


Arquiteturas Paralelas

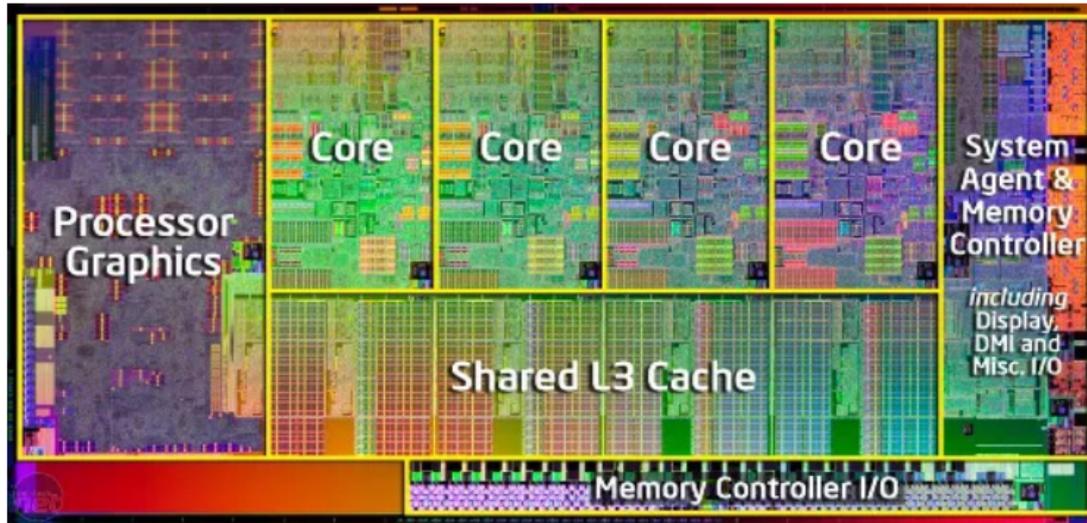
Classificação de Flynn



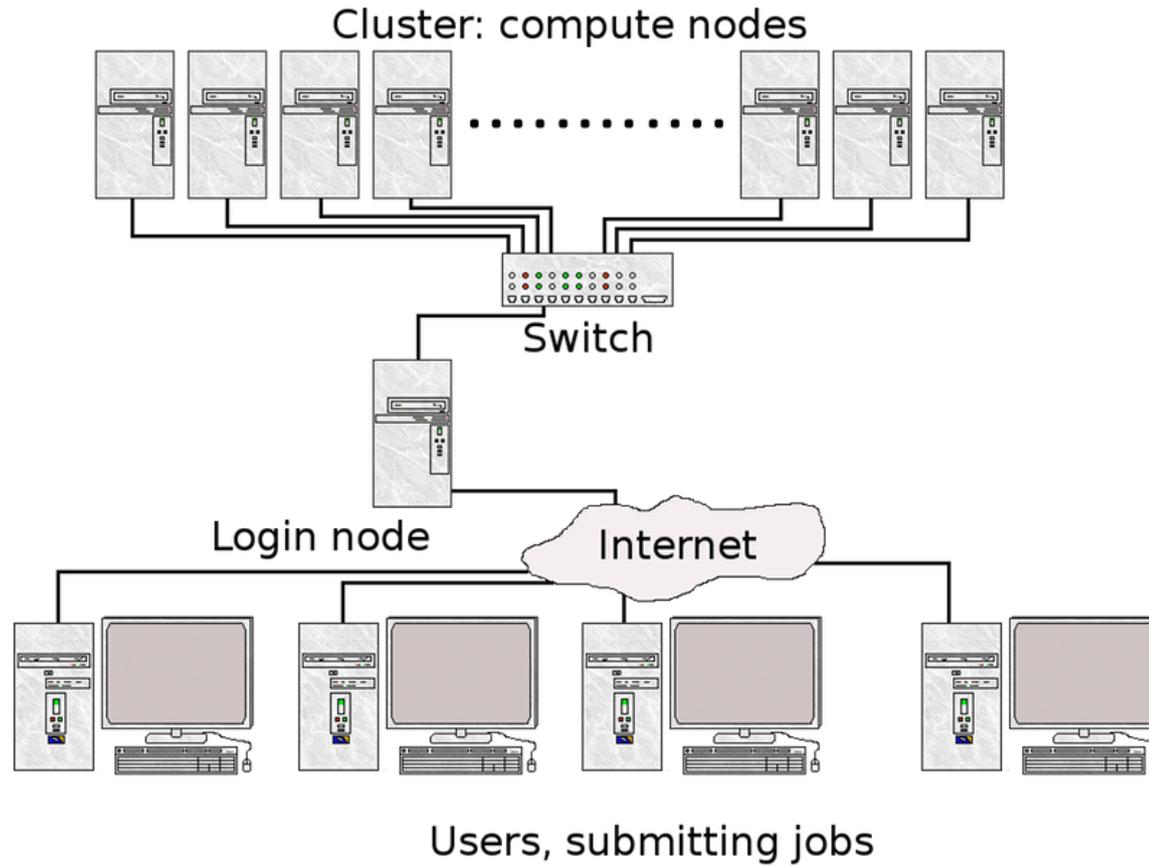
Arquiteturas Paralelas



Multicore



Clusters



Cluster Santos Dummont (LNCC)

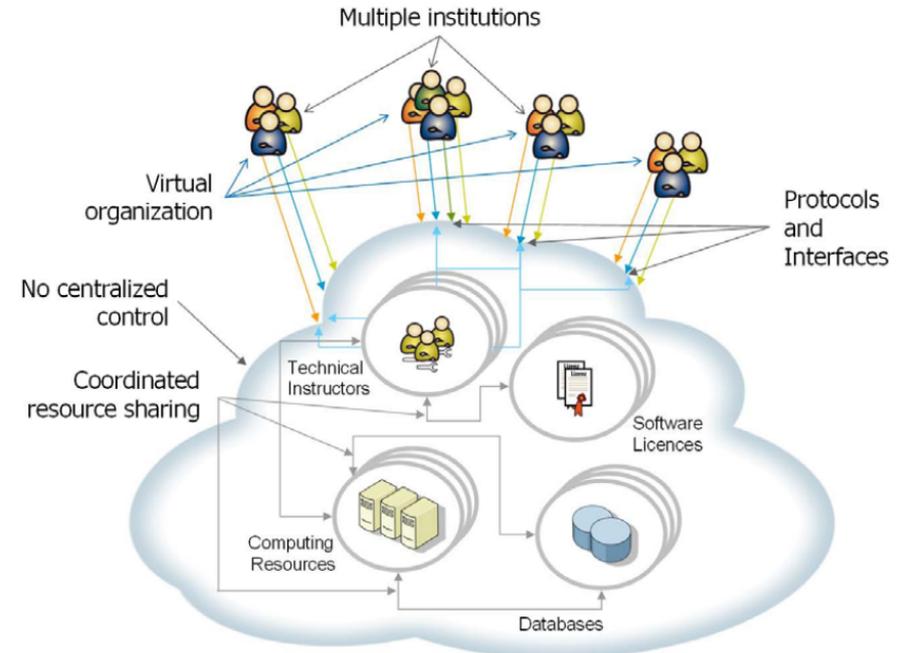


- 504 nós de computação B710 (*thin node*), onde cada nó possui 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
- 198 nós de computação B715 (*thin node*) com GPUs K40, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
- 54 nós de computação B715 (*thin node*) com XEON PHI, onde cada nó possui 2 x CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) e 64Gb de memória RAM
- 1 nó de computação MESCA 2 com memória compartilhada, 16 x CPU Intel Xeon Ivy Bridge (15c @2,4GHz) e 6TB de memória RAM
- 246 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 384Gb de memória RAM
- 36 nós computacionais (CPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252 e 768 Gb de memória RAM
- 94 nós computacionais (GPU), cada um com 2x Intel Xeon Cascade Lake Gold 6252, 4x NVIDIA Volta V100 GPU e 384Gb de memória RAM
- 1 nó para Inteligência Artificial com 2x Intel Xeon Skylake Gold 6148 (20c @2,4GHz), 8x NVIDIA Tesla V100-16GB com NVLink e 384Gb de memória RAM

