

Um Algoritmo de Escalonamento para Redução do Consumo de Energia em Computação em Nuvem

Pedro Paulo Vezzà Campos

Orientador: Prof. Dr. Daniel Macêdo Batista

MACo499 – Trabalho de Formatura Supervisionado

Instituto de Matemática e Estatística

Universidade de São Paulo, São Paulo, Brasil

[pedrovc, batista]@ime.usp.br

11 de novembro de 2013

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

Conclusões

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

Conclusões

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Exatidão e Intuições

Conclusões

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Êxitos e frustrações

Conclusões

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Êxitos e frustrações

Conclusões

Trabalho de conclusão de curso

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Êxitos e frustrações

Conclusões

- Análise das contribuições e resultados obtidos

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Êxitos e frustrações

Conclusões

- Análise das contribuições e resultados obtidos

Agenda

Motivação & Conceitos

- Consumo energético
- Escalonamento de fluxos de trabalho
- Um algoritmo clássico: *Heterogeneous Earliest Finish Time*

Algoritmo Proposto & Experimentos

- O desenvolvimento de um novo algoritmo: Êxitos e frustrações

Conclusões

- Análise das contribuições e resultados obtidos

Agenda

1 Motivação

2 Conceitos

3 Algoritmo Proposto

4 Experimentos

5 Conclusões

Número de Transistores

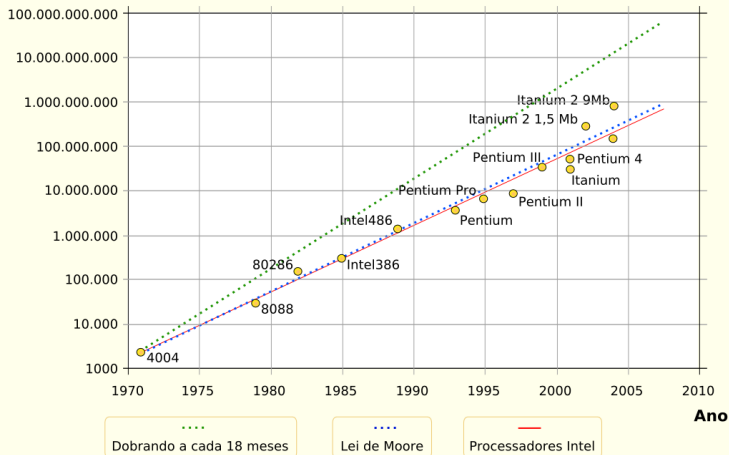


Figura: Lei de Moore ¹

¹ Fonte: Wikipédia, http://pt.wikipedia.org/wiki/Ficheiro:Lei_de_moore_2006.svg.png, em domínio público



Exercício 1: Criar um algoritmo para calcular a soma dos números pares entre 1 e 100.



- Economia de escala
- Capacidade de poder computacional, armazenamento e transmissão de dados



- Economia de **escala**
- **Consolidação** de poder computacional, armazenamento e transferência de dados



- Economia de **escala**
- **Consolidação** de poder computacional, armazenamento e transferência de dados



- Economia de **escala**
- **Consolidação** de poder computacional, armazenamento e transferência de dados

