



# USING THE LATEST PERFORMANCE ANALYSIS TOOLS TO PREPARE FOR INTEL® OPTANE™ DC PERSISTENT MEMORY

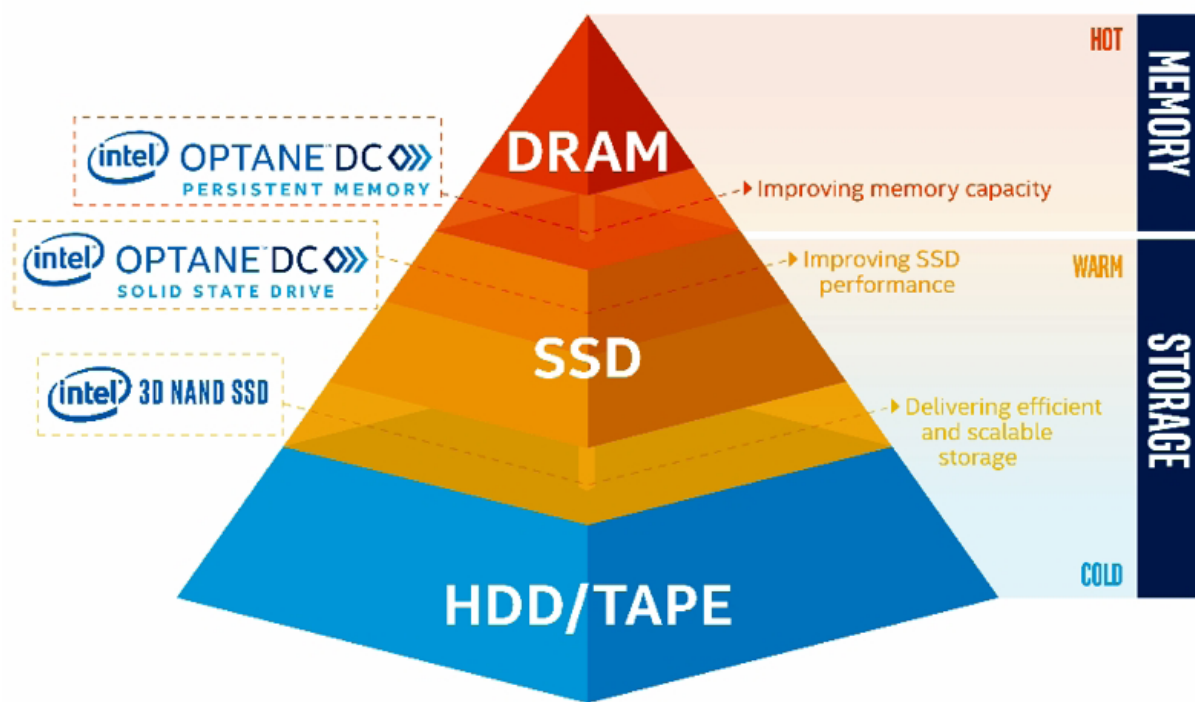
## Getting Past Bottlenecks and Storage Issues

*Jackson Marusarz, Technical Consulting Engineer, and Kevin O'Leary, Senior Technical Consulting Engineer, Intel Corporation*

We have some good news and some bad news. First, the bad news: With the exponential growth in data year after year, and advances in fields like data analytics and artificial intelligence, many applications are becoming bottlenecked by the available system memory or fast storage on a platform. The good news: **Intel® Optane™ DC persistent memory** has arrived.

This new technology introduces a nonvolatile memory/storage tier that's faster than SSDs or hard drives, with latencies near DRAM and much larger capacity. It has implications for any workloads that are currently bound by memory capacity or the slow speeds of storage devices.

**Figure 1** shows how Intel Optane DC persistent memory slots into the memory hierarchy of current platforms. This article will help you understand how you can use Intel® tools to profile your existing workloads and evaluate how they can benefit from this new hardware.



