Intel® Xeon® 6 Processors – Performance and Power Profiles

Default, Latency-Optimized Mode, and Other Options

Technical Article

April 2025

Revision 1.06

Document Number: 826934

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

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

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more on the Performance Index site.

Your costs and results may vary.

"Conflict-free" refers to products, suppliers, supply chains, smelters, and refiners that, based on our due diligence, do not contain or source tantalum, tin, tungsten or gold (referred to as "conflict minerals" by the U.S. Securities and Exchange Commission) that directly or indirectly finance or benefit armed groups in the Democratic Republic of the Congo or adjoining countries.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

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.

Altering clock frequency or voltage may void any product warranties and reduce stability, security, performance, and life of the processor and other components. Check with system and component manufacturers for details.

Intel® Turbo Boost Technology requires a PC with a processor with Intel Turbo Boost Technology capability. Intel Turbo Boost Technology performance varies depending on hardware, software, and overall system configuration. Check with your PC manufacturer on whether your system delivers Intel Turbo Boost Technology.

Results have been estimated or simulated.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel contributes to the development of benchmarks by participating in, sponsoring, and/or contributing technical support to various benchmarking groups, including the BenchmarkXPRT Development Community administered by Principled Technologies.

SPECpower®, SPEjbb® and SPEC CPU® are registered trademarks of the Standard Performance Evaluation Corporation. See www.spec.org for more information.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting the Resource and Documentation Center.

Contents

Figures

Tables

Revision History

Revision Number	Description	Date
1.06	• Fixed typo in Table 2-1.	April 2025
1.05	• Added note on Latency Optimized Mode and Energy Performance Bias (EPB) in Chapter 3.	February 2025
1.04	• Added a new profile for Energy Efficiency Benchmark in Table 4.7. • Added section 3.2. BIOS Knob for Latency Optimized Mode.	January 2025
1.03	• Updated Figure 3-1 for clarity, Added Table 6-2. And other minor updates.	October 2024
1.02	• Added instance option for pcm-tpmi in Table 4.2.	July 2024
1.01	• Removed Section 5.1.	July 2024
1.0	• Initial release of the document.	July 2024

1 Introduction

With the following generation of Intel® Xeon® processors—the Intel® Xeon® 6900/6700-series processors with E-cores and Intel® Xeon® 6900/6700/6500-series processors with P-cores on the Birch Stream platform—Intel has a new Performance Per Watt (PPW) biased power mode called "default" along with a lower latency mode called "latency optimized mode." This paper aims to highlight the difference between the two and the PPW benefits of the new "default" mode.

Default

Out-of-the-box mode for newer disaggregated SoC architecture with significant power savings across the load line. Better PPW across the load line. There is no significant difference in performance compared to Latency Optimized mode at 100% load level.

Latency Optimized Mode

The highest performance mode (the default on previous generations) involves core and uncore frequencies running up to their maximum limits within the RAPL budget. This model is not PPW optimized across the load line.

Details of the architecture for these modes are described, and other example profiles and possibilities are also discussed.

2 Background

Traditionally performance-per-power settings have been set through Energy Performance Bias (EPB) and Energy Performance Preference (EPP) configurations, either during the BIOS or by the Operating System (OS) performance governor modes.

These settings allow customers to choose between various performance-per-power profiles ranging from highest performance to highest power savings.

Previous 4th and 5th Gen Intel® Xeon® processor Scalable families introduced an Optimized Power Mode (OPM) specifically tuned for power savings at lower utilization portions of the load line with minimal performance impact (5% or less).

Table 2-1. Workload Performance OOB versus Optimized Power Mode 2.0 for 5th Gen Intel® Xeon® Processor Scalable Families

Setting	Active Idle	Uncore Maximum Frequency	SPECpower* (Savings in W)		SPECjbb (perf)/Java*			SPECcpu SIR (perf)			NGINX*	FIO (Small Packet Latency)
			At 0% Load	At 30% Load	Maximum JOps	Critical JOps	Average Power Saved	Geomean	Load Line at 16C	Average Power Saved	Throughput	1/Latency
Out-Of-Box (OOB) – 5th Gen Intel® Xeon® processor Scalable family	UP = 0, Uncore = 1.4 GHz	2.5 GHz	0W	0W	100%	100%	0W	100%	100%	0W	100%	100%
Optimized Power Mode 2.0	UP = 3, Uncore = 1.4 GHz	2.2 GHz	4W	108W	100%	100%	16W	100%	97%	109W	100%	100%

NOTES: Supplier contact information is provided for convenience. Intel does not formally endorse any particular supplier.

Intel® Xeon® 6 processors extend the OPM methodology to tune for both performance at 100% utilization (example: EST. SPECCPU2017 INTEGER RATE) and efficiency across the load line (example: Energy Efficiency Workload). Intel® Xeon® 6 processors also add new control knobs to adjust the performance-per-power profile to customize it for different usages.

A major performance-per-power tradeoff is the uncore frequency. Uncore in this this technical article refers to the coherent fabrics and L3 caches in the compute chiplets. Setting the uncore frequency to the maximum results in large power consumption even at lower utilization points (for example, 30%-50% utilized). On Intel® Xeon® 6 SoCs, there are separate fabrics for each chiplet (compute and I/O). These fabrics can be controlled separately. The new knobs give better control over the uncore frequency policies.

The ability to control the compute and I/O part of the uncore independently enables power savings when the I/O frequency can be set lower when the I/O die is not connected to I/O devices, for example, core-centric workloads.

Typical ranges for uncore on the compute chiplet are 0.8-2.2 GHz, and on the I/O chiplet are 0.8-2.5 GHz. In idle conditions, 1.2 GHz (C) and 0.8GHz (I/O) are defaults, but for a core-centric workload like an EST. SPECCPU2017 INTEGER RATE exchange, these can be lowered to 0.8 GHz/0.8 GHz (dynamically, based on SoC heuristics).

