フラッシュメモリ構成のストレージ環境における商用アウトオブオーダ型データベースエンジンの性能にプロセッサ省電力モードが与える影響の評価

出射 英臣† 久木 和也† 藤原 真二‡ 茂木 和彦‡ 合田 和生¶ 喜連川 優¶§

†㈱日立製作所 横浜研究所 〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地
‡㈱日立製作所 情報・通信システム社 IT プラットフォーム事業本部
〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地
¶東京大学生産技術研究所 〒153-8503 東京都目黒区駒場 4-6-1
§国立情報学研究所〒101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †{hideomi.idei.ub, kazuya.hisaki.bx}@hitachi.com,
‡{shinji.fujiwara.yc, kazuhiko.mogi.uv}@hitachi.com, ¶{kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし 近年, 企業や社会活動で発生するデータが増加しており, ビッグデータ利活用への期待が高まっている. このような中，我々は内閣府最先端研究開発支援プログラムにおいて，アウトオブオーダ型データベースエンジン(OoODE)と称する実行原理に基づく超高速データベースエンジンの研究開発を推進している．近年注目を集めているフラッシュメモリ構成のストレージ環境では，I/O のレスポンス時間が従来 HDD と比較して２桁以上短いため，プロセッサの省電力モードが OoODE の性能に影響を与えることが考えられる．そこで，今回本研究成果を基に開発した商用 OoODE を用い，フラッシュメモリ構成のストレージ環境においてプロセッサの省電力モードが性能に与える影響について評価した．本稿では，その評価に関して報告する．

キーワード OoODE，アウトオブオーダ型，データベースエンジン，フラッシュメモリ，省電力モード

Hideomi IDEI† Kazuya HISAKI† Shinji FUJIWARA‡ Kazuhiko MOGI‡
Kazuo GODA¶ and Masaru KITSUREGAWA¶

†Yokohama Research Laboratory, Hitachi, Ltd. 292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817 Japan
‡IT Platform Division Group, Information & Telecommunication Systems Company, Hitachi, Ltd.
292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817 Japan
¶Institute of Industrial Science, The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
§National Institute of Infomatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
E-mail: †{ hideomi.idei.ub, kazuya.hisaki.bx }@hitachi.com,
‡{ shinji.fujiwara.yc, kazuhiko.mogi.uv }@hitachi.com, ¶{kgoda, kitsure}@tkl.iis.u-tokyo.ac.jp

1. はじめに

近年，クラウドコンピューティングの拡大や，多機能情報端末の急速な普及等により，企業や社会活動で発生するデータが増加している．また，グローバルでの事業拡大や，新事業の創出，より豊かでスマートな社会の実現に向けて，ビッグデータ利活用に対する期待が急速に高まっており，データの超高速な検索処理を可能にするデータベース製品が求められている．

このような中，我々は，内閣府最先端研究開発支援プログラムにおいて，アウトオブオーダ型データベースエンジン(OoODE)と称する実行原理に基づく超高速データベースエンジンの研究開発を進めてきている[1]．日立は 2012 年 6 月に，本研究開発成果を基にした商用のアウトオブオーダ型データベースエンジン Hitachi Advanced Data Binder（HADB）を製品化した[2].

一方，DB（Database）のデータを記憶するストレージの記憶デバイスとして，半導体メモリの一種であるフラッシュメモリが注目されている．フラッシュメモリは，従来の HDD（Hard Disk Drive）と比較してデータの入出力が高速で消費電力が低いという特徴がある[3]．その反面，記憶容量やビットコスト等に課題があったが，ハードウェア技術の進歩とともに大容量化や低価格化が進み，小・中規模程度の DB については全データをフラッシュメモリ構成のストレージ環境（以下，フラッシュストレージ環境）に格納することが可能となりつつある．このような背景を踏まえ，OoODE においては，従来の HDD のストレージ環境（以下，HDD 環境）だけでなく，フラッシュストレージ環境でも高い性能を発揮することが明らかにされている[4].

