DBサーバー統合におけるNUMA対応IAサーバーの性能評価

谷本 雄一郎*

Performance Evaluation of IA Servers based on NUMA Architecture in DB Server Consolidation

Yuichiro Tanimoto*

本稿では, NUMA(Non-Uniform Memory Access)アーキテクチャを採用したIBM社のハイエンドIA(Intel® Architecture)サーバーであるxSeries® 440と, NUMA対応OSであるMicrosoft®社のWindows® Server 2003 Enterprise Editionを用い, 32ビット環境におけるDBサーバー統合の現実性について性能面から考察した. 考察において, 具体的な性能数値を測定して評価を行うことにより, NUMA対応IAサーバーがDBサーバー統合の負荷に耐えうるプラットフォームであるという結論を導き出している. 加えて, NUMA環境において効果的な性能を実現するための方法や64ビット環境への展望についても言及する.

This paper discusses the feasibility of DB server consolidation from performance viewpoint under 32-bit computing environment. The evaluation utilizes a combination of IBM xSeries® 440 and Microsoft® Windows® Server 2003 Enterprise Edition. IBM xSeries 440 is one of the IA (Intel® Architecture) servers, adopting the NUMA (Non-Uniform Memory Access) architecture. Microsoft Windows Server 2003 Enterprise Edition is a NUMA-aware operating system. The performance of the servers is evaluated by detailed numerical output from tests. The output shows that DB server consolidation with NUMA architecture IA Server is effective in a real world context. In addition, the most effective implementation of the NUMA environment is also discussed, as well as the prospect on 64-bit environment.

Key Words & Phrases：データベース, サーバー統合, 非均等メモリ・アクセス, IAサーバー, 性能
DB, Server Consolidation, NUMA, IA Server, Performance

1. はじめに

昨今, IAサーバーの高性能化により, 従来では考えられなかったような膨大な処理を行うことが可能となってきている. 加えて, 分散環境でのサーバー管理が高コストのため, TCO(Total Cost of Ownership)削減への関心が強まっている. そのため, IAサーバーの世界ではハイエンド・マシンによるサーバー統合の需要が高まっている. このニーズに応えるべく, 多くのベンダーがハイエンドIAサーバーを発表している. その中の一つであるIBM社 xSeries® 440は, 最大16プロセッサ, 64GBメモリをサポートしているが(2003年8月時点), 特徴的であるのは, 他ベンダーのサーバーと異なるNUMAと呼ばれる技術を採用したことである. 詳細については後ほど触れてゆくが, この技術により, 従来のSMP(Symmetric Multi Processing)マシンの課題点であるパフォーマンスのスケーラビリティについて改善を図っている.

本稿では, 2003年6月にMicrosoft®社より発売された, NUMA対応OSであるWindows® Server 2003 Enterprise Edition(以下Windows 2003)とxSeries 440の組み合わせが, DBサーバー統合の膨大な負荷に耐えうるプラットフォームであることを検証してゆく. 検証にあたっては, 2000年2月に発売されたWindows 2000 Advanced Server(以下Windows 2000)と当時のハイエンドIAサーバーであるIBM社 Netfinity 8500Rを統合前サーバーと仮定し, 具体的な性能数値をもとにサーバー統合の現実性を考察してゆく.

提出日：2003年8月29日
*yusa@jp.ibm.com

2．DBサーバー統合の種類

ここでは, どのような統合の種類があるかを述べ, 実際に検証する統合について確認する.

まず, 統合の種類であるが, 統合サーバーに関してのWebページ[1]を参考に, **表1**のように4種類考えた.

Aの垂直統合型は, 全国展開しているような組織において, データを都道府県ごとに管理している場合を想像してもらえると分かりやすい. 各DBのスキーマは同一であるため, データを統合対象のサーバーに垂直に積み上げてゆく形となる. この場合, 管理対象のDB個数とサーバーマシンの台数の両者を削減することが可能である.

