Tuning UEFI Settings for Performance and Energy Efficiency on AMD Processor-Based ThinkSystem Servers

Last Update: January 2023

Covers the ThinkSystem SR635, SR645, SR655 and SR665 servers with AMD EPYC 7002 and 7003 processors

Defines UEFI preset operating modes for Maximum Performance and Maximum Efficiency

Provides detailed information on key UEFI tuning parameter settings

Explains AMD NUMA domains and tuning options and the Determinism Slider tuning parameter

John Encizo

Kin Huang

Joe Jakubowski

Robert R. Wolford

Abstract

Properly configuring UEFI parameters in a server is important for achieving a desired outcome such as maximum performance or maximum energy efficiency. This paper defines preset UEFI operating modes for Lenovo® ThinkSystem™ servers running AMD EPYC 7002 and 7003 Series processors that are optimized for each of these outcomes. In addition, a thorough explanation is provided for each of the UEFI parameters so they can be fine-tuned for any particular customer deployment.

This paper is for customers and for business partners and sellers wishing to understand how to optimize UEFI parameters to achieve maximum performance or energy efficiency of Lenovo ThinkSystem servers with second-generation and third-generation AMD EPYC processors.

At Lenovo Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges. See a list of our most recent publications:

http://lenovopress.com

Contents

Introduction

The Lenovo ThinkSystem UEFI provides an interface to the server firmware that controls boot and runtime services. The system firmware contains numerous tuning parameters that can be set through the UEFI interface. These tuning parameters can affect all aspects of how the server functions and how well the server performs.

The Lenovo ThinkSystem SR635, SR645, SR655 and SR665 server UEFI contains operating modes that pre-define tuning parameters for maximum performance or maximum energy efficiency. This paper describes the tuning parameter settings for each operating mode and other tuning parameters to consider for performance and efficiency.

> **AMD 1S and 2S differences:** The menu structure for 1-socket servers (SR635 and SR655) differs from 2-socket servers (SR645 and SR665). Some menu items between the two groups of servers may appear very similar but the location and associated Redfish, OneCLI and ASU variable can be subtly different.

Summary of operating modes

The ThinkSystem SR635, SR655, SR645 and SR665 servers with 2nd and 3rd Gen AMD EPYC processors offer two preset operating modes, Maximum Efficiency and Maximum Performance. These modes are a collection of predefined low-level UEFI settings that simplify the task of tuning the server for either maximum performance or energy efficiency.

The two pre-defined modes are as follows:

► **Maximum Efficiency** (the default): Maximizes performance/watt efficiency while maintaining reasonable performance

► **Maximum Performance**: Achieves maximum performance at the expense of higher power consumption and lower energy efficiency

Summary for SR635 and SR655 (1-socket servers)

Table 1 summarizes the settings that are made for each mode selected for the SR635 and SR655 servers.

The values in the Category column (column 3) in each table are as follows:

► Recommended: Settings follow Lenovo's best practices and should not be changed without sufficient justification.

► Suggested: Settings follow Lenovo's general recommendation for a majority of workloads but these settings can be changed if justified by workload specific testing.

► Test: The non-default values for the Test settings can optionally be evaluated because they are workload dependent.

Table 1   UEFI Settings for Maximum Efficiency and Maximum Performance for SR635 and SR655

UEFI Setting	Page	Category	Maximum Efficiency	Maximum Performance
Operating Mode	9	Recommended	Maximum Efficiency	Maximum Performance
Determinism Slider	10	Recommended	Performance	Power

UEFI Setting	Page	Category	Maximum Efficiency	Maximum Performance
Core Performance Boost	10	Recommended	Auto (Enabled)	Auto (Enabled)
cTDP Control	11	Recommended	Auto	Manual and set cTDP to 300watt
Memory Speed	12	Recommended	2933	► 3200 if highest memory bandwidth is required and if higher memory latency can be tolerated ► 2933 if lower memory latency is required but with lower memory bandwidth vs. 3200
L1 Stream HW Prefetcher	13	Suggested	Auto (Optionally experiment with Disable)	Auto
L2 Stream HW Prefetcher	13	Suggested	Auto (Optionally experiment with Disable)	Auto
Global C-state Control	13	Recommended	Enabled	Enabled
SMT Mode	14	Suggested	Auto (Enabled)	Auto (Enabled)
Package Power Limit	15	Recommended	Auto	300watt
Memory Interleaving	15	Recommended	Auto	Auto
NUMA Nodes Per Socket	16	Test	NPS1 (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads	NPS1 (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads
EfficiencyModeEn	17	Recommended	Enabled	Auto
Chipselect interleaving	17	Recommended	Auto	Auto
LLC as NUMA Node	17	Test	Auto	Auto
SOC P-States	18	Recommended	Auto	Auto
P-State 1	19	Recommended	Enabled	Enabled
P-State 2	19	Recommended	Enabled	Enabled
DF C-states	19	Recommended	Enabled	Enabled
Memory Power Down Enable	20	Recommended	Enabled	Enabled

For the SR635 and SR655, Table 2 lists additional UEFI settings that you should consider for tuning for performance or energy efficiency. These setting are not part of the Maximum Efficiency and Maximum Performance modes.

Table 2   Other UEFI settings to consider for performance and efficiency for the SR635 and SR655

Menu Item	Page	Category	Comments
CPU Cores Activated (Downcore)	20	Test	This setting allows an administrator to power down some processor cores.
Preferred IO Bus	21	Test	This setting provides some I/O performance benefit, however, you should perform tests on using your workload and evaluate the results.

Menu Item	Page	Category	Comments
L1 Stream HW Prefetcher	22	Test	Optionally experiment with Disable for maximum efficiency
L2 Stream HW Prefetcher	22	Test	Optionally experiment with Disable for maximum efficiency
L1 Stride Prefetcher	22	Test	Optionally experiment with Disable for maximum efficiency
L1 Region Prefetcher	22	Test	Optionally experiment with Disable for maximum efficiency
L2 Up/Down Prefetcher	22	Test	Optionally experiment with Disable for maximum efficiency

Summary for SR645 and SR665 (2-socket servers)

Table 3 summarizes the settings that are made for each mode selected for the SR645 and SR665. The values in the Category column (column 3) in each table are as follows:

► Recommended: Settings follow Lenovo's best practices and should not be changed without sufficient justification.

► Suggested: Settings follow Lenovo's general recommendation for a majority of workloads but these settings can be changed if justified by workload specific testing.

► Test: The non-default values for the Test settings can optionally be evaluated because they are workload dependent.

Table 3   UEFI Settings for Maximum Efficiency and Maximum Performance for SR645 and SR665

Menu Item	Page	Category	Maximum Efficiency	Maximum Performance
Operating Mode	24	Recommended	Maximum Efficiency	Maximum Performance
Determinism Slider	25	Recommended	Performance	Power
Core Performance Boost	26	Recommended	Enable	Enable
cTDP	26	Recommended	Auto	Maximum cTDP supported by the CPU
Package Power Limit	27	Recommended	Auto	Maximum cTDP supported by the CPU
Memory Speed	28	Recommended	1 speed bin down from maximum speed (for example, if the maximum speed is 3200 MHz, the memory speed for this selection will be 2933 MHz.)	Maximum (For example with 2nd Gen AMD EPYC processors, 3200 MHz if highest memory bandwidth is required and if higher memory latency can be tolerated, or 2933 MHz if lower memory latency is required but with lower memory bandwidth vs. 3200. 3200 MHz provides the highest memory performance with 3rd Gen AMD EPYC processors)
Efficiency Mode	29	Recommended	Enable	Disable
4-Link xGMI Max Speed	30	Recommended	Minimum The value is 10.667GT/s.	Maximum supported speed (N). The value is 18GT/s for SR645 and SR665.

Menu Item	Page	Category	Maximum Efficiency	Maximum Performance
Global C-state Control	31	Recommended	Enable	Enable
SOC P-states	32	Recommended	Auto	Auto
DF C-States	32	Recommended	Enable	Enable
P-State 1	33	Recommended	Enable	Enable
P-State 2	33	Recommended	Enable	Enable
Memory Power Down Enable	34	Recommended	Enable	Enable
NUMA Nodes per Socket	34	Test	NPS1 (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads	NPS1 (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads
Memory Interleave	29	Recommended	Auto	Auto
ACPI SRAT L3 Cache as NUMA Domain	38	Test	Disable	Disable

For the SR645 and SR665, Table 4 lists additional UEFI settings that you should consider for tuning for performance or energy efficiency. These settings are not part of the Maximum Efficiency and Maximum Performance modes.

Table 4   Other UEFI settings to consider for performance and efficiency for the SR645 and SR665

Menu Item	Page	Category	Comments
SMT Mode	35	Suggested	It is recommended to keep enabled as default.
L1 Stream HW Prefetcher	36	Suggested	Optionally experiment with Disable for maximum efficiency
L2 Stream HW Prefetcher	36	Suggested	Optionally experiment with Disable for maximum efficiency
L1 Stride Prefetcher	36	Test	Optionally experiment with Disable for maximum efficiency
L1 Region Prefetcher	36	Test	Optionally experiment with Disable for maximum efficiency
L2 Up/Down Prefetcher	36	Test	Optionally experiment with Disable for maximum efficiency
Preferred IO Bus	37	Test	This setting provides some I/O performance benefit, however, you should perform tests on using your workload and evaluate the results.
xGMI Maximum Link Width	31	Suggested	This setting is only available for the SR645 and SR665. Auto sets maximum width based on the system capabilities.
PCIe Gen Speed Selection	38	Suggested	Suggest keep Maximum
CPPC	39	Suggested	Suggest keep Enable
BoostFmax	39	Suggested	Suggest keep Auto
DRAM Scrub Time	40	Suggested	Suggest keep 24 hour interval
Number of CPU Cores Activated	40	Test	This setting allows an administrator to power down some processor cores.
PCIe Ten Bit Tag Support	42	Test	This setting enabled provides PCIe performance benefits. If the PCIe device doesn't have Ten Bit Tag support, it will cause issues, suggest disable.

How to use OneCLI and Redfish to access these settings

In addition to using UEFI Setup, Lenovo also provides OneCLI/ASU variables and Redfish UEFI Setting Attribute names for managing system settings.

SR635 and SR655

The methods to use OneCLI/ASU variables and Redfish attributes in the SR635 and SR655 servers are as follows:

► OneCLI/ASU variable usage

Show current setting:

```
Onecli config show "Bios.<Attribute>" --override --log 5 --imm
<userid>:<password>@<IP Address>
```

Example:

```
onecli config show "Bios.Q00301_Operating_Mode" --override --log 5 --imm
USERID:PASSWORD@10.240.55.226
```

Set a setting:

```
Onecli config set "Bios.<Attribute>" "<choice>" –override –log 5 –imm
<userid>:<password>@<IP Address>
```

Example:

```
onecli config set "Bios.Q00301_Operating_Mode" "Maximum_Efficiency" --override
--log 5 --imm USERID:PASSWORD@10.240.55.226
```

► Redfish Attributes configure URL

Setting get URL: `https://<BMC IP>/redfish/v1/Systems/Self/Bios`

Setting set URL: `https://<BMC IP>/redfish/v1/Systems/Self/Bios/SD`

Example:

Get URL: `https://10.240.55.226/redfish/v1/Systems/Self/Bios`

Set URL: `https://10.240.55.226/redfish/v1/Systems/Self/Bios/SD`

► Redfish Value Names of Attributes

If no special description, choice name is same as possible values. If there is a space character (' '), dash character ('-') or forward slash character ('/') in the possible values, replace them with underline ("_"). This is because the Redfish standard doesn't support those special characters.

If you use OneCLI to configure the setting, OneCLI will automatically replace those characters with an underline character. However, if you use other Redfish tools, then you may need to replace them manually.

For example, "Operating Mode" has two choices: "Maximum Efficiency" and "Maximum Performance", their Redfish value names are `Maximum_Efficiency` and `Maximum_Performance`.

SR645 and SR665

The methods to use OneCLI/ASU variables and Redfish attributes in the SR645 and SR665 servers are as follows:

► OneCLI/ASU variable usage

Show current setting:

```
Onecli config show "<OneCLI/ASU Var>" --override --log 5 --imm
<userid>:<password>@<IP Address>
```

Example:

```
onecli config show "OperatingModes.ChooseOperatingMode" --override --log 5
--imm USERID:PASSWORD@10.240.218.89
```

Set a setting:

```
Onecli config set "<OneCLI/ASU Var>" "<choice>" —override —log 5 —imm
<userid>:<password>@<IP Address>
```

Example:

```
onecli config set "OperatingModes.ChooseOperatingMode" "Maximum Efficiency"
--override --log 5 --imm USERID:PASSWORD@10.240.218.89
```

► Redfish Attributes configure URL

Setting get URL: `https://<BMC IP>/redfish/v1/Systems/1/Bios`

► Redfish Value Names of Attributes

If no special description, choice name is same as possible values. If there are space character (' '), dash line ('-') or slash line ('/') in the possible values, remove them. Because Redfish standard doesn't support those special characters.

If you use OneCLI to configure the setting, OneCLI will automatically remove them. But if you use other Redfish tools, then you may need to remove those characters by yourselves.

Example:

"Operating Mode" has three choices: "Maximum Efficiency", "Maximum Performance" and "Custom Mode", their Redfish value names are "MaximumEfficiency", "MaximumPerformance" and "CustomMode".

For more detailed information on the BIOS schema, please refer to the DMTF website:

https://redfish.dmtf.org/redfish/schema_index

Usually, postman can be used for get/set BIOS schema:

https://www.getpostman.com/

The remaining sections in this paper provide details about each of these settings. We describe how to access the settings via System Setup (Press F1 during system boot).