また，最近の CPU（Central Processing Unit）はハイ

パワー化による消費電力増加の対策として，CPU コアがアイドル状態となった際に CPU クロックと電圧を調整して消費電力を抑える機能（本稿では省電力モードと呼ぶ）を有している[5]．尚，本機能では，CPU コアが再度動作を始めた場合に CPU クロックと電圧を元に戻すため，その処理でオーバヘッドが生じる．従来の HDD 環境では I/O 応答時間が長いため，相対的にそのオーバヘッドの影響は小さかった．しかし，フラッシュストレージ環境においては，I/O 応答時間が HDD 環境よりも 2 桁以上短くなるため，CPU 省電力モードの切り替えオーバヘッドの影響が顕著に現れる懸念があった．

そこで，前述の商用 OoODE である HADB，OS（Operating System）に Linux，ストレージにフラッシュストレージを用いた環境において，CPU 省電力モードが性能に与える影響について評価した．その結果，従来の順序実行型のデータベースエンジン（以下，IODE）では CPU 省電力モードにより性能が 20%程度低下したが，OoODE では性能差を 1~2%程度に抑えることを確認した．以上より，フラッシュストレージ環境におけるいても OoODE の技術は，動作をしない時は CPU 省電力モードによって消費電力を抑えることができ，且つ動作する時は従来 IODE と比較して CPU 省電力モードの影響をほぼ受けずに動作することが可能であることを明らかにした．

本稿の構成は以下の通りである．2 章では OoODE の概要を説明し，3 章では性能評価の内容について述べる．4 章では性能評価結果について報告し，5 章で本稿を纏める．

2. OoODE の概要

2.1 OoODE 実行原理

アウトオブ型データベースエンジンは，図 1 に示すように問合せ処理をアンフォールドすることにより多数のプロセッサコアを活用し，また，複数のスレッドが非同期 I/O を同時に発行する高多重 I/O によって標準的な HDD が有する Native Command Queuing 機能を効率的に活用する．また，図 1 のディスクドライブがフラッシュメモリに置き換わった場合，フラッシュメモリは内部で並列動作が可能であるため，OoODE の高多重 I/O によってその性能を最大限引き出すことができる．このように OoODE は，サーバとストレージの性能を最大限引き出すことで性能向上を図る．

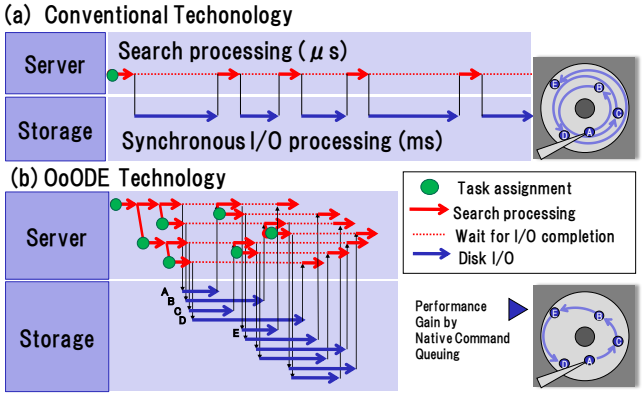

図 1 OoODE 実行原理

2.2 フラッシュストレージ環境における懸念事項

最近の CPU は消費電力を低減するため，アイドル状態となっている CPU コアの CPU クロックと電圧を調整して消費電力を抑える機能（省電力モード）を搭載している．例えば，図 1 (a)のように I/O 完了待ちをしている間の CPU コアはアイドル状態となるため，省電力モードを有効にしている場合は，該当 CPU コアの CPU クロックと電圧を下げてローパワー側に移行し，消費電力を小さくする．I/O が完了して CPU コアが再度動作を始めると元の CPU クロック・電圧に復帰する．その処理でオーバヘッドが生じるが，従来の HDD 環境の場合は I/O 応答時間が数十ミリ秒オーダと長いため，相対的にそのオーバヘッドの影響は小さい．しかし，フラッシュストレージ環境においては，I/O 応答時間が HDD 環境よりも 2 桁以上短くなるため，CPU 省電力モードの切り替えオーバヘッドの影響が顕著に現れることが考えられる．そこで，フラッシュストレージ環境において，CPU 省電力モードが OoODE の性能に与える影響について評価した．

