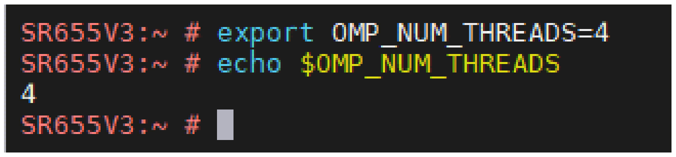
Tuning SPEComp2012 Performance on Lenovo ThinkSystem AMD Servers

Planning / Implementation

Introduction to OpenMP

OpenMP (Open Multi-Processing) is an application programming interface (API) that supports multi-platform shared-memory multiprocessing programming in C, C++, and Fortran. It consists of a set of compiler directives, library routines, and environment variables that influence run-time behavior. OpenMP use a portable, scalable model that gives programmers a simple and flexible interface for developing parallel applications for platforms ranging from the standard desktop computer to the supercomputer.

To setup the number of threads used for the openmp program, the value of openmp environment variable OMP_NUM_THREADS can be modified and verified under Linux Bash shell environment as shown below.

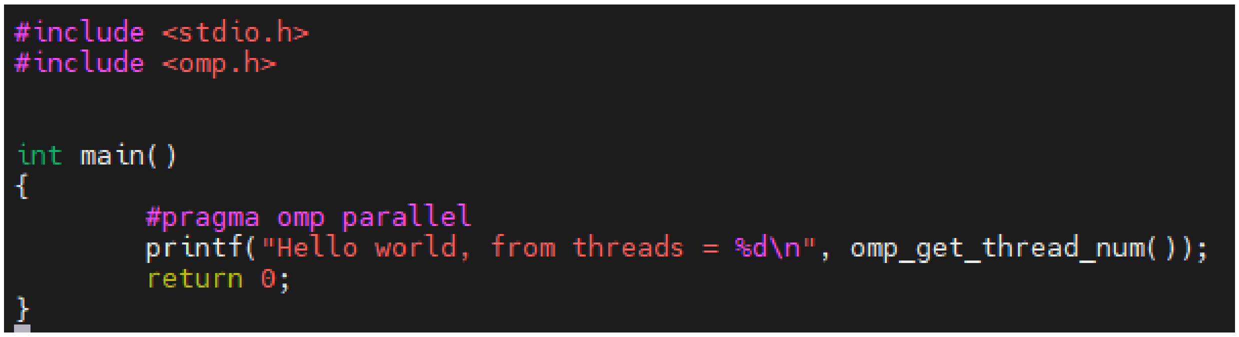


```
SR655V3:~ # export OMP_NUM_THREADS=4
SR655V3:~ # echo $OMP_NUM_THREADS
4
SR655V3:~ #
```

Figure 1. Set the number of threads for parallel

In OpenMP programming, we need to specify the region which we are going to define as parallel using the keywords `pragma omp parallel`. The keywords are used to fork additional threads to carry out the work enclosed in the construct in parallel. The original thread will be denoted as master thread with thread ID 0.

The `omp_get_thread_num()` function in the following C based program will display “Hello, world” multiple time with threads ID according to user defined value of OMP_NUM_THREADS environment variable.



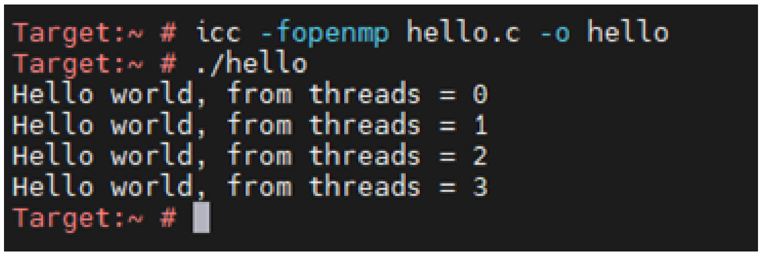
```
#include <stdio.h>
#include <omp.h>


int main()
{
        #pragma omp parallel
        printf("Hello world, from threads = %d\n", omp_get_thread_num());
        return 0;
}
```

