Intel Software

NUMA CHARACTERIZATION ON INTEL® XEON® PROCESSORS USING VTUNE® AMPLIFIER™

Michael Steyer

Technical Consulting Engineer

Intel Architecture, Graphics & Software

Analysis Tools

ASPECTS OF HPC/THROUGHPUT APPLICATION PERFORMANCE

Intel Hardware Features

Intel® Omni Path Architecture	HBM	Intel® Optane™ DC persistent memory	Multi-core Intel® Xeon® processor	Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
Distributed memory	**Memory**	**I/O**	**Threading**	**CPU Core**
Message size Rank placement Rank Imbalance RTL Overhead Network Bandwidth	False Sharing Access with strides Latency Bandwidth NUMA	File I/O I/O latency I/O waits System-wide I/O	Threaded/serial ratio Thread Imbalance RTL overhead (scheduling, forking) Synchronization	uArch issues (IPC) Vectorization FPU usage efficiency
Cluster	Node			Core

INTEL PARALLEL STUDIO TOOLS COVERING THE ASPECTS

Intel Hardware Features

Intel® Trace Analyzer and Collector

Intel® VTune™ Amplifier

Intel® Advisor

Intel® Omni-Path Architecture	HBM	Intel® Optane™ DC persistent memory	Multi-core Intel® Xeon® processor	Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
Distributed memory	**Memory**	**I/O**	**Threading**	**CPU Core**
Message size Rank placement Rank Imbalance RTL Overhead Network Bandwidth	False Sharing Access with strides Latency Bandwidth NUMA	File I/O I/O latency I/O waits System-wide I/O	Threaded/serial ratio Thread Imbalance RTL overhead (scheduling, forking) Synchronization	uArch issues (IPC) Vectorization FPU usage efficiency
Cluster	Node			Core

THE LONG & SHORT OF PERFORMANCE ANALYSIS

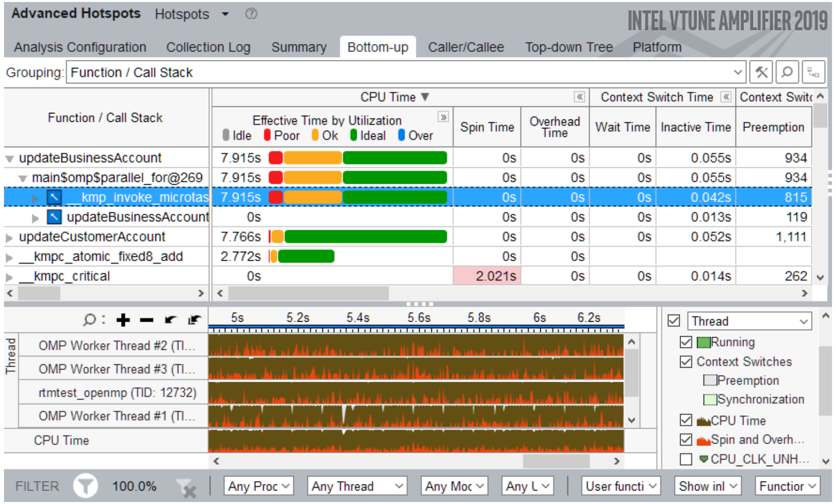
Get the big picture first with a Snapshot or Platform Profiler

	Snapshot Quickly size potential performance gain. Run a test "during a coffee break".	**In-Depth** Advanced collection & analysis. Insight for effective optimization.
Application Focus • HPC App developer focus • 1 app running during test	**Intel® VTune™ Amplifier** **Application Performance Snapshot** L	**Intel® VTune™ Amplifier** • Many profiles S-M **Intel® Advisor** • Vectorization S **Intel® Trace Analyzer and Collector** S-L • MPI Optimization
System Focus • Deployed system focus • Full system load test	**Intel® VTune™ Amplifier's** **Storage Performance Snapshot** L	**Intel® VTune™ Amplifier** **- System-wide sampling** S-M **- Platform Profiler** L

Maximum collection times: L=long (hours) M=medium (minutes) S=short (seconds-few minutes)

Analyze & Tune Application Performance

Intel® VTune™ Amplifier—Performance Profiler



Learn More: software.intel.com/intel-vtune-amplifier-xe

Save Time Optimizing Code

- Accurately profile C, C++, Fortran*, Python*, Go*, Java*, or any mix
- Optimize CPU, threading, memory, cache, storage & more
- Save time: rich analysis leads to insight
- Take advantage of Priority Support
  - Connects customers to Intel engineers for confidential inquiries (paid versions)

What's New in 2019 Release (partial list)

- New Platform Profiler! - Longer Data Collection
- A more accessible user interface provides a simplified profiling workflow
- Smarter, faster Application Performance Snapshot: Analyze CPU utilization of physical cores, pause/resume, more… (Linux*)
- Improved JIT profiling for server-side/cloud applications

Two Great Ways to Collect Data

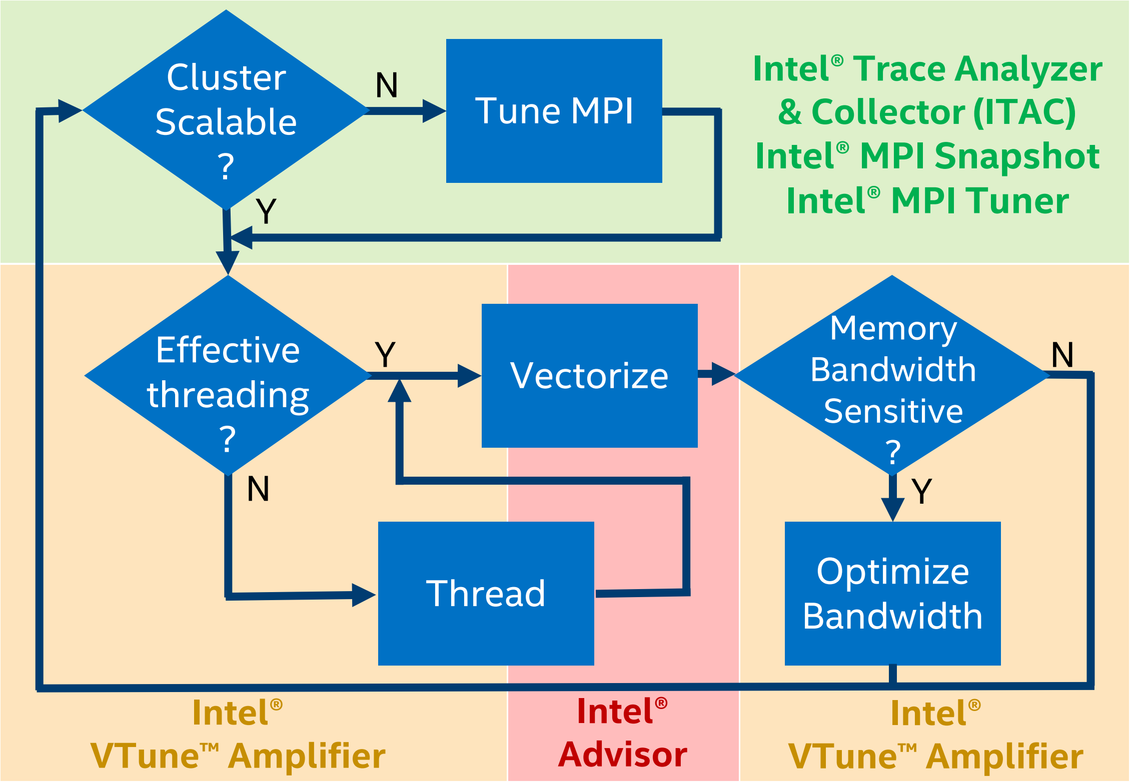
Intel® VTune™ Amplifier

Software Collector	Hardware Collector
Uses OS interrupts	Uses the on chip Performance Monitoring Unit (PMU)
Collects from a single process tree	Collect system wide or from a single process tree.
~5ms default resolution	~1ms default resolution (finer granularity - finds small functions)
Either an Intel® or a compatible processor	Requires a genuine Intel® processor for collection
Call stacks show calling sequence	Optionally collect call stacks
Works in virtual environments	Works in a VM only when supported by the VM (e.g., vSphere*, KVM)
No driver required	Uses Intel driver or perf if driver not installed

No special recompiles - C, C++, C#, Fortran, Java, Python, Assembly

Performance Analysis Tools for Diagnosis

Intel® Parallel Studio XE



HANDS-ON INSTRUCTIONS

Login to RRZE System & Setup

1) $ ssh –L <user> cshpc.rrze.fau.de
2) $ screen
3) $ ssh meggie
4) $ cp /home/hpc/k_m85q/m85q0066/vtune_labs.tar.gz .
5) $ tar -xzvf vtune_labs.tar.gz
6) $ cd vtune_labs
7) $ module load oneapi
8) $ ./compile.sh

Run Benchmark & VTune on the Compute Node

1) $ srun -N 1 -t 90 --reservation=PRACE-day3 -C hwperf --pty /bin/bash -l
2) $ module load oneapi
3) $ export OMP_PLACES=threads
4) $ ./stream.x   #get baseline performance
5) $ ./stream_mod.x   #broken stream performance
6) $ diff ./stream.c ./stream_mod.c   #no, that would be too easy
7) $ vtune -c hotspots -r r_hs_mod -- ./stream_mod.x   #first VTune analysis
8) $ vtune –c ...   #will be discussed in the presentation

Start VTune Backend Server & Connect

1) Open another screen terminal (CTRL & a + c) and navigate to vtune_labs
2) $ source /home/woody/unrz/unrz139/inteloneapi/setvars.sh #no module
3) $ vtune-backend --web-port <UNIQUE PORT> --data-directory .
4) Detach from screen (CTRL & a + d) & logout (exit)
5) Reconect with Tunnel ssh … -L <UNIQUE PORT> :localhost: <UNIQUE PORT>
6) $ screen –x
7) Copy server URL (Serving GUI at) into your browser –> accept certificate
8) Go back to screen terminal 1 (CTRL & a + p) and continue to follow the presentation

STREAM

The Stream Benchmark

John D. McCalpin (TACC)

Arithmetic intensity (>9 for peak DP DRAM)

```
#pragma omp parallel for
        for (j=0; j<STREAM_ARRAY_SIZE; j++)
            c[j] = a[j];
```

Copy

0 Flop / 2 * 8 Bytes = 0

```
#pragma omp parallel for
        for (j=0; j<STREAM_ARRAY_SIZE; j++)
            b[j] = scalar*c[j];
```

Scale

1 Flop / 3 * 8 Bytes = 0.042

```
#pragma omp parallel for
        for (j=0; j<STREAM_ARRAY_SIZE; j++)
            c[j] = a[j]+b[j];
```

Add

1 Flop / 3 * 8 Bytes = 0.042

```
#pragma omp parallel for
        for (j=0; j<STREAM_ARRAY_SIZE; j++)
            a[j] = b[j]+scalar*c[j];
```

Triad

2 Flop / 4 * 8 Bytes = 0.0625

Note that Stream is reporting the best Bandwidth rate out of 10 iterations per default

Performance Note

Xeon Scalable 2nd Generation 8260

Name: Intel(R) Xeon(R) Processor code named Cascadelake

Frequency: 2.4 GHz

Logical CPU Count: 96

Max DRAM Single-Package Bandwidth: 128.0 GB/s (MHZ * ...)

Performance figures are reported OOB without further optimizations like hugepages

We modified Stream

```
$ icc -qopenmp -DSTREAM_ARRAY_SIZE=1000000000 -mcmodel large ../stream.c
        -O2 -g -xHost -o stream.x
$ icc -qopenmp -DSTREAM_ARRAY_SIZE=1000000000 -mcmodel large ../stream_mod.c
        -O2 -g -xHost -o stream_mod.x
$ export OMP_PLACES=threads
```

STREAM Baseline vs Modified

stream.x

```
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 1000000000 (elements), Offset = 0 (elements)
Memory per array = 7629.4 MiB (= 7.5 GiB).
Total memory required = 22888.2 MiB (= 22.4 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 96
Number of Threads counted = 96
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 80675 microseconds.
   (= 80675 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:          179386.1     0.089317     0.089193     0.089479
Scale:         187742.8     0.085394     0.085223     0.085636
Add:           201301.6     0.119386     0.119224     0.119830
Triad:         200889.1     0.119528     0.119469     0.119599
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

stream_mod.x

```
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 1000000000 (elements), Offset = 0 (elements)
Memory per array = 7629.4 MiB (= 7.5 GiB).
Total memory required = 22888.2 MiB (= 22.4 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 96
Number of Threads counted = 96
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 223591 microseconds.
   (= 223591 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           95325.4     0.205046     0.167846     0.230016
Scale:          84390.4     0.214661     0.189595     0.275163
Add:           107456.1     0.263129     0.223347     0.317161
Triad:         100594.8     0.289895     0.238581     0.347999
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

Problem Investigation

What causes the memory bandwidth drop?

Collecting Hotspots

```
vtune -c hotspots -r r_hs_mod -- ./stream_mod.x
```

Intel VTune Amplifier

r_hs_mod

Hotspots Hotspots by CPU Utilization

INTEL VTUNE AMPLIFIER 2019

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

Elapsed Time: 22.803s

- CPU Time: 1116.020s
  - Effective Time: 979.859s
  - Spin Time: 136.098s
    - Imbalance or Serial Spinning: 123.784s
    - Lock Contention: 0s
    - Other: 12.314s
  - Overhead Time: 0.063s
- Total Thread Count: 96
- Paused Time: 0s

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 view to track critical paths for these hotspots.

Explore Additional Insights

Parallelism: 44.8%

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

Microarchitecture Usage: 6.1%

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

INSIGHTS

Top Hotspots

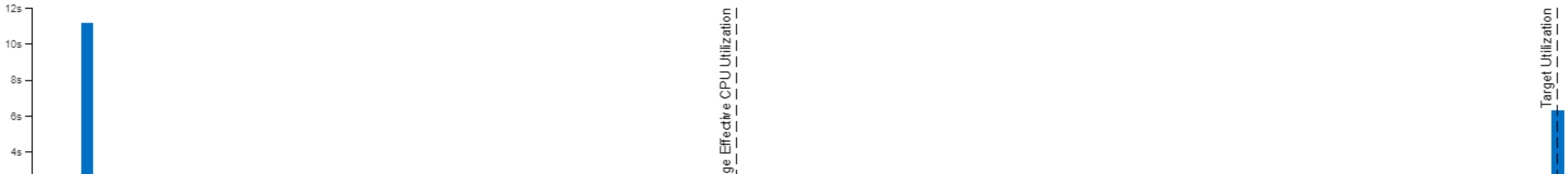
This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

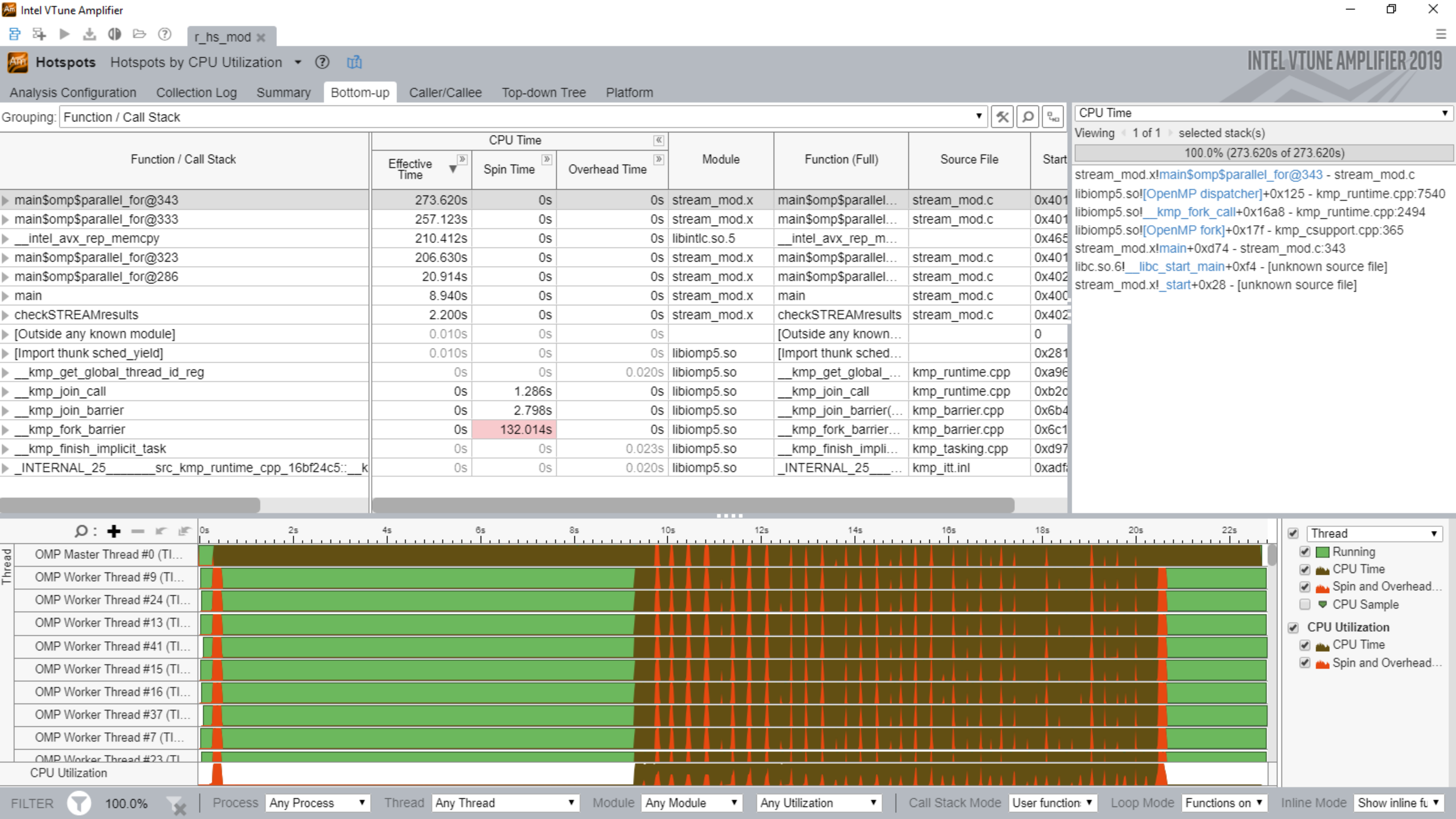
Function	Module	CPU Time
mainompparallel_for@343	stream_mod.x	273.620s
mainompparallel_for@333	stream_mod.x	257.123s
__intel_avx_rep_memcpy	libintlc.so.5	210.412s
mainompparallel_for@323	stream_mod.x	206.630s
__kmp_fork_barrier	libiomp5.so	132.014s
[Others]		36.221s

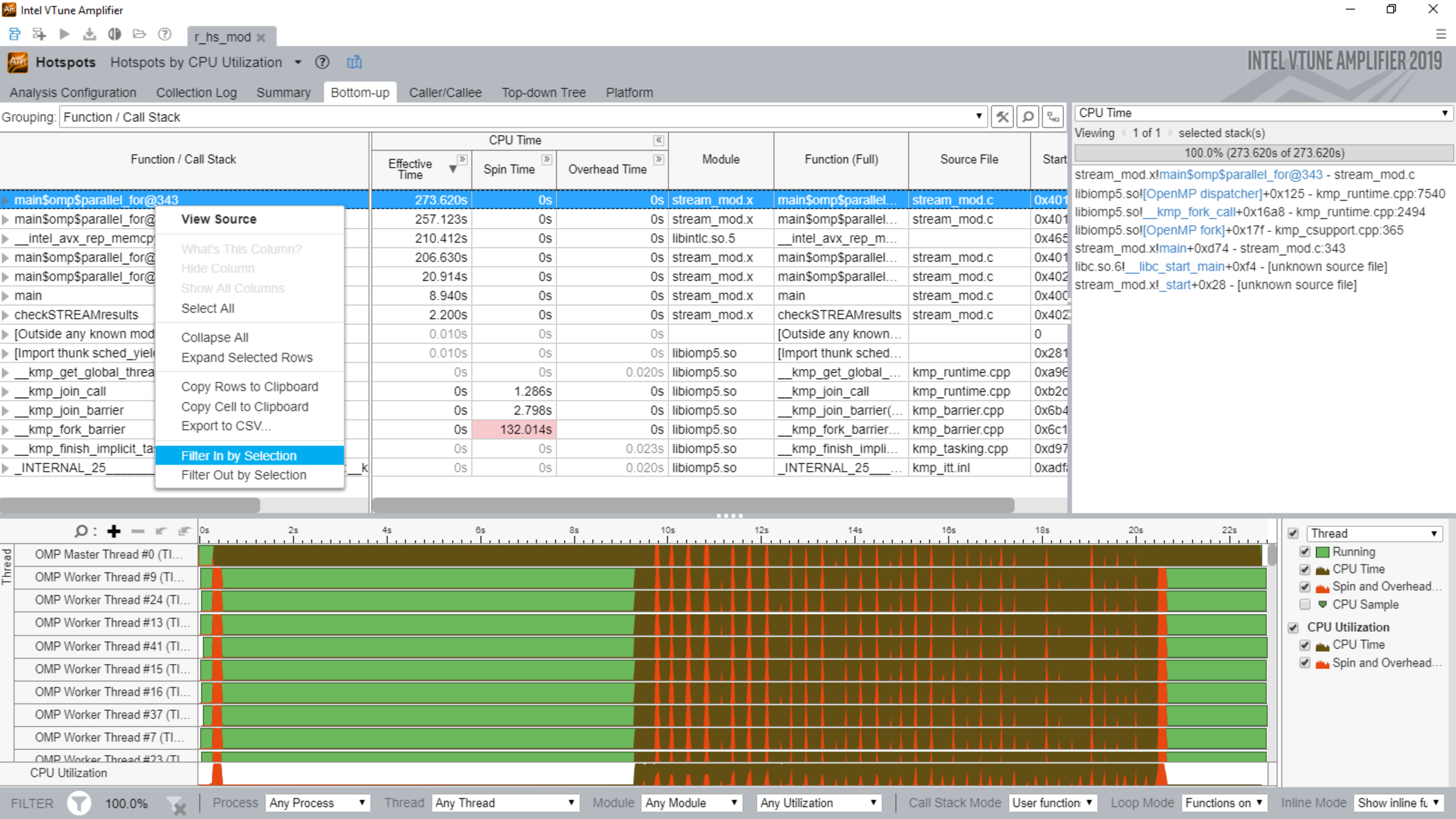
*N/A is applied to non-summable metrics.

