Uma Ferramenta de Software para a Predição de Desempenho de Workflows Científicos*

Lucas Magno[†], ***Kelly Rosa Braghetto***[‡]

Universidade de São Paulo / [†]Instituto de Física, [‡]Instituto de Matemática e Estatística

lucas.magno@usp.br | kellyrb@ime.usp.br

Resumo

Experimentos científicos frequentemente lidam com grandes quantidades de dados e processos com altos custos computacionais, exigindo a análise de seus desempenhos através de métodos que não envolvam suas execuções. Um destes métodos é a representação dos experimentos em modelos de *workflow* por meio de linguagens formais, que permitem a extração de predições quantitativas de desempenho. Entretanto, o uso dessas linguagens requer familiaridade com formalismos complexos, restringindo-o às áreas da ciência com maior domínio em computação.

Neste projeto, portanto, foi desenvolvida uma ferramenta em *Python* capaz de, a partir da descrição de um *workflow* em uma linguagem simples e intuitiva, gerar suas predições de desempenho através da álgebra de processos estocástica *PEPA*, permitindo a otimização de experimentos científicos por usuários não-especialistas.

Abstract

*Este trabalho foi financiado por uma bolsa de iniciação científica CNPq/PIBIC (processo: 155544/2013-6).

Introdução

Inicialmente desenvolvidos para automatizar processos industriais e empresariais, os *workflows* se popularizaram e passaram a ser usados na modelagem e automatização de experimentos científicos em diversas áreas da ciência. Um *workflow científico* é a descrição completa ou parcial de um experimento científico em termo de suas atividades, controles de fluxo e dependência de dados [6].

Por ser comum em *workflows* científicos a manipulação de enormes quantidades de dados e a presença de processos que consomem muitos recursos computacionais, ferramentas que forneçam predições de desempenho desse tipo de sistema fazem-se necessárias. As predições auxiliam na escolha dos recursos computacionais apropriados para a execução e na identificação de possíveis problemas na modelagem dos *workflows*, possibilitando assim execuções mais eficientes.

Há várias maneiras de se representar um *workflow científico*, como, por exemplo, por meio de *grafos direcionados*, *UML* (Unified Modeling Language), *redes de Petri* e *álgebras de processos* [7]. Estes mecanismos de representação são usados para criar modelos que especificam a ordem de execução das atividades pertencentes aos *workflows*. Linguagens formais como as redes de Petri e as álgebras de processos, vão além de uma simples representação, pois permitem também que se verifique propriedades qualitativas e quantitativas dos modelos de *workflow* nelas representados. Em particular, extensões estocásticas dessas linguagens frequentemente são usadas para a criação de modelos preditivos do desempenho de sistemas computacionais.

Apesar dos inúmeros benefícios que esses formalismos podem trazer à modelagem de *workflows*, o seu uso na prática impõe algumas dificuldades. Ele exige que o usuário tenha familiaridade com linguagens e modelos estocásticos (de compreensão pouco intuitiva) e com seus programas de simulação ou análise numérica. No entanto, os *workflows* científicos são criados e manipulados por cientistas das mais diversas áreas da ciência e que, geralmente, não são especialistas em computação.

Objetivos

Este trabalho aborda o problema da predição de desempenho de *workflows* científicos, propondo uma ferramenta de software que automatiza a geração de predições baseadas em modelos estocásticos. O objetivo da ferramenta é esconder do usuário final a complexidade associada à criação e análise desse tipo de modelo.

A ferramenta – chamada de `wkf2pepa` – gera automaticamente modelos estocásticos a partir de modelos de *workflows* descritos em uma linguagem bastante simples e de compreensão intuitiva, fácil de ser usada por cientistas de qualquer domínio. A ferramenta obtém a solução numérica dos modelos estocásticos e então extrai índices de desempenho relacionados aos modelos de *workflows* fornecidos como entrada.

Materiais e Métodos

Os modelos de *workflows* usados como entrada para a ferramenta `wkf2pepa` são descritos textualmente na forma de um grafo dirigido – uma representação simples e que pode ser usada com facilidade por usuários não especialistas. Para a criação dos modelos estocásticos, optou-se pelo uso da linguagem *PEPA – Performance Evaluation Process Algebra*, uma álgebra de processos estocástica bem estabelecida e que conta com várias ferramentas de apoio.