Figure 2. hello.c

Compiler the C program with Intel oneAPI compiler’s option “ *-fopenmp*” to generate a parallel execution binary file:

```
$ icc -fopenmp hello.c -o hello
```

Set to 4 threads by command `export OMP_NUM_THREADS=4`, the program will print out 4 times “Hello, world” with thread ID from 0 to 3 as shown in the figure below:



```
Target:~ # icc -fopenmp hello.c -o hello
Target:~ # ./hello
Hello world, from threads = 0
Hello world, from threads = 1
Hello world, from threads = 2
Hello world, from threads = 3
Target:~ #
```

Figure 3. Execution result

Introduction to SPEComp2012

SPEComp2012 contains a suite that focuses on parallel computing performance using the OpenMP 3.1 parallelism standard. SPEC OMP is not intended to stress other computer components such as networking, the operating system, graphics, or the I/O system. SPEComp2012 focuses on compute intensive performance, which means these benchmarks emphasize the performance of:

- The computer processor (CPU)
- The memory architecture
- The parallel support libraries
- The compilers

SPEComp2012 is based on compute-intensive applications provided as source code, it contains 14 benchmarks: 8 use Fortran, 5 use C, and 1 use C++.

Table 1. Details of SPEComp2012 Benchmarks

Benchmark	**Language**	**Description**
350.md	Fortran	Physics: Molecular Dynamics
351.bwaves	Fortran	Physics: Computational Fluid Dynamics (CFD)
352.nab	C	Molecular Modeling
357.bt331	Fortran	Physics: Computational Fluid Dynamics (CFD)
358.botsalgn	C	Protein Alignment
359.botsspar	C	Sparse LU
360.ilbdc	Fortran	Lattic Boltzmann
362.fma3d	Fortran	Mechanical Response Simulation
363.swim	Fortran	Weather Prediction
367.imagick	C	Image Processing
370.mgrid331	Fortran	Physics: Computation Fluid Dynamics (CFD)
371.applu331	Fortran	Physics: Computation Fluid Dynamics (CFD)
372.smithwa	C	Optimal Pattern Matching
376.kdtree	C++	Sorting and Searching

The SPEComp2012 benchmark uses base and peak metrics to evaluate the performance of a server. The base metric is the geometric mean of medians of the base ratios, and the peak metric is the geometric mean of median of the peak ratios.

- The **base** metrics are required for all reported results and have stricter guidelines for compilation. For example, the same flags must be used in the same order for all benchmarks of a given language. This is the point closer to those who might prefer a relatively simple build process.
- The **peak** metrics are optional and have less strict requirements. For example, different compiler options may be used on each benchmark, and feedback-directed optimization is allowed. This point is closer to those who may be willing to invest more time and effort in development of build procedures.

The following table is an example of SPECompG_base2012.

Table 2. SPEComp2012 results example

Benchmark	**Base # Threads**	**Base Run Time**	**Base Ratio**	**Peak # Threads**	**Peak Run Time**	**Peak Ratio**
350.md	192	55.4	83.6	192.0	54.8	84.4
351.bwaves	192	132.3	34.2	96.0	131.4	34.5
352.nab	192	86.0	45.2	192.0	86.3	45.1
357.bt331	192	84.7	56.0	192.0	84.2	56.3
358.botsalgn	192	99.0	44.0	192.0	98.7	44.1
359.botsspar	192	166.9	31.5	192.0	167.0	31.4
360.ilbdc	192	131.5	27.1	96.0	131.2	27.1
362.fma3d	192	115.5	32.9	192.0	115.4	32.9
363.swim	192	171.4	26.4	96.0	169.5	26.7
367.imagick	192	87.1	80.7	96.0	79.5	88.4
370.mgrid331	192	185.6	23.8	96.0	164.9	26.8
371.applu331	192	59.9	101.1	192.0	59.8	101.3
372.smithwa	192	48.0	111.7	192.0	47.9	111.9
376.kdtree	192	77.5	58.0	192.0	77.3	58.2
SPECompG_base2012			**47.5**			
SPECompG_peak2012						**48.4**

Notes about the table:

1. For the given OMP2012 suite, the elapsed time in seconds for each of its benchmark runs is reported.
2. The ratio of the reference system (Sun Fire X4140) time divided by the corresponding measured time is reported.
3. Separately for base and peak, the median of three runs of these ratios is reported per benchmark.

