

ホワイトペーパー

日立アドバンスサーバ(HA8000)
フラッシュドライブ(SSD, PCIe FLASH ドライブボード)と
SQL Server 2014 の有効性検証

2014年8月発行

株式会社 日立製作所

用語

HA8000	インテル Xeonプロセッサを搭載したタワーおよびラック型PCサーバ 2014年6月時点の最新チップセット、ドライバを利用する。
フラッシュドライブ	記憶媒体にフラッシュメモリを使用したドライブの総称。 複数のフラッシュメモリで構成し高速アクセス可能な記憶装置。
SSD	フラッシュドライブの1種。solid state driveの略。
PCIe FLASH ドライブボード	フラッシュドライブの1種。 サーバとの接続インターフェースにPCI Expressを利用する。
Windows Server 2012R2	マイクロソフト社が2013年にリリースした、Windows Server OS の最新バージョン。 2014年6月の最新バージョンを利用する。
SQL Server 2014	マイクロソフト社が2014年にリリース予定の、SQL Server の最新バージョン。 Microsoft SQL Server 2014 CU1(2931693)を利用する。
LU	logical Unitの略。
T-SQL	Transaction-SQLの略。マイクロソフト社のSQL言語。
AlwaysOn 可用性グループ	Microsoft SQL Server2012及び2014で提供される高可用性データベースを構成する機能。
トランザクションログ	データベースの重要なコンポーネントの1つであり、データベースの一貫性を保つデータ群。
AG	AlwaysOn可用性グループの略

登録商標および商標について

Windows Server、SQL Server、Active Directory、Microsoft Failover Cluster は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Intel および Intel Xeon は、アメリカ合衆国およびその他の国におけるインテルコーポレーションまたはその子会社の商標または登録商標です。

その他、このドキュメントで記載する製品名および会社名は、各社の商標または登録商標です。

本文中では、® および ™ は明記していません。

注意事項および免責事項について

1. 本書の内容は一部または全部を無断転載することは禁止されています。
2. 本書の内容に関しては将来予告なしに変更することがあります。
3. 日立製作所の許可なく複製・改変などを行うことはできません。
4. 日立製作所が製品やサービスについて行う保証は、製品添付の保証文章に記載した内容のみに限定され、本書のどの箇所であっても何ら新規の保証を行うものではありません。
5. 運用した結果の影響については、責任を負いかねますのでご了承ください。
6. 本書に技術的あるいは編集上の誤りや欠陥があったとしても、日立製作所は一切の責任を負わないものとします。
7. 性能の結果は保証するものではありません。実案件ではベンチマーク値を使った机上計算のみでなく、実際の運用環境と同等構成(ハード・ソフト)において、要件を満たしていることを確認してください。
8. サイトに掲載している性能評価指標、または、その他の性能に関連する情報の使用によって、いかなる損害が生じた場合も、情報提供者である日立製作所は責任を負いかねますのでご了承ください。

目次

1	はじめに	4
2	製品および機能概要	5
2.1	HA8000	5
2.1.1	フラッシュドライブ	6
2.2	SQL Server の機能	7
2.2.1	AlwaysOn 可用性グループ	7
2.2.2	インメモリ OLTP	9
3	検証概要	10
3.1	検証シナリオ	10
3.2	検証環境	11
3.3	DB 構成	12
3.4	検証方法	12
4	想定内容と結果	14
4.1	各デバイスの特徴を確認	14
4.1.1	SQLIO による確認	14
4.1.2	SQLIOSIM による確認	15
4.2	AlwaysOn 可用性グループ有無による性能傾向を確認	16
4.2.1	AlwaysOn 可用性グループ有無による性能	16
4.2.2	AlwaysOn 可用性グループと各デバイスの性能	17
4.2.3	AlwaysOn 可用性グループの同期型と非同期型の性能	18
4.3	各デバイスの混在環境における性能傾向を確認	19
5	まとめ	20
6	注意事項	21
7	付録	22

1 はじめに

システム処理性能をハードウェアレイヤーで向上させるための手段の一つとして、NAND 型フラッシュメモリを搭載したフラッシュドライブが注目されています。このうち、日立アドバンスサーバ HA8000 シリーズではシステム装置内蔵タイプの SSD ドライブと高速な PCI Express インターフェースに直接接続する PCIe FLASH ドライブボードをサポートしています。

一方、DB システムの可用性を高める手段の一つとして、SQL Server 2012 から AlwaysOn 可用性グループ機能がサポートされ、SQL Server 2014 にて機能拡張されました。これにより、エンタープライズレベルの OLTP や DB システムの可用性をさらに高めることが期待できます。

本ドキュメントでは、システムの高可用性、高性能化をご検討されている方を対象に、日立アドバンスサーバ HA8000 シリーズにて、SQL Server 2014 AlwaysOn 可用性グループ機能が動作することを検証、また SQL Server 機能の性能をハードディスクおよびフラッシュドライブを用いて測定し、分析した内容に基づき、SQL Server 各機能と各種ドライブとの組み合わせにおける有効性を解説します。

本内容は株式会社日立製作所と日本マイクロソフト株式会社の共同で実施した検証に基づき作成しています。

対象読者は、Windows Server 2012R2 及び SQL Server 2014 の概要をご理解頂いている方を想定しています。

2 製品および機能概要

本ホワイトペーパーで使用したラックマウントサーバの製品概要と SQL Server の機能概要を示します。

2.1 HA8000

日立アドバンスサーバ HA8000 シリーズ(以降、HA8000 と記載)は、広がり続けるビジネスニーズに即応できる、タワーサーバからラックサーバまでの幅広いラインアップを備えた PC サーバ。効率の良い運用と導入のしやすさで、コスト削減とビジネスのスピードアップを支援致します。

HA8000 ラックサーバは高密度実装(1U~4U)によりラックキャビネットへの高集積化を実現、インテル Xeon プロセッサを最大 4 個搭載し、ネットワークインタフェースは 1000BASE-T や 100BASE-TX に対応、内蔵ストレージは RAID5(SAS 6Gbps) 2.5 型 HDD や SSD、PCIe FLASH ドライブボードを採用、優れた処理性能と高可用性、拡張性を備え、小規模から大規模システムまで幅広い用途に適応しています。

以下に HA8000/RS220 の概観を示します。



図 2.1 HA8000/RS220 概観

システム運用をサポートする機能として、HA8000 を構成する各コンポーネント(プロセッサ、メモリ、ディスクなど)を管理するため、ソフトウェア及びツールを提供しています。

高性能・高信頼性を求めるシステムには、サーバの稼働状況をモニタリングする機能が重要です。日立製作所のシステム運用管理ソフトウェアの JP1 と連携してトータルな運用支援を行います。以下に HA8000 の内蔵ストレージを管理するソフトウェア及びツール間の連携フローを示します。なお、管理対象サーバは管理される HA8000、通知先のサーバはマネージャサーバ、端末は運用者の操作端末を指します。

