



Intel[®] VTune[™] Profiler 2020

Release Notes

18 February 2020

Customer Support

For technical support, including answers to questions not addressed in this product, visit the technical support forum, FAQs, and other support information at:

- <https://software.intel.com/en-us/vtune/support/get-help>
- <http://www.intel.com/software/products/support/>
- <https://software.intel.com/en-us/vtune>

Register your product at <https://registrationcenter.intel.com/> by providing your email address. Registration entitles you to free technical support, product updates and upgrades for the duration of the support term. It also helps Intel recognize you as a valued customer in the support forum.

NOTE: If your distributor provides technical support for this product, contact them directly for support issues.

Contents

1	Introduction	2
2	What's New	3
3	System Requirements	4
4	Where to Find the Release	8
5	Installation Notes	8
6	Known Issues	9
7	Attributions	19
8	Legal Information	20

1 Introduction

Intel® VTune™ Profiler 2020 provides an integrated performance analysis and tuning environment with a graphical user interface that helps you to analyze code performance on systems with Intel® 64 architectures.

This document provides system requirements, issues, limitations, and legal information pertaining to Intel VTune™ Profiler.

Intel VTune™ Profiler for Windows and Linux OS has a standalone graphical user interface (GUI) as well as a command-line interface (CLI). VTune™ Profiler for macOS* supports viewing of results collected on other platforms. Native collection on a macOS* system is currently not available.

To learn more about Intel VTune™ Profiler, see:

- New features listed in the [What's New](#) section below.
- [Help Document](#)
- Installation Guide for:
 - [Linux*](#)
 - [macOS*](#)
 - [Windows*](#)

2 What's New

Detailed [new features](#).

Intel® VTune™ Profiler 2020

- There is a new, more descriptive name: “Intel® VTune™ Profiler” (formerly “Intel® VTune™ Amplifier”).
 - Command line interface `amplxe-cl` and GUI interface `amplxe-gui` were re-named to `vtune` and `vtune-gui` respectively
- Intel® VTune™ Profiler has been updated to include more recent versions of 3rd party components, which include functional and security updates. Users should update to the latest version.
- GPU accelerators support:
 - New [GPU Offload](#) analysis added to explore and correlate code execution across CPUs and GPUs. You can identify a kernel of interest for GPU-bound applications and explore further with GPU Compute/Media Hotspots analysis.
 - [GPU Compute/Media Hotspots](#) analysis updated with GPU in-kernel analysis for OpenCL™ code and an option to filter by a kernel of interest.
 - Command line [hotspots](#) report now supports GPU analysis types. You can apply the `computing-task` and `computing-instance` groupings to your collected data to focus on time-intensive computing tasks.
 - [Dynamic instruction count](#) collection (available as part of the GPU Compute/Media Hotspots Analysis) improved to provide better accuracy for basic block Assembly analysis.
 - Support for Intel® Processor Graphics Gen11.
- Platform analysis support:
 - [System Overview](#) analysis updated to serve as an entry point to platform analysis. Use this analysis to assess system (IO, accelerators and CPU) performance and review guidance for next steps.
 - New [Hardware Tracing mode](#) in the System Overview analysis enables application analysis on the micro-second level and identification of causes for latency issues.
- HPC analysis improvements:
 - Max and Bound Bandwidth metrics added to Application Performance Snapshot to better estimate the efficiency of the DRAM, MCDRAM, Persistent Memory and Intel® Omni-Path usage.
- Platform Profiler new features and improvements:
 - Overview and Memory views extended with new metrics to analyze Non-Uniform Memory Access (NUMA) behavior.
 - User authentication and authorization implemented to enable access control to user data.
 - Added a new option for users to choose or modify the location of Platform Profiler data files.
- Energy analysis improvements:
 - New [Throttling analysis](#) added to identify causes for system throttling, including violation of safe thermal or power limits.
 - Options for [Energy analysis](#), based on the Intel SoC Watch data collector, extended to monitor processor package energy consumption over time and identify how it correlates with CPU throttling.
- Cloud and containerization support:
 - [Containerization support](#) extended with an option to install and run VTune™ Profiler in a Docker* container and profile targets inside and outside the same container.

Intel® VTune™ Profiler 2020

- Added support to profile applications running in Amazon Web Services* (AWS) EC2 Instances based on Intel microarchitecture code name Cascade Lake X.
- New [Fabric Profiler](#) performance tool added to VTune™ Profiler in Preview mode. Use Fabric Profiler to identify detailed characteristics of the runtime behavior for an OpenSHMEM application.
- Quality and usability improvements:
 - Symbol resolution for effective source-level analysis enabled for crossgen (Ahead-of-JIT compilation) functions on Linux* systems.
 - Interactive **Help Tour** (available on the Welcome page) guides you through the product interface using a sample project.
- New hardware/operating systems/IDEs support:
 - 10th Gen Intel® Core™ processors
 - Ubuntu* 19.10
 - Microsoft* Windows* 10, November 2019 Update

3 System Requirements

Supported Architectures and Terminology

Intel® VTune™ Profiler 2020 supports the following architecture:

- **Intel® 64 Architecture** refers to systems based on IA-32 architecture processors which have 64-bit architectural extensions (like Intel® Core™ architecture processors, running a 64-bit operating system such as Microsoft Windows* 10 x64 or a Linux* "x86_64" variant).

