SPARC M10 の Software on Chip を活用した Oracle Database 12*c* In-Memory のクエリー高速化

2016 年 6 月

第 2.0 版

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

目次

1. はじめに.....3

2. Oracle Database In-Memory.....4

3. SPARC M10.....6

4. Oracle Database In-Memory および Software on Chip.....8

5. 検証.....14

6. 総括.....20

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

はじめに

企業を取り巻く環境が急速に変化している今日において、迅速かつ効率的な経営判断が求められています。特に、ビジネスの中核を担うデータベースにおいては、多種多様なデータの高速処理に対する要求が高まっています。

Oracle Database 12*c* が提供する機能「Oracle Database In-Memory」は、あらゆるデータベース処理を高速化します。また、SPARC M10 のプロセッサ SPARC64 X/X+が提供する演算機能「Software on Chip」は、高い検索処理性能や高速メモリ操作を実現します。

本資料では、Oracle Database 12*c* を SPARC M10 上で運用し、それぞれの機能 Oracle Database In-Memory と Software on Chip を利用することで、高性能なデータベースが実現することを示します。

本資料は、データベース管理者、データベースの性能向上に従事している方、そしてデータベース導入を検討している方を対象にしています。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Database In-Memory

Oracle Database 12*c* は従来のロー型に加えて、「Oracle Database In-Memory」と呼ばれる、全く新しいインメモリ・カラム型をオプション機能として提供しています。Oracle Database In-Memory は最新トランザクションデータのリアルタイム処理を可能にする分析処理を高速にするだけでなく、OLTP 処理性能も向上させます。

Oracle Database In-Memory はリスクフリーかつ簡単に運用できます。Oracle Database 上で動く全てのアプリケーションで、Oracle Database In-Memory 機能を有効にし、高速データ処理を実現することができます。導入には複雑なデータ移行やアプリケーションの書き換えなどの必要がありません。また、Oracle Database がサポートしているいずれのハードウェア・プラットホームでも動作します。Oracle Database 12*c* が動作している環境で Oracle Database In-Memory 機能を有効にするだけで、既存のアプリケーションをそのまま使うことができます。

Oracle Database In-Memory は、OLTP 処理に適したロー型と、集計・分析処理を得意とするカラム型の 2 つのフォーマットを同時にメモリ上に保持するため、両方の処理を高速化します。OLTP 処理は従来通り、SGA（システムグローバル領域）上のデータベース・バッファ・キャッシュを用い、DWH 処理は SGA 上に新たに用意されたインメモリ列ストアという場所を利用します。

インメモリ列ストアでは、列方向にデータを扱います。そのため、カラム型データをメモリ上に展開することが可能になります。これにより、クエリーが要求する列のみにアクセスし、複数の列の値を同時に処理することで、分析処理が高速になります。また、Oracle Database では列方向にデータ型が同一なので効率よくデータ圧縮され、クエリー処理の性能向上が期待されます。

インメモリ列ストアという新しいメモリ領域は、OLTP 処理の性能改善も可能にします。従来の DWH 処理では索引が多く作成されるため、表のデータ更新の際に索引を書き換えることが OLTP 処理のオーバーヘッドを引き起こしていました。Oracle Database In-Memory では、DWH 処理はインメモリ列ストアを利用するので、従来のデータベースで検索用に利用されていた索引はほとんど不要になります。

さらに、他のインメモリはロー型またはカラム型を選択してから実行するのに対し、Oracle Database In-Memory はアプリケーションが両方のフォーマットにアクセスすることが可能です。アプリケーションからの要求に応じてオプティマイザが自動判別し、集計・分析処理はカラム型に、OLTP 処理はロー型に対し実行します。つまり、アプリケーションはロー型・カラム型を意識することなく、要求に適した処理を高速に実行することが可能です。

加えて、ロー型とカラム型はリアルタイムに同期するので、データベース・バッファ・キャッシュとインメモリ列ストアのデータは常に一貫性が保たれます。例えば、OLTP 処理を実行しながら、検索系処理によって更新された最新情報にアクセスすることが可能です。

また、Oracle Database In-Memory は、DML 向け、クエリー向けなど、用途に応じて異なる 6 種類の圧縮レベルを提供しています。これらを使い分けることで、限られたメモリ領域を効率的に利用することができます。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Database In-Memory に関する詳しい情報は、下記 URL をご覧ください:
https://www.oracle.com/database/database-in-memory/index.html

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

SPARC M10

1.1 製品概要

SPARC M10 は、最先端テクノロジーによる世界 No.1 の拡張性と柔軟性、メインフレームおよび SPARC Enterprise M シリーズから継承する高い信頼性と高水準の技術力を備えた、データベースサーバ等の基幹システムに最適な高性能・高信頼 UNIX サーバです。

1U の省スペースに高い性能、信頼性、仮想化機能を備え、あらゆる用途に最適なエントリーサーバである SPARC M10-1、データセンターの統合に最適なミッドレンジサーバの SPARC M10-4、そして、「Building Block」方式の筐体間接続により、高いスケーラビリティを発揮するハイエンドサーバである SPARC M10-4S の 3 モデルをラインナップしています（図 1）。

全てのモデルが「Software on Chip」に対応し、大量データの高速処理を実現します。また「CPU コア アクティベーション」により、サーバの稼働状況に応じた柔軟な CPU リソースの追加が可能です。さらに SPARC M10-4S は、「Building Block 方式」による筐体間接続を採用しており、ビジネスの成長に合わせて「Dynamic Reconfiguration（物理パーティションの動的再構成）」による筐体追加が可能です。

