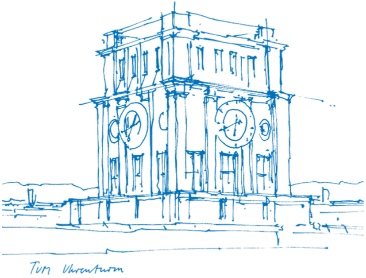
HPC Lab

Session 4: Profiler

Carsten Uphoff, Chaulio Ferreira, Michael Bader
TUM – SCCS

4 December 2017



Profiling

Profiling allows you to learn where your program spent its time [. . .]. This information can show you which pieces of your program are slower than you expected, and might be candidates for rewriting to make your program execute faster.

Source: `https://sourceware.org/binutils/docs/gprof/Introduction.html`

GNU gprof

- Compile the program with profiling enabled:
  `icpc -pg program.c -o program`
- Execute the program and generate profile data:
  `./program`
  (will generate a file `gmon.out`)
- Analyze the data with `gprof`:
  `gprof program`

GNU gprof - Flat profile

```
 Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 21.75      0.05     0.05  5050000     0.01     0.01  solver::vec2<float>::vec2(floa
 13.05      0.08     0.03     5000     6.00     6.00  WavePropagation::updateUnknown
  8.70      0.10     0.02  1010000     0.02     0.03  solver::FWave<float>::f(solver
  8.70      0.12     0.02  1010000     0.02     0.02  solver::matrix2x2<float>::matr
  ...
```

GNU gprof - Call tree

```
        Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 4.35% of 0.23 seconds

index % time  self  children   called     name
...
               0.02    0.00    5001/5001       main [1]
[12]      8.7  0.02    0.00    5001          writer::VtkWriter::write(float, float const*, float c
               0.00    0.00    5001/5001       writer::VtkWriter::generateFileName() [33]
               0.00    0.00    5001/5002       std::operator|(std::_Ios_Openmode, std::_Ios_Oper
...
```

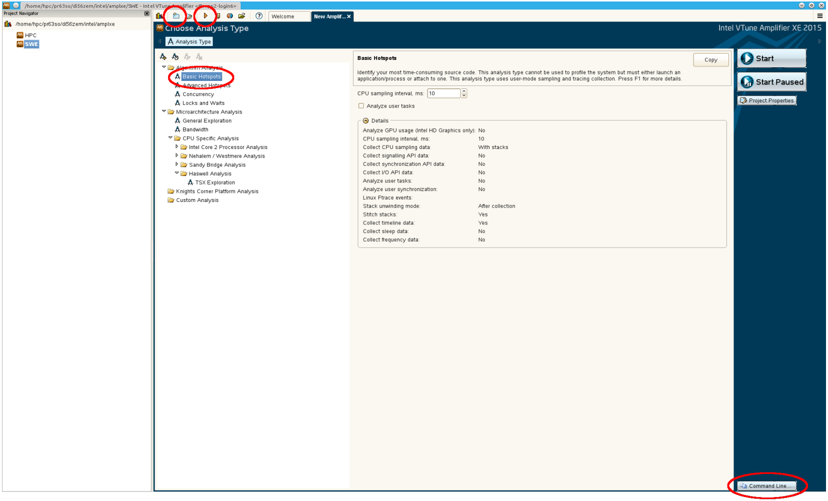
Intel VTune Amplifier XE

On the cluster:

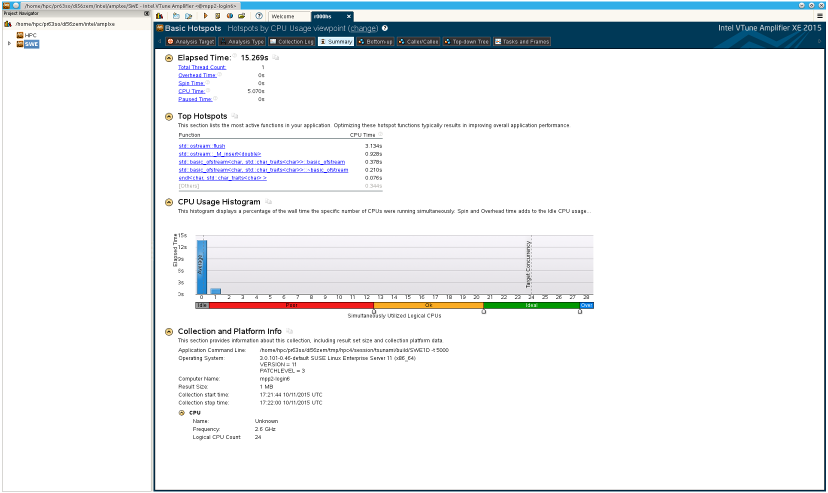
- `module load amplifier_xe`
- GUI: `amplxe-gui`
- Command line tool: `amplxe-cl`

