

# Profiling

Soner Steiner

Intel oneAPI Certified Instructor

VSC

December 2023



# The Cookbooks

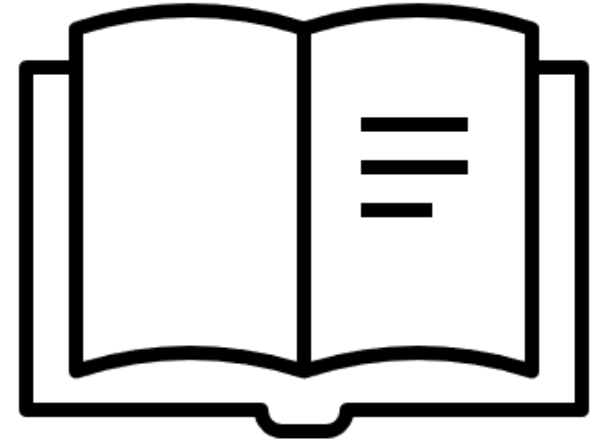
## ▪ **VTune Cookbook**

- Methodologies
- Configuration
- Tuning

- <https://www.intel.com/content/www/us/en/develop/documentation/vtune-cookbook>

## ▪ **Advisor Cookbook**

- <https://www.intel.com/content/www/us/en/develop/documentation/advisor-cookbook>



# Advisor Cookbook Contents

- [Use Intel Advisor Command Line Interface to Analyze a GPU Application](#)
- [Use Intel Advisor Command Line Interface to Model GPU Performance](#)
- [Model GPU Application Performance for a Different GPU Device](#)
- **[Estimate the C++ Application Speedup on a Target GPU](#)**
- [Identify Code Regions to Offload to GPU and Visualize GPU Usage](#)
- [Identify Bottlenecks Iteratively: Cache-Aware Roofline](#)
- [Optimize Vectorization Aspects of a Real-Time 3D Cardiac Electrophysiology Simulation](#)
- [Optimize Memory Access Patterns using Loop Interchange and Cache Blocking Techniques](#)
- [Visualize Performance Improvements with Roofline Compare](#)
- [Analyze Vectorization and Memory Aspects of an MPI Application](#)
- [Analyze Performance Remotely and Visualize Results on a Local macOS\\* System](#)
- [Analyze Performance in Amazon Web Services\\* \(AWS\\*\) EC2\\* Instances](#)
- [Analyze Performance on Cray\\* Systems](#)

# VTune Cookbook Methodologies

- [Top-down Microarchitecture Analysis Method](#)
- [OpenMP\\* Code Analysis Method](#)
- [Software Optimization for Intel® GPUs \(NEW\)](#)
- [Core Utilization in DPDK Apps](#)
- [PCIe Traffic in DPDK Apps](#)
- [DPDK Event Device Profiling](#)
- [Effective Utilization of Intel® Data Direct I/O Technology](#)
- [Compile a Portable Optimized Binary with the Latest Instruction Set](#)

# VTune Cookbook Configuration Recipes - I

- **Analyzing Hot Code Paths Using Flame Graphs (NEW)**
- Improving Hotspot Observability in a C++ Application Using Flame Graphs
- Profiling Games built with Unity\* (NEW)
- Profiling Games built with Unreal Engine\* (NEW)
- Profiling Java Applications as a Remote User (NEW)
- Profiling JavaScript\* Code in Node.js\*
- Measuring Performance Impact of NUMA in Multi-Processor Systems (NEW)
- Analyzing CPU and FPGA (Intel® Arria® 10 GX) Interaction
- Profiling a .NET\* Core Application

