In-Depth Performance Analysis
for OpenACC/CUDA/OpenCL Applications
with Score-P and Vampir

Hands-on-Lab @ GTC2015

Guido Juckeland (guido.juckeland@tu-dresden.de)
Agenda

- Motivation
- Performance Analysis 101
- Generating Traces with Score-P
- Visualizing Traces with Vampir
- Special Treat: OpenACC Tracing
- Looking a Little Deeper
Motivation

Guido Juckeland (guido.juckeland@tu-dresden.de)
Why are you here?

SO, YOU HAVE DECIDED TO UNDERSTAND WHAT A PROGRAM EXACTLY DOES?

CONGRATULATIONS!!!
YOU ARE AHEAD OF 99% OF YOUR COLLEAGUES

ARE YOU SERIOUS ABOUT THIS?

WHAT KIND OF PROFESSIONAL ARE YOU?

GO USE gprof OR THE LIKE...

FIND INTERESTING SPOTS

SELECT MIGNIFICATION

LOW
MED
HIGH

RUN IN PROFILING MODE
USE COMPILER WRAPPERS + FILTERS
INSTRUMENT THE CRITICAL PARTS

GET + INSTALL THE TOOLS

SCIENTIST
ENGINEER

MAYBE

SLACKER

Guido Juckeland
Performance engineering workflow

- Calculation of metrics
- Identification of performance problems
- Presentation of results
- Modifications intended to eliminate/reduce performance problem
- Collection of performance data
- Aggregation of performance data
- Prepare application with symbols
- Insert extra code (probes/hooks)
- Preparation
- Measurement
- Optimization
- Analysis
- Insert extra code (probes/hooks)
- Preparation
- Measurement
- Optimization
- Analysis
Performance Analysis 101

Guido Juckeland (guido.juckeland@tu-dresden.de)
Sampling vs. Tracing

2011/06/30 10:15:12.672865 Enter foo
2011/06/30 10:15:12.894341 Leave foo

Foo: Total Time 0.0815
Bar: Total Time 0.4711

Guido Juckeland – Slide 7
Terms Used and How They Connect

Data Presentation

Data Recording

Data Acquisition

Analysis Layer

Profiling
- Profiles

Tracing
- Timelines

Summarization

Loggging

Sampling

Event-based Instrumentation

Guido Juckeland – Slide 8
Score-P/Vampir Workflow for Small-Medium Sized Applications

Multi-Core Program

Score-P

Trace File (OTF2)

Vampir 8

1.0 ms

2.0 ms

Thread parallel

Small/Medium sized trace
Score-P Overview

- **Vampir**
- **Scalasca**
- **CUBE**
- **TAU**
- **TAUdb**
- **Periscope**

Event traces (OTF2)

Call-path profiles (CUBE4, TAU)

Online interface

Hardware counter (PAPI, rusage)

Score-P measurement infrastructure

Instrumentation wrapper

Process-level parallelism (MPI, SHMEM)

Thread-level parallelism (OpenMP, Pthreads)

Accelerator-based parallelism (CUDA, OpenCL)

Source code instrumentation

User instrumentation

Application
Partners

- Forschungszentrum Jülich, Germany
- German Research School for Simulation Sciences, Aachen, Germany
- Gesellschaft für numerische Simulation mbH Braunschweig, Germany
- RWTH Aachen, Germany
- Technische Universität Dresden, Germany
- Technische Universität München, Germany
- University of Oregon, Eugene, USA
Hands-on: CUDA Tracing in Your Own AWS Instance

Guido Juckeland (guido.juckeland@tu-dresden.de)
Connection Instructions

• Navigate to nvlabs.qwiklab.com
• Login or create a new account
• Select the “Instructor-Led Hands-on Labs” class
• Find the lab called “Analysis for OpenACC/CUDA/OpenCL Applications with Score-P and Vampir (S5721 - GTC 2015)” and click Start
• After a short wait, lab instance connection information will be shown
• Please ask Lab Assistants for help!
Performance Analysis Steps

