半導体ディスク装置 RamSan-400 による Oracle 10g Database パフォーマンス改善 検証報告書（本編）

システムクリエイト株式会社　技術部

白井紀行

野村公挙

須藤マルコス

平成 19 年 9 月 27 日

1　検証概要

Oracle 10g Database 上に 100GB（≒2 億件）の大きさのデータベースを作成して、Disk Array への負荷がほぼ 100%となる OLTP をシミュレートするベンチマークテストを実施し、特に I/O アクセスの多いオブジェクトを半導体ディスク RamSan-400(32GB)に移動する前後のパフォーマンスを比較することで効果を検証した。

1.1　検証環境

1.1.1 実施期間

日程	平成 19 年 8 月 16 日(月)～平成 19 年 8 月 31 日（金）	土日を除く 10 営業日
時間	09:30～19:00	DB の作成は夜間

1.1.2 検証場所

富士通 Platform Solution Center（浜松町 世界貿易センタービル 30F）

1.1.3 検証担当

責任者	技術部　白井紀行
ハードウェア設定＆ベンチマーク	技術部　野村公挙
Oracle 設定＆ベンチマーク	技術部　須藤マルコス

1.1.4 ハードウェア環境

1.1.4.1　サーバ

表 1-1 にある 5 つのプラットフォーム（3 種類のハードウェア、3 種類の OS の組み合わせ）で検証を行いました。
ただし、PRIMEQUEST520 の Windows 環境については、時間の関係でインターオペラビリティのみの確認となっています。

Server	CPU	MEM	OS	HBA	HBA Driver
SPARC Enterprise M4000	SPARC64VI/2150Mhz 4CPU/8core/16 ｽﾚｯﾄﾞ	16GB	Solaris 10 OS 11/06	4Gbps SE0X7F11F×2	FJSVpfcaV40L00
PRIMERGY RX600 S2	IntelXeonMP/3Ghz 8CPU	2GB	Windows Server 2003 R2, Enterprise x64 Edition	2Gbps Emulex LP1050	7-1.10A4 x64
			Red Hat Enterprise Linux AS V4U4 （2.6.9-42.ELsmp x86_64）		lpfc_2.6_ioctl_module-2.0.15-1
PRIMEQUEST 520	Intel Itanium2 /1500Mhz 8CPU	32GB	Windows 2003 Server Enterprise Edition SP1 for Itanium	4Gbps Emulex LPe11002-M4	6-1.11A3 64bit IA64
			RedHat Enterprise Linux ASV4U4(2.6.9-42.EL)		lpfc-fjext-RHEL4-U4-8.0.16.27-1

表 1-1　サーバ仕様一覧

1.1.4.2　ストレージ

① Disk Array - Fujitsu ETERNUS4000（モデル E408R224A）

<table>
<tr><td>キヤッシュ</td><td>1GB</td></tr>
<tr><td>ディスクドライブ</td><td>FC73GB×15 台（3 台で RAID0 を作成し、上記 5 種のプラットホームに対し 1 LUN ずつ割り当て）
ただし、PQ（Win2003 は SAN boot のため、15GB の OS 領域も割り当て）
<table>
<tr><th colspan="2">WWN</th><th>Zone 名</th><th>LUN-Vol</th><th>Capacity</th></tr>
<tr><td>SPARC Enterprise</td><td>1000000b5d660062(Slot#2)</td><td>SE_Sol</td><td>0000</td><td>200GB</td></tr>
<tr><td>PRIMERGY(L)</td><td>10000000c94a6ee3(Slot#3)</td><td>PG_Linux</td><td>0001</td><td>200GB</td></tr>
<tr><td>PRIMERGY (W)</td><td>10000000c94a6e97(Slot#1)</td><td>PG_Win</td><td>0002</td><td>200GB</td></tr>
<tr><td>PRIMEQUEST(L)</td><td>10000000c953c042(Slot#0)</td><td>PQ_linux</td><td>0003</td><td>200GB</td></tr>
<tr><td rowspan="2">PRIMEQUEST(W)</td><td rowspan="2">10000000c953c043(Slot#1)</td><td rowspan="2">PQ_Win</td><td>0005</td><td>15GB</td></tr>
<tr><td>0004</td><td>190GB</td></tr>
</table></td></tr>
<tr><td>ホストインターフェイス</td><td>4Gbps(CM#0 CA#0 Port#0 のみ使用)</td></tr>
<tr><td>ディスクインターフェイス</td><td>2Gbps</td></tr>
</table>

表 1-2 ETERNUS4000 仕様

② Texas Memory Systems 半導体ディスク装置 RamSan-400（モデル　RamSan-400/32GB）※詳細は 4.1 章を参照のこと

メモリサイズ	32GB（1LUN を作成）
ホストインターフェイス	4Gbps(FC-AL)
ファームウェア	2.7.0n

表 1-3 RamSan-400 仕様

1.1.5 ハードウェア接続図

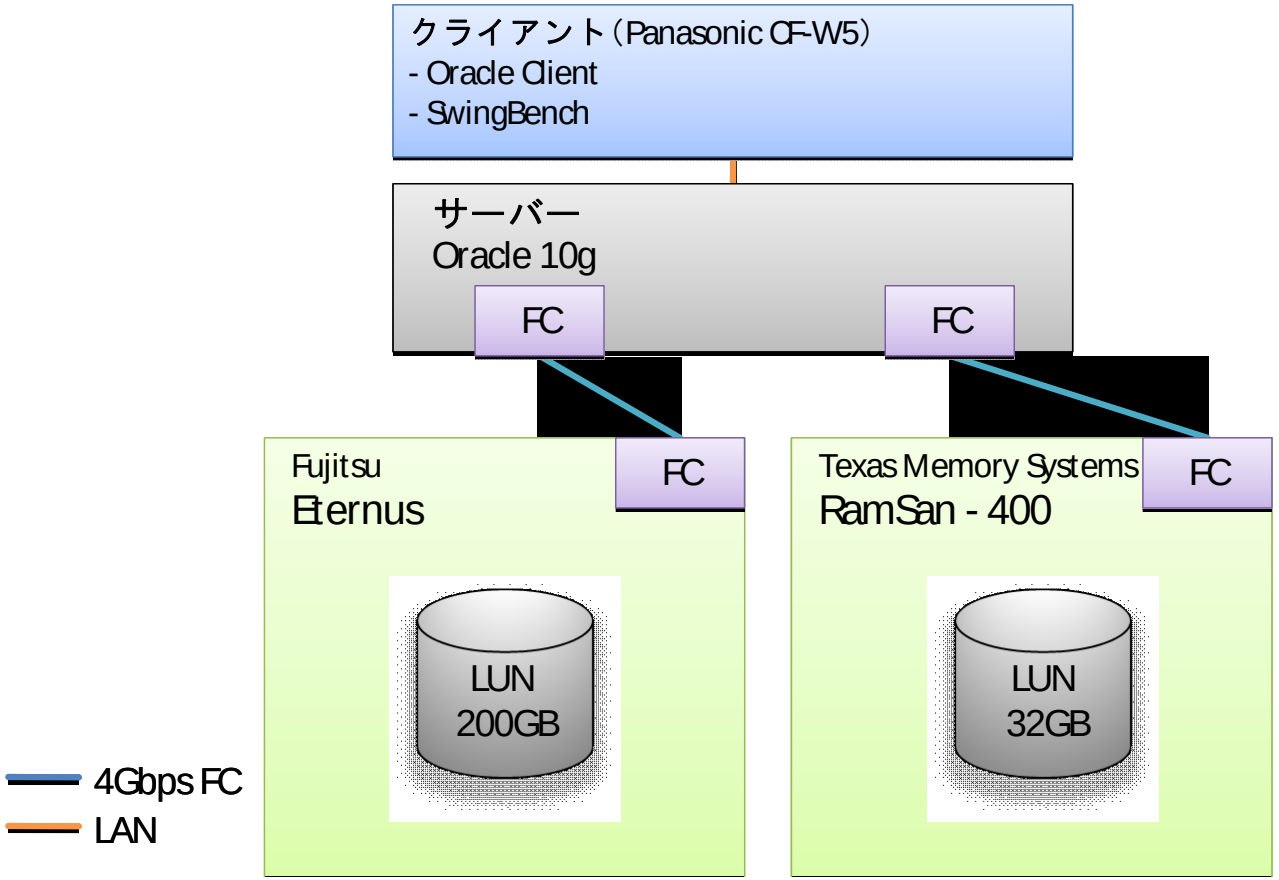


図 1-1　検証用ハードウェア接続図

1.2 検証内容

今回の検証は Oracle 10g Database における RamSan-400 の導入効果がメインとなっていますが、これに付随した形でインターオペラビリティ（接続性）の確認も実施しました。

