



Universidade Federal Fluminense

Programa de Pós-Graduação em Computação

Instituto de Computação IC-UFF

Escalonamento de Tarefas e Alocação de Arquivos de Dados de Workflows Científicos em Nuvens Computacionais

Luan Teylo Gouveia Lima

Orientadora: Lúcia M. A. Drummond

Co-Orientador: Yuri Abitbol de Menezes Frota

Dissertação de Mestrado
Niterói, 17 de Março 2017

Agenda

- 1 Introdução
- 2 Trabalhos Relacionados
- 3 Descrição do Problema e Solução Proposta
- 4 Resultados
- 5 Conclusão e Trabalhos Futuros

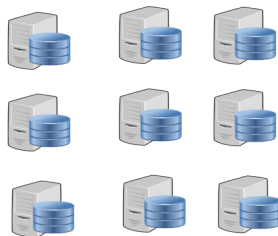
Introdução

O problema de escalonamento

Escalonamento é o processo de mapear a execução de aplicações em um ambiente computacional [Yu e Buyya, 2006]



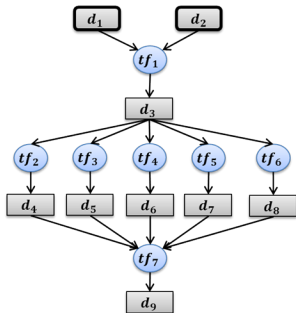
Escalonador

A thick black arrow pointing from the "Aplicações" box to the server grid.

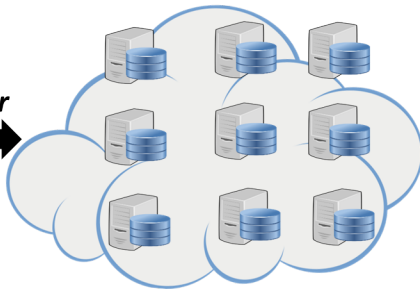
Introdução

O problema de escalonamento

Este trabalho propõe uma solução para o problema de escalonamento de tarefas e alocação de arquivos de *workflows científicos (WfCs)* executados em ambientes de *nuvens computacionais*.



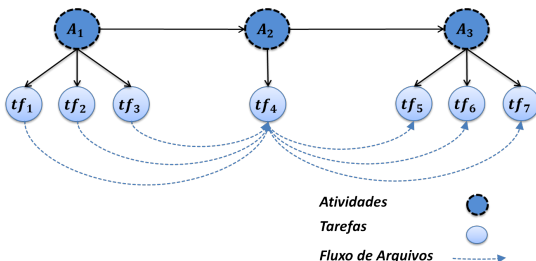
Escalonador



Introdução

Workflows Científicos

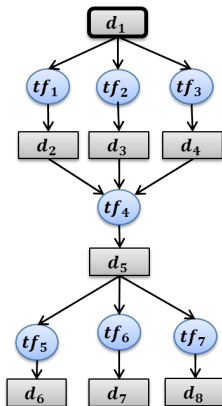
- ▶ Abstração que estrutura as etapas de um experimento como um grafo acíclico dirigido (DAG), no qual os nós correspondem às atividades de processamento e as arestas representam os fluxos de dados entre elas
- ▶ Experimentos científicos são comumente executados por uma cadeia de aplicações, nas quais a saída de um programa é a entrada para outro



Introdução

Workflows Científicos

Diferente dos trabalhos encontrados na literatura, neste trabalho os arquivos **não** são representados como arcs do grafo e sim como parte do conjunto de vértices.



Introdução

Nuvens Computacionais

- ▶ Vantagens técnicas e econômicas em relação as outras plataformas (como *grids* e *clusters*).
 - Recursos sob demanda e com fácil acesso
 - Recursos virtualmente infinitos
 - Pagamento baseado no modelo *pay-per-use*
 - Serviços integrados de processamento de dados e armazenamento escalável
- ▶ O uso da nuvem para o processamento de WfCs é feito através da alocação de máquinas virtuais (MV) que forma um *cluster* virtual

Introdução

Principais Contribuições

As principais contribuições deste trabalho são:

- ▶ Modelo de representação para WfCs
- ▶ Formulação do problema de Escalonamento de Tarefas e Alocação de Arquivos de Dados (ETAA) como um problema de programação inteira.
- ▶ Algoritmo evolutivo híbrido para solução do problema
- ▶ Validação da solução proposta por meio de experimentos práticos e teóricos

Trabalhos Relacionados

O escalonamento de tarefas em sistemas distribuídos faz parte dos chamados problemas NP-Difícil
– [Ullman, 1973]

Por conta disso, heurísticas ou metaheurísticas são comumente utilizadas

- ▶ ***Heterogeneous Earliest-Finish-Time* HEFT** [Topcuoglu *et al.*, 2002].
 - *Adaptative* HEFT (AHEFT) [Yu e Shi, 2007]
 - *Multi-Objective* HEFT (MOHEFT) [Durillo, Fard e Prodan, 2012]
 - *HEFT-Based* [Chopra e Singh, 2013]
- ▶ **Heurísticas de Particionamento**
 - *Partitioned Balanced Time Scheduling* (PBTS) [Byun *et al.*, 2011]
 - IaaS Cloud Partial Critical Paths (IC-PCP) [Abrishami *et al.*, 2009]
- ▶ **MinMin** [Blythe *et al.*, 2005]

Trabalhos Relacionados

Metaheurísticas

- ▶ **Genetic Algorithm** [Goldberg, 1989]
 - *Genetic Algorithm* para Escalonamento estático [Yu e Buyya, 2006]
- ▶ **Particle Swarm Optimization (PSO)** [Kennedy e Eberhart, 1995]
 - PSO como heurística dinâmica [Pandey *et al.*, 2010]
 - PSO para escalonamento estático [Rodriguez e Buyya, 2014]
- ▶ **Ant Colony Optimization (ACO)** [Dorigo e Di Caro, 1999]
 - Multi-objetivo ACO [Chen e Zhang, 2009]
 - *Knowledge-based ant colony optimization* (KBACO) [Hu *et al.*, 2010]

Trabalhos Relacionados

Metaheurísticas

- ▶ **Genetic Algorithm** [Goldberg, 1989]
 - *Genetic Algorithm* para Escalonamento estático [Yu e Buyya, 2006]
- ▶ **Particle Swarm Optimization (PSO)** [Kennedy e Eberhart, 1995]
 - PSO como heurística dinâmica [Pandey *et al.*, 2010]
 - PSO para escalonamento estático [Rodriguez e Buyya, 2014]
- ▶ **Ant Colony Optimization (ACO)** [Dorigo e Di Caro, 1999]
 - Multi-objetivo ACO [Chen e Zhang, 2009]
 - *Knowledge-based ant colony optimization* (KBACO) [Hu *et al.*, 2010]

Não consideram a localização dos arquivos na decisão de escalonamento.

