Optimisation et modèle *roofline*

Luca Cirrottola (INRIA)

luca.cirrottola@inria.fr

Bordeaux INP ENSEIRB-MATMECA, Université de Bordeaux

Automne 2023

Usage

- **[Rappel:] Les performances sont beaucoup liées à la localité de la mémoire...**

Storage Area	Register	L1 Cache	L2 Cache	RAM	Swap
Cycles to Access	≤ 1	≈ 3	≈ 14	≈ 240	$\approx 10^7$
Town	Talence	Pessac	Cestas	Toulouse	Mars

Usage

Outils

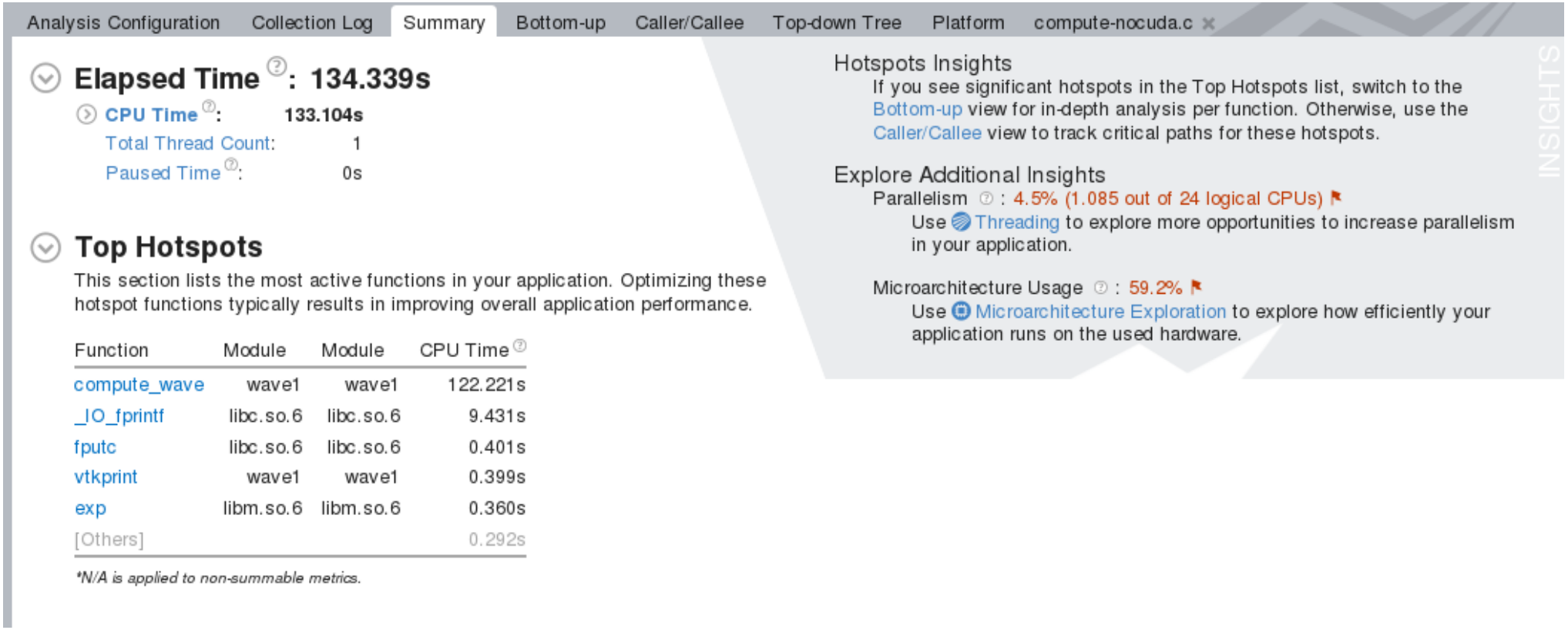
- **Exemples:**
  - gprof
  - Valgrind
  - PAPI
  - Intel VTune Amplifier
  - Intel Advisor
  - Extrae/Paraver [traces]
  - NVIDIA Visual Profiler [GPUs]
  - Running Average Power Limit (RAPL) [energy]
  - ...

Intel Vtune Amplifier XE

Intel Vtune Amplifier XE

- **Basic hotspots**

  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect hotspots -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000



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

Elapsed Time: 134.339s

CPU Time: 133.104s
Total Thread Count: 1
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	Module	CPU Time
compute_wave	wave1	wave1	122.221s
_IO_fprintf	libc.so.6	libc.so.6	9.431s
fputc	libc.so.6	libc.so.6	0.401s
vtkprint	wave1	wave1	0.399s
exp	libm.so.6	libm.so.6	0.360s
[Others]			0.292s

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

INSIGHTS

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 : 4.5% (1.085 out of 24 logical CPUs)