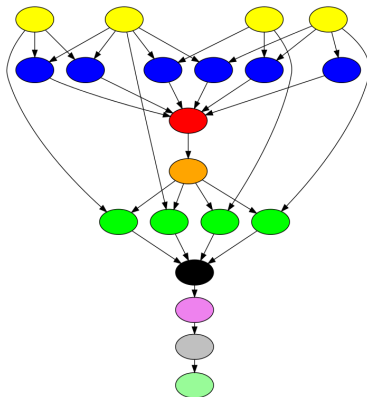


Figura: Montage: Gerador de mosaicos astronômicos



Figura: Aglomerado de Plêiades, figura em domínio público

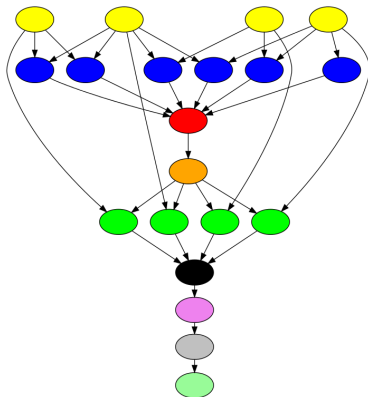


Figura: Montage: Gerador de mosaicos astronômicos



Figura: Aglomerado de Plêiades, figura em domínio público

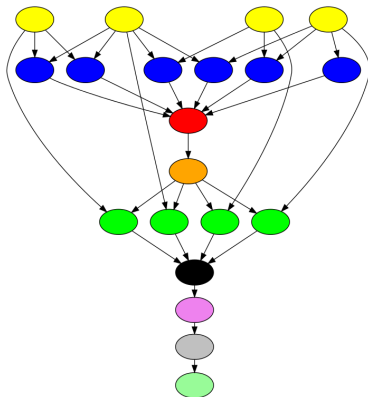


Figura: Montage: Gerador de mosaicos astronômicos



Figura: Aglomerado de Plêiades, figura em domínio público

Agenda

1 Motivação

2 Conceitos

3 Algoritmo Proposto

4 Experimentos

5 Conclusões

Custo total de posse de um *rack* em um *datacenter*

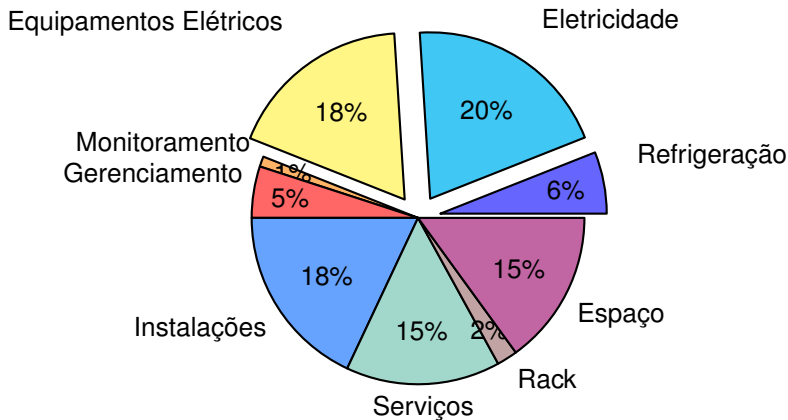


Figura: Custo total de posse de um *rack* em um *data center* típico de alta disponibilidade [Ras11]

Estratégias para economia de energia

- **DVFS**: *Dynamic Voltage and Frequency Scaling*
- Migração de máquinas virtuais
- Algoritmos de escalonamento energeticamente eficientes

Estratégias para economia de energia

- **DVFS**: *Dynamic Voltage and Frequency Scaling*
- **Migração** de máquinas virtuais
- **Algoritmos de escalonamento** energeticamente eficientes

Estratégias para economia de energia

- **DVFS**: *Dynamic Voltage and Frequency Scaling*
- **Migração** de máquinas virtuais
- **Algoritmos de escalonamento** energeticamente eficientes

HEFT: *Heterogeneous Earliest Finish Time*

- Publicado em **2002**
- Heurística para um problema **NP-difícil!**
- Bastante aceito na comunidade científica (Quase **mil citações**)
- Duas fases: **priorização** e **seleção**

HEFT: *Heterogeneous Earliest Finish Time*

- Publicado em **2002**
- Heurística para um problema **NP-difícil!**
- Bastante aceito na comunidade científica (Quase **mil citações**)
- Duas fases: **priorização** e **seleção**

HEFT: *Heterogeneous Earliest Finish Time*

- Publicado em **2002**
- Heurística para um problema **NP-difícil!**
- Bastante aceito na comunidade científica (Quase **mil citações**)
- Duas fases: **priorização** e **seleção**

