



UPPSALA
UNIVERSITET

PARALUTION - a Library for Iterative Sparse Methods on CPU and GPU Hands-on

Dimitar Lukarski

Division of Scientific Computing
Department of Information Technology
Uppsala Programming for Multicore Architectures Research Center
(UPMARC)
Uppsala University, Sweden

GPU-enabled Libraries Course Sept, 2013

Outline
Compilation
SpMV - Host
SpMV - Accelerator
SpMV - No Accelerator
SpMV - Format
SpMV - Performance



UPPSALA
UNIVERSITET

Outline

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

Sparse Matrix-Vector Multiplication - Without Accelerator

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance



UPPSALA
UNIVERSITET

WWW/Download

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

www.paralution.com

- ▶ User manual - design, usage, examples, etc
- ▶ Doxygen - source documentation
- ▶ White papers - reports

Download it

- ▶ Download the package (tar-gz-ball)
- ▶ Save it in $\sim/$
- ▶ Unpack it



Compilation

cmake

- ▶ Go to *paralution-0.3.0/build*
- ▶ `mkdir` if the directory does not exist
- ▶ `cmake` checks your system
- ▶ *\$ cmake .. && make*

Output *build/*

- ▶ *bin* - all examples (*cg*, *bicgstab*, *amg*, *benchmark*)
- ▶ *inc* - all includes (including the plug-ins)
- ▶ *lib* - static lib (*libparalution.a*)



Test Matrices

- ▶ The library supports matrix-market format
- ▶ You need to download the matrices (see [fomics-WiKi](#))

Four test matrices - SPD

- ▶ `gr_30_30` - Laplace (small)
- ▶ `G3_circuit` - circuit simulation (large)
- ▶ `ecology2` - animal/gene flow simulation (large)
- ▶ `thermal2` - steady thermal problem (large)



UPPSALA
UNIVERSITET

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

Sparse Matrix-Vector Multiplication - Without Accelerator

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance



SpMV - example

Perform SpMV

- ▶ Initialize PARALUTION
- ▶ Read Matrix A
- ▶ Allocate vectors x, y
- ▶ SpMV $y = Ax$
- ▶ Stop PARALUTION

Step-by-step

- ▶ Compile PARALUTION with cmake
- ▶ Pick your favorite editor
- ▶ Create a new file test.cpp in *paralution-0.3.0/*
- ▶ `$ g++ -fopenmp test.cpp -o test
build/lib/libparalution.a -Ibuild/inc/ -lcudart -lcublas
-lcusparses -L$CUDATOOLKIT_HOME/lib64`
- ▶ `./test mat/gr_30_30.mtx`



Main Body - src

```
#include <iostream>
#include <paralution.hpp>

using namespace paralution;

int main(int argc, char* argv[]) {

    init_paralution();

    info_paralution();

    // ...

    stop_paralution();

}
```




SpMV - src

```
LocalVector<double> x, y;  
LocalMatrix<double> A;
```

```
A.ReadFileMTX(std::string(argv[1]));  
x.Allocate("x", A.get_ncol());  
y.Allocate("y", A.get_nrow());
```

```
x.Ones(); // x = 1  
y.Zeros(); // y = 0
```

```
A.info();  
A.Apply(x, &y); // y = Ax
```

```
std::cout << x.Dot(y) << std::endl;
```

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance



SpMV OpenMP - Output

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

```
$ ./test ~/test-matrices/small/sym/gr_30_30.mtx  
Paralution platform is initialized
```

```
(platform info)
```

```
ReadFileMTX: filename=gr_30_30.mtx; reading...  
ReadFileMTX: filename=gr_30_30.mtx; done  
LocalMatrix name=gr_30_30.mtx; rows=900; cols=900  
nnz=7744; prec=64bit; format=CSR; host backend=  
={CPU(OpenMP)}; accelerator backend={GPU(CUDA)};  
current=CPU(OpenMP)  
356
```