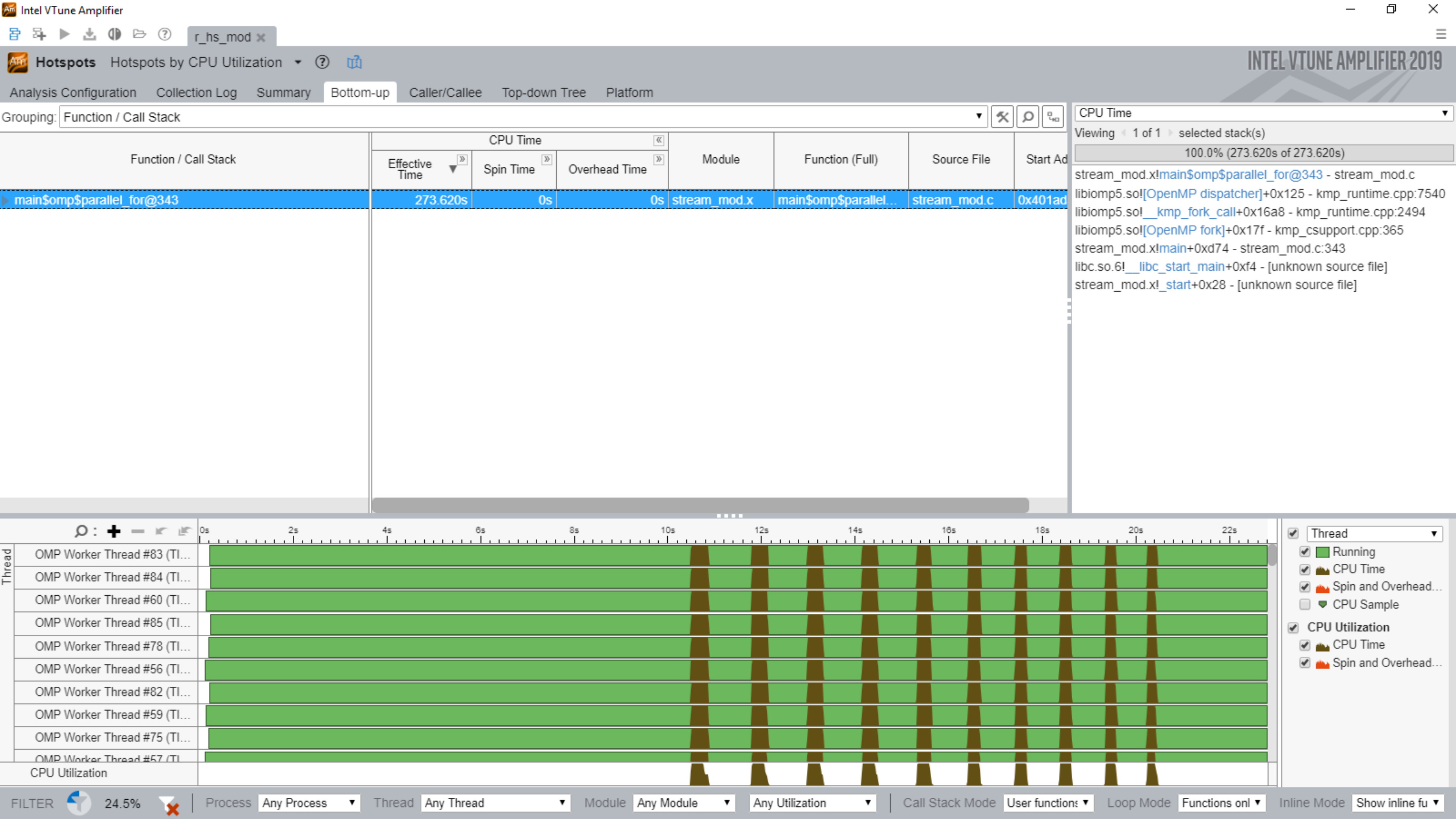
Effective CPU Utilization Histogram

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.

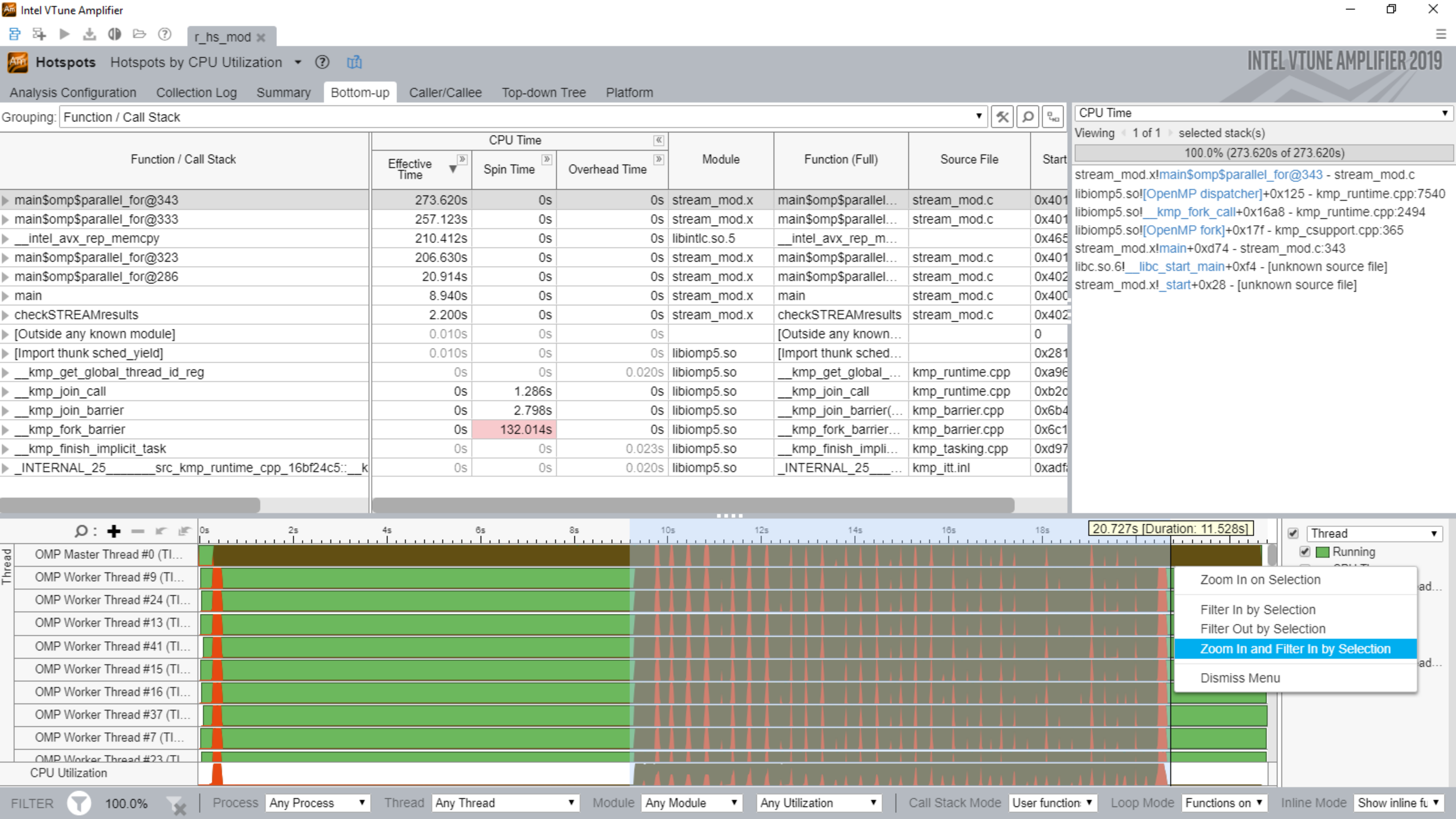



Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Function / Call Stack
CPU Time
Function / Call Stack
Effective Time
Spin Time
Overhead Time
Module
Function (Full)
Source File
Start
mainompparallel_for@343 273.620s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@333 257.123s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
__intel_avx_rep_memcpy 210.412s 0s 0s libintlc.so.5 __intel_avx_rep_m... 0x465
mainompparallel_for@323 206.630s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@286 20.914s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x402
main 8.940s 0s 0s stream_mod.x main stream_mod.c 0x400
checkSTREAMresults 2.200s 0s 0s stream_mod.x checkSTREAMresults stream_mod.c 0x402
[Outside any known module] 0.010s 0s 0s [Outside any known... 0
[Import thunk sched_yield] 0.010s 0s 0s libiomp5.so [Import thunk sched... 0x281
__kmp_get_global_thread_id_reg 0s 0s 0.020s libiomp5.so __kmp_get_global_... kmp_runtime.cpp 0xa96
__kmp_join_call 0s 1.286s 0s libiomp5.so __kmp_join_call kmp_runtime.cpp 0xb2c
__kmp_join_barrier 0s 2.798s 0s libiomp5.so __kmp_join_barrier(... kmp_barrier.cpp 0x6b4
__kmp_fork_barrier 0s 132.014s 0s libiomp5.so __kmp_fork_barrier... kmp_barrier.cpp 0x6c1
__kmp_finish_implicit_task 0s 0s 0.023s libiomp5.so __kmp_finish_impli... kmp_tasking.cpp 0xd97
_INTERNAL_25________src_kmp_runtime_cpp_16bf24c5::__k 0s 0s 0.020s libiomp5.so _INTERNAL_25___... kmp_itt.inl 0xadf
CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (273.620s of 273.620s)
stream_mod.x!mainompparallel_for@343 - stream_mod.c
libiomp5.so![OpenMP dispatcher]+0x125 - kmp_runtime.cpp:7540
libiomp5.so!__kmp_fork_call+0x16a8 - kmp_runtime.cpp:2494
libiomp5.so![OpenMP fork]+0x17f - kmp_csupport.cpp:365
stream_mod.x!main+0xd74 - stream_mod.c:343
libc.so.6!__libc_start_main+0xf4 - [unknown source file]
stream_mod.x!_start+0x28 - [unknown source file]
0s 2s 4s 6s 8s 10s 12s 14s 16s 18s 20s 22s
Thread
OMP Master Thread #0 (TI...
OMP Worker Thread #9 (TI...
OMP Worker Thread #24 (TI...
OMP Worker Thread #13 (TI...
OMP Worker Thread #41 (TI...
OMP Worker Thread #15 (TI...
OMP Worker Thread #16 (TI...
OMP Worker Thread #37 (TI...
OMP Worker Thread #7 (TI...
OMP Worker Thread #23 (TI...
CPU Utilization
Thread
Running
CPU Time
Spin and Overhead...
CPU Sample
CPU Utilization
CPU Time
Spin and Overhead...
FILTER 100.0%
Process Any Process
Thread Any Thread
Module Any Module
Any Utilization
Call Stack Mode User function
Loop Mode Functions on
Inline Mode Show inline fu

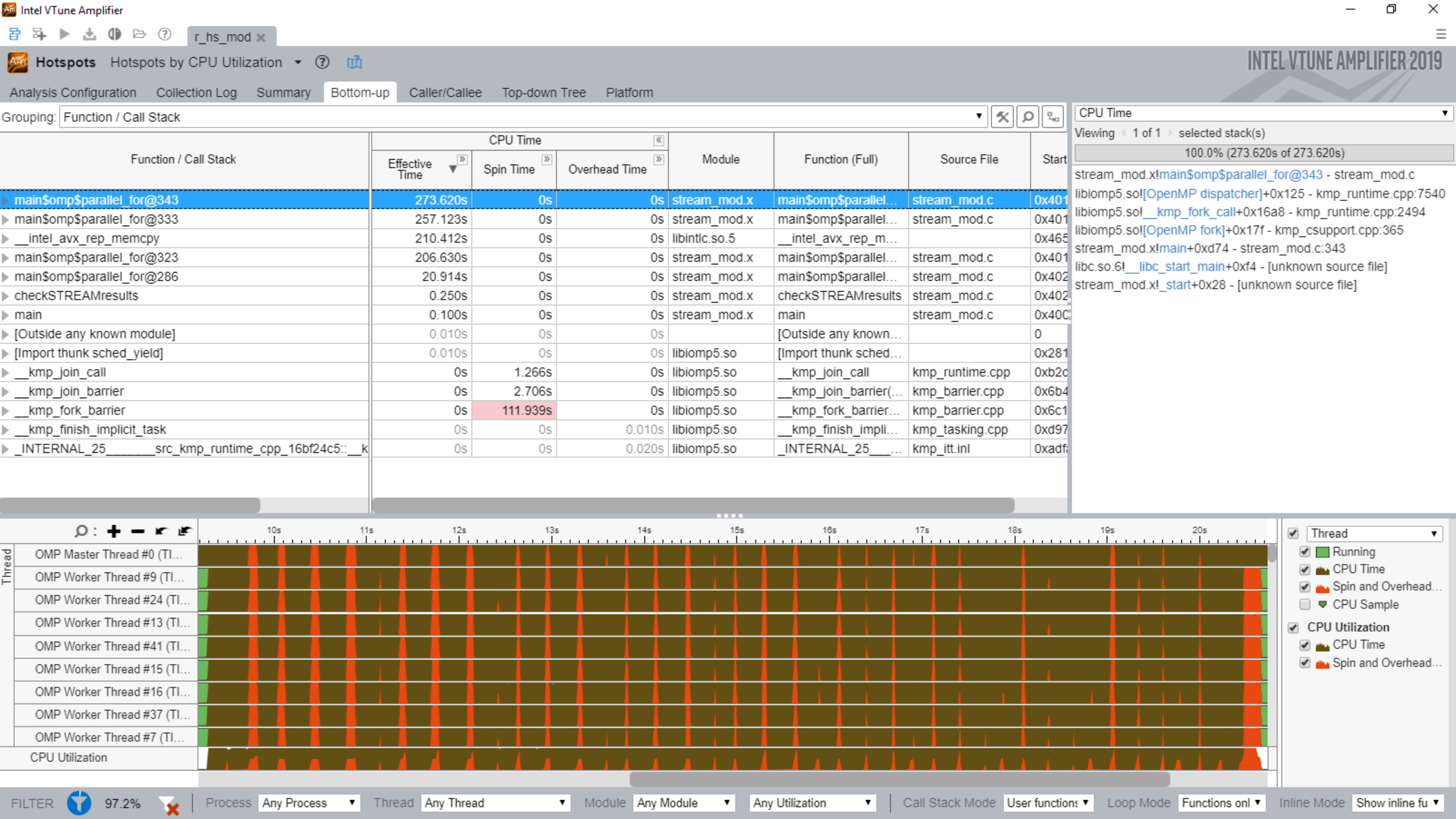

Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time
Spin Time
Overhead Time
Module
Function (Full)
Source File
Start
mainompparallel_for@343 273.620s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@ 257.123s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
__intel_avx_rep_memcpy 210.412s 0s 0s libintlc.so.5 __intel_avx_rep_m... 0x465
mainompparallel_for@ 206.630s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@ 20.914s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x402
main 8.940s 0s 0s stream_mod.x main stream_mod.c 0x400
checkSTREAMresults 2.200s 0s 0s stream_mod.x checkSTREAMresults stream_mod.c 0x402
[Outside any known mod 0.010s 0s 0s [Outside any known... 0
[Import thunk sched_yiel 0.010s 0s 0s libiomp5.so [Import thunk sched... 0x281
__kmp_get_global_threa 0s 0s 0.020s libiomp5.so __kmp_get_global_... kmp_runtime.cpp 0xa96
__kmp_join_call 0s 1.286s 0s libiomp5.so __kmp_join_call kmp_runtime.cpp 0xb2c
__kmp_join_barrier 0s 2.798s 0s libiomp5.so __kmp_join_barrier(... kmp_barrier.cpp 0x6b4
__kmp_fork_barrier 0s 132.014s 0s libiomp5.so __kmp_fork_barrier... kmp_barrier.cpp 0x6c1
__kmp_finish_implicit_ta 0s 0s 0.023s libiomp5.so __kmp_finish_impli... kmp_tasking.cpp 0xd97
_INTERNAL_25 0s 0s 0.020s libiomp5.so _INTERNAL_25___... kmp_itt.inl 0xadf
View Source
What's This Column?
Hide Column
Show All Columns
Select All
Collapse All
Expand Selected Rows
Copy Rows to Clipboard
Copy Cell to Clipboard
Export to CSV...
Filter In by Selection
Filter Out by Selection
CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (273.620s of 273.620s)
stream_mod.x!mainompparallel_for@343 - stream_mod.c
libiomp5.so![OpenMP dispatcher]+0x125 - kmp_runtime.cpp:7540
libiomp5.so!__kmp_fork_call+0x16a8 - kmp_runtime.cpp:2494
libiomp5.so![OpenMP fork]+0x17f - kmp_csupport.cpp:365
stream_mod.x!main+0xd74 - stream_mod.c:343
libc.so.6!__libc_start_main+0xf4 - [unknown source file]
stream_mod.x!_start+0x28 - [unknown source file]
0s 2s 4s 6s 8s 10s 12s 14s 16s 18s 20s 22s
Thread
OMP Master Thread #0 (TI...
OMP Worker Thread #9 (TI...
OMP Worker Thread #24 (TI...
OMP Worker Thread #13 (TI...
OMP Worker Thread #41 (TI...
OMP Worker Thread #15 (TI...
OMP Worker Thread #16 (TI...
OMP Worker Thread #37 (TI...
OMP Worker Thread #7 (TI...
OMP Worker Thread #23 (TI...
CPU Utilization
Thread
Running
CPU Time
Spin and Overhead...
CPU Sample
CPU Utilization
CPU Time
Spin and Overhead...
FILTER 100.0%
Process Any Process
Thread Any Thread
Module Any Module
Any Utilization
Call Stack Mode User function
Loop Mode Functions on
Inline Mode Show inline fu



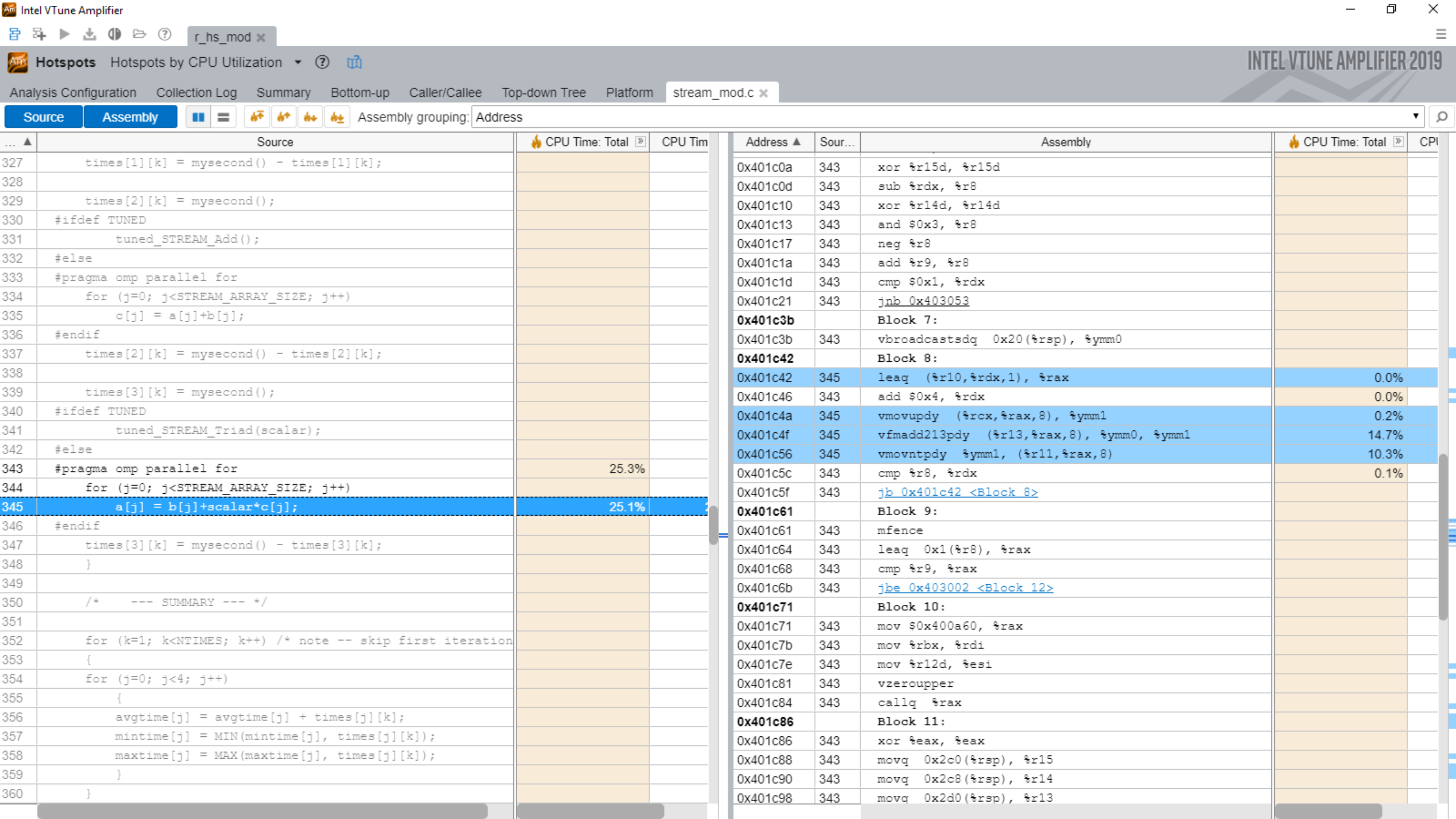
Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time
Spin Time
Overhead Time
Module
Function (Full)
Source File
Start Ad
mainompparallel_for@343
273.620s
0s
0s
stream_mod.x
mainompparallel...
stream_mod.c
0x401ad
CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (273.620s of 273.620s)
stream_mod.x!mainompparallel_for@343 - stream_mod.c
libiomp5.so![OpenMP dispatcher]+0x125 - kmp_runtime.cpp:7540
libiomp5.so!__kmp_fork_call+0x16a8 - kmp_runtime.cpp:2494
libiomp5.so![OpenMP fork]+0x17f - kmp_csupport.cpp:365
stream_mod.x!main+0xd74 - stream_mod.c:343
libc.so.6!__libc_start_main+0xf4 - [unknown source file]
stream_mod.x!_start+0x28 - [unknown source file]
0s
2s
4s
6s
8s
10s
12s
14s
16s
18s
20s
22s
Thread
OMP Worker Thread #83 (TI...
OMP Worker Thread #84 (TI...
OMP Worker Thread #60 (TI...
OMP Worker Thread #85 (TI...
OMP Worker Thread #78 (TI...
OMP Worker Thread #56 (TI...
OMP Worker Thread #82 (TI...
OMP Worker Thread #59 (TI...
OMP Worker Thread #75 (TI...
CPU Utilization
Thread
Running
CPU Time
Spin and Overhead...
CPU Sample
CPU Utilization
CPU Time
Spin and Overhead...
FILTER
24.5%
Process
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
Call Stack Mode
User functions
Loop Mode
Functions onl
Inline Mode
Show inline fu



Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time
Spin Time
Overhead Time
Module
Function (Full)
Source File
Start
mainompparallel_for@343 273.620s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@333 257.123s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
__intel_avx_rep_memcpy 210.412s 0s 0s libintlc.so.5 __intel_avx_rep_m... 0x465
mainompparallel_for@323 206.630s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@286 20.914s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x402
main 8.940s 0s 0s stream_mod.x main stream_mod.c 0x400
checkSTREAMresults 2.200s 0s 0s stream_mod.x checkSTREAMresults stream_mod.c 0x402
[Outside any known module] 0.010s 0s 0s [Outside any known... 0
[Import thunk sched_yield] 0.010s 0s 0s libiomp5.so [Import thunk sched... 0x281
__kmp_get_global_thread_id_reg 0s 0s 0.020s libiomp5.so __kmp_get_global_... kmp_runtime.cpp 0xa96
__kmp_join_call 0s 1.286s 0s libiomp5.so __kmp_join_call kmp_runtime.cpp 0xb2d
__kmp_join_barrier 0s 2.798s 0s libiomp5.so __kmp_join_barrier(... kmp_barrier.cpp 0x6b4
__kmp_fork_barrier 0s 132.014s 0s libiomp5.so __kmp_fork_barrier... kmp_barrier.cpp 0x6c1
__kmp_finish_implicit_task 0s 0s 0.023s libiomp5.so __kmp_finish_impli... kmp_tasking.cpp 0xd97
_INTERNAL_25_______src_kmp_runtime_cpp_16bf24c5::__k 0s 0s 0.020s libiomp5.so _INTERNAL_25___... kmp_itt.inl 0xadf
CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (273.620s of 273.620s)
stream_mod.x!mainompparallel_for@343 - stream_mod.c
libiomp5.so![OpenMP dispatcher]+0x125 - kmp_runtime.cpp:7540
libiomp5.so!__kmp_fork_call+0x16a8 - kmp_runtime.cpp:2494
libiomp5.so![OpenMP fork]+0x17f - kmp_csupport.cpp:365
stream_mod.x!main+0xd74 - stream_mod.c:343
libc.so.6!__libc_start_main+0xf4 - [unknown source file]
stream_mod.x!_start+0x28 - [unknown source file]
0s
2s
4s
6s
8s
10s
12s
14s
16s
18s
20.727s [Duration: 11.528s]
Thread
OMP Master Thread #0 (TI...
OMP Worker Thread #9 (TI...
OMP Worker Thread #24 (TI...
OMP Worker Thread #13 (TI...
OMP Worker Thread #41 (TI...
OMP Worker Thread #15 (TI...
OMP Worker Thread #16 (TI...
OMP Worker Thread #37 (TI...
OMP Worker Thread #7 (TI...
CPU Utilization
Thread
Running
Zoom In on Selection
Filter In by Selection
Filter Out by Selection
Zoom In and Filter In by Selection
Dismiss Menu
FILTER
100.0%
Process
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
Call Stack Mode
User function
Loop Mode
Functions on
Inline Mode
Show inline fu



Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time
Spin Time
Overhead Time
Module
Function (Full)
Source File
Start
mainompparallel_for@343 273.620s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@333 257.123s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
__intel_avx_rep_memcpy 210.412s 0s 0s libintlc.so.5 __intel_avx_rep_m... 0x465
mainompparallel_for@323 206.630s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x401
mainompparallel_for@286 20.914s 0s 0s stream_mod.x mainompparallel... stream_mod.c 0x402
checkSTREAMresults 0.250s 0s 0s stream_mod.x checkSTREAMresults stream_mod.c 0x402
main 0.100s 0s 0s stream_mod.x main stream_mod.c 0x40C
[Outside any known module] 0.010s 0s 0s [Outside any known... 0
[Import thunk sched_yield] 0.010s 0s 0s libiomp5.so [Import thunk sched... 0x281
__kmp_join_call 0s 1.266s 0s libiomp5.so __kmp_join_call kmp_runtime.cpp 0xb2c
__kmp_join_barrier 0s 2.706s 0s libiomp5.so __kmp_join_barrier(... kmp_barrier.cpp 0x6b4
__kmp_fork_barrier 0s 111.939s 0s libiomp5.so __kmp_fork_barrier... kmp_barrier.cpp 0x6c1
__kmp_finish_implicit_task 0s 0s 0.010s libiomp5.so __kmp_finish_impli... kmp_tasking.cpp 0xd97
_INTERNAL_25_______src_kmp_runtime_cpp_16bf24c5::__k 0s 0s 0.020s libiomp5.so _INTERNAL_25___... kmp_itt.inl 0xadf
CPU Time
Viewing 1 of 1 selected stack(s)
100.0% (273.620s of 273.620s)
stream_mod.x!mainompparallel_for@343 - stream_mod.c
libiomp5.so![OpenMP dispatcher]+0x125 - kmp_runtime.cpp:7540
libiomp5.so!__kmp_fork_call+0x16a8 - kmp_runtime.cpp:2494
libiomp5.so![OpenMP fork]+0x17f - kmp_csupport.cpp:365
stream_mod.x!main+0xd74 - stream_mod.c:343
libc.so.6!__libc_start_main+0xf4 - [unknown source file]
stream_mod.x!_start+0x28 - [unknown source file]
10s
11s
12s
13s
14s
15s
16s
17s
18s
19s
20s
Thread
OMP Master Thread #0 (TI...
OMP Worker Thread #9 (TI...
OMP Worker Thread #24 (TI...
OMP Worker Thread #13 (TI...
OMP Worker Thread #41 (TI...
OMP Worker Thread #15 (TI...
OMP Worker Thread #16 (TI...
OMP Worker Thread #37 (TI...
OMP Worker Thread #7 (TI...
CPU Utilization
Thread
Running
CPU Time
Spin and Overhead...
CPU Sample
CPU Utilization
CPU Time
Spin and Overhead...
FILTER
97.2%
Process
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
Call Stack Mode
User functions
Loop Mode
Functions onl
Inline Mode
Show inline fu



Intel VTune Amplifier
r_hs_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
stream_mod.c
Source
Assembly
Assembly grouping: Address
Source
CPU Time: Total
CPU Tim
327 times[1][k] = mysecond() - times[1][k];
328
329 times[2][k] = mysecond();
330 #ifdef TUNED
331 tuned_STREAM_Add();
332 #else
333 #pragma omp parallel for
334 for (j=0; j<STREAM_ARRAY_SIZE; j++)
335 c[j] = a[j]+b[j];
336 #endif
337 times[2][k] = mysecond() - times[2][k];
338
339 times[3][k] = mysecond();
340 #ifdef TUNED
341 tuned_STREAM_Triad(scalar);
342 #else
343 #pragma omp parallel for 25.3%
344 for (j=0; j<STREAM_ARRAY_SIZE; j++)
345 a[j] = b[j]+scalar*c[j]; 25.1%
346 #endif
347 times[3][k] = mysecond() - times[3][k];
348 }
349
350 /* --- SUMMARY --- */
351
352 for (k=1; k<NTIMES; k++) /* note -- skip first iteration
353 {
354 for (j=0; j<4; j++)
355 {
356 avgtime[j] = avgtime[j] + times[j][k];
357 mintime[j] = MIN(mintime[j], times[j][k]);
358 maxtime[j] = MAX(maxtime[j], times[j][k]);
359 }
360 }
Address
Sour...
Assembly
CPU Time: Total
CPI
0x401c0a 343 xor %r15d, %r15d
0x401c0d 343 sub %rdx, %r8
0x401c10 343 xor %r14d, %r14d
0x401c13 343 and $0x3, %r8
0x401c17 343 neg %r8
0x401c1a 343 add %r9, %r8
0x401c1d 343 cmp $0x1, %rdx
0x401c21 343 jnb 0x403053
0x401c3b Block 7:
0x401c3b 343 vbroadcastsdq 0x20(%rsp), %ymm0
0x401c42 Block 8:
0x401c42 345 leaq (%r10,%rdx,1), %rax 0.0%
0x401c46 343 add $0x4, %rdx 0.0%
0x401c4a 345 vmovupdy (%rcx,%rax,8), %ymm1 0.2%
0x401c4f 345 vfmadd213pdy (%r13,%rax,8), %ymm0, %ymm1 14.7%
0x401c56 345 vmovntpdy %ymm1, (%r11,%rax,8) 10.3%
0x401c5c 343 cmp %r8, %rdx 0.1%
0x401c5f 343 jb 0x401c42 <Block 8>
0x401c61 Block 9:
0x401c61 343 mfence
0x401c64 343 leaq 0x1(%r8), %rax
0x401c68 343 cmp %r9, %rax
0x401c6b 343 jbe 0x403002 <Block 12>
0x401c71 Block 10:
0x401c71 343 mov $0x400a60, %rax
0x401c7b 343 mov %rbx, %rdi
0x401c7e 343 mov %r12d, %esi
0x401c81 343 vzeroupper
0x401c84 343 callq %rax
0x401c86 Block 11:
0x401c86 343 xor %eax, %eax
0x401c88 343 movq 0x2c0(%rsp), %r15
0x401c90 343 movq 0x2c8(%rsp), %r14
0x401c98 343 movq 0x2d0(%rsp), %r13

Collecting Hotspots via EBS

```
vtune -c hotspots -knob sampling-mode=hw -r r_hshw_mod -- ./stream_mod.x
```

Intel VTune Amplifier

r_hshw_mod

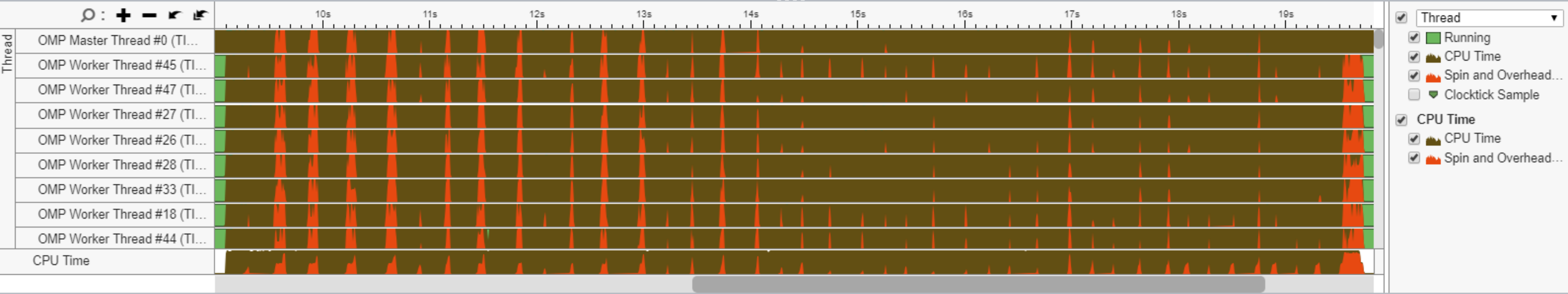
Hotspots Hotspots by CPU Utilization

INTEL VTUNE AMPLIFIER 2019

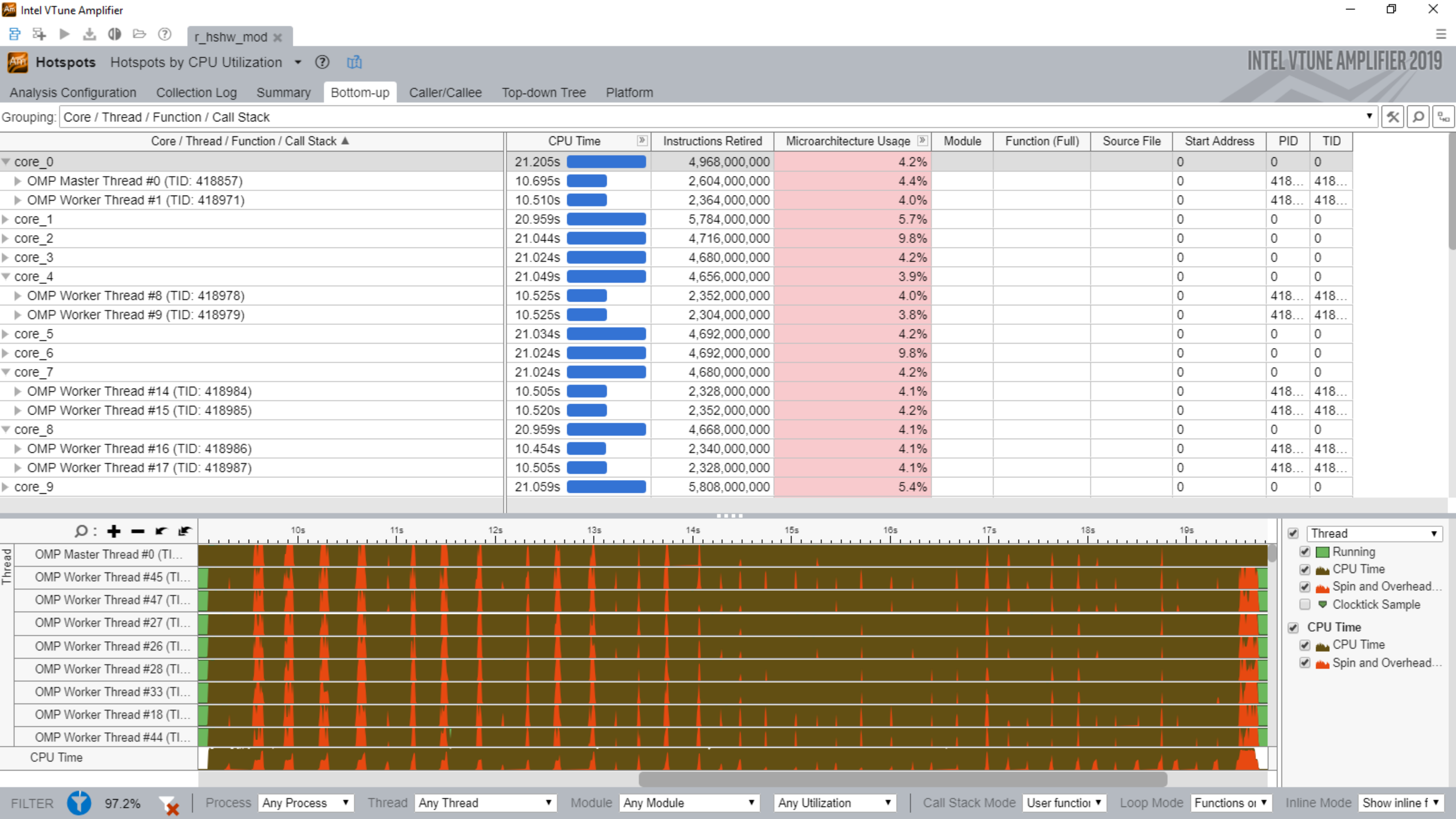
Analysis Configuration | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree | Platform

Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼	Instructions Retired	Microarchitecture Usage	Module	Function (Full)	Source File	Start Address
mainompparallel_for@343	255.238s	17,844,000,000	1.4%	stream_mod.x	mainompparallel...	stream_mod.c	0x401ada
mainompparallel_for@333	230.836s	17,400,000,000	1.6%	stream_mod.x	mainompparallel...	stream_mod.c	0x401cb5
mainompparallel_for@323	186.121s	15,012,000,000	1.7%	stream_mod.x	mainompparallel...	stream_mod.c	0x401e70
__intel_avx_rep_memcpy	185.084s	7,728,000,000	0.8%	libintlc.so.5	__intel_avx_rep_m...		0x46580
_INTERNAL_25_______src_kmp_barrier_cpp_38a91946::__kmp_wait_template<kmp	78.815s	130,584,000,000	44.5%	libiomp5.so	bool _INTERNAL_2...	kmp_wait_release.h	0x628e0
[vmlinux]	48.674s	18,984,000,000	11.5%	vmlinux	[vmlinux]		0
mainompparallel_for@286	17.892s	1,884,000,000	1.8%	stream_mod.x	mainompparallel...	stream_mod.c	0x40218b
_INTERNAL_25_______src_kmp_barrier_cpp_38a91946::__kmp_wait_template<kmp	2.817s	4,476,000,000	45.6%	libiomp5.so	bool _INTERNAL_2...	kmp_wait_release.h	0x63435
kmp_flag_native<unsigned long long>::get	0.922s	8,064,000,000	63.6%	libiomp5.so	kmp_flag_native<u...	kmp_wait_release.h	0x62c3f
sched_yield	0.301s	36,000,000	37.0%	libc-2.17.so	sched_yield		0xe2e00
checkSTREAMresults	0.271s	540,000,000	0.0%	stream_mod.x	checkSTREAMresults	stream_mod.c	0x4029da
main	0.035s	12,000,000	6.2%	stream_mod.x	main	stream_mod.c	0x400c90
__kmp_x86_pause	0.025s	0	0.0%	libiomp5.so	__kmp_x86_pause	kmp.h	0x62b18
kmp_basic_flag_native<unsigned long long>::notdone_check	0.025s	24,000,000	0.0%	libiomp5.so	kmp_basic_flag_nat...	kmp_wait_release.h	0x62c3f
__kmp_finish_implicit_task	0.020s	0	0.0%	libiomp5.so	__kmp_finish_impli...	kmp_tasking.cpp	0xd97a0
_INTERNAL_25_______src_kmp_barrier_cpp_38a91946::__kmp_hyper_barrier_gathe	0.015s	216,000,000	100.0%	libiomp5.so	_INTERNAL_25___...	kmp_barrier.cpp	0x632a0
__kmp_yield	0.015s	0	0.0%	libiomp5.so	__kmp_yield	z_Linux_util.cpp	0xf0ef0
__kmpc_for_static_fini	0.015s	0	0.0%	libiomp5.so	__kmpc_for_static_...	kmp_csupport.cpp	0x70a00



FILTER 97.2% | Process Any Process | Thread Any Thread | Module Any Module | Any Utilization | Call Stack Mode User functio | Loop Mode Functions o | Inline Mode Show inline f


Intel VTune Amplifier
r_hshw_mod
Hotspots
Hotspots by CPU Utilization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Platform
Grouping: Core / Thread / Function / Call Stack

Core / Thread / Function / Call Stack	CPU Time	Instructions Retired	Microarchitecture Usage	Module	Function (Full)	Source File	Start Address	PID	TID
core_0	21.205s	4,968,000,000	4.2%				0	0	0
OMP Master Thread #0 (TID: 418857)	10.695s	2,604,000,000	4.4%				0	418…	418…
OMP Worker Thread #1 (TID: 418971)	10.510s	2,364,000,000	4.0%				0	418…	418…
core_1	20.959s	5,784,000,000	5.7%				0	0	0
core_2	21.044s	4,716,000,000	9.8%				0	0	0
core_3	21.024s	4,680,000,000	4.2%				0	0	0
core_4	21.049s	4,656,000,000	3.9%				0	0	0
OMP Worker Thread #8 (TID: 418978)	10.525s	2,352,000,000	4.0%				0	418…	418…
OMP Worker Thread #9 (TID: 418979)	10.525s	2,304,000,000	3.8%				0	418…	418…
core_5	21.034s	4,692,000,000	4.2%				0	0	0
core_6	21.024s	4,692,000,000	9.8%				0	0	0
core_7	21.024s	4,680,000,000	4.2%				0	0	0
OMP Worker Thread #14 (TID: 418984)	10.505s	2,328,000,000	4.1%				0	418…	418…
OMP Worker Thread #15 (TID: 418985)	10.520s	2,352,000,000	4.2%				0	418…	418…
core_8	20.959s	4,668,000,000	4.1%				0	0	0
OMP Worker Thread #16 (TID: 418986)	10.454s	2,340,000,000	4.1%				0	418…	418…
OMP Worker Thread #17 (TID: 418987)	10.505s	2,328,000,000	4.1%				0	418…	418…
core_9	21.059s	5,808,000,000	5.4%				0	0	0

10s
11s
12s
13s
14s
15s
16s
17s
18s
19s
Thread
OMP Master Thread #0 (TI…
OMP Worker Thread #45 (TI…
OMP Worker Thread #47 (TI…
OMP Worker Thread #27 (TI…
OMP Worker Thread #26 (TI…
OMP Worker Thread #28 (TI…
OMP Worker Thread #33 (TI…
OMP Worker Thread #18 (TI…
OMP Worker Thread #44 (TI…
CPU Time
Thread
Running
CPU Time
Spin and Overhead…
Clocktick Sample
CPU Time
CPU Time
Spin and Overhead…
FILTER
97.2%
Process
Any Process
Thread
Any Thread
Module
Any Module
Any Utilization
Call Stack Mode
User functio
Loop Mode
Functions o
Inline Mode
Show inline f

Collecting HPC Performance

```
vtune -c hpc-performance -r r_hpc_mod -- ./stream_mod.x
```

Intel VTune Amplifier

r_hpc_mod

HPC Performance Characterization HPC Performance Characterization

INTEL VTUNE AMPLIFIER 2019

Analysis Configuration Collection Log Summary Bottom-up

Elapsed Time: 21.772s

SP GFLOPS:	0.000
DP GFLOPS:	0.990
x87 GFLOPS:	0.000
CPI Rate:	9.293
Average CPU Frequency:	2.4 GHz
Total Thread Count:	96

Effective Physical Core Utilization: 44.3% (21.274 out of 48)

Effective Logical Core Utilization: 43.9% (42.101 out of 96)

Serial Time (outside parallel regions): 11.449s (52.6%)

Top Serial Hotspots (outside parallel regions)

This section lists the loops and functions executed serially in the master thread outside of any OpenMP region and consuming the most CPU time. Improve overall application performance by optimizing or parallelizing these hotspot functions. Since the Serial Time metric includes the Wait time of the master thread, it may significantly exceed the aggregated CPU time in the table.

