



Estudo Comparativo de Algoritmos de Escalonamento para Grades Computacionais

Alvaro H. Mamani-Aliaga e Alfredo Goldman

Instituto de Matemática e Estatística - Universidade de São Paulo (USP)



Introdução

- ▶ Computação em Grade é a agregação de recursos computacionais sem importar a distância entre eles.
- ▶ 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.

Algoritmos de Escalonamento

- ▶ **HEFT**
 - ▶ *Heterogeneous Earliest Finish Time*[1]
 - ▶ Cria uma lista baseada nas prioridades das tarefas em ordem não crescente.
 - ▶ Escalona as tarefas no processador que minimiza o tempo de término, respeitando o ordenamento da lista.
- ▶ **CPOP**
 - ▶ *Critical Path On a Processor*[1]
 - ▶ Agrupa às tarefas do caminho crítico.
 - ▶ Escalona às tarefas do caminho crítico no processador que oferece melhor tempo de término.
- ▶ **PCH**
 - ▶ *Path Clustering Heuristic*[2]
 - ▶ Cria grupos das tarefas, (*clustering*).
 - ▶ Escalona cada grupo no processador que ofereça melhor tempo no término das tarefas.

Aplicações

- ▶ **Aplicação Montage**
É usada para gerar mosaicos personalizados do céu usando pontos de múltiplas imagens de entrada.
- ▶ **A aplicação Epigenomics**
é usada no mapeamento do estado epigenético de células humanas sobre uma grande escala genômica.

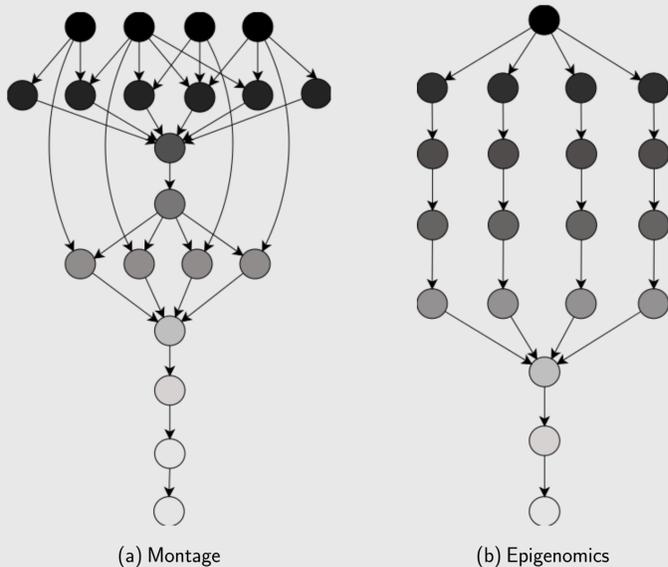
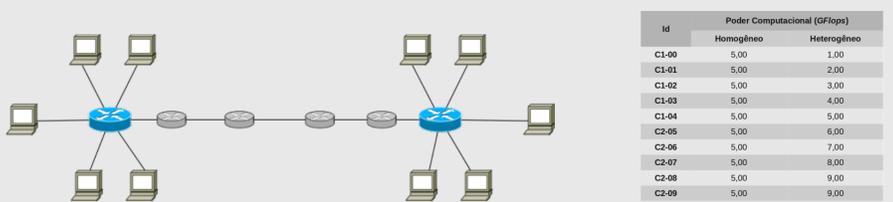


Figura: Representações das aplicações usadas nas simulações.

Arquitetura

- ▶ Para simular os algoritmos foi utilizado o simulador SimGrid[3].
- ▶ Também foram especificados dois aglomerados: homogêneo e heterogêneo.
- ▶ Foram implementados além dos algoritmos de escalonamento *HEFT*, *CPOP* e *PCH*, um escalonamento de tipo *FIFO*.