UEFI menu items for SR635 and SR655

The items below are exposed to server administrators in UEFI menus that can be accessed by pressing F1 when a server is rebooted, through the BMC service, or through command line utilities such as Lenovo's Advanced Settings Utility (ASU) or OneCLI. These parameters are exposed because they are commonly changed from their default values to fine tune server performance for a wide variety of customer use cases.

Menu items described for the SR635 and SR655:

Operating Mode

This setting is used to set multiple processor and memory variables at a macro level.

Choosing one of the predefined Operating Modes is a way to quickly set a multitude of processor and memory variables. It is less fine grained than individually tuning parameters but does allow for a simple "one-step" tuning method for two primary scenarios.

> **Tip:** Prior to optimizing a workload for maximum performance, it is recommended to set the Operating Mode to "Maximum Performance" and then reboot rather than simply starting from the Maximum Efficiency default mode and then modifying individual UEFI parameters. If you don't do this, some settings may be unavailable for configuration.

This setting is accessed as follows:

- ► System setup: **Main** → **Operating Modes** → **Operating Mode**
- ► Redfish: `Q00301_Operating_Mode`

Possible values:

- ► **Maximum Efficiency** (default)

  Maximizes the performance / watt efficiency with a bias towards power savings.

- ► **Maximum Performance**

  Maximizes the absolute performance of the system without regard for power savings. Most power savings features are disabled and additional memory power / performance settings are exposed.

Determinism Slider

The determinism slider allows you to select between uniform performance across identically configured systems in you data center (by setting all servers to the Performance setting) or maximum performance of any individual system but with varying performance across the data center (by setting all servers to the Power setting).

When setting Determinism to Performance, ensure that cTDP and PPL are set to the same value (see "cTDP Control" on page 11 and "PPL (Package Power Limit)" on page 15 for more details). The default (Auto) setting for most processors will be Performance Determinism mode.

Determinism Slider should be set to Power for maximum performance and set to Performance for lower performance variability.

This setting is accessed as follows:

► System Setup:

- **Main** → **Operating Modes** → **Determinism Slider**
- **Advanced** → **CPU Configuration** → **Determinism Slider**

► Redfish: `Q00059_Determinism_Slider`

Possible values:

► **Auto**

Use the default determinism for the processor. For all EPYC 7002 and 7003 Series processors, the default is Performance.

► **Power**

Ensure maximum performance levels for each CPU in a large population of identically configured CPUs by throttling CPUs only when they reach the same cTDP. Forces processors that are capable of running at the rated TDP to consume the TDP power (or higher).

► **Performance** (default)

Ensure consistent performance levels across a large population of identically configured CPUs by throttling some CPUs to operate at a lower power level

Core Performance Boost

Core Performance Boost (CPB) is similar to Intel Turbo Boost Technology. CPB allows the processor to opportunistically increase a set of CPU cores higher than the CPU's rated base clock speed, based on the number of active cores, power and thermal headroom in a system.

Two levels of boosted P-states are supported.

► Compute units can be placed in the first level of boosted P-states if the processor power consumption remains within the TDP limit.

► The second level of boosted P-states can only be achieved if a subset of compute units are in CC6 low-power sleep state and the processor power consumption remains within the TDP limit.

When more than two physical CPU cores facilitating four hardware threads (2C4T) per die are active, the maximum CPU frequency achieved is hardware P0.

Consider using CPB when you have applications that can benefit from clock frequency enhancements. Avoid using this feature with latency-sensitive or clock frequency-sensitive applications, or if power draw is a concern. Some workloads do not need to be able to run at the maximum capable core frequency to achieve acceptable levels of performance.

To obtain better power efficiency, there is the option of setting a maximum core boost frequency. This setting does not allow you to set a fixed frequency. It only limits the maximum boost frequency. If the BoostFmax is set to something higher than the boost algorithms allow, the SoC will not go beyond the allowable frequency that the algorithms support.

This setting is accessed as follows:

▶ System setup:

— **Main** → **Operating Modes** → **Core Performance Boost**
— **Advanced** → **CPU Configuration** → **Core Performance Boost**

▶ Redfish: Q00052_Core_Performance_Boost

Possible values:

▶ **Auto** (default)

The Auto setting maps to enabling Core Performance Boost.

▶ **Disabled**

Disables Core Performance Boost so the processor cannot opportunistically increase a set of CPU cores higher than the CPU's rated base clock speed.

cTDP Control

Configurable Thermal Design Power (cTDP) allows you to modify the platform CPU cooling limit. A related setting, Package Power Limit (PPL), discussed in the next section, allows the user to modify the CPU Power Dissipation Limit.

Many platforms will configure cTDP to the maximum supported by the installed CPU. For example, an EPYC 7502 part has a default TDP of 180W but has a cTDP maximum of 200W. Most platforms also configure the PPL to the same value as the cTDP. Please refer to Table 9 on page 48 and Table 10 on page 49to get maximum cTDP of you installed processor.

If the Determinism slider parameter is set to Performance (see "Determinism Slider" on page 10), cTDP and PPL must be set to the same value, otherwise, the user can set PPL to a value lower than cTDP to reduce system operating power. The CPU will control CPU boost to keep socket power dissipation at or below the specified Package Power Limit.

For maximum performance, set cTDP and PPL to the maximum cTDP value supported by the CPU. For increased energy efficiency, set cTDP and PPL to Auto which sets both parameters to the CPU's default TDP value.

This setting is accessed as follows:

▶ System setup:

— **Main** → **Operating Modes** → **cTDP Control**
— **Advanced** → **CPU Configuration** → **cTDP Control**

▶ Redfish: Q00060_cTDP_Control

Possible values:

- ► cTDP
  - – Manual

    Set customized configurable TDP
  - – **Auto** (default)

    Use the platform and the default TDP for the installed processor.
- ► cTDP
  - – Set the configurable TDP (in Watts)

Memory Speed

The memory speed setting determines the frequency at which the installed memory will run. Consider changing the memory speed setting if you are attempting to conserve power, since lowering the clock frequency to the installed memory will reduce overall power consumption of the DIMMs.

With the second-generation AMD EPYC processors, setting the memory speed to 3200 MHz results in higher memory bandwidth when compared to operation at 2933 MHz, but will also result in a higher memory latency. This higher latency is due to the memory bus clock and the memory/IO die clock not being synchronized when the memory speed is set to 3200 MHz. Customers should evaluate both memory speeds for their applications if supported by 2nd generation AMD EPYC 7002 processors.

With the third-generation AMD EPYC processors, setting the memory speed to 3200MHz does not result in higher memory latency when compared to operation at 2933 MHz. The highest memory performance is achieved when the memory speed is set to 3200 MHz and the memory DIMMs are capable of operating at 3200 MHz.

This setting is accessed as follows:

- ► System setup:
  - – **Main → Memory Speed**
  - – **Main → Operating Modes → Memory Speed**
- ► Redfish: Q00304_Memory_Speed

Possible values:

- ► **Auto**

  Max Performance sets the memory to the maximum allowed frequency as dictated by the type of CPU installed. Will support 3200 MHz operation with 1 DIMM Per Channel on 1-socket servers if 3200 MHz memory is installed and if the processor supports a 3200 MHz memory speed.
- ► **2933 MHz** (default)

  Redfish value name is Memory_Speed_2933_MHz.
- ► **2666 MHz**

  Redfish value name is Memory_Speed_2666_MHz.
- ► **2400 MHz**

  Redfish value name is Memory_Speed_2400_MHz.

Data Prefetchers

Most workloads will benefit from the L1 & L2 Stream Hardware prefetchers gathering data and keeping the core pipeline busy. By default, both prefetchers are enabled.

Application information access patterns, which tend to be relatively predictable, benefit greatly from prefetching. Most typical line-of-business, virtualization and scientific applications benefit from having pre-fetching enabled, however, there are however some workloads (for example, the SPECjbb 2015 Java application benchmark) that are very random in nature and will actually obtain better overall performance by disabling one or both of the prefetchers.

Further, the L1 and L2 stream hardware prefetchers can consume disproportionately more power vs. the gain in performance when enabled. Customers should therefore evaluate the benefit of prefetching vs. the non-linear increase in power if sensitive to energy consumption.

Note: L1 Stride Prefetcher, L1 Region Prefetcher and L2 Up/Down Prefetcher settings are only available for 3rd Gen EPYC processor.

This setting is accessed as follows:

► System Setup:

   — **Main → Operating Modes → L1 Stream HW Prefetcher**
   — **Main → Operating Modes → L2 Stream HW Prefetcher**
   — **Advanced → CPU Configuration → L1 Stream HW Prefetcher**
   — **Advanced → CPU Configuration → L2 Stream HW Prefetcher**

► Redfish:

   — Q00054_L1_Stream_HW_Prefetcher
   — Q00055_L2_Stream_HW_Prefetcher

Possible values:

► **Disabled**

   Disable L1/L2 Stream HW Prefetcher

► **Enabled**

   Enable L1/L2 Stream HW Prefetcher

► **Auto** (default)

   Maps to Enabled, enable the prefetcher.

Global C-State Control

C-states are idle power saving states. This setting enables and disables C-states on the server across all cores. When disabled, the CPU cores can only be in C0 (active) or C1 state. C1 state can never be disabled. A CPU core will be in the C1 state if the core is halted by the operating system.

Lenovo generally recommends that Global C-State Control remain enabled, however consider disabling it for low-jitter use cases.

This setting is accessed as follows:

► System setup:

— **Main** → **Operating Modes** → **Global C-State Control**
— **Advanced** → **CPU Configuration** → **Global C-State Control**

► Redfish: `Q00056_Global_C_state_Control`

Possible values:

► **Disabled**

I/O based C-state generation and Data Fabric (DF) C-states are disabled.

► **Enabled** (default)

I/O based C-state generation and DF C-states are enabled.

► **Auto**

Map to Enabled.

SMT Mode

Simultaneous multithreading (SMT) is similar to Intel Hyper-Threading Technology, the capability of a single core to execute multiple threads simultaneously. An OS will register an SMT-thread as a logical CPU and attempt to schedule instruction threads accordingly. All processor cache within a Core Complex (CCX) is shared between the physical core and its corresponding SMT-thread.

In general, enabling SMT benefits the performance of most applications. Certain operating systems and hypervisors, such as VMware ESXi, can schedule instructions such that both threads execute on the same core. SMT takes advantage of out-of-order execution, deeper execution pipelines and improved memory bandwidth in today's processors to be an effective way of getting all of the benefits of additional logical CPUs without having to supply the power necessary to drive another physical core.

Start with SMT enabled since SMT generally benefits the performance of most applications, however, consider disabling SMT in the following scenarios:

► Some workloads, including many HPC workloads, observe a performance neutral or even performance negative result when SMT is enabled.

► Using multiple execution threads per core requires resource sharing and is a possible source of inconsistent system response. So disabling SMT could give benefit on some low-jitter use case.

► Some application license fees are based on the number of hardware threads enabled, not just the number of physical cores present. For this reason, disabling SMT on your EPYC 7002 and 7003 Series processor may be desirable to reduce license fees.

► Some older operating systems that have not enabled support for the x2APIC within the EPYC 7002 and 7003 Series   processor, which is required to support beyond 255 threads. If you are running an operating system that does not support AMD's x2APIC implementation, and have two 64-core processors installed, you will need to disable SMT.

Operating systems such as Windows Server 2012 and Windows Server 2016 do not support x2APIC. Please refer to the following article for details:

https://support.microsoft.com/en-in/help/4514607/windows-server-support-and-installation-instructions-for-amd-rome-proc

This setting is accessed as follows:

- ► System setup:
    - — **Main** $\rightarrow$ **Operating Modes** $\rightarrow$ **SMT Mode**
    - — **Advanced** $\rightarrow$ **CPU Configuration** $\rightarrow$ **SMT Mode**
- ► Redfish: `Q00051_SMT_Mode`

Possible values:

- ► **Auto** (default)

    Enables simultaneous multithreading

- ► **Disabled**

    Disables simultaneous multithreading so that only one thread or CPU instruction stream is run on a physical CPU core

PPL (Package Power Limit)

The parameter sets the CPU package power limit. The maximum value allowed for PPL is the cTDP limit. Set PPL to the cTDP Maximum value when maximum performance is desired. PPL can be set to the cTDPminimum value or lower but reaching the set value of PPL is not guaranteed when it is set to less than cTDPminimum.

This setting is accessed as follows:

- ► System setup: **Main** $\rightarrow$ **Operating Modes** $\rightarrow$ **Package Power Limit Control**
- ► Redfish: `Q00305_Package_Power_Limit_Control`

Possible values:

- ► **Manual**

    If a manual value entered that is larger than the maximum value allowed (cTDP Maximum), the value will be internally limited to maximum allowable value.

- ► **Auto** (default)

    Set to maximum value allowed by installed CPU

Memory Interleaving

This setting allows interleaved memory accesses across multiple memory channels in each socket, providing higher memory bandwidth. Interleaving generally improves memory performance so the Auto (default) setting is recommended.

This setting is accessed as follows:

- ► System Setup:
    - — **Main** $\rightarrow$ **Operating Modes** $\rightarrow$ **Memory Interleaving**
    - — **Advanced** $\rightarrow$ **Memory Configuration** $\rightarrow$ **Memory Interleaving**
- ► Redfish: `Q00081_Memory_interleaving`

Possible values:

- ► **Disabled**

    No interleaving

▶ **Auto** (default)

  Interleaving is automatically enabled if memory DIMM configuration supports it.

