

Predição do impacto do fluxo de dados no desempenho de workflows científicos

Duílio Henrique Haroldo Elias

RELATÓRIO FINAL DE INICIAÇÃO CIENTÍFICA

Programa: Programa Institucional de Bolsas de
Iniciação Científica em Desenvolvimento Tecnológico
e Inovação – PIBITI/CNPq/USP
Orientadora: Kelly Rosa Braghetto

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

24 de junho de 2014

Resumo

Nas últimas décadas os workflows científicos receberam grande atenção da comunidade acadêmica, devido à sua grande capacidade de manipulação, análise e simulação de experimentos que trabalham com grandes quantidades de dados. No entanto, as ferramentas existentes para tal fim são ainda escassas e muitos cientistas afirmam que isso é um grande gargalo da produção científica. Neste trabalho, busca-se contribuir com este processo, a partir da identificação e caracterização de linguagens de modelagem de workflows científicos, criando modelos estocásticos capazes de descrever analiticamente alguns workflows bem estudado pela comunidade acadêmica. Com isso, pretende-se extrair índices capazes de prever o impacto do uso de diferentes abordagens de controle de fluxo de dados no desempenho dos workflows e compara-los com os dados das simulações.

Palavras-chaves: Workflows científicos. Redes de Petri. Análise de Desempenho. Gerenciadores de Workflows.

Sumário

Sumário	3
1 Introdução	4
2 Objetivo	4
3 Conceitos Preliminares	5
3.1 Workflows científicos	5
3.2 Sistemas gerenciadores de workflows científicos	7
3.2.1 Taverna	7
3.2.2 Pegasus	8
3.3 Redes de Petri	9
3.3.1 PIPE	11
3.4 Simuladores de workflows científicos	11
4 Modelagem Analítica	12
5 Metodologia	12
6 Análise	13
7 Conclusões	13
Referências	14
8 Anexos	15
8.1 Anexo 1	15
8.2 Anexo 2	15
8.3 Anexo 3	17

1 Introdução

Um workflow científico(WfC) pode ser compreendido como um conjunto de tarefas computacionais que constituem um experimento científico[Júnior, 2012]. As principais etapas do ciclo de vida de um WfC são a modelagem, execução e análise. O Pesquisador pode defini-las com o auxílio de uma ferramenta de software, com uma interface gráfica simples e intuitiva. Depois de criado um modelo, o sistema gerenciador de WfC auxilia o pesquisador a executar as tarefas do seu WfC de forma eficiente, automaticamente, com pouca ou nenhuma intervenção manual.

Os WfCs são conhecidos pelo seu poder de manipulação e execução de experimentos científicos automatizáveis que trabalham com grandes quantidades de dados. Neste trabalho, estamos interessados em melhorar o desempenho da execução de workflows em ambientes computacionais distribuídos, o que muitas vezes não é uma tarefa fácil. Para isso, o sistema gerenciador precisa considerar a relação de dependência que existe entre as tarefas do workflow, ou seja, precisa definir quais são os dados de entrada de uma determinada tarefa, a fonte desses dados e destino dos dados processados. Há também um custo envolvido nas transferências de dados, em termos de tempo, que precisa ser considerado e que depende do ambiente computacional.

Todas essas observações se relacionam ao modelo de workflow de duas formas diferentes: primeiro, pela relação de dependência existente entre as tarefas, ou seja, como os dados irão percorrer o workflow e, segundo, quais as características computacionais do ambiente em que o workflow será executado.

O conjunto de conexões que interligam as atividades pode ser caracterizado como fluxo de dados, esse fluxo de dados define as dependências entre as atividades e os dados manipulados[Teixeira, 2013].

2 Objetivo

O objetivo deste trabalho foi identificar e caracterizar os principais construtores de fluxo de dados encontrados em linguagens de modelagem de workflows científicos, bem como criar modelos estocásticos em Redes de Petri capazes de prever o impacto causado pelo fluxo de dados no desempenho da execução de workflows. Os estudos conduzidos no trabalho se basearam em um conjunto de modelos de workflow científicos que são amplamente usados na avaliação de sistema gerenciadores de workflows. As previsões obtidas a partir do modelo de Redes de Petri foram comparadas com as obtidas a partir da simulação da execução dos

mesmos workflows na ferramenta de simulação WorkflowSim¹.