Hardware Requirements

- At least 8 GB of RAM recommended.
- 1.6 GB free disk space required for all product features and all architectures.
- For general operations with user interface and all data collection except Hardware event-based sampling analysis:
 - A PC based on an Intel® 64 architecture processor supporting the Intel® Streaming SIMD Extensions 2 (Intel® SSE2) instructions (Intel® Pentium® 4 processor or later, or compatible non-Intel processor).
 - For the best experience, a multi-core or multi-processor system is recommended.
 - Because VTune™ Profiler requires specific knowledge of assembly-level instructions, its analysis may not operate correctly if a program contains non-Intel instructions. In this case, run the analysis with a target executable that contains only Intel® instructions. After using VTune™ Profiler, you can use the assembler code or optimizing compiler options that provide the non-Intel instructions.
- For Hardware event-based sampling analysis (EBS):
 - EBS analysis makes use of the on-chip Performance Monitoring Unit (PMU) and requires a genuine Intel® processor for collection. EBS analysis is supported on Intel® Xeon® processors and second generation (or later) Intel® Core™ processors. Note that a system does not need to support EBS analysis collection in order to open and view existing EBS results; systems that meet the less restrictive general operation requirements are sufficient for viewing results.
 - EBS analysis within a virtual machine is available in the following environments virtualizing the on-chip Performance Monitoring Unit (PMU): VMware*, KVM*, XEN*, Hyper-V*, Parallels

Desktop*. See support details in the <https://software.intel.com/en-us/vtune-help-on-virtual-machine>

- Hardware Event-based Sampling (EBS) analysis can run a driverless sampling collection via Linux Perf* tool. The source code for the perf tool version shipped with this release of Intel® VTune™ Profiler can be found on the Software Heritage archive by searching for the persistent id `swH:1:dir:eb948822426ae45875837f885573356602451953` or at <https://archive.softwareheritage.org/browse/directory/12ab80173cf69d1384fc31460c8e9d5df6f1049c/>
- For GPU analyses:
 - All GPU analyses requires a PC based on Intel® 64 architecture processor with 8th generation of Intel HD or Iris Graphics (formerly Broadwell) or newer and Linux kernel 4.14 or newer
- For Android* Targets:
 - Intel Atom® E39XX Processors (code name Apollo Lake)
 - For collection:
 - The collection utilities make use of the Performance Monitoring Unit (PMU) and Architectural Model Specific Registers (MSR) and therefore require a genuine Intel Atom® processor for collection.
 - Collected results can be analyzed using any system meeting the less restrictive general operation requirements.

Supported operating systems

- To collect and analyze profiling data using the graphical and command line user interface of Intel® VTune™ Profiler, use a 64-bit operating system host.
- Intel® VTune™ Profiler does not support 32-bit operating systems. However, Intel® VTune™ Profiler can profile 32-bit applications on 64-bit OSs (cross mode).
- To use the VTune Profiler graphical interface on Linux* OS, ensure that you install the required packages from <https://software.intel.com/en-us/vtune-install-guide-linux-prerequisites>.

Linux* hosts and targets:

- Supported operating systems:
 - Red Hat* Enterprise Linux 7.X and 8.X
 - CentOS* versions equivalent to Red Hat* Enterprise Linux* versions listed above
 - SUSE* Linux* Enterprise Server (SLES) 12.x and 15.x
 - Fedora* 30 and 31
 - Ubuntu* 16.04 LTS, 18.04 LTS and 19.10
 - Amazon Linux* 2.0
 - Clear Linux* OS
 - Debian* 9.0 and 10.0

NOTE: Intel® VTune™ Profiler Platform Profiler host functionality (viewing and analyzing results) does not support Debian* or SUSE* Linux* Enterprise Server.

- For embedded Linux*, the target platform should be based on one of the following environments:

- Yocto Project* 2.7 and 3.0 based environment
- Wind River* Linux* 8 and 9-based environment
- Supported compilers:
 - Intel® C/C++ Compiler 11 and higher
 - Intel® Fortran Compiler 11 and higher
 - GNU* C/C++ Compiler 3.4.6 and higher
- Supported kernels:
 - Hardware event-based sampling analysis with stacks requirements:
 - Linux kernel version 2.6.32 up to 5.0
 - Driverless hardware event-based sampling analysis requirements:
 - Linux kernel version 2.6.32 up to 5.0, exporting CPU PMU programming details over `/sys/bus/event_source/devices/cpu/format` file system

Windows* hosts and targets:

- Supported operating systems:
 - Microsoft* Windows* 10 x64
 - Microsoft* Windows Server* 2016 and 2019
- Supported compilers:
 - Intel® C/C++ Compiler 11 and higher
 - Intel® Fortran Compiler 11 and higher
 - Intel® Parallel Composer
 - Microsoft* Visual Studio* C/C++ Compiler
 - Cygwin (tested Cygwin 1.7.17 with GCC 4.5.3)
 - MinGW (tested MinGW with GCC 4.6.2)
- Supported versions of Microsoft* Visual Studio*:
 - Microsoft* Visual Studio* 2017 (v15.9.10 or above)
 - Microsoft* Visual Studio* 2019 (v16.2.3 or above)

Intel® VTune™ Profiler requires .NET Framework (v4.6.01586 or above) for Microsoft* Visual Studio* integration.

macOS* hosts:

- Supported operating systems:
 - macOS* 10.14.X and 10.15.X

NOTE: Intel® VTune™ Profiler Platform Profiler host functionality (viewing and analyzing results) does not support macOS*.

FreeBSD* targets:

- Supported operating systems:
 - FreeBSD* 11, and 12 based environments on Broadwell server and higher.

