Modeling Heterogeneous Computing Performance with Offload Advisor

Vladimir Tsymbal (Intel) - presenter
Cédric Andreolli (Intel)
Zakhar Matveev (Intel)

Association for Computing Machinery
Advancing Computing as a Science & Profession

IWOCL '20, April 27–29, 2020, Munich, Germany

ACM ISBN 978-1-4503-7531-3/20/04.
https://doi.org/10.1145/3388333.3388665

Association for Computing Machinery

Advancing Computing as a Science & Profession

Agenda

- Introduction to Offload Advisor
- Command line tips
- Understanding the performance modelization
- GPU Roofline Analysis

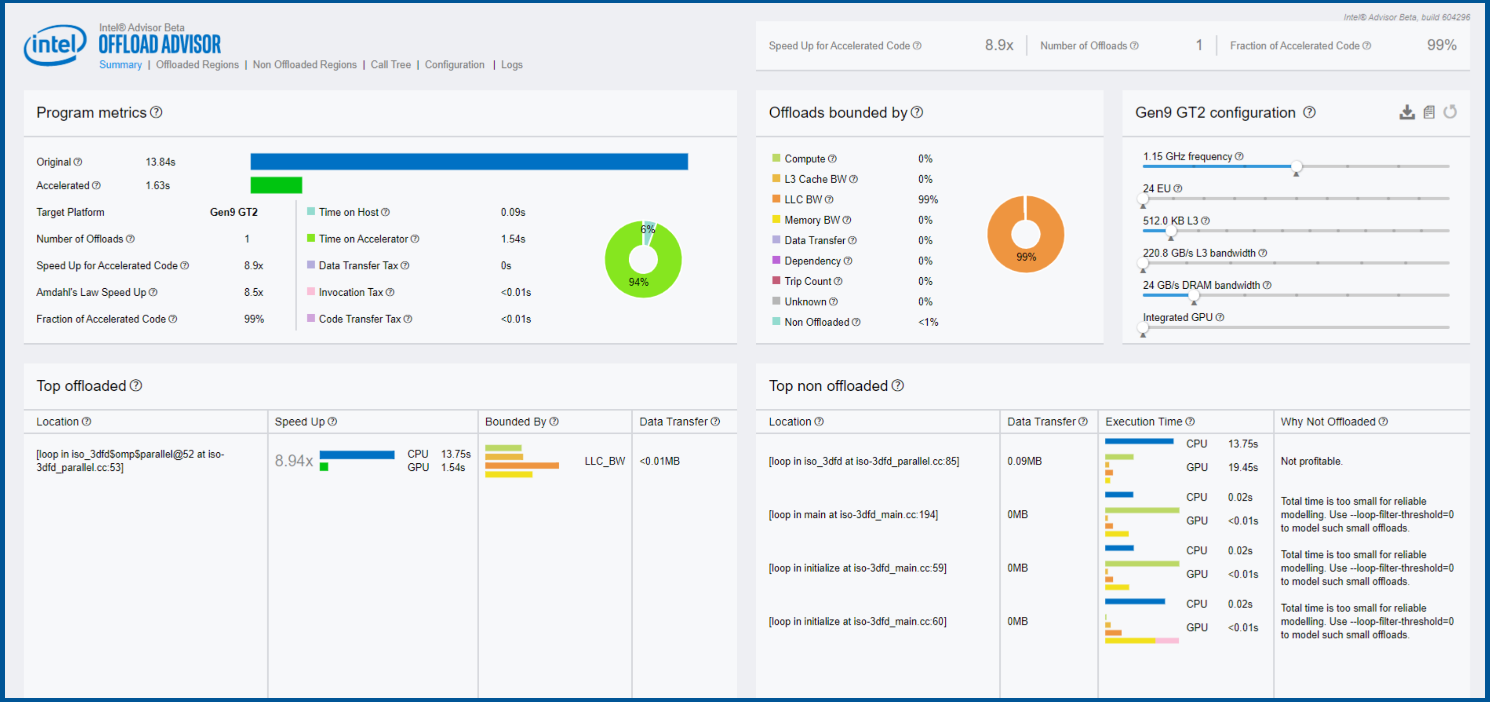
Introduction to Offload Advisor

Association for
Computing Machinery
Advancing Computing as a Science & Profession

Optimization Notice

Intel Offload Advisor (Beta)