Trabalhos Relacionados

data-aware scheduling

- ▶ Clusterização por Matriz de dependência [Yuan *et al.*, 2010]
- ▶ Algoritmo de Particionamento [Çatalyürek *et al.*, 2011]
- ▶ Algoritmo Evolutivo [Szabo *et al.*, 2013]
- ▶ K-means [Wang *et al.*, 2014]
- ▶ Heurística *File Locality-Aware Scheduling* [Bryk *et al.*, 2015]

Trabalhos Relacionados

data-aware scheduling

- ▶ Clusterização por Matriz de dependência [Yuan *et al.*, 2010]
- ▶ Algoritmo de Particionamento [Çatalyürek *et al.*, 2011]
- ▶ Algoritmo Evolutivo [Szabo *et al.*, 2013]
- ▶ K-means [Wang *et al.*, 2014]
- ▶ Heurística *File Locality-Aware Scheduling* [Bryk *et al.*, 2015]

A alocação dos arquivos não é definida pelos algoritmos.

Modelo Matemático

O Problema de Escalonamento de Tarefas e Alocação de Arquivos (ETAA)

O ETAA foi formulado como um problema de **Programação Inteira**

Modelo Matemático

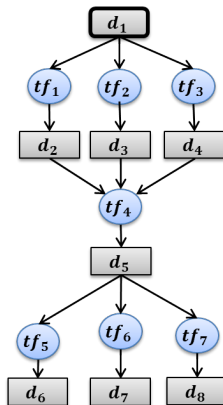
Modelo da Aplicação

Denotada por um DAG G , na forma:

$$G = (V, A, a, \omega)$$

$$V = N \cup D$$

- ▶ V : Vértices
- ▶ A : Arcos
- ▶ a_i : Tempo base associado à tarefa $i \in N$
- ▶ ω_k : Tempo base associado ao arco $k \in A$
- ▶ N : Tarefas
- ▶ $D = D_s \cup D_d$: Arquivos

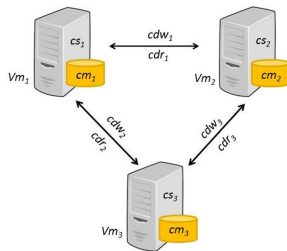


Modelo Matemático

Modelo do Ambiente

O conjunto M representa os recursos (MVs) utilizados na execução.

- ▶ cm : Capacidade de Armazenamento
- ▶ cs : Valor de *slowdown* (variação de desempenho)
- ▶ cdr_l : Latência do enlace l (leitura)
- ▶ cdw_l : Latência do enlace l (escrita)



Modelo Matemático

O tempo de execução:

$$t_{ij} = a_i \times cs_j$$

Tempo base da tarefa i vezes o **slowdown** da MV_j

O tempo de comunicação:

$$\begin{aligned}\overrightarrow{t}_{djp} &= \omega_{di} \times cdr_l \text{ (leitura)} \\ \overleftarrow{t}_{djp} &= \omega_{id} \times cdw_l \text{ (escrita)}\end{aligned}$$

Tempo base de transferência do arquivo d vezes a **latência** do enlace l

Modelo Matemático

Função Objetivo

Minimiza o tempo total de execução (*makespan*):

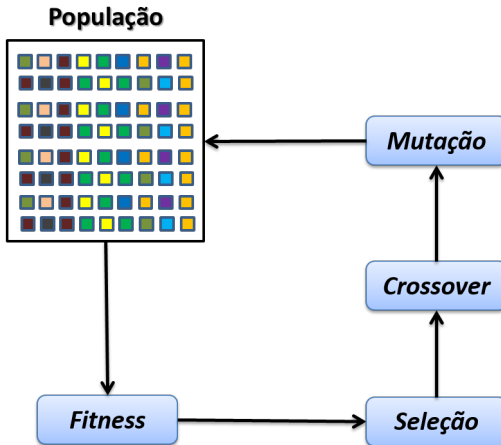
$$\min z_T$$

- z_T : Tempo total de execução do *workflow*

- ▶ Definidas 14 restrições para garantir que:
 - As operações de execução, leitura e escrita sejam realizadas nos tempos corretos
 - Todas as tarefas sejam executadas
 - Todas as operações de leitura e escrita ocorram
 - As restrições de armazenamento sejam satisfeitas
 - As relações de dependências sejam respeitadas

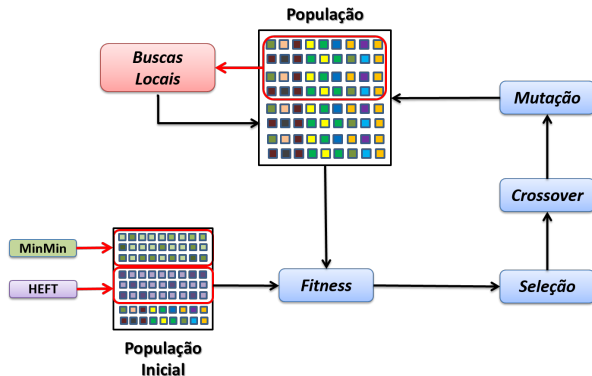
Algoritmo Evolutivo

São métodos de otimização inspirados nos mecanismos de evolução biológica observados na natureza.



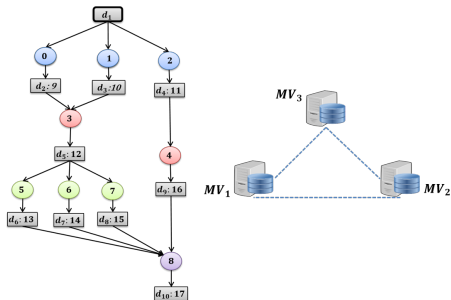
Algoritmo Evolutivo Híbrido para Escalonamento de Tarefas e Alocação de Arquivos (AEH-ETAA)

- ▶ População Inicial contém soluções geradas a partir das heurísticas HEFT e MinMin.
- ▶ Buscas Locais são efetuadas nas melhores soluções.



