

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

Lucas Magno¹
Kelly Rosa Braghetto²

¹Instituto de Física
²Instituto de Matemática e Estatística

Universidade de São Paulo

PIBIC/CNPq

INTRODUÇÃO

- ▶ Workflows científicos

INTRODUÇÃO

- ▶ Workflows científicos
- ▶ Custo de execução

INTRODUÇÃO

- ▶ Workflows científicos
- ▶ Custo de execução
- ▶ Previsão de desempenho

INTRODUÇÃO

- ▶ Workflows científicos
- ▶ Custo de execução
- ▶ Previsão de desempenho
- ▶ Modelagem analítica
 - ▶ Redes de Petri
 - ▶ Álgebras de processos

DIFICULDADES

- ▶ Linguagens e modelos estocásticos

DIFICULDADES

- ▶ Linguagens e modelos estocásticos
- ▶ Programas de simulação e análise numérica

DIFICULDADES

- ▶ Linguagens e modelos estocásticos
- ▶ Programas de simulação e análise numérica
- ▶ Diversas áreas da ciência

OBJETIVOS

- Ferramenta de software

OBJETIVOS

- ▶ Ferramenta de software
 - ▶ Descrição simples do workflow

OBJETIVOS

- ▶ Ferramenta de software
 - ▶ Descrição simples do workflow
 - ▶ Geração do modelo analítico

OBJETIVOS

- ▶ Ferramenta de software
 - ▶ Descrição simples do workflow
 - ▶ Geração do modelo analítico
 - ▶ Extração dos índices de desempenho

O PROGRAMA

► *wkf2pepa*

O PROGRAMA

- ▶ *wkf2pepa*
- ▶ *Python*

O PROGRAMA

- ▶ *wkf2pepa*
- ▶ *Python*
 - ▶ Alto nível
 - ▶ Bibliotecas

DESCRIÇÃO DO WORKFLOW

- ▶ Linguagem textual simples e intuitiva

DESCRIÇÃO DO WORKFLOW

- ▶ Linguagem textual simples e intuitiva
- ▶ Baseada na linguagem *DOT*

DESCRIÇÃO DO WORKFLOW

- ▶ Linguagem textual simples e intuitiva
- ▶ Baseada na linguagem *DOT*
- ▶ Grafos direcionados

DESCRIÇÃO DO WORKFLOW

EXEMPLO

```
1  digraph {
2      a      → b;
3      b      → and1;
4      and1 [and] → e, xor1;
5      xor1 [xor] → [0.15] c, [0.85] d;
6      e      [0.5] → and2;
7      c      → xor2;
8      d      → xor2;
9      xor2    → and2;
10     and2    → f;
11 }
```

LEITURA DO WORKFLOW DE ENTRADA

- Analisadores léxico (*lexer*) e sintático (*parser*)

LEITURA DO WORKFLOW DE ENTRADA

- ▶ Analisadores léxico (*lexer*) e sintático (*parser*)
- ▶ PLY - *Python Lex-Yacc*

ESTRUTURA DE DADOS NA MEMÓRIA

- ▶ Grafo
 - ▶ Generalidade
 - ▶ Independência de linguagem

ESTRUTURA DE DADOS NA MEMÓRIA

- ▶ Grafo
 - ▶ Generalidade
 - ▶ Independência de linguagem
- ▶ Classes
 - ▶ Flexibilidade
 - ▶ Clareza

ESTRUTURA DE DADOS NA MEMÓRIA

- ▶ Grafo
 - ▶ Generalidade
 - ▶ Independência de linguagem
- ▶ Classes
 - ▶ Flexibilidade
 - ▶ Clareza
 - ▶ *Node, Edge, Workflow*