Allowing the uncore to scale based on utilization and heuristics results in better power efficiency (Performance Per Watt [PPW]) across utilization points while still maintaining performance at high utilization points. One consequence of a scaled uncore frequency is that idle latency micros will degrade, and there may be some performance impact (<5%) on some latency-sensitive workloads. Some improvements have been made to mitigate the impact.

This paper describes the new Intel® Xeon® 6 processor policies and control knobs for optimizing performance and power efficiency and recommended Intel profiles for different usages.

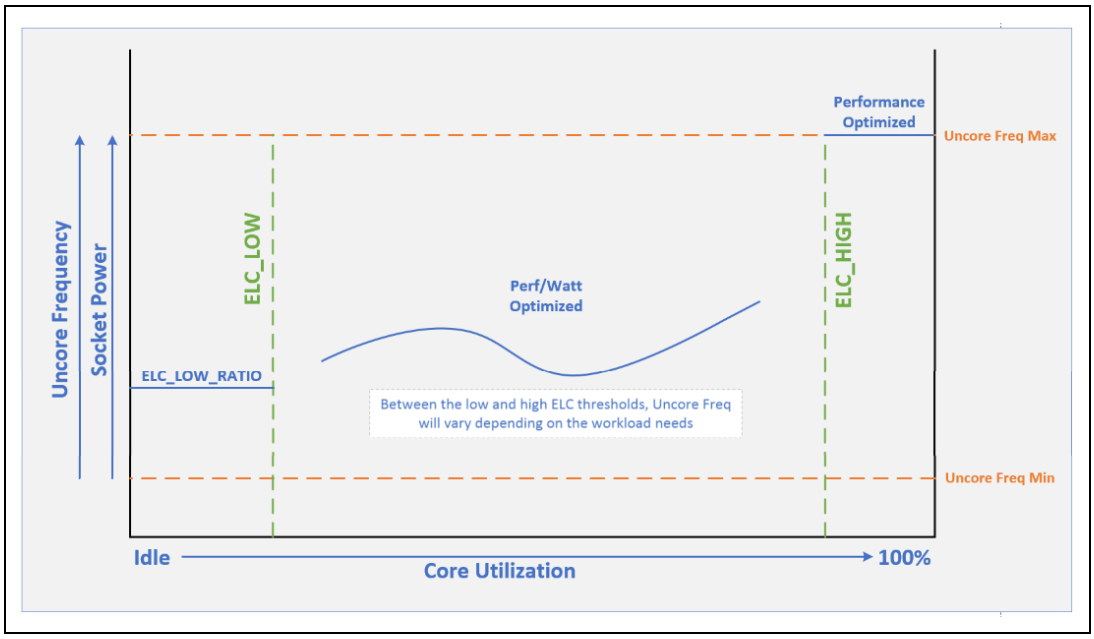
Note: The EPP policy is orthogonal to the control knobs described in this technical article. (EPP is consistent with past products, with values from 0-255 that tune the Hardware P-States algorithm (HWP); 0 is the maximum performance (turbo all the time), and 255 is minimum energy).

In addition to the knobs described, other power management features that help to optimize power in Intel® Xeon® 6 processors include various power states (C1, FC1E (P-core processors only), C6, PC6), and Intel® Ultra Path Interconnect (Intel® UPI) L0p.

3 Uncore Frequency Scaling Policies for Intel® Xeon® 6 Processors

Intel® Xeon® 6 processors offer more optimization opportunities than the 5th Gen Intel® Xeon® processor Scalable families for performance (latency) and PPW depending on the CPU load. Intel® Xeon® 6 processors include new knobs that can help users select up to three uncore frequency scaling policies depending on the CPU C0 utilization.

Figure 3-1. Load Line Optimization Options in Birch Stream



The previous figure illustrates the optimization options for a load line in Intel® Xeon® 6 processors. The X-axis is the CPU C0 utilization, and the Y-axis is the socket power consumption. There are two CPU C0 utilization thresholds: ELC_LOW and ELC_HIGH. Efficiency Latency Control (ELC) refers to the set of knobs that allows users to trade-off between performance (latency) and power consumption.

These two thresholds divide the entire load line into three parts, with different optimization objectives, as follows:

if(Utilization ≤ ELC_Low)
Set a floor frequency (ELC_RATIO) to offer high responsiveness when idle.
else-if(Utilization > ELC _High)
Consume full TDP to push uncore frequency higher (up to the MAX_RATIO).
else-if(ELC_Low < Utilization ≤ ELC_High)
Heuristics-based fabric frequency.

Typically, the ELC_LOW_RATIO is chosen such that in idle condition (0% load) with core C6 state disabled the socket consumes 40% of TDP. Users can choose to increase or decrease the ELC_LOW_RATIO for Compute and IO die depending on the idle latency and idle power requirements for their workload. The ELC_LOW_ THRESHOLD of 10% provides ample margin for latency tests following low activity workloads. The ELC _HIGH_THRESHOLD is chosen high enough to ensure the increase in uncore frequency if there is TDP headroom even at 100% C0 utilization.

In between the high and low thresholds, the native Uncore Frequency Scaling (UFS) algorithm optimizes the socket for Performance Per Watt (PPW), for example, default UFS allows socket to achieve much lower power consumption at 50% load when running SPECpower*, than when the socket is configured to be latency optimized mode.

Note on Latency Optimized Mode and Energy Performance Bias (EPB):

Latency optimized mode sets *ELC_Low=ELC_High=0%* and within TDP limit makes the uncore frequency minimum=maximum.

Any deviation of EPB from a value of 0, induced by the OS or any other agent will allow the uncore frequency to not be at the maximum, and thus will violate the desire to be at maximum uncore frequency, which is provided via Latency Optimized mode. If EPB changes occur on an underutilized socket in a 2S system, the drop in uncore frequency in the underutilized socket can result in longer than expected remote socket latencies.

To overcome any unexpected variation in latencies due to the OS changing the EPB values to something >0, customers can do one of these two things:

- Choose “BIOS Controls EPB” in the BIOS in place of “OS Controls EPB”.

