www.locuz.com

Analyze Vectorization and Memory Aspects of an MPI Application

Presented By: Mandeep Kumar

Converge to the Cloud

Overview

- Since a distributed HPC application runs on a collection of several discrete nodes, apart from optimizing MPI communications across nodes and within nodes, you must also account for optimizations like vectorization on a per-node basis.

- This recipe explains how to use the vectorization and memory-specific capabilities and recommendations of the Intel® Advisor features to analyze an MPI application.

Scenario

You can collect data for MPI applications only with the Intel Advisor CLI, but you can view the results with the standalone GUI, as well as the command line. You can also use the GUI to generate required command lines.

This recipe describes an example workflow of analyzing the **Weather Research and Forecasting (WRF)** Model which is a popular MPI-based numerical application for weather prediction. Depending on a type of your MPI application, you can collect data on a different number of ranks:

- For profiling MPI applications written under the Single Program Multiple Data (SPMD) framework, like WRF, it is enough to collect data on a single MPI rank only, since all ranks execute the same code for a different subset of data. This also decreases the collection overhead.

- For Multiple Program Multiple Data (MPMD) applications, you should analyze all MPI ranks.

Ingredients

This section lists the hardware and software used to produce the specific result shown in this recipe:

- **Performance analysis tools:** Intel® Advisor 2020

- **Application:** Weather Research and Forecasting (WRF) Model version 3.9.1.1. The WRF workload used is Conus12km.
  IMPORTANT: You also must install the following application dependencies: zlib-1.2.11, szip-2.1.1, hdf5-1.8.21, netcdf-c-4.6.3 and netcdf-fortran-4.4.5.

- **Compiler:**
  - Intel® C++ Compiler 2019 Update 5
  - Intel® Fortran Compiler 2019 Update 5

- **Other tools:** Intel® MPI Library 2019 Update 6

- **Operating system:** Linux

- **CPU:** Intel(R) Xeon(R) Platinum 8260L

Prerequisites: Set Up Environment

1. Set up the environment for the required software:

```
$ source <compiler-install-dir>/linux/bin/compilervars.sh intel64
$ source <mpi-install-dir>/intel64/bin/mpivars.sh
$ source <advisor-install-dir>/advixe-vars.sh
```

To verify that you successfully set up the tools, you can run the following commands. You should get the product versions.

```
$ mpiicc -v
$ mpiifort -v
$ mpiexec -V
$ advixe-cl -version
```

2. Set the environment variables required for the WRF application:

```
export LD_LIBRARY_PATH=/path_to_IO_libs/lib:$LD_LIBRARY_PATH
ulimit -s unlimited
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export KMP_STACKSIZE=512M
export OMP_NUM_THREADS=1
```

3. Build the application in the Release mode. The –g compile-time flag is recommended so that Intel Advisor can show source names and locations.

Survey Your Target Application

- The first step is to run the Survey analysis on the target application using the Intel Advisor CLI. This analysis type collects high-level details about the target application. To run the analysis:
  - Pass the advixe-cl with collection options as an argument to the mpiexec launcher.
  - Use the -gtool flag to attach analysis to the ranks specified after the name of a project directory.
- To execute the WRF application with 48 ranks on a single node of the Intel® Xeon® processor and attach the Survey analysis to the rank 0 only:

```
mpiexec -genvall -n 48 -ppn 48 -gtool "advixe-cl --collect=survey --project-dir=<project_dir>/project1:0" ./wrf.exe
```

  This command will generate a result folder for the rank 0 only containing the survey data.

- You can also run the analysis for a set of ranks, for example 0, 10 through 15, and 47:

```
mpiexec -genvall -n 48 -ppn 48 -gtool "advixe-cl --collect=survey --project-dir=<project_dir>/project1:0,10-15,47" ./wrf.exe
```

Collect Trip Counts and FLOP Data and Review the Results

After running the Survey analysis, you can view the Survey data collected for your application or collect additional information about trip counts and FLOP. To run the Trip Counts and FLOP analysis only for the rank 0 of the WRF application, execute the following command:

```
mpiexec -genvall -n 48 -ppn 48 -gtool "advixe-cl --collect=tripcounts --flop --project-dir=<project_dir>/project1:0" ./wrf.exe
```

This command reads the collected survey data and adds details on trip counts and FLOP to it.

