



Universidade Estadual de Feira de Santana



Tutorial de utilização do Command-Line-Profiler

Feira de Santana - BA

Julho, 2016

## 1 Introdução

Este tutorial é um resultado do projeto de pesquisa do estudante bolsista (Fapesb) do LaCAD, Cássio Silva de Sá Santos e tem como objetivo apresentar um manual de utilização da ferramenta de medição de desempenho conhecida como Command-Line-Profilers que é disponibilizada através do Cuda ToolKit da Nvidia o qual já teve o seu tutorial de instalação desenvolvido e publicado por este bolsista.

Este tutorial tem como principal fonte a documentação oficial da Nvidia [4] e como complemento os testes e estudos feitos pelo bolsista.

O conteúdo deste tutorial foi criado para fins de pesquisa e pode ser usado livremente desde que citada a fonte. O LaCAD não se responsabiliza pelo uso dessas informações.

## 2 Command-Line-Profiler

O Command Line Profiler permite aos usuários obterem informações de tempo sobre as execuções dos kernels e operações de transferências de memórias para aplicações CUDA e OpenCL. As opções de perfil são controladas através de variáveis de ambiente e arquivos de configuração de perfil [2].

Os perfis de saída podem ser gerados em arquivos de texto nos formatos de par (Chave-Valor) ou separados por ponto-e-virgula.

**Obs.:** Esta ferramenta foi descontinuada e informa ao usuário que o mesmo deve migrar para o nvprof, pois o Command-Line-Profiler pode não estar mais presente nas versões mais atualizadas do Cuda-Tool-Kit [1].

### 2.1 Controle de perfis

As opções de perfil podem ser controladas através de variáveis de ambiente e um arquivo de configuração de perfil.

Para isso, simplesmente digite no terminal a variável com o valor correspondente seguindo o exemplo abaixo:

```
$ export VARIAVEL_DE_AMBIENTE=VALOR
```

Os valores dessas variáveis podem indicar um lógico para sim(1)/não(0), um valor inteiro ou um caminho de arquivo.

A **Tabela 1** exhibe as variáveis que podem ser usadas:

Variável	Valor	Descrição
COMPUTE_PROFILE	1(ativo) ou 0(inativo)	Habilita ou desabilita o profiling
COMPUTE_PROFILE_LOG		Serve para designar o caminho de um arquivo de saída para o perfil
COMPUTE_PROFILE_CSV	1(ativo) ou 0(inativo)	Ativa ou desativa a versão de saída do log separada por virgula (,)
COMPUTE_PROFILE_CONFIG		Usado para especificar um arquivo de configuração para ativar os contadores de performance da GPU
CUDA_PROFILE		Versão antiga, ainda suportada, para o COMPUTE_PROFILE
CUDA_PROFILE_LOG		Versão antiga, ainda suportada, para o COMPUTE_PROFILE_LOG
CUDA_PROFILE_CSV		Versão antiga, ainda suportada, para o COMPUTE_PROFILE_CSV
CUDA_PROFILE_CONFIG		Versão antiga, ainda suportada, para o COMPUTE_PROFILE_CONFIG

**Tabela 1:** Variáveis de ambiente usadas para controle do Command-Line-Profiler

Por padrão, se não for especificado nenhum caminho de log, o perfil vai registrar os dados em “cuda\_profile\_%d.log” em caso de um contexto CUDA e “opencl\_profile\_%d.log” em caso de um contexto OpenCL, nos quais %d é substituído pelo número do contexto.

**Obs.:** Caso as variáveis de ambiente CUDA\_PROFILE e COMPUTE\_PROFILE estiverem especificadas ao mesmo tempo, o Command Line Profiler vai priorizar o COMPUTE\_PROFILE.

## 2.2 Executando o Command Line Profiler

Para visualizar os dados gerados pelo Command Line Profiler execute os seguintes passos:

1. Ative a variável de ambiente que habilita o profiling.

```
$ export COMPUTE_PROFILE=1
```

2. Execute normalmente sua aplicação pré compilada utilizando:

```
$ ./seu_exec
```

3. Por padrão, o arquivo de saída será: cuda\_profile\_0.log, abra-o:

```
$ vi cuda_profile_0.log
```

Um exemplo de um arquivo de saída pode ser visualizado na **Figura 1** na qual o tempo é apresentado em microsegundos.

```

NV Warning: The legacy Command Line Profiler is deprecated and will be no longer available as of
# CUDA_PROFILE_LOG_VERSION 2.0
# CUDA_DEVICE 0 Tesla C2070
# CUDA_CONTEXT 1
# TIMESTAMPFACTOR 1456ffa32d46d766
method,gputime,cputime,occupancy
method=[ memset128 ] gputime=[ 528.480 ] cputime=[ 33.960 ] occupancy=[ 0.333 ]
method=[ memcpyHtoDasync ] gputime=[ 33302.945 ] cputime=[ 10.245 ]
method=[ _Z16increment_kernelPii ] gputime=[ 2005.120 ] cputime=[ 13.148 ] occupancy=[ 1.000 ]
method=[ memcpyDtoHasync ] gputime=[ 40782.398 ] cputime=[ 8.551 ]
~

```

**Figura 1:** Exemplo de execução usando o Command-Line-Profiler

Caso a variável de ambiente `COMPUTE_PROFILE_CSV` esteja ativada, o resultado será como o apresentado na **Figura 2**.

```

# CUDA_PROFILE_LOG_VERSION 2.0
# CUDA_DEVICE 0 Tesla C2070
# CUDA_CONTEXT 1
# CUDA_PROFILE_CSV 1
# TIMESTAMPFACTOR 1456ffa32dbaab22
method,gputime,cputime,occupancy
memset128,528.992,38.893,0.333
memcpyHtoDasync,33297.184,9.920
_Z16increment_kernelPii,2007.008,14.852,1.000
memcpyDtoHasync,40762.176,8.281

```

**Figura 2:** Exemplo de execução usando o Command-Line-Profiler

Neste modo, por padrão, podemos observar em cada uma das linhas do resultado, da esquerda para a direita, separadas por vírgula, as informações apresentadas na **Tabela 2** [3].

Coluna	Descrição
<b>method</b>	O nome do Kernel GPU ou método de copia de memória. No caso de Kernels o nome do método é o nome desconfigurado gerado pelo compilador.
<b>gputime</b>	O tempo de execução de um Kernel GPU ou de um método de copia de memória.
<b>cputime</b>	Para métodos não bloqueantes esta coluna equivale apenas ao tempo de overhead do lado do host necessário para executar o método. Para métodos bloqueante esta coluna equivale a soma do gputime e o overhead do CPU.
<b>occupancy</b>	Esta coluna indica a ocupância do multiprocessador que diz respeito a razão entre o número de <i>warps</i> ativos e o número máximo de <i>warps</i> suportados no multiprocessador de uma GPU.

**Tabela 2:** Informações apresentadas na tabela gerada pelo Command-Line-Profiler

## Referências

- [1] <http://docs.nvidia.com/cuda/profiler-users-guide/#nvprof-overview>
- [2] <http://docs.nvidia.com/cuda/profiler-users-guide/#compute-command-line-profiler-overview>
- [3] [https://hpc.oit.uci.edu/nvidia-doc/sdk-cuda-doc/C/doc/Compute\\_Command\\_Line\\_Profiler\\_User\\_Guide.pdf](https://hpc.oit.uci.edu/nvidia-doc/sdk-cuda-doc/C/doc/Compute_Command_Line_Profiler_User_Guide.pdf)
- [4] <http://docs.nvidia.com/cuda/profiler-users-guide/>