Or

- Set *Uncore Freq Min=Max* to the specified maximum uncore frequency, thereby eliminating any latency impacts due to changes in EPB values made by OS in Latency Optimized mode.

3.1 Control Knobs

The knobs are accessed through the Topology Aware Register and Power Management Capsule Interface (TPMI), which allows individual control over each die (compute and I/O).

Details on accessing TPMI registers can be found here:

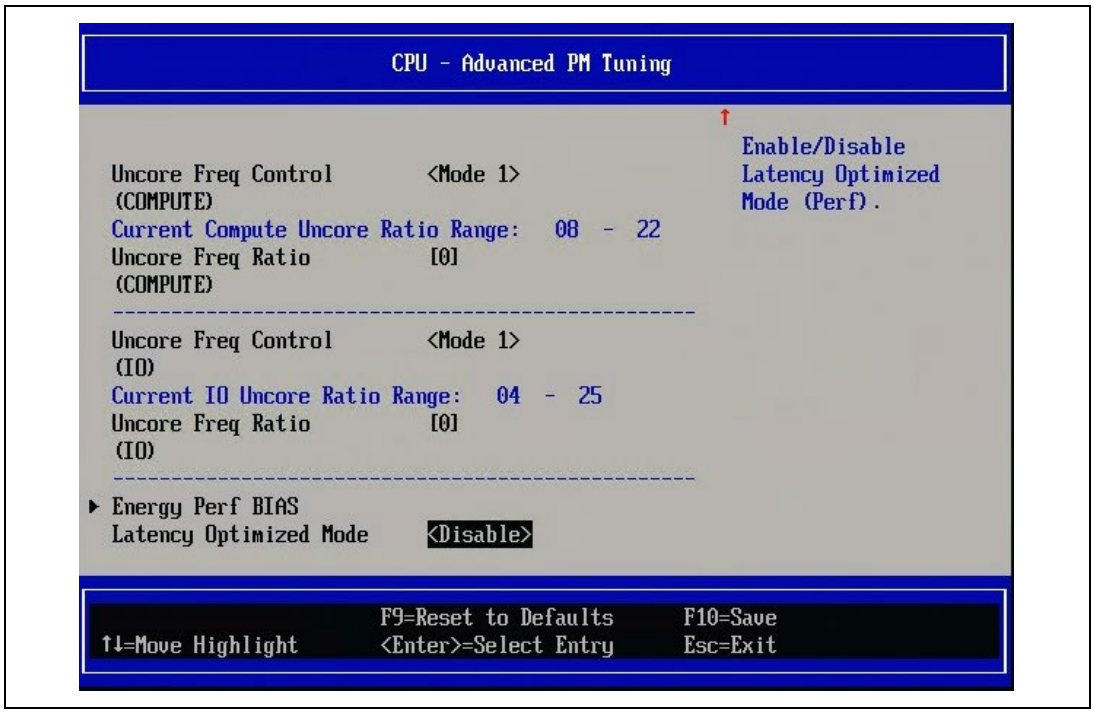
- TPMI Registers: UFS Control Register (UFS_CONTROL): UFS_TPMI_public_disclosure.
- For advanced usages, a register can be locked by a TPMI lock bit. For details, refer to the TPMI documentation IC_TPMI_SET_STATE or contact an Intel PAE.

3.2 BIOS Knob for Latency Optimized Mode

On Birch Stream platforms, Intel added a Latency Optimized Mode BIOS knob. It will be visible on BIOS ID 3544.P05 and later.

- Intel's BIOS knob, Latency Optimized mode, is found under CPU – Advanced PM Tuning
- This knob is "disabled" by default.
- Enable this, if better latency is desired. It maintains Uncore Freq high all the time unless limited by RAPL/TDP or other constraints.

Figure 3-2. BIOS knob for Latency Optimized Mode (Intel Reference Validation Platform [RVP] BIOS Shown)



4 Recommended Profiles

The two primary profiles that are evaluated are the Default mode and the Latency Optimized mode.

Note: Settings are subject to change once final tuning is done.

Table 4-1. TPMI ELC Settings

TPMI Register Settings	Scope	Default	Latency optimized (Traditional Perf mode)
EFFICIENCY_LATENCY_CTRL_RATIO	Die (Compute/IO)	12, 8 (1.2 GHz (c)/0.8 GHz (io))	0
EFFICIENCY_LATENCY_CTRL_LOW_THRESHOLD	Socket	13 (13/127 = 10%)	0
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD	Socket	120 (120/127 = 95%)	0%
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD_ENABLE	Socket	1	1
MAX_RATIO	Die (Compute/IO)	22, 25 (2.2 GHz (c)/2.5 GHz (io))[1]	22,25 (2.2 GHz (c)/2.5 GHz (io))
MIN_RATIO	Die (Compute/IO)	8, 8 (0.8 GHz (c)/0.8 GHz (io))	8, 8 (0.8 GHz (c)/0.8 GHz (io))
EPB	SYS	0	0
EPP	SYS	0	0
OS mode name	SYS	Perf	Perf
Description of Mesh/Fabric Frequency Policy	**-**	**<10% Mesh = 1.2 GHz/0.8 GHz, 10-95% Mesh = Uncore Frequency Scaling (UFS) >95% Mesh = maximize mesh freq under TDP constraint UFS tuning will vary depending on SKU**	**Mesh freq = maximize mesh freq under TDP constraint**

NOTES: 1. Maximum frequency is SKU-dependent. Some SKUs may have lower MAX_RATIO values.

Influence of EPB

For the Efficiency Latency Control (ELC) mechanism to work effectively, the EPB needs to be 0. Uncore frequency only increases at the ELC high threshold when EPB=0. EPB > 0 will disable ELC_High_Threshold. The ELC Low Threshold can be set and used in any EPB mode (EPB>=0).

Influence of C-States

C-states do not influence UFS behavior differently on the two power profiles, but enabling CC6 or PC6 will change the core and uncore frequencies at idle depending on the idle state (Active Idle/Perf Idle/OS Idle).