# VTune Cookbook Configuration Recipes - II

- [Profiling Applications in Amazon Web Services\\* \(AWS\) EC2 Instances](#)
- [Enabling Performance Profiling in GitLab\\* CI](#)
- [Configuring a Hyper-V\\* Virtual Machine for Hardware-Based Hotspots Analysis](#)
- [Profiling an Application for Performance Anomalies \(NEW\)](#)
- [\*\*Profiling an OpenMP\\* Offload Application running on a GPU \(NEW\)\*\*](#)
- [Profiling a SYCL\\* Application running on a GPU](#)
- [Using the Command-Line Interface to Analyze the Performance of a SYCL\\* Application running on a GPU \(NEW\)](#)

# VTune Cookbook Configuration Recipes - III

- [Profiling an FPGA-driven SYCL\\* Application](#)
- [Profiling Hardware Without Intel Sampling Drivers](#)
- [Profiling MPI Applications](#)
- [Profiling Docker\\* Containers](#)
- [Profiling a Remote Target Through a Proxy Server \(NEW\)](#)
- [\*\*Using Intel® VTune™ Profiler Server with Visual Studio Code and Intel® DevCloud for oneAPI \(NEW\)\*\*](#)
- [Using Intel® VTune™ Profiler Server in HPC Clusters](#)
- [Profiling in a Singularity\\* Container](#)
- [Profiling Linux\\*, Android\\*, and QNX\\* System Boot Time](#)

# VTune Cookbook Tuning Recipes -I

- [Cache-Related Latency Issues in Segmented Cache Environment](#)
- [False Sharing](#)
- [Frequent DRAM Accesses](#)
- [Poor Port Utilization](#)
- [Page Faults](#)
- [Instruction Cache Misses](#)
- [Inefficient Synchronization](#)

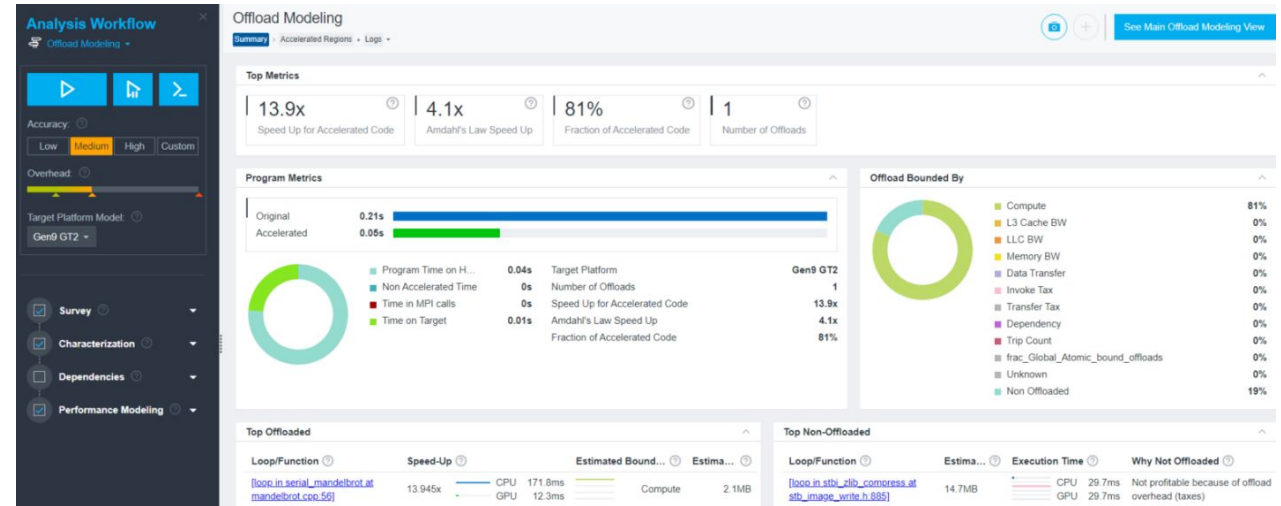


# VTune Cookbook Tuning Recipes II

- Inefficient TCP/IP Synchronization
- OS Thread Migration
- OpenMP\* Imbalance and Scheduling Overhead
- Processor Cores Underutilization: OpenMP\* Serial Time
- Scheduling Overhead in Intel® Threading Building Blocks (Intel® TBB) Apps
- PMDK Application Overhead

# Estimate the C++ Application Speedup on a Target GPU

1. Prerequisites.
2. Compile the C++ Mandelbrot sample.
3. Run Offload Modeling without Dependencies analysis.
4. View estimated performance results.
5. Run Offload Modeling with Dependencies analysis.
6. Rewrite the code in SYCL.
7. Compare estimations and real performance on GPU.