Bの水平統合型は, 財務DB, 人事DB, 経理DBなどが, それぞれ財務部門, 人事部門, 経理部門により, 別々のサーバー上で運用されているような場合である. これらを1台のサーバーに統合する場合, 各DBのスキーマには手を加えず, そのまま統合先のサーバーのインスタンスに横並びに, つまり水平に配置する. この結果, 管理対象のDB数は変化しないものの, 管理対象のサーバーマシンを削減することができる.

Cの再設計統合型は, 業務プロセスのリエンジニアリングに伴うDB統合などのケースに多いと考えられる. これについては, DBスキーマの再構築をした後は, 統合対象がデータとなるため, 最終形は垂直統合型と同じであると見なすことができる. また, Dに関しては, AからCまでを組み合わせたものであり, それらの検証の結果をもとに推測可能だと考え, 検証対象からはずした. したがって, Aの垂直統合型と Bの水平統合型の二つを検証対象とする.

3．検証環境

検証では, TPC-Cベンチマーク[2]のスタイルで, クライアントマシンであるxSeries 360（Xeon MP 1.6GHz×4, 8GBメモリ）からDBサーバーに負荷を与えた.

DBサーバー部分は, 統合前サーバーであるNetfinity 8500Rと統合後サーバーであるxSeries 440を入れ替え, 各サーバーが1秒間に処理するトランザクション量を計測した. ストレージの部分に関しては, ハードウェア, 論理ディスク構成とも全く同一のものを使用した. 理由は, ストレージの違いから発生するI/O処理性能の差をできるだけ極小化するためである.

DBMSについては, Microsoft SQL Server 2000 Enterprise Edition（以下SQL Server 2000）を選択した. 選択した一番の理由は, TPC-Cベンチマークなどの代表的なベンチマークにおいて, 既にWindows 2003上での稼動実績があることがあげられる. また, Windowsプラットフォーム上での販売実績の豊富さという点で[3], 本稿が多くの読者に役立つと考えた.

表1. DBサーバー統合の4つの類型

種　類	内　容
A　垂直統合型	同一のスキーマを持ちデータが異なる複数のDBを一つのDBに統合
B　水平統合型	異なるスキーマを持つ複数のDBを一つまたは複数のインスタンスに統合
C　再設計統合型	異なるスキーマを持つDB同士のスキーマ構造を再設計し一つのDBに統合
D　その他	AからCまでの任意の組み合わせ

3.1　検証対象のハードウェア

図1は, 今回の検証で使用するxSeries 440とNetfinity 8500Rの特徴をまとめたものである. 図中のHTは, Hyper Threadingテクノロジの略で, Xeon MPの方にのみ実装されている. この技術を用いて製造されたプロセッサは, その内部に, アーキテクチャ・ステートと呼ばれるスレッドを受付けるモジュールを通常の倍の2個実装している. このため, 1スレッドの処理だけではプロセッサの実行リソースに余裕があるような場合, 2スレッドを同時に処理することで同時実行性が上がり, 性能が向上するという訳である[4]. この技術を使用すると, OSからは物理的に1個のプロセッサが, 論理的に二つのプロセッサとしてカウントされる[5]. つまり, xSeries 440の8個のXeon MPプロセッサは, OSからは論理的に16個のプロセッサとして認識される. また, FSBはFront Side Busの略で, CPU, メモリ・コントローラ, メモリなどをつなぐデータ伝送路であり, 数値は伝送スピードを表している. この数値が大きければ大きいほど, プロセッサやメモリなどのデバイス間のデータ交換が高速化される. 設計については次で詳しく説明する.

