White paper
PRIMEQUEST 2800E
With All Flash Array and Microsoft SQL Server 2014
Best Practice

Fujitsu Mission Critical server PRIMEQUEST 2000 with much affordable performance scalability is the best server platform for database applications.
This whitepaper explains how PRIMEQUEST 2800E with Violin 6610 and Microsoft SQL Server 2014 helps improve database performance.

Content

Introduction

Through reading whitepaper, you will understand how PRIMEQUEST 2000 combined with All Flash Array including Violin is effective to much improve database performance. You will also obtain precious information about new features of Microsoft SQL Server 2014 which helps improve database performance.
Fujitsu appreciates great contributions by Macnica Network Corporation in entire processes of performance tests and creation of this whitepaper.

Test outline

This whitepaper explains technical knowledge about performance improvement of PRIMEQUEST and its relationship with hardware and software configurations as stated in following sections. Tests are outlined as shown below.

- Tests related to PRIMEQUEST and All Flash Array
  - Maximum IO throughput and its relationship with random or sequential read/write accesses
  - Best practice for tuning of hardware configurations to maximize IO performance
  - Best practice for tuning of configurations including LUN block size of Violin and Allocation Unit size of Operating System to maximize IO performance
  - How PRIMEQUEST configurations such as the number of System Boards and CPUs influence DB performance
- Software related tests
  - How Buffer Pool Extension influences processing time for DB operations
  - How In-Memory OLTP and Memory Optimization Table influence processing time for DB operations

For details of PRIMEQUEST, please visit: www.fujitsu.com/PRIMEQUEST
For details of All Flash Array product used in these tests, please visit: http://www.violin-memory.com/products/6000-flash-memory-array/

Tests focused on Hardware

Four types of performance tests are explained here.

- Maximum IO throughput performance and its comparison with HDD configuration
- How the number of LUN influences the performance
- How the storage configurations such as LUN block size and Allocation Unit Size of Windows Server OS influence the performance
- How the number of System Board influences the performance

Tests for maximum throughput performance

IO throughput such as the number of IO per second and MB/s for random and sequential IO operations are measured.

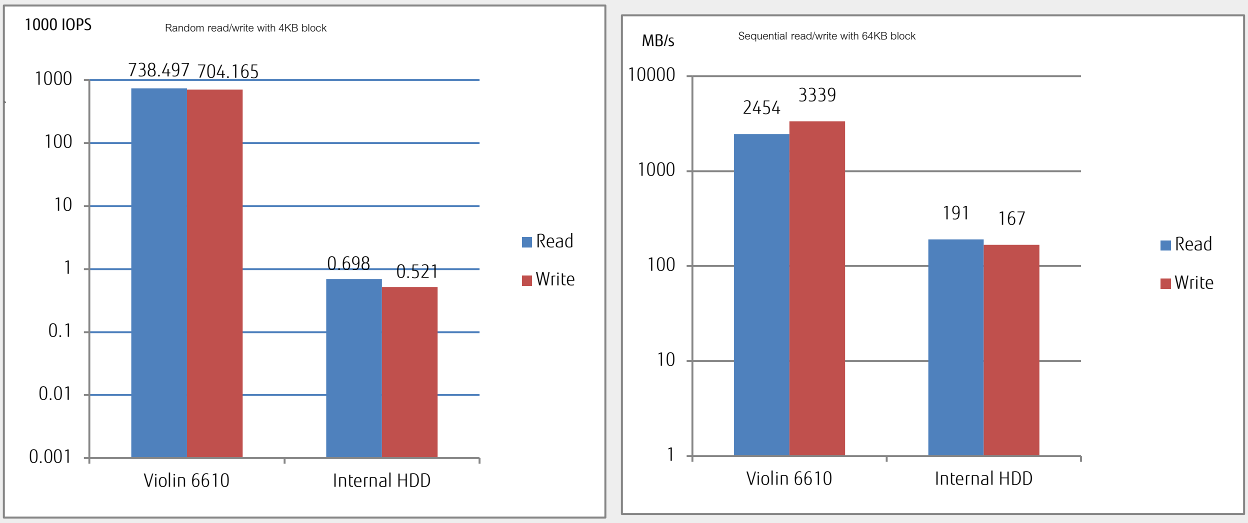
Configuration

- PRIMEQUEST 2800E with 8 CPU & 8 fiber channel ports (4 System Boards and 4 IO Units)
- Violin 6610
- Benchmark tool : SQLIO Disk Subsystem Benchmark Tool
- Data size : 25GB, one data file is stored for one data file
- Data access pattern : random read/write, sequential read/write
- IO pattern : read 100%, write 100%
- IO block size : 4KB for random read/write, 64KB for sequential read/write
- Comparison with 2x SAS-HDD 10Krpm RAID1
- Elapsed time : 30 seconds per one command execution
- Number of IO request waiting for execution per thread : 64 per command execution

Test Results

We confirmed very high performance with over 700,000 IOPS with random read/write and 4KB block size and over 2,500 MB/s throughput with sequential read/write and 64KB block size. Looking at the IOPS measurement, you can find 1,000 or 1,350 times the performance of HDD access.