NUMA nodes per socket

This setting lets you specify the number of desired NUMA nodes per socket. NPS0 will attempt to interleave two sockets together into one NUMA node. NPS0 is not supported on the SR635 and SR655 one-socket servers.

Second and third-generation AMD EPYC processors support a varying number of NUMA Nodes per Socket depending on the internal NUMA topology of the processor. In one-socket servers, the number of NUMA Nodes per socket can be 1, 2 or 4 though not all values are supported by every processor. See Table 7 on page 47 and Table 8 on page 48 for the NUMA nodes per socket options available for each processor.

Applications that are highly NUMA optimized can improve performance by setting the number of NUMA Nodes per Socket to a supported value greater than 1.

This setting is accessed as follows:

▶ System Setup: **Main → Operating Modes → NUMA nodes per socket**

▶ Redfish: Q00302_NUMA_nodes_per_socket

Possible values:

▶ **Auto**

  One NUMA node per socket, NPS1.

▶ **NPS4**

  – Four NUMA nodes per socket, one per Quadrant.

  – Requires symmetrical Core Cache Die (CCD) configuration across Quadrants of the SoC.

  – Preferred Interleaving: 2-channel interleaving using channels from each quadrant.

▶ **NPS2**

  – Two NUMA nodes per socket, one per Left/Right Half of the SoC.

  – Requires symmetrical CCD configuration across left/right halves of the SoC.

  – Preferred Interleaving: 4-channel interleaving using channels from each half.

▶ **NPS1** (default)

  – One NUMA node per socket.

  – Available for any CCD configuration in the SoC.

  – Preferred Interleaving: 8-channel interleaving using all channels in the socket.

EfficiencyModeEn

This setting enables an energy efficient mode of operation internal to the AMD EPYC Gen 2 and Gen 3 processors at the expense of performance. The setting should be enabled when energy efficient operation is desired from the processor. If desired, set it to Auto which disables the feature when maximum performance is desired.

This setting is accessed as follows:

▶ System Setup: **Main** → **Operating Modes** → **EfficiencyModeEn**

▶ Redfish: `Q00303_EfficiencyModeEn`

Possible values:

▶ **Auto**

Use performance optimized CCLK DPM settings

▶ **Enabled** (default)

Use power efficiency optimized CCLK DPM settings

Chipselect Interleaving (Rank Interleaving)

This setting specifies if the system should use a DRAM rank also known as chipselect interleaving. This feature will spread memory accesses across the ranks of memory within a memory channel and will increase memory block access performance.

This feature requires the following memory configuration:

▶ Populated DIMMs have same rank size and type. Strongly recommend you populate DIMMs which have same part number.

▶ Number of ranks is a power of two.

This setting is accessed as follows:

▶ System Setup:

   — **Main** → **Operating Modes** → **Chipselect Interleaving**
   — **Advanced** → **Memory Configuration** → **Chipselect Interleaving**

▶ Redfish: `Q00083_Chipselect_interleaving`

Possible values:

▶ **Disabled**

No interleaving. When your application data access is random, and very sensitive about latency, then you may disable it.

▶ **Auto** (default)

Interleaving across the ranks within the memory channel.

LLC as NUMA node

Specifies whether the processor last-level caches (LLCs) are exposed to the operating system as NUMA nodes.

In certain server applications where workloads are managed by a remote job scheduler, it is desirable to pin execution to a single NUMA node, and preferably to share a single L3 cache

within that node. Hence BIOS Setup should support an L3AsNumaNode (Boolean) option to create a NUMA node for each Core Complex (CCX) L3 Cache in the system.

When enabled, this setting can improve performance for highly NUMA optimized workloads if workloads or components of workloads can be pinned to cores in a CCX and if they can benefit from sharing an L3 cache.

This setting is accessed as follows:

► System Setup: **Main** → **Operating Modes** → **LCC as NUMA Node**

► Redfish: Q00308_LCC_as_NUMA_Node

Possible values:

► **Disabled**

  LLCs are not exposed to the operating system as NUMA nodes.

► **Enabled**

  LLCs are exposed to the operating system as NUMA nodes.

► **Auto** (default)

  Maps to Disable where the LLCs are not exposed to the operating system as NUMA nodes.

SOC P-states

Infinity Fabric is a proprietary AMD bus that connects all the Core Cache Dies (CCDs) to the IO die inside the CPU as shown in Figure 5 on page 47. SOC P-states is the Infinity Fabric (Uncore) Power States setting. When Auto is selected the CPU SOC P-states will be dynamically adjusted. That is, their frequency will dynamically change based on the workload. Selecting P0, P1, P2, or P3 forces the SOC to a specific P-state frequency.

SOC P-states functions cooperatively with the Algorithm Performance Boost (APB) which allows the Infinity Fabric to select between a full-power and low-power fabric clock and memory clock based on fabric and memory usage. Latency sensitive traffic may be impacted by the transition from low power to full power. Setting APBDIS to 1 (to disable APB) and SOC P-states=0 sets the Infinity Fabric and memory controllers into full-power mode. This will eliminate the added latency and jitter caused by the fabric power transitions.

The following examples illustrate how SOC P-states and APBDIS function together:

► If SOC P-states=Auto then APBDIS=0 will be automatically set. The Infinity Fabric can select between a full-power and low-power fabric clock and memory clock based on fabric and memory usage.

► If SOC-P-states=<P0, P1, P2, P3> then APBDIS=1 will be automatically set. The Infinity Fabric and memory controllers are set in full-power mode.

► If SOC P-states=P0 which results in APBDIS=1, the Infinity Fabric and memory controllers are set in full-power mode. This results in the highest performing Infinity Fabric P-state with the lowest latency jitter.

This setting is accessed as follows:

► System setup: **Main** → **Operating Modes** → **SOC P-states**

► Redfish: Q00311_SOC_P_states

Possible values:

- ▶ **Auto** (default)

  When Auto is selected the CPU SOC P-states (uncore P-states) will be dynamically adjusted.
- ▶ **P0**: Highest-performing Infinity Fabric P-state
- ▶ **P1**: Next-highest-performing Infinity Fabric P-state
- ▶ **P2**: Next next-highest-performing Infinity Fabric P-state
- ▶ **P3**: Minimum Infinity Fabric power P-state

P-State 1

This setting enables or disable the CPU's P1 operating state.

This setting is accessed as follows:

- ▶ System setup: **Main** → **Operating Modes** → **P-State 1**
- ▶ Redfish: Q00313_P_State_1

Possible values:

- ▶ **Enabled** (default)

  Enables CPU P1 P-state.
- ▶ **Disabled**

  Disable CPU P1 P-state

P-State 2

This setting enables or disable the CPU's P2 operating state.

This setting is accessed as follows:

- ▶ System setup: **Main** → **Operating Modes** → **P-State 2**
- ▶ Redfish: Q00314_P_State_2

Possible values:

- ▶ **Enabled** (default)

  Enables CPU P2 P-state.
- ▶ **Disabled**

  Disable CPU P2 P-state.

DF (Data Fabric) C-States

Much like CPU cores, the Infinity Fabric can go into lower power states while idle. However, there will be a delay changing back to full-power mode causing some latency jitter. In a low latency workload, or one with bursty I/O, one could disable this feature to achieve more performance with the tradeoff of higher power consumption.

This setting is accessed as follows:

- ▶ System setup: **Main** → **Operating Modes** → **DF C-states**

▶ Redfish: `Q00312_DF_C_states`

Possible values:

▶ **Enabled** (default)

Enables Data Fabric C-states. Data Fabric C-states may be entered when all cores are in CC6.

▶ **Disabled**

Disable Data Fabric (DF) C-states

Memory Power Down Enable

Low-power feature for DIMMs. Lenovo generally recommends that Memory Power Down remain enabled, however consider disabling it for low-latency use cases.

This setting is accessed as follows:

▶ System setup:

— **System Settings → Operating Modes → Memory Power Down Enable**
— **System Settings → Memory → Memory Power Down Enable**

▶ Redfish: `Q00310_Memory_Power_Down_Enable`

Possible values:

▶ **Enabled** (default)

Enables low-power features for DIMMs.

▶ **Disabled**

CPU Cores Activated (Downcore)

UEFI allows the administrator to shut down cores in a server. This setting powers off a set number of cores for each processor in a system. As opposed to restricting the number of logical processors an OS will run on, this setting directly affects the number of cores powered on by turning off the core level power gates on each processor.

Manipulating the number of physically powered cores is primarily used in three scenarios:

▶ Where users have a licensing model that supports a certain number of active cores in a system

▶ Where users have poorly threaded applications but require the additional LLC available to additional processors, but not the core overhead

▶ Where users are looking to limit the number of active cores in an attempt to reclaim power and thermal overhead to increase the probability of Performance Boost being engaged.

This setting is accessed as follows:

▶ System Setup: **Advanced → CPU Configuration → CPU Cores Activated**

▶ Redfish: `Q00053_CPU_Cores_Activated`

Possible values:

▶ **Auto** (default):

Enable all cores

- ► **2 Cores Per Die**:

    Enable 2 cores for each Core Cache Die. Redfish value name is
    `CPU_Cores_Activated_2_Cores_Per_Die`.

- ► **4 Cores Per Die**:

    Enable 4 cores for each Core Cache Die. Redfish value name is
    `CPU_Cores_Activated_4_Cores_Per_Die`.

- ► **6 Cores Per Die**:

    Enable 6 cores for each Core Cache Die. Redfish value name is
    `CPU_Cores_Activated_6_Cores_Per_Die`.

AMD EPYC Gen 2 and Gen 3 processors have a multi-die topology consisting of two or more
Core Cache Die (CCD). Each CCD contains processor cores plus Level 2 and Level 3
caches. See Figure 5 on page 47 and Table 7 on page 47 and Table 8 on page 48 for more
information on CCDs.

Preferred IO Bus

AMD Preferred I/O complements Relaxed Ordering (RO) and ID-Based Ordering (IDO),
rather than replace them. As with most performance tuning options, your exact workload may
not perform better with Preferred I/O enabled but rather with RO or IDO. Lenovo recommends
that you test your workload under all scenarios to find the most efficient setting.

If your PCIe device supports IDO or RO, and it is enabled, you can try enabling Preferred IO
to determine whether your specific use case realizes an additional performance
improvement. If the device does not support either of those features for your high bandwidth
adapter, you can experiment with the Preferred IO mode capability available on the EPYC
7002 and 7003 Series processors. This setting allows devices on a single PCIe bus to obtain
improved DMA write performance.

Note: The Preferred I/O can be enabled for only a single PCIe root complex (RC) and will
affect all PCIe endpoints behind that RC. This setting gives priority to the I/O devices
attached to the PCIe slot(s) associated with the enabled RC for I/O transactions.

This setting is accessed as follows:

- ► System Setup:
    - — **Advanced → CPU Configuration → Preferred IO Bus**
    - — **Advanced → CPU Configuration → Preferred I/O Bus Number**
- ► Redfish:
    - — `Q00309_Preferred_IO_Bus`
    - — `Q00307_Preferred_I_O_Bus_Number`

Possible values for Preferred IO Bus:

- ► **No Priority** (default):

    All PCIe buses have equal priority. Use this setting unless you have an I/O device, such as
    a RAID controller or network adapter, that has a clear need to be prioritized over all other
    I/O devices installed in the server.

- ► **Preferred**

    Enter a preferred I/O bus number in the range 00h-FFh to setup item "Preferred I/O
    Device Bus Number" to specify the bus number for which device(s) you wish to enable
    preferred I/O.

Data Prefetchers

- ► L1 Stream Prefetcher: Uses history of memory access patterns to fetch next line into the L1 cache when cached lines are resued within a certain time period or access sequentially.

- ► L1 Stride Prefetcher: Uses memory access history to fetch additional data lines into L1 cache when each access is a constant distance from previous.

- ► L1 Region Prefetcher: Uses memory access history to fetche additional data line into L1 cache when the data access for a given instruction tends to be followed by a consistent pattern of subsequent access

- ► L2 Stream Prefetcher: Uses history of memory access patterns to fetch next line into the L1 cache when cached lines are resued within a certain time period or access sequentially.

- ► L2 Up/Down Prefetcher: Uses memory access history to determine whether to fetch the next or previous line for all memory accesses.

These fetchers use memory access history to determine whether to fetch the text or previous line for all memory access. Most workloads will benefit from these prefetchers gathering data and keeping the core pipeline busy. By default, these prefetchers all are enabled.

Application information access patterns, which tend to be relatively predictable, benefit greatly from prefetching. Most typical line-of-business, virtualization and scientific applications benefit from having pre-fetching enabled, however, there are however some workloads (for example, the SPECjbb 2015 Java application benchmark) that are very random in nature and will actually obtain better overall performance by disabling some of the prefetchers.

Further, the L1 and L2 stream hardware prefetchers can consume disproportionately more power vs. the gain in performance when enabled. Customers should therefore evaluate the benefit of prefetching vs. the non-linear increase in power if sensitive to energy consumption.

> **Note:** L1 Stride Prefetcher, L1 Region Prefetcher and L2 Up/Down Prefetcher settings are only available for 3rd Gen EPYC processor.