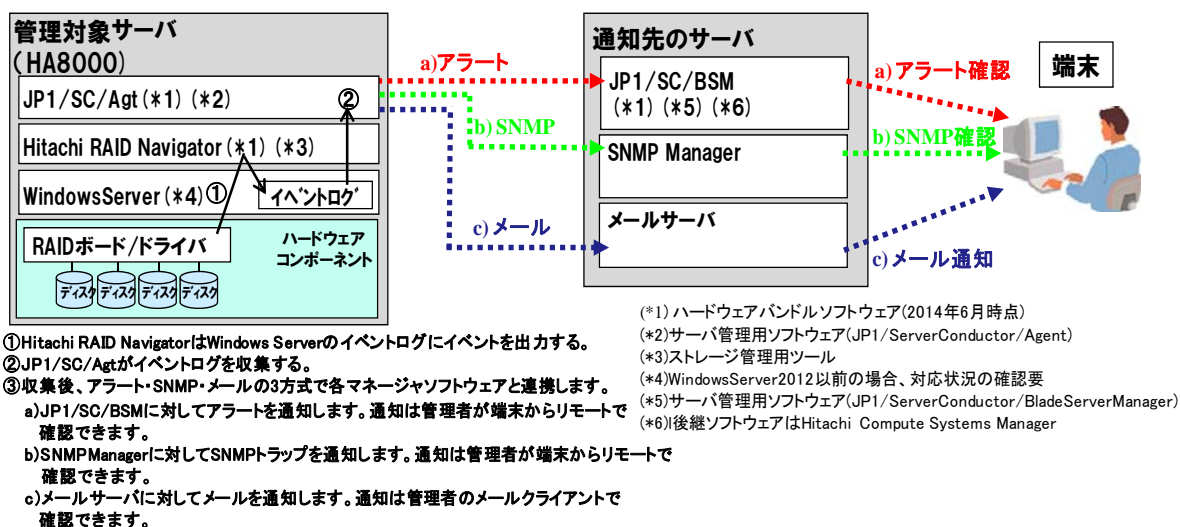


図 2.2 HA8000 内蔵ストレージの管理を想定した連携フロー

2.1.1 フラッシュドライブ

HA8000 の内蔵ストレージとして搭載できるフラッシュドライブは、SSD と PCIe FLASH ドライブボードの 2 点があります。

以下に HA8000 内蔵ストレージ(以降、PCIe FLASH ドライブボード、SSD、SAS HDD、SAS HDD (ニアライン)を総称してデバイスと記載)のラインナップを示します。

■HA8000内蔵ストレージ ラインナップ

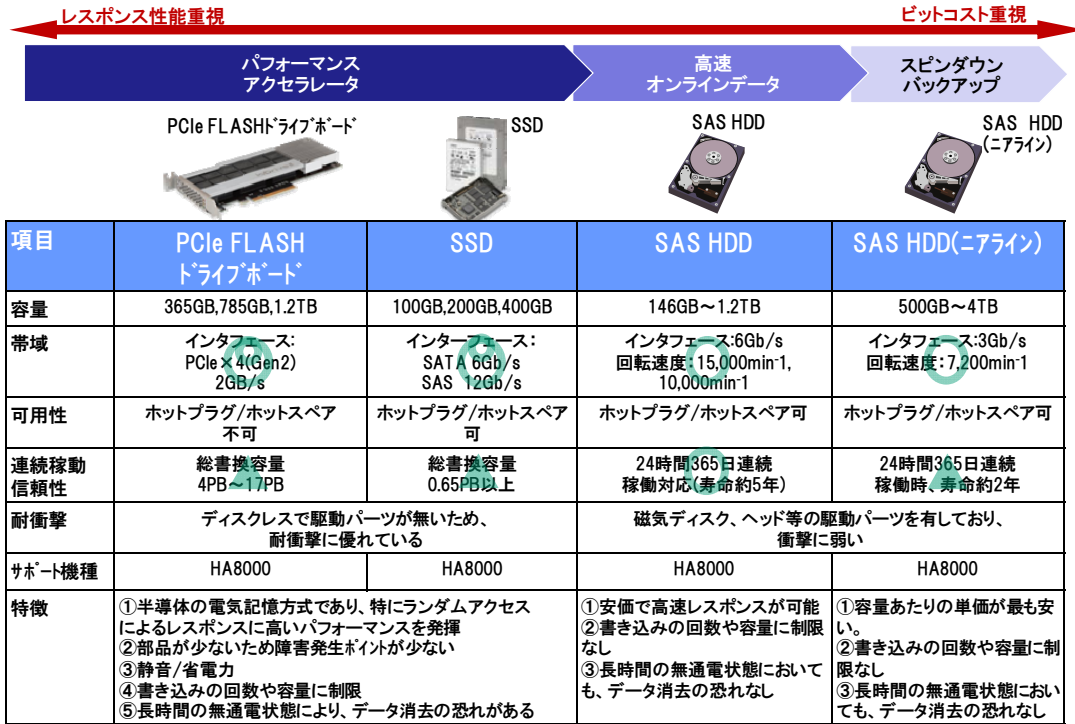


図 2.3 HA8000 内蔵ストレージ ラインナップ

2.2 SQL Server の機能

SQL Server は複数のコンポーネントとツールで構成されており、コンポーネントの 1 つとなる SQL Server データベースエンジンは、システムデータベースとユーザが定義する(アプリケーションで利用するデータベースを格納する)ユーザデータベースがあります。さらに、ユーザデータベースのデータは、実データ(テーブル・インデックスなど)を格納するデータ領域とトランザクションログ領域(以降、ログ領域と記載)の 2 つで構成します。また、データ領域群をグルーピングするファイルグループがあります。

以下に、SQL Server 2014 で拡張された高可用性を実現する AlwaysOn 可用性グループと高性能を実現するインメモリ OLTP(インメモリ最適化)について紹介します。

2.2.1 AlwaysOn 可用性グループ

(1)AlwaysOn 可用性グループ概要

SQL Server 2014 が提供する高可用性を実現する機能として、トランザクションログをセカンダリノードのデータベースに送信するログ配布、 AlwaysOn可用性グループと比べて仕様制限があるデータベースミラーリング、 AlwaysOn可用性グループ、共有ディスクが必要となる AlwaysOn FCI があります。 AlwaysOn可用性グループと AlwaysOnFCI は SQL Server 2012 から実装した機能となり、ログ配布とデータベースミラーリングは SQL Server2012 以前からの機能となります。

以下に SQL Server 可用性構成におけるバージョン・エディションと前提条件を記載します。

表 2.1 SQL Server 高可用性構成におけるバージョン・エディションと前提条件

機能名	SQL Server												前提技術			
	2014				2012				2008R2		2008		Active Directory	Windows Server フェールオーバー クラスタ	共有ディスク	
	Enterprise	Business Intelligence	Standard	Enterprise	Business Intelligence	Standard	Datacenter	Enterprise	Standard	Enterprise	Standard					
ログ配布	○												-			
データベース ミラーリング ※4	○	○※1	○	○	○※1	○	○	○※2	○	○※3	○※1	-				
AlwaysOn 可用性グループ ()は最大セカンダリ数	○ (8)	-	○ (4)	-								必須		-		
AlwaysOn フェールオーバー クラスタ インスタンス ()はノード数	○	○(2)	○	○(2)	○	○	○(2)	○	○(2)	必須						