Function	Module	Serial CPU Time
[vmlinux]	vmlinux	4.962s
[Loop at line 268 in main]	stream_mod.x	4.210s
[Loop at line 462 in checkSTREAMresults]	stream_mod.x	2.225s
func@0x7d340	libittnotify_collector.so	0.005s
[sep5]	sep5	0.005s

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

Parallel Region Time: 10.323s (47.4%)

Effective CPU Utilization Histogram

Memory Bound: 86.7% of Pipeline Slots

Cache Bound:	32.5%	of Clockticks
DRAM Bound:	**51.1%**	**of Clockticks**
DRAM Bandwidth Bound:	43.9%	of Elapsed Time
NUMA: % of Remote Accesses:	32.8%	

Bandwidth Utilization Histogram

Vectorization: 100.0% of Packed FP Operations

Intel VTune Amplifier

r_hpc_mod

HPC Performance Characterization HPC Performance Characterization

INTEL VTUNE AMPLIFIER 2019

Analysis Configuration Collection Log Summary Bottom-up

[Loop at line 462 in checkSTREAMresults]	stream_mod.x	2.225s
func@0x7d340	libittnotify_collector.so	0.005s
[sep5]	sep5	0.005s

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

Parallel Region Time: 10.323s (47.4%)

Effective CPU Utilization Histogram

Memory Bound: 86.7% of Pipeline Slots

Cache Bound:	32.5%	of Clockticks
DRAM Bound:	**51.1%**	**of Clockticks**
DRAM Bandwidth Bound:	43.9%	of Elapsed Time
NUMA: % of Remote Accesses:	32.8%	

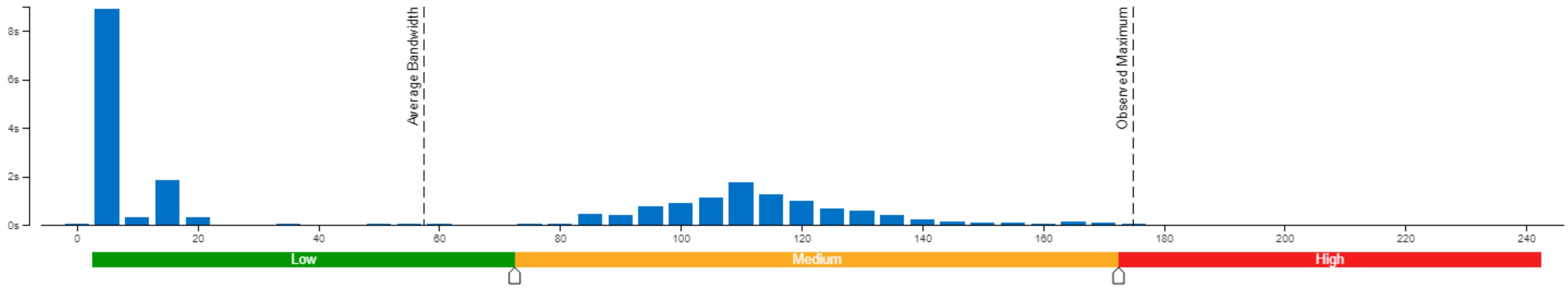
Bandwidth Utilization Histogram

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

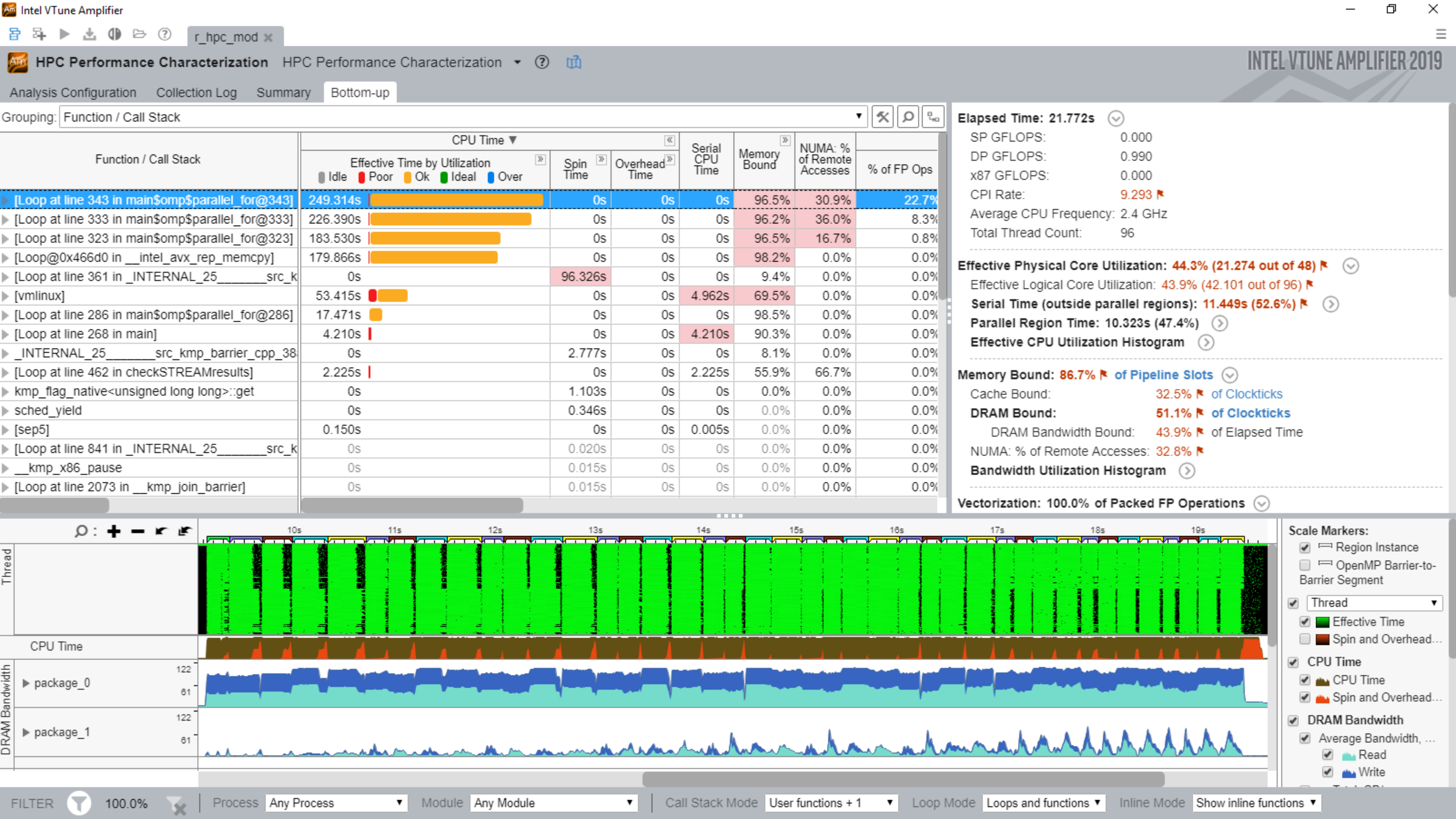
Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

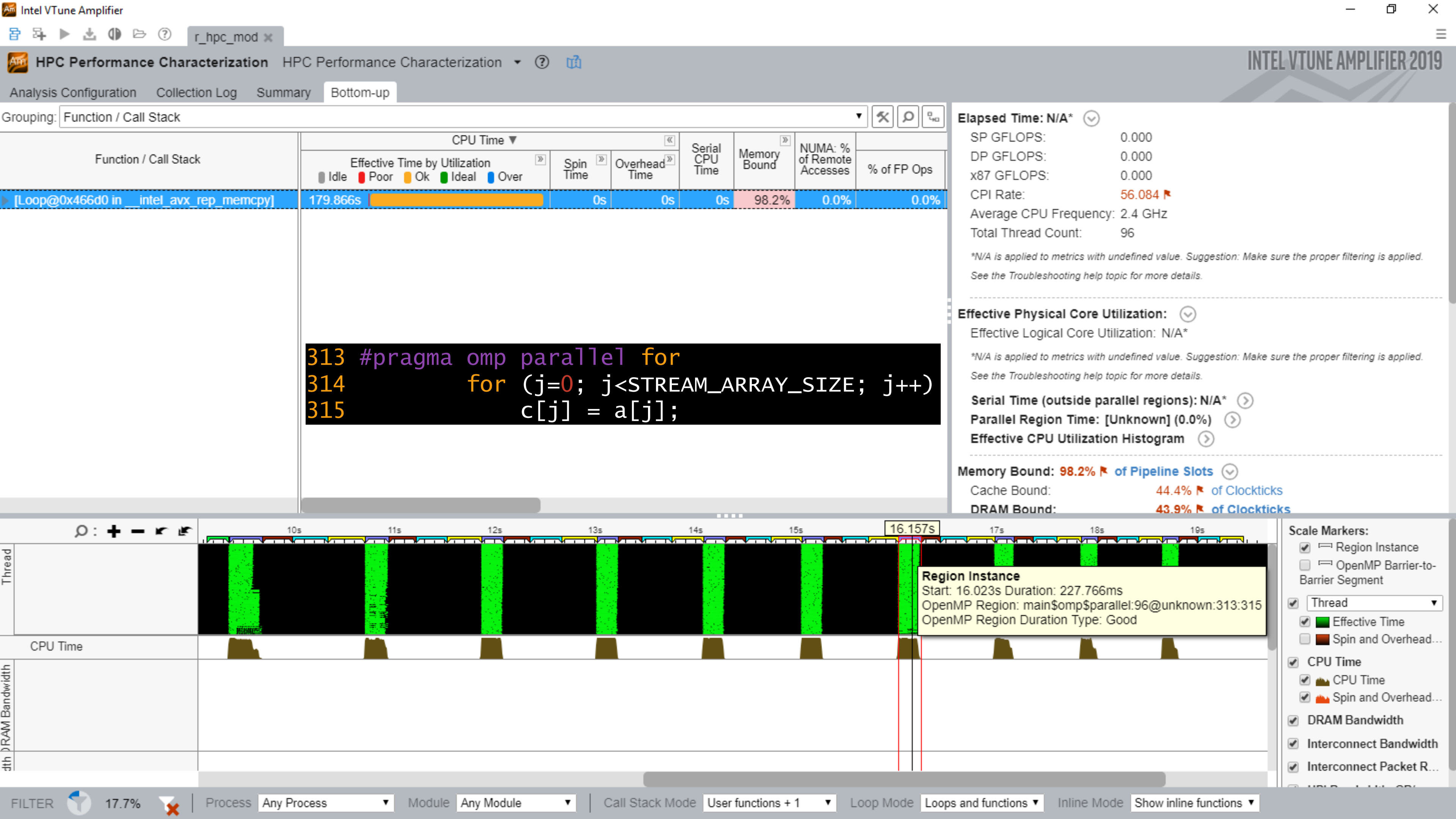
This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed 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.

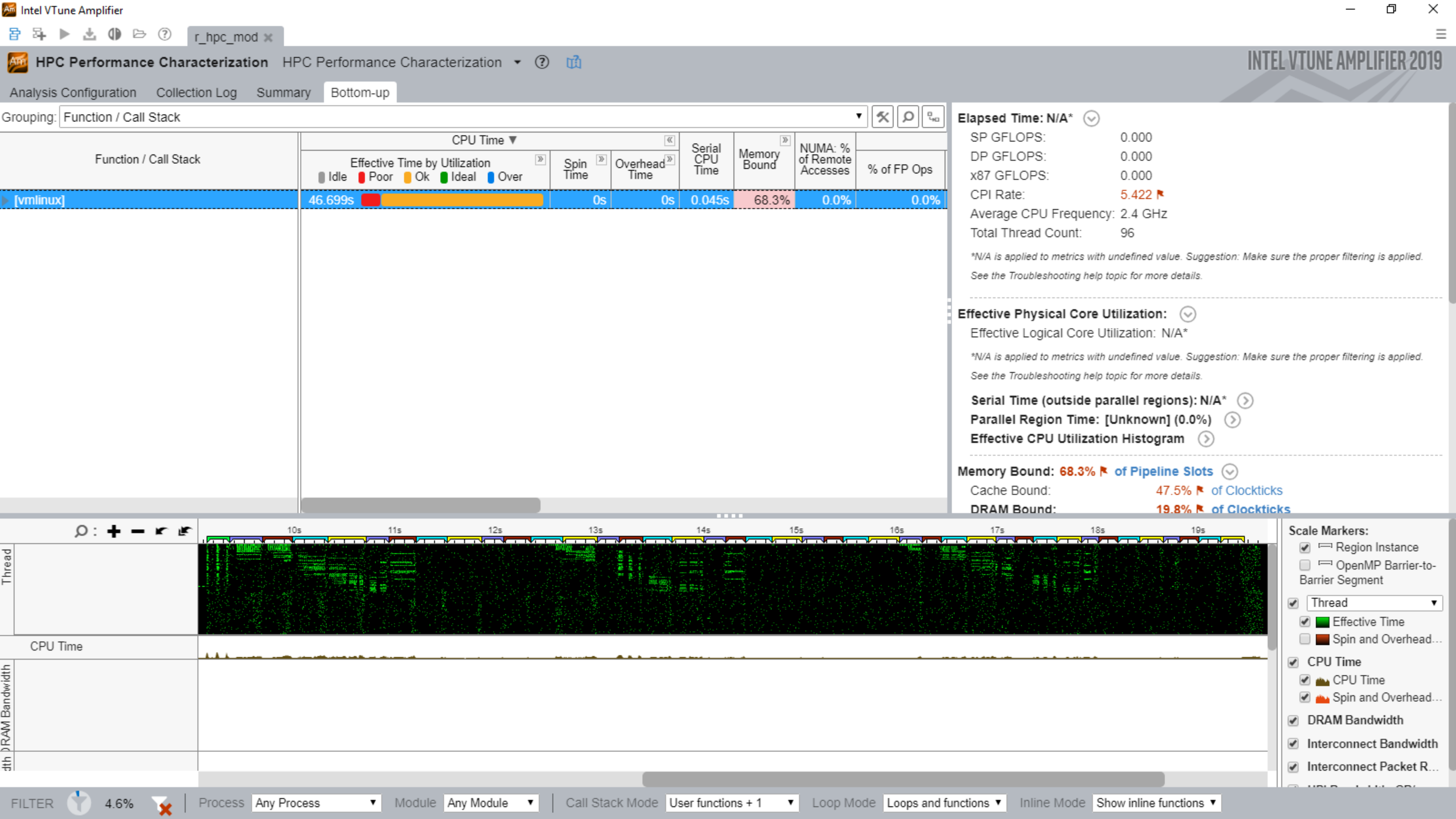


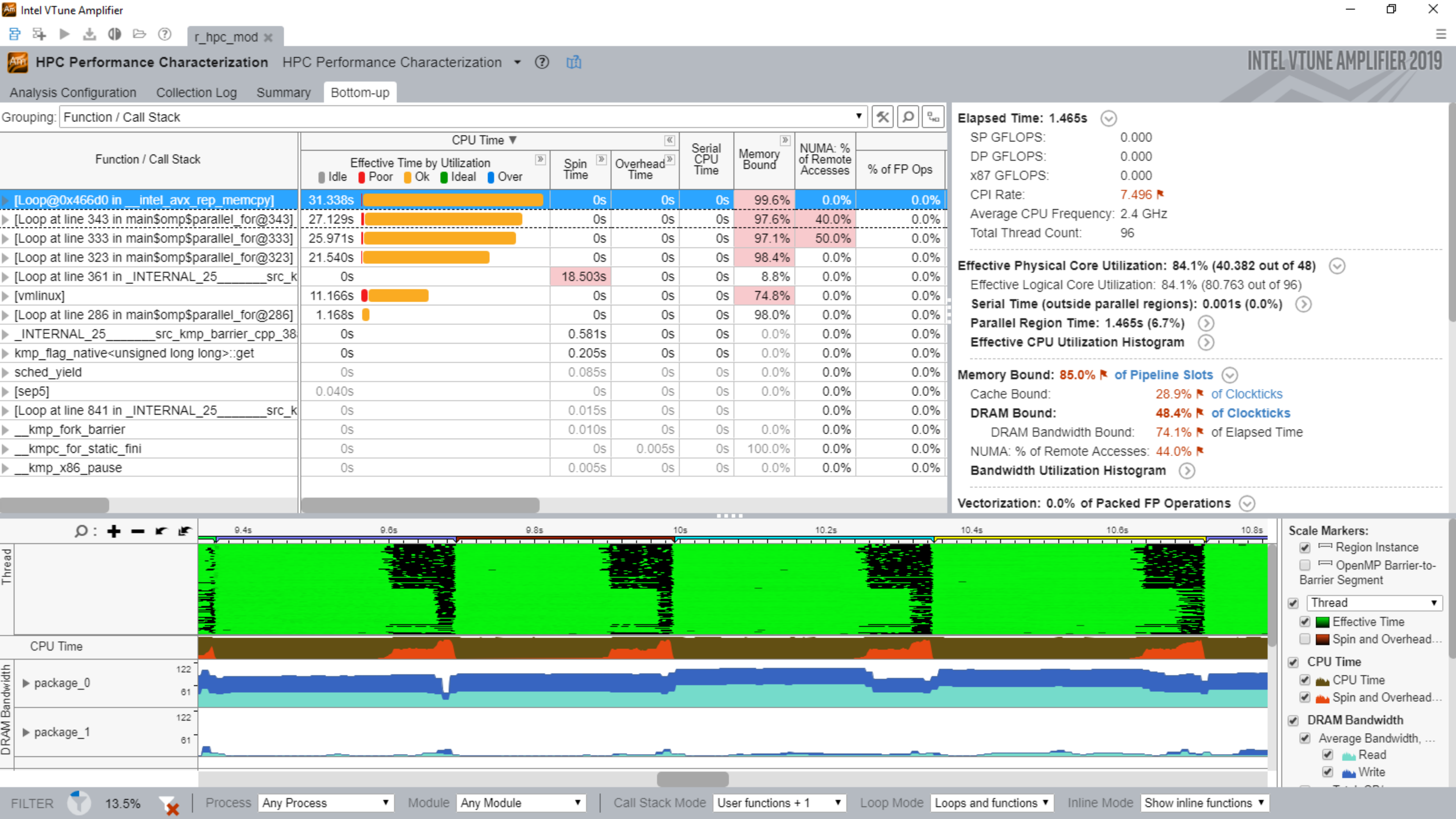
Top Functions with High Bandwidth Utilization