1.2.1 Oracle ベンチマーク

1.2.1.1 検証用ソフトウェア

バージョン	Oracle 10g Database　ver. 10.2.0（各プラットフォームとも同じバージョンを使用）
ベンチマークツール	SwingBench ver. 2.3（詳細については、4.2 章を参照のこと）

表 1-4　ソフトウェア一覧

1.2.1.2 検証手順

① Oracle10g Database を標準インストール
② データベースを作成して ETERNUS 上に全てのデータファイルを置く。
③ Swing Bench にてダミーデータを作成する。
④ ベンチマークを実施
⑤ 表 1-5 のオブジェクトを RamSan-400 に移動。
⑥ 再度、④と同じ条件でベンチマークを実施し、移動前後のパフォーマンスを比較する。

1.2.1.3　移動オブジェクト

RamSan-400 は、400,000IOPS というパフォーマンスで I/O ボトルネックを解消できるソリューションですが、テラバイトクラスにもなるデータベースにおいて、ストレージを全て RamSan-400 に置き換えてシステムを構築するのは、費用面から考えても現実的ではありません。

そこで、容量では全体の数%でありながら、I/O アクセスは全体の 60 パーセント以上を占めるホットファイルを RamSan-400 に移動してパフォーマンスを向上させるというアプローチを取るのが一般的です。

今回の検証でも、このような状況をシミュレートしています。

RamSan-400 の容量(32GB)の 3 倍以上にあたる 100GB のデータベースを作成し、リハーサル（予備テスト）で得られた結果を元に、I/O アクセスの多かった合計 13GB のオブジェクト――Statspack の Segments by Physical Reads にランキングされた索引、Redo ログファイル、Undo ファイル――を RamSan-400 に移動して前後のパフォーマンスを比較しました。

索引	CUST_EMAIL_IX, CUST_UPPER_NAME_IX, ORD_CUSTOMERS_PK
ログファイル	REDO01~03.LOG（移動前後では Redo ログファイルの設定は変更していません）
データファイル	UNDOTBS01.DBF, USERS01.DBF

表 1-5　RamSan-400 への移動オブジェクト

1.2.1.4　効果確認

オブジェクト移動前後の結果を、表 1-6 に示す 5 項目で比較して、効果を確認しました。

パフォーマンス効果	● 1 時間の総トランザクション数(Total Transactions) ● 平均応答時間
Statspack 解析	Top5 Timed Event ● % Total Call ● トランザクション当たりの Wait（Event per min. / Total Transactions per min.）
SwingBench	実行（キャプッチャー）画面による評価※
4 つのトランザクションの応答時間	①Customer Registration（顧客の作成）、②Browse Products（製品の表示）、③Browse Orders（注文の確認）、④Process Orders（発注）
OS ツールでのパフォーマンス	● CPU の Idle ● IO Status

表 1-6　効果確認

1.2.2 インターオペラビリティ

各プラットフォームの HBA に RamSan-400 に繋いで認識させ、通常の Fibre Channel ディスク装置を扱う時に使うコマンド（または、ユーティリティ）――ディスクの認識、パーティション設定、ファイルシステムの作成――を実行して、正常にマウントが出来るまでを確認しました。

1.2.2.1 Solaris10

ディスクの認識	①┌ dmesg　②iostat -E
パーティション作成	format　（今回は、全領域(s2)を使用）
ファイルシステムの作成	Newfs

表 1-7　Solaris ディスクコマンド

1.2.2.2 Windows2003

ディスクの認識	コンピュータの管理→システムツール→デバイスマネージャー→ディスク
パーティション作成	コンピュータの管理→記憶域→ディスクの管理
ファイルシステムの作成	同上

表 1-8　Windows2003 ディスクコマンド

1.2.2.3 Red Hat Enterprise Linux AS Ver4

ディスクの認識	①dmesg　②cat /proc/scsi/scsi
パーティション作成	fdisk　(今回は、1 パーティションのみ)
ファイルシステムの作成	mkfs –t ext3

表 1-9　Linux ディスクコマンド

2　Oracle 10g Database ベンチマーク

2.1　Oracle 10g Database インストール

2.1.1 ファイルシステムのマウント

Solaris と Linux は、/u02/etfc に ETERNUS4000 を/u02/rsfc に RamSan-400 をマウント。Windows2003 は、E ドライブ（ボリューム名 ETFC）に ETERNUS4000 を、F ドライブ（ボリューム名：RSFC）に RamSan-400 をマウントしています。

ファイルシステム	サイズ	使用済み	使用可能	容量	マウント先	
/dev/dsk/c0t0d0s0	52G	14G	37G	28%	/	
/devices	0K	0K	0K	0%	/devices	
ctfs	0K	0K	0K	0%	/system/contract	
proc	0K	0K	0K	0%	/proc	
mnttab	0K	0K	0K	0%	/etc/mnttab	
swap	22G	1.3M	22G	1%	/etc/svc/volatile	
objfs	0K	0K	0K	0%	/system/object	
fd	0K	0K	0K	0%	/dev/fd	
swap	22G	32K	22G	1%	/tmp	
swap	22G	72K	22G	1%	/var/run	
/dev/dsk/c1t0d0s2	192G	46G	145G	24%	/u02/etfc	【ETERNUS4000】
/dev/dsk/c2t0d0s2	32G	32M	31G	1%	/u02/rsfc	【RamSan-400】

図 2-1　df コマンドの実行例

2.1.2 インストール方法

Oracle10g Database のインストールは、標準のインストール（Oracle Universal Installer）を実施。
OS 固有の設定はインストールマニュアルの設定値を使用しました。

インストールの方法	拡張インストール
インストールタイプ	カスタム
ホームの詳細の指定	名前：デフォルト、ホーム：システムディスク上
使用可能な製品コンポーネント	Enterprise Manager
データベースの作成	ソフトウェアインストールのみ

表 2-1　Oracle インストール

2.1.3 データベースの作成

データベースの作成は、データベースコンフィグレーションアシスタント（DBCA）を使って行いました。

テンプレート	カスタム
データベース名	グローバルデータベース名：SID.local SID：sedbX（SPARC Enterprise）、pgdbX（PRIMERGY）、pqdbX(PRIMEQUEST)
ファイルシステム	ファイルシステム（Solaris：UFS、Windows：NTFS、RHEL4：ext3）
ファイルの位置	/u02/etfc/DB_NAME　（Windows は、E:¥etfc/DB_NAME）
Oracle 用合計メモリ	SPARC Enterprise：5.7GB(40%)、PRIMERGY：1GB(50%)、PRIMEQUEST：25GB(80%)

