Estudo Comparativo de Algoritmos de Escalonamento para Grades Computacionais

Alvaro Henry Mamani Aliaga e Alfredo Goldman

Instituto de Matemática e Estatística
Departamento de Ciência da Computação
Universidade de São Paulo

{alvaroma,gold}@ime.usp.br

II Escola Regional de Alto Desempenho de São Paulo

Julho, 2011

Roteiro

Introdução

- Necessidade de poder computacional: mineração de dados, previsão do tempo, processamento de imagens médicas, ...
- Aumento na disponibilidade de computadores poderosos e na interligação de redes de alta velocidade
- Computação em grade

  Uma alternativa para obter grande capacidade processamento
- Escalonamento de tarefas consiste em alocar tarefas de uma aplicação em recursos computacionais, com o intuito de minimizar o *Makespan*
- Aplicações com tarefas dependentes são modeladas mediante um grafo acíclico dirigido, $G = (V, E)$, onde, G representa o DAG, V o conjunto de tarefas t e E o conjunto de dependências entre cada tarefa da aplicação

HEFT, *Heterogeneous Earliest Finish Time*

Priorização de tarefas

- Atribuir prioridade às tarefas
- Calculo da prioridade, baseado na média dos custos de computação e custos de comunicação
- lista das tarefas

Seleção de recursos

- Selecionar a tarefa t_i da lista com maior prioridade
- Para cada recurso $r \in R$ é calculado o *EST* e *EFT* de cada tarefa t_i
- r_j é alocada ao recurso que minimiza o EFT da tarefa t_i

Topcuouglu, Haluk et Al., Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing, IEEE Trans. Parallel Distrib. Syst., 2002.

CPOP, *Critical Path On a Processor*

Priorização de tarefas

- Atribuir prioridade às tarefas
- Calculo das prioridades baseados no custo de computação e comunicação
- $|CP|$ é o caminho crítico

Seleção de recursos

- *PCP* (*critical-path processor*)
- Se a tarefa selecionada está no caminho crítico, então é escalonada no recurso de caminho crítico
- ela é atribuída a um recurso que minimiza o EFT

Topcuouglu, Haluk et Al., Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing, IEEE Trans. Parallel Distrib. Syst., 2002.

PCH, *Path Clustering Heuristic*

Seleção de tarefas e agrupamento

- seleciona tarefas que formarão cada *cluster* que serão escalonadas no mesmo recurso
- A primeira tarefa que compõe um *cluster* cls_k é a tarefa não escalonada com maior prioridade

Seleção de recursos

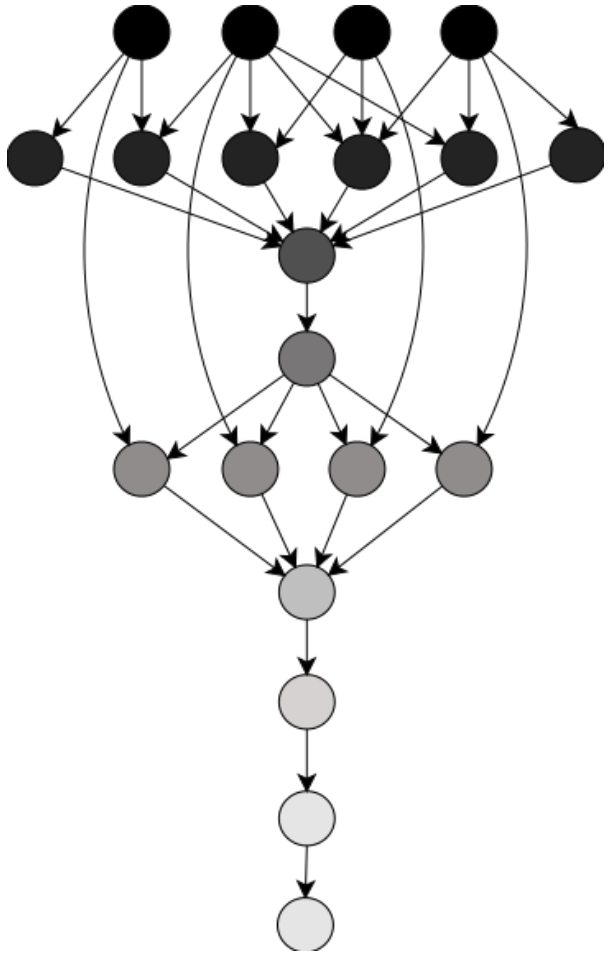
- A seleção de recursos se dá através do cálculo de valores
- qual recurso terminará a execução do *cluster* em menor tempo
- O fator que determina em qual recurso um *cluster* será escalonado é o *EST* do sucessor da último tarefa do *cluster* considerado