This setting is accessed as follows:

- ► System setup:

    - — **Advanced → CPU Configuration → L1 Stream HW Prefetcher**
    - — **Advanced → CPU Configuration → L2 Stream HW Prefetcher**
    - — **Advanced → CPU Configuration → L1 Stride Prefetcher**
    - — **Advanced → CPU Configuration → L1 Region Prefetcher**
    - — **Advanced → CPU Configuration → L2 Up/Down Prefetcher**

- ► Redfish:

    - — Q00054_L1_Stream_HW_Prefetcher
    - — Q00055_L2_Stream_HW_Prefetcher
    - — Q00066_L1_Stride_Prefetcher
    - — Q00067_L1_Region_Prefetcher
    - — Q00068_L2_Up_Down_Prefetcher

Possible values:

- ► **Auto** (default)

    Maps to Enabled

- ▶ **Disabled**

  Disable corresponding Prefetcher
- ▶ **Enabled**

  Enable corresponding Prefetcher

UEFI menu items for SR645 and SR665

Below items are exposed to server administrators in UEFI menus that can be accessed by pressing F1 when a server is rebooted, through the XClarity Controller (XCC) service processor, or through command line utilities such as Lenovo's Advanced Settings Utility (ASU) or OneCLI. These parameters are exposed because they are commonly changed from their default values to fine tune server performance for a wide variety of customer use cases.

Menu items described for the SR645 and SR665:

Operating Mode

This setting is used to set multiple processor and memory variables at a macro level.

Choosing one of the predefined Operating Modes is a way to quickly set a multitude of processor, memory, and miscellaneous variables. It is less fine grained than individually tuning parameters but does allow for a simple "one-step" tuning method for two primary scenarios.

> **Tip:** Prior to optimizing a workload for maximum performance, it is recommended to set the Operating Mode to "Maximum Performance" and then reboot rather than simply starting from the Maximum Efficiency default mode and then modifying individual UEFI parameters. If you don't do this, some settings may be unavailable for configuration.

This setting is accessed as follows:

- ► System setup: **System Settings** → **Operating Modes** → **Choose Operating Mode**
- ► OneCLI/ASU variable: `OperatingModes.ChooseOperatingMode`

► Redfish: `OperatingModes_ChooseOperatingMode`

Default Setting: **Maximum Efficiency**

Possible values:

► **Maximum Efficiency**

Maximizes the performance / watt efficiency with a bias towards power savings.

► **Maximum Performance**

Maximizes the absolute performance of the system without regard for power savings. Most power savings features are disabled, and additional memory power / performance settings are exposed.

► **Custom Mode**

Allow user to customize the performance settings. Custom Mode will inherit the UEFI settings from the previous preset operating mode. For example, if the previous operating mode was the Maximum Performance operating mode and then Custom Mode was selected, all the settings from the Maximum Performance operating mode will be inherited.

Determinism Slider

The determinism slider allows you to select between uniform performance across identically configured systems in you data center (by setting all servers to the Performance setting) or maximum performance of any individual system but with varying performance across the data center (by setting all servers to the Power setting).

When setting Determinism to Performance, ensure that cTDP and PPL are set to the same value (see Configurable TDP control and PPL (Package Power Limit) for more details). The default (Auto) setting for most processors will be Performance Determinism mode.

Determinism Slider should be set to Power for maximum performance and set to Performance for lower performance variability.

This setting is accessed as follows:

► System setup:

– **System Settings → Operating Modes → Determinism Slider**
– **System Settings → Processors → Determinism Slider**

► OneCLI/ASU variable: `Processors.DeterminismSlider`

► Redfish: `Processors_DeterminismSlider`

Possible values:

► **Power**

Ensure maximum performance levels for each CPU in a large population of identically configured CPUs by throttling CPUs only when they reach the same cTDP. Forces processors that are capable of running at the rated TDP to consume the TDP power (or higher).

► **Performance** (default)

Ensure consistent performance levels across a large population of identically configured CPUs by throttling some CPUs to operate at a lower power level

Core Performance Boost

Core Performance Boost (CPB) is similar to Intel Turbo Boost Technology. CPB allows the processor to opportunistically increase a set of CPU cores higher than the CPU's rated base clock speed, based on the number of active cores, power and thermal headroom in a system.

Two levels of boosted P-states are supported.

► Compute units can be placed in the first level of boosted P-states if the processor power consumption remains within the TDP limit.

► The second level of boosted P-states can only be achieved if a subset of compute units are in CC6 state and the processor power consumption remains within the TDP limit.

When more than two physical CPU cores facilitating four hardware threads (2C4T) per die are active, the maximum CPU frequency achieved is hardware P0.

Consider using CPB when you have applications that can benefit from clock frequency enhancements. Avoid using this feature with latency-sensitive or clock frequency-sensitive applications, or if power draw is a concern. Some workloads do not need to be able to run at the maximum capable core frequency to achieve acceptable levels of performance.

To obtain better power efficiency, there is the option of setting a maximum core boost frequency. This setting does not allow you to set a fixed frequency. It only limits the maximum boost frequency. If the BoostFmax is set to something higher than the boost algorithms allow, the SoC will not go beyond the allowable frequency that the algorithms support.

This setting is accessed as follows:

► System setup:

— **System Settings → Operating Modes → Core Performance Boost**
— **System Settings → Processors → Core Performance Boost**

► OneCLI/ASU variable: `Processors.CorePerformanceBoost`

► Redfish: `Processors_CorePerformanceBoost`

Possible values:

► **Disable**

Disables Core Performance Boost so the processor cannot opportunistically increase a set of CPU cores higher than the CPU's rated base clock speed.

► **Enable** (default)

When set to Enable, cores can go to boosted P-states.

cTDP (Configurable TDP)

Configurable Thermal Design Power (cTDP) allows you to modify the platform CPU cooling limit. A related setting, Package Power Limit (PPL), discussed in the next section, allows the user to modify the CPU Power Dissipation Limit.

Many platforms will configure cTDP to the maximum supported by the installed CPU. For example, an EPYC 7502 part has a default TDP of 180W but has a cTDP maximum of 200W. Most platforms also configure the PPL to the same value as the cTDP. Please refer to AMD EPYC Processor cTDP Range Table to get maximum cTDP of your installed processor.

If the Determinism slider parameter is set to Performance (see Determinism slider), cTDP and PPL must be set to the same value, otherwise, the user can set PPL to a value lower

than cTDP to reduce system operating power. The CPU will control CPU boost to keep socket power dissipation at or below the specified Package Power Limit.

For maximum performance, set cTDP and PPL to the maximum cTDP value supported by the CPU. For increased energy efficiency, set cTDP and PPL to Auto which sets both parameters to the CPU's default TDP value.

This setting is accessed as follows:

▶ System setup:

    — **System Settings → Operating Modes → cTDP**
    — **System Settings → Processors → cTDP**

▶ OneCLI/ASU variable: `Processors.cTDP`

▶ Redfish: `Processors_cTDP`

Possible values:

▶ cTDP

    — **Auto** (default)

        Use the platform and the default TDP for the installed processor. cTDP = TDP.

    — **Maximum**

        Maximum sets the maximum allowed cTDP value for the installed CPU SKU. Maximum could be greater than default TDP. Please refer to Table 9 on page 48 and Table 10 on page 49 for maximum cTDP of each CPU SKU.

    — **Manual**

        Set customized configurable TDP

▶ cTDP Manual

    Set the configurable TDP (in Watts). If a manual value is entered that is larger than the max value allowed, the value will be internally limited to the maximum allowable value.

PPL (Package Power Limit)

The parameter sets the CPU package power limit. The maximum value allowed for PPL is the cTDP limit. Set PPL to the cTDP Maximum value when maximum performance is desired. PPL can be set to the cTDP Minimum value or lower but reaching the set value of PPL is not guaranteed when it is set to less than cTDP Minimum.

This setting is accessed as follows:

▶ System setup:

    — **System Settings → Operating Modes → Package Power Limit**
    — **System Settings → Processors → Package Power Limit**

▶ OneCLI/ASU variable: `Processors.PackagePowerLimit`

▶ Redfish: `Processors_PackagePowerLimit`

Possible values:

▶ **Auto** (default)

    Set to maximum value allowed by installed CPU

▶ **Maximum**

    The maximum value allowed for PPL is the cTDP limit.

► **Manual**

If a manual value entered that is larger than the maximum value allowed (cTDP Maximum), the value will be internally limited to maximum allowable value.

Memory Speed

The memory speed setting determines the frequency at which the installed memory will run. Consider changing the memory speed setting if you are attempting to conserve power, since lowering the clock frequency to the installed memory will reduce overall power consumption of the DIMMs.

With the second-generation AMD EPYC processors, setting the memory speed to 3200 MHz results in higher memory bandwidth when compared to operation at 2933 MHz, but will also result in a higher memory latency. This higher latency is due to the memory bus clock and the memory/IO die clock not being synchronized when the memory speed is set to 3200 MHz. Customers should evaluate both memory speeds for their applications if supported by the second-generation AMD processor.

With the third-generation AMD EPYC processors, setting the memory speed to 3200MHz does not result in higher memory latency when compared to operation at 2933 MHz. The highest memory performance is achieved when the memory speed is set to 3200 MHz and the memory DIMMs are capable of operating at 3200 MHz.

This setting is accessed as follows:

► System setup:

— **System Settings** → **Memory** → **Memory Speed**
— **System Information** → **System Summary** → **Memory Speed**

► OneCLI/ASU variable: `Memory.MemorySpeed`

► Redfish: `Memory_MemorySpeed`

Possible values:

► **N**

N is the actual maximum supported speed and is auto-calculated based on the CPU SKU, DIMM type, number of DIMMs installed per channel, and the capability of the system. The values in setup show the actual frequency (for example, 3200 MHz, 2933 MHz, 2666 MHz, 2400 MHz).

► **N-1** (default)

1 speed bin down from maximum speed. If the maximum supported is 3200 MHz, then this "N-1" is 2933 MHz.

► **N-2**

2 speed bins down from maximum speed

► **Minimum**

The system operates at the rated speed of the slowest DIMM in the system when populated with different speed DIMMs. Installing DIMMs which have a rated speed below 2400 MHz will result in the memory speed getting set to the Minimum value.

Possible Redfish value names are:

► `MemorySpeed_3200MHz`
► `MemorySpeed_2933MHz`
► `MemorySpeed_2666MHz`

- ► `MemorySpeed_2400MHz`
- ► `Minimum`

> **Tip:** The ThinkSystem SR645 and SR665 servers support memory speed operation at 3200 MHz when the memory is configured with two DIMMs per memory channel and when Performance+ RDIMMs are installed.
>
> Consult the ThinkSystem SR645 and SR665 product guides for more information:
> - ► SR645: https://lenovopress.com/lp1280-thinksystem-sr645-server
> - ► SR665: https://lenovopress.com/lp1269-thinksystem-sr665-server

Memory Interleave

This setting allows interleaved memory accesses across multiple memory channels in each socket, providing higher memory bandwidth. Interleaving generally improves memory performance so the Auto (default) setting is recommended.

> **Note:** This setting is hidden in setup for 2nd Gen EPYC processor, only OneCLI/ASU variable and Redfish attribute are available.

This setting is accessed as follows:

- ► System setup: **System Settings** → **Memory** → **Interleave**
- ► OneCLI/ASU variable: `Memory.Interleave`
- ► Redfish attribute: `Memory_Interleave`

Possible values for Memory Interleave:

- ► **Auto** (default):

  Memory interleaving is automatically enabled if DIMM configuration supports it.

- ► **Disable**

Efficiency Mode

This setting enables an energy efficient mode of operation internal to the AMD EPYC Gen 2 and Gen 3 processors at the expense of performance. The setting should be enabled when energy efficient operation is desired from the processor. If desired, set it to Auto which disables the feature when maximum performance is desired.

This setting is accessed as follows:

- ► System setup:
  - – System **Settings** → **Operating Modes** → **Efficiency Mode**
  - – System **Settings** → **Power** → **Efficiency Mode**
- ► OneCLI/ASU variable: `Power.EfficiencyMode`
- ► Redfish: `Power_EfficiencyMode`

Possible values:

- ► **Disable**

  Use performance optimized CCLK DPM settings

► **Enable** (default)

Use power efficiency optimized CCLK DPM settings

xGMI settings

xGMI (Global Memory Interface) is the Socket SP3 processor socket-to-socket interconnection topology comprised of four x16 links. Each x16 link is comprised of 16 lanes. Each lane is comprised of two unidirectional differential signals.

Since xGMI is the interconnection between processor sockets, these xGMI settings are not applicable for ThinkSystem SR635 and SR635 which are one-socket platforms.

NUMA-unaware workloads may need maximum xGMI bandwidth/speed while other compute efficient NUMA-aware platforms may be able to minimize the xGMI speed and achieve adequate performance with power savings from the lower speed. The xGMI speed can be lowered, lane width can be reduced from x16 to x8 (or x2), or an xGMI link can be disabled if power consumption is too high.

4-Link xGMI Max Speed

The ThinkSystem SR645 and SR665 servers use 4-Link xGMI for maximizing socket-to-socket interconnection performance. Some other vendors' products may use 3-Link xGMI connection to provide 16 more PCIe I/O lanes.

This setting is used to set the xGMI speed. N is the maximum speed and is auto-calculated from the system board capabilities. For NUMA-aware workloads, users can lower the xGMI speed setting to reduce power consumption.

This setting is accessed as follows:

► System setup:

— **System Settings → Operating Modes → 4-Link xGMI Max Speed**
— **System Settings → Processors → 4-Link xGMI Max Speed**

► OneCLI/ASU variable: `Processors.4-LinkxGMIMaxSpeed`

► Redfish: `Processors_4_LinkxGMIMaxSpeed`

Possible values:

► **N**

N is the actual maximum supported speed and is auto-calculated based on the system board capabilities. The values in setup show the actual frequency. It is 18 GT/s on the SR645 and SR665 servers, but the choice is "18Gbps" for OneCLI/ASU because of a historical mistake.

