4th Gen AMD EPYC™ Processor Benchmark Report

Document ID: CC00204
Version: 1.0
Writer: Technology Group
HPC Division
HPC SYSTEMS Inc.
Last updated: February 1st, 2023

Table of Contents

1. Abstract

4th Gen AMD EPYC™ Processor (code name: Genoa) was released on November 10th, 2022, emerging with its remarkable features: adoption of new microarchitecture "Zen 4", miniaturization by the 5nm manufacturing process, a many-core system with up to 96 cores in 1 socket and up to 192 cores in 2 sockets, wider memory bandwidth due to an increase of memory channels to 12 lines (see chapter 2 for the details). Moreover, functional enhancements of the processor were also made for AI & HPC workloads to support AVX-512 instructions, and IPC (instruction per cycle) increased by about 14% on average compared with that of the former generation owing to its improvement in cache hierarchy and branch prediction, which is expected to accelerate both parallel and single thread applications through these advancements.

In order to investigate realistic performance of 4th Gen AMD EPYC™ Processor, our benchmark survey of various applications had taken place by comparatively evaluating each effective performance in 3 different environments with a 2 socket machine of 4th Gen AMD EPYC™ Processor, a 2 socket machine of the previous generation processor (code name: Milan-X, Zen 3 adopted), and a machine of 3rd Gen Intel Xeon® Scalable Processor (IceLake architecture).

2. Features of 4th Gen AMD EPYC™ Processor

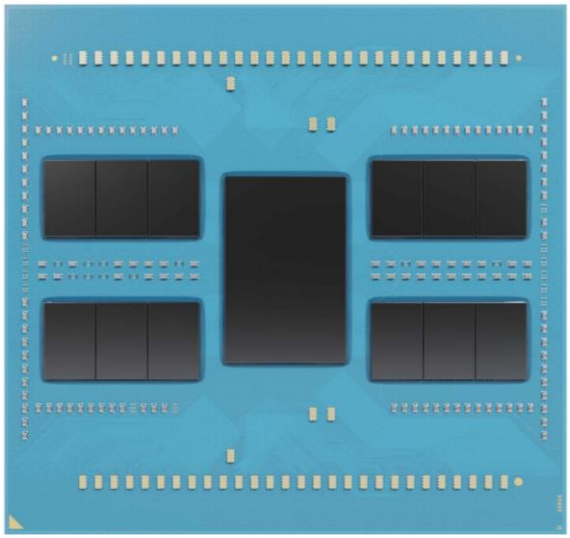


Figure 1 Overview of 4th Gen AMD EPYC™ Processor

- Miniaturized by 5nm manufacturing process that takes advantage of 4th Gen FinFET, a top-bin of SKU "EPYC 9654" contains as many as 96 cores in one socket. In addition, a maximum of 12 CCD per CPU package can be loaded due to its miniaturization.
- Hybrid multi-die architecture is adopted, allowing users to utilize optimized process technology to respective CPU cores by separating them from I/O.
- AVX-512 instructions which is optimal for AI and HPC acceleration is supported, including BFloat16 and VNNI instructions.
- IPC (instruction per cycle) increased by about 14% on average compared with that of the previous generation owing to repeated improvements in L2 cache, execution engine, branch prediction, load / store, and frontend.
- 1 MB of L2 cache per core is contained, which is larger than that of the previous generation.
- 32 MB of L3 cache is contained per CCD.
- Inter-processor connection by AMD Infinity Fabric became two times stronger than that of previous generation.
- A maximum of 160 lanes in PCIe Gen5.0 (two times faster transfer speed than 4.0) are available.
- CXL 1.1+ and CXL 2.0 memory device is supported.
- 12 lines of DDR5-4800 memory channel are loaded, and its memory bandwidth became enhanced to be 2.3 times larger than that of the previous generation. The flexibility of memory configuration can be increased by interleaving with 2, 4, 6, 8, 10, or 12 lines of bus.
- Its memory supports 256 bit AES-XTS encryption.
- The number of SEV-SNP guest increased up to twice as many as that of the previous generation, thanks to the new AMD Infinity Guard function.

3. Benchmark environments

This benchmark test had been done with the following three computers.

- 4th Gen AMD EPYC™ Processor computer ("Genoa env." from the following)
  CPU: AMD EPYC 9654 (2.4 GHz / 96 core) x 2
  L1 data cache: 32 K
  L1 instruction cache: 32 K
  L2 cache: 1024 K
  L3 cache: 32768 K/CCD, 384 M/CPU
  Memory: 768GB (32GB DDR5-4800 ECC RDIMM x 24)
  Theoretical memory bandwidth: 921.6 GB/s (total 2 CPU)
  Storage: 500GB NVMe SSD x 1
  OS: AlmaLinux 8.7

- 3rd Gen AMD EPYC™ Processor computer ("Milan-X env." from the following)
  CPU: AMD EPYC 7773X (2.2 GHz / 64 core) x 2
  L1 data cache: 32 K
  L1 instruction cache: 32 K
  L2 cache: 512 K
  L3 cache: 98304 K / CCD, 768 M / CPU
  Memory: 1024 GB (64 GB DDR4-3200 ECC RDIMM x 16)
  Theoretical memory bandwidth: 409.6 GB / s (total 2 CPU)
  Storage: 1.92 TB NVMe SSD x 1
  OS: AlmaLinux 8.5

- 3rd Gen Intel® Xeon® Scalable Processor computer ("IceLake env." from the following)
  CPU: Engineering sample (2.6 GHz / 24 core) x 2
  L1 data cache: 48 K
  L1 instruction cache: 32 K
  L2 cache: 1280 K
  L3 cache: 36864 K
  Memory: 512 GB (32 GB DDR4-3200 ECC RDIMM x 16)
  Theoretical memory bandwidth: 409.6 GB / s (total 2 CPU)
  Storage: 1.9 TB SATA 6 Gbps TLC SSD x 1
  OS: CentOS 8.2

The versions of compliers, libraries, and applications which were used in this benchmarking are shown below.