3 Conceitos Preliminares

3.1 Workflows científicos

O termo workflow tem sua origem ligada a processos de automação de escritórios que surgiram na década de 70. Com o objetivo de gerar uma tecnologia capaz de fornecer uma base de apoio à distribuição de documentos nos escritórios, o que resultou na diminuição do uso de papéis[Ogasawara, 2011]. Embora a sua origem esteja mais relacionada com a automação de tarefas de logística, essa teoria foi expandida para outras áreas do conhecimento, como Economia, Administração, até mesmo Física, Biologia e Oceanográfica. Segundo *Workflow Management Coalition*[WfM, 2014], um workflow pode ser definido como: A automação de um processo de negócio, completo ou apenas parte dele, através do qual documentos, informações ou tarefas são transmitidos de um participante a outro por ações, de acordo com regras procedimentais.

Já o termo workflow científico(WfC) está mais ligado com as ciências do que o termo workflow, por ter o seu foco na manipulação de dados e na execução de tarefas que estão diretamente relacionadas com o processamento de dados. Essa grande utilização de dados caracteriza os WCs como uma estrutura baseada no fluxo de dados, ou seja, as conexões entre as atividades representam o fluxo de dados.[Teixeira, 2013]

Os sistemas de WC geralmente são compostos por um grande número de tarefas que podem depender dos dados produzidos por outras tarefas. Essas tarefas podem ser manuais ou automáticas e são responsáveis pela gerência de todo o ciclo de vida do experimento científico: composição, mapeamento, execução e análise. Cada etapa do ciclo do experimento é muito importante e precisa ser analisada com cuidado[Júnior, 2012] e estão descritas com mais detalhes logo abaixo:

- Composição: especificações das tarefas que compõem o experimento, dos tipos de dados de entrada e saída e das dependências de dados existentes entre as atividades e o ambiente computacional em que o workflows será executado.
- Mapeamento: associação do modelo de workflow com os recursos computacionais disponíveis.

¹ <http://www.workflowsim.org/>

- Execução: realização das tarefas definidas anteriormente no ambiente computacional.
- Análise: estudo dos dados gerados na execução.

Todo esse processo pode ser automatizado de forma que o cientista possa reexecutá-lo até que os dados obtidos na fase de análise sejam os dados esperados.

A estrutura de um WfC é composta pelas atividades a serem executadas e as conexões entre elas. Compreende-se por atividade cada ação realizada que recebe um dado de entrada e produz um dado ou uma informação de saída. O conjunto de conexões entre as atividades pode ser caracterizado como fluxo de controle ou fluxo de dados. Neste trabalho estamos interessados em analisar apenas o fluxo de dados, pois é a estrutura que está mais relacionada com os WCs. O fluxo de dados representa a dependência entre as atividades e os dados manipulados, e a forma como os dados são utilizados no workflow, com atividades produzindo dados que serão consumidos por outras atividades[Teixeira, 2013].

As principais construções de fluxo de dados são[Teixeira, 2013]:

- *Pipeline*: são várias atividades combinadas sequencialmente, em que cada atividade recebe como entrada os dados produzidos como saída da atividade anterior;
- *Distribuição de dados*: é feita por atividades que produzem dados de saída que são recebidos como entrada por múltiplas atividades, ou que apenas particionam grandes conjuntos de dados em subconjuntos menores para serem processados por outras atividades no workflow;
- *Redistribuição de dados*: é feita por atividades que funciona como um ponto de sincronização com relação ao processamento de dados, recebendo como entrada várias porções de dados e produzindo também múltiplas saídas como resultado.
- *Agregação de dados*: é feita por atividades que processam múltiplos conjuntos de dados de entrada, gerando uma combinação dos dados como saída;

A figura ?? representa algumas dessas construções e pode facilitar a compreensão. Observe também que nesta figura temos a modelagem analítica do Montage que tópico será melhor abordado nas próximas seções.

3.2 Sistemas gerenciadores de workflows científicos

Para apoiar os WfC, diversos Sistemas de Gerenciamento de Workflows Científicos(SGWfC) foram desenvolvidos. Cada um deles apresentam uma particularidades para representação dos workflows e visa atender determinados tipos de aplicações de áreas específicas. Estes SGWfC são responsáveis pela coordenação das chamadas a programas, seja em ambiente local ou para ambientes distribuídos[Ogasawara, 2011]. Os Sistemas mais utilizados, segundo o número de workflows compartilhados no *myexperiment*², repositório de workflows científicos, estão descritos de maneira sucinta nesta nas próximas subseções:

3.2.1 Taverna

O Taverna foi desenvolvido como parte do projeto *mygrid*³, é uma ferramenta de código aberto, multiplataforma e possui uma boa documentação. Com um ambiente para modelagem, execução e análise de workflows científicos, tem como foco principal apoiar cientistas da área de Ciências Biológicas que normalmente não possuem conhecimento profundo de computação, para facilitar a construção o Taverna possui uma interface gráfica que permite ao pesquisador construir workflows de forma muito parecida com fluxogramas.[Teixeira, 2013].