HEFT: *Heterogeneous Earliest Finish Time*

- Publicado em **2002**
- Heurística para um problema **NP-difícil!**
- Bastante aceito na comunidade científica (Quase **mil citações**)
- Duas fases: **priorização** e **seleção**

Fase de priorização

- **Qual** tarefa **escalonar primeiro?**
- Algoritmo **offline**
- Ordenação topológica:

$$rank_u(n_i) = \overline{w}_i + \max_{n_j \in succ(n_i)} (\overline{c}_{i,j} + rank_u(n_j))$$

Fase de priorização

- Qual tarefa **escalonar primeiro**?
- Algoritmo **offline**
- Ordenação topológica:

$$rank_u(n_i) = \overline{w}_i + \max_{n_j \in succ(n_i)} (\overline{c}_{i,j} + rank_u(n_j))$$

Fase de priorização

- Qual tarefa **escalonar primeiro**?
- Algoritmo **offline**
- Ordenação topológica:

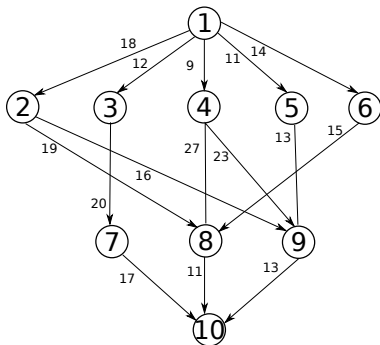
$$rank_u(n_i) = \overline{w_i} + \max_{n_j \in succ(n_i)} (\overline{c_{i,j}} + rank_u(n_j))$$

Fase de seleção

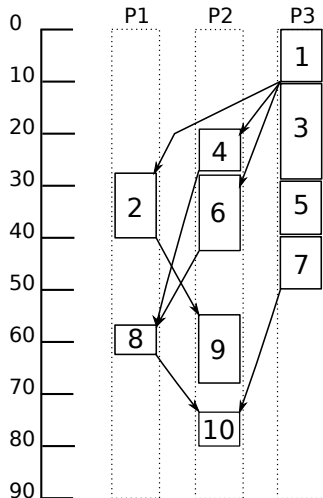
- Minimizar o **tempo mais cedo de conclusão** (*Earliest finish time*)
- Busca por um **espaço vago** grande o suficiente

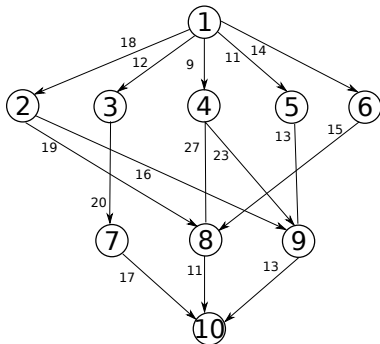
Fase de seleção

- Minimizar o **tempo mais cedo de conclusão** (*Earliest finish time*)
- Busca por um **espaço vago** grande o suficiente

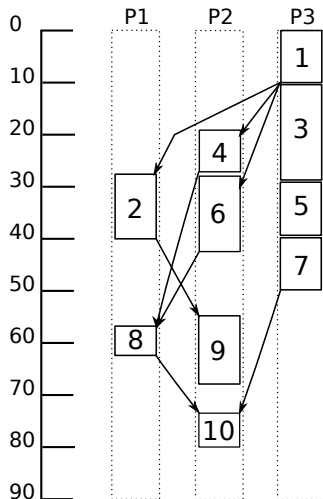


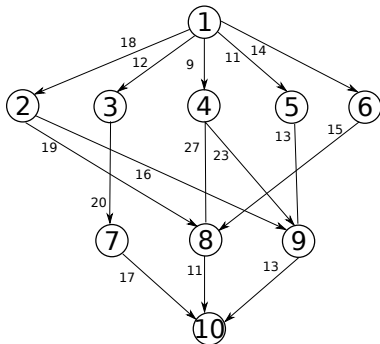
Tarefa	P1	P2	P3	$rank_U(n_i)$
1	14	16	9	108.000
2	13	19	18	77.000
3	11	13	19	80.000
4	13	8	17	80.000
5	12	13	10	69.000
6	13	16	9	63.333
7	7	15	11	42.667
8	5	11	14	35.667
9	18	12	20	44.333
10	21	7	16	14.667