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

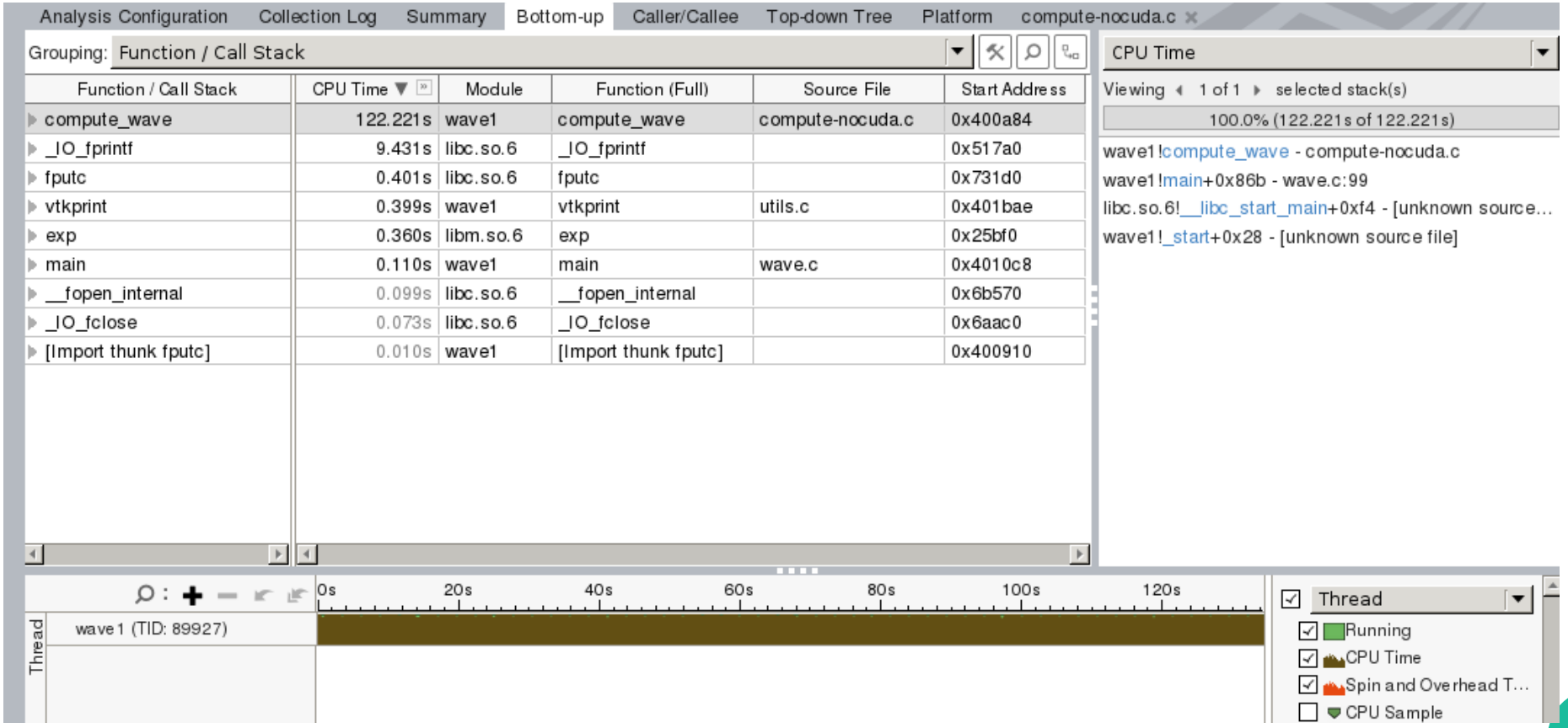
Microarchitecture Usage : 59.2%

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

Intel Vtune Amplifier XE

- **Basic hotspots**

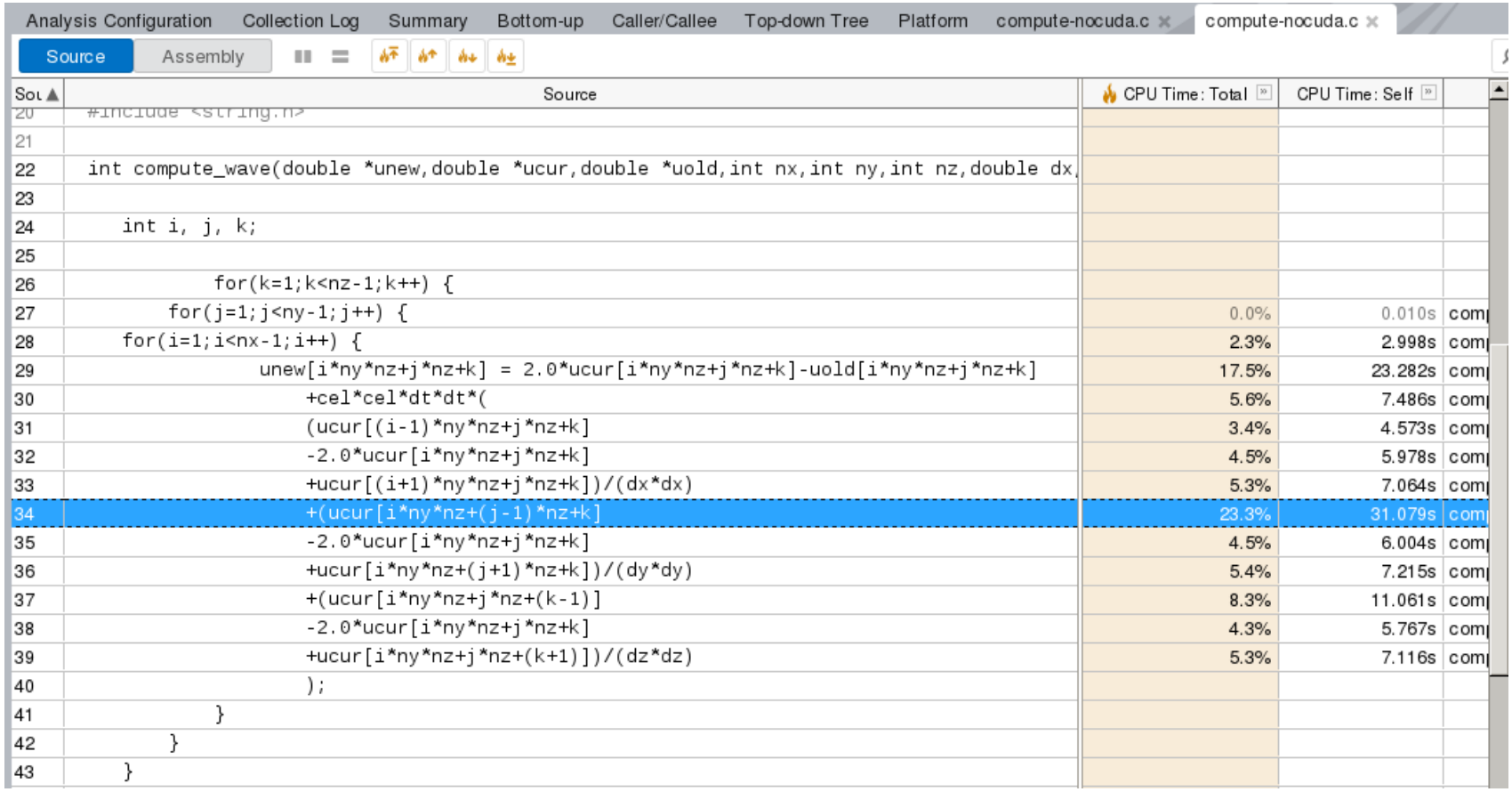
  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect hotspots -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000