1. Reference preparation for validation
2. Program instrumentation
3. Event trace collection
4. Event trace examination & analysis
Go to CUDA Example and Compile

Go to CUDA Example

% cd codes/cuda

Compile

% make
scorep --cuda /usr/local/anaconda/bin/mpicxx -Icommon/inc
-o simpleMPI_mpi.o -c simpleMPI.cpp
scorep --cuda "/usr/local/cuda-6.5"/bin/nvcc -ccbin g++ -Icommon/inc
-m64 -gencode arch=compute_30,code=sm_30 -gencode arch=compute_35,
code=sm_35 -gencode arch=compute_37,code=sm_37 -gencode
arch=compute_50,code=sm_50 -gencode arch=compute_50,code=compute_50
-o simpleMPI.o -c simpleMPI.cu
scorep --cuda /usr/local/anaconda/bin/mpicxx -o simpleMPI
simpleMPI_mpi.o simpleMPI.o -L"/usr/local/cuda-6.5"/lib64 -lcudart
Run Example

Run

% mpiexec -np 4 ./simpleMPI
Running on 4 nodes
Average of square roots is: 0.667305
PASSED

Find Tracefile appearing

% ls
Makefile
NsightEclipse.xml
readme.txt
scorep-20150311_2045_907655747320
simpleMPI
simpleMPI.cpp
simpleMPI.cu
simpleMPI_mpi.o
simpleMPI.o
simpleMPI.h
What Happened Behind the Scenes?

Score-P performance monitor loaded on login

- Done via an environment module
- Also sets the following environment variables (it would be up to you)

```
% export SCOREP_ENABLE_TRACING=true
% export SCOREP_ENABLE_PROFILING=false
% export SCOREP_OPENCL_ENABLE=true
% export SCOREP_CUDA_ENABLE=driver,kernel,memcpy,flushatexit
% export SCOREP_OPENACC_ENABLE=true
```
What Happened Behind the Scenes? (2)

Makefile modified to instrument application

- Using `scorep` compiler wrapper

Before:

```
NVCC := $(CUDA_PATH)/bin/nvcc -ccbin $(GCC)
MPICXX ?= $(shell which mpicxx 2>/dev/null)
```

After:

```
NVCC := scorep --cuda $(CUDA_PATH)/bin/nvcc -ccbin $(GCC)
MPICXX ?= scorep --cuda $(shell which mpicxx 2>/dev/null)
```
Trace Visualization with Vampir

Guido Juckeland (guido.juckeland@tu-dresden.de)
Typical questions that Vampir helps to answer:

- What happens in my application execution during a given time in a given process or thread?
- How do the communication patterns of my application execute on a real system?
- Are there any imbalances in computation, I/O or memory usage and how do they affect the parallel execution of my application?
Event Trace Visualization with Vampir

- Alternative and supplement to automatic analysis
- Show dynamic run-time behavior graphically at any level of detail
- Provide statistics and performance metrics

Timeline charts
- Show application activities and communication along a time axis

Summary charts
- Provide quantitative results for the currently selected time interval
The main displays of Vampir

**Timeline Charts:**
- Master Timeline
- Process Timeline
- Counter Data Timeline
- Performance Radar

**Summary Charts:**
- Function Summary
- Message Summary
- Process Summary
- Communication Matrix View
Let’s Open Your Tracefile

Start Vampir

ScoreP version 1.4 loaded
SCOREP_ROOT=/opt/scorep
scorep --cuda /usr/local/anaconda/bin/mplcxx -Icommon/lnc -o simpleMPI_mpi.o -c simpleMPI.cpp
scorep --cuda /usr/local/cuda-6.5/bin/nvcc -ccbin g++ -Icommon/lnc -m64 -encode arch=compute_30,code=sm_30 -encode arch=compute_35,code=sm_35 -encode arch=compute_50,code=sm_50 -o simpleMPI.o -c simpleMPI.cpp
scorep --cuda /usr/local/anaconda/bin/mplcxx -o simpleMPI simpleMPI_mpi.o simpleMPI.o -L/usr/local/cuda-6.5.1/lib64 -lcudart
Running on 8 nodes
Average of square roots is: 0.607337
PASSED
Let’s Open Your Tracefile (2)

Click on “Open Other”
Let’s Open Your Tracefile (3)

Select “Local File”
Let’s Open Your Tracefile (4)

Navigate to “home”, “ubuntu”, “codes”, “cuda”, “scorep*”, Open “traces.otf2”
Let’s Open Your Tracefile (5)

Maximize the Vampir window
What Do You See?

