Fermilab U.S. DEPARTMENT OF ENERGY | Office of Science

Profiling Software with Intel OneAPI Toolset Profilers: Vtune and Advisor

Patrick Gartung

TAC-HEP

28 Feb 2023

Prerequisites

- This Vtune tutorial is based on one provided by Intel, but updated for the newer version of tools available
  https://www.intel.com/content/www/us/en/docs/vtune-profiler/tutorial-common-bottlenecks-linux/2020/overview.html
- This tutorial assumes you have access to a Linux PC or VM and X11 or a VNC client to allow running the Linux gui.
  - The install for MacOS didn't work because there is no current DPC++ compiler
  - The install for Windows was not attempted.
- Follow the links in the Use Case and Prerequisites page to download and install
  - Vtune Profiler
    sh l_oneapi_vtune_p_2023.0.0.25339.sh
  - DPC++ compiler
    sh l_dpcpp-cpp-compiler_p_2023.0.0.25393.sh
- Cmake v3.4+ is also required to generate the makefiles

Getting the code

git clone -b 2021.2.1 https://github.com/oneapi-src/oneAPI-samples.git

```
Cloning into 'oneAPI-samples'...
remote: Enumerating objects: 24289, done.
remote: Counting objects: 100% (735/735), done.
remote: Compressing objects: 100% (408/408), done.
remote: Total 24289 (delta 353), reused 639 (delta 316), pack-reused 23554
Receiving objects: 100% (24289/24289), 256.31 MiB | 61.87 MiB/s, done.
Resolving deltas: 100% (15792/15792), done.
Note: switching to 'cb1440584bb64554d573bf7b03225926c2da3651'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

Updating files: 100% (3353/3353), done.
```

Setting up the environment

source /opt/intel/oneapi/setvars.sh

:: initializing oneAPI environment ...
   -bash: BASH_VERSION = 5.1.8(1)-release
   args: Using "$@" for setvars.sh arguments:
:: compiler -- latest
:: debugger -- latest
:: dev-utilities -- latest
:: tbb -- latest
:: vtune -- latest
:: oneAPI environment initialized ::

Compiling the sample code (first try)

```
cd oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/
cmake .
-- The C compiler identification is GNU 11.3.1
-- The CXX compiler identification is IntelLLVM 2023.0.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /opt/intel/oneapi/compiler/2023.0.0/linux/bin/icpx - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune
[gartung@gartung-desktop matrix_multiply_vtune]$ make
[ 33%] Building CXX object CMakeFiles/matrix.dpcpp.dir/src/matrix.cpp.o
In file included from /home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp:12:
/opt/intel/oneapi/dev-utilities/2021.8.0/include/dpc_common.hpp:14:36: error: use of undeclared identifier 'cl'
static auto exception_handler = [](cl::sycl::exception_list eList) {
                                   ^
1 error generated.
make[2]: *** [CMakeFiles/matrix.dpcpp.dir/build.make:76: CMakeFiles/matrix.dpcpp.dir/src/matrix.cpp.o] Error 1
make[1]: *** [CMakeFiles/Makefile2:110: CMakeFiles/matrix.dpcpp.dir/all] Error 2
make: *** [Makefile:91: all] Error 2
```

How is cl::sycl::exception_list used and defined?

- **Find other examples of cl::sycl::exception_list in samples**
  cd oneAPI-samples
  git grep cl::sycl::exception_list
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp: auto error_handler = [&] (**cl::sycl::exception_list** exceptions) {
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:static auto exception_handler = [](**cl::sycl::exception_list** exception_list) {
  Tools/Advisor/matrix_multiply_advisor/src/multiply.hpp:auto exception_handler = [](**cl::sycl::exception_list** exceptionList) {
  Tools/VTuneProfiler/matrix_multiply_vtune/src/multiply.hpp:auto exception_handler = [](**cl::sycl::exception_list** exceptionList) {
  common/dpc_common.hpp:static auto exception_handler = [](**cl::sycl::exception_list** eList) {
- **Check what headers are included in each**
  git grep cl::sycl::exception_list | cut -d: -f1 | grep -v ' '| xargs grep include
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp:#include <cstdint>
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp:#include <iostream>
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp:#include <vector>
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp:#include <CL/sycl.hpp>
  Libraries/oneMKL/block_cholesky_decomposition/solve.cpp:#include "oneapi/mkl.hpp"
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <algorithm>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <cstdio>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <cstdlib>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <cstring>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <exception>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include <vector>
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp: #include "CL/sycl.hpp"
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include "vpl/mfxdispatcher.h"
  Libraries/oneVPL/dpcpp-blur/src/dpcpp-blur.cpp:#include "vpl/mfxvideo.h"
  Tools/Advisor/matrix_multiply_advisor/src/multiply.hpp:#include <CL/sycl.hpp>
  Tools/VTuneProfiler/matrix_multiply_vtune/src/multiply.hpp:#include <CL/sycl.hpp>
  common/dpc_common.hpp:#include <stdlib.h>
  common/dpc_common.hpp:#include <exception>
  common/dpc_common.hpp:#include <CL/sycl.hpp>

Fix and build the sample

- Add #include <CL/sycl.hpp> to **Tools/Advisor/matrix_multiply_advisor/src/matrix.cpp** and **Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp**

```
git diff
diff --git a/Tools/Advisor/matrix_multiply_advisor/src/matrix.cpp b/Tools/Advisor/matrix_multiply_advisor/src/matrix.cpp
index 5914031d..54f25696 100644
--- a/Tools/Advisor/matrix_multiply_advisor/src/matrix.cpp
+++ b/Tools/Advisor/matrix_multiply_advisor/src/matrix.cpp
@@ -6,7 +6,7 @@

 #include <malloc.h>
 #include <iostream>
-
+#include <CL/sycl.hpp>
 // dpc_common.hpp can be found in the dev-utilities include folder.
 // e.g., $ONEAPI_ROOT/dev-utilities//include/dpc_common.hpp
 #include "dpc_common.hpp"
diff --git a/Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp b/Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp
index 5914031d..54f25696 100644
--- a/Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp
+++ b/Tools/VTuneProfiler/matrix_multiply_vtune/src/matrix.cpp
@@ -6,7 +6,7 @@

 #include <malloc.h>
 #include <iostream>
-
+#include <CL/sycl.hpp>
 // dpc_common.hpp can be found in the dev-utilities include folder.
 // e.g., $ONEAPI_ROOT/dev-utilities//include/dpc_common.hpp
 #include "dpc_common.hpp"
```

- Compile the sample with the changed header

```
cd oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/
make
Consolidate compiler generated dependencies of target matrix.dpcpp
[ 33%] Building CXX object CMakeFiles/matrix.dpcpp.dir/src/matrix.cpp.o
[ 66%] Building CXX object CMakeFiles/matrix.dpcpp.dir/src/multiply.cpp.o
[100%] Linking CXX executable matrix.dpcpp
[100%] Built target matrix.dpcpp
```

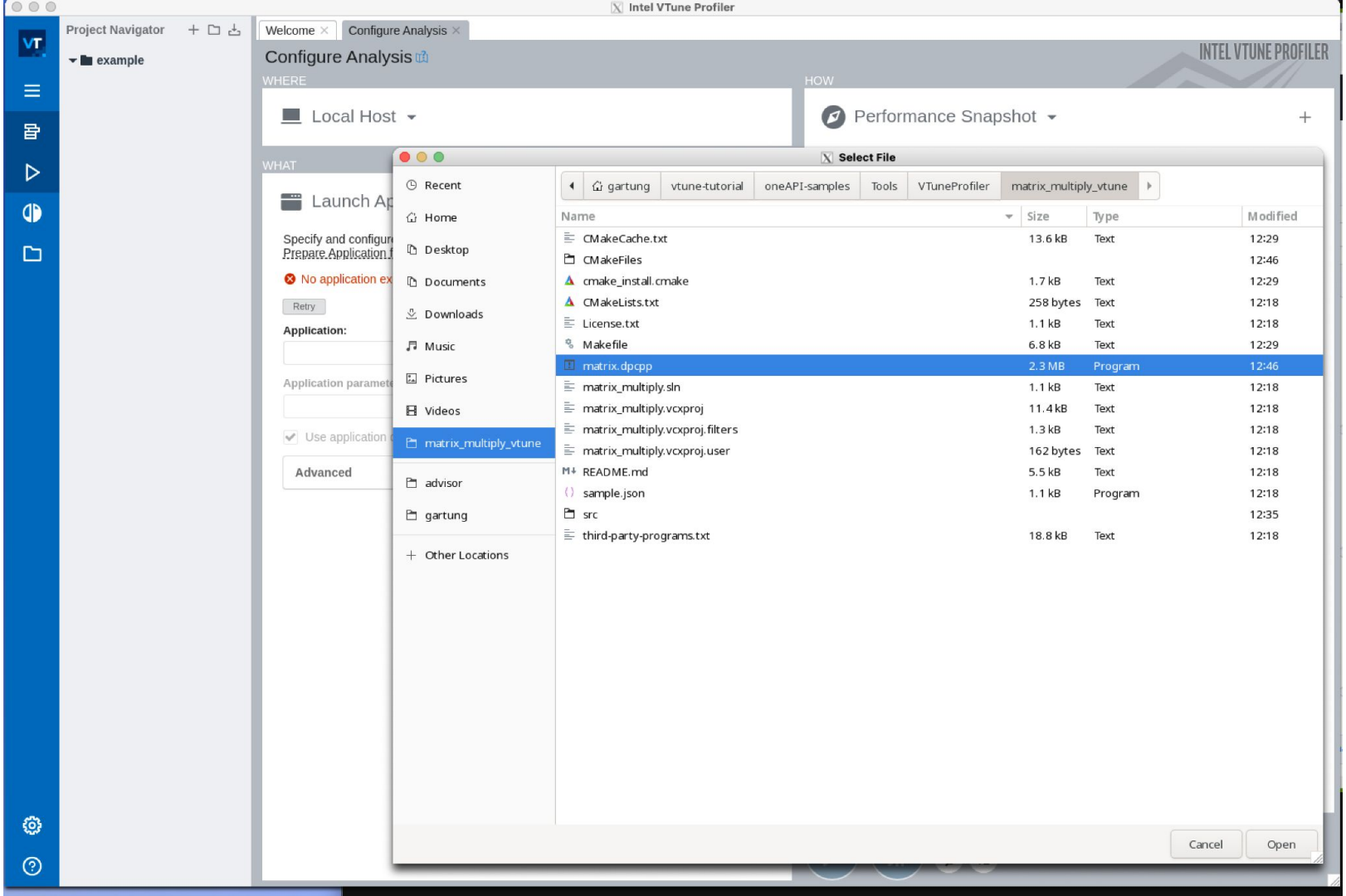
Start Vtune-gui and configure an analysis