データファイルのパス、サイズ又は表領域設定画面

表領域に＊がついたものは、RamSan-400 への移動オブジェクトが含まれています。

表領域	データファイル	備考
SYSAUX	sysaux01.dbf	SYSTEM 表領域の補助表領域。関連ツールのリポジトリ
SYSTEM	system01.dbf	データディクショナリや管理データなどを格納する
TEMP	temp01.dbf	一時データ（ソートデータなど）を格納する
UNDOTBS*	undotbs01.dbf	UNDO データ（変更前に戻すためのデータ）を格納する
USERS*	users01.dbf	非システムユーザーが作成するセグメントを格納する。
SCOTT_DATA	scott_data.dbf	実データのデータファイル
SCOTT_INDEX*	scott_index.dbf	索引のデータファイル
SCOTT_SSD	scott_sdd.dbf	移動オブジェクト（索引）を納めるファイル（RamSan-400 上）
コントロールファイルグループ		
Control01-03.ctl		データファイルや REDO ログファイルの情報、Oracle データベースの管理情報を格納する。
REDO ロググループ		
Redo01-03.log*		データベースに対して行われた変更（DML、DDL）が記録され、データベースに障害があったときの回復に使用する。

表 2-2　データベース記憶域

2.1.4 Oracle 初期パラメータ（Solaris 10 の例）

init.ora Parameters

Parameter Name	Begin value	End value (if different)
audit_file_dest	/u01/app/oracle/10.2.0/db_1/admin/sedb1/adump	
background_dump_dest	/u01/app/oracle/10.2.0/db_1/admin/sedb1/bdump	
compatible	10.2.0.1.0	
control_files	/u02/etfc/sedb1/control01.ctl, /u02/etfc/sedb1/control02.ctl, /u02/etfc/sedb1/control03.ctl	
core_dump_dest	/u01/app/oracle/10.2.0/db_1/admin/sedb1/cdump	
db_block_size	2048	
db_domain	local	
db_file_multiblock_read_count	16	
db_name	sedb1	
job_queue_processes	10	
nls_language	JAPANESE	
nls_territory	JAPAN	
open_cursors	300	
pga_aggregate_target	1707081728	
processes	500	
remote_login_passwordfile	EXCLUSIVE	
sessions	555	
sga_target	4294967296	
statistics_level	ALL	
undo_management	AUTO	
undo_tablespace	UNDOTBS1	
user_dump_dest	/u01/app/oracle/10.2.0/db_1/admin/sedb1/udump	

表 2-3　Oracle 初期化パラメータ

2.1.5 ファイル配置の例（Solaris10）

SwingBench によってダミーデータを作成すると/u02/etfc の下に図 2-2 のようなファイルが配置されます。

```
bash-3.00# ls -lh
合計 148728768
-rw-r-----   1 oracle   oinstall     21G  8 月 23 日  11:00 SCOTT_DATA.DBF
-rwxr-xr-x   1 oracle   oinstall     39G  8 月 23 日  11:00 SCOTT_INDEX.DBF
-rw-r-----   1 oracle   oinstall    8.6M  8 月 23 日  11:00 control01.ctl
-rw-r-----   1 oracle   oinstall    8.6M  8 月 23 日  11:00 control02.ctl
-rw-r-----   1 oracle   oinstall    8.6M  8 月 23 日  11:00 control03.ctl
-rw-r-----   1 oracle   oinstall     50M  8 月 23 日  10:54 redo01.log
-rw-r-----   1 oracle   oinstall     50M  8 月 23 日  10:57 redo02.log
-rw-r-----   1 oracle   oinstall     50M  8 月 23 日  10:59 redo03.log
-rw-r-----   1 oracle   oinstall    180M  8 月 23 日  11:00 sysaux01.dbf
-rw-r-----   1 oracle   oinstall    300M  8 月 23 日  11:00 system01.dbf
-rw-r-----   1 oracle   oinstall    8.0G  8 月 23 日  10:40 temp01.dbf
-rw-r-----   1 oracle   oinstall    2.5G  8 月 23 日  11:00 undotbs01.dbf
-rw-r-----   1 oracle   oinstall    5.0M  8 月 23 日  10:57 users01.dbf
```

図 2-2　/u02/etfc のファイル配置

2.2 Swing Bench

2.2.1 Order Entry ウィザード

Swingbench に用意されている Order Entry ウィザードを使ってダミーデータを作成します。

作成オブジェクト	データファイル	オブジェクト名
表	SCOTT_DATA.DBF	WAREHOUSES、PRODUCT_INFORMATION、PRODUCT_DESCRIPTIONS、LOGON、INVENTORIES、ORDERS、ORDER_ITEMS、CUSTOMERS
索引	SCOTT_INDEX.DBF	WHS_LOCATION_IX、INV_PRODUCT_IX、INV_WAREHOUSE_IX、ITEM_ORDER_IX、ITEM_PRODUCT_IX、ORD_SALES_REP_IX、ORD_CUSTOMER_IX、 ORD_ORDER_DATE_IX、ORD_STATUS_IX、 CUST_ACCOUNT_MANAGER_IX、 CUST_LNAME_IX、CUST_EMAIL_IX、PROD_NAME_IX、 PROD_NAME_IX、 PROD_SUPPLIER_IX、CUST_UPPER_NAME_IX

表 2-4　作成されたダミーデータ

2.2.2 Benchmark 環境

Swingbench のパラメータを表 2-5 のように設定してベンチマークを実施しました。

ユーザ数	100
ベンチマーク実行時間	1 時間
データベースサイズ	100GB、約 2 億件
ログイン Delay	0　（全てのユーザが同時ログイン）
トランザクション Delay	0　（各ユーザが同時に 4 つのトランザクションを実行）
各トランザクション負荷	1:1:1:1
DML 負荷	Select 50%, Insert 17,7%, Update 6,8%, Commit 23,4%, Rollback 1,7%

表 2-5　SwingBench 設定パラメータ

2.3　ベンチマーク結果

2.3.1 SPARC Enterprise M4000（Solaris10）

2.3.1.1　パフォーマンス改善効果

オブジェクトの移動によって、1 時間の総トランザクション数は 75 万から 152 万と 77 万の増加、1 トランザクションの平均応答時間は 496ms から 242ms と 254ms 短縮し、約 2 倍の改善効果が見られました。

比較項目	1 時間の総トランザクション数	平均応答時間(ms)
オブジェクト移動前	745,112	496.75
オブジェクト移動後	1,524,845	242.50
パフォーマンス改善効果	2.0 倍	2.0 倍

表 2-6　パフォーマンス改善効果

2.3.1.2　Statspack

Top 5 Timed Events

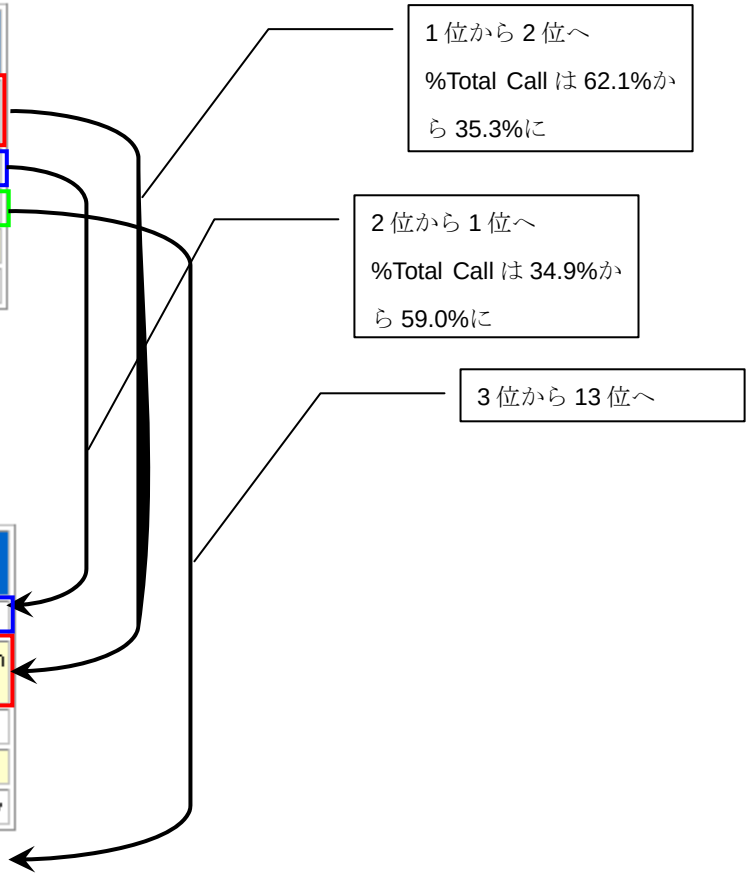
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log file switch (checkpoint incomplete)	156,022	151,291	970	62.1	Configuration
db file sequential read	741,112	84,904	115	34.9	User I/O
log file sync	212,656	4,843	23	2.0	Commit
db file parallel write	4,849	4,813	993	2.0	System I/O
buffer busy waits	3,846	1,526	397	.6	Concurrency

