Activities of Cyberscience Center and Performance Evaluation of the SX-9 Supercomputer

KOBAYASHI Hiroaki, EGAWA Ryusuke, OKABE Kouki, ITO Eiichi, OIZUMI Kenji

Abstract

The Cyberscience Center at Tohoku University is an inter-university cooperative institute that maintains and provides supercomputer systems with high-performance computation capabilities and also conducts R&D on supercomputing and networking to advance the ICT-driven science named Cyberscience. In order to extract the maximum performances of the supercomputers, the center supports the users in tuning their programs and also conducts joint research with users to realize advanced high-performance computing. This paper presents the new supercomputing system SX-9 that was installed in March 2008 and the user-support activities of our center. It also presents the early performance evaluations of the SX-9 system using real scientific application codes.

Keywords

supercomputer, vector processing, supercomputing system, high-speed network, large-scale simulation
ADB, crossbar switch, Building Cube Method

1. Introduction

The year 2008 is the 50th anniversary of the completion of the SENAC-1 parametron electronic computer (NEAC1102), which has been developed jointly by NEC Corporation and Tohoku University. In this memorable year, owing to 50-years history of joint research with NEC, Tohoku University carried out the world's first installation of the SX-9 supercomputer system. We are planning to continue the joint research for the development of the world's leading supercomputer systems.

The Cyberscience Center of Tohoku University was founded in 1969 as the Large-Scale Computer Center inheriting the developmental spirit of the SENAC-1. The objectives of the center are to maintain and provide advanced academic information infrastructures such as supercomputer systems and high-speed networks across the country. It also aims to promote education and research that will lead to creative and innovative scientific activities by utilizing these infrastructures.

In the following sections of this paper, we introduce the new supercomputing system of the center and our user-support activities. We also describe early performance evaluations of the SX-9 that started operation in March 2008. The performance evaluation includes single CPU and node performance of SX-9, and effects of an on-chip vector cache named ADB (Assignable Data Buffer). In particular, we report results of a large-scale CFD MPI simulation with the 16-node SX-9 system.

2. Supercomputing System

The center installs and maintains Supercomputing Systems for carrying out large-scale simulations that are not possible with ordinary servers or PCs. **Fig. 1** shows the configuration

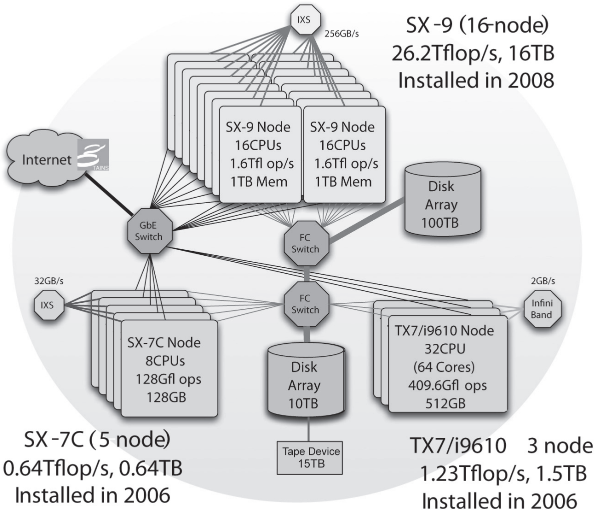


Fig. 1 Supercomputers at Tohoku University.

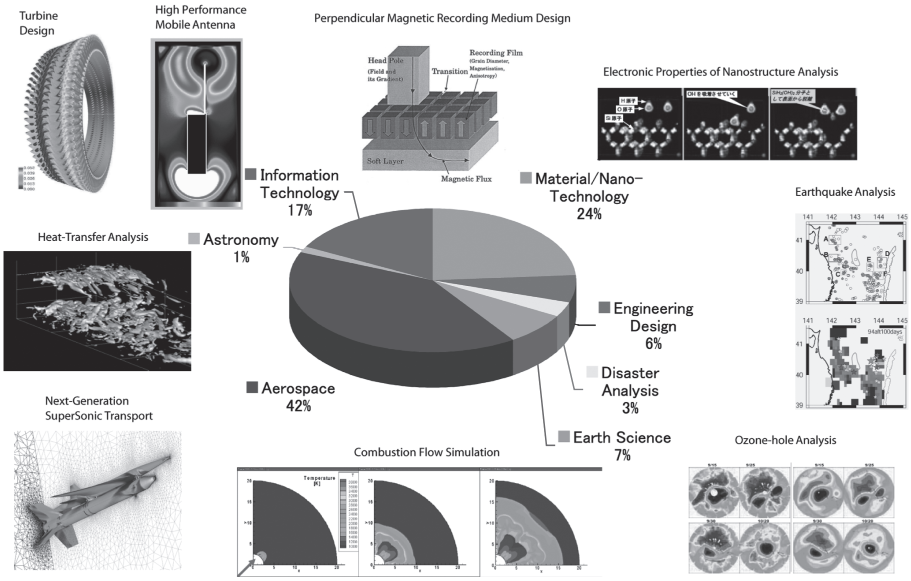


Fig. 2   A wide variety of vector supercomputer's contributions.

of the current supercomputer system. In March 2008, we replaced the previous main computing system from the SX-7 to the latest supercomputer SX-9. The system consists of SX-9 with 16 nodes, SX-7C (corresponding to SX-8) with 5 nodes and a scalar type computer TX7/i9610 with 3 nodes, which together offer a total operational capability of 28Tflop/s.