O cromossomo é composto por duas estruturas que representam:

- ▶ A alocação das tarefas e dos dados
- ▶ A ordem de distribuição das tarefas



Vetor de Alocação

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
MV_1	MV_5	MV_2	MV_1	MV_5	MV_5	MV_1	MV_3	MV_2	MV_2	MV_5	MV_2	MV_1	MV_1	MV_2	MV_1	MV_1	MV_1

Lista de Execução

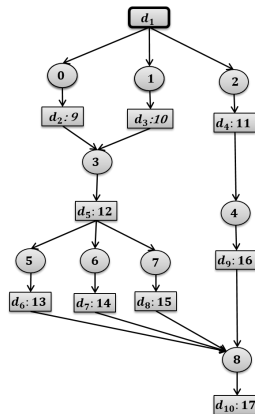


A ordem de execução das tarefas é definida conforme as equações de altura apresentadas por Tsujimura e Gen (1997):

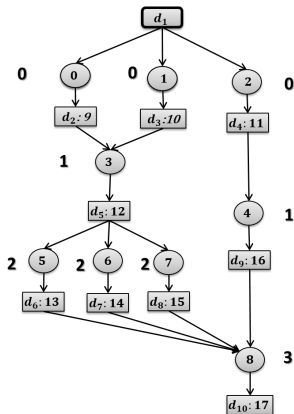
$$altura_inicial(tf_i) = \begin{cases} 0, & \text{se } pred(tf_i) = \emptyset \\ 1 + \max_{tf_j \in pred(tf_i)} altura_inicial(tf_j), & \text{caso contrário} \end{cases}$$

$$altura(tf_i) = \begin{cases} altura_inicial(tf_i), & \text{se } suc(tf_i) = \emptyset \\ rand \in [altura_inicial(tf_i), \min_{\forall tf_k \in suc(tf_i)} \{altura(tf_k)\} - 1], & \text{caso contrário} \end{cases}$$

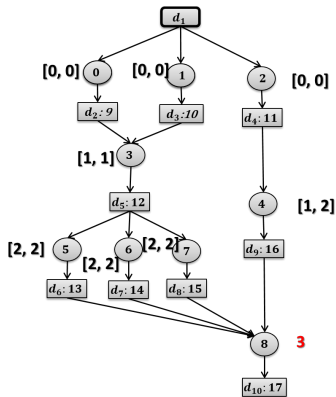
$$altura_inicial(tf_i) = \begin{cases} 0, & \text{se } pred(tf_i) = \emptyset \\ 1 + \max_{tf_j \in pred(tf_i)} altura_inicial(tf_j), & \text{caso contrário} \end{cases}$$



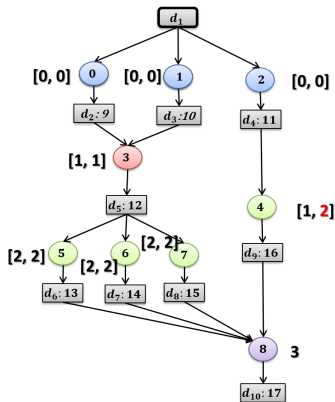
$$altura_inicial(tf_i) = \begin{cases} 0, & \text{se } pred(tf_i) = \emptyset \\ 1 + \max_{tf_j \in pred(tf_i)} altura_inicial(tf_j), & \text{caso contrário} \end{cases}$$



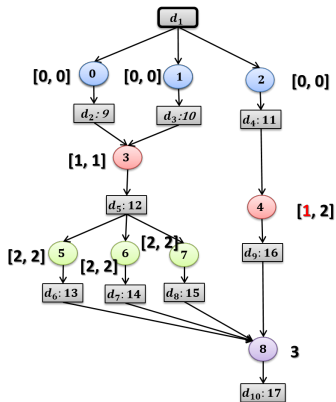
$$altura(tf_i) = \begin{cases} altura_inicial(tf_i), & \text{se } suc(tf_i) = \emptyset \\ rand \in [altura_inicial(tf_i), \min_{\forall tf_k \in suc(tf_i)} \{altura(tf_k)\} - 1], & \text{caso contrário} \end{cases}$$



$$altura(tf_i) = \begin{cases} altura_inicial(tf_i), & \text{se } suc(tf_i) = \emptyset \\ rand \in [altura_inicial(tf_i), \min_{\forall tf_k \in suc(tf_i)} \{altura(tf_k)\} - 1], & \text{caso contrário} \end{cases}$$



$$altura(tf_i) = \begin{cases} altura_inicial(tf_i), & \text{se } suc(tf_i) = \emptyset \\ rand \in [altura_inicial(tf_i), \min_{\forall tf_k \in suc(tf_i)} \{altura(tf_k)\} - 1], & \text{caso contrário} \end{cases}$$



AEH-ETAA

Operador de *Crossover*

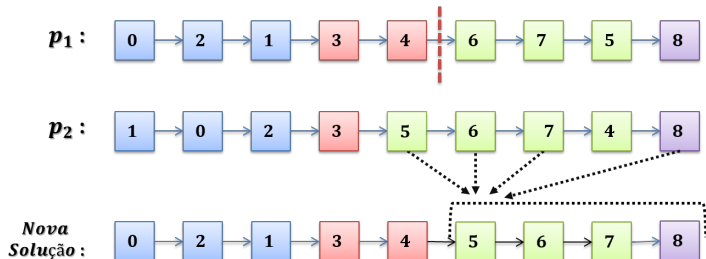
Lista de Execução:



AEH-ETAA

Operador de Crossover

Lista de Execução:



AEH-ETAA

Operador de Crossover

Vetor de Alocação:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p_1 :	MV_1	MV_3	MV_2	MV_1	MV_3	MV_3	MV_1	MV_3	MV_2	MV_2	MV_3	MV_2	MV_1	MV_1	MV_2	MV_1	MV_2	MV_1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p_2 :	MV_3	MV_3	MV_1	MV_2	MV_1	MV_3	MV_1	MV_2	MV_1	MV_3	MV_1	MV_2	MV_2	MV_3	MV_1	MV_1	MV_1	MV_1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<i>Nova</i> Solução :																		

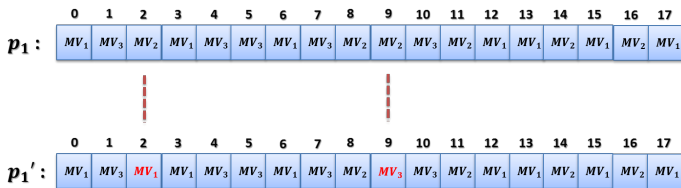
Vetor de Alocação:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p_1 :	MV_1	MV_3	MV_2	MV_1	MV_3	MV_3	MV_1	MV_3	MV_2	MV_2	MV_3	MV_2	MV_1	MV_1	MV_2	MV_1	MV_2	MV_1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p_2 :	MV_3	MV_3	MV_1	MV_2	MV_1	MV_3	MV_1	MV_2	MV_1	MV_3	MV_1	MV_2	MV_2	MV_3	MV_1	MV_1	MV_1	MV_1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Nova Solução :	MV_1	MV_3	MV_2	MV_1	MV_3	MV_3	MV_1	MV_3	MV_1	MV_3	MV_1	MV_2	MV_2	MV_3	MV_1	MV_1	MV_1	MV_1

AEH-ETAA

Operador de Mutaç o

Executado apenas no *vetor de aloca o*:



----- **M quina Virtual Alterada**

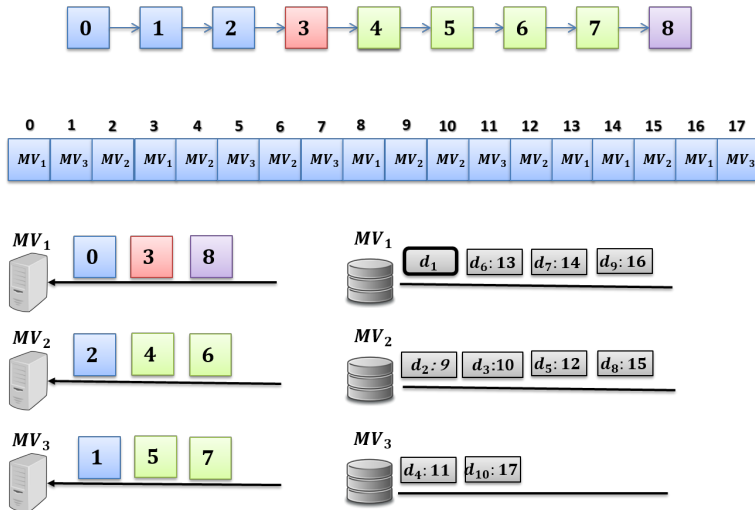
Foram definidas três buscas locais e uma heurística de busca:

- ▶ Troca-MV
- ▶ Troca-Posição
- ▶ Move-Elemento
- ▶ Heurística *Path Relinking*

Todas as buscas locais são *first-improvement*.

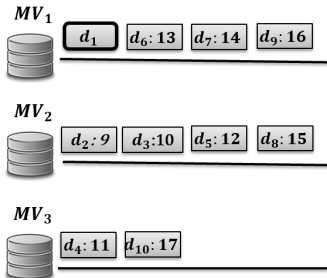
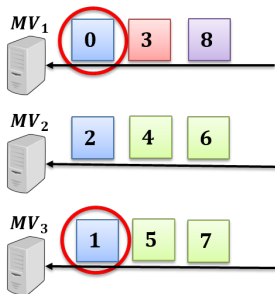
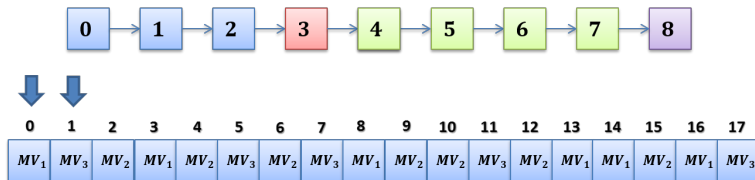
AEH-ETAA

Busca Local Troca-MV



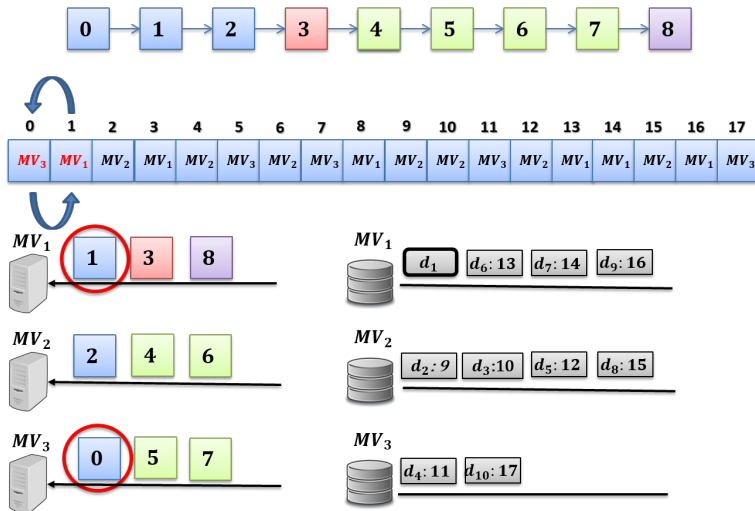
AEH-ETAA

Busca Local Troca-MV



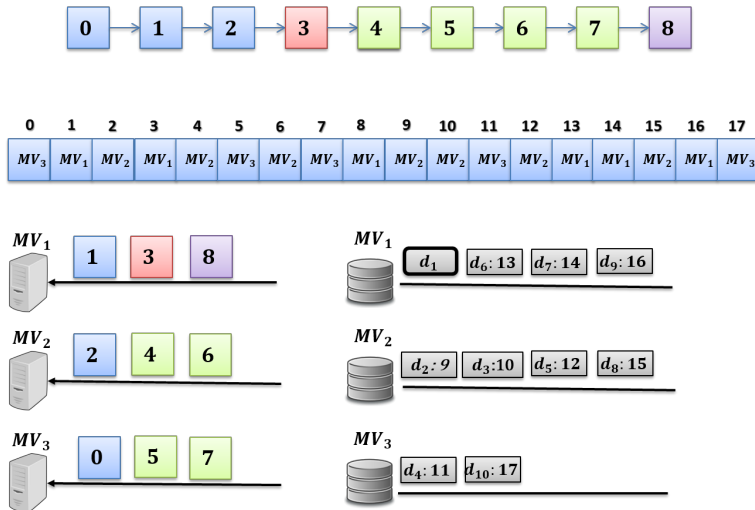
AEH-ETAA

Busca Local Troca-MV



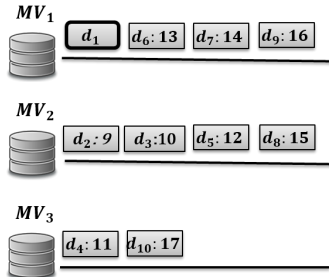
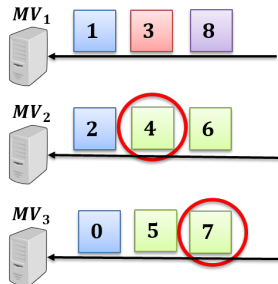
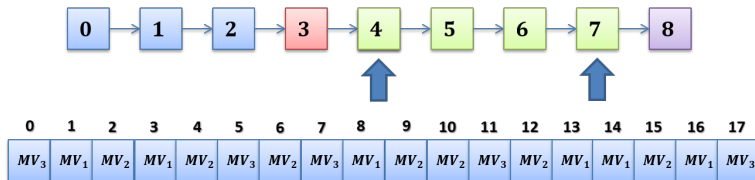
AEH-ETAA

Busca Local Troca-Posição



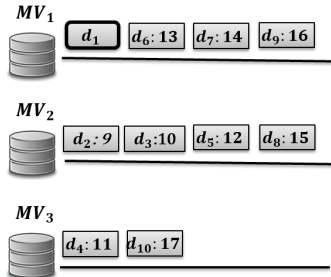
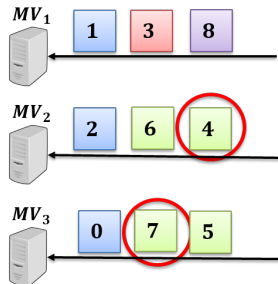
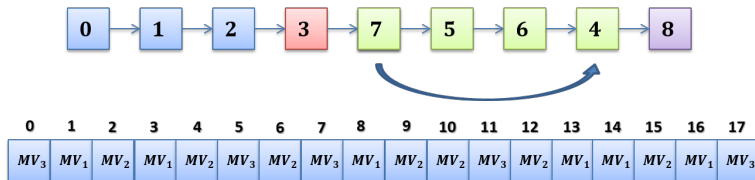
AEH-ETAA