(a) Representação do aglomerado usado nos experimentos, cada roteador azul representa um aglomerado e a interligação com o outro.

| Id | Poder Computacional (GFlops) | |
|-------|------------------------------|-------------|
| | Homogêneo | Heterogêneo |
| C1-00 | 5,00 | 1,00 |
| C1-01 | 5,00 | 2,00 |
| C1-02 | 5,00 | 3,00 |
| C1-03 | 5,00 | 4,00 |
| C1-04 | 5,00 | 5,00 |
| C2-05 | 5,00 | 6,00 |
| C2-06 | 5,00 | 7,00 |
| C2-07 | 5,00 | 8,00 |
| C2-08 | 5,00 | 9,00 |
| C2-09 | 5,00 | 9,00 |

(b) Tabela de poderes computacionais dos recursos

Figura: Representação dos recursos computacionais.

Análise das Aplicações

- ▶ A Aplicação Montage possui uma estrutura complexa para o escalonamento, com dois gargalos e nós altamente dependentes uns dos outros.
- ▶ A aplicação Epigenomics possui somente um gargalo, os nós pode ser executados paralelamente, dada a pouca dependência entre eles.

Resultados Experimentais

- ▶ **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.

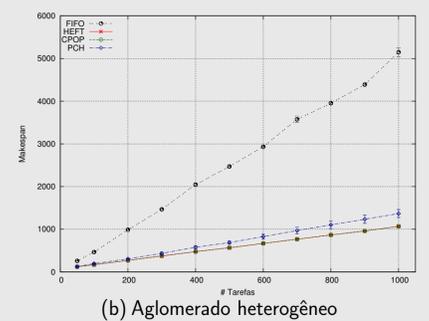
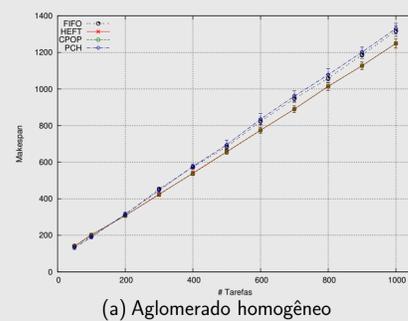


Figura: Resultado da simulação do *Workload Montage* com as estratégias: *HEFT*, *CPOP*, *PCH* e o tipo de escalonamento *FIFO*.

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.

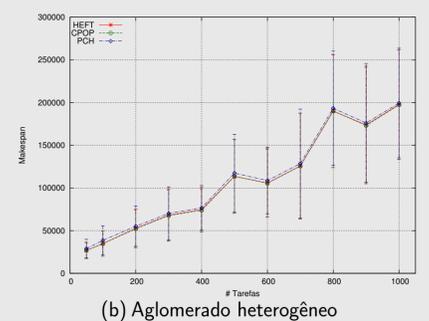
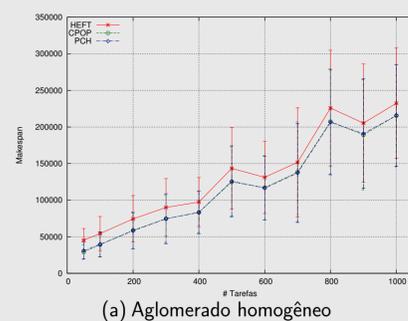


Figura: Resultado da simulação do *Workload Epigenomics*.

Conclusões

- ▶ **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 almejado 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.

Agradecimentos

- ▶ Os autores agradecem ao CNPq pelo apoio financeiro.

Referências

- (1) Topcuouglu, Haluk et Al., *Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing*, IEEE Trans. Parallel Distrib. Syst., 2002.
- (2) Bittencourt, Luiz F et Al., *Uma Heurística de Agrupamento de Caminhos para Escalonamento de Tarefas em Grades Computacionais*, SBRC, 2006.
- (3) Casanova, Henri and Legrand, Arnaud and Quinson, Martin, *SimGrid: a Generic Framework for Large-Scale Distributed Experiments*, IEEE Computer Society, 2008.