Figure 1. Throughput performance (the higher the better)



Performance transition based on the number of LUN

Through this test, you can find how the number of LUN influences performance of Violin 6610.

Configuration

- PRIMEQUEST 2800E with 8 CPU & 8 fiber channel ports (4 System Boards and 4 IO Units)
- Violin 6610
- Benchmark tool : SQLIO Disk Subsystem Benchmark Tool
- Data size : 25GB, one data file is stored for one data file
- Data access pattern : random read/write, and sequential read/write
- IO pattern : read 100%, and write 100%
- IO block size : 4KB, and 64KB

- Comparison with SAS-HDD x 2/10Krpm RAID 1
- Elapsed time : 30 seconds per one command execution
- Number of IO request waiting for execution per thread : 64 per command execution

Test Methods

What are measured using SQLIO is listed here:

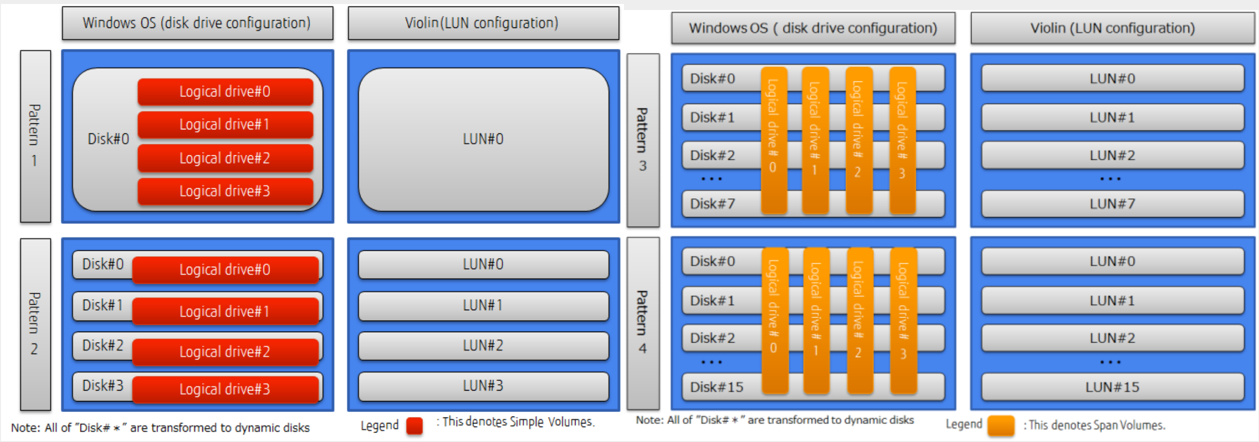
- Number of IO executions per second for read and write operations with random access as shown below.
- Windows OS setup for kind of OS volume, Allocation Unit Size, number of drives are included here.

Table 1 LUN configuration

Pattern	Setup for Violin		Setup for Windows		
	Number of LUN	LUN Block size	Volume type	Allocation Unit Size	Number of disk drives
1	1	4KB	Simple volume	64KB	4
2	4	4KB	Simple volume	64KB	4
3	8	4KB	Span volume	64KB	4
4	16	4KB	Span volume	64KB	4

Figure below explains disk drive configurations of patterns 1 to 4.

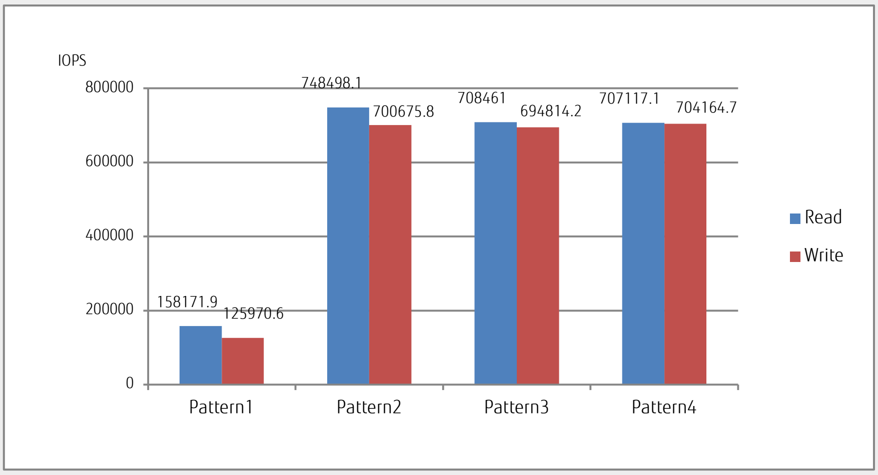
Figure 2 Disk drive configurations



Test Result

We confirmed all configurations with four or more LUN performs more than 700,000 IOPS. Then, Fujitsu recommends the number of LUN be set to four or more.