Um workflow no Taverna é especificado como um grafo direcionado onde os nós, chamados de processadores, são as atividades e representam componentes de software. Um nó processador consome dados que chegam nas suas entradas e produzem dados de saída. Cada arco no grafo conecta um entrada com uma saída de dados e expressa uma dependência entre as atividades o que denota o fluxo de dados entre as atividades[Teixeira, 2013].

Toda a construção e execução dos workflows no Taverna é pode ser feita com auxílio da interface gráfica, que é representado posteriormente na linguagem Scufl(*Simplified Conceptual Workflow Language*) que descreve o workflow em um formato baseada em XML.[Teixeira, 2013].

A linguagem Schfl possui uma serie de componentes que permite a representar no seu formato todo o workflow. A construção mais básica do Taverna são os *Pipelines*, porém é possível criar construções de fluxos de dados como Agregação, Distribuição e Redistribuição de dados [Teixeira, 2013].

Como pode ser visto na figura 1 abaixo o Taverna possui uma integração com o *myexperiment* para compartilhamento dos workflows

² <http://www.myexperiment.org/>

³ <http://www.mygrid.org.uk/>

modelados nele o que possivelmente o tornou um dos SGWC mais conhecidos, além de todo seu poder de construção de workflows científicos voltados para área de Biológicas.

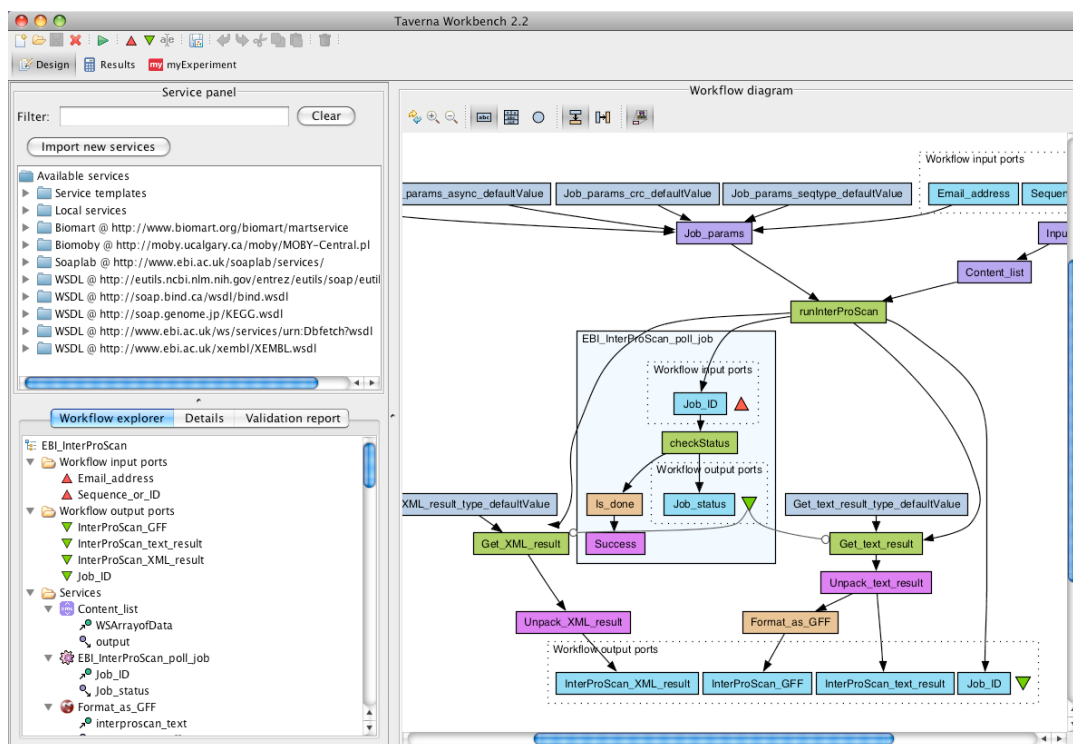


Figura 1 – Interface gráfica do Taverna [Tav, 2014]

3.2.2 Pegasus

O SGWC Pegasus é um *framework* que permite o mapeamento de instâncias de workflows em um conjunto de recursos distribuídos, como uma grade. A descrição de um workflow no Pegasus é feita a partir de um arquivo de entrada na forma de um DAG em formato XML, na linguagem DAX (*Directed Acyclic Graph in XML*), que representa todo o fluxo de processamento e as dependências entre as tarefas, além dos recursos computacionais em que o workflow será executado [Teixeira, 2013].