## 1 The new memory hierarchy

Intel Optane DC persistent memory can be configured in two different modes:

- 1. **Memory Mode**
- 2. **App Direct Mode**

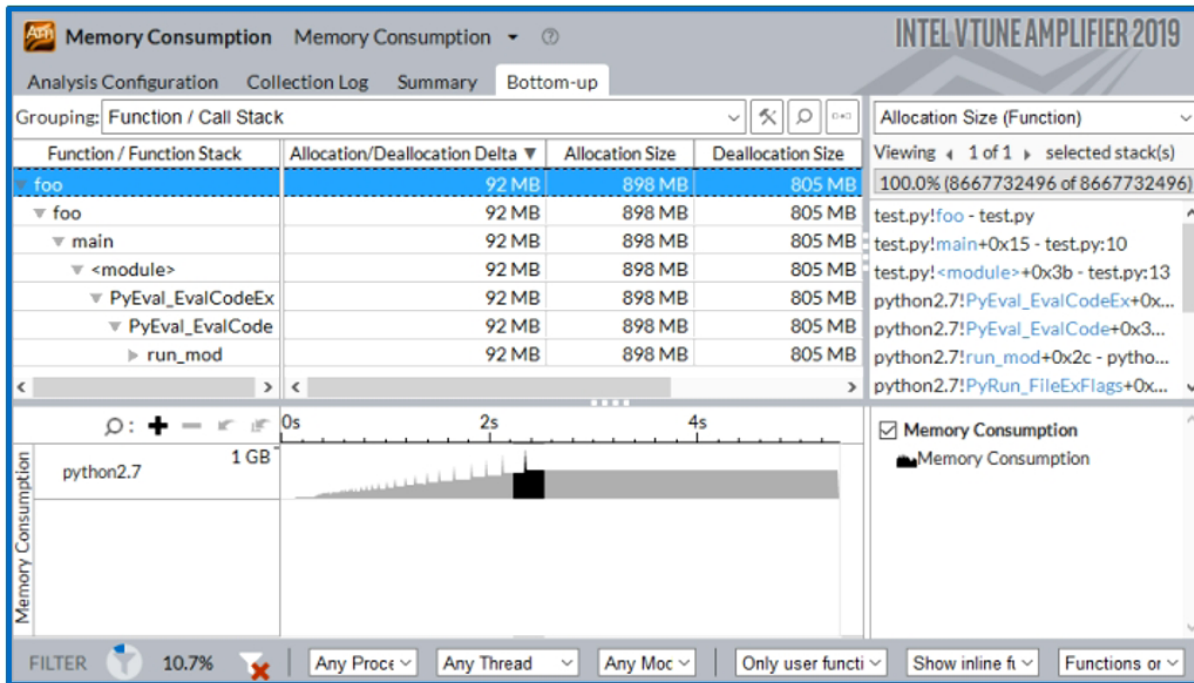
In Memory Mode, Intel Optane DC persistent memory extends the system memory available to the operating system. DRAM is used as a cache for Intel Optane DC persistent memory, and all the memory management is transparent to the user. No code modifications are required.

In App Direct Mode, users manually allocate objects on Intel Optane DC persistent memory via APIs and can also use the memory as traditional storage. This mode enables the non-volatile (persistent) capabilities of the technology.

To determine how your workloads can benefit from Intel Optane DC persistent memory, and which mode to choose, it's important to characterize the behavior and understand specific performance metrics. Intel has tools to help with this process.

## Measure the Memory Footprint of the Application

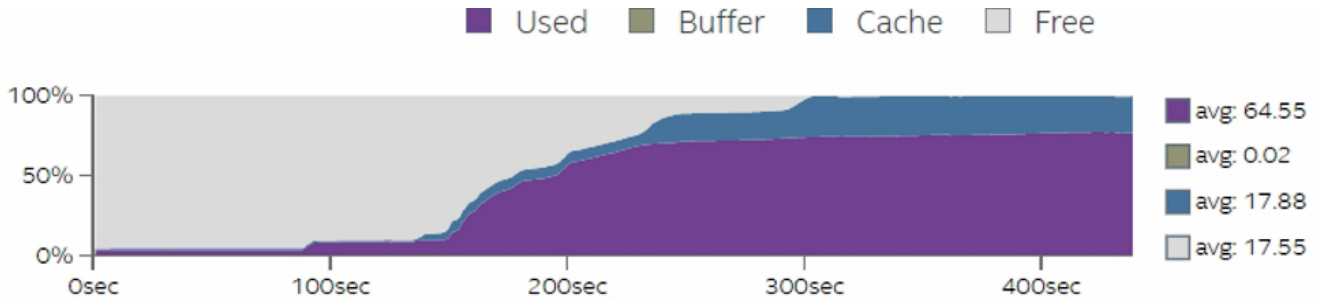
If you're planning to use Intel Optane DC persistent memory as additional system memory—in either mode—the first metric to understand is the memory footprint of your workload. There are many tools that can measure memory consumption, including **Intel® VTune™ Amplifier**. The **Memory Consumption analysis** in Intel VTune Amplifier will monitor the allocations and deallocations of an application and track the memory consumption over time (**Figure 2**).