Intel Vtune Amplifier XE

- **Basic hotspots**

  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect hotspots -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000

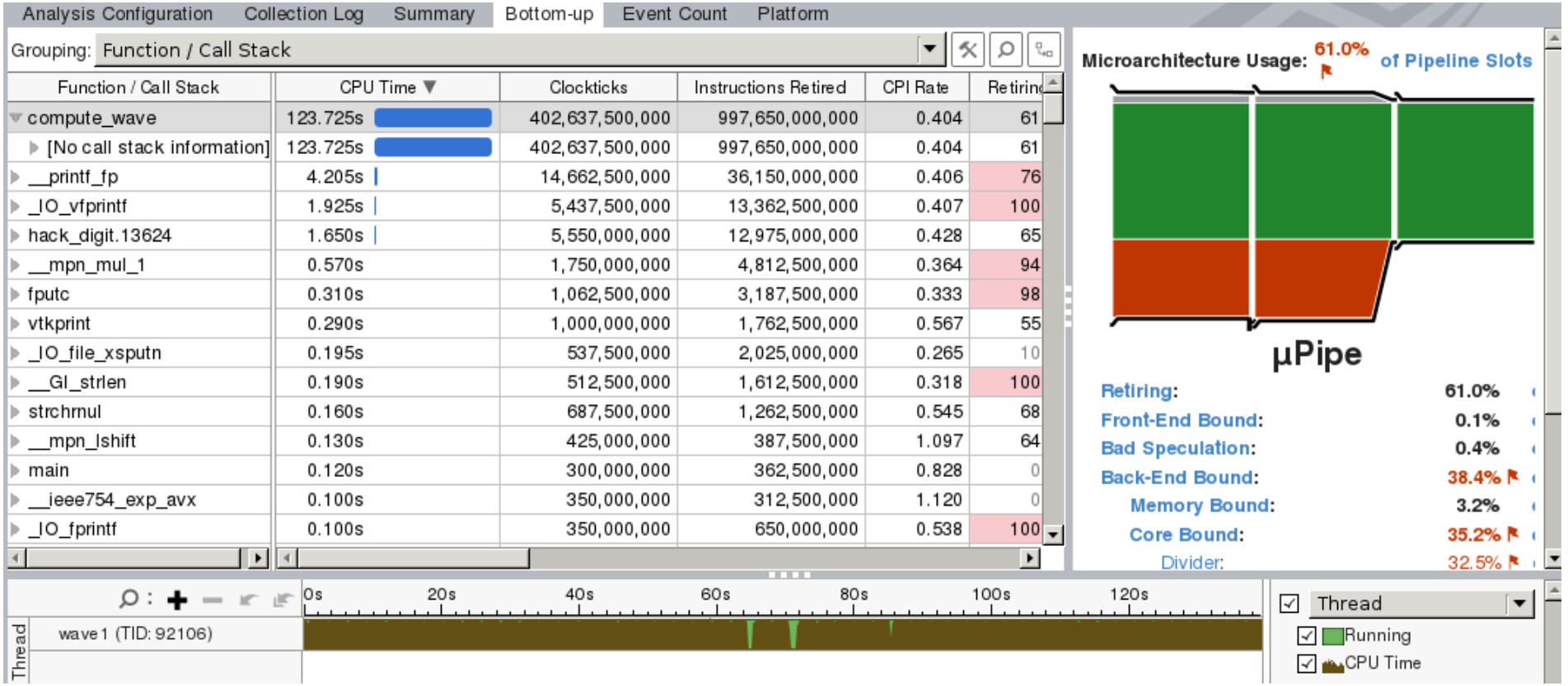


Source	CPU Time: Total	CPU Time: Self
20 #include <string.h>		
21		
22 int compute_wave(double *unew,double *ucur,double *uold,int nx,int ny,int nz,double dx,		
23		
24 int i, j, k;		
25		
26 for(k=1;k<nz-1;k++) {		
27 for(j=1;j<ny-1;j++) {	0.0%	0.010s
28 for(i=1;i<nx-1;i++) {	2.3%	2.998s
29 unew[i*ny*nz+j*nz+k] = 2.0*ucur[i*ny*nz+j*nz+k]-uold[i*ny*nz+j*nz+k]	17.5%	23.282s
30 +cel*cel*dt*dt*(	5.6%	7.486s
31 (ucur[(i-1)*ny*nz+j*nz+k]	3.4%	4.573s
32 -2.0*ucur[i*ny*nz+j*nz+k]	4.5%	5.978s
33 +ucur[(i+1)*ny*nz+j*nz+k])/(dx*dx)	5.3%	7.064s
34 +(ucur[i*ny*nz+(j-1)*nz+k]	23.3%	31.079s
35 -2.0*ucur[i*ny*nz+j*nz+k]	4.5%	6.004s
36 +ucur[i*ny*nz+(j+1)*nz+k])/(dy*dy)	5.4%	7.215s
37 +(ucur[i*ny*nz+j*nz+(k-1)]	8.3%	11.061s
38 -2.0*ucur[i*ny*nz+j*nz+k]	4.3%	5.767s
39 +ucur[i*ny*nz+j*nz+(k+1)])/(dz*dz)	5.3%	7.116s
40);		
41 }		
42 }		
43 }		