Figure 3. Throughput performance for LUN configurations (the higher the better)

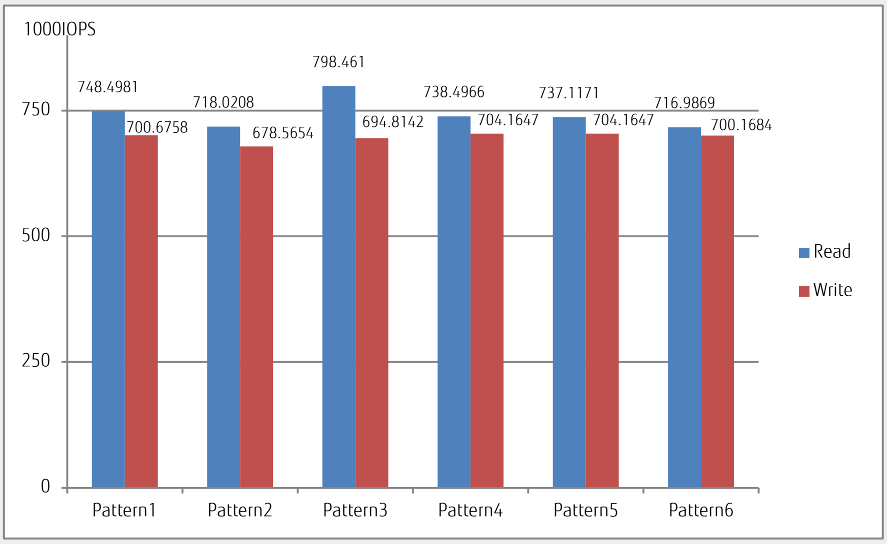


Reference
We compared IOPS between striped volume and other types of volumes such as simple or span volume in order to determine how software RAID influences performance. We selected striped volume because of its high read/write performance in general storage environment under Windows OS.

Table 2. Test patterns for LUN configuration

	Setup of Violin		Setup of Windows		
	Number of LUN	LUN block size	Volume type	Allocation Unit Size	number of disk drives
1	4	4KB	Simple volume	64KB	4
2	4	4KB	Striped volume	64KB	4
3	8	4KB	Span volume	64KB	4
4	8	4KB	Striped volume	64KB	4
5	16	4KB	Span volume	64KB	4
6	16	4KB	Striped volume	64KB	4

Figure 4. Throughput performance for difference of volume types (the higher the better)



Performance transition based on storage configuration changes

This section explains how LUN Block size of Violin and Allocation Unit Size of Windows OS influences performance.

Configurations
- PRIMEQUEST 2800E with 8 CPU & 8 fiber channel ports (4 System Boards and 4 IO Units)
- Violin 6610
- Benchmark tool : SQLIO Disk Subsystem Benchmark Tool
- Data size : 25GB, one data file is stored for one data file
- Data access pattern : random read/write, sequential read/write
- IO pattern : read 100%, and write 100%
- IO block size : 4KB, and 64KB
- Comparison with SAS-HDD x 2/10Krpm RAID1
- Elapsed time : 30 seconds per one command execution
- Number of IO request waiting for execution per thread : 64 per command execution

Test Methods

IOPS with random read/write access is measured by SQLIO. Adding to 4KB of block size, which is used for Windows OS, 512B block is used to determine whether if there is performance overhead, which may be caused by emulation – if certain OS does not make 4KB Block available, 512B block will be available as emulation mode.

Table 3. Test patterns for LUN block size / Allocation LUN Size

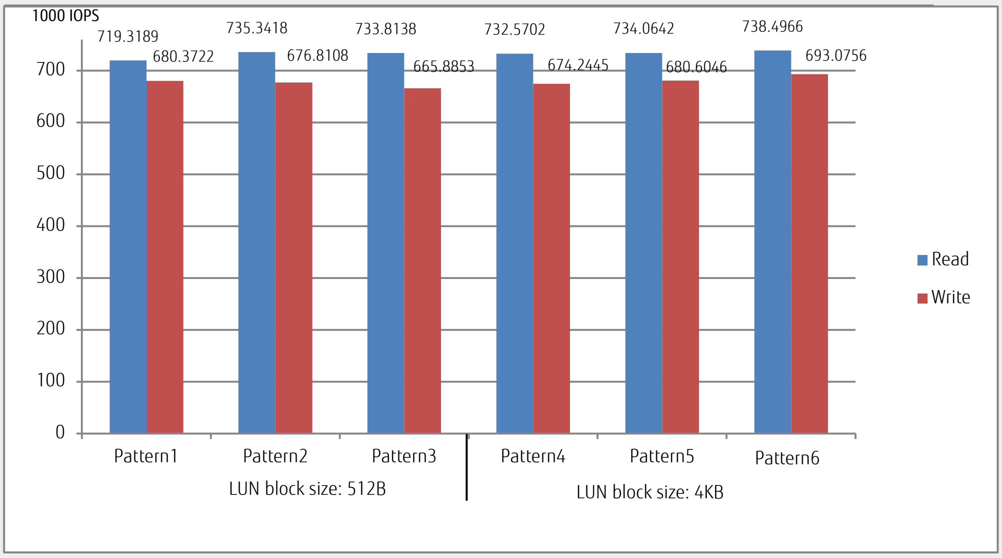
Pattern	Set up of Violin		Set up of Windows		
	Number of LUN	LUN block size	Volume type	Allocation Unit Size	number of disk drives
1	8	512B	Striped volume	4KB	4
2	8	512B	Striped volume	8KB	4
3	8	512B	Striped volume	64KB	4
4	8	4KB	Striped volume	4KB	4
5	8	4KB	Striped volume	8KB	4
6	8	4KB	Striped volume	64KB	4