表 2-7　オブジェクト移動前

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
db file sequential read	1,866,971	142,725	76	59.0	User I/O
log file switch (checkpoint incomplete)	92,424	85,477	925	35.3	Configuration
buffer exterminate	8,614	8,424	978	3.5	Other
db file parallel write	9,175	4,795	523	2.0	System I/O
buffer busy waits	4,225	2,175	515	.9	Concurrency

表 2-8　オブジェクト移動後



- % Total Call

TOP5 Timed Events を比較すると、オブジェクト移動前の 1 位は log file switch (checkpoint incomplete)――ログファイルスイッチの終了待ちで発生する待機イベント――が 62.1%。2 位は Db sequential read――トランザクションによる IO 待ち――が 34.9%でした。

オブジェクト移動後は Db sequential read が 1 位になって待機イベントの 59.0%、log file switch (checkpoint incomplete)は 2 位に下がり、待機イベントの 35.3%になりました。log file sync は移動前には 3 位でしたが移動後では 13 位に下がりました。

- **トランザクション当たりの Wait**

	Txn/min.	Log file switch		Log file sync		Db file sequential read	
		Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})
移動前	12,017.9	3879.3	322.8	124.2	10.33	2177.0	181.1
移動後	24,594.3	2084.8	84.8	1.9	0.08	3481.1	141.5
改善効果	2.05	3.81		137.10		1.28	

表 2-9　トランザクション当たりの Wait の比較

オブジェクトを移動する前のトランザクション当たりの Wait は log file switch (checkpoint incomplete)は 3.81 倍、log file sync は 137.10 倍、Db file sequential read は 1.28 倍かかっていました。特に log file sync に大きな効果があったことが分かります。

2.3.1.3　Swing Bench のキャプチャー画面

オブジェクト移動前はログファイルの影響（Log file Sync、Log file switch など）で一時的にトランザクションが完全に止まる事がありましたが、オブジェクトを RamSan-400 へ移動後は、停止することはありませんでした。

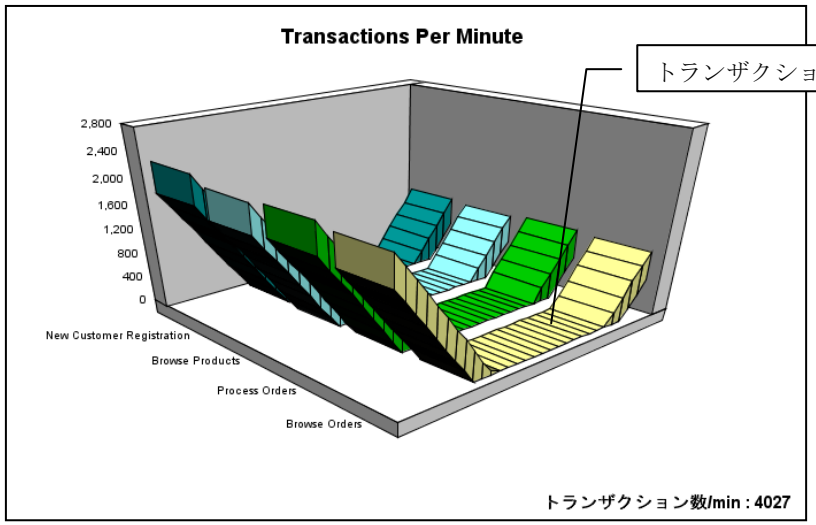


図 2-3　オブジェクト移動前

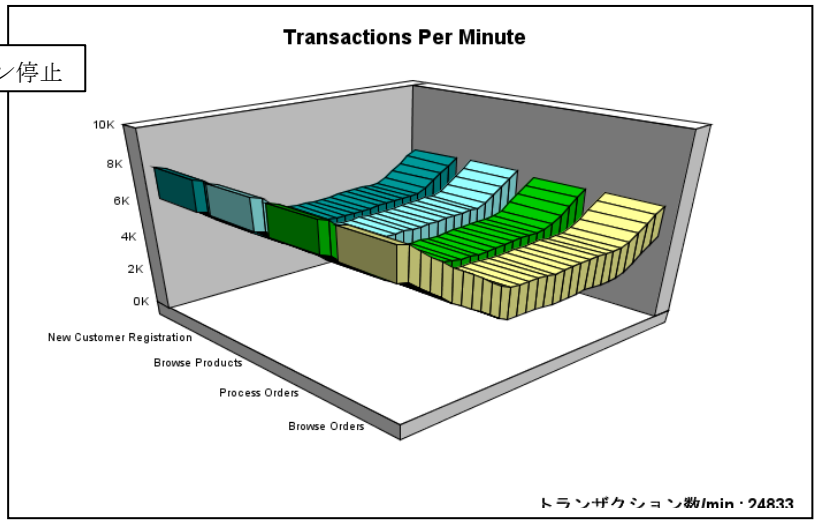
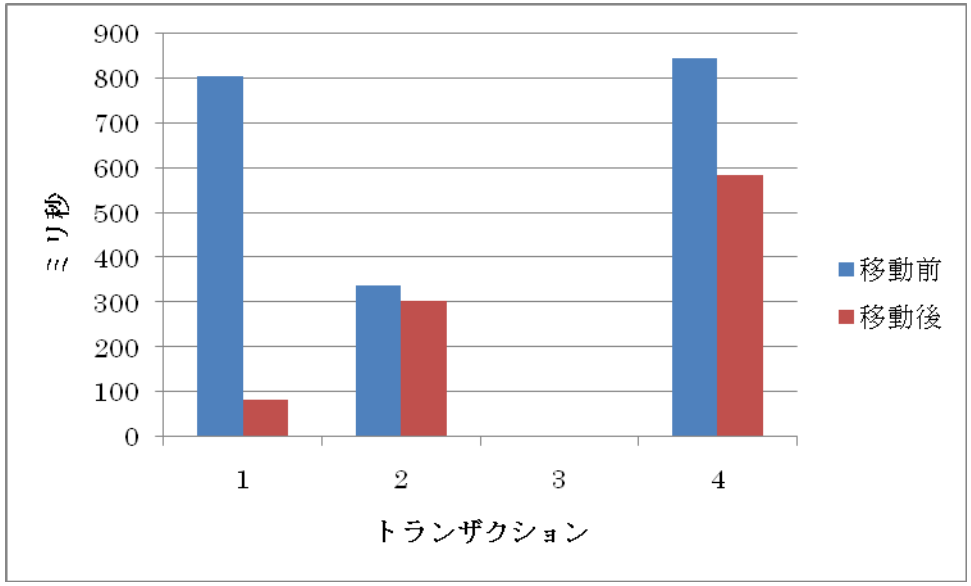


図 2-4　オブジェクト移動後

2.3.1.4　4 つのトランザクションの応答時間

4 つのトランザクションの応答時間は、オブジェクトを RamSan-400 に移動すると、最大で約 10 分の 1 に、平均的に 3 分の 1 に改善されました。



① Customer Registration（顧客の作成）
② Browse Products（製品の表示）
③ Browse Orders（注文の確認）
④ Process Orders（発注）

図 2-5　各トランザクションの応答時間

2.3.1.5　OS ツールでのパフォーマンス測定

	CPU Idle	IOBusy	
		Disk Array	RamSan-400
オブジェクト移動前	96,3%	99,3%	―
オブジェクト移動後	93,7%	98,6%	9,53%