Intel Vtune Amplifier XE

- **Memory Access – General Exploration**

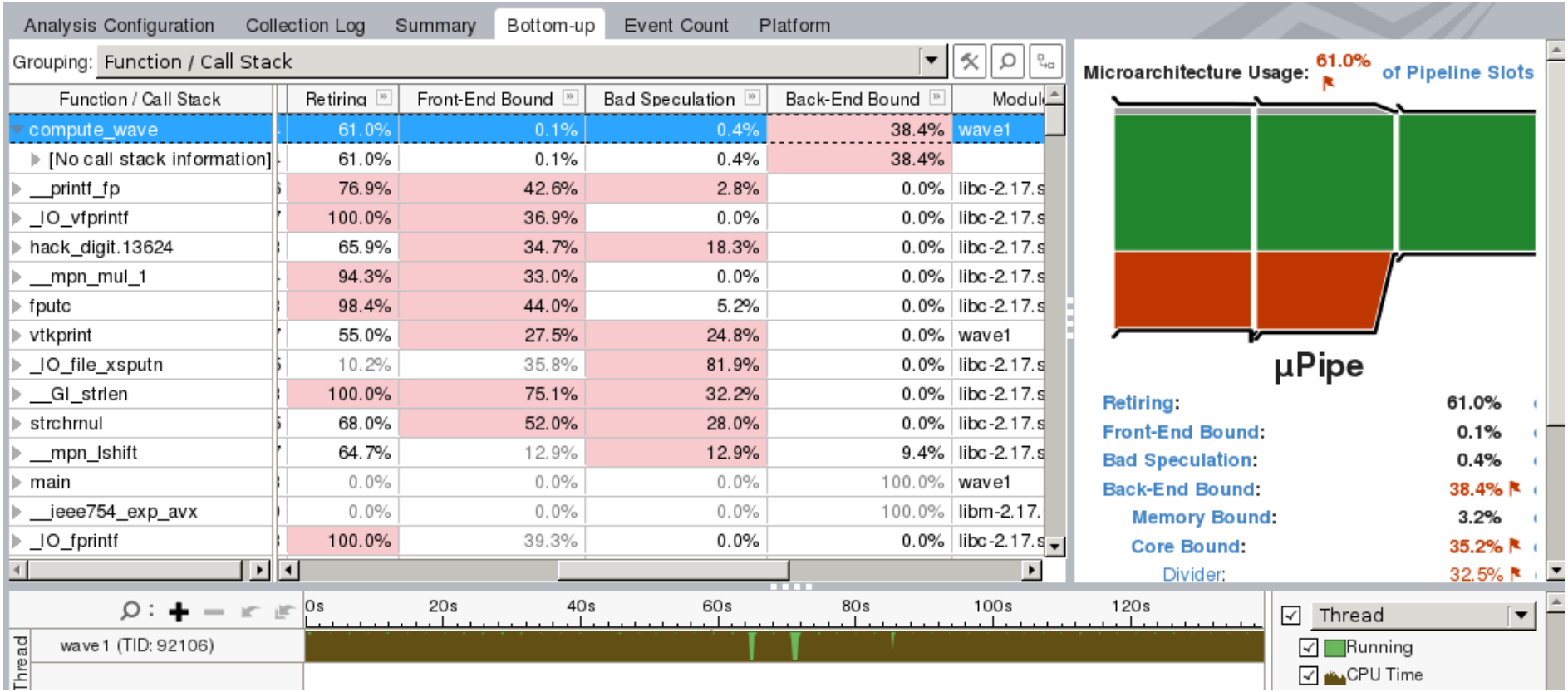
  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect general-exploration -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000



Intel Vtune Amplifier XE

- **Memory Access – General Exploration**

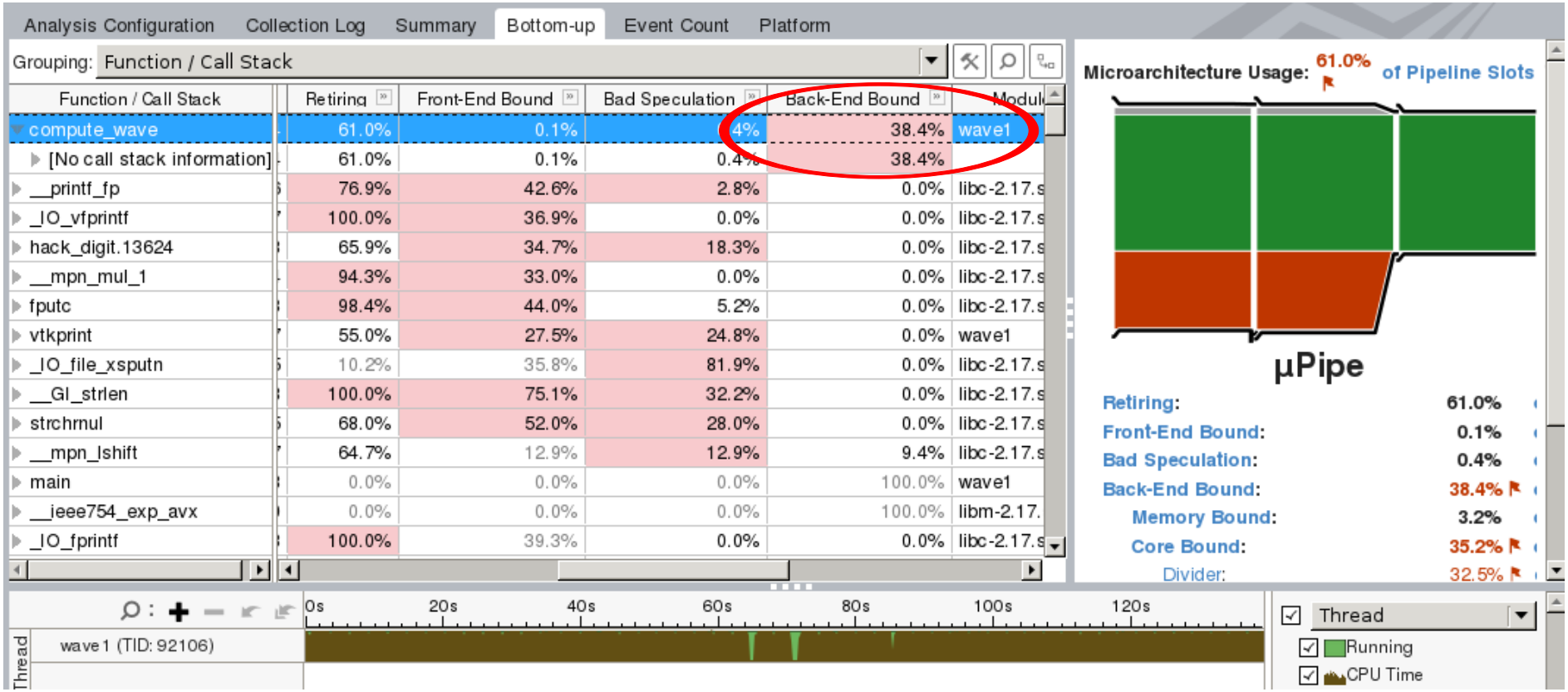
  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect general-exploration -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000