As shown in Fig. 1, one of the features of the supercomputer system is a heterogeneous computing environment of the two vector parallel supercomputers and the scalar parallel supercomputer. In combination with high memory bandwidth for a large shared memory space, vector processors of the supercomputer realize very efficient processing for computation-intensive, large-scale applications. The SX-9 system consists of 16 SMP nodes, and each node has 16 vector processors sharing a large memory space of 1TB. The 16 nodes are interconnected via a custom designed high-bandwidth crossbar network named IXS (Interconnected Crossbar Switch) at 256GB/s (bidirection). As each processor has a peak performance of 102.4Gflop/s, the total performance of the system reaches 26.2Tflop/s. The system of SX-7C consists of 5 SMP nodes, and each node has 8 vector processors sharing a memory space of 128GB. As each processor has a peak performance of 16Gflop/s, the total peak performance of the SX-7C realizes 640Gflop/s.

The TX7/i9610 system employs 192 Itanium2 CPUs with 1536GB of total memory. The system achieves a total performance of 1228.8Gflop/s. Each CPU has a peak performance of 6.4Gflop/s and 64-way parallel processing is available in one computing node. The TX7/i9610 is effective for processing programs, which are non-vectorizable and have instruction-level and thread-level parallelisms. In addition, this system is working as a front-end server of the SX-9 and SX-7C systems.

At present, our supercomputing system is used by 1500 academic researchers for various cutting-edge researches as shown in **Fig. 2** .

3. User-Support and R&D Activities

To obtain the maximum performance of the supercomputer system, it is necessary to optimize user programs by analyzing it from the point of view of supercomputer experts. The support for the tuning of user programs is one of the important services of the center, same as running and maintaining our supercomputers. At the center, faculty members, technical staff and manufacturer's engineers collaborate to optimize user programs. Through analyses of user programs and consultations, program optimization/tuning techniques for supercomputers are developed, and these techniques are fed back to users via forums, lectures and publications. In addition, the faculty member provides users with new computational

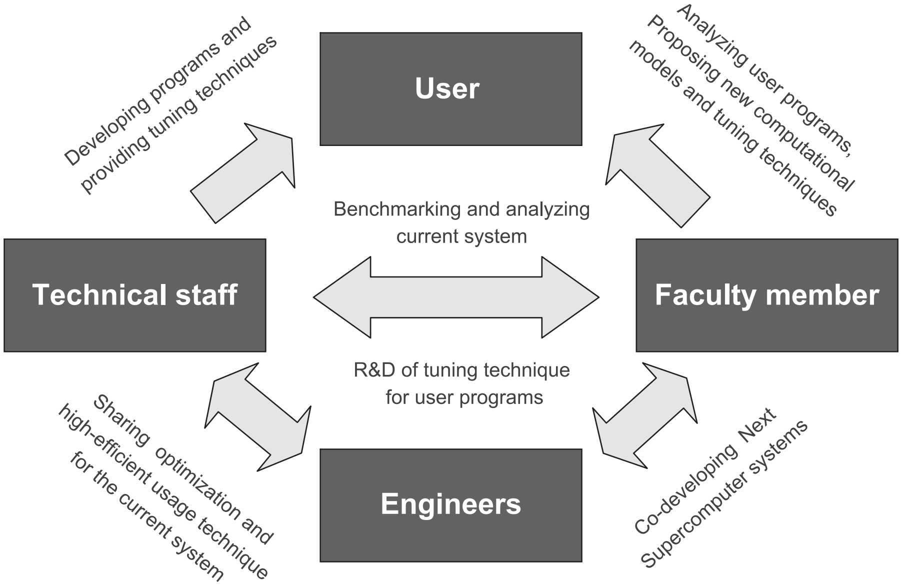

User
Technical staff
Faculty member
Engineers
Developing programs and providing tuning techniques
Analyzing user programs, Proposing new computational models and tuning techniques
Benchmarking and analyzing current system
R&D of tuning technique for user programs
Sharing optimization and high-efficient usage technique for the current system
Co-developing Next Supercomputer systems

Fig. 3 User-support system.