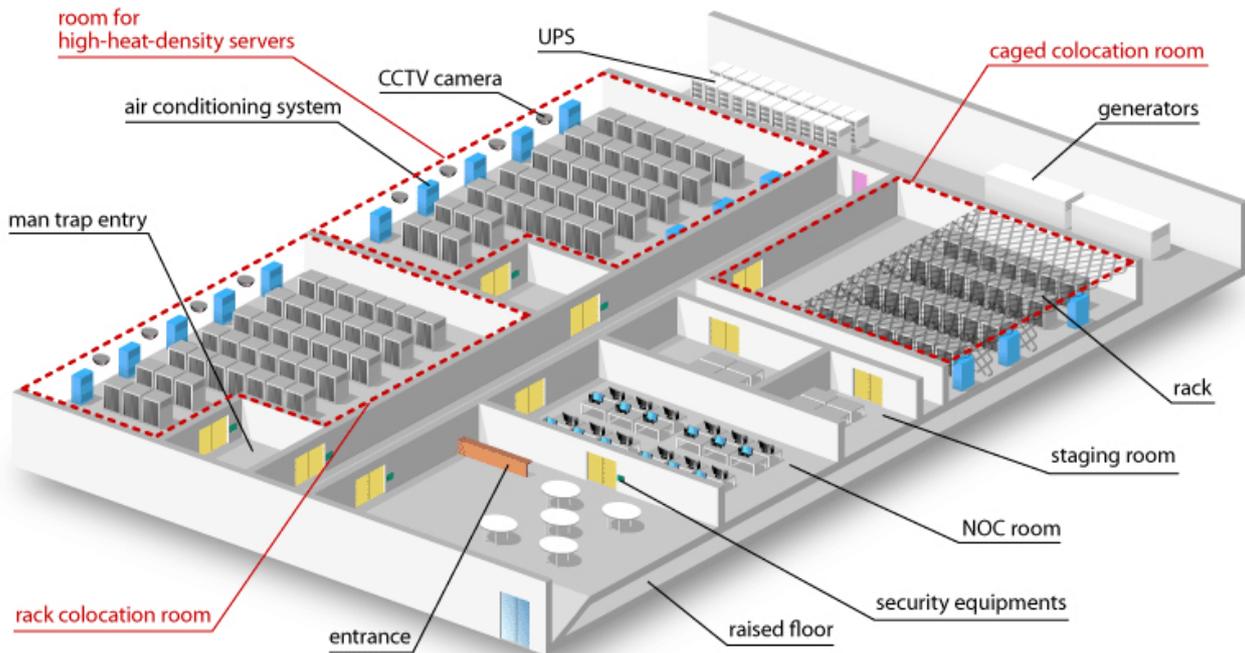
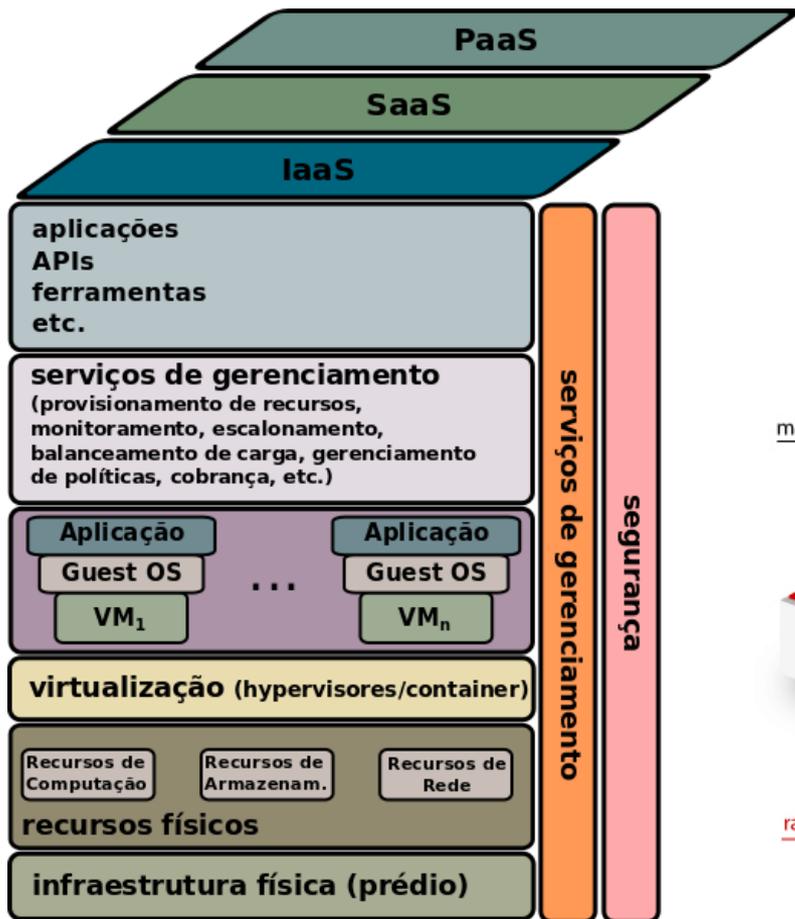
Grids



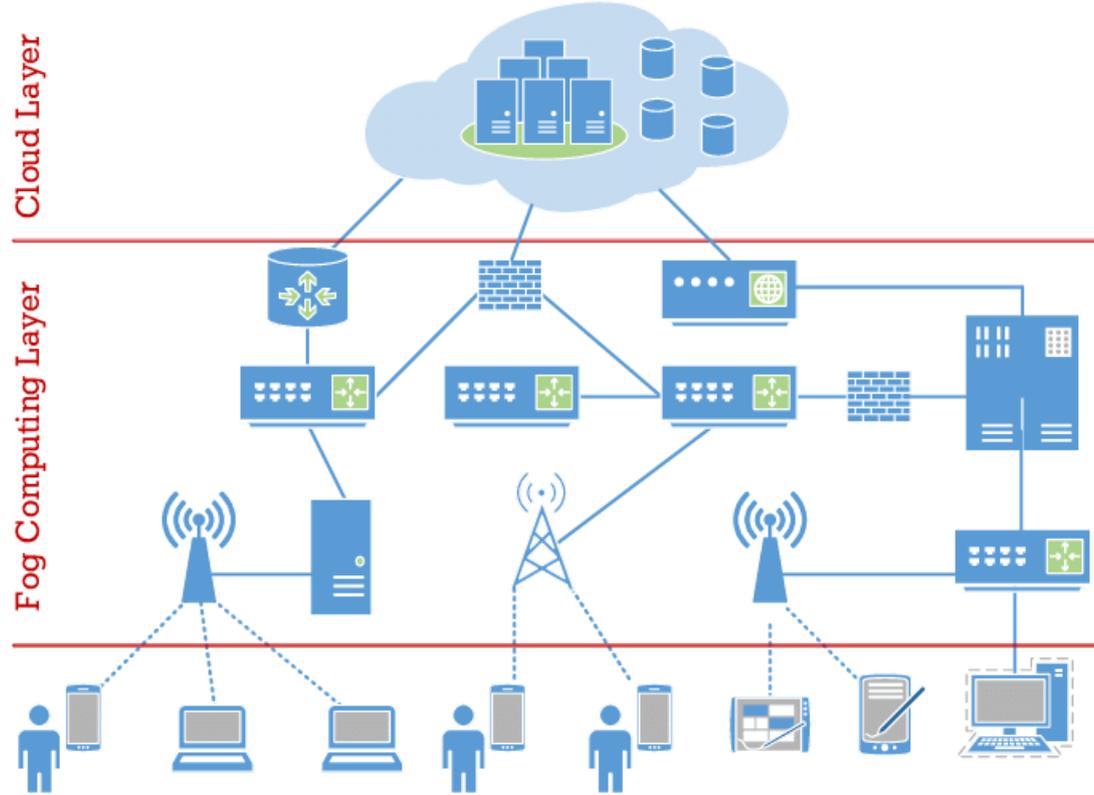
U. Schwiegelshohn et al. / Future Generation Computer Systems 26 (2010) 1104–1115



Cloud

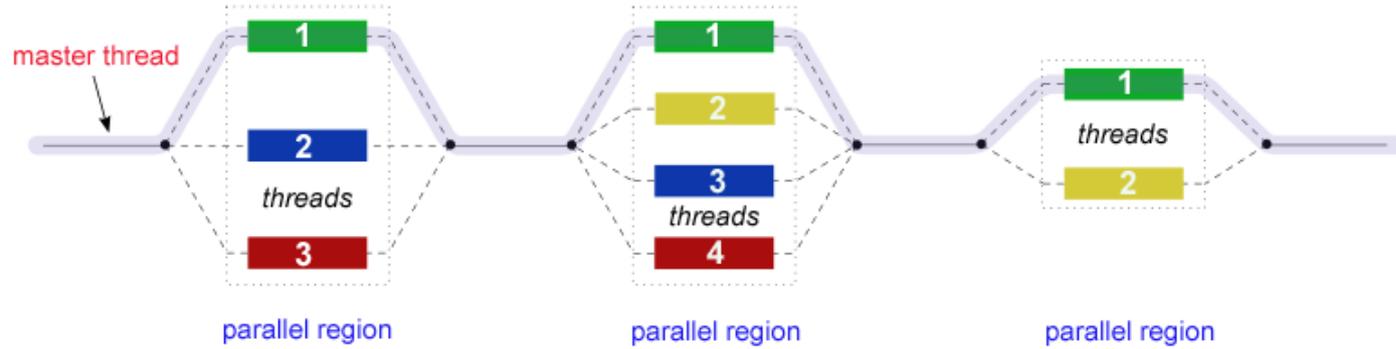


Fog



Modelos de Programação: OpenMP

- Baseia-se na criação de várias **threads** que compartilham o mesmo recurso de memória
- Modelo Fork-Join



O programa inicia como uma única thread, a master thread. Um conjunto de threads é criada no início de uma região paralela (fork) e unida no final (join).

