

Título a Definir

Bruno César Ribas

UTFPR-PB

30 de agosto de 2016

RGVzYWZpbwo=

RGVzYWZpbwo=

```
1 ribas@jaguaririca:~$ echo 'RGVzYWZpbwo=' | base64 -d
2 Desafio
3 ribas@jaguaririca:~$
```

```
1 $ sed -i -e 's/T.tulo a Definir/Desafio/g' \  
2 apresentacao.tex  
3 ribas@jaguaririca:~$ pdflatex apresentacao
```


Desafio

Bruno César Ribas

UTFPR-PB

30 de agosto de 2016

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 QWNobyBxdWUgZXNjb25kZXJpYSBhbGdvIHN1Y3JldG8gZGUgZm9ybW  
2 EgbWFpcyBhZGVxdWFkYSwgbs0jbyBhY2hhPwo=
```

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 $ base64 -d ladapagina
2 Acho que esconderia algo secreto de forma mais
3 adequada, n.o acha?
4 $
```

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 TXVpdG8gYmVtIGNvbWJhdGVudGUhIEVuY29udHJvdSBhbGd1bWEgY
2 29pc2Egc2VjcmVOYSwgbWFzIHRhbHZleiBhcyBjb2lzYXMKbs0jby
3 BzZWphbSB0w6NvIHNpbXBsZXMgaGVpbi4KCklGVFc2NFRCRUJaV1M
4 zSkFNvlpYSVlMTk41WlNBWTNQTlZTNEhKMOJOWlNHNk1EQkVCUl c2
5 M1RXTVZaSEdZTFNFQ1pXN1l1UU01VUUcKRzMzTO9SUzRIT1RFTjRRS
6 EdaTERPS1NYSTNaTOJJRkUyWUxURUJVVzRaVEZOU1VYVTNMRk5aMk
7 dLTEJBT0JRWEVZSkFPWlhXSFE1SwpGUVFHNVE1RE4OUU1IS0pBTUZ
8 ZWEsySkFPRjJXS01EVO1GV1c2NFpBT1JTWEVJRFBFQ1RHUzNUQk5R
9 UudJM1pBTVJTWEdZTEdORlhTCjRDUUUtJRlJXUTNaQU9CWkhLWkRGT
0 loyR0tJRFdONVIOSEtSQU10WFc0NVRGTOpaVOMOUkFNTlhXMk1EQk
1 9GM1dLMORCRUJZWEtaSkEKT1IyV0kzWkFPMOIyVU1ERkVCMkhLWkR
2 QRUJUWEtZTFNNU1FTQTREqk9KUVNBNEExWTVVGSE0zMORZT1ZDQV1M
3 RE5CU1NBWkRGTOJYVwpTNFpBTVJTUOE1REZPSVFHSzQzUk9WU1dHM
4 kxFTjRYQOFRSkFPQ1NYRVozVk5aMkdDUFpBS1pYV0hRNUtFQ1NHSz
5 VURk9MQjJDSURFCk1WW1dHMzNDTOpVWEVDVFRONTVHUzNUSU40WEF
6 VPT09Cg==
```

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 $ base64 -d ladapagina2
2 Muito bem combatente! Encontrou alguma coisa secreta,
3 mas talvez as coisas n.o sejam t.o simples hein.
4
5 IFTW64TBEBZWS3JAMVZXIYLNN5ZSAY3PNVS4HJ3BNZSG6IDBEBRW6
6 3TWMVZHGYLSEBZW6YTSMUQGG330ORS4HOTEN4QHGZLDOJSXI3ZOBI
7 FE2YLTEBUW4ZTFNRUXU3LFNZ2GKLBAOBQXEYJAOZXWHQ5KFQQG5Q5
8 DN4QMHKJAMFYXK2JAOF2WKIDWMFWW64ZAORSXEIDPEBTGS3TBNQQG
9 I3ZAMRSXGYLGNFXS4CQKIFRWQ3ZA0BZHKZDFNZ2GKIDWN5R4HKRAM
0 NXW45TF0JZWC4RAMNXW2IDBOF2WK3DBEBYXKZJAOR2WI3ZA03B2UI
1 DFEB2HKZDPEBTXKYLMSRQSA4DB0JQSA4LVMUFHM33DYOVCAYLDNBS
2 SAZDFOBXWS4ZAMR5SA5DFOIQGK43ROVSWG2LEN4XCAQJA0BSXEZ3V
3 NZ2GCPZAKZXWHQ5KEBSGK5TFOLB2CIDEMVZWG33COJUXECTTN55GS
4 3TIN4XAU===
5 $
```