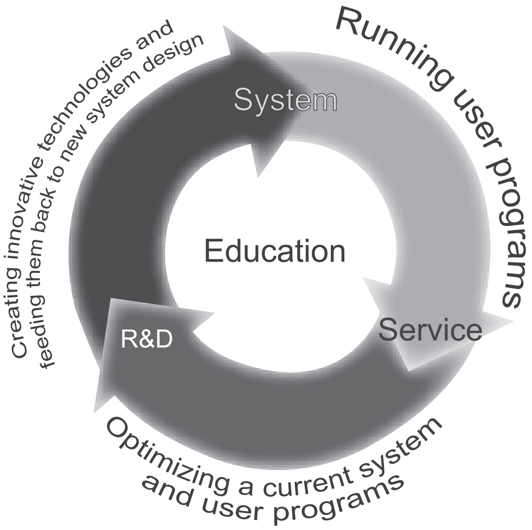

Running user programs
System
Education
Service
R&D
Creating innovative technologies and feeding them back to new system design
Optimizing a current system and user programs

Fig. 4 R&D cycle at Cyberscience Center.

model and research efforts at their division for efficient simulations as shown in **Fig. 3** . The knowledge obtained from these activities is shared among faculty members, technical staff and manufacturer's engineers for the efficient use of current supercomputers.

Every year, the center conducts joint research with users for advancement, vectorization and parallelization of programs. In 2007, eleven joint research projects were organized with researchers from both inside and outside the Tohoku University to promote R&D on large-scale simulations for the SX-9. One of the results of these projects is described in Section 5. The expertise and R&D results obtained through analyses and tuning of user programs are also shared among the users, the faculty members and technical staff of the center. The results and knowledge obtained by joint research is fed back to the design and development of the next supercomputing systems via the R&D cycle as shown in **Fig. 4** , so that we can always provide and maintain supercomputers with the highest performances and efficiencies. Also, this R&D cycle assists the human resource development of young researchers in computational and computer science.

4. Performance of SX–9

In this section, we examine performance of the new SX-9 system. We have used six real application programs for the performance evaluation, which are representative programs in the fields of earth science, aerospace science, engineering design and space physics designed and developed by our users as shown in Fig. 2 [1] . We evaluated performance of the SX-9 by comparing it with those of SX-8, SX-8R, SX-7 and TX7/i9610.

Table 1 System specifications.

System name	Number of CPUs (cores)	Clock frequency (GHz)	Per-CPU data			
			Computing performance (Gflop/s)	Memory bandwidth (GB/s)	B/flop	Cache ADB
SX-9	16	3.2	102.4	256	2.5	256KB
SX-8R	8	2.2	35.2	70.4	2	N/A
SX-8	8	2	16	64	4	N/A
SX-7	32	1.1	8.83	35.3	4	N/A
TX7/i9610	32(64)	1.6	12.8(6.4)	8.5(4.25)	0.66	L1: 16KB L2: 256KB L3: 12MB

Table 1 shows the specifications of the evaluated systems. In this table, "B/flop" is the value obtained by dividing the memory bandwidth by the peak computing performance, and it represents the ratio of the memory performance to the CPU one. In the previous work, we are conducting quantitative study on the B/flop and computing performance and have clarified that 4B/flop is required to maintain the CPU performance[2]. We also demonstrated that with a system of around 2B/flop, the cache mechanism for vector operations, such as ADB (Assignable Data Buffer) introduced with the SX-9, is effective for improving the sustained performance. In the following sections, we compare sustained single CPU (core) performance and scalable performance inside a node of each system. Then we demonstrate effects of ADB, one of the new functions of the SX-9 vector processor.

4.1 Single CPU Performance

Fig. 5 shows the sustained single CPU performance of the six benchmark programs for five NEC supercomputer systems in Table 1. The ADB function is not used in this evaluation. The SX-9 performs significantly better compared to the other systems in all programs. Particularly, when compared with the TX7, the SX-9 achieves speedups of 30 up to 155, although the peak performance ratio between the two systems is only 16. It can be understood that the SX-9 has a high capability for more efficient program execution than the TX7 in all benchmarks. In the "Land Mine" code case, the SX-9 achieves 155 times higher performance than the TX7. This program has a low data locality and a cache hit rate with the TX7 is just around 77%. As a result, the performance with the TX7 is only about 130Mflop/s. A scalar supercomputer such as the TX7 has a lower memory bandwidth than the SX Series. This causes a large difference in a performance to the SX Series when the caches are not used effectively. Even in the case of "Antenna" code that has a high data locality, the TX7 can achieve a sustained performance of only about 1.8Gflop/s in spite of a cache hit rate of 99.4%. The performance with SX-9 is 30 times higher than the TX7. This result shows that the SX-9 with a high memory bandwidth can execute programs more efficiently even with programs capable of sufficient cache usage in scalar processors.

