Roofline Performance model:
How-tos

Roofline access and how-to **command line** example

FLOP/S = #FLOP/Seconds

```
> source advixe-vars.sh
```

1st pass Obtain "Seconds" 1.1x overhead

```
> advixe-cl --collect survey --project-dir ./your_project --
<your-executable-with-parameters>
```

2nd pass Obtain #FLOP count: 3x-5x overhead

```
> advixe-cl --collect tripcounts -flops-and-masks --project-dir
./your_project -- <your-executable-with-parameters>
```

```
> advixe-gui ./your_project
```

Launch GUI

MPI example (slurm)

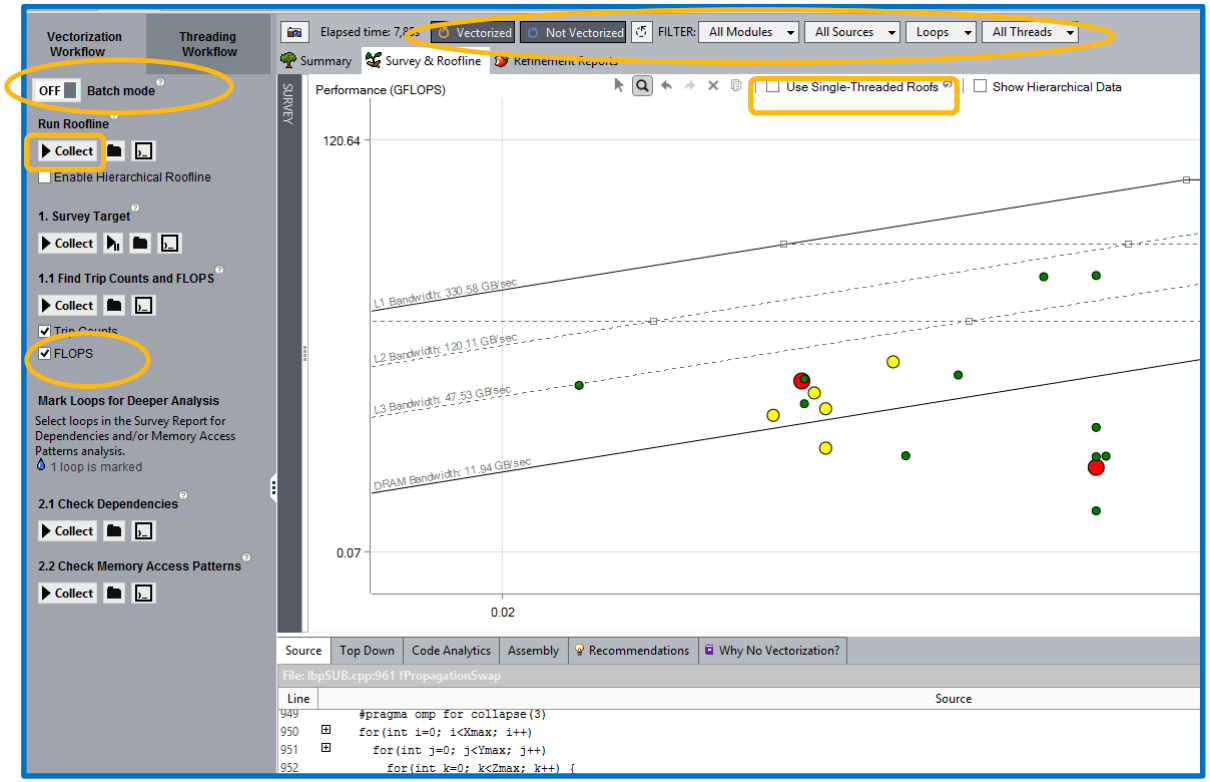
1st step:

srun -n <num-of-ranks> -c <num_of_cores_per_rank> **advixe-cl** -v -**collect survey** -project-dir=<same_dir_name> -data-limit=0 <your_executable>

2nd step:

srun -n <num-of-ranks> -c <num_of_cores_per_rank> **advixe-cl** -v -**collect tripcounts -flops-and-masks** -project-dir=<same_dir_name> -data-limit=0 <your_executable>