Android* targets:

- Supported operating systems:
 - Android* 6.x or higher based environment
- Supported compilers:
 - Intel® C/C++ Compiler 12.1 for Android* and higher
 - GNU* C/C++ Compiler 4.2 and higher for Intel® architecture

- While samples can be collected for any programming language running on the system, code source can only be viewed for the following:
 - C
 - C++
 - Java* (Dalvik and ART*)

Software Requirements

- General requirements:
 - Supported programming languages (see Android* targets bullet for restrictions):
 - Fortran
 - C
 - C++
 - C#
 - Java*
 - OpenCL™
 - Python*
 - Go
 - Threading analysis type interprets the use of constructs from the following threading methodologies:
 - Intel® Threading Building Blocks
 - Posix* Threads on Linux
 - Win32* Threads on Windows*
 - OpenMP*
 - Intel's C/C++ Parallel Language Extensions
 - Supported .NET environments:
 - .NET (Windows only)
 - .NET Core 2.0 and newer
 - Supported Java* environments:
 - Oracle* JVM 6, 7, 8 and 9 – Hotspots and Hardware event-based analysis types
 - OpenJDK* 8 and 9 (Linux only)
 - Supported OpenCL™ environments:
 - Intel® CPU Runtime for OpenCL™ Applications 18.1 and newer
 - Intel® Graphics Compute Runtime for OpenCL™
NOTE: On Windows*, this is part of the GPU driver. On Linux*, it is available on Github*: <https://github.com/intel/compute-runtime>
 - You must have the latest drivers to collect GPU usage events and processor graphics hardware events on Windows*. Request drivers from Intel Support or download them from <https://downloadcenter.intel.com/product/80939/Graphics-Drivers> .
 - Intel® VTune™ Profiler Platform Profiler data collector requires Python* 2.7.
 - To view PDF documents, use a PDF reader, such as Adobe Reader*.

4 Where to Find the Release

If you have a qualifying license for this product (i.e. an unexpired free license or a paid license with active support at the time of the build date for this product release), you can download the installer by logging in to <https://registrationcenter.intel.com/> and selecting the appropriate product.

You may need to create an account and/or register your product. For additional information please see:

- <https://software.intel.com/en-us/faq/registration>
- <https://software.intel.com/en-us/faq/downloading>

5 Installation Notes

For instructions on installing, updating, or removing Intel® VTune™ Profiler, see the appropriate installation guide for your operating system:

- Linux*: <https://software.intel.com/en-us/VTune-Install-Guide-Linux>
- macOS*: <https://software.intel.com/en-us/VTune-Install-Guide-macOS>
- Windows*: <https://software.intel.com/en-us/VTune-Install-Guide-Windows>

Intel® Software Manager

Intel® Software Manager installs with Intel Software Development Products on Windows* operating systems. Intel Software Manager is a utility that allows you to perform these tasks:

- Download and install updates for your Intel® Software Development Products.
- Manage subscription status of installed software.
- Activate serial numbers.
- Find out about the latest news for Intel Software Development Products.

Intel Software Manager requires an Internet connection to connect to a remote server for information and updates.

For more information about the Intel Software Manager, see <https://registrationcenter-ssl.intel.com/Docs/ism.htm>.

Default Installation Folders

The default top-level installation folders for this product are:

- Windows: `C:\Program Files (x86)\IntelSWTools\VTune Profiler 2020`
 - If you are installing on a system with a non-English language version of Windows* OS, the name of the `Program Files` folder may be different.
- Linux: `/opt/intel/vtune_profiler_2020`

In both cases, there is also a generic soft link to the most recently installed version.

Setting Up the Command Line

Before using Intel® VTune™ Profiler from the command line, you must set up the environment. Run the appropriate script for your OS

- Windows: `<installation_dir>\vtune-vars.bat`
- Linux: `source <installation_dir>/vtune-vars.sh`

NOTE: Commands to invoke the command line interface (`amplxe-cl`) and GUI interface (`amplxe-gui`) have been renamed to `vtune` and `vtune-gui` respectively.

6 Known Issues

Use these links to see OS-specific limitations:

- [Windows* Specific Limitations](#)
- [Linux* Specific Limitations](#)
- [macOS* Specific Limitations](#)

Common Limitations

RUNNING TIME IS ATTRIBUTED TO THE NEXT INSTRUCTION (VASP-3854)

To collect data about time-intensive running regions of the target, Intel® VTune™ Profiler interrupts the execution of target threads and attributes the time to the context IP address.

Due to the collection mechanism, the captured IP address points to the instruction *after* the most time-intensive one. This causes the running time to get attributed to the next instruction (or sometimes to subsequent instructions) in the Assembly view. In rare cases, this can also lead to incorrect attribution of running time in the source. The time can be erroneously attributed to the source line *after* the actual hot line.

If the inline mode is ON and the program has small functions inlined at the hotspots, this can cause the running time to be attributed to the wrong function since the next instruction can belong to a different function in tightly inlined code.

VTUNE CRASHES ON FREEBSD 11/12 AFTER PRESSING CTRL-C DURING COLLECTION (VASP-17676)

Upon termination of collection on FreeBSD with a Ctrl-C operation, the next VTune hardware event-based collection fails with this error- PMU resource(s) currently being used by another profiling tool or process: another PMU collection session may be running.

As a workaround, use `-command stop` to stop the collection.

COLLECTION MISSING DATA FOR CORE_0 ON SKYLAKE-X SYSTEMS (VASP-11480)

On some systems based on Intel microarchitecture code name Skylake-X with enabled SMT mode, VTune™ Profiler hardware sampling analysis can collect data against less logical CPUs than available on the system. For example, the problem was reproduced on Intel® Core™ i9-7900X 10C20T CPU installed on an Asus* PRIME X299-DELUXE motherboard.

As a workaround, upgrade the BIOS and the firmware of the board to 0802 or a newer version.