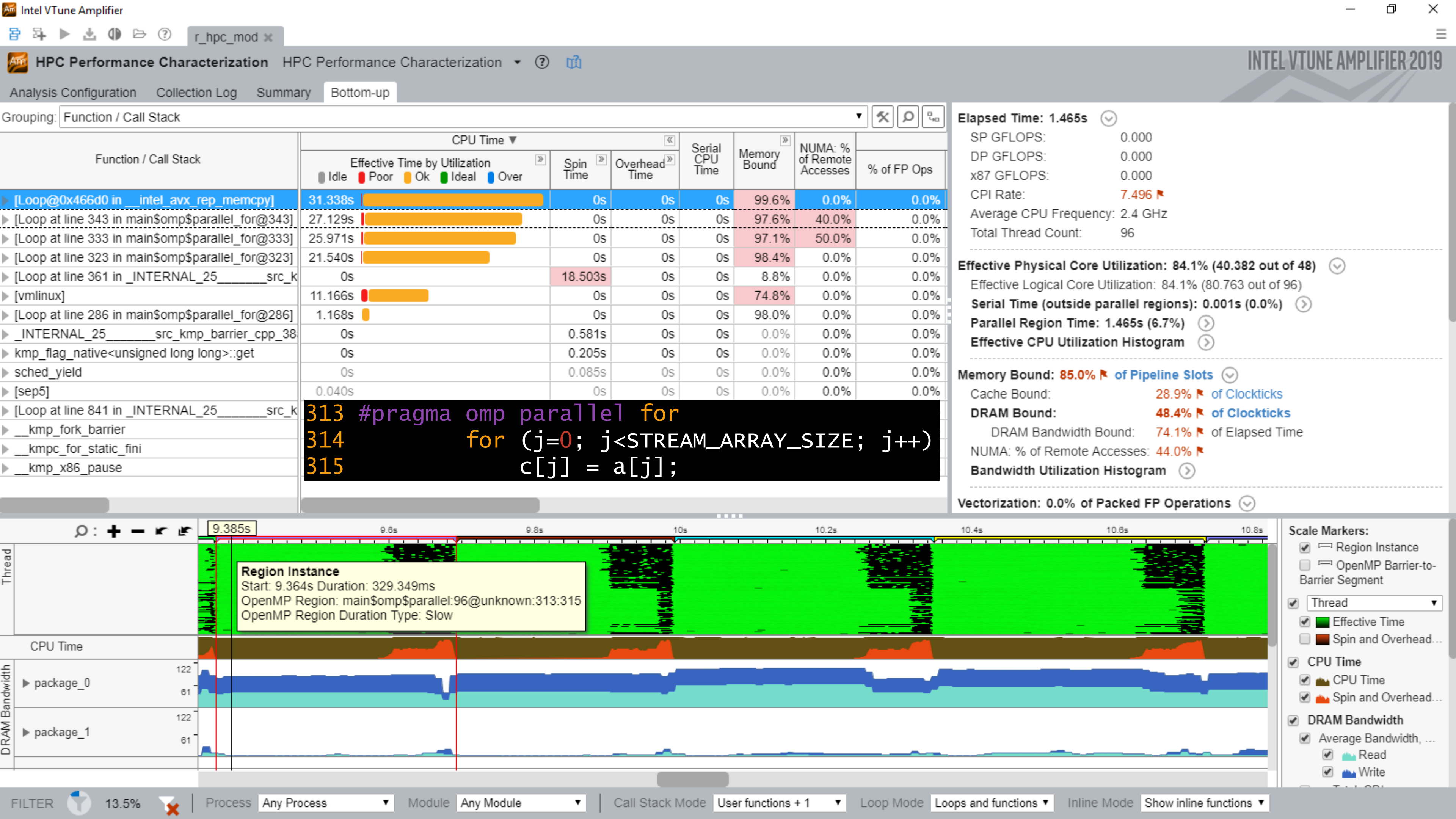
Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	CPU Time: Spin Time	CPU Time: Overhead Time	Serial CPU Time	Memory Bound	NUMA: % of Remote Accesses	% of FP Ops
[Loop at line 343 in mainompparallel_for@343]	249.314s	0s	0s	0s	96.5%	30.9%	22.7%
[Loop at line 333 in mainompparallel_for@333]	226.390s	0s	0s	0s	96.2%	36.0%	8.3%
[Loop at line 323 in mainompparallel_for@323]	183.530s	0s	0s	0s	96.5%	16.7%	0.8%
[Loop@0x466d0 in __intel_avx_rep_memcpy]	179.866s	0s	0s	0s	98.2%	0.0%	0.0%
[Loop at line 361 in _INTERNAL_25_______src_k	0s	96.326s	0s	0s	9.4%	0.0%	0.0%
[vmlinux]	53.415s	0s	0s	4.962s	69.5%	0.0%	0.0%
[Loop at line 286 in mainompparallel_for@286]	17.471s	0s	0s	0s	98.5%	0.0%	0.0%
[Loop at line 268 in main]	4.210s	0s	0s	4.210s	90.3%	0.0%	0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38	0s	2.777s	0s	0s	8.1%	0.0%	0.0%
[Loop at line 462 in checkSTREAMresults]	2.225s	0s	0s	2.225s	55.9%	66.7%	0.0%
kmp_flag_native<unsigned long long>::get	0s	1.103s	0s	0s	0.0%	0.0%	0.0%
sched_yield	0s	0.346s	0s	0s	0.0%	0.0%	0.0%
[sep5]	0.150s	0s	0s	0.005s	0.0%	0.0%	0.0%
[Loop at line 841 in _INTERNAL_25_______src_k	0s	0.020s	0s	0s	0.0%	0.0%	0.0%
__kmp_x86_pause	0s	0.015s	0s	0s	0.0%	0.0%	0.0%
[Loop at line 2073 in __kmp_join_barrier]	0s	0.015s	0s	0s	0.0%	0.0%	0.0%

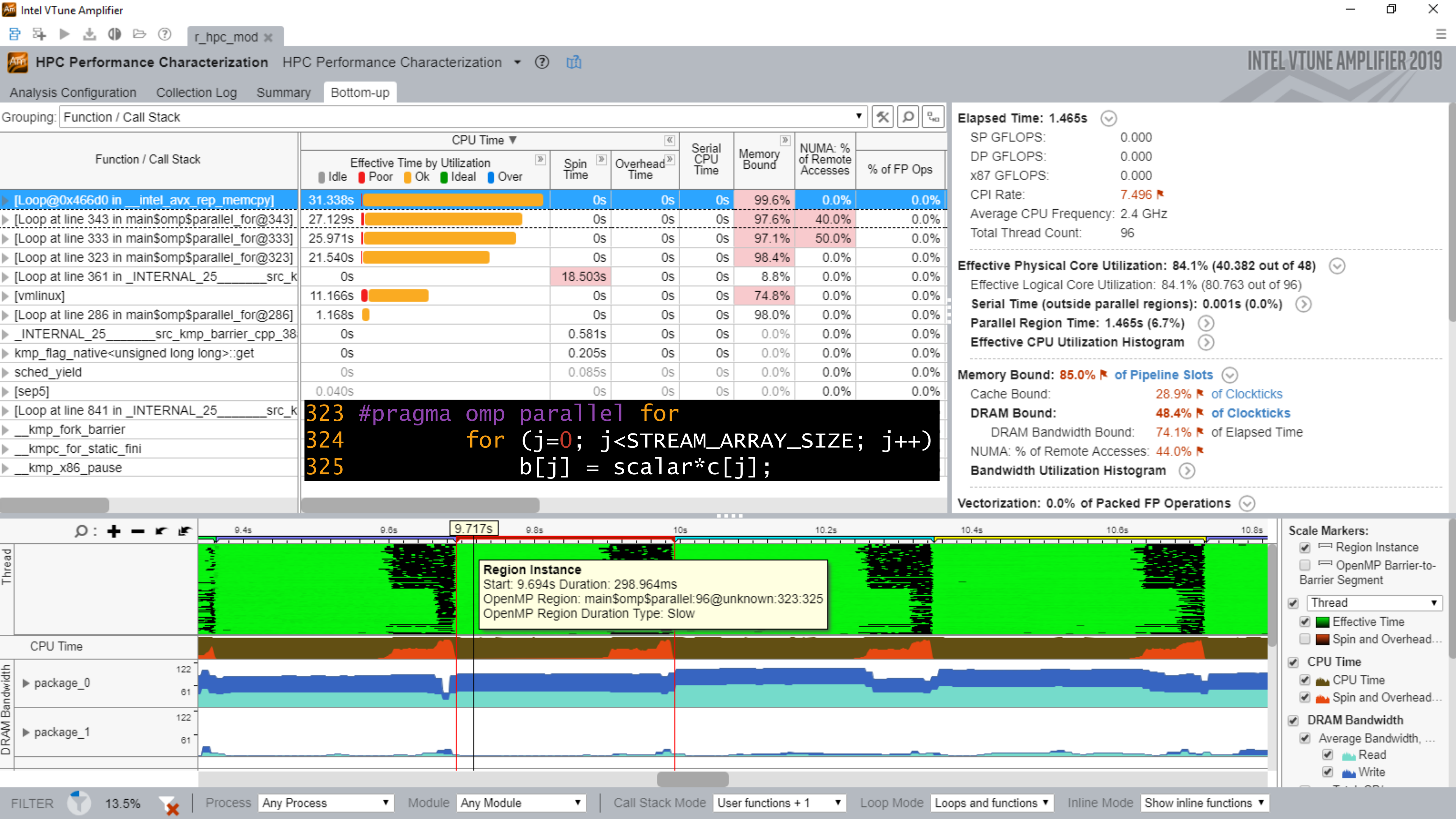

Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy]
179.866s
0s
0s
0s
98.2%
0.0%
0.0%
313 #pragma omp parallel for
314 for (j=0; j<STREAM_ARRAY_SIZE; j++)
315 c[j] = a[j];
Elapsed Time: N/A*
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 56.084
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
*N/A is applied to metrics with undefined value. Suggestion: Make sure the proper filtering is applied.
See the Troubleshooting help topic for more details.
Effective Physical Core Utilization:
Effective Logical Core Utilization: N/A*
*N/A is applied to metrics with undefined value. Suggestion: Make sure the proper filtering is applied.
See the Troubleshooting help topic for more details.
Serial Time (outside parallel regions): N/A*
Parallel Region Time: [Unknown] (0.0%)
Effective CPU Utilization Histogram
Memory Bound: 98.2% of Pipeline Slots
Cache Bound: 44.4% of Clockticks
DRAM Bound: 43.9% of Clockticks
10s
11s
12s
13s
14s
15s
16.157s
17s
18s
19s
Thread
CPU Time
DRAM Bandwidth
Region Instance
Start: 16.023s Duration: 227.766ms
OpenMP Region: mainompparallel:96@unknown:313:315
OpenMP Region Duration Type: Good
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
CPU Time
Spin and Overhead...
DRAM Bandwidth
Interconnect Bandwidth
Interconnect Packet R...
FILTER
17.7%
Process
Any Process
Module
Any Module
Call Stack Mode
User functions + 1
Loop Mode
Loops and functions
Inline Mode
Show inline functions

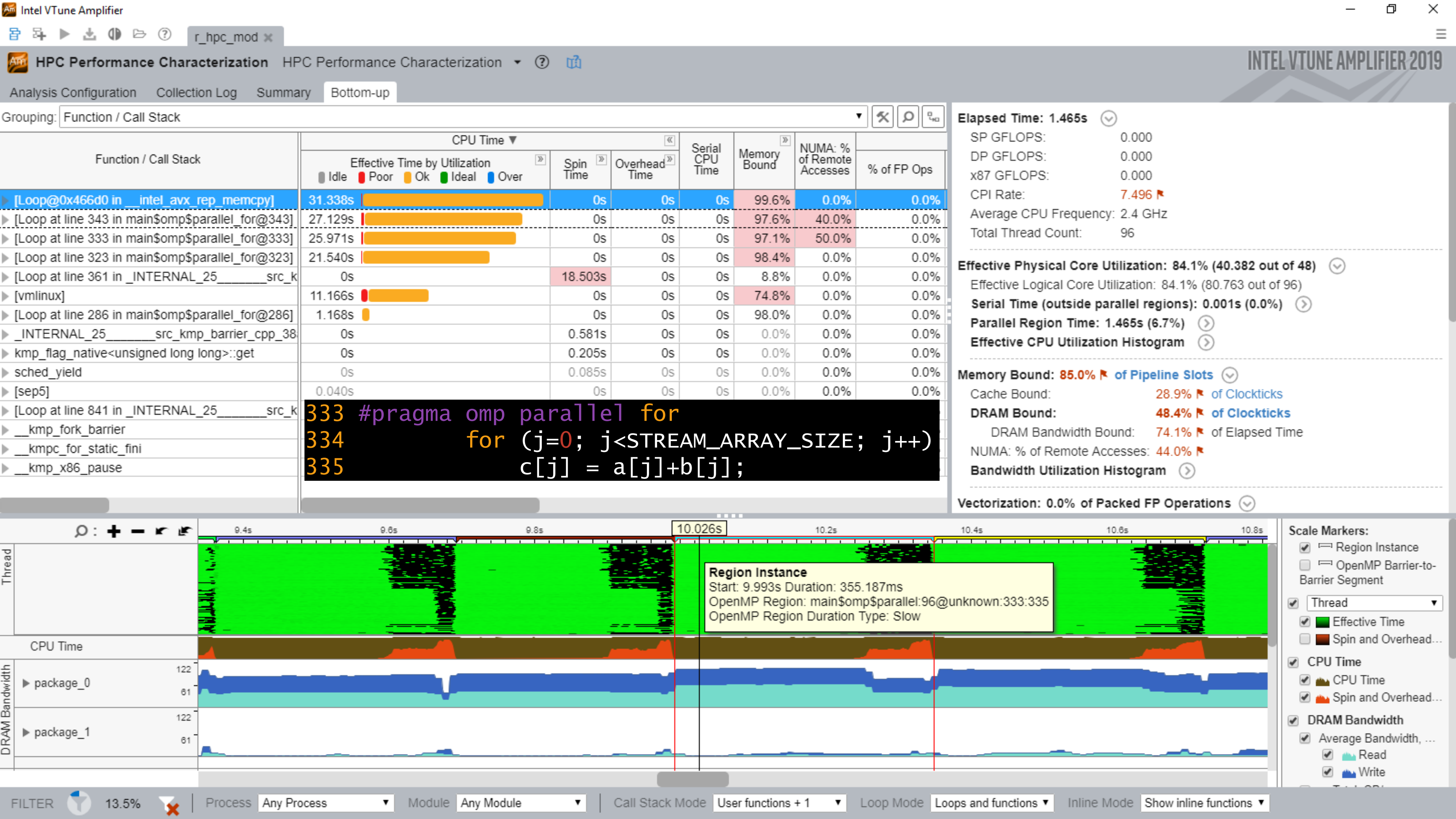

Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[vmlinux] 46.699s 0s 0s 0.045s 68.3% 0.0% 0.0%
Elapsed Time: N/A*
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 5.422
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
*N/A is applied to metrics with undefined value. Suggestion: Make sure the proper filtering is applied.
See the Troubleshooting help topic for more details.
Effective Physical Core Utilization:
Effective Logical Core Utilization: N/A*
*N/A is applied to metrics with undefined value. Suggestion: Make sure the proper filtering is applied.
See the Troubleshooting help topic for more details.
Serial Time (outside parallel regions): N/A*
Parallel Region Time: [Unknown] (0.0%)
Effective CPU Utilization Histogram
Memory Bound: 68.3% of Pipeline Slots
Cache Bound: 47.5% of Clockticks
DRAM Bound: 19.8% of Clockticks
10s 11s 12s 13s 14s 15s 16s 17s 18s 19s
Thread
CPU Time
DRAM Bandwidth
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
CPU Time
Spin and Overhead...
DRAM Bandwidth
Interconnect Bandwidth
Interconnect Packet R...
FILTER 4.6%
Process Any Process
Module Any Module
Call Stack Mode User functions + 1
Loop Mode Loops and functions
Inline Mode Show inline functions

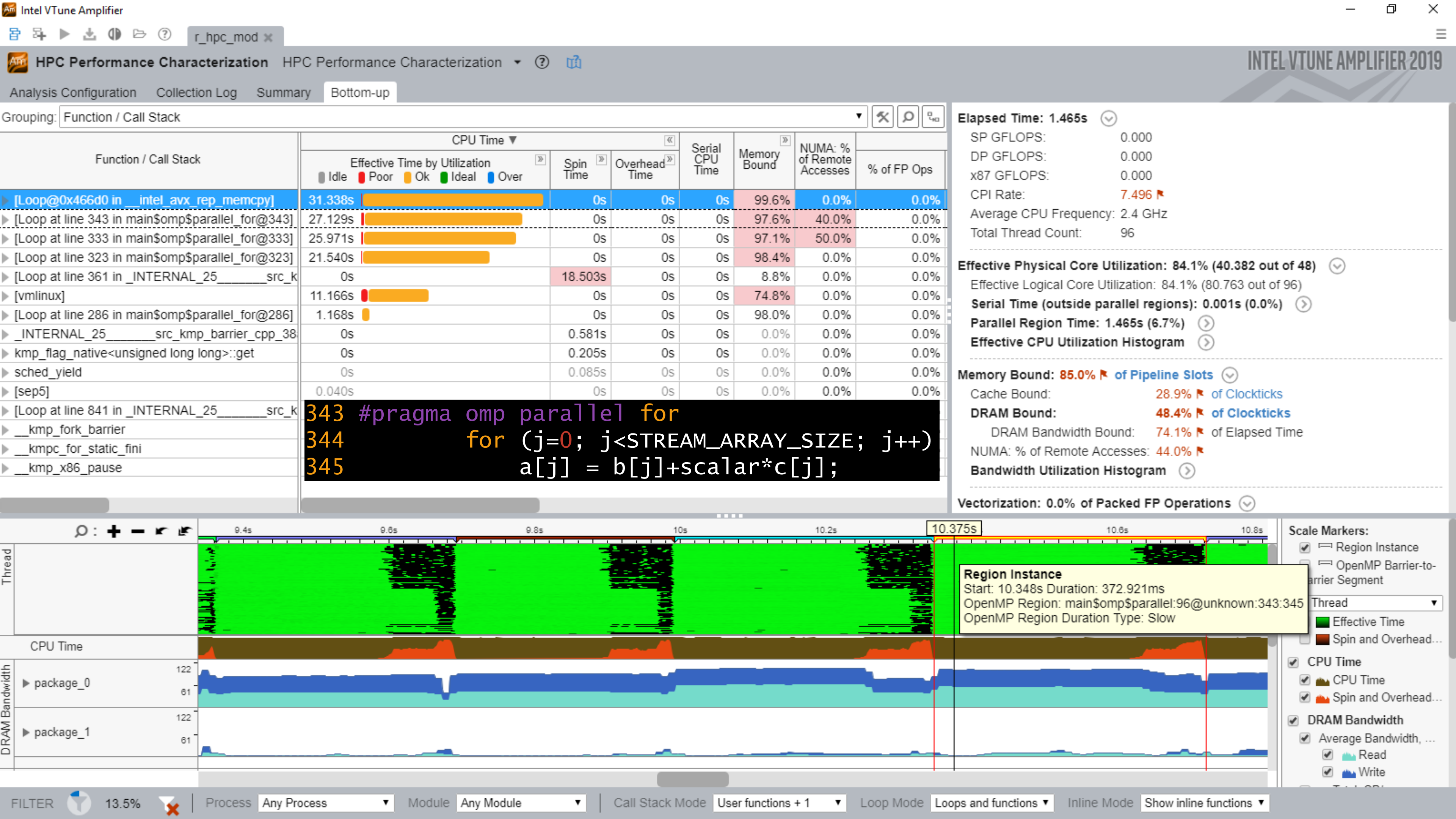


Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Time	Overhead Time	Serial CPU Time	Memory Bound	NUMA: % of Remote Accesses	% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy]	31.338s	0s	0s	0s	99.6%	0.0%	0.0%
[Loop at line 343 in mainompparallel_for@343]	27.129s	0s	0s	0s	97.6%	40.0%	0.0%
[Loop at line 333 in mainompparallel_for@333]	25.971s	0s	0s	0s	97.1%	50.0%	0.0%
[Loop at line 323 in mainompparallel_for@323]	21.540s	0s	0s	0s	98.4%	0.0%	0.0%
[Loop at line 361 in _INTERNAL_25_______src_k	0s	18.503s	0s	0s	8.8%	0.0%	0.0%
[vmlinux]	11.166s	0s	0s	0s	74.8%	0.0%	0.0%
[Loop at line 286 in mainompparallel_for@286]	1.168s	0s	0s	0s	98.0%	0.0%	0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38	0s	0.581s	0s	0s	0.0%	0.0%	0.0%
kmp_flag_native<unsigned long long>::get	0s	0.205s	0s	0s	0.0%	0.0%	0.0%
sched_yield	0s	0.085s	0s	0s	0.0%	0.0%	0.0%
[sep5]	0.040s	0s	0s	0s	0.0%	0.0%	0.0%
[Loop at line 841 in _INTERNAL_25_______src_k	0s	0.015s	0s	0s		0.0%	0.0%
__kmp_fork_barrier	0s	0.010s	0s	0s	0.0%	0.0%	0.0%
__kmpc_for_static_fini	0s	0s	0.005s	0s	100.0%	0.0%	0.0%
__kmp_x86_pause	0s	0.005s	0s	0s	0.0%	0.0%	0.0%


Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy] 31.338s 0s 0s 0s 99.6% 0.0% 0.0%
[Loop at line 343 in mainompparallel_for@343] 27.129s 0s 0s 0s 97.6% 40.0% 0.0%
[Loop at line 333 in mainompparallel_for@333] 25.971s 0s 0s 0s 97.1% 50.0% 0.0%
[Loop at line 323 in mainompparallel_for@323] 21.540s 0s 0s 0s 98.4% 0.0% 0.0%
[Loop at line 361 in _INTERNAL_25_______src_k 0s 18.503s 0s 0s 8.8% 0.0% 0.0%
[vmlinux] 11.166s 0s 0s 0s 74.8% 0.0% 0.0%
[Loop at line 286 in mainompparallel_for@286] 1.168s 0s 0s 0s 98.0% 0.0% 0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38 0s 0.581s 0s 0s 0.0% 0.0% 0.0%
kmp_flag_native<unsigned long long>::get 0s 0.205s 0s 0s 0.0% 0.0% 0.0%
sched_yield 0s 0.085s 0s 0s 0.0% 0.0% 0.0%
[sep5] 0.040s 0s 0s 0s 0.0% 0.0% 0.0%
[Loop at line 841 in _INTERNAL_25_______src_k
__kmp_fork_barrier
__kmpc_for_static_fini
__kmp_x86_pause
313 #pragma omp parallel for
314 for (j=0; j<STREAM_ARRAY_SIZE; j++)
315 c[j] = a[j];
Elapsed Time: 1.465s
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 7.496
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
Effective Physical Core Utilization: 84.1% (40.382 out of 48)
Effective Logical Core Utilization: 84.1% (80.763 out of 96)
Serial Time (outside parallel regions): 0.001s (0.0%)
Parallel Region Time: 1.465s (6.7%)
Effective CPU Utilization Histogram
Memory Bound: 85.0% of Pipeline Slots
Cache Bound: 28.9% of Clockticks
DRAM Bound: 48.4% of Clockticks
DRAM Bandwidth Bound: 74.1% of Elapsed Time
NUMA: % of Remote Accesses: 44.0%
Bandwidth Utilization Histogram
Vectorization: 0.0% of Packed FP Operations
9.385s
9.6s
9.8s
10s
10.2s
10.4s
10.6s
10.8s
Thread
Region Instance
Start: 9.364s Duration: 329.349ms
OpenMP Region: mainompparallel:96@unknown:313:315
OpenMP Region Duration Type: Slow
CPU Time
DRAM Bandwidth
package_0
package_1
122
61
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
DRAM Bandwidth
Average Bandwidth, ...
Read
Write
FILTER
13.5%
Process
Any Process
Module
Any Module
Call Stack Mode
User functions + 1
Loop Mode
Loops and functions
Inline Mode
Show inline functions