Busca Local Troca-Posição



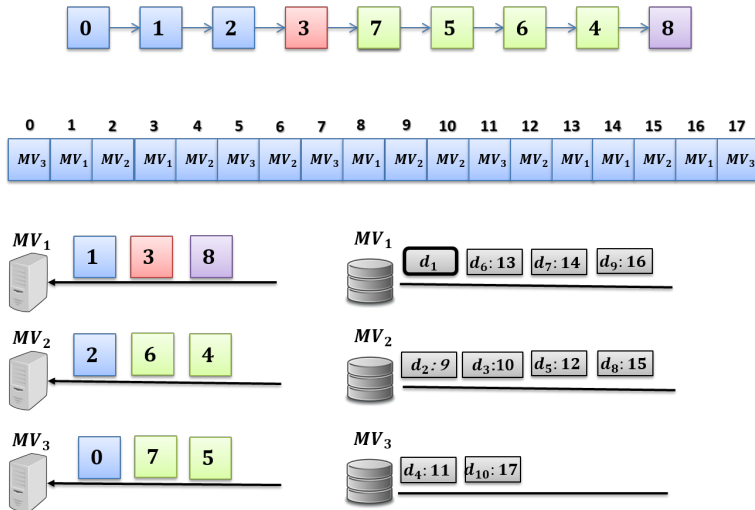
AEH-ETAA

Busca Local Troca-Posição



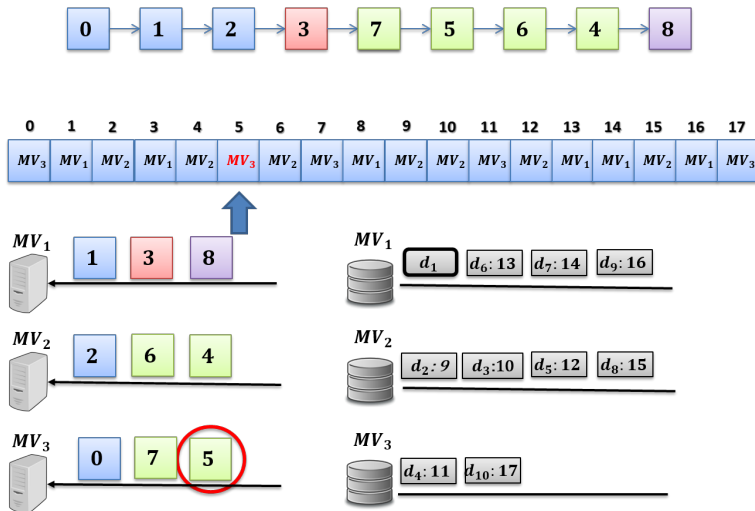
AEH-ETAA

Busca Local Move-Elemento



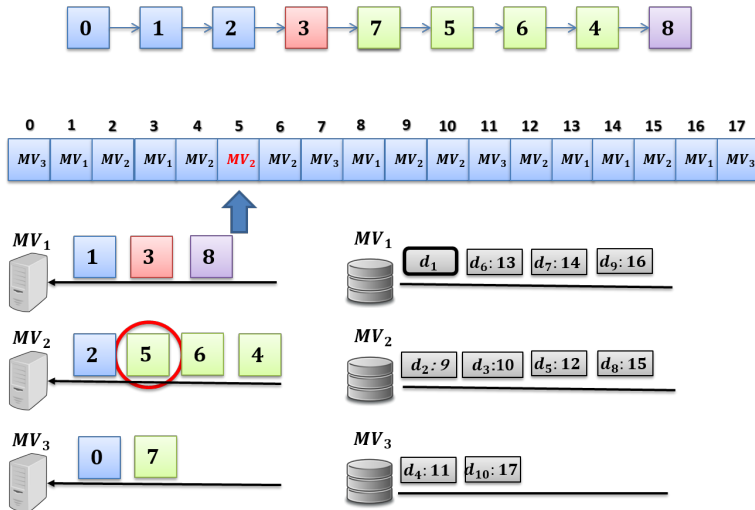
AEH-ETAA

Busca Local Move-Elemento



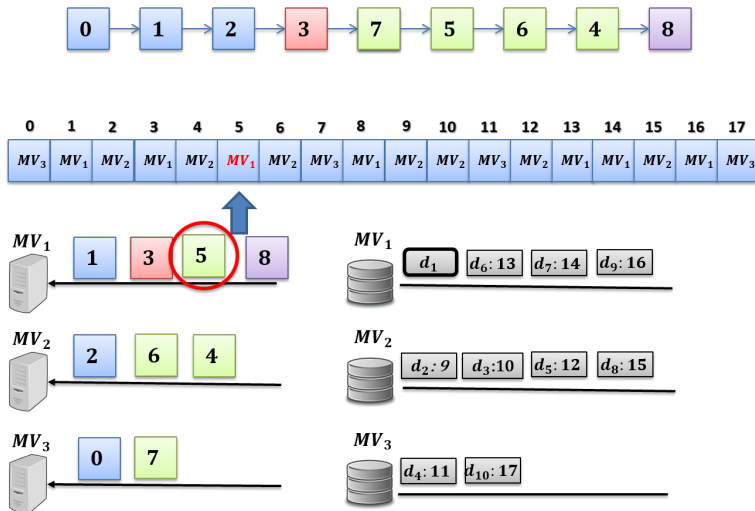
AEH-ETAA

Busca Local Move-Elemento



AEH-ETAA

Busca Local Move-Elemento



- ▶ As melhores soluções encontradas pela metaheurística são guardadas em um **conjunto elite**
- ▶ A estratégia explora soluções intermediárias ao longo da trajetória que conecta duas soluções pertencentes a esse conjunto

