J Supercomput (2017) 73:3948–3976
DOI 10.1007/s11227-017-1993-y

Potential of a modern vector supercomputer for practical applications: performance evaluation of SX-ACE

Ryusuke Egawa[1] · Kazuhiko Komatsu[1] · Shintaro Momose[3] · Yoko Isobe[3] · Akihiro Musa[3] · Hiroyuki Takizawa[1] · Hiroaki Kobayashi[2]

Published online: 7 March 2017

Abstract Achieving a high sustained simulation performance is the most important concern in the HPC community. To this end, many kinds of HPC system architectures have been proposed, and the diversity of the HPC systems grows rapidly. Under this

This paper is an extended version of a following poster papers of SC14 and SC15. R. Egawa, S. Momose, K. Komatsu, Y. Isobe, H. Takizawa, A. Musa and H. Kobayashi,: "Early Evaluation of the SX-ACE Processor," in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC14), Poster, Nov. 2014, pp. 1–2 (USB).
K. Komatsu, R. Egawa, R. Ogata, Y. Isobe, H. Takizawa, and H. Kobayashi,: "An Approach to the Highest Efficiency of the HPCG Benchmark on the SX-ACE Supercomputer," in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC15), Poster, Nov. 2015, pp. 1–2 (USB).
The key additions of this journal version are detailed performance evaluation and analysis of multiple nodes system using real scientific applications and the HPCG Benchmark.

✉ Ryusuke Egawa
egawa@tohoku.ac.jp

Kazuhiko Komatsu
komatsu@tohoku.ac.jp

Shintaro Momose
s-momose@ak.jp.nec.com

Yoko Isobe
y-isobe@pi.jp.nec.com

Akihiro Musa
a-musa@bq.jp.nec.com

Hiroyuki Takizawa
takizawa@tohoku.ac.jp

Hiroaki Kobayashi
koba@tohoku.ac.jp

[1] Cyberscience Center, Tohoku University, Sendai 980-8578, Japan

circumstance, a vector-parallel supercomputer SX-ACE has been designed to achieve a high sustained performance of memory-intensive applications by providing a high memory bandwidth commensurate with its high computational capability. This paper examines the potential of the modern vector-parallel supercomputer through the performance evaluation of SX-ACE using practical engineering and scientific applications. To improve the sustained simulation performances of practical applications, SX-ACE adopts an advanced memory subsystem with several new architectural features. This paper discusses how these features, such as MSHR, a large on-chip memory, and novel vector processing mechanisms, are beneficial to achieve a high sustained performance for large-scale engineering and scientific simulations. Evaluation results clearly indicate that the high sustained memory performance per core enables the modern vector supercomputer to achieve outstanding performances that are unreachable by simply increasing the number of fine-grain scalar processor cores. This paper also discusses the performance of the HPCG benchmark to evaluate the potentials of supercomputers with balanced memory and computational performance against heterogeneous and cutting-edge scalar parallel systems.

Keywords Vector architecture · Memory-intensive applications · Memory bandwidth · Sustained performance

1 Introduction

Nowadays, supercomputers have become requisite facilities to accelerate various kinds of simulations in sciences, engineering and economics fields. To satisfy ever-increasing demands of computational scientists for a higher computational capability, the peak performance of a supercomputer has drastically been improved. Thanks to the technology scaling and the maturation of many core architectures including accelerators, the theoretical peak performance of the world's fastest supercomputer achieves 125 petaflop/s (Pflop/s) [1]. However, mainly due to the memory wall problem and overhead to handle massive parallelism of the system, there is a big gap between the theoretical and sustained performances of a recent supercomputer for practical applications. Thus, it is getting harder to fully exploit the potential of these systems, and only a limited number of applications can reap the benefit of their extremely high theoretical performance. As clearly demonstrated in the previous research efforts [2], keeping a high ratio of a memory bandwidth to a high floating-point operation (flop/s) ratio, known as Bytes per Flop ratio; B/F ratio, of a supercomputer is a key factor to achieve a high sustained performance [3,4].

However, B/F ratios of current supercomputers have steeply been dropping as shown in Fig. 1. Figure 1 shows the B/F ratios of the No.1 systems in the past 15 years. Note that the B/F ratios of heterogeneous systems are obtained by the peak performance and local memory bandwidths of accelerators/GPUs. The No.1 system in 2002, Earth

2 GSIS, Tohoku University, Sendai 980-8578, Japan

3 NEC Corporation, Tokyo 108-8001, Japan

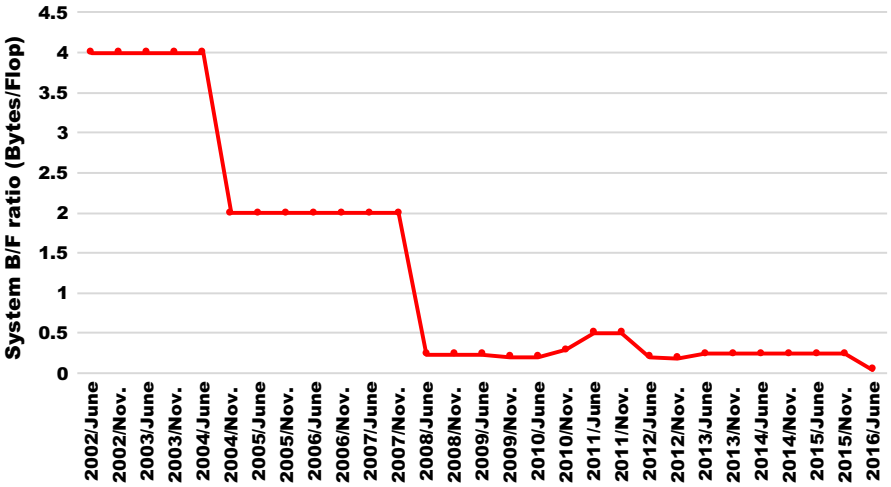

Fig. 1 BF ratios of Top No.1 Supercomputers for past 15 years

Simulator, had a B/F ratio of 4 but that of Sunway Taihulight, which is the No.1 system in 2016 is less than 0.05 [2,5]. It is mainly due to the limitations of the chip size, the number of I/Os integrated on a chip, and their I/O bandwidths.

Furthermore, the I/O logic of a chip consumes huge amounts of area and power compared with logic circuits. Because of the strong requirements for a high theoretical peak performance, recent microprocessors for supercomputers spend given silicon and power budgets to enhance the computational performance rather than the memory bandwidth. Therefore, under a tight power restriction, the current major supercomputers are designed to gain higher aggregated system performances with a huge number of cores at the expense of their memory bandwidths, which is appropriate only for a limited number of computation-intensive applications in addition to the High Performance LINPACK benchmark (HPL) [1].

Since the single core performance has not been improved drastically, the performance improvement of a modern supercomputer has been achieved mainly by increasing the number of cores in the system. However, this approach will reach a limit in the near future because of the following two reasons. One is that it is difficult to extract more parallelism from practical codes consisting of various kernels with different performance characteristics, even though increasing the number of cores requires more massive parallelism to exploit their potentials. The other reason is that the fabrication cost per transistor has begun to increase. Although technology scaling will continue for a while, it will not drastically reduce the price per core, differently from the past. Due to the cost and power limitations, it could be unaffordable to build a larger inefficient system to achieve a certain level of sustained performance.

As described above, there are difficulties in improving the sustained performance by just adding more cores to the system. Amdahl's well-known law implies that, if two systems have the same peak performance, the system with more powerful cores can achieve a higher sustained performance. To this end, a vector-parallel supercomputer

SX-ACE is launched to the market, and several supercomputer centers have started the operation of SX-ACE systems. Inheriting the advantages of conventional vector processors, the SX-ACE processor of four powerful vector cores provides a high memory bandwidth commensurate with its high computational capability.

The SX-ACE processor is designed so as to satisfy the following two requirements for a higher sustained performance under the limited power and silicon budgets. One requirement is to achieve a high sustained memory bandwidth for accelerating memory-intensive applications, and the other is to achieve a high single core performance for obtaining a certain level of sustained performance with fewer cores. In addition, SX-ACE is designed to overcome the drawbacks of conventional vector supercomputers by introducing several architectural features. A large capacity of Assignable Data Buffer (ADB) with Miss Status Handling Register (MSHR) [6] is implemented to avoid redundant data transfers for vector load operations by holding reusable data on a chip, and thus to make full use of high memory bandwidth of SX-ACE. Furthermore, SX-ACE can efficiently execute an application even if the application needs short vector calculations and/or indirect memory accesses. As a result, SX-ACE can accelerate practical memory-intensive applications in various research fields.

Aiming to examine the potentials of the modern vector supercomputer, this paper discusses the sustained performance of SX-ACE for practical scientific applications. Especially, the contributions of the new features of SX-ACE to the sustained performance are evaluated in detail. The performance of SX-ACE is compared with those of other major supercomputers equipped with modern scalar processors. In addition to these evaluations, to clarify the importance of a high sustained memory bandwidth and power efficiency of the system, the performance of the High Performance Conjugate Gradients (HPCG) benchmark [7] is also discussed. These evaluation results clearly indicate that the high sustained performance per core enables the SX-ACE system to achieve outstanding performances that are unreachable by simply increasing the number of fine-grain scalar processor cores.

The rest of this paper is organized as follows. Section 2 presents the system architecture of SX-ACE from the processor to the overall system. Section 3 evaluates the new features of SX-ACE using several standard benchmark programs. Section 4 discusses the sustained performances and scalabilities of seven practical applications and the HPCG benchmark on SX-ACE. Section 5 concludes this paper.

2 An overview of the SX-ACE vector supercomputer

The basic configuration of the SX-ACE supercomputer is composed of up to 512 nodes connected via a custom interconnect network. Each node of the SX-ACE system consists of one processor and several memory modules, and the processor of four powerful cores can provide a double-precision floating-point operating ratio of 256 Gflop/s, a memory bandwidth of 256 GB/s, and a memory capacity of 64 GB. Thanks to this powerful node, the aggregated performance and memory bandwidth of the 512-node configuration reach 131 Tflop/s and 131 TB/s, respectively. It should be noted that its B/F ratio is 1.0 that is higher than those of other supercomputers. Table 1 shows

Table 1 Specifications of SX-ACE

Core	
Theoretical performance	64 Gflop/s
ADB capacity	1 MB
ADB bandwidth	256 GB/s
Memory bandwidth	64–256 GB/s
B/F ratio	1.0–4.0
CPU	
Number of cores	4
Theoretical performance	256 Gflop/s
Memory bandwidth	256 GB/s
B/F ratio	1.0
Node	
Number of CPUs	1
Memory capacity	64 GB
Multi-node system	
Number of nodes	512
Number of CPUs	512
Number of cores	2048
Theoretical performance	131 Tflop/s
Memory capacity	32 TB
Memory bandwidth	131 TB/sec

the basic specifications of the SX-ACE system. The following subsections describe the details.