Bittencourt, Luiz F et Al., Uma Heurística de Agrupamento de Caminhos para Escalonamento de Tarefas em Grades Computacionais, SBRC, 2006.

Aplicações

Montage

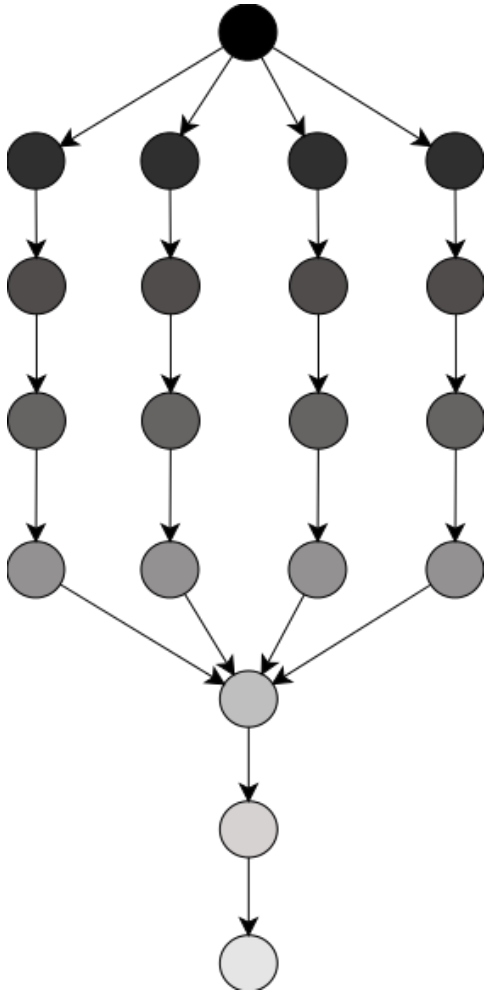
- É usada para gerar mosaicos personalizados do céu usando pontos de múltiplas imagens de entrada.



Aplicações

Epigenomics

- É usada no mapeamento do estado epigenético de células humanas sobre uma grande escala genômica.



Arquitetura

Características da Arquitetura

- Foram especificados dois aglomerados, com duas instâncias: homogênea e heterogênea