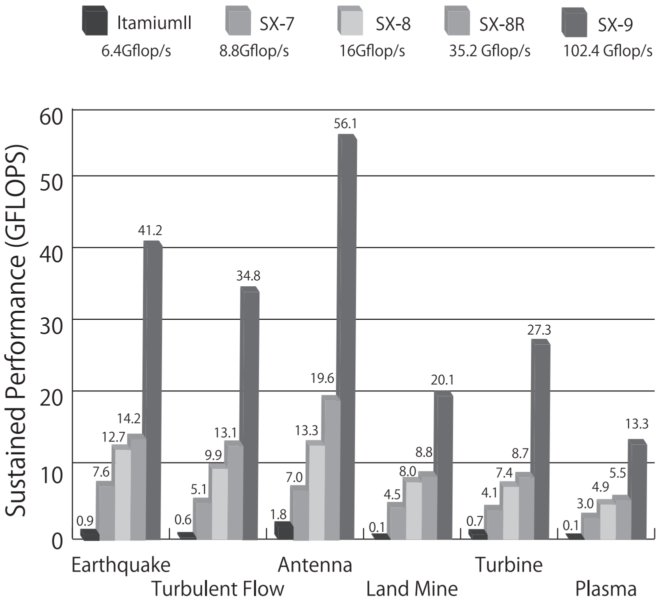


Fig. 5 Single CPU performance.

On the other hand, in comparing the performance between SX-9 and the other SX systems, the performance of the SX-9 shows 4~8 times higher than the SX-7, and 2.5~4.2 times higher than the SX-8. This is because the sustained performance varies being greatly influenced by the memory access frequency, which depends on an arithmetic operation density (number of arithmetic operations per number of memory access instructions) of each program. Also, a drop of the B/flop severely affects the sustained performance. The B/flop value, which used to be 4 with the SX-7 and SX-8, is decreased to 2.5 for the SX-9. Also, the sustained performance is strongly dependent on whether each program can efficiently utilize the second additional arithmetic pipelines (addition and multiplication) installed on SX-8R and later SX Series. For example, in the case of the "Antenna" code, the SX-9 shows a highly effective performance of 56.1Gflop/s. This program has a high arithmetic operation density of 2.3 with a few memory accesses. This has resulted in very little difference in the single CPU performance whether the B/flop value is 2.5 or 4. However, this program does not often use the additional arithmetic pipelines due to arithmetic operations inside loops. Consequently, the execution efficiency is only 54.8% (the execution efficiency is 79% with the SX-7).

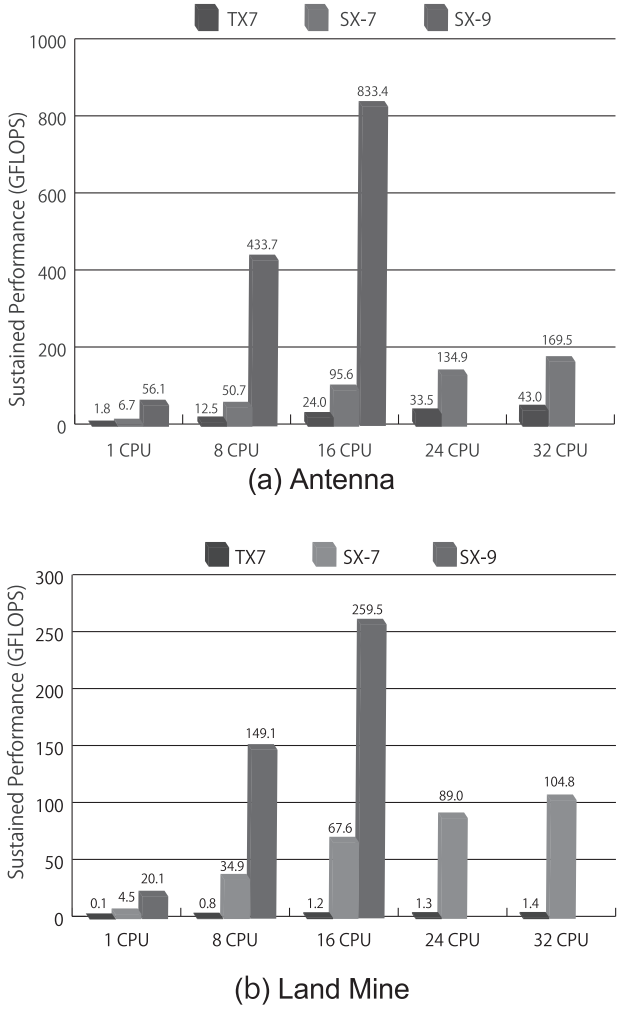
In the case of the "Plasma" code, the sustained performance of the SX-9 is 13.3Gflop/s, which is the lowest among the six programs. This is because the program uses list vectors frequently, which cannot hide the memory latency of vector-gather instructions and increase the execution time. The resulting sustained performance of the SX systems is not so high due to the bottleneck in memory accesses, but it is still impressive compared to the TX7. Other programs have frequent memory accesses with the arithmetic operation densities around 1.0, so the sustained performance of the SX-9 is determined by the B/flop value of 2.5 and additional arithmetic pipelines utilization. The more detailed discussion on these two factors and sustained performance of SX-9 is our future work.