2.1 Node and CPU configurations

Figure 2 depicts an overview of the SX-ACE processor. The processor is comprised of four cores, a memory control unit (MCU), a remote access control unit (RCU), and a memory crossbar. Each core is composed of four major parts: a scalar processing unit (SPU), a vector processing unit (VPU), ADB, and MSHR. Each core is connected to MCU at a bandwidth of 256 GB/s through the memory crossbar, and four cores of one processor share the memory bandwidth. Due to this configuration, one core can exclusively utilize the entire memory bandwidth of 256 GB/s if the other three cores do not access the memory. Accordingly, the B/F ratio of each core can change from 1.0 up to 4.0 at maximum. The implementation of such an ample datapath is a key design feature of the SX-ACE processor, which is quite beneficial to high sustained performances of memory-intensive practical applications.

MCU has 16 memory interfaces of DDR3 and is connected to the memory composed of 16 DDR3 DIMMs with a 64 GB capacity through these interfaces. The memory is accessible from each core with a 128 B granularity. In order to improve the sustained memory bandwidth for indirect memory accesses and the performance of short vector

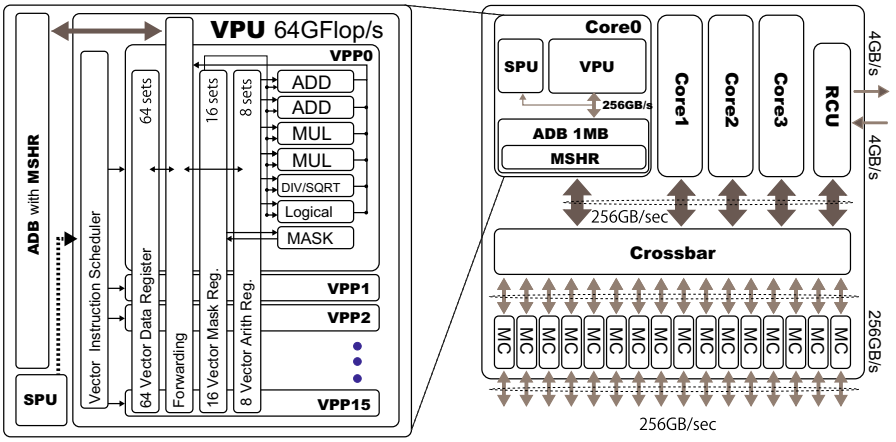

Fig. 2 SX-ACE processor

operations, the memory access latency is halved in comparison with its predecessor model of SX-9. RCU integrated on the SX-ACE processor is a remote direct memory access engine. RCU is directly connected to the dedicated interconnect network at a transfer rate of 8 GB/s per direction with minimizing communication latencies among nodes, and is also connected to MCU through the memory crossbar.

SPU is a scalar processing unit with a 1 Gflop/s peak performance. The main roles of SPU are decoding all instructions and processing scalar instructions. SPU transfers all vector instructions to VPU, and these vector instructions are issued and processed by VPU. Besides, since the roles of SPU are limited as mentioned above, its power consumption is negligibly small compared to other units.

VPU is a key component of the SX-ACE vector architecture with its single core performance of 64 Gflop/s. As well as previous SX series, SX-ACE can process up to 256 vector elements, 8 B each, by a single vector instruction. VPU can process 256 operations by a single instruction, and they are performed in 16 cycles using 16 vector pipelines in a SIMD (single instruction multiple data) manner. This is one of the advantages in hiding long operation latencies, and differentiates the SX vector architecture from other conventional SIMD architectures adopted into modern scalar processors and accelerators. For example, the most recent accelerator Xeon Phi is equipped with a large number of cores, and each core can process up to eight vector elements at one cycle [8]. Xeon Phi is designed so as to achieve a high performance by exploiting high thread-level parallelism using many cores. Although SX-ACE and Xeon Phi employ SIMD/vector processing mechanisms, their design concepts and architecture are different.

Furthermore, in order to accelerate short vector operations that generally inhibit such a latency hiding feature, a data forwarding latency between vector pipelines in VPU is shortened by approximately 40%. This reduction is realized by implementing a bypass mechanism among vector pipelines. An issue rate of the vector instruction is also enhanced to provide vector instructions to VPU at a sufficient rate to minimize vector pipeline stalls even in short vector operations.

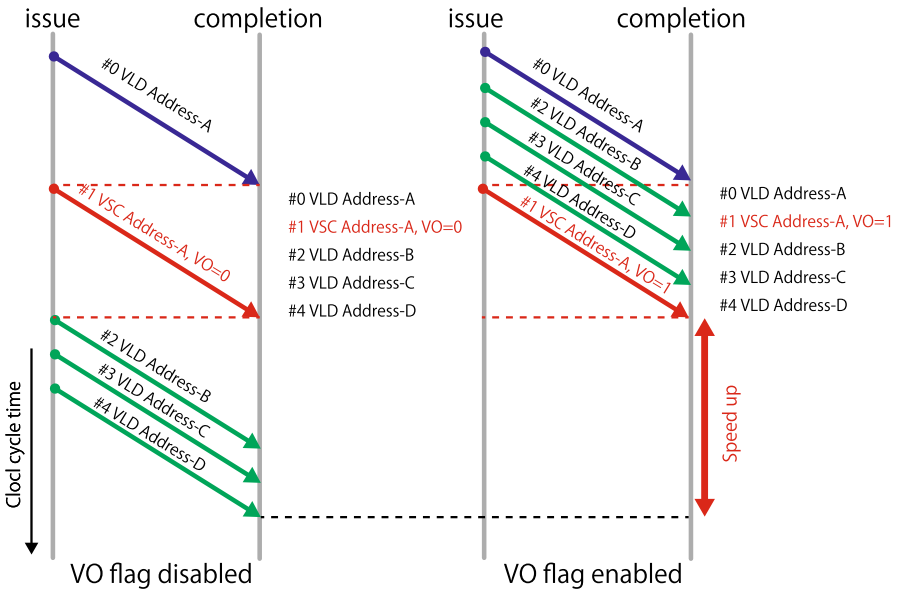

Fig. 3 Behavior of an out-of-order vector memory accesses instruction

Each core can also issue memory access instructions for any set of 256 elements. These vector instructions are reordered by checking their data dependencies, so as to minimize their stalls that often hamper effective use of the memory bandwidth. For reordering, a vector overtaking (VO) flag is newly added to vector store and scatter instructions. The VO flag is set automatically by NEC's compiler or by manually inserting directives into a code. The vector store or scatter instruction with the VO flag can be overtaken by subsequent vector and/or gather instructions without checking data dependencies among these instructions. Figure 3 shows the behavior of the out-of-order memory access with the VO flag. Since the VO flag is given to the vector scatter (VSC) operation #1, its subsequent three vector load instructions can overtake VSC operation #1. As a result, the SX-ACE processor can effectively use the given memory bandwidth, resulting in a decrease in execution cycles.

Each core can access data in the 1 MB ADB at the rate of 256 GB/s. ADB is a software controllable private cache with 4 ways and 16 banks. While the features of ADB are similar to those of conventional caches, ADB can selectively store reusable data or explicitly prevent non-reusable data from being stored. These functions can be controlled by a programmer or the SX compiler. By manually inserting directives into a source code, programmers can specify whether data are reusable or not. Besides, since the SX compiler appropriately detects the reusable data and stores them to ADB, ADB can work as a conventional cache memory. Then ADB retains only reusable data, resulting in a higher ADB hit rate and a decrease in memory transactions. Note that ADB can directly be accessed by VPU at the word-level granularity. Therefore, ADB enables efficient random memory accesses, such as stride and indirect access patterns, resulting in a higher sustained bandwidth for such irregular memory accesses.

MSHR is also implemented to achieve a higher sustained memory bandwidth by reducing the number of redundant memory transactions [9]. MSHR withholds identical load requests with in-flight load requests on ADB misses, and avoids redundant memory requests if the subsequent memory requests that cause ADB misses can be solved by the in-flight load requests. Avoidance of redundant load requests allows the memory bandwidth to be utilized efficiently.

Moreover, SX-ACE has a potential to achieve a 10× higher performance per watt ratio than SX-9. This improvement in power is mainly provided by redesigning of the memory system of SX-ACE. Since SX-9 employs a large-scale SMT node with a 1TB shared memory, it requires power-hungry modules such as an enormous number of DRAMs and I/O pins, custom-made memory interfaces, and custom network controllers. However, as introduced in the previous section, the node of SX-ACE just consists of a single CPU and 64GB DDR3 memory. Therefore, SX-ACE can successfully reduce the number of I/O pins by employing the standard DDR3, and then can remove power-hungry modules of the memory system. Furthermore, since the activation power of DRAM strongly depends on the cache line size, SX-ACE employs 128-Byte cache line size by default to reduce the power consumption while keeping a high sustained memory bandwidth.

Besides all that, SX-ACE is designed so as to obtain a high sustained performance by enhancing memory functions and performance, rather than just improving theoretical peak performance. This design decision achieves balanced memory and computational performance and certainly improves the sustained performance of memory-intensive applications. Hence, in addition to the benefit of technologies scaling, SX-ACE successfully improves its power efficiency [10].