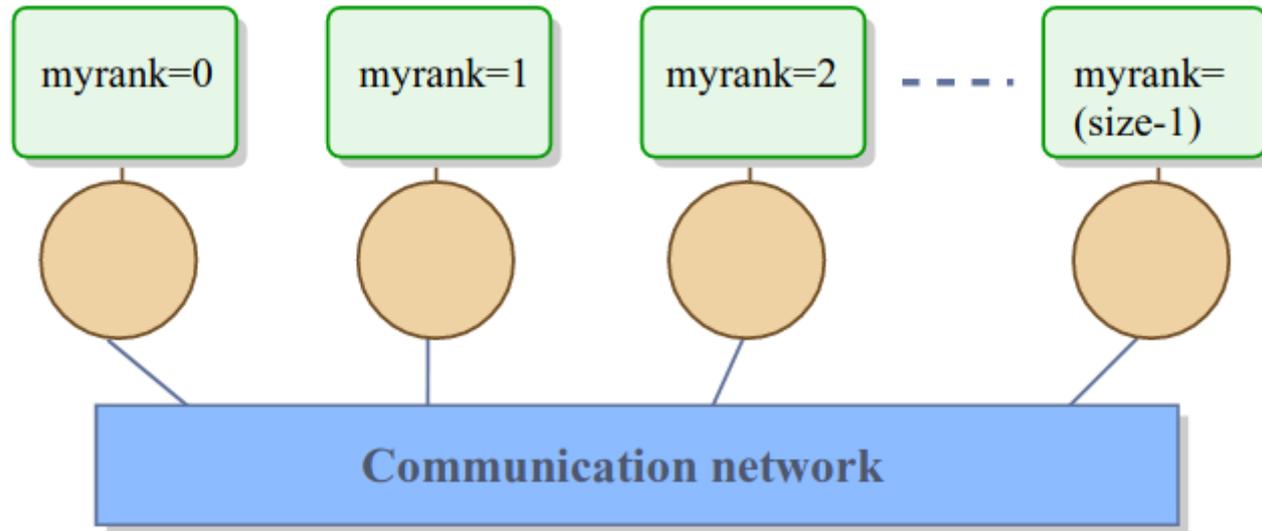
Modelos de Programação: OpenMP

```
#include <stdio.h>
#include <omp.h>

int main()
{
    #pragma omp parallel
    {
        printf("Hello, world!\n");
    }
    return 0;
}
```

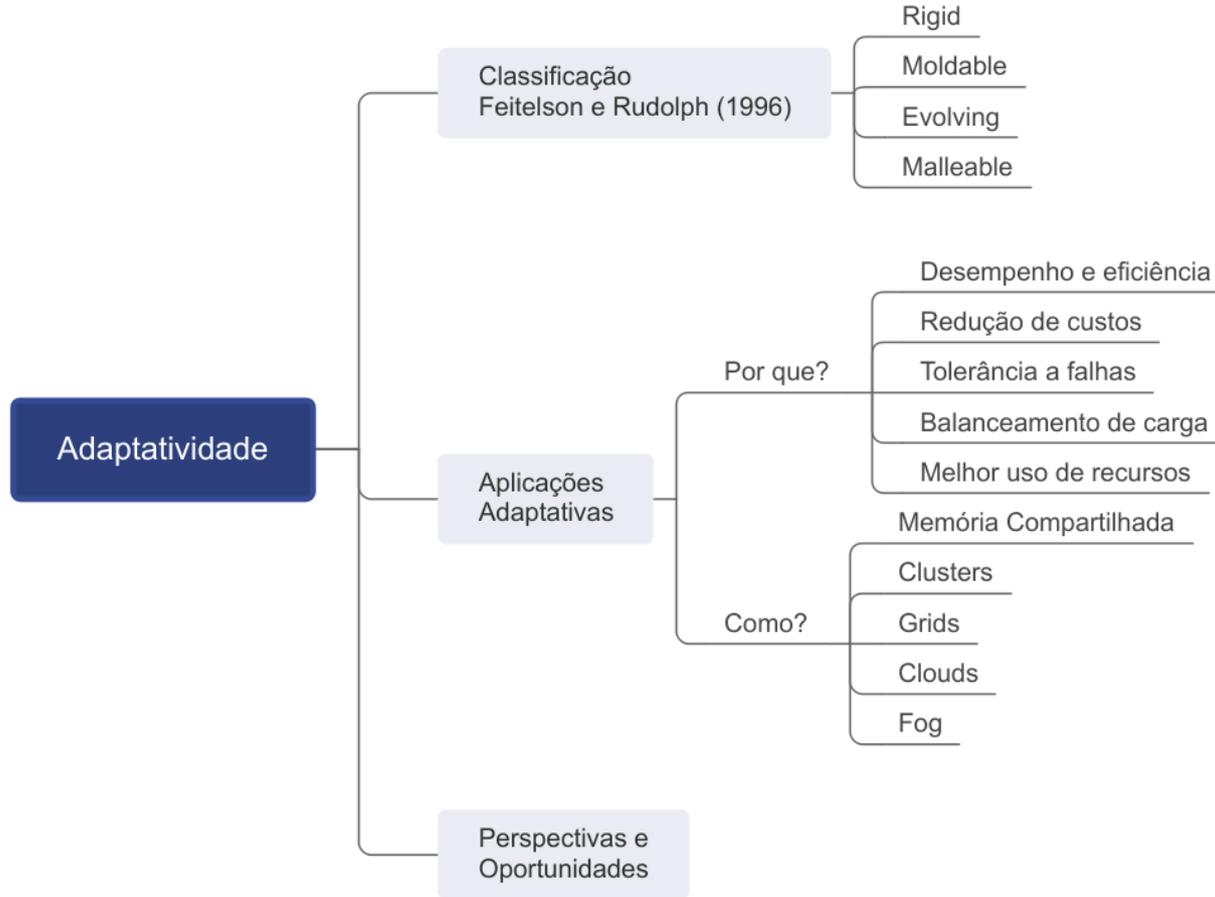
Modelos de Programação: MPI

- Memória distribuída
- Troca de mensagens



PARTE II – Aplicações Adaptativas

ROADMAP – Parte II



INTRODUÇÃO (Classificação quanto à adaptabilidade)

Feitelson e Rudolph (1996):

- Classificação das aplicações paralelas considerando a adaptabilidade
- 4 Classes de aplicações:
 - Rígidas
 - Moldáveis
 - Evolutivas (aplicação)
 - Maleáveis (sistema)

Adaptive

INTRODUÇÃO (Contexto e motivação)

- ▶ Aplicações adaptativas vem sendo desenvolvidas desde os anos 90
Não há trabalhos revisando o estado da arte cobrindo a adaptatividade de aplicações paralelas em arquiteturas paralelas
 - ▷ Memória Compartilhada
 - ▷ Clusters
 - ▷ Grids

- ▶ Para clouds a situação é diferente
 - ▷ Diversos surveys abordando a questão da elasticidade:
 - **Galante e Bona (2012)**
 - **Lorido-Botran et al. (2014)**
 - **Coutinho et al. (2015)**
 - **Al-Dhuraibi et al. (2018)**
 - **Kehrer e Blochinger (2019)**

Objetivo:

- Levantamento sobre adaptabilidade de recursos e de aplicações nos últimos 20 anos (2001 - 2021)
 - Memória compartilhada, clusters, grades e nuvens
- Principais iniciativas de pesquisa no campo de aplicações paralelas adaptativas
- Identificar lacunas e oportunidades de pesquisa nesta área
 - Possíveis trabalhos de IC e PG

É possível apontar diversas possibilidades ao usar recursos dinâmicos :

- Desempenho e eficiência
- Redução de custos
- Energia
- Tolerância a falhas
- Balanceamento de carga
- Melhor uso dos recursos

APLICAÇÕES ADAPTATIVAS (Como?)