Collect Trip Counts and FLOP Data and Review the Results

You can view the collected results on a remote machine or move the results to a local machine and view them with the Intel Advisor GUI. To visualize results on the local machine, you can do the following:

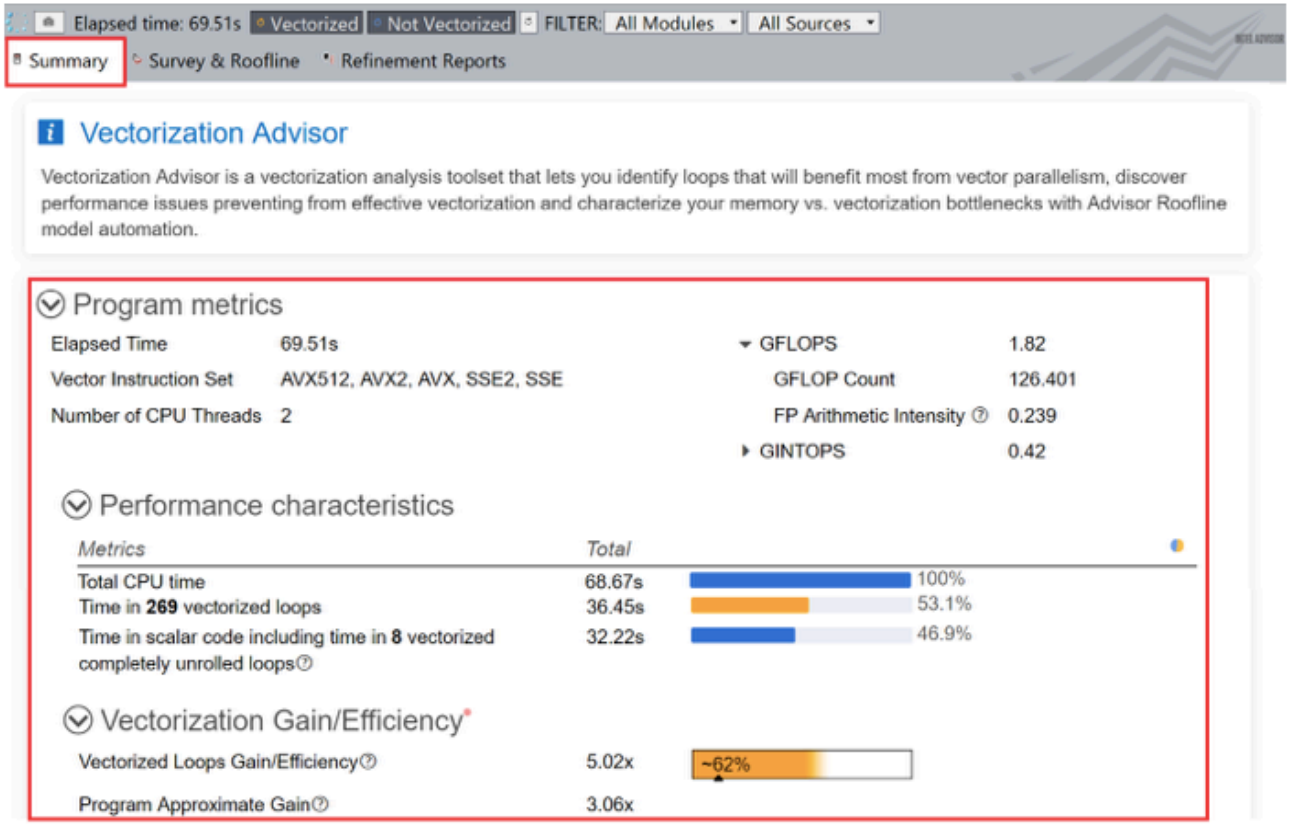
1. Pack the result files, corresponding sources and binaries in a single snapshot file with .advixeexpz extension:

```
  advixe-cl --snapshot --project-dir=<project_dir>/project1 --pack --cache-sources --cache-binaries
-- <snapshot_name>
```