- Starting from a baseline binary (running on CPU):
  - Helps defining which sections of the code should run on a given accelerator
  - Provides performance projection on accelerators (currently gen9 and gen11)



Modeling Flows supported: NOW

	Baseline HW	(Programming model)		Target HW	
1.	**CPU** measured	(C,C++,Fortran, Py)	→	**CPU** + measured	**GPU** estimated
1.a	**CPU** measured	(DPC++, OCL, OMP, “target=host”)	→	**CPU** + measured	**GPU** estimated

Modeling Flows supported: NOW + Coming Soon

	Baseline HW	(Programming model)		Target HW	
1.	**CPU** measured	(C,C++,Fortran, Py)	→	**CPU** + measured	**GPU** estimated
1.a	**CPU** measured	(DPC++, OCL, OMP, “target=host”)	→	**CPU** + measured	**GPU** estimated
2	**CPU+iGPU** measured	(C, C++, Fortran, DPC++, OCL, OMP)	→	**CPU** + measured	**GPU** estimated

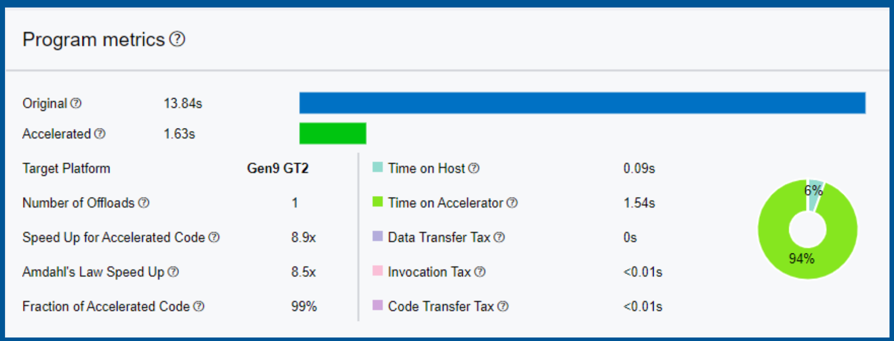
Association for Computing Machinery

Advancing Computing as a Science & Profession

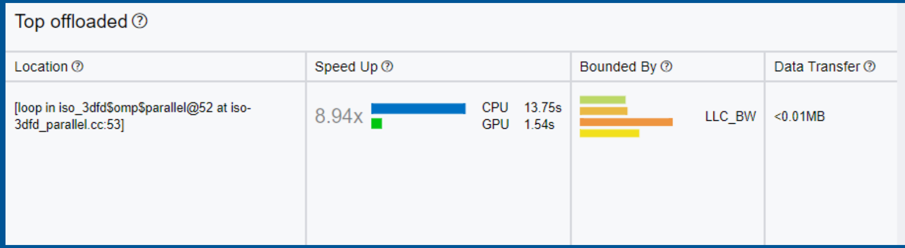
Optimization Notice

From Your CPU Application, you wonder:

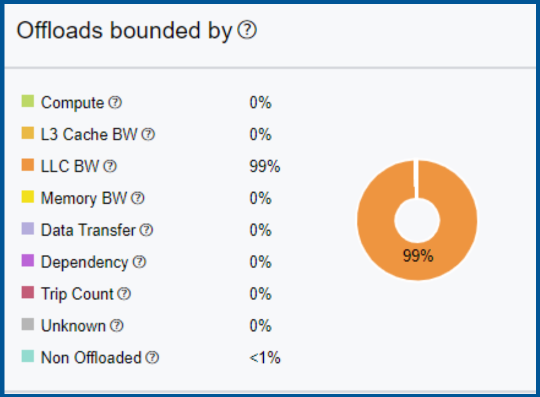
- How your code might perform on an accelerator ?



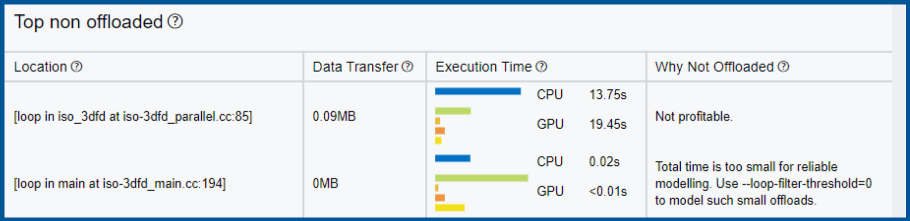
- What should you offload ?