Test result

We confirmed that parameters such as LUN block size nor Allocation Unit Size of Windows OS do not influence performance. Fujitsu recommends that LUN Block Size be set to 4KB. Recommended parameters for Allocation Unit Size are shown below.

- Online Transaction (OLTP)
  - 4KB is recommended for LUN Block Size because this LUN block size can optimize performance of Violin.
- Data Ware House
  - 64KB is recommended because expected large block size. Microsoft also recommends this value for this usage.

Figure 5. Throughput performance for LUN block size / Allocation LUN Size (the higher the better)



Performance transition based on server configuration changes

This section explains performance scalability of PRIMEQUEST based on number of transaction according to the increase of number of System Boards and CPUs.

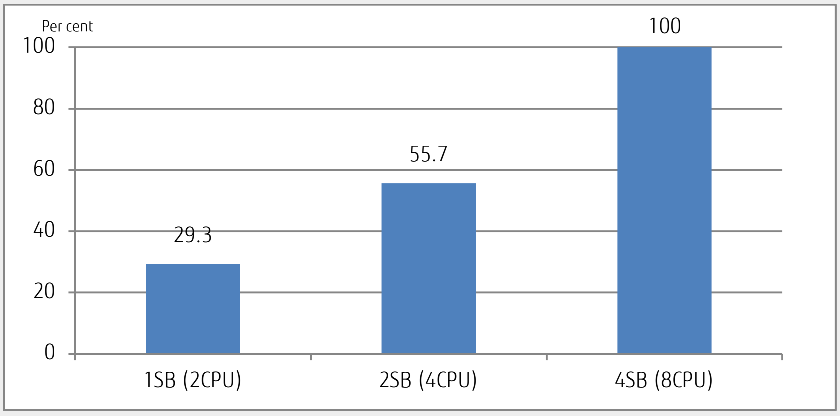
Configurations

- PRIMEQUEST 2800E with 4 IO Units and 8 fiber channel ports
  - 1 System Board with 2 CPU, 2 System Boards with 4 CPU, and 4 System Boards with 8 CPU
- Violin 6610
- Benchmark environment
  - This test measures number of transations
  - DB size: 130GB with 10,000 users
  - Number of users signed on to DB is adjusted so that CPU utilization rate stays at around 60%

Test results

With addition of System Boards and CPU, scores of online transaction increases : 1.9 times the performance for two System Boards compared to one System Board, 1.8 times the performance for four System Boards compared to two System Boards.

Figure 6. Performance on number of System Boards and CPU (the higher the better)



Note:
This graph shows relative DB transaction performance where the performance with 4SB PRIMEQUEST is set to100.

Tests focused on Microsoft SQL Server 2014

How Buffer Pool Extension influences processing time for DB operations

Buffer Pool Extension, the new function in Microsoft SQL Server 2014, helps improve DB performance. This section explains how PCIe-SSD used as buffer pool contributes DB performance improvement.

Configurations

- PRIMEQUEST 2800E with 8 CPU, 4 System Boards, 4 IO Units, and 8 fiber channel ports
  - One PCIe SSD card with1.2 TB of capacity
- Violin 6610
- Benchmark tool : Stored Procedure which executes SELECT operations
  - Retrieval of all records
  - DB size: 20GB with 50 million records

Test method

Process time for SELECT operation in test patterns below is measured.

Table 4. Test patterns for Buffer Pool Setting

Pattern	Areas DB is deployed	DB size	max server memory	Buffer Pool	Remarks
1-1	Internal HDD	20GB	16GB *	Not set	Memory capacity ;16GB Part of DB which cannot be contained in memory is stored to HDD
1-2	Internal HDD	20GB	16GB *	256GB	Memory capacity ;16GB Part of DB which cannot be contained in memory is stored to PCIe SSD
2-1	Violin	20GB	16GB *	Not set	Memory capacity ;16GB Part of DB which cannot be contained in memory is stored to Violin
2-2	Violin	20GB	16GB *	256GB	Memory capacity ;16GB Part of DB which cannot be contained in memory is stored to PCIe SSD
3	Violin	20GB	256GB	Not set	All record of DB are stored in memory