4.1 Recipes for Linux* and Windows*

Registers are accessed through the TPMI interface. In both Linux* and Windows*, UFS TPMI can be accessed through the PCM tool.

On Linux you can use "pepc" on kernel versions 6.6 and later.

- PCM can be accessed via this external link: https://github.com/intel/pcm
- PEPC can be accessed via this link: https://github.com/intel/pepc

Table 4-2. Die ID Decoder

-	Intel® Xeon® 6900-Series Processor with P-Cores	Intel® Xeon® 6700/6500-Series Processor with P-Cores	Intel® Xeon® 6900-Series Processors with E-Cores	Intel® Xeon® 6700-Series Processors with E-Cores
Config	ICCCI	ICCI	ICCI	ICI
Uncore Compute	0,1,2	0,1	0,1	0
Uncore IO	3,4	3,4	3,4	3,4
Uncore IO (PCM tool only) -e	3,4	2,3	2,3	1,2
Instances (PCM tool only) -i	examples: -i 1 or -i 0,1 or -i 0,2-3 1S=0 2S=0,1 4S=0,1,2,3 ***Usage example:*** pcm-tpmi 2 0x18 -d -i 1 -e 3 -b 28:22 -w 15 The above command changes the Uncore IO freq on Die 3 on the second socket to 1.5 GHz. -e option can be used to change ELC low ratio on the IO dies. ***Changing frequencies per die on compute dies is not recommended.***			

pcm-tpmi example commands to set Latency Optimized mode:

```
./pcm-tpmi.exe 2 0x18 -d -b 39:39 -w 1 # enable ELC high bit
./pcm-tpmi.exe 2 0x18 -d -b 28:22 -w 0 # ELC low ratio zero
./pcm-tpmi.exe 2 0x18 -d -b 46:40 -w 0 # ELC high threshold zero
./pcm-tpmi.exe 2 0x18 -d -b 38:32 -w 0 # ELC low threshold zero
```

Or

Use bhs-power-mode.sh on Linux:

Script

Usage: `bhs-power-mode.sh --latency-optimized-mode`

pcm-tpmi example commands to set "default" mode:

Use `bhs-power-mode.sh` on Linux:

Script

Usage: `bhs-power-mode.sh –default`

Table 4-3. PCM Tool Instructions (Windows* and Linux*)

TPMI UFS CSR	Pcm-tpmi Equivalent (Windows*/Linux*)	Comments
EFFICIENCY_LATENCY_CTRL_RATIO (Uncore IO)	pcm-tpmi 2 0x18 -d -e<ios> -b 28:22 -w 8 -i 0,1	Sets uncore IO die frequency to 0.8 GHz. Use the previous table to identify –e die IDs. -i 0,1 sets it on both sockets.
EFFICIENCY_LATENCY_CTRL_RATIO (Uncore Compute)	pcm-tpmi 2 0x18 -d -e <computes> -b 28:22 -w 12 -i 0,1	Sets uncore compute die frequency to 1.2 GHz. Use the previous table to identify –e die IDs **Changing frequencies per die on computes dies is not recommended.** -i 0,1 sets it on both sockets.
EFFICIENCY_LATENCY_CTRL_LOW_THRESHOLD (Uncore Root IO)	pcm-tpmi 2 0x18 -d -e <io0> -b 38:32 -w 13 -i 0,1	Sets ELC low threshold to 10% (13/127). Use the previous table to identify –e die IDs. Need to be set on the Uncore Root IO but will apply across all IO and Compute dies. -i 0,1 sets it on both sockets.
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD(Uncore Root IO)	pcm-tpmi 2 0x18 -d -e<io0> -b 46:40 -w 120 -i 0,1	Sets ELC high threshold to 95% (120/127). Use the previous table to identify –e die IDs. Need to be set on the Uncore Root IO but will apply across all IO and Compute dies. -i 0,1 sets it on both sockets.
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD_ENABLE (Uncore Root IO)	pcm-tpmi 2 0x18 -d -e<io0> -b 39:39 -w 1 -i 0,1	Enables ELC high threshold. Use the previous table to identify –e die IDs. Need to be set on the Uncore Root IO but will apply across all IO and Compute dies. -i 0,1 sets it on both sockets.
MIN_RATIO (Uncore Compute)	pcm-tpmi 2 0x18 –d –e <computes> -b 15:21 -w 15 -i 0,1	Example: Sets uncore compute minimum frequency to 1.5 GHz. -i 0,1 sets it on both sockets. **Changing frequencies per die on computes dies is not recommended.**

TPMI UFS CSR	Pcm-tpmi Equivalent (Windows*/Linux*)	Comments
MAX_RATIO (Uncore Compute)	pcm-tpmi 2 0x18 –d –e <computes> -b 14:8 -w 22 -i 0,1	Example: Sets uncore compute maximum frequency to 2.2 GHz. -i 0,1 sets it on both sockets. **Changing frequencies per die on computes dies is not recommended.**
MIN_RATIO (Uncore IO)	pcm-tpmi 2 0x18 –d –e <ios> -b 15:21 -w 15 -i 0,1	Example: Sets uncore IO minimum frequency to 1.5 GHz. -i 0,1 sets it on both sockets.
MAX_RATIO (Uncore IO)	pcm-tpmi 2 0x18 –d –e <ios> -b 14:8 -w 22 -i 0,1	Example: Sets uncore IO maximum frequency to 2.2 GHz. -i 0,1 sets it on both sockets.