2. Move this snapshot to a local machine and open it with the Intel Advisor GUI.

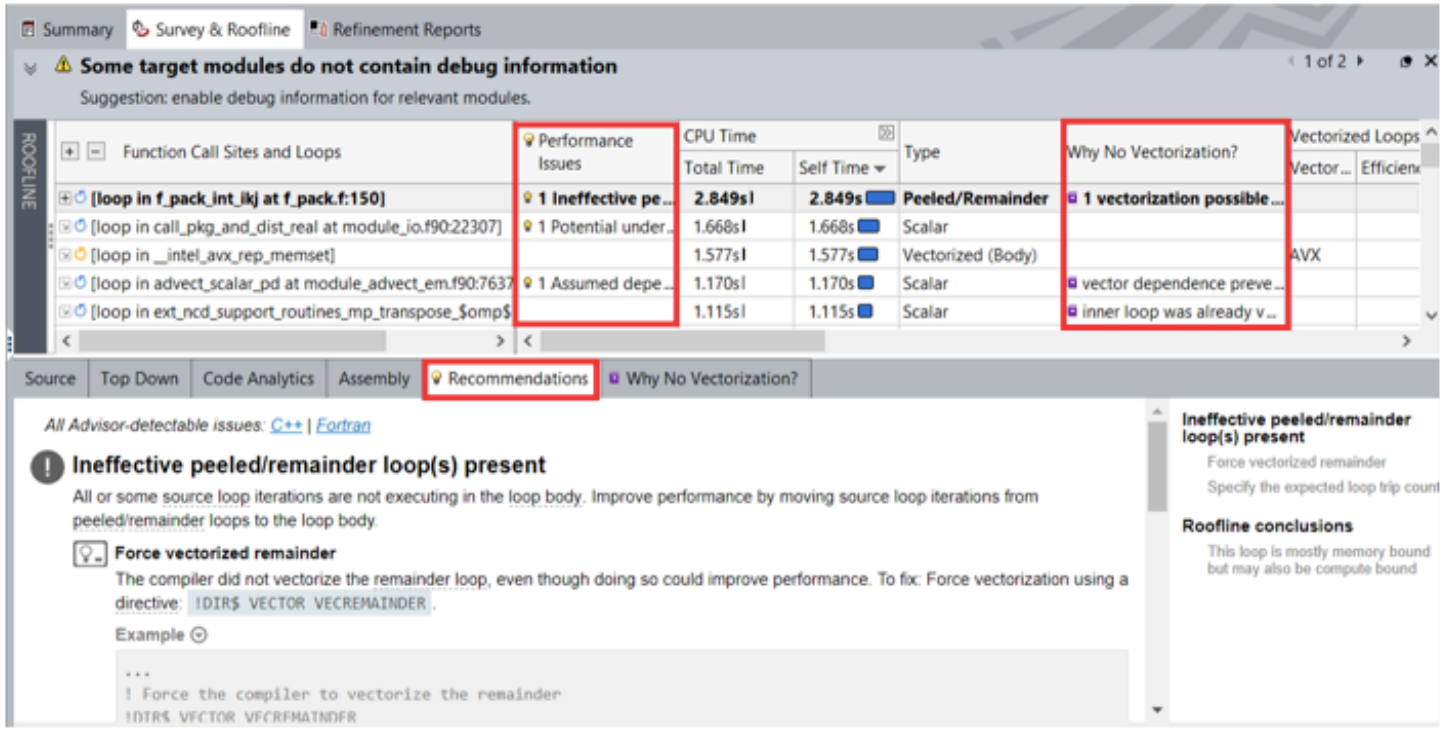
Generated Results

- In the Summary tab of the survey report, review the program metrics like elapsed time, number of vectorized loops, vector instruction sets used, GFLOPS.



Generated Results

- In the **Survey & Roofline** tab, review the details about the application performance with a list of loops/functions with the most time-consuming ones at the top. Use the messages in the **Performance Issues** and **Why No Vectorization** columns identify the next steps for improving the application performance.