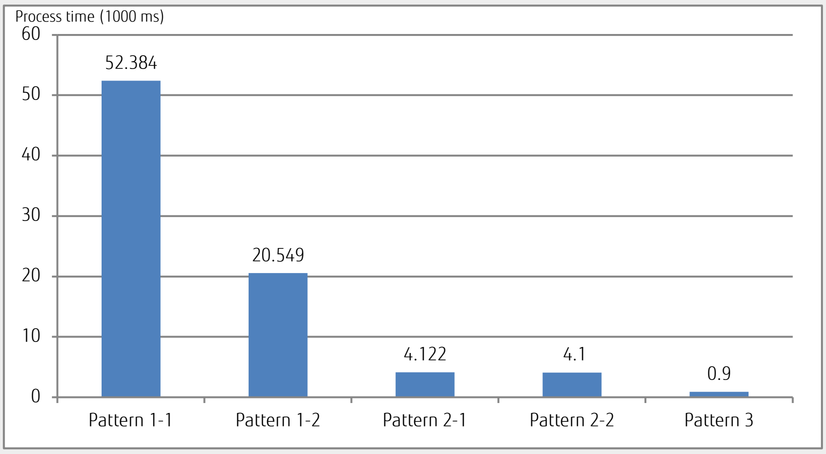
Note:
* In this test maximum memory size is set to the largest value (16GB) because instances of SQL Server may not start if this value is too small.

Test results

- (Pattern 1-1 & 1-2) Pattern 1-2 has 2.5 times the performance compared to pattern 1-1 because in pattern 1-2 all the records searched are placed in memory or PCIe-SSD but in pattern 1-1 part of DB records are placed in HDD
- (Pattern 2-1 & 2-2) Pattern 2-1 and 2-2 has similar performance results because in both patterns all DB records are placed in memory or flash storage
- (Pattern 3) This patterns has the best performance of all five patterns because all DB records are placed in memory

If you use HDD for storage for DB or if you use server with insufficient capacity of memory, Buffer Pool Extension is effective to improve performance. If all database records are placed to memory, you can obtain the best performance.

Figure 7. Process time for different Buffer Pool settings (the lower the better)



Performance test for In-Memory OLTP

In-Memory OLTP, the new function in SQL Server 2014, can much reduce time to access database tables placed in memory. This section explains how much process time for database operations In-Memory OLTP can reduce.

Configurations

- PRIMEQUEST 2800E with 8 CPUs, 4 System Boards, 4 IO Units, and 8 fiber channel ports
- Violin 6610
- Benchmark tool: Stored procedure for database operations below
  - DB size:12GB with 60 million database records
  - UPDATE operation: Operations to a specific column of all rows
  - DELETE operation: removal of all records
  - SELECT operation: retrieval of all records with specified values
- This test measures process time for operations above

Test method

Process time of SQL operations below is measured for two types of storages such as Violin and internal HDD.

Table 5. Test patterns for different types of database tables

Pattern	Optimization for In-Memory OLTP	Native compilation of Stored Procedure	Settiing of Durable Option
1	No	No	No
2-1	Yes	No	No
2-2	Yes	No	Yes
3-1	Yes	Yes	No
3-1	Yes	Yes	Yes

Note:

Durability option is used to assure durable data access to Memory-Optimized Tables even after restart of OS or SQL Server. To do so, Microsoft SQL Server refers to log in which all the updated data are retained, and restores In-Memory Database as it was before the restart of OS or SQL Server. If Durability option is disabled, updated data may be vanished.

Test results

Using Memory-Optimized Table even without native compilation of Stored Procedure and without Durable Option helps improve performance for UPDATE and DELETE operations compared to conventional Tables.

Table 6. Performance improvement by In-Memory OLTP in UPDATE and DELETE operations

Storage	SQL Operations	Pattern	Process time (ms)	Performance improvement
Violin	UPDATE	1	77	64.1 times the performance
Violin	UPDATE	2-1	1.2	
Violin	DELETE	1	128.2	106.8 times the performance
Violin	DELETE	2-1	1.2	

Using Memory-Optimized Table even without native compilation of Stored Procedure and with Durable Option helps improve performance for UPDATE and DELETE operations compared to conventional Tables. (3.8 to 8.7 times the improvement) Performance improvement in this pattern is lower than that without Durable Option because write execution with Durable Option includes write to transaction log.

Table 7. Performance influences by Durable Option

Storage	SQL Operations	Pattern	Process time (ms)	Performance improvement
Violin	UPDATE	1	77	3.8 times the performance
Violin	UPDATE	2-2	20.1	
Violin	DELETE	1	128.2	8.7 times the performance
Violin	DELETE	2-2	14.7	

Using native compilation of Stored Procedure helps improve performance to 257 times the performance.