Netfinity 8500R 	CPU	Pentium® III Xeon 550 MHz×4 - 32KB L1 Cache - 2MB L2 Cache
	HT	なし
	FSB	100MHz（800MB/sec）
	メモリ	8GB
	設計	SMPアーキテクチャ
xSeries 440 	CPU	Xeon MP 2.0GHz×8 - 20KB L1 / 512KB L2 Cache - 2MB L3 Cache - 64MB L4 Cache（Chipsetに実装）
	HT	あり
	FSB	400MHz（3.2GB/sec）
	メモリ	16GB
	設計	NUMAアーキテクチャ

図1. 比較対象となる新旧IAサーバー

3.2 ハイパフォーマンス・アーキテクチャ

図2は, SMPアーキテクチャを採用するNetfinity 8500Rの内部構造図である. このアーキテクチャは, 比較的単純な設計であるわりには, 4プロセッサ程度までであれば問題なく性能を発揮する. しかし, プロセッサやI/Oデバイスが増えた場合, 負荷が単一のメモリ・コントローラやFSBに集中してしまう. つまり, プロセッサが8個, 16個と増加してゆくと, スケーラビリティを失う[6].

一方, NUMAアーキテクチャを採用するxSeries 440の内部構造は**図3**のようになる. 4個のプロセッサ, メモリ, メモリ・コントローラ, I/Oコントローラを1セットとして, CEC（Central Electronics Complex）と呼ばれる箱型のモジュール内に配置している. そして, インター・コネクト用のスケーラビリティ・ケーブルでCEC同士を連結している[7]. この設計では, メモリ・コントローラやFSBに対する負荷が複数のCECで分散されるため, プロセッサの数を増やした際にSMPと比較してリニアなパフォーマンス向上が期待できる. また, 複数のCECを持つxSeries 440は, SMPのサーバーと同様に, OSからは1台のサーバーとして認識される. したがって, SMP環境下で動作する一般的なアプリケーションは, xSeries 440上で修正することなく実行することができる.

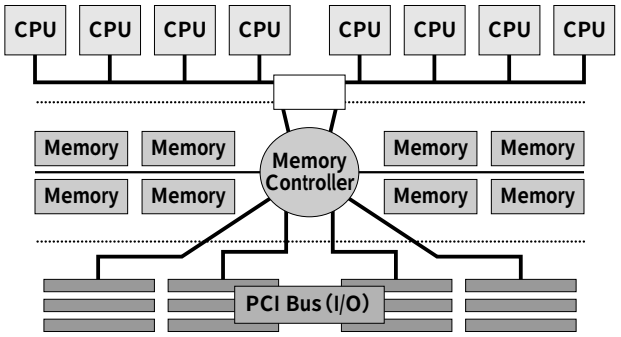

図2. Netfinity 8500R内部構造図

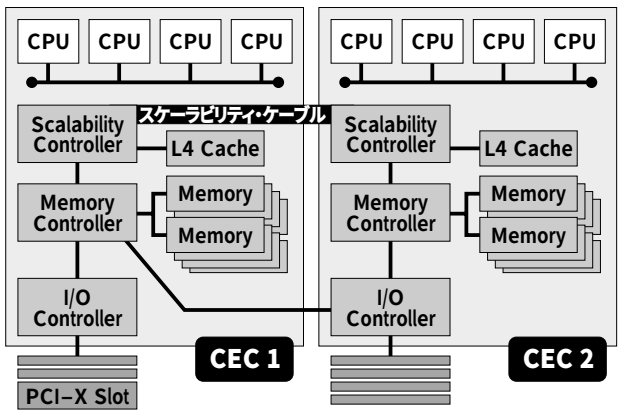

図3. xSeries 440内部構造図

4. 性能評価

4.1 概要

図4は, xSeries 440による2種類のDBサーバー統合を, Netfinity 8500Rの性能と比較した結果である. 垂直統合型の場合で4倍, 水平統合型の場合で約6倍の性能向上を実現することができた. この結果から, 垂直統合の場合は4台のNetfinity 8500Rを1台のxSeries 440に, 水平統合の場合は6台の Netfinity 8500Rを1台のxSeries 440に統合可能だと推測することができる.

次は, 二つの統合型の性能差がどのような原因から発生しているか, 性能を左右する要因を確認した後, 垂直統合型, 水平統合型と順を追って考察することにより明らかにしてゆく.