2.2 Multi-node system

The nodes of SX-ACE are connected by a custom interconnect network with an 8 GB/s bandwidth per direction, and the network is composed of the internode crossbar switch (IXS) and RCU. IXS is made up of two independent fat-tree topological planes, which are configured by dedicated router (RTR) switches and cables. Since the network is organized by a fat-tree topology, a unique path to a target node can be identified when the destination is decided. If a conflict occurs at any port, dedicated buffers provided on each port will hold data packets until the port becomes available. Hence, IXS does not have complicated routing control and congestion management mechanisms. In each IXS plane, 16 nodes are connected to one edge RTR, and all the edge RTRs are connected to all spine RTRs as shown in Fig. 4. Communications among nodes are executed by using two IXS planes. In order to accelerate internode collective communications, such as an internode global barrier synchronization, special hardware mechanisms such as global communication registers/counters are built in IXS.

3 Evaluation of the SX-ACE processor using standard benchmark programs

As mentioned in the previous section, the SX-ACE processor introduces several functions to overcome the drawbacks of current supercomputing systems including SX-9.

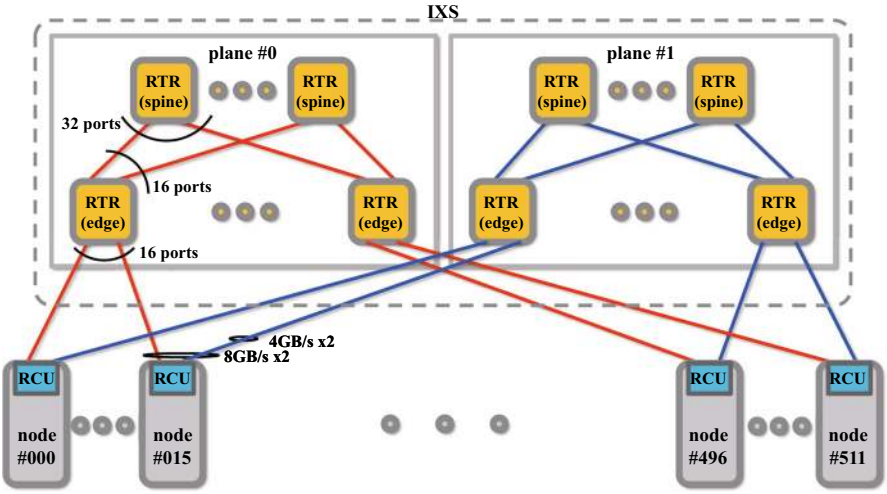

Fig. 4 Multiple nodes configuration of SX-ACE

In this section, the potentials of the SX-ACE processor are evaluated using standard benchmark programs. SX-ACE and its compiler are used to evaluate sustained performances.

3.1 Sustained data transfer performance

First, the STREAM (TRIAD) benchmark program is used to evaluate the data transfer performance of SX-ACE. The STREAM benchmark program evaluates the sustained memory bandwidth by using arrays whose sizes are much larger than the capacity of the last level cache [11]. The sustained bandwidths of SX-ACE as a function of the number of threads on a processor are shown in Fig. 5. The number of cores integrated in each processor limits the maximum number of threads. For comparison, the sustained bandwidths of representative scalar processors of NEC LX 406(Ivy Bridge), Fujitsu FX10(SPARC64 IXfx), and Hitachi SR16000M1(Power7) are also shown in Fig. 5. Here, the peak memory bandwidth and the number of cores of SX-ACE, LX406, FX10, and SR16000M1 are 256 GB/s with four cores, 59.7 GB/s with 12 cores, 85 GB/s with 16 cores, and 128 GB/s with eight cores, respectively. The detailed specifications of these scalar processors are described in Sect. 4.

The STREAM memory bandwidth of the SX-ACE processor reaches 220 GB/s even in the single thread case, and is almost constant irrespective of the number of threads. On the other hand, although the STREAM bandwidths of the other systems gradually increase with the number of threads, scalar-based systems cannot effectively use the theoretical peak memory bandwidths when the number of threads is small. We think that each single core in these scalar processors doe not have sufficient bandwidth between memory interfaces and cores, and cannot issue memory requests at a sufficiently high rate to extract their theoretical memory bandwidth. As a result, these processors only use around 20% of the peak memory bandwidth in the single

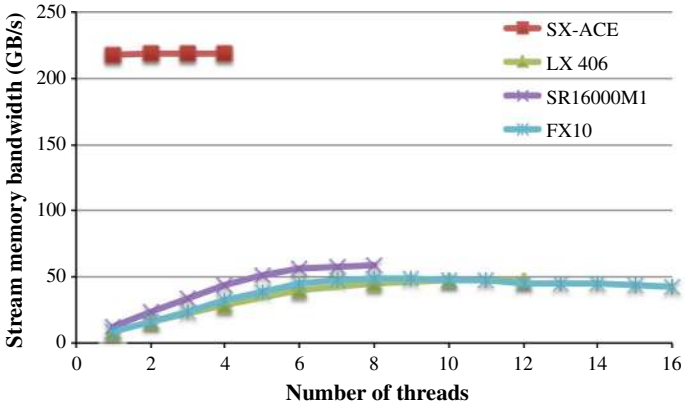

Fig. 5 Stream memory bandwidth

core cases. Since SX-ACE maintains high memory bandwidths in all the cases, practical memory-intensive applications, which consist of various kernels with individual characteristics, can be processed at a high throughput.

3.2 Effects of ADB with MSHR

SX-ACE shows a higher system B/F ratio than the scalar-based systems. However, due to the limitations of current chip technologies and power/cost constraints, the B/F ratio of SX-ACE is just 1.0, while that of SX-9 is 2.5 [4].

To compensate for the dropped system B/F ratio, SX-ACE introduces advanced mechanisms, such as an enlarged ADB with MSHR, to the memory subsystem. This subsection discusses the potentials of a larger capability of ADB with MSHR using the Himeno benchmark with the XL data set ($1024 \times 512 \times 512$) [12]. The Himeno benchmark measures the performance in solving the Poisson equation with the Jacobi iterative method, which is highly memory-intensive. Although a user can manually specify reusable data by using a directive, the SX compiler has a capability to automatically find out reusable data within the code and selectively store them in ADB.

Figure 6 shows the evaluation results in the cases of ”without ADB and MSHR” by disabling both ADB and MSHR, “with MSHR” by disabling only ADB, and “with ADB and MSHR” by enabling both ADB and MSHR. The vertical axis indicates the sustained performance in Gflop/s of the Himeno benchmark. In comparison with the performance without using both ADB and MSHR, use of only MSHR increases the performance by approximately 40%. Use of both ADB and MSHR doubles the performance.

To analyze the reasons behind the performance improvements, Fig. 7 shows the hit rates of ADB with MSHR, the code B/F ratios, and the actual B/F ratios of all the cases. Here, the code B/F ratio is defined as the necessary data in bytes per floating-point operation. The code B/F ratio can be obtained by counting the numbers of memory operations and floating-point operations in the object codes. To achieve a high

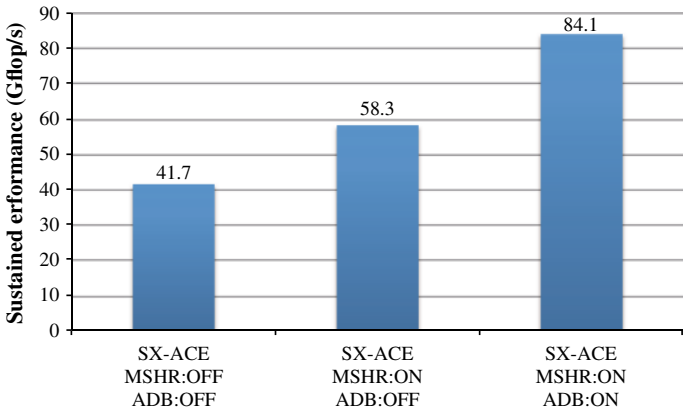

Fig. 6 Effects of ADB and MSHR

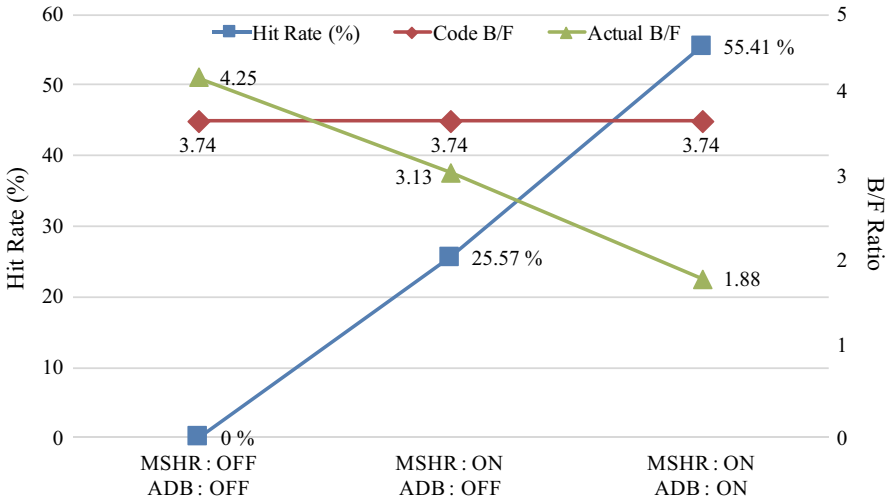

Fig. 7 ADB hit rates and actual B/F ratio

memory bandwidth, recent processors employ the block data transfer. Since the block data transfer reads and writes data by the block size, the block contains unnecessary data unless the program accesses a continuous memory region aligned to the block boundary. Unnecessary data are also contained within the blocks in the cases of stride and indirect memory access patterns. In those cases, the number of transferred data is increased compared with that considered in the code B/F ratio. Therefore, the actual B/F ratio given by Eq. (1) indicates the ratio of the sustained memory bandwidth to the sustained floating-point operation ratio, and hence reflects the actual behavior of each system.

$$Actual\ B/F = \frac{N_{block\ mem} \times D_{size}}{Flops}, \tag{1}$$

where $N_{block\ mem}$ indicates the actual number of block memory accesses. Here, if the data are loaded from ADB, $N_{block\ mem}$ decreases. Hence, detail profiling data that include ADB or cache hit rates are needed to obtain $N_{block\ mem}$. D_{size} denotes the unit of block data transfers, and $Flops$ indicates the number of floating-point operations.

Since the kernel of the Himeno benchmark has many misaligned memory accesses, it cases a lot of ADB misses and the number of block accesses increases. Then, the actual B/F ratio of the "without ADB and MSHR" case becomes larger than its code B/F ratio. As shown in Fig. 7, both MSHR and ADB effectively reduce the actual B/F ratio by storing reusable data in ADB and reducing block memory accesses. These results demonstrate that the sustained performance of real memory-intensive applications can be boosted by collaboration of ADB and MSHR.