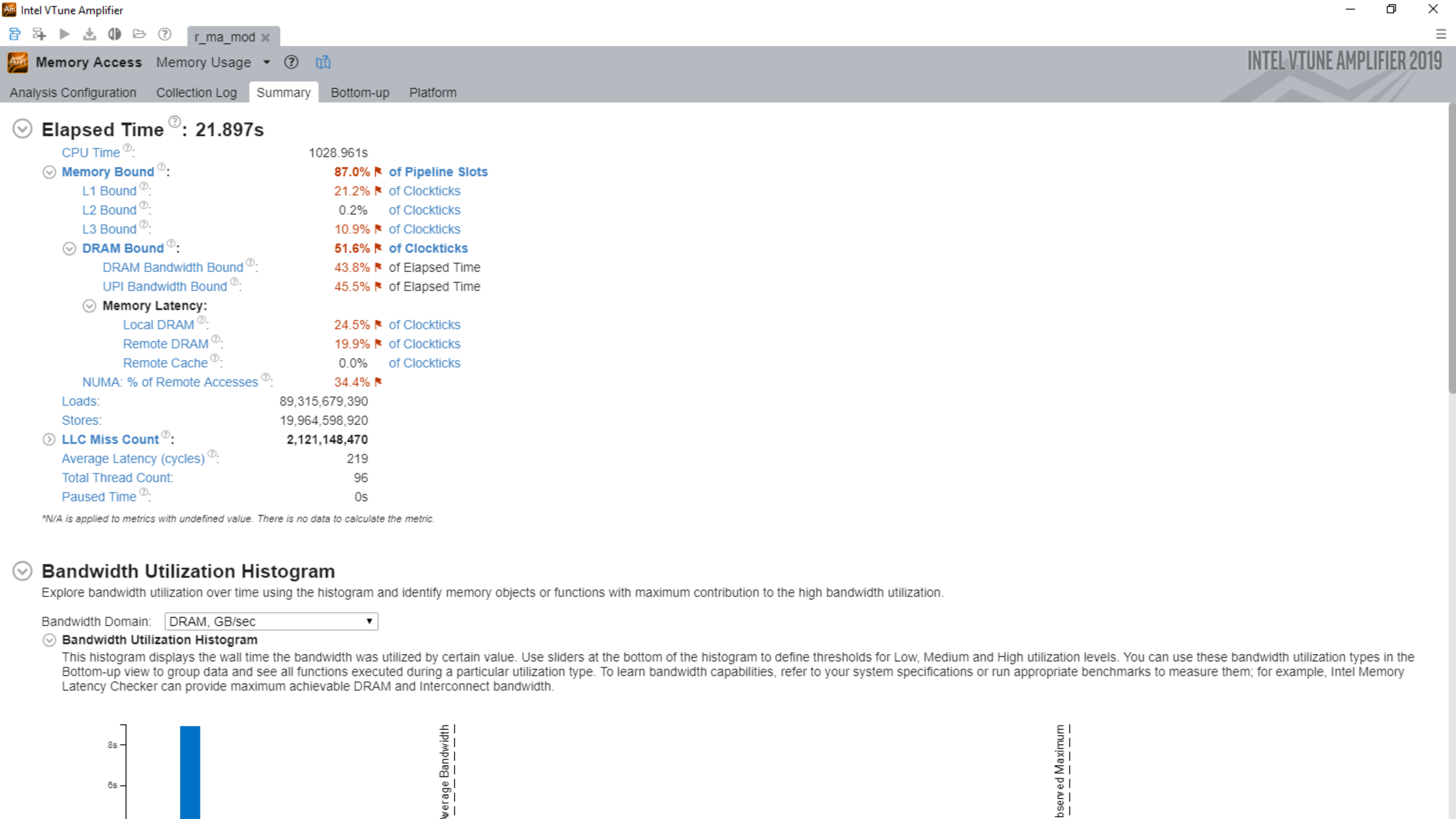

Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy] 31.338s 0s 0s 0s 99.6% 0.0% 0.0%
[Loop at line 343 in mainompparallel_for@343] 27.129s 0s 0s 0s 97.6% 40.0% 0.0%
[Loop at line 333 in mainompparallel_for@333] 25.971s 0s 0s 0s 97.1% 50.0% 0.0%
[Loop at line 323 in mainompparallel_for@323] 21.540s 0s 0s 0s 98.4% 0.0% 0.0%
[Loop at line 361 in _INTERNAL_25_______src_k 0s 18.503s 0s 0s 8.8% 0.0% 0.0%
[vmlinux] 11.166s 0s 0s 0s 74.8% 0.0% 0.0%
[Loop at line 286 in mainompparallel_for@286] 1.168s 0s 0s 0s 98.0% 0.0% 0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38 0s 0.581s 0s 0s 0.0% 0.0% 0.0%
kmp_flag_native<unsigned long long>::get 0s 0.205s 0s 0s 0.0% 0.0% 0.0%
sched_yield 0s 0.085s 0s 0s 0.0% 0.0% 0.0%
[sep5] 0.040s 0s 0s 0s 0.0% 0.0% 0.0%
[Loop at line 841 in _INTERNAL_25_______src_k
__kmp_fork_barrier
__kmpc_for_static_fini
__kmp_x86_pause
323 #pragma omp parallel for
324 for (j=0; j<STREAM_ARRAY_SIZE; j++)
325 b[j] = scalar*c[j];
Elapsed Time: 1.465s
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 7.496
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
Effective Physical Core Utilization: 84.1% (40.382 out of 48)
Effective Logical Core Utilization: 84.1% (80.763 out of 96)
Serial Time (outside parallel regions): 0.001s (0.0%)
Parallel Region Time: 1.465s (6.7%)
Effective CPU Utilization Histogram
Memory Bound: 85.0% of Pipeline Slots
Cache Bound: 28.9% of Clockticks
DRAM Bound: 48.4% of Clockticks
DRAM Bandwidth Bound: 74.1% of Elapsed Time
NUMA: % of Remote Accesses: 44.0%
Bandwidth Utilization Histogram
Vectorization: 0.0% of Packed FP Operations
9.4s 9.6s 9.717s 9.8s 10s 10.2s 10.4s 10.6s 10.8s
Thread
Region Instance
Start: 9.694s Duration: 298.964ms
OpenMP Region: mainompparallel:96@unknown:323:325
OpenMP Region Duration Type: Slow
CPU Time
DRAM Bandwidth
package_0
package_1
122
61
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
CPU Time
Spin and Overhead...
DRAM Bandwidth
Average Bandwidth, ...
Read
Write
FILTER 13.5%
Process Any Process
Module Any Module
Call Stack Mode User functions + 1
Loop Mode Loops and functions
Inline Mode Show inline functions


Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy] 31.338s 0s 0s 0s 99.6% 0.0% 0.0%
[Loop at line 343 in mainompparallel_for@343] 27.129s 0s 0s 0s 97.6% 40.0% 0.0%
[Loop at line 333 in mainompparallel_for@333] 25.971s 0s 0s 0s 97.1% 50.0% 0.0%
[Loop at line 323 in mainompparallel_for@323] 21.540s 0s 0s 0s 98.4% 0.0% 0.0%
[Loop at line 361 in _INTERNAL_25_______src_k 0s 18.503s 0s 0s 8.8% 0.0% 0.0%
[vmlinux] 11.166s 0s 0s 0s 74.8% 0.0% 0.0%
[Loop at line 286 in mainompparallel_for@286] 1.168s 0s 0s 0s 98.0% 0.0% 0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38 0s 0.581s 0s 0s 0.0% 0.0% 0.0%
kmp_flag_native<unsigned long long>::get 0s 0.205s 0s 0s 0.0% 0.0% 0.0%
sched_yield 0s 0.085s 0s 0s 0.0% 0.0% 0.0%
[sep5] 0.040s 0s 0s 0s 0.0% 0.0% 0.0%
[Loop at line 841 in _INTERNAL_25_______src_k
__kmp_fork_barrier
__kmpc_for_static_fini
__kmp_x86_pause
333 #pragma omp parallel for
334 for (j=0; j<STREAM_ARRAY_SIZE; j++)
335 c[j] = a[j]+b[j];
Elapsed Time: 1.465s
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 7.496
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
Effective Physical Core Utilization: 84.1% (40.382 out of 48)
Effective Logical Core Utilization: 84.1% (80.763 out of 96)
Serial Time (outside parallel regions): 0.001s (0.0%)
Parallel Region Time: 1.465s (6.7%)
Effective CPU Utilization Histogram
Memory Bound: 85.0% of Pipeline Slots
Cache Bound: 28.9% of Clockticks
DRAM Bound: 48.4% of Clockticks
DRAM Bandwidth Bound: 74.1% of Elapsed Time
NUMA: % of Remote Accesses: 44.0%
Bandwidth Utilization Histogram
Vectorization: 0.0% of Packed FP Operations
9.4s 9.6s 9.8s 10.026s 10.2s 10.4s 10.6s 10.8s
Thread
CPU Time
DRAM Bandwidth
package_0
package_1
122
61
Region Instance
Start: 9.993s Duration: 355.187ms
OpenMP Region: mainompparallel:96@unknown:333:335
OpenMP Region Duration Type: Slow
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
CPU Time
Spin and Overhead...
DRAM Bandwidth
Average Bandwidth, ...
Read
Write
FILTER 13.5%
Process Any Process
Module Any Module
Call Stack Mode User functions + 1
Loop Mode Loops and functions
Inline Mode Show inline functions

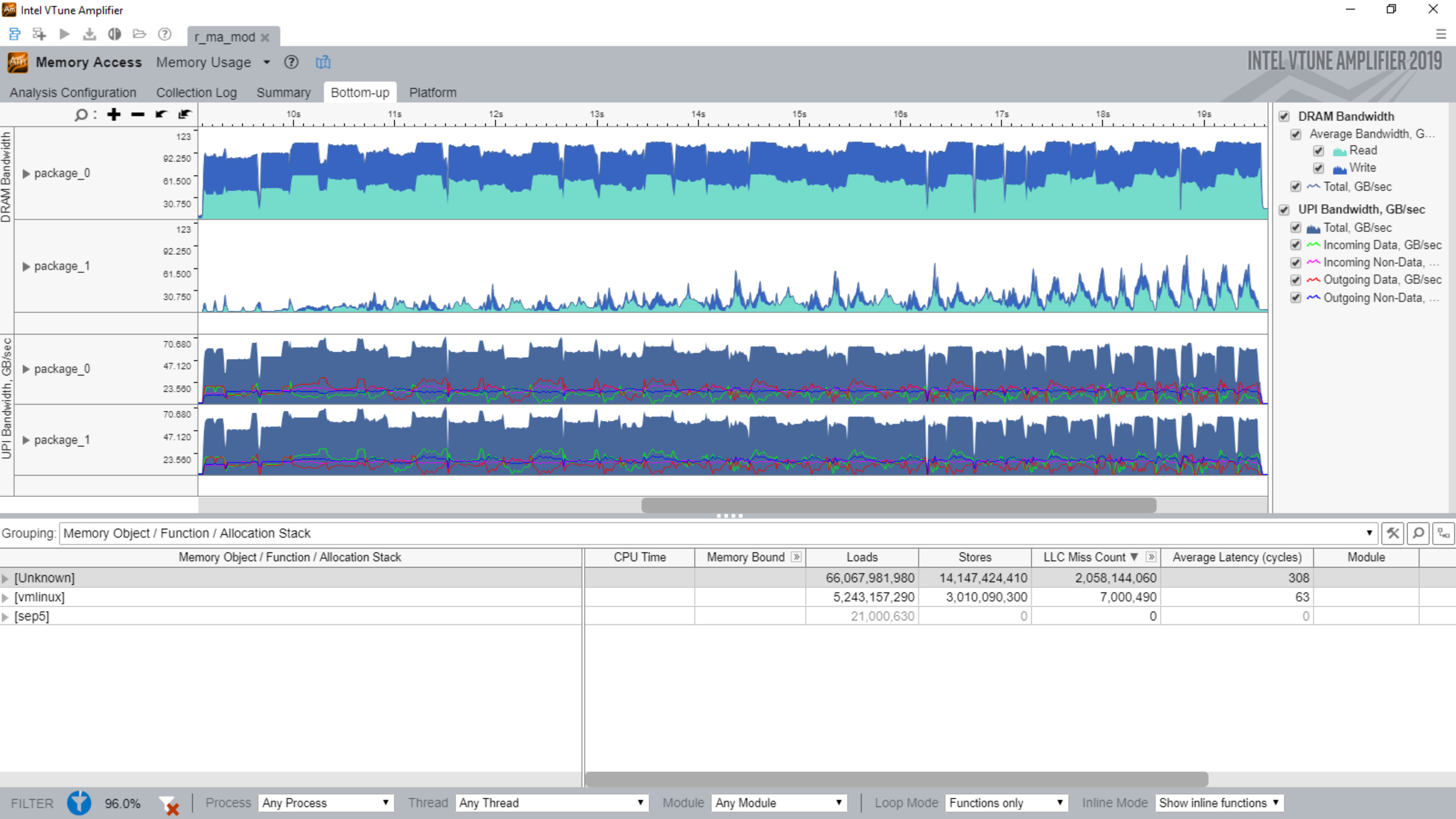

Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
[Loop@0x466d0 in __intel_avx_rep_memcpy] 31.338s 0s 0s 0s 99.6% 0.0% 0.0%
[Loop at line 343 in mainompparallel_for@343] 27.129s 0s 0s 0s 97.6% 40.0% 0.0%
[Loop at line 333 in mainompparallel_for@333] 25.971s 0s 0s 0s 97.1% 50.0% 0.0%
[Loop at line 323 in mainompparallel_for@323] 21.540s 0s 0s 0s 98.4% 0.0% 0.0%
[Loop at line 361 in _INTERNAL_25_______src_k 0s 18.503s 0s 0s 8.8% 0.0% 0.0%
[vmlinux] 11.166s 0s 0s 0s 74.8% 0.0% 0.0%
[Loop at line 286 in mainompparallel_for@286] 1.168s 0s 0s 0s 98.0% 0.0% 0.0%
_INTERNAL_25_______src_kmp_barrier_cpp_38 0s 0.581s 0s 0s 0.0% 0.0% 0.0%
kmp_flag_native<unsigned long long>::get 0s 0.205s 0s 0s 0.0% 0.0% 0.0%
sched_yield 0s 0.085s 0s 0s 0.0% 0.0% 0.0%
[sep5] 0.040s 0s 0s 0s 0.0% 0.0% 0.0%
[Loop at line 841 in _INTERNAL_25_______src_k
__kmp_fork_barrier
__kmpc_for_static_fini
__kmp_x86_pause
343 #pragma omp parallel for
344 for (j=0; j<STREAM_ARRAY_SIZE; j++)
345 a[j] = b[j]+scalar*c[j];
Elapsed Time: 1.465s
SP GFLOPS: 0.000
DP GFLOPS: 0.000
x87 GFLOPS: 0.000
CPI Rate: 7.496
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
Effective Physical Core Utilization: 84.1% (40.382 out of 48)
Effective Logical Core Utilization: 84.1% (80.763 out of 96)
Serial Time (outside parallel regions): 0.001s (0.0%)
Parallel Region Time: 1.465s (6.7%)
Effective CPU Utilization Histogram
Memory Bound: 85.0% of Pipeline Slots
Cache Bound: 28.9% of Clockticks
DRAM Bound: 48.4% of Clockticks
DRAM Bandwidth Bound: 74.1% of Elapsed Time
NUMA: % of Remote Accesses: 44.0%
Bandwidth Utilization Histogram
Vectorization: 0.0% of Packed FP Operations
9.4s 9.6s 9.8s 10s 10.2s 10.375s 10.6s 10.8s
Thread
Region Instance
Start: 10.348s Duration: 372.921ms
OpenMP Region: mainompparallel:96@unknown:343:345
OpenMP Region Duration Type: Slow
CPU Time
DRAM Bandwidth
package_0
package_1
122
61
Scale Markers:
Region Instance
OpenMP Barrier-to-Barrier Segment
Thread
Effective Time
Spin and Overhead...
CPU Time
CPU Time
Spin and Overhead...
DRAM Bandwidth
Average Bandwidth, ...
Read
Write
FILTER 13.5%
Process Any Process
Module Any Module
Call Stack Mode User functions + 1
Loop Mode Loops and functions
Inline Mode Show inline functions

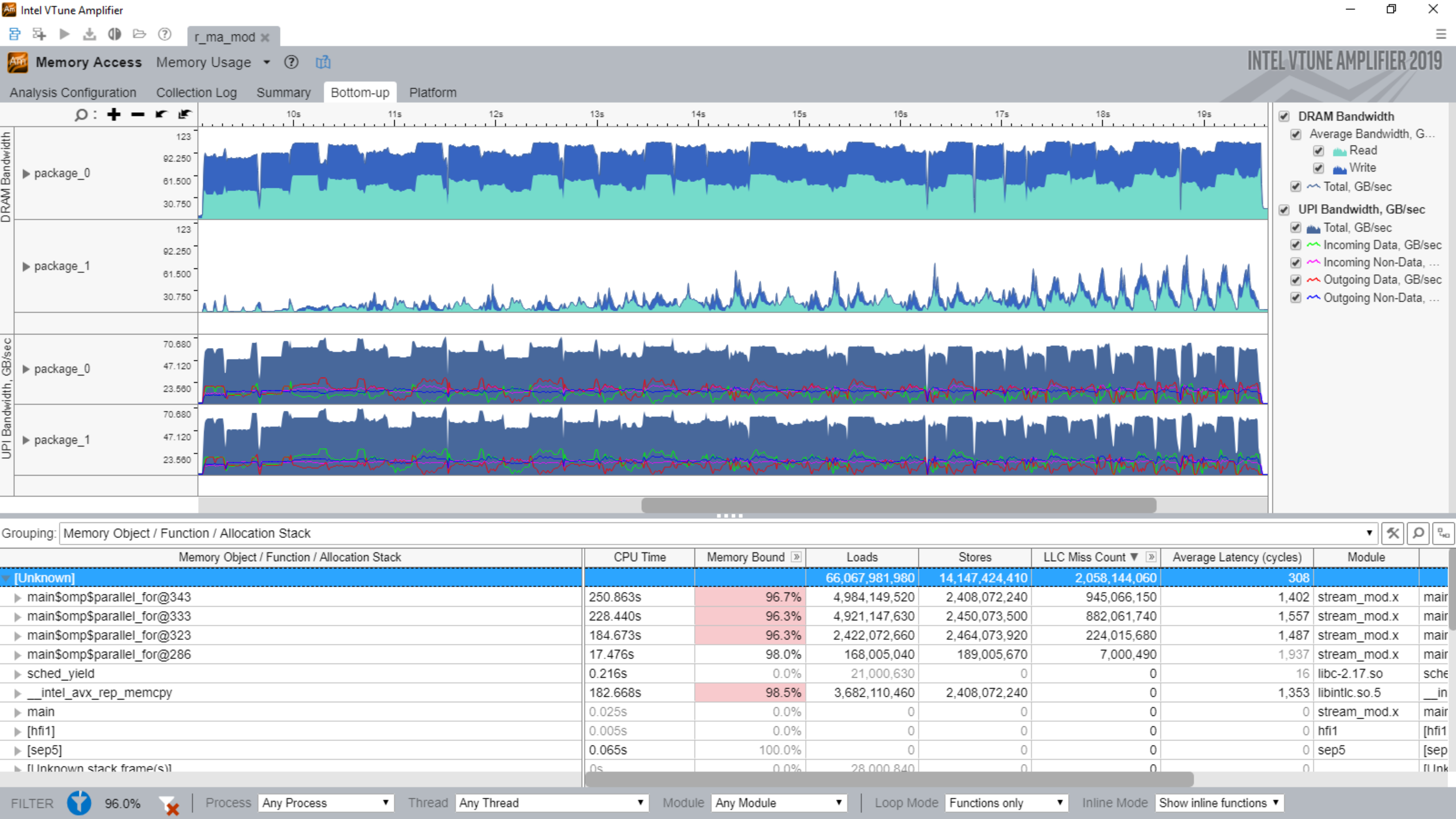
Collecting Memory Access

```
vtune -c memory-access -r ./r_ma_mod -- ./stream_mod.x
```



Intel VTune Amplifier
r_ma_mod
Memory Access
Memory Usage
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Platform
Elapsed Time: 21.897s
CPU Time: 1028.961s
Memory Bound: 87.0% of Pipeline Slots
L1 Bound: 21.2% of Clockticks
L2 Bound: 0.2% of Clockticks
L3 Bound: 10.9% of Clockticks
DRAM Bound: 51.6% of Clockticks
DRAM Bandwidth Bound: 43.8% of Elapsed Time
UPI Bandwidth Bound: 45.5% of Elapsed Time
Memory Latency:
Local DRAM: 24.5% of Clockticks
Remote DRAM: 19.9% of Clockticks
Remote Cache: 0.0% of Clockticks
NUMA: % of Remote Accesses: 34.4%
Loads: 89,315,679,390
Stores: 19,964,598,920
LLC Miss Count: 2,121,148,470
Average Latency (cycles): 219
Total Thread Count: 96
Paused Time: 0s
*N/A is applied to metrics with undefined value. There is no data to calculate the metric.
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.
8s
6s


Intel VTune Amplifier
r_ma_mod
Memory Access
Memory Usage
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Platform
10s
11s
12s
13s
14s
15s
16s
17s
18s
19s
DRAM Bandwidth
package_0
package_1
123
92.250
61.500
30.750
UPI Bandwidth, GB/sec
70.680
47.120
23.560
DRAM Bandwidth
Average Bandwidth, G...
Read
Write
Total, GB/sec
UPI Bandwidth, GB/sec
Total, GB/sec
Incoming Data, GB/sec
Incoming Non-Data, ...
Outgoing Data, GB/sec
Outgoing Non-Data, ...
Grouping: Memory Object / Function / Allocation Stack
Memory Object / Function / Allocation Stack
CPU Time
Memory Bound
Loads
Stores
LLC Miss Count
Average Latency (cycles)
Module
[Unknown] 66,067,981,980 14,147,424,410 2,058,144,060 308
[vmlinux] 5,243,157,290 3,010,090,300 7,000,490 63
[sep5] 21,000,630 0 0 0
FILTER
96.0%
Process
Any Process
Thread
Any Thread
Module
Any Module
Loop Mode
Functions only
Inline Mode
Show inline functions