4.2 性能に影響を与える要因

今回のように, 32ビット版Windowsの環境では, アプリケーションのプロセスが使用できるメモリ空間は基本的に4GB以内に制限される. 1プロセスがそれ以上のメモリ空間を使用する場合, PAE（Physical Addressing Extension）[8]と呼ばれるIntel®社のプロセッサに実装された機能と, AWE（Address Windowing Extensions）[9]と呼ばれるSQL Server 2000に実装された機能を使用することにより可能となる. ただし, この場合もプロセスからネイティブに使用できるメモリ空間はあくまで4GBであり, PAE, AWEで得られた4GB以上のメモリ空間というのは高速なスワップ領域という位置付けに近い. このため, 32ビット環境において, 1プロセスが4GB以上のメモリを使用する際はオーバーヘッドが発生する. 例えば, 8GBのメモリ空間がある場合, 1プロセスで全てのメモリを使用するよりも, 2プロセスで4GBずつ分割した方が, プロセスがネイティブに使用できるメモリが多くなる. その結果として, システム全体で見た場合のスループットは向上する.

次に, メモリ・アクセスのレイテンシー（遅延）につ

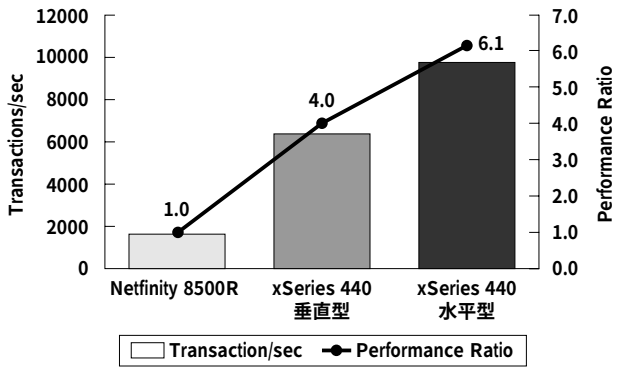

図4. 各統合型での性能向上の差

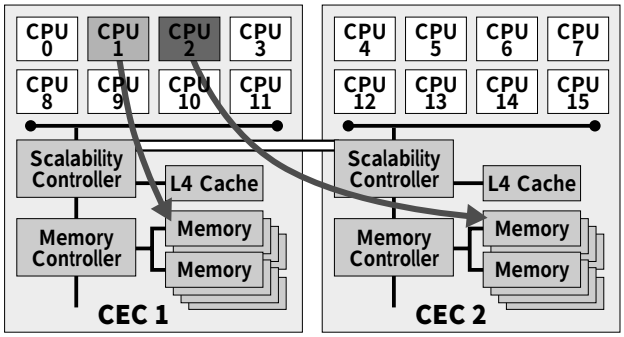

図5. 異なるメモリ・アクセス

いて述べる. 既に確認したように, NUMAアーキテクチャを実装するxSeries 440で8Wayのプロセッサ構成をするためには, 二つのCECをスケーラビリティ・ケーブルで接続する必要がある. この場合, **図5**にあるように2通りのメモリ・アクセスのパターンが考えられる. 一つは, CPU1のように, 一番近いCEC1のメモリにアクセスする, ローカル・メモリ・アクセスである. もう一つは, CPU2のように, 他のCECのメモリにアクセスする, リモート・メモリ・アクセスである. 両者の差異は, データの取得に要する時間である. ローカル・メモリ・アクセスの場合, 少ないCPUサイクルでデータを取得できるのに対し, リモート・メモリ・アクセスでは, より多くのCPUサイクルを必要とする. したがって, システム性能を向上させるためには, できるだけリモート・メモリ・アクセスを抑えることが重要となる[10].

このような要件に対し, OSであるWindows 2003がどのように動作するかを確認する.