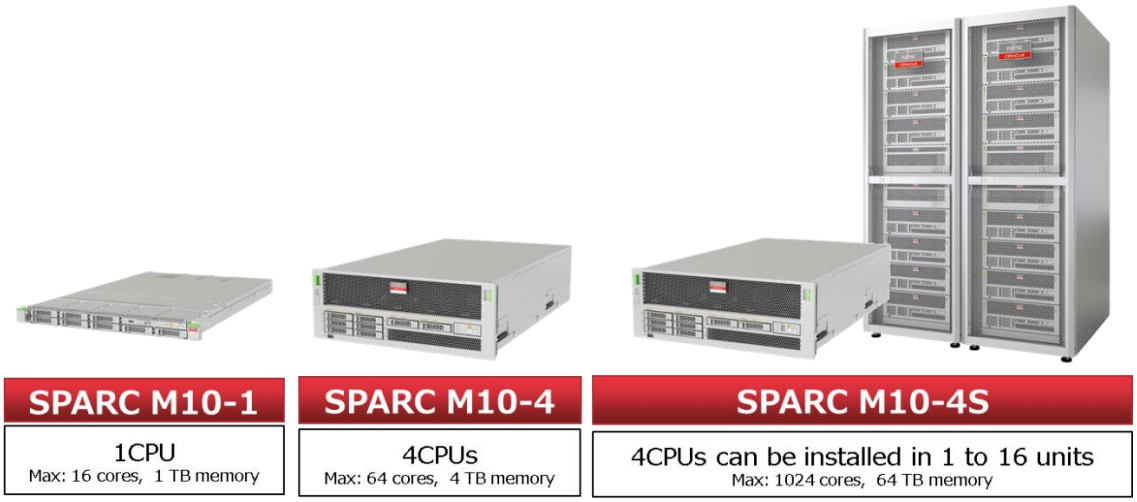

図 1. SPARC M10 ラインナップ

1.2 Software on Chip

「Software on Chip」とは富士通独自の命令セット、マイクロアーキテクチャを用いて特定のソフトウェア機能を強化したもので、これには富士通がこれまで培ってきたスーパーコンピュータ技術、ビジネスコンピューティング技術およびメインフレーム技術が活かされています。この Software on Chip により、SPARC M10 では、オペレーティングシステム、Oracle Database をはじめ、様々なアプリケーションの高速化を実現しています。

Software on Chip は、SPARC64 X/X+で拡張された、「拡張レジスタ」「SIMD（Single Instruction Multiple Data)」、「十進浮動小数点演算」、「暗号演算」の 4 つの機能の組み合わせで実現されています。Oracle Database では、以下の Software on Chip の機能でパフォーマンスを飛躍的に向上させています*1（表 1）。

*1　表 1 に示した Software on Chip の機能が全て有効になるのは、Oracle Database 12.1.0.2 以降です。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

表 1. Software on Chip の各機能と動作環境

Software on Chip 機能	Oracle Database の パフォーマンス向上	動作環境
In-Memory 検索処理	In-Memory 検索処理	Oracle Database 12*c* Enterprise Edition Oracle Database In-Memory Oracle Solaris 11.1 以降
In-Memory 展開処理	MEMCOMPRESS FOR CAPACITY LOW による展開処理(OZIP)	
十進浮動小数点演算	Oracle NUMBER の演算	Oracle Database 12*c* 以降 Oracle Solaris 11.1 以降
暗号演算	TDE 表領域暗号化 通信暗号化	Oracle Database 12*c* Enterprise Edition Oracle Advanced Security Oracle Solaris 11.1 以降
HASH	HASH Join	Oracle Database 12*c*
Copy/Compare	メモリ処理	Oracle Solaris 11.1 以降

本資料では、これら Software on Chip 機能のうち In-Memory の検索/展開処理について言及します。

1.3 高速・大容量メモリ搭載

Oracle Database In-Memory でその効果を最大限に発揮するために、SPARC M10 は大量のデータをメモリに載せるための大容量メモリと、大量データを高速に処理するための高速メモリを提供しています。SPARC M10 は、メモリを 1CPU あたり最大 1TB、システムあたり最大 64TB 搭載することが可能です。大量のデータをメモリ上に展開するために、インメモリ処理には大容量のメモリを搭載することが理想的です。(図 2)。

一般に、メモリ容量が増大すると、メモリアクセスの遅延が発生すると言われています。しかし SPARC M10 は、プロセッサ - メモリをダイレクトに接続することで、高速データ転送を可能にしました。さらに、プロセッサ - メモリ間のバスを多くすることで、メモリ帯域幅を増やしました。このように SPARC M10 は、低レイテンシおよび高帯域のメモリアクセスを提供します。

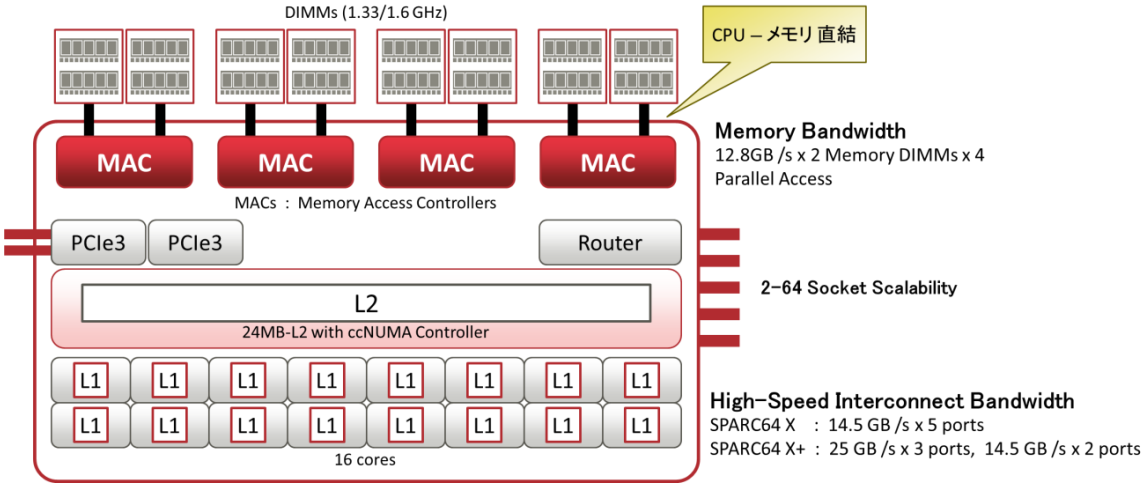

図 2. SPARC M10 大容量/高速メモリアクセス

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Database In-Memory および Software on Chip

本章では、Oracle Database In-Memory と SPARC M10 の組み合わせにより、インメモリデータベース処理がより高速に実行される仕組みについて解説します。Oracle Database In-Memory が SIMD ベクタ演算処理に適したフォーマットでデータをメモリに展開するので、SPARC M10 の高速ベクタ演算は検索処理を飛躍的に向上させます。

1.4 ロー型フォーマットとカラム型フォーマット

Oracle Database In-Memory はデュアルフォーマットのインメモリアーキテクチャを提供しており、データをロー型、カラム型の両方のフォーマットで同時に格納します。(図 3)。

ロー型フォーマットは、Oracle Database で伝統的に用いられてきたフォーマットで、トランザクションやレコードは 1 つの行として表現されます。ロー型フォーマットは、レコードに含まれるすべての属性値をひとかたまりとしてメモリ上、または、ストレージ上に格納するため、すべての属性値にすばやくアクセスすることが出来ます。このため、ロー型フォーマットはトランザクション処理に適しています。