In all cases, a higher ratio means “better performance” on the given workload.

ThinkSystem SR655 V3

We used the Lenovo ThinkSystem SR655 V3 for our testing in the lab. The Lenovo ThinkSystem SR655 V3 is a 1-socket 2U server that features the AMD EPYC 9004 "Genoa" family of processors. With up to 96 cores per processor and support for the new PCIe 5.0 standard for I/O, the SR655 V3 offers the ultimate in one-socket server performance in a 2U form factor. The server also supports for DDR5 memory DIMMs to maximize performance of the memory subsystem, includes 12 memory channels (1 DIMM per channel), DIMM speeds up to 4800 MHz.

Figure 4. Lenovo ThinkSystem SR655 V3

For information about the SR655 V3, see the Lenovo Press product guide:
https://lenovopress.lenovo.com/lp1610-thinksystem-sr655-v3-server

Our configuration to measure SPEComp2012 in our lab is as follows.

Table 3. Lab configuration

System Configuration	**Lab configuration**
CPU	1x AMD EPYC 9654P processor, 96 cores, 2.4GHz
Memory	12x 64GB 2Rx4 DDR5 4800MHz
Disk	1x 960 GB SATA SSD
OS	SLES 15.4

SPEComp2012 performance tunning

To obtain the SPEComp2012 best performance recipe for the 4th Gen AMD EPYC processor, we examined the hardware, firmware, and software components of ThinkSystem SR655 V3 server.

Topics in this section:

- Processors
- Memory
- UEFI configuration
- Linux utilities
- Compiler flags
- OpenMP environment variables

Processors

Designed for parallel computing, the performance of SPEComp2012 benefit from CPU cores and hardware threads. The SMT (Simultaneously Multithreading) of AMD EPYC CPU provide one more hardware thread for each physical cores, which can support more OpenMP threads runs on the system to improve the performance.

The following chart shows the SPECcomp2012 scaling result from 1 to 96 cores with SMT enabled. The baseline result was measure by one core, two threads (1C2T), the performance scale up to 2424.95% on 96C192T.