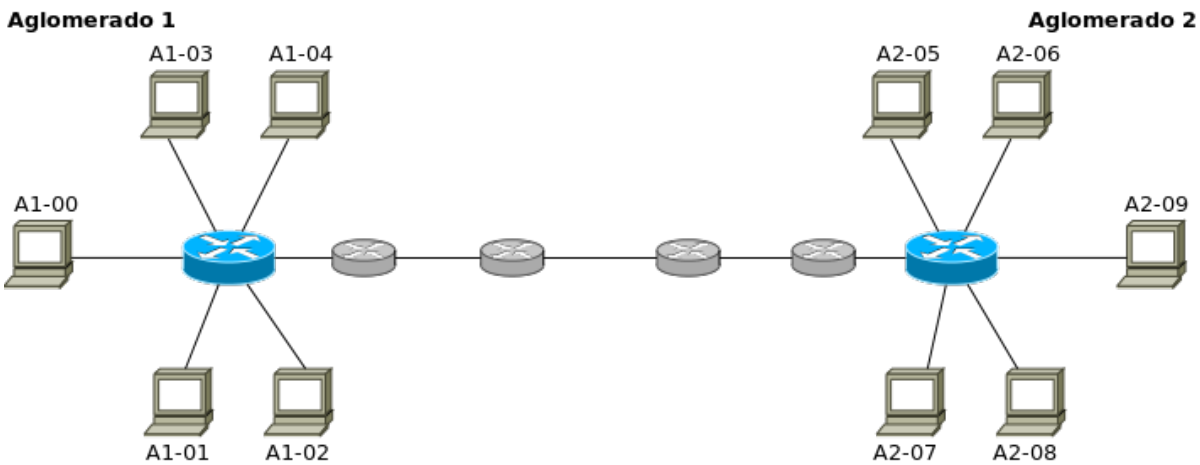


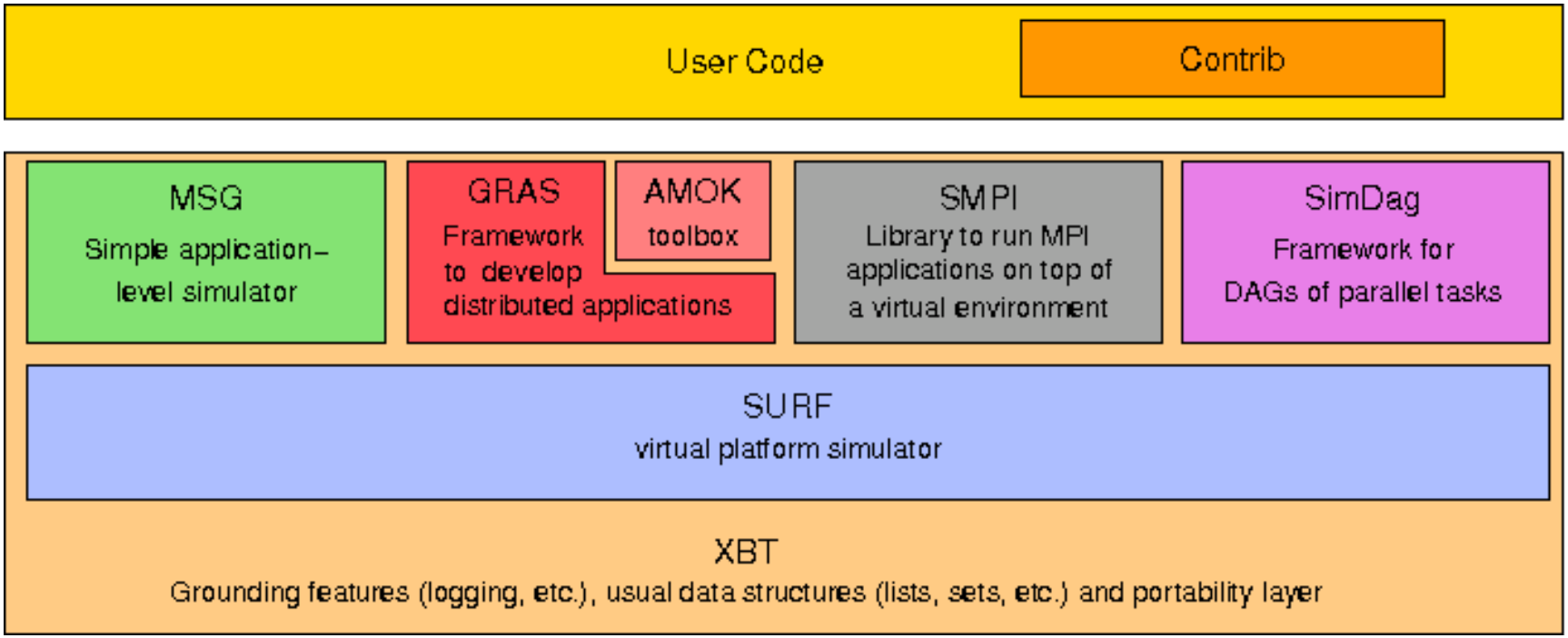
Figura: Dois aglomerados usados na simulação.

Arquitetura

Id	Poder Computacional (GFlops)	
	Homogêneo	Heterogêneo
A1-00	5,00	1,00
A1-01	5,00	2,00
A1-02	5,00	3,00
A1-03	5,00	4,00
A1-04	5,00	5,00
A2-05	5,00	6,00
A2-06	5,00	7,00
A2-07	5,00	8,00
A2-08	5,00	9,00
A2-09	5,00	9,00

Tabela: Poder computacional dos recursos.