一方で、新しいカラムフォーマットはデータを 90 度回転させ、それぞれの属性値は分離されたカラム構造で格納されます。このカラム型フォーマットは少数の列、多量の行にアクセスする検索処理に優れるので、分析に適しています。

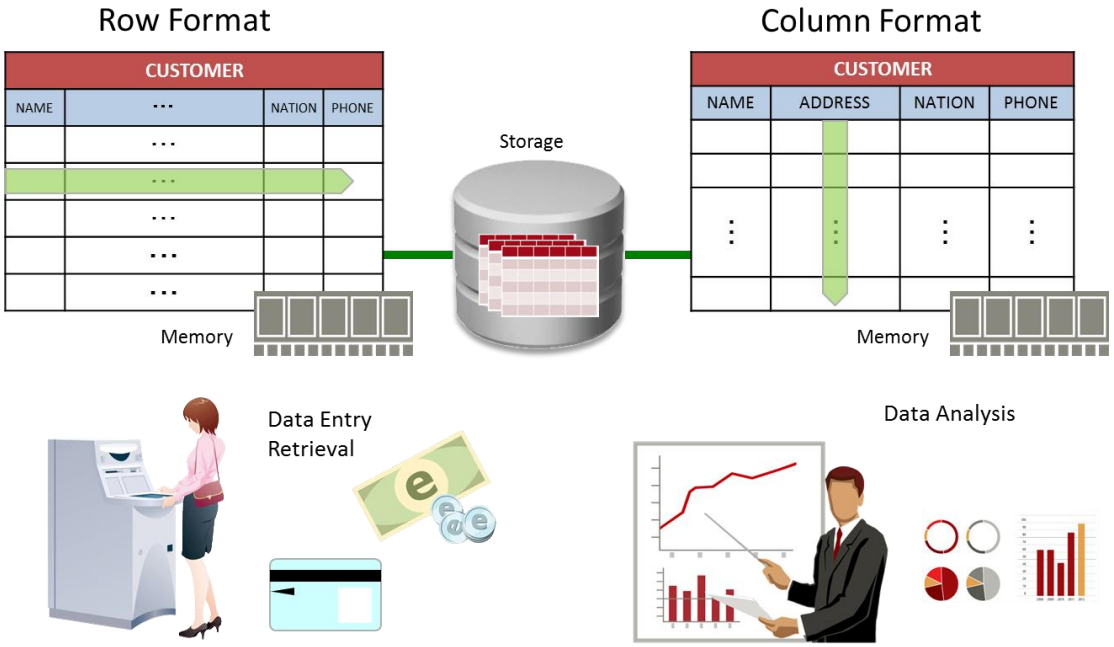

図 3．ロー型フォーマットとカラム型フォーマット

1.5 カラム型フォーマットの検索処理高速化

1.5.1 辞書化による最適化

Oracle Database In-Memory のカラム型フォーマットは、メモリと CPU 性能を最大限

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

に生かすために最適化されています。ディスク上に格納されている各行のデータを辞書化により短い固定長のデータに変換し、メモリ上の連続した領域に格納します（図 4）

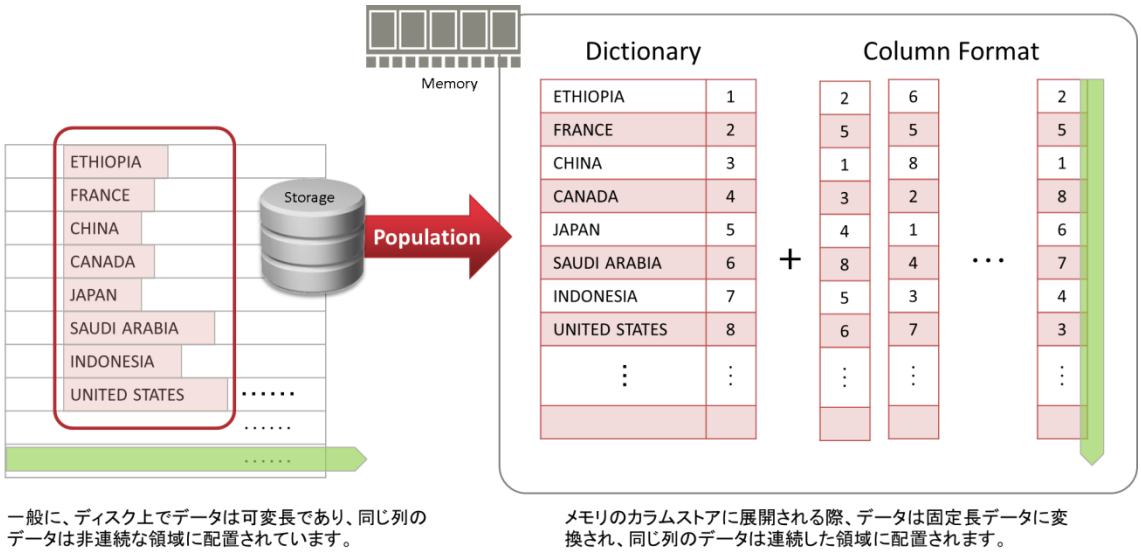

図 4. 辞書化による最適化

これにより、大量のデータをメモリ上に載せることができるだけでなく、列データを検索する際には、多くのエントリを効率よくレジスタにロードすることができるため、プロセッサの SIMD 機能と組合わせることで効率よくかつ高速に演算することが可能になっています。

1.5.2 SIMD による検索処理の高速化

以下に SIMD を使用した検索例(NATION='JAPAN')を示します。まず、JAPAN を符号化した値(この場合 =5)を調べるために辞書を参照します。そして、符号化された値 "5" に対し検索処理を実行します。

次にメモリ上に格納されているデータをレジスタにローディングします。SIMD を利用しない検索処理では、１つのレジスタに１つのデータのみローディング可能です。このため、一度に１つの比較しか実行できません。一方、SIMD を用いたときには、メモリ上は固定長のデータとしてメモリ上に連続に格納されているため、1 つのレジスタに 8 つのデータを同時にローディングすることができます。このレジスタと比較対象の 5 を格納したレジスタを比較することで同時に 8 つの比較を行うことができます。このように辞書化をすることにより、複数のデータを一つのレジスタに入れ、SIMD により同時に比較することができるため、高速に検索を行うことができます（図 5）。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