With the SX-9, the execution efficiency may differ from the previous SX Series due to a drop of B/flop and effects of new added pipelines. Nevertheless, the SX-9 is always capable of achieving highly effective performances over 10Gflop/s per CPU with any of these programs. Consequently, we think SX-9 can provide the users with powerful vector operation environments than ever before.

4.2 Single Node Performance

To evaluate the node performance, we compared the parallel processing performances of the TX7, SX-7 and SX-9 with the same application codes used in the previous section. Although the TX7 uses two cores per CPU, our evaluation uses only one core per CPU and measures up to the 32-CPU (32-core) system. We introduce the results of the programs, "Antenna" and "Land Mine": the former features a lower number of memory accesses compared with the number of arithmetic operations, on the other hand, the latter features a higher number of memory accesses compared with the number of arithmetic operations. For parallelizing the programs, we utilize the auto parallelization function of the Fortran compiler for each system.

Fig. 6 shows the scalable performances of those systems using the "Antenna" and "Land Mine" codes. In the "Antenna" case, the performance of the SX-9 with 16 CPUs achieves 833Gflop/s, which is 460 times higher than the TX7 with 1 CPU, 124 times higher than the SX-7 with 1 CPU, 19 times higher than the TX7 with 32 cores and 5 times higher than the SX-7 with 32 CPUs. Also, in the case of "LAND MINE," the performance of the SX-9 with 16 CPUs reaches 260Gflop/s, which is lower than that with "Antenna." This is because the single CPU performance with "LAND MINE" is about half that with "Antenna" and also because of the scalability issue to be discussed later. However, the performance of the SX-9 with 16 CPUs is still 2000 times higher than that of the TX7 with 1 CPU, 58 times higher than that of the SX-7 with 1 CPU, 187 times higher than that of TX7 with 32 cores and 2.5 times higher than that of the SX-7 with 32 CPUs. As seen above, in spite of the SX-9 using half the number of CPUs compared to the previous SX-7, the high single CPU performance of the SX-9 can achieve a higher node performance than the SX-7.



(a) Antenna

(b) Land Mine

Fig. 6 Single node performance.

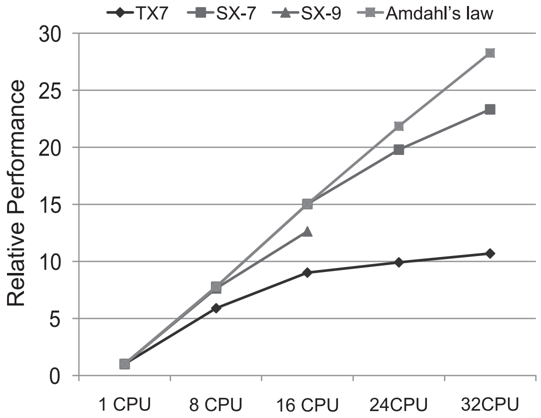


Fig. 7 Performance improvement rate (Land Mine).

In the case of the "Antenna" code, the sustained performances of all systems are improved as the number of CPUs increases. However, in the case of "Land Mine," the rate of performance improvement slows down as the number of CPUs increases. **Fig. 7** shows a relationship between a performance improvement rate of three systems and the number of CPUs of the "Land Mine" case. In this figure, all performances are normalized by the performance of each CPU. Especially, almost no performance improvement is observed above 16 CPUs in the TX7. This is because the "Land Mine" is a program with frequent memory accesses. The TX7 packages four CPUs on each cell board and the cell boards are connected through crossbar switches. The reason of the drop in the performance improvement rate of the TX7 is that the data transfer rate is limited due to simultaneous data transfer across cell boards by the CPUs. The SX-7 and SX-9 present performance improvements according to Amdhal's law up to 16 CPUs with the SX-7 and up to 8 CPUs with the SX-9, however the improvement rates also slow down with more CPUs. This is because an increase in the bank conflicts due to memory access from CPUs increases the memory access time. As seen here, the SX-9 presents the tendency that the parallel processing performance is affected by the memory load, just like previous SX Series. Nevertheless, as shown in Fig. 6, the SX-9 is capable of offering a high node performance thanks to the auto parallelization function.

4.3 Effect of ADB

ADB is a 256KB on-chip memory newly introduced in the vector processor of the SX-9. As ADB can work as a software-controllable cache, we can specify some arrays with the high locality for selectively bringing them into the cache for the effective use of the limited cache space. This section examines a usefulness and effects of ADB to improve the sustained performance. The B/flop value of the SX-9 is 2.5, however, supplying data from both the memory and ADB can improve the sustained performance by increasing the B/flop value to about 4. We examine the cases of "Earthquake" and "Land Mine" in the following.

```
   do i=1,ncells
!CDIR ON_ADB(wary)
     do j=1,ncells
       wf_dip(i)=wf_dip(i)+gd_dip(j,i)*wary(j)
     end do
   end do
```

Fig. 8 Main loop of earthquake.