- ▶ Para permitir que as aplicações utilizem eficientemente a alocação dinâmica de recursos são necessários frameworks e plataformas específicas
- ▶ Diversas soluções podem ser encontradas no estado da arte
 - ▷ Após analisar essas soluções, pode-se apontar quais são as principais abordagens usadas em cada uma das arquiteturas

Importante:

A adaptatividade deve ser suportada pelas arquiteturas e pelas aplicações!

APLICAÇÕES ADAPTATIVAS (Arquiteturas de Memória Compartilhada)

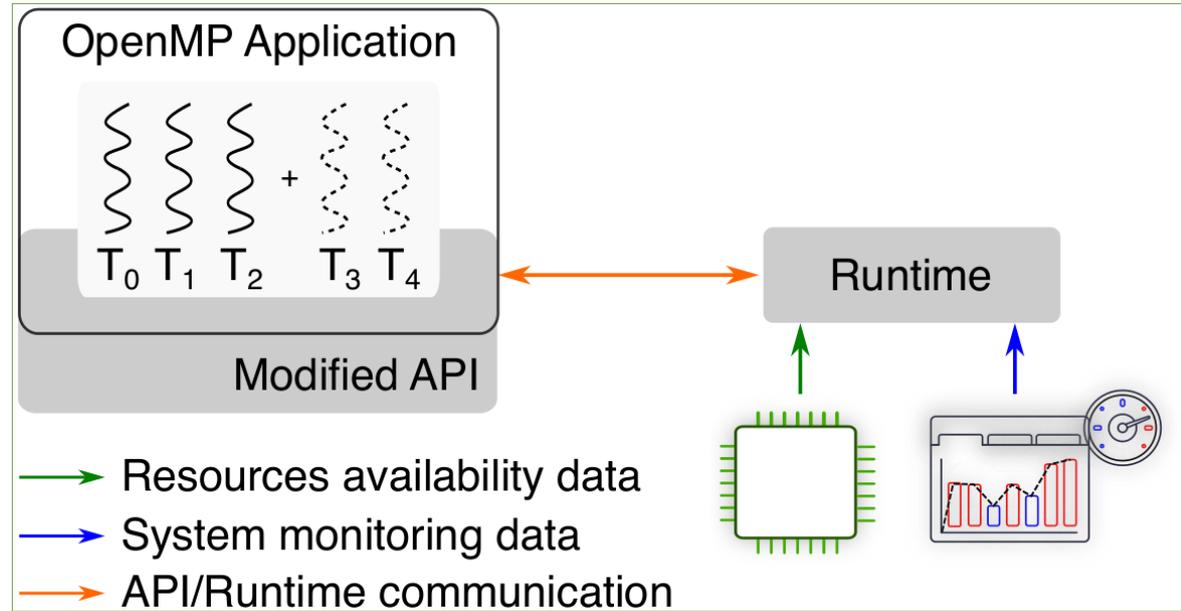
Foco: aplicações multithread

Como a adaptabilidade é explorada:

Ajuste do número de threads

- A maioria das soluções é baseada em uma biblioteca e runtime/middleware
- Em geral, o uso desses frameworks exige algumas modificações no código-fonte original e, em alguns casos, nenhuma alteração é necessária se a biblioteca modificada fizer o trabalho pesado.

Objetivo: melhorar o uso de recursos (compartilhamento), desempenho ou eficiência

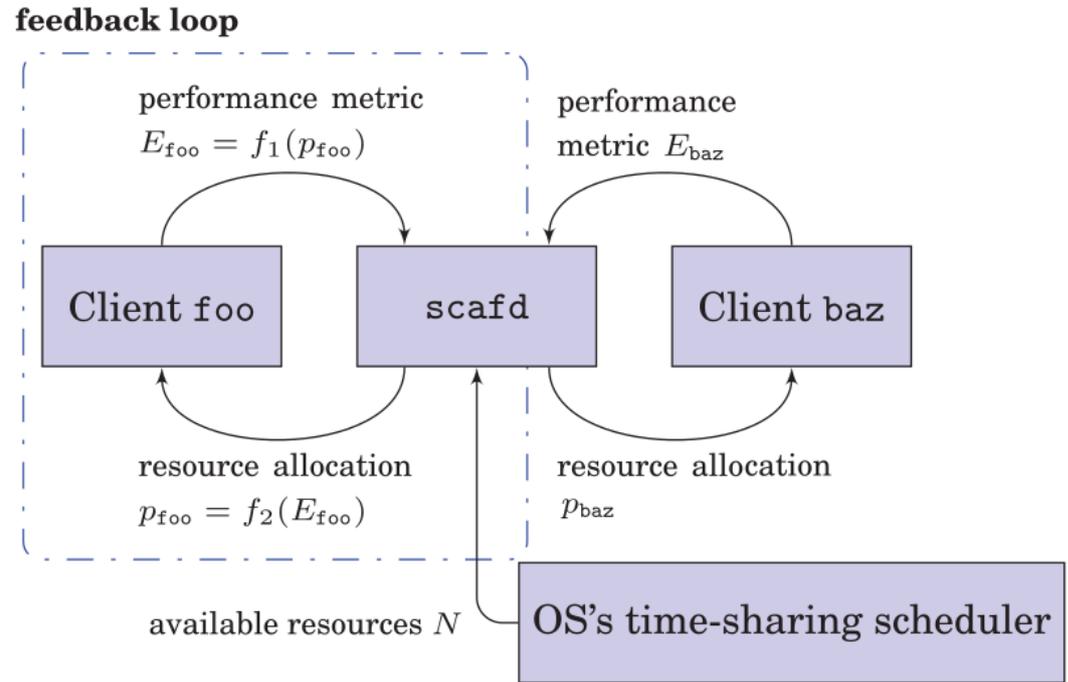


APLICAÇÕES ADAPTATIVAS (Arquiteturas de Memória Compartilhada)

Exemplo:

SCAF – Creech e Barua (2016)

- ▶ Aplicações OpenMP
- ▶ Alocação de recursos baseada na eficiência
- ▶ Calcula no início de uma seção paralela a eficiência de cada aplicação para estimar a alocação de recursos
- ▶ O objetivo geral é melhorar a soma de speedups das aplicações que compartilham recursos de máquina



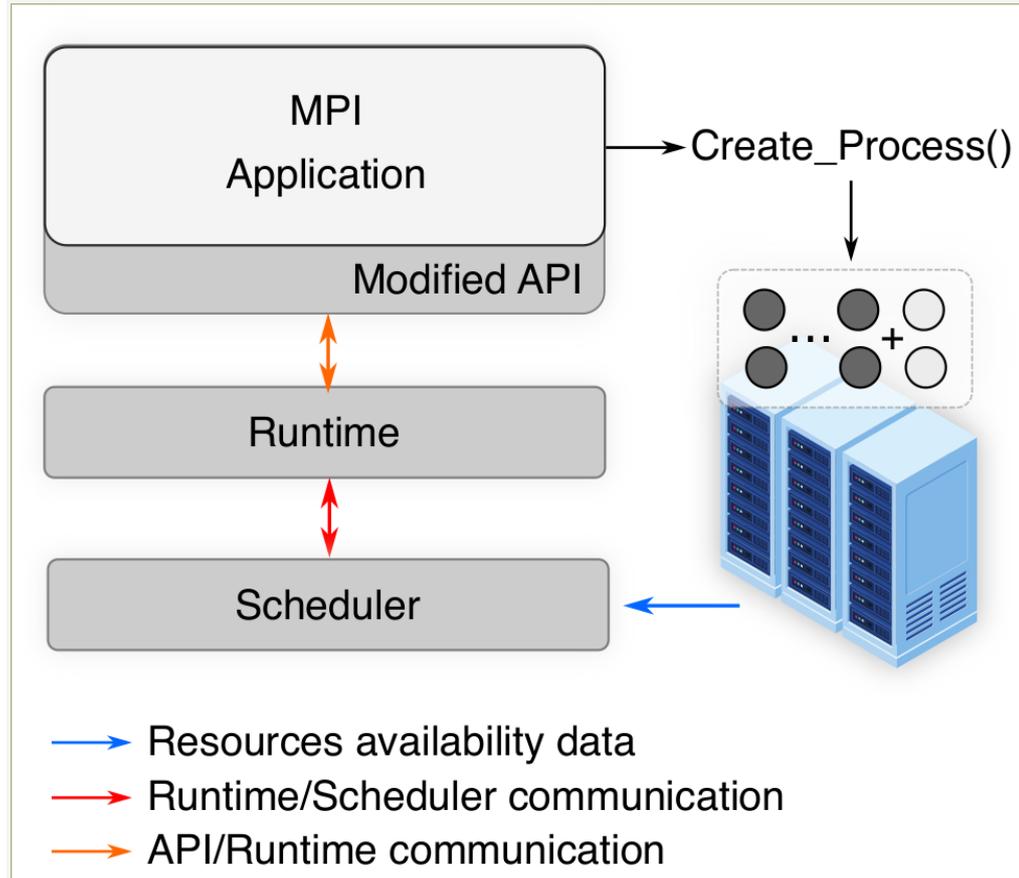
APLICAÇÕES ADAPTATIVAS (Clusters)