3. 評価内容

3.1 システム構成

今回の評価に用いたシステムは，サーバに 10CPU コア(Intel® Xeon® E7-8870，2.4GHz)×4 ソケット，メモリ 512GB，RedHat® Enterprise Linux 6.2(OS)の Hitachi Blade Symphony BS2000，DB 用のストレージに 1.6TB の FMD(Flash Memory Device)を 9 台(7D+1P の RAID5 構成)搭載した Hitachi Unified Storage VM(HUS-VM)を用い，8GbpsFC（Fibre Channel）×8Port で接続したシステム構成である．また，8 台の FMD 全領域に 1 つの Raid Group を作成し，そこから 8 個の LU を切り出して，各 Port に 1 対 1 で割当てた構成としている(図 2)．

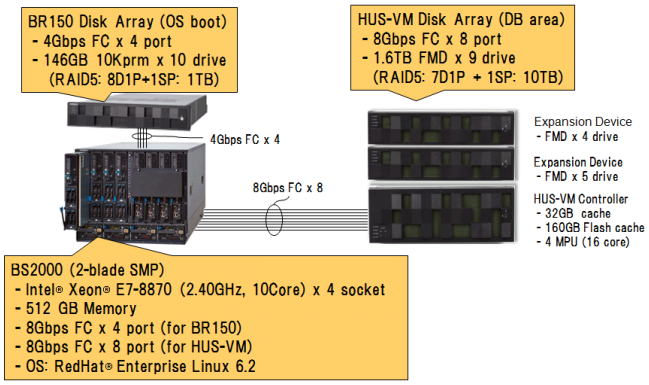

図 2 システム構成

3.2 OS，HBA 等のチューニング

3.1 章に記載の通り，OS には RedHat® Enterprise Linux6.2 を利用したが，インストールしたままのデフォルトの設定では，ストレージに十分な I/O を積めない等 OoODE の性能を最大限引き出すことができない．そこで，以下のパラメータのチューニングを実施した．

(1)CPU アフィニティの設定

特定の CPU コアに処理が集中してしまうことを防ぐため，DB 処理を行う CPU コア（OoODE の場合のみ），IRQ を受け付ける CPU コア，モニタ系（iostat/mpstat）の CPU コアの割当てを行う（図 3）．

Socket0		Socket1		Socket2		Socket3	
0	1	20	21	40	41	60	61
2	3	22	23	42	43	62	63
4	5	24	25	44	45	64	65
6	7	26	27	46	47	66	67
8	9	28	29	48	49	68	69
10	11	30	31	50	51	70	71
12	13	32	33	52	53	72	73
14	15	34	35	54	55	74	75
16	17	36	37	56	57	76	77
18	19	38	39	58	59	78	79

モニタ系CPU　IRQ処理CPU　DB処理CPU

※OoODEのみ

図 3 CPU アフィニティ設定

(2)HBA の設定

HBA 本体に対して以下を設定．

動的 QueueDepth を無効
ストレージ装置から Queue-Full が返ってきた場合，動的 QueueDepth が有効だと該当デバイスの queue_depth を OS が自動的に減少させ，I/O 性能にばらつきが発生するため無効にする．

HBA の全ポートに対して以下を設定．

QueueDepth に 256 を設定
ストレージに十分な I/O を発行するため．
割込み方式に msi を設定
IRQ が均等に分散するように，割込み機構に MSI(Message Signalled Interrupt)を設定する．