- Navigation Toolbar
- Display Toolbar
- Function Summary
- Function Legend
- Master Timeline
- Context View
Clicking on anything provides details in the context view
Zooming is done by click, hold, release
  - Horizontal (Undo: Ctrl+Z, Reset: Ctrl+R)
  - Vertical (Undo: Ctrl+Z, Reset: Ctrl+Shift+R)
Navigation Toolbar provides ways of sliding and zooming
Adding more displays via display toolbar
Moving displays around, dock to any border

Now you go ahead!
Changing displays

Right click on anything
Tasks

- Right click into Master Timeline
- Adjust Process Bar Height to fit Chart Height
- Determine length of initialization phase
- Determine length of compute phase
- Determine kernel runtime
- Determine message sizes
Detailed information about functions, communication and synchronization events for collection of processes.
Detailed information about different levels of function calls in a stacked bar chart for an individual process.
Detailed profiles on the messages sent/received in the application (includes CUDA memcpy).
Profiling At Its Best

- All displays are updated to the currently zoomed time interval

Function Summary
- Include/exclude functions
- Change metric
- Select processes used for profile

Message Summary
- Change metric
- Select only specific senders/receivers
There Is an Example Trace to Play With

Go and look under /home/ubuntu/traces/cuda for more traces

Now go and play with your or my trace –
tell me how to improve the application
A Look Ahead: OpenACC Tracing

Guido Juckeland (guido.juckeland@tu-dresden.de)
Disclaimer

- Your are looking at a prototype
- Only works with PGI compilers and developer version of Score-P
- If you find it cool – talk to your OpenACC compiler vendor 😊
Start a Terminal

Guido Juckeland
Switch to developer version of Score-P

% ubuntu@ip-172-31-3-169:~$ module purge
ScoreP version 1.4 unloaded

% module av
-------------------- /usr/share/modules/versions ------
3.2.10
-------------------- /usr/share/modules/modulefiles ----
dot               modules         scorep/dev-openacc
module-git       null             use.own
module-info      scorep/1.4(default)

% module load scorep/dev-openacc
ScoreP version openacc loaded
    SCOREP_ROOT=/opt/scorep-openacc
Go to OpenACC Example

% cd codes/openacc

Compile

% make

scorep --cuda pgcc -mp -ta=nvidia matmul_openacc.c -o matmul_openacc
Run Example

Run

% export OMP_NUM_THREADS=8

% ./matmul_openacc

CPU MM with 8 threads
MM on CPU: 1.658984 sec
mm_oacc_kernel(): 0.207447 sec
OpenACC matrix multiplication test was successful!
mm_oacc_kernel_with_init(): 0.052948 sec
OpenACC matrix multiplication test was successful!
mm_oacc_parallel_with_init(): 0.051797 sec
OpenACC matrix multiplication test was successful!
Total runtime: 0.325640 sec
Open the New Tracefile
There Is an Example Trace to Play With

Go and look under /home/ubuntu/traces/openacc for more traces

Now go and play with your or my trace – tell me how to improve the application
Looking a Little Further

Guido Juckeland (guido.juckeland@tu-dresden.de)
How About Very Large Tracefiles?

VampirServer

Score-P

Many-Core Program

Large Trace File (OTF2)

LAN/WAN

MPI parallel application

Large Trace File (stays on remote machine)
Performance Analysis is valuable

Use “easy” tools first

Score-P can record any concurrent activity

Vampir can visualize all that activity

The rest is experience and up to you 😊
Vampir is available at http://www.vampir.eu,
get support via vampirsupport@zih.tu-dresden.de