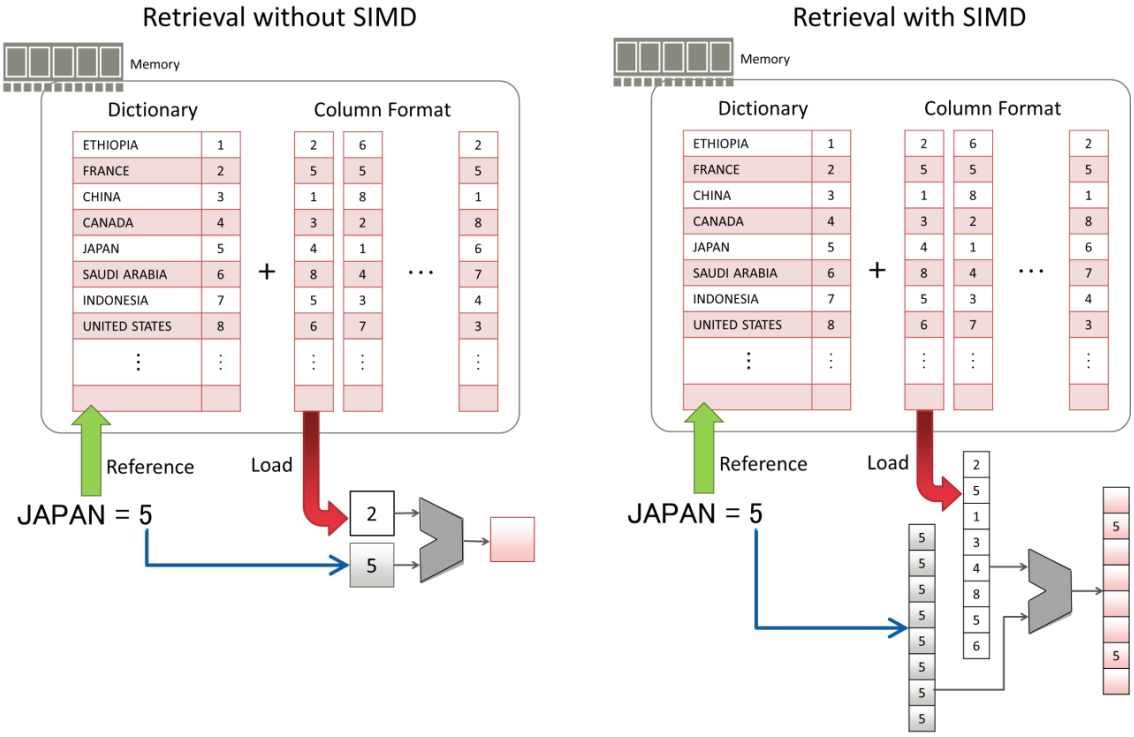

図 5. SIMD による検索処理の高速化

1.5.3 SPARC64 X/X+ による高速検索

SPARC M10 に採用されている SPARC64 X/X+ プロセッサは高度な SIMD ベクタ処理能力を有しており、より高い検索性能を実現します。Software on Chip 機能の一つである In-Memory 検索処理は SIMD ベクタ処理を使っています。

SPARC M10 の先行機種「SPARC Enterprise」では、1 命令で 1 個のデータを処理し、それを 1 サイクルに 2 命令実行できました。一方、SPARC64 X の SIMD 命令は、最大 8 個のデータを処理し、プロセッサは 1 サイクルあたり 1 命令の実行が可能です。 SPARC64 X+では SIMD がさらに強化され、1 バイトデータを 1 命令あたり最大 16 個処理することができます。また、そのような命令を同時に 2 命令実行することが可能です。したがって、1 サイクルに最大 32 個のデータを処理することができます。Oracle Database 12.1.0.2 以降にパッチ 20342458[*2] を適用すると、SPARC64 X+の演算器 4 つ全てがサポートされます。SPARC64 X+の各演算器は最大で 8 個のデータを処理することが可能で、4 つ全ての演算器を有効にすれば、最大 32 個のデータに SIMD 比較命令を並列に実行できるようになります(図 6)。パッチ 20342458 を適用できるのは、Oracle Solaris 11.2 以降です。

また、演算能力の向上にあわせて、主記憶からキャッシュへのリードバンド幅、キャッシュからレジスタへの転送能力、および、ベクタレジスタの個数も大幅に増強されており、強化された演算能力がインメモリ・データベース上で最大限発揮されるように設計されています。

---

*2 Database Proactive Bundle Patch 12.1.0.2.7 またはそれ以降を使用している環境であれば、パッチ 20342458 を個別に適用する必要はありません。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