四つのEditionを持つWindows Server 2003のうち, Enterprise EditionとDatacenter Editionの二つは, NUMA対応サーバーのノード(CEC)情報を持つことが可能である. xSeries 440のFirmware内部にはSRAT(Static Resource Affinity Table)と呼ばれる, CPUやメモリの位置情報などが格納されるテーブルが存在する. Windows 2003は, 起動時にこの情報を読み取ることでプロセッサとメモリの位置関係を把握することが可能となる. これにより, アプリケーションに最適なノードのCPUを割り振り, 必要なデータを可能な限りローカル・メモリに配置する[11].

4.3 垂直統合型

今回の検証では, データを1個のDBに統合しているので, DBは1インスタンス上で実行されている. SQL Server 2000では, インスタンスごとにプロセスが生成されるため, このケースではプロセスは1個となる. そして, 実際にはプロセスから派生する多数のスレッドにより処理が行われる. この際, 本来なら避けるべきリモート・メモリ・アクセスがある程度発生してしまうことに気づく(**図6**). 例えば, CPU8が使用するデータはWindows 2003のNUMA対応によりローカル・メモリであるCEC1のメモリにロードされる. その後CPU4が同じデータを使用する場合, CEC1のメモリに対して, リモート・メモリ・アクセスが発生してしまう. しかし, このリモート・メモリ・アクセスは必ずしも不適切とは言えない. なぜならば, 1プロセスが全てのCPUを使用する場合において, リモート・メモリ・アクセスを発生させない設計とは, SMPアーキテクチャであるからだ. これについては, 既に, スケーラビリティに問題があると述べた. 一般的に, CPUが増加すればする程, NUMAのリモート・メモリ・アクセスのペナルティは, SMPの単一メモリ・コントローラとFSB帯域の共有によるペナルティよりも相対的に軽くなる.

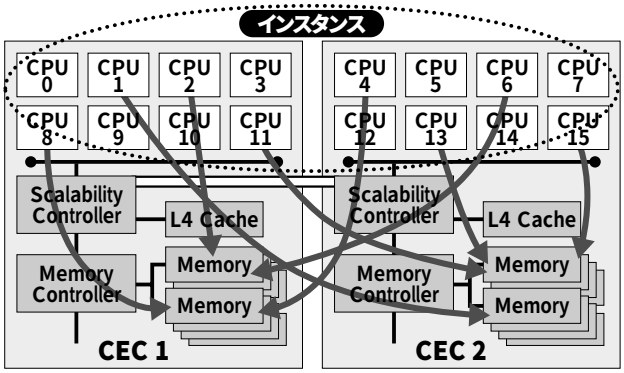

図6. 1プロセスでのメモリ・アクセス

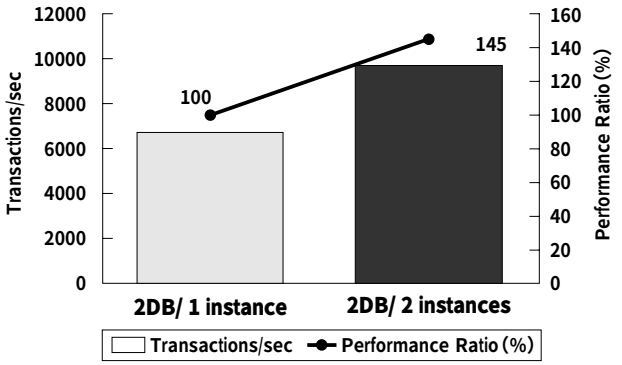

図7. インスタンス数と性能

4.4 水平統合型

複数のDBをSQL Server 2000上で運用する際, 最小で1インスタンス, 最大でDBの数のインスタンスで運用することが可能である. 性能評価では, 2個のDBを使用した. その際, インスタンスの数を1から2と変えることにより, 性能は**図7**のように変化した. 2インスタンスの性能の方が, 45%程度高くなっているが, これには二つ理由がある.