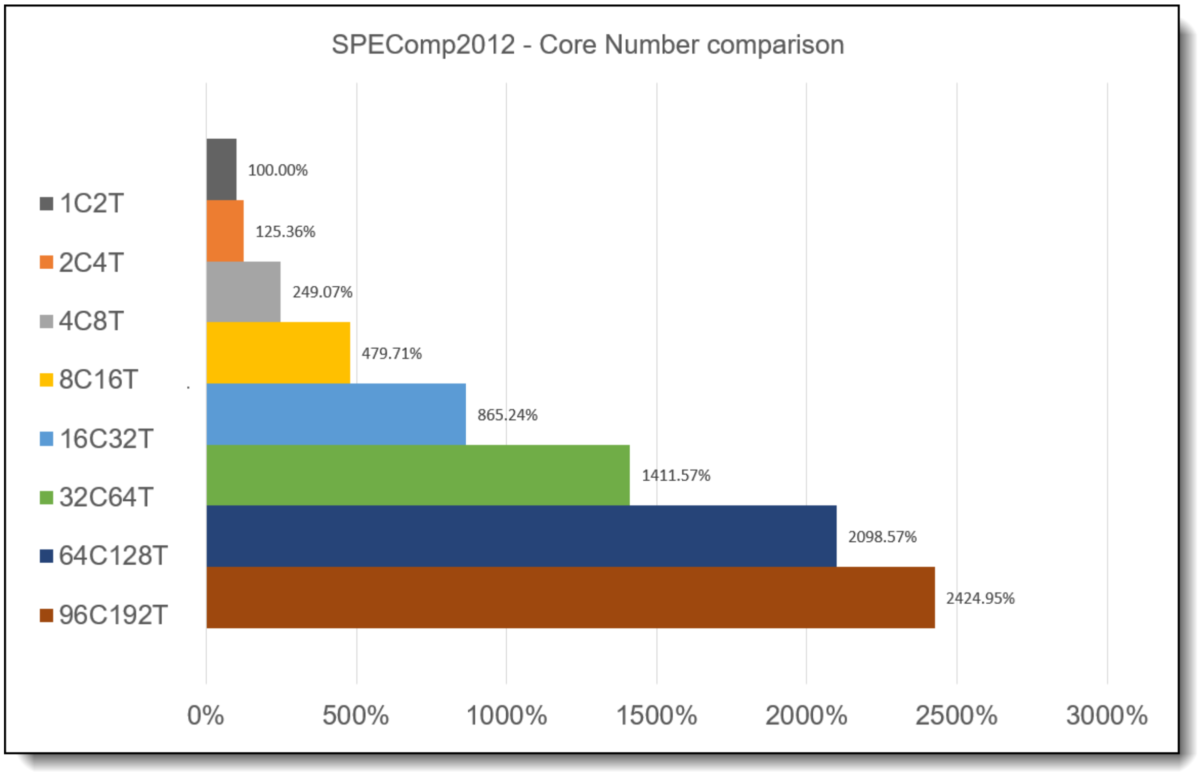


Figure 5. SPECcomp2012 result with different cores/thread

SMT

In Lenovo UEFI Maximum Performance Operating Mode, the SMT default is enabled. It not only improves the performance, but also impacts the numbers of the threads. Before start to run the SPECcomp2012 benchmark, please check the OpenMP environment variables OMP_NUM_THREADS, mapping to correct number of threads for best performance.

Enable SMT significantly improve the performance of SPECcomp 2012 benchmark. The following chart is comparing the SMT enable and disable results on SR655 V3 server.

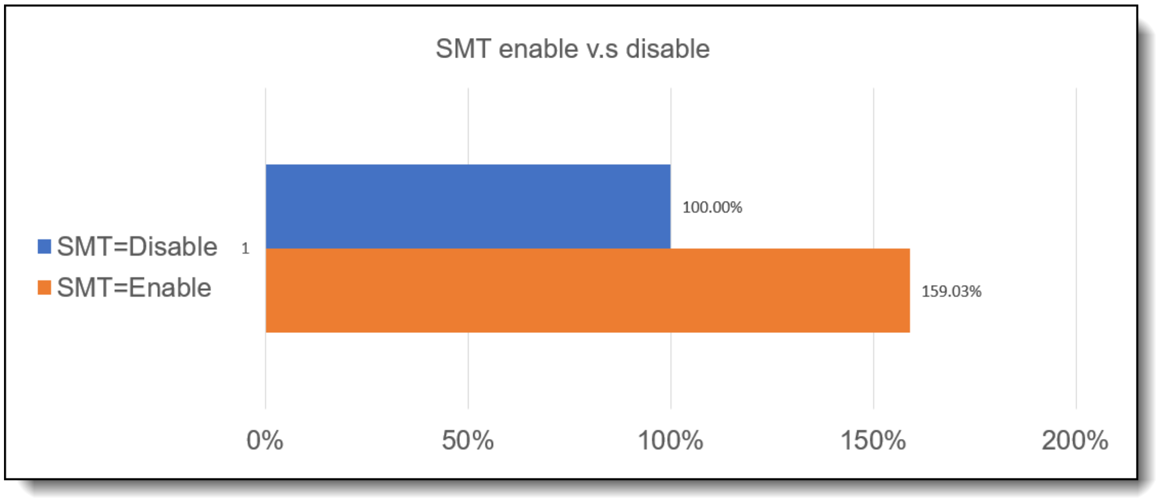


Figure 6. SMT feature results

Memory

The Lenovo ThinkSystem SR655 V3 supports the DDR5 memory frequency 4000MHz, 4400MHz and 4800MHz. Configure to the highest memory frequency further improve the performance up to 5% compared to lowest memory frequency as show in the picture below.

