Hands-on:
CloverLeaf_OpenACC w/ Score-P (I)

Markus Geimer
Jülich Supercomputing Centre

CloverLeaf_OpenACC: Makefile

```
#Crown Copyright 2012 AWE
#
# This file is part of CloverLeaf.
#
# CloverLeaf is free software...
#
# Agnostic, platform independent Makefile for the CloverLeaf benchmark code.
# It is not meant to be clever in any way, just a simple build script.
#
# this works as well:-
#
# make COMPILER=PGI # == NVHPC
#

...

#PREP=scorep --openacc --cuda --user

MPI_COMPILER=mpif90
C_MPI_COMPILER=mpicc


clover_leaf: *.f90 Makefile
      $(PREP) $(MPI_COMPILER) $(FLAGS) …
...
```

No instrumentation by default

Set/uncomment PREP macro for instrumenter preposition

Instrumenting CloverLeaf

Score-P instrumenter options:
--compiler: source code routines (default)
--mpp=mpi: MPI determined by heuristics
--openacc: enable OpenACC
--cuda: enable CUDA
--user: enable Score-P user API (optional)

```
% make clean
% make PREP="scorep --openacc --cuda --user"

mpicc -c timer_c.c
scorep --openacc --cuda --user  mpif90 -O3 -fastsse -acc -Minfo=acc -gpu=cc80 \
    data.f90 definitions.f90 pack_kernel.f90 clover.F90 report.f90 timer.f90 \
    parse.f90 read_input.f90 initialise_chunk_kernel.f90 initialise_chunk.f90 build_field.f90 \
    update_tile_halo_kernel.f90 update_tile_halo.f90 update_halo_kernel.f90 update_halo.f90 \
    ideal_gas_kernel.f90 ideal_gas.f90 start.f90 generate_chunk_kernel.f90 generate_chunk.f90 \
    initialise.f90 field_summary_kernel.f90 field_summary.f90 viscosity_kernel.f90 viscosity.f90 \
    calc_dt_kernel.f90 calc_dt.f90 timestep.f90 accelerate_kernel.f90 accelerate.f90 \
    revert_kernel.f90 revert.f90 PdV_kernel.f90 PdV.f90 flux_calc_kernel.f90 flux_calc.f90 \
    advec_cell_kernel.f90 advec_cell_driver.f90 advec_mom_kernel.f90 advec_mom_driver.f90 \
    reset_field_kernel.f90 reset_field.f90 hydro.F90 clover_leaf.F90 visit.f90 \
    timer_c.o \
    -o bin.scorep/clover_leaf
```

Mastering build systems

- Hooking up the Score-P instrumenter `scorep` into complex build environments like *Autotools* or *CMake* was always challenging
- Score-P provides convenience wrapper scripts to simplify this
- *Autotools* and *CMake* need the used compiler already in the *configure step,* but instrumentation should not happen in this step, only in the *build step*

```
% SCOREP_WRAPPER=off \
> cmake .. \
> -DCMAKE_C_COMPILER=scorep-nvcc \
> -DCMAKE_CXX_COMPILER=scorep-nvc++ \
> -DCMAKE_Fortran_COMPILER=scorep-nvfortran
```

Disable instrumentation in the *configure step*

Specify the wrapper scripts as the compiler to use

- Allows to pass addition options to the Score-P instrumenter and the compiler via environment variables without modifying the *Makefile*s
- Run `scorep-wrapper --help` for a detailed description and the available wrapper scripts of the Score-P installation

Measurement configuration: scorep-info

```
% scorep-info config-vars --full
SCOREP_ENABLE_PROFILING
  Description: Enable profiling
 [...]
SCOREP_ENABLE_TRACING
  Description: Enable tracing
 [...]
SCOREP_TOTAL_MEMORY
  Description: Total memory in bytes for the measurement system
 [...]
SCOREP_EXPERIMENT_DIRECTORY
  Description: Name of the experiment directory
 [...]
SCOREP_FILTERING_FILE
  Description: A file name which contain the filter rules
 [...]
SCOREP_METRIC_PAPI
  Description: PAPI metric names to measure
 [...]
SCOREP_METRIC_RUSAGE
  Description: Resource usage metric names to measure
 [...]
SCOREP_OPENACC_ENABLE
  Description: OpenACC measurement features
 [... More configuration variables ...]
```

- Score-P measurements are configured via environmental variables

Required for OpenACC measurements. [yes|default] recommended to start. Additional CUDA measurement optional.