### 2 Memory Consumption report

The timeline in the Bottom-Up view of the Memory Consumption report can be used to identify the high-water mark of memory usage for the workload. Also, the **Platform Profiler** feature in Intel VTune Amplifier can track memory consumption using OS statistics and provide a timeline as a percentage of available memory (**Figure 3**).

To improve performance with Intel Optane DC persistent memory, the application should benefit from more physical memory. This means the memory consumption should be close to the total amount of DRAM available on the system. Since physical memory is a finite resource, you need to consider that the operating system and other processes also consume memory. If the memory footprint, plus the expected usage of these other memory consumers, is near the available DRAM size, it ensures the application can use the Intel Optane DC persistent memory because it can't fit all of its data in DRAM. If available memory isn't the limiting factor for your workload, then adding more memory probably isn't going to improve performance.



### 3 Platform Profiler Memory Utilization analysis

## Identify the Hot Working Set Size

If you determine that your workload is consuming most of the available memory, then you may have a good candidate for Intel Optane DC persistent memory.

The next step is to determine how your application might behave in each mode, Memory or App Direct. The key metric for this step is the hot working set size. The hot working set is made up of the set objects frequently accessed by your application. And the hot working set size is the sum of the sizes of these objects. This metric isn't as straightforward to calculate as the footprint, since the line of what is frequently and infrequently accessed isn't always clearly defined. However, the Memory Access Analysis in Intel VTune Amplifier, with the knob to analyze dynamic memory objects enabled, can help.

After running a Memory Access analysis, the Bottom-Up view in the GUI will display a grid that lists each memory object that was allocated by the application, its size in parentheses, and the number of loads and stores that accessed it (**Figure 4**). Identify the objects with the most loads and stores. Sum up the sizes (the values in parentheses) of these objects to get the hot working set size.