Compiler:	[for HPL] AMD Optimizing C/C++ and Fortran Compilers (AOCC) 4.0.0
	[for others] Intel® oneAPI Base & HPC Toolkit Classic Compiler 2022.2.0
MPI:	[for HPL] Open MPI 4.1.4
	[for others] Intel® MPI Library 2021.6.0
BLAS:	[for HPL] AMD Optimizing CPU Libraries (AOCL) 4.0
	[for others] Intel® oneAPI Math Kernel Library 2022.2.0
HPL:	Version 2.3
STREAM	Version 5.10 (stream.c)
Gaussian:	Gaussian16 Rev. C.01 AVX2-optimized binary
Amber:	Amber22 patch 1, AmberTools22 patch 3
VASP:	Version 6.3.2

4. Performance results

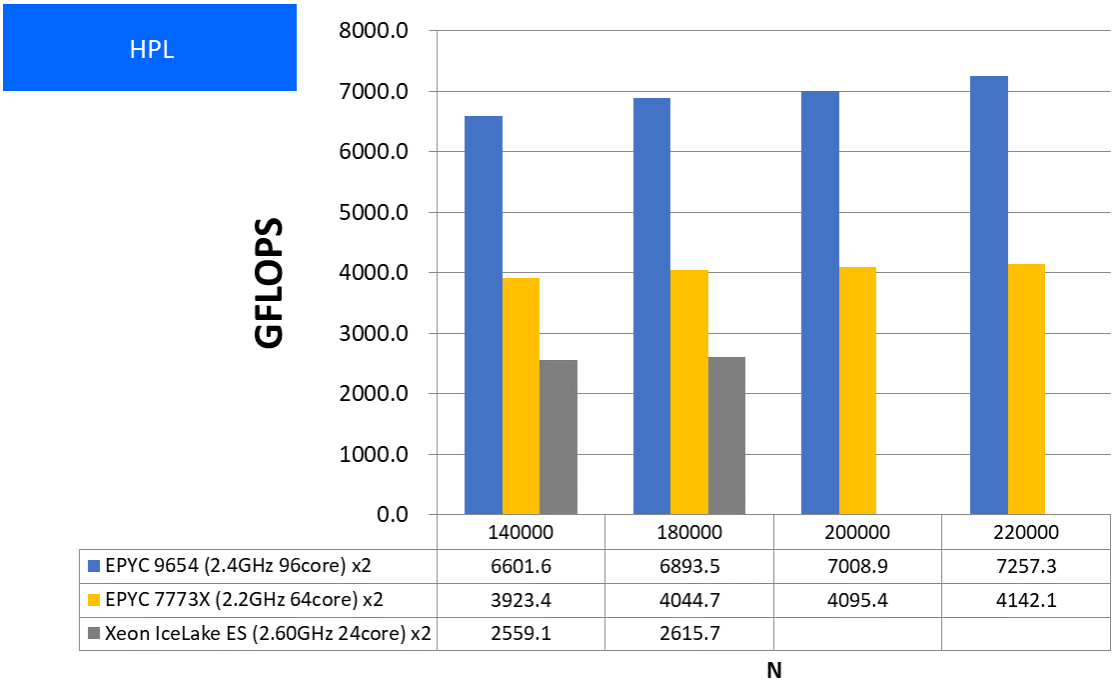
4.1. HPL

HPL is a benchmark program that is used in the supercomputer performance ranking "Top 500". The program is for solving simultaneous equation and evaluates the floating-point arithmetic performance by FLOPS unit (the number of floating-point operation instruction that can be processed per second). Considering that HPL is well known as a computationally intensive benchmark program, we exploited it for understanding floating-point arithmetic performance of CPU.

As for building and execution of HPL, those setups took place according to AOCL User Guide from AMD Inc. The following are the results of benchmarking.

Table 1 Performance of HPL

Number of nodes	Number of parallels	Number of floating-point operation instruction per second in HPL [GFLOPS]		
		Genoa env.	Milan-X env.	IceLake env.
1	N=140,00 Full core (192 / 128 / 48)	6601.6	3923.4	2559.1
	N=180,000 Full core (192 / 128 / 48)	6893.5	4044.7	2615.7
	N=200,000 Full core (192 / 128 / 48)	7008.9	4095.4	(Lack of memory)
	N=220,000 Full core (192 / 128 / 48)	7257.3	4142.1	(Lack of memory)



	140000	180000	200000	220000
EPYC 9654 (2.4GHz 96core) x2	6601.6	6893.5	7008.9	7257.3
EPYC 7773X (2.2GHz 64core) x2	3923.4	4044.7	4095.4	4142.1
Xeon IceLake ES (2.60GHz 24core) x2	2559.1	2615.7		

Figure 2 Performance of HPL

Feature:

Achievement of considerably higher floating-point arithmetic performance compared with that of previous generation and other CPU

A great deal of cores (96 cores in 1-socket) contributed directly to its high floating-point arithmetic performance. For example, the performance became 1.75 times higher in case of N=220,000 compared with that of Milan-X env., increasing dramatically against the previous. Please refer to these results as a rough measure of effective performance advancements regarding computationally intensive benchmark program.

