PMPP WiSe 2024 - Exercise 2

Johannes S. Mueller-Roemer and Sebastian Besler

TECHNISCHE UNIVERSITÄT DARMSTADT

WiSe 2024 – v1.00 (2023/07/21)
Sheet 2

Task 2.1: Setup, execution and submission

2.1a) Setup and execution

- **Download:** Acquire the framework from moodle
- **Login:** Use your TU-ID to gain access to the cluster. The compute nodes can't be accessed directly. Use one of the login nodes[1]
  me@home: `ssh <TU-ID>@lcluster<X>.hrz.tu-darmstadt.de`
- **Upload the framework (from your own machine):**
  me@home: `scp pmpp_ex2.tar.gz <TU-ID>@lcluster<X>.hrz.tu-darmstadt.de:~/pmpp_ex2.tar.gz`
- **Load cuda:**
  me@cluster: `module load cuda/12.5 gcc/13.1.0`
- **Extract framework:**
  me@cluster: `tar -xzvf pmpp_ex2.tar.gz` and `cd pmpp_ex2`
- **Setup build dir:**
  me@cluster: `mkdir build`
  me@cluster: `cd build`
- **Configure build:**
  me@cluster: `cmake -DCMAKE_BUILD_TYPE=Release ..`
  Note: Always profile the release build to avoid the overhead of debug builds and incorrect results
  Note: The project sets the commandline option `-lineinfo` for `nvcc` to allow `ncu` to include source files in the report
- **Build:**
  me@cluster: `make`
- **Run:**
  me@cluster: `cd ..`
  me@cluster: `sbatch run.sh`
- **List jobs** You may list your currently enqueued jobs using:
  me@cluster: `squeue`
- **Cancel job** You may cancel your currently enqueued jobs using:
  me@cluster: `scancel <job_id>`
- **Commandline parameters:**
  -v Print matrices to stdout
  -u Run unoptimized kernels
  -o Run optimized kernels
  -s Use small matrices/vectors

[1] List of login nodes: https://www.hrz.tu-darmstadt.de/hlr/betrieb_hlr/hardware_hlr_1/aktuell_verfuegbar_hlr/index.en.jsp

-m Use medium-sized matrices/vectors
-t<*x*> Run task <*x*>
When no arguments are given, all kernels will be executed using large matrices and vectors

2.1b) Viewing profiles

The script `run.sh` will call the script `prof.sh` which profiles the application code using the Nsight Compute CLI[2] and writes the results to the `out` directory (and packs them as `out.tar.gz`). Open the resulting reports using a locally installed version of Nsight Compute[3] (Windows/Linux/WSL).

2.1c) Optimization of kernels

The code provided includes the unoptimized kernels (the optimized and unoptimized versions are identical at the start). Use Nsight Compute to profile the kernels and determine the limiting factors. The commandline parameters `--section` and `--set` are used to profile subsets of metrics, reducing the amount of data collected and letting you focus on the specific issues.

Feel free to also alter the size or dimensionality of the grid and blocks of the optimized implementation.

2.1d) Submission

Submit your solution via moodle. Please provide

- a PDF with your answers to text assignments and the corresponding screenshots
- a zipped tar archive of the code (**without** the `build` and `out` directories):
  me@cluster: `tar -czvf pmpp_ex2.tar.gz pmpp_ex2`

2.1e) Grading

Each task (except for 2.1 and 2.4) will give you 1 point. Task 2.4 gives you 2 points due to comparatively high complexity. In order to get the bonus, you will need to achieve 75% of the total points of both exercises.

Task 2.2: Scalar multiplication (1 Point)

The framework contains the function `scale_vectors` which scales an array of vectors by a scalar value. The access pattern in the kernel prohibits proper coalescing.

2.2a) Analysis (Text)

Show the issue exists using Nsight Compute (screenshot and explanation of where the problem can be seen) and briefly explain the issue.
Note: Define the proper `--section` in `prof.sh`. A list of all `sections` is available through
me@cluster: `ncu --list-sections`.

2.2b) Implementation

Implement an optimized kernel `scale_vectors_kernel_optimized` in `kernels_optimized.cu` using an optimized access pattern.

[2] https://docs.nvidia.com/nsight-compute/NsightComputeCli/index.html
[3] https://docs.nvidia.com/nsight-compute/NsightCompute/index.html

2.2c) Verification (Text)

Show you solved the issue using Nsight Compute (screenshot and explanation of where the improvement can be seen) and explain the remaining stalls.

Task 2.3: Matrix transpose (1 Point)

The framework contains the function `transpose_matrix` which transposes a matrix. The implementation suffers from bank conflicts.

2.3a) Analysis (Text)

Show the issue exists using Nsight Compute (screenshot and explanation of where the problem can be seen) and briefly explain the issue.
Note: Define the proper `--set` in `prof.sh`. A list of all `sets` is available through
`me@cluster: ncu --list-sets`.

2.3b) Implementation

Implement an optimized kernel `transpose_matrix_kernel_optimized` in `kernels_optimized.cu` using an optimized access pattern.

2.3c) Verification (Text)

Show you solved the issue using Nsight Compute (screenshot and explanation of where the improvement can be seen).

Task 2.4: Roofline (2 Points)

The framework contains the function `compute_matrix_vector_product` to multiply a vector by a matrix.

2.4a) Analysis (Text)

Analyze the performance issues using Nsight Compute, list them and provide evidence (where applicable).
Note: Define the proper `--section` in `prof.sh`. A list of all `sections` is available through
`me@cluster: ncu --list-sections`.

2.4b) Implementation

Implement an optimized kernel `compute_matrix_vector_product_kernel_optimized` in `kernels_optimized.cu`.

2.4c) Verification (Text)

Show you solved the issues using Nsight Compute's roofline graph and explain the improvement.

Exercise2
LastName, FirstName: _____ EnrollmentID: ␣␣␣␣␣␣␣␣

Task 2.5: Divergence (1 Point)

The framework contains the function `cwise_op_vectors` which applies functions to an array of vectors (component-wise). The branches in control flow lead to thread divergence.

2.5a) Analysis (Text)

Show the issue exists using Nsight Compute (screenshot and explanation of where the problem can be seen) and briefly explain the issue.
Note: Define the proper `--section` in `prof.sh`. A list of all `sections` is available through
`me@cluster:` `ncu --list-sections`.

2.5b) Implementation

Implement an optimized kernel `cwise_op_vectors_kernel_optimized` in `kernels_optimized.cu` which does not diverge.

2.5c) Verification (Text)

Show you solved the issue using Nsight Compute (screenshot and explanation of where the improvement can be seen).