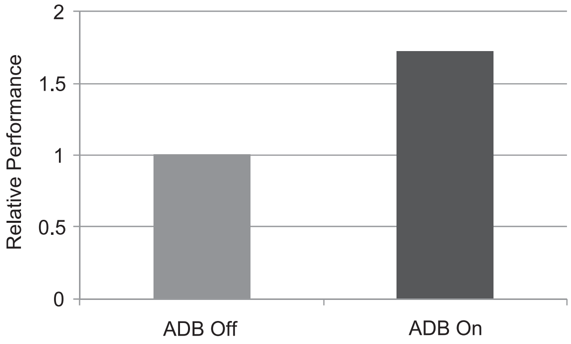


Fig. 9 Performance improvement by ADB (Earthquake).

Fig. 8 shows one of high cost kernels of "Earthquake" code, where compiler directive "ON_ADB" specifies array to cache the data in ADB. The loop in Fig. 8 is vectorized with index *j* and arrays *gd_dip* and *wary* are loaded by vector-load operations. The capacity of array *wary* is 250KB, which means that all of its data can be cached in ADB. As a result, array *gd_dip* is supplied from the memory while array *wary* is supplied from ADB, and the data supply rate for vector pipelines reaches 4B/flop. **Fig. 9** shows the performance with ADB normalized by that without ADB. The sustained performance is 1.7 times that of the case in which ADB is not used.

Now let us look at the case of "Land Mine." **Fig. 10** shows the main loop of "Land Mine." The loop is the kernel of the FDTD technique, and most loops have identical structures. With this program, the loop length of index *i* is as small as 50 so the compiler vectorizes the loop of index *j* . In this evaluation, each array has a size of 858MB, so all of the data cannot be cached in the 256KB ADB. However, we can find out some arrays with high locality in this code. For example, data references in the differential equation of the FDTD model show a high special locality. If we focus on " *H_y (i, j, k) - H_y (i-1, j, k)* " in the third line from the bottom in Fig. 10, the data of *H_y (i, j, k)* is reused by *H_y (i-1, j, k)* of the subsequent *i+1* . Also " *H_z (i, j, k)- H_z (i-1, j, k)* " in the fifth line from the bottom is also reusable, and the ADB capacity required for the reuse is 24KB. These arrays are good candidates for selective caching with ADB. **Fig. 11** also shows the results of the case in which only arrays *H_y* and *H_z* are selectively cached in ADB by the directive lines. In this case, the performance improvement is about 1.18 times, and the use of ADB has increased sustained single CPU performance to 24Gflop/s.

It is demonstrated that ADB can improve the program performance significantly when data with high locality are selectively cached, although the size of ADB is only 256KB. Apart from the methods shown here, there may also be other utilization methods such as caching the list vector table in ADB. We intend to continue studies for efficient utilization methods of ADB as tuning techniques for the SX-9.

```
   DO 10 k=0,Nz
    DO 10 i=0,Nx
!CDIR ON_ADB(H_y,H_z)
      DO 10 j=0,Ny
        E_x(i,j,k) = C_x_a(i,j,k)*E_x(i,j,k)
   &     + C_x_b(i,j,k) * (( H_z(i,j,k) -H_z(i,j-1,k))/dy
   &     -(H_y(i,j,k) -H_y(i,j,k-1))/dz -E_x_Current(i,j,k))
        E_y(i,j,k) = C_y_a(i,j,k)*E_y(i,j,k)
   &     + C_y_b(i,j,k) * ((H_x(i,j,k) -H_x(i,j,k-1) )/dz
   &     -(H_z(i,j,k) -H_z(i-1,j,k))/dx -E_y_Current(i,j,k))
        E_z(i,j,k) = C_z_a(i,j,k)*E_z(i,j,k)
   &     + C_z_b(i,j,k) * ((H_y(i,j,k)-H_y(i-1,j,k) )/dx
   &     -(H_x(i,j,k)-H_x(i,j-1,k) )/dy -E_z_Current(i,j,k))
  10 CONTINUE
```

Fig. 10 Main loop of "Land Mine" code.

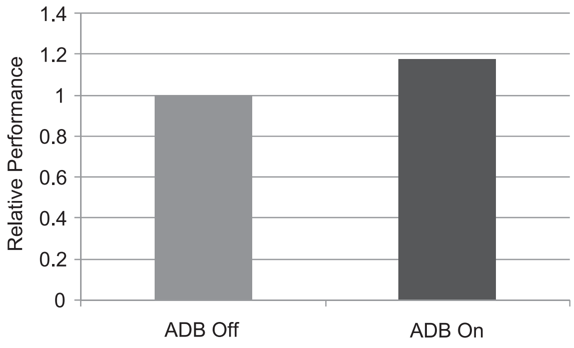


Fig. 11 Performance improvement by ADB (Land Mine).