なお、辞書のビット幅が 8bit のとき、1 コアあたりの検索性能は SPARC64 X+で毎秒 300 億行、SPARC64 X で毎秒 140 億行です。

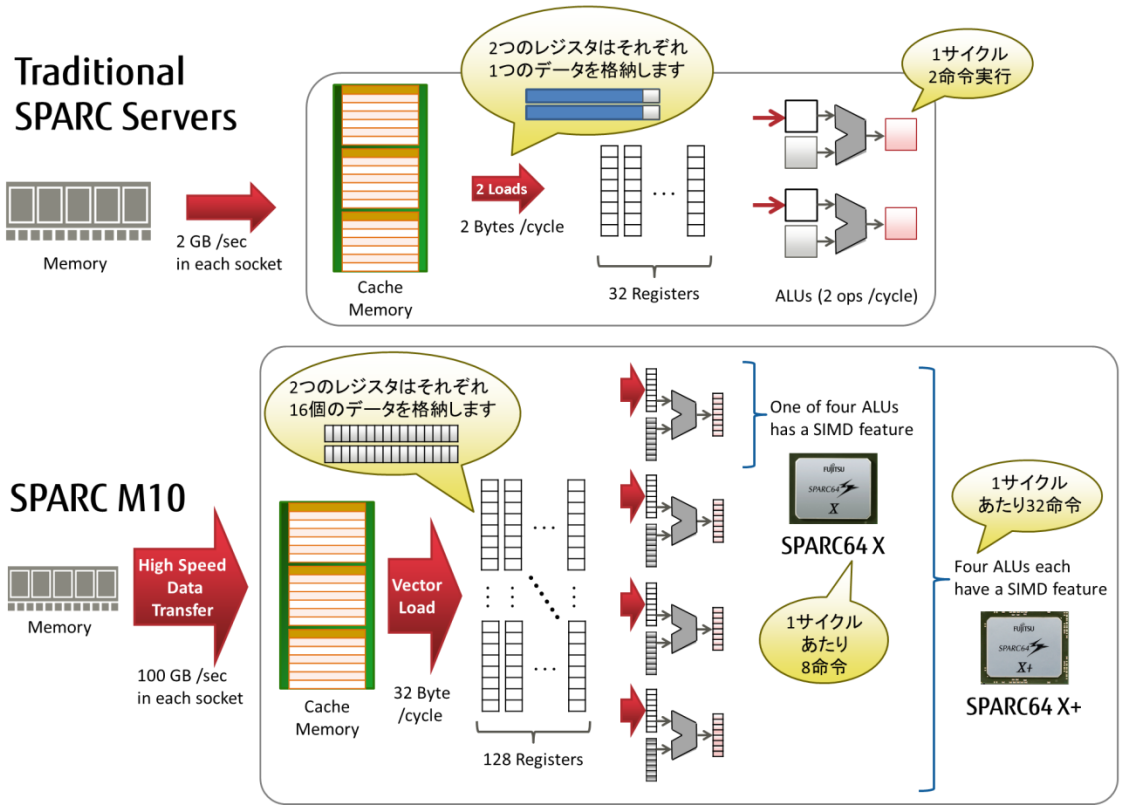

図 6 . SPARC M10 と SPARC Enterprise の SIMD ベクタ演算処理.

1.6 インメモリ・ストレージ・インデックス

カラム型フォーマットにおいて、各列の情報は複数の「IMCU（In-Memory Compression Unit）」に分割されます。このようにデータを分割することで、複数プロセスによる処理を効率よく行うことが可能となっています。

また、各 IMCU に付与されるインメモリ・ストレージ・インデックスによって、データの処理量を大幅に削減することができます。

「インメモリ・ストレージ・インデックス」は、「インメモリ列ストア」に対して自動的に作成・維持され、データ処理量を削減する効果をもっています。インメモリ・ストレージ・インデックスは、各 IMCU に含まれる各列について最大値と最小値を保持しており、SQL 文におけるフィルタ述語にもとづく処理データの削減を可能にします（図 7）。

クエリーにおいて WHERE 節の述語が指定された場合には、まず指定された値が対応する IMCU に含まれるかを判定するために、インメモリ・ストレージ・インデックスが検索されます。もし、指定された値が含まれないことがわかれば、その IMCU に対する検索を省くことができます。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

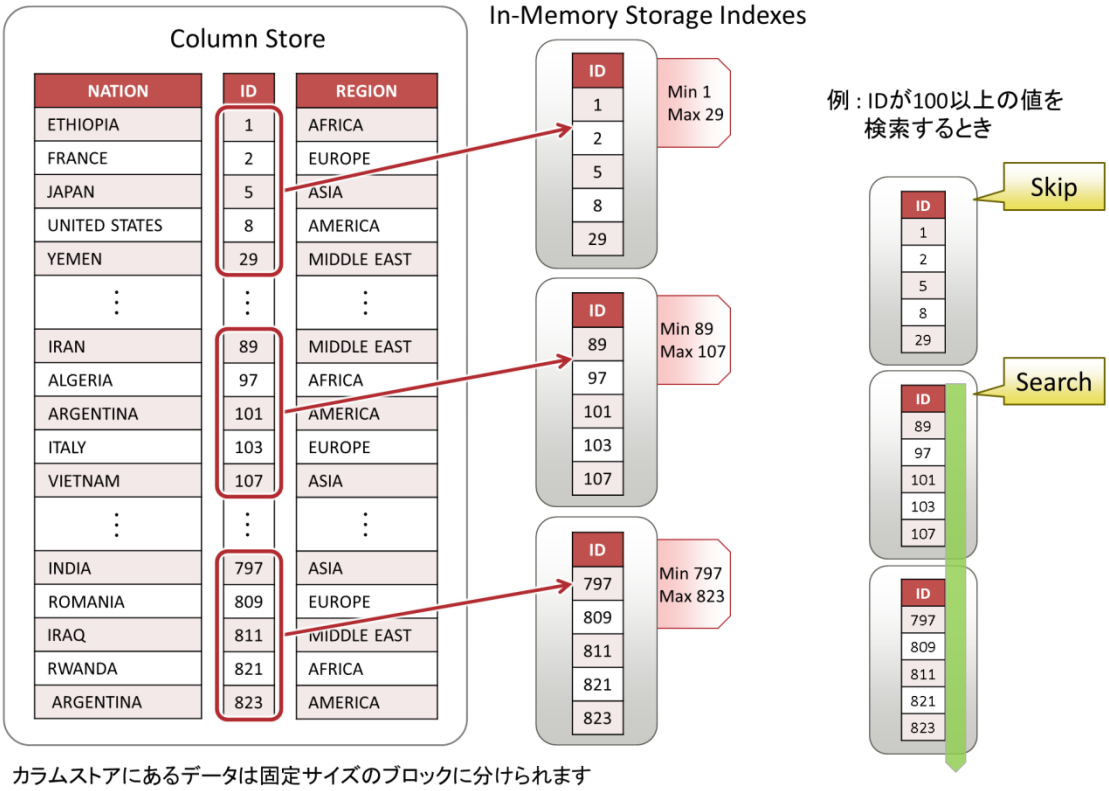

図 7. インメモリ・ストレージ・インデックスを利用した検索処理

1.7 In-Memory 圧縮

初期設定では、データは圧縮されて In-Memory Column Store に格納されます。Oracle Database In-Memory ではメモリを効率よく使用するために、6 段階のデータ圧縮レベルを用意しており、業務に合わせてパフォーマンスとデータ容量のバランスが取れるようになっています。

- NO MEMCOMPRESS

  圧縮を一切せずに、データのポピュレーションを行います。

- MEMCOMPRESS FOR DML

  最小限の圧縮です。DML 操作に優れています。

- MEMCOMPRESS FOR QUERY LOW

  デフォルトの圧縮です。クエリー処理に優れています。先述した辞書化の他に、ランレングス符号化やビットパッキングを使用します。

- MEMCOMPRESS FOR QUERY HIGH

  省スペースを実現し、クエリー処理を高速に行います。

- MEMCOMPRESS FOR CAPACITY LOW

  OZIP と呼ばれる Oracle Database に特化した圧縮技術を用いて、より高い圧縮で省スペースを実現し、高速処理を行います。

- MEMCOMPRESS FOR CAPACITY HIGH

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

省スペースに最も適した圧縮です。gzip や Deflate に似た圧縮技術を利用しています。

1.8 In-Memory と Software on Chip SIMD の組み合わせ

これまで説明してきたように Oracle Database In-Memory のインメモリ・ストレージ・インデックスによる性能向上と、Software on Chip の SIMD ベクタ処理による検索処理を組み合わせることで、従来のデータベース処理の 100 倍を超える高速処理が期待できます。