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 $ base64 -d ladapagina3
2 base64: entrada inv.lida
3 $
```

Desafio

Tudo começou na página de Algoritmos 1 e Segurança Computacional

```
1 $ base32 -d ladapagina3
2 Agora sim estamos começando a conversar sobre
3 conte.do secreto.
4
5 Mas infelizmente, para voc., não é aqui que vamos
6 ter o final do desafio.
7
8 Acho prudente você conversar com aquela que tudo
9 vê e tudo guarda para que você ache depois de
0 ter esquecido. A pergunta? Você deverá descobrir
1 sozinho.
2
3 $
```

Desafio - A Espiã



Desafio - A Espiã



Desafio - A Espiã

```
1 Vejo que voce^ encontrou a pessoa que tudo ve^ 8^)  
2 Qr code nao guarda muitos segredos, uma trivã  
3 basica ha de guardar.  
4  
5 RXN0YW1vcyBhcXVpIG5vdmFtZW50ZSA4XikuCgpUZW50byBzZXIgy  
6 nJldmUhIE9SSU90LCBwb3NzdWkgdW0gYXJxdWl2byBzZW5yZXIvIG  
7 5vIERFViBTSE0uClZvY80qIGrDoSBvdXZpdSBmYWxhciBkbyBjb2I  
8 hbmRvIGhvc3Q/Cg==
```

Desafio - A Espiã

```
1 $ base64 -d qr-code-nice
2 Estamos aqui novamente 8^).
3
4 Tento ser breve! ORION, possui um arquivo secreto no
5 DEV SHM.
6 Voce ja ouviu falar do comando host?
7 $
```

Desafio - ORION

```
1 ribas@jaguatirica:~$ ssh brunoribas@orion -YC
2 brunoribas@orion's password:
3 brunoribas@orion:~$
```

Desafio - ORION

```
1 brunoribas@orion:~$ ls -a /dev/shm
2  ..  . arquivo-meio-secreto  .arquivo-super-secreto
3 nao-muito-secreto           onde-esta.interrogacao
```

Desafio - ORION

REAL CODE NOW

Desafio - ORION

INICIO/DSC/Y-P/.secret

printf Nice | md5sum

Desafio - ORION

Desafio - ORION



Desafio - ORION

```
1 $ sed -i -e 's/Desafio/SAT, PBO e Virtuais/g' \  
2 apresentacao.tex  
3 ribas@jaguatirica:~$ pdflatex apresentacao
```

Satisfatibilidade e Otimização pseudo-Booleana na Consolidação de Máquinas virtuais

Bruno César Ribas

UTFPR-PB

30 de agosto de 2016

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Visão Geral

- Sistemas computacionais usados em aplicações críticas;
 - ▶ ex: automóveis, aviões, trens
- garantir o funcionamento é crucial;
- a complexidade de garantia de funcionamento aumenta com a complexidade do sistema;
- verificação automática necessária;
- descrições formais de circuitos podem ser convertidas para instâncias SAT e verificadas por um resolvidor SAT;

Definição Satisfatibilidade

É o problema de decidir se uma fórmula é satisfatível, ou seja, verificar se há uma valoração para as variáveis da fórmula que a torne verdadeira.

Visão Geral - Definições

Literal

é uma variável x ou a sua negação $\neg x$

Literal Puro

é o literal que aparece em apenas uma forma em toda a fórmula

Visão Geral - Definições

Cláusula

é a disjunção de literais

- ex: $(x_1 \vee x_2)$

Cláusula Unitária

é uma cláusula representada por apenas um literal

- ex: x_1

CNF - Forma Normal Conjuntiva

É a conjunção de cláusulas

- ex: $(p \vee q) \wedge (q \vee \neg p) \wedge (\neg p \vee \neg q)$

Satisfatibilidade - 4 pombos

p	cnf	12	22			
1	2	3	0	-2	-8	0
4	5	6	0	-2	-11	0
7	8	9	0	-5	-8	0
10	11	12	0	-5	-11	0
-1	-4	0		-8	-11	0
-1	-7	0		-3	-6	0
-1	-10	0		-3	-9	0
-4	-7	0		-3	-12	0
-4	-10	0		-6	-9	0
-7	-10	0		-6	-12	0
-2	-5	0		-9	-12	0

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Satisfatibilidade

- Tabela verdade
- Primeiro Algoritmo em 1960, DP
- DPLL refinamento em 1962
- Algoritmos completos baseados no DPLL
- SATO(1996), GRASP(1996), ZCHAFF(2001), MINISAT(2003)
 - ▶ Otimizações do DPLL

DPLL

Tautologia

Remove Cláusulas Tautológicas

Cláusula Unitária

Remove Cláusulas Unitárias

Literal Puro

Remove Literais Puros

Ramificação

Escolhe uma variável e assinala Verdadeiro ou Falso e abre a fórmula em duas: um lado com x verdadeiro; outro com x falso

DPLL

Decisão

É o análogo da Ramificação, onde alguma variável é escolhida e valorada

Nível

Cada decisão corresponde a um nível de decisão

BCP (Boolean Constraint Propagation)

É o algoritmo que propaga o efeito da valoração de uma variável.
Corresponde às regras: Cláusula Unitária, Literal Puro e Tautologia

Retrocesso

Sempre que o BCP encontrar alguma cláusula falsa o retrocesso é feito

DPLL