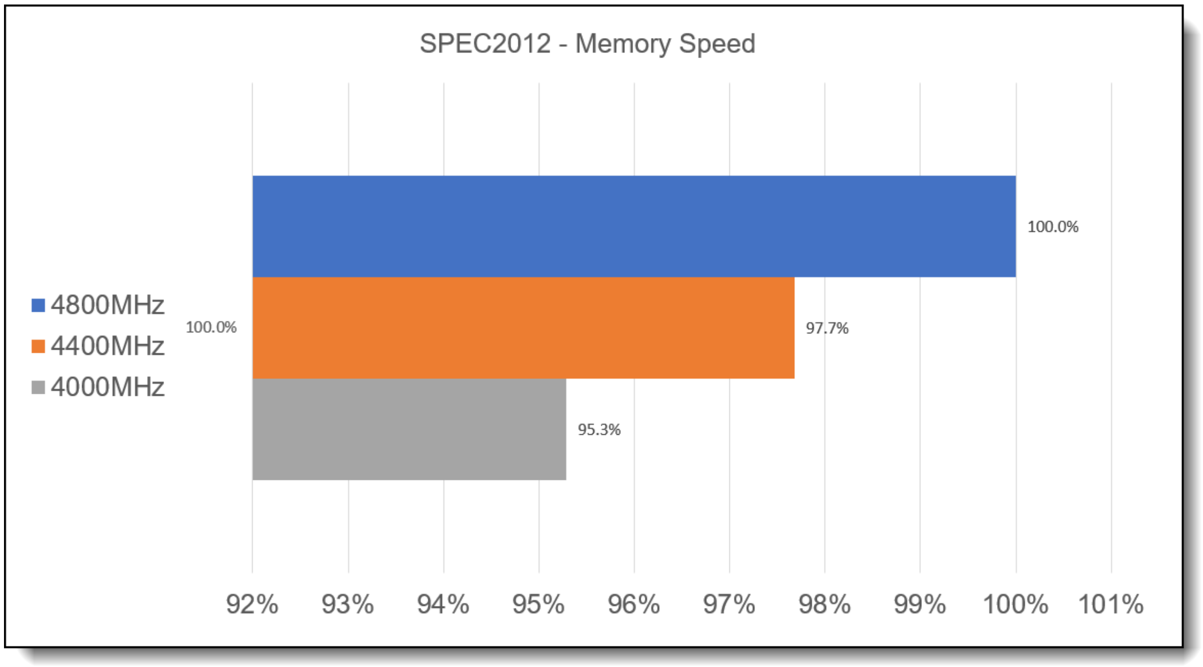
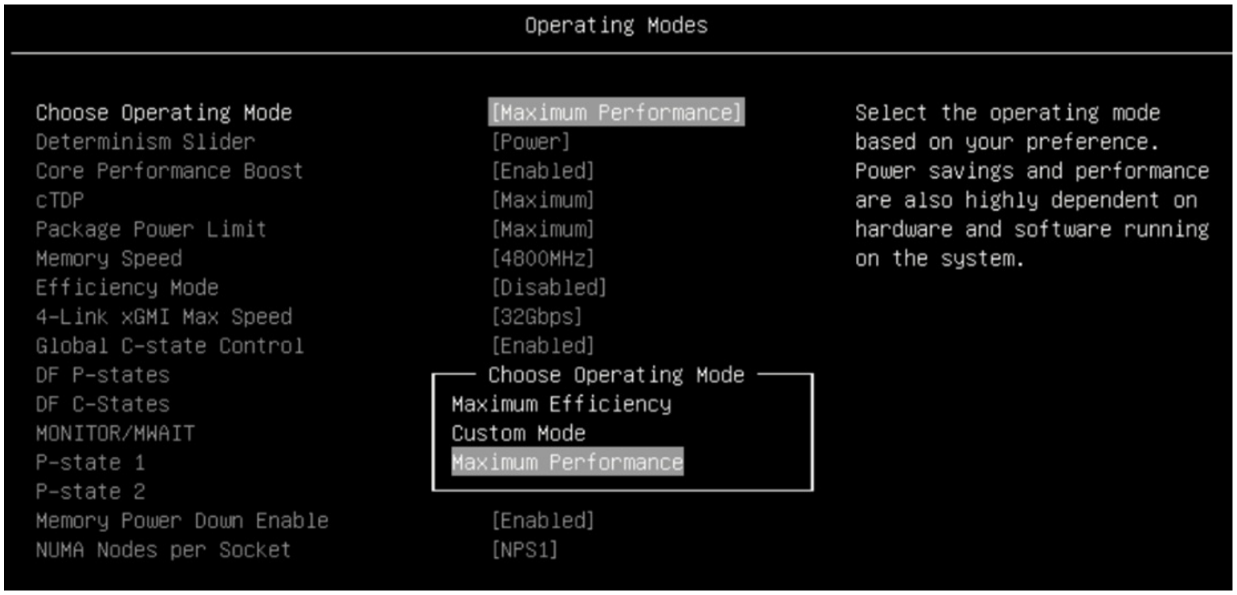