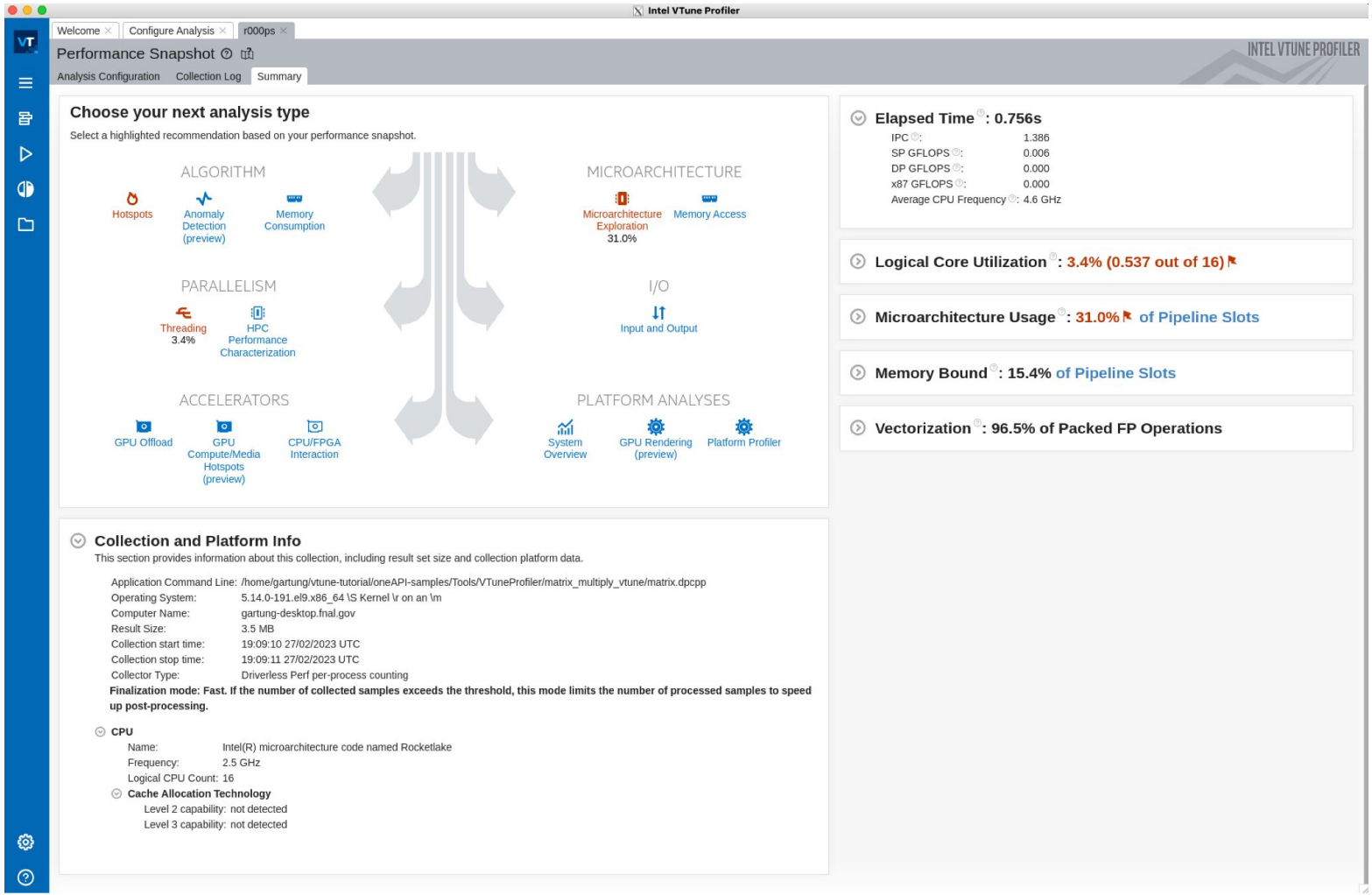
- Start vtune-gui

  vtune-gui

  libva error: vaGetDriverNameByIndex() failed with unknown libva error, driver_name = (null)
  [2798917:0227/125808.021651:ERROR:gpu_memory_buffer_support_x11.cc(44)] dri3 extension not supported.
  [2798884:0227/125808.374982:ERROR:cert_verify_proc_builtin.cc(690)] CertVerifyProcBuiltin for 127.0.0.1 failed:
  ----- Certificate i=0 (CN=gartung-desktop.fnal.gov) -----
  ERROR: No matching issuer found

- Click Menu(stacked bars) ->New->Project
- Name the project, eg tutorial
- Click on the folder icon next to application and navigate to location of sample directory and select maxtrix.dpcpp


Intel VTune Profiler
Project Navigator
example
Welcome
Configure Analysis
Configure Analysis
INTEL VTUNE PROFILER
WHERE
Local Host
HOW
Performance Snapshot
WHAT
Launch Ap
Specify and configur
Prepare Application f
No application ex
Retry
Application:
Application parameter
Use application
Advanced
Select File
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
matrix_multiply_vtune
advisor
gartung
Other Locations
gartung
vtune-tutorial
oneAPI-samples
Tools
VTuneProfiler
matrix_multiply_vtune
Name
Size
Type
Modified
CMakeCache.txt 13.6 kB Text 12:29
CMakeFiles 12:46
cmake_install.cmake 1.7 kB Text 12:29
CMakeLists.txt 258 bytes Text 12:18
License.txt 1.1 kB Text 12:18
Makefile 6.8 kB Text 12:29
matrix.dpcpp 2.3 MB Program 12:46
matrix_multiply.sln 1.1 kB Text 12:18
matrix_multiply.vcxproj 11.4 kB Text 12:18
matrix_multiply.vcxproj.filters 1.3 kB Text 12:18
matrix_multiply.vcxproj.user 162 bytes Text 12:18
README.md 5.5 kB Text 12:18
sample.json 1.1 kB Program 12:18
src 12:35
third-party-programs.txt 18.8 kB Text 12:18
Cancel
Open


Intel VTune Profiler
Welcome
Configure Analysis
r000ps
Performance Snapshot
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Choose your next analysis type
Select a highlighted recommendation based on your performance snapshot.
ALGORITHM
Hotspots
Anomaly Detection (preview)
Memory Consumption
MICROARCHITECTURE
Microarchitecture Exploration
31.0%
Memory Access
PARALLELISM
Threading
3.4%
HPC Performance Characterization
I/O
Input and Output
ACCELERATORS
GPU Offload
GPU Compute/Media Hotspots (preview)
CPU/FPGA Interaction
PLATFORM ANALYSES
System Overview
GPU Rendering (preview)
Platform Profiler
Elapsed Time: 0.756s
IPC: 1.386
SP GFLOPS: 0.006
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
Average CPU Frequency: 4.6 GHz
Logical Core Utilization: 3.4% (0.537 out of 16)
Microarchitecture Usage: 31.0% of Pipeline Slots
Memory Bound: 15.4% of Pipeline Slots
Vectorization: 96.5% of Packed FP Operations
Collection and Platform Info
This section provides information about this collection, including result set size and collection platform data.
Application Command Line: /home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/matrix.dpcpp
Operating System: 5.14.0-191.el9.x86_64 \S Kernel \r on an \m
Computer Name: gartung-desktop.fnal.gov
Result Size: 3.5 MB
Collection start time: 19:09:10 27/02/2023 UTC
Collection stop time: 19:09:11 27/02/2023 UTC
Collector Type: Driverless Perf per-process counting
Finalization mode: Fast. If the number of collected samples exceeds the threshold, this mode limits the number of processed samples to speed up post-processing.
CPU
Name: Intel(R) microarchitecture code named Rocketlake
Frequency: 2.5 GHz
Logical CPU Count: 16
Cache Allocation Technology
Level 2 capability: not detected
Level 3 capability: not detected