Intel Vtune Amplifier XE

- **Memory Access – General Exploration**

  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect general-exploration -no-auto-finalize ./wave1 5 5 5 100 100 100 0.0005 5000

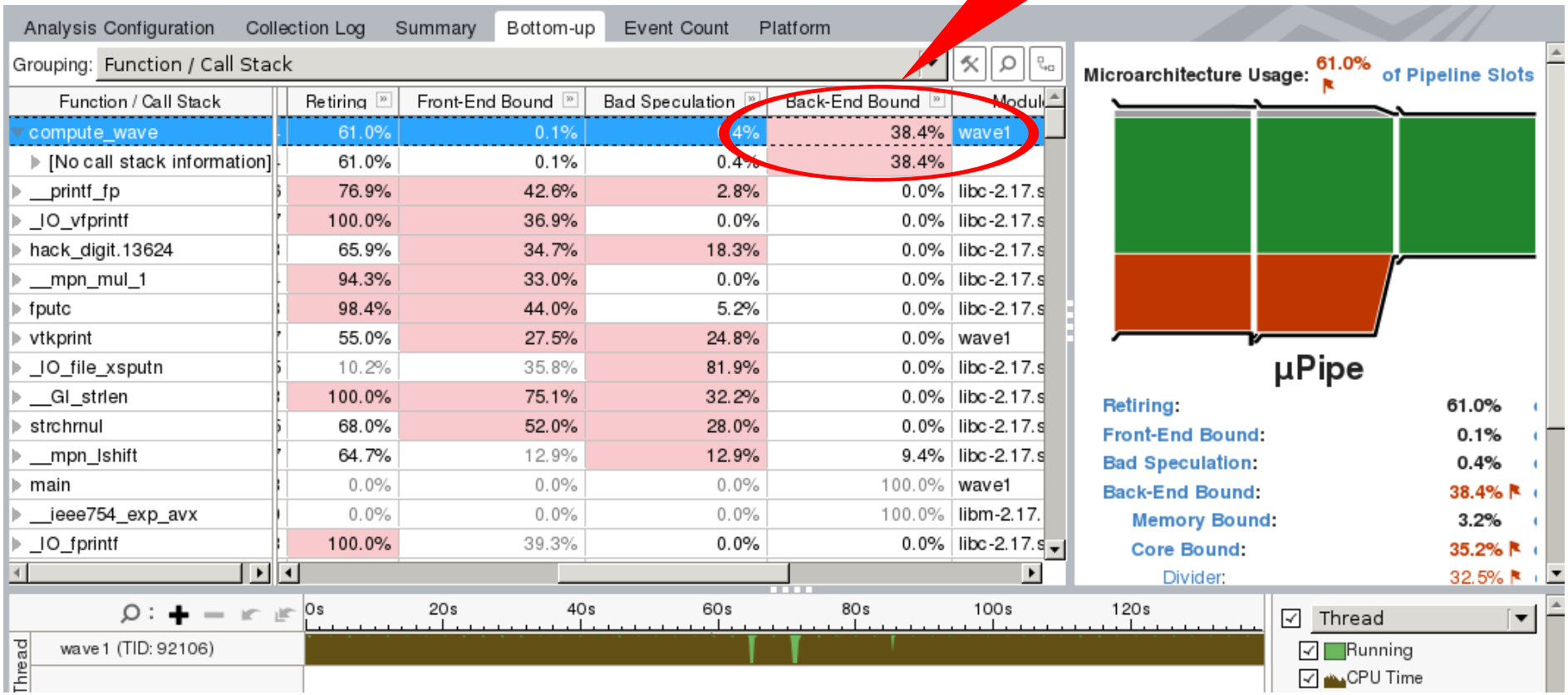


Intel Vtune Amplifier XE

Slower execution
Long latency operation like divides or memory operations
....

- **Memory Access – General Exploration**

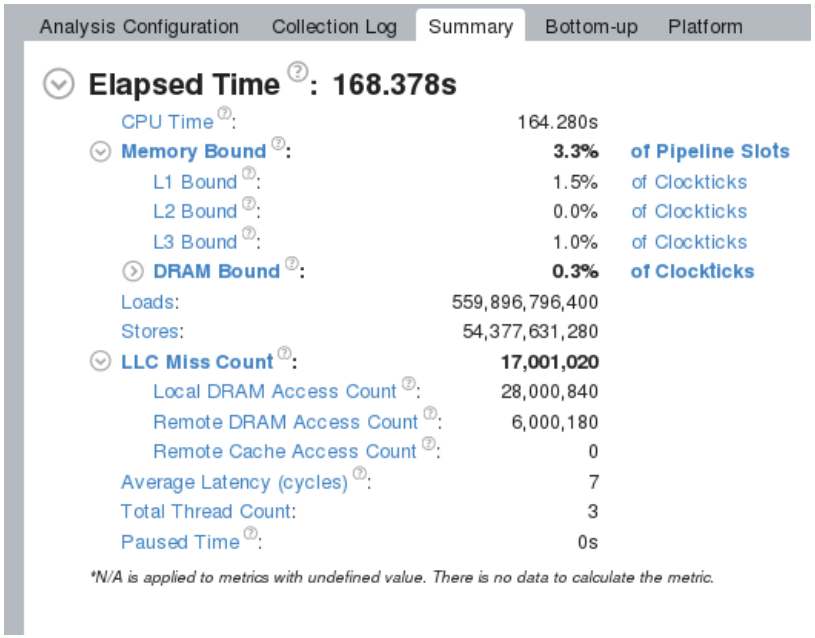
>/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect general-exploration -no-auto-finalize ./wave1 5 100 100 100 0.0005 5000