► N-1

1 speed bin down from maximum speed. If the maximum supported speed is 18GT/s, then this "N-1" is 16GT/s. It is 16GT/s on the SR645 and SR665 servers. The choice is "16Gbps" for OneCLI/ASU.

► **N-2**

2 speed bin down from maximum speed. It is 13 GT/s on the SR645 and SR665 servers. The choice is "13Gbps" for OneCLI/ASU.

► **Minimum** (default)

Minimum will result in the xGMI speed getting set to the minimum value. It is equal to 10.667 GT/s on the SR645 and SR665 servers with 2nd Gen AMD EPYC processors and

13 GT/s when using 3rd Gen AMD EPYC processors. The choice is "Minimum" for OneCLI/ASU.

Possible Redfish value names are:

► `4-LinkxGMIMaxSpeed_18Gbps`
► `4-LinkxGMIMaxSpeed_16Gbps`
► `4-LinkxGMIMaxSpeed_13Gbps`
► `Minimum`

xGMI Maximum Link Width

Sets the xGMI width of each of the four links.

This setting is accessed as follows:

► System setup: **System Settings → Processors → xGMI Maximum Link Width**

► OneCLI/ASU variable: `Processors.xGMIMaximumLinkWidth`

► Redfish: `Processors_xGMIMaximumLinkWidth`

Possible values:

► **Auto** (default)

Auto sets maximum width based on the system capabilities. For the SR665 and SR645, the maximum link width is set to x16.

► **0**

Sets the maximum link width to x8. Redfish value name is `xGMIMaximumLinkWidth_0`.

► **1**

Sets the maximum link width to x16. Redfish value name is `xGMIMaximumLinkWidth_1`.

Global C-State Control

C-states are idle power saving states. This setting enables and disables C-states on the server across all cores. When disabled, the CPU cores can only be in C0 (active) or C1 state. C1 state can never be disabled. A CPU core is considered to be in C1 state if the core is halted by the operating system.

Lenovo generally recommends that Global C-State Control remain enabled, however consider disabling it for low-jitter use cases.

This setting is accessed as follows:

► System setup:

  – **System Settings → Operating Modes → Global C-state Control**
  – **System Settings → Processors → Global C-state Control**

► OneCLI/ASU variable: `Processors.GlobalC-stateControl`

► Redfish: `Processors_GlobalC_stateControl`

Possible values:

► **Disable**

I/O based C-state generation and Data Fabric (DF) C-states are disabled.

► **Enable** (default)

I/O based C-state generation and DF C-states are enabled.

SOC P-states

Infinity Fabric is a proprietary AMD bus that connects all the Core Cache Dies (CCDs) to the IO die inside the CPU as shown in Figure 5 on page 47. SOC P-states is the Infinity Fabric (Uncore) Power States setting. When Auto is selected the CPU SOC P-states will be dynamically adjusted. That is, their frequency will dynamically change based on the workload. Selecting P0, P1, P2, or P3 forces the SOC to a specific P-state frequency.

SOC P-states functions cooperatively with the Algorithm Performance Boost (APB) which allows the Infinity Fabric to select between a full-power and low-power fabric clock and memory clock based on fabric and memory usage. Latency sensitive traffic may be impacted by the transition from low power to full power. Setting APBDIS to 1 (to disable APB) and SOC P-states=0 sets the Infinity Fabric and memory controllers into full-power mode. This will eliminate the added latency and jitter caused by the fabric power transitions.

The following examples illustrate how SOC P-states and APBDIS function together:

► If SOC P-states=Auto then APBDIS=0 will be automatically set. The Infinity Fabric can select between a full-power and low-power fabric clock and memory clock based on fabric and memory usage.

► If SOC-P-states=<P0, P1, P2, P3> then APBDIS=1 will be automatically set. The Infinity Fabric and memory controllers are set in full-power mode.

► If SOC P-states=P0 which results in APBDIS=1, the Infinity Fabric and memory controllers are set in full-power mode. This results in the highest performing Infinity Fabric P-state with the lowest latency jitter.

This setting is accessed as follows:

► System setup:

— **System Settings → Operating Modes → SOC P-states**
— **System Settings → Processors → SOC P-states**

► OneCLI/ASU variable: `Processors.SOCP-states`

► Redfish: `Processors_SOCP_states`

Possible values:

► **Auto** (default)

When Auto is selected the CPU SOC P-states (uncore P-states) will be dynamically adjusted.

► **P0**: Highest-performing Infinity Fabric P-state

► **P1**: Next-highest-performing Infinity Fabric P-state

► **P2**: Next next-highest-performing Infinity Fabric P-state

► **P3**: Minimum Infinity Fabric power P-state

DF (Data Fabric) C-States

Much like CPU cores, the Infinity Fabric can go into lower power states while idle. However, there will be a delay changing back to full-power mode causing some latency jitter. In a low latency workload, or one with bursty I/O, one could disable this feature to achieve more performance with the tradeoff of higher power consumption.

This setting is accessed as follows:

- ▶ System setup:
  - — **System Settings** → **Operating Modes** → **DF C-States**
  - — **System Settings** → **Processors** → **DF C-States**
- ▶ OneCLI/ASU variable: `Processors.DFC-States`
- ▶ Redfish: `Processors_DFC_States`

Possible values:

- ▶ **Enable** (default)

  Enables Data Fabric C-states. Data Fabric C-states may be entered when all cores are in CC6.

- ▶ **Disable**

  Disable Data Fabric (DF) C-states

P-State 1

This setting enables or disable the CPU's P1 operating state.

This setting is accessed as follows:

- ▶ System setup:
  - — **System Settings** → **Operating Modes** → **P-State 1**
  - — **System Settings** → **Processors** → **P-State 1**
- ▶ OneCLI/ASU variable: `Processors.P-state1`
- ▶ Redfish: `Processors_P_state1`

Possible values:

- ▶ **Enable** (default)

  Enables CPU P1 P-state.

- ▶ **Disable**

  Disable CPU P1 P-state

P-State 2

This setting enables or disable the CPU's P2 operating state.

This setting is accessed as follows:

- ▶ System setup:
  - — **System Settings** → **Operating Modes** → **P-State 2**
  - — **System Settings** → **Processors** → **P-State 2**
- ▶ OneCLI/ASU variable: `Processors.P-state2`
- ▶ Redfish: `Processors_P_state2`

Possible values:

- ▶ **Enable** (default)

  Enables CPU P2 P-state.

▶ **Disable**

    Disable CPU P2 P-state.

Memory Power Down Enable

Low-power feature for DIMMs. Lenovo generally recommends that Memory Power Down remain enabled, however consider disabling it for low-latency use cases.

This setting is accessed as follows:

▶ System setup:
  – System Settings > Operating Modes > Memory Power Down Enable
  – System Settings > Memory > Memory Power Down Enable

▶ OneCLI/ASU variable: `Memory.MemoryPowerDownEnable`

▶ Redfish: `Memory_MemoryPowerDownEnable`

Possible values:

▶ **Enable** (default)

  Enables low-power features for DIMMs.

▶ **Disable**

NUMA nodes per socket

This setting lets you specify the number of desired NUMA nodes per socket. NPS0 will attempt to interleave the two sockets together into one NUMA node.

AMD EPYC 7002 and 7003 processors support a varying number of NUMA Nodes per Socket depending on the internal NUMA topology of the processor. In one-socket servers, the number of NUMA Nodes per socket can be 1, 2 or 4 though not all values are supported by every processor. See Table 7 on page 47 and Table 8 on page 48 NUMA nodes per socket (NPSx) options for the NUMA nodes per socket options available for each processor.

Applications that are highly NUMA optimized can improve performance by setting the number of NUMA Nodes per Socket to a supported value greater than 1.

This setting is accessed as follows:

▶ System setup:

  – **System Settings → Operating Modes → NUMA Nodes per Socket**
  – **System Settings → Memory → NUMA Nodes per Socket**

▶ OneCLI/ASU variable: `Memory.NUMANodesperSocket`

▶ Redfish: `Memory_NUMANodesperSocket`

Possible values:

▶ **NPS0**

  NPS0 will attempt to interleave the 2 CPU sockets together (non-NUMA mode).

▶ **NPS1** (default)
  – One NUMA node per socket.
  – Available for any CCD configuration in the SoC.
  – Preferred Interleaving: 8-channel interleaving using all channels in the socket.

- ► **NPS2**
  - – Two NUMA nodes per socket, one per Left/Right Half of the SoC.
  - – Requires symmetrical CCD configuration across left/right halves of the SoC.
  - – Preferred Interleaving: 4-channel interleaving using channels from each half.
- ► **NPS4**
  - – Four NUMA nodes per socket, one per Quadrant.
  - – Requires symmetrical Core Cache Die (CCD) configuration across Quadrants of the SoC.
  - – Preferred Interleaving: 2-channel interleaving using channels from each quadrant.

SMT Mode

Simultaneous multithreading (SMT) is similar to Intel Hyper-Threading Technology, the capability of a single core to execute multiple threads simultaneously. An OS will register an SMT-thread as a logical CPU and attempt to schedule instruction threads accordingly. All processor cache within a Core Complex (CCX) is shared between the physical core and its corresponding SMT-thread.

In general, enabling SMT benefits the performance of most applications. Certain operating systems and hypervisors, such as VMware ESXi, are able to schedule instructions such that both threads execute on the same core. SMT takes advantage of out-of-order execution, deeper execution pipelines and improved memory bandwidth in today's processors to be an effective way of getting all of the benefits of additional logical CPUs without having to supply the power necessary to drive a physical core.

Start with SMT enabled since SMT generally benefits the performance of most applications, however, consider disabling SMT in the following scenarios:

- ► Some workloads, including many HPC workloads, observe a performance neutral or even performance negative result when SMT is enabled.
- ► Using multiple execution threads per core requires resource sharing and is a possible source of inconsistent system response. As a result, disabling SMT could give benefit on some low-jitter use case.
- ► Some application license fees are based on the number of hardware threads enabled, not just the number of physical cores present. For this reason, disabling SMT on your EPYC 7002 and 7003 Series processor may be desirable to reduce license fees.
- ► Some older operating systems that have not enabled support for the x2APIC within the EPYC 7002 and 7003 Series processor, which is required to support beyond 255 threads. If you are running an operating system that does not support AM's x2APIC implementation, and have two 64-core processors installed, you will need to disable SMT.

  Operating systems such as Windows Server 2012 and Windows Server 2016 do not support x2APIC. Please refer to the following article for details:

  https://support.microsoft.com/en-in/help/4514607/windows-server-support-and-installation-instructions-for-amd-rome-proc

This setting is accessed as follows:

- ► System setup: **System Settings → Processors → SMT Mode**
- ► OneCLI/ASU variable: `Processors.SMTMode`
- ► Redfish: `Processors_SMTMode`

Possible values:

- **Disable**

  Disables simultaneous multithreading so that only one thread or CPU instruction stream is run on a physical CPU core

- **Enable** (default)

  Enables simultaneous multithreading.

Data Prefetchers

- L1 Stream Prefetcher: Uses history of memory access patterns to fetch next line into the L1 cache when cached lines are resued within a certain time period or access sequentially.

- L1 Stride Prefetcher: Uses memory access history to fetch additional data lines into L1 cache when each access is a constant distance from previous.

- L1 Region Prefetcher: Uses memory access history to fetche additional data line into L1 cache when the data access for a given instruction tends to be followed by a consistent pattern of subsequent access

- L2 Stream Prefetcher: Uses history of memory access patterns to fetch next line into the L1 cache when cached lines are resued within a certain time period or access sequentially.

- L2 Up/Down Prefetcher: Uses memory access history to determine whether to fetch the next or previous line for all memory accesses.

These fetchers use memory access history to determine whether to fetch the text or previous line for all memory access. Most workloads will benefit from these prefetchers gathering data and keeping the core pipeline busy. By default, these prefetchers all are enabled.

Application information access patterns, which tend to be relatively predictable, benefit greatly from prefetching. Most typical line-of-business, virtualization and scientific applications benefit from having pre-fetching enabled, however, there are however some workloads (for example, the SPECjbb 2015 Java application benchmark) that are very random in nature and will actually obtain better overall performance by disabling some of the prefetchers.

Further, the L1 and L2 stream hardware prefetchers can consume disproportionately more power vs. the gain in performance when enabled. Customers should therefore evaluate the benefit of prefetching vs. the non-linear increase in power if sensitive to energy consumption.

Note: L1 Stride Prefetcher, L1 Region Prefetcher and L2 Up/Down Prefetcher settings are only available for 3rd Gen EPYC processor.

This setting is accessed as follows:

- System setup:

  - **System Settings → Processors → L1 Stream HW Prefetcher**
  - **System Settings → Processors → L2 Stream HW Prefetcher**
  - **System Settings → Processors → L1 Stride Prefetcher**
  - **System Settings → Processors → L1 Region Prefetcher**
  - **System Settings → Processors → L2 Up/Down Prefetcher**

- ► OneCLI/ASU variables:
  - — `Processors.L1StreamHWPrefetcher`
  - — `Processors.L2StreamHWPrefetcher`
  - — `Processors.L1StridePrefetcher`
  - — `Processors.L1RegionPrefetcher`
  - — `Processors.L2UpDownPrefetcher`
- ► Redfish:
  - — `Processors_L1StreamHWPrefetcher`
  - — `Processors_L2StreamHWPrefetcher`
  - — `Processors_L1StridePrefetcher`
  - — `Processors_L1RegionPrefetcher`
  - — `Processors_L2UpDownPrefetcher`

Possible values:

- ► **Disable**

  Disable corresponding Prefetcher

- ► **Enable** (default)

  Enable corresponding Prefetcher

Preferred IO bus

AMD Preferred I/O complements Relaxed Ordering (RO) and ID-Based Ordering (IDO), rather than replace them. As with most performance tuning options, your exact workload may not perform better with Preferred I/O enabled but rather with RO or IDO. Lenovo recommends that you test your workload under all scenarios to find the most efficient setting.