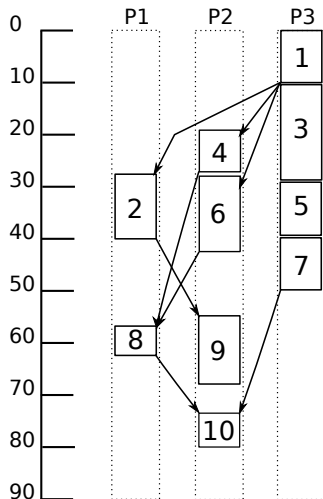


Tarefa	P1	P2	P3	$rank_U(n_i)$
1	14	16	9	108.000
2	13	19	18	77.000
3	11	13	19	80.000
4	13	8	17	80.000
5	12	13	10	69.000
6	13	16	9	63.333
7	7	15	11	42.667
8	5	11	14	35.667
9	18	12	20	44.333
10	21	7	16	14.667





Tarefa	P1	P2	P3	$rank_U(n_i)$
1	14	16	9	108.000
2	13	19	18	77.000
3	11	13	19	80.000
4	13	8	17	80.000
5	12	13	10	69.000
6	13	16	9	63.333
7	7	15	11	42.667
8	5	11	14	35.667
9	18	12	20	44.333
10	21	7	16	14.667



Agenda

1 Motivação

2 Conceitos

3 Algoritmo Proposto

4 Experimentos

5 Conclusões

PowerHEFT

- **Variante** do HEFT, faz uso de uma estratégia de **lookahead**
- **Motivação**: Otimizar o consumo energético de uma **tarefa sozinha não é eficiente**
- Desenvolvido em conjunto com a mestrand **Elaine Naomi Watanabe** (elainew@ime.usp.br)

PowerHEFT

- **Variante** do HEFT, faz uso de uma estratégia de **lookahead**
- **Motivação**: Otimizar o consumo energético de uma **tarefa sozinha não é eficiente**
- Desenvolvido em conjunto com a mestranda **Elaine Naomi Watanabe** (`elainew@ime.usp.br`)

PowerHEFT

- **Variante** do HEFT, faz uso de uma estratégia de **lookahead**
- **Motivação**: Otimizar o consumo energético de uma **tarefa sozinha não é eficiente**
- Desenvolvido em conjunto com a mestrand **Elaine Naomi Watanabe** (`elainew@ime.usp.br`)

Algoritmo

POWERHEFTLOOKAHEAD()

```
1   $V = \{VmMaisRapida\}$  // VMs usadas ao escalonar
2   $O =$  os tipos de VMs que podem ser instanciadas
3  Ordene o conjunto de tarefas segundo o critério  $rank_u$ 
4  enquanto há tarefas não escalonadas
5       $t =$  a tarefa não escalonada de maior  $rank_u$ 

6      // Vamos tentar escalonar  $t$  em uma VM existente
7      para cada  $v$  em  $V$ :
8          ESCALONARPOWERHEFT( $t, v$ )

9      // Vamos tentar escalonar  $t$  em uma nova VM
10     para cada  $o$  em  $O$ :
11          $V = V \cup \{o\}$ 
12         Atualize os valores de  $rank_u$ 
13          $t =$  a tarefa não escalonada de maior  $rank_u$ 
14         ESCALONARPOWERHEFT( $t, o$ )
15     Escalone  $t$  na VM que minimiza a energia consumida
16     Atualize  $V$  e  $rank_u$  caso necessário
```