Foco: aplicações MPI

Como a adaptabilidade é explorada: Aumento ou redução do número de processos de acordo com as variações na disponibilidade de recursos

- As operações de maleabilidade são tratadas por uma API que interage com um runtime (ou escalonador) para obter informações sobre disponibilidade de recursos ou desempenho do aplicativo
- Suporte do escalonador é importante

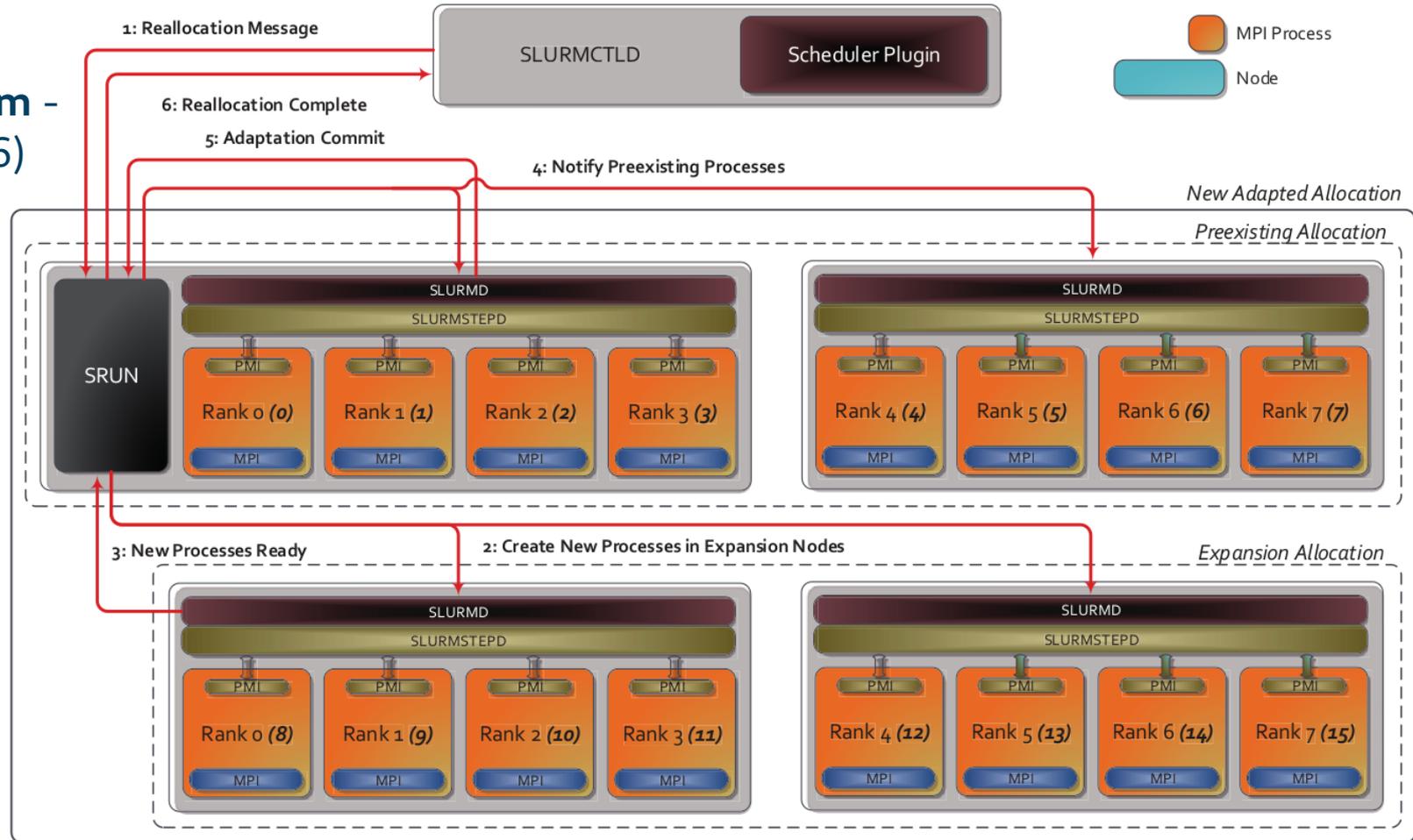
Objetivo: melhorar o uso de recursos, desempenho, balanceamento de carga, tolerância a falhas



APLICAÇÕES ADAPTATIVAS (Clusters)

Exemplo:

Extended MPI/Slurm -
Comprés et al (2016)



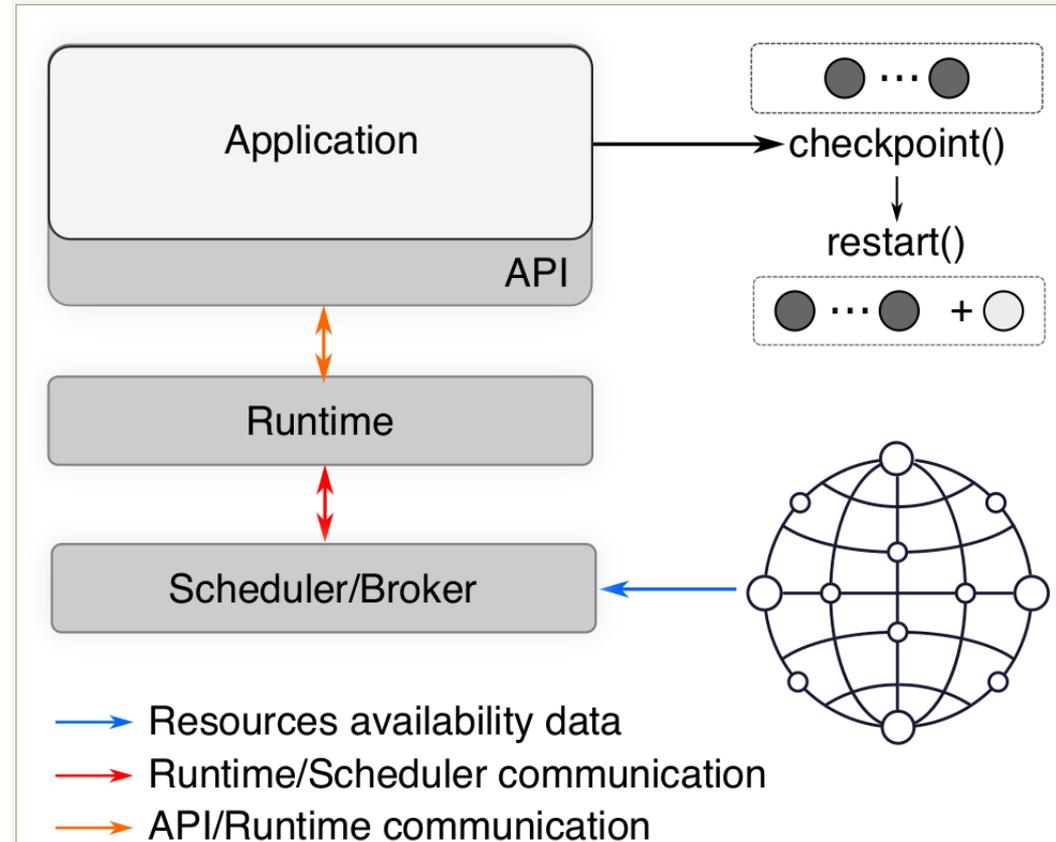
APLICAÇÕES ADAPTATIVAS (Grids)

Foco: aplicações fracamente acopladas

Como a adaptabilidade é explorada: A maioria dos frameworks são baseados em técnicas de checkpoint/restart

- Os aplicativos são checkpointed e reiniciados e com um conjunto diferente de recursos
- A principal vantagem do checkpointing é que é uma técnica muito geral que pode ser aplicada a aplicações paralelas.

