



Universidade Estadual de Feira de Santana



Tutorial de Utilização do MAGMA

Feira de Santana - BA

Fevereiro, 2016

1 Introdução

Tutorial criado por Lucas Santana, estudante da Universidade Estadual de Feira de Santana e bolsista do Laboratório de Computação de Alto Desempenho, com o objetivo de orientar sobre a utilização da ferramenta Matrix Algebra for GPU and Multicore Architectures (MAGMA) em máquina com placa GPU.

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 Passos do tutorial

2.1 Inclusão de bibliotecas necessárias para o funcionamento das funções MAGMA no código

- `cuda.h`
- `cublas_v2.h`
- `magma.h`
- `magma_lapack.h`

2.2 Inicializa o MAGMA

```
magma_init();
```

2.3 Parâmetros da Função

- Criação das matrizes e variáveis. O magma possui alguns tipos de dados próprios iniciados por `magma_` (para cada função verificar as variáveis necessárias e seus tipos em [2]);
- Alocação do espaço com `magma_xmalloc` ou `magma_Xmalloc_cpu` (X indica o tipo de dado, s-simples, d-double, c-simples complexo, z- double complexo, i-inteiro, o sufixo `_cpu` indica que a matriz será alocada na memória do host);
- Inicialização das matrizes (Para matrizes alocadas no device, utilizar a função `magma_Xsetmatrix` para enviar os dados e `magma_Xgetmatrix` para receber os dados).
Ex: `magma_dsetmatrix(n, m, A, n, dA, n)`; (Envia o conteúdo de A, tamanho `nxm`, alocada no host para `dA`, alocada no device)
`magma_dgetmatrix(n, m, dA, n, A, n)`; (Captura o conteúdo de `dA`, tamanho `nxm`, alocada no device para A, alocada no host)

2.4 Chamada da função a ser utilizada.

Nome similar ao do LAPACK, com o termo `magma_` na frente. Pode ter o sufixo `_cpu` se a matriz a ser trabalhada foi alocada em memória do host ou não se for alocada na memória do device.

2.5 Liberar as matrizes e ponteiros

- `free()`; para matrizes alocadas no host.
- `magma_free()`; para matrizes alocadas no device.

2.6 Finalizar o MAGMA

```
magma_finalize();
```

2.7 Executando o código com MAGMA

2.7.1 Compilando

```
gcc -Wall -DADD_ -I/usr/local/magma/include -I/usr/local/cuda/include -c -o magma.o magma.c  
gcc -Wall -o magma magma.o -L/usr/local/magma/lib -L/usr/local/cuda/lib64 -L/usr/local/  
openblas/lib -lmagma -lcublas -lcudart -lopenblas
```

2.7.2 Executando

```
nohup ./magma &
```

Referências

[1] <http://icl.cs.utk.edu/magma/>

[2] <http://icl.cs.utk.edu/projectsfiles/magma/doxygen/installing.html>