3.3 Performance of indirect memory accesses

As VPU can handle 256 operations with a single vector instruction, VPU has advantages in hiding long operation latencies. Therefore, the SX vector architecture is differentiated from the conventional SIMD architecture. To fully exploit the potential of VPUs, as described in Sect. 2, SX-ACE has out-of-order memory access functions by using VO flags through hardware and software controls that can improve the sustained memory bandwidth, as well as ADB with MSHR.

In the SX-ACE processor, indirect memory accesses usually degrade the sustained memory bandwidth because subsequent memory access instructions have to wait until completion of the precedent indirect memory accesses. However, since run-time memory disambiguation for 256 indirect memory accesses just using hardware control is quite challenging, this disambiguation is performed by software control. The VO flag is introduced into vector store and scatter instructions in order to enhance the reordering capability even in the case of indirect memory accesses. The VO flag is set to an instruction, such as vector store/scatter instruction, only if it does not depend on its subsequent vector load/gather accesses. The dependency among vector memory instructions is automatically analyzed by the compiler or is manually provided by a directive in the code.

The sustained performance of indirect memory accesses is evaluated by using a Legendre transformation program [13], which needs frequent indirect memory accesses. Figure 8 shows that SX-ACE provides a 6.5× higher performance and a 2.6× higher efficiency than those of the SX-9, respectively. In this graph and hereafter, the efficiency is defined as the ratio of the sustained performance to the peak performance. The out-of-order memory access mechanism, the shorter memory access latency, and the higher issue capability of SX-ACE for memory access instructions can accelerate indirect accesses to the memory even though SX-ACE and SX-9 processors have the same memory bandwidth of 256 GB/s.

3.4 Performance of shorter vector processing

As the characteristics of practical scientific applications are getting more diverse, the vector lengths of kernel loops in practical applications widely vary. Since conventional

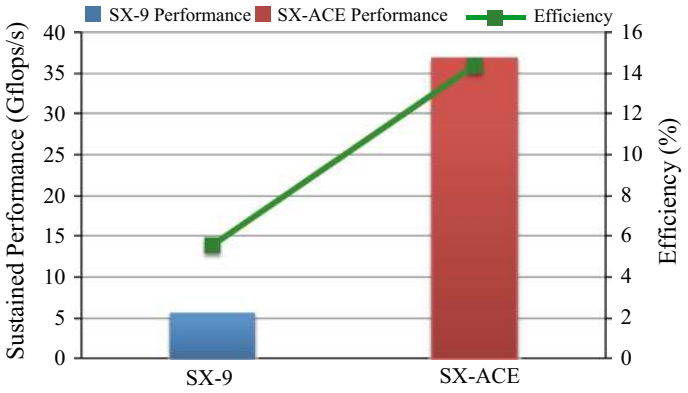

Fig. 8 Performance of indirect memory access

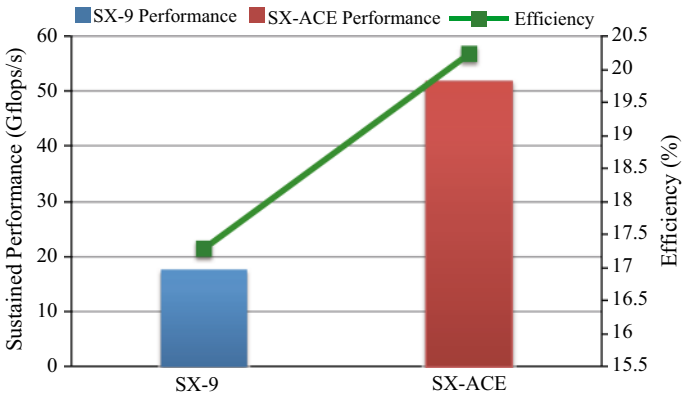

Fig. 9 Performance of short vector processing

vector processors require a longer vector length to hide the long memory access latency by deeper pipelines with 256 vector elements, they cannot efficiently handle short vector operations whose vector lengths are much shorter than 256. SX-ACE improves the performance of short vector processing by halving the memory access latency of SX-9. This reduction in the latency is achieved by employing a simple memory network configuration with a smaller memory capacity than SX-9. Moreover, SX-ACE has the advanced data forwarding function of vector pipelines described in Sect. 2.

The performance of the SX-ACE processor for short vector operations is evaluated using a reverse Legendre transformation program that requires short vector processing. Figure 9 shows the evaluation results in terms of the sustained performance and the computational efficiency. As SX-9 and SX-ACE can perform 256 elements of vector calculations at the same time, 320 elements of the main kernel, whose calculations are dominant in the kernel, are divided into 256 elements and 64 elements. Since SX-ACE can process the 64 elements efficiently by virtue of the vector data forwarding, the short memory access latency, and the enhanced vector instruction issue, SX-ACE can achieve about 2.9× higher sustained performance than SX-9.

4 Performance analysis of SX-ACE using practical applications

4.1 Experimental environments

In this section, the performance of SX-ACE is evaluated by using seven practical applications. For comparison, the performances of various supercomputers, which are listed in Table 2, are also evaluated. SX-9 is a vector parallel supercomputer consisting of large symmetric multi-processing nodes, each of which has 16 vector processors with a 4096 GB/s memory bandwidth. The effective use of a 256 KB ADB is a key to exploit the potential of SX-9 [4]. As the main memory and ADB can simultaneously provide data to vector pipelines, the vector processor can access those data at a high sustained bandwidth.

NEC LX 406 and Hitachi SR16000M1 are scalar parallel supercomputers that are equipped with Intel Ivy Bridge and IBM Power7 processors, respectively. As shown in Table 2, these scalar processors also have large on-chip cache memories. On-chip L2 and/or L3 caches should be used for data with high locality to hide memory access latencies and to reduce the number of memory accesses. This evaluation is carried out by only using compiler optimizations without code modifications for individual systems.

Table 2 Specifications of HPC systems used in the evaluations

	System			Node	CPU	
	Tflop/s	#Nodes	NW GB/s	#CPUs	Gflop/s	#Cores
SX-ACE	131.1	512	2 × 8 IXS	1	256	4
SX-9	26.2	16	2 × 128 IXS	16	102.4	1
ES2(SX-9)	104.9	128	2 × 64 IXS	8	102.4	1
LX 406	29.4	64	5 IB	2	230.4	12
(Ivy Bridge)						
Fujitsu FX10	0.24	1	5-50 Tofu NW	1	236.5	16
SR16000M1	62.7	2 × 24-96	64	4	245.1	8
(Power7)			Custom NW			

	CPU			Core
	Mem. GB/s	On-Chip Mem.	System B/F	Gflop/s
SX-ACE	256	1MB ADB/core	1.0	64
SX-9	256	256KB ADB/core	2.5	102.4
ES2(SX-9)	256	256KB ADB/core	2.5	102.4
LX 406	59.7	256KB L2/core	0.26	19.2
(Ivy Bridge)		30MB shared L3		
Fujitsu FX10	85	12MB shared L2	0.36	14.78
SR16000M1	128	256KB L2/core	0.52	30.6
(Power7)		32MB shared L3		

Table 3 Evaluated applications

Applications	Method	Memory access characteristic
Barotropic ocean	Shallow water model	Sequential memory access
BCM	Navier–Stokes equation	Indirect memory access
MHD(FDM)	Finite difference method	Sequential memory access
MHD(Spectral)	Pseudospectral method	Stride memory access
QSFDM GLOBE	Spherical 2.5D FDM	Sequential memory access
TURBINE	DNS	Indirect memory access
Seism3D	Finite difference method	Sequential memory access

Applications	Mesh size	Code B/F
Barotropic ocean	4322×2160	1.97
BCM	$128 \times 128 \times 128 \times 64$	7.01
MHD(FDM)	$2000 \times 1920 \times 32$	3.04
MHD(Spectral)	$900 \times 768 \times 96$	2.21
	$3600 \times 3072 \times 2048$ (multi-node)	
QSFDM GLOBE	4.3×10^7 grids	2.16
TURBINE	$91 \times 91 \times 91 \times 13$	1.78
Seism3D	$1024 \times 512 \times 512$	2.15
	$4096 \times 2048 \times 2048$ (multi-node)	

Table 3 shows the outlines and memory access characteristics of seven practical applications. The code B/F ratios of the applications in the table are obtained from the object codes of SX-ACE. These codes are designed and developed by independent researchers.

Barotropic ocean This simulation code numerically models the global ocean mass variations forced by atmospheric pressure and wind on the sea surface [14]. The code is implemented by the finite difference method for the partial differential equation of a single-layer ocean. The time integration is done using an explicit method.

BCM (Building-Cube Method) This code is a block-structured Cartesian-mesh Computational Fluid Dynamics solver [15]. The BCM has been developed to quickly evaluate the aerodynamic performance and flows of real-world complicated geometries using large parallel computers. The fractional step method is employed to solve the incompressible Navier–Stokes equations with a higher-order scheme by making use of the advantage of Cartesian meshes.

MHD_Spectral & MHD_FDM These codes are distributed-memory parallel implementation for a direct numerical simulation (DNS) of the magneto-hydro-dynamic (MHD) turbulent channel flows, and they have aimed to investigate the MHD pressure loss and heat transfer characteristics of the high-Reynolds number and the high-Prandtl number fluid such as the molten salt FLiBe. Governing equations of these codes are the continuity equations, the incompressible Navier–Stokes equations with the electric

field, and the energy equations. With the second-order central differencing method, the MHD_Spectral code is computed by a hybrid Fourier spectral [16,17], and the MHD_FDM code is computed by the 12th order accurate finite difference method [18].

QSFDM_GLOBE This code simulates global seismic wave fields generated by a moment-tensor point source [19]. It solves a 3D wave equation in spherical coordinates on a 2D structural cross section, assuming that the structure is rotationally symmetric about the vertical axis including the source, so called axisymmetric 2.5D modeling. This code uses the finite difference method for discretization.