Solução Inicial



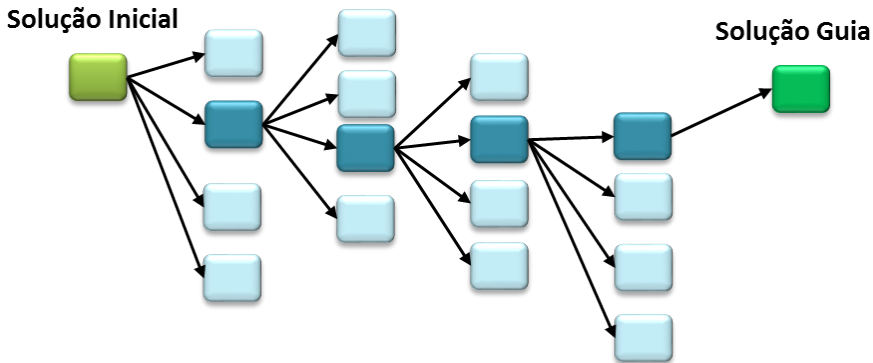
**Caminho na vizinhança
de soluções**



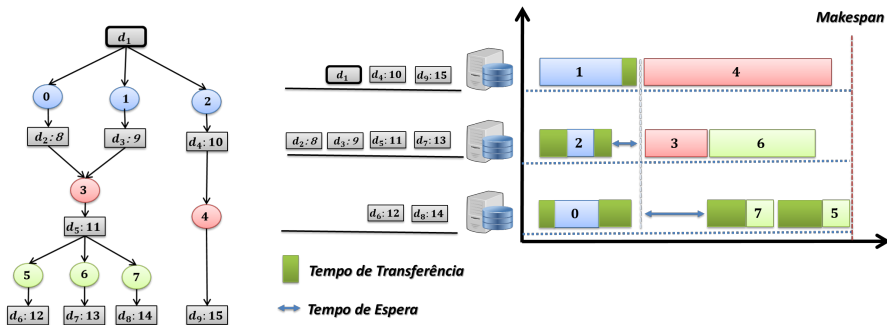
Solução Guia



- ▶ O caminho é gerado selecionando movimentos que introduzem características da **Solução Guia** na **Solução Inicial**.



A qualidade de um indivíduo é estimada conforme o *makespan* de sua solução.



Resultados Experimentais

Para avaliar a abordagem proposta, foram conduzidos experimentos práticos e teóricos

- ▶ Processador Intel I7-3770, CPU 3,40GHZ com 12GB Memória RAM e Ubuntu 14.04. Algoritmos implementados em C++ e compilados com G++ 5.3.0
- ▶ Para a comparação teórica foram utilizadas instâncias sintéticas produzidas pelo *workflow generator* [Silva, 2014]
- ▶ A comparação prática foi feita no Amazon EC2

Resultados Experimentais

Avaliação Teórica

O AEH-ETAA foi comparado com:

- ▶ Modelo matemático IP-ETAA (CPLEX 12.5.1)
- ▶ Heurísticas MinMin e HEFT

Resultados Experimentais

Avaliação Teórica

AEH-ETAA vs IP-ETAA

Instâncias	AEH-ETAA		Formulação Matemática		Dif. Percentual
5A_m3	10,0	0,83	10,0	4,2	0,0
5B_m3	11,0	0,79	11,0	14,0	0,0
5C_m3	13,0	0,71	13,0	2,8	0,0
7A_m3	21,0	0,96	21,0	140,8	0,0
7B_m3	16,0	1,17	16,0	7,3	0,0
7C_m3	14,0	1,11	14,0	105,0	0,0
10A_m3	21,0	1,80	21,0	1644,6	0,0
10B_m3	12,0	1,74	12,0	21,4	0,0
10C_m3	21,0	1,22	21,0	316,7	0,0
15A_m3	16,0	2,47	16,0	2606,6	0,0
15B_m3	11,4	3,01	11,0	600,9	3,6
15C_m3*	19,2	2,42	94,0	86395,5	–
5A_m5	10,0	0,90	10,0	14,4	0,0
5B_m5	9,0	0,50	9,0	462,6	0,0
5C_m5	10,0	0,75	10,0	9,0	0,0
7A_m5	27,0	0,97	27,0	1836,4	0,0
7B_m5	26,0	1,26	26,0	21,0	0,0
7C_m5	16,0	1,09	16,0	2426,9	0,0
10A_m5*	25,0	1,58	–	86489,2	–
10B_m5	14,0	1,75	13,0	103,0	7,6
10C_m5	47,6	1,88	47,0	7353,5	1,2
15A_m5	16,0	2,64	16,0	15783,9	0,0
15B_m5	11,0	2,97	10,0	10764,3	10,0
15C_m5*	20,0	2,49	–	86243,0	–
Média	17,4	1,5	20,2	12640,3	1,1

Resultados Experimentais

Avaliação Teórica

AEH-ETAA vs IP-ETAA

Instâncias	AEH-ETAA		Formulação Matemática		Dif. Percentual
5A_m3	10,0	0,83	10,0	4,2	0,0
5B_m3	11,0	0,79	11,0	14,0	0,0
5C_m3	13,0	0,71	13,0	2,8	0,0
7A_m3	21,0	0,96	21,0	140,8	0,0
7B_m3	16,0	1,17	16,0	7,3	0,0
7C_m3	14,0	1,11	14,0	105,0	0,0
10A_m3	21,0	1,80	21,0	1644,6	0,0
10B_m3	12,0	1,74	12,0	21,4	0,0
10C_m3	21,0	1,22	21,0	316,7	0,0
15A_m3	16,0	2,47	16,0	2606,6	0,0
15B_m3	11,4	3,01	11,0	600,9	3,6
15C_m3*	19,2	2,42	94,0	86395,5	—
5A_m5	10,0	0,90	10,0	14,4	0,0
5B_m5	9,0	0,50	9,0	462,6	0,0
5C_m5	10,0	0,75	10,0	9,0	0,0
7A_m5	27,0	0,97	27,0	1836,4	0,0
7B_m5	26,0	1,26	26,0	21,0	0,0
7C_m5	16,0	1,09	16,0	2426,9	0,0
10A_m5*	25,0	1,58	—	86489,2	—
10B_m5	14,0	1,75	13,0	103,0	7,6
10C_m5	47,6	1,88	47,0	7353,5	1,2
15A_m5	16,0	2,64	16,0	15783,9	0,0
15B_m5	11,0	2,97	10,0	10764,3	10,0
15C_m5*	20,0	2,49	—	86243,0	—
Média	17,4	1,5	20,2	12640,3	1,1

Resultados Experimentais

Avaliação Teórica

Em relação a abordagem exata o AEH-ETAA obteve:

- ▶ Diferença percentual média de **1,1%**
- ▶ Tempo de execução médio significativamente menor (**1,5s** contra 12.640,3s)

Resultados Experimentais

Avaliação Teórica

AEH-ETAA vs HEFT e MinMin

Inst.	MinMin	HEFT	AEH-ETAA		
	<i>Makesp. (min)</i>	<i>Makesp. (min)</i>	<i>Makesp. (min)</i>	<i>RSD</i>	<i>Exec. (min)</i>
Cybershake30	11,98	11,28	10,25	0,0019	0,39
Cybershake50	14,88	16,65	12,46	0,0264	2,19
Cybershake100	26,93	28,08	14,52	0,0261	8,11
Genome.3510	530,88	469,38	444,91	0,0109	4,05
Genome.7020	923,46	865,45	833,98	0,0004	6,79
Epigenomics24	67,36	57,30	55,80	0,0000	0,03
Epigenomics46	119,95	104,07	99,27	0,0140	0,48
Epigenomics100	1,004,23	916,73	889,65	0,0001	4,02
Montage25	1,81	1,16	0,95	0,0242	0,05
Montage50	3,06	2,08	1,88	0,0036	0,10
Montage100	5,31	4,66	3,93	0,0038	3,33
Inspiral30	18,26	14,61	13,95	0,0014	0,17
Inspiral50	29,96	23,36	22,41	0,0005	0,79
Inspiral100	44,65	41,80	40,76	0,0036	1,55