Simulador SimGrid



Casanova, Henri and Legrand, Arnaud and Quinson, Martin, SimGrid: a Generic Framework for Large-Scale Distributed Experiments, IEEE Computer Society, 2008.

Aplicação Montage - Arquitetura Homogênea

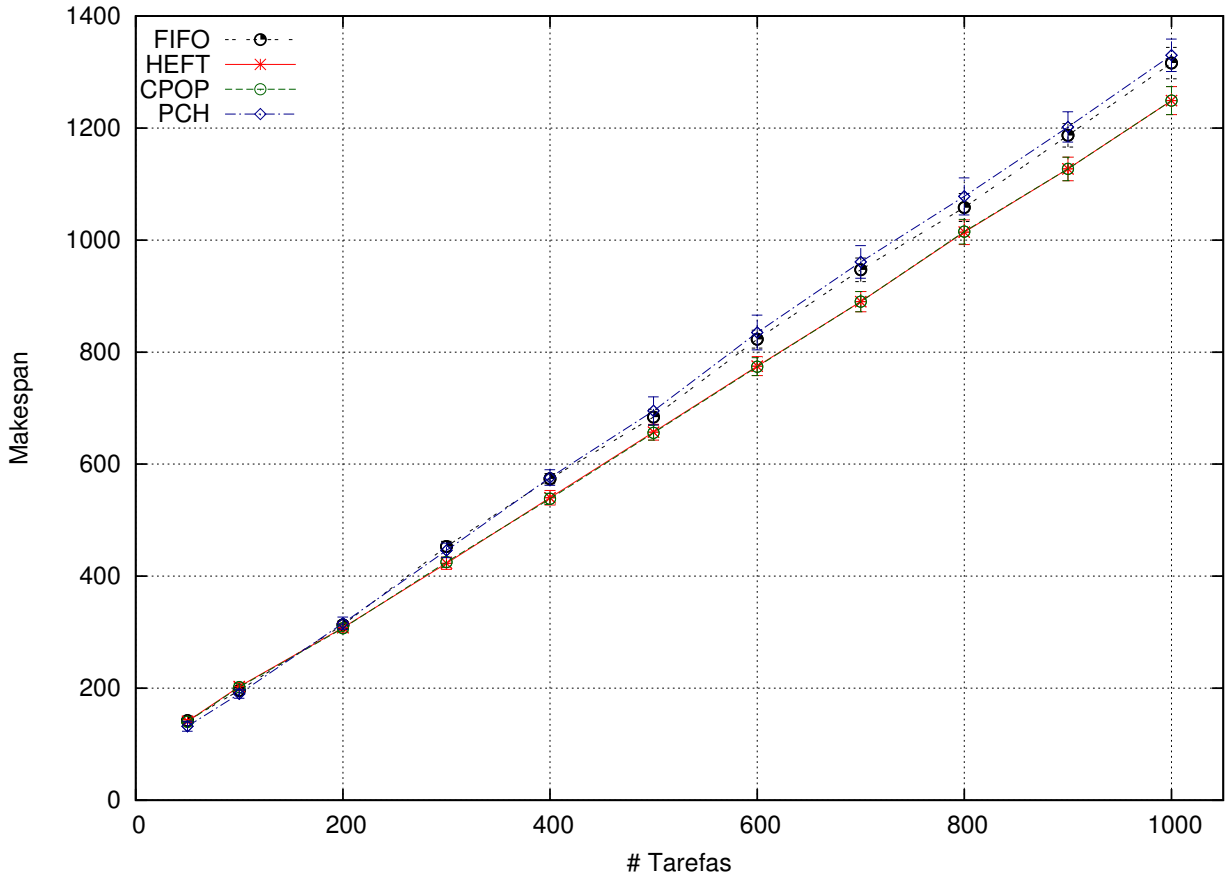


Figura: Resultado da simulação do *Workflow Montage*.