4.1.1. Supplement: Comparison between different development environments

Additionally, HPL was also built and executed by oneAPI Base & HPC Toolkit Classic Compiler 2022.2.0, oneAPI MKL 2022.2.0, and Intel MPI 2021.6 under Genoa env. As a result, the performance that was built with AOCC・AOCL・OpenMPI showed higher FLOPS scores as the following figure.

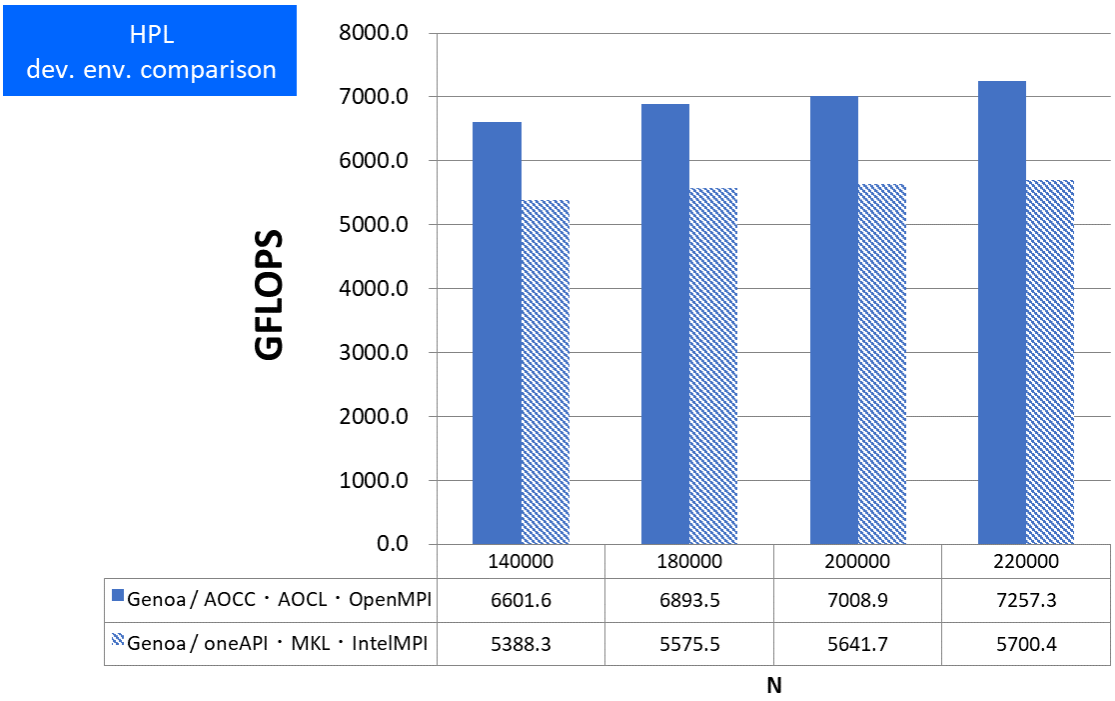


Figure 3 Comparison of HPL performance between different development environments

Considering BLAS library influences the most of HPL performance, there will be a case that AMD's AOCL works better than Intel's MKL under the Genoa env., although HPL was only evaluated this time. Also, it may be possible that AOCL will contribute to speed-up more than MKL in some cases depending on practical applications[1]. Therefore, users should select their development environment carefully in case of making high-speed application binary for AMD processor, not just simply thinking "Always OK to select MKL for BLAS".

[1] From our knowledge by the previous works, we will also note a case that SIMD optimization of Intel MKL don't work effectively in AMD processor. Also, as another notice, MKL_DEBUG_CPU_TYPE environment variable was not applied in all benchmark tests written in this report, because it could not be used in the present version of MKL.

4.2. STREAM

STREAM is a benchmark program that is often used to measure performance of memory bandwidth. Especially, Triad inside STREAM is an OpenMP parallel program that performs multiply-accumulate operation of huge one-dimension vector and calculates whole node bandwidth of memory input and output by parallel operation.

STREAM was benchmarked with a system which was built by AVX-512 optimized option in oneAPI 2022.2.0. The results of peak period in a node are as follows.

Table 2 Performance of STREAM (Traid)

Number of nodes	Number of parallels	Memory bandwidth of STREAM (Triad) [GB/s]		
		Genoa env.	Milan-X env.	IceLake env.
1	Full core (192 / 128 / 48)	674.8	332.6	283.0