Resultados Experimentais

Avaliação Teórica

AEH-ETAA vs HEFT e MinMin

Inst.	MinMin	HEFT	AEH-ETAA		
	<i>Makesp. (min)</i>	<i>Makesp. (min)</i>	<i>Makesp. (min)</i>	<i>RSD</i>	<i>Exec. (min)</i>
Cybershake30	11,98	11,28	10,25	0,0019	0,39
Cybershake50	14,88	16,65	12,46	0,0264	2,19
Cybershake100	26,93	28,08	14,52	0,0261	8,11
Genome.3510	530,88	469,38	444,91	0,0109	4,05
Genome.7020	923,46	865,45	833,98	0,0004	6,79
Epigenomics24	67,36	57,30	55,80	0,0000	0,03
Epigenomics46	119,95	104,07	99,27	0,0140	0,48
Epigenomics100	1,004,23	916,73	889,65	0,0001	4,02
Montage25	1,81	1,16	0,95	0,0242	0,05
Montage50	3,06	2,08	1,88	0,0036	0,10
Montage100	5,31	4,66	3,93	0,0038	3,33
Inspiral30	18,26	14,61	13,95	0,0014	0,17
Inspiral50	29,96	23,36	22,41	0,0005	0,79
Inspiral100	44,65	41,80	40,76	0,0036	1,55

Resultados Experimentais

Avaliação Teórica

- ▶ Os resultados mais significativos estão presentes nos *workflows* com o maior número de arquivos.
- ▶ O AEH-ETAA apresentou uma melhora média de **22,72%** em relação ao MinMin e de **11,15%** em relação ao HEFT

Resultados Experimentais

Avaliação Prática

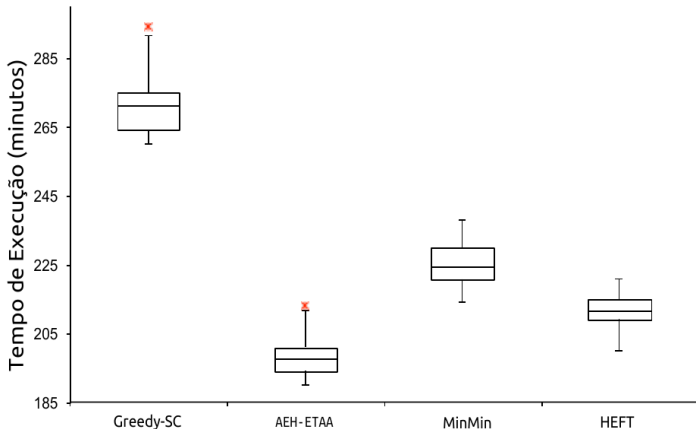
Avaliação prática com *Workflow Sciphy* [Ocaña *et al.*, 2011] utilizando o sistema de gerenciamento de WfCs SciCumulus [Oliveira *et al.*, 2011]

- ▶ Comparação entre AEH-ETAA, Algoritmo Guloso do Sciumulus (Greedy-SC) e as heurísticas MinMin e HEFT
- ▶ Foi utilizado o serviço de nuvem da Amazon EC2
- ▶ Ambiente definido pelo GRASPCC [Coutinho *et al.*, 2015]
- ▶ 1 m3.small (1 vCPU e 4GB RAM) e 2 m3.2xlarge (8 VCPU e 30GB RAM)

Resultados Experimentais

Avaliação Prática

Diagrama de caixa comparando as execuções do SciPhy com Greedy-SC, AEH-ETAA, MinMin, HEFT.



Resultados Experimentais

Avaliação Prática

- ▶ Melhora de **27,4%** em comparação com o Greedy-SC, **11,7%** em comparação com o MinMin e **8,1%** em comparação com o HEFT.

Conclusão e Trabalhos Futuros

- ▶ O AEH-ETAA obteve soluções melhores em relação aos algoritmos avaliados com um tempo de execução aceitável.

Conclusão e Trabalhos Futuros

- ▶ O AEH-ETAA obteve soluções melhores em relação aos algoritmos avaliados com um tempo de execução aceitável.
- ▶ A avaliação com a abordagem exata mostrou que para a maioria das instâncias avaliadas o AEH-ETAA foi capaz de encontrar a solução ótima com um tempo de execução significativamente menor (média de 1,5s).

Conclusão e Trabalhos Futuros

- ▶ O AEH-ETAA obteve soluções melhores em relação aos algoritmos avaliados com um tempo de execução aceitável.
- ▶ A avaliação com a abordagem exata mostrou que para a maioria das instâncias avaliadas o AEH-ETAA foi capaz de encontrar a solução ótima com um tempo de execução significativamente menor (média de 1,5s).
- ▶ A metaheurística apresenta resultados mais significativas para instâncias com um maior número de transferências.

Conclusão e Trabalhos Futuros

Sugestão de trabalhos futuros:

- ▶ Incluir técnicas de tolerância a falhas
- ▶ Redimensionamento dinâmico
- ▶ Incluir outros objetivos, como minimizar os custos financeiros juntamente com o *makespan* (multiobjetivo)

Obrigado!

Restrições do Modelo Matemático

Dados	Descrição
D_s	Conjunto de arquivos estáticos.
D_d	Conjunto de arquivos dinâmicos.
$D = D_s \cup D_d$	Conjunto de arquivos.
$O(d)$	Máquina de origem do arquivo estático $d \in D_s$.
$W(d)$	Tamanho do arquivo $d \in D$.
N	Conjunto de tarefas.
a_i	Quantidade de trabalho da tarefa $i \in N$.
M	Conjunto de Mvs.
t_{ij}	Tempo de processamento da tarefa $i \in N$ na Mv $j \in M$.
\bar{t}_{djp}	Tempo gasto pela Mv $j \in M$ para ler o arquivo $d \in D$ armazenado na Mv $p \in M$
$\leftarrow \bar{t}_{djp}$	Tempo gasto pela Mv $j \in M$ para escrever o arquivo $d \in D_d$ na Mv $p \in M$.
$\Delta_{in}(i) \subseteq D$	Conjunto de arquivos de entrada necessários para a execução da tarefa $i \in N$.
$\Delta_{out}(i) \subseteq D_d$	Conjnto de arquivos de saída gerados pela tarefa $i \in N$.
cm_j	Capacidade de armazenamento da Mv $j \in M$.
Variáveis	Descrição
x_{ijt}	Variável binária que indica se a tarefa $i \in N$ iniciou sua execução na Mv $j \in M$ no período $t \in T$ ou não.
\vec{x}_{idjpt}	Variável binária que indica se a tarefa $i \in N$ executando na Mv $j \in M$ começou a ler o arquivo $d \in \Delta_{in}(i)$, que está armazenado na Mv $p \in M$, no período $t \in T$ ou não.
$\leftarrow x_{djp}$	Variável binária que indica se o arquivo $d \in D_d$ começou a ser escrito a partir da Mv $j \in M$ para a Mv $p \in M$ no período $t \in T$ ou não.
y_{djte}	Variável binária que indica se o arquivo $d \in D$ está armazenado na Mv $j \in M$ no período $t \in T$ ou não.
z_T	Variável contínua que indica o tempo total de execução (<i>makespan</i>) do workflow.