- What might be limiting your performance on the accelerator ?



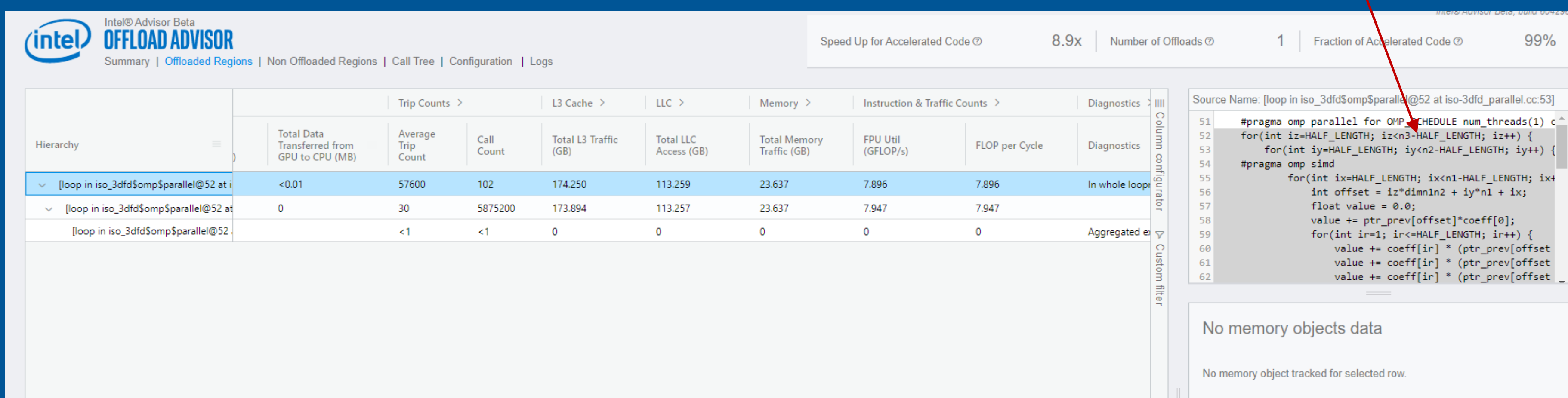
- What are the bad candidates for offload and Why ?



Top Offloaded in depth

- Provides a detailed description of each loop interesting for offload
  - Timings (total time, time on the accelerator, speedup)
  - Offload metrics (offload taxe, data transfers)
  - Memory traffic (DRAM, L3, L2, L1), trip count
  - Highlight which part of the code should run on the accelerator

This is where you will use DPCPP or OMP target for offload



Association for Computing Machinery
Advancing Computing as a Science & Profession

Non Offloaded in depth

- Explains why Advisor doesn’t recommend a given loop for offload
  - Dependency issues
  - Not profitable
  - Total time is too small

Intel® Advisor Beta

OFFLOAD ADVISOR

Summary | Offloaded Regions | Non Offloaded Regions | Call Tree | Configuration | Logs