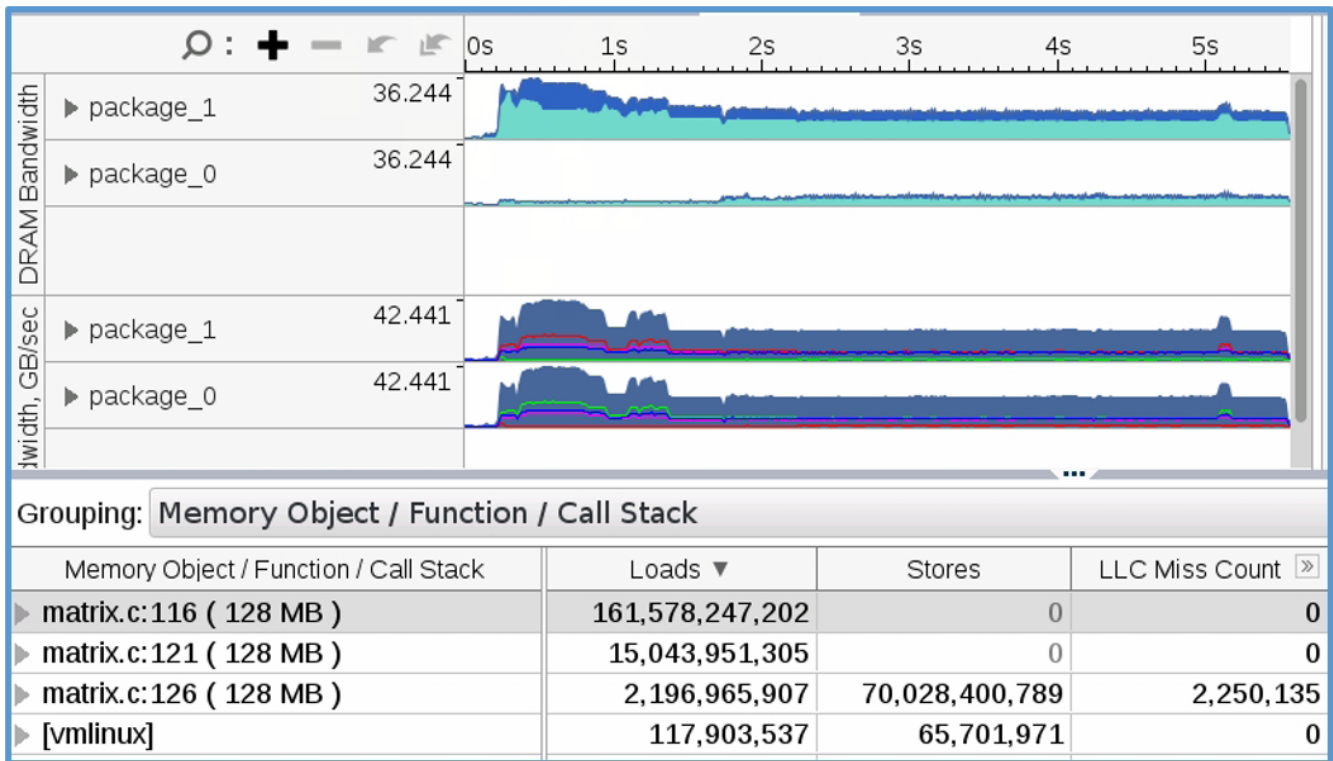
The size of your hot working set is important for determining how your application will behave in each of the memory modes.

## Considerations for Choosing a Memory Configuration and Mode

The important concept to remember when you're thinking about persistent memory performance is that you still want the majority of memory accesses to come from DRAM. The persistent memory acts as additional memory that can be used when DRAM isn't available.

Based on that concept, Memory Mode could be a good solution for applications whose hot working set fits into DRAM (i.e., the hot working set size calculated in the last step should be smaller than the available DRAM

on the system). This will ensure that the working set will routinely be cached in DRAM and, as long as the memory footprint is smaller than the available persistent memory, the remaining data will sit in Intel Optane DC persistent memory instead of out on disk.



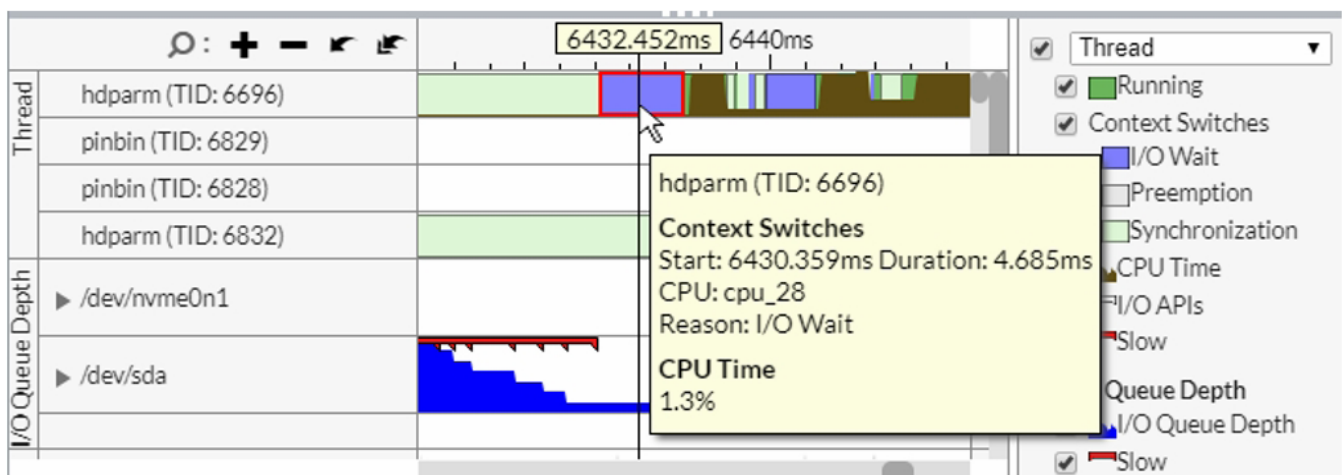
**4 Memory Access Analysis report with Dynamic Memory Object Analysis**