Algoritmo

ESCALONARPOWERHEFT(*tarefa*, *VM*)

- 1 F = filhos diretos da *tarefa* no DAG
- 2 Escalone *tarefa* em *VM*
- 3 Escalone F utilizando o algoritmo HEFT
- 4 // A modelagem energética utilizada está descrita em [GMDC+13]
- 5 *energia* = ESTIMARENERGIACONSUMIDA()
- 6 Volte para o escalonamento do começo do laço
- 7 **retorne** *energia*

Agenda

1 Motivação

2 Conceitos

3 Algoritmo Proposto

4 Experimentos

5 Conclusões

Simuladores

CloudSim v3 lançado em **2010**, quase 300 citações

WorkflowSim lançado em **abril de 2013**

CloudSim_DVFS lançado em **junho de 2013** (!!)

Simuladores

CloudSim v3 lançado em **2010**, quase 300 citações

WorkflowSim lançado em **abril de 2013**

CloudSim_DVFS lançado em **junho de 2013** (!!)

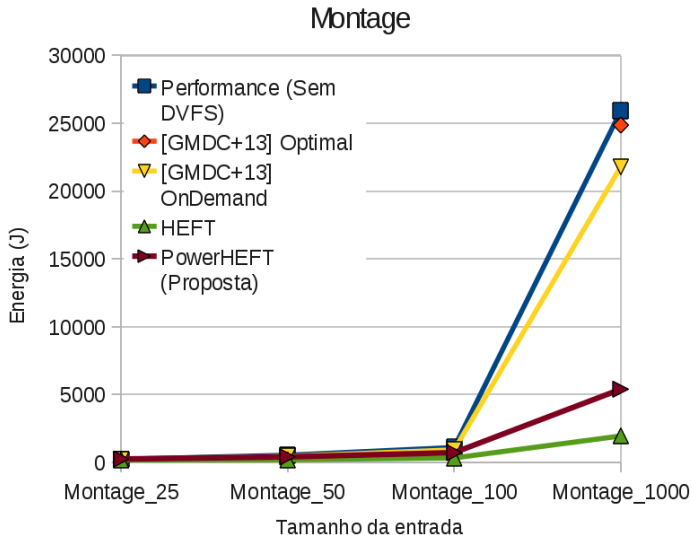
Simuladores

CloudSim v3 lançado em **2010**, quase 300 citações

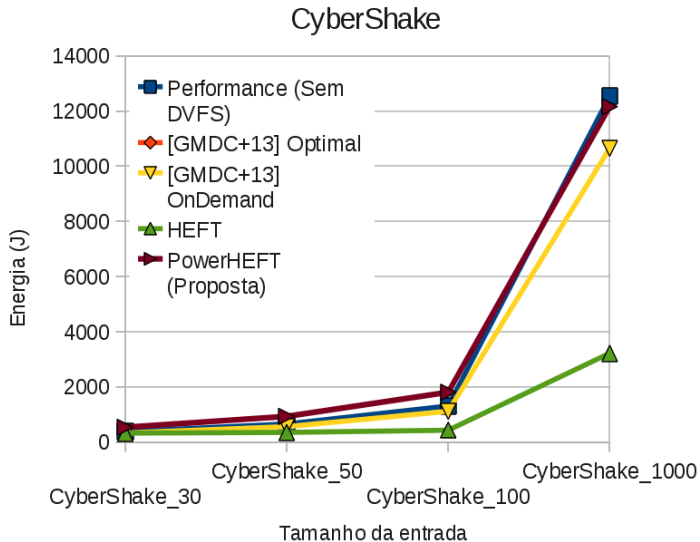
WorkflowSim lançado em **abril de 2013**

CloudSim_DVFS lançado em **junho de 2013** (!!)