```
1 enquanto  $1 == 1$  faça
2     se Decide() == OK então
3         BCP()
4         se BCPRetornouConflito() == SIM então
5             se Diagnostico() == NaoPodeSerResolvido então
6                 retorna INSTATISFATIVEL
7             fim
8             senão
9                 RETROCESSO()
10            fim
11        fim
12    fim
13    senão
14        retorna SAT
15    fim
16 fim
```


DPLL - Pontos Críticos

- Escolha de variável
- Propagação da valoração
- Retrocesso

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Heurísticas para ramificação

- Consome até 10% do tempo de processamento
- Escolha aleatória (RAND)
- MOM(1995), BOHM(1992)
- VSIDS(2001) (*Variable State Independent Decaying Sum*)
 - ▶ Maior eficiência com informação dinâmica
 - ▶ Melhorou desempenho em uma ordem de grandeza;

Heurística para ramificação VSIDS

- 1 Todo literal tem um contador iniciado como 0;
- 2 Quando uma cláusula nova é adicionada, o contador associado com cada literal presente na cláusula é incrementado;
- 3 A variável (não valorada) e seu literal com o contador mais alto é escolhida para a decisão;
- 4 Periodicamente todos os contadores são divididos por uma constante.

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

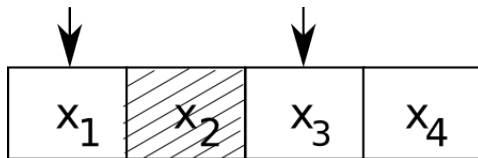
5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

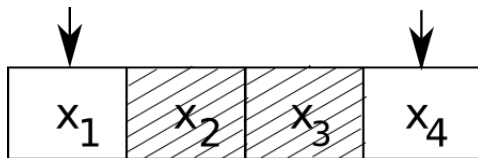
BCP rápido

- Consome 80% do tempo de processamento
- Lista de observação
- Estado crítico de uma cláusula
- Inferência



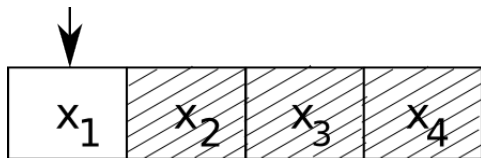
BCP rápido

- Consome 80% do tempo de processamento
- Lista de observação
- Estado crítico de uma cláusula
- Inferência



BCP rápido

- Consome 80% do tempo de processamento
- Lista de observação
- Estado crítico de uma cláusula
- Inferência



Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Retrocesso não-cronológico

- Retrocesso para um nível de decisão mais distante
- Aprender valorações parciais conflitantes, evita repetir erros
- Um grafo é mantido com as decisões e implicações
- Atualizado sempre a cada Decisão e execução do BCP

Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$

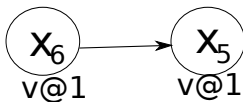
Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$

x_6
v@1

Retrocesso não-cronológico

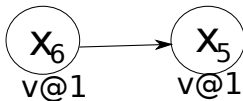
$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$



Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$

f@2



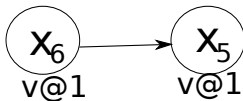
Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$

f@2

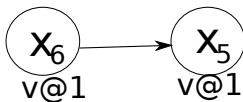
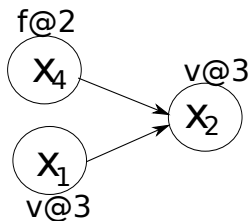


v@3



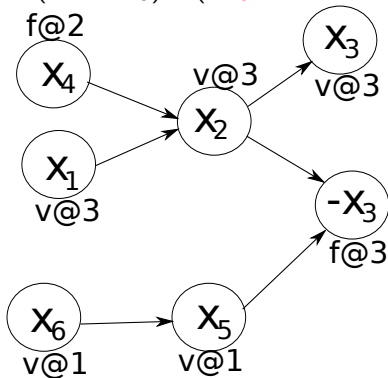
Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$



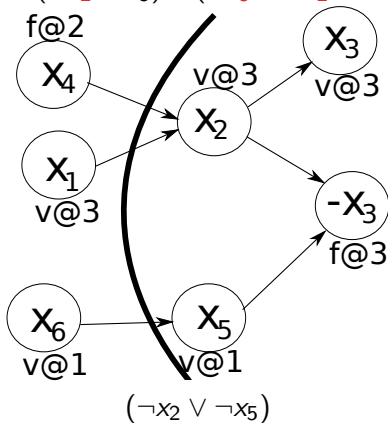
Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$



Retrocesso não-cronológico

$$(\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_5 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_6 \vee x_5)$$



Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Pseudo-Boolean

- Em uma definição mais direta, uma função pseudo-Boolean é uma função que mapeia valores Boolean para um número inteiro;
- As funções não são Boolean mas ainda permanecem muito próximas de funções estritamente Boolean;
- Nesse domínio as variáveis possuem domínio Boolean e as restrições são desigualdades com coeficientes inteiros;
- As restrições oferecem uma maneira mais expressiva e natural de representação que cláusulas e ainda, é um formalismo que fica muito próximo de SAT.
- Em geral restrições são do tipo: $\sum a_i x_i \geq k$

Pseudo-Boolean - 4 pombos