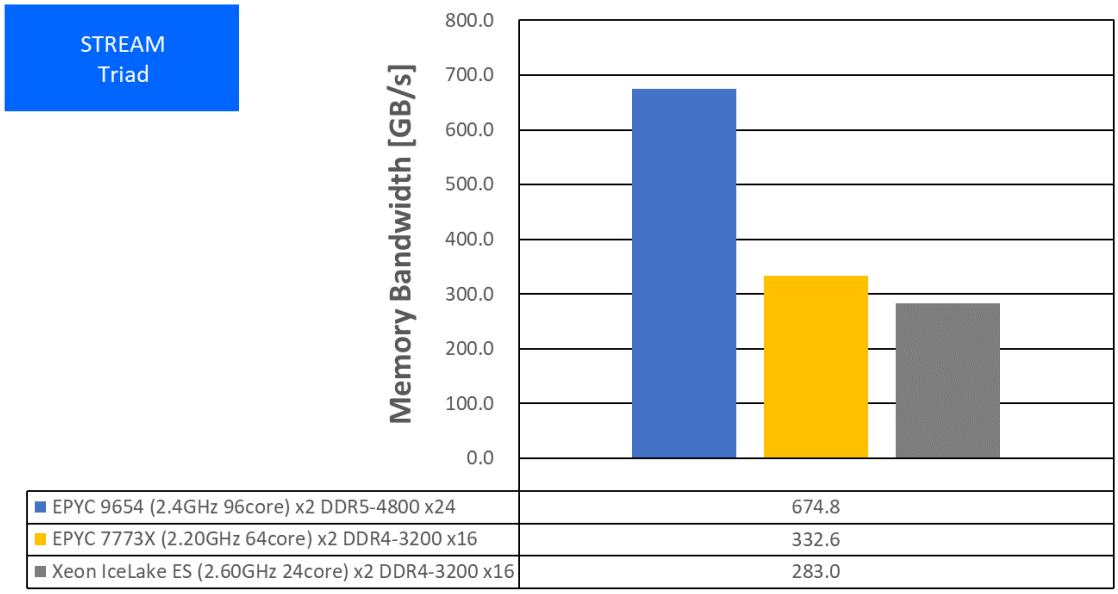


Figure 4 Performance of STREAM (Triad)

Feature:

Attained effective memory bandwidth which is two times larger than that of previous generation

The number of memory channels increased from 8 to 12, and DDR5-4800 support has been added to achieve 2.03 times the memory bandwidth performance of the previous generation. These results are significant for users who perform computation mainly influenced by the memory bandwidth such as stencil and FFT.

4.3. Gaussian

Benchmarking of Gaussian (de facto standard program of quantum chemistry) was performed. A binary version package of Gaussian Inc.'s standard was optimized in AVX2, and was used for the evaluation (not supported in AVX-512).

Elapsed time of test0397 input (Valinomycin molecular $C_{54}H_{90}N_6O_{18}$, force calculation with RB3LYP/3-21G, attached in Gaussian package) were measured. In addition, we also evaluated elapse time of rkest0397 input of which basis function was altered to 6-31G (d,p).

Table 3 Elapsed time of Gaussian16 (test0397)

Number of nodes	Number of parallels	Elapsed time in computation of Gaussian 16 Rev. C.01 (test0397) [sec]		
		Genoa env.	Milan-X env.	IceLake env.
1	48	45.2	55.6	71.6
	64	37.3	46.5	
	96	30.0	41.4	
	128	27.6	41.0	
	192	27.1		

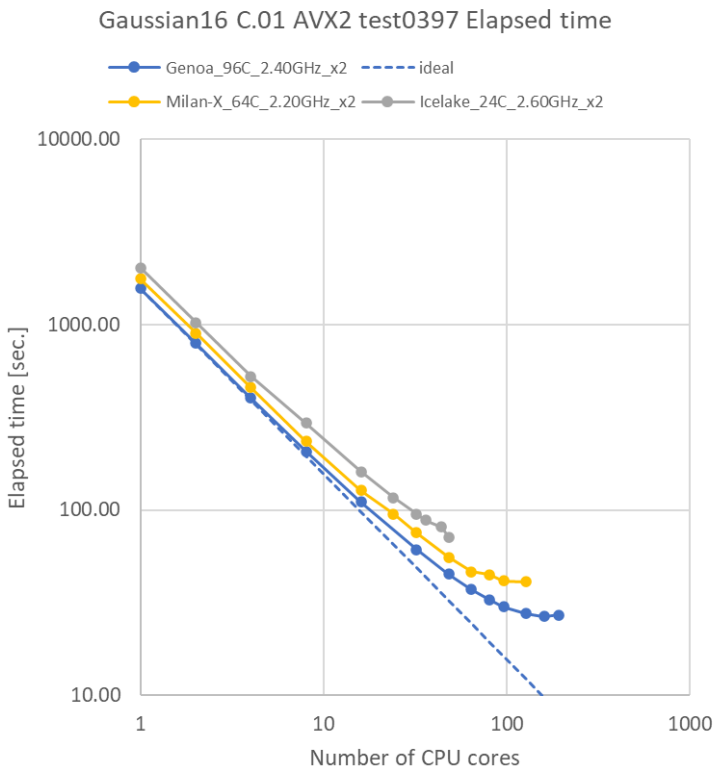


Figure 5 Elapsed time of Gaussian16 (test0397)

Table 4 Elapsed time of Gaussian16 (rkest0397)

Number of nodes	Number of parallels	Elapsed time in computation of Gaussian 16 Rev. C.01 (rkest0397) [sec]		
		Genoa env.	Milan-X env.	IceLake env.
1	48	164.5	200.4	257.6
	64	134.2	166.0	
	96	105.0	143.8	
	128	95.8	135.7	
	192	89.4		