Speed Up for Accelerated Code 8.9x | Number of O

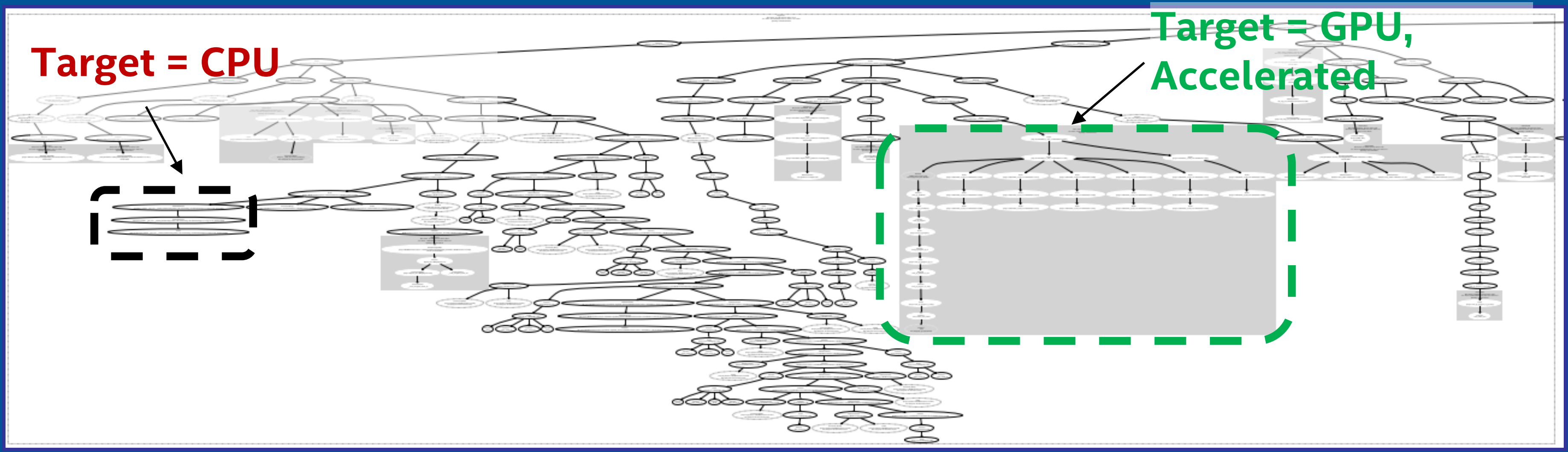
	Information				Potential Offload	
Hierarchy	oop ion ed?	Estimated Execution Time on Accelerator (+Host) (s)	Bounded By	Fraction Offloaded (%)	Why Not Offloaded	Potential Speed Up for Whole Region
[loop in iso_3dfd at iso-3dfd_parallel.cc:		1.539		100.00	Not profitable.	0.7068x
[loop in main at iso-3dfd_main.cc:194]		0.020		0	Total time is too small for reliable modelling. Use --loop-filter-threshold=0 to model such small offloads.	41338.1565x
› [loop in initialize at iso-3dfd_main.cc:59					Total time is too small for reliable modelling. Use --loop-filter-threshold=0 to model such small offloads.	640.4016x

Column configurator

Custom filte

Program Tree

- The program tree offers another view of the proportion of code that can be offloaded to the accelerator.



Association for Computing Machinery

Advancing Computing as a Science & Profession

Command Line Tips

Association for Computing Machinery
Advancing Computing as a Science & Profession

Before you start to use Offload Advisor

- The only strict requirement for compilation and linking is full debug information:

  -g: Requests full debug information (compiler and linker)

- Offload Advisor supports any optimization level, but the following settings are considered the optimal requirements:

  -O2: Requests moderate optimization

  -no-ipo: Disables inter-procedural optimizations that may inhibit Offload Advisor to collect performance data (Intel® C++ & Fortran Compiler specific)

Source Offload Advisor

- To set up the Intel® Advisor Beta environment, run one of the shell script:

```
source <ONEAPI_INSTALL_DIR>/setvars.sh
```

or

```
source <ADV_INSTALL_DIR>/env/vars.sh
```

- This script sets all required Intel Advisor environment variables, including APM, which points to `<ADV_INSTALL_DIR>/perfmodels`
- This is the location of the Offload Advisor scripts in the Intel® Advisor Beta installation directory

Intel® Advisor Beta OFFLOAD ADVISOR

The performance modeling functionality is available on Linux* OS only

Association for Computing Machinery
Advancing Computing as a Science & Profession

How does it work ?

- Easy to collect data and generate output with batch mode:

```
advixe-python <ADV_INSTALL_DIR>/perfmodels/run_oa.py
<path_to_result_dir>  -config gen9 --out-dir <path_to_result_dir> [--
options] -- <app>
```

- By default, **run_oa.py** marks up all regions and only selects the most profitable ones for analysis

- To generate the report.html, uses the following command:

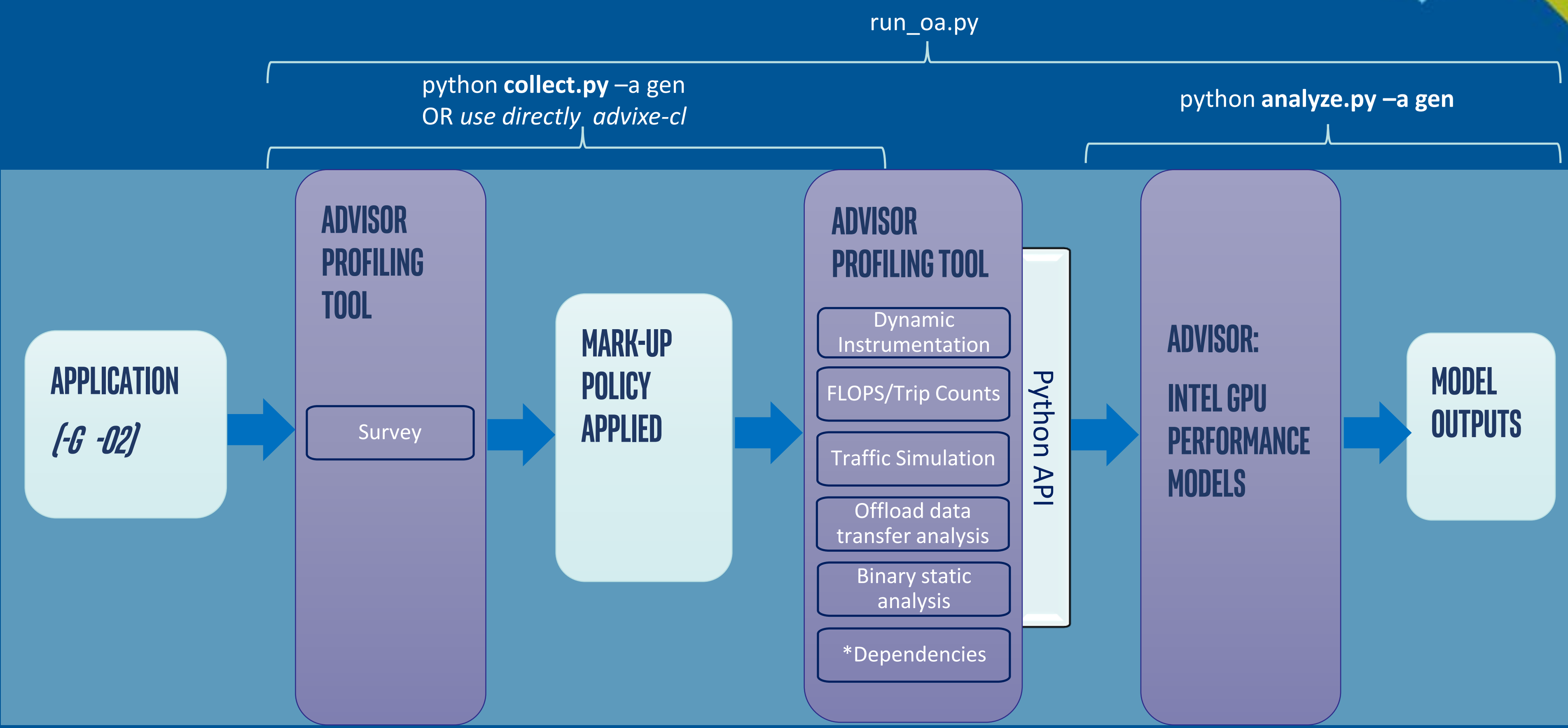
```
advixe-python $APM/analyse.py <project_dir> --config gen9 [--options] --
<app_binary> [app_options]
```

```
u31313@s001-n004:/opt/intel/inteloneapi/advisor/latest/perfmodels$ ls
accelerators      analyze.py        collect.py        debug.so        environ.py   oa_wrapper.so   shared.so   toml
analyze_impl.so   collect_impl.so   compute_stats.py  dot_graph.so    helpers      run_oa.py       template    tree.so
```

Association for Computing Machinery

Advancing Computing as a Science & Profession

Run_oa.py: What is running behind?



Association for Computing Machinery

Advancing Computing as a Science & Profession

Offload advisor Output Overview

- **`report.html`:** Main report in HTML format
- `report.csv` and `whole_app_metric.csv:` Comma-separated CSV files
- `program_tree.dot:` A graphical representation of the call tree showing the offloadable and accelerated regions
- **`program_tree.pdf`:** A graphical representation of the call tree

  Generated if the DOT(GraphViz*) utility is installed

  1:1 conversion from the `program_tree.dot` file
- `JSON` and `LOG` files that contain data used to generate the HTML report and logs, primarily used for debugging and reporting bugs and issues

Want to avoid dependency checking?