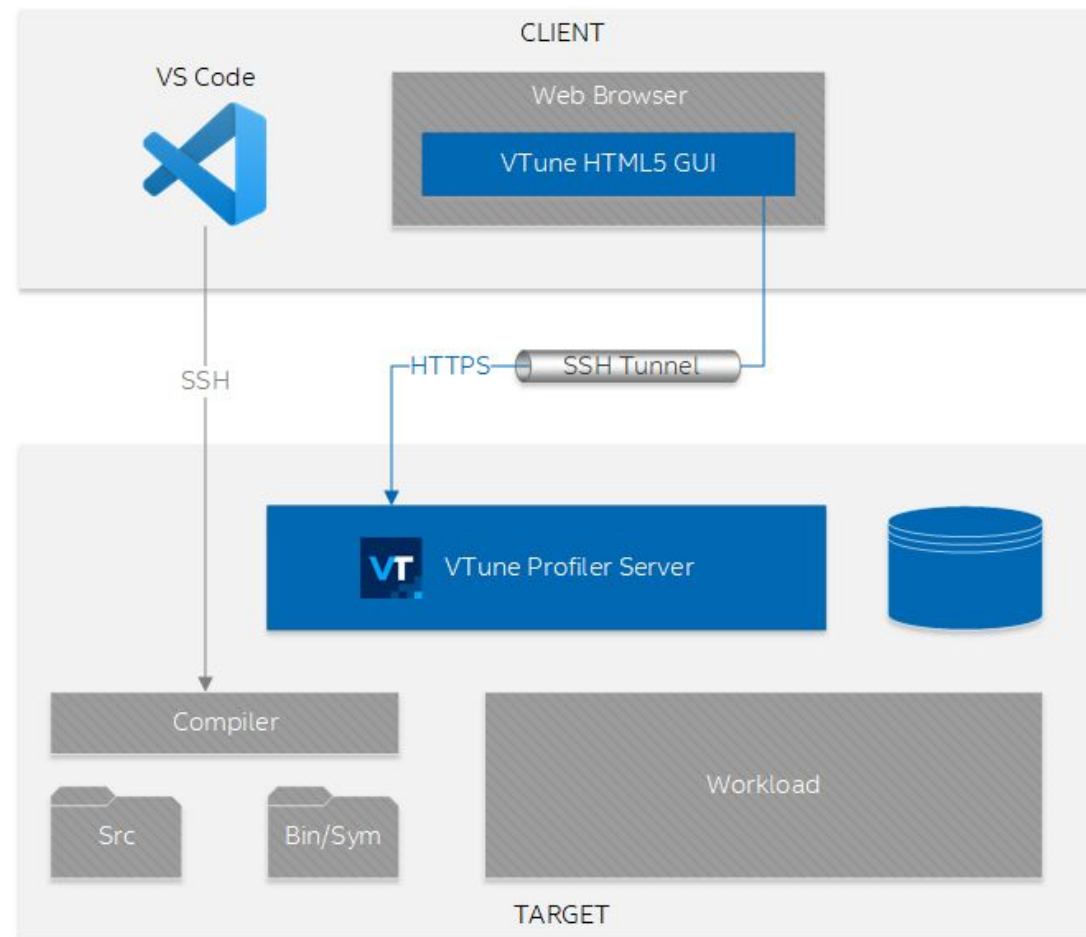
Source		Top-Down				
Line	Source	Is Offloaded	Speed-Up	Time		
52	<code>_mm_malloc(width * height * sizeof(unsig</code>					
53						
54	<code>// Traverse the sample space in equally spac</code>					
55	<code>// samples</code>					
56	<code>for (int j = 0; j &lt; height; ++j) {</code>	Yes	13.945x	12.3ms		
57	<code>for (int i = 0; i &lt; width; ++i) {</code>					
58	<code>double z_real = x0 + i * xstep;</code>					