Example: `amplxe-cl -collect hotspots ./program`

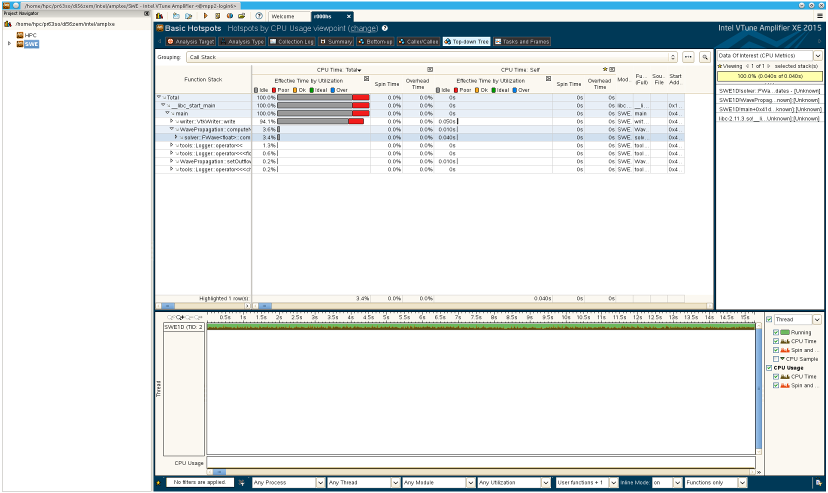
Intel VTune Amplifier XE



Intel VTune Amplifier XE



Intel VTune Amplifier XE



Scalasca

Open source project:
Forschungszentrum Jülich,
Technische Universität Darmstadt,
German Research School for Simulation Sciences

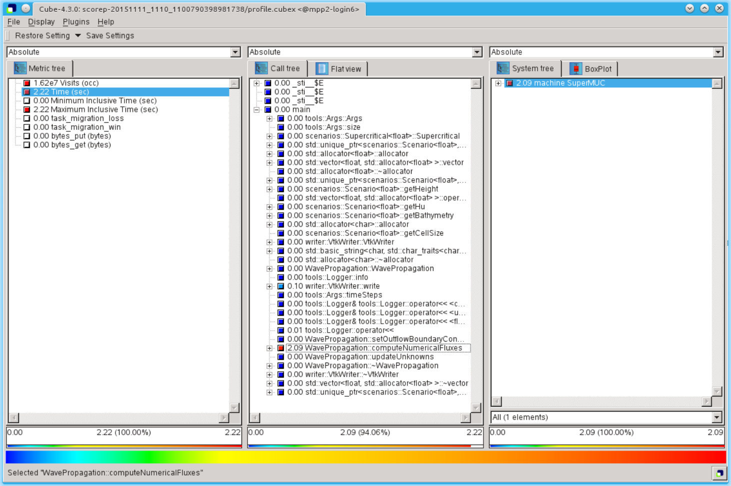
On the Linux Cluster:

- `module load scalasca`
- Also loads:
  - `Score-P` (Code instrumentation)
  - `Cube` (Visualisation)

Scalasca – instrumentation

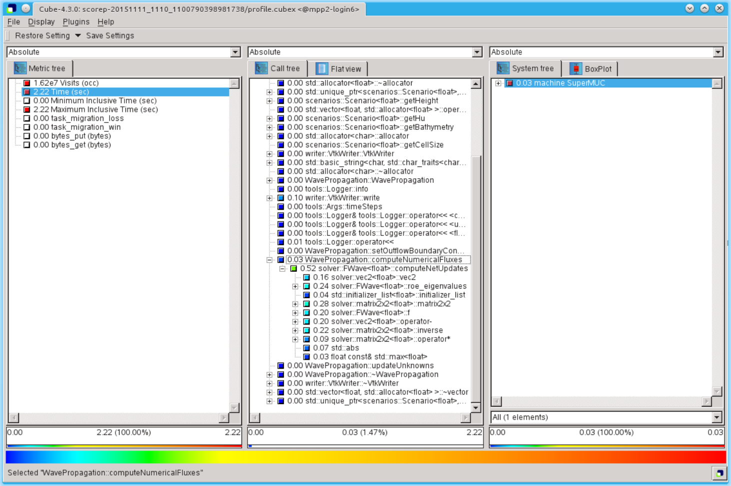
- `scalasca -instrument [options] compiler ...`
  Installation on the Linux Cluster only works with the Intel compiler
  Custom installation for GCC possible
- Options:
  - `--mpp=mpi`
  - `--thread=omp`
  - `--nocompiler`
  - `--user`
  - ...
- Instrumented code generates a folder `scorep-*`