Aplicação Montage - Arquitetura Heterogênea

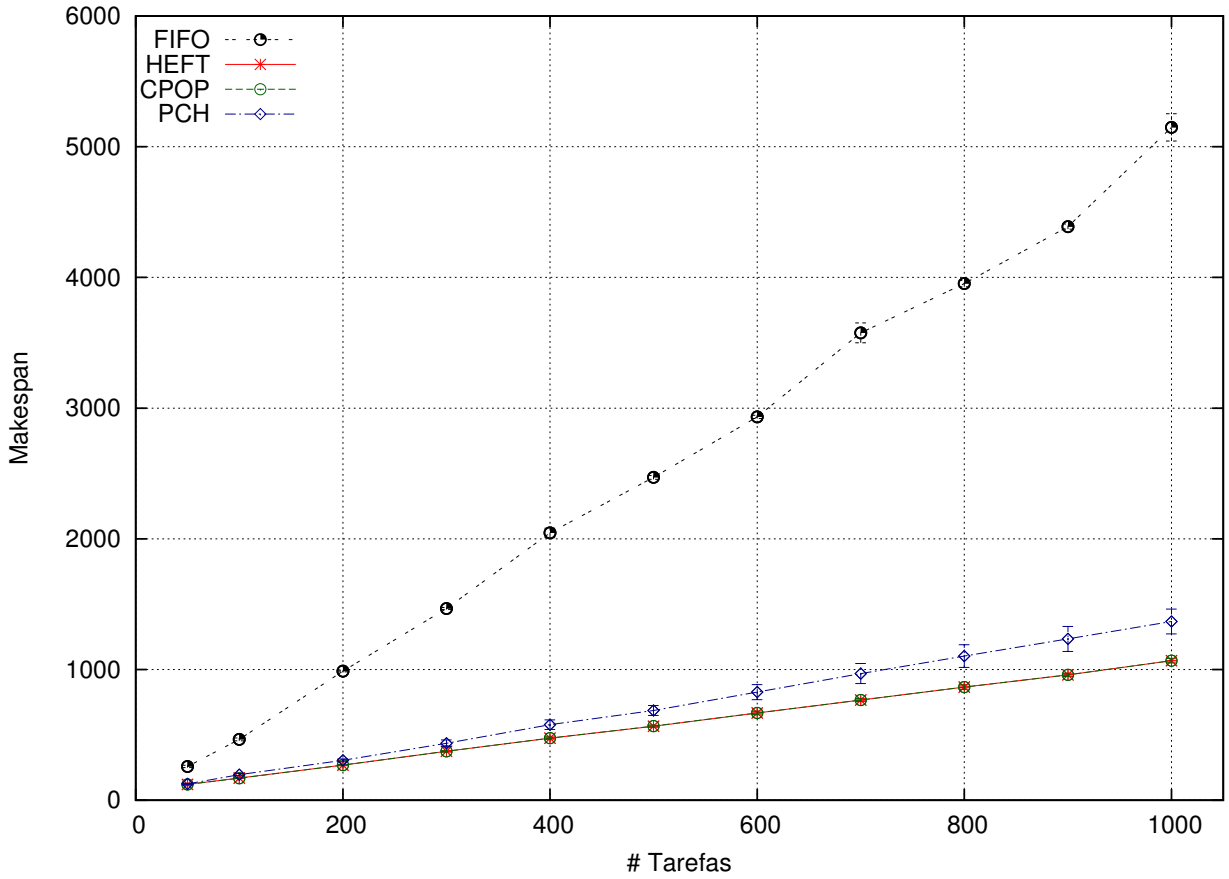


Figura: Resultado da simulação do *Workflow Montage*.