PEPC Instructions:

Install PEPC and run the following:

1. To set power mode to "Latency Optimized mode":
```
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_RAT IO –V 0 -i 3,4
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_RAT IO –V 0 -i 0,1,2
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_LO W_THRESHOLD –V 0 –i 3
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_HIG H_THRESHOLD –V 0 –i 3
```
2. To set power mode back to "default mode":
```
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_RAT IO –V 8 -i 3,4
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_RAT IO –V 12 -i 0,1,2
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_LO W_THRESHOLD –V 13 –i 3
pepc tpmi write -F uncore -R UFS_CONTROL -b
EFFICIENCY_LATENCY_CTRL_HIG H_THRESHOLD –V 120 –i 3
```
3. To get the current state of all the registers:
```
pepc tpmi read -F uncore -R UFS_CONTROL
```

Table 4-4. PEPC Tool Instructions (Linux)

TPMI UFS CSR	Pepc equivalent (Linux)	Comments
EFFICIENCY_LATENCY_CTRL_RATIO (Uncore IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b EFFICIENCY_LATENCY_CTRL_RATIO –V 8 -i 3,4	Sets Uncore IO die Freq to 0.8 GHz. Use the die id decoder table to identify –i die IDs.
EFFICIENCY_LATENCY_CTRL_RATIO (Uncore Compute)	pepc tpmi write -F uncore -R UFS_CONTROL -b EFFICIENCY_LATENCY_CTRL_RATIO –V 12 -i 0,1,2	Sets Uncore Compute die Freq to 1.2 GHz . Use the die id decoder table to identify –i die IDs.

TPMI UFS CSR	Pepc equivalent (Linux)	Comments
EFFICIENCY_LATENCY_CTRL_LOW _THRESHOLD (Uncore Root IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b EFFICIENCY_LATENCY_CTRL_LOW_ THRESHOLD –V 13 –i 3	Sets ELC low threshold to 10% (13/127). Use the die id decoder table to identify –i die IDs.
EFFICIENCY_LATENCY_CTRL_HIGH _THRESHOLD (Uncore Root IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b EFFICIENCY_LATENCY_CTRL_HIGH_ THRESHOLD –V 120 –i 3	Sets ELC high threshold to 95% (120/127). Use the die id decoder table to identify –i die IDs.
EFFICIENCY_LATENCY_CTRL_HIGH _THRESHOLD_ENABLE (Uncore Root IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b EFFICIENCY_LATENCY_CTRL_HIGH_ THRESHOLD_ENABLE –V 1 –i 3	Enables ELC high threshold. Use the die id decoder table to identify –i die IDs.
MIN_RATIO (Uncore Compute)	pepc tpmi write -F uncore -R UFS_CONTROL -b MIN_RATIO –V 15 –i 0,1	Example: Sets Uncore Compute Min Freq to 1.5 GHz (Computes die values 0,1 or 0&1) Use the die id decoder table to identify –i die IDs. **Changing frequencies per die on computes dies is not recommended.**
MAX_RATIO (Uncore Compute)	pepc tpmi write -F uncore -R UFS_CONTROL -b MAX_RATIO –V 22 –i 0,1	Example: Sets Uncore Compute Max Freq to 2.2 GHz (Computes die values 0,1 or 0&1) Use the die id decoder table to identify –i die IDs. **Changing frequencies per die on computes dies is not recommended.**
MIN_RATIO (Uncore IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b MIN_RATIO –V 15 –i 3,4	Example: Sets Uncore IO Min Freq to 1.5 GHz (IO die values 3,4, or 3&4) Use the die id decoder table to identify –i die IDs.
MAX_RATIO (Uncore IO)	pepc tpmi write -F uncore -R UFS_CONTROL -b MAX_RATIO –V 22 –i 3,4	Example: Sets Uncore IO Max Freq to 2.2 GHz (IO die values 3,4, or 3&4) Use the die id decoder table to identify –i die IDs.

4.2 Other Example Profiles

Here are some additional profiles that could be tried, depending on the usage and requirements.

An example of setting EFFICIENCY_LATENCY_CTRL_RATIO to a value of 50% would be to set it to 64 (64/127=50%).

Table 4-5. Example Profiles

TPMI Register Settings	Efficiency	Latency Friendly	Lowest Idle power	Networking (5G UPF)
EFFICIENCY_LATENCY_CTRL_RATIO	Same as default.	**18,16 (1.8 GHz (c)/1.6 GHz (io))** Can improve remote/IO idle latencies with still reasonable idle power.	**8,8 (0.8 GHz (c)/0.8 GHz (io))** Same as default settings, but with lower idle power (and longer remote and I/O idle latencies)	Low and high thresholds same as default. Frequency fixed using (Max_Ratio/Min_Ratio set to 1.5 GHz)
EFFICIENCY_LATENCY_CTRL_LOW_THRESHOLD	10% (same as default)	10% (same as default)	10% (same as default)	Low and high thresholds same as default. Frequency fixed using (Max_Ratio/Min_Ratio set to 1.5 GHz)
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD	95% (same as default)	95% (same as default)	95% (same as default)	Low and high thresholds same as default. Frequency fixed using (Max_Ratio/Min_Ratio set to 1.5 GHz)
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD_ENABLE	1 (same as default)	1 (same as default)	1 (same as default)	Low and high thresholds same as default. Frequency fixed using (Max_Ratio/Min_Ratio set to 1.5 GHz)
MAX_RATIO	Same as default	Same as default	Same as default	**15,15 (1.5 GHz (c)/1.5 GHz (io))** Requires medium range uncore frequency to achieve peak throughput and PPW
MIN_RATIO	Same as default	Same as default	Same as default	**15,15 (1.5 GHz (c)/1.5 GHz (io))**
EPB	6	0	0	0
EPP	128	0	0	0
OS mode name	Balanced Perf (Windows*)	Perf	Perf	Perf
Mesh/fabric frequency policy description	TBD	• <10% Mesh = low ratio • 10-95% Mesh = Uncore Frequency Scaling (UFS) • >95% Mesh = max	• <10% Mesh = low ratio • 10-95% Mesh = Uncore Frequency Scaling (UFS) • >95% Mesh = max	Fixed Freq = 1.5 GHz