- ※1 サポートされている (安全性レベルが FULL の場合のみ)
- ※2 シングル スレッド、同期のみ
- ※3 完全
- ※4 SQL Server 2014 では推奨していない機能

SQL Server 機能の詳細に関しては以下をご参照ください。

【高可用性ソリューション(SQL Server)】

<http://msdn.microsoft.com/ja-jp/library/ms190202.aspx>

SQL Server の高可用性構成の中でも、AlwaysOn 可用性グループはトランザクションログをプライマリノードからセカンダリノードに複製します。共有ディスクや監視サーバを必要とせず動作し、プライマリノードがダウンした場合にセカンダリノードがサービスを提供できます。本ドキュメントにおいて、AlwaysOn 可用性グループを高可用性構成として記載します。

以下に AlwaysOn 可用性グループの構成例を示します。

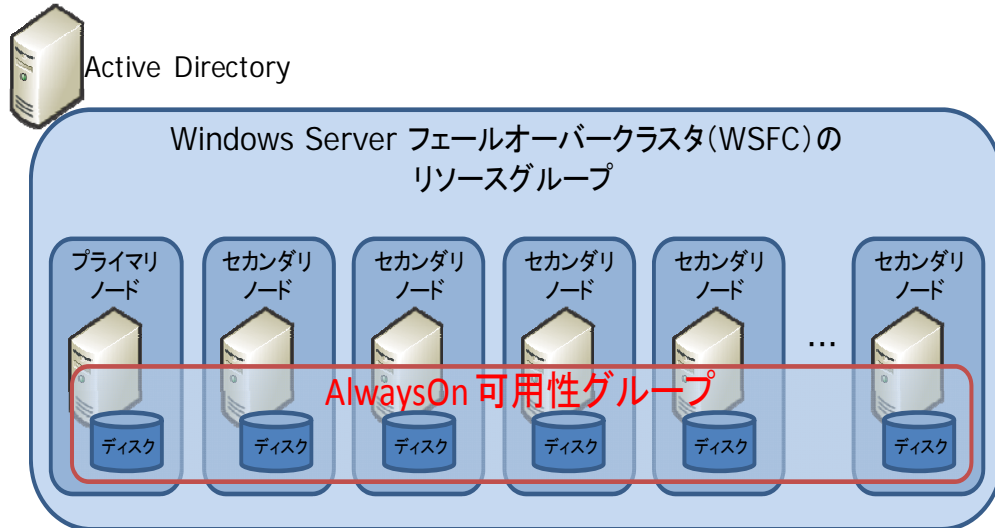


図 2.4 AlwaysOn 可用性グループの構成例

(2) AlwaysOn 可用性グループの同期型と非同期型

AlwaysOn 可用性グループにおけるデータベースの複製は、同期型と非同期型を選択できます。同期型はプライマリノードからセカンダリノードへのトランザクションログの転送が完了するのを待ち、クライアントに応答処理を返します。特徴は、障害が発生したときにプライマリノードからセカンダリノードに自動で切り替えることができます。

非同期型はプライマリノードからセカンダリノードへのトランザクションログ転送の完了を待たず、クライアントに応答処理を返します。片道の転送処理のため同期型と比べるとクライアントへの応答速度は速くなります。

用途としては、ディザスタリカバリ対策用、セカンダリノードはデータベーススナップショットを作成せずに読み取り専用で公開することができるため、BI 用のデータベース用が想定されます。

2.2.2 インメモリ OLTP

SQL Server 2014 が提供する性能向上を実現する機能として、インメモリ OLTP (インメモリ最適化) が利用できます。
本機能は SQL Server 2014 から実装した機能となります。

インメモリ OLTP は SQL Server のデータベースエンジンに統合しており、データ領域をメモリ内に展開してトランザクション処理を行うため性能向上が期待できます。一方、ログ領域はメモリ内に展開せずディスクを指定する必要があるためディスク性能を考慮した設計が必要となります。

インメモリ OLTP を使用すると、SQL Server システムの大幅な性能改善が望める可能性がありますが、サポートする Transact-SQL (以降、T-SQL と記載) に制限がありますので留意が必要です。

【インメモリ OLTP でサポートされていない Transact-SQL の構造】

[http://msdn.microsoft.com/ja-jp/library/dn246937\(v=sql.120\).aspx](http://msdn.microsoft.com/ja-jp/library/dn246937(v=sql.120).aspx)

【メモリ最適化テーブルを使用するための要件】

[http://msdn.microsoft.com/ja-jp/library/dn170449\(v=sql.120\).aspx](http://msdn.microsoft.com/ja-jp/library/dn170449(v=sql.120).aspx)

3 検証概要

3.1 検証シナリオ

HA8000 のフラッシュドライブと SQL Server の高可用性・性能向上の機能を利用した高性能、高可用性システムを検討するため、3 点の検証を実施しました。

(1)各デバイス(SAS HDD,SSD,PCIe FLASH ドライブボード)の特徴を確認

SQLIO(3.3 にて説明)と SQLIOSIM(3.3 にて説明)を利用して各デバイスの処理速度を測定し、各デバイスの特長を確認しました。

(2)AlwaysOn 可用性グループ有無による性能傾向を確認

SQL Server Management Studio を利用して 3 点確認しました。

①AlwaysOn 可用性グループ有無による性能

AlwaysOn 可用性グループ環境と AlwaysOn 可用性グループを利用しないスタンドアロン環境との処理時間を測定し、後者の性能劣化の割合を確認しました。

②AlwaysOn 可用性グループ有無と各デバイスの性能

SAS HDD で構成したスタンドアロン環境とフラッシュドライブで構成した AlwaysOn 可用性グループ環境との処理時間を測定し、後者の性能向上を確認しました。

③AlwaysOn 可用性グループの同期型と非同期型の性能

AlwaysOn 可用性グループの同期型と非同期型での処理時間を測定し、後者の性能向上を確認しました。

(3)各デバイスの混在環境における性能傾向を確認

SQL Server Management Studio を利用して、データ領域とログ領域に SSD と PCIe FLASH ドライブボードを組み合わせた場合での処理時間を測定し、性能差異を確認しました。

3.2 検証環境

本検証環境は、ドメイン管理用に Active Directory を 1 台、正系の DB サーバとしてプライマリサーバ 1 台、副系の DB サーバとしてセカンダリサーバ 1 台を用意しました。プラットフォームは HA8000、Windows Server 2012R2、SQL Server 2014 を利用しました。

デバイスの構成は、SAS HDD で構成した RAID グループから 1LU を作成し、Windows Server と SQL Server のシステム領域及びシステムデータベースを格納しました。ユーザデータベース用のデータ領域とログ領域としては、SAS HDD、SSD で構成した RAID グループから複数 LU を用意しました。PCIe FLASH ドライブボードは RAID 未構成とし、複数 LU を用意しました。

ネットワーク構成は、クライアントから可用性グループリスナーに T-SQL 文を実行するアクセス用ネットワークと MSFC のハートビート通信で利用するハートビート用ネットワークとしました。今回検証で利用するクライアントとは、PC などの端末ではなくプライマリサーバを指します。