Mastering heterogeneous applications

- Record CUDA application events and device activities

```
% export SCOREP_CUDA_ENABLE=default
```

- Record OpenCL application events and device activities

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

- Record OpenACC application events

```
% export SCOREP_OPENACC_ENABLE=regions,wait,enqueue
```

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

```
% export SCOREP_CUDA_ENABLE=kernel,kernel_callsite,idle
```

For all available options check:
scorep-info config-vars --full

CloverLeaf_OpenACC summary measurement collection

```
% cd bin.scorep
% cat scorep.sbatch
...
# Score-P measurement configuration
export SCOREP_OPENACC_ENABLE=default
export SCOREP_CUDA_ENABLE=default
export SCOREP_EXPERIMENT_DIRECTORY=scorep_clover_leaf_8_sum
#export SCOREP_FILTERING_FILE=scorep.filter

srun ./clover_leaf

% sbatch scorep.sbatch
```

- Change to the directory containing the new executable before running it with the desired configuration
- Check settings

Leave these lines commented out for the moment

- Submit job

CloverLeaf_OpenACC summary measurement execution

```
% cat CloverLeaf.<job_id>.out
 MPI rank     0 (0) using device 0/4 on jrc0319
 MPI rank     1 (1) using device 1/4 on jrc0319
 MPI rank     2 (2) using device 2/4 on jrc0319
 MPI rank     3 (3) using device 3/4 on jrc0319
 MPI rank     4 (0) using device 0/4 on jrc0320
 MPI rank     5 (1) using device 1/4 on jrc0320
 MPI rank     6 (2) using device 2/4 on jrc0320
 MPI rank     7 (3) using device 3/4 on jrc0320

    Clover Version 1.300
       MPI Version
   OpenACC Version 201711
   Task Count      8

 Output file clover.out opened. All output will go there.

 Step    1 time   0.0000000 control sound timestep  1.93E-04   1, 1 x  1.63E-04 y  1.63E-04
 Step    2 time   0.0001926 control sound timestep  1.18E-04   1, 1 x  1.63E-04 y  1.63E-04
 [...]
 Step   87 time   0.0155351 control sound timestep  1.83E-04   1, 1 x  1.63E-04 y  1.63E-04

 Wall clock     17.18185114860535
 First step overhead   4.0292978286743164E-002
```

- Verify the reported execution configuration and that the test execution passed

Compare to previous reference execution without instrumentation

CloverLeaf_OpenACC summary analysis report examination

```
% ls
clover_leaf* clover.in clover.out  CloverLeaf.<job_id>.out
scorep.sbatch  scorep_clover_leaf_8_sum/

% ls scorep_clover_leaf_8_sum
MANIFEST.md  profile.cubex  scorep.cfg




% cube_remap2 -d -o scorep_clover_leaf_8_sum/summary.cubex \
                   scorep_clover_leaf_8_sum/profile.cubex

% cube scorep_clover_leaf_8_sum/summary.cubex

      [CUBE GUI showing summary analysis report]
```

- Creates experiment directory including
  - A brief content overview (MANIFEST.md)
  - A record of the measurement configuration (scorep.cfg)
  - The analysis report that was collated after measurement (profile.cubex)
- Interactive exploration with Cube

Hint:
Copy ‘profile.cubex’ to local system (laptop) using ‘scp’ to improve responsiveness of GUI