Cube

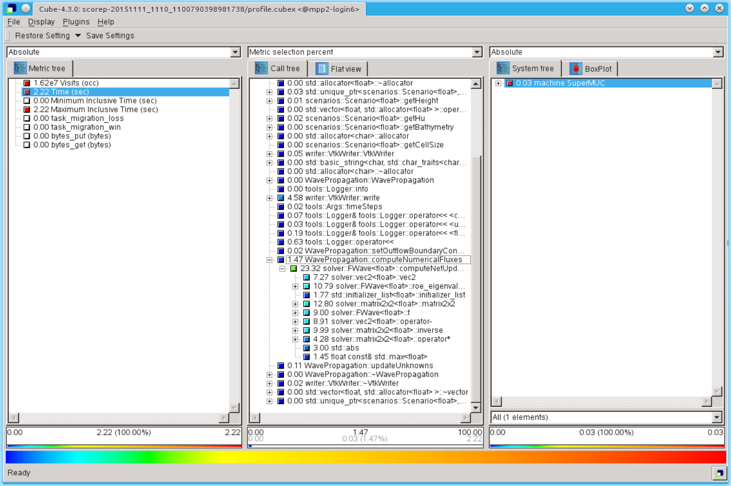


Cube-4.3.0: scorep-20151111_1110_1100790398981738/profile.cubex <@mpp2-login6>
File Display Plugins Help
Restore Setting Save Settings
Absolute
Metric tree
1.62e7 Visits (occ)
2.22 Time (sec)
0.00 Minimum Inclusive Time (sec)
2.22 Maximum Inclusive Time (sec)
0.00 task_migration_loss
0.00 task_migration_win
0.00 bytes_put (bytes)
0.00 bytes_get (bytes)
Absolute
Call tree
Flat view
0.00 _sti__$E
0.00 _sti__$E
0.00 _sti__$E
0.00 main
0.00 tools::Args::Args
0.00 tools::Args::size
0.00 scenarios::Supercritical<float>::Supercritical
0.00 std::unique_ptr<scenarios::Scenario<float>,...
0.00 std::allocator<float>::allocator
0.00 std::vector<float, std::allocator<float> >::vector
0.00 std::allocator<float>::~allocator
0.00 std::unique_ptr<scenarios::Scenario<float>,...
0.00 scenarios::Scenario<float>::getHeight
0.00 std::vector<float, std::allocator<float> >::oper...
0.00 scenarios::Scenario<float>::getHu
0.00 scenarios::Scenario<float>::getBathymetry
0.00 std::allocator<char>::allocator
0.00 scenarios::Scenario<float>::getCellSize
0.00 writer::VtkWriter::VtkWriter
0.00 std::basic_string<char, std::char_traits<char...
0.00 std::allocator<char>::~allocator
0.00 WavePropagation::WavePropagation
0.00 tools::Logger::info
0.10 writer::VtkWriter::write
0.00 tools::Args::timeSteps
0.00 tools::Logger& tools::Logger::operator<< <c...
0.00 tools::Logger& tools::Logger::operator<< <u...
0.00 tools::Logger& tools::Logger::operator<< <fl...
0.01 tools::Logger::operator<<
0.00 WavePropagation::setOutflowBoundaryCon...
2.09 WavePropagation::computeNumericalFluxes
0.00 WavePropagation::updateUnknowns
0.00 WavePropagation::~WavePropagation
0.00 writer::VtkWriter::~VtkWriter
0.00 std::vector<float, std::allocator<float> >::~vector
0.00 std::unique_ptr<scenarios::Scenario<float>,...
Absolute
System tree
BoxPlot
2.09 machine SuperMUC
All (1 elements)
0.00 2.22 (100.00%) 2.22
0.00 2.09 (94.06%) 2.22
0.00 2.09 (100.00%) 2.09
Selected "WavePropagation::computeNumericalFluxes"