Intel VTune Profiler

Welcome | r000ps | r001hs

Hotspots

INTEL VTUNE PROFILER

Analysis Configuration | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree | Flame Graph | Platform

Elapsed Time: 4.380s

- CPU Time: 3.460s
- Total Thread Count: 16
- Paused Time: 0s

Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
sycl::_V1::queue::queue	matrix.dpcpp	0.706s	20.4%
_GLOBAL__sub_I_CommuteCond.cpp	libcommon_clang.so.2022.15.12.0	0.630s	18.2%
__internal_atexit	libc.so.6	0.340s	9.8%
Intel::OpenCL::Utils::OclDynamicLib::Load	libcpu_device.so.2022.15.12.0	0.208s	6.0%
std::basic_string<char, std::char_traits<char>, __cxxabiv1::(anonymous namespace)::malloc_alloc<char>>::_M_append.isra.0	libstlport-dynamic.so	0.170s	4.9%
[Others]	N/A*	1.406s	40.6%

*N/A is applied to non-summable metrics.

Top Tasks

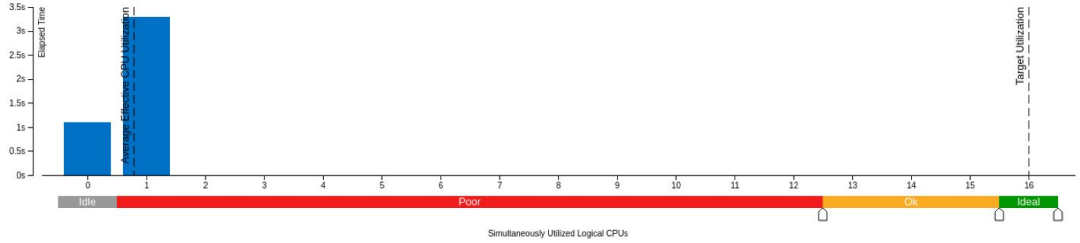
This section lists the most active tasks in your application.

Task Type	Task Time	Task Count	Average Task Time
tbb_parallel_for	0.232s	40	0.006s
tbb_custom	0.032s	6	0.005s

*N/A is applied to non-summable metrics.

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



Collection and Platform Info

Hotspots Insights

If you see significant hotspots in the Top Hotspots list, switch to the Bottom-up view for in-depth analysis per function. Otherwise, use the Caller/Callee or the Flame Graph view to track critical paths for these hotspots.

Explore Additional Insights

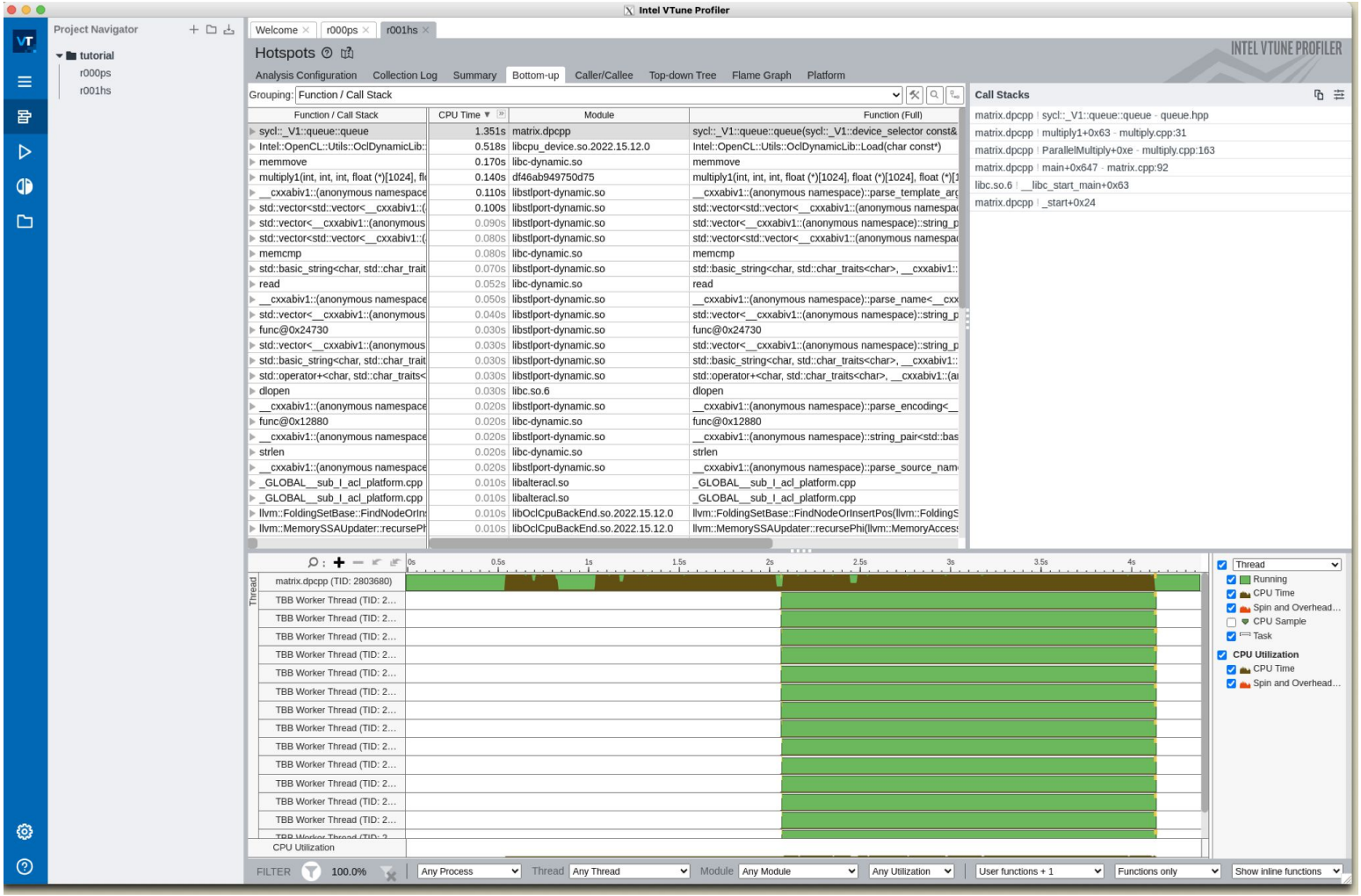
Parallelism: 4.9%

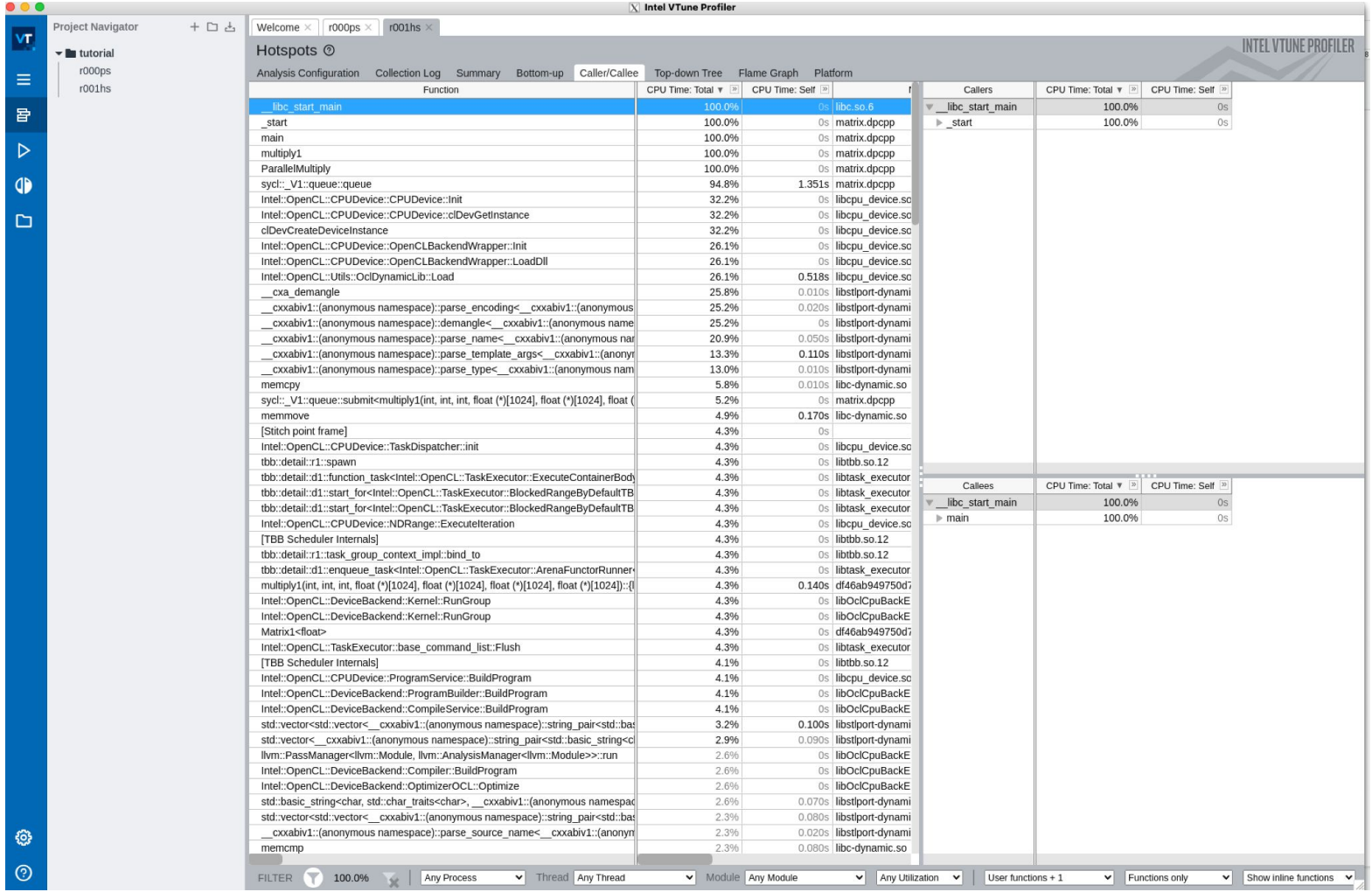
Use Threading to explore more opportunities to increase parallelism in your application.