Table 8. Performance influences by native compilation of Stored Procedure

Storage	SQL Operation	Pattern	Process time (ms)	Performance improvement
Violin	UPDATE	1	77	257 times the performance
Violin	UPDATE	3-1	0.3	
Violin	DELETE	1	128.2	142.4 times the performance
Violin	DELETE	3-1	0.9	

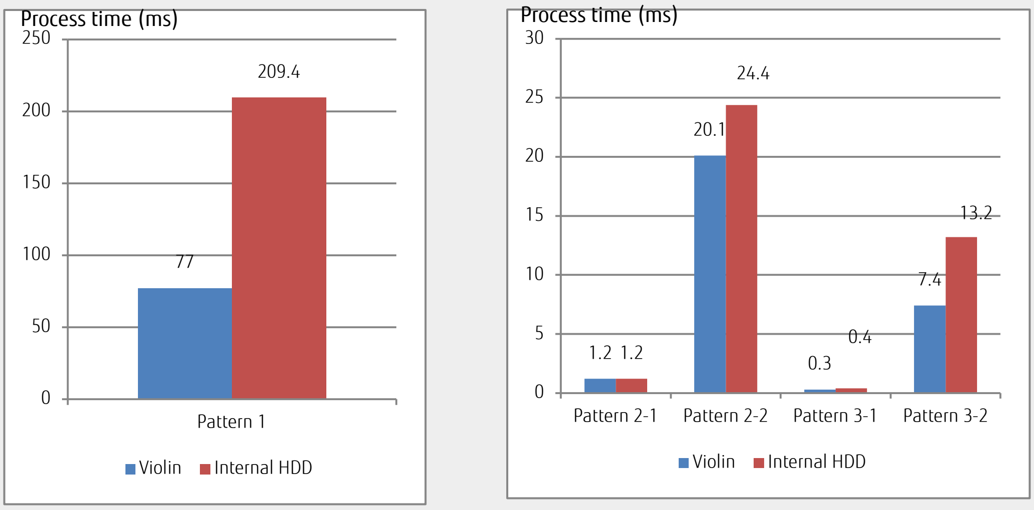
Using native compilation of Stored Procedure with Durable Option =Yes helps improve performance to 10.4 times.

Table 9. Performance influences by native compilation of Stored Procedure (Durable Option =Yes)

Storage	SQL Operation	Pattern	Process time (ms)	Performance improvement
Violin	UPDATE	1	77	10.4 times the performance
Violin	UPDATE	3-2	7.4	
Violin	DELETE	1	128.2	10.4 times the performance
Violin	DELETE	3-2	12.3	

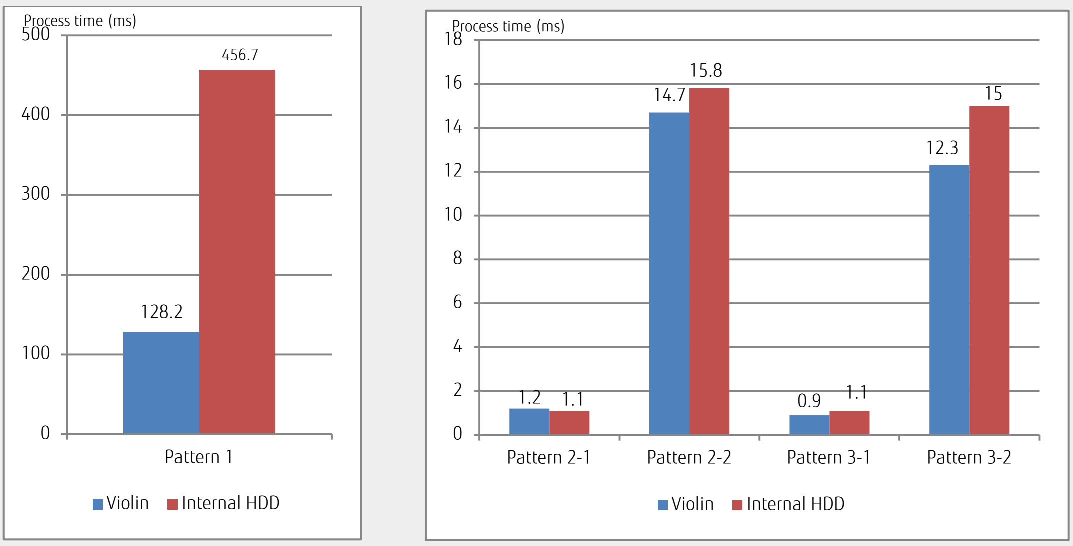
Process time for UPDATE operations is shown below.

Figure 8. In-Memory OLTP process time for UPDATE operations (the lower the better)



Process time for DELETE operation is show below.

Figure 9. In-Memory OLTP process time for DELETE operations (the lower the better)



Effectiveness to use Memory-Optimized Table for SELECT operations is lower than that of UPDATE or DELETE operations because SQL Server with Look Ahead function can reduce the process time.