5. Large–Scale Simulation

Finally, we introduce an example of large-scale simulation using the whole SX-9 system in our center. Before the installation of the 16-node SX-9 system, the center has conducted joint research with the Nakahashi Laboratory, Department of Aerospace Engineering, Tohoku University, for developing a simulation program for efficient parallel operations of 256 vector processors.

The Nakahashi Laboratory uses supercomputers in the studies for solving various flow-related problems, such as the computation techniques of computational fluid dynamics (CFD), aerodynamic design techniques based on CFD and the application of mathematical optimization algorithms to aerodynamic design. The Nakahashi Laboratory has also proposed the Building Cube Method as the next generation [3)-4)] CFD algorithm. As shown in **Fig. 12** , the Building Cube Method is a computation technique that stacks various sized cubic blocks (cubes) in the computational space and provides each cube with high-density, equal-interval orthogonal mesh (cells), which can be given a grid density according to the locality of flow. The use of the equal-interval orthogonal mesh in the cube is advantageous in that it can increase the precision as well as maintaining a simple algorithm.

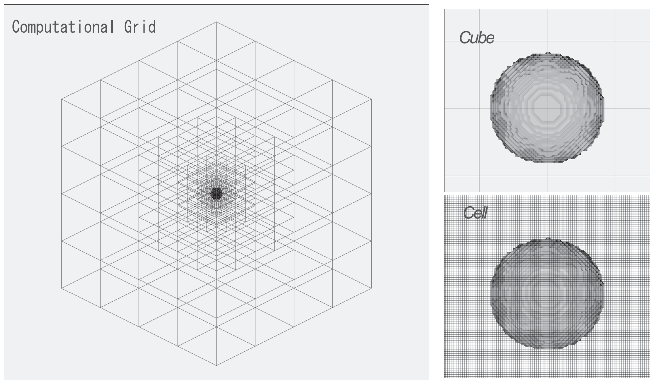


Fig. 12 Examples of spatial mesh structure.

With regard to our collaboration with Nakahashi's group, the faculty member and postgraduates of the Nakahashi Laboratory, the faculty member of our laboratory, the technical staff of our center and engineers from NEC have held several meetings in order to study the characteristic analysis and parallelization of the Building Cube Method. The simulation program that is consequently developed is intended to solve the incompressible Navier-Stokes equations using the third-order upwind difference approximation for the convection terms, the second-order central difference approximation for the viscosity terms and the second-order explicit scheme for the time integration based on the staggered grid fractional step method and to solve the Poisson equations on the pressure using the second-order central difference and the SOR method.

From the study, we confirm the Building Cube Method has high parallelism and the ability to adjust the granularity and load balance of the large-scale parallel processing by assigning multiple cubes for each processor. For the program parallelization, we have employed the global memory function [5] of the SX-9 for improving the data transfer efficiency by the MPI (Message Passing Interface) library and adopted MPI_GET and MPI_PUT as the data transfer functions. For the vectorization, we introduced the planar method for the solution of Poisson equations, which has the highest cost among the calculations related to incompressible fluids and also vectorized the iterative computations of the SOR method.

For the performance evaluation, the surface pressure distribution is obtained by reproducing the wind flow around a racing car using 2 million cells in 5,930 cubes. **Fig. 13** shows the visualization result, where the dark parts above the driver's seat and behind the rear wing indicate a drop in pressure. The rise in pressure is obtained on the white parts on the body and in front of the rear wing. The areas in the proximity of the racing car use small cubes in order to improve the simulation precision. **Table 2** shows the effective performance ratio between the TX7/i9610 and the SX-9. The SX-9 with 256 CPUs required 50 minutes in order to execute the simulation in Fig. 13. When this result is applied to other systems, the table indicates that the TX7 with 1 CPU would take an execution time of 128 days for the same simulation and the SX-9 with 1 CPU would take 6 days to complete it. Our collaboration has made it possible to execute larger scale CFD MPI simulations than hitherto and in shorter periods.

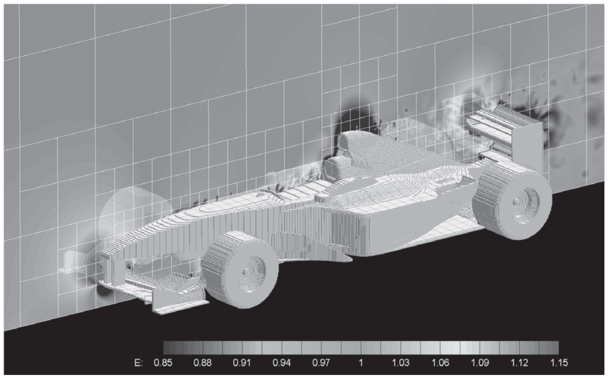


Fig. 13 A visualization result of Building Cube Method.

Table 2 Performance comparison of TX7 and SX-9 (Building Cube Method).