# Using Intel® VTune™ Profiler Server with Visual Studio Code and Intel® DevCloud for oneAPI (NEW)

- [INGREDIENTS](#)

- DIRECTIONS:

- [Setup Overview](#)
- [Option 1: Use Intel® VTune™ Profiler Server for Remote Development with Visual Studio Code](#)
- [Option 2: Use Intel® VTune™ Profiler Server on a Remote System via SSH Terminal](#)
- [Finish Setup](#)
- [Usage Considerations](#)

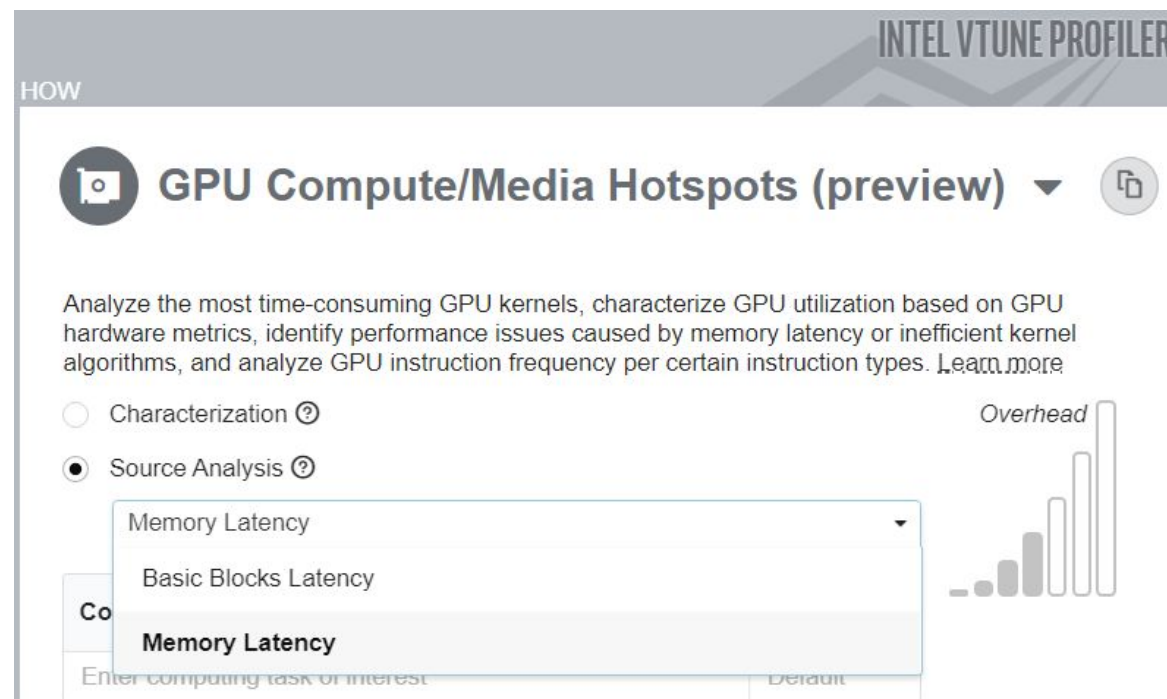


# Profiling an OpenMP\* Offload Application running on a GPU (NEW)

## •INGREDIENTS

## •DIRECTIONS:

- 1.[Build and Compile an OpenMP Offload Application](#)
- 2.[Run HPC Performance Characterization Analysis on the OpenMP Offload Application](#)
- 3.[Analyze HPC Performance Characterization Data](#)
- 4.[Run GPU Offload Analysis on the OpenMP Offload Application](#)
- 5.[Analyze GPU Offload Analysis Data](#)
- 6.[Run GPU Compute/Media Hotspots Analysis](#)
- 7.[Analyze Your Compute Task](#)