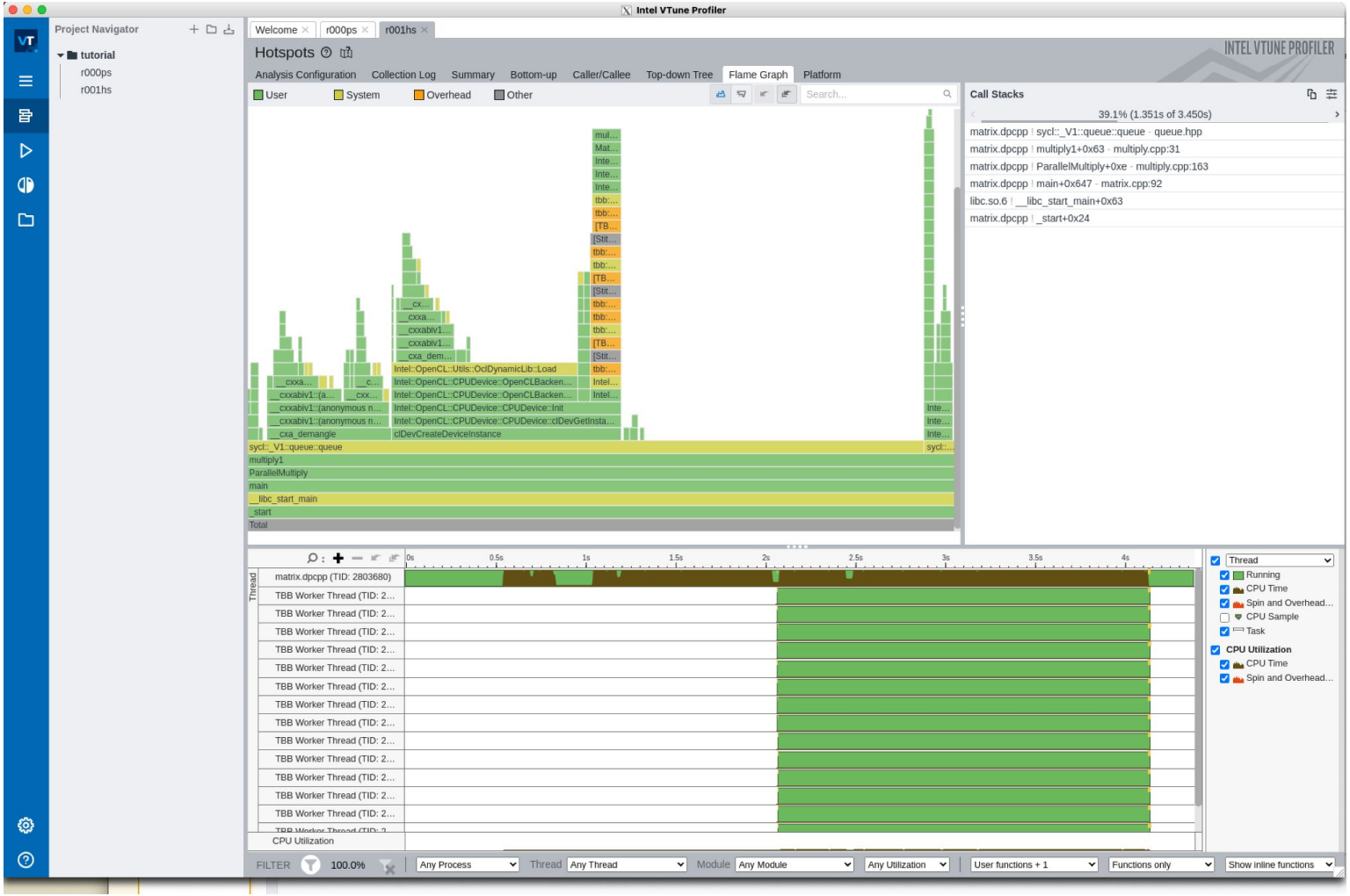
Microarchitecture Usage: 46.7%

Use Microarchitecture Exploration to explore how efficiently your application runs on the used hardware.

INSIGHTS


Intel VTune Profiler
Project Navigator
tutorial
r000ps
r001hs
Welcome
r000ps
r001hs
Hotspots
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Flame Graph
Platform
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Module
Function (Full)
sycl::_V1::queue::queue 1.351s matrix.dpcpp sycl::_V1::queue::queue(sycl::_V1::device_selector const&
Intel::OpenCL::Utils::OclDynamicLib:: 0.518s libcpu_device.so.2022.15.12.0 Intel::OpenCL::Utils::OclDynamicLib::Load(char const*)
memmove 0.170s libc-dynamic.so memmove
multiply1(int, int, int, float (*)[1024], fl 0.140s df46ab949750d75 multiply1(int, int, int, float (*)[1024], float (*)[1024], float (*)[
__cxxabiv1::(anonymous namespace 0.110s libstlport-dynamic.so __cxxabiv1::(anonymous namespace)::parse_template_ar
std::vector<std::vector<__cxxabiv1::(0.100s libstlport-dynamic.so std::vector<std::vector<__cxxabiv1::(anonymous namespa
std::vector<__cxxabiv1::(anonymous 0.090s libstlport-dynamic.so std::vector<__cxxabiv1::(anonymous namespace)::string_p
std::vector<std::vector<__cxxabiv1::(0.080s libstlport-dynamic.so std::vector<std::vector<__cxxabiv1::(anonymous namespa
memcmp 0.080s libc-dynamic.so memcmp
std::basic_string<char, std::char_trait 0.070s libstlport-dynamic.so std::basic_string<char, std::char_traits<char>, __cxxabiv1::
read 0.052s libc-dynamic.so read
__cxxabiv1::(anonymous namespace 0.050s libstlport-dynamic.so __cxxabiv1::(anonymous namespace)::parse_name<__cxx
std::vector<__cxxabiv1::(anonymous 0.040s libstlport-dynamic.so std::vector<__cxxabiv1::(anonymous namespace)::string_p
func@0x24730 0.030s libstlport-dynamic.so func@0x24730
std::vector<__cxxabiv1::(anonymous 0.030s libstlport-dynamic.so std::vector<__cxxabiv1::(anonymous namespace)::string_p
std::basic_string<char, std::char_trait 0.030s libstlport-dynamic.so std::basic_string<char, std::char_traits<char>, __cxxabiv1::
std::operator+<char, std::char_traits< 0.030s libstlport-dynamic.so std::operator+<char, std::char_traits<char>, __cxxabiv1::(a
dlopen 0.030s libc.so.6 dlopen
__cxxabiv1::(anonymous namespace 0.020s libstlport-dynamic.so __cxxabiv1::(anonymous namespace)::parse_encoding<__
func@0x12880 0.020s libc-dynamic.so func@0x12880
__cxxabiv1::(anonymous namespace 0.020s libstlport-dynamic.so __cxxabiv1::(anonymous namespace)::string_pair<std::bas
strlen 0.020s libc-dynamic.so strlen
__cxxabiv1::(anonymous namespace 0.020s libstlport-dynamic.so __cxxabiv1::(anonymous namespace)::parse_source_nam
_GLOBAL__sub_I_acl_platform.cpp 0.010s libalteracl.so _GLOBAL__sub_I_acl_platform.cpp
_GLOBAL__sub_I_acl_platform.cpp 0.010s libalteracl.so _GLOBAL__sub_I_acl_platform.cpp
llvm::FoldingSetBase::FindNodeOrIn 0.010s libOclCpuBackEnd.so.2022.15.12.0 llvm::FoldingSetBase::FindNodeOrInsertPos(llvm::FoldingS
llvm::MemorySSAUpdater::recursePh 0.010s libOclCpuBackEnd.so.2022.15.12.0 llvm::MemorySSAUpdater::recursePhi(llvm::MemoryAcces
Call Stacks
matrix.dpcpp | sycl::_V1::queue::queue - queue.hpp
matrix.dpcpp | multiply1+0x63 - multiply.cpp:31
matrix.dpcpp | ParallelMultiply+0xe - multiply.cpp:163
matrix.dpcpp | main+0x647 - matrix.cpp:92
libc.so.6 | __libc_start_main+0x63
matrix.dpcpp | _start+0x24
0s
0.5s
1s
1.5s
2s
2.5s
3s
3.5s
4s
Thread
matrix.dpcpp (TID: 2803680)
TBB Worker Thread (TID: 2…
CPU Utilization
Thread
Running
CPU Time
Spin and Overhead…
CPU Sample
Task
CPU Utilization
CPU Time
Spin and Overhead…
FILTER
100.0%
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
User functions + 1
Functions only
Show inline functions