Objetivo: melhorar a robustez, garantir um certo nível de QoS, melhorar o desempenho, fornecer tolerância a falhas e usar recursos de forma eficiente

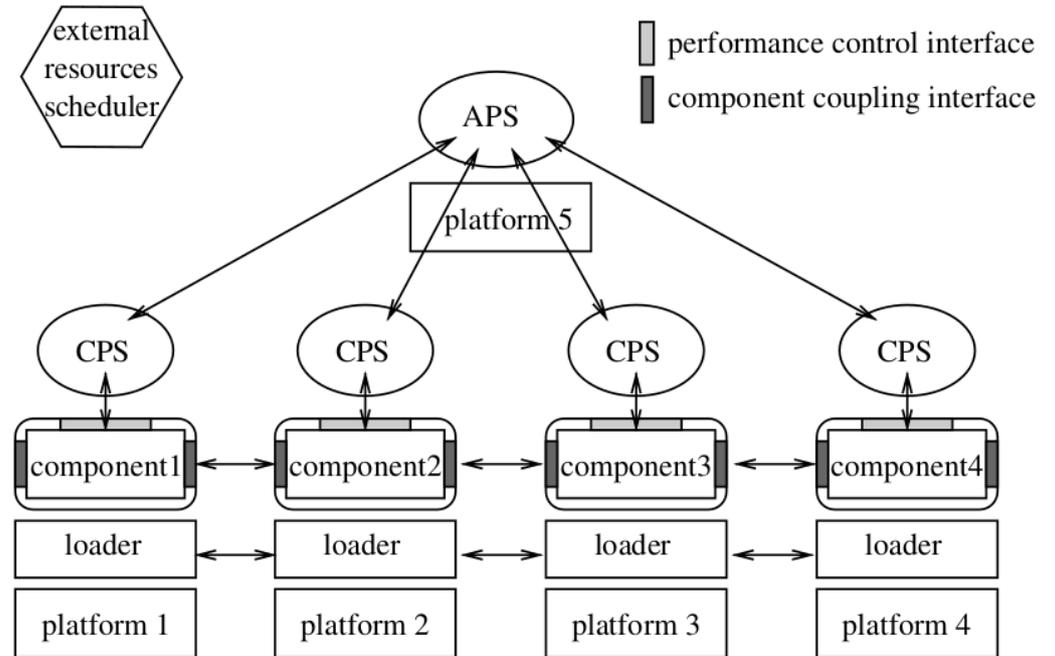


APLICAÇÕES ADAPTATIVAS (Grids)

Exemplo:

PerCo – Mayes et al (2010)

- ▶ O alvo são aplicações fracamente acopladas que podem ser realocadas e reiniciadas com um número diferente de processos.
- ▶ O sistema monitora o progresso das aplicações e as realoca para otimizar o desempenho



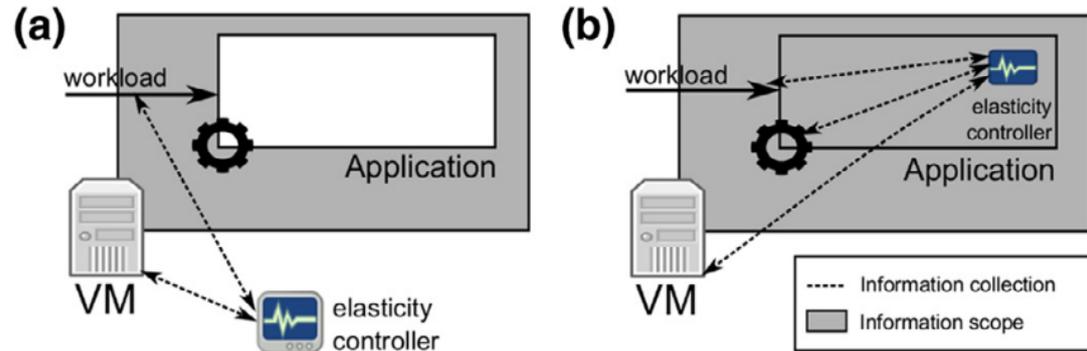
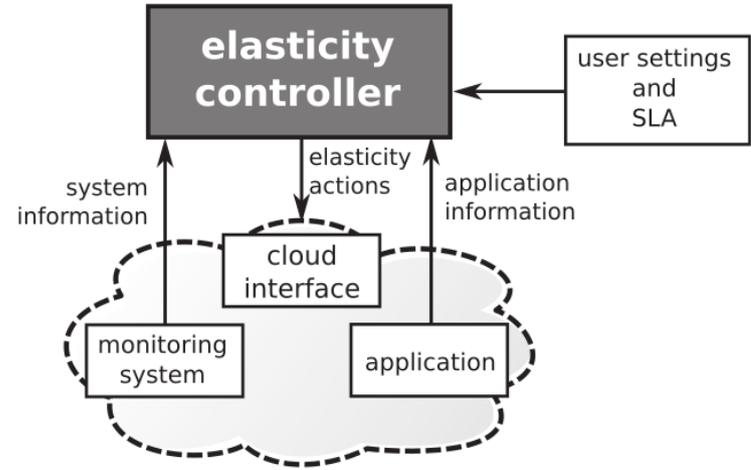
APLICAÇÕES ADAPTATIVAS (Clouds)

Foco: aplicações com diferentes características

Como a adaptabilidade é explorada:
elasticidade vertical e horizontal

- ▶ Sistema de monitoramento que age sob certas circunstâncias
- ▶ Linguagens/API para exploração da elasticidade
- ▶ Pode agregar ainda as soluções apresentadas para as outras arquiteturas

Objetivo: melhorar o desempenho, fornecer tolerância a falhas, balanceamento de carga, uso recursos de forma eficiente e custo

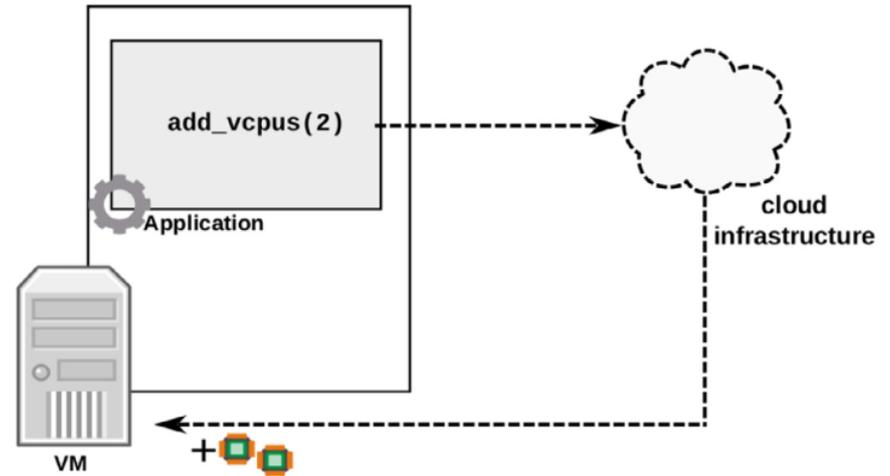


APLICAÇÕES ADAPTATIVAS (Clouds)

Exemplo:

Cloudine - Galante e Bona (2015)

- Primitivas de elasticidade
- Possibilidade de programar as ações de elasticidade



APLICAÇÕES ADAPTATIVAS (Fog)