Table 4-6. Example Profiles Tailored to Workload Characteristics

TPMI Register Settings	Core Bound	Memory Bound	I/O Bound
EFFICIENCY_LATENCY_CTRL_RATIO	8,8 (0.8 GHz (c)/0.8 GHz (io))	22,8 (2.2 GHz (c)/0.8 GHz (io))	22,25 (2.2 GHz (c)/2.5 GHz (io))
EFFICIENCY_LATENCY_CTRL_LOW_THRESHOLD	100%	100%	100%
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD	100%	100%	100%
EFFICIENCY_LATENCY_CTRL_HIGH_THRESHOLD_ENABLE	0	0	0
MAX_RATIO	Default	Default	Default
MIN_RATIO	Default	Default	Default
EPB	0	0	0
EPP	0	0	0
OS mode name	Perf	Perf	Perf
Mesh/fabric frequency policy description	<100% Mesh = 0.8 GHz	<100%: compute mesh = 2.2 GHz, IO mesh = 0.8 GHz	<100%: compute mesh = 2.2 GHz, IO mesh = 2.5 GHz

4.3 Intel® Xeon® 6900-Series Processor with P-Cores Energy Efficiency Workload – Improved TCO with Customized Latency Optimized Mode

For TCO calculations when using "Energy Efficiency Workload" power data, making some changes to the Latency Optimized mode can result in ~7-10% power savings at 50% load level.

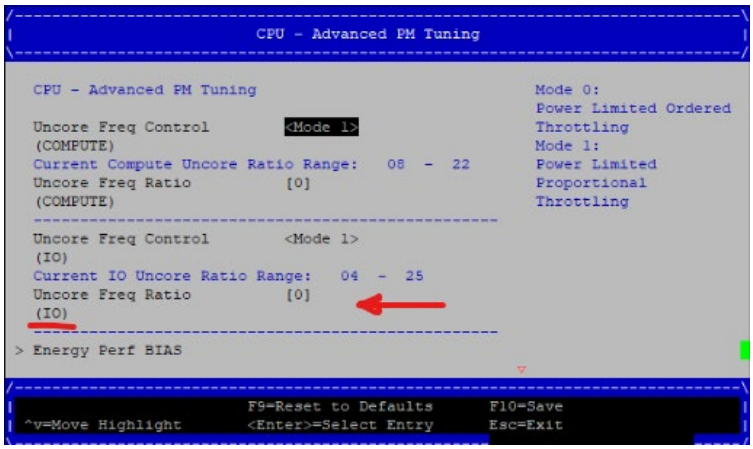
The uncore frequencies can be set for the compute and IO dies, and in latency-optimized mode, both are set to min=max unless constrained by RAPL/TDP.

The Energy Efficiency Workload does not stress the Uncore IO, and hence, the uncore IO frequency can be set lower to realize TCO savings in the lower utilization levels.

This can be done in two ways:

1. Using TPMI to set the uncore_io_max_ratio to a lower value like 08 from the latency optimized mode default of 24 (or 25) when in Latency Optimized mode.
2. Setting the Uncore IO frequency ratio in the BIOS to 08.

Figure 4-1. Uncore IO Frequency Ratio



Expected results: About 7-10% power savings at 50% load with lower uncore IO freq compared to Latency Optimized mode and comparable to "default" mode.

5 Workload Tradeoffs Between Default and Latency Optimized Profiles

This section describes the tradeoffs between Default and Latency Optimized profiles.

This high-level comparison briefly outlines the pros and cons of each profile. The following table includes more details on measurements.

Table 5-1. Workload Power, PPW Expectations

Expectation	Default	Latency Optimized (Traditional Perf)
EST. SPECCPU2017 INTEGER RATE/EST. SPECCPU2017 FLOATING POINT RATE (100% utilized)	Same	Same
Server-Side Java* Workload	Same	Same
Energy Efficiency Workload[1]	Better	Worse
EST. SPECCPU2017 INTEGER RATE load line[2]	Better	Worse
Network latencies (Sockperf*)	Worse	Better
Micros		
Latency when TDP constrained	Same	Same
Local memory/LLC latencies that are helped by "stall detect"	Same	Same
All other idle latencies	Worse	Better
Perf Idle	Better	Worse
Active Idle	Better	Worse

NOTES: 1. As measured by PPW at 50% utilized.
2. As measured by PPW at 50% utilized (performance-per-active VCPU will be listed, also to ensure base performance requirement is met).
Legend: Green indicates "better" performance and red indicates "worse" performance.

6 Intel® Xeon® 6900-Series Processors with P-Cores Workload Comparison – Measurements

Note: See the backup section for workloads and configurations. Results may vary.

Table 6-1. Intel® Xeon® 6900-Series Processors with P-Cores Workload Performance

Workload	Latency Optimized/Default
EST. SPECCPU2017 INTEGER RATE	1.00
Server-side Java* workload	1.00
Energy Efficiency[1] Workload C6 Disabled	0.87
EST. SPECCPU2017 INTEGER RATE Load line[2] C6 Disabled	50% PPW 0.89
EST. SPECCPU2017 INTEGER RATE Load line[2] C6 Disabled	50% Perf/vCPU 1.04
OLAP (TPC-H type (Throughput Runtime(sec))* 10 TB SF (Lower is better)	0.96
L2 core to core(ns)	1.00
LLC latency – local socket L3 Hit (ns) (Uncore IO Freq Impact)	1.00
Local memory latency (ns)	0.99
Remote socket memory (ns)latency	0.65
Network wake latency (µs)	0.79
Remote socket remotely homed L2 hit latency (ns)	0.61
Remote socket remotely homed L3 hit latency (ns)	0.59
Perf Idle (FC1E enabled) (W) 1S	1.53
Active Idle (C6 enabled) (W) 1S	1.61

NOTES:

1. Higher is better, except for latency and power numbers, where lower is better.
2. Energy efficiency workload as measured by PPW at 50% utilization. See the following measured load lines.
3. EST. SPECCPU2017 INTEGER RATE load line as measured by PPW at 50% utilized.
4. Legend: Green indicates “better" performance and red indicates "worse" performance.

7 Intel® Xeon® 6700-Series Processors with E- and P-Cores Workload Comparison – Measurements

Note: Energy Efficiency Workload as measured by PPW at 50% utilization. See the following measured load lines.