Memory Object / Function / Allocation Stack	CPU Time	Memory Bound	Loads	Stores	LLC Miss Count	Average Latency (cycles)	Module	
[Unknown]			66,067,981,980	14,147,424,410	2,058,144,060	308		
mainompparallel_for@343	250.863s	96.7%	4,984,149,520	2,408,072,240	945,066,150	1,402	stream_mod.x	mair
mainompparallel_for@333	228.440s	96.3%	4,921,147,630	2,450,073,500	882,061,740	1,557	stream_mod.x	mair
mainompparallel_for@323	184.673s	96.3%	2,422,072,660	2,464,073,920	224,015,680	1,487	stream_mod.x	mair
mainompparallel_for@286	17.476s	98.0%	168,005,040	189,005,670	7,000,490	1,937	stream_mod.x	mair
sched_yield	0.216s	0.0%	21,000,630	0	0	16	libc-2.17.so	sche
__intel_avx_rep_memcpy	182.668s	98.5%	3,682,110,460	2,408,072,240	0	1,353	libintlc.so.5	__in
main	0.025s	0.0%	0	0	0	0	stream_mod.x	mair
[hfi1]	0.005s	0.0%	0	0	0	0	hfi1	[hfi1
[sep5]	0.065s	100.0%	0	0	0	0	sep5	[sep
[Unknown stack frame(s)]	0s	0.0%	28,000,840	0	0	0		[Unk

Collecting Memory Access with Objects

```
vtune -c memory-access -knob analyze-mem-objects=true -r ./r_mao_mod -- ./stream_mod.x
```

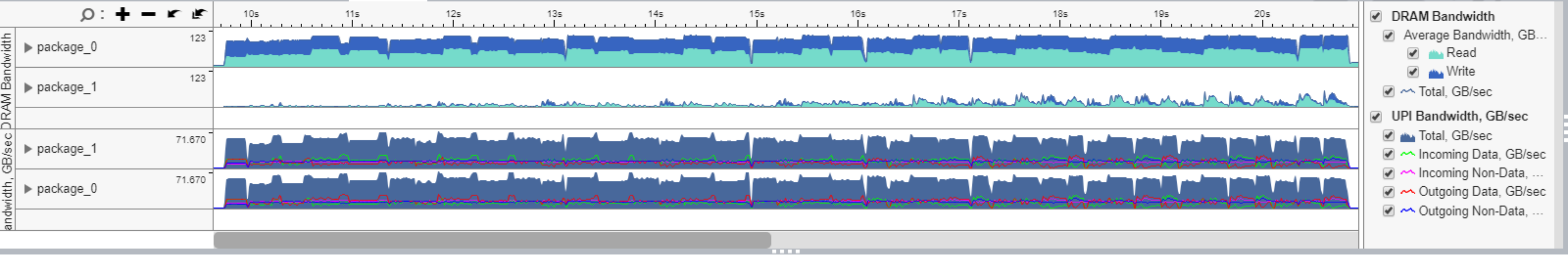
Intel VTune Amplifier

r_mao_mod

Memory Access Memory Usage

INTEL VTUNE AMPLIFIER 2019

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



Grouping: Memory Object / Function / Allocation Stack

Memory Object / Function / Allocation Stack	CPU Time	Memory Bound	Loads	Stores	LLC Miss Count	Average Latency (cycles)	Module	Function (Ful
stream_mod.x!c (7 GB)			5,530,165,900	4,942,148,260	840,058,800	1,773		
stream_mod.x!b (7 GB)			3,570,107,100	2,443,073,290	714,049,980	1,716		
stream_mod.x!a (7 GB)			6,986,209,580	2,618,078,540	560,039,200	1,599		
mainompparallel_for@333	0s	0.0%	3,143,094,290	0	518,036,260	1,656	stream...	mainompp
mainompparallel_for@286	0s	0.0%	189,005,670	203,006,090	28,001,960	0	stream...	mainompp
__intel_avx_rep_memcpy	0s	0.0%	3,654,109,620	0	14,000,980	1,531	libintlc....	__intel_avx_r
mainompparallel_for@343	0s	0.0%	0	2,415,072,450	0	0	stream...	mainompp
[vmlinux]			5,887,176,610	3,773,113,190	7,000,490	58		
[Unknown]			266,007,980	0	0	0		
stream_mod.x!main (2 MB)			140,004,200	637,019,110	0	7		
libiomp5.so![OpenMP fork] (1 MB)			182,005,460	581,017,430	0	7		
libiomp5.so![OpenMP fork] (1 MB)			1,582,047,460	1,232,036,960	0	7		
[Stack]			32,823,984,690	2,492,074,760	0	7		
[libiomp5.so]			26,488,794,640	0	0	7		
libiomp5.so!__kmp_global (128 B)			70,002,100	0	0	7		
libiomp5.so!__kmp_nth (4 B)			105,003,150	0	0	7		
libiomp5.so!__kmp_itt_fsync_prepare_ptr__3_0 (8 B)			49,001,470	0	0	7		
libiomp5.so!__kmp_tasking_mode (4 B)			0	0	0	7		
libiomp5.so!__kmp_dflt_blocktime (4 B)			245,007,350	0	0	6		
libiomp5.so!__kmp_avail_proc (4 B)			21,000,630	0	0	7		

FILTER 96.3% | Process: Any Process | Thread: Any Thread | Module: Any Module | Loop Mode: Functions only | Inline Mode: Show inline functions

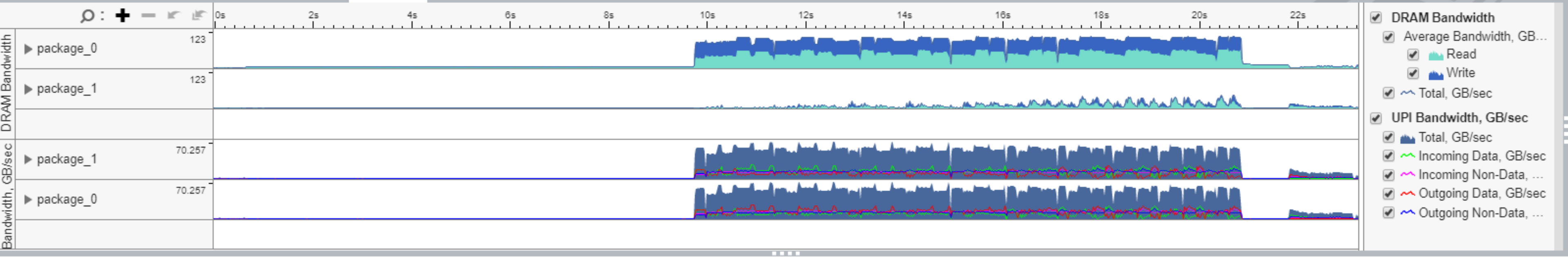
Intel VTune Amplifier

r_mao_mod

Memory Access Memory Usage

INTEL VTUNE AMPLIFIER 2019

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



Grouping: Memory Object / Function / Allocation Stack

Memory Object / Function / Allocation Stack	CPU Time	Memory Bound	Loads	Stores	LLC Miss Count	Average Latency (cycles)	Module	
stream_mod.x!c (7 GB)			5,908,177,240	5,019,150,570	847,059,290	1,695		
stream_mod.x!b (7 GB)			3,801,114,030	2,548,076,440	714,049,980	1,552		
stream_mod.x!a (7 GB)			8,337,250,110	3,073,092,190	581,040,670	1,501		
mainompparallel_for@333	0s	0.0%	3,143,094,290	0	518,036,260	1,656	stream_mod.x	main$
mainompparallel_for@286	0s	0.0%	189,005,670	203,006,090	28,001,960	0	stream_mod.x	main$
checkSTREAMresults	0s	0.0%	1,351,040,530	0	21,001,470	61	stream_mod.x	check
__intel_avx_rep_memcpy	0s	0.0%	3,654,109,620	0	14,000,980	1,531	libintlc.so.5	__inte
main	0s	0.0%	0	455,013,650	0	0	stream_mod.x	main
mainompparallel_for@343	0s	0.0%	0	2,415,072,450	0	0	stream_mod.x	main$
[vmlinux]			9,128,273,840	5,656,169,680	7,000,490	50		
[Unknown]			441,013,230	14,000,420	0	0		
stream_mod.x!main (2 MB)			161,004,830	756,022,680	0	7		
libiomp5.so![OpenMP fork] (1 MB)			210,006,300	728,021,840	0	7		
libiomp5.so![OpenMP fork] (1 MB)			1,862,055,860	1,610,048,300	0	7		
[Stack]			40,475,214,220	3,178,095,340	0	7		
[ld-linux-x86-64.so.2]			245,007,350	147,004,410	0	0		
[libiomp5.so]			32,403,972,090	0	0	7		
libiomp5.so!__kmp_global (128 B)			84,002,520	0	0	7		
libiomp5.so!__kmp_nth (4 B)			119,003,570	0	0	7		
libiomp5.so!__kmp_itt_fsync_prepare_ptr__3_0 (8 B)			56,001,680	0	0	7		

FILTER 100.0% Process Any Process Thread Any Thread Module Any Module Loop Mode Functions only Inline Mode Show inline functions

Intel VTune Amplifier

r_mao_mod

Memory Access Memory Usage

INTEL VTUNE AMPLIFIER 2019

Analysis Configuration | Collection Log | Summary | Bottom-up | Platform | stream_mod.c

Source | Assembly

Source	CPU Time: Total	CPU Time: Self	Memory Bound: Total

```
            printf ("Number of Threads requested = %i\n",k);
            }
        }
#endif

#ifdef _OPENMP
        k = 0;
#pragma omp parallel
#pragma omp atomic
            k++;
        printf ("Number of Threads counted = %i\n",k);
#endif

        /* Get initial value for system clock. */
//#pragma omp parallel for
        for (j=0; j<STREAM_ARRAY_SIZE; j++) {
            a[j] = 1.0;
            b[j] = 2.0;
            c[j] = 0.0;
        }

        printf(HLINE);

        if  ( (quantum = checktick()) >= 1)
        printf("Your clock granularity/precision appears to be "
            "%d microseconds.\n", quantum);
        else {
        printf("Your clock granularity appears to be "
            "less than one microsecond.\n");
        quantum = 1;
        }

        t = mysecond();
#pragma omp parallel for
```

Source	CPU Time: Total	CPU Time: Self	Memory Bound: Total
a[j] = 1.0;	0usec	0usec	0.0%

Linux first touch policy

- Memory is assigned to NUMA domains
  - not during the (default) allocation
  - but when the memory is being touched by the first time
- The NUMA domain that will get the memory assigned as local memory, is therefore the domain from where the corresponding thread touched the memory for the first time

Let’s fix it!

Intel VTune Amplifier

r_hpc_baseline

HPC Performance Characterization HPC Performance Characterization

INTEL VTUNE AMPLIFIER 2019

Analysis Configuration Collection Log Summary Bottom-up

Elapsed Time: 7.490s

SP GFLOPS:	0.000
DP GFLOPS:	3.818
x87 GFLOPS:	0.000
CPI Rate:	10.039
Average CPU Frequency:	2.4 GHz
Total Thread Count:	96

Effective Physical Core Utilization: 61.1% (29.339 out of 48)

Effective Logical Core Utilization: 60.8% (58.327 out of 96)

Serial Time (outside parallel regions): 2.872s (38.3%)

Top Serial Hotspots (outside parallel regions)

This section lists the loops and functions executed serially in the master thread outside of any OpenMP region and consuming the most CPU time. Improve overall application performance by optimizing or parallelizing these hotspot functions. Since the Serial Time metric includes the Wait time of the master thread, it may significantly exceed the aggregated CPU time in the table.

Function	Module	Serial CPU Time
[Loop at line 462 in checkSTREAMresults]	stream.x	2.140s
[vmlinux]	vmlinux	0.692s
func@0x7d340	libittnotify_collector.so	0.005s
strcmp	ld-2.17.so	0.005s

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

Parallel Region Time: 4.618s (61.7%)

Effective CPU Utilization Histogram

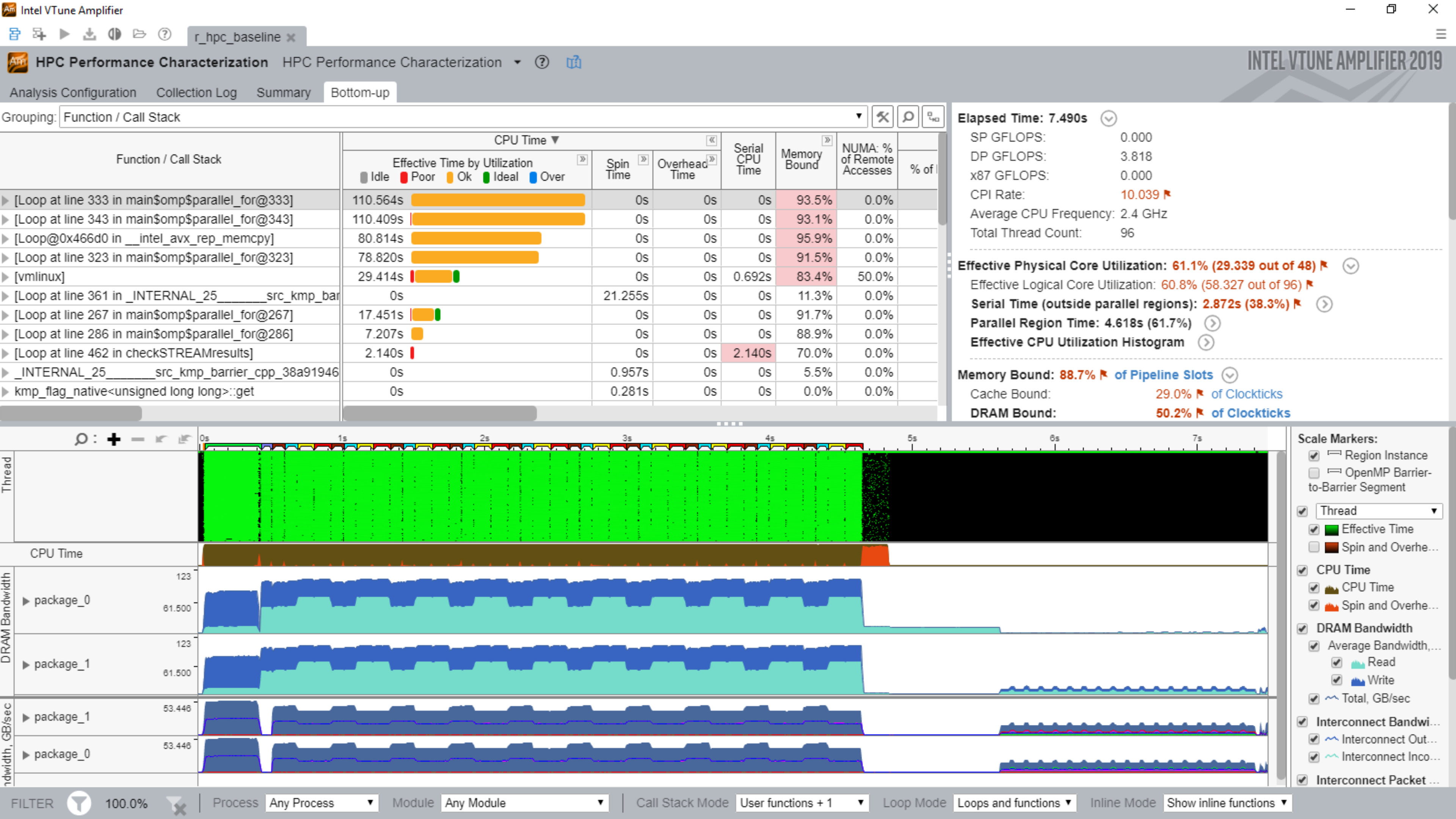
Memory Bound: 88.7% of Pipeline Slots

Cache Bound:	29.0%	of Clockticks
DRAM Bound:	**50.2%**	**of Clockticks**
DRAM Bandwidth Bound:	56.2%	of Elapsed Time
NUMA: % of Remote Accesses:	0.3%	

Bandwidth Utilization Histogram

Vectorization: 100.0% of Packed FP Operations

Instruction Mix:



Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	CPU Time: Spin Time	CPU Time: Overhead Time	Serial CPU Time	Memory Bound	NUMA: % of Remote Accesses	% of
[Loop at line 333 in mainompparallel_for@333]	110.564s	0s	0s	0s	93.5%	0.0%	
[Loop at line 343 in mainompparallel_for@343]	110.409s	0s	0s	0s	93.1%	0.0%	
[Loop@0x466d0 in __intel_avx_rep_memcpy]	80.814s	0s	0s	0s	95.9%	0.0%	
[Loop at line 323 in mainompparallel_for@323]	78.820s	0s	0s	0s	91.5%	0.0%	
[vmlinux]	29.414s	0s	0s	0.692s	83.4%	50.0%	
[Loop at line 361 in _INTERNAL_25_______src_kmp_bar	0s	21.255s	0s	0s	11.3%	0.0%	
[Loop at line 267 in mainompparallel_for@267]	17.451s	0s	0s	0s	91.7%	0.0%	
[Loop at line 286 in mainompparallel_for@286]	7.207s	0s	0s	0s	88.9%	0.0%	
[Loop at line 462 in checkSTREAMresults]	2.140s	0s	0s	2.140s	70.0%	0.0%	
_INTERNAL_25_______src_kmp_barrier_cpp_38a91946	0s	0.957s	0s	0s	5.5%	0.0%	
kmp_flag_native<unsigned long long>::get	0s	0.281s	0s	0s	0.0%	0.0%	

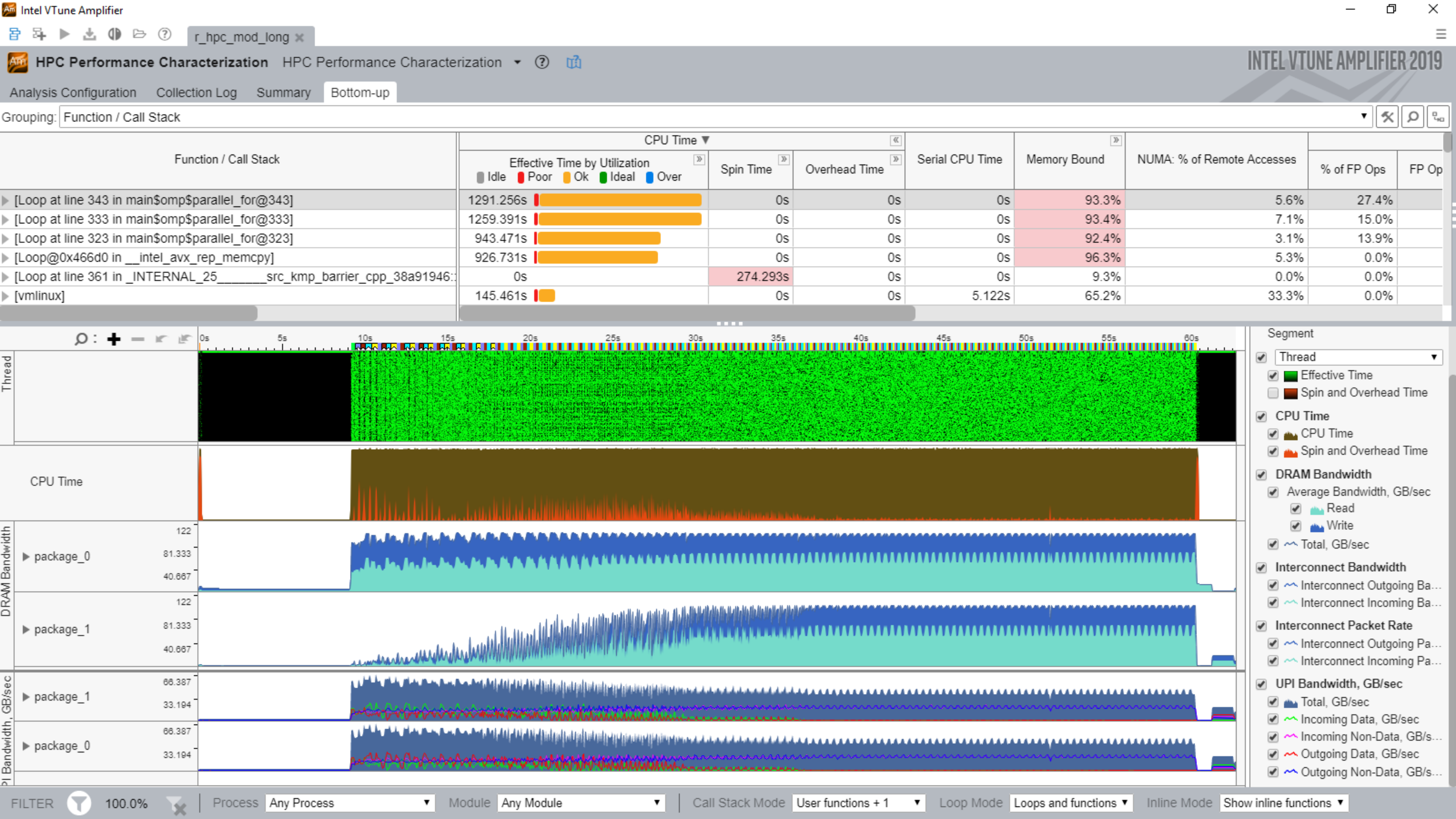
Fine, but what about the mem BW increase on socket #2?