Figure 7. SPEComp2012 results for different memory speed

UEFI configuration

The Lenovo ThinkSystem SR655 V3 UEFI provides three presets for Operating Mode: Maximum Performance Mode, Maximum Efficiency and Custom Mode. We recommend loading UEFI default settings first and choose “Maximum Performance” preset in Operating Mode before start to run the SPEComp2012 benchmark.

Lenovo ThinkSystem UEFI firmware design several operating modes for difference purposes, the number of operating modes will depend on different system design. We always chose the **Maximum Performance Mode** for best system performance.



```
                                Operating Modes

Choose Operating Mode          [Maximum Performance]    Select the operating mode
Determinism Slider             [Power]                  based on your preference.
Core Performance Boost         [Enabled]                Power savings and performance
cTDP                           [Maximum]                are also highly dependent on
Package Power Limit            [Maximum]                hardware and software running
Memory Speed                   [4800MHz]                on the system.
Efficiency Mode                [Disabled]
4-Link xGMI Max Speed          [32Gbps]
Global C-state Control         [Enabled]
DF P-states                ┌── Choose Operating Mode ──┐
DF C-States                │Maximum Efficiency         │
MONITOR/MWAIT              │Custom Mode                │
P-state 1                  │Maximum Performance        │
P-state 2                  └───────────────────────────┘
Memory Power Down Enable       [Enabled]
NUMA Nodes per Socket          [NPS1]
```

Figure 8. UEFI setup menu for Operating Modes

The modes are as follows:

- **Maximum Efficiency**

  Maximum Efficiency mode maximizes the performance/watt efficiency. It provides the best features for reducing power and increasing performance in application where maximum bus speeds are not critical.

- **Maximum Performance**Maximum Performance mode will maximize the absolute performance of the system without regard for power. In this mode, power consumption is not taken into consideration. Attributes like fan speed and heat output of the system may increase in addition to power consumption. Efficiency of the system may go down in this mode, but the absolute performance may increase depending on the workload that is running.

- **Custom Mode**

  Custom Mode allows the user to individually modify any of the low-level settings that are preset and unchangeable in any of the other preset modes. Custom Mode will inherit the UEFI settings from the previous preset operating mode.