AN APPLICATION THAT ALLOCATES MASSIVE CHUNKS OF MEMORY MAY FAIL TO WORK UNDER VTUNE™ PROFILER

If a 32-bit application allocates massive chunks of memory (close to 2 GB) in the heap, it can fail to launch with VTune™ Profiler but execute independently without error. This happens because VTune™ Profiler requires additional memory when profiling an application. As a workaround, use a larger address space. For example, convert the project to 64-bit.

HARDWARE EVENT-BASED ANALYSIS MAY CRASH CERTAIN INTEL® CORE™ I7 PROCESSOR-BASED SYSTEMS WHEN DEEP SLEEP STATES ARE ENABLED

On some Intel® Core™ i7 processor-based (based on Intel microarchitecture code named Nehalem) systems with C-states enabled, sampling may cause system hanging due to a known hardware issue. To avoid this, disable the “Cn (ACPI Cn) report to OS” BIOS option before sampling with the VTune™ Profiler on Intel Core i7 processor-based systems.

INSTRUCTION SET REFERENCE DOCUMENT IS NOT POSITIONED ON DESCRIPTION OF PROPER INSTRUCTION

The reference information for assembly instructions can be opened in any PDF viewer, but only Adobe* Acrobat Reader* supports the positioning of the instruction reference document on the required page. To ensure correct functionality of this feature, install the latest version of Adobe Acrobat Reader.

VTUNE™ PROFILER DOES NOT SUPPORT PROFILING AN APPLICATION LAUNCHED UNDER DEBUGGER

The Hotspots, Threading, Memory Access with object tracking, and Input and Output analysis types provide incorrect results when analyzing an application launched under a debugger. VTune™ Profiler does not detect whether a debugger is attached to a profiled application. Make sure that no debugging tools are attached to the application that is profiled with the VTune™ Profiler.

ONLY ONE STACK FRAME IS SHOWN IN JAVA* CODE IF IBM* J9* JVM IS USED (VASP-3535)

Currently, IBM* J9* JVM does not support Java* stack unwinding.

RESULTS NOT COLLECTED FOR PROCESSES TERMINATED WITH `TERMINATEPROCESS ()`

No results are displayed for a profiled process that is terminated by a different process calling `TerminateProcess ()`. Use an alternate method to terminate the process.

DO NOT USE `-IPO (LINUX*)//QIPO (WINDOWS*)` OPTION WITH INTEL COMPILER VERSION EARLIER THAN 13.0. (VASP-3534)

If you are using a version of Intel® compiler that is 13.0 or older, use the additional option `-inline-debug-info (Linux*)//debug:inline-debug-info (Windows*)` to get performance data on inline functions. However, avoid using the `/Qipo` option. Currently this option disables the generation of the inline debug information in the compiler. Note that the Intel compiler integrated into the Microsoft Visual Studio* IDE uses the `-ipo//Qipo` by default in the Release configuration.

JAVA* SOURCE LINE MAY BE SHOWN INCORRECTLY FOR ORACLE JDK* 6U25 x64

Drilling down to Java* source code from VTune™ Profiler results may show incorrect source line. The issue occurs with Java* applications built with Oracle JDK* 6u25 x64. This is a JDK* problem. Refer to issues JDK-7051767 and JDK-7047624 in the Oracle bug database.

RESULT FINALIZATION MAY BE SLOW IF RESULTS ARE LOCATED IN A REMOTE NETWORK DIRECTORY (VASP-995)

The problem is caused by slow network access. To improve performance, use local directory for result storage.

CALL STACK CANNOT BE UNWOUND VIA NO RETURN FUNCTIONS (VASP-3350)

If the analyzed application contains functions without `ret` instruction (for example, calling `exit ()`), unwinding the call stack to its caller and higher may fail (no stack frames shown).

ATTACHING THE VTUNE™ PROFILER TO A PROCESS MAY TAKE SOME TIME (VASP-2289)

Sometimes, your target application may complete execution before data collection begins. In this case, VTune™ Profiler may report an error. Increase the duration of your test application if necessary.

Command line option "`vtune -command status`" is currently supported only for user-mode sampling and tracing based analysis types, and not for EBS analysis types.

HARDWARE EVENT-BASED ANALYSIS MAY CAUSE UNPREDICTABLE SYSTEM BEHAVIOR ON PROCESSORS BASED ON INTEL MICROARCHITECTURE CODE NAMED SANDY BRIDGE

On processors based on Intel microarchitecture code name Sandy Bridge, hardware event-based sampling may cause unpredictable system behavior due to a known hardware issue (see erratum BK105 in <http://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/2nd-gen-core-family-mobile-specification-update.pdf>). To avoid this, avoid running Microarchitecture Exploration (formerly, General Exploration) or Custom Hardware event-based analysis with precise events on these systems unless a BIOS workaround for the erratum is provided.

VTUNE™ PROFILER MAY DETECT WHICH TIMER SOURCE TO USE INCORRECTLY ON INTEL® XEON® PROCESSOR E5-XXXX PROCESSORS

In User-mode Sampling and Tracing based analysis (Hotspots, Threading), the following message may be displayed in command line or Debug Window:

Warning: Cannot load data file ``<path>\data.0\18385-18389.0.trace`' (SampleCallback: timestamps aren't ascended!).

As a workaround, collect results from the command line using the `-run-pass-thru` option:

```
vtune -collect hotspots -run-pass-thru=-timestamp=sys -- <application>
```

HARDWARE EVENT-BASED ANALYSIS DOES NOT WORK IF MORE THAN 128 EVENTS ARE GATHERED SIMULTANEOUSLY (VASP-1992)

Decrease number of PMU events in analysis settings to resolve it.