```
* #variable= 12 #constraint= 7
+1 x1 +1 x2 +1 x3 >= 1;
+1 x4 +1 x5 +1 x6 >= 1;
+1 x7 +1 x8 +1 x9 >= 1;
+1 x10 +1 x11 +1 x12 >= 1;
+1 ~x1 +1 ~x4 +1 ~x7 +1 ~x10 >= 3;
+1 ~x2 +1 ~x5 +1 ~x8 +1 ~x11 >= 3;
+1 ~x3 +1 ~x6 +1 ~x9 +1 ~x12 >= 3;
```

Pseudo-Boolean

Data: Fórmula pseudo-Boolean C

Result: SAT ou UNSAT para a fórmula C

```
1 enquanto verdade faça
2   | se Solucao_Encontrada( $C$ ) então
3   |   | retorna SAT
4   | senão
5   |   | Decida( $p$ )
6   | enquanto  $BCP(C) == CONFLITO$  faça
7   |   | se  $Diagnostico(C) == CONFLITO$  então
8   |   |   | retorna UNSAT
```

Pseudo-Boolean Propagação

Seja c um conjunto de literais e P uma valoração parcial. Definimos os conjuntos:

$$S_p(c) = \{x_i \mid x_i \in c \text{ e } x_i \in P\}$$

$$V_p(c) = \{x_i \mid x_i \in c \text{ e } \neg x_i \notin P\}$$

- $S_p(c)$ representa os literais que estão satisfeitos
- $V_p(c)$ representa os literais que são, ou satisfeitos ou não valorados

Pseudo-Boolean Propagação

Seja c uma restrição pseudo-Boolean na forma $\sum a_i x_i \geq k$, T um subconjunto de literais em c , e P uma valoração parcial dos valores das variáveis. As contagens *current* e *possible* são definidas como:

$$current(T, P) = \sum_{i|x_e \in S_p(T)} a_i - k$$

$$possible(T, P) = \sum_{i|x_e \in V_p(T)} a_i - k$$

- Se $current(c, P) \geq 0$ então está satisfeita
- Se $possible(c, P) \geq 0$ então é possível satisfazer, senão não é mais possível

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

Consolidação de Máquinas Virtuais

- Maximizar o uso dos recursos de hardware;
- A consolidação pode aumentar o uso dos servidores de 50% a 85%, quando o consumo de energia é mais eficiente;
- Relocação permite o desligamento de hardware, reduzindo o custo de energia e refrigeração.

Consolidação de Máquinas Virtuais

Definição do problema

Colocar K VMs $\{vm_1 \dots vm_K\}$ em execução dentro de N hardware $\{hw_1 \dots hw_N\}$ enquanto minimiza o número total de hardware ligados. Cada VM vm_i possui associada as suas necessidades com a quantidade de núcleos de processamento (VCPU) e a quantidade de memória RAM (VRAM), enquanto que cada hardware hw_j possui a quantidade de recursos disponível, em processamento (CPU) e memória (RAM).

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

- Cada hardware possui duas variáveis:

hw_i^{ram} relaciona a quantidade de RAM para cada hw_i

hw_i^{proc} relaciona a quantidade de CPU para cada hw_i

- Para cada hardware, a máquina virtual possui duas variáveis:

$vm_j^{ram \cdot hw_i}$ relaciona para a VM vm_j a quantidade de VRAM vm_j^{ram} necessária do hardware hw_i

$vm_j^{proc \cdot hw_i}$ em VCPU

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

- Minimizar a quantidade de hardware ligados

$$\text{minimize} : \sum_{i=1}^N hw_i \quad (1)$$