以下に検証環境を示します。7 付録(5)にファームウェア・ドライバー一覧を記載します。

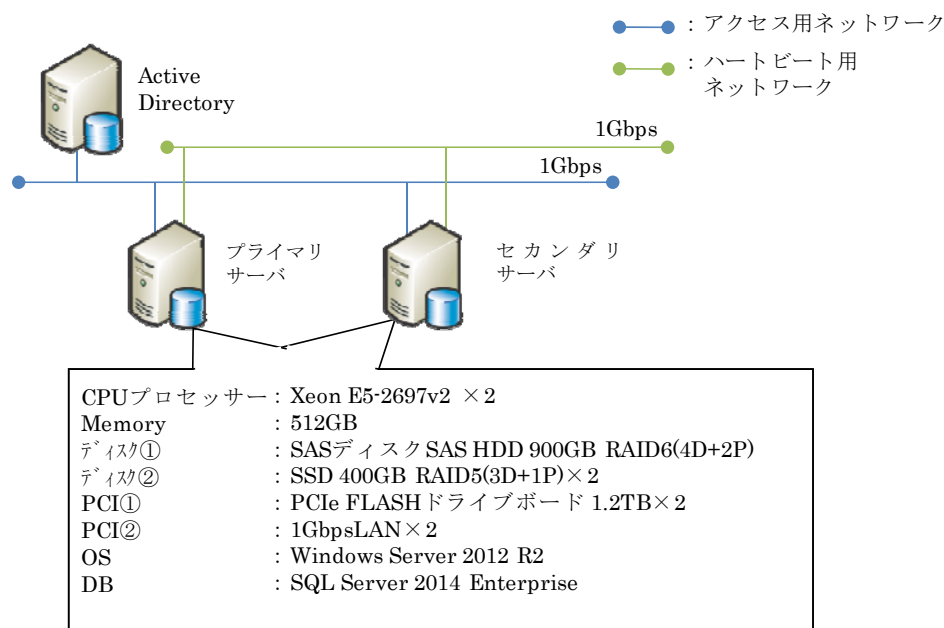


図 3.1 検証環境

3.3 DB 構成

本検証で利用するデータ領域は、ユーザデータベース用に用意した複数 LU に対して Windows Server2012 からドライブマッピングしました。ログ領域はデータ領域と別 RAID グループの LU に対してドライブマッピングしました。

データ領域の DB 構成は、1DB 当たりカラム 7 列(varchar 型)で構成した 1 テーブルを 10 テーブル分用意し、DB 数は 4DB を作成しました。

以下に DB 構成を示します。

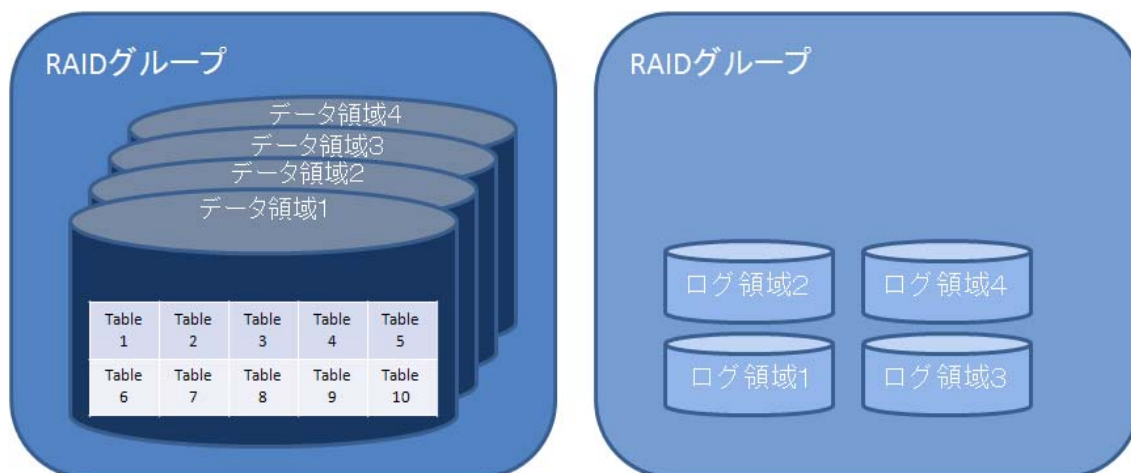


図 3.2 DB 構成

3.4 検証方法

今回のシナリオを確認するため、検証で使用したツール、作業前提条件を記載します。

(1) ツール

① SQLIO

SQLIO は SQL Server 2008 までをサポートするマイクロソフト社が提供する SQL Server に使用するストレージの I/O 性能を確認するテスト負荷ツール。ツールでは各ドライブに対してシーケンシャルまたはランダムな read・write 処理を実行し、IO 数や処理速度を確認することができる。SQL Server 2014 では正式サポートが表明されないが、デバイスの処理性能を確認するのに有効です。

② SQLIOSIM

SQL Server 2008 以降に SQLIO の後継としてリリースされたマイクロソフト社が提供する DB 用のテスト負荷ツール。ツールでは各ドライブに対してシーケンシャルまたはランダムな read・write 処理を実行し、IO 数や処理時間を確認する。

SQLIO は 1 度の処理でシーケンシャルまたはランダムな read・write 処理の 4 種類から 1 つの I/O タイプを実行することとなるが、SQLIOSIM は 1 度に複数の I/O タイプを実行することができます。

③ SQL Server Management Studio

SQL Server を統合的に管理・運用するツール。アカウント管理、データベース管理や分析、SQL 実行など実行できます。今回は、作成したテーブルに対してランダム write となることを想定して insert の T-SQL 文、update の T-SQL 文を実行し、処理時間を計測する。

(2)前提条件

①SQL Server の設定

- ・インメモリ OLTP について、ネイティブコンパイルストアプロシージャ機能は無効化、遅延永続化機能は無効化、データの遅延永続化は有効化する。
- ・SQL Server は機能を利用するために必要となる最低限の設定のみとし、他はデフォルト設定とする。

②検証準備

- ・常時、パフォーマンスモニタを有効化する。
- ・検証実行前は、SQL Server のサービスを再起動する。(事前の処理で SQL Server が使用したメモリを開放するため)

③検証内容

- ・insert の T-SQL 文は、100 万件/テーブルに行う。1DB に 10 テーブルあるため、合計 1000 万行を書き込む。
- ・update の T-SQL 文は 7 列/テーブルに対して 2 列分を全更新する。

4 想定内容と結果

4.1 各デバイスの特徴を確認

4.1.1 SQLIO による確認

SQLIO ツールを利用して、各デバイス(SAS HDD、SSD、PCIe FLASHドライブボード)のドライブに対して read・write 処理を実行しました。IO サイズは小ブロックサイズ(8KB)としました。