6 種類の圧縮のうち、MEMCOMPRESS FOR CAPACITY LOW に使われている OZIP は、SPARC M10 の SIMD ベクタ演算による Software on Chip 機能の一つである、In-Memory 圧縮処理を用いて、高速なデータ圧縮を実現します。なお、展開後の検索処理には、全ての圧縮レベルで Software on Chip が適用されます（表 2）。

表 2　圧縮レベルと Software on Chip の対応

圧縮レベル	Software on Chip SIMD		
	圧縮	展開	検索
NO MEMCOMPRESS	—	—	●
FOR DML	—	—	●
FOR QUERY LOW	—	—	●
FOR QUERY HIGH	—	—	●
FOR CAPACITY LOW	—	●	●
FOR CAPACITY HIGH	—	—	●

●：オン　—：オフ

本資料ではこれ以降、In-Memory 検索処理および圧縮処理を「Software on Chip SIMD」と表記します。

次章では、ベンチマークを用いた検証により、SPARC M10 がデータベースの高速処理に適しているということを説明します。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

検証

前章までに述べたように、SPARC M10 は Oracle Database In-Memory と Software on Chip を組み合わせることにより、データベース処理の高速化を実現します。本章では、検証によって SPARC M10 が高速データベースに最適なサーバであることを示します。

1.9 検証環境

本検証では、データベースサーバとして SPARC M10-4S、Oracle Solaris11.1 および 11.2、Oracle Database 12*c* Enterprise Edition を使用しています。また、ストレージとして Flash Accelerator F80 カードを用いています（図 8）。

図 8. 検証システムの構成

1.9.1 使用ハードウェアおよびソフトウェア

Database Server

SPARC M10-4S(1unit)

Processor	3.7GHz SPARC64 X+, 4 sockets (16 cores per socket)
Memory	1TB
Operating System	Oracle Solaris 11.1.16.5.0 および 11.2.15.5.1

Storage System

Sun Flash Accelerator F80 PCIe Card (4 台)

Oracle Databese

Oracle Database 12*c* R1 (12.1.0.2)

OPatch version　12.1.0.1.10

1.9.2 ストレージ設定

4 台の Sun Flash Accelerator F80 PCIe Card を 1 つの ZFS ストレージプールとして作成し、その上に ZFS ファイルシステムを作成しデータベースファイル、REDO ログを配置します。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

1.9.3 検証ツール

本検証では、ベンチマークとして「SSB (Star Schema Benchmark)」を使用します。SSB は、TPC-H を元に開発されたデータウェアハウス性能を測定するためのベンチ―マークで、多くの商用データベース製品の性能評価に用いられています。

テーブル

SSB のテーブル構成は、データウェアハウスにおける代表的なデータモデルである、スタースキーマ型の構成になっています（図 9)。SSB の詳細については以下 URL 等を参照してください:

http://www.cs.umb.edu/~poneil/StarSchemaB.PDF

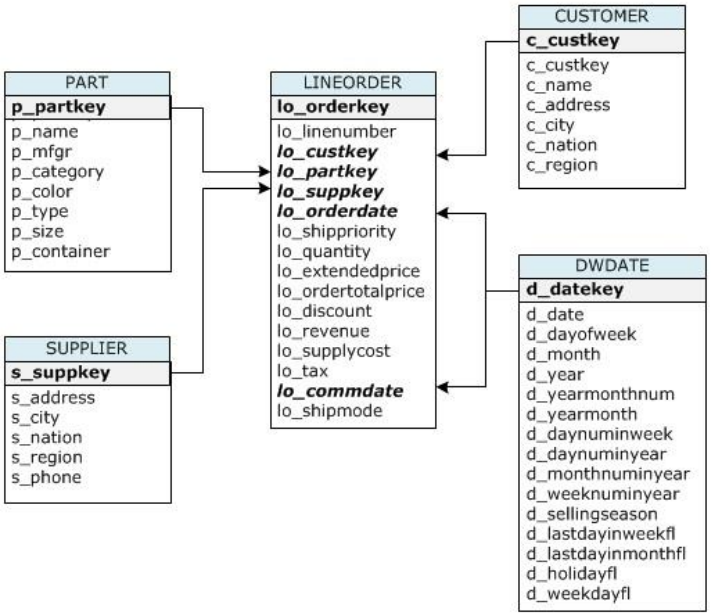

図 9. SSB のテーブル構成

今回の検証では Oracle Database In-Memory の効果を検証するためにテーブルのサイズがメモリサイズとほぼ等しくなるよう、スケールファクタを 1,400 に設定します。このときのテーブルのレコード数、サイズは以下になります。

TABLE_NAME	REC_COUNT	TABLESIZE(MB)
LINEORDER	8399997494	951955
PART	2200000	224
SUPPLIER	2800000	332
CUSTOMER	42000000	5171
DATE_DIM	2556	0

クエリー

インメモリ列ストアにデータを格納した場合、以下の操作を行うとパフォーマンスが大幅に向上します。

- 大量の行をスキャンし、演算子を使用するフィルタ述語を適用する問合せ
- 100 列ある表から 5 列を選択する問合せなど、列が大量にある表またはマテリアライズド・ビューから少量の列を選択する問合せ

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

- 小規模表を大規模表に結合する問合せ
- データを集計する問合せ

なお測定プログラムには、上記条件を含む SSB ベンチマークのクエリーQ1 をカスタマイズした以下の独自のクエリーを使用します。

```
select count(lo_discount)
 from lineorder, date_dim
   where lo_orderdate = d_datekey
   and d_year = 1994
   and d_weeknuminyear = 7
   and d_daynuminweek = 1
   and lo_discount between 2 and 8
   and lo_tax between 2 and 6
   and lo_quantity between 10 and 45;
```

なおクエリーでオプティマイザが自動で選択した実行計画は以下の通りです。

```
-----------------------------------------------------
| Id  | Operation                     | Name      |
-----------------------------------------------------
|   0 | SELECT STATEMENT              |           |
|   1 |  SORT AGGREGATE               |           |
|*  2 |   HASH JOIN                   |           |
|   3 |    JOIN FILTER CREATE         | :BF0000   |
|*  4 |     TABLE ACCESS INMEMORY FULL| DATE_DIM  |
|   5 |    JOIN FILTER USE            | :BF0000   |
|*  6 |     TABLE ACCESS INMEMORY FULL| LINEORDER |
-----------------------------------------------------
```