- Para garantir que a quantidade necessária de hardware está ativa mais duas restrições são adicionadas que implicam que a soma de RAM e CPU ativas deve ser maior ou igual que a soma dos recursos necessários pela VM

$$\sum_{i=1}^N RAM_{hw_i} \cdot hw_i^{ram} \geq \sum_{j=1}^K RAM_{vm_j} \cdot vm_j^{ram} \quad (2)$$

$$\sum_{i=1}^N PROC_{hw_i} \cdot hw_i^{proc} \geq \sum_{j=1}^K PROC_{vm_j} \cdot vm_j^{proc} \quad (3)$$

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

Para definir os limites de cada hardware mais duas restrições são colocadas:

RAM disponível por hardware: Esta restrição dita que a quantidade de RAM necessária para as máquinas virtuais não pode exceder a quantidade de RAM disponível no hardware. A restrição é definida em 4.

CPU disponível por hardware: Esta restrição dita que a quantidade de CPU virtuais necessárias não pode extrapolar a quantidade de CPU no hardware como definida em 5.

$$\forall hw_i^{ram} \in hw_N^{ram} \left(\sum_{j=1}^K RAM_{vm_j} \cdot vm_j^{ram \cdot hw_i} \leq RAM_{hw_i} \right) \quad (4)$$

$$\forall hw_i^{proc} \in hw_N^{proc} \left(\sum_{j=1}^K PROC_{vm_j} \cdot vm_j^{proc \cdot hw_i} \leq PROC_{hw_i} \right) \quad (5)$$

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

- Finalmente mais uma restrição por VM é adicionada para garantir que ela esteja rodando em exatamente um hardware

$$\forall vm_i \in vm_K \left(\sum_{j=1}^N vm_i^{proc \cdot hw_j} \cdot vm_i^{ram \cdot hw_j} \cdot hw_j^{proc} \cdot hw_j^{ram} = 1 \right) \quad (6)$$

- Nesse modelo temos $(2 \times N + 2 \times N \times K)$ variáveis e;
- $(2 + 2 \times N + K)$ restrições

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

- Principais problemas:
 - ▶ Baseado na formulação do Bin Packing
 - ▶ Resolvedores demoram muito tempo nas fórmulas maiores
 - ▶ Equações de igualdade são difíceis de se resolver
 - ★ São, geralmente, substituídas por duas restrições, \leq e \geq
 - ▶ \leq nem sempre é implementada nos resolvedores

Sumário

- 1 **Introdução**
 - Visão Geral
- 2 **Satisfatibilidade**
 - Heurísticas para ramificação
 - BCP rápido
 - Retrocesso não-cronológico
- 3 **Pseudo-Boolean**
- 4 **Consolidação de Máquinas Virtuais**
- 5 **Fortalecimento de restrições Pseudo-Boolean**
- 6 **PBFVMC**
- 7 **Conclusão e Trabalhos Futuros**

Fortalecimento de restrições Pseudo-Boolean

FORTALECIMENTO

- É uma técnica de construção automática de inferências;
- O objetivo de construir inferências é a de inferir novas restrições de um conjunto de restrições clausais;
- O efeito é construir uma representação mais forte e concisa de um problema, partindo de uma representação mais fraca;
- O método de fortalecimento é uma forma de redução de coeficientes;
- Herdada da área de pesquisa operacional.

Fortalecimento de restrições Pseudo-Boolean

