1 Profiling with NVIDIA Tools

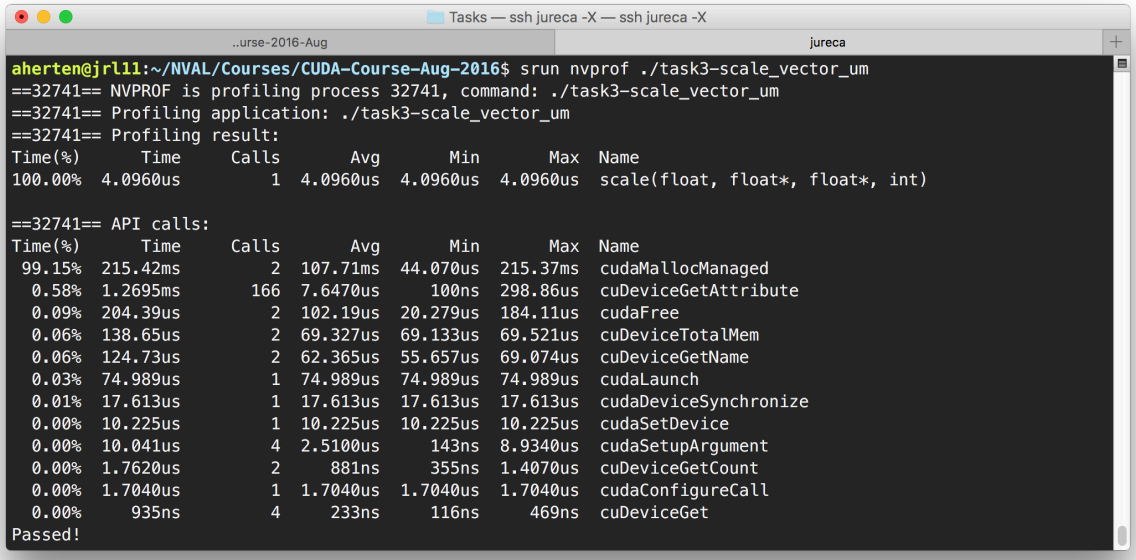
The CUDA Toolkit comes with two solutions for profiling an application: `nvprof`, which is a command line program, and the GUI application *NVIDIA Visual Profiler* (NVVP).

`nvprof` can be used in batch jobs or smaller interactive runs; NVVP can either import an `nvprof`-generated profile or run interactively through X forwarding.[1]

On JURON, the CUDA Toolkit can be loaded by `module load nvidia/cuda/8.0`; on JURECA, load it by `module load CUDA`.

1.1 Command Line: `nvprof`

For a quick overview of the GPU-invocations of an application, prefix its run with `nvprof`: `nvprof ./APP`.[2] `nvprof` instruments the application at run-time.



Among the many options of `nvprof` (see `nvprof --help`) it can export a profile for further usage through NVVP: `nvprof --export-profile FILE ./APP` will export the profile to `FILE`.

To make use of NVVP performance experiments, certain metrics need to be measured by `nvprof`: `nvprof --analysis-metrics --export-profile FILE ./APP` will export the metrics to `FILE`.

Further options of potential interest:

- `--print-gpu-trace`: Show trace of function calls
- `--openacc-profiling on`: Profile OpenACC as well (*on* by default)
- `--cpu-profiling on`: Enable some CPU profiling
- `--csv --log-file FILE`: Generate CSV output and save to `FILE`; handy for plots or benchmarked analysis
- `--metrics M1`: Measure only metric `M1` which is one of the NVIDIA-provided metrics which can be listed via `--query-metrics`.

→ docs.nvidia.com/cuda/profiler-users-guide/

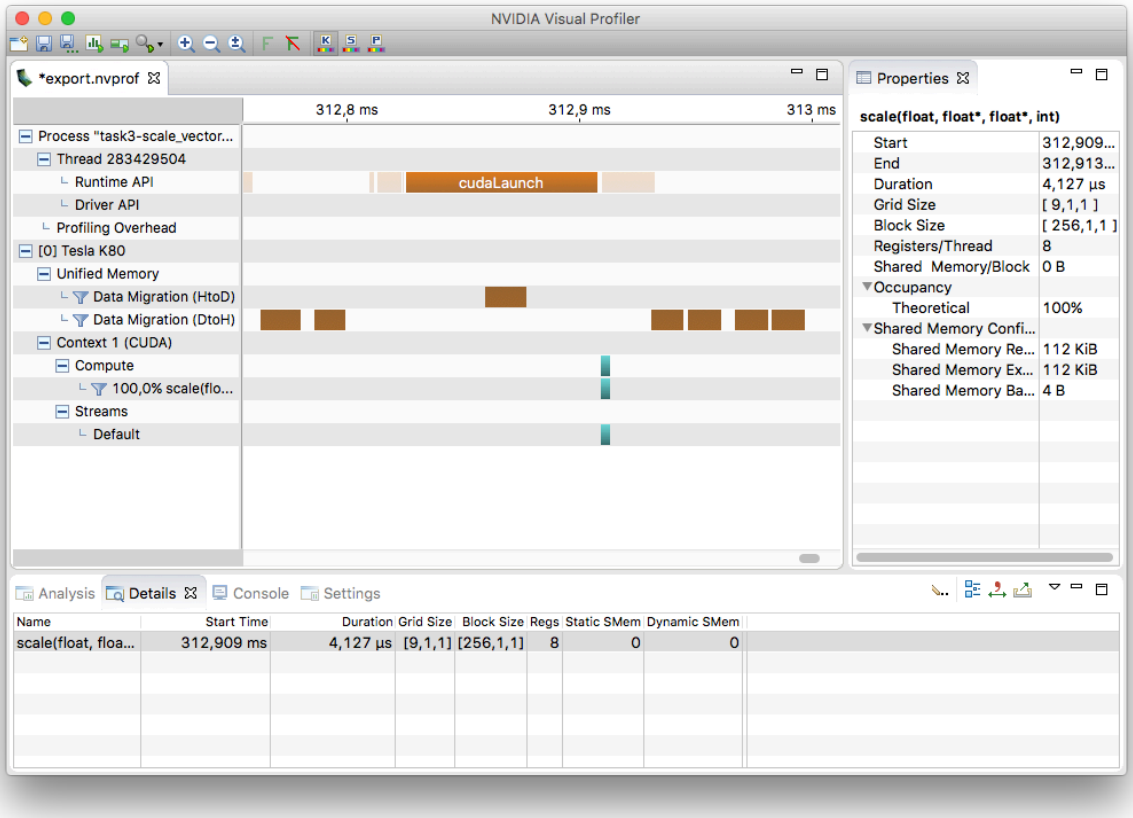
[1] The CUDA Toolkit is freeware and can be installed on your local machine; even on a laptop without an NVIDIA GPU. This allows for downloading generated `nvprof` profiles and importing them locally or even for connecting to a remote server with NVVP.

[2] Since your application might be run via a batch system, the call to `nvprof` might need to be prefixed by an `srun`. Like in the screenshot.

1.2 GUI: NVIDIA Visual Profiler

While `nvprof` can be used to collect information or display it concisely on the command line, the Visual Profiler (NVVP) can be helpful to understand an application through a timeline view and by running performance analyses.

NVVP can be launched from the command line with `nvvp` or by installing the CUDA Toolkit on a local machine.



→ developer.nvidia.com/nvidia-visual-profiler