O DAX descreve os workflows termos de atividades, onde cada atividade consiste de um código de identificação da atividade, os nomes dos arquivos de entrada e saída de dados e uma relação de dependência entre as atividades. A partir dessa definição de dependências é possível construir as principais estruturas de fluxos de dados, *pipelines*, distribuição, agregação e redistribuição de dados. O DAX também permite a definição de vários arquivos de saída que podem ser de tipos diferentes, dependendo da necessidade da atividade [Teixeira, 2013].

Código abaixo mostra como é representada uma atividade na linguagem DAX⁴:

Código- 1 – Exemplo de uma atividade DAX

```

1
2 <job id="ID000002" namespace="vahi" name="findrange"
   version="1.0"
3 level="2" dv-namespace="vahi" dv-name="left" dv-version="
   1.0">
4   <argument>
5     -a left -T 6 -i
6     <filename file="f.b1"/> -o <filename file="f.c1"/>
7     -p 0.5
8   </argument>
9   <uses file="f.b1" link="input" dontRegister="false"
      dontTransfer="false" temporaryHint="true"/>
10  <uses file="f.c1" link="output" dontRegister="true"
      dontTransfer="false" temporaryHint="true"/>
11 </job>

```

Numa próxima fase o DAX é mapeado pelo Pegasus e as atividades definidas são executadas.

3.3 Redes de Petri

As Redes Petri(RdP) são um dos formalismos mais utilizados para modelagem analítica e análise de sistemas concorrentes mais utilizados, e isso deve-se à fácil compreensão, representação gráfica e devido ao seu potencial matemático para análise de modelos. Algumas características importantes de workflows científicos podem ser facilmente representadas utilizando RdP, tais como a sincronização de tarefas, concorrência e controle de dependências[Braghetto, 2011]. Uma RdP pode ser formalmente definida pela tupla $RdP = L, T, A, P, m_0$, em que:

- $L = \{l_1, l_2, \dots, l_L\}$ é um conjunto de lugares;
- $T = \{t_1, t_2, \dots, t_T\}$ é um conjunto de transições;
- $A \subseteq (L \times T) \cup (T \times L)$ é um conjunto de arcos;
- $P : A \rightarrow \mathbb{N}$ é a função peso dos arcos;
- $m_0 = \{m_{01}, m_{02}, \dots, m_{0L}\}$ é a marcação inicial da rede;

⁴ <https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator>, Este exemplo foi retirado do DAX que representa o Montage

Cada uma dessas definições está associadas a diferentes e importantes conceitos de RdP, que são:

- *lugar* (representado graficamente por um círculo)- modela uma condição que deve ser satisfeita para que o disparo da transição seja realizado;
- *transição* (representado graficamente por um retângulo) - pode ser compreendido por uma por uma ação ou um evento;
- *arco orientado* liga um lugar a uma transição ou o contrário, ligando condições e eventos;
- *fichas, marcas ou tokens* (representado por um ponto preto)- Representam um recurso ou um estado de um sistema;
- *peso* - os arcos possuem um peso associado a eles; os pesos indica quantas fichas uma transição consome de um lugar de entrada ou quantas fichas uma transição acrescenta em um lugar de saída. Por *default* quando um arco não possui um peso o peso é 1;

Da necessidade de modelagem de sistemas grandes e complexos, como os encontrados na natureza e na sociedade, foram definidas sobre a base de Redes de Petri clássica outras redes de alto nível, como as Redes de Petri Temporizadas(RdPTs) e Redes de Petri Estocásticas(RdPEs). As RdPTs como o nome diz, permite modelar situações que ocorrem entre o início e fim de cada tarefa em termos do tempos. O que permitiu diferentes possibilidades para modelagem, como lugares temporizados, fichas temporizadas e todas as outras estruturas descritas acima puderam ser temporizadas e permitiram modelar uma série de sistemas reais. Já as RdPEs é uma subclasse das RdPTs, onde o tempo de atraso de uma transição é uma variável aleatória exponencialmente distribuída[Braghetto, 2011].