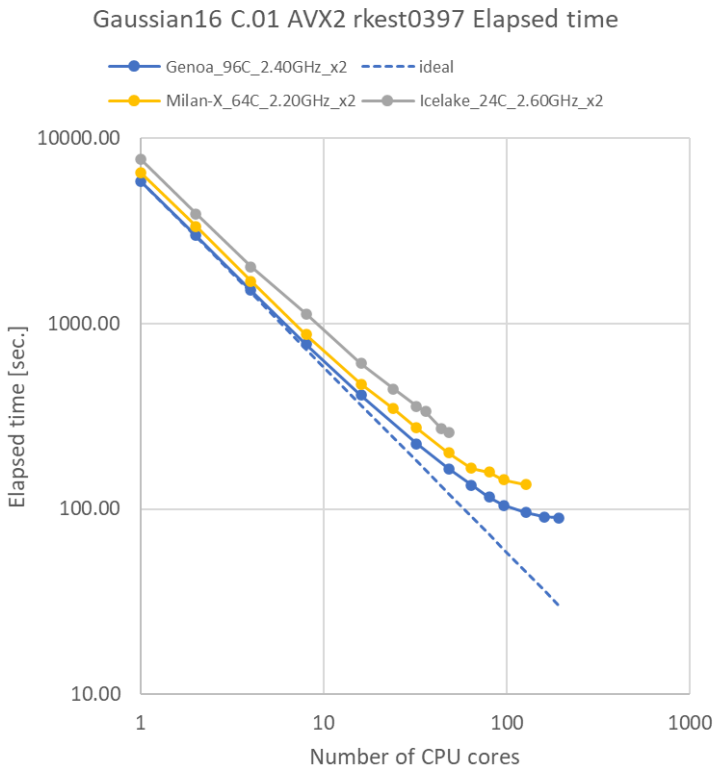


Figure 6 Elapsed time of Gaussian16 (rkest0397)

Feature:

Achieved acceleration in all number of parallel from previous generation

There were total speed ups over all parallel numbers, and parallel scalability until around 96 cores was similar to a case of the previous generation (Milan-X env.). In 48 parallel of rkest0397, the computing speed of Genoa env. became 1.22 times faster than that of Milan-X env. and 1.57 times faster than that of IceLake env. respectively. This acceleration was driven by the improvement of IPC as well as a point that CPU clock got boosted during AVX2 operation thanks

to AMD Turbo Core technology[2].

As for parallel scalability of Genoa system, there was a struggle in growth when the number of parallel was over 128 parallel, which had not been seen in IceLake env. Taking account of Gaussian's computationally intensive tendency, it is thought that scale of this benchmarking was not enough for the number of parallel to make its growth.

Considering that the Gaussian's computationally intensive tendency and the parallel scalability until around 96 cores, configuration with many CPU cores is basically preferred for computation of Gaussian. Beyond this number of parallel, there can be a situation that a lack of performance improvement will occur due to the very large amount of parallel number against computing scale like this time. In this case, a countermeasure during operation, making full use of CPU core by running several computing jobs parallelly for example, will allow users to exploit the computing power of Genoa systems thoroughly.

4.4. Amber

Amber is one of the biomolecule simulation software. Our benchmarking was performed in a system that was built by CPU optimization of AVX-512, AVX2 and AVX with Intel oneAPI Base & HPC Toolkit Classic Compiler. However, since its numerical accuracy was not enough to pass our test when the system was built with MKL, we used binary built by compiling OpenBLAS which is attached in Amber, instead of using MKL.

In this benchmarking, we measured performance (ns/day) regarding input of Cellulose NVE including 408,000 atoms (distributed in a GPU version official website of pmemd in Amber) by MPI parallel computing with pmemd. Also, performance by input of Nucleosome (Implicit Solvent, GB) including 25,095 atoms was measured as well.

[2] As a result of profiling by using perf command attached in Linux, IPC in Genoa env. was about 12% higher than that in Milan-X env.

Table 5 Performance of Amber22 (Cellulose NVE, PME)

Number of nodes	Number of parallels	Performance of Amber22 (Cellulose NVE, PME) [ns/day]		
		Genoa env.	Milan-X env.	IceLake env.
1	16	3.31	3.05	2.83
	32	6.10	5.81	5.00
	48	(Not measured)	7.29	6.89
	64	10.01	8.55	
	96	11.99	9.72	
	128	12.26	9.81	
	160	11.49		
	192	10.58		

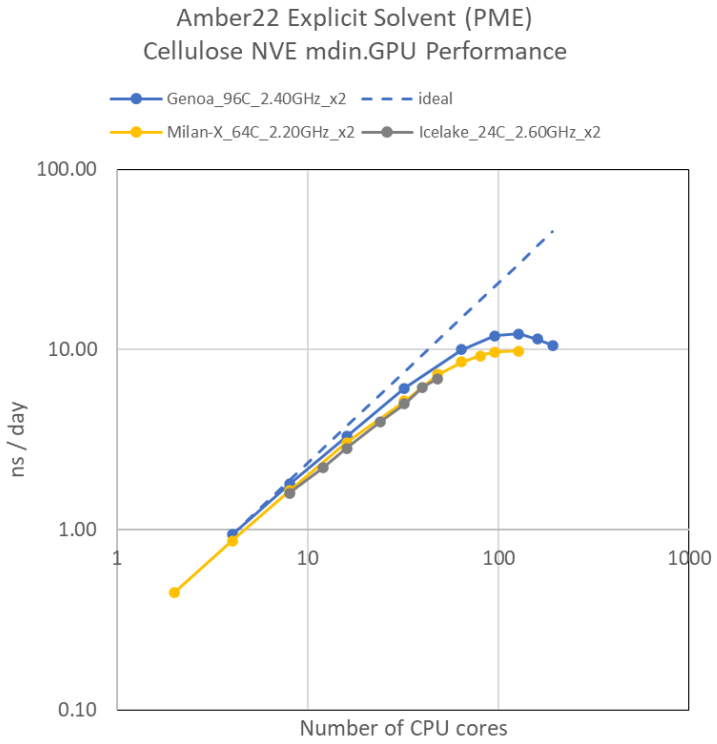


Figure 7 Performance of Amber22 (Cellulose NVE, PME)