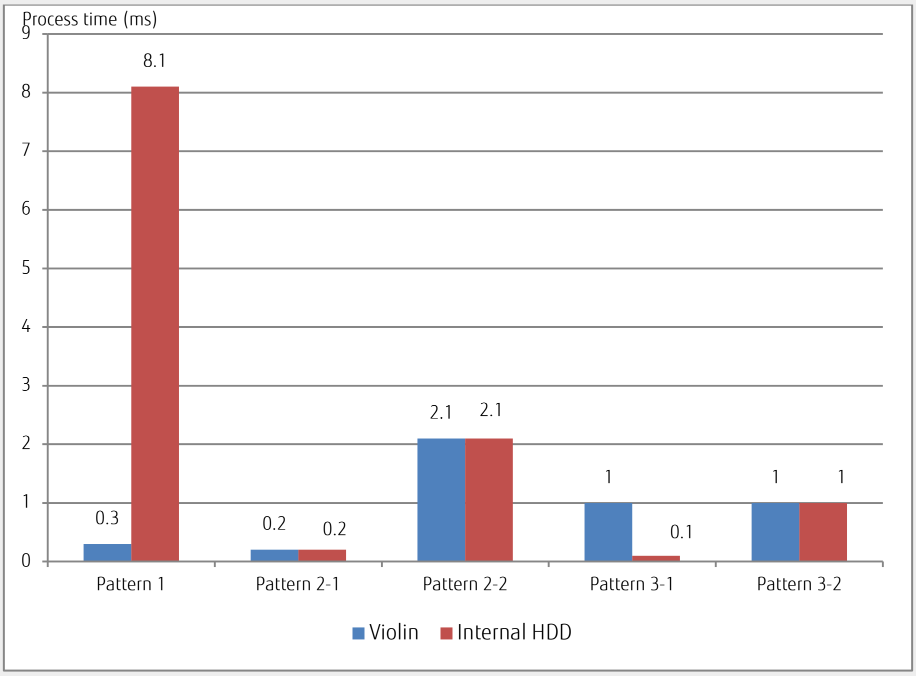
Table 10. Performance influences by native compilation of Stored Procedure

Storage	SQL Operation	Pattern	Process time (ms)	Performance improvement
Violin	SELECT	1	0.3	1.5 times the performance
Violin	SELECT	2-1	0.2	
Violin	UPDATE	1	77	64.1 times the performance
Violin	UPDATE	2-1	1.2	

Process time for SELECT operation is show below.
Comparing process time of different storage configurations using Violin and internal HDD, Violin used to store conventional Tables helps improve process time. (Max. 27 times the performance for SELECT operation as shown in pattern1 of Figure 9). For Memory-Optimized Tables, configuration patterns for Violin and internal HDD have similar performance because the Tables in both patterns are deployed in memory. (For details, refer to pattern 2-1, 2-2, 3-1, and 3-2 of Figure 9) However, if Durability Option is used, Violin helps improve performance for UPDATE operations. This is because Violin has superior performance of write operations to log compared to internal HDD. For details, refer to pattern 2-2 and 3-2 of Figure 7.

Through tests in this section, it is proved In-Memory OLTP of Microsoft SQL Server is effective for database applications with high load in update operations. If Durability Option of In-Memory OLTP is used, Violin helps improve performance due to its reduction of log-write time.

Figure 10. In-Memory OLTP process time for SELECT operations (the lower the better)



Performance tests for Updatable Raw Stored Index

Updatable Raw Store Index is the enhancement of read-only Raw Store Index provided in Microsoft SQL Server 2014. This section explains how Updatable Raw Store Index helps improve performance.

Configuration

- PRIMEQUEST 2800E with 8 CPU, 4 System Boards, 4 IO Units, and 8 fiber channel ports
- Violin 6610
- Benchmark tool : Stored procedure which execute SELECT operations
  - DB size: 5GB with 60 million records
    - Approximately 500 MB of raw data
    - Raw Data Index: Approximately 200MB of row data
  - Select operations : aggregation of specified rows using GROUP BY

Test Methods

* Archived Option: Updatable Raw Stored Indexes are archived. This used to reduce used disk area.

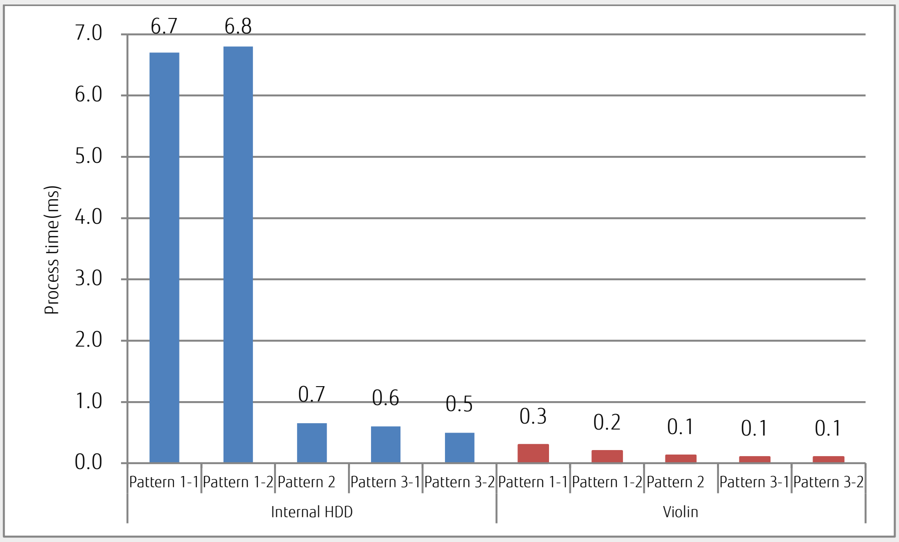
Table 11. Test patterns for Updatable Raw Stored Index