Restrições do Modelo Matemático

A restrição 1 garante que cada tarefa seja executada. As restrições 2 e 3 certificam que todas as operações de leitura e escrita sejam realizadas, respectivamente.

Sujeito a

$$\sum_{j \in M} \sum_{t \in T} x_{ijt} = 1, \quad \forall i \in N \quad (1)$$

$$\sum_{j, p \in M} \sum_{t \in T} \vec{x}_{idjpt} = 1, \quad \forall i \in N, \forall d \in \Delta_{in}(i) \quad (2)$$

$$\sum_{j, p \in M} \sum_{t \in T} \overleftarrow{x}_{djpt} = 1, \quad \forall d \in D_d \quad (3)$$

Restrições do Modelo Matemático

A inequação (4) garante que o dado $d \in \Delta_{out}(i)$ seja escrito apenas se a tarefa i tenha sido executada no tempo correto.

$$\begin{aligned} \overleftarrow{x}_{djpt} &\leq x_{ij}(t - t_{ij}), & \forall d \in D_d, \forall j, p \in M, \\ & \forall t = (t_{ij} + 1) \cdots T_M \text{ tal que } d \in \Delta_{out}(i) \end{aligned} \quad (4)$$

Restrições do Modelo Matemático

As restrições definidas em (5) asseguram que o dado d não possa ser escrito antes do tempo de processamento da tarefa i (responsável pela sua escrita). Note que ambas as restrições (4 e 5) trabalham em conjunto para garantir um tempo de escrita factível.

$$\overleftarrow{x}_{djpt} = 0 \quad \forall d \in D_d, \forall j, p \in M, \quad 1 \leq t \leq t_{ij} \text{ tal que } d \in \Delta_{out}(i) \quad (5)$$

Restrições do Modelo Matemático

A restrição definida em (6) assegura que uma tarefa só possa ser executada quando todas as operações de leitura estiverem concluídas.

$$x_{ijt} \leq \sum_{p \in M} \vec{x}_{idjp(t - \vec{t}_{djp})}, \quad \forall i \in N, \forall d \in \Delta_{in}(i), \forall j \in M, \\ \forall t \in T, \text{ tal que } (t - \vec{t}_{djp}) \geq 1 \quad (6)$$

Restrições do Modelo Matemático

A desigualdade (7) garante que apenas uma ação (execução, leitura ou escrita) possa ser realizada em cada período de tempo em cada MV. Ou seja, a MV não pode executar uma tarefa e escrever, ou ler, um dado ao mesmo tempo.

$$\begin{aligned}
 & \sum_{i \in N} \sum_{q=\max(1, t-t_{ij}+1)}^t x_{ijq} + \sum_{d \in D_d} \sum_{p \in M} \\
 & \sum_{r=\max(1, t-\overleftarrow{t}_{djp}+1)}^t \overleftarrow{x}_{djpr} + \\
 & \sum_{i \in N} \sum_{d \in \Delta_{in}(i)} \sum_{p \in M} \sum_{r=\max(1, t-\overrightarrow{t}_{djp}+1)}^t \overrightarrow{x}_{idjpr} \leq 1, \quad \forall j \in M, \forall t \in T \quad (7)
 \end{aligned}$$

Restrições do Modelo Matemático

A restrição (8) estabelece que não há arquivos dinâmicos no tempo inicial. A restrição (9) garante que todos os arquivos estáticos estejam prontamente armazenados em suas máquinas de origem.

$$y_{dj1} = 0, \quad \forall d \in D_d, \forall j \in M \quad (8)$$

$$y_{djt} = 1, \quad \forall d \in D_s \mid j \in O(d), \forall t \in T \quad (9)$$

Restrições do Modelo Matemático

As restrições (10) e (11) relacionam a variável de armazenamento y com as variáveis de escrita \overleftarrow{x} e de leitura \overrightarrow{x} , garantindo um processo viável de escrita e leitura.

$$y_{dp(t+1)} \leq y_{dpt} + \sum_{j \in M} \overleftarrow{x}_{djp(t - \overleftarrow{t}_{djp})}, \forall d \in D, \forall p \in M, \\ \forall t \in T, \\ \text{tal que } (t - \overrightarrow{t}_{djp}) \geq 1 \quad (10)$$

Restrições do Modelo Matemático

A restrição (11) garante que os arquivos serão lidos apenas se estes estiverem previamente armazenados em uma MV, e a restrição (10) assegura que um arquivo seja armazenado em MV apenas caso ele tenha sido produzido (escrito).

$$\sum_{j \in M} \vec{x}_{idjpt} \leq y_{dpt} \quad , \forall i \in N, \forall d \in \Delta_{in}(i), \forall p \in M, \forall t \in T \quad (11)$$

Restrições do Modelo Matemático

A capacidade de armazenamento das MVs é estabelecida pela restrição (12). A restrição (13) relaciona a última operação de escrita com o tempo total de execução da aplicação (*makespan*). Note que no modelo da aplicação, uma tarefa sempre escreve ao menos um arquivo.

$$\sum_{d \in D} y_{djt} W(d) \leq cm_j, \forall j \in M, \forall t \in T \quad (12)$$

$$\overleftarrow{x}_{djpt} \cdot (t + \overleftarrow{t}_{dj p}) \leq z_T, \forall d \in D_d, \forall j, p \in M, \forall t \in T \quad (13)$$

Restrições do Modelo Matemático

A restrição operacional (14) deve ser satisfeita: uma tarefa i pode iniciar um processo de leitura se todos os arquivos $d \in \Delta_{in}(i)$ estiverem disponíveis (isto é, se todos os arquivos $d \in (\Delta_{in}(i) \cap D_d)$ forem escritos).

$$\vec{x}_{idjpt} \cdot |\Delta_{in}(i) \cap D_d| \leq \sum_{g \in \{\Delta_{in}(i) \cap D_d\}} \sum_{l, o \in M} \sum_{u=1}^{t - \overleftarrow{t}_{glo}} \overleftarrow{x}_{glou}, \quad \forall i \in N,$$

$$\forall d \in \Delta_{in}(i),$$

$$\forall j, p \in M,$$

$$\forall t \in T$$

(14)