Intel Vtune Amplifier XE

- **Memory Access**

  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect memory-access -knob dram-bandwidth-limits=false -knob analyze-mem-objects=true -data-limit=0 -- ./wave1 5 5 5 100 100 100 0.0005 5000

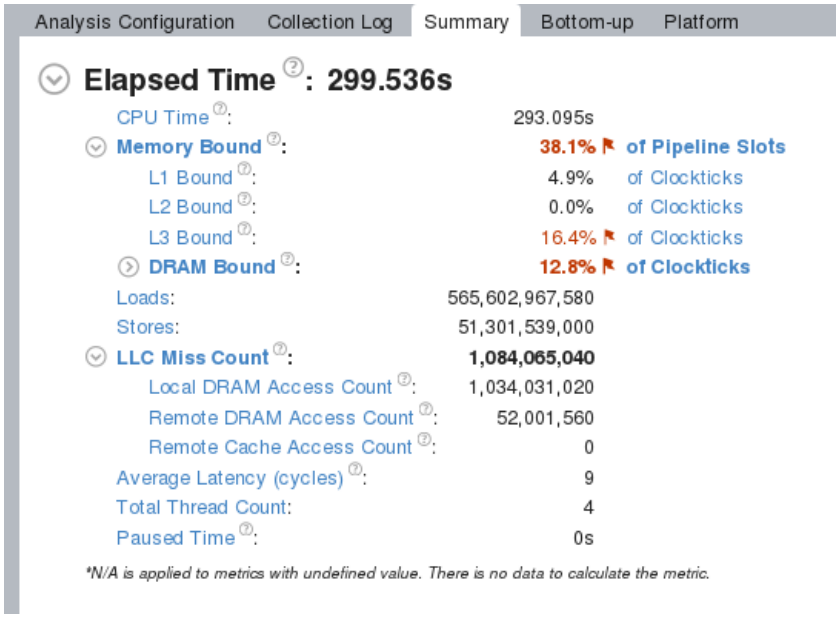


Analysis Configuration | Collection Log | Summary | Bottom-up | Platform

Elapsed Time: 168.378s

CPU Time:	164.280s	
Memory Bound:	**3.3%**	**of Pipeline Slots**
L1 Bound:	1.5%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	1.0%	of Clockticks
DRAM Bound:	**0.3%**	**of Clockticks**
Loads:	559,896,796,400	
Stores:	54,377,631,280	
LLC Miss Count:	**17,001,020**	
Local DRAM Access Count:	28,000,840	
Remote DRAM Access Count:	6,000,180	
Remote Cache Access Count:	0	
Average Latency (cycles):	7	
Total Thread Count:	3	
Paused Time:	0s	

*N/A is applied to metrics with undefined value. There is no data to calculate the metric.



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

Elapsed Time: 299.536s

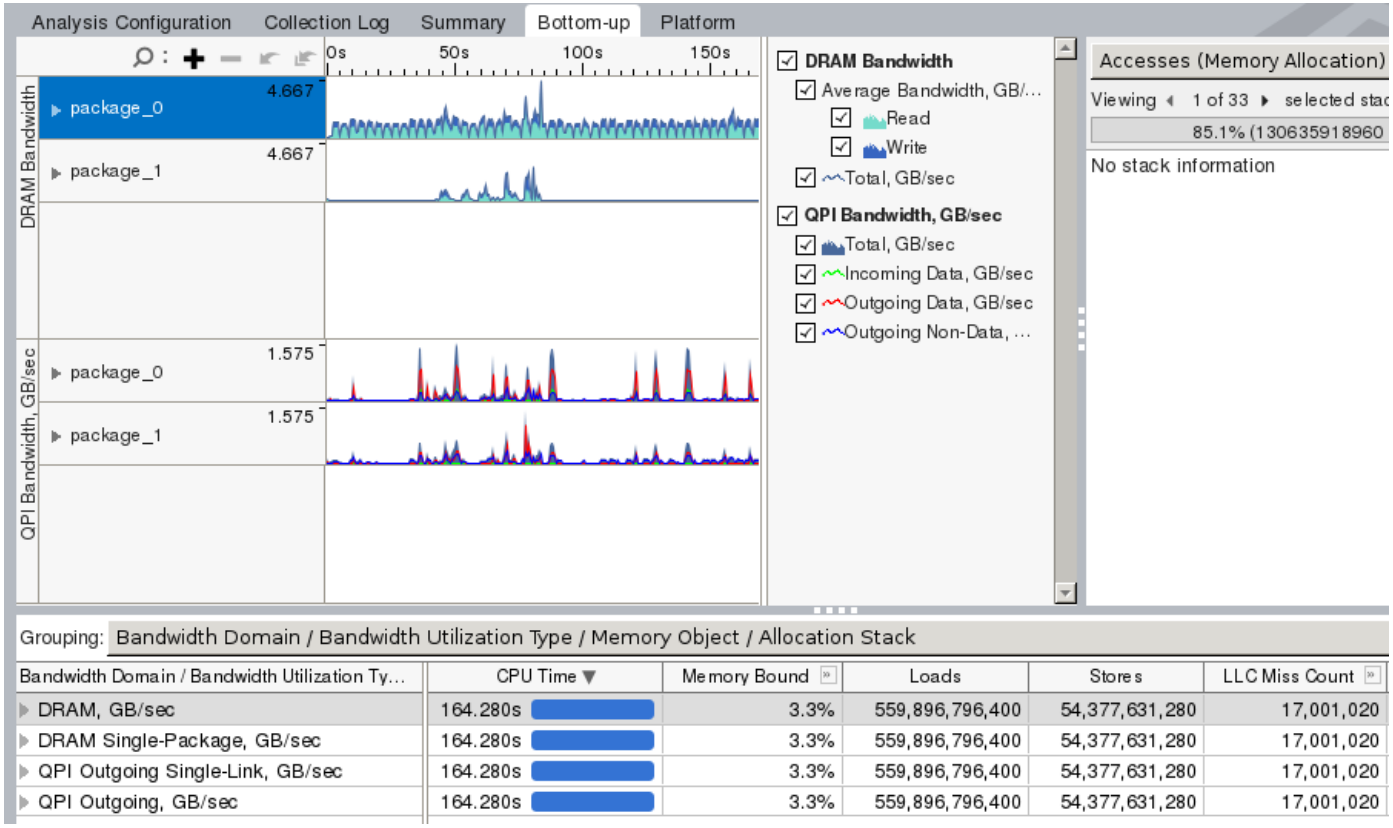
CPU Time:	293.095s	
Memory Bound:	**38.1%**	**of Pipeline Slots**
L1 Bound:	4.9%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	16.4%	of Clockticks
DRAM Bound:	**12.8%**	**of Clockticks**
Loads:	565,602,967,580	
Stores:	51,301,539,000	
LLC Miss Count:	**1,084,065,040**	
Local DRAM Access Count:	1,034,031,020	
Remote DRAM Access Count:	52,001,560	
Remote Cache Access Count:	0	
Average Latency (cycles):	9	
Total Thread Count:	4	
Paused Time:	0s	