```
1 //Procedimento fortalecer
2 //Entrada: Fórmula pseudo-Boolean C
3 //Saída: Fórmula fortalecida
4 para cada literal  $l$  em  $C$  faça
5    $P \leftarrow l$ 
6    $P \leftarrow \text{Propagação}(C, P)$ 
7   para cada restrição  $c = \sum a_i x_i \geq k$  com  $\text{current}(c, P) > 0$  faça
8      $c \leftarrow \sum a_i x_i + \text{current}(c, P) \neg l \geq k + \text{current}(c, P)$ 
```

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + c \geq 1 \quad (8)$$

$$b + c \geq 1 \quad (9)$$

- Se $P = \{a\}$,
- Se $P = \{\neg a\}$, $\{b, c\}$ será propagado
- A restrição 9 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + c \geq 1 \quad (8)$$

$$\mathbf{b + c \geq 1} \quad (9)$$

- Se $P = \{a\}$,
- Se $P = \{\neg a\}$, $\{b, c\}$ será propagado
- A restrição 9 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + c \geq 1 \quad (8)$$

$$a + b + c \geq 2 \quad (9)$$

- Se $P = \{b\}$,
- Se $P = \{\neg b\}$, $\{a, c\}$ será propagado
- A restrição 8 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + c \geq 1 \quad (8)$$

$$a + b + c \geq 2 \quad (9)$$

- Se $P = \{b\}$,
- Se $P = \{\neg b\}$, $\{a, c\}$ será propagado
- A restrição 8 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + b + c \geq 2 \quad (8)$$

$$a + b + c \geq 2 \quad (9)$$

- Se $P = \{c\}$,
- Se $P = \{\neg c\}$, $\{a, b\}$ será propagado
- A restrição 7 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b \geq 1 \quad (7)$$

$$a + b + c \geq 2 \quad (8)$$

$$a + b + c \geq 2 \quad (9)$$

- Se $P = \{c\}$,
- Se $P = \{\neg c\}$, $\{a, b\}$ será propagado
- A restrição 7 é super satisfeita e pode ser substituída pela restrição:

$$a + b + c \geq 2$$

Fortalecimento de restrições Pseudo-Boolean

$$a + b + c \geq 2 \quad (7)$$

$$a + b + c \geq 2 \quad (8)$$

$$a + b + c \geq 2 \quad (9)$$

- Duas restrições podem ser removidas da fórmula.

Fortalecimento de restrições Pseudo-Boolean

$$a + b + c \geq 2 \tag{7}$$

(8)

(9)

- Duas restrições podem ser removidas da fórmula.

Fortalecimento de restrições Pseudo-Boolean

Tabela: As colunas Restrições mostram a quantidade de restrições a fórmula original possuía e quantas restrições sobraram após o pré-processamento

		Original	Após Pré-processamento
Fórmula	Variáveis	Restrições	Restrições
pigeon-8	56	204	15
pigeon-9	72	297	17
pigeon-10	90	415	19
pigeon-11	110	561	21
pigeon-12	132	738	23
pigeon-20	380	3630	39
pigeon-30	870	12645	59
pigeon-40	1560	30460	79
pigeon-50	2450	60075	99
pigeon-60	3540	104490	119

Fortalecimento de restrições Pseudo-Boolean

Tabela: Tempo de resolução de várias instâncias do Princípio da Casa dos Pombos. A coluna de Pré-processamento é o tempo gasto de pré-processamento para cada instância. Os resolvedores SCIP-pb, Clasp e Sat4J-PB resolvem as fórmulas preprocessadas. O tempo limite foi de 600s e quando extrapolado um “-” é colocado.

Fórmula	Pré-processamento	SCIP	Clasp	Sat4J-PB
pigeon-8	0.00	0.01	0.056	0.203
pigeon-9	0.01	0.00	0.454	0.552
pigeon-10	0.01	0.01	4.939	3.207
pigeon-11	0.01	0.01	51.372	34.449
pigeon-12	0.02	0.01	570.511	-
pigeon-20	0.25	0.02	-	-
pigeon-30	1.78	0.03	-	-
pigeon-40	7.15	0.07	-	-
pigeon-50	21.25	0.10	-	-
pigeon-60	52.40	0.14	-	-

Sumário

1 Introdução

- Visão Geral

2 Satisfatibilidade

- Heurísticas para ramificação
- BCP rápido
- Retrocesso não-cronológico

3 Pseudo-Boolean

4 Consolidação de Máquinas Virtuais