(3)デバイスの設定

全デバイスに対して以下を設定．

I/O スケジューラに noop を設定
I/O スケジューラがデフォルトの cfq の場合，OS が I/O キュー内で I/O の並び変えを行う．その処理に CPU を使われることを防ぐため，I/O の並び変えを行わない noop を設定する．
nr_request に 16,384 を設定
ストレージに十分な I/O を発行するため，nr_request パラメータに 16,384 を設定する．
queue_depth に 256 を設定
ストレージに十分な I/O を発行するため，queue_depth パラメータに 16,384 を設定する．
rq_affinity に 0 を設定
I/O 完了時の softirq 処理を I/O 発行した CPU コアで処理するデフォルト設定（rq_affinity=1）の場合，HBA・SCSI 構造体のアクセス競合が発生するため，I/O 完了時の softirq 処理を IRQ 受付け CPU コアで処理する設定（rq_affinity=0）にする．
add_random に 0 を設定
I/O 完了時のエントロピー収集を行うデフォルト設定（add_random=1）の場合，エントロピー構造体のアクセス競合が発生するため，I/O 完了時のエントロピー収集を行わない設定(add_random=0)にする．

3.3 評価環境

(1)DB 環境

DB 環境は，TPC-H[4]で規定されているスキーマで，データ規模として Scale factor が 3,000 のものを使用した．

(2)LVM（Logical Volume Maneger）構成

全 8LU に対して 1 つの Volume Group を作成し，以下の表・索引それぞれにストライプサイズ 4MB で 80 個の Logical Volume（LV）を作成した．これは，本評価で用いる商用 OoODE が複数のスキャン処理を並列実行する方式を採用しており，複数の LV を用いることで LV のロック競合を避けるためである．また，以下の表以外の表・索引及びワーク用としてそれぞれ 1 個の LV を作成した．

【表】
part/supplier/partsupp/customer/orders/lineitem
【索引】
idx_part/idx_supplier/idx_partsupp/idx_customer/
idx_orders/idx_lineitem

(3)性能評価用クエリ

性能評価用のクエリとして，TPC-H の Q08 をベースに，最外表である Part 表の絞込み条件を変えたクエリ（以下，Q08’）を使用した．図 4 に Q08’の SQL，図 5 に Q08’の実行プランを示す．尚，本稿では最外表（Q08’では Part 表）の全行数に対する選択行数の割合を絞込み率としている．

```
SELECT
   count(l_shipdate) l_year,
   sum( case
      when n2.n_name = 'CHINA'
      then l_extendedprice
      else 0
   end) /
      sum(l_extendedprice * (1 - l_discount)) mkt_share,
   n2.n_name
FROM
( ( ( ( ( ( (part
                inner join
                lineitem
                on p_partkey = l_partkey          Part表の絞込み条件
                 and p_type = 'STANDARD ANODIZED STEEL'
                 and p_brand = 'Brand#24'
                 and p_size < 3)
             inner join
             orders
             on l_orderkey = o_orderkey
               and o_orderdate between DATE '1995-01-01' and DATE '1996-12-31')
          inner join
          customer
          on o_custkey = c_custkey)
        inner join
        supplier
        on s_suppkey = l_suppkey)
      inner join
      nation n1
      on c_nationkey = n1.n_nationkey)
    inner join
    nation n2
    on s_nationkey = n2.n_nationkey)
 inner join
 region
 on n1.n_regionkey = r_regionkey
    and r_name = 'AMERICA')
GROUP BY
   n2.n_name
ORDER BY
   n2.n_name;
```