TURBINE This code is a numerical simulation of unsteady 3D flows of wet steam through turbine multi-stage stator-rotor channels [20]. The numerical integration is solved by the LU-SGS method, and the space finite difference is solved by Roe's flux difference splitting method and the fourth-order accurate compact MUSCL TVD scheme. The viscous term is the second-order central difference. The turbulence model is Shear Stress Transport.

Seism 3D This code is a 3D simulation of seismic wave propagations on a regional scale including higher frequencies [21]. It solves the equations of 3D motion and the constitutive relationship between stress and strain of the wavefield, using a parallel finite difference method.

4.2 Performance evaluation of the SX-ACE processor

Figures 10 and 11 show the sustained performances and efficiencies of the SX-ACE processor by changing the number of threads, respectively. Figure 10 shows that the sustained performances of the practical applications increase with the number of threads. As the aggregated peak performance increases with the number of threads, the sustained performance becomes the highest when all the four cores are used for the execution.

Figure 11 indicates that the efficiencies decrease as the number of threads increases. Since four cores of one processor share the memory bandwidth of 256GB/s as

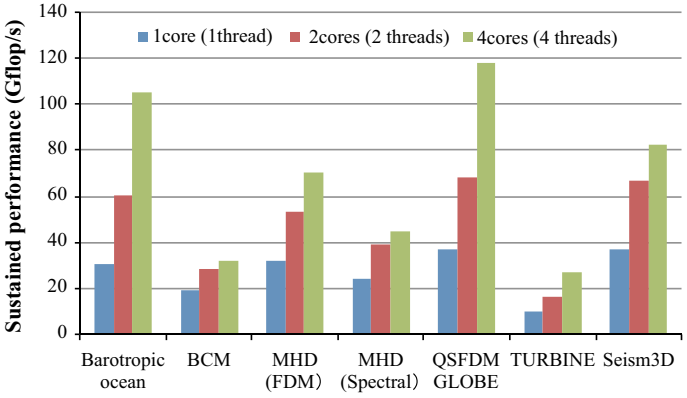

Fig. 10 Performance of the practical applications on the SX-ACE processor

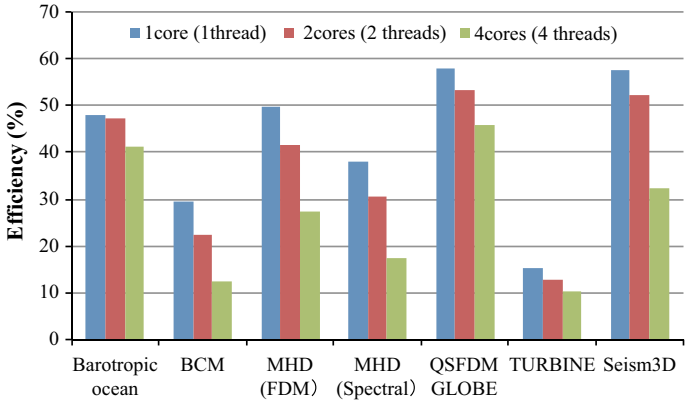

Fig. 11 Efficiencies of the SX-ACE processor

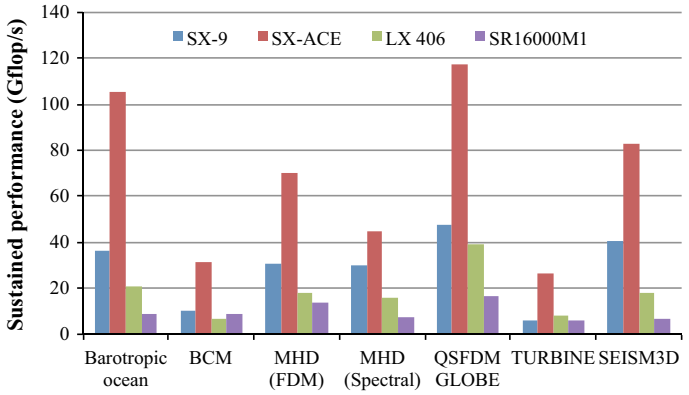

Fig. 12 Comparison in a single CPU performance among multiple supercomputers

described in Sect. 2.1, the memory bandwidth per core decreases as the number of activated cores increases. Therefore, the system B/F ratio decreases as the number of threads increases. In particular, the efficiencies of the memory-intensive applications whose actual B/F ratios are high drastically degrade. For example, as the actual B/F ratio of BCM is high, its efficiency decreases by about 68% when the number of threads changes from one to four. Thus, it is important to keep the system B/F ratio high to achieve high efficiency especially for the memory-intensive application.

Figure 12 shows the single processor performances of different supercomputers for practical applications. The horizontal axis indicates the practical applications used in the evaluation. The vertical axis indicates the sustained performances of a processor on each supercomputer. From Fig. 12, it is clarified that the SX-ACE processor achieves the highest performance among processors of all the evaluated systems. This is due to the high sustained memory bandwidth of SX-ACE compared with the scalar-based systems. Since the actual B/F ratios of the applications are high, the memory performance of each system greatly affects the sustained performance of the applications.

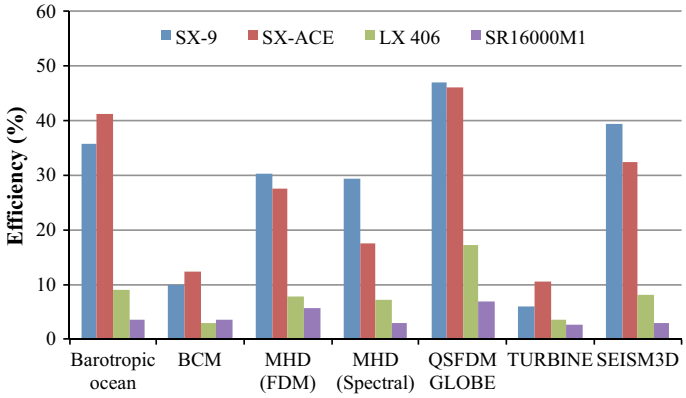

Fig. 13 Comparison in efficiencies of single CPU among multiple supercomputers

Thus, SX-ACE can achieve about 3.9× and 7.2× higher sustained performances than those of LX 406 and SR16000M1, respectively.

Compared with the SX-9 processor, SX-ACE achieves a higher sustained performance even though the memory bandwidth of SX-9 is the same as that of SX-ACE. This is because the new features of SX-ACE could alleviate the high B/F ratio requirements in addition to the higher peak performance.

Figure 13 shows the efficiencies of the practical applications on each supercomputing system. This figure shows that the vector processors, SX-ACE and SX-9, achieve higher efficiencies than the scalar processors. The high system B/F ratios of the vector processors bring the high efficiencies. Furthermore, the efficiencies of SX-ACE are comparable to those of SX-9 even though the system B/F ratio of SX-ACE is 40% lower than that of SX-9. Since the new features of SX-ACE contribute to the high sustained performance, the efficiencies of SX-ACE remain almost the same as those of SX-9. In particular, in the cases of Barotropic ocean, BCM, and TURBINE, the efficiencies are even higher than those of SX-9. To analyze the high efficiencies, the new features of SX-ACE such as the large-capacity ADB with MSHR, the high-speed indirect memory accesses, and the enhanced short vector processing are further discussed by using these applications.

Taking BCM as an example, the effects of the enhanced ADB with MSHR on the performance and efficiencies are discussed. Figure 14 indicates that ADB and MSHR can mitigate the high B/F requirements of BCM. By enabling both ADB and MSHR, the sustained performance increases with the number of threads. This is because both of them can reduce the amount of data transferred from/to the memory, and hence the actual B/F ratio decreases. As the hit rate of ADB with MSHR is about 47.1%, the actual B/F ratio required for BCM can be reduced from 11.07 to 5.86, resulting in the performance improvements. In the case where both ADB and MSHR are disabled, the sustained performance hardly increases with the number of threads. This is because indirect memory access patterns of BCM prevent one core from efficiently using the memory bandwidth, even though the code B/F ratio of BCM is high. Two cores can exhaust the memory bandwidth, and thus the sustained performance is slightly

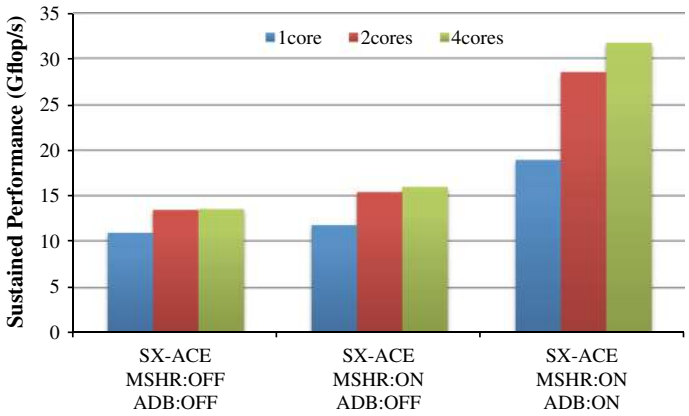

Fig. 14 Performances of BCM

```
DO 200 K=1,KF;
 DO 200 J=1,JF;
  DO 200 L=lstart,lend;
   DO 200 I=IS(L),IT(L)
    DQL =RH(AMINO(I-2,IS(L)),J,K)- RH(AMINO(I-3,IS(L)),J,K)
    DQM = RH(I-1,J,K) - RH(AMINO(I-2,IS(L)),J,K)
    DQN = RH(I,J,K) - RH(I-1,J,K)
    DQP = RH(AMINO(I+1,IT(L)),J,K) - RH(I,J,K)
    DQR = RH(AMINO(I+2,IT(L)),J,K) - RH(AMINO(I+1,IT(L)),J,K)
    ...
    DQLM=SQL*DMAX1(0.0D0,DMIN1(AQL,COEFQ*SQL*DQM,COEFQ*SQL*DQN))
    DQMM=SQM*DMAX1(0.0D0,DMIN1(AQM,COEFQ*SQM*DQL,COEFQ*SQM*DQN))
    DQNM=SQN*DMAX1(0.0D0,DMIN1(AQN,COEFQ*SQN*DQM,COEFQ*SQN*DQL))
    DQNN=SQN*DMAX1(0.0D0,DMIN1(AQN,COEFQ*SQN*DQM,COEFQ*SQN*DQP))
    ...
    DDQM=DQNM-2.0D0*DQMM+DQLM
    DDQN=DQPN-2.0D0*DQNN+DQMN
    ...
      QLL(I,J,K,M)=DQL0+COEFB*DPQM0+COEFC*DMQN0
    QRR(I,J,K,M)=DQR0-COEFB*DMQP0-COEFC*DPQN0
200 CONTINUE
```

Fig. 15 Indirect memory accesses in the TURBINE application

improved by increasing the number of threads from one to two. However, since the memory bandwidth is fully utilized by two cores, the performance does not change by further increasing the number of threads from two to four.