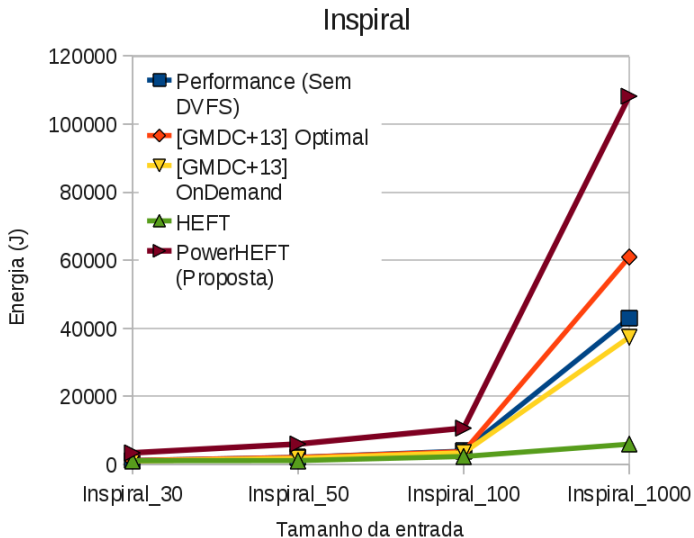
Resultados



Resultados



Resultados



Agenda

1 Motivação

2 Conceitos

3 Algoritmo Proposto

4 Experimentos

5 Conclusões

Motivação
○○○

Conceitos
○○○○○

Algoritmo Proposto
○○○

Experimentos
○○○○

Conclusões
●○○○○

Dificuldades



Dificuldades

Técnicas

Simuladores muito novos e
pouco testados

Dispositivos para simulação
nem sempre adequados para as
necessidades do TCC

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Conceituais

- O LEFT é um algoritmo de programação dinâmica para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele é baseado no algoritmo de Dijkstra, mas com algumas modificações para lidar com grafos ponderados.
- O algoritmo de Dijkstra é um algoritmo de programação dinâmica para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele é baseado no algoritmo de Bellman-Ford, mas com algumas modificações para lidar com grafos ponderados.
- O algoritmo de Bellman-Ford é um algoritmo de programação dinâmica para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele é baseado no algoritmo de Floyd-Warshall, mas com algumas modificações para lidar com grafos ponderados.
- O algoritmo de Floyd-Warshall é um algoritmo de programação dinâmica para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele é baseado no algoritmo de Warshall, mas com algumas modificações para lidar com grafos ponderados.
- O algoritmo de Warshall é um algoritmo de programação dinâmica para encontrar o caminho mais curto entre dois pontos em um grafo ponderado. Ele é baseado no algoritmo de Warshall, mas com algumas modificações para lidar com grafos ponderados.

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Conceituais

- O HEFT é um algoritmo **simples e elegante**
- Trabalho ainda **em andamento** para superá-lo
- Escalonamento energeticamente eficiente é uma **área de pesquisa** inaugurada há **meses**

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Conceituais

- O HEFT é um algoritmo **simples e elegante**
- Trabalho ainda **em andamento** para superá-lo
- Escalonamento energeticamente eficiente é uma **área de pesquisa** inaugurada há **meses**

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Conceituais

- O HEFT é um algoritmo **simples e elegante**
- Trabalho ainda **em andamento** para superá-lo
- Escalonamento energeticamente eficiente é uma **área de pesquisa** inaugurada há **meses**

Dificuldades

Técnicas

- Simuladores **muito novos** e **pouco testados**
- Dificuldades para **estender as funcionalidades** para as necessidades do TCC

Conceituais

- O HEFT é um algoritmo **simples e elegante**
- Trabalho ainda **em andamento** para superá-lo
- Escalonamento energeticamente eficiente é uma **área de pesquisa** inaugurada há **meses**

Este TCC

- Fez um **estudo** de diversas técnicas de **escalonamento** e **simulação** em computação em nuvem
- Implementou com a ajuda da mestrande Elaine Watanabe um **algoritmo novo** para resolver o problema do escalonamento energeticamente eficiente
- Contribuiu com projetos de **software livre**