以下に処理速度の比較結果を示します。

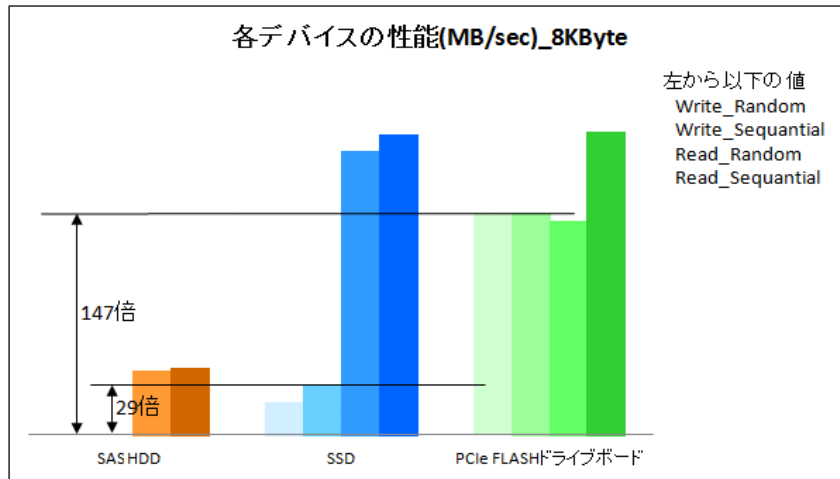


図 4.1 SQLIO を利用した各デバイスの性能比較(8KB)

上記結果から以下を確認できました。

①SSD

SAS HDD と SSD との性能差は最大 29 倍と確認しました。特に、Read 処理の速度が速いことを確認しました。

②PCIe FLASHドライブボード

SAS HDD と PCIe FLASHドライブボードとの性能差は最大 147 倍と確認しました。特に、小ランザクション(8KB)での Write 処理が早いことを確認しました。

4.1.2 SQLIOSIM による確認

SQLIOSIM ツールを利用して、各デバイス(SAS HDD、SSD、PCIe FLASHドライブボード)のドライブに対して read・write 処理を実行しました。以下にツール実行時間の結果を示します。結果は各デバイス当たり6ドライブの平均値を採用します。

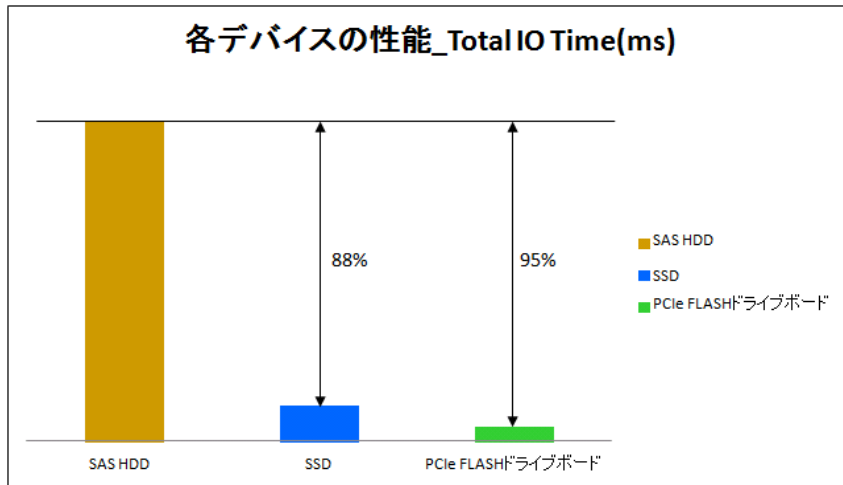


図 4.2 SQLIOSIM を利用した各デバイスの性能比較

上記結果から以下を確認できました。

①SSD

SAS HDD と SSD との性能差は約 88%と確認しました。

②PCIe FLASHドライブボード

SAS HDD と PCIe FLASHドライブボードとの性能差は約 95%と確認しました。

4.2 AlwaysOn 可用性グループ有無による性能傾向を確認

4.2.1 AlwaysOn 可用性グループ有無による性能

AlwaysOn 可用性グループ(図は AG と記載)を利用した場合と利用しない場合における insert の T-SQL 文と update の T-SQL 文をインメモリ(SSD)のドライブに対して実施しました。インメモリ(SSD)とは、インメモリ OLTP を利用し、ファイルグループとログ領域を格納するデバイスに SSD を利用することを指します。AlwaysOn 可用性グループは同期型としました。insert と update の際には SQL Server Management Studio を利用しました。以下に処理時間の結果を示します。

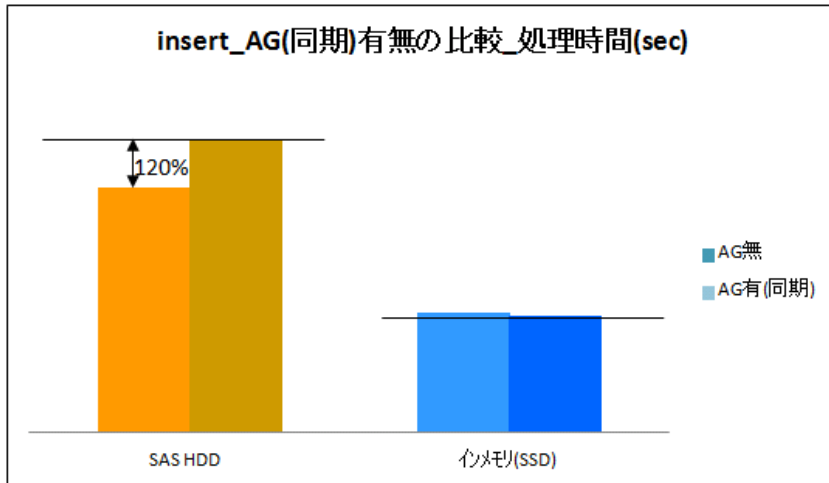


図 4.3 insert 時の AG 無と AG 有(同期)の処理時間

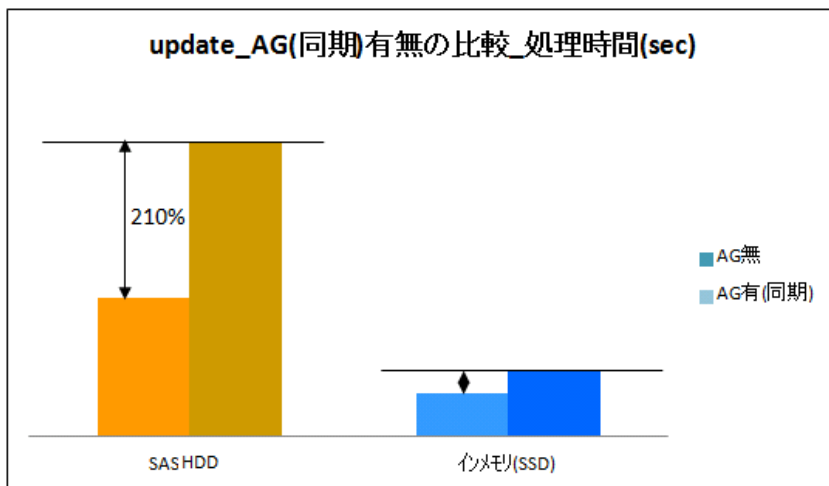


図 4.4 update 時の AG 無と AG 有(同期)の処理時間

上記結果から以下を確認しました。

- ① Insert 処理において、 AlwaysOn 可用性グループ無と AlwaysOn 可用性グループ有との性能差は最大 120%と確認しました。また、フラッシュドライブを利用することで AlwaysOn 可用性グループ有無による差異を軽減できます。
- ② update 処理において、 AlwaysOn 可用性グループ無と AlwaysOn 可用性グループ有との性能差は最大 210%と確認しました。また、フラッシュドライブを利用することで AlwaysOn 可用性グループ有無による差異を軽減できます。

