Intel performance analysis tools for HPC

Mathieu Lobet, Matthieu Haefele

Maison de la Simulation, CEA, CNRS, Université Paris-Sud, UVSQ,
Universite Paris-Saclay, F-91191 Gif-sur-Yvette, France
(mathieu.lobet@cea.fr)

Intel Advisor

Advisor description

Intel Advisor : Vectorization advisor and Threading advisor

Vectorization advisor is a vectorization optimization tool:

Help to identify time-consuming loops that can benefit from vectorization or already vectorized

Help to identify vetorization and efficiency issues (dependencies, spilling, memory access...) and propose solutions

Help to ensure that vectorization is safe and quantify effects of vectorization (vectorization efficiency, roofline performance model)

Compilation for Advisor

Classical compilation with additional flags:

- -g
- -debug inline-debug-info
- -shared-intel
- -parallel-source-info=2

https://software.intel.com/en-us/vtune-amplifier-help-compiler-switches-for-performance-analysis-on-linux-targets

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

$ANALYSIS=survey: general overview of the performances and the vectorization state of the code.

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

$ANALYSIS=tripcounts: improves the survey by dynamically exploring loop iteration execution and propose better decisions about your vectorization strategy. It measures #FLOP count and cumulative data traffic necessary for the Roofline performance model.

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

$ANALYSIS=dependencies: refine analysis by checking for real data dependencies in loops the compiler did not vectorize because of assumed dependencies.

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

$ANALYSIS=map: (Memory Access Pattern) refine analysis by checking for various memory issues, such as non-contiguous memory accesses and unit stride vs. non-unit stride accesses.

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

-no-auto-finalize: Results are not finalize after the run. This is useful for collections on KNL that takes longer time than on Haswell. Finalization can then be performed on Haswell or when opening the results via the GUI.

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

-flop: take into account the masks in the vector operations on AVX512

Execution with Advisor

```
> mpirun -np 1 advixe-cl -collect $ANALYSIS --project-dir
$ANALYSIS_DIR --search-dir src:r=$SRCDIR -verbose -flop - ./
$APPNAME
```

: enable to target specific loops for the analysis

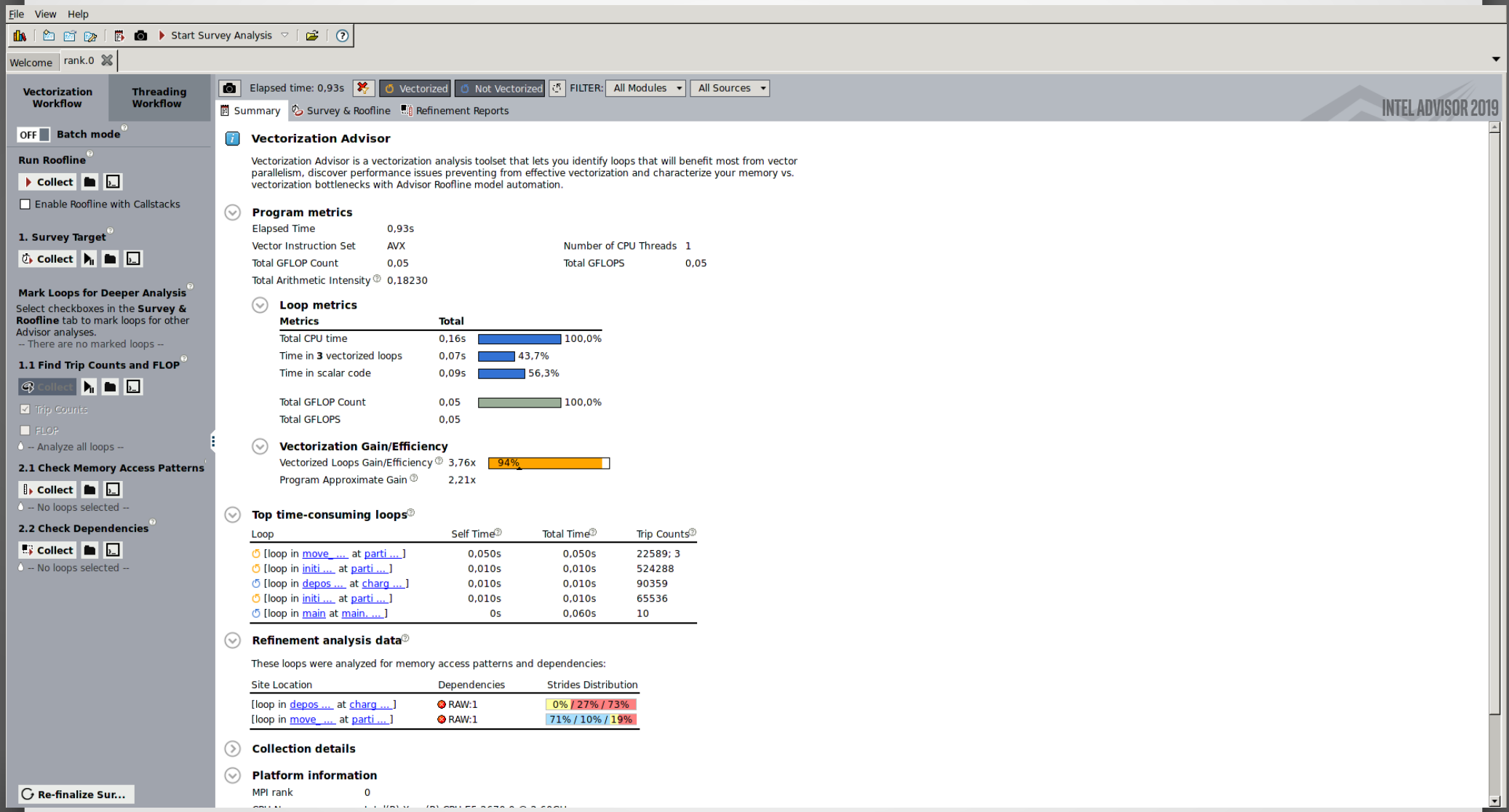
Visualization of the results

```
> advixe-gui ./$ANALYSIS_DIR
```

