



Universidade Estadual de Feira de Santana



Tutorial de instalação do Ambiente Cuda para programação
paralela em GPU's

Feira de Santana - BA

Fevereiro, 2015

1 Introdução

Tutorial criado por Cássio Silva de Sá Santos, estudante da Universidade Estadual de Feira de Santana e bolsista do Laboratório de Computação de Alto Desempenho, com o objetivo de orientar como se instala o ambiente Cuda para programação paralela em GPU's.

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 Download dos Conteúdos

Você precisará baixar os apropriados Toolkit, drivers e SDK do site da Nvidia (<https://developer.nvidia.com/cuda-downloads>).

- `NVIDIA-Linux-x86_64-352.55.run`

```
$ wget http://us.download.nvidia.com/XFree86/Linux-x86_64/352.55/NVIDIA-Linux-x86_64-352.55.run
```

- `cuda_7.5.18_linux.run`

```
$ wget http://developer.download.nvidia.com/compute/cuda/7.5/Prod/local_installers/cuda_7.5.18_linux.run
```

- `gpucomputingsdk_4.0.17_linux.run`

```
$ wget http://developer.download.nvidia.com/compute/cuda/4_0/sdk/gpucomputingsdk_4.0.17_linux.run
```

3 Preparação para a instalação do Driver da Nvidia

RHEL e seus derivados vêm com o código aberto do driver da Nvidia chamado nouveau. Antes de instalar os drivers da Nvidia, você precisará garantir que os drivers nouveau não estejam carregados. Para isso, você deve editar `/boot/grub/grub.conf`:

```
$ sudo nano /boot/grub/grub.conf
```

Adicionar a seguinte linha de opção de Kernel:

- `rdblacklist=nouveau nouveau.modeset=0`

Agora você precisará instalar o grupo de pacotes de ferramentas de desenvolvimento e de bibliotecas de desenvolvimento, além de alguns outros pacotes listados abaixo, para isso, execute as seguintes instruções no terminal:

```
$ sudo yum groupinstall 'Development Tools' 'Development Libraries'
$ sudo yum install kernel-devel gcc-c++ freeglut freeglut-devel libX11
-devel mesa-libGLU-devel libXmu-devel libXi-devel gcc* compat-gcc* compat
-glibc* compat-lib*
```

Feito isso, você deverá reiniciar o sistema. Ao reinicia você deve notar que a resolução da tela ficou desconfigurada. Isso é devido a termos adicionado o drive de código aberto Nouveau na lista negra, e isso é um sinal de que estamos no caminho correto (Não se assuste).

Nesta etapa, devemos ir para o modo no SO sem interface gráfica (non-GUI mode) Conhecida como runlevel 3. Para isso, abra o terminal e digite o seguinte comando:

```
$ sudo init 3
```

Após entrar no modo texto. Mude o diretório para `/usr/src/kernels`:

```
$ cd /usr/src/kernels
```

Execute o comando `ls` para verificar a pasta presente dentro da diretório `kernels` e anote completamente o caminho para este diretório, exemplo:

```
/usr/src/kernels/2.6.32-573.7.1.el6.x86_64/
```

4 Instalação do Driver da Nvidia

Agora estamos prontos para começar a instalação do driver da Nvidia, para isso, dirija-se ao diretório dos downloads efetuados no passo 1 e torne os arquivos .run executáveis utilizando o comando chmod:

```
$ cd ~/Downloads
$ chmod a+x NV*; chmod a+x cuda*; chmod a+x gpu*
```

Finalmente você poderá executar o instalador. Primeiro é a instalação do Driver Nvidia (Tenha certeza de estar colocando o caminho encontrado no final do passo 2:

```
$ sudo sh NVIDIA-Linux-x86_64-352.55.run --kernel-source-path=/usr/src/
kernels/2.6.32-573.7.1.el6.x86_64/
```

Durante a instalação, aceite o termo de licença mostrado, bem como todas as opções e reinicie após a conclusão.

```
$ sudo reboot
```

Você deve notar que a resolução da interface voltou ao normal, indicando que obteve-se sucesso na instalação do driver da Nvidia. Agora vamos para a instalação do cudatoolkit.

5 Instalação do CudaToolkit

Abra o terminal e vá até o diretório onde foram baixados os arquivos do passo 1 e execute o arquivo `cuda_7.5.18_linux.run`

Obs.: Entre novamente no modo texto (`sudo init 3`)

```
$ cd ~/Downloads
$ sudo sh cuda_7.5.18_linux.run
```

Durante a instalação você será perguntando sobre o caminho da instalação. Entre com o caminho padrão (`/usr/local/cuda`)

Uma vez concluído, é necessário adicionar o caminho anterior às variáveis de caminho de ambiente. Para isso, crie o arquivo `cuda.conf`:

```
$ sudo nano /etc/ld.so.conf.d/cuda.conf
```

E adicione as seguintes linhas neste arquivo:

```
/usr/local/cuda/lib64
/usr/local/cuda/lib
```

Salve o arquivo criado pressionando `Ctrl+x`, seguido da tecla `'y'` e `Enter`. Agora execute:

```
$ sudo ldconfig
```

Para adicionar o caminho da instalação do cuda para às variáveis de caminho de ambiente, edite `~/.bash_profile` com um editor de texto:

```
$ cd
$ nano .bash_profile
```

Adicione as seguintes linhas:

- `export CUDA_INSTALL_PATH=/usr/local/cuda`
- `export PATH=(PATH: /usr/local/cuda/bin)`
- `export LD_LIBRARY_PATH=/usr/local/cuda/lib64`
- `export PATH=(PATH: /usr/local/cuda/lib)`

Depois, faça:

```
$ cd
$ nano .bashrc
```

Adicione:

- `export PATH=/usr/local/cuda-7.5/bin:$PATH`
- `export LD_LIBRARY_PATH=/usr/local/cuda-7.5/lib64:$LD_LIBRARY_PATH`

```
source ~/.bashrc
bash
```

6 Instalação do SDK

Para instalar o `gpucomputingsdk` volte a pasta `Downloads` e execute o ultimo arquivo de download restante.

```
$ sh gpucomputingsdk_4.0.17_linux.run
```

Durante a instalação você será perguntando o caminho para ser instalado. Como exemplo, o `sdk` aqui foi instalado na pasta: `~/Documentos/NVIDIA_GPU_Computing_SDK`. Concluido, você precisará compilar os arquivos no SDK.

```
$ cd ~/Documents/NVIDIA_GPU_Computing_SDK/C/  
$ make
```

Para chegar se tudo ocorreu corretamente, nós iremos executar o arquivo `deviceQuery` provando que o SDK foi instalado:

```
$ cd ~/Documents/NVIDIA\GPU\Computing\SDK/C/bin/linux/release/  
$ ./deviceQuery
```

Obs.: Para executar códigos Cuda deverá utilizar o sistema operacional desprovido de interface devido ao cancelamento de compito pelo `WatchDog` da placa de vídeo quando são executados códigos que exigem muita utilização de memória. (Ctrl + Alt + F3)