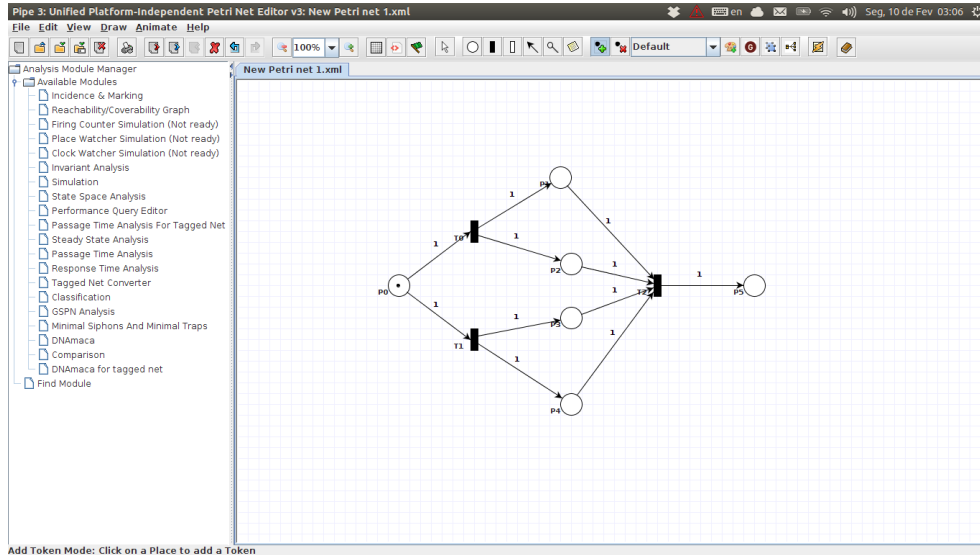
Ao se usar as redes de Petri para se modelar a execução de um workflow, cada vértice pode armazenar um ou mais tokens, os vértices de transição de redes de Petri podem consumir e produzir tokens de múltiplas localizações. Os vértices de transições agem em tokens de entrada por um processo denominado disparo. Quando uma transição é disparada, ela consome os tokens de suas posições de entrada, realiza alguma tarefa de processamento, e realoca um número específico de tokens nas suas posições de saída. Isso é feito atomicamente. Como os disparos são não-determinísticos, as redes de Petri são muito utilizadas para modelar comportamento concorrente em sistemas distribuídos[Ogasawara, 2011],

assim como é possível associar a cada transição uma taxa de disparo. Neste trabalho utilizaremos o *PIPE* para fazer a modelagem analítica que será descrito brevemente a seguir.

3.3.1 PIPE

O PIPE[Bonet et al., 2007] é uma ferramenta *open source* criada para análise de RdPEs, possui uma interface gráfica simples e intuitiva. Com um conjunto de módulos para avaliar o comportamento estatístico das RdPEs, a representação de Lugares, Transições, Arcos e Fichas, segue o modelo gráfico descrito na seção anterior, como pode ser visto na figura abaixo:

Figura 2 – Interface gráfica-PIPE



Após a construção gráfica do modelo é possível observar a evolução dinâmica do sistema modelado e fazer a análise dos dados gerados, assim como exportar em formato XML ou o PNG da Rede Petri representada.

3.4 Simuladores de workflows científicos

Depois de criados os modelos em RdPEs para workflows científicos há algumas maneiras de validar o modelo, a primeira possibilidade é execução do workflows. Essa alternativa possui como problema o custo monetário envolvido na instanciação das máquinas, o que a torna inviável no primeiro momento. Uma outra alternativa mais viável é a utilização de um ambiente de simulação computacional. Essa alternativa abre a

possibilidade de criar-se diferentes ambientes computacionais e facilidade de reprodução da simulação, o que torna viável a busca por gargalos no workflows utilizados. Além da possibilidade avaliar modelos analíticos em um ambiente controlado que é um dos objetivos desse trabalho.

Nesse trabalho utilizou-se o WorkflowSim[Chen and Deelman, 2012] que é uma ferramenta de simulação de workflows científicos baseada no CloudSim[Google, 2013], que também é uma ferramenta de simulação. No entanto, o CloudSim não possui certas características presentes no WorkflowSim, por exemplo, a simulação de ambientes heterogêneos e possibilidade de incluir falhas no sistema. As atividades no WorkflowSim são descritas na linguagem DAX (linguagem para a descrição de workflows em formato XML usada pelo SGWC Pegasus). O WorkflowSim, possibilita também, modelar todo um ambiente computacional para execução dos workflows(Quantidade de *Virtual Machine(VM)* e *DataCenters*). Observe no Anexo 4 como é descrito um ambiente computacional no WorkflowSim.

4 Modelagem Analítica

Ao se usar as redes de Petri para se modelar a execução de um workflow, cada vértice pode armazenar um ou mais tokens. Diferentemente de sistemas mais tradicionais de processamento de dados, que comumente processam somente um único fluxo de tokens entrantes, os vértices de transição de redes de Petri podem consumir e produzir tokens de múltiplas localizações. Os vértices de transições agem em tokens de entrada por um processo denominado disparo. Quando uma transição é disparada, ela consome os tokens de suas posições de entrada, realiza alguma tarefa de processamento,e realoca um número específico de tokens nas suas posições de saída. Isso é feito atomicamente. Como os disparos são não determinísticos, as redes de Petri são muito utilizadas para modelar comportamento concorrente em sistemas distribuídos.

5 Metodologia

Neste projeto foi realizado um estudo sobre os conceitos básicos envolvendo workflows científicos, Sistemas Gerenciadores de Workflows Científicos, Redes de Petri, ferramentas de representação de redes de Petri e modelagem analítica de workflows científicos, assim como um estudo de ferramentas para simulação de workflows.