Intel VTune Profiler
Project Navigator
tutorial
r000ps
r001hs
Welcome
r000ps
r001hs
Hotspots
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Flame Graph
Platform
Function
CPU Time: Total
CPU Time: Self
__libc_start_main 100.0% 0s libc.so.6
_start 100.0% 0s matrix.dpcpp
main 100.0% 0s matrix.dpcpp
multiply1 100.0% 0s matrix.dpcpp
ParallelMultiply 100.0% 0s matrix.dpcpp
sycl::_V1::queue::queue 94.8% 1.351s matrix.dpcpp
Intel::OpenCL::CPUDevice::CPUDevice::Init 32.2% 0s libcpu_device.sc
Intel::OpenCL::CPUDevice::CPUDevice::clDevGetInstance 32.2% 0s libcpu_device.sc
clDevCreateDeviceInstance 32.2% 0s libcpu_device.sc
Intel::OpenCL::CPUDevice::OpenCLBackendWrapper::Init 26.1% 0s libcpu_device.sc
Intel::OpenCL::CPUDevice::OpenCLBackendWrapper::LoadDll 26.1% 0s libcpu_device.sc
Intel::OpenCL::Utils::OclDynamicLib::Load 26.1% 0.518s libcpu_device.sc
__cxa_demangle 25.8% 0.010s libstlport-dynami
__cxxabiv1::(anonymous namespace)::parse_encoding<__cxxabiv1::(anonymous 25.2% 0.020s libstlport-dynami
__cxxabiv1::(anonymous namespace)::demangle<__cxxabiv1::(anonymous name 25.2% 0s libstlport-dynami
__cxxabiv1::(anonymous namespace)::parse_name<__cxxabiv1::(anonymous nar 20.9% 0.050s libstlport-dynami
__cxxabiv1::(anonymous namespace)::parse_template_args<__cxxabiv1::(anonyr 13.3% 0.110s libstlport-dynami
__cxxabiv1::(anonymous namespace)::parse_type<__cxxabiv1::(anonymous nam 13.0% 0.010s libstlport-dynami
memcpy 5.8% 0.010s libc-dynamic.so
sycl::_V1::queue::submit<multiply1(int, int, int, float (*)[1024], float (*)[1024], float (5.2% 0s matrix.dpcpp
memmove 4.9% 0.170s libc-dynamic.so
[Stitch point frame] 4.3% 0s
Intel::OpenCL::CPUDevice::TaskDispatcher::init 4.3% 0s libcpu_device.sc
tbb::detail::r1::spawn 4.3% 0s libtbb.so.12
tbb::detail::d1::function_task<Intel::OpenCL::TaskExecutor::ExecuteContainerBody 4.3% 0s libtask_executor
tbb::detail::d1::start_for<Intel::OpenCL::TaskExecutor::BlockedRangeByDefaultTB 4.3% 0s libtask_executor
tbb::detail::d1::start_for<Intel::OpenCL::TaskExecutor::BlockedRangeByDefaultTB 4.3% 0s libtask_executor
Intel::OpenCL::CPUDevice::NDRange::ExecuteIteration 4.3% 0s libcpu_device.sc
[TBB Scheduler Internals] 4.3% 0s libtbb.so.12
tbb::detail::r1::task_group_context_impl::bind_to 4.3% 0s libtbb.so.12
tbb::detail::d1::enqueue_task<Intel::OpenCL::TaskExecutor::ArenaFunctorRunner 4.3% 0s libtask_executor
multiply1(int, int, int, float (*)[1024], float (*)[1024], float (*)[1024])::{l 4.3% 0.140s df46ab949750d7
Intel::OpenCL::DeviceBackend::Kernel::RunGroup 4.3% 0s libOclCpuBackE
Intel::OpenCL::DeviceBackend::Kernel::RunGroup 4.3% 0s libOclCpuBackE
Matrix1<float> 4.3% 0s df46ab949750d7
Intel::OpenCL::TaskExecutor::base_command_list::Flush 4.3% 0s libtask_executor
[TBB Scheduler Internals] 4.1% 0s libtbb.so.12
Intel::OpenCL::CPUDevice::ProgramService::BuildProgram 4.1% 0s libcpu_device.sc
Intel::OpenCL::DeviceBackend::ProgramBuilder::BuildProgram 4.1% 0s libOclCpuBackE
Intel::OpenCL::DeviceBackend::CompileService::BuildProgram 4.1% 0s libOclCpuBackE
std::vector<std::vector<__cxxabiv1::(anonymous namespace)::string_pair<std::bas 3.2% 0.100s libstlport-dynami
std::vector<__cxxabiv1::(anonymous namespace)::string_pair<std::basic_string<cl 2.9% 0.090s libstlport-dynami
llvm::PassManager<llvm::Module, llvm::AnalysisManager<llvm::Module>>::run 2.6% 0s libOclCpuBackE
Intel::OpenCL::DeviceBackend::Compiler::BuildProgram 2.6% 0s libOclCpuBackE
Intel::OpenCL::DeviceBackend::OptimizerOCL::Optimize 2.6% 0s libOclCpuBackE
std::basic_string<char, std::char_traits<char>, __cxxabiv1::(anonymous namespac 2.6% 0.070s libstlport-dynami
std::vector<std::vector<__cxxabiv1::(anonymous namespace)::string_pair<std::bas 2.3% 0.080s libstlport-dynami
__cxxabiv1::(anonymous namespace)::parse_source_name<__cxxabiv1::(anonym 2.3% 0.020s libstlport-dynami
memcmp 2.3% 0.080s libc-dynamic.so
Callers
CPU Time: Total
CPU Time: Self
__libc_start_main 100.0% 0s
_start 100.0% 0s
Callees
CPU Time: Total
CPU Time: Self
__libc_start_main 100.0% 0s
main 100.0% 0s
FILTER
100.0%
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
User functions + 1
Functions only
Show inline functions


Intel VTune Profiler
Project Navigator
tutorial
r000ps
r001hs
Welcome
r000ps
r001hs
Hotspots
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Flame Graph
Platform
User
System
Overhead
Other
Search...
Call Stacks
39.1% (1.351s of 3.450s)
matrix.dpcpp ! sycl::_V1::queue::queue - queue.hpp
matrix.dpcpp ! multiply1+0x63 - multiply.cpp:31
matrix.dpcpp ! ParallelMultiply+0xe - multiply.cpp:163
matrix.dpcpp ! main+0x647 - matrix.cpp:92
libc.so.6 ! __libc_start_main+0x63
matrix.dpcpp ! _start+0x24
Intel::OpenCL::Utils::OclDynamicLib::Load
Intel::OpenCL::CPUDevice::OpenCLBacken...
Intel::OpenCL::CPUDevice::CPUDevice::Init
Intel::OpenCL::CPUDevice::CPUDevice::clDevGetInsta...
_cxa_demangle
clDevCreateDeviceInstance
sycl::_V1::queue::queue
multiply1
ParallelMultiply
main
__libc_start_main
_start
Total
Thread
matrix.dpcpp (TID: 2803680)
TBB Worker Thread (TID: 2...
CPU Utilization
Running
CPU Time
Spin and Overhead...
CPU Sample
Task
FILTER
100.0%
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
User functions + 1
Functions only
Show inline functions

Intel VTune Profiler

Welcome | r000ps | r001hs | r002tr

Threading

INTEL VTUNE PROFILER

Analysis Configuration | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree | Platform

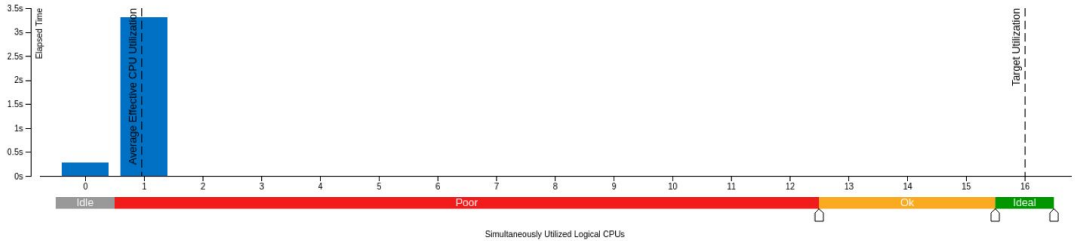
Elapsed Time: 3.583s

Paused Time: 0s

Effective CPU Utilization: 6.0% (0.963 out of 16 logical CPUs)

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



Total Thread Count: 16

Thread Oversubscription: 0s (0.0% of CPU Time)

Wait Time with poor CPU Utilization: 30.846s (100.0% of Wait Time)

Top Waiting Objects

This section lists the objects that spent the most time waiting in your application. Objects can wait on specific calls, such as sleep() or I/O, or on contended synchronizations. A significant amount of Wait time associated with a synchronization object reflects high contention for that object and, thus, reduced parallelism.