Open the graphic interface to visualilze the results

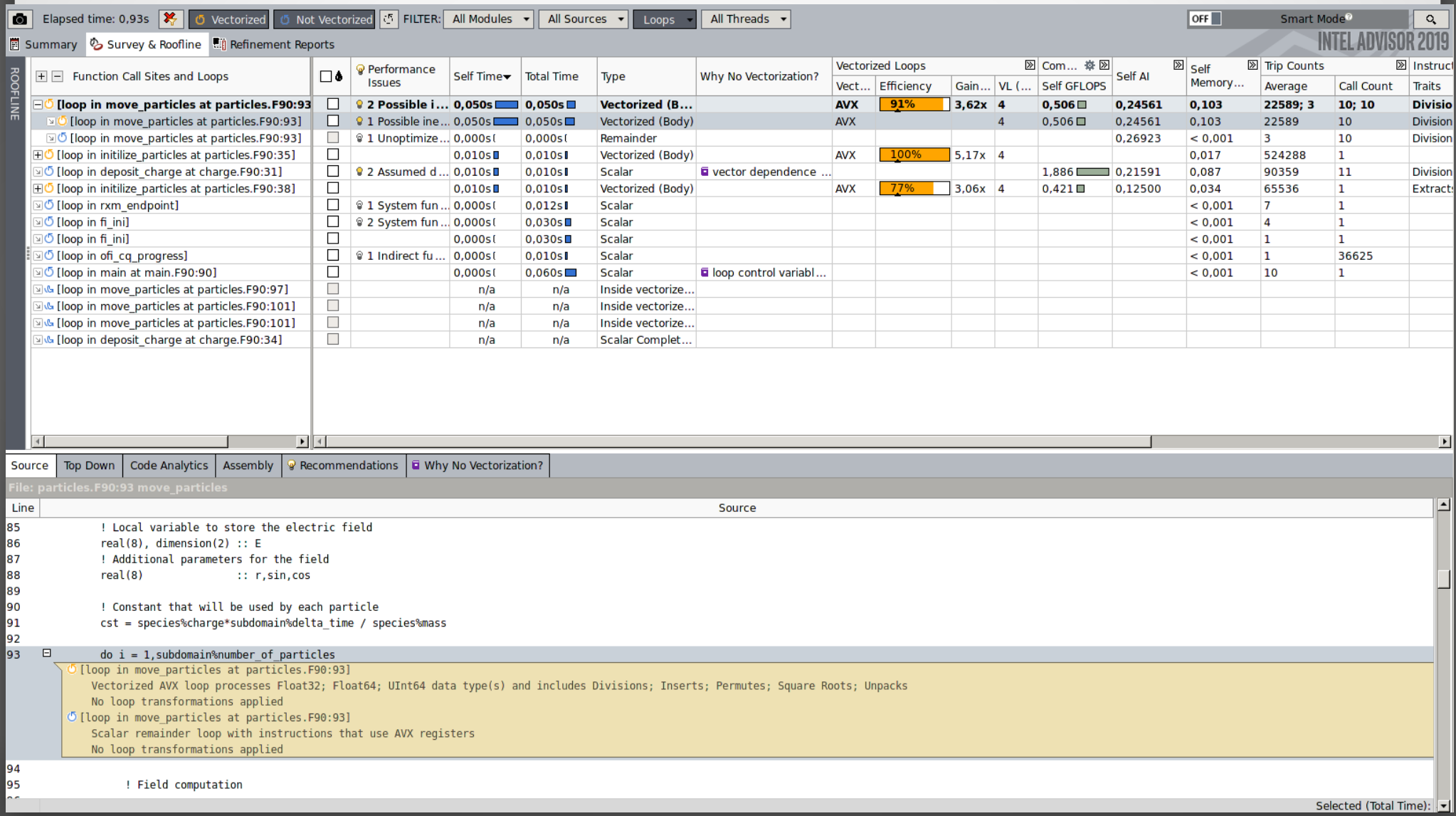
Visualization of the results

Global interface



Visualization of the results

Survey interface



Useful links

- Compilation flags: https://software.intel.com/en-us/vtune-amplifier-help-compiler-switches-for-performance-analysis-on-linux-targets
- Command line interface: https://software.intel.com/en-us/advisor-user-guide-intel-advisor-cli#F673BC4E-187B-4A08-AAF9-366304468295
- Command line interface: https://software.intel.com/en-us/advisor-user-guide-command-line-interface-reference
- Collection description: https://software.intel.com/en-us/advisor-user-guide-collect