Neste trabalho, considera-se que *workflows* sejam compostos por *atividades*, que representam atividades reais de um experimento, e estruturas para descrever o fluxo de controle, como *sequência*, *paralelismo*, *escolha* e *sincronização*, definidas por meio dos operadores *AND* (paralelismo/sincronização), *XOR* (escolha exclusiva/junção) e *OR* (escolha múltipla/junção). Para que possua um modelo correspondente em *PEPA*, um modelo de *workflow* precisa ser bem estruturado e não possuir ambiguidades semânticas. Por essa razão, neste trabalho são considerados apenas modelos de *workflow* que apresentam somente um ponto de entrada e um ponto de saída, têm sua estrutura em forma de "blocos" e não apresentam ciclos, ou laços.

A `wkf2pepa` foi implementada na linguagem *Python*. Ela usa a bibllioteca *pyPEPA*[5], uma implementação recente de um solucionador para *PEPA* em *Python*, para calcular as probabilidades no regime estacionário de cada um dos estados possíveis do *workflow* descrito no modelo em *PEPA*. A partir dessas probabilidades, a *pyPEPA* consegue fornecer o rendimento (*throughput*) das atividades do *workflow* e também a taxa de utilização de seus componentes.

Além do modelo em *PEPA* e sua solução, a `wkf2pepa` também gera uma representação gráfica do modelo de *workflow*, que permite que o usuário possa verificá-lo mais facilmente.

O funcionamento completo da ferramenta `wkf2pepa` pode ser descrito pelos seguintes passos:

1. Recebe como entrada uma descrição textual (modelo) de um *workflow* que segue uma sintaxe simples, criada com base na linguagem *DOT* [2];
2. Por meio de analisadores léxico e sintático gerados com a biblioteca *PLY* (*Python Lex-Yacc*) [4], lê o modelo de workflow e gera uma representação em memória dele. Essa representação, baseada em grafos, é feita através de classes criadas na ferramenta para a manipulação de nós, arestas e *workflows*.
3. Gera uma descrição do *workflow* de entrada em linguagem *DOT* e sua representação gráfica (visualização) através da biblioteca *Graphviz* [3].
4. Gera um modelo analítico (estocástico) do *workflow* na linguagem *PEPA*.
5. Gera a solução numérica do modelo analítico e extrai os índices de desempenho com a *pyPEPA* .

Resultados

O código fonte da `wkf2pepa`, suas dependências, informações sobre o seu uso, exemplos de *workflows* de entrada (e seus respectivos modelos em *PEPA*) podem ser vistos na página do projeto [1].

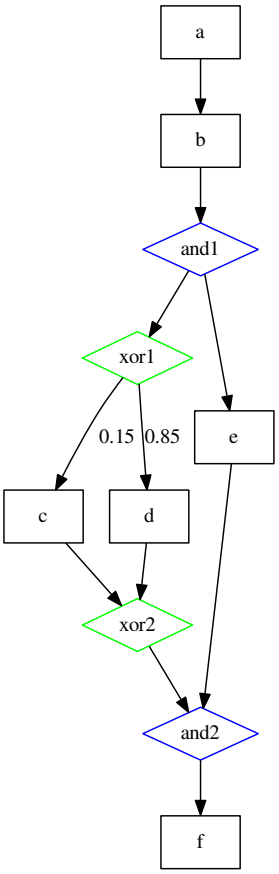
Ao processar um *workflow*, a ferramenta cria arquivos de saída contendo a descrição do *workflow* em *DOT*, sua representação gráfica, sua descrição em *PEPA* e os índices de desempenho dela extraídos. Um exemplo de descrição de *workflow* que pode ser usada como entrada para a ferramenta e parte dos resultados de saída gerados para ela são mostrados na Figura 1. Outros exemplos de *workflows* com estruturas mais complexas e seus respectivos resultados encontram-se na página do projeto [1].

```
digraph {
    a           -> b;
    b           -> and1;
    and1 [and]  -> e, xor1;
    xor1 [xor]  -> [0.15] c, [0.85] d;
    e [0.5]     -> and2;
    c           -> xor2;
    d           -> xor2;
    xor2        -> and2;
    and2        -> f;
}
```