表 2-10　IOSTAT によるリソースの使用状況

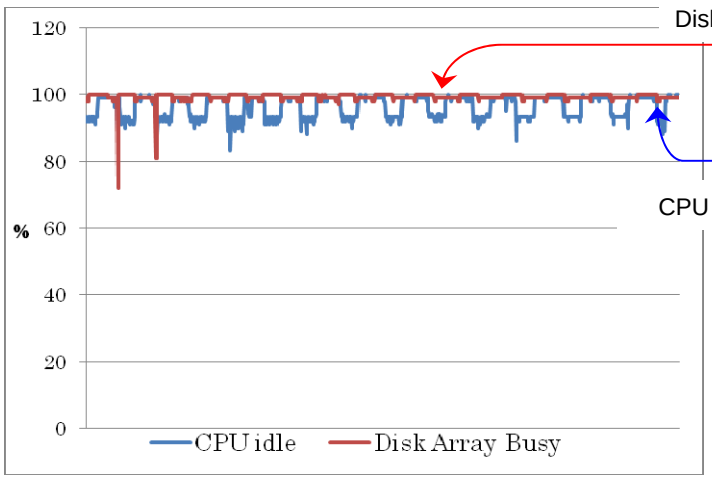


図 2-5　オブジェクト移動前

図 2-6　オブジェクト移動後

2.3.1.6　考察

移動後では全体的に 2.0 倍(トランザクション数)のパフォーマンス向上が確認されました。

移動前の場合では 12,017 トランザクション／min に対して CPU の平均使用率は 3.3%と Disk Array の平均使用率は 99.3%に達してシステムのボトルネックになっていました。

オブジェクトを RamSan-400 へ移動後は 24,594.3 トランザクション／min に増加しました。CPU の平均使用率は 6.3%に上がりましたが Disk Array は 99%近くの平均使用率で、RamSan-400 の平均使用率は 10%となりましたので、更に適切なオブジェクトを RamSan-400 に移動することでパフォーマンスの改善が見込まれます。

2.3.1.7　参考データ

データベースサイズを 30GB にして、すべてを RamSan-400 に載せた場合の 1 時間の総トランザクション数は、9,434,342 でした。

データベースサイズは異なりますが、単純に Disk Array のみのトランザクション数 745,112 と比較すると 12.66 倍のパフォーマンス改善になります。実際のデータアクセス Db file sequential read に注目して Txn/Wait を計算すると 121 倍速くなっていました。

この場合 CPU の平均使用率は 68.1%でオブジェクト移動前の 3.3%と比べると約 21 倍リソースが使われることになります。この時の RamSan の平均使用率は 43.1%でした。

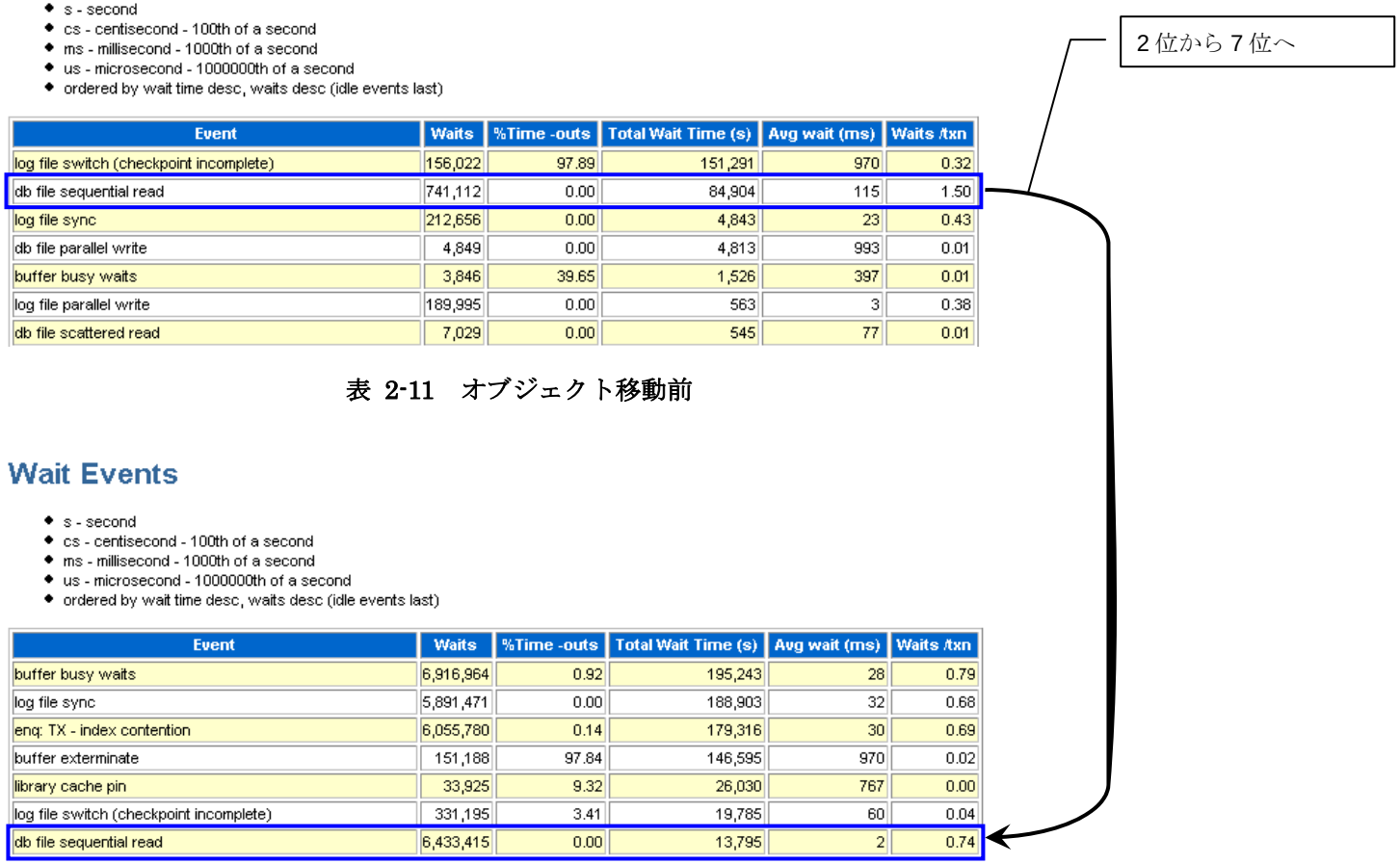
Statspack では Concurrency(同時アクセス)の待機イベントが 44%を占めるようになりました。これは Oracle10g Database のチューニングで解決することで、パフォーマンスを上げることが可能と考えられます。

Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (idle events last)

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
log file switch (checkpoint incomplete)	156,022	97.89	151,291	970	0.32
db file sequential read	741,112	0.00	84,904	115	1.50
log file sync	212,656	0.00	4,843	23	0.43
db file parallel write	4,849	0.00	4,813	993	0.01
buffer busy waits	3,846	39.65	1,526	397	0.01
log file parallel write	189,995	0.00	563	3	0.38
db file scattered read	7,029	0.00	545	77	0.01

表 2-11　オブジェクト移動前



Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (idle events last)

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
buffer busy waits	6,916,964	0.92	195,243	28	0.79
log file sync	5,891,471	0.00	188,903	32	0.68
enq: TX - index contention	6,055,780	0.14	179,316	30	0.69
buffer exterminate	151,188	97.84	146,595	970	0.02
library cache pin	33,925	9.32	26,030	767	0.00
log file switch (checkpoint incomplete)	331,195	3.41	19,785	60	0.04
db file sequential read	6,433,415	0.00	13,795	2	0.74