Aplicação Montage

- Observamos que a estratégia FIFO somente apresenta bom desempenho na arquitetura homogênea
- O algoritmo *PCH* não oferece bom desempenho em ambas arquiteturas
- Os algoritmos *HEFT* e o *CPOP* apresentam bom desempenho em ambas das arquiteturas com uma diferença quase desprezível

Aplicação Epigenomics - Arquitetura Homogênea

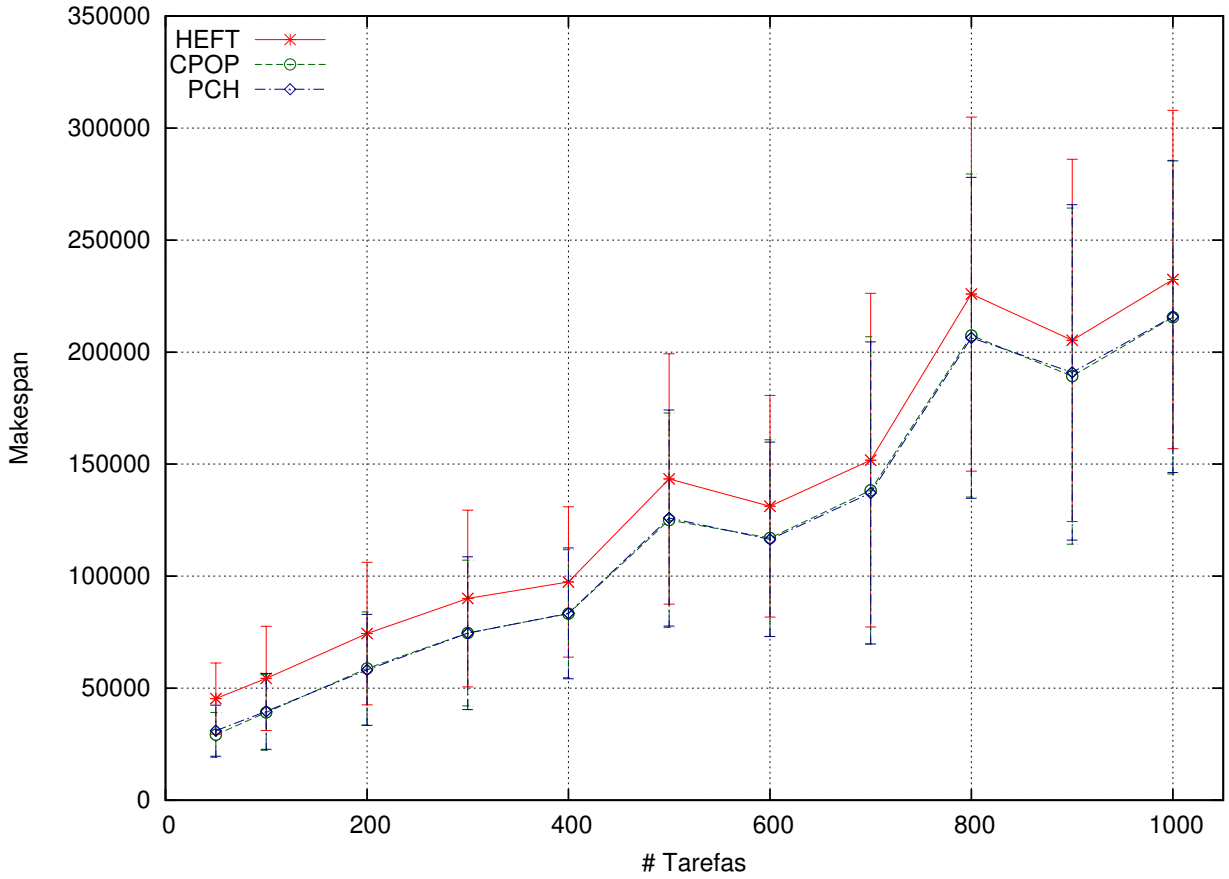


Figura: Resultado da simulação do *Workflow Epigenomics*.