Table 6 Performance of Amber22 (Nucleosome, GB)

Number of nodes	Number of parallels	Performance of Amber22 (Nucleosome, GB) [ns/day]		
		Genoa env.	Milan-X env.	IceLake env.
1	16	0.46	0.52	0.69
	32	0.93	0.81	1.31
	48	(Not measured)	1.11	1.69
	64	1.85	1.58	
	96	2.65	2.17	
	128	3.26	2.66	
	160	3.75		
	192	4.00		

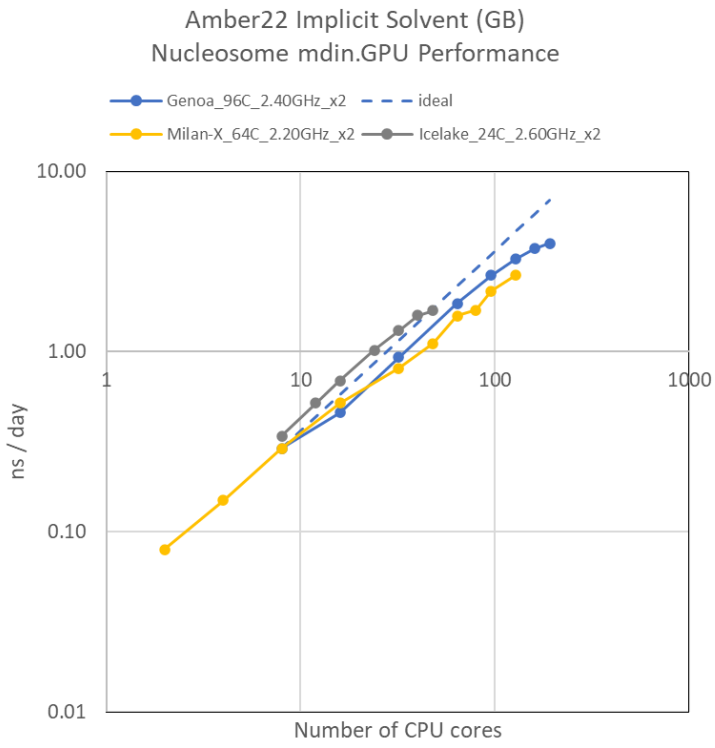


Figure 8 Performance of Amber22 (Nucleosome GB)

Feature:

Attained higher effective performance than that of previous generation in almost all number of parallel

As for the computation of Cellulose NVE in Explicit Solvent (PME), it was revealed that the effective performance got higher than that of Milan-X env. in all parallel number cases. However, there was the struggle in increasing parallel scalability when the number of parallel was over 64, which is similar to the previous generation, and the computing speed became rather slower when the parallel number was over 128. Thus, we recommend users to choose the appropriate number of cores taking account of parallel scalability.

On the other hand, regarding the case of Nucleosome in Implicit Solvent (GB), there was an achievement of the effective performance which was higher than that of Milan-X env. (the previous generation model) in the condition of over 32 parallel, showing excellent scalability up to full core of parallel. Also, in comparison between each performance with the same parallel number, the computing speed in IceLake env. became the fastest of the three environments. We presume that instruction sequence and cache access with comparatively low parallelisms are dominant in workloads of GB, from our knowledge of previous benchmarking. Moreover, considering results of VASP (written later) that comparatively high performance was attained in Genoa env. despite SSE2 compatible mode, we speculate that CPU of Genoa have been designed to enhance parallel execution throughput of instructions while IceLake env. are optimized to shorten execution latency of instruction sequence. As a consequence, it is supposed that IceLake env. was able to execute the GB workloads more efficiently owing to these facts as a whole.

4.5. VASP

VASP is a first-principle electronic structure program package which is based on the density functional method with plane wave & pseudopotential basis. VASP tends to use a lot of bandwidth between CPU and memory during parallel operation.

PAW GGA and USPP computations of 1000 atoms was benchmarked by binary which was built with SIMD optimization of AVX2[3] using Intel oneAPI Base & HPC Toolkit Classic Compiler and linking MKL, followed by comparison between each elapsed time. In addition, results of elapsed time made by input files of practical materials are shown below as well, which was requested from our customers before (computation by DFT: PAW-PBE in which 40 atoms involved, the details are confidential).

[3] Because of a result that binary in which AVX-512 was contained in SIMD optimization failed in our numerical accuracy test.

Table 7 Elapsed time of VASP 6.3.2 (1000 atoms, PAW GGA)

Number of nodes	Number of parallels	Elapsed time in computation of VASP 6.3.2 (1000 atoms, PAW GGA) [sec]		
		Genoa env.	Milan-X env.	IceLake env.
1	16	1381.4	1429.7	1133.1
	32	871.2	958.7	808.0
	48	(Not measured)	852.1	756.2
	64	636.4	764.6	
	96	607.0	2111.1	
	128	609.9	1007.2	
	192	849.1		