表 2-12　オブジェクト移動後

2.3.2 PRIMERGY RX600 S2（Windows2003）

2.3.2.1　パフォーマンス改善効果

オブジェクトの移動によって、1 時間あたりの総トランザクション数は 118 万から 197 万と 79 万の増加、1 トランザクションの平均応答時間は 306ms から 180.5ms と 125ms 短縮し、約 1.7 倍の改善効果が見られました。

比較項目	1 時間の総トランザクション数	平均応答時間(ms)
オブジェクト移動前	1,180,471	306.0
オブジェクト移動後	1,968,168	180.5
パフォーマンス改善効果	1.7 倍	1.7 倍

表 2-13　パフォーマンス改善効果

2.3.2.2　Statspack 解析

Top 5 Timed Events

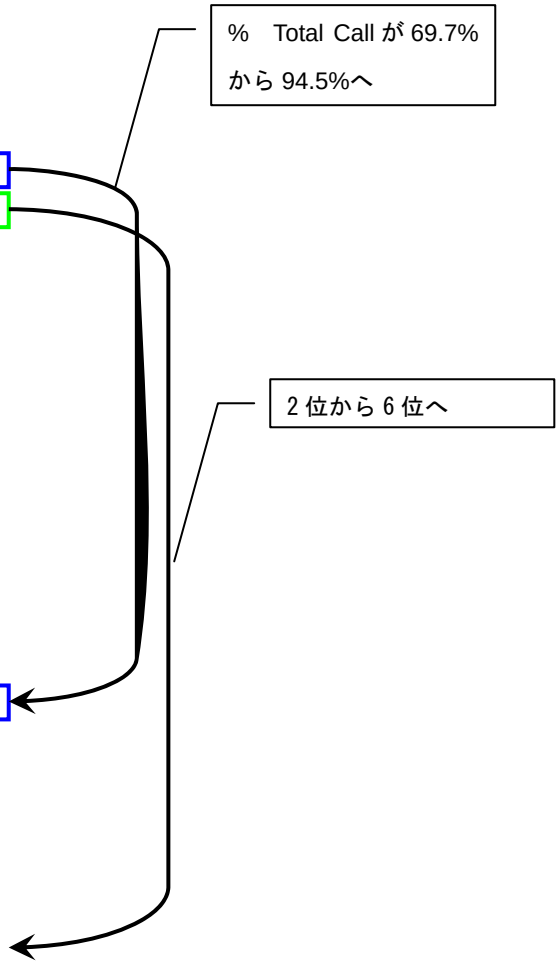
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
db file sequential read	1,406,511	145,234	103	69.7	User I/O
log file sync	441,556	49,098	111	23.6	Commit
CPU time		2,286		1.1	
log file parallel write	27,648	2,009	73	1.0	System I/O
enq: TX - index contention	42,929	1,738	40	.8	Concurrency

表 2-14　オブジェクト移動前

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
db file sequential read	1,598,863	127,627	80	94.5	User I/O
CPU time		1,990		1.5	
log file switch (checkpoint incomplete)	3,005	1,281	426	.9	Configuration
log file switch completion	3,022	1,117	370	.8	Configuration
db file parallel write	64,685	872	13	.6	System I/O

表 2-15　オブジェクト移動後



● % Total Call

移動前はデータアクセスの待機イベント、Db file sequential read、と log file sync の２つで全待機イベントの 93.3%を占めています。

移動後では Db file sequential read の順位は 1 位と変わりませんが、これだけで、待機イベントは 94.5%を占めるように変化しました。

log file sync は 2 位から 6 位になり、%Total Call の 23.6%から１%に下がっていますので、RamSan-400 に移動した効果が大きいことが分かります。

● トランザクション当たりの Wait

	Txn/min.	Log file switch		Log file sync		Db file sequential read	
		Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})
移動前	19,039.9	データ無	データ無し	1,444.1	75.84	4,271.6	224.3
移動後	31,744.6	データ無	データ無し	31.4	0.99	5,549.0	174.8
改善効果	1.67			76.59		1.28	

表 2-16　トランザクション当たりの Wait の比較

オブジェクトを移動する前のトランザクション辺りの Wait は、log file sync は 76.59 倍、Db file sequential read は 1.28 倍かかっていました。特に Top5 Events でも 2 位から 6 位に下がった log file sync に大きな効果があったことが分かります。

2.3.2.3　SwingBench のキャプチャー画面

SPARC Enterprise のようなトランザクションが停止するような現象は見られず安定した動作となりました。

トランザクション数/min を比較すると移動前は 19,039.9Txn/min、移動後は 31,744.6Txn/min と、RamSan-400 へオブジェクトを移動することで、約 1.7 倍の高いトランザクション数で安定することが分かります。

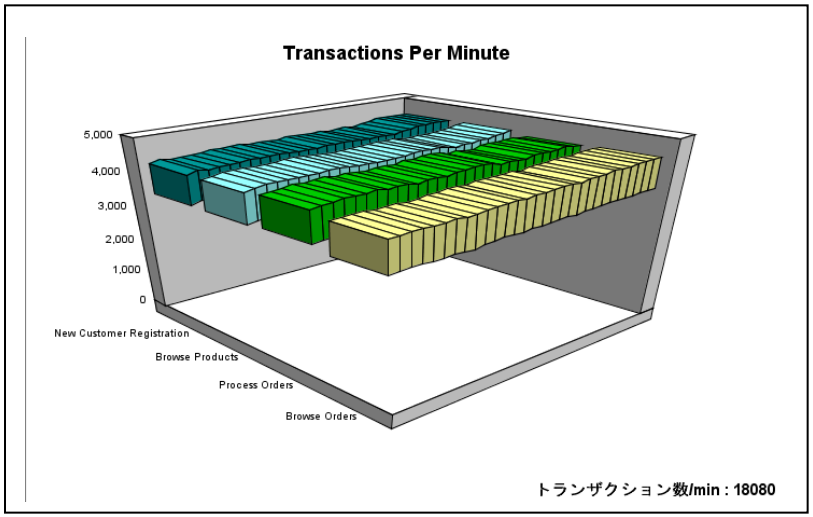


図 2-7　オブジェクト移動前

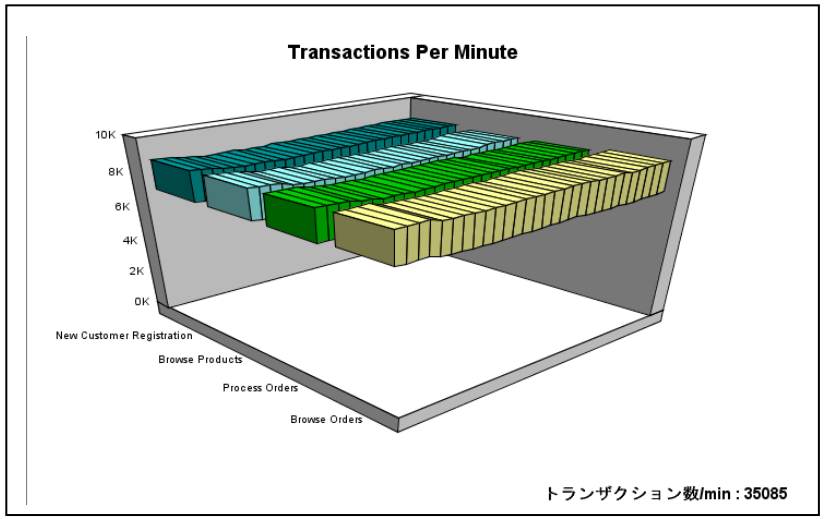
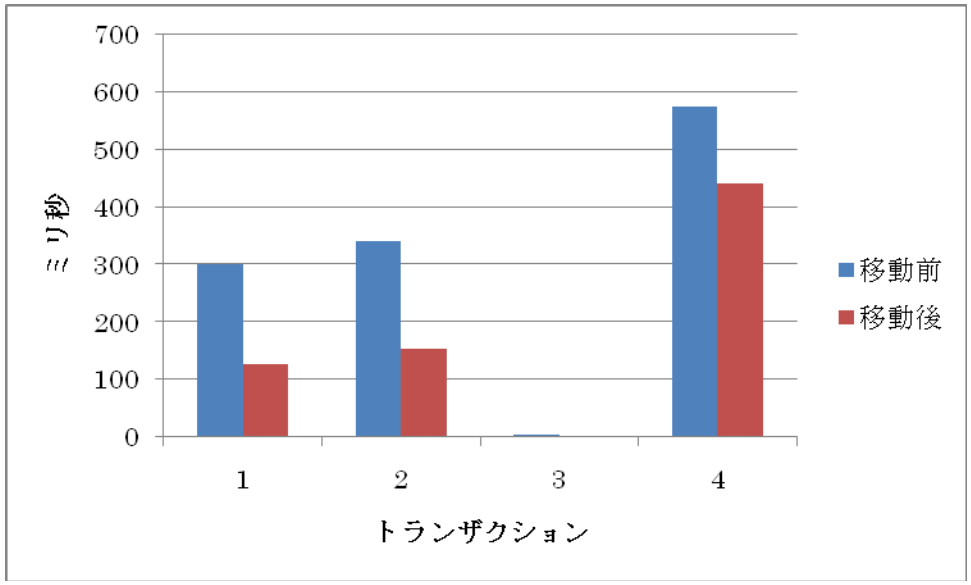