ITT API TASK OR FRAME IS NOT SHOWN IN RESULTS IF ITS END OCCURS WHEN COLLECTOR IS INACTIVE (VASP-3354)

When ITT task or frame end notification occurs after collection was paused, or detach or stop command issued, the frame or task is not displayed in the result.

AN APPLICATION MAY EXPERIENCE THE STACK OVERFLOW EXCEPTION WHEN RUNNING UNDER THE VTUNE™ PROFILER (DOC-200)

An application allocating massive chunks of memory on a thread stack may experience the stack overflow exception and fail when running under the VTune™ Profiler, while running flawlessly on its own. This happens because VTune™ Profiler requires additional space on an application thread stack for profiling needs. To work around this problem, consider using larger thread stack space.

VTUNE™ PROFILER CANNOT LOAD DRIVERS INTO KERNEL

Most Android* kernels only allow signed drivers. If you try and install a driver (`sep3_10.ko`, `pax.ko`, `vtsspp.ko`, `SOCWATCH1_3.ko` or `apwr3_1.ko`), which was built after the kernel was built, you will get an

error message stating that no signing key is found. To work around this problem, you must build drivers with the exact signing key created when the Android* OS (`boot.bin`) was built. Use drivers which have been integrated into the kernel.

DATA COLLECTED WITH PREVIOUS VERSIONS OF PLATFORM PROFILER IS NOT AVAILABLE AFTER UPGRADE TO VTUNE™ PROFILER 2020 (DXET-2014)

Platform Profiler 2020 allows users to choose the location for data storage. Previous versions of Platform Profiler stored the data in these locations:

- **Linux:** `/opt/intel/vtune-Profiler-platform-profiler/vpp-data`
- **Windows:** `C:\ProgramData\Intel\VTune Profiler Platform Profiler\data`

To view the data collected with previous versions of Platform Profiler:

- **Linux:** Start the Platform Profiler server with the `--data-dir` option:
`<vtune install dir>/vpp/server/vpp-server start --data-dir /opt/intel/vtune-Profiler-platform-profiler/vpp-data/`
- **Windows:** In the Platform Profiler server startup dialog, click the "Browse Directory" button. Select "`C:\ProgramData\Intel\VTune Profiler Platform Profiler\data`" and then click "Start" to start the sever.

PLATFORM PROFILER IS UNABLE TO UPLOAD RESULTS OR VIEW RESULT DIRECTORIES AND FILES WITH GOOGLE CHROME VERSION 80 OR LATER (DXET-2045)

Users of Google Chrome version 80 or later are unable to browse Platform Profiler collected results or upload new result files. Clicking the "View Results" on the Platform Profiler home page displays an empty page.

To work around this problem:

- Start Google Chrome with the `--enable-blink-features=HTMLImports`. Prior to launching Google Chrome, make sure to stop all Chrome running processes.
- Use Firefox. Platform Profiler is fully functional with Firefox. Expect some degradation in user experience (e.g. topology diagrams are not fully rendered)

SIZE LIMITATIONS WHEN UPLOADING DATA COLLECTION RESULTS TO THE PLATFORM PROFILER SERVER (DXET-1200)

The Platform Profiler server currently supports data collection results up to 1GB in size (compressed).

DATA IMPORT FAILURES ON SYSTEMS WITH NON-US LOCALES (DXET-2011)

Platform Profiler may fail to import data on systems configured with a numeric locale that uses "." as the thousands separator (e.g. Russian/Belorussian locale). A typical import error in this case is:

```
InfluxDBClientError: 400: {"error":"unable to parse 'CPU//s-operating//s-frequency//s- (in//s-GHz,device=core,package=package0,unit=CPU0,dataset=BC-SR1-4R3-K8S-VA1_20191002-1540 value=2.89 15700200670691000000': strconv.ParseInt: parsing \"15700200670691000000\": value out of range
```

To work around this problem:

1. Configure the system to use the "en_US" locale
2. Restart the Platform Profiler server
3. Upload the data file

Windows* Specific Limitations

JIT WRITER API CANNOT HANDLE TOO LONG FILE NAMES ON WINDOWS (VASP-484)

This issue happens when file name of JIT trace file is longer than 260 symbols. VTune™ Profiler is unable to generate a proper result when the result directory has such a long name.

THREAD STACK SIZE LIMITATION (VASP-1808)

VTune™ Profiler may crash with the following error message:

Error: failed to create a sampling thread: not enough storage is available to process this command.

This happens when the profiled application has big reserved and committed thread stack size (`/STACK:reserve[,commit]` command line switch of `link.exe`). The suggested workaround is reducing the reserved/committed thread stack size to profile the target.

INCORRECT TIMING RESULTS WHEN RUNNING ON A 32-BIT VIRTUAL MACHINE

VTune™ Profiler may fail to collect correct timing data when running on a virtual machine with problematic virtualization of time stamp counters. In this case the VTune™ Profiler throws a warning message:

Warning: Cannot load data file '<path_to_a_trace_file>.trace' (syncAcquiredHandler: timestamps aren't ascended!)

UNINSTALLING LIMITATION: PIN.EXE STAYS RUNNING AFTER DETACHING.

Once attached to a target, VTune™ Profiler cannot be uninstalled until execution on the target is over. The cause is that `pin.exe` keeps working after detaching from the target and exits only after the profiled application/process execution finishes.

SECOND ATTACH TO THE SAME APPLICATION SHOULD PRINT AN ERROR AND EXIT IMMEDIATELY

VTune™ Profiler allows you to run an analysis while the previous one is in progress. Howeverm it does not store any data from the second analysis run.