VISUALIZAÇÃO DO WORKFLOW

- ▶ Verificação da estrutura em memória

VISUALIZAÇÃO DO WORKFLOW

- ▶ Verificação da estrutura em memória
- ▶ Linguagem *DOT*

VISUALIZAÇÃO DO WORKFLOW

- ▶ Verificação da estrutura em memória
- ▶ Linguagem *DOT*
- ▶ *PDF*

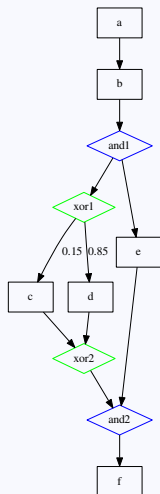
VISUALIZAÇÃO DO WORKFLOW

EXEMPLO

```

1  digraph workflow1 {
2      a      [shape=box, label=a];
3      xor1   [shape=diamond, label=xor1, color=green];
4      c      [shape=box, label=c];
5      b      [shape=box, label=b];
6      e      [shape=box, label=e];
7      d      [shape=box, label=d];
8      f      [shape=box, label=f];
9      xor2   [shape=diamond, label=xor2, color=green];
10     and1   [shape=diamond, label=and1, color=blue];
11     and2   [shape=diamond, label=and2, color=blue];
12     and1  -> e;
13     and2  -> f;
14     c     -> xor2;
15     e     -> and2;
16     xor2  -> and2;
17     xor1  -> d [label="0.85"];
18     and1  -> xor1;
19     b     -> and1;
20     a     -> b;
21     xor1  -> c [label="0.15"];
22     d     -> xor2;
23 }

```



MODELAGEM ANALÍTICA

- ▶ Álgebra de processos estocástica

MODELAGEM ANALÍTICA

- ▶ Álgebra de processos estocástica
 - ▶ Composicionalidade
 - ▶ Formalismo
 - ▶ Abstração

MODELAGEM ANALÍTICA

- ▶ Álgebra de processos estocástica
 - ▶ Composicionalidade
 - ▶ Formalismo
 - ▶ Abstração
- ▶ *PEPA - Performance Evaluation Process Algebra*

MODELAGEM ANALÍTICA

- ▶ Álgebra de processos estocástica
 - ▶ Composicionalidade
 - ▶ Formalismo
 - ▶ Abstração
- ▶ *PEPA - Performance Evaluation Process Algebra*
 - ▶ Bem desenvolvida
 - ▶ Ferramentas de apoio
 - ▶ Cadeia de Markov

MODELAGEM ANALÍTICA

EXEMPLO

```

1  r_a = 1.0; r_b = 1.0; r_e = 0.5;
2  r_c = 1.0; r_d = 1.0; r_f = 1.0;
3
4  r_AND = 100.0; r_XOR = 100.0; r_OR = 100.0;
5
6  prob_xor1_c = 0.15;
7  prob_xor1_d = 0.85;
8
9  r_xor1_c = prob_xor1_c * r_XOR;
10 r_xor1_d = prob_xor1_d * r_XOR;
11
12 P = (a, r_a) . (b, r_b) . (and1, r_AND) . (and2, r_AND) . (f, r_f) . P;
13
14 P_and1_e = (and1, r_AND) . (e, r_e) . (and2, r_AND) . P_and1_e;
15 P_and1_xor1 = (and1, r_AND) . P_xor1;
16 P_xor1_c = (c, r_c) . P_xor2;
17 P_xor1_d = (d, r_d) . P_xor2;
18 P_xor1 = (xor1, r_xor1_c) . P_xor1_c + (xor1, r_xor1_d) . P_xor1_d;
19 P_xor2 = (xor2, r_XOR) . (and2, r_AND) . P_and1_xor1;
20
21 P <and1, and2> (P_and1_e <and1, and2> P_and1_xor1)

```

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO

► *pyPEPA*

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO

- ▶ *pyPEPA*
- ▶ Probabilidades dos estados no regime estacionário

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO

- ▶ *pyPEPA*
- ▶ Probabilidades dos estados no regime estacionário
- ▶ Rendimentos (*throughput*)

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO

- ▶ *pyPEPA*
- ▶ Probabilidades dos estados no regime estacionário
- ▶ Rendimentos (*throughput*)
- ▶ Taxas de utilização

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO

EXEMPLO

```
1  Throughput:
2  xor1      -2.62364490393e-16
3  c         9.38551843388e-18
4  xor2      7.5563162709e-20
5  e         0.49504950495
6  and1      0.49504950495
7  and2      0.49504950495
8
9  Utilization:
10 P          1.0
11
12 P_and1_e    0.0049504950495
13 (and2       0.0049504950495
14 (e          0.990099009901
15
16 P_xor1_d    0.0049504950495
17 P_xor1_c    3.43509599203e-20
18 P_xor1      -6.60131287319e-19
19 r_AND).P_and1_e 0.0049504950495
20 P_xor2      3.49163606874e-22
21 (and2       4.07395329479e-22
22 r_e ).((and2 0.990099009901
23 P_and1_xor1 -1.31596606031e-18
```

VISUALIZAÇÃO DO WORKFLOW

MODELAGEM ANALÍTICA

EXTRAÇÃO DOS ÍNDICES DE DESEMPENHO