2 つのテーブル LINEORDER と DATE_DIM をインメモリ列ストアをフルスキャンし HASH 結合をするプランになります。

1.9.4 データベース設定

検証に際し SSB で使用する全てのテーブルをインメモリ列ストアに格納するために SGA およびインメモリ列ストアのサイズを変更します。なお今回は、検証のためにインメモリ列ストアのサイズを大きく設定していますが、実際の運用環境ではユーザプロセスやソート用のメモリ領域などを考慮する必要があります。In-Memory はデフォルトで無効になっています。これを有効にするには、パラメータ INMEMORY_SIZE を下記の様に設定する必要があります。

```
SQL> ALTER SYSTEM SET SGA_TARGET=900G SCOPE=SPFILE;
SQL> ALTER SYSTEM SET INMEMORY_SIZE=800G SCOPE=SPFILE;
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
```

1.10 検証内容

次の 2 つの検証行い Software on Chip の効果を検証します。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

1) パッチ 20342458 適用時の In-Memory および Software on Chip SIMD の効果
2) パッチ 20342458 適用時の圧縮モードにおける Software on Chip SIMD の効果

1.10.1 In-Memory および Software on Chip SIMD の効果の検証

パッチ 20342458 を適用した時の In-Memory および Software on Chip SIMD の効果を測定するために以下の 4 つのモードの比較を行います(表 3)。パッチ 20342458 を適用し、In-Memory および Software on Chip SIMD を有効にしたモードは、Oracle Solaris 11.2 で走行します。

表 3 In-Memory と Software on Chip SIMD の設定

モード名	In-Memory	Software on Chip SIMD	パッチ 20342458	Oracle Solaris バージョン
通常モード	—	—	—	11.1
In-Memory	●	—*3	—	11.1
In-Memory + Software on Chip SIMD	●	●	—	11.1
パッチ適用+ In-Memory + Software on Chip SIMD	●	●	●	11.2

●：オン —：オフ

1.10.2 FOR CAPACITY LOW における Software on Chip SIMD の効果の検証

Oracle Database In-memory には 6 種類の圧縮モードが用意されており、このうち FOR CAPACITY LOW オプションは Oracle データ型に特化した圧縮 OZIP を適用しています。SPARC M10 には、FOR CAPACITY LOW の展開処理に Software on Chip SIMD が使用されています。パッチ 20342458 適用時の Software on Chip SIMD の効果を検証するために、下の 4 つのモードで測定します。パッチ 20342458 を適用し、OZIP および Software on Chip SIMD を有効にしたモードは、Oracle Solaris 11.2 で走行します。

表 4 OZIP と Software on Chip SIMD の設定

モード名	In-Memory	OZIP	Software on Chip SIMD	パッチ Patch 20342458	Oracle Solaris バージョン
通常モード	●	—	—	—	11.1
OZIP	●	●	—*3	—	11.1
OZIP + Software on Chip SIMD	●	●	●	—	11.1
パッチ適用+ OZIP + Software on Chip SIMD	●	●	●	●	11.2

●：オン —：オフ

1.11 検証結果

1.11.1 In-Memory および Software on Chip SIMD の効果

In-Memory を有効にすると、辞書化やインメモリ・ストレージ・インデックスにより検索処理性能が向上します。さらに Software on Chip SIMD を使用すると、SIMD ベクタ演算により、さらなる高速処理が可能になります。

*3 検証に限り、特別な方法を用いて Software on Chip SIMD を無効にしています。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

検証の結果、処理時間は In-Memory を使用しない場合に比べ、In-Memory 機能を使用すると 16～18 倍、In-Memory + Software on Chip SIMD を使用すると 49～190 倍、さらにパッチ適用と In-Memory + Software on Chip SIMD を組み合わせると 73～278 倍に向上しました（図 10、表 5）。

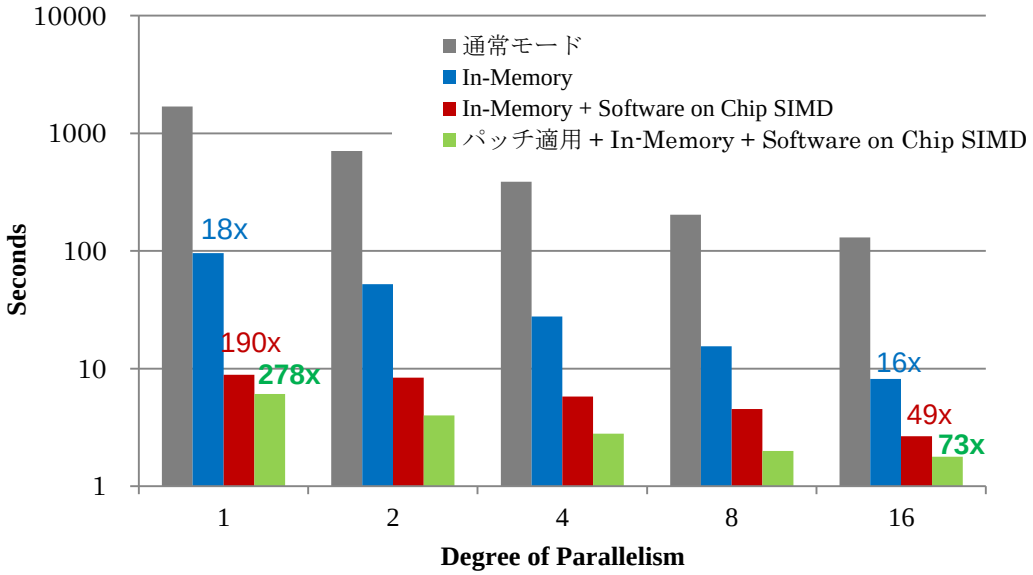

図 10　In-Memory および Software on Chip SIMD の効果

表 5　各モードにおける処理時間（同時並列度=1）

モード名	処理時間［秒］	通常モードとの比較［倍］
通常モード	1691	1
In-Memory	96	18
In-Memory+Software on Chip SIMD	9	190
パッチ適用+ In-Memory + Software on Chip SIMD	6	278

本結果により、Oracle Database 12*c* の Oracle Database In-Memory と、SPARC M10 の Software on Chip の組み合わせが、高速検索処理を実現するデータベースとして有効であることを実証できました。処理をさらに高速化するために、パッチ 20342458 の適用が効果的です。