5 Fortalecimento de restrições Pseudo-Boolean

6 PBFVMC

7 Conclusão e Trabalhos Futuros

- Baseado em uma formulação do princípio da Casa dos Pombos
- Retrabalhado para ser mais amigável com os resolvedores
- Algumas variáveis unificadas
 - ▶ hw_i^r e hw_i^p para hw_i
 - ▶ vm_j^r e vm_j^p para vm_j
- Todas restrições são PosiForm
 - ▶ Apenas \geq
 - ▶ Nenhum coeficiente negativo

- As variáveis no novo modelo, são:
 - N : número total de hardware (hw);
 - K : número total de máquinas virtuais (VM);
 - hw_i : Hardware $i \in N$;
 - $vm_j^{hw_i}$: Virtual Machine $j \in K$ executando em hw_i ;
 - R_{hw_i} e P_{hw_i} : RAM e CPU por hardware
 - R_{vm_j} e P_{vm_j} : RAM e CPU por virtual

- Minimizar a quantidade de hardware ligados:

$$\text{minimize : } \sum_{i=1}^N hw_i \quad (10)$$

- Soma da memória e poder de processamento tem que ser o suficiente para todas máquinas virtuais:

$$\sum_{i=1}^N R_{hw_i} \cdot hw_i \geq \sum_{j=1}^K R_{vm_j} \quad (11)$$

$$\sum_{i=1}^N P_{hw_i} \cdot hw_i \geq \sum_{j=1}^K P_{vm_j} \quad (12)$$

- Definir os limites totais de recursos que cada hardware pode prover em relação as máquinas virtuais que estão rodando no hardware:

$$\forall i \in 1..N \left(\sum_{j=1}^K (R_{vm_j} \cdot \neg vm_j^{hw_i}) + R_{hw_i} \cdot hw_i \geq \sum_{j=1}^K R_{vm_j} \right) \quad (13)$$

$$\forall i \in 1..N \left(\sum_{j=1}^K (P_{vm_j} \cdot \neg vm_j^{hw_i}) + P_{hw_i} \cdot hw_i \geq \sum_{j=1}^K P_{vm_j} \right) \quad (14)$$

PBFVMC - Restrições Pombais

- Cada máquina virtual deve rodar em algum hardware

$$\forall j \in 1..K \left(\sum_{i=1}^N vm_j^{hw_i} \geq 1 \right) \quad (15)$$

- A máquina virtual deve executar em apenas um único hardware

$$\forall j, i, k \in 1..K, 1..N, i+1..K (\neg vm_j^{hw_i} + \neg vm_k^{hw_i}) \geq 1 \quad (16)$$

- A formulação da casa dos pombos é facilmente modelada com cláusulas, tendo $(n + 1)$ cláusulas, onde n é a quantidade de casas, dizendo que cada pombo deve ser acomodado em alguma casa
- O problema é que o número de cláusulas cresce rapidamente conforme o número de pombos aumenta

PBFVMC - Restrições Pombais Fortalecidas

- Cada máquina virtual deve rodar em algum hardware

$$\forall j \in 1..K \left(\sum_{i=1}^N vm_j^{hw_i} \geq 1 \right) \quad (17)$$

- A máquina virtual deve executar em apenas um único hardware

$$\forall j \in 1..K \left(\sum_{i=1}^N \neg vm_j^{hw_i} \geq N - 1 \right) \quad (18)$$

- Pontos Positivos

- ▶ O total de variáveis é $(N + N \times K)$
- ▶ O total de restrições é $(2 + 2 \times N + 2 \times K)$
- ▶ Simples manipulação algébrica das restrições (4) e (5) para gerar (13) e (14)
- ▶ Todas restrições não-lineares foram removidas

On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

Para definir os limites de cada hardware mais duas restrições são colocadas:

RAM disponível por hardware: Esta restrição dita que a quantidade de RAM necessária para as máquinas virtuais não pode exceder a quantidade de RAM disponível no hardware. A restrição é definida em 4.

CPU disponível por hardware: Esta restrição dita que a quantidade de CPU virtuais necessárias não pode extrapolar a quantidade de CPU no hardware como definida em 5.

$$\forall hw_i^{ram} \in hw_N^{ram} \left(\sum_{j=1}^K RAM_{vm_j} \cdot vm_j^{ram \cdot hw_i} \leq RAM_{hw_i} \right) \quad (4)$$