4.2.2 AlwaysOn 可用性グループと各デバイスの性能

AlwaysOn 可用性グループを利用した場合における insert の T-SQL 文と update の T-SQL 文をインメモリ(SSD)のドライブに対して実施しました。AlwaysOn 可用性グループは同期型としました。insert と update の実行には SQL Server Management Studio を利用しました。

以下に処理時間の結果を示します。

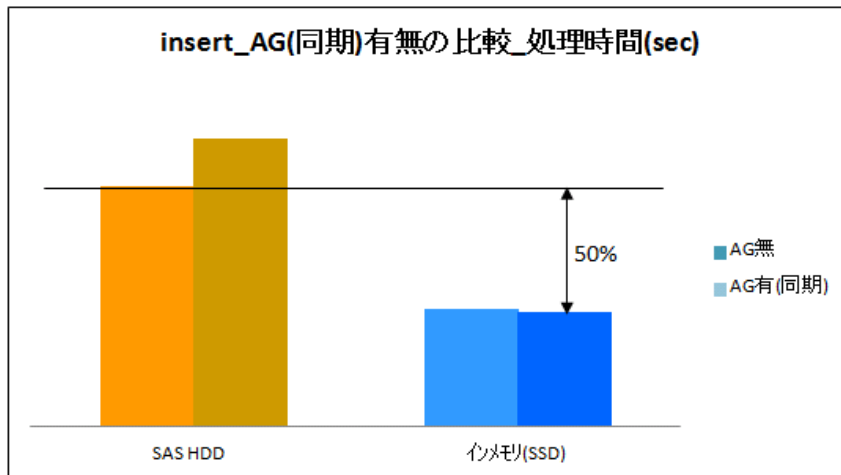


図 4.5 insert 時の AG 無と AG 有(同期)の処理時間

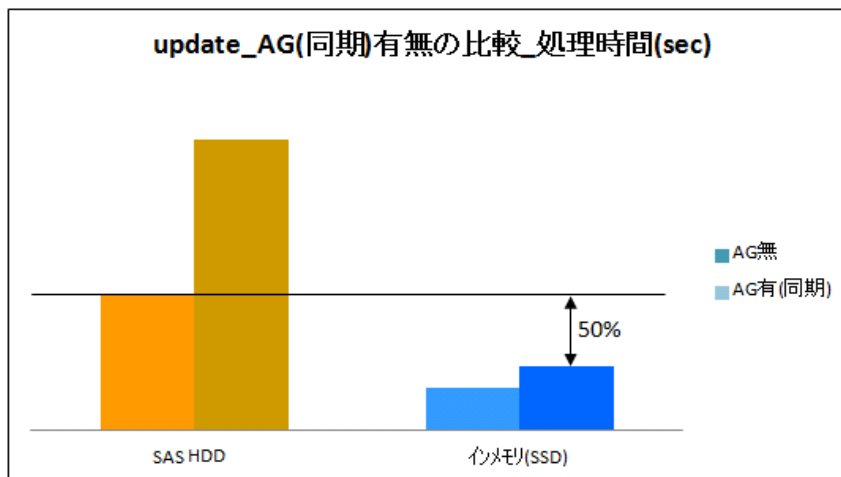


図 4.6 update 時の AG 無と AG 有(同期)の処理時間

上記結果から以下を確認しました。

- ①insert 処理において、AlwaysOn 可用性グループ無での SAS HDD と AlwaysOn 可用性グループ有でのインメモリ OLTP (ファイルグループとログ領域は SSD)を比較すると最大 50%の性能の向上を確認しました。
- ②update 処理において、AlwaysOn 可用性グループ無での SAS HDD と AlwaysOn 可用性グループ有でのインメモリ OLTP (ファイルグループとログ領域は SSD)を比較すると最大 50%の性能の向上を確認しました。

4.2.3 AlwaysOn 可用性グループの同期型と非同期型の性能

AlwaysOn 可用性グループの同期型と非同期型を用いて insert の T-SQL 文と update の T-SQL 文を SSD のドライブに対して実施しました insert と update の実行には SQL Server Management Studio を利用しました。

以下に処理時間の結果を示します。

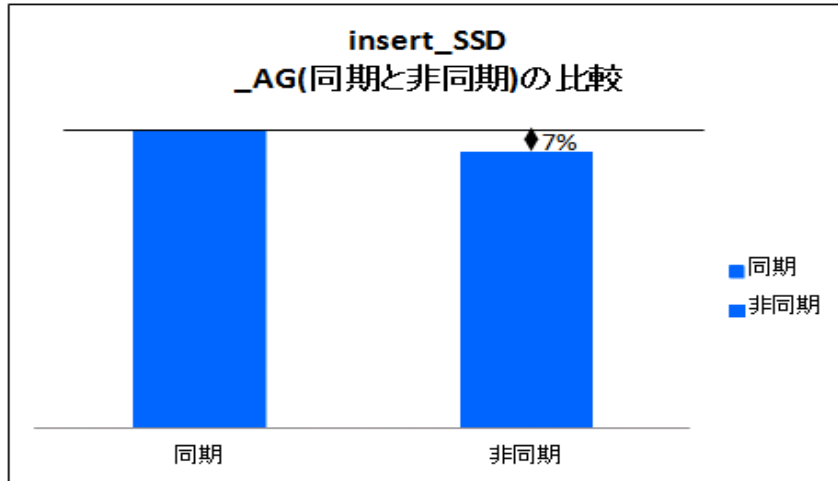


図 4.7 insert 時の AG(同期)と AG 有(非同期)の処理時間

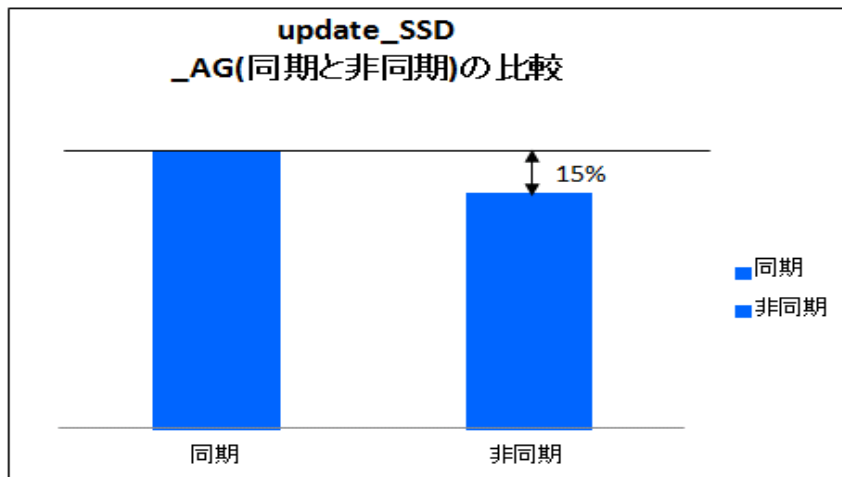


図 4.8 update 時の AG(同期)と AG(非同期)の処理時間

上記結果から以下を確認しました。

- ① insert 処理において、同期型と比べて非同期型が最大 7%の性能の向上を確認しました。
- ② update 処理において、同期型と比べて非同期型が最大 15%の性能の向上を確認しました。