TRUNCATED .NET* MODULE NAMES MAY BE DISPLAYED IN RESULTS VIEW (VASP-3772)

When viewing results collected for a .NET application, you may observe truncated .NET module names. Ensure that the system was rebooted after installing the .NET application and before profiling with the VTune™ Profiler.

.NET MODULES LOADED DYNAMICALLY VIA REFLECTION API ARE DISPLAYED AS "UNKNOWN" HOTSPOTS (VASP-1945)

If dynamic .NET modules (e.g. loaded via `Assembly.LoadFrom`) are used in target application, VTune™ Profiler will show them as "unknown" functions and modules in hotspots list.

ATTACHMENT AS ADMINISTRATOR TO A PROCESS THAT LAUNCHED BY THE 'SYSTEM' USER FAILS (VASP-476)

You may use a utility from <http://technet.microsoft.com/en-us/sysinternals/bb897553> to profile a system service (for example, w3wp.exe-based code) from the command line using vtune.exe that is located in <product_install_dir>/bin64. Do the following:

1. Configure the w3wp service to run with the permissions you use to log in: Open IIS Manager, right-click an application pool you are using and set “Process model:Identity” to the account under which w3wp needs to be run.
2. Run the w3wp service and make sure you run it with proper credentials and remember its PID.
3. Start data collection:

```
psexec -i 0 /path/to/vtune.exe -c=hotspots -r /path/for/your/data_dir --
target-pid=PID
```

4. Run your workload.
5. Stop data collection:

```
psexec -i 0 /path/to/vtune.exe -command detach -r /path/for/your/data_dir
```

6. Open /path/for/your/data_dir in GUI:

```
<product_install_dir>/bin64/vtune-gui /path/to/your/data_dir
```

LIMITED SUPPORT OF WINDOWS STORE* APPLICATIONS (VASP-531)

VTune™ Profiler supports attaching to Windows Store* applications, but not launching them. Only hardware event-based analysis types are supported.

VTUNE™ PROFILER ANALYSIS MAY FAIL ON MACHINE WITH VERDASYS DIGITAL GUARDIAN* SOFTWARE INSTALLED. (VASP-2701)

User mode sampling and tracing based analysis types (Hotspots, Threading) run from GUI may fail if Verdasys Digital Guardian* software is installed on the machine. However, it is still possible to run the collection from the command line.

You can also use EBS collection with stacks, either from the GUI or command line.

ANALYZED APPLICATION MAY CRASH DURING ANALYSIS IF "SYMANTEC ENDPOINT PROTECTION APPLICATION AND DEVICE CONTROL" SOFTWARE IS INSTALLED (VASP-2286)

The issue may occur with user mode sampling and tracing based analysis types (Hotspots, Threading). Ways to overcome the problem:

- If user mode sampling and tracing based analysis is needed exclude the analyzed application from the list of monitored applications of "Symantec Endpoint Protection Application and Device Control" software.
- Use Hotspots analysis with Hardware Event based Sampling and **Collect stacks** option turned on.

DOCUMENTATION CANNOT BE OPENED IN MICROSOFT INTERNET EXPLORER* 10 ON WINDOWS SERVER* 2012 (DOC-204)

To fix the error, modify security settings in Microsoft Internet Explorer* 10. From **Tools > Internet Options > Security**, add “about:internet” to the list of trusted sites.

CHILD PROCESS ANALYSIS DOES NOT WORK FOR EBS ANALYSIS TYPES IF PROCESS IS RUN FROM SCRIPT USING START COMMAND (DOC-231)

If a process is started from a batch file by the `start` command, it does not have original command interpreter as a parent process. So the new process cannot be identified as a child and not profiled. This is true for all Hardware event-based analyses. Possible workarounds:

- Use user-mode sampling and tracing based analysis types instead
- Use `call` command in a batch file instead of `start`
- Use "attach to process" collection mode to connect directly to the process of interest

RESULTS MAY CONTAIN PROCESS NAME CUT ON THE 14TH CHARACTER (VASP-2702)

VTune™ Profiler may truncate a lengthy process name at the 14th character in results collected with EBS analysis types with stacks. For example, a result should contain `find_hotspots.exe`, but contains only `find_hotspots..` This is an operating system limitation.

WARNING "UNSIGNED DRIVER" DURING INSTALLATION FOR SEPDRV AND VTSS DRIVERS ON WINDOWS* 7 AND WINDOWS SERVER* 2008 R2 (VASP-234)

VTune™ Profiler hardware event-based sampling drivers (`sepdrv.sys` and `vtss.sys`) are signed with digital SHA-2 certificate key for compliance with Windows 10 requirements.

To install the drivers on Windows* 7 and Windows Server* 2008 R2 operating systems, add functionality for the SHA-2 hashing algorithm to the systems by applying Microsoft Security Update 2033929:

<https://technet.microsoft.com/en-us/library/security/3033929>.

Linux* Specific Limitations**SOME FEATURES REQUIRE INSTALLING AS ROOT**

Most of Intel® VTune™ Profiler profiling features work with a non-root install. Many features work on either a genuine Intel processor or a compatible processor.

Some advanced features that use event-based sampling require the latest OS kernel or sampling driver to be installed. Intel Atom® processors also require this driver for analysis.

To install the driver on a system with a genuine Intel processor, launch the installer as root or ask your system administrator to install the driver later. For information on building and setting up the drivers, see

https://software.intel.com/en-us/sep_driver.

SLES 15 MACHINE REBOOTS WHEN STARTING INPUT/OUTPUT ANALYSIS WITH BOTH PERF/SEP (VASP-14205)

When running Input and Output analysis on SUSE* Enterprise Linux (SLES) 15, consider switching to **Custom Analysis** and turn off page faults collection. This helps prevent a system crash that may happen when kernel trace points for page faults are turned on.