Sync Object	Wait Time with poor CPU Utilization	(% from Object Wait Time)	Wait Count
Futex 0xa6a7ecf3	30.764s	100.0%	142
Sleep	0.057s	100.0%	49
Multiple Objects	0.013s	100.0%	6
Stream /proc/cpuinfo 0x2ad8b2c7	0.012s	100.0%	1
Mutex 0x4a3d2244	0.000s	100.0%	6
[Others]	0.001s	100.0%	108

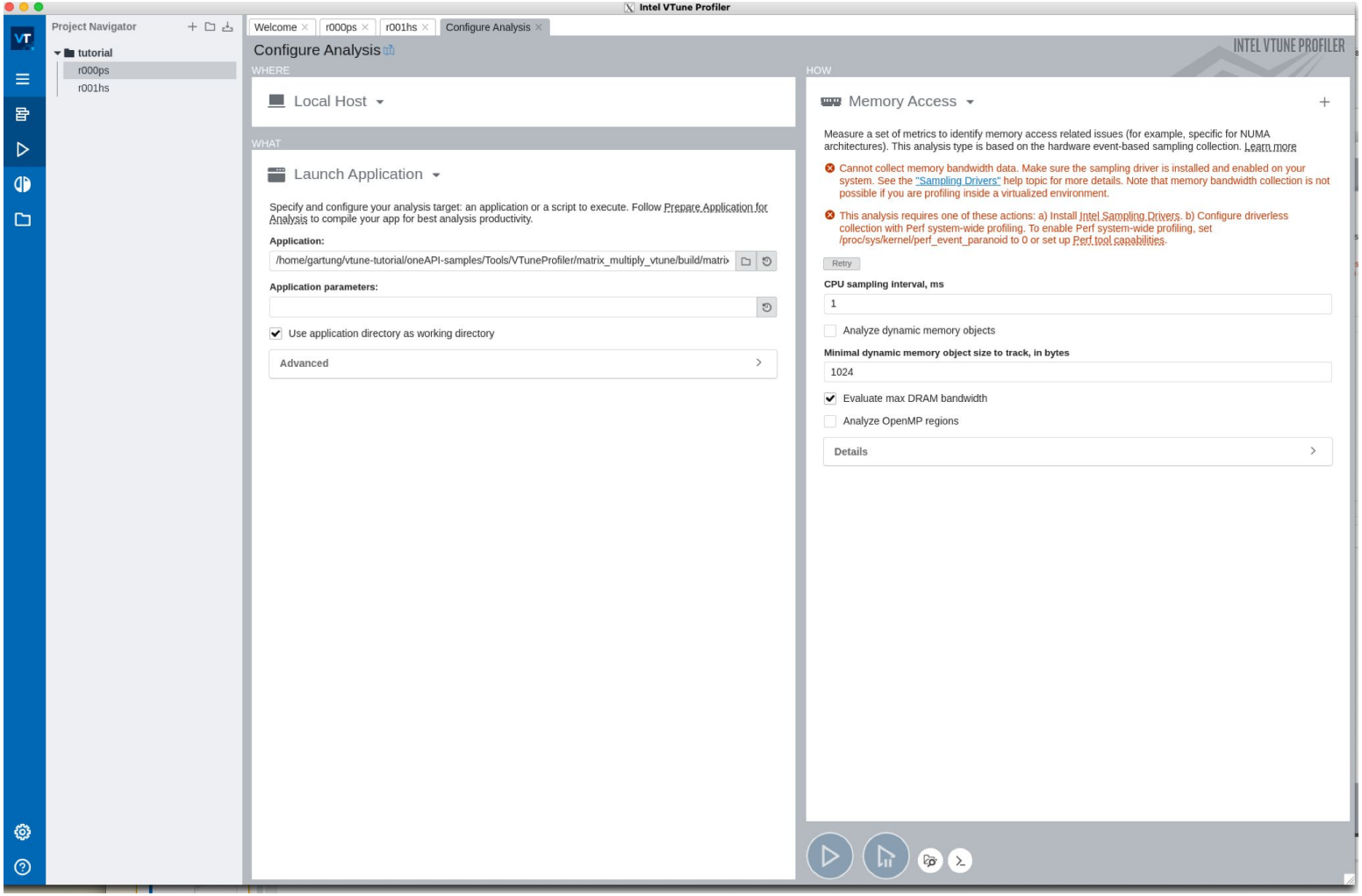
*N/A is applied to non-summable metrics.

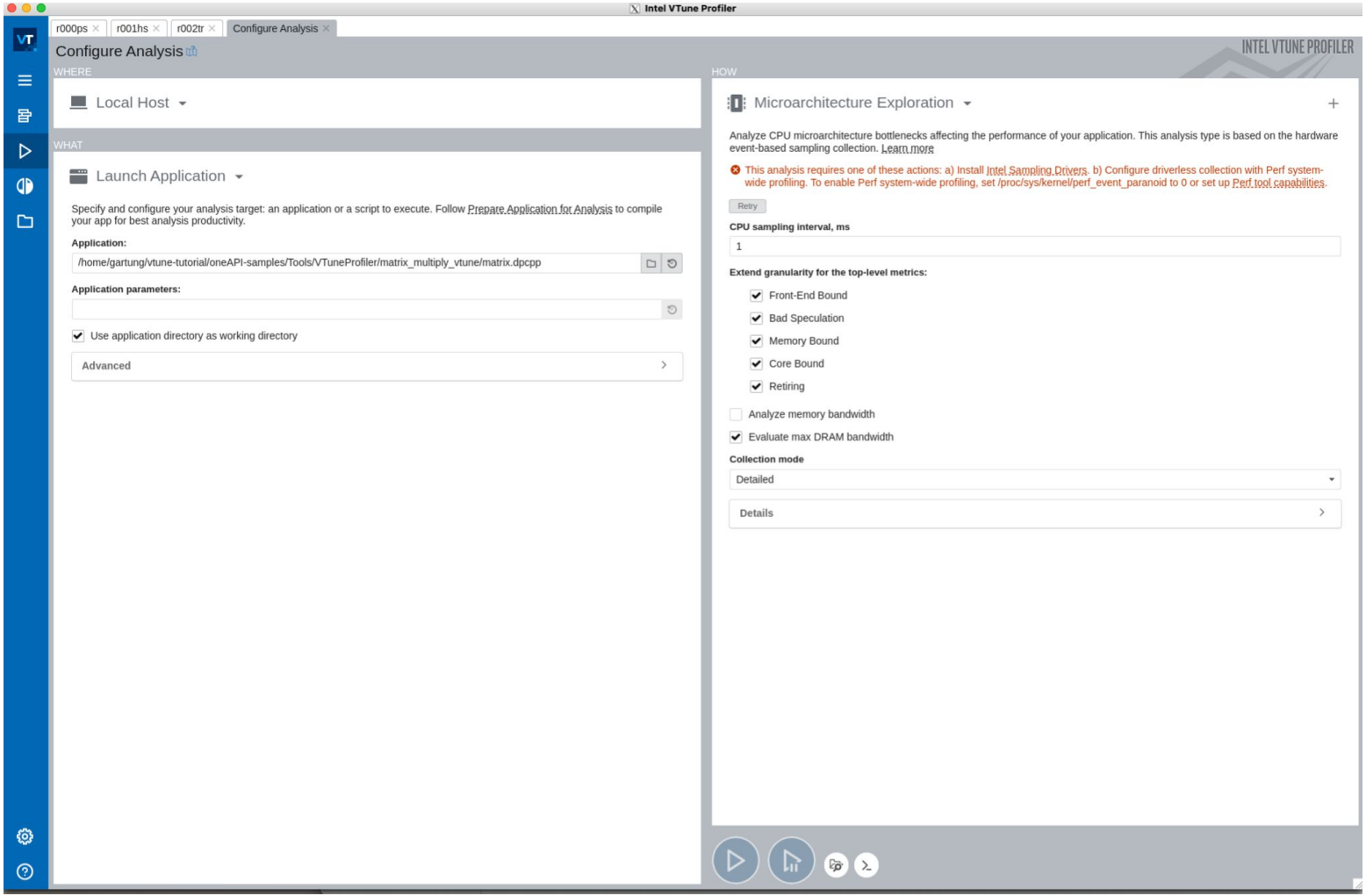
Spin and Overhead Time: 0s (0.0% of CPU Time)

Collection and Platform Info

This section provides information about this collection, including result set size and collection platform data.