4.3 各デバイスの混在環境における性能傾向を確認

デバイスを組み合わせて性能測定を実施しました。組み合わせは、データ領域に PCIe FLASH ドライブボード、デバイス障害時に RAID 再構成ができるようログ領域に SSD(図は PCIe FLASH ドライブボード+ SSD と記載)としました。AlwaysOn 可用性グループは同期型としました。insert と update の実行には SQL Server Management Studio を利用しました。以下に応答時間の結果を示します。

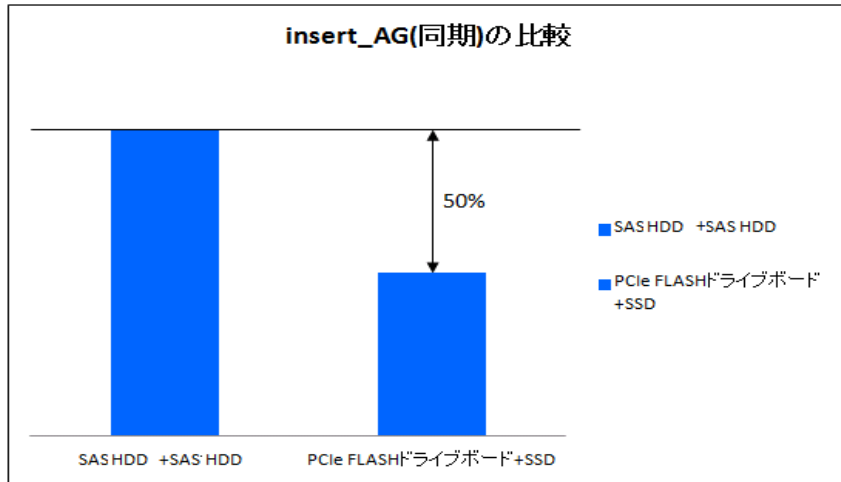


図 4.9 insert 時の組み合わせの処理時間

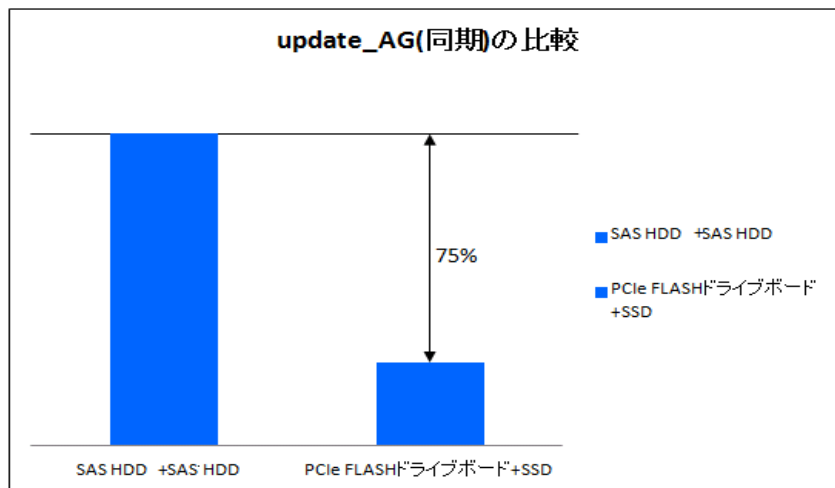


図 4.10 update 時の組み合わせの処理時間

上記結果から以下を確認しました。

(1)insert の T-SQL 文

データ領域に SSD、ログ領域に PCIe FLASH ドライブボードを搭載した構成での処理時間は、両領域に SAS HDD を搭載した構成と比べて 50%向上することを確認しました。

(2)update の T-SQL 文

データ領域に SSD、ログ領域に PCIe FLASH ドライブボードを搭載した構成での処理時間は、両領域に SAS HDD を搭載した構成と比べて 75%向上することを確認しました。

5 まとめ

今回、HA8000 と SQL Server の可用性の機能を利用した高性能、高可用性システムを検証した結果を以下にまとめます。

(1)フラッシュドライブと AlwaysOn 可用性グループを利用することは有効です。

SAS HDD を利用したスタンドアロン環境と比べて、フラッシュドライブを利用した AlwaysOn 可用性グループ環境は高性能と高可用性を備えたシステムとなります。今回、最大 50%の処理時間の向上を確認しました。

(2)AlwaysOn 可用性グループの同期型と非同期型を用途に合わせて選択することは有効です。

性能を重視し、災害対策の用途を選択する場合は非同期型を検討し、障害時の自動切換えを重視する場合は非同期型を検討します。今回、同期型と比べて非同期型は最大 15%の処理時間の向上を確認しました。

(3)フラッシュドライブの混在環境を検討することは有効です。

ユーザ要件に対してフラッシュドライブの特性(RAID 制限など)が満たせる場合はデバイスの組み合わせは有効です。今回、データ領域に SSD デバイス、ログ領域に PCIe FLASH ドライブボードを搭載した環境では、両領域に SAS HDD を搭載した環境と比べて最大 75%の処理時間の向上を確認しました。

6 注意事項

(1)非同期型の同期タイミングについて

AlwaysOn 可用性グループの非同期型について、セカンダリサーバでのデータの反映状態 つまりプライマリサーバとの差は `sys.dm_hadr_database_replica_states` 関数で確認できます。確認できる項目としては、`log_send_queue_size` にてプライマリサーバからセカンダリサーバへの転送状況が確認でき、`redo_queue_size` にてログ領域への書き込み状況が確認できます。

特に、ディザスタリカバリ対策で非同期型を利用する際は復元ポイントに関わります。

`log_send_queue_size`:セカンダリサーバのデータベースに送信されていない、プライマリサーバのデータベースのログ レコードの量 (KB 単位)。

`redo_queue_size`:まだ再実行されていないセカンダリサーバのトランザクションログ内のログ レコードの量 (KB 単位)。

【状態確認コマンド】

< <http://msdn.microsoft.com/ja-jp/library/ff877972.aspx>>

(2)ツールの効果について

ツールは参考値であり、ユーザ環境に合わせてデバイスを選択する様注意してください。

今回、ツールの結果から SSD は Read 処理の速度が速く、PCIe FLASH ドライブボードは小トランザクション(8KB)での Write 処理が速いと確認できました。しかし、SQL Server Management Studio を利用した insert 処理、update 処理では大きな違いが確認できませんでした。

(3)フラッシュドライブの性能について

フラッシュメモリでは既にデータが存在する領域に直接データを上書きすることができず、事前にメモリ上のデータの消去が必要です。そのため、フラッシュメモリではデータの上書きが発生した場合に、まずデータが存在しない領域に書き込み、後から不要になったデータ領域を消去します。このようにして消去処理にかかる時間を隠蔽します。しかし、長時間にわたる大量の書き込みが発生した場合に、この消去処理が追いつかず、一時的に Write 性能が低下する場合があります。