Os workflows científicos foram escolhidos para serem modelados neste trabalho possuem estruturas bem definidas, que explicam o uso

de construções de fluxo de dados mais comuns e são bastante estudados comunidade acadêmica.

Depois do estudo da bibliografia básica envolvendo workflow científicos e as motivações acadêmicas para a pesquisa de workflows científicos, foi feito o estudo de alguns SGWC, quanto a implementação de workflows e a representação do fluxo de dados em cada um deles.

Para aparar o formalismo matemático envolvendo workflows científicos e fazer uma modelagem analítica de alguns workflows bem conhecidos foram estudas Redes de Petri, Redes de Petri Temporizadas e Redes de Petri Estocásticas, assim como uma ferramenta para representação gráfica e análise das RdPEs(PIPE).

Para finalizar a primeira parte do trabalho foi feito um estudo do WorkFlowSim, quanto possibilidade de ferramenta de simulação para o projeto.

6 Análise

Na primeira etapa desse projeto foram obtidos conhecimentos básico de workflows científicos, SGWC, RdPEs, modelagem analítica de workflow científicos. Para uma melhor interpretação foi feita a modelagem analítica do fluxo de dados do Montage, partir do Anexo 1, [Pegasus Project, 2012] com o auxilio PIPE. Como pode vista na figura do Anexo 2, que permitirá extrair índices de desempenho e verificar a validade do modelo a partir da simulação do Montage utilizando o Work-flowSim.