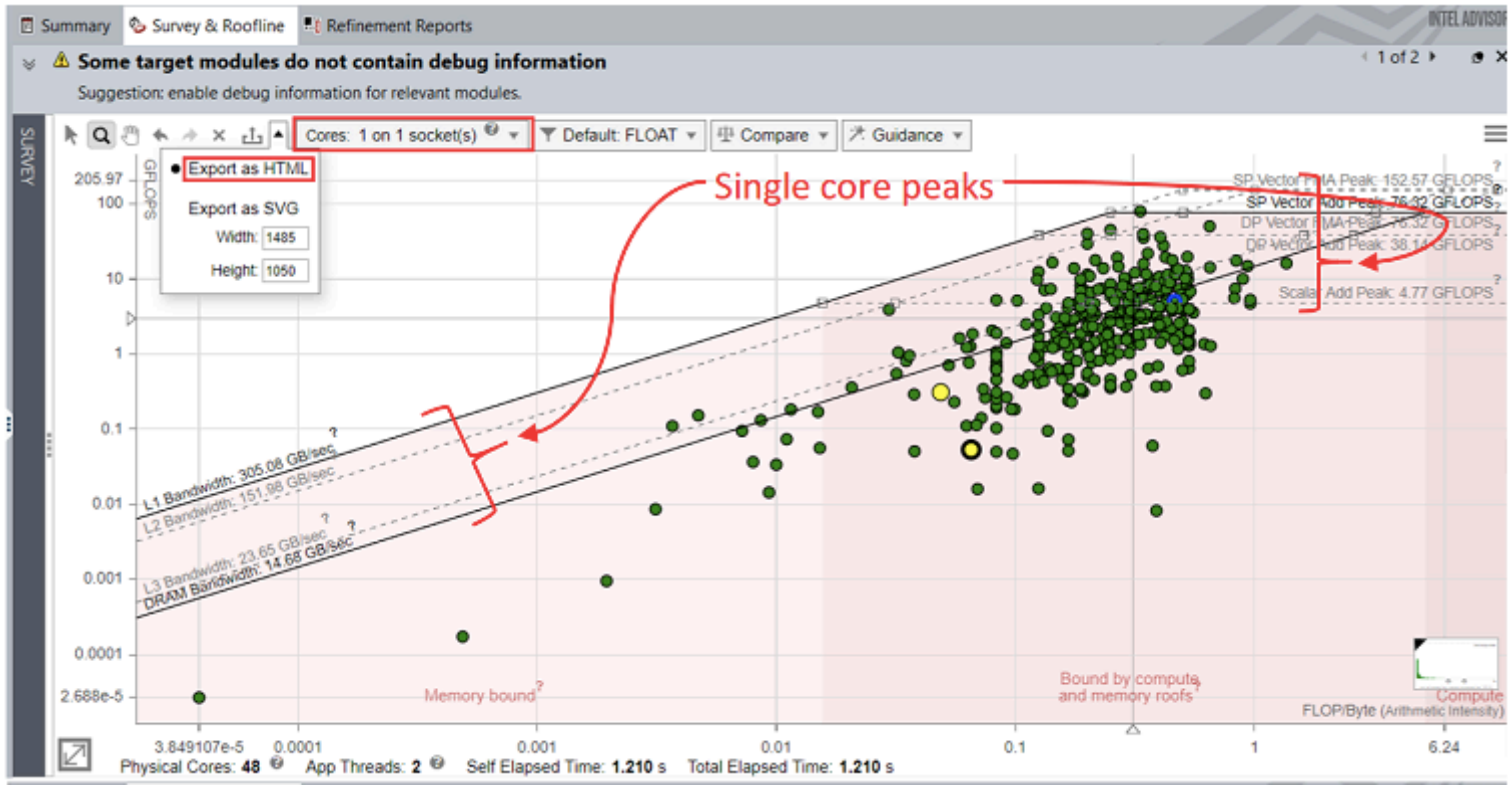


Review the Roofline Chart

- Intel Advisor can plot roofline charts for applications that visualize application performance levels relatively to the system's peak compute performance and memory bandwidths.
- To generate a roofline report for an MPI application, you must run the Survey and Trip Counts and FLOP analyses one after the other as described in the previous sections.
- These analyses collect all the data required to plot roofline charts for MPI applications.
- To open the Roofline report, click the **Roofline** toggle button on the top left pane of the analysis results opened in Intel Advisor GUI. Note the following:
  - Based on dot positions along the horizontal, we can see that WRF loops are mostly bounded both by compute and memory.
  - Most loops take similar amount of time, which is denoted by size and color of dots.