一つ目は, 2インスタンスの場合, プロセスが2個生成されることで, それぞれのプロセスがネイティブに

アクセスできる4GB以内のメモリが増加するためである. 二つ目は, 2インスタンスの場合, リモート・メモリ・アクセスを減少させる構成が可能なためである. 1インスタンスの場合は, 2DBでも1プロセスとなるため, 図6と同様のメモリ・アクセスとなる.

4.5 複数インスタンス構成の際の注意

今回の性能測定では, 二つのインスタンスに対して, 事前にNUMAの特性を利用してSQL Server 2000のインスタンスに最適な設定を行っている. そこで, 次は, インスタンスに適切な設定を行った場合と, そうでない場合で, どの程度の性能の差が生じるかを検証する. ここでは, インスタンスに対するプロセッサの割り当て方を変えることにより, 三つのパターンを用意した. 具体的には, SQL Server 2000の二つのインスタンスのそれぞれに, **表2**のようにプロセッサを設定した(数字はCPU番号). この時, プロセッサが所属するCECにも注目してほしい. Pattern A, Pattern Bではインスタンスにそれぞれ8個ずつプロセッサを設定しているが, 割り当て方が異なっている. Pattern Cでは, 各インスタンスに全てのプロセッサを割り当てている(インストール時のデフォルトはPattern C).

図8は性能測定の結果である. 図8では, Pattern Aを基準(=100%)に設定している. つまり, Pattern Aが最適な設定という訳である. これ以外のパターンでいずれも性能は低下しているため, 次にPattern A以外がなぜ効率的でないかを考察してゆく.

Pattern Bでは, 各インスタンスに割り当てられているプロセッサがCECをまたいでいるために, メモリに展開されるデータも各CECに分散されることになる. 例えば, **図9**でCPU0が使用するデータが, ローカル・メモリであるCEC1のメモリに展開されたとする. その後, CEC2のCPU5がそのデータを処理する場合は, CEC1にリモート・メモリ・アクセスを行わなければならない.

これに対し, Pattern Aでは, 各インスタンスを実行するプロセッサはCECごとに分割されている(**図10**). したがって, 各インスタンスが使用するデータも, 可能な限り各CECのメモリ上に配置される. この結果, プロセッサはローカルのメモリにアクセスすることになり, 性能が最適化される.

このことを実証するために, 特殊なツールを使用してPattern AとPattern Bのリモート・メモリ・アクセスの割合を計測した(**表3**). これは, 全てのメモリ・アクセスのうちリモート・メモリ・アクセスが占める割合を示したものである. Pattern AはPattern Bに比べリモート・メモリ・アクセスが少なく, その結果として性能が高くなっているということができる.

以上述べた理由から, 複数インスタンスを構成する場合, Pattern Bのように各インスタンスがCECをまたぐようなCPUの設定はすべきではない. 他方, SMP環境下においては, 各CPUからメモリまでの距離が等しいため, これらの設定を必要としない. しかし, 何度も繰り返すように, CPUの増加に値するパフォーマンスが得られないことが懸念される.

また, ここまで触れていないPattern Cについては, CPUリソースが競合し, CEC間に大量のリモート・メモリ・アクセスが発生しているために, パフォーマンスが著しく劣化しているケースである. 一般的に, このような構成はNUMAであるかどうかに関わらず, 行うべきではない.