7 Conclusões

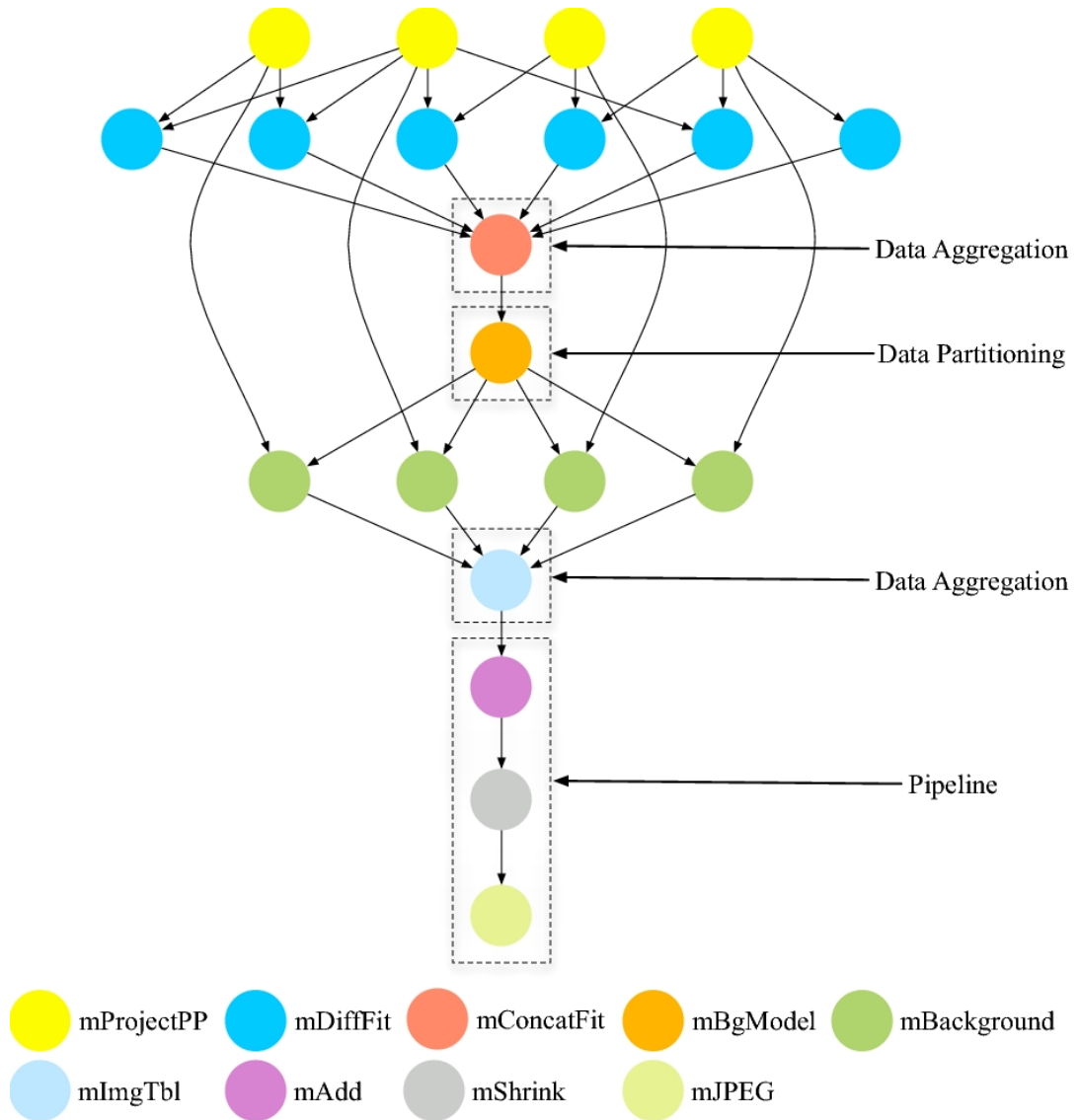
Referências

- [Tav, 2014] (2014). Taverna. www.taverna.org.uk/pages/wp-content/uploads/2010/06/. Último acesso: 2014-06-23.
- [WfM, 2014] (2014). Workflow management coalition. <http://www.wfmc.org/>. Último acesso: 2014-02-06.
- [Bonet et al., 2007] Bonet, P., Lladó, C. M., Puijaner, R., and Knottenbelt, W. J. (2007). Pipe v2. 5: A petri net tool for performance modelling. In *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*.
- [Braghetto, 2011] Braghetto, K. (2011). Técnicas de modelagem para a análise de desempenho de processos de negócio. Doutorado, Universidade de São Paulo, USP-São Paulo.
- [Chen and Deelman, 2012] Chen, W. and Deelman, E. (2012). Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–8.
- [Google, 2013] Google (2013). Calheiros: Cloudsim: a toolkit for modeling and simulation of ... - google scholar. http://scholar.google.com.br/scholar?cites=272054103506592999&as_sdt=2005&scioldt=0,5&hl=pt-BR. [Online; acessado em 01 de fevereiro de 2014].
- [Júnior, 2012] Júnior, L. (2012). Gerência de proveniência em workflow científicos para e distribuídos. Doutorado, Universidade Federal do Rio de Janeiro, UFRJ-Rio de Janeiro.
- [Ogasawara, 2011] Ogasawara, E. S. (2011). Uma abordagem algébrica para workflows científicos com dados em larga escala. Doutorado, Universidade Federal do Rio de Janeiro, UFRJ-Rio de Janeiro.
- [Pegasus Project, 2012] Pegasus Project (2012). WorkflowGenerator. <https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator>. [Online; acessado em 1 de setembro de 2013].
- [Teixeira, 2013] Teixeira, E. (2013). Modelagem e escalonamento de workflows científicos para execução em plataforma de computação em nuvem. Doutorado-qualificação, Universidade de São Paulo, USP-São Paulo.