In order to clarify the potential of SX-ACE for the indirect memory accesses, a kernel that includes indirect memory accesses is excerpted from the TURBINE application. Figure 15 shows an indirect memory access part of the kernel. Figure 16

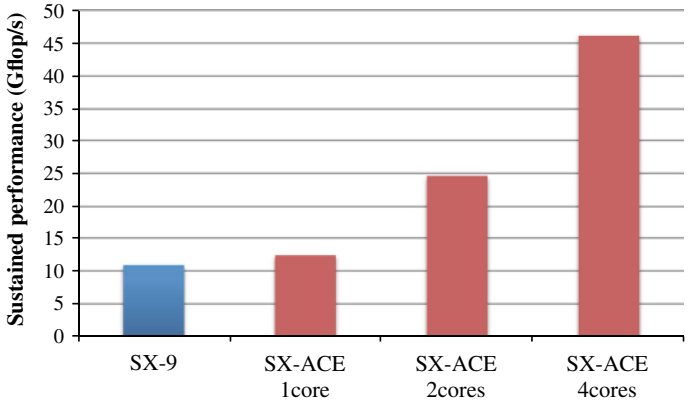

Fig. 16 Performance of indirect memory accesses in TURBINE

shows the sustained performances of SX-ACE and SX-9 for this indirect memory access kernel. In this kernel, the address of *RH* is determined by its first, second, and third elements. In addition, in *RH(AMIN0(I-2,IS(L)),J,K)*, the first elements is decided by the values of *I-2* and *IS(L)*. To effectively load data by the vector load operations, a list of *AMIN0(I-2,IS(L))* is created in advance. Since accesses to *RH(AMIN0(I-2,IS(L)),J,K)* are performed using the list, accesses to this array becomes indirect memory accesses in SX vector supercomputers. This figure shows that one thread of the SX-ACE processor achieves a higher sustained performance than the SX-9 processor, although the peak performance of one core of the SX-ACE processor is just a 60% of the peak performance of the SX-9 processor. This is because SX-ACE can effectively perform indirect memory accesses. As with the results in Sect. 3, the indirect memory access performance is significantly enhanced by the out-of-order memory access function and the short memory access latency.

Moreover, because the data of *RH* in this kernel are reused several times, the enhanced ADB also contributes to the sustained performance improvement with a 13.4% ADB hit rate. Even with a low ADB hit rate, the actual B/F ratio of this kernel becomes 1.30, where the code B/F ratio is 1.48. This fact indicates that ADB effectively improves the sustained memory bandwidth of this kernel. In addition, since the memory bandwidth of the SX-ACE processor is shared by multiple cores, the memory bandwidth available to each core changes from 64 GB/s up to 256 GB/s. Therefore, the effects of ADB become larger when the number of threads increases. As a result, four threads of SX-ACE achieve a 4.22× higher sustained performance than the SX-9 processor.

In order to discuss the performance of SX-ACE in short vector processing, another kernel that needs short vector processing is excerpted from the TURBINE application. Figure 17 shows the sustained performance of the kernel. SX-ACE achieves a higher performance. Even in the case of one thread, SX-ACE achieves comparable performance in spite of the difference in their peak performances. One of the reasons is the short memory access latency. Compared with SX-9, the memory access latency of SX-ACE becomes half. In addition, vector data forwarding between vector pipelines

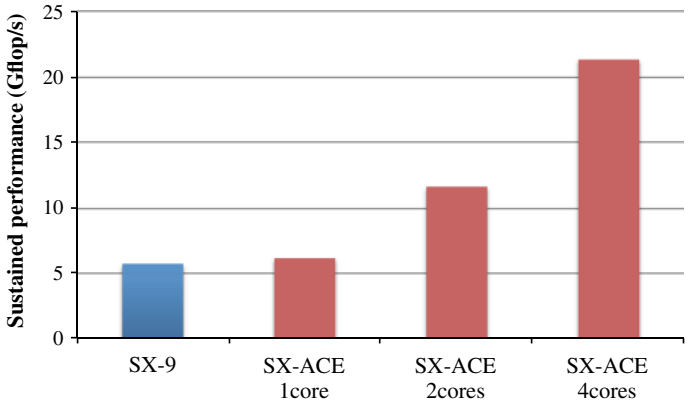

Fig. 17 Performance of short vector processing in TURBINE

in VPU also reduces the latency in vector pipelines. Therefore, SX-ACE can achieve a high sustained performance for the kernel since short vector processing can effectively be performed as discussed in Sect. 3.

4.3 Performance evaluation of multiple nodes of SX-ACE

In order to evaluate multi-node performance of SX-ACE, three applications in Table 3 are used; Barotropic ocean, Seism3D, and MHD_Spectral. Moreover, due to the limitation of the number of nodes of SX-9, ES2 is used in this evaluation. As shown in Table 2, although the SX-9 and ES2 employ the same CPU and memory system, ES2 has half the number of CPUs per node of SX-9. Figure 18 shows the sustained performance of Barotropic ocean by hybrid parallel processing. In the hybrid parallel processing, MPI parallel processing is performed using multiple nodes, and thread parallel processing is performed within one node. The scalability of SX-ACE is better

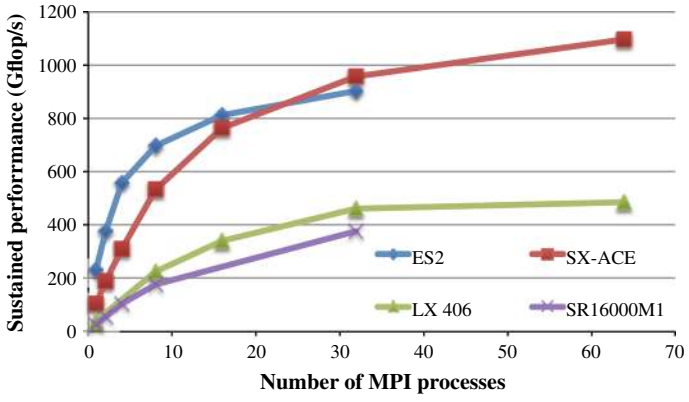

Fig. 18 Performance of hybrid parallelization

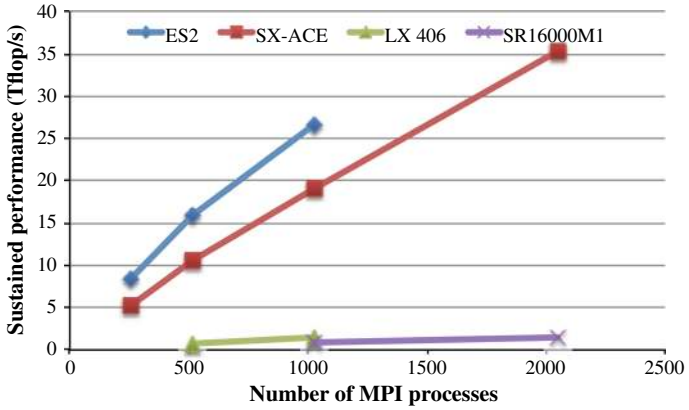

Fig. 19 Performance of flat MPI parallelization

than those of LX 406 and ES2. Since the sustained performance of the SX-ACE processor is much higher than that of the Ivy Bridge processor, which is shown in Fig. 12, SX-ACE achieves outstanding performances that are unreachable by just increasing the number of low-performance scalar nodes.

However, the performance of SX-ACE is lower than that of ES2 when the number of nodes is small. Due to the higher sustained bandwidth per node of ES2, ES2 achieves a higher sustained performance than SX-ACE, in the cases of 24 or less nodes. On the other hand, when the number of nodes is 32 or more, the performance of SX-ACE is superior to that of ES2. Since a scalar processing unit (SPU) of SX-ACE is faster than that of ES2 by virtue of a larger ADB, the performance of SX-ACE becomes higher when the fraction of the serial portion increases after parallel portion can be reduced with a large number of nodes.

Figure 19 shows the sustained performance of Seism3D by flat MPI parallel processing. In this evaluation, although a single socket of LX406 has 12 cores, just eight cores are used. This is because the eight-cores case achieves the highest performance when changing the number of cores in a socket. The result shows that the performances of SX-ACE and ES2 are better than those of the scalar processor based systems. Since this evaluation is carried out by flat MPI parallel processing, the performance is strongly affected by the sustained performance per core as shown in Figs. 10 and 12.

Comparing SX-ACE with ES2, ES2 outperforms SX-ACE if the same number of MPI processes are executed. However, the performance of ES2 does not improve linearly with MPI processes even when the number of MPI processes is 1,024. On the other hand, SX-ACE can still achieve linear speedup with 2,048 MPI processes. Therefore, even in the case of Fig. 19 as well as Fig. 18, the performance of SX-ACE is expected to exceed that of ES2 when the number of MPI processes becomes much larger. This high scalability is important to attain a sustained peta-scale performance for practical simulations.