Roofline GUI access and how-to: **GUI**



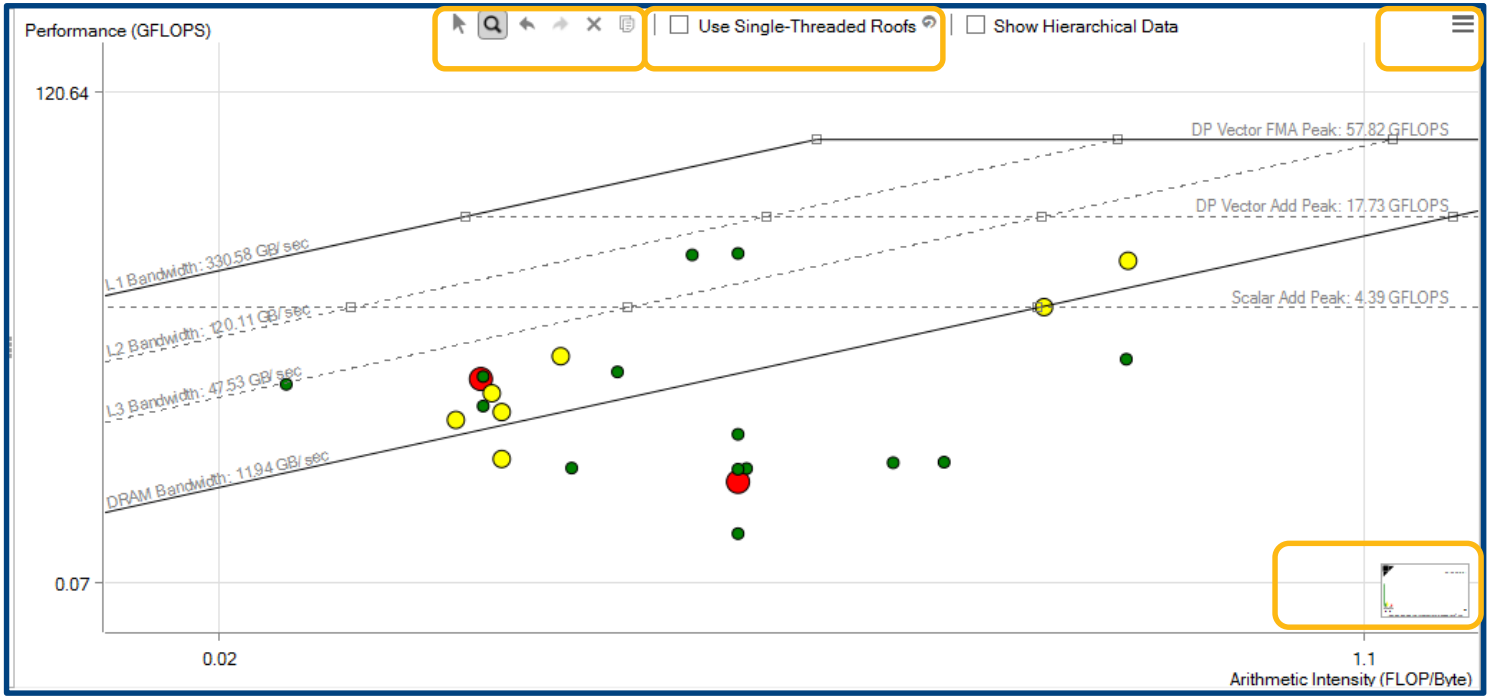
1) “**Run Roofline**” : most automated way.

2) You can also use **two separate runs**:

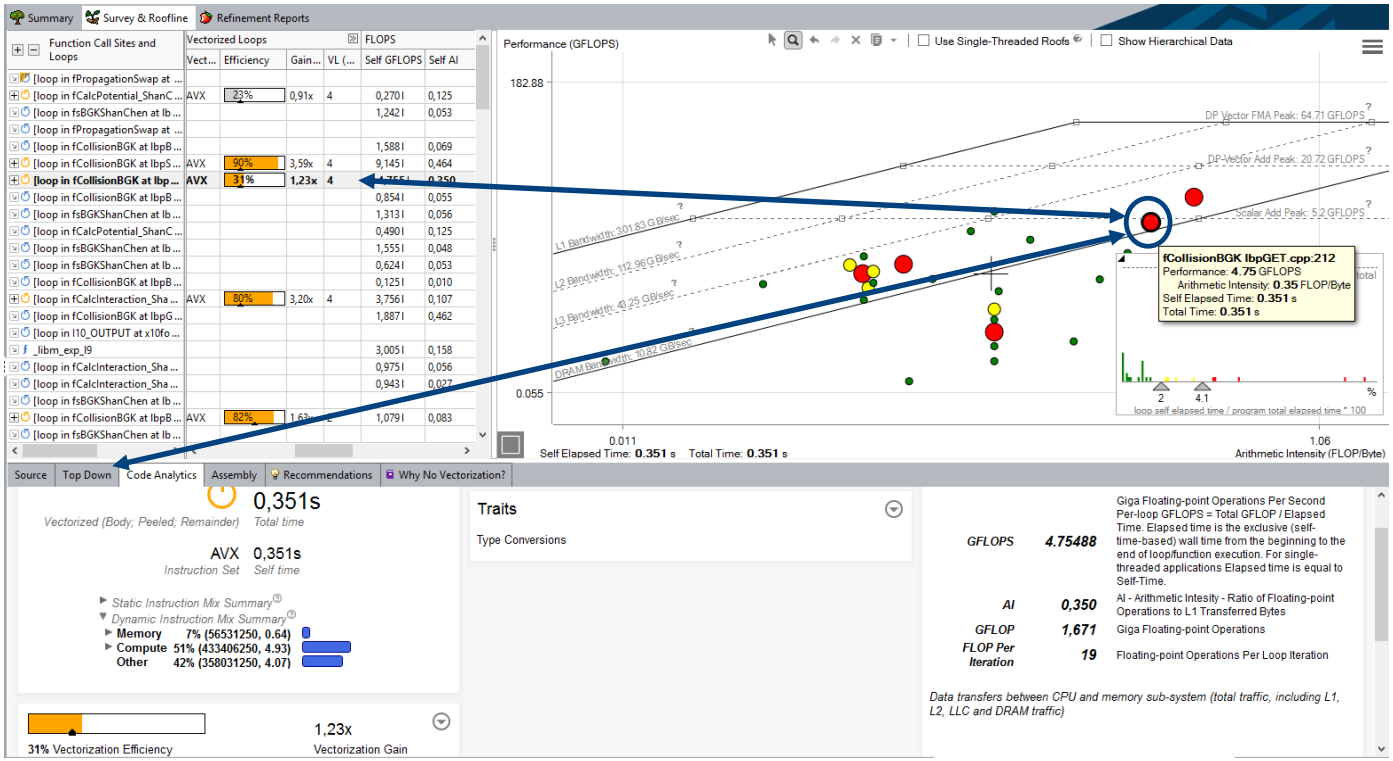
1. Survey
2. TripCounts (remember to switch **FLOPs** ON)