7 付録

(1)フラッシュドライブの交換方法

HA8000 のフラッシュドライブが容量制限となった場合、2 点の手段で交換できます。

①業務稼働中の交換

HA8000 に新 SSD を搭載してリビルドにより旧 SSD から新 SSD の RAID グループに交換します。

ただし、リビルドによる性能劣化のため負荷が低い時間帯に作業する必要があります。

②業務停止中の交換

データを他媒体にバックアップし、旧 SSD 環境から新 SSD 環境へのディスク交換後に、バックアップした媒体からリストアします。

(2)フラッシュドライブの利用制限の確認

サーバ管理ツールである Hitachi Server Navigator - Hitachi RAID Navigator をインストールして、コマンドラインにて「hrccli -pdinfo -aall」(SSD)、「hrccli -pssdinfo -aall」(PCIe FLASH ドライブボード)を実行することで利用算容量を確認できます。

表 7.1 コマンド実行例

###	SSD Predictive Information ###
Replacement Necessity(wearout Only) :	Not replacement
Replacement Reason	: -
Rated PBW	: 17.00 PB
Current PBW(MB)	: 66481296.57 MB
Remaining PBW(%)	: 99.64 % remaining
Remaining PBW(%)	: Warning at 15 %
Remaining PBW(%)	: Error at 10 %
Reserve space	: 100 %

【Hitachi Server Navigator ダウンロードページ】

<http://www.hitachi.co.jp/products/bladesymphony/download/driver/12345900.html>

<http://www.hitachi.co.jp/products/bladesymphony/download/driver/12345901.html>

(3)SQL Server の参考情報

以下に SQL Server 2014 の代表的な新機能とエディションを記載します。

表 7.2 SQL Server 2014 の代表的な新機能とエディション

項目	機能	新機能・改善ポイント	搭載 エディション		
			Enterprise	Business Intelligence	Standard
パフォーマンス向上	インメモリ OLTP	データベースをサーバーのメモリ上に配置することが可能となりました	○	—	—
	更新可能なインメモリカラムストア (列ストアインデックス)	SQL Server 2012 で新機能としてサポートされた、読み取り専用として列ストアインデックスが更新可能となりました	○	—	—
	バッファプール拡張 (SSD へのメモリ拡張)	サーバーのメモリが枯渇した場合に SSD をデータベース バッファプール領域として利用可能となりました	○	—	—
	IO リソース ガバナンス	リソース プールにおいて、ボリュームごとの最小および最大の IOPS の構成がサポートされ、リソース分離をより包括的に制御できるようになりました	○	—	○
高可用性	Always On 機能拡張 (AlwaysOn 可用性グループ)	可用性グループが最高で 8 つまでのセカンダリレプリカをサポートするようになり、ネットワークエラーの場合でも、常にレプリカを読み取ることができるようになりました。フェールオーバー クラスタ インスタンスが Windows クラスタ共有ボリュームをサポートするようになり、共有記憶域の使用が改善され、フェールオーバーの回復性が高まりました	○	—	—
セキュリティ	バックアップの暗号化	社内および Windows Azure におけるバックアップの暗号化ができるようになりました	○	○	○
クラウド機能強化	SQL XI (XStore 統合)	Windows Azure BLOB ストレージを SQL DB として統合利用ができるようになりました	○	○	○
クラウドと オンプレミスとの連携	Windows Azure ストレージへの バックアップ	Windows Azure ストレージへの社内および Windows Azure の SQL Server バックアップを管理したり、自動化することができます	○	○	○
	ウィザード形式での Azure レプリカ	Windows Azure のレプリカを簡単に社内の可用性グループに追加できます	○	○	○

SQL Server 2014 の各エディションがサポートする機能については、必要に応じて確認ください。

【SQL Server 2014 の各エディションがサポートする機能】

[http://msdn.microsoft.com/ja-jp/library/cc645993\(v=sql.120\).aspx](http://msdn.microsoft.com/ja-jp/library/cc645993(v=sql.120).aspx)

(4)インメモリ最適化アドバイザの参考情報

インメモリ最適化アドバイザ機能は SQL Server 標準の機能で、インメモリ OLTP を利用することで性能向上の効果が得られるか確認するツールです。以下に既存のデータベースのテーブルをインメモリに展開する処理を操作した例を記載します。



図 7.1 メモリ最適化アドバイザ機能

(5)ファームウェア・ドライバー一覧

今回、検証で利用したファームウェア・ドライバー一覧です。

(3)HA8000/220 (BIOS:American Megatrends Inc. MA.1.01.00, 2013/08/29)

- ①intel_05.exe
- ②matrox_06.exe
- ③h8kw2012r2bcmlan.exe
- ④heci_02.exe
- ⑤Fusion-io_3.2.3.950_x64.exe
- ⑥Fusion-io_Management_Server_3.3.4.267_x64.exe
- ⑦win_hrn_1.5.0.4.exe

(2)WindowsServer2012R2 SP1 SE (Version:6.3.9600 ビルド 9600)

- ①Windows NIC 用ドライバ 16.2.0.4 Broadcom
- ②Windows HBA 用ドライバ 4.4.7.1110 Hitachi
- ③.NET Framework Microsoft .NET Framework 3.5 SP1
- ④セキュリティパッチ 適応なし

(3)SQLServer2014 (Version:12.0.2342.0)

- ①セキュリティパッチ cu1 2931693

(6)SQL Server の参考情報

【AlwaysOn 可用性グループ 実習書】

<http://www.google.co.jp/url?url=http://www.microsoft.com/ja-jp/SQLServer/2014/technology/self-learning.aspx&rct=j&frm=1&q=&esrc=s&sa=U&ei=yhipU6GWGISvIQWwh4CADg&ved=0CB0QFjAB&usg=AFQjCNE4AdmEOMKwlm20pB2mOYVfeEjGTg>

【インメモリ OLTP 実習書】

<http://www.google.co.jp/url?url=http://www.microsoft.com/ja-jp/SQLServer/2012/technology/self-learning.aspx&rct=j&frm=1&q=&esrc=s&sa=U&ei=IRmpU4-yJ8PIkAWDzYHoAw&ved=0CBcQFjAA&usg=AFQjCNFkmi2BhfpX22S5WXNwthWC0Mk2g>

【SQL Server 2014 CTP1 の新機能の概要/インメモリ OLTP (Hekaton) 機能の概要】

<http://www.microsoft.com/ja-jp/sqlserver/2014/technology/self-learning.aspx>

【SQL Server 2014 のエディションとコンポーネント】

[http://msdn.microsoft.com/ja-jp/library/ms144275\(v=sql.120\).aspx](http://msdn.microsoft.com/ja-jp/library/ms144275(v=sql.120).aspx)

【SQL Server 2014 の各エディションがサポートする機能】

[http://msdn.microsoft.com/ja-jp/library/cc645993\(v=sql.120\).aspx](http://msdn.microsoft.com/ja-jp/library/cc645993(v=sql.120).aspx)