If your PCIe device supports IDO or RO, and it is enabled, you can try enabling Preferred IO to determine whether your specific use case realizes an additional performance improvement. If the device does not support either of those features for your high bandwidth adapter, you can experiment with the Preferred IO mode capability available on the EPYC 7002 and 7003 Series processors. This setting allows devices on a single PCIe bus to obtain improved DMA write performance.

Note: The Preferred I/O can be enabled for only a single PCIe root complex (RC) and will affect all PCIe endpoints behind that RC. This setting gives priority to the I/O devices attached to the PCIe slot(s) associated with the enabled RC for I/O transactions.

This setting is accessed as follows:

- ► System setup: System **Settings** → **Processors** → **Preferred IO Bus**
- ► OneCLI/ASU variable:
  - — `Processors.PreferredIODevice`
  - — `Processors.PreferredIOBusNumber` (for specifying the bus number)
- ► Redfish:
  - — `Processors_PreferredIOBus`
  - — `Processors_PreferredIOBusNumber` (for specifying the bus number)

Possible values for Preferred IO Bus:

► **No Priority** (default)

All PCIe buses have equal priority. Use this setting unless you have an I/O device, such as a RAID controller or network adapter, that has a clear need to be prioritized over all other I/O devices installed in the server.

► **Preferred**

Select and then enter a preferred I/O bus number in the range 00–FF to specify the bus number for which device(s) you wish to enable preferred I/O. The number entered is hex.

If you use OneCLI (or Redfish), please set `Processors.PreferredIODevice` (or Redfish Attribute `Processors_PreferredIOBus`) to **Preferred**, then specify the bus number to `Processors.PreferredIOBusNumber` (or Redfish Attribute `Processors_PreferredIOBusNumber`).

ACPI SRAT L3 Cache as NUMA Domain

This parameter is named LLC as NUMA mode in the SR635 and SR655. When enabled, each Core Complex (CCX) in the system will become a separate NUMA domain. This setting can improve performance for highly NUMA optimized workloads if workloads or components of workloads can be pinned to cores in a CCX and if they can benefit from sharing an L3 cache. When disabled, NUMA domains will be identified according to the NUMA Nodes per Socket parameter setting.

This setting is accessed as follows:

► System setup: **System Settings → Processors → ACPI SRAT L3 Cache as NUMA Domain**

► OneCLI/ASU variable: `Processors.ACPISRATL3CacheasNUMADomain`

► Redfish: `Processors_ACPISRATL3CacheasNUMADomain`

Possible values:

► **Disable** (default)

When disabled, NUMA domains will be identified according to the NUMA Nodes per Socket parameter setting.

► **Enable**

When enabled, each Core Complex (CCX) in the system will become a separate NUMA domain.

PCIe (PCI Express) Gen Speed Selection

Choose the generation speed for available PCIe slots. Set the PCIe slot as Auto or generation 1, 2, 3 or 4. Note: Some adapters may not operate correctly in Gen2, Gen3 or Gen4.

Make sure to power off and power on the system for these settings to take effect.

This setting is accessed as follows:

► System setup: System **Settings → Devices and I/O Ports → PCIe Gen Speed Selection → Slot N** (or **NVMe Bay N**)

- ▶ OneCLI/ASU variable: `DevicesandIOPorts.PCIeGen_SlotN` (N is the slot number, for example `PCIeGen_Slot4`. If it is NVMe, then the name is `DevicesandIOPorts.PCIeGen_NVMeBayN`, where N is the bay number)
- ▶ Redfish: `DevicesandIOPorts_PCIeGen_SlotN` (`DevicesandIOPorts_PCIeGen_NVMeBayN` for NVMe Bay) ((N is the slot number, for example `DevicesandIOPorts_PCIeGen_Slot4`)

Possible values:

- ▶ **Auto** (default): Maximum PCIE speed by installed PCIE device support
- ▶ **Gen1**: 2.5 GT/s
- ▶ **Gen2**: 5.0 GT/s
- ▶ **Gen3**: 8.0 GT/s
- ▶ **Gen4**: 16.0 GT/s

CPPC

CPPC (Cooperative Processor Performance Control) was introduced with ACPI 5.0 as a mode to communicate performance between an operating system and the hardware. This mode can be used to allow the OS to control when and how much turbo can be applied in an effort to maintain energy efficiency. Not all operating systems support CPPC, but Microsoft began support with Windows Server 2016.

This setting is accessed as follows:

- ▶ System setup: **System Settings → Processors → CPPC**
- ▶ OneCLI/ASU variable: `Processors.CPPC`
- ▶ Redfish: `Processors_CPPC`

Possible values:

- ▶ **Enable** (default)
- ▶ **Disable**

BoostFmax

This value specifies the maximum boost frequency limit to apply to all cores. If the BoostFmax is set to something higher than the boost algorithms allow, the SoC will not go beyond the allowable frequency that the algorithms support.

This setting is accessed as follows:

System setup: **System Settings → Processors → BoostFmax**

- ▶ OneCLI/ASU variable:
  - — `Processors.BoostFmax`
  - — `Processors.BoostFmaxManual` (for specifying the frequency number)
- ▶ Redfish:
  - — `Processors_BoostFmax`
  - — `Processors_BoostFmaxManual` (for specifying the frequency number)

Possible values:

► **Manual**

A 4 digit number representing the maximum boost frequency in MHz.

If you use OneCLI (or Redfish), please set `Processors.BoostFmax` (or Redfish Attribute `Processors_BoostFmax`) to **Manual**, then specify the maximum boost frequency number in MHz to `Processors.BoostFmaxManual` (or Redfish Attribute `Processors_BoostFmaxManual`).

► **Auto** (default)

Auto set the boost frequency to the fused value for the installed CPU.

DRAM Scrub Time

Memory reliability parameter that sets the period of time between successive DRAM scrub events. Performance may be reduced with more frequent DRAM scrub events.

This setting is accessed as follows:

► System setup: **System Settings → Memory → DRAM Scrub Time**

► OneCLI/ASU variable: `Memory.DRAMScrubTime`

► Redfish: `Memory_DRAMScrubTime`

Possible values:

► **Disable**

► **1 hour**

Redfish value name is `DRAMScrubTime_1Hour`.

► **4 hour**

Redfish value name is `DRAMScrubTime_4Hour`.

► **8 hour**

Redfish value name is `DRAMScrubTime_8Hour`.

► **16 hour**

Redfish value name is `DRAMScrubTime_16Hour`.

► **24 hour** (default)

Redfish value name is `DRAMScrubTime_24Hour`.

► **48 hour**

Redfish value name is `DRAMScrubTime_48Hour`.

Number of CPU Cores Activated (Downcore)

UEFI allows the administrator to shut down cores in a server. This setting powers off a set number of cores for each processor in a system. As opposed to restricting the number of logical processors an OS will run on, this setting directly affects the number of cores powered on by turning off the core level power gates on each processor.

Manipulating the number of physically powered cores is primarily used in three scenarios:

Where users have a licensing model that supports a certain number of active cores in a system

Where users have poorly threaded applications but require the additional LLC available to additional processors, but not the core overhead

Where users are looking to limit the number of active cores in an attempt to reclaim power and thermal overhead to increase the probability of Performance Boost being engaged.

This setting is accessed as follows:

- ► System setup: **System Settings → Processors → Number of CPU Cores Activated**
- ► OneCLI/ASU variable: Processors.CPUCoresActivated
- ► Redfish: `Processors_CPUCoresActivated`

Possible values:

- ► **All** (default):

  Enable all cores

- ► **2 Cores Per Die**:

  Enable 2 cores for each Core Cache Die. Redfish value name is CPUCoresActivated_2CoresPerDie.

- ► **4 Cores Per Die**:

  Enable 4 cores for each Core Cache Die. Redfish value name is CPUCoresActivated_4CoresPerDie.

- ► **6 Cores Per Die**:

  Enable 6 cores for each Core Cache Die. Redfish value name is CPUCoresActivated_6CoresPerDie.

The second and third-generation AMD EPYC processors have a multi-die topology consisting of two or more Core Cache Die (CCD). Each CCD contains processor cores plus Level 2 and Level 3 caches. See Figure 5 on page 47 and Table 7 on page 47 and Table 8 on page 48 for more information on CCDs.

Hidden UEFI Items for SR645 and SR665

The UEFI items in this section are more limited in their applicability to customer use cases and are not exposed in UEFI menus. However, they can be accessed using the command line utilities such as Lenovo's Advanced Settings Utility (ASU) or OneCLI.

- ► "DLWM (Dynamic xGMI Link Width Management) Support"
- ► "PCIe Ten Bit Tag Support" on page 42

DLWM (Dynamic xGMI Link Width Management) Support

DLWM saves power during periods of low socket-to-socket data traffic by reducing the number of active xGMI lanes per link from 16, 8 to 2. Disabling this parameter can achieve an effect similar to setting "xGMI Force Link Width Control" to Force.

This setting is accessed as follows:

- ► OneCLI/ASU variable: Processors.DLWMSupport
- ► Redfish attribute: Processors_DLWMSupport

Possible values for DLWM Support:

► Auto (default):

  DLWM is enabled when two CPU are installed.

► Disable:

  DLWM is disabled, xGMI link width is fixed.

PCIe Ten Bit Tag Support

This setting enables the PCIe Ten Bit Tag which is optionally supported by PCIe device since Gen4. Enable Ten Bit Tag to increase the number of non-posted request from 256 to 768 for better performance. As latency increases, the increase in unique tags is required to maintain the peak performance at 16GT/s.

This setting is accessed as follows:

► OneCLI/ASU variable: `DevicesandIOPorts.PCIeTenBitTagSupport`

► Redfish attribute: `DevicesandIOPorts_PCIeTenBitTagSupport`

Possible values for PCIe Ten Bit Tag:

► **Auto** (default)

  The Auto setting maps to Enable.

► **Enable**

► **Disable**

Low Latency and Low Jitter UEFI parameter settings

The tables in this section show the recommended settings when tuning for either Low Latency or Low Jitter.

► The Low Latency settings should be used when a workload or application relies on the lowest possible local/remote memory, storage, and/or PCIe adapter latency.

► The Low Jitter settings should be used when minimal run-to-run and system-to-system variation is desired, i.e. more consistent performance.

Note that peak performance may be impacted to achieve lower latency or more consistent performance.

> **Tip:** Prior to optimizing a workload for Low Latency or Low Jitter, it is recommended you first set the Operating Mode to "Maximum Performance", save settings, then reboot rather than simply starting from the Maximum Efficiency default mode and then modifying individual UEFI parameters. If you don't do this, some settings may be unavailable for configuration.

Table 5   UEFI Settings for Low Latency and Low Jitter for SR635 and SR655

UEFI Setting	Page	Category	Low Latency	Low Jitter
Determinism Slider	10	Recommended	Power	Performance

UEFI Setting	Page	Category	Low Latency	Low Jitter
Core Performance Boost	10	Recommended	Enable	Enable
cTDP Control	11	Recommended	Manual; Set cTDP to 300W	Manual; Set cTDP to 300W
Memory Speed	12	Recommended	2933MHz for EPYC Gen 2 3200MHz for EPYC Gen 3	2933MHz for EPYC Gen 2 3200MHz for EPYC Gen 3
L1 Stream HW Prefetcher	13	Recommended	Auto	Auto
L2 Stream HW Prefetcher	13	Recommended	Auto	Auto
Global C-state Control	13	Recommended	Enable	Disable
SMT Mode	14	Test (For Low Jitter, disabling SMT Mode can adversely reduce performance)	Enable	Disable
Package Power Limit	15	Recommended	Manual; Set PPL to 300W	Manual; Set PPL to 300W
Memory Interleaving	15	Suggested	Auto	Auto
NUMA Nodes Per Socket	16	Test	NPS=4; Note that available NPS options vary depending on processor SKU. Set to highest available NPS setting.	Auto (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads)
EfficiencyModeEn	17	Recommended	Auto	Auto
Chipselect Interleaving	17	Suggested	Auto	Auto
LLC as NUMA Node	17	Test	Auto (Optionally experiment with Enable if application threads can be pinned to a NUMA node and can share an L3 cache)	Auto (Optionally experiment with Enable if application threads can be pinned to a NUMA node and can share an L3 cache)
SOC P-States	18	Recommended	Auto	Auto
P-State 1	19	Recommended	Enabled	Disabled
P-State 2	19	Recommended	Enabled	Disabled
DF C-states	19	Recommended	Enabled	Disabled
Memory Power Down Enable	20	Recommended	Enabled	Disabled

Table 6   UEFI Settings for Low Latency and Low Jitter for SR645 and SR665

Menu Item	Page	Category	Low Latency	Low Jitter
Determinism Slider	25	Recommended	Power	Performance
Core Performance Boost	26	Recommended	Enable	Enable
cTDP	26	Recommended	Maximum	Maximum
Package Power Limit	27	Recommended	Maximum	Maximum
Memory Speed	28	Recommended	2933MHz for EPYC Gen 2 3200MHz for EPYC Gen 3	2933MHz for EPYC Gen 2 3200MHz for EPYC Gen 3
Efficiency Mode	29	Recommended	Disable	Disable
4-Link xGMI Max Speed	30	Recommended	18Gbps	18Gbps
Global C-state Control	31	Recommended	Enable	Disable
SOC P-states	32	Recommended	P0	P0
DF C-States	32	Recommended	Enable	Disable
P-State 1	33	Recommended	Enable	Disable
P-State 2	33	Recommended	Enable	Disable
Memory Power Down Enable	34	Recommended	Disable	Disable
NUMA Nodes per Socket	34	Test	NPS=4; Note that available NPS options vary depending on processor SKU. Set to highest available NPS setting.	NPS1 (Optionally experiment with NPS=2 or NPS=4 for NUMA optimized workloads)
ACPI SRAT L3 Cache as NUMA Domain	38	Test	Disable (Optionally experiment with Enable if application threads can be pinned to a NUMA node and can share an L3 cache)	Disable (Optionally experiment with Enable if application threads can be pinned to a NUMA node and can share an L3 cache)
SMT Mode	35	Test (For Low Jitter, disabling SMT Mode can adversely reduce performance)	Enable	Disable
CPPC	39	Recommended	Enable	Disable
xGMI Maximum Link Width	30	Recommended	Auto	1
DRAMScrubTime	40	Recommended	24 Hour	Disable
Memory Interleave	29	Recommended	Auto	Auto

ThinkSystem server platform design

The following figures show the block diagrams of the 2nd and 3rd Gen AMD EPYC processor-based ThinkSystem servers.