- Dependency adds a lot of time to the collection and you might want to remove it.
- Add the option –c basic for the collection:

```
advixe-python <ADV_INSTALL_DIR>/perfmodels/run_oa.py
<path_to_result_dir>  -config gen9 –c basic --out-dir
<path_to_result_dir> [--options] -- <app>
```

- Add the option --assume-parallel for the analysis:

```
advixe-python $APM/analyse.py <project_dir> --assume-parallel --config
gen9 [--options] -- <app_binary> [app_options]
```

Understanding the performance modelization

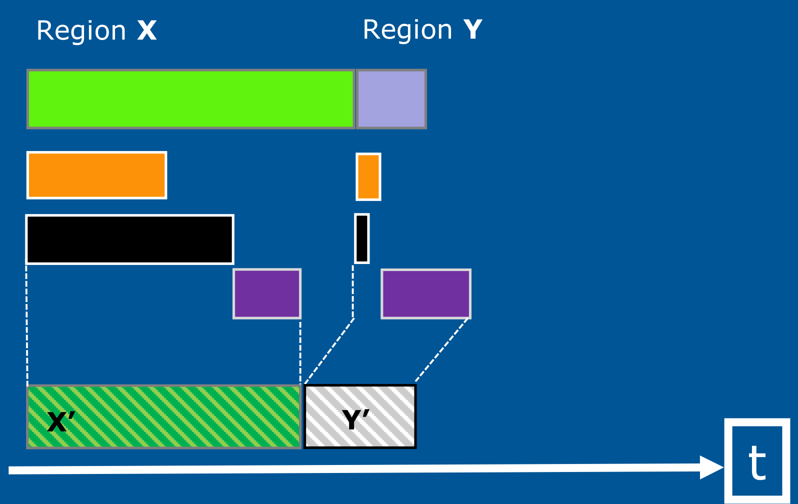
The mechanisms behind 1/2

First order analytical modeling pillars:

- Compute throughput model
- Memory sub-system model
- Offload data transfer modeling

Execution time on baseline platform (CPU)

- Execution time on accelerator. Estimate assuming bound exclusively by Compute
- Execution time on accelerator. Estimate assuming bound exclusively by caches/memory
- Offload Tax estimate (data transfer + invoke)



Final estimated time on target platform (eg GPU)

X – profitable to accelerate, $t(X) > t(X')$

Y - too much overhead, not accelerable, $t(Y) < t(Y')$

$$t_{\text{region}} = \max(t_{\text{compute}}, t_{\text{memory subsystem}}) + t_{\text{data transfer tax}} + t_{\text{kernel launch}}$$

Association for Computing Machinery

Advancing Computing as a Science & Profession

The mechanisms behind 2/2

We minimize the total time spent in this loop hierarchy by varying offload strategies U (offload/non-offload, #threads for each component $loop_i$ of loopnest)

$$\textbf{\textit{Objective function}} : \boldsymbol{T_{all}} = \min_{U=\{uf_1,uf_2,..\}} \left(\sum_i \boldsymbol{T_i} + t_{data\ transfer} + t_{invoke} + T_{cpu} \right)$$

Reject loopnests for which
T (x86) / T_{all} (x86+"X") < 1.0

$$\boldsymbol{T_i} = max \begin{cases} T_i^{Comp\_only}(\quad) \\ T_i^{M_k\_only}(M_i^k) = \dfrac{\boldsymbol{M_i^k}}{\boldsymbol{BW_k}} \end{cases}$$

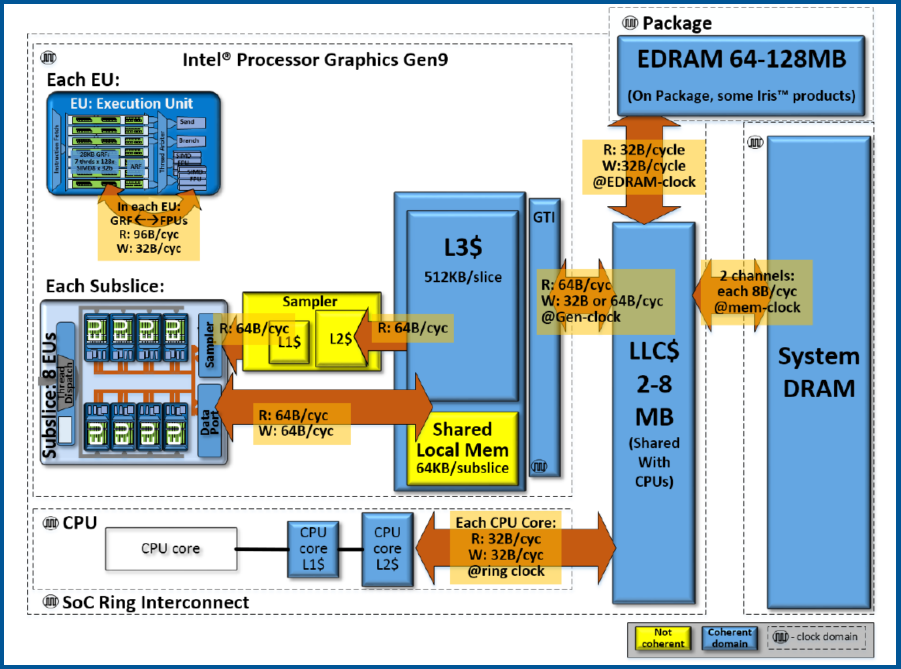
This is effectively "balance" (throughput) model

