

Geração de traços para o SPEC CPU 2006

Esse manual visa facilitar a criação de traços do SPEC CPU 2006 com o Sinuca PinPoints para o OrCS.

Consideraremos que o Sinuca Tracer já foi instalado de acordo com o manual **Readme OrCS tracer** e que o Sinuca PinPoints já foi configurado de acordo com o manual **Gerando traços com o Sinuca PinPoints**. Além disso, o Spec Cpu 2006 deve ter sido instalado e compilado corretamente, de forma a estar funcional para comandos como:

```
>> runspec --size=test -noreportable -tune=base -iterations=1 all
```

Configurando

Visto que a criação de traços para esse conjunto de benchmarks é uma tarefa corriqueira, a pasta **trace_generator/scripts** já contém um script preparado para sua criação. Porém, para seu uso, alguns parâmetros importantes devem ser considerados. Primeiramente, o arquivo **pinplay.cfg** contido na pasta **trace_generator/scripts/apps_benchmarks**, que é o arquivo contendo as configurações básicas dos traços a serem gerados deve ser editado.

Nosso único parâmetro de interesse é o **slice_size**, que define o tamanho dos traços gerados.

Não modifique parâmetros definindo o número de processos ou processadores, visto que as aplicações do SPEC CPU 2006 usam apenas um processo, essa alteração apenas tornará a criação dos traços mais lenta (eu testei :P)

Se não soucer um bom tamanho, 200M de instruções parece uma boa :D

Após essa pequena mudança, precisamos editar o script **spec_cpu2006_tracer.py** presente no diretório: **trace_generator/scripts/**

Dois parâmetros precisam ser configurados:

- **SPEC_FOLDER**: Pasta onde foi instalado o spec na sua máquina, diretório raiz do SPEC CPU 2006.
- **PIN_POINTS_HOME**: Caminho completo até a pasta **trace_generator** do OrCS. *Pode-se descobrir isso ao executar o comando **pwd** dentro dessa pasta*

Execução

Após todas as configurações, basta executar o script **spec_cpu2006_tracer.py** do diretório **trace_generator/scripts/**:

```
>> python3 spec_cpu2006_tracer.py <id do benchmark> <sufixo dos traços>
```

Os dois parâmetros passados ao programa correspondem à aplicação da qual será retirado o traço e qual a terminação dos nomes dos arquivos gerados, um identificador para os resultados.

Um exemplo seria:

```
>> python3 spec_cpu2006_tracer.py 1 “.trace”
```

Lembre-se que as aspas são copiadas incorretamente dos pdfs

Ao fim da execução, seus traços estarão salvos em **trace_generator/scripts/Traces**.

Os identificadores de cada aplicação podem ser vistos na seção seguinte.

Identificadores de aplicações

Cada aplicação do SPEC é indicada por um identificador durante a chamada do script. Os identificadores de cada aplicação são listados abaixo:

Identificador – Aplicação

1 – astar

29 – zeusmp

2 – bzip2

3 – gcc

4 – gobmk

5 – h264ref

6 – hmmer

7 – libquantum

8 – mcf

9 – omnetpp

10 – perlbench

11 – sjeng

12 – xalancbmk

13 – bwaves

14 – cactusADM

15 – calculix

16 – dealII

17 – gamess

18 – GemsFDTD

19 – gromacs

20 – lbm

21 – leslie3d

22 – milc

23 – namd

24 – povray

25 – soplex

26 – sphinx3

27 – tonto

28 – wrf