The architecture of the ThinkSystem SR635 is shown in Figure 1.

Figure 1   ThinkSystem SR635 (1U server) block diagram

The architecture of the ThinkSystem SR655 is shown in Figure 2.

Figure 2   ThinkSystem SR655 (2U server) block diagram

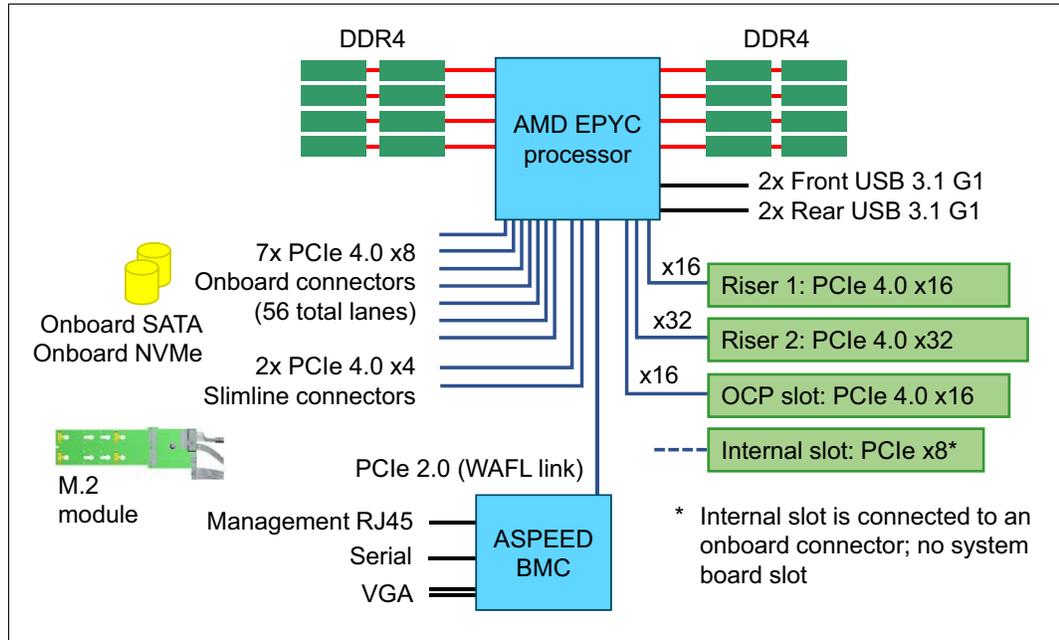The architecture of the ThinkSystem SR645 is shown in Figure 3.

Figure 3   ThinkSystem SR645 (1U server) block diagram

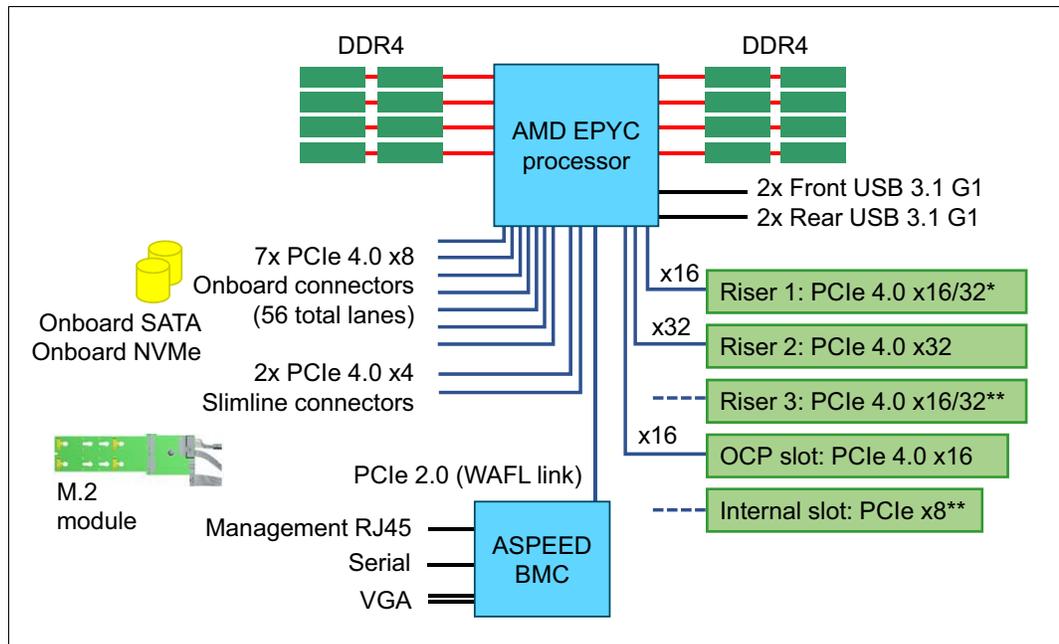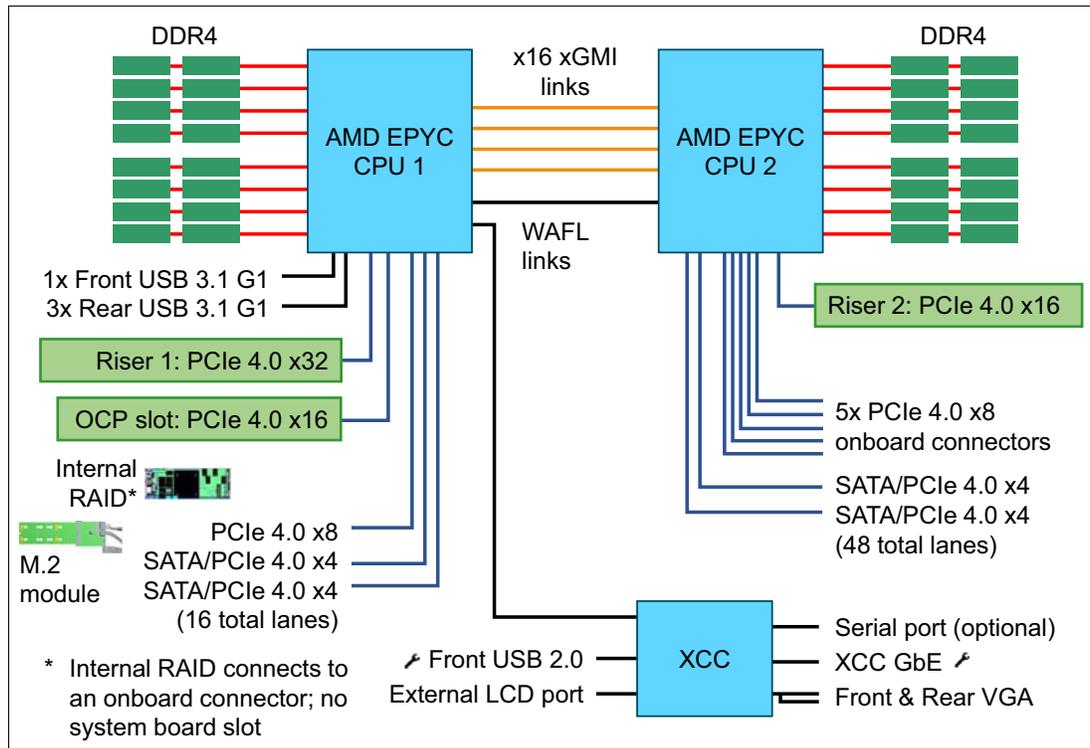The architecture of the ThinkSystem SR665 is shown in Figure 4.

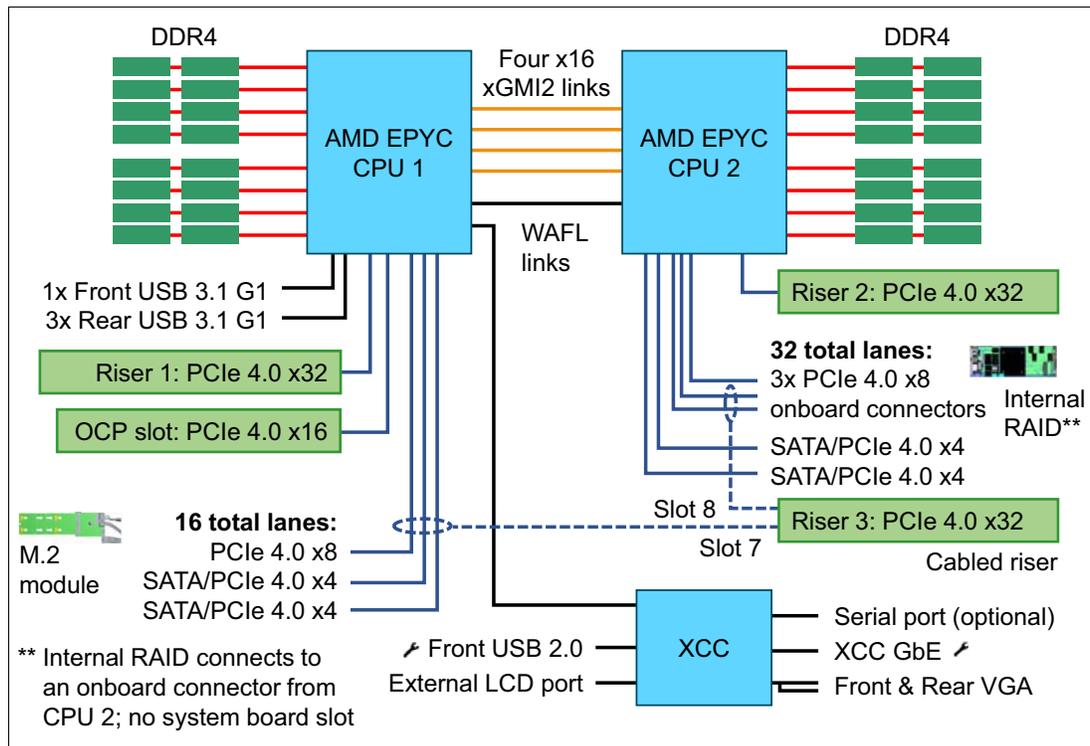*Figure 4   ThinkSystem SR665 (2U server) block diagram*

Figure 5 shows the architecture of the second-generation AMD EPYC processor codenamed "Rome".
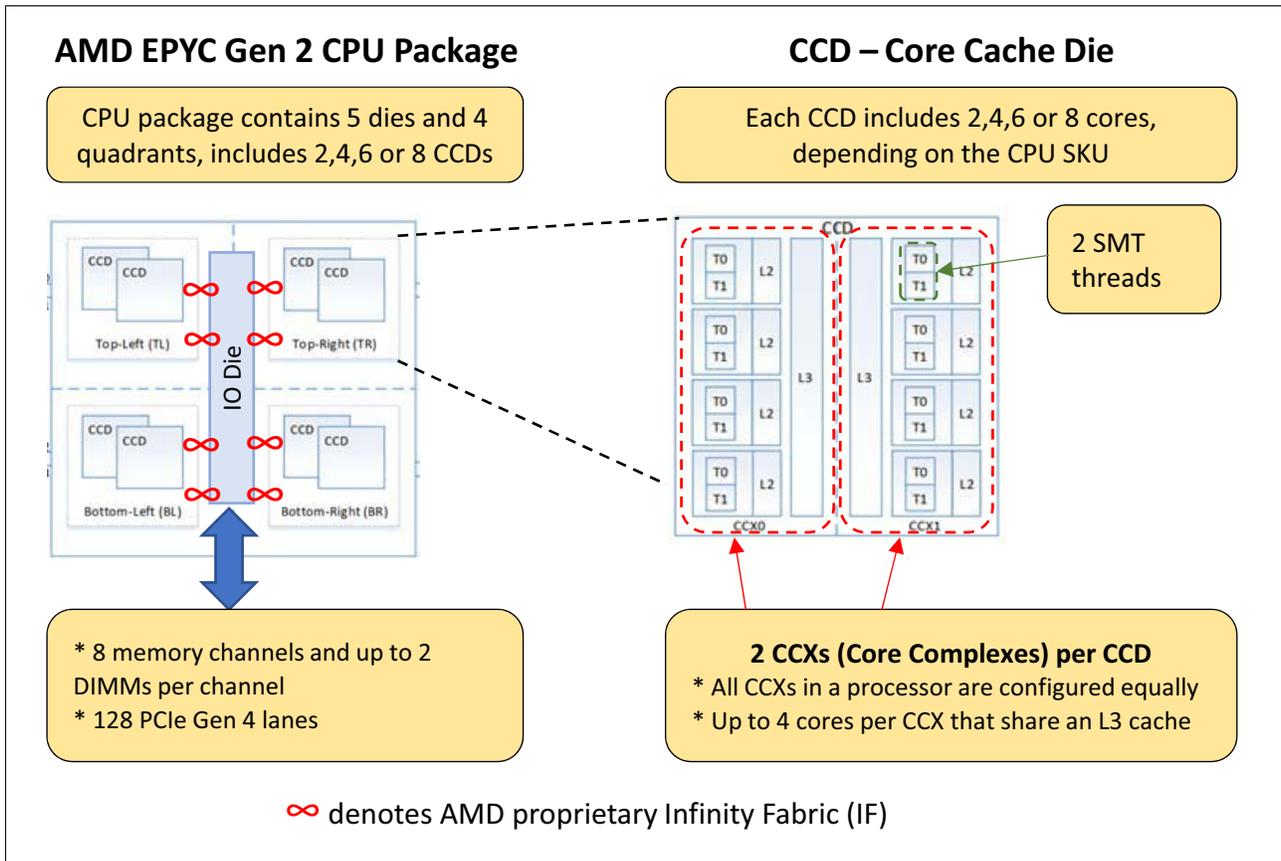

Figure 5   Illustration of the 2nd Gen AMD EPYC "Rome" Processor CCD and CCX

The difference in the 3rd Gen AMD EPYC "Milan" processor is that each CCD only has one CCX and each CCX has up to 8 cores that share an L3 cache. That means a CCD is equal to a CCX in EPYC Gen 3 processors.