Application Command Line:	/home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/matrix.dpcpp
Operating System:	5.14.0-191.el9.x86_64 \S Kernel \r on an \m
Computer Name:	gartung-desktop.fnal.gov
Result Size:	12.2 MB
Collection start time:	19:14:35 27/02/2023 UTC
Collection stop time:	19:14:39 27/02/2023 UTC


Intel VTune Profiler
Project Navigator
tutorial
r000ps
r001hs
Welcome
r000ps
r001hs
Configure Analysis
Configure Analysis
INTEL VTUNE PROFILER
WHERE
Local Host
WHAT
Launch Application
Specify and configure your analysis target: an application or a script to execute. Follow Prepare Application for Analysis to compile your app for best analysis productivity.
Application:
/home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/build/matri
Application parameters:
Use application directory as working directory
Advanced
HOW
Memory Access
Measure a set of metrics to identify memory access related issues (for example, specific for NUMA architectures). This analysis type is based on the hardware event-based sampling collection. Learn more
Cannot collect memory bandwidth data. Make sure the sampling driver is installed and enabled on your system. See the "Sampling Drivers" help topic for more details. Note that memory bandwidth collection is not possible if you are profiling inside a virtualized environment.
This analysis requires one of these actions: a) Install Intel Sampling Drivers. b) Configure driverless collection with Perf system-wide profiling. To enable Perf system-wide profiling, set /proc/sys/kernel/perf_event_paranoid to 0 or set up Perf tool capabilities.
Retry
CPU sampling interval, ms
1
Analyze dynamic memory objects
Minimal dynamic memory object size to track, in bytes
1024
Evaluate max DRAM bandwidth
Analyze OpenMP regions
Details


Intel VTune Profiler
r000ps
r001hs
r002tr
Configure Analysis
Configure Analysis
INTEL VTUNE PROFILER
WHERE
Local Host
WHAT
Launch Application
Specify and configure your analysis target: an application or a script to execute. Follow Prepare Application for Analysis to compile your app for best analysis productivity.
Application:
/home/gartung/vtune-tutorial/oneAPI-samples/Tools/VTuneProfiler/matrix_multiply_vtune/matrix.dpcpp
Application parameters:
Use application directory as working directory
Advanced
HOW
Microarchitecture Exploration
Analyze CPU microarchitecture bottlenecks affecting the performance of your application. This analysis type is based on the hardware event-based sampling collection. Learn more
This analysis requires one of these actions: a) Install Intel Sampling Drivers. b) Configure driverless collection with Perf system-wide profiling. To enable Perf system-wide profiling, set /proc/sys/kernel/perf_event_paranoid to 0 or set up Perf tool capabilities.
Retry
CPU sampling interval, ms
1
Extend granularity for the top-level metrics:
Front-End Bound
Bad Speculation
Memory Bound
Core Bound
Retiring
Analyze memory bandwidth
Evaluate max DRAM bandwidth
Collection mode
Detailed
Details

Install sampling drivers or set kernel parameters

- Directions for installing sampling drivers
- Drivers would not compile on AlmaLinux 9 because of error
  -Werror=implicit-function-declaration
- Set kernel parameters for perf

```
sudo su -
cat /proc/sys/kernel/perf_event_paranoid
2
echo 0 > /proc/sys/kernel/perf_event_paranoid
cat /proc/sys/kernel/kptr_restrict
1
[root@gartung-desktop ~]# echo 0 > /proc/sys/kernel/kptr_restrict
```

Intel VTune Profiler

Project Navigator

- tutorial
  - r000ps
  - r001hs
  - r002ue
  - r003macc

Welcome | r000ps | r001hs | r002ue | r003macc

Microarchitecture Exploration

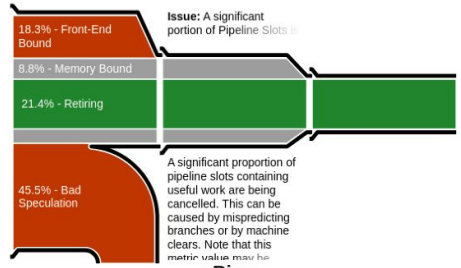
Microarchitecture Exploration

INTEL VTUNE PROFILER

Analysis Configuration | Collection Log | Summary | Bottom-up | Event Count | Platform

Elapsed Time: 0.769s

Clockticks:	3,460,000,000	
Instructions Retired:	5,900,000,000	
CPI Rate:	0.586	
Retiring:	21.4%	of Pipeline Slots
Front-End Bound:	18.3%	of Pipeline Slots
Front-End Latency:	6.8%	of Pipeline Slots
Front-End Bandwidth:	11.5%	of Pipeline Slots
Front-End Bandwidth MITE:	3.3%	of Pipeline Slots
Front-End Bandwidth DSB:	2.8%	of Pipeline Slots
Front-End Bandwidth LSD:	0.0%	of Pipeline Slots
(Info) DSB Coverage:	70.9%	
(Info) LSD Coverage:	4.2%	
(Info) DSB Misses:	100.0%	of Pipeline Slots
Bad Speculation:	45.5%	of Pipeline Slots
Branch Mispredict:	45.5%	of Pipeline Slots
Machine Clears:	0.0%	of Pipeline Slots
Back-End Bound:	14.8%	of Pipeline Slots
Average CPU Frequency:	4.4 GHz	
Total Thread Count:	18	
Paused Time:	0s	



μPipe

This diagram represents inefficiencies in CPU usage. Treat it as a pipe with an output flow equal to the "pipe efficiency" ratio: (Actual Instructions Retired)/(Maximum Possible Instruction Retired). If there are pipeline stalls decreasing the pipe efficiency, the pipe shape gets more narrow.

Top Tasks

This section lists the most active tasks in your application.

Task Type	Task Time	Task Count	Average Task Time
tbb_parallel_for	0.218s	32	0.007s
tbb_custom	0.028s	6	0.005s

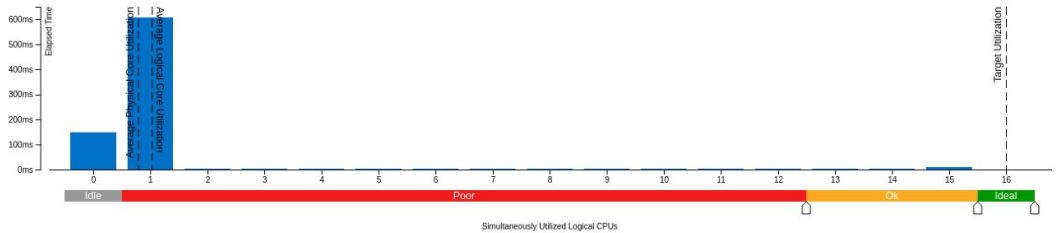
*N/A is applied to non-summable metrics.

Effective Physical Core Utilization: 9.8% (0.787 out of 8)

Effective Logical Core Utilization: 6.4% (1.031 out of 16)

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



Intel VTune Profiler

Project Navigator

tutorial
- r000ps
- r001hs
- r002ue
- r003macc

Welcome | r000ps | r001hs | r002ue | r003macc

Memory Access

Memory Usage

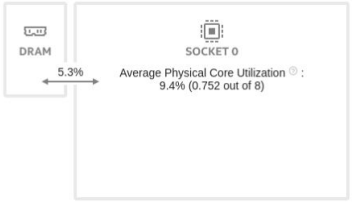
INTEL VTUNE PROFILER

Analysis Configuration | Collection Log | Summary | Bottom-up | Platform

Elapsed Time : 0.788s

CPU Time:	0.801s	
Memory Bound:	7.0%	of Pipeline Slots
Loads:	1,128,929,607	
Stores:	738,537,456	
LLC Miss Count:	0	
Total Thread Count:	18	
Paused Time:	0.026s	

Platform Diagram



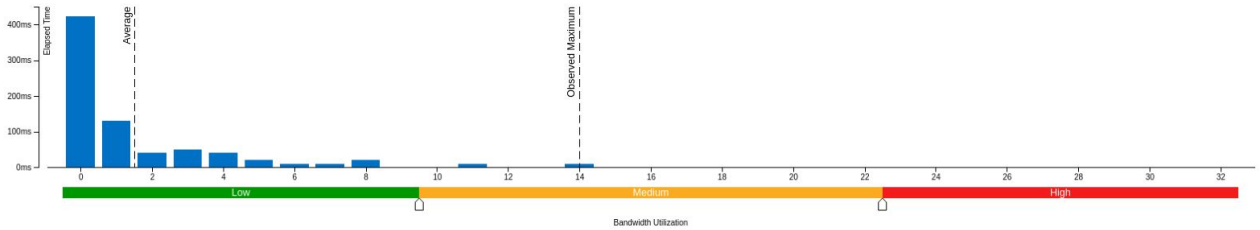
Bandwidth Utilization Histogram

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and Interconnect bandwidth.



Top Functions with High Bandwidth Utilization

This section shows top functions, sorted by LLC Misses that were executing when bandwidth utilization was high for the domain selected in the histogram area.

No data to show. The collected data is not sufficient.