図 2-8　オブジェクト移動後

2.3.2.4　4 つのトランザクションの比較

4 つのトランザクションの応答時間は、オブジェクトを RamSan-400 に移動すると、平均的には 1.7 分の 1 に短縮しました。



① Customer Registration（顧客の作成）
② Browse Products（製品の表示）
③ Browse Orders（注文の確認）
④ Process Orders（発注）

図 2-9　各トランザクションの応答時間

2.3.2.5　OS ツールでのパフォーマンス測定

	CPU Idle	IOBusy	
		Disk Array	RamSan-400
オブジェクト移動前	85%	100%	―
オブジェクト移動後	65%	100%	45%

表 2-17　Performance Monitor によるリソースの使用状況

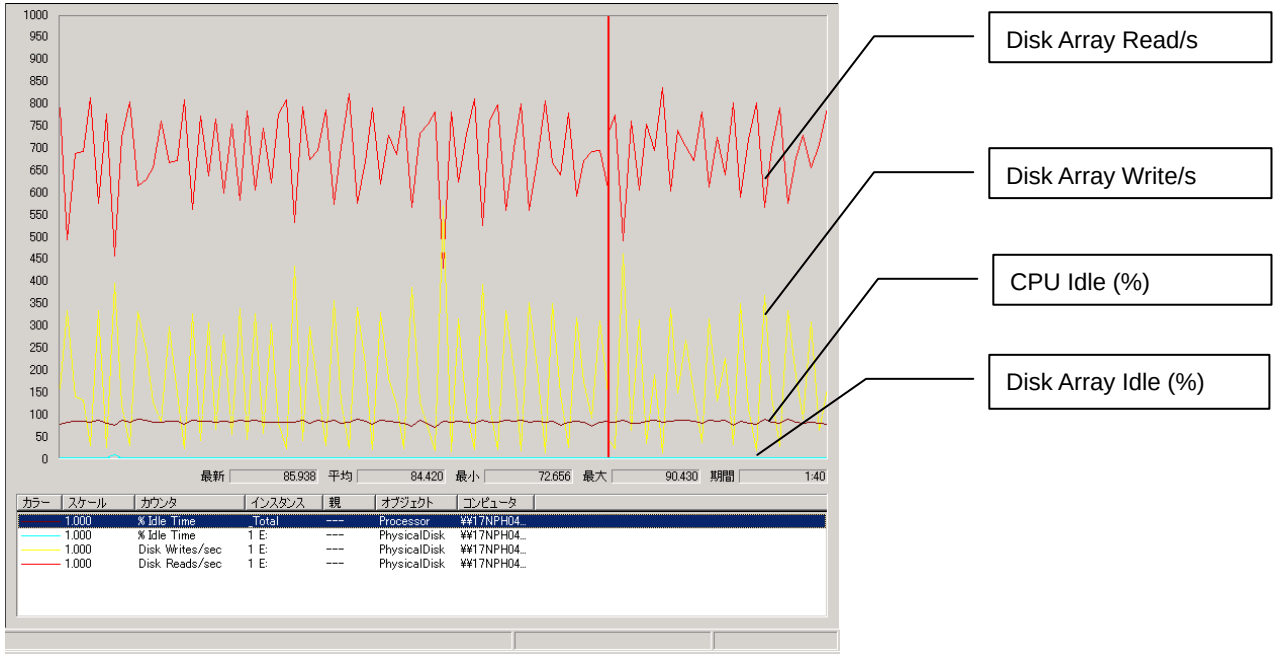


図 2-10　オブジェクト移動前（パフォーマンスメータ）

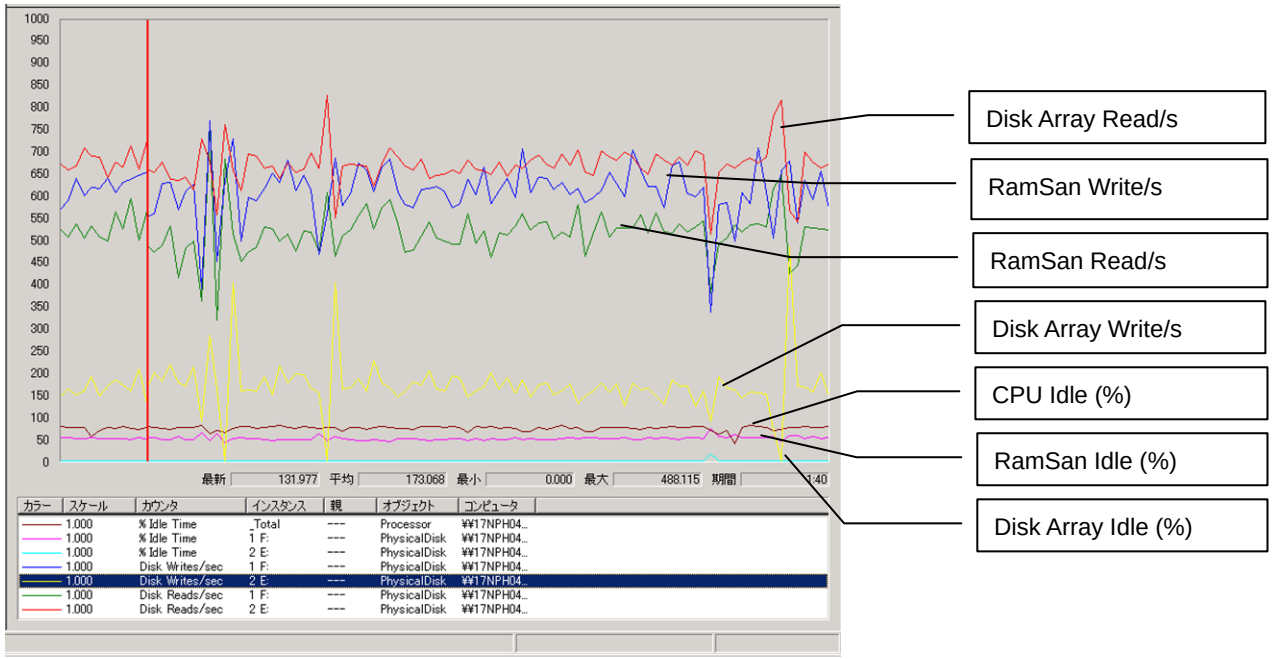


図 2-11　オブジェクト移動後（パフォーマンスメータ）

2.3.2.6　考察

このシステムではメモリサイズは2GBだったのでOracle 10g Databaseへは1GBを割り当てました。

そのため、ディスクのI/Oアクセスは常に100%となっていて、RamSan-400へファイルを移動したことによる効果が良く分かります。

トランザクション数は1.7倍になり、RamSan-400に移動されたRedoログファイルに対するイベントの内log file syncは76.59倍となりました。また、log file parallel writeは4倍の改善が見られていました。