Table 7 shows the architectural geometry of the second-generation AMD EPYC processor including the core cache die (CCD), core complex (CCX) and cores per CCX for each processor. The NUMA Nodes per Socket (NPSx) options for each processor are also listed.

Table 7   NUMA nodes per socket (NPSx) options for EPYC 7002 series CPU SKUs

Cores	CCDs x CCXs x cores/CCX	NPSx Options (1P)	NPS x Options (2P)	EPYC Gen 2 CPU SKUs[a]
64	8 x 2 x 4	4, 2, 1	4, 2, 1, 0	7H12, 7742, 7702, 7702P, 7662
48	8 x 2 x 3	2, 1	2, 1, 0	7642
48	6 x 2 x 4	2, 1	2, 1, 0	7552
32	4 x 2 x 4	4, 2, 1	4, 2, 1, 0	7542, 7502, 7502P, 7452
32	8 x 2 x 2	2, 1	2, 1, 0	7532
24	6 x 2 x 2	2, 1	2, 1, 0	7F72
24	4 x 2 x 3	4, 2, 1	4, 2, 1, 0	7352, 7402, 7402P
16	8 x 2 x 2	4, 2, 1	4, 2, 1, 0	7F52

Cores	CCDs x CCXs x cores/CCX	NPSx Options (1P)	NPS x Options (2P)	EPYC Gen 2 CPU SKUs[a]
16	4 x 2 x 2	4, 2, 1	4, 2, 1, 0	7302, 7302P
16	2 x 2 x 4	1	1, 0	7282
12	2 x 2 x 3	1	1, 0	7272
8	2 x 2 x 2	1	1, 0	7232P, 7252
8	4 x 2 x 1	4, 2, 1	4, 2, 1, 0	7262, 7F32

a. Consult the relevant ThinkSystem server product guides for specific AMD EPYC Gen 2 CPU SKUs that are supported.

Table 8 shows the architectural geometry of the third-generation AMD EPYC processor.

Table 8   NUMA nodes per socket (NPSx) options for EPYC Gen 3 CPU SKUs

Cores	CCDs x CCXs/CCD x cores/CCX	NPSx Options – 1P	NPSx Options – 2P	EPYC Gen 3 CPU SKUs
64	8 x 1 x 8	4, 2, 1	4, 2, 1, 0	7763, 7713, 7713P
56	8 x 1 x 7	4, 2, 1	4, 2, 1, 0	7663
48	8 x 1 x 6	4, 2, 1	4, 2, 1, 0	7643
32	8 x 1 x 4	4, 2, 1	4, 2, 1, 0	75F3, 7543, 7543P,
32	4 x 1 x 8	4, 2, 1	4, 2, 1, 0	7513
28	4 x 1 x 7	4, 2, 1	4, 2, 1, 0	7453
24	8 x 1 x 3	4, 2, 1	4, 2, 1, 0	74F3
24	4 x 1 x 6	4, 2, 1	4, 2, 1, 0	7443, 7443P, 7413
16	8 x 1 x 2	4, 2, 1	4, 2, 1, 0	73F3
16	4 x 1 x 4	4, 2, 1	4, 2, 1, 0	7343, 7313, 7313P
8	8 x 1 x 1	4, 2, 1	4, 2, 1, 0	72F3

Table 9 shows EPYC 7002 series CPU allowed maximum and minimum configurable TDP values.

Note: If the cTDP setting is set outside the limits that are supported by the installed CPU SKU, the cTDP value will automatically be limited to the minimum or maximum supported value.

Table 9   AMD EPYC 7002 Series processor cTDP range table

Model	2P/1P	Production OPN	Default TDP	Min cTDP	Max cTDP
7H12	2P/1P	100-000000055	280 W	225 W	280 W
7742	2P/1P	100-000000053	225 W	225 W	240 W
7702	2P/1P	100-000000038	200 W	165 W	200 W
7702P	1P	100-000000047	200 W	165 W	200 W
7662	2P/1P	100-000000137	225 W	225 W	240 W
7642	2P/1P	100-000000074	225 W	225 W	240 W

Model	2P/1P	Production OPN	Default TDP	Min cTDP	Max cTDP
7552	2P/1P	100-000000076	200 W	165 W	200 W
7542	2P/1P	100-000000075	225 W	225 W	240 W
7532	2P/1P	100-000000136	200 W	180 W	200 W
7502	2P/1P	100-000000054	180 W	165 W	200 W
7502P	1P	100-000000045	180 W	165 W	200 W
7452	2P/1P	100-000000057	155 W	155 W	180 W
7F72	2P/1P	100-000000141	240 W	225 W	240 W
7402	2P/1P	100-000000046	180 W	165 W	200 W
7402P	1P	100-000000048	180 W	165 W	200 W
7352	2P/1P	100-000000077	155 W	155 W	180 W
7F52	2P/1P	100-000000140	240 W	225 W	240 W
7302	2P/1P	100-000000043	155 W	155 W	180 W
7302P	1P	100-000000049	155 W	155 W	180 W
7282	2P/1P	100-000000078	120 W	120 W	150 W
7272	2P/1P	100-000000079	120 W	120 W	150 W
7F32	2P/1P	100-000000139	180 W	165 W	200 W
7262	2P/1P	100-000000041	155 W	155 W	180 W
7252	2P/1P	100-000000080	120 W	120 W	150 W
7232P	1P	100-000000081	120 W	120 W	150 W

Table 10 shows EPYC 7003 series CPU allowed maximum and minimum configurable TDP values.

Table 10   AMD EPYC 7003 Series processor cTDP range table

Model	2P/1P	Production OPN	Default TDP	Min cTDP	Max cTDP
7763	2P/1P	100-000000312	280 W	225 W	280 W
7713	2P/1P	100-000000344	225 W	225 W	240 W
7713P	1P	100-000000337	225 W	225 W	240 W
7663	2P/1P	100-000000318	240 W	225 W	240 W
7643	2P/1P	100-000000326	225 W	225 W	240 W
75F3	2P/1P	100-000000313	280 W	225 W	280 W
7543	2P/1P	100-000000345	225 W	225 W	240 W
7543P	1P	100-000000341	225 W	225 W	240 W
7513	2P/1P	100-000000334	200 W	165 W	200 W
7453	2P/1P	100-000000319	225 W	225 W	240 W
74F3	2P/1P	100-000000317	240 W	225 W	240 W

Model	2P/1P	Production OPN	Default TDP	Min cTDP	Max cTDP
7443	2P/1P	100-000000340	200 W	165 W	200 W
7443P	1P	100-000000342	200 W	165 W	200 W
7413	2P/1P	100-000000323	180 W	165 W	200 W
73F3	2P/1P	100-000000321	240 W	225 W	240 W
7343	2P/1P	100-000000338	190 W	165 W	200 W
7313	2P/1P	100-000000329	155 W	155 W	180 W
7313P	1P	100-000000339	155 W	155 W	180 W
72F3	2P/1P	100-000000327	180 W	165 W	200 W

References

► ThinkSystem SR635 product guide

  https://lenovopress.com/lp1160-thinksystem-sr635-server

► ThinkSystem SR655 product guide

  https://lenovopress.com/lp1161-thinksystem-sr655-server

► ThinkSystem SR645 product guide

  https://lenovopress.com/lp1280-thinksystem-sr645-server

► ThinkSystem SR665 product guide

  https://lenovopress.com/lp1269-thinksystem-sr665-server

► AMD white paper: *Power / Performance Determinism — Maximizing performance while driving better flexibility & choice*:

  https://www.amd.com/system/files/2017-06/Power-Performance-Determinism.pdf

► Lenovo performance paper: Balanced Memory Configurations with Second-Generation AMD EPYC Processors

  https://lenovopress.com/lp1268-balanced-memory-configurations-with-second-generation-amd-epyc-processors

► WikiChip Chip & Semi - Users can get detailed information on any CPU family or SKU

  https://en.wikichip.org/wiki/amd

► Microsoft Powercfg command-line options and usage

  https://datacentersupport.lenovo.com/us/en/solutions/ht514546

  In addition to the UEFI settings described in this document, the command line options offered by the Microsoft powercfg program can also be used to tune the operating system

Change History

January 2023:

► Added a link to a Lenovo support tip on Microsoft Powercfg command-line options — "References" on page 50

June 2022:

► Updated the Data Prefetcher sections

– SR635 & SR655: "Data Prefetchers" on page 22
– SR645 & SR665: "Data Prefetchers" on page 36

► Added "PCIe Ten Bit Tag Support" on page 42

June 2021:

► Added tuning information for AMD EPYC 7003 Series processors

► Added hidden setup options which are used for performance and power saving tuning

January 2021:

► Added section "Low Latency and Low Jitter UEFI parameter settings" on page 42

► Added section "How to use OneCLI and Redfish to access these settings" on page 7

► Added Redfish parameters for each setting in "UEFI menu items for SR645 and SR665" on page 24

► Updated the default for setting "NUMA nodes per socket" on page 16

August 2020:

► Now includes settings for SR645 and SR665 (AMD 2S servers)

Authors

This paper was produced by the following team of specialists:

John Encizo is a Senior Technical Sales Engineer on the Brand Technical Sales team for Lenovo across the Americas region. He is part of the field engineering team responsible for pre-GA support activities on the ThinkSystem server portfolio and works closely with customers and Lenovo development on system design, BIOS tuning, and systems management toolkits. He has also been the lead field engineer for high-end server product launches. When not in the field doing technical design and onboarding sessions with customers or pre-GA support activities, he works in the BTS lab to do remote customer proof-of-technology engagements, consult on systems performance and develop documentation for the field technical sellers.

Kin Huang is a Principal Engineer for firmware in Lenovo Infrastructure Solutions Group in Beijing. He has 2 years of experience in software application programming, and 17 years of experience in x86 server, focusing on BIOS/UEFI, management firmware and software. His current role includes firmware architecture for AMD platforms, boot performance optimization, RAS (Reliability, Availability and Serviceability).

Joe Jakubowski is the Principal Engineer and Performance Architect in the Lenovo ISG Performance Laboratory in Morrisville, NC. Previously, he spent 30 years at IBM. He started his career in the IBM Networking Hardware Division test organization and worked on various

token-ring adapters, switches and test tool development projects. He has spent the past 26 years in the x86 server performance organization focusing primarily on database, virtualization and new technology performance. His current role includes all aspects of Intel and AMD x86 server architecture and performance. Joe holds Bachelor of Science degrees with Distinction in Electrical Engineering and Engineering Operations from North Carolina State University and a Master of Science degree in Telecommunications from Pace University.

Robert R. Wolford is the Principal Engineer for power management and efficiency at Lenovo. He covers all technical aspects that are associated with power metering, management, and efficiency. Previously, he was a lead systems engineer for workstation products, video subsystems building block owner for System x®, System p, and desktops, signal quality and timing analysis engineer, gate array design, and product engineering. In addition, he was a Technical Sales Specialist for System x. Bob has several issued patents centered around computing technology. He holds a Bachelor of Science degree in Electrical Engineering with Distinction from the Pennsylvania State University.

Thanks to the following people for their contributions to this project:

- ► Helen Lan1 He
- ► Jack MJ1 He
- ► David Watts
- ► Mengjiao MJ12 Li
- ► Jamal Ayoubi
- ► Sampson Chien
- ► Arun Krishnamoorthy

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on January 12, 2023.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp1267

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. These and other Lenovo trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by Lenovo at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of Lenovo trademarks is available from https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Lenovo®   System x®
Lenovo(logo)®   ThinkSystem™

The following terms are trademarks of other companies:

Intel, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Other company, product, or service names may be trademarks or service marks of others.