Pattern	Index type
1-1	Index (not as Raw Stored Index)
1-2	Nonclustered index
2	Read-only Stored Index
3-1	Updatable Raw Stored Index (Archived Option disabled)*
3-2	Updatable Raw Stored Index (Archived Option enabled)*

Test Results

Comparing with performance of normal index tested in Pattern 1-1 (Violin), you can find Updatable Raw Stored Index tested in Pattern 3-2 (Violin) has around three times the performance. If you compare this with Nonclustered Index tested in Pattern 1-2 (Violin), you can find Updatable Raw Stored Index has doubled the performance.
Comparing the corresponding test results with Internal HDD, you can find performance with Violin differences are widened to 22 to 34 times from Pattern 1-1 to 3-2.the performance. With these test results, conclusion is that Updatable Raw Stored Index can improve performance with Violinbetter if these indices are stored in storage of lower speed.
Another conclusion is that Updatable Raw Stored Index is effective to process large size of data where aggregation using GROUP BY, daily or monthly data aggregation, and Data Ware House are included.

Figure 11. Test results for Updatable Raw Stored Index (the lower the better)



Conclusion

This whitepaper provides insight for performance improvement of database system.

- All Flash Array including Violin can be the solution to improve throughput performance for database read/write if all of database is stored in the storage
- Database system formed of PRIMEQUEST and Violin has performance scalability to accommodate business growth
- In-Memory OLTP of Microsoft SQL Server 2014 has excellent performance but you need to take care about right tuning and configurations
    - Database log should be stored to flash storage or PCIe-SSD
    - If memory space is insufficient to store all of database, it is effective to use flash storage or PCIe-SSD for the storage of parts of database
    - Fujitsu recommends to enable Durable Option for assurance of data recovery
    - New features of Microsoft SQL Server 2014 are effective to improve database performance.
        - Adjustment of database tables to In-Memory OLTP
        - Native compilation of Stored Procedure
        - Updatable Raw Stored Index

Appendix : Test environment

This section summarizes detailed configurations used in performance tests.

Table 10. Tested Configurations

	Hardware	Software
DB server	PRIMEQUEST 2800E x 1 CPU: Xeon E7-8890v2 (2.80GHz/15core/37.5MB) *8 RAM: 4TB (32GB 1600 LRDIMM *128) DISK: Internal SSD dive (SAS): 200GB x 2 Internal HDD drive SAS, 10Krpm): 300GB x 2 PCIe SSD 1.2TB x 2 (Firmware: v7.1.15 rev110356/ Driver: 3.2.6) LAN interface : dual channel LAN card(10H BASE-T) FC interface : dual channel FC card (16Gbps) x 4 (Firmware: 1.1.43.202 / Driver: 2.76.002.001) BIOS Version:01.44 Firmware:BA14041	Windows Server 2012 R2 Standard SQL Server 2014 Enterprise
Storage	Violin 6610 x1 Flash type: SLC Capacity: 16TB FC interface : FC8Gbps(8port)	-
Application server	These application servers receive and handle online transactions from clients PRIMERGY BX920 S4 x 1 mounted in PRIMERGY BX400 S1 CPU: Xeon E5-2470v2 (2.40GHz/10core/25MB) x 2 RAM: 64GB (8GB 1600 LV-RDIMM x 8) DISK: internal HDD drive (SAS/10Krpm): 600GB *2 LAN interface: 2port onboard (10Gbps) x 2	Windows Server 2012 R2 Standard
Client	These clients invoke online transactions and issue them to application servers PRIMERGY BX920 S3 x 2 mounted in PRIMERGY BX400 S1 CPU: Xeon E5-2470 (2.30GHz/8core/20MB) x 2 RAM: 64GB (8GB 1600 LV-RDIMM *8) DISK: internal HDD drive (SAS/10Krpm): 600GB x 22 LAN interface: 2port onboard (10Gbps) x 2	Windows Server 2012 R2 Standard

Table 11. Tested configurations in detail

	Disk	RAID level	Effective disk capacity	Format	Usage
DB server (PQ2800E)	Internal SSD	Not available	185GB	NTFS / 4KB	C: (system)
	PCIe SSD	Not available	1.09TB	NTFS / 64KB	E: (buffer pool area)
	Internal SSD	RAID1	279GB	NTFS / 64KB	J: (Work area)
Storage (Violin6610)	Flash memory (SLC)	Striped volume	1.95TB	NTFS / 64KB	F: (User Data)
			1.95TB	NTFS / 64KB	G: (User Data)
			1.95TB	NTFS / 64KB	H: (User Data)
			1.95TB	NTFS / 64KB	I: (Log)

Contact
FUJITSU LIMITED
Website: www.fujitsu.com
2014-12-11-WW-EN