Figure 20 shows the sustained performance of MHD_Spectral on SX-ACE, SX-9, ES2, and the K computer [22]. The performances of SX-9, ES2 and the K computer are quoted from [17], and the code is optimized for individual systems. The performance

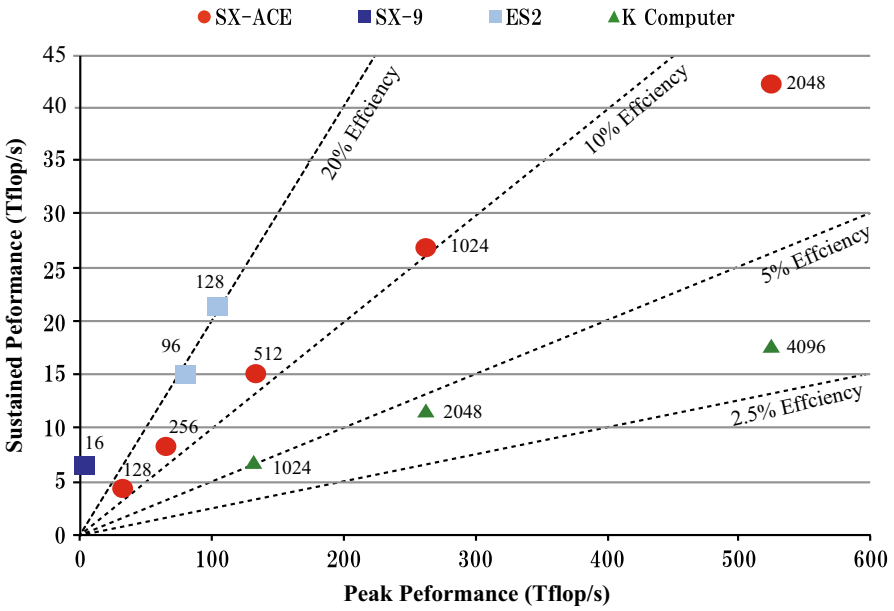

Fig. 20 Sustained performance of MHD in parallel execution

of SX-ACE is newly evaluated in this paper. The code is executed by the hybrid parallel processing. In this figure, the vertical axis indicates the sustained performance. The horizontal axis indicates the aggregated peak performance according to the number of nodes used in the evaluation. The dotted lines indicate the sustained performance corresponding to each efficiency, and the numbers of individual marks denote the number of nodes.

Since this code frequently executes FFTs, high memory and network bandwidths are required. As shown in Table 2, since SX-9 and ES2 have high system B/F ratios and network bandwidth, they can provide higher efficiencies than SX-ACE and the K computer. Besides, they can reach approximately 20% efficiencies that are higher efficiency than those of SX-ACE and the K computer. Comparing SX-ACE with the K computer, SX-ACE achieves higher efficiencies than the K computer due to a high peak node performance of a node and the high system B/F ratio. Even in the case of the number of nodes increases, SX-ACE keeps almost 10 % efficiency. In contrast, the K computer can achieve only 5 % efficiency at a maximum. As a result, SX-ACE realizes more than twice performances than the K computer under the same peak performance. These results suggest the advantage of the high node performance with balanced memory bandwidth to attain high sustained performances.

Although SX-ACE achieves a high sustained performance compared with SX-9 and ES2 by using a larger number of nodes, the efficiency of SX-ACE has been dropped as the number of nodes increases. Since this code frequently executes FFTs that need all-to-all communications among nodes, the internode communication overhead severely limits the scalability. As a result, in the case of 2048 nodes of SX-ACE, the efficiency becomes less than 10%. The main reason of this fact is the lower network bandwidth of

SX-ACE compared with the other systems. To obtain and exploit the high potentials of memory bandwidth and powerful cores, some approaches to communication avoidance should be required for an efficient large-scale simulation.

4.4 HPCG Benchmark Results and Discussions

The performance comparisons among vector and scalar supercomputers have been discussed in this paper. However, heterogeneous supercomputers with accelerators become major players in the Top500 list. Due to the difference in programming models of heterogeneous HPC systems and traditional HPC systems, a quantitative performance comparison among these supercomputers using practical applications is quite challenging. Since High Performance LINPACK benchmark (HPL), which is a highly scalable MPI program with computation-intensive kernels, is used as a benchmark program in the Top 500 list, HPL results become close to the theoretical peak performance of supercomputers [23]. However, the majority of real applications include kernels that are memory-intensive. Thus, the sustained performance of HPL is estranged from those of real applications. To overcome these issues, the HPCG benchmark has been proposed [7] .

In this section, to clarify the potentials of SX-ACE against heterogeneous supercomputers, the performance of HPCG is discussed. The HPCG contains four kernels; DDOT, WAXPBY, SpMV, and MG. Since a sparse matrix-vector multiplication (SpMV) kernel that is the most expensive kernel of HPCG needs a higher B/F ratio than system B/F ratios of current HPC systems as shown in Table 4. In this table, HPCG indicates the total code B/F ratio of this benchmark, and the actual B/F ratios obtained by running HPCG on SX-ACE are also listed. The actual B/F ratios of SpMV, MG, and HPCG on SX-ACE become half of their code B/F ratios by effective usage of ADB, respectively.

Various optimization techniques such as a hyperplane method, selective data caching, and the problem size tuning are introduced to exploit high potential of SX-ACE for HPCG. To eliminate data dependencies for parallelization, the hyperplane method that accesses an array in the diagonal way have been implemented [24]. In addition, to avoid inefficient use of ADB, only highly reusable data are manually stored in ADB by exploiting programmer's knowledge. Moreover, the problem size of HPCG has been tuned considering both the capacity of ADB and the size of the hyperplane. More details of performance tuning can be found in [25].

Figure 21 shows the HPCG results of SX-ACE. The left vertical axis and the bars indicate the sustained performance in Tflop/s, and the right vertical axis and

Table 4 Bytes/flop ratios of HPCG kernels

	DDOT	WAXPBY	SpMV	MG	HPCG
Code B/F	6.64	8.05	12.01	12.08	11.90
Actual B/F	6.64	8.06	6.43	6.38	6.42

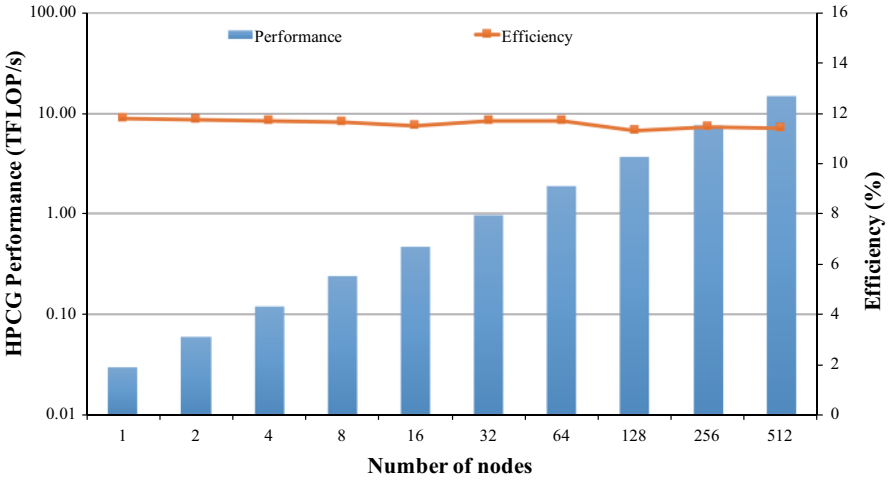

Fig. 21 HPCG performance and efficiency on SX-ACE

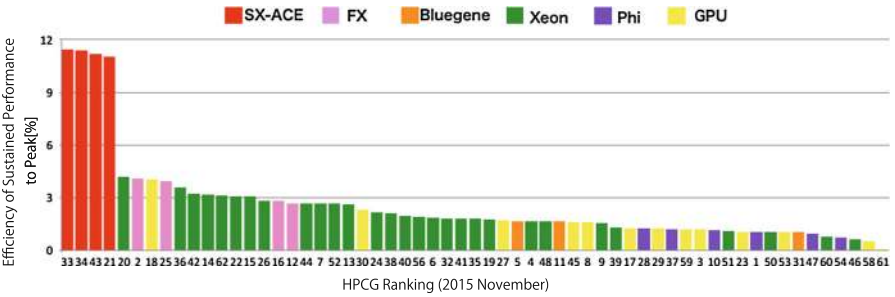

Fig. 22 HPCG efficiency ranking

Table 5 Average HPCG efficiency

	SX	FX	Xeon	Xeon Phi	GPU	Bluegene
Ave. efficiency	11.259	3.372	2.196	1.052	1.452	1.438

the line show the efficiency. From this weak scaling evaluation, it is clarified that the performance of HPCG scales almost linearly according to the number of nodes. Moreover, SX-ACE achieves a high efficiency of 11.4% in the case of 512 nodes, which is much higher efficiency compared with a few percent for the rest of the HPC systems including heterogeneous supercomputers listed in the HPCG ranking.

Figure 22 shows the efficiency of the HPCG on various HPC systems, which are listed in the HPCG ranking, and the systems are sorted by the efficiency. The vertical axis shows the efficiency and the indexes of horizontal axis show the HPCG ranking as of Nov. 2015 [7]. In this figure, HPC systems are classified by processor architectures, SX-ACE, K computer/FX, Xeon, Xeon Phi, GPUs, and Blue Gene, respectively. Table 5 summarizes the average efficiency of individual system architectures. From

Table 6 Power efficiencies of HPCG on SX-ACE and Xeon systems

	Sustained performance (Gflop/s)	Efficiency (%)	Power (W)	Power efficiency (Gflops/W)
HPCG SX-ACE (1 node)	30.2	11.8	234.5	0.128
HPCG SX-ACE (512 nodes)	14,934	11.4	124,800	0.119
HPL SX-ACE (1 node)	246.3	96.2	226.2	1.09
HPL SX-ACE (512 nodes)	123,100	93.9	121,800	1.00

this result, although Xeon Phi and GPUs with a higher peak performance are listed at the upper ranking in the list, we can confirm a relatively low efficiency of Xeon Phi and GPU systems against the other systems.

The main reason is that both of the accelerators (Xeon Phi and GPUs) have an extremely high peak performance, but a relatively low B/F ratio compared with the other systems. Since the performance of HPCG is mainly limited by memory bandwidth of the systems, accelerators cannot feed data to their powerful arithmetic units at a sufficiently high bandwidth.

Next, the power efficiencies of HPCG on SX-ACE are evaluated. Table 6 shows the sustained performance, efficiency, maximum power consumption, and power efficiency of HPCG on SX-ACE. The performance of SX-ACE is evaluated in two cases of single node execution and 512 nodes execution. To analyze the results, evaluation results of HPL are also listed in Table 6. We can confirm that the power consumptions of HPCG and HPL are almost the same even though the sustained performance and efficiency of HPL are nearly eight to nine times higher than those of HPCG. As the power consumption of SX-ACE can be considered constant, the power efficiency becomes almost proportional to the efficiency.