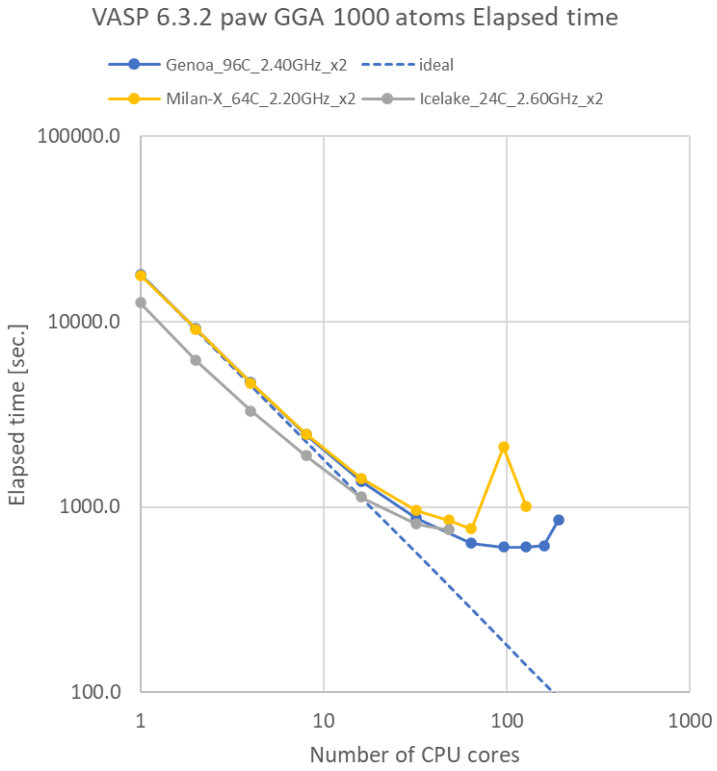


Figure 9 Elapsed time of VASP 6.3.2 (1000 atoms, PAW GGA)

Table 8 Elapsed time of VASP 6.3.2 (1000 atoms, USPP)

Number of nodes	Number of parallels	Elapsed time in computation of VASP 6.3.2 (1000 atoms, USPP) [sec]		
		Genoa env.	Milan-X env.	IceLake env.
1	16	786.1	797.9	611.6
	32	539.0	596.3	458.0
	48	(Not measured)	536.4	421.4
	64	415.2	550.7	
	96	1222.7	1779.1	
	128	407.6	1621.5	
	192	842.9		

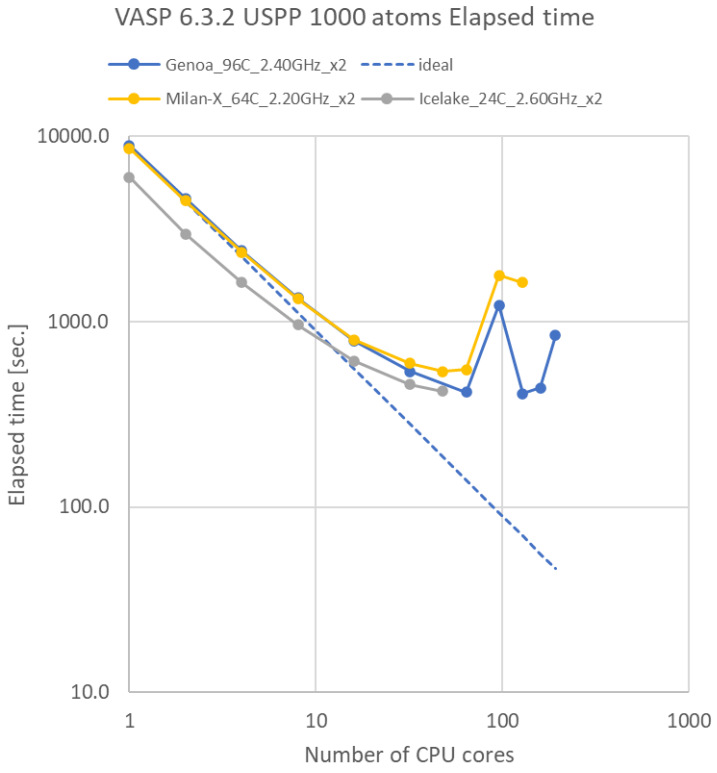


Figure 10 Elapsed time of VASP 6.3.2 (1000 atoms, USPP)

Table 9 Elapsed time of VASP 6.3.2 (40 atoms, DFT: PAW-PBE)

Number of nodes	Number of parallels	Elapsed time in computation of VASP 6.3.2 (40 atoms, DFT: PAW-PBE) [sec]		
		Genoa env.	Milan-X env.	IceLake env.
1	16	925.8	1107.9	956.5
	32	584.1	709.7	645.5
	48	(Not measured)	600.0	562.2
	64	488.6	620.9	
	96	525.7	751.9	
	128	476.9	665.2	
	192	914.0		

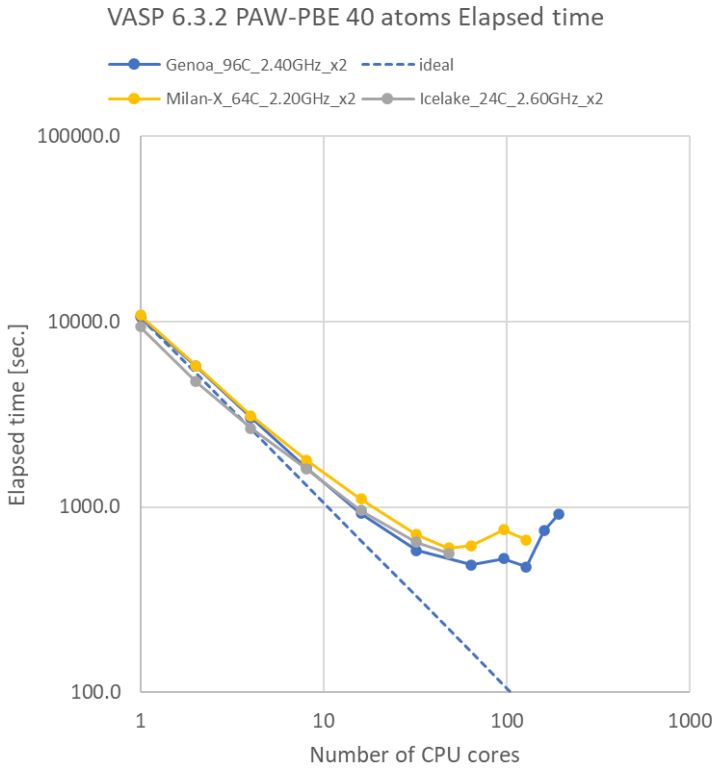


Figure 11 Elapsed time of VASP 6.3.2 (40 atoms, DFT: PAW-PBE)