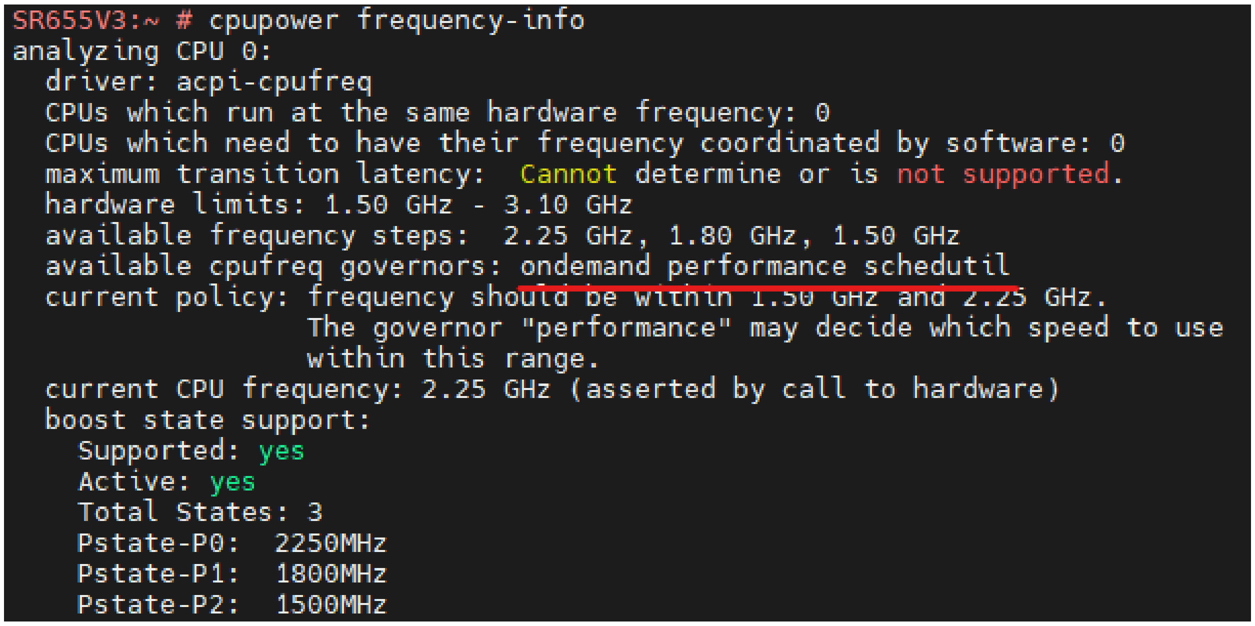
  For example, if the previous operating mode was the Maximum Performance operating mode and then Custom Mode was selected, all the settings from the Maximum Performance operating mode will be inherited. Note that there are certain settings that may be mutually exclusive or interdependent. For those settings an error will be surfaced if one of the pre-requisite or interrelated settings is set in such a way as to make configuration of the setting in question non-valid.

Linux utilities

To achieve best performance for AMD Genoa Processor we suggest using commercial Enterprise Linux OS with latest kernel, such as RHEL 9 or SLES 15. Besides kernel version, `cpupower` is one of user-level utilities that provides predefined governors and abilities for tuning CPU frequency and power features.

Use the following command to check processor state and list available governors:

```
cpupower frequency-info
```

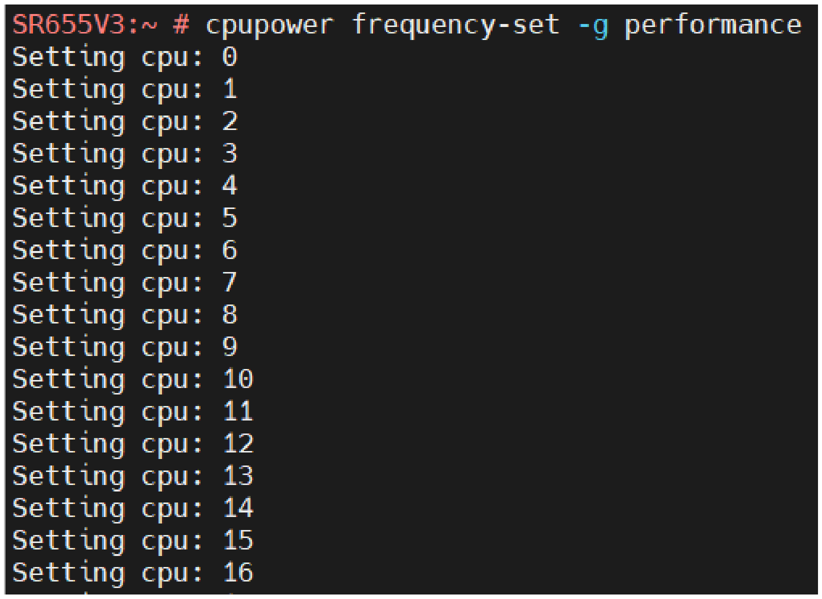


```
SR655V3:~ # cpupower frequency-info
analyzing CPU 0:
  driver: acpi-cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency:  Cannot determine or is not supported.
  hardware limits: 1.50 GHz - 3.10 GHz
  available frequency steps:  2.25 GHz, 1.80 GHz, 1.50 GHz
  available cpufreq governors: ondemand performance schedutil
  current policy: frequency should be within 1.50 GHz and 2.25 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency: 2.25 GHz (asserted by call to hardware)
  boost state support:
    Supported: yes
    Active: yes
    Total States: 3
    Pstate-P0:  2250MHz
    Pstate-P1:  1800MHz
    Pstate-P2:  1500MHz
```

