

GPU Measurements with Score-P

Marc Schlütter, JSC



Environment for Score-P/Scalasca on Grete Phase 2

- Using GCC + OpenMPI tool:

```
% module use $PROJECT/modules
% module load gcc/13.2.0 openmpi/5.0.7
% module load scorep/10.1-grete-gcc-openmpi scalasca/2.6.2-grete-gcc-openmpi cubelib/4.9.1
```

- Setting up the hands-on materials for Score-P in the training scratch space:
 - Today: Looking at MPI+(OpenMP)+CUDA with the TeaLeaf CUDA mini-app
 - https://github.com/UK-MAC/TeaLeaf_CUDA

```
% mkdir -p /scratch/lustre-grete/projects/tr_20260831-vi-hps-tuning/$USER/scorep
% cd /scratch/lustre-grete/projects/tr_20260831-vi-hps-tuning/$USER/scorep
% tar xfv $PROJECT/training-materials/scorep/TeaLeaf_CUDA.tar.gz

% cd TeaLeaf_CUDA
```

TeaLeaf_CUDA configuration and compilation

■ The Makefile and its options

```
% vi Makefile
...
# GWDG Grete
# make clean
# make COMPILER=GNU MPI_COMPILER=mpif90 C_MPI_COMPILER=mpicc NV_ARCH=AMPERE PREP="scorep --cuda"
```

■ Running the compilation for the GCC OpenMPI toolchain

```
% vi make clean; make COMPILER=GNU MPI_COMPILER=mpif90 C_MPI_COMPILER=mpicc \
NV_ARCH=AMPERE PREP="scorep -cuda"
...
rm -f *.o *.mod *genmod* *.lst *.cub *.ptx tea_leaf *.s *.i
scorep --cuda mpif90 -O3 -fopenmp ... -g -c data.f90 -o data.o
scorep --cuda mpif90 -O3 -fopenmp ... -g -c definitions.f90 -o definitions.o
scorep --cuda mpif90 -O3 -fopenmp ... -g -c global_mpi.f90 -o global_mpi.o
...
scorep --cuda nvcc -ccbin mpicxx ... -c -g" -DNO_ERR_CHK -O3 -c cuda_errors.cu -o cuda_errors.o
scorep --cuda nvcc -ccbin mpicxx ... -c -g" -DNO_ERR_CHK -O3 -c cuda_strings.cu -o cuda_strings.o
...
-o bin.scorep/tea_leaf
```

Available options for CUDA and other paradigms

```
% scorep-info config-vars --full
...
SCOREP_CUDA_ENABLE
  Description: CUDA measurement features
    Type: Set
    Default: `yes`

Full description:
Sets the CUDA measurement mode to capture.

Notes:
- Options required by other options will be included automatically.
The following options or sets are available:
  runtime:          CUDA runtime API
  driver:           CUDA driver API
  kernel:           CUDA kernels on the accelerator device
  launch_parameters: Parameters passed to launch call of CUDA kernels
  memcpy:           CUDA memory copies
  malloc:           CUDA allocation and free calls
  user:             Record markers defined via NVTX
  default/yes/1:    CUDA runtime API and GPU activities.
...
```

▪ **scorep-info infos:**

- **config-summary:** Current Score-P configuration
- **config-vars:** Options for given configuration

Accelerator Paradigms

- Record CUDA applications and device activities

```
% export SCOREP_CUDA_ENABLE=runtime, kernel
```

- Record HIP applications and device activities

```
% export SCOREP_HIP_ENABLE=all
```

- Record OpenCL applications and device activities

```
% export SCOREP_OPENCL_ENABLE=api, kernel
```

- Record OpenACC applications

```
% export SCOREP_OPENACC_ENABLE=yes
```

- Can be combined with CUDA if it is a NVIDIA device

```
% export SCOREP_CUDA_ENABLE=kernel
```

TeaLeaf_CUDA summary and trace measurement collection

```
% cd bin.scorep
% cp ../jobscript/gretep2/scorep.slurm .
% vi scorep.slurm
#!/bin/bash
#SBATCH --partition=grete:shared
#SBATCH --gpus=A100:4
#SBATCH --reservation=vi-hps-tuning
...
##### Score-P measurement configuration
export SCOREP_CUDA_ENABLE=default

export SCOREP_FILTERING_FILE=../config/scorep.filter
#export SCOREP_ENABLE_TRACING=true
#export SCOREP_TOTAL_MEMORY=250M
export SCOREP_EXPERIMENT_DIRECTORY=scorep-tealeaf-${SLURM_NPROCS}
...
<save and exit>
% sbatch scorep.slurm
```

Note: instrumented
executable + enable
settings required

- Set the relevant paradigm vars
- Run with tracing enabled if analysis in Vampir is intended

TeaLeaf_CUDA Scalasca workaround

```
% cd bin.scorep
% cp ../jobscript/gretep2/scorep.slurm .
% vi scorep.slurm
#!/bin/bash
#SBATCH --partition=grete:shared
#SBATCH --gpus=A100:4
#SBATCH --reservation=vi-hps-tuning
...
##### Score-P measurement configuration
export SCOREP_CUDA_ENABLE=driver

export SCOREP_FILTERING_FILE=../config/scorep.filter
export SCOREP_TOTAL_MEMORY=250M
...
scan -t srun ./tea_leaf

<save and exit>
% sbatch scorep.slurm
```

Note: Be aware that kernel side effects are attributed to the host side waiting for the return of the kernel!

- Only record host side events (driver or runtime for CUDA)
- Run scan as before
- Use case: Analysis of the inter-process behavior of a MPI + CUDA app

Score-P: Further information

- Scalable Performance Measurement Infrastructure for Parallel Codes
 - Instrumenter, libraries, and tools to generate profile and trace measurements
 - Bundled with OTF2 (tracing), OPARI2 (OpenMP instrumentation), CubeWriter, and CubeLib (profiling)
- Available under 3-clause BSD open-source license
- Documentation & sources:
 - <https://score-p.org>
- User guide also part of installation:
 - `<prefix>/share/doc/scorep/pdf/scorep.pdf`
- Contact:
 - [mailto: support@score-p.org](mailto:support@score-p.org)