1.11.2　FOR CAPACITY LOW における Software on Chip SIMD の効果

圧縮モード FOR CAPACITY LOW において、Software on Chip SIMD は検索処理だけでなく、圧縮されたデータの展開処理を高速にします。

本検証では、圧縮モード FOR CAPACITY LOW における処理性能は、OZIP を有効にすると約 1.2～1.6 倍、OZIP と Software on Chip SIMD を両方有効にすると約 2.0～3.5 倍、さらにパッチ 20342458 を適用し、OZIP と Software on Chip SIMD を両方有効にすると約 3.4～3.7 倍に向上しました（図 11、表 6）。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

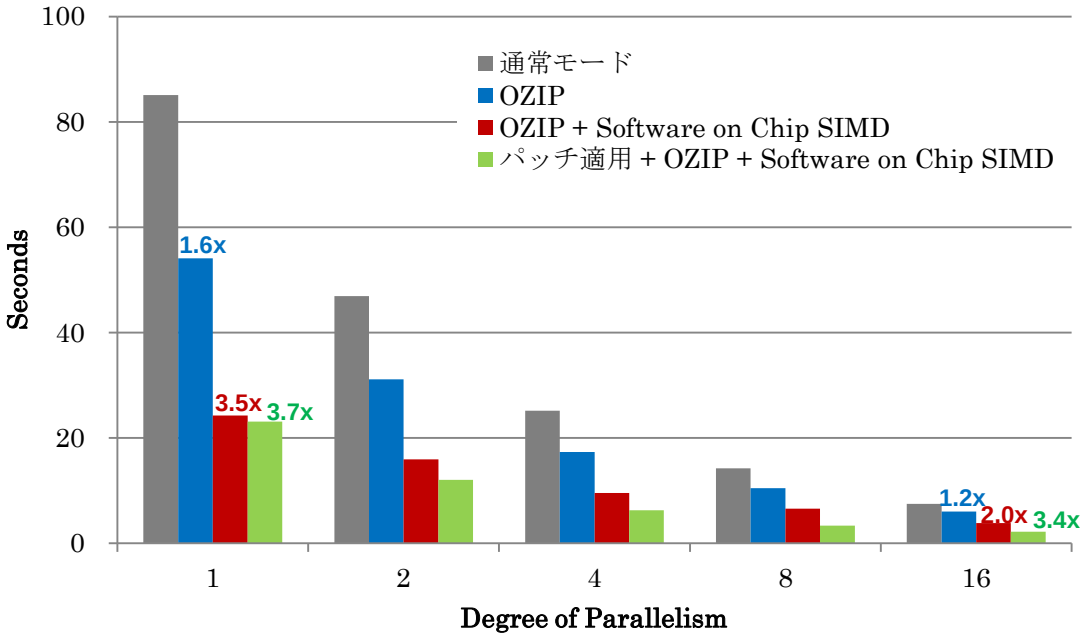

図 11　FOR CAPACITY LOW における OZIP および Software on Chip SIMD の効果

表 6　各モードにおける処理時間（同時並列度=1）

モード名	処理時間［秒］	通常モードとの比較［倍］
通常モード	85	1
OZIP	54	1.6
OZIP + Software on Chip SIMD	24	3.5
パッチ適用+ OZIP + Software on Chip SIMD	23	3.7

本検証により、圧縮モード FOR CAPACITY LOW において、Software on Chip SIMD が OZIP で圧縮されたデータの展開処理に機能することで、性能向上を実現することが確認されました。さらに、パッチ 20342458 を使用することで全体性能が向上します。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

総括

本検証より、Oracle Database 12*c*の Oracle Database In-Memory によるインメモリ・カラム処理と、SPARC M10 の Software on Chip SIMD による高速ベクタ演算処理を組み合わせることで、高速処理性能が得られることが分かりました。

最初の検証では、Oracle Database In-Memory と Software on Chip SIMD を両方有効にし、パッチ 20342458 を適用することで最大 278 倍の性能を得られることが確認できました。

次に、圧縮モード FOR CAPACITY LOW における Software on Chip SIMD の効果を調べる検証では、OZIP と Software on Chip SIMD を両方とも有効にし、パッチ 20342458 を適用することで、全体性能が最大 3.7 倍向上することが分かりました。

Oracle Database In-Memory 機能と SPARC M10 の Software on Chip SIMD を使用することで、圧倒的な性能向上が実現します。Oracle Database 12*c* と SPARC M10 で構築したデータベースサーバは、これら二つの技術を組み合わせて大量データ検索の処理時間を大幅に短縮することで、お客様のビジネスをさらに加速し、より効率的な投資を促進します。

Oracle Database In-Memory と SPARC M10 の Software on Chip SIMD の利用は、単なるデータベースサーバ統合の基盤を提供するだけでなく、処理性能、効率性、柔軟性、投資費用抑制の点で、お客様のビジネスに変革をもたらす優れたサーバを実現します。

SPARC64 X+が提供する全ての Software on Chip 機能が有効になるのは、Oracle Database 12.1.0.2 以降です。SPARC64 X+を搭載した SPARC M10 は、SPARC64 X の Software on Chip 機能を全てサポートします。4.2.3 章で示した SPARC64 X+の SIMD 比較演算の複数演算器による並列実行は、パッチ 20342458 を適用した Oracle Database 12.1.0.2 以降でサポートされます。このパッチを適用することにより Oracle Database は、データベースの性能を飛躍的に向上させます。パッチ 20342458 の適用により、SPARC64 X＋を搭載した SPARC M10 は、SPARC64 X+の Software on Chip 機能の性能を最大限に活用することができます。

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

改版履歴

改版年月	版数	改版内容
2015 年 10 月	1.0	初版
2016 年 6 月	2.0	SPARC64 X+の Software on Chip 機能を有効にするパッチを使用した検証を追加

富士通株式会社

〒105-7123
東京都港区東新橋1-5-2
汐留シティセンター

本書に記載されている内容は改善のため、予告なく変更することがあります。
富士通株式会社は、本書の内容に関して、いかなる保証もいたしません。
また、本書の内容に関連した、いかなる損害についてもその責任を負いません。
UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。
SPARC64、SPARC64 ロゴ、およびすべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
その他各種製品名は、各社の製品名称、商標または登録商標です。
本書に記載されているシステム名、製品名等には、必ずしも商標表示（(R)、TM）を付記していません。

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.0115

Copyright 2016 FUJITSU LIMITED
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.