移動前後でのDisk Arrayの使用率は100%で変わりませんでしたが、CPUのIdleは85%から65%と使用率は20ポイント上昇しました。

移動後のRamSan-400の平均使用率は45%でしたので、RamSan-400に相当のI/O負荷が分散されたことが分かります。

Swingbenchのキャプチャー画面を見ると、このシステムのトランザクション/minは他のプラットホームと比較して安定していることが分かります。

2.3.3 PRIMERGY RX600 S2（Red Hat Enterprise Linux AS V4U4）

2.3.3.1　パフォーマンス改善効果

オブジェクトの移動によって、1 時間あたりの総トランザクション数は 68 万から 96 万と 28 万の増加、1 トランザクションの平均応答時間は 534ms から 381.0ms と 153ms 短縮し、約 1.4 倍の改善効果が見られました。

比較項目	1 時間の総トランザクション数	平均応答時間(ms)
オブジェクト移動前	681,119	534.5
オブジェクト移動後	959,000	381.0
パフォーマンス改善効果	1.4 倍	1.4 倍

表 2-18　パフォーマンス改善効果

2.3.3.2　Statspack

Top 5 Timed Events

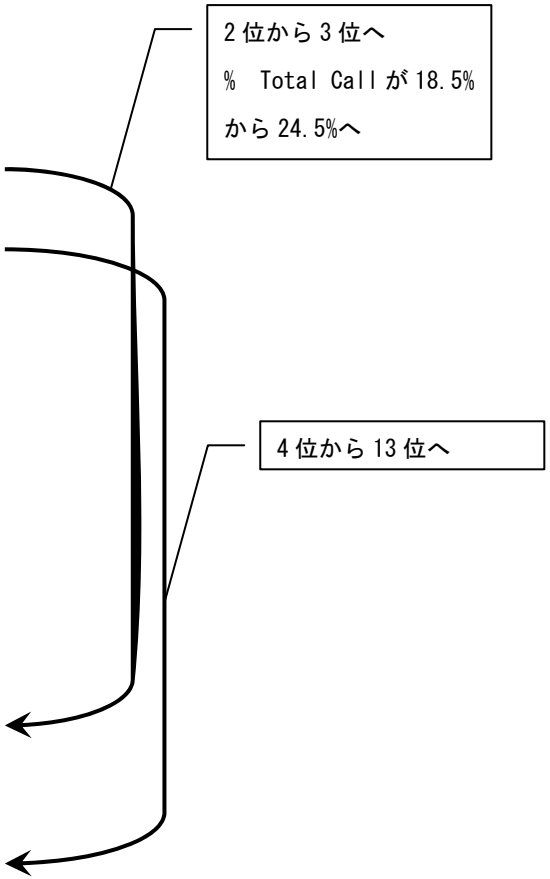
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log file switch (checkpoint incomplete)	72,676	68,198	938	39.2	Configuration
db file sequential read	644,346	32,246	50	18.5	User I/O
buffer busy waits	60,350	30,229	501	17.4	Concurrency
log file sync	223,713	22,223	99	12.8	Commit
enq: TX - index contention	15,377	10,739	698	6.2	Concurrency

表 2-19　オブジェクト移動前

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log file switch (checkpoint incomplete)	112,055	107,987	964	35.2	Configuration
enq: HW - contention	33,058	84,375	2,552	27.5	Configuration
db file sequential read	1,602,214	75,302	47	24.5	User I/O
buffer busy waits	46,002	24,841	540	8.1	Concurrency
enq: TX - index contention	8,899	5,189	583	1.7	Concurrency

表 2-20　オブジェクト移動後



- % Total Call

移動前後で、Log file switch の変化は見られませんでした。

オブジェクトの移動前は 4 位になっていた log file sync がランクから外されて 13 位に相当する順位に下がりました。トランザクションの増加によって enq: HW -contention が 2 位になりました。Db file sequential read は 2 位から 3 位に下がりましたがその比率は 18.5%から 27.5%を占めるようになり、I/O の傾向にかなり変化が見られます。

- トランザクション当たりの Wait

	Txn/min.	Log file switch		Log file sync		Db file sequential read	
		Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})	Wait/min.	Wait/txn(10^{-3})
移動前	10,985.8	2,199.9	200.2	716.9	65.25	1040.2	94.7
移動後	15,467.7	1,928.0	124.6	5.7	0.37	1344.7	86.9
改善効果	1.41	1.61		177.75		1.09	

表 2-21　トランザクション当たりの Wait の比較

オブジェクトを移動する前のトランザクション当たりの Wait は、log file sync は 177.75 倍掛かっていました。Db file sequential read はほとんど変化がありませんので、依然ここにボトルネックがあることが分かります。

2.3.3.3　SwingBench のキャプチャー画面

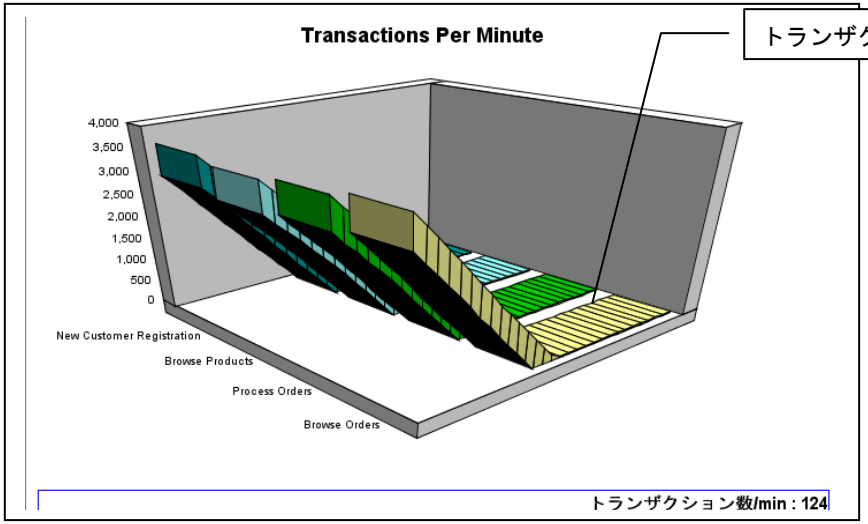


図 2-12　オブジェクト移動前

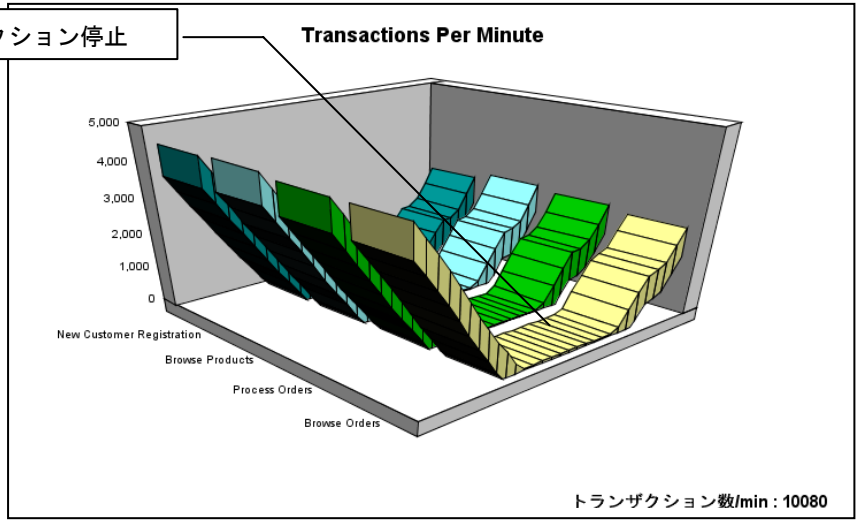