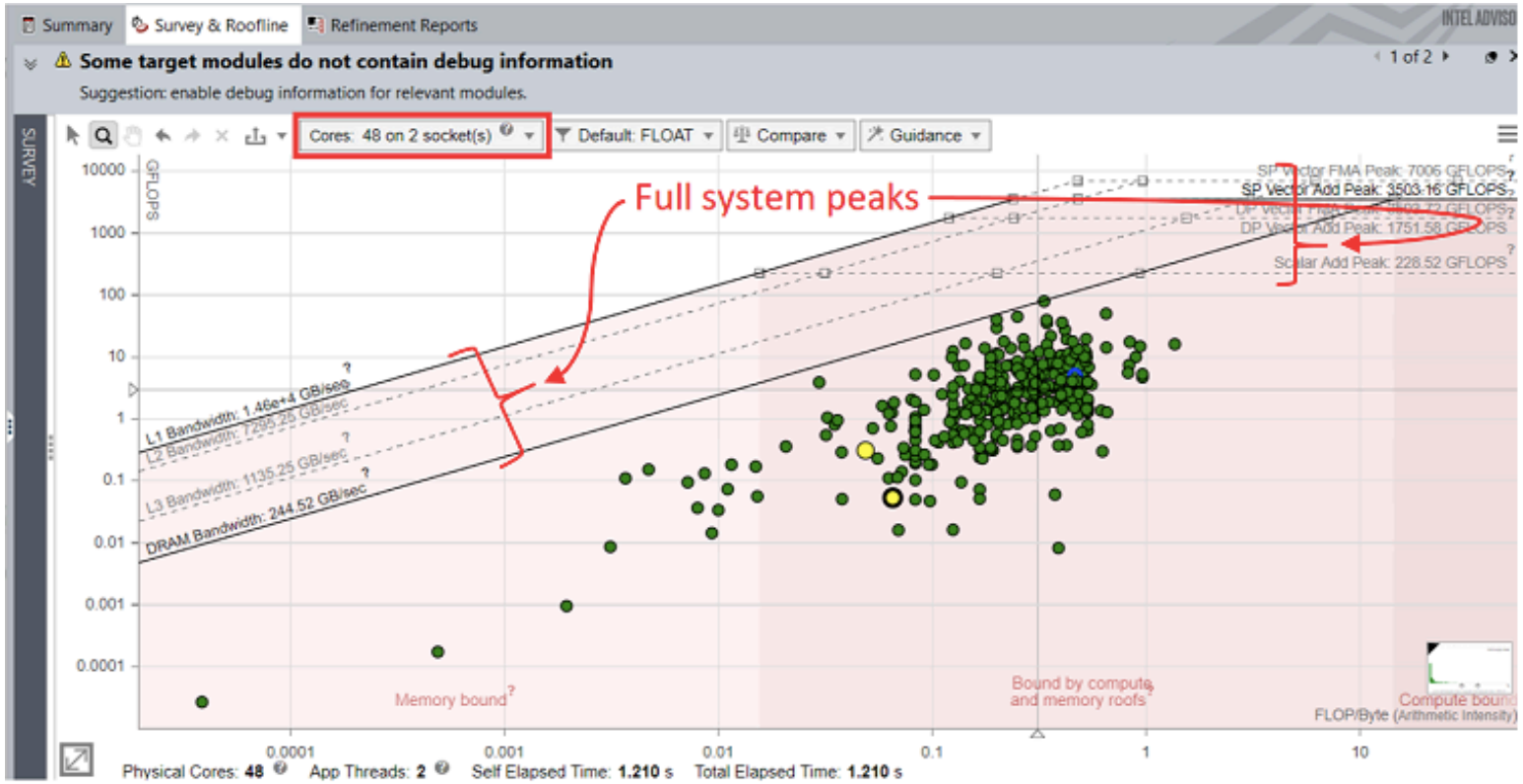
Adjust the Roofline Chart

- Since Intel® Advisor was executed on a single rank 0, the dots show performance of the rank 0 only and not the full application performance. As a result, the distance between loops and roofs (relative dots positions) in the Roofline chart shows a poorer performance than it is in reality.