*N/A is applied to metrics with undefined value. There is no data to calculate the metric.

Intel Vtune Amplifier XE

- **Memory Access**

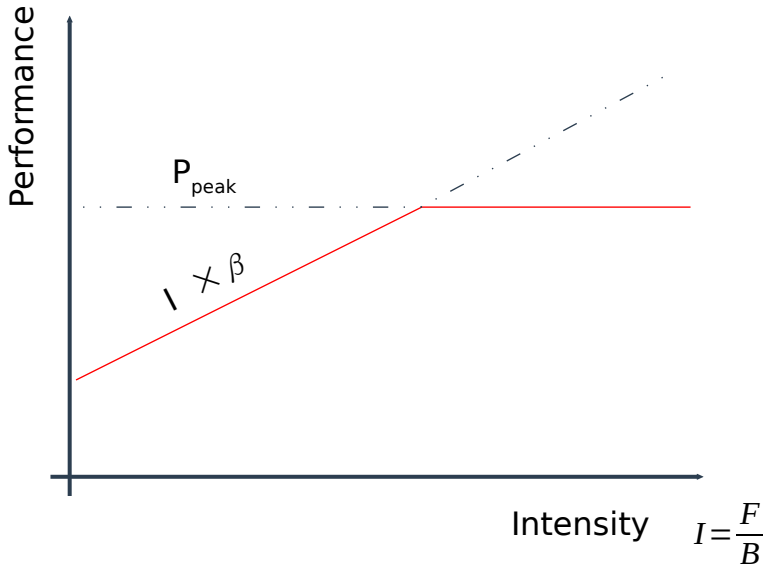
  >/cm/shared/apps/intel/composer_xe/2019.0-117/vtune_amplifier/bin64/amplxe-cl -collect memory-access -knob dram-bandwidth-limits=false -knob analyze-mem-objects=true -data-limit=0 -- ./wave1 5 5 5 100 100 100 0.0005 5000



Roofline Model

Roofline model

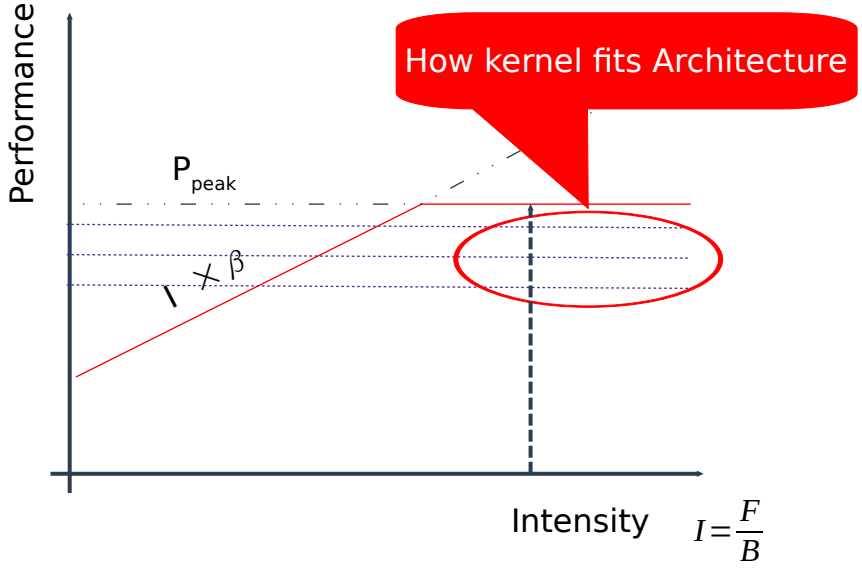
- **Roofline Model (by Williams et.al. 2009)**
  - **P: performance (in FLOP/s)**
  - **F : number of operations performed by kernel or application (in FLOP)**
  - **B : number of bytes of memory transferred (in bytes)**
  - **I: Arithmetic Intensity (in FLOP/byte)**
  - **β: Streaming bandwitdh (in byte/s)**



- **Roofs:**
  - **Maximum processing capability**
    - **Peak performance P_{peak}**
  - **Peak bandwidth**
    - **Intensity * bandwidth (stream benchmark)**