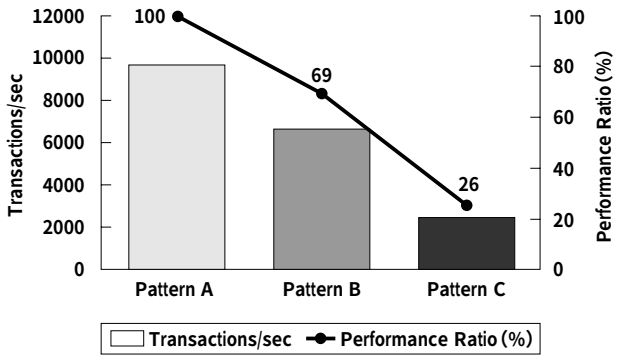

図8. インスタンスの設定と性能

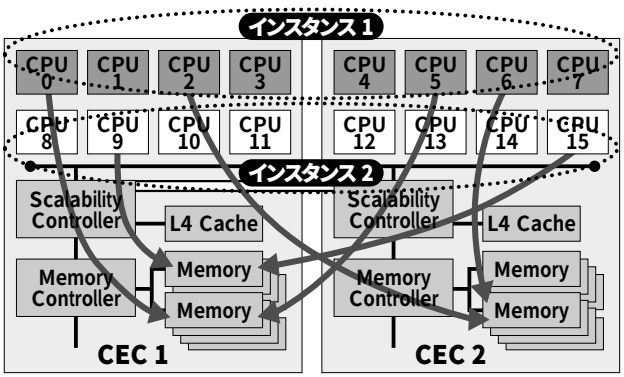

図9. Pattern Bのメモリ・アクセス

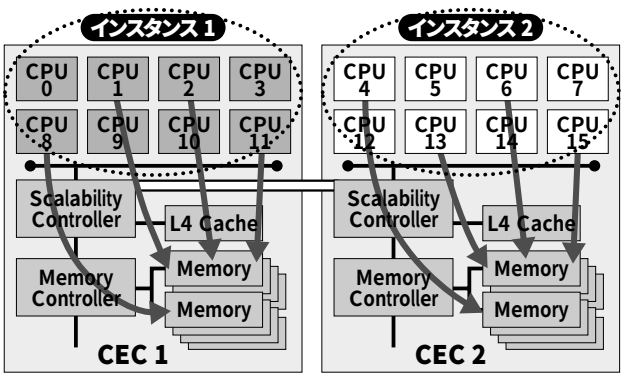

図10. Pattern Aのメモリ・アクセス

表2. 各Patternで使用されるCPU

	Instance 1								Instance 2							
	CEC 1				CEC 2				CEC 1				CEC 2			
Pattern A	0	1	2	3									4	5	6	7
	8	9	10	11									12	13	14	15
Pattern B	0	1	2	3	4	5	6	7								
									8	9	10	11	12	13	14	15
Pattern C	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15	8	9	10	11	12	13	14	15

表3. リモート・メモリ・アクセスの割合

	CEC 1	CEC 2
Pattern A	約6%(5.780777325%)	約6%(6.279928528%)
Pattern B	約33%(33.09079834%)	約35%(35.10074226%)

5. まとめと展望

これまで見てきたように, Windows 2003とxSeries 440を用いたシステムは, 適切な設定を行えばDBサーバー統合の負荷に耐えうるプラットフォームであると確認できた. このことから, NUMAアーキテクチャIAサーバーにおけるDBサーバー統合は, 現実性のあるソリューションであると考える. ただし, 本稿で直接触れた以外にもいくつかの考慮点がある.

一例として, NUMA環境では, ネットワーク・アダプタやディスク・サブシステムの接続方法などによって, パフォーマンスは大きく変化する場合がある. 例えば, SQL Server 2000と複数のVI(バーチャル・インターフェイス)ネットワーク・アダプタを使用することにより, どのノードのアダプタを使用すればノード間のリモート・メモリ・アクセスを極小化できるかを判断し, 性能を向上させることが可能である.

このように, NUMA大規模サーバーシステムの設計にはハードウェア, ソフトウェアの両方に精通したスキルが必須となる. ただし, それらは前提条件であり, 真に高性能なシステムを実現しようとするならば, システム導入前には実機を用いた性能評価をすることをお勧めしたい.