System name	TX7		SX-9		
Number of CPUs (cores)	1 cores	64 cores	1 CPU	16 CPU	256 CPU
Peak performance (flop/s)	6.4 G	409.6 G	102.4 G	1.6 T	26 T
Performance ratio	1	36	20	285	3700

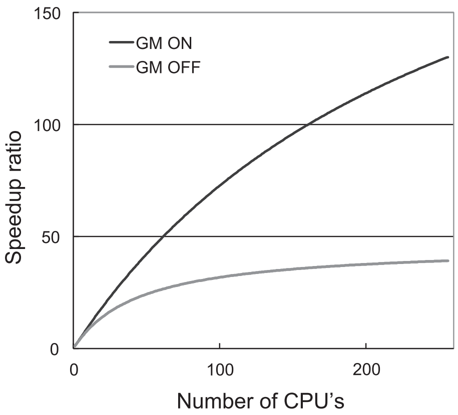


Fig. 14 Performance of SX-9's global memory function.

Fig. 14 shows a scalable performance with and without the global memory function of the SX-9. When the global memory function is used, the SX-9 with 256 CPUs can provide a 130 times higher performance. However, without global memory function, the performance does not improve above 100 CPUs and the performance improvement with parallel processing remains at 40 times. This fact indicates that the global memory function of the SX-9 can reduce the overhead of data transfer through MPI and is therefore necessary for large-scale simulations in multi-node environments.

6. Conclusion

In this paper, we presented our activities, an overview of the new supercomputing system SX-9 and early performance evaluation of the SX-9 using real scientific and engineering applications. Owing to the powerful 100+ Gflop/s vector processor and high-speed network technology, the experimental results show remarkable improvement for the considered applications. We believe that the NEC SX-9 has enough capability for accelerating users' research activities.

Since the installation of the Supercomputer SX-1 system in 1986, the Cyberscience Center of Tohoku University has been conducting researches into the maintenance and efficient use of vector type supercomputers and enhancements of hardware and software. With the installation of the SX-9, we are now able to provide an efficient high-performance computing environment more than ever before. The center will continue its user-support activities for allowing the academic researchers in Japan to make use of the new functions of the SX-9 and its 26.2Tflop/s performance using 16 nodes. With regard to the problems that the future supercomputers such as the next-generation supercomputer will encounter, e.g., those of power consumption, thermal issue and memory performance, the center is also determined to contribute to solutions through research into chip stacking, memory architecture and parallelization technologies.

We sincerely wish that the activities of the center will lead to further significant advancements in Japanese computer and simulation technologies.

In closing, we would like to express our gratitude for the cooperation and support of the many parties concerned. These include: Professor Emeritus Akira Hasegawa, Professor Kunio Fukazawa, Professor Motoyuki Sato, Professor Satoru Yamamoto, Associate Professor Masahide Iijima of the Tohoku University, Dr. Takahiro Tsukahara of the Tokyo University of Science and Dr. Keisuke Ariyoshi of the Japan Agency for Marine-Earth Science and Technology for their provision of evaluation programs. For their important cooperation in the joint research into large-scale simulations we also thank: Professor Kazuhiro Nakahashi and Mr. Shun Takahashi of Tohoku University, Mr. Akihiro Musa of NEC Corporation, Mr. Takashi Soga of NEC System Technologies, Ltd., and Mr. Yoichi Shimomura of NEC Software Tohoku, Ltd.

References

1) Egawa, R. et. al., "New Supercomputing System - Performance Evaluation of Supercomputer SX-9," SENAC, Vol.41, No.2, pp.11-18, 2008. (in Japanese)

2) Kobayashi, H., A. Musa, et al., "The potential of On-chip Memory Systems for Future Vector Architectures", In M. Resch et al. (Eds.), High Performance Computing on Vector Systems 2007, pp.247-264. Springer-Verlag (2007).

3) Nakahashi, K., "Building-Cube Method for Flow Problems with Broadband Characteristic Length," Computational Fluid Dynamics 2002, edited by S. Armfield, R. Morgan, K. Srinivas, Springer, 2003, pp.77-81.

4) Nakahashi, K., "High-Density Mesh Flow Computations with Pre-/Post-Data Compressions," AIAA 05-4876, 17th AIAA Computational Fluid Dynamics Conference, Toronto, June 2005.

5) NEC Corporation: SUPER-UX - MPI/SX RIYO NO TEBIKI (SUPER-UX - MPI/SX Operation Guide)," 2008.

Authors' Profiles

KOBAYASHI Hiroaki
Professor,
Director,
Cyberscience Center Tohoku University

EGAWA Ryusuke
Assistant Professor,
Research Division on Supercomputing Systems,
Cyberscience Center Tohoku University

OKABE Kouki
Research Associate,
Research Division on Supercomputing Systems,
Cyberscience Center Tohoku University

ITO Eiichi
Technical Assistant Director,
System Management Section,
Information Infrastructure Division,
Information Department, Administration Bureau,
Tohoku University

OIZUMI Kenji
Technical Associate Expert,
System Management Section,
Information Infrastructure Division,
Information Department, Administration Bureau,
Tohoku University