Cube


Cube-4.3.0: scorep-20151111_1110_1100790398981738/profile.cubex <@mpp2-login6>
File Display Plugins Help
Restore Setting Save Settings
Absolute
Metric tree
1.62e7 Visits (occ)
2.22 Time (sec)
0.00 Minimum Inclusive Time (sec)
2.22 Maximum Inclusive Time (sec)
0.00 task_migration_loss
0.00 task_migration_win
0.00 bytes_put (bytes)
0.00 bytes_get (bytes)
Absolute
Call tree
Flat view
0.00 std::allocator<float>::~allocator
0.00 std::unique_ptr<scenarios::Scenario<float>,...
0.00 scenarios::Scenario<float>::getHeight
0.00 std::vector<float, std::allocator<float> >::oper...
0.00 scenarios::Scenario<float>::getHu
0.00 scenarios::Scenario<float>::getBathymetry
0.00 std::allocator<char>::allocator
0.00 scenarios::Scenario<float>::getCellSize
0.00 writer::VtkWriter::VtkWriter
0.00 std::basic_string<char, std::char_traits<char...
0.00 std::allocator<char>::~allocator
0.00 WavePropagation::WavePropagation
0.00 tools::Logger::info
0.10 writer::VtkWriter::write
0.00 tools::Args::timeSteps
0.00 tools::Logger& tools::Logger::operator<< <c...
0.00 tools::Logger& tools::Logger::operator<< <u...
0.00 tools::Logger& tools::Logger::operator<< <fl...
0.01 tools::Logger::operator<<
0.00 WavePropagation::setOutflowBoundaryCon...
0.03 WavePropagation::computeNumericalFluxes
0.52 solver::FWave<float>::computeNetUpdates
0.16 solver::vec2<float>::vec2
0.24 solver::FWave<float>::roe_eigenvalues
0.04 std::initializer_list<float>::initializer_list
0.28 solver::matrix2x2<float>::matrix2x2
0.20 solver::FWave<float>::f
0.20 solver::vec2<float>::operator-
0.22 solver::matrix2x2<float>::inverse
0.09 solver::matrix2x2<float>::operator*
0.07 std::abs
0.03 float const& std::max<float>
0.00 WavePropagation::updateUnknowns
0.00 WavePropagation::~WavePropagation
0.00 writer::VtkWriter::~VtkWriter
0.00 std::vector<float, std::allocator<float> >::~vector
0.00 std::unique_ptr<scenarios::Scenario<float>,...
Absolute
System tree
BoxPlot
0.03 machine SuperMUC
All (1 elements)
0.00 2.22 (100.00%) 2.22
0.00 0.03 (1.47%) 2.22
0.00 0.03 (100.00%) 0.03
Selected "WavePropagation::computeNumericalFluxes"

Cube


Cube-4.3.0: scorep-20151111_1110_1100790398981738/profile.cubex <@mpp2-login6>
File Display Plugins Help
Restore Setting Save Settings
Absolute
Metric selection percent
Absolute
Metric tree
Call tree
Flat view
System tree
BoxPlot
1.62e7 Visits (occ)
2.22 Time (sec)
0.00 Minimum Inclusive Time (sec)
2.22 Maximum Inclusive Time (sec)
0.00 task_migration_loss
0.00 task_migration_win
0.00 bytes_put (bytes)
0.00 bytes_get (bytes)
0.00 std::allocator<float>::~allocator
0.03 std::unique_ptr<scenarios::Scenario<float>,...
0.01 scenarios::Scenario<float>::getHeight
0.00 std::vector<float, std::allocator<float> >::oper...
0.02 scenarios::Scenario<float>::getHu
0.00 scenarios::Scenario<float>::getBathymetry
0.00 std::allocator<char>::allocator
0.00 scenarios::Scenario<float>::getCellSize
0.05 writer::VtkWriter::VtkWriter
0.00 std::basic_string<char, std::char_traits<char...
0.00 std::allocator<char>::~allocator
0.00 WavePropagation::WavePropagation
0.00 tools::Logger::info
4.58 writer::VtkWriter::write
0.02 tools::Args::timeSteps
0.07 tools::Logger& tools::Logger::operator<< <c...
0.03 tools::Logger& tools::Logger::operator<< <u...
0.19 tools::Logger& tools::Logger::operator<< <fl...
0.63 tools::Logger::operator<<
0.02 WavePropagation::setOutflowBoundaryCon...
1.47 WavePropagation::computeNumericalFluxes
23.32 solver::FWave<float>::computeNetUpd...
7.27 solver::vec2<float>::vec2
10.79 solver::FWave<float>::roe_eigenval...
1.77 std::initializer_list<float>::initializer_list
12.80 solver::matrix2x2<float>::matrix2x2
9.00 solver::FWave<float>::f
8.91 solver::vec2<float>::operator-
9.99 solver::matrix2x2<float>::inverse
4.28 solver::matrix2x2<float>::operator*
3.00 std::abs
1.45 float const& std::max<float>
0.11 WavePropagation::updateUnknowns
0.00 WavePropagation::~WavePropagation
0.02 writer::VtkWriter::~VtkWriter
0.00 std::vector<float, std::allocator<float> >::~vector
0.00 std::unique_ptr<scenarios::Scenario<float>,...
0.03 machine SuperMUC
All (1 elements)
0.00 2.22 (100.00%) 2.22
0.00 1.47 100.00
0.00 0.03 (1.47%) 2.22
0.00 0.03 (100.00%) 0.03
Ready