Table 7-1. Intel® Xeon® 6700-Series Processors E- and P-Cores Workload Performance

Workload	Latency Optimized/Default
EST. SPECCPU2017 INTEGER RATE	1.00
Server-Side Java workload	1.00
Energy Efficiency workload[1]	0.90
L2 core to core (ns)	1.00
LLC latency – local socket (ns)	1.00
Local memory latency (ns)	0.99
Remote socket memory (ns) latency	0.54
Network wake latency (µs)	0.86
Perf Idle (W)	1.18
Active Idle (C6 enabled) (W)	1.15

NOTES:

1. Energy Efficiency Workload as measured by PPW at 50% utilization.
2. Higher is better, except for latency and power numbers, where lower is better.
3. Legend: Green indicates “better" performance and red indicates "worse" performance.

8 Summary

With Intel® Xeon® 6 processors, Intel introduced an out-of-box mode that supports the newer disaggregated SoC architecture with significant power savings across the load line. Intel has the new Efficiency Latency Control (ELC) mechanism with multi-policy capable architecture and user-controllable ranges for ELC. TPMI provides a direct architectural power management feature tuning from the OS (Linux/Windows) to control uncore frequency scaling parameters for optimal perf or PPW workload performance.

9 Notices and Disclaimers

Note: Supplier contact information is provided for convenience. Intel does not formally endorse any particular supplier.
GNR, Granite Rapids: Intel® Xeon® 6900/6700/6500-series processors with P-cores
Granite Rapids-AP: Intel® Xeon® 6900-series processors with P-cores
Sierra Forest-SP: Intel® Xeon® 6700-series processors with E-cores
Intel® Xeon® Platinum processor
HT: Intel® Hyper-Threading Technology (Intel® HT Technology)
Turbo: Intel® Turbo Boost Technology
DLB: Intel® Dynamic Load Balancer (Intel® DLB)
DSA: Intel® Driver & Support Assistant (Intel® DSA)
IAA: Intel® In-Memory Analytics Accelerator (Intel® IAA)
QAT: Intel® QuickAssist Technology (Intel® QAT)
Samsung*
Micron*
Ubuntu*
CentOS* Stream 9
Java*
KIOXIA*

1. Integer Throughput (SIR), Floating-Point Throughput (SFR) (Intel Xeon 6900-series processors with P-cores): Estimated performance on two-socket pre-production platform with Intel Xeon Platinum processor (Intel Xeon 6900/6700/6500-series processors with P-cores) 96C, 550W TDP; DDR5-6400, 1-node, 2x Genuine Intel Xeon, 96 cores, HT On, Turbo On, NUMA 6, Integrated Accelerators Available [used]: DLB 8 [0], DSA 8 [0], IAA 8 [0], QAT 8 [0], Total Memory 1536 GB (24x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.86B.2899.D04.2401220310, microcode 0x80000594, 1x I210 Gigabit Network Connection, 2x Ethernet Controller E810-XXV for SFP, 1x 894.3G Micron_7450_MTFDKCC960TFR, 1x 894.3G Micron_7450_MTFDKBG960TFR, 1x 57.8G USB DISK 3.0, 4x 1.5T INTEL SSDPE2KE016T7, Ubuntu 23.10, 6.5.0-15-generic Power and Perf Policy: Performance. Test by Intel as of 06/05/24.
2. Server Side Java (Intel Xeon 6900/6700/6500-series processors with P-cores): Estimated performance on two-socket pre-production platform with Intel Xeon Platinum processor (Intel Xeon 6900/6700/6500-series processors with P-cores) 96C, 550W TDP; DDR5-6400, 1-node, 2x Genuine Intel Xeon, 96 cores, HT On, Turbo On, NUMA 6, Integrated Accelerators Available [used]: DLB 8 [0], DSA 8 [0], IAA 8 [0], QAT 8 [0], Total Memory 1536 GB (24x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.0032.D97.2405300022, microcode 0x81000251, 1x I210 Gigabit Network Connection, 4x Ethernet Controller E810-XXV for SFP, 1x 1G SAMSUNG MZQL27T6HBLA-00A07, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 1x 1.7T Micron_7450_MTFDKBG1T9TFR, CentOS Stream 9, 6.6.0-Gnr.bkc.6.6.17.9.24.x86_64. Power and Perf Policy: Performance Test by Intel as of 06/13/24.