図 4　Q08’の SQL

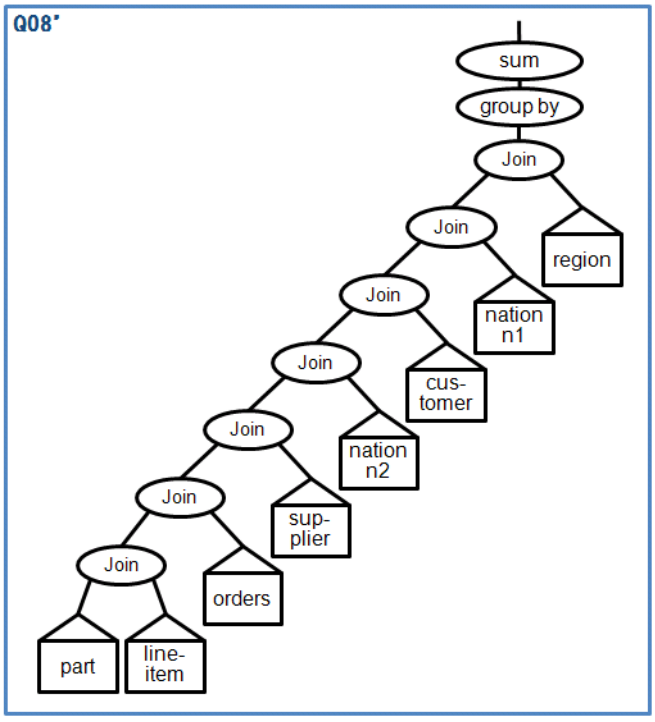

図 5　Q08’の実行プラン

(4)性能測定内容

本評価では，CPU 省電力モードを有効にした場合と無効にした場合でそれぞれ 3 回測定を行い，それらの平均値を結果としている．尚，比較のため OoODE だけでなく IODE についても性能測定を実施した．IODE と OoODE では同一クエリでもクエリ実行時間が大幅に異なるため，今回の評価ではクエリ実行時間が同程度となるように，IODE と OoODE で最外表の絞込み条件を調整し，以下の絞込み率で測定を実施した．

[IODE]
絞込み率：5e-7/1e-6/5e-6/1e-5/2e-5/3e-5/5e-5
[OoODE]
絞込み率：3e-5/5e-5/1e-4/5e-4/1e-3/3e-3/5e-3

4.　評価結果

4.1　測定結果

(1)IODE 測定結果

IODE 測定結果のグラフを図 6 に示す．本グラフでは，省電力モード OFF で絞込み率 5e-7 のクエリ実行時間を 1 としたクエリ実行相対時間を縦軸，絞込み率を横軸として，各絞込み率のポイントで省電力モードが無効（OFF）の結果と有効（ON）の結果を並べて表示している．

本評価において，IODE では，CPU 省電力モードを有効にした場合，CPU 省電力モードが無効の場合と比較して性能が 20%程度低下した．

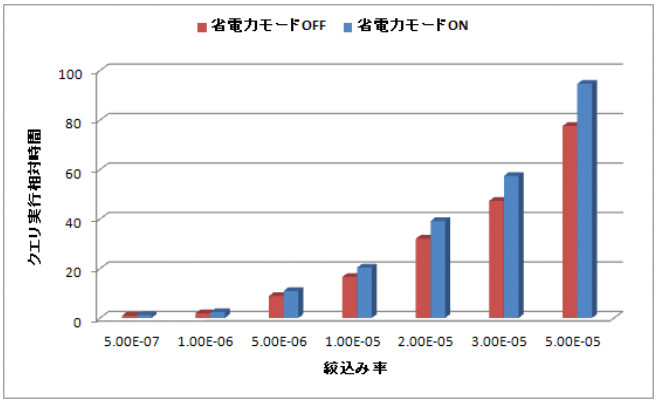

図 6 IODE 測定結果

(2)OoODE 測定結果

OoODE 測定結果のグラフを図 7 に示す. 本グラフも省電力モード OFF で絞込み率 3e-5 のクエリ実行時間を 1 としたクエリ実行相対時間を縦軸，絞込み率を横軸として，各絞込み率のポイントで省電力モードが無効（OFF）の結果と有効（ON）の結果を並べて表示している.

本評価において，OoODE では，CPU 省電力モードを有効にした場合でも CPU 省電力モードが無効の場合とほぼ同等の 1~2%の性能差となった.