Adjust the Roofline Chart

- To adjust the dot positions, change the number of cores from a drop-down list in the top pane of the Roofline chart to the total number of MPI ranks in the application. For the WRF application, choose 48 cores. Changing the number of cores adjusts the system memory and compute roofs accordingly in the Roofline chart. The relative positions of dots change based on the roofs plotted, but their absolute values do not change.



Export the Roofline Report (optional)

For MPI applications, the recommended way to get a separate Roofline report to share is to export it as an HTML or SVG file. Do one of the following:

- In the Roofline report opened in the Intel Advisor GUI, click the Export button in the report toolbar and choose Export as HTML or Export as SVG.

- Run the CLI command with the --report=roofline option. For example:

```
advixe-cl --report=roofline --project-dir=<project_dir>/project1 -report-output=./wrf_roofline.html
```

For MPI applications, using the CLI command is the recommended because you do not need to have an installation of Intel Advisor GUI to use it.

Run the Dependencies Analysis (optional)

- Intel Advisor may require more details about your application performance to make useful recommendation. For example, Intel Advisor may recommend running the Dependencies analysis for some loops that have Assumed dependency present message in the **Performance Issues** column of the Survey report.

- To collect the dependencies data, you must choose specific loops to analyze. For MPI applications, choose one of the following:

  - **Loop IDs-based collection**
  - **Source location-based collection**

Loop IDs-based collection:

1. Generate a Survey report to get loop IDs:

```
advixe-cl --report=survey –project-dir=./<project_dir>/project1
```

This command will create an advisor-survey.txt file with metrics for all loops in your application sorted by self time. The loop IDs are in the first column of the table:

MPI rank:0

ID	Function Call Sites and Loops	Total Time	Self Time	Type
1752	[loop in f_pack_int_ikj at f_pack.f:150]	2.849s	2.849s	Peeled/Remainder
126	-[loop in f_pack_int_ikj at f_pack.f:150]	2.849s	2.849s	Remainder
104	[loop in call_pkg_and_dist_real at module_io.f90:22307]	1.668s	1.668s	Scalar
144	[loop in __intel_avx_rep_memset]	1.577s	1.577s	Vectorized (Body)
332	[loop in advect_scalar_pd at module_advect_em.f90:7637]	1.170s	1.170s	Scalar
105	[loop in ext_ncd_support_routines_mp_transpose_ompparallel_for@776 at wrf_io.f:782]	1.115s	1.115s	Scalar
1801	[loop in w_damp at module_big_step_utilities_em.f90:2542]	1.033s	1.033s	Scalar Versions
355	-[loop in w_damp at module_big_step_utilities_em.f90:2542]	1.033s	1.033s	Scalar
109	[loop in __intel_avx_rep_memcpy]	0.909s	0.909s	Vectorized (Body)
287	[loop in set_physical_bc3d at module_bc.f90:874]	0.878s	0.878s	Scalar
1753	[loop in f_unpack_int_ikj at f_pack.f:201]	0.797s	0.797s	Peeled/Remainder
428	-[loop in f_unpack_int_ikj at f_pack.f:201]	0.797s	0.797s	Remainder

2. Identify loops to run the deeper analysis on.

3. Run the Dependencies analysis for the selected loops on the rank 0 of the WRF application. In this case, we select loops 235 and 355:

```
mpiexec –genvall –n 48 –ppn 48 –gtool "advixe-cl --collect=dependencies --mark-up-list=235,355
--project-dir=<project_dir>/project1:0" ./wrf.exe
```

Source location-based collection:

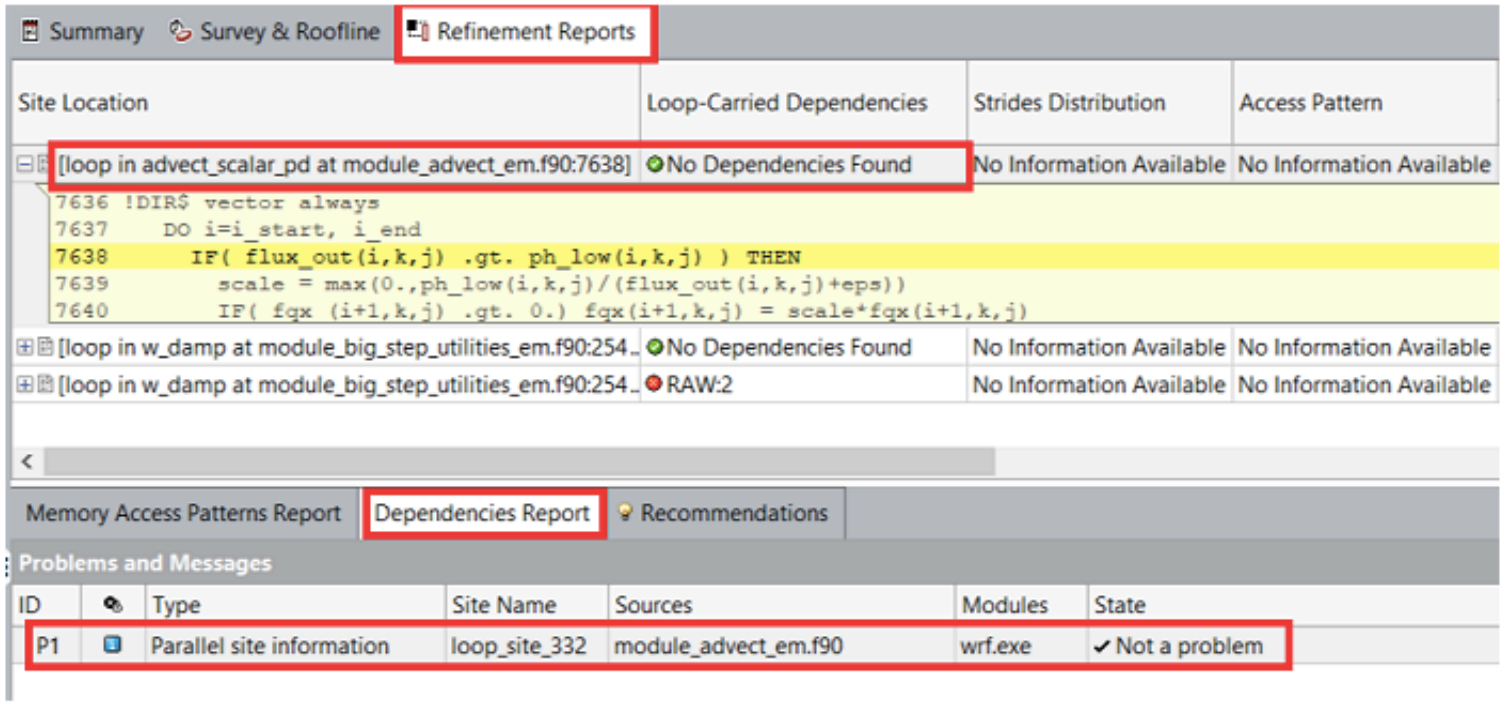
- Specify the source location of loops to analyze in the file1:line1 format and run the Dependencies analysis for the selected loops on the rank 0 of the WRF application:

```
mpiexec -genvall -n 48 -ppn 48 -gtool "advixe-cl --collect=dependencies --mark-up-list=module_advect_em.f90:7637,module_big_step_utilities_em.f90:2542 --project-dir=<project_dir>/project1:0" ./wrf.exe
```

Dependencies Analysis Results

- After you run the Dependencies analysis, the results will be added to the **Refinement Reports** tab of the analysis results.

- For the WRF application, the Dependencies analysis confirms that there were no dependencies in the selected loops and the **Recommendations** tab suggests related optimization steps.