Roofline model

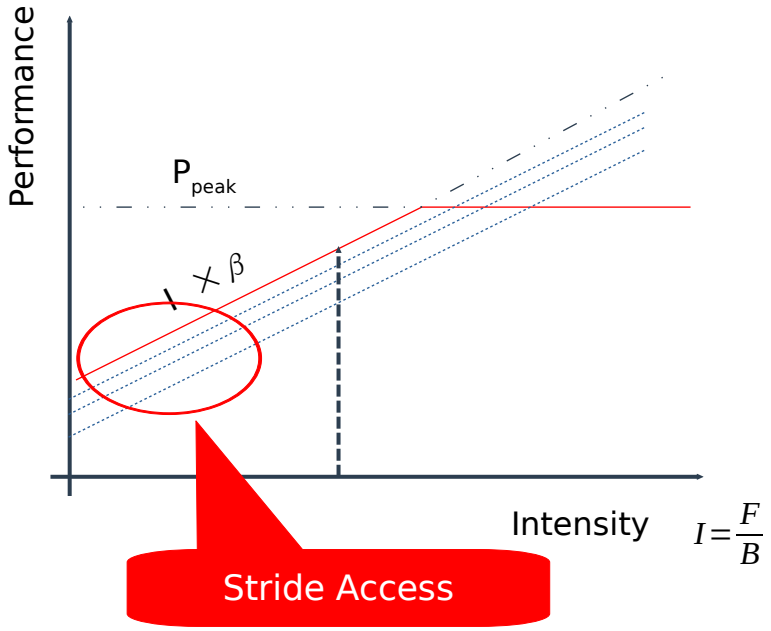
- **Roofline Model (by Williams et.al. 2009)**
  - **P: performance (in FLOP/s)**
  - **F : number of operations performed by kernel or application (in FLOP)**
  - **B : number of bytes of memory transferred (in bytes)**
  - **I: Arithmetic Intensity (in FLOP/byte)**
  - **β: Streaming bandwitdh (in byte/s)**



” Compute-bound”!

Roofline model

- **Roofline Model (by Williams et.al. 2009)**
  - **P: performance (in FLOP/s)**
  - **F : number of operations performed by kernel or application (in FLOP)**
  - **B : number of bytes of memory transferred (in bytes)**
  - **I: Arithmetic Intensity (in FLOP/byte)**
  - **β: Streaming bandwitdh (in byte/s)**



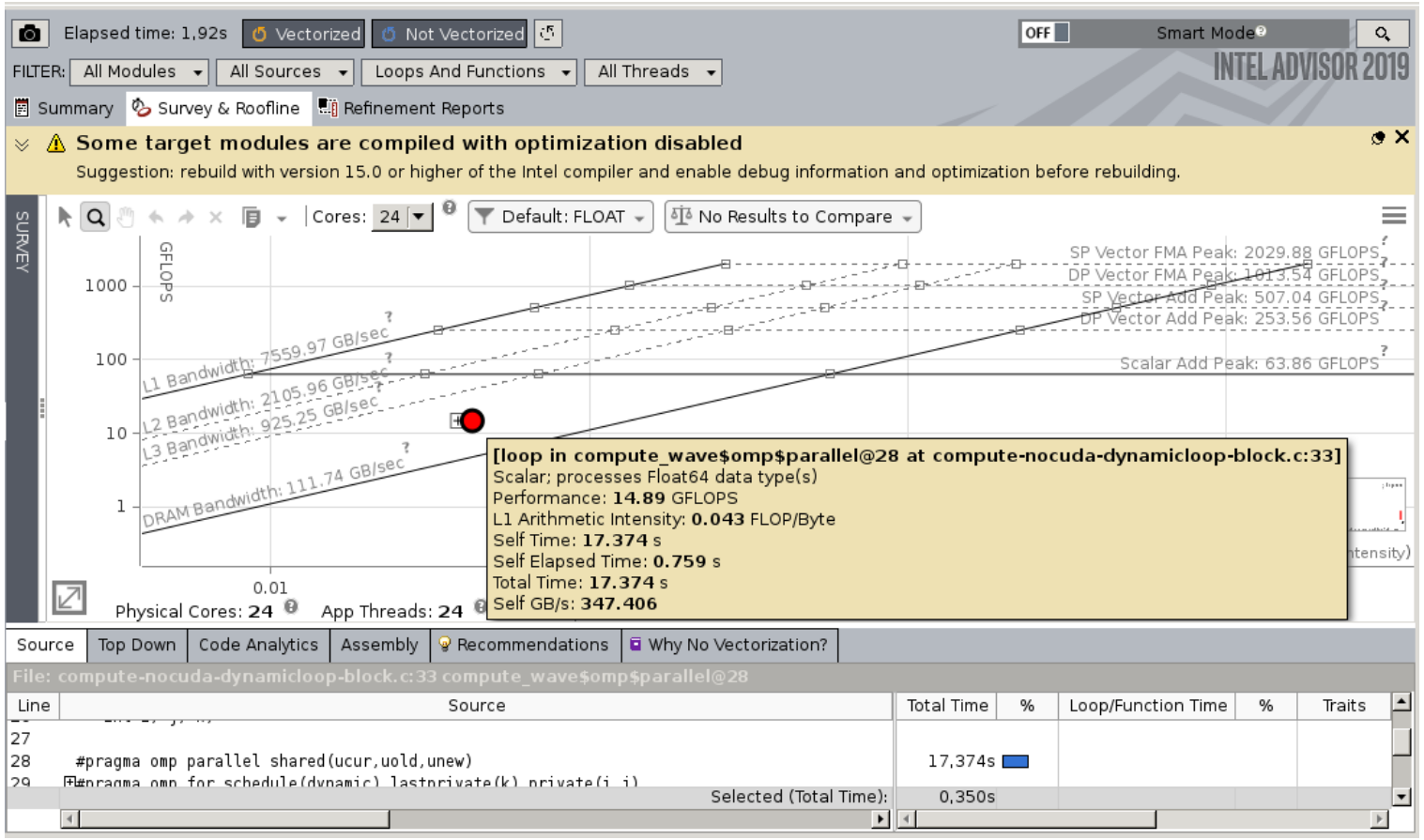
”Memory-bound”!

Intel Vtune Advisor

Intel Vtune Advisor

- **roofline**

>/cm/shared/apps/intel/composer_xe/2019.0-117/advisor/bin64/advixe-cl -collect roofline --project-dir=wave5-dir ./wave5 5 5 5 100 100 100 0.0005 500



… À nos claviers!