Figure 9. cpupower frequency-info

Use the following command to switch to the “performance” governor to get better performance:

```
cpupower frequency-set -g performance
```



```
SR655V3:~ # cpupower frequency-set -g performance
Setting cpu: 0
Setting cpu: 1
Setting cpu: 2
Setting cpu: 3
Setting cpu: 4
Setting cpu: 5
Setting cpu: 6
Setting cpu: 7
Setting cpu: 8
Setting cpu: 9
Setting cpu: 10
Setting cpu: 11
Setting cpu: 12
Setting cpu: 13
Setting cpu: 14
Setting cpu: 15
Setting cpu: 16
```

Figure 10. set performance governor

Compiler flags

Support C/C++, Fortran with OpenMP 3.1, the Intel oneAPI DPC/C++ is one of the best commercial compiler toolsets we recommend building SPEComp 2012 benchmark under x86_64 architecture.

Here are suggested compiler flags use for SPEComp 2012 performance optimization.

- `-O3`: The higher compiler’s optimizations level, which generate performance optimized binary and reduced size of the binary.
- `-fopenmp`: Activating the OpenMP features based on OpenMP directives in the source codes.

- `-march`: Direct the compiler to generate binary for specific architecture. For example, “-march=core-avx2” would generates binary for the processors that support Intel Advanced Vector Extension2 (Intel AVX2) instructions.

OpenMP environment variables

The OpenMP environment variables configure the CPU resources allocation, CPU binding, memory binding and preferred runtime library when running OpenMP processes, the detail can find in the table below.

Table 4. OpenMP environment variables

Variables	**Description**	**Setting Value Example**
`KMP_AFFINITY`	Pin OpenMP threads to hardware threads	`compact,1`
`KMP_SCHEDULE`	sets the run-time schedule type and an optional chunk size, default is static.	`static`
`KMP_LIBRARY`	selects the OpenMP run-time library execution mode. The values are serial, turnaround, or throughput.	`turnaround`
`KMP_STACKSIZE`	sets the number of bytes to allocate for each OpenMP* thread to use as the private stack for the threads.	`768M`
`KMP_BLOCKTIME`	use the optional characters suffixes: s (seconds), m (minutes), h (hours), or d (days) to specify the units, specific infinite for an unlimited wait time.	`infinite`
`OMP_DYNAMIC`	Enables (TRUE) or disables (FALSE) the dynamic adjustment of the number of threads.	`FALSE`
`OMP_NUM_THREADS`	sets the specifies the number of threads to use for parallel regions.	`256`

In SPEComp2012 benchmark configuration file, add `ENV_` for all OpenMP environment variables. For example:

```
ENV_OMP_NUM_THREADS=256
```

```
ENV_KMP_AFFINITY=compact,1
```

Conclusion

For the best SPEComp2012 performance, we provided tuning suggestions from hardware configurations to firmware settings, compiler flags and OpenMP environment variables. Besides, the performance delta with and without suggestions also been provided for user reference.

Applying all the tuning steps mentioned here, the ThinkSystem SR655 V3 set a new performance world recorded on SPEComp2012. More detailed information can be found from the publish results URL: https://lenovopress.lenovo.com/lp1758-sr655-v3-specompg-benchmark-result-2023-07-01

Author

Sinper Liang is a performance engineer in the Lenovo Infrastructure Solution Group laboratory located at Taipei, Taiwan. Sinper joined Lenovo in 2019 and focuses on system performance validation and the SPEC OMP2012 benchmark. Prior to Lenovo, he worked at the IBM Taiwan Systems and Technology Laboratory as a system UEFI firmware assurance and validation engineer, and at Wiwynn as the UEFI test leader.

Related product families

Product families related to this document are the following:

- Processors
- SPEComp Benchmark Results

Notices

This document, LP1840, was created or updated on November 2, 2023.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP1840
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP1840.

Trademarks