Run the Memory Access Patterns Analysis (optional)

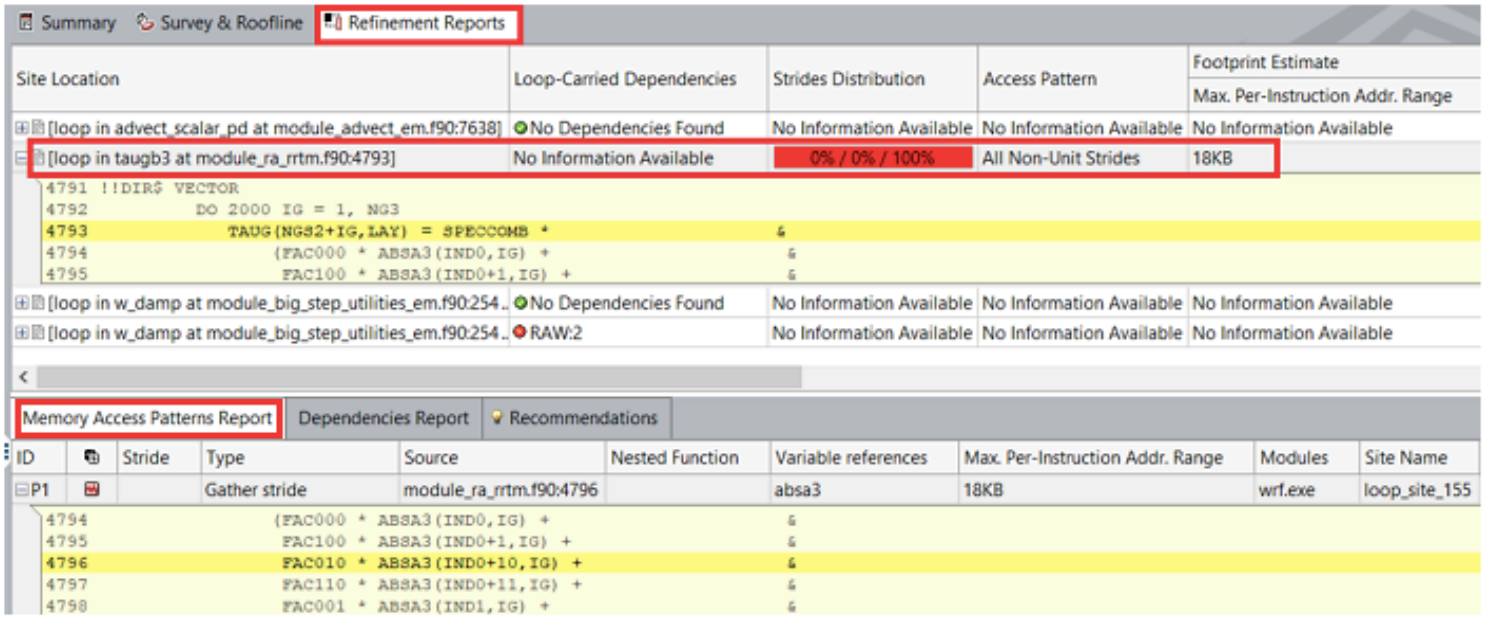
- If you want to check your MPI application for various memory issues, such as non-contiguous memory accesses and unit stride vs. non-unit stride accesses, run the Memory Access Patterns (MAP) analysis.
- Intel Advisor may recommend running the MAP analysis for some loops that have Possible inefficient memory access patterns present message in the Performance Issues column of the Survey report.
- To run the MAP analysis:

1. Identify loop IDs or source locations to run the deeper analysis on.
2. Run the MAP analysis for the selected loops (155 and 200 in this case) on the rank 0 of the WRF application:

```
mpiexec -genvall -n 48 -ppn 48 -gtool "advixe-cl --collect=map --mark-up-list=155,200 --project-dir=<project_dir>/project1:0" ./wrf.exe
```

MAP Analysis Results

- After you run the MAP analysis, the results will be added to the **Refinement Reports** tab of the analysis results.
- For the WRF application, the MAP analysis reported that all strides for the selected loops are random in nature, which could cause suboptimal memory and vectorization performance.
- See the messages in the **Recommendations** tab for potential next steps.



Key Take-Aways

- You can use the Intel® Advisor to analyze your MPI applications on one, several, or all ranks. This recipe used the WRF Conus12 km workload.

- To run the Survey, Trip Counts and FLOP, Roofline, Dependencies, or Memory Access Pattern analysis on an MPI application, you can use only Intel Advisor CLI commands, but you can visualize the results generated in the Intel Advisor GUI.

Thanks!