Aplicação Epigenomics - Arquitetura Heterogênea

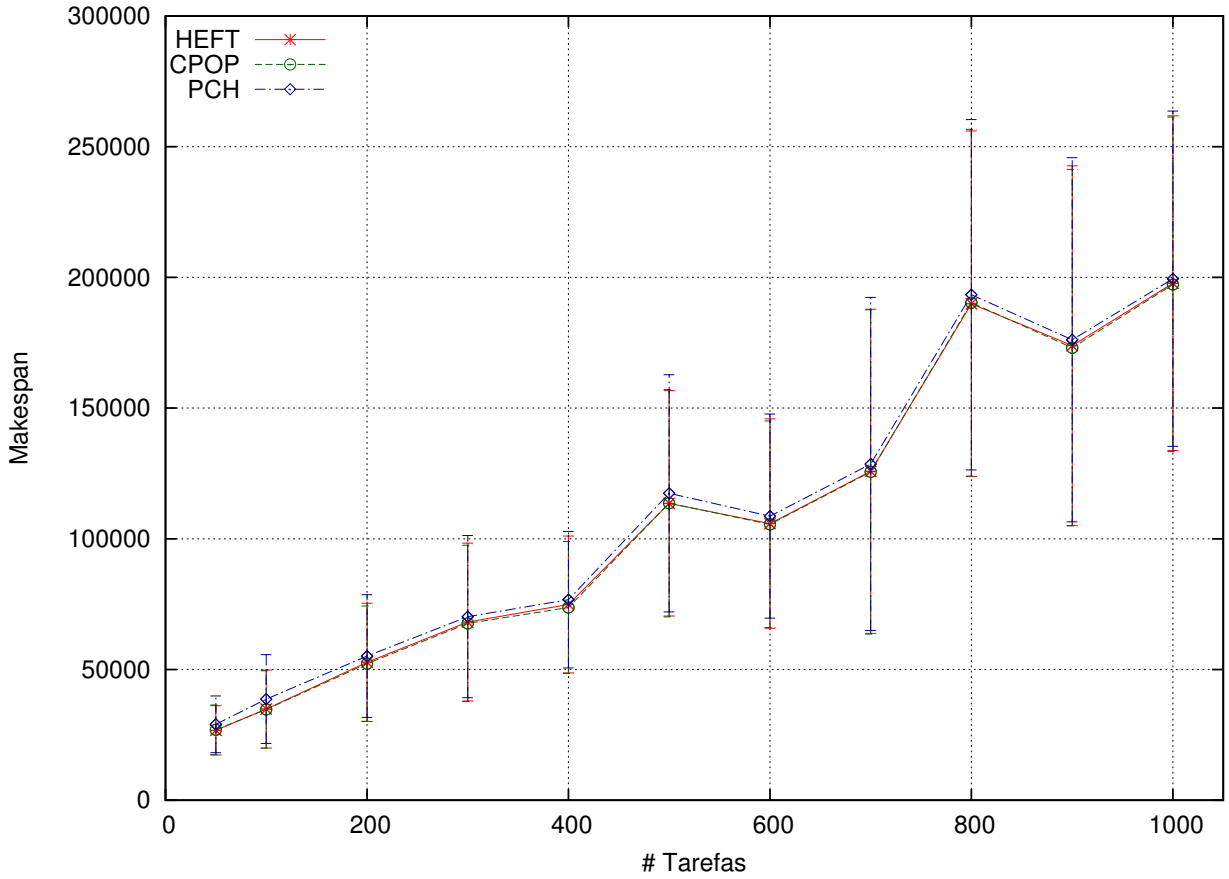


Figura: Resultado da simulação do *Workflow Epigenomics*.

Aplicação Epigenomics

- O algoritmo *PCH* melhora o desempenho neste tipo de aplicação.
- O algoritmo *CPOP* conserva o bom desempenho em ambas das arquiteturas.
- O algoritmo *HEFT* apresenta bom desempenho na arquitetura heterogênea, mas na arquitetura homogênea não

Conclusões

- Uma estratégia de tipo *FIFO* é adequada somente em arquiteturas homogêneas
- Em arquiteturas heterogêneas é preciso um estudo tanto dos recursos computacionais quanto das tarefas
- Os algoritmos HEFT e CPOP apresentaram desempenhos bons e ao mesmo tempo similares na aplicação Montage, enquanto o PCH não
- Em aplicações que possuem gargalos, como a aplicação Montage, não é conveniente o agrupamento de tarefas

Trabalhos Futuros

- É almejada a simulação dos algoritmos em arquiteturas reais, por exemplo, DAS-3, Grid5000, GridPP, entre outras
- Também, extensões dos algoritmos para melhorar o desempenho em determinadas situações

Muchas Gracias!!!

?

Perguntas e sugestões são muito bem-vindas