If the hot working set size is much larger than the available DRAM, it's a good indication that persistent memory in App Direct mode could be a better solution than Memory mode. App Direct mode requires the user to explicitly define which objects should be allocated in DRAM and which should be allocated in Intel Optane DC persistent memory. It's important to make educated choices, since allocating incorrectly could hurt application performance. A good starting heuristic for choosing where to allocate objects is identifying the objects with the most last-level core cache (LLC) misses and allocating as many as possible into the available DRAM. The Memory Access analysis in Intel VTune Amplifier (**Figure 4**) has this information. This ensures they will have lower access latency compared to the latency of Intel Optane DC persistent memory. As for the remaining objects that have fewer LLC misses or are too large to put in DRAM, allocate them in Intel Optane DC persistent memory.

One additional consideration for allocation is the load/store ratio for object accesses. Intel Optane DC persistent memory loads are generally much faster than stores. Identify objects with high load/store ratios (load-heavy objects) and allocate them in persistent memory. Allocate the store-heavy objects in DRAM. The load and store counts can also be found with the Memory Access analysis.

## Using Intel Optane DC Persistent Memory for Non-Volatile Storage

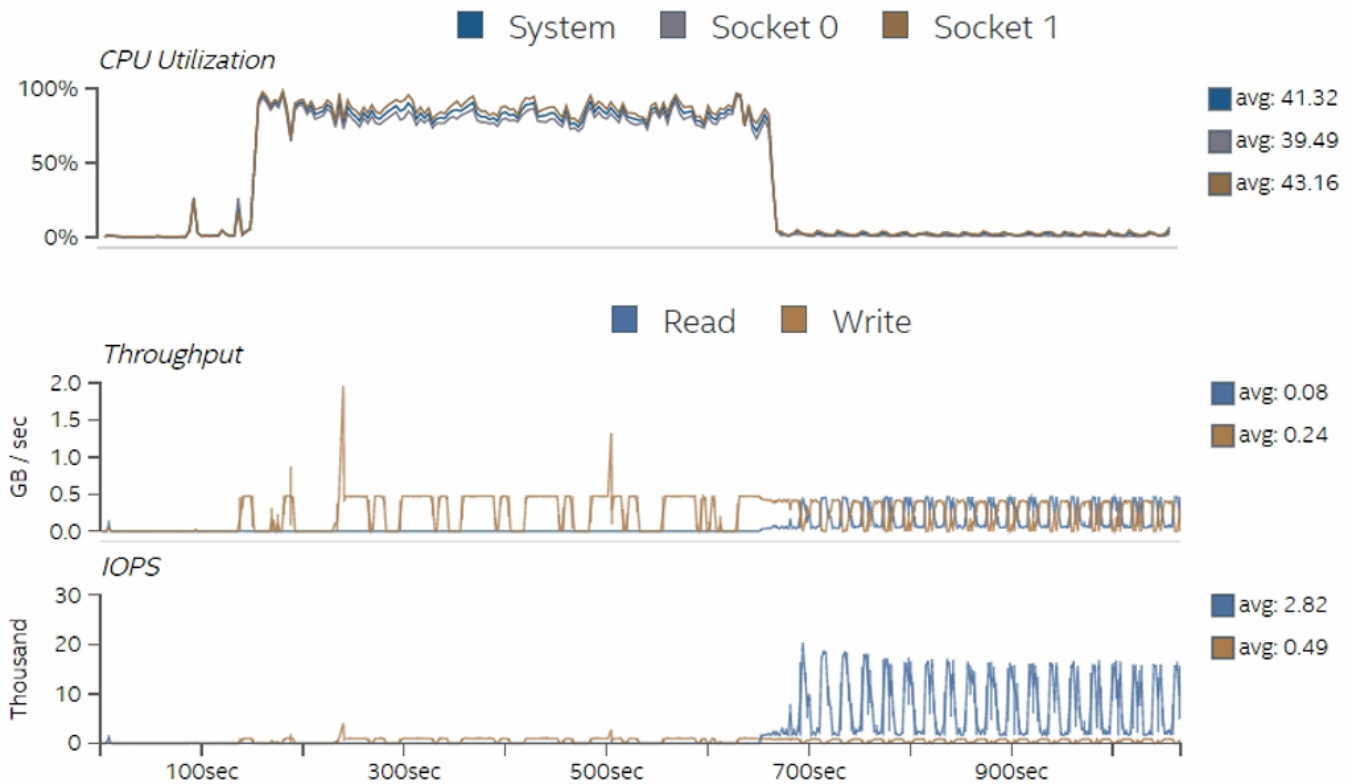
The uses for Intel Optane DC persistent memory as non-volatile storage are fairly straightforward. If your application has any performance issues related to reading and writing to disk, this new technology could give you a boost. Many developers are already aware of disks being their bottleneck. If this is you, then you're one step ahead. If you aren't sure whether storage is causing performance issues, there are features in Intel® tools to help. For instance, the **Input and Output Analysis** in Intel VTune Amplifier helps diagnose CPU stalls correlated with disk accesses (**Figure 5**).