$$\forall hw_i^{proc} \in hw_N^{proc} \left(\sum_{j=1}^K PROC_{vm_j} \cdot vm_j^{proc \cdot hw_i} \leq PROC_{hw_i} \right) \quad (5)$$

- Definir os limites totais de recursos que cada hardware pode prover em relação as máquinas virtuais que estão rodando no hardware:

$$\forall i \in 1..N \left(\sum_{j=1}^K (R_{vm_j} \cdot \neg vm_j^{hw_i}) + R_{hw_i} \cdot hw_i \geq \sum_{j=1}^K R_{vm_j} \right) \quad (13)$$

$$\forall i \in 1..N \left(\sum_{j=1}^K (P_{vm_j} \cdot \neg vm_j^{hw_i}) + P_{hw_i} \cdot hw_i \geq \sum_{j=1}^K P_{vm_j} \right) \quad (14)$$

Tamanho das Fórmulas

HW	VMS	Anterior		PBFVMC	
		Vars	Constr	Vars	Constr
hw32-vm25p	98	6336	164	3168	262
hw32-vm50p	173	11136	239	5568	412
hw32-vm75p	278	17856	344	8928	622
hw32-vm85p	320	20544	386	10272	706
hw32-vm90p	325	20864	391	10432	716
hw32-vm95p	348	22336	414	11168	762
hw32-vm98p	364	23360	430	11680	794
hw32-vm99p	366	23488	432	11744	798

Tamanho das Fórmulas

		Anterior		PBFVMC	
HW	VMS	Vars	Constr	Vars	Constr
hw512-vm25p	1432	1467392	2458	733696	3890
hw512-vm50p	2771	2838528	3797	1419264	6568
hw512-vm75p	4035	4132864	5061	2066432	9096
hw512-vm85p	4431	4538368	5457	2269184	9888
hw512-vm90p	4745	4859904	5771	2429952	10516
hw512-vm95p	5068	5190656	6094	2595328	11162
hw512-vm98p	5319	5447680	6345	2723840	11664
hw512-vm99p	5402	5532672	6428	2766336	11830

Experimentos - Decide SAT

Formula	Anterior	PBFVMC
hw32-vm25p	92.756	0.433
hw32-vm50p	35.643	0.542
hw32-vm75p	3.43	0.588
hw32-vm85p	4.516	0.911
hw32-vm90p	6.795	9.716
hw32-vm95p	3442.92	8.129
hw32-vm98p	TLE	45.589
hw32-vm99p	TLE	5566.28
hw512-vm25p	TLE	28.436
hw512-vm50p	TLE	162.289
hw512-vm75p	TLE	508.322
hw512-vm85p	TLE	287.437
hw512-vm90p	TLE	5604.022
hw512-vm95p	TLE	4222.892

Experimentos - Optimize

Formula	Anterior	PBFVMC
hw32-vm25p	249.897/7	191.912/7
hw32-vm50p	35.696/16	4.134/16
hw32-vm75p	23.628/25	772.657/24
hw32-vm85p	1175.103/29	159.86/28
hw32-vm90p	108.361/31	948.924/29
hw32-vm95p	3442.92/32	319.041/31
hw32-vm98p	TLE	45.651/32
hw32-vm99p	TLE	5566.491/32
hw512-vm25p	TLE	4471.005/140
hw512-vm50p	TLE	5406.047/281
hw512-vm75p	TLE	4378.66/408
hw512-vm85p	TLE	4919.328/461
hw512-vm90p	TLE	14426.6/487
hw512-vm95p	TLE	6864.151/510

Sumário

- 1 **Introdução**
 - Visão Geral
- 2 **Satisfatibilidade**
 - Heurísticas para ramificação
 - BCP rápido
 - Retrocesso não-cronológico
- 3 **Pseudo-Boolean**
- 4 **Consolidação de Máquinas Virtuais**
- 5 **Fortalecimento de restrições Pseudo-Boolean**
- 6 **PBFVMC**
- 7 **Conclusão e Trabalhos Futuros**

Proposta de um modelo para Consolidação de Máquinas Virtuais

- O modelo apresentado ainda não está completo;
- Nesta proposta a abordagem é criar um modelo para a consolidação de máquinas virtuais levando em consideração a relocação (migração) das máquinas entre diversas servidoras;
- Além disso, focar na formulação para ser amigável aos resolvedores, afim de aproveitar o maquinário já existente.

Satisfatibilidade e Otimização pseudo-Booleana na Consolidação de Máquinas virtuais

Bruno César Ribas

UTFPR-PB

30 de agosto de 2016

Calma! tem mais

```
1 $ sed -i -e 's/SAT, PBO e Virtuais/Maratona/g' \  
2 apresentacao.tex  
3 ribas@jaguaririca:~$ pdflatex apresentacao
```

Maratona

Bruno César Ribas

UTFPR-PB

30 de agosto de 2016

Maratona de Programação



Maratona de Programação



Maratona de Programação



Maratona de Programação



Maratona de Programação