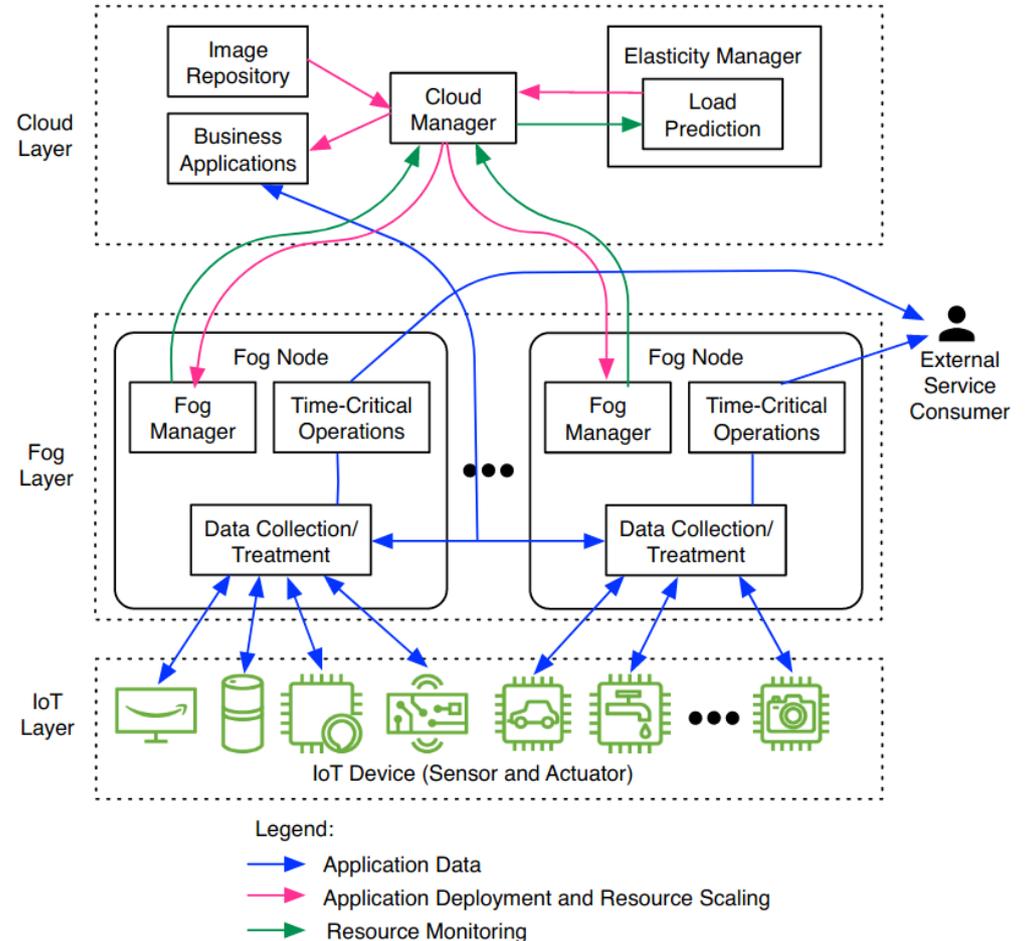
Foco: IoT

Como a adaptabilidade é explorada: alocação de recursos na nuvem/cloud

Objetivo: redução de custos, desempenho

Exemplo: ElasticFog (Nguyen et al. 2020)

Alocação dinâmica de recursos de aplicações baseados em contêiner em Fog Computing



É possível apontar algumas tendências e oportunidades de pesquisa:

- Soluções baseadas em IA
- Combinação de abordagens de exploração da elasticidade em nuvens
- Uso eficiente dos recursos em arquiteturas heterogêneas
- Exploração transparente e com pouco esforço do paralelismo e adaptatividade
- Mecanismos para redistribuição de dados/carga de trabalho

PARTE III – Exemplos de Aplicações Adaptativas

Exemplos de Aplicações Adaptativas

▸ Scalable Assembler at Notre Dame – SAND

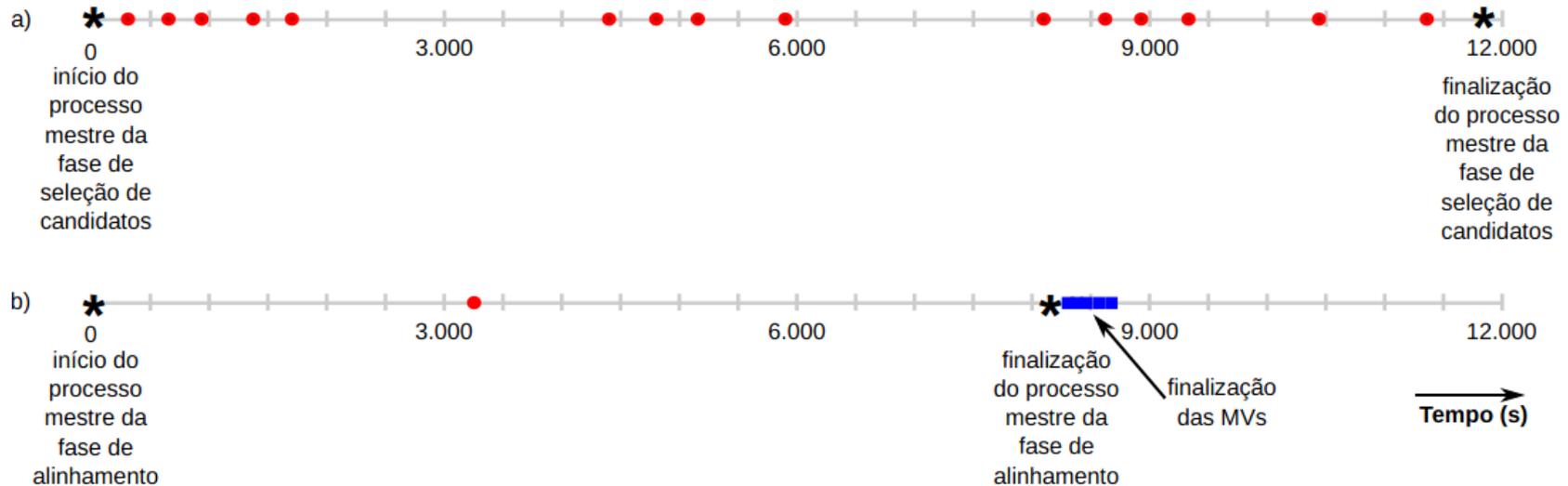
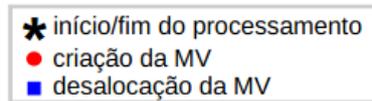
- O SAND é um conjunto de módulos para a montagem do genoma
 - Duas fases principais: seleção de candidatos e alinhamento de sequências
 - Mestre-Escravo

▹ Work Queue + Cloudine

```
mestre_SAND()  
...  
enquanto(1):  
    se recursos_livres  
        clne_add_node(1)  
        //aguarda a inicializacao da MV  
  
        upload_trabalhador_MV  
        inicia_trabalhador_MV  
    ...
```

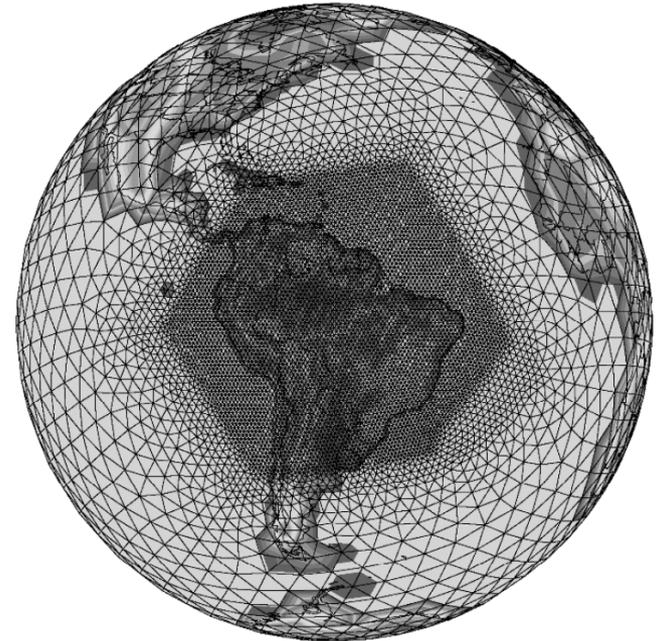
Exemplos de Aplicações Adaptativas

▸ Scalable Assembler at Notre Dame – SAND



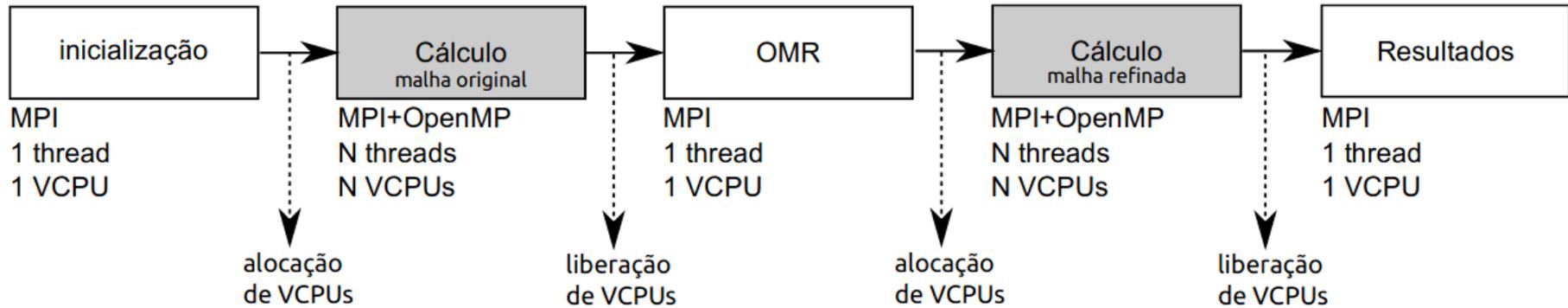
Exemplos de Aplicações Adaptativas

- Ocean-Land-Atmosphere Model – OLAM
 - ▾ Modelo de simulação numérica para a climatologia
 - ▾ Implementação MPI+OpenMP
 - ▾ Decomposição de domínio
 - ▾ Cloudine