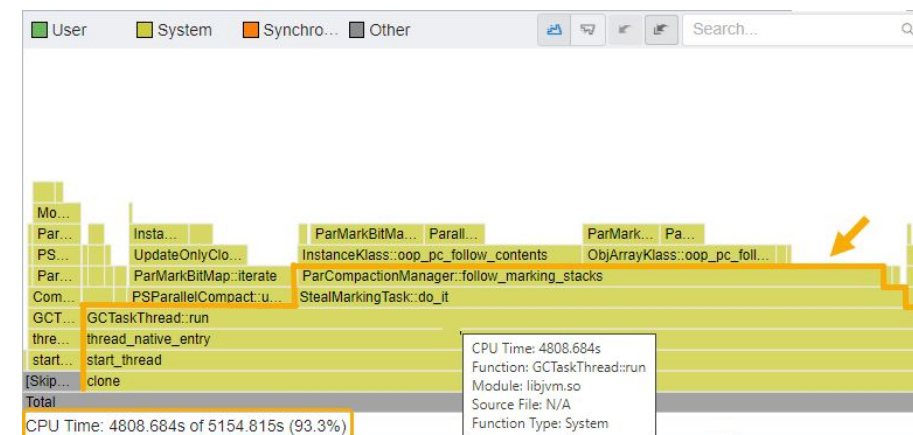
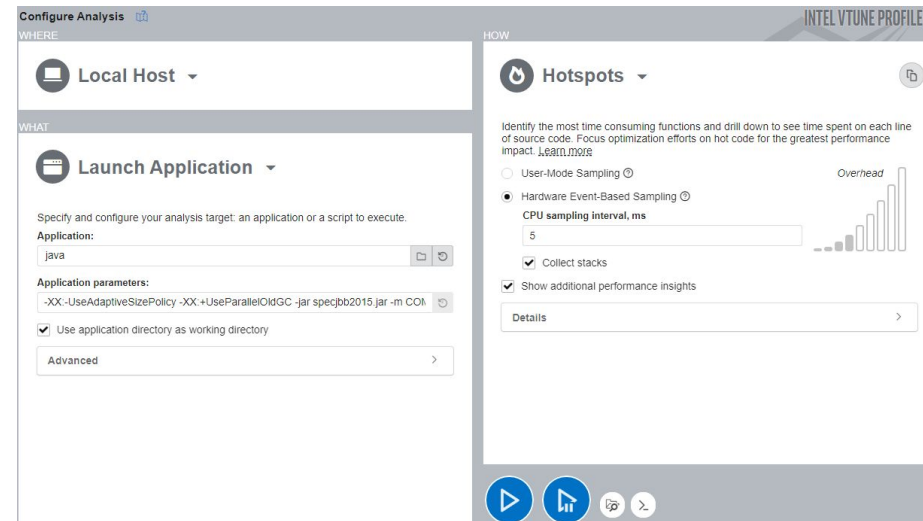


# Analyzing Hot Code Paths Using Flame Graphs (NEW)

## • INGREDIENTS

## • DIRECTIONS:

1. Create a Baseline
2. Run Hotspots Analysis
3. Analyze Hotspots Information
4. Identify Hot Code Paths in the Flame Graph
5. Change JVM Options





# Notices & Disclaimers

Performance varies by use, configuration, and other factors. Learn more at [www.Intel.com/PerformanceIndex](https://www.intel.com/PerformanceIndex).

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.