3. Energy Efficiency workload (Intel Xeon 6900/6700/6500-series processors with P-cores): Estimated performance on two-socket pre-production platform with Intel Xeon Platinum processor (Intel Xeon 6900/6700/6500-series processors with P-cores) 96C, 550W TDP; DDR5-6400, 1-node, 2x Genuine Intel Xeon, 96 cores, HT On, Turbo On, NUMA 6, Integrated Accelerators Available [used]: DLB 8 [0], DSA 8 [0], IAA 8 [0], QAT 8 [0], Total Memory 1536 GB (24x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.0032.D97.2405300022, microcode 0x81000251, 1x I210 Gigabit Network Connection, 4x Ethernet Controller E810-XXV for SFP, 1x 1G SAMSUNG MZQL27T6HBLA-00A07, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 1x 1.7T Micron_7450_MTFDKBG1T9TFR, CentOS Stream 9, 6.6.0-Gnr.bkc.6.6.17.9.24.x86_64. Power and Perf Policy: Performance Test by Intel as of 06/05/24.
4. MLC Latency (Intel Xeon 6900/6700/6500-series processors with P-cores): Estimated performance on two-socket pre-production platform with Intel Xeon Platinum processor (Intel Xeon 6900/6700/6500-series processors with P-cores) 96C, 550W TDP; DDR5-6400, 1-node, 2x Genuine Intel Xeon, 96 cores, HT On, Turbo On, NUMA 6, Integrated Accelerators Available [used]: DLB 8 [0], DSA 8 [0], IAA 8 [0], QAT 8 [0], Total Memory 1536 GB (24x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS superpatch21 BHSDCRB1.IPC.0032.D97.2405300022 / 0xf15803fc, 1x I210 Gigabit Network Connection, 2x Ethernet Controller E810-XXV for SFP, 1x 894.3G Micron_7450_MTFDKCC960TFR, 1x 894.3G Micron_7450_MTFDKBG960TFR, 1x 57.8G USB DISK 3.0, 4x 1.5T INTEL SSDPE2KE016T7, Ubuntu 23.10, 6.5.0-15-generic, Power and Perf Policy: Performance. Test by Intel as of 06/06/24. https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html, Version: v3.11a.
5. Sockperf Latency test: Estimated performance on two-socket pre-production platform with Intel Xeon Platinum processor (Intel Xeon 6900/6700/6500-series processors with P-cores) 96C, 550W TDP; DDR5-6400, 1-node, 2x Genuine Intel Xeon, 96 cores, HT On, Turbo On, NUMA 6, Integrated Accelerators Available [used]: DLB 8 [0], DSA 8 [0], IAA 8 [0], QAT 8 [0], Total Memory 1536 GB (24x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.E9I.0031.D44.2405031510, microcode 0xf15802b8, 1x I210 Gigabit Network Connection, 4x Ethernet Controller E810-C for QSFP, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 1x 894.3G Micron_7450_MTFDKBG960TFR, CentOS Stream 9, 6.6.0-gnr.bkc.6.6.17.9.24.x86_64. Power and Perf Policy: Performance Test by Intel as of 06/06/24.
6. Integer Throughput (SIR), Floating-Point Throughput (SFR) (Intel Xeon 6700-series processors with E-cores): Estimated performance on two-socket production platform with Intel Xeon 6780E processor 144C, 330W TDP; DDR5-6400, 1-node, 2x Intel Xeon 6780E processors, 144 cores, HT N/A, Turbo On, NUMA 2, Integrated Accelerators Available [used]: DLB 4 [0], DSA 4 [0], IAA 4 [0], QAT 4 [0], Total Memory 1024 GB (16x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.3085.P38.2405202106, microcode 0x30001b2, 4x Ethernet Controller E810-C for QSFP, 1x I210 Gigabit Network Connection, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 1x 894.3G Micron_7450_MTFDKBG960TFR, CentOS Stream 9, 6.6.0-

srf.bkc.6.6.13.4.11.x86_64. Power and Perf Policy: Performance Test by Intel as of 06/05/24.

7. Server Side Java (Intel Xeon 6700-series processors with E-cores): Estimated performance on two-socket production platform with Intel Xeon 6780E processor 144C, 330W TDP; DDR5-6400, 1-node, 2x Intel Xeon 6780E processor, 144 cores, HT Off, Turbo On, NUMA 2, Integrated Accelerators Available [used]: DLB 4 [0], DSA 4 [0], IAA 4 [0], QAT 4 [0], Total Memory 1024 GB (16x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.3085.P38.2405202106, microcode 0x30001b2, 1x I210 Gigabit Network Connection, 4x Ethernet Controller E810-C for QSFP, 3x 1.7T SAMSUNG MZWLO1T9HCJR-00A07, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 2x 1.7T KIOXIA KCDY1RUG1T92, 1x 894.3G Micron_7450_MTFDKBG960TFR, Ubuntu 22.04.4 LTS, 6.5.0-35-generic, Power and Perf Policy: Performance Test by Intel as of 06/13/24.
8. Energy Efficiency workload (Intel Xeon 6700-series processors with E-cores): Estimated performance on two-socket production platform with Intel Xeon 6780E processor 144C, 330W TDP; DDR5-6400, 1-node, 2x Intel Xeon 6780E processors, 144 cores, HT N/A, Turbo On, NUMA 2, Integrated Accelerators Available [used]: DLB 4 [0], DSA 4 [0], IAA 4 [0], QAT 4 [0], Total Memory 1024 GB (16x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.3085.P38.2405202106, microcode 0x30001b2, 4x Ethernet Controller E810-C for QSFP, 1x I210 Gigabit Network Connection, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 1x 894.3G Micron_7450_MTFDKBG960TFR, CentOS Stream 9, 6.6.0-srf.bkc.6.6.13.4.11.x86_64. Power and Perf Policy: Performance Test by Intel as of 06/05/24.
9. MLC Latency (Intel Xeon 6700-series processors with E-cores): Estimated performance on two-socket production platform with Intel Xeon 6780E processor 144C, 330W TDP; DDR5-6400, 1-node, 2x Intel Xeon 6780E processors, 144 cores, HT Off, Turbo On, NUMA 2, Integrated Accelerators Available [used]: DLB 4 [0], DSA 4 [0], IAA 4 [0], QAT 4 [0], Total Memory 1024 GB (16x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.3085.P38.2405202106, microcode 0x130001b0, 1x I210 Gigabit Network Connection, 4x Ethernet Controller E810-C for QSFP, 3x 1.7T SAMSUNG MZWLO1T9HCJR-00A07, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, 2x 1.7T KIOXIA KCDY1RUG1T92, 1x 894.3G Micron_7450_MTFDKBG960TFR, Ubuntu22.04.4LTS, 6.5.0-28-generic Power and Perf Policy: Performance. Test by Intel as of 06/06/24. https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html, Version: v3.11a.
10. Sockperf Latency test: Estimated performance on two-socket pre-production platform with Intel Xeon 6780E processor 144C, 330W TDP; DDR5-6400, 1-node, 2x Intel Xeon 6780E processors, 144 cores, HT N/A, Turbo On, NUMA 2, Integrated Accelerators Available [used]: DLB 4 [0], DSA 4 [0], IAA 4 [0], QAT 4 [0], Total Memory 1024 GB (16x 64 GB DDR5 6400 MT/s [6400 MT/s]), BIOS BHSDCRB1.IPC.3085.P38.2405202106, microcode 0x30001b2, 4x Ethernet Controller E810-C for QSFP, 1x I210 Gigabit Network Connection, 1x 894.3G Micron_7450_MTFDKBG960TFR, 1x 7T SAMSUNG MZQL27T6HBLA-00A07, CentOS Stream 9, 6.6.0-srf.bkc.6.6.13.4.11.x86_64.Power and Perf Policy: Performance Test by Intel as of 06/07/24.