UPPSALA
UNIVERSITET

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

Sparse Matrix-Vector Multiplication - Without
Accelerator

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance



SpMV on the GPU

Step-by-step

- ▶ Insert just before the SpMV
 - ▶ *A.MoveToAccelerator();*
 - ▶ *x.MoveToAccelerator();*
 - ▶ *y.MoveToAccelerator();*
- ▶ Recompile the code
- ▶ *\$ g++ -fopenmp test.cpp -o test
build/lib/libparalution.a -Ibuild/inc/ -lcudart -lcublas
-lcusparses -L/usr/local/cuda/lib64*
- ▶ Run it again
- ▶ *./test matrix.mtx*



SpMV - Output

```
$ ./spmv ~/test-matrices/small/sym/gr_30_30.mtx  
Paralution platform is initialized
```

```
(platform info)...
```

```
ReadFileMTX: filename=gr_30_30.mtx; reading...  
ReadFileMTX: filename=gr_30_30.mtx; done  
LocalMatrix name=gr_30_30.mtx; rows=900; cols=900  
nnz=7744; prec=64bit; format=CSR; host backend=  
={CPU(OpenMP)}; accelerator backend={GPU(CUDA)};  
current=GPU(CUDA)  
356
```



UPPSALA
UNIVERSITET

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

**Sparse Matrix-Vector Multiplication - Without
Accelerator**

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance



No Acclertor

- ▶ Run the code on the login node (it has no GPUs) – the code will run on the CPU/OpenMP

or

- ▶ Compile the code without CUDA, just OpenMP
 - ▶ *\$ module unload cudatoolkit*
 - ▶ *\$ cd build*
 - ▶ *\$ rm -rf **
 - ▶ *\$ cmake .. && make -j*
 - ▶ Recompile the SAME code
 - ▶ Run it again
 - ▶ *./spmv matrix.mtx*

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance



SpMV - Output (no CUDA)

```
$ ./spmvp ~/test-matrices/small/sym/gr_30_30.mtx  
Paralution platform is initialized
```

```
(platform info)...
```

```
ReadFileMTX: filename=gr_30_30.mtx; reading...  
ReadFileMTX: filename=gr_30_30.mtx; done  
LocalMatrix name=gr_30_30.mtx; rows=900; cols=900  
nnz=7744; prec=64bit; format=CSR; host backend=  
{CPU(OpenMP)}; accelerator backend={No Accelerator}  
current=CPU(OpenMP)  
356
```

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance



UPPSALA
UNIVERSITET

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

Sparse Matrix-Vector Multiplication - Without
Accelerator

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance



SpMV in Various Formats

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Step-by-step

- ▶ The default format is CSR
- ▶ Insert before the SpMV one of
 - ▶ *A.ConvertToCOO();*
 - ▶ *A.ConvertToELL();*
 - ▶ *A.ConvertToHYB();*
 - ▶ *A.ConvertToDIA();*
- ▶ DIA format is not always possible



UPPSALA
UNIVERSITET

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance

Download and Compilation

Sparse Matrix-Vector Multiplication - Host

Sparse Matrix-Vector Multiplication - Accelerator

Sparse Matrix-Vector Multiplication - Without Accelerator

Sparse Matrix-Vector Multiplication - Various Formats

Sparse Matrix-Vector Multiplication - Performance



SpMV - Performance

Step-by-step

- ▶ Measure the time for 1000 SpMV
- ▶ Include *sys/time.h* and use the function *gettimeofday()* for the time measurement
- ▶ What is the performance on the Host?
 - ▶ You can adjust the number of OpenMP threads by *set_omp_threads_parallelism(N)*;
- ▶ What is the performance on the GPU?
- ▶ Determine the best matrix format for this problem

Outline

Compilation

SpMV - Host

SpMV - Accelerator

SpMV - No Accelerator

SpMV - Format

SpMV - Performance