また, NUMA環境においてパフォーマンスを向上させるには, ソフトウェア側でもこれを意識した設計が必要だと考える. 特に, 垂直統合型では, プロセス内のスレッドはCECをまたがって実行されることになり, 比較的多くのリモート・メモリ・アクセスが発生してしまう. これを改善する一つの案として, 1インスタンスであっても複数のプロセスが生成され, それらがCECごとに配置されるような仕組みが考えられる. 具体的には, CEC1とCEC2にそれぞれ一つずつプロセスを生成し, データアクセスの際に, 極力ローカル・メモリ・アクセス行うように設計すればよい.

別のアプローチとしては, 64ビット環境を利用する方法がある. Windows 2003とSQL Server 2000の場合, 従来の32ビット版だけでなく64ビット版も用意されている. そのため, ハイエンド・ソリューションの分野では, ソフトウェアの対応や周辺機器の拡充とともに, IA64が採用されるケースも増えてゆくと予想される. これは, DBサーバー統合にもパフォーマンス面で良い影響を与えてくれる. 例えば, 今回の検証でIA64を使用したならば, 4GB以上のメモリ・アクセスの際に, オーバーヘッドなしにアクセス可能となるため性能は向上する. その結果, 1台のCECに16GB(xSeries 440の制限)以上のメモリを搭載することで, パフォーマンスが得やすいローカル・メモリ・アクセスの機会を増やすことができる. ただし, CECごとの搭載メモリが増加すると, リモート・メモリ・アクセスが発生した際, スケーラビリティ・ケーブルを流れるデータ量がIA32環境よりも大量になることが懸念される. しかし, これはL4キャッシュ[12]を大容量化することにより緩和できると考える.

謝辞

執筆にあたり, 米国Microsoft社のSQL Server製品マーケティングチーム, 米国IBM社のJoseph J. Jakubowski氏, Mark V. Kapoor氏, 日本アイ・ビー・エム株式会社の溝上敏文氏, 田坂岳氏をはじめ, 多くの方々よりさまざまなサポート及びアドバイスを頂きました. あらためて感謝いたします.

参考文献

[1] e-business時代をリードするS/390エンタープライズ・サーバー, http://www-6.ibm.com/jp/software/s390/conference/images/J_m6.pdf, 2003.8.28

[2] TPC-C V5, http://www.tpc.org/tpcc/default.asp 2003.8.28

[3] SQL Server 2000の市場優位性, https://www.microsoft.com/japan/sql/evaluation/compare/prk/vsOracle1_1.asp, 2003.8.28

[4] Deborah T. Marr, *Hyper-Threading Technology*

Architecture and Microarchitecture, Intel Technology Journal Q1, 2002
［5］John Borozan, *Microsoft Windows-Based Servers and Intel Hyper-Threading Technology*, Microsoft Corporation, 2002
［6］Ian Bramley, *Intel Server Architectures -Diversity Now Blooms*, Butler Direct Limited, pp.12-15, 2000
［7］David Watts, et al., *IBM xSeries 440 Planning and Installation Guide*, *IBM Red Book*, 2002
［8］Physical Address Extension, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/memory/base/physical_address_extension.asp, 2003.8.28
［9］Using AWE Memory on Windows 2000, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_sa_6b3k.asp, 2003.8.28
［10］B. C. Brock, et al., *Experience with building a commodity Intel-based ccNUMA system*, IBM Journal of Research and Development, Volume 45 Issue 2, 2001
［11］Application Software Considerations for NUMA-Based Systems, http://www.microsoft.com/whdc/system/platform/server/datacenter/numa_isv.mspx, 2003.8.28
［12］Mark T. Chapman, *Introducing IBM Enterprise X-Architecture Technology*, IBM Server Group, IBM Corporation, pp.17-18, 2001

日本アイ・ビー・エム株式会社
副主任ITスペシャリスト
谷本 雄一郎 Yuichiro Tanimoto

［プロフィール］
1997年，日本アイ・ビー・エム入社．長野オリンピック・プロジェクトに派遣され，大会を支えるシステムの技術支援を行う．その後，クライアントPC，IAサーバーの技術支援に従事する中で，一貫して，大規模システムの構築支援やパフォーマンス分析に携わっている．