3) **Batch** Mode

Roofline Chart



Use Vectorization and Roofline views together



Observe slower Survey analysis or “finalization”?

(1.5x analysis slow-down or more)

Change default call stacks processing mode (***especially for Fortran***)

advixe-cl -collect survey –stackwalk-mode=online –no-stack-stitching

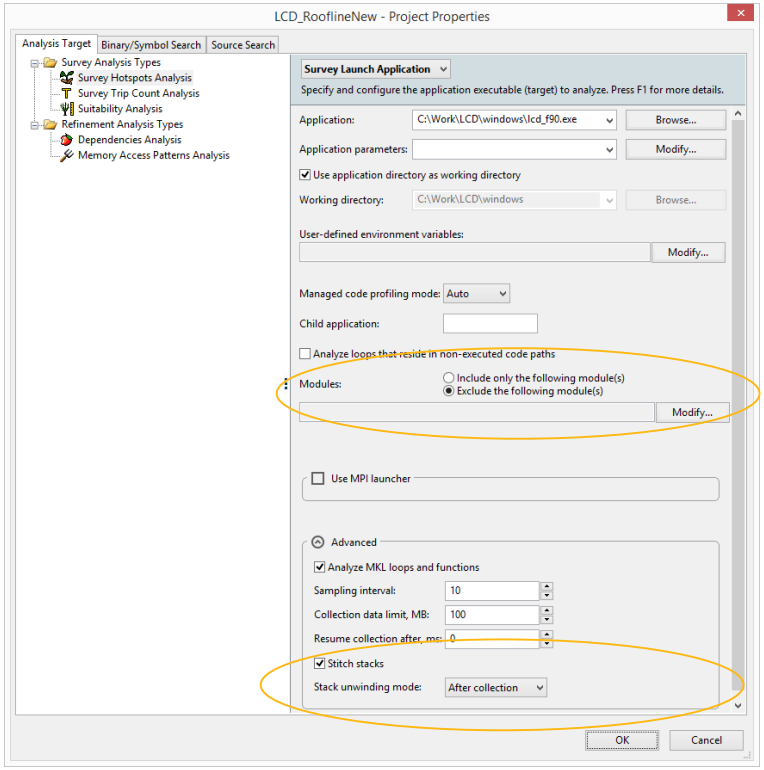
Consider disabling system modules and non-interesting modules processing:

advixe-cl -collect survey -module-filter-mode=include -module-filter=**foo.so**

Observe slower Survey analysis or “finalization”?

(1.5x slower than native run and more)

Configuration via GUI:



Observe slow tripcounts/FLOP analysis ??