PROXY-RELATED ERRORS WHEN STARTING THE PLATFORM PROFILER SERVER (DXET-1338)

On some systems, starting the server generates the following error:

Malformed Proxy variables. Please set your proxy variables with "http" scheme and restart the PPE server
:*[Errno socket error] [Errno 10060] A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond*

To work around this problem:

1. Remove the following proxy-related environment variables and start the server again: `http_proxy`, `https_proxy`, `noproxy`
2. Start the server using the following commands:

```
cd <server install directory> dist/3rd/python-dist/python scripts/start.py -  
-skip-proxy-check
```

SECURITY-ENHANCED LINUX* IS NOT SUPPORTED

Security-enhanced Linux settings (SELinux) are currently not supported by the VTune™ Profiler and need to be either disabled or set to permissive for a successful tool suite installation. If your Linux distribution has SELinux enabled the following error message will be issued by the installer:

Your system is protected with Security-enhanced Linux (SELinux). We currently support only "Permissive" mode, which is not found on the system. To rectify this issue, you may either disable SELinux by - setting the line "SELINUX=disabled" in your /etc/sysconfig/selinux file - adding "selinux=0" kernel argument in lilo.conf or grub.conf files or make SELinux mode adjustment by - setting the line "SELINUX=permissive" in your /etc/sysconfig/selinux file or ask your system administrator to make SELinux mode adjustment. You may need to reboot your system after changing the system parameters. More information about SELinux can be found at <http://www.nsa.gov/selinux/>

VTUNE™ PROFILER USER MODE COLLECTION MAY FAIL TO RUN ON UBUNTU* (VASP-3352)

VTune™ Profiler may fail to collect data for Hotspots and Threading analysis types on the Ubuntu operating system. Once a collection is started, this message appears in the output:

Failed to start profiling because the scope of ptrace() system call application is limited. To enable profiling, please set /proc/sys/kernel/yama/ptrace_scope to 0. See the Release Notes for instructions on enabling it permanently.

To work around this problem for the current session, set contents of the `/proc/sys/kernel/yama/ptrace_scope` `sysctl` file to 0.

To make this change permanent, set `kernel.yama.ptrace_scope` value to 0 at `/etc/sysctl.d/10-ptrace.conf` file using root permissions and reboot the machine.

THREADING ANALYSIS MAY PROVIDE INCORRECT WAIT TIME DATA IF LINUX FUTEXES ARE USED

In certain cases, custom synchronization constructions based on Linux futexes are not detected by the VTune™ Profiler. System libraries may also use such constructions. To ensure Wait time correctness in the Threading analysis, use `syscall(SYS_futex, ...)` API in your code.

VTUNE™ PROFILER MAY CAUSE KVM GUEST VIRTUAL MACHINE TO CRASH (VASP-3806)

In some cases, when you run Event-Based Sampling analysis on a host machine, a guest KVM virtual machine may experience kernel panic. This happens only if precise events are used, for example, with

Microarchitecture Exploration (formerly, General Exploration) analysis type. Upgrade KVM to a newer version; there is a KVM patch that fixes the problem.

SYMBOLS ARE NOT RESOLVED FOR BINARIES REMAPPED WITH HUGETLBFS UTILITY (VASP-707)

`hugetlbfs` remaps segments from executable file to its temporary file system or to anonymous memory regions. VTune™ Profiler can't map those segments to the original executable.

The workaround for this problem is to rename the original executable to the one that `hugetlb` exposes, i.e. the module name that you see in VTune™ Profiler results.

`my_exe` (original executable) → `my_exe_32_2` (file name on `hugetlb` file system in case of sharing between processes)

`my_exe` (original executable) → `libhugetlbfs.tmp.<random hash>` (memory region name in case of anonymous mappings)

RUNNING OTHER PMU-BASED TOOLS SIMULTANEOUSLY WITH HARDWARE EVENT-BASED SAMPLING COLLECTION IS NOT SUPPORTED

The Sampling Driver requires exclusive use of the PMU resources. Running multiple PMU based tools will cause incorrect results and/or system instability.

HOTSPOTS AND THREADING ANALYSIS TYPES MAY NOT WORK ON EXECUTABLES THAT DO NOT DEPEND ON THE LIBPTHREAD.SO.0 LIBRARY

There is currently a limitation in the product regarding profiling application targets where the executable does not depend on the `libpthread.so.0` library. The message "*Link libpthread.so to the application statically and restart profiling*" appears when profiling an application where program image does not depend on `libpthread.so.0` but then it `dlopen()`s a shared library which does depend on `libpthread.so.0`. The collector is not able to follow the program execution and module load/unload so the collection results are likely to be misleading. A workaround is to set "`LD_PRELOAD=libpthread.so.0`" before running the collection.

OPENING RESULTS SIGNIFICANTLY DELAYED ON LINUX SYSTEMS CONFIGURED WITH "LARGE PAGES"

To work around this issue make sure that you have `hugectl` utility and `libhugetlbfs` library on the system. VTune™ Profiler can be started this way:

```
hugectl --heap <vtune install dir>/bin64/vtune-gui
```

```
LD_PRELOAD=libhugetlbfs.so <vtune install dir>/bin64/vtune-gui
```

HELP TOC IS TRUNCATED IN SOME LINUX BROWSERS

When using certain HTML browsers to view the VTune™ Profiler offline documentation, if the Contents tab width appears truncated and hides some of the topic titles:

- Click the **Index** (or **Search**) button
- Click the **Contents** button

COLLECTION FAILS WITH ERROR "NO CONFIG_PERF_EVENTS=Y KERNEL SUPPORT CONFIGURED" (DOC-241)

Driverless EBS collection on Linux can fail with the error: "No CONFIG_PERF_EVENTS=y kernel support configured" even if the kernel config has been set. This is a known Linux kernel issue that is resolved in kernel version 4.1. There is a simple patch that provides a workaround: <http://halobates.de/ivb-allow-mem-load-uops>

USER TASKS, EVENTS AND FRAMES MARKED WITH ITT APIS MAY NOT BE SHOWN IF TARGET APPLICATION IS FINISHED BY CTRL+C (VASP-1367)

Use another method to stop application, or VTune™ Profiler `stop` command.