$$Under\ algorithmic\ constraints\ (Dependencies\ and\ TripCount/Granularity)$$

GPU Roofline Analysis

Intel® Gen9 Memory Hierarchy

- Intel® Graphics Compute Architecture uses the same DRAM with the CPU
- Level-3 (L3) data cache: slice-shared asset
- Shared Local Memory (SLM): a dedicated structure within the L3 that supports the work-group local memory address space
- Graphics Technology Interface (GTI): a dedicated interface unit connects the entire architecture interfaces to the rest of the SoC components
- The rest of SoC memory hierarchy includes the large Last-Level Cache (LLC, which is shared between CPU and GPU), possibly embedded DRAM and finally the system DRAM



A view of the SoC chip level memory hierarchy and its theoretical peak bandwidths for the compute architecture of Intel processor graphics gen9

Association for Computing Machinery

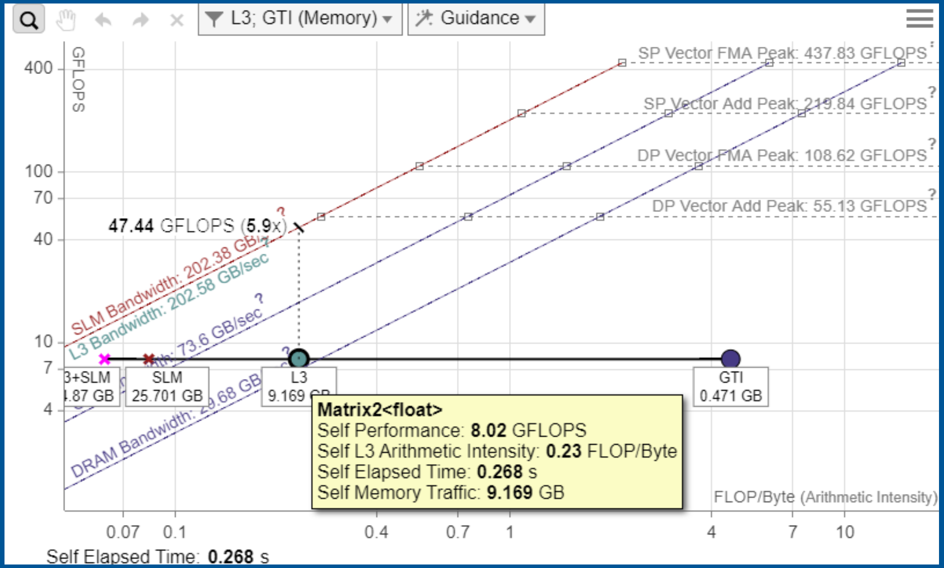
Advancing Computing as a Science & Profession

Optimization Notice

Find Effective Optimization Strategies

GPU Roofline Performance Insights

- Highlights poor performing loops
- Shows performance 'headroom' for each loop
  - Which can be improved
  - Which are worth improving
- Shows likely causes of bottlenecks
  - Memory bound vs. compute bound
- Suggests next optimization steps



Association for Computing Machinery

Advancing Computing as a Science & Profession

How to run?

The Roofline model on GPU is a technical preview feature and is not available by default.