(a) Descrição do workflow de entrada.

```
r_a = 1.0; r_b = 1.0; r_e = 0.5;
r_c = 1.0; r_d = 1.0; r_f = 1.0;

r_AND = 100.0; r_XOR = 100.0; r_OR  = 100.0;

prob_xor1_c = 0.15; prob_xor1_d = 0.85;

r_xor1_c = prob_xor1_c * r_XOR;
r_xor1_d = prob_xor1_d * r_XOR;

P = (a, r_a) . (b, r_b) . (and1, r_AND) . (and2, r_AND) . (f, r_f) . P;

P_and1_e = (and1, r_AND) . (e, r_e) . (and2, r_AND) . P_and1_e;
P_and1_xor1 = (and1, r_AND) . P_xor1;
P_xor1_c = (c, r_c) . P_xor2;
P_xor1_d = (d, r_d) . P_xor2;
P_xor1 = (xor1, r_xor1_c) . P_xor1_c + (xor1, r_xor1_d) . P_xor1_d;
P_xor2 = (xor2, r_XOR) . (and2, r_AND) . P_and1_xor1;

P <and1, and2> (P_and1_e <and1, and2> P_and1_xor1)
```

(b) Modelo em PEPA gerado.



(c) Visualização criada a partir da saída em DOT

Figura 1: Exemplo de uma execução da `wkf2pepa`.

Conclusões

Este trabalho apresenta a ferramenta de software `wkf2pepa`, que converte de forma automática modelos de *workflow* em modelos estocásticos e, a partir destes últimos, extrai predições do desempenho dos *workflows*. A predição do desempenho de *workflows* é importante porque auxilia a identificação de problemas em sua modelagem e o provisionamento dos recursos necessários para a execução eficiente desses sistemas.

Os modelos de *workflow* usados como entrada para a `wkf2pepa` são definidos por meio de uma notação textual simples e intuitiva, que permite descrever as estruturas de fluxos de atividades mais comumente encontradas nos experimentos científicos. Assim, a `wkf2pepa` possibilita que usuários obtenham predições sobre o desempenho de *workflows* de forma prática, sem a necessidade de conhecer detalhes sobre modelagem estocástica e análise numérica.

A `wkf2pepa` gera modelos estocásticos na álgebra de processos *PEPA*. A ferramenta usa a biblioteca *pyPEPA* para obter a solução numérica dos modelos estocásticos e índices de desempenho tais como a taxa de utilização dos componentes do modelo e o rendimento de cada atividade e do *workflow* como um todo.

Como trabalhos futuros, pretende-se estender a `wkf2pepa` para que ela lide com modelos de *workflows* que incluam uma descrição dos recursos disponíveis para a execução. Com isso, os modelos estocásticos gerados e os índices de desempenho extraídos a partir deles fornecerão uma boa aproximação do desempenho real esperado para os *workflows*.

Referências

[1] *Código fonte da ferramenta de software desenvolvida, exemplos de workflow de entrada e seus respectivos modelos em PEPA*. `www.ime.usp.br/~kellyrb/ic/#lucas`.

[2] *The DOT Language | Graphviz - Graph Visualization Software*. `http://www.graphviz.org/content/dot-language`.

[3] *Graphviz | Graphviz - Graph Visualization Software*. `http://www.graphviz.org/`.

[4] *PLY (Python Lex-Yacc)*. `http://www.dabeaz.com/ply/`.

[5] *pypepa - Python toolset for PEPA*. `https://github.com/tdi/pyPEPA`.

[6] Gadelha, L. M. R.: *Gerência de Proveniência em Workflows Científicos Paralelos e Distribuídos*. Tese de Doutoramento, Universidade Federal do Rio de Janeiro, 2012.

[7] Ogasawara, E. S.: *Uma Abordagem Algébrica para Workflows Científicos com Dados em Larga Escala*. Tese de Doutoramento, Universidade Federal do Rio de Janeiro, 2011.