WHEN INSTALLING ON A XEN* VIRTUAL MACHINE, IT MAY HAPPEN THAT INSTALLATION SCRIPT EXITS CLAIMING CPU IS UNSUPPORTED

To skip the check for CPU model, add `--ignore-cpu` parameter to install script command line:

```
./install.sh --ignore-cpu
```

VTUNE™ PROFILER HELP DOES NOT OPEN FROM GUI ON SOME LINUX SYSTEMS (VASP-3322)

On some Linux* systems, like Wind River Linux, default ways to open web browser may not work. You may export the environment variable `BROWSER` (before starting VTune™ Profiler) with the path to the web browser to get VTune™ Profiler help working.

For example:

```
$ export BROWSER=/usr/bin/firefox
```

VTUNE™ PROFILER USER APIS ARE NOT SUPPORTED FOR STATIC BINARIES (VASP-2287)

To use ITT and JIT APIs with software-based collectors (Hotspots, Threading), you must link their application dynamically. Otherwise ITT and JIT API notifications will be missing in the result.

VTUNE™ PROFILER TARGET EXECUTABLES DO NOT WORK ON YOCTO PROJECT X64 DUE TO DIFFERENT PATHS TO LD (VASP-8531)

Error message: "`-sh: ./amplxe-perrunss: No such file or directory`". Usually `ld` is located in `/lib64/ld-linux-x86-64.so.2` but on Yocto x64 it is located in `/lib/ld-linux-x86-64.so.2`. The workaround is to create `/lib64/ld-linux-x86-64.so.2` as a symlink to `/lib/ld-linux-x86-64.so.2`.

EVENT-BASED SAMPLING DRIVER FAILS TO BUILD ON YOCTO PROJECT 1.6 INTEL X86_64 BSPs DUE TO MISSING CONFIG_COMPAT #DEFINE IN LINUX KERNEL HEADERS (VASP-1558)

Sampling driver fails to build on Yocto x86_64 BSPs images that do not have "CONFIG_COMPAT=y" setting (https://bugzilla.yoctoproject.org/show_bug.cgi?id=6777)

To resolve the issue, you need to re-build the image with the option manually.

Change kernel configuration according to <http://www.yoctoproject.org/docs/1.6.1/kernel-dev/kernel-dev.html#changing-the-configuration> :

1. Create the `compat.cfg` file in `meta-yocto-bsp/recipes-kernel/linux/files/` directory with content:

Intel® VTune™ Profiler 2020

```
CONFIG_IA32_EMULATION=y
```

```
CONFIG_COMPAT=y
```

2. Add the following strings to meta-yocto-bsp/recipes-kernel/linux/linux-yocto_3.10.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/files:"
```

```
SRC_URI += file://compat.cfg
```

3. Clean and build a modified kernel according to <http://www.yoctoproject.org/docs/1.6.1/dev-manual/dev-manual.html#patching-the-kernel> and <http://www.yoctoproject.org/docs/1.6.1/dev-manual/dev-manual.html#build-the-modified-qemu-kernel-image> :

```
$ bitbake linux-yocto -c cleansstate
```

```
$ bitbake -k linux-yocto
```

4. Build image:

```
$ bitbake core-image-sato
```

COLLECTING PERFORMANCE DATA FOR INTEL® OPTANE™ DC PERSISTENT MEMORY (DXET-1844)

If you are using an account with root privileges and you try to collect data on a system enabled with Intel® Optane™ DC Persistent Memory, this warning appears:

User needs sudo privilege to collect Persistent Memory metrics. Persistent Memory metrics will not be collected.

Use the following command to start data collection with elevated privileges:

```
sudo <vtune install dir>/vpp/collector/vpp-collect start
```

Use the following command to stop data collection:

```
sudo <vtune install dir>/vpp/collector/vpp-collect stop
```

NOTE: elevated privileges are required to stop a data collection session that was started with elevated privileges.

macOS* Specific Limitations

NATIVE COLLECTION IS NOT AVAILABLE ON MACOS*

Intel VTune™ Profiler for macOS* supports viewing of results collected on other OSs. Native collection on macOS is not currently available.

SOME SYSTEMS DO NOT SUPPORT CONTEXT-SENSITIVE HELP

On some macOS* systems, such as macOS 10.13, VTune™ Profiler context-sensitive help is not supported.

7 Attributions

The license information for third party software that was used to develop Intel® VTune™ Profiler can be found in <install_dir>/licensing/third-party-programs.txt .

For the avoidance of doubt, the Intel® VTune™ Profiler is solely governed by the terms and conditions of the End User License Agreement for Intel® Software Development Product that accompanies the Intel® VTune™ Profiler.

8 Legal Information

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

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

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.

The products and services described may contain defects or errors which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.html.

Intel, the Intel logo, Intel Core, Intel Xeon Phi, VTune, and Xeon 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.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

Copyright 2010 - 2020 Intel Corporation.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you (**License**). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.