To enable it:

```
export ADVIXE_EXPERIMENTAL=gpu-profiling
```

To run the GPU Roofline analysis in the Intel® Advisor CLI:

Run the Survey analysis with the `--enable-gpu-profiling` option:

```
advixe-cl –collect=survey --enable-gpu-profiling --project-dir=<my_project_directory> --search-dir
        src:r=<my_source_directory> -- ./myapp [app_parameters]
```

Run the Trip Counts and FLOP analysis with `--enable-gpu-profiling` option:

```
advixe-cl –collect=tripcounts --stacks --flop --enable-gpu-profiling --project-dir=<my_project_directory>
--search-dir    src:r=<my_source_directory> -- ./myapp [app_parameters]
```

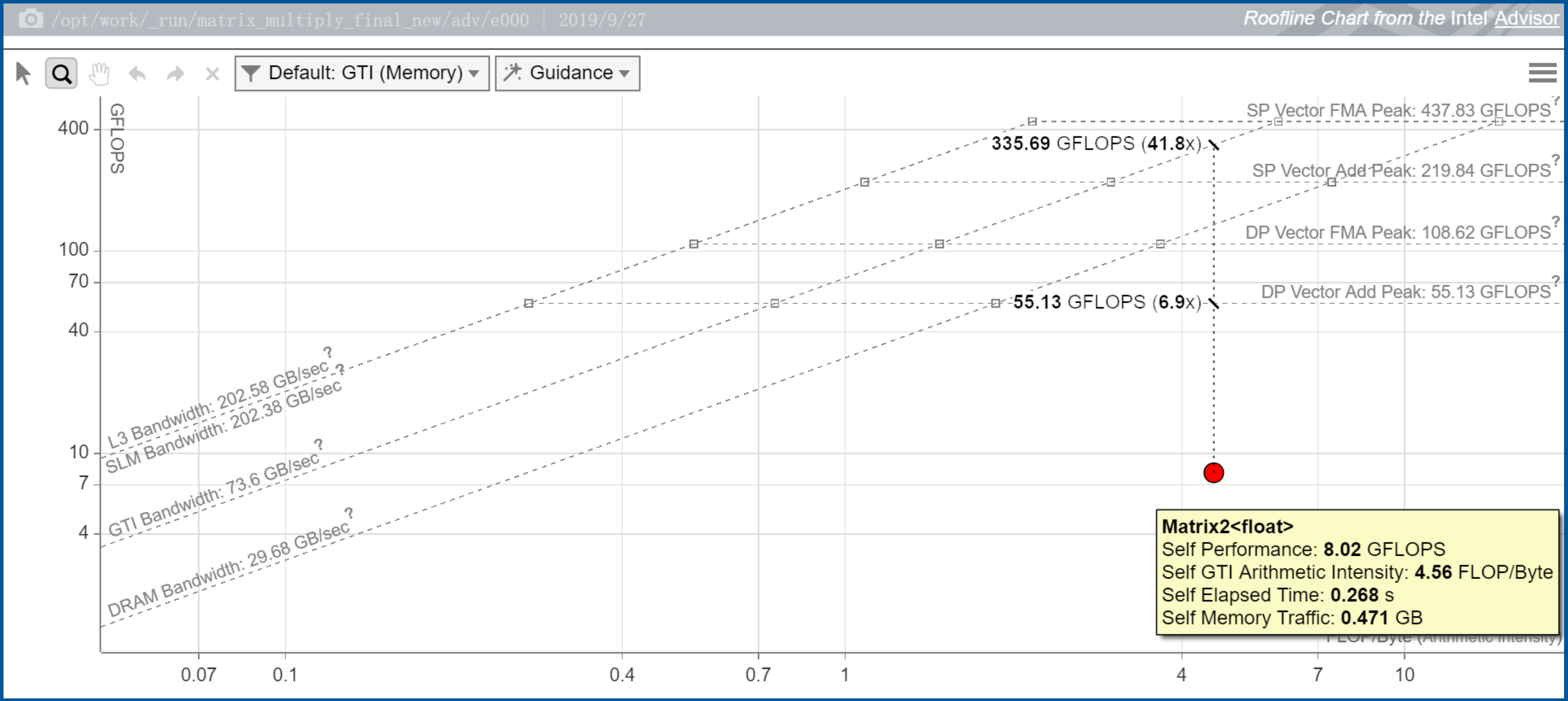
Generate a GPU Roofline report:

```
advixe-cl --report=roofline --gpu --project-dir=<my_project_directory> --report-output=roofline.html
```

Open the generated roofline.html in a web browser to visualize GPU performance.

Association for Computing Machinery
Advancing Computing as a Science & Profession

Roofline Analysis on Intel® GPU



Association for Computing Machinery

Advancing Computing as a Science & Profession

Legal notices & disclaimers

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, Xeon, Movidius and others are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as the property of others.

Association for Computing Machinery
Advancing Computing as a Science & Profession

Optimization Notice