Collecting profiles from the command line

- These example command lines are taken from a document I wrote up for a student to gather profiles on the CMS Reconstruction process and generate text reports.
  - Run vtune command to collect profile of reconstruction job

source /uscms/home/gartung/nobackup/intel/oneapi/setvars.sh
vtune -collect hotspots -r r35234.21 -resume-after=120 -data-limit=0 -knob enable-stack-collection=true -knob stack-size=4096 -knob sampling-mode=sw -- cmsRun step3-35234.21.py 2>&1 | tee step3-35234.21.log

  - Generate a Vtune hotspots report to get the top functions by CPU usage

vtune -report hotspots -r r35234.21 -format=csv -csv-delimiter=semicolon >step3-35234.21.hotspots.csv

  - Generate a Vtune gprof_cc report to get the callgraph of reconstruction

vtune -report gprof-cc -r r35234.21 -format=csv -csv-delimiter=semicolon >step3-35234.21.gprof_cc.csv

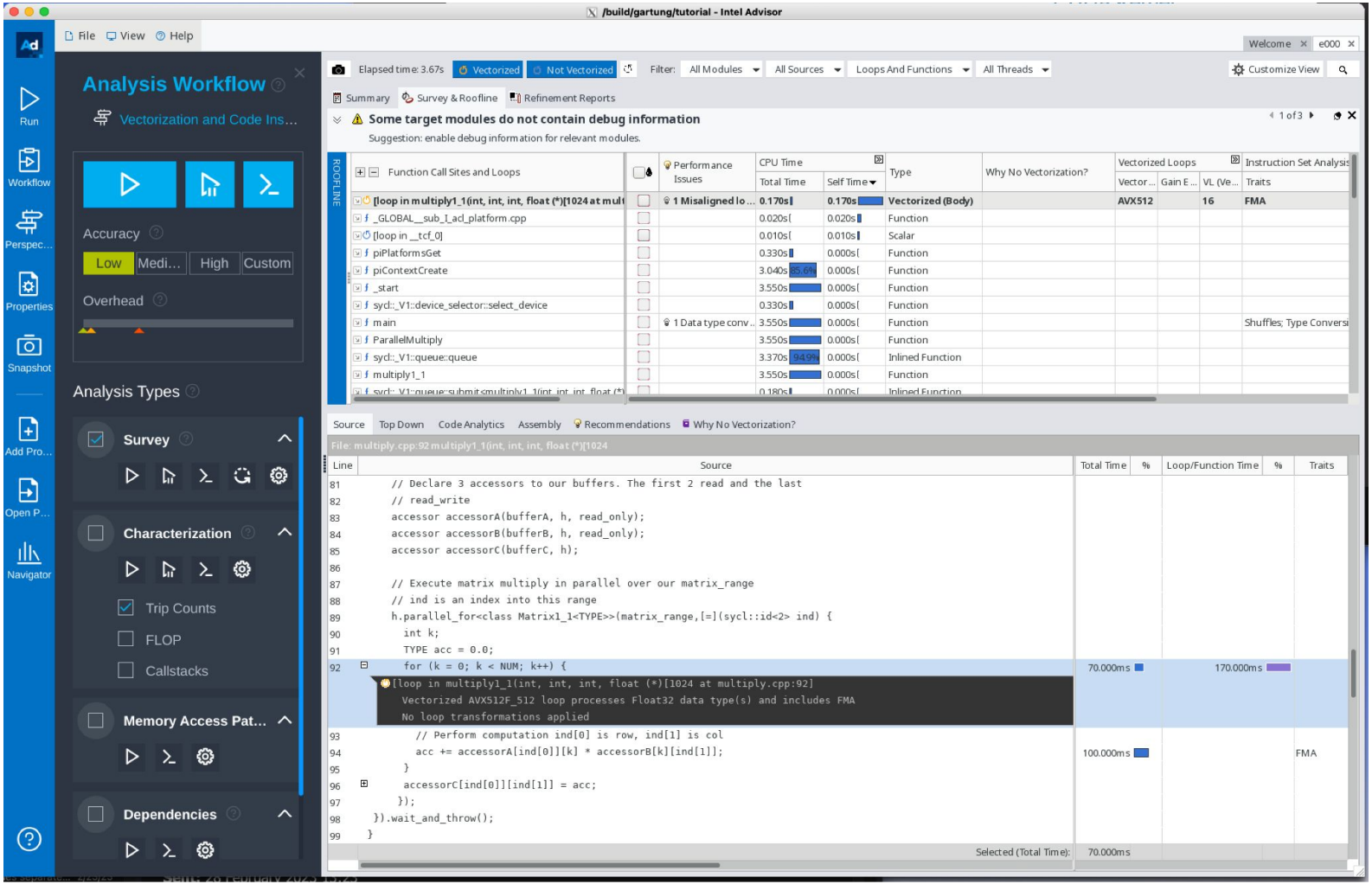
Intel Advisor

- Intel Advisor gives insights into vectorization by identifying loops involving floating point operations that can potentially be vectorized.
- Intel Advisor tutorial
- Download and install Intel Advisor standalone

```
 sudo sh l_oneapi_advisor_p_2023.0.0.25338.sh
Extract l_oneapi_advisor_p_2023.0.0.25338 to /home/gartung/vtune-tutorial/l_oneapi_advisor_p_2023.0.0.25338...
[########################################################################################################################################################]
Extract l_oneapi_advisor_p_2023.0.0.25338 completed!
X11 connection rejected because of wrong authentication.
Could not detect graphical display, installation will continue in console mode. If you aim to launch the installer graphical user interface under root try `xhost si:localuser:root`
command and then restart the application.
Checking system requirements...
Done.
Wait while the installer is preparing...
Done.
Launching the installer...
Remove extracted files: /home/gartung/vtune-tutorial/l_oneapi_advisor_p_2023.0.0.25338…
```

- Run advisor-gui and create a project in a similar way to Vtune example and use the example example generated by the code in

```
cd oneAPI-samples/Tools/Advisor/matrix_multiply_advisor; mkdir build; cd build; cmake ..; make
```


/build/gartung/tutorial - Intel Advisor
File View Help
Welcome e000
Analysis Workflow
Vectorization and Code Ins...
Run
Workflow
Perspec...
Properties
Snapshot
Add Pro...
Open P...
Navigator
Accuracy
Low Medi... High Custom
Overhead
Analysis Types
Survey
Characterization
Trip Counts
FLOP
Callstacks
Memory Access Pat...
Dependencies
Elapsed time: 3.67s
Vectorized
Not Vectorized
Filter: All Modules All Sources Loops And Functions All Threads
Customize View
Summary
Survey & Roofline
Refinement Reports
Some target modules do not contain debug information
Suggestion: enable debug information for relevant modules.
1 of 3
ROOFLINE
Function Call Sites and Loops
Performance Issues
CPU Time
Total Time
Self Time
Type
Why No Vectorization?
Vectorized Loops
Vector ...
Gain E ...
VL (Ve...
Instruction Set Analysis
Traits
[loop in multiply1_1(int, int, int, float (*)[1024 at mul
1 Misaligned lo ...
0.170s
0.170s
Vectorized (Body)
AVX512
16
FMA
_GLOBAL__sub_I_acl_platform.cpp
0.020s
0.020s
Function
[loop in __tcf_0]
0.010s
0.010s
Scalar
piPlatformsGet
0.330s
0.000s
piContextCreate
3.040s
85.6%
0.000s
_start
3.550s
0.000s
sycl::_V1::device_selector::select_device
0.330s
main
1 Data type conv...
3.550s
0.000s
Shuffles; Type Conversi
ParallelMultiply
3.550s
0.000s
sycl::_V1::queue::queue
3.370s
94.9%
0.000s
Inlined Function
multiply1_1
3.550s
0.000s
Source
Top Down
Code Analytics
Assembly
Recommendations
Why No Vectorization?
File: multiply.cpp:92 multiply1_1(int, int, int, float (*)[1024
Line
Source
Total Time
%
Loop/Function Time
%
Traits
81 // Declare 3 accessors to our buffers. The first 2 read and the last
82 // read_write
83 accessor accessorA(bufferA, h, read_only);
84 accessor accessorB(bufferB, h, read_only);
85 accessor accessorC(bufferC, h);
86
87 // Execute matrix multiply in parallel over our matrix_range
88 // ind is an index into this range
89 h.parallel_for<class Matrix1_1<TYPE>>(matrix_range,[=](sycl::id<2> ind) {
90 int k;
91 TYPE acc = 0.0;
92 for (k = 0; k < NUM; k++) {
70.000ms
170.000ms
[loop in multiply1_1(int, int, int, float (*)[1024 at multiply.cpp:92]
Vectorized AVX512F_512 loop processes Float32 data type(s) and includes FMA
No loop transformations applied
93 // Perform computation ind[0] is row, ind[1] is col
94 acc += accessorA[ind[0]][k] * accessorB[k][ind[1]];
100.000ms
FMA
95 }
96 accessorC[ind[0]][ind[1]] = acc;
97 });
98 }).wait_and_throw();
99 }
Selected (Total Time):
70.000ms