To validate this assumption on another system, we also evaluate performance, efficiency, power, and power efficiency of HPCG and HPL on a single node of an Xeon system. The node has two processors with a 64 GB memory, and each processor has eight cores running at a 3.3 GHz clock frequency. Since the peak performance and memory bandwidth of the node are 422.4 Gflop/s and 118 GB/s, respectively, its B/F ratio is 0.28. The evaluation results are shown in Table 7.

In this case, the difference in power consumption between HPL and HPCG is only 11%, and the power efficiency is nearly proportional to the performance efficiency. Pedretti et al. [26] also present that the power consumptions of HPL and HPCG are almost the same as those of Xeon-based clusters, which consist of 100 nodes without any power control. Although Table 7 shows that SX-ACE achieves a 16% lower power efficiency than the Intel Xeon processor in the HPL case, SX-ACE achieves eight times higher power efficiency in the HPCG case. Since SX-ACE achieves five times higher performance efficiency on average than the Xeon-based systems as shown in Table 5, SX-ACE would perform power efficient processing of memory-intensive applications compared with commodity scalar systems.

Table 7 Power efficiency of HPCG on Xeon system

	Sustained performance (Gflop/s)	Efficiency (%)	Power (W)	Power efficiency (Gflops/W)
HPCG Xeon (1 node)	14.4	3.4	246.1	0.014
HPL Xeon (1 node)	357.9	84.8	275	1.3

Currently, the power consumptions of supercomputers in the HPCG list are not disclosed. Therefore, it is impossible to compare and evaluate the power efficiencies of all systems in the list. However, considering eight to ten times higher efficiency of SX-ACE than those of accelerator and GPU systems, SX-ACE systems have a potential to achieve an almost-comparable power efficiency with those of power-aware heterogeneous supercomputers. These evaluation results clearly indicate that a high B/F ratio with a powerful core is mandatory to achieve a high sustained performance on the future highly productive supercomputer under a severe power-budget limitation.

5 Conclusions

This paper discusses the sustained performance of a new-generation vector-parallel supercomputer, SX-ACE, using practical scientific and engineering applications. First, by focusing on new features of the SX-ACE processor, the fundamental performance is analyzed using standard benchmark programs. The advanced memory subsystem with ADB, MSHR, and the out-of-order memory access function effectively accelerates the memory-intensive applications, even with indirect memory accesses. In addition, while inheriting the advantage of conventional vector processors, the SX-ACE processor achieves a higher sustained performance even in processing short vector operations by shortening the memory access latencies and the vector data forwarding.

Then, the sustained performance of SX-ACE is evaluated using practical scientific applications and the HPCG benchmark. The evaluation results show that SX-ACE provides a higher sustained performance than conventional vector system and scalar-based systems. In particular, the evaluation results using multiple nodes indicate that the high sustained performance per core with a high sustained memory bandwidth given by powerful off-chip memory bandwidth and advanced memory subsystems enables the SX-ACE system to achieve outstanding performances that are unreachable by simply increasing the number of fine-grain scalar processor cores. Besides, from the evaluation results of the HPCG benchmark, it is clarified that SX-ACE can achieve comparable and/or higher power efficiency with heterogeneous and scalar systems.

This paper also discusses the impact of a B/F ratio on performances of memory-intensive applications. The results presented in this paper clearly show the strong correlation between the B/F ratio and computational efficiencies of the system. In addition, since recent microprocessors employ the block data transfer, the importance of the actual B/F ratio considering the block data transfer is discussed. Through our evaluations and discussions, this paper concludes that a high B/F ratio with a high-

performance core is mandatory to achieve a high sustained performance. Especially, the evaluation results of SX-ACE and SX-9 indicate that there are certain possibilities to obtain a high sustained performance by enhancing memory functions and performance, rather than improving theoretical peak performance. Since it becomes more difficult to improve the theoretical peak performance of supercomputers, this paper provides a new direction for developing a future large-scale, highly productive supercomputer under a severe power-budget limitation.

Acknowledgements The authors would like to express our gratitude for the cooperation and supports of the many parties concerned. These include: Satoru Yamamoto, Genti Toyokuni, Yuma Fukushima of Tohoku University, Yoshinobu Yamamoto of Yamanashi University, Takashi Furumura of the University of Tokyo, Daisuke Inazu of the National Research Institute for Earth Science and Disaster Prevention, and Yasuhiro Sasao of Teikyo University for providing their programs for the performance evaluations in this paper. Our thanks go to Koki Okabe of Tohoku University, Hiroshi Takahara, Osamu Watanabe, Souya Fujimoto of NEC Corporation, Youichi Shimomura, and Kenta Yamaguchi of NEC Solution Innovators, Ltd. for their supports on the performance evaluations. This research is conducted as the joint research project between Tohoku University and NEC. This research uses the NEC SX-ACE, SX-9, and LX systems in Cyberscience Center of Tohoku University, the Hitachi SR16000M1 system in the Information Initiative Center of Hokkaido University, the Fujitsu PRIMEHPC FX10 system in the Information Technology Center of Nagoya University, and the ES2 and ES3 systems of Earth Simulation Center. This research was partially supported by Core Research of Evolutional Science and Technology of Japan Science and Technology Agency (JST CREST) "An Evolutionary Approach to Construction of a Software Development Environment for Massively Parallel Heterogeneous Systems," and "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan.

References

1. TOP500 Supercomputer Site. http://www.top500.org/
2. Oliker L, Canning A, Carter J, Shalf J, Ethier S (2004) Scientific computations on modern parallel vector systems. In: Supercomputing, 2004. Proceedings of the ACM/IEEE SC2004 Conference, November, pp 10–25
3. Momose S (2013) Next generation vector supercomputer for providing higher sustained performance. In: Proceedings of COOLChips XVI, COOLChips 19
4. Soga T, Musa A, Shimomura Y, Egawa R, Itakura K, Takizawa H, Okabe K, Kobayashi H (2009) Performance evaluation of Nec sx-9 using real science and engineering applications. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis(SC09), November, pp 1–12
5. Dongarra Jack "Report on the Sunway TaihuLight System". http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016
6. Kroft D (1981) Lockup-free instruction fetch/prefetch cache organization. In: Proceedings of the 8th Annual Symposium on Computer Architecture, ser. ISCA '81. IEEE Computer Society Press, Los Alamitos, CA, USA, pp 81–87
7. HPCG Benchmark. http://www.hpcg-benchmark.org/
8. Jeffers J, Reinders J, Sodani A (2016) Intel Xeon phi processor high performance programming. Morgan Kaufmann, Burlington Second Edition: Knights Landing Edition
9. Musa A, Sato Y, Soga T, Egawa R, Takizawa H, Okabe K, Kobayashi H (2008) Effects of MSHR and Prefetch mechanisms on an on-chip cache of the vector architecture. In: Proceedings of International Symposium on Parallel and Distributed Processing with Applications, December, pp 335–342

10. Momose S (2014) SX-ACE processor: NECs brand-new vector processor. In: Proceedings of HOTChips26, August 11
11. McCalpin JD (1995) Memory bandwidth and machine balance in current high performance computers. In: IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, December, pp 19–25
12. Himeno benchmark. http://accc.riken.jp/2444.htm
13. Takahashi K, Azami A, Tochihara Y, Kubo Y, Itakura K, Goto K, Kataumi K, Takahara H, Isobe Y, Okura S, Fuchigami H, Yamamoto J-I, Takei T, Tsuda Y, Watanabe K (2011) World-highest resolution global atmospheric model and its performance on the earth simulator. In: State of the Practice Reports, ser. SC '11, ACM, New York, NY, USA, pp 21:1–21:12
14. Inazu D, Hino R, Fujimoto H (2012) A global barotropic ocean model driven by synoptic atmospheric disturbances for detecting seafloor vertical displacements from in situ ocean bottom pressure measurements. Mar Geophys Res 33(2):127–148
15. Sasaki D, Akihito D, Onda H, Nakahashi K (2012) Landing gear aerodynamic noise prediction using building-cube method. Model Simul Eng 2012:7:7–7:7 Jan
16. Yamamoto Y, Kunugi T (2011) Direct numerical simulation of a high-froude-number turbulent open-channel flow. Phys Fluids (1994-present) 23(12):125108.1–125108.11
17. Yamamoto Y (2012) Structuring a database for mhd high-resolution heat transfer under the fusion reactor design conditions using large scale numerical simulations. In: *Progress report of JHPCN12*
18. Yamamoto Y, Kunugi T, Serizawa A (2001) Turbulence statistics and scalar transport in an open-channel flow. J Turbul 2(1):10–10, 2001-01-01T00:00:00
19. Toyokuni G, Takenaka H (2012) Accurate and efficient modeling of global seismic wave propagation for an attenuative earth model including the center. Phys Earth Planet Inter 200–201(0):45–55
20. Yamamoto S, Yasuhiro S (2005) Numerical prediction of unsteady flows through turbine stator-rotor channels with condensation. In: Proceedings of ASME Fluids Engineering Summer Conference, pp 855–861
21. Furumura T, Chen L (2004) Large scale parallel simulation and visualization of 3D seismic wavefield using the earth simulator. Comput Model Eng Sci 6(2):153–168
22. Yokokawa M, Shoji F, Uno A, Kurokawa M, Watanabe T (2011) The K Computer: Japanese next-generation supercomputer development project. In: Proceedings of the 17th IEEE/ACM International Symposium on Low-power Electronics and Design (ISLPED '11), August, pp 371–372
23. Marjanovi V, Gracia J, Glass CW (2014) Performance modeling of the HPCG benchmark. In: High Performance Computing Systems, Performance Modeling, Benchmarking, and Simulation, Springer, pp 172 –192
24. Oyanagi Y (1987) Hyperplane vs. multicolor vectorization of incomplete LU preconditioning for the Wilson Fermion on the lattice. J Inf Process 11(1):32–37
25. Komatsu K, Egawa R, Ogata R, Isobe Y, Takizawa H, Kobayashi H (2015) An approach to the highest efficiency of the HPCG benchmark on the SX-ACE supercomputer. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC15), Poster, November, pp 1–2 (USB)
26. Pedretti K, Olivier SL, Ferreira KB, Shipman G, Shu W (2015) Early experiences with node-level power capping on the Cray XC40 platform. In: The Proceedings of E2SC Workshop, November