Feature:

Improvement of scalability in 32-64 parallel

There was a development of scalability from the previous generation in 32-64 parallel. This growth is considered to be driven by implementation of large memory bandwidth comprising 12 channels of DDR5-4800. In addition, the higher performance was gained than that of IceLake env. when the number of parallel was over 16 in DFT: PAW-PBE computation of 40 atoms. On the other hand, a struggle in growth was observed when the number of parallel processing was over 64, which is same as before. This result is considered to happen due to the memory bandwidth influenced by characteristic of VASP in which data traffics of $O(N^2)$ are generated.

As a whole, the computing speed of IceLake env. showed the fastest result of the 3 environments. The reasons for this can be described by a fact that SIMD optimization of MKL didn't work effectively resulting in operation just by SSE2 compatible mode in Genoa and Milan-X env. Accordingly, it is assumed that performance in the SSE2 compatible mode becomes about a little less than one-fourth of that in AVX-512 compatible mode because of the negative effects of performance degradation caused by decrease of SIMD bit-length, latency for concurrent execution of SIMD instructions, and increase of clock. However, a difference in performance between Genoa and IceLake env. is not so big compared to the above-mentioned, which can be explained by the following reasons.

- Memory bandwidth became larger in Genoa env. (refer to the results of STREAM in section 4.2).
- Genoa env. doesn't cause the decrease of operation clock, which happens under operation of AVX2 in IceLake env.

In addition, a difference in performance between IceLake and Genoa env. became smaller in DFT: PAW-PBE computation of 40 atoms compared with that in PAW GGA and USPP computations of 1000 atoms. The reason for this can be explained by a point that the former is the computation in which same structure is repeated with a single type of atoms while the latter is the one with complex structures consisting of various types of atoms. Accordingly, the former shows high cache hit ratio in computation, and is input which has comparatively small influence of memory read/write speed. Therefore, this is why the difference in performance of MKL presumably got bigger in the former case.

5. Summary

What is revolutionary about 4th Gen AMD EPYC™ Processor is that a high degree of integration came true thanks to its miniaturization by the 5nm manufacturing process leading to 96 cores in 1 socket and to 192 cores in 2 sockets (it is thought to be a 4-socket machine in a conventional sense). Therefore, the processor will be a potential platform which makes a chance to acquire new viewpoints in HPC, taking account of the results that the performance exceeded 7TFLOPS in HPL despite 1 node of computer and the scalability struggled to level up when the number of parallel was over 128 in Gaussian.

In addition, the enhancement of IPC (Instruction per cycle) was confirmed in benchmarking the practical applications (about +12% in Gaussian16) as AMD Inc. have noticed in advance. Since the increase of IPC will also contribute to acceleration regarding workloads in which performance reach those peaks by issues of operation clock and the number of parallel, this feature can be good news for those who have troubles in such workloads.

As for Amber, the higher effective performance by the PME computation of Cellulose NVE was achieved in all parallel processing number compared with the performance of Milan-X env. (the previous generation), but the struggle of growth in parallel scalability was also observed when the number of parallel processing was over 64. Moreover, the effective performance by GB computation of Nucleosome successfully became higher than that of Milan-X env. in 32 parallel, showing excellent scalability up to full core of parallel. Taking account of these results of parallel scalability and computation methods comparatively both together, it is essential for users to decide the appropriate number of CPU cores for these kinds of computation.

Regarding memory bandwidth, we confirmed overwhelmingly large bandwidth which is more than two time larger than bandwidths of the previous generation and other CPU owing to 24 lines of memory channels (12 lines per socket) in DDR5-4800 in benchmarking of STREAM. Thus, this new platform can be greatly expected to enhance performance in applications that strongly require memory bandwidth such as stencil computation and FFT.

Even in VASP which is the practical application influenced by memory bandwidth, the growth in scalability against the previous generation was also observed in 32-64 parallel. On the other hand, the growth reached to its peak as before when the number of parallel processing was over 64 due to memory bandwidth influenced by characteristic of VASP in which traffics with a squared amount of parallel number are generated. Additionally, the performance got lower than that of IceLake env. because MKL was only to be operated by SSE2 compatible mode (benchmarking was only carried out with VASP built by MKL due to our schedule). Therefore, in case of making fast application binary for AMD processor, there will be a situation carefully starting from selection of development environment such as BLAS library.

At the end, HPC SYSTEMS Inc. will continue to pursue efficiency and acceleration of practical computation by acquiring and analyzing supportive information of computing speed through operational verification of the latest hardware, compilers, libraries and applications provided from various makers, supported by our technology group staffs who specialize in both scientific computation and high-performance computer.

6. Revision history

Version No.	Last Updated	Summary
1.0	Feb. 1st, 2022	Created the first draft (English ver.).