### 5 Intel VTune Amplifier Input and Output analysis

Also, the Platform Profiler analysis in Intel VTune Amplifier displays disk statistics that can be correlated with CPU performance (**Figure 6**).

Use these metrics to identify performance bottlenecks from storage accesses. If this is causing a significant performance issue, using Intel Optane DC persistent memory as fast and persistent storage could increase performance. The persistent memory can be configured as part of the filesystem, and you can put your most accessed files directly on the memory modules.



## 6 Platform Profiler metrics for CPU utilization and disk usage

### Verifying the Correctness of Persistent Memory Applications

Besides identifying performance issues, there are also some software challenges to programming persistent memory applications. One challenge is that a store to persistent memory is not actually persistent until after it's out of the cache hierarchy and visible to the memory subsystem. **Intel® Inspector Persistence Inspector** is a new runtime tool developers can use to detect potential errors (**Figure 7**). In addition to cache flush misses, this tool detects:

- **Redundant** cache flushes and memory fences
- **Out-of-order** persistent memory stores
- **Incorrect** undo logging for the Persistent Memory Development Kit (PMDK)

The screenshot shows the Intel Inspector interface. At the top, the 'Problems' window lists three issues:

ID	Type	Sources	Modules	State
P1	Missing cache flush	MSVCR120D.dll:0x4B...	MSVCR120D.dll	New
P2	Missing cache flush	trace.pmem.cpp:201...	tachyon.exe	New
P3	Missing cache flush before unmap()	pmem_windows.cpp...	tachyon.exe	New

Below this, the 'Code Locations: Missing cache flush' window is open, showing two code snippets:

**Controlled variable** (trace.pmem.cpp:243, do\_render, tachyon.exe)

```
241 color_t *pixel = srb[width * y + x];
242 *pixel = render_one_pixel(x, y, local
243 rs->pixels_stored++;
244 drawing.put_pixel(*pixel);
245 }
```

**Unflushed memory store** (trace.pmem.cpp:242, do\_render, tachyon.exe)

```
240 {
241 color_t *pixel = srb[width * y + x];
242 *pixel = render_one_pixel(x, y, local
243 rs->pixels_stored++;
244 drawing.put_pixel(*pixel);
```

## 7 Intel® Inspector Persistence Inspector

### Getting Past Bottlenecks and Storage Issues

We've just scratched the surface of the possibilities this new technology enables. If you've been struggling with the rise of big data and have performance issues related to limited system memory or fast storage, Intel Optane DC persistent memory is here to help. Intel also has tools like Intel VTune Amplifier and Intel Inspector to help you understand how your workloads may be limited by these issues and how you can take advantage of persistent memory.

To learn more, check out the Intel Optane DC persistent memory [webpage](#) and the software tools [landing page](#).