8 Anexos

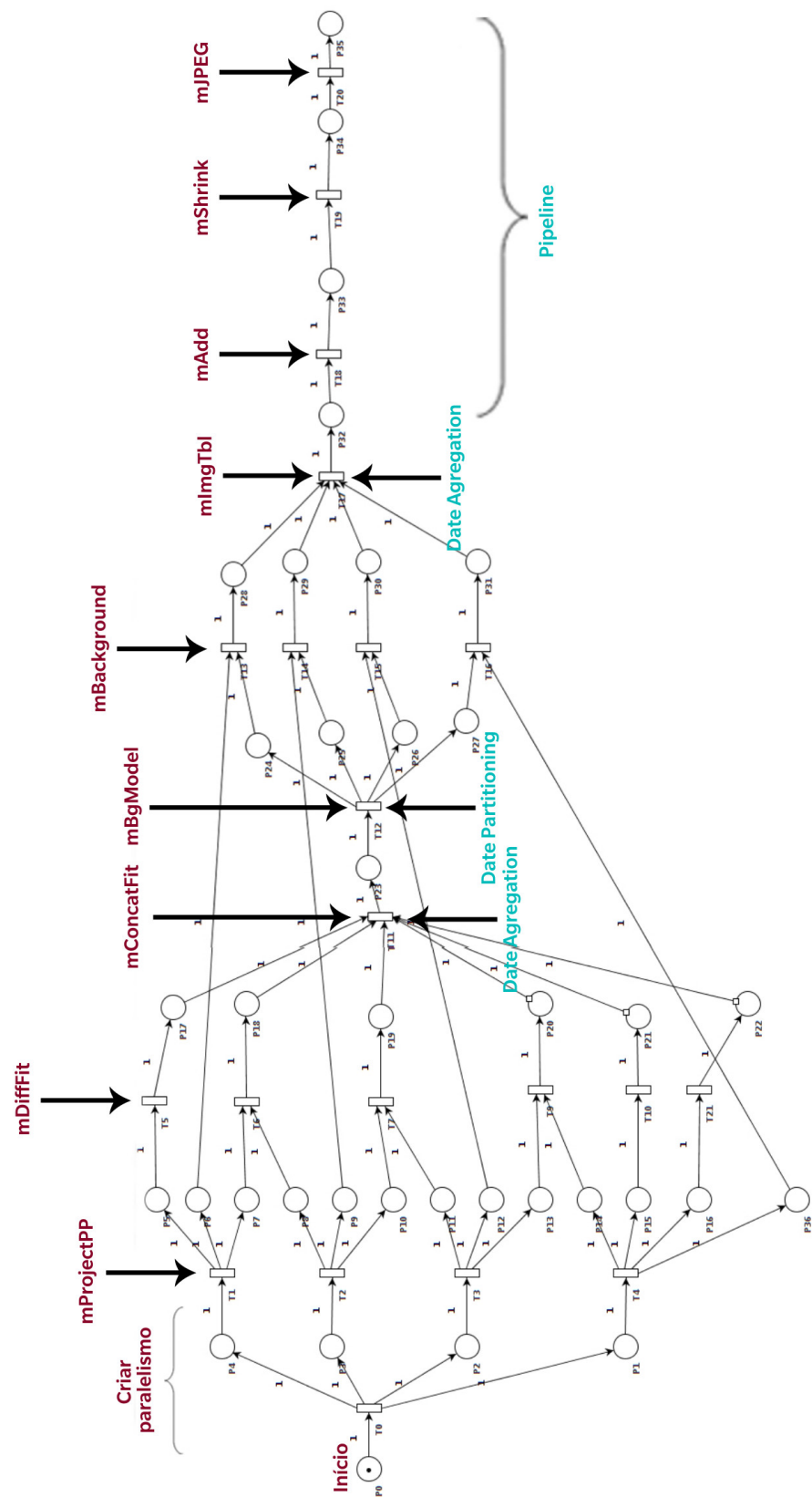
8.1 Anexo 1

Figura 3 – Ilustração do workflow do Montage



8.2 Anexo 2

Figura 4 – Representação Gráfica do Montagem usando Redes de Petri



8.3 Anexo 3

Código- 2 – Descrição de uma ambiente computacional no WorkflowSim

```
1
2 #####Overhead Related Parameters
3 #####The Workflow Engine Delay for level 1
4 #d1      = 8.1
5 #####The Workflow Engine Delay for level 2
6 #d2      = 18.44
7 #####The Workflow Engine Delay for other levels that are
      not specified
8 #d0      = 6.74
9 #####The Postscript Delay for level 1
10 #p1     = 78.05
11 #####The Postscript Delay for level 2
12 #p2     = 13.12
13 #####The Postscript Delay for other levels that are not
      specified
14 #p0     = 6.81
15 #####The Queue Delay for level 1
16 #q1     = 77.75
17 #####The Queue Delay for level 2
18 #q2     = 343.47
19 #####The Queue Delay for other levels that are not
      specified
20 #q0     = 161.96
21 #####The Clustering Delay for level 1
22 #c1     = 428.28
23 #####The Clustering Delay for level 2
24 #c2     = 67.44
25 #####The interval for Workflow Engine
26 #interval = 5
27
28 #####Failure related parameters
29 #####The Task Failure Rate for level 1
30 #a1     = 0.02
31 #####The Task Failure Rate for level 2
32 #a2     = 0.02
33 #####Fault tolerant clustering method
34 #ftc.method = FTCLUSTERING_DC
35 #####Fault tolerant clustering monitor mode
36 #ftc.monitor = MONITOR_ALL
37 #####Fault tolerant clustering failure generation mode
38 #ftc.failure = FAILURE_JOB
39
40 #####Clustering Related parameters
41 #####The number of clustered jobs per level
42 #clusters.num = 20
43 #####The size of a clustered job (number of tasks per job
      )
44 #clusters.size = 20
45 #####The clustering method
```

```

46 #clusters.method = horizontal
47 #clusters.method = balanced
48 #####The type of file system to use
49 file.system = LOCAL
50 #file.system    = SHARED
51
52 #####Scheduling Related Parameters
53 #####If you have specified planner.method, it will be
    disabled
54 scheduler.method= MINMIN_SCH
55
56 #####Planning Related Parameters
57 #planner.method= RANDOM
58 #deadline = 100000
59
60 #####Please replace it with your real physical path
61 dax.path = /home/duilio/workspace/WorkflowSim-1.0-master/
    config/dax/Montage_1000.xml
62 #####Reducer mode used to remove duplicate dependencies
63 #reduce.method = montage

```