(> 8x slower than native and more)

Consider combinations:

1. FLOPS **only**, no TripCounts:

   advixe-cl -collect tripcounts –flops-and-masks –no-trip-countss

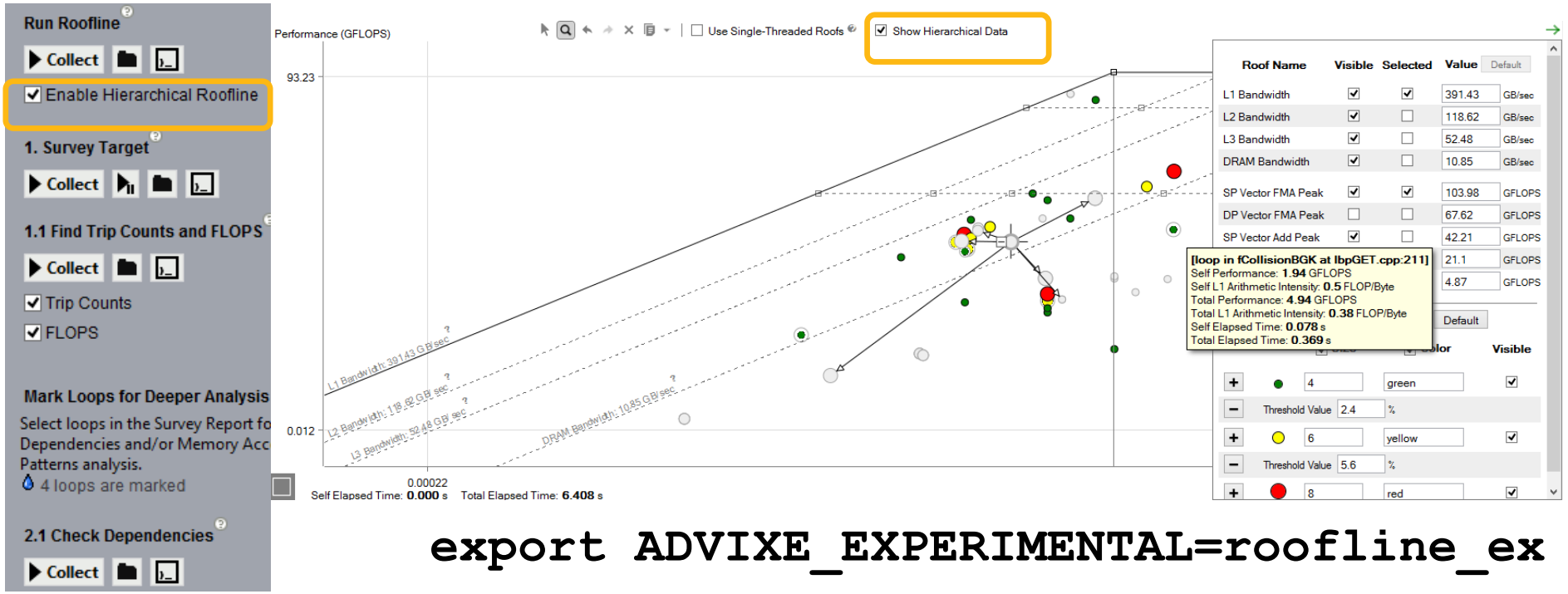
2. no FLOPS , TripCounts **only**, (->No Roofline):

   advixe-cl -collect tripcounts

3. FLOPS **and** TripCounts :

   advixe-cl -collect tripcounts –flops-and-masks

Hierarchical (top-down) Roofline: new in 2018 release

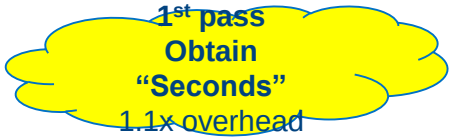



```
export ADVIXE_EXPERIMENTAL=roofline_ex
```

Hierarchical Roofline (based on stacks w/ FLOPS)

```
> source advixe-vars.sh
```

```
> export ADVIXE_EXPERIMENTAL=roofline_ex
```




```
> advixe-cl --collect survey --project-dir ./your_project --
<your-executable-with-parameters>
```


```
> advixe-cl --collect tripcounts -flops-and-masks -callstack-flops --project-dir
./your_project -- <your-executable-with-parameters>
```

```
> export ADVIXE_EXPERIMENTAL=roofline_ex
```

```
> advixe-gui ./your_project
```

Legal Disclaimer & Optimization Notice

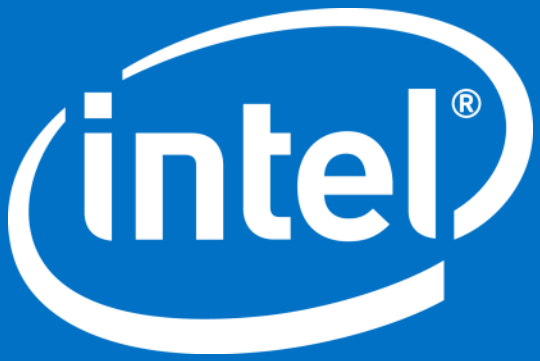
INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING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

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.

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®