Linux Kernel page migration

- New in RHEL 7 / SLES 12
- Default configuration is ON
- Introduces background noise, bad for benchmarking

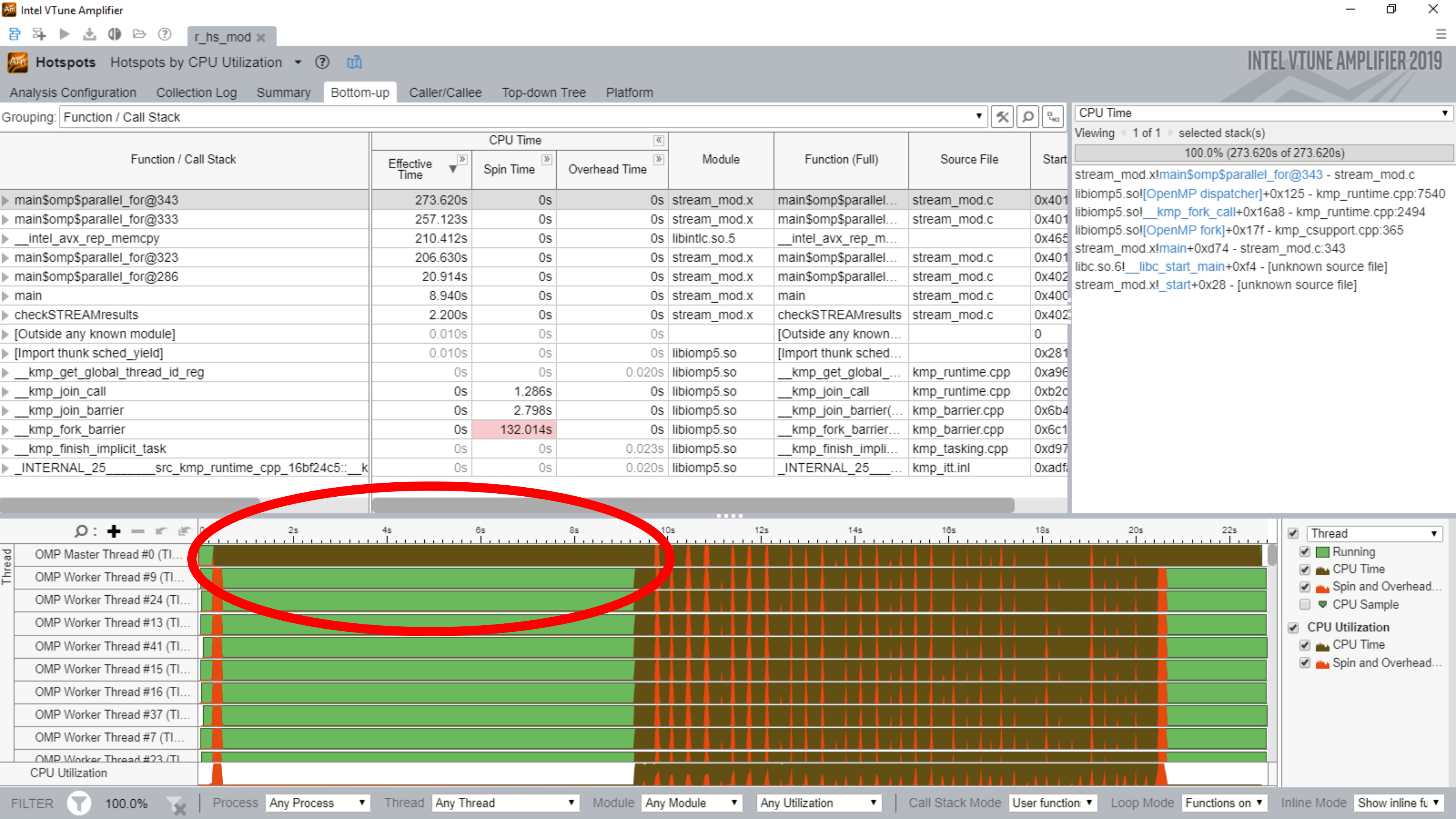
```
$ cat /proc/sys/kernel/numa_balancing
```

What if we would increase the runtime from 10 iterations to 100?

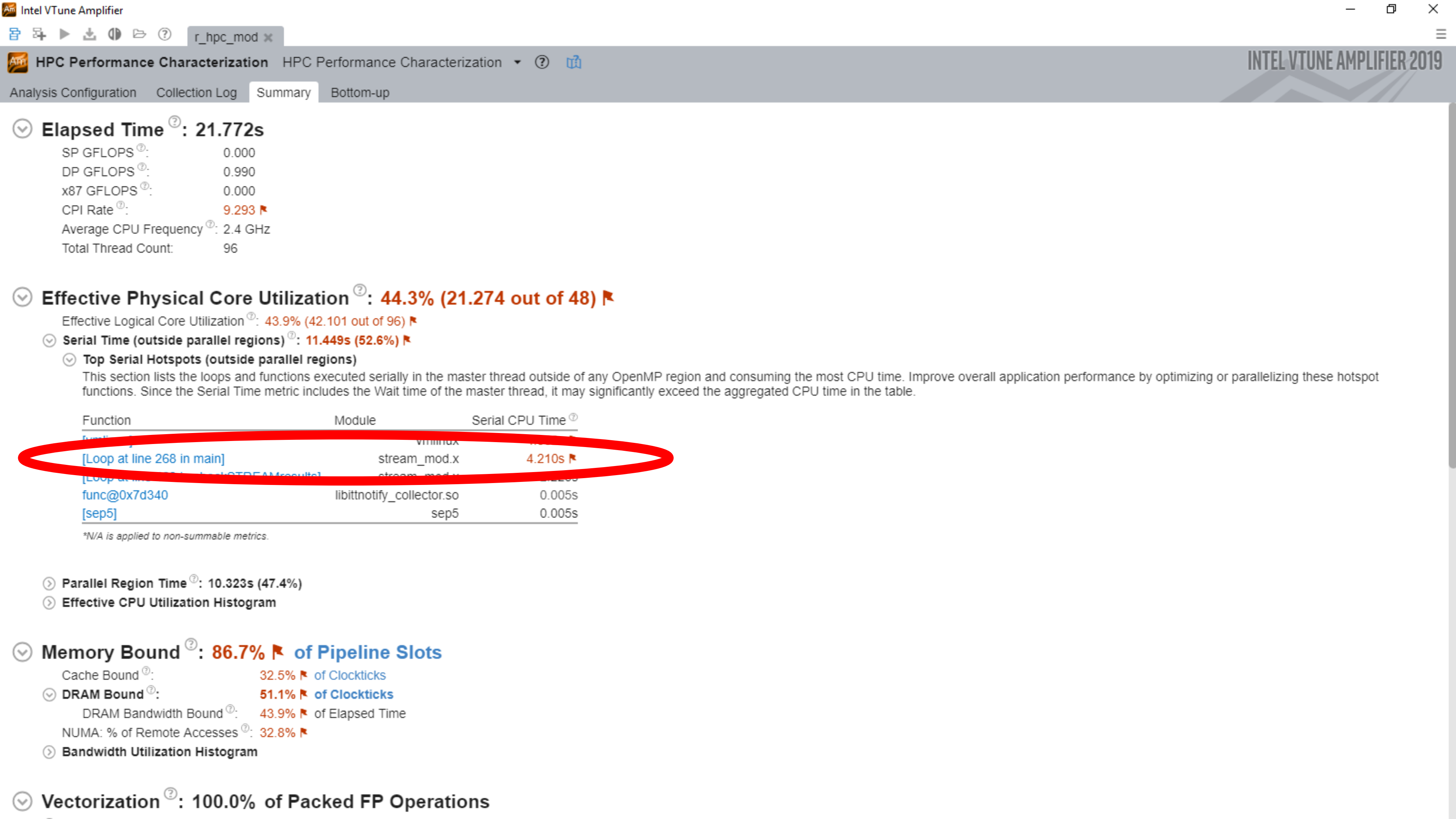

Intel VTune Amplifier
r_hpc_mod_long
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Grouping: Function / Call Stack
Function / Call Stack
CPU Time
Effective Time by Utilization
Idle Poor Ok Ideal Over
Spin Time
Overhead Time
Serial CPU Time
Memory Bound
NUMA: % of Remote Accesses
% of FP Ops
FP Op
[Loop at line 343 in mainompparallel_for@343] 1291.256s 0s 0s 0s 93.3% 5.6% 27.4%
[Loop at line 333 in mainompparallel_for@333] 1259.391s 0s 0s 0s 93.4% 7.1% 15.0%
[Loop at line 323 in mainompparallel_for@323] 943.471s 0s 0s 0s 92.4% 3.1% 13.9%
[Loop@0x466d0 in __intel_avx_rep_memcpy] 926.731s 0s 0s 0s 96.3% 5.3% 0.0%
[Loop at line 361 in _INTERNAL_25_______src_kmp_barrier_cpp_38a91946: 0s 274.293s 0s 0s 9.3% 0.0% 0.0%
[vmlinux] 145.461s 0s 0s 5.122s 65.2% 33.3% 0.0%
0s 5s 10s 15s 20s 25s 30s 35s 40s 45s 50s 55s 60s
Thread
CPU Time
DRAM Bandwidth
package_0
122
81.333
40.667
package_1
122
81.333
40.667
package_1
66.387
33.194
package_0
66.387
33.194
Segment
Thread
Effective Time
Spin and Overhead Time
CPU Time
CPU Time
Spin and Overhead Time
DRAM Bandwidth
Average Bandwidth, GB/sec
Read
Write
Total, GB/sec
Interconnect Bandwidth
Interconnect Outgoing Ba...
Interconnect Incoming Ba...
Interconnect Packet Rate
Interconnect Outgoing Pa...
Interconnect Incoming Pa...
UPI Bandwidth, GB/sec
Total, GB/sec
Incoming Data, GB/sec
Incoming Non-Data, GB/s...
Outgoing Data, GB/sec
Outgoing Non-Data, GB/s...
FILTER 100.0%
Process Any Process
Module Any Module
Call Stack Mode User functions + 1
Loop Mode Loops and functions
Inline Mode Show inline functions

Some earlier problem indicators we missed?

Besides being fooled by filters and zoom



Function / Call Stack	CPU Time: Effective Time	CPU Time: Spin Time	CPU Time: Overhead Time	Module	Function (Full)	Source File	Start
mainompparallel_for@343	273.620s	0s	0s	stream_mod.x	mainompparallel...	stream_mod.c	0x401
mainompparallel_for@333	257.123s	0s	0s	stream_mod.x	mainompparallel...	stream_mod.c	0x401
__intel_avx_rep_memcpy	210.412s	0s	0s	libintlc.so.5	__intel_avx_rep_m...		0x465
mainompparallel_for@323	206.630s	0s	0s	stream_mod.x	mainompparallel...	stream_mod.c	0x401
mainompparallel_for@286	20.914s	0s	0s	stream_mod.x	mainompparallel...	stream_mod.c	0x402
main	8.940s	0s	0s	stream_mod.x	main	stream_mod.c	0x400
checkSTREAMresults	2.200s	0s	0s	stream_mod.x	checkSTREAMresults	stream_mod.c	0x402
[Outside any known module]	0.010s	0s	0s		[Outside any known...		0
[Import thunk sched_yield]	0.010s	0s	0s	libiomp5.so	[Import thunk sched...		0x281
__kmp_get_global_thread_id_reg	0s	0s	0.020s	libiomp5.so	__kmp_get_global_...	kmp_runtime.cpp	0xa96
__kmp_join_call	0s	1.286s	0s	libiomp5.so	__kmp_join_call	kmp_runtime.cpp	0xb2c
__kmp_join_barrier	0s	2.798s	0s	libiomp5.so	__kmp_join_barrier(...	kmp_barrier.cpp	0x6b4
__kmp_fork_barrier	0s	132.014s	0s	libiomp5.so	__kmp_fork_barrier...	kmp_barrier.cpp	0x6c1
__kmp_finish_implicit_task	0s	0s	0.023s	libiomp5.so	__kmp_finish_impli...	kmp_tasking.cpp	0xd97
_INTERNAL_25_______src_kmp_runtime_cpp_16bf24c5::__k	0s	0s	0.020s	libiomp5.so	_INTERNAL_25___...	kmp_itt.inl	0xadf


Intel VTune Amplifier
r_hpc_mod
HPC Performance Characterization
HPC Performance Characterization
INTEL VTUNE AMPLIFIER 2019
Analysis Configuration
Collection Log
Summary
Bottom-up
Elapsed Time: 21.772s
SP GFLOPS: 0.000
DP GFLOPS: 0.990
x87 GFLOPS: 0.000
CPI Rate: 9.293
Average CPU Frequency: 2.4 GHz
Total Thread Count: 96
Effective Physical Core Utilization: 44.3% (21.274 out of 48)
Effective Logical Core Utilization: 43.9% (42.101 out of 96)
Serial Time (outside parallel regions): 11.449s (52.6%)
Top Serial Hotspots (outside parallel regions)
This section lists the loops and functions executed serially in the master thread outside of any OpenMP region and consuming the most CPU time. Improve overall application performance by optimizing or parallelizing these hotspot functions. Since the Serial Time metric includes the Wait time of the master thread, it may significantly exceed the aggregated CPU time in the table.
Function
Module
Serial CPU Time
[Loop at line 268 in main]
stream_mod.x
4.210s
func@0x7d340
libittnotify_collector.so
0.005s
[sep5]
sep5
0.005s
*N/A is applied to non-summable metrics.
Parallel Region Time: 10.323s (47.4%)
Effective CPU Utilization Histogram
Memory Bound: 86.7% of Pipeline Slots
Cache Bound: 32.5% of Clockticks
DRAM Bound: 51.1% of Clockticks
DRAM Bandwidth Bound: 43.9% of Elapsed Time
NUMA: % of Remote Accesses: 32.8%
Bandwidth Utilization Histogram
Vectorization: 100.0% of Packed FP Operations

Intel VTune Amplifier

r_hpc_mod

HPC Performance Characterization | HPC Performance Characterization | INTEL VTUNE AMPLIFIER 2019

Analysis Configuration | Collection Log | Summary | Bottom-up

[Loop at line 462 in checkSTREAMresults]	stream_mod.x	2.225s
func@0x7d340	libittnotify_collector.so	0.005s
[sep5]	sep5	0.005s

N/A is applied to non-summable metrics.

Parallel Region Time: 10.323s (47.4%)

Effective CPU Utilization Histogram

Memory Bound: 86.7% of Pipeline Slots

Cache Bound: 32.5% of Clockticks

DRAM Bound: 51.1% of Clockticks

DRAM Bandwidth Bound: 43.9% of Elapsed Time

NUMA: % of Remote Accesses: 32.8%

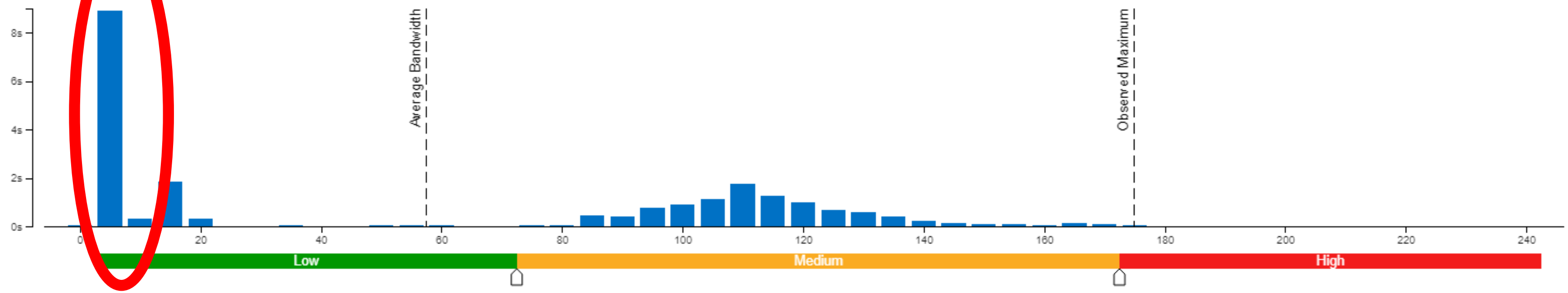
Bandwidth Utilization Histogram

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

Better, Faster Application Performance Snapshot

Intel® VTune™ Amplifier – Performance Profiler

Better Answers

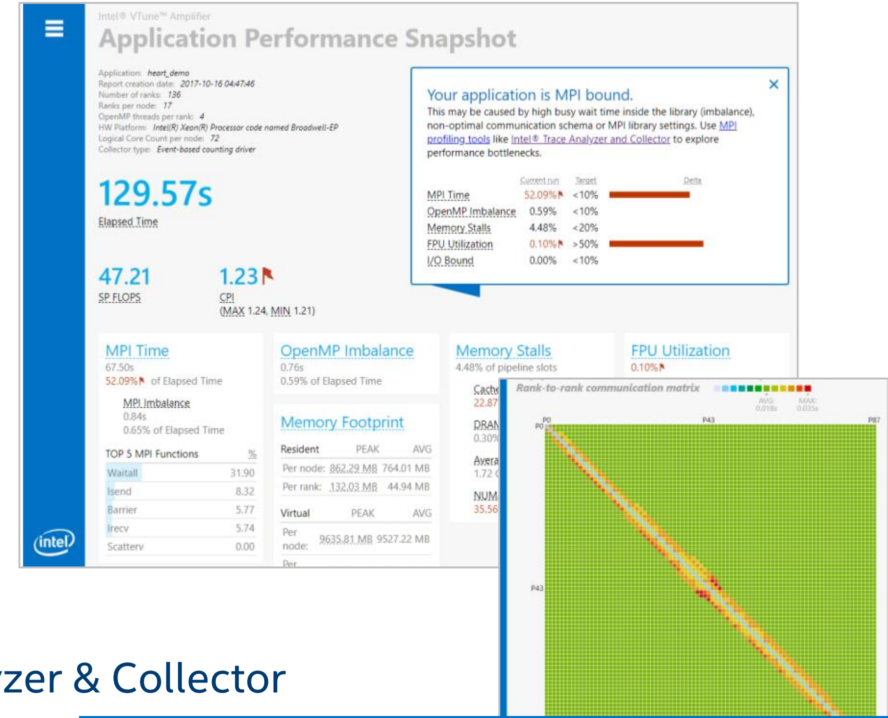
- CPU utilization analysis of physical cores

Less Overhead

- Lower MPI trace overhead
  & faster result processing
- New data selection & pause/resume
  let you focus on useful data

Easier to Use

- Visualize rank-to-rank &
  node-to-node MPI communications
- Easily configure profiling for Intel® Trace Analyzer & Collector



Free Download: intel.com/performance-snapshot

Intel® VTune™ Amplifier

Application Performance Snapshot

Application: *stream.x*
Report creation date: *2019-05-09 08:12:19*
OpenMP threads: *96*
HW Platform: *Intel(R) Xeon(R) Processor code named Cascadelake*
Logical Core Count per node: *96*
Collector type: *Event-based counting driver*

7.49s Elapsed Time

6.38 CPI

13.19 SP GFLOPS

Your application is memory bound.

Use memory access analysis tools like Intel® VTune™ Amplifier for a detailed metric breakdown by memory hierarchy, memory bandwidth, and correlation by memory objects.

	Current run	Target	Delta
Serial Time	37.38%	<15%	
OpenMP Imbalance	0.95%	<10%	
Memory Stalls	88.30%	<20%	
Vectorization	99.90%	>70%	

Serial Time
2.76s
37.38% of Elapsed Time

Memory Footprint
Resident total: 22913.84 MB
Virtual total: 29479.83 MB

OpenMP Imbalance
0.07s
0.95% of Elapsed Time

Memory Stalls
88.30% of pipeline slots

Cache Stalls
34.00% of cycles

DRAM Stalls
57.00% of cycles

DRAM Bandwidth
AVG 125.88 GB/s

NUMA
0.90% of remote accesses

Vectorization
99.90%of Packed FP Operations

Instruction Mix:

SP FLOPs
0.00% of uOps

DP FLOPs
14.90% of uOps
Packed: 100.00% from DP FP
128-bit: 0.00%
256-bit: 100.00%
512-bit: 0.00%
Scalar: 0.00% from DP FP

Non-FP
85.10% of uOps

FP Arith/Mem Rd Instr. Ratio
0.23

FP Arith/Mem Wr Instr. Ratio
0.82

Intel® VTune™ Amplifier

Application Performance Snapshot

Application: *stream_mod.x*
Report creation date: *2019-05-09 08:12:33*
OpenMP threads: *96*
HW Platform: *Intel(R) Xeon(R) Processor code named Cascadelake*
Logical Core Count per node: *96*
Collector type: *Event-based counting driver*

21.81s
Elapsed Time

6.29
CPI

3.19
SP GFLOPS

Your application is memory bound.

Use memory access analysis tools like Intel® VTune™ Amplifier for a detailed metric breakdown by memory hierarchy, memory bandwidth, and correlation by memory objects.

	Current run	Target	Delta
Serial Time	52.18%	<15%	
OpenMP Imbalance	3.70%	<10%	
Memory Stalls	85.10%	<20%	
Vectorization	99.80%	>70%	

Serial Time
11.32s
52.18% of Elapsed Time

Memory Footprint
Resident total: 22907.89 MB
Virtual total: 29479.82 MB

OpenMP Imbalance
0.81s
3.70% of Elapsed Time

Memory Stalls
85.10% of pipeline slots

Cache Stalls
35.50% of cycles

DRAM Stalls
50.50% of cycles

DRAM Bandwidth
AVG 59.68 GB/s

NUMA
29.20% of remote accesses

Vectorization
99.80%of Packed FP Operations

Instruction Mix:

SP FLOPs
0.00% of uOps

DP FLOPs
3.60% of uOps
Packed: 100.00% from DP FP
128-bit: 0.00%
256-bit: 100.00%
512-bit: 0.00%
Scalar: 0.00% from DP FP

Non-FP
96.40% of uOps

FP Arith/Mem Rd Instr. Ratio
0.09

FP Arith/Mem Wr Instr. Ratio
0.41

Legal Disclaimer & Optimization Notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

intel®

Software