Score-P – Manual Instrumentation

- Option: `--user`
- Functions:

```
#include <scorep/SCOREP_User.h>

void foo(x) {
  SCOREP_USER_REGION( "foo", SCOREP_USER_REGION_TYPE_FUNCTION )
  // Do something
}
```

- Regions:

```
#include <scorep/SCOREP_User.h>
void foo() {
  SCOREP_USER_REGION_DEFINE( handle )
  // Do something
  SCOREP_USER_REGION_BEGIN( handle, "region", SCOREP_USER_REGION_TYPE_COMMON )
  // Do something else
  SCOREP_USER_REGION_END( handle )
  // Do more
}
```

Score-P – Parameter-Based Profiling

```c
#include <scorep/SCOREP_User.h>

void foo(int64_t myint)
{
  SCOREP_USER_REGION_DEFINE( handle )
  SCOREP_USER_REGION_BEGIN( handle, "foo", SCOREP_USER_REGION_TYPE_COMMON )
  SCOREP_USER_PARAMETER_INT64( "myint", myint )

  // do something

  SCOREP_USER_REGION_END( handle )
}
```

Hardware Performance Counters

- Hardware counters are special registers
- Count events, e.g.
  - total instructions
  - cache misses
  - branch misses
  - ...
- Automatically incremented by the hardware
  $\rightarrow$ no instrumentation required
  $\rightarrow$ minimal overhead
- Likwid-perfctr
  - Linux tool/kernel module to read hardware counters
  - Developed by Regionales Rechenzentrums Erlangen (RRZE)

Likwid-perfctr

```
(...)
--------------------------------------------------------------------------------
DGEMM with options:
Time/FLOPS with timeofday(): 1.020164e+10
--------------------------------------------------------------------------------
Group 1: MEM
+-----------------------+---------+-------------+
|        Event          | Counter |   Core 1    |
+-----------------------+---------+-------------+
|   INSTR_RETIRED_ANY   |  FIXC0  | 68146405143 |
| CPU_CLK_UNHALTED_CORE |  FIXC1  | 52138809008 |
|  CPU_CLK_UNHALTED_REF |  FIXC2  | 41113421180 |
(...)
+-----------------------------------+--------------+
|              Metric               |    Core 1    |
+-----------------------------------+--------------+
(...)
|  Memory read bandwidth [MBytes/s] | 1.014995e+04 |
|  Memory read data volume [GBytes] | 1.621715e+02 |
| Memory write bandwidth [MBytes/s] | 4.532920e+01 |
| Memory writo data volume [GBytes] | 0.72425024 |
|    Memory bandwidth [MBytes/s]    | 1.019528e+04 |
|    Memory data volume [GBytes]    | 1.628957e+02 |
+-----------------------------------+--------------+
```

Likwid-perfctr

```
(...)
--------------------------------------------------------------------------------
DGEMM with options: blocked packed vector
Time/FLOPS with timeofday(): 2.770782e+10
--------------------------------------------------------------------------------
Group 1: MEM
+-----------------------+---------+-------------+
|         Event         | Counter |   Core 1    |
+-----------------------+---------+-------------+
|   INSTR_RETIRED_ANY   |  FIXC0  | 38479616403 |
| CPU_CLK_UNHALTED_CORE |  FIXC1  | 19245252158 |
|  CPU_CLK_UNHALTED_REF |  FIXC2  | 15179510632 |

(...)
+-----------------------------------+--------------+
|              Metric               |    Core 1    |
+-----------------------------------+--------------+
(...)
|  Memory read bandwidth [MBytes/s] | 5.727931e+03 |
|  Memory read data volume [GBytes] | 34.298501696 |
| Memory write bandwidth [MBytes/s] | 3.039028e+02 |
| Memory writo data volume [GBytes] | 1.81975136 |
|    Memory bandwidth [MBytes/s]    | 6.031834e+03 |
|    Memory data volume [GBytes]    | 36.118253056 |
+-----------------------------------+--------------+
```

Things to keep in mind

- Measurements almost always create overhead.
- Check if a metric conforms with your performance model.
- Do not trust measurements blindly.