Este TCC

- Fez um **estudo** de diversas técnicas de **escalonamento** e **simulação** em computação em nuvem
- Implementou com a ajuda da mestrande Elaine Watanabe um **algoritmo novo** para resolver o problema do escalonamento energeticamente eficiente
- Contribuiu com projetos de **software livre**

Este TCC

- Fez um **estudo** de diversas técnicas de **escalonamento** e **simulação** em computação em nuvem
- Implementou com a ajuda da mestrande Elaine Watanabe um **algoritmo novo** para resolver o problema do escalonamento energeticamente eficiente
- Contribuiu com projetos de **software livre**

Obrigado!
Perguntas?

Bibliografia



L.A. Barroso e U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.



Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya e Mihai Alexandru. Energy-aware simulation with dvfs. *Simulation Modelling Practice and Theory*, v.39, i.1, p.76-91, 2013.



Neil Rasmussen. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure. Relatório técnico, Schneider Electric, Paris, 2011.



H. Topcuoglu, S. Hariri e Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

Bibliografia



L.A. Barroso e U. Hözlze. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.



Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya e Mihai Alexandru. Energy-aware simulation with dvfs. *Simulation Modelling Practice and Theory*, v.39, i.1, p.76-91, 2013.



Neil Rasmussen. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure. Relatório técnico, Schneider Electric, Paris, 2011.



H. Topcuoglu, S. Hariri e Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

Bibliografia



L.A. Barroso e U. Hözlze. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.



Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya e Mihai Alexandru. Energy-aware simulation with dvfs. *Simulation Modelling Practice and Theory*, v.39, i.1, p.76-91, 2013.



Neil Rasmussen. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure. Relatório técnico, Schneider Electric, Paris, 2011.



H. Topcuoglu, S. Hariri e Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

Bibliografia



L.A. Barroso e U. Hözlze. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.



Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya e Mihai Alexandru. Energy-aware simulation with dvfs. *Simulation Modelling Practice and Theory*, v.39, i.1, p.76-91, 2013.



Neil Rasmussen. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure. Relatório técnico, Schneider Electric, Paris, 2011.



H. Topcuoglu, S. Hariri e Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

Um Algoritmo de Escalonamento para Redução do Consumo de Energia em Computação em Nuvem

Pedro Paulo Vezzà Campos

Orientador: Prof. Dr. Daniel Macêdo Batista

MACo499 – Trabalho de Formatura Supervisionado

Instituto de Matemática e Estatística

Universidade de São Paulo, São Paulo, Brasil

[pedrovc, batista]@ime.usp.br

11 de novembro de 2013

Algoritmo

HETEROGENEOUS-EARLIEST-FINISH-TIME()

- 1 Defina os custos computacionais das tarefas e os custos de comunicação das arestas com valores médios
- 2 Calcule $rank_u$ para todas as tarefas varrendo o grafo de “baixo para cima”, iniciando pela tarefa final.
- 3 Ordene as tarefas em uma lista de escalonamento utilizando uma ordem não crescente de valores de $rank_u$.
- 4 **enquanto** há tarefas não escalonadas na lista
- 5 Selecione a primeira tarefa, n_i da lista de escalonamento.
- 6 **para** cada processador p_k no conjunto de processadores
- 7 Calcule o tempo mais cedo de conclusão da tarefa n_i , considerando que ela execute em p_k
- 8 Defina a tarefa n_i para executar no processador p_j que minimiza o tempo mais cedo de conclusão da tarefa n_i .

Modelagem energética

Velocidades (GHz)	0.8	1.0	1.2	1.5	1.7
Nível ocioso (W)	140	146	153	159	167
Nível máximo (W)	228	238	249	260	272

Tabela: Frequências do Grid'5000 Reims com as medidas de potência durante cargas mínima e máxima (0% e 100% de uso dos 24 núcleos de um nó de processamento) [GMDC+13]

Resultados