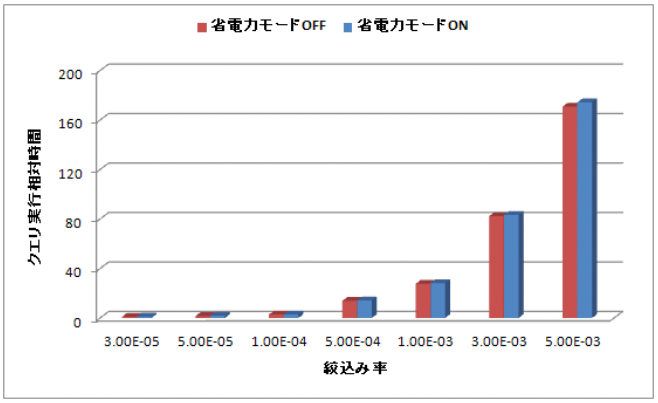

図 7 OoODE 測定結果

4.2 考察

フラッシュストレージ環境において，CPU 省電力モードを有効にした場合，従来の IODE では 20%程度性能が低下した．IODE はシングルスレッドで動作しているため，I/O 完了待ちでアイドル状態（ローパワー）となっている CPU コアが I/O 完了後に再度動作を始める際，通常時の CPU クロック・電圧に復帰する処理のオーバヘッドが性能に直接影響を与えているためだと考えられる。

これに対し，フラッシュストレージ環境において，OoODE では CPU 省電力モードの影響を 1~2%程度に抑えることができた．OoODE では，アウトオブオーダ型データベースエンジンの実行原理に基づいて複数の処理を並列に実行しており，各 CPU コアをアイドルに落とさずに効率良く使用する．そのため，IODE と比較してアイドル状態になる割合が低く CPU 省電力モードの影響が小さいと考えられる.

5. おわりに

本論文では，著者らが研究開発を進めている成果を利用した商用のアウトオブオーダ型データベースエンジン Hitachi Advanced Data Binder（HADB）を用い，近年注目を集めているフラッシュストレージ環境において，CPU 省電力モードが性能に与える影響について評価した．その結果，同データベースエンジンは，OS に Linux，ストレージにフラッシュストレージを用いた環境において，動作をしない時は CPU 省電力モードによって消費電力を抑えることができ，且つ動作する時はアウトオブオーダ型データベースエンジンの実行原理により，従来の IODE と比較して CPU 省電力モードの影響をほぼ受けずに動作することが可能であることを確認した.

謝　辞

本研究は，内閣府最先端研究開発支援プログラム「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の助成により行われた.

文　献

[1] 喜連川優, 合田和生,アウトオブオーダ型データベースエンジン OoODE の構想と初期実験，日本データベース学会論文誌，Vol.8，No.1，pp.131-136，2009.

[2] 日立，東京大学との超高速データベースエンジンの共同研究開発成果を製品化，http://www.hitachi.co.jp/New/cnews/month/2012/05/0528.html, 2012.

[3] 森山正秋，Hitachi Accelerated Flash がもたらすストレージシステムの変革，IDC Japan，Feb-2013，http://www.hitachi.co.jp/products/it/storage-solutions/techsupport/whitepaper/pdf/500170_hitachi_wp.pdf

[4] 早水悠登，合田和生，喜連川優，フラッシュストレージ環境におけるアウトオブオーダ型データベースエンジン OoODE の実験的クエリ処理性能評価，第 6 回データ工学と情報マネジメントに関するフォーラム, 2014 (to appear)

[5] Alon Naveh , Doron Rajwan , Avinash Ananthakrishnan、Eli Weissmann, Power management architecture of the 2nd generation Intel® Core™ microarchitecture, formerly codenamed Sandy Bridge, Hot Chips Aug-2011, http://www.hotchips.org/wp-content/uploads/hc_archives/hc23/HC23.19.9-Desktop-CPUs/HC23.19.921.SandyBridge_Power_10-Rotem-Intel.pdf

[6] Transaction Processing Performance Council, TPC-H All Results – Sorted by Performance, http://www.tpc.org/tpch/results/tpch_results.asp.