Exemplos de Aplicações Adaptativas

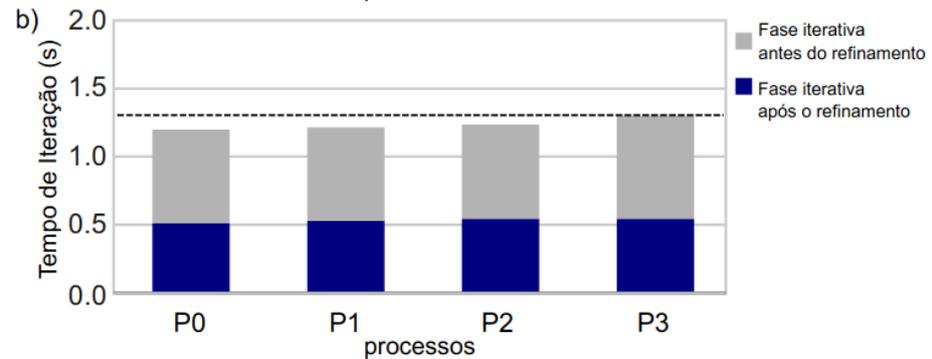
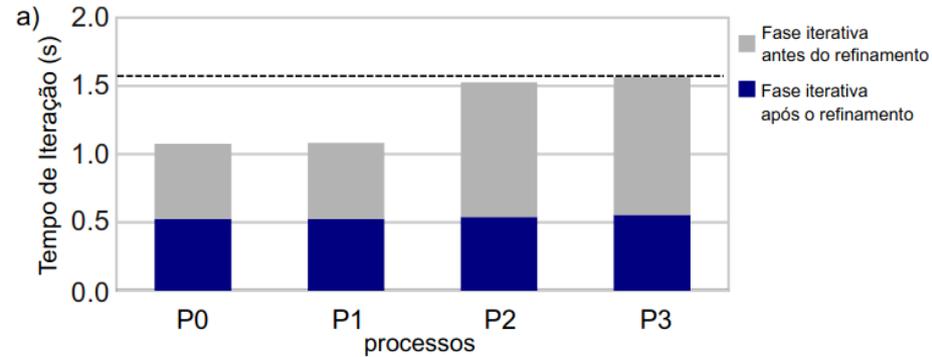
▸ Ocean-Land-Atmosphere Model – OLAM



Exemplos de Aplicações Adaptativas

► Ocean-Land-Atmosphere Model – OLAM

Processo	Elementos antes do OMR	Elementos após OMR	Diferença em relação à P0
P0	50.556	50.556	–
P1	50.812	50.812	0,5%
P2	51.110	88.875	75%
P3	50.886	85.143	68%

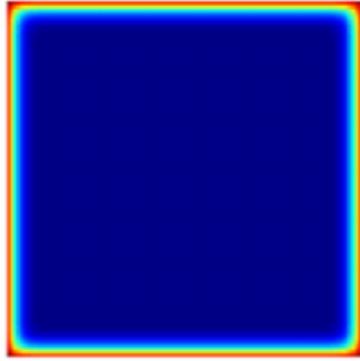


Exemplos de Aplicações Adaptativas

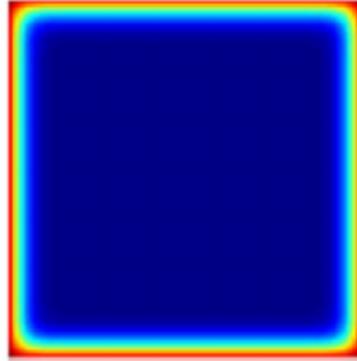
- ▶ Transferência de Calor – Elastic OpenMP



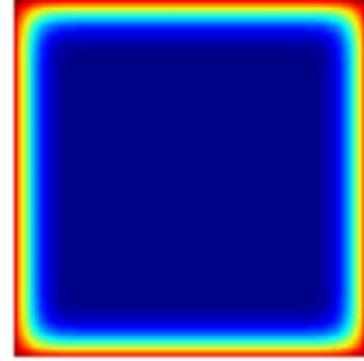
passo 1



passo 60



passo 120

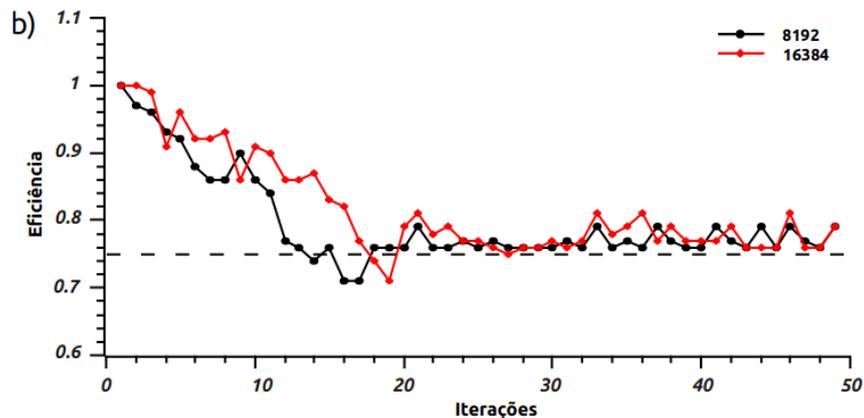
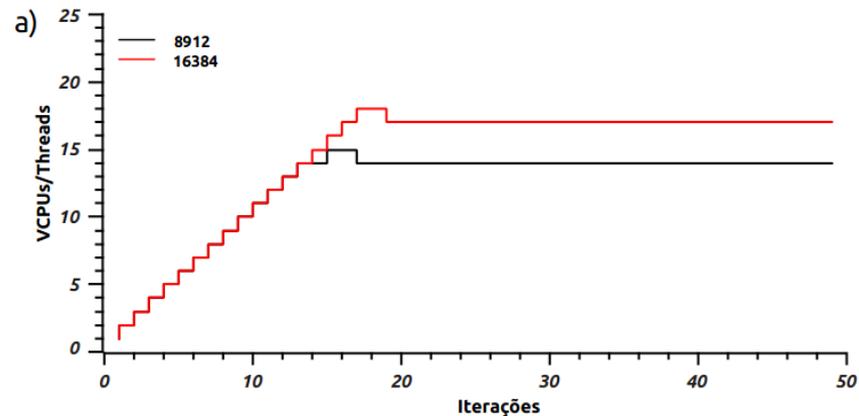


passo 180

Exemplos de Aplicações Adaptativas

► Transferência de Calor – Elastic OpenMP

```
transferência_calor()  
    ...  
    para cada iteração:  
        calcula_eficiencia()  
  
        se eficiencia >= 0.75  
        | threads++ //OpenMP automaticamente adiciona VCPU  
  
        se eficiencia < 0.75 por duas iterações consecutivas  
        | threads-- //OpenMP automaticamente remove VCPU  
  
        calcula_calor_OpenMP(threads)
```



CONSIDERAÇÕES FINAIS

- Adaptatividade pode levar a melhorias significativas no desempenho e na eficiência, redução de custos, incorporação de tolerância a falhas e balanceamento de carga e melhor utilização dos recursos
- Embora a adaptabilidade seja uma área que tem sido pesquisada há mais de 20 anos, ainda há alguns desafios de pesquisa e questões em aberto para serem abordadas
 - O santo graal é a disponibilidade de mecanismos transparentes que exploram com eficiência a adaptabilidade em diferentes modelos de aplicações
 - Isso pode ser conseguido usando novas linguagens de domínio específico e provavelmente, com a ajuda de inteligência artificial

OBRIGADO!

APLICAÇÕES ADAPTATIVAS EM ARQUITETURAS PARALELAS

Guilherme Galante
guilherme.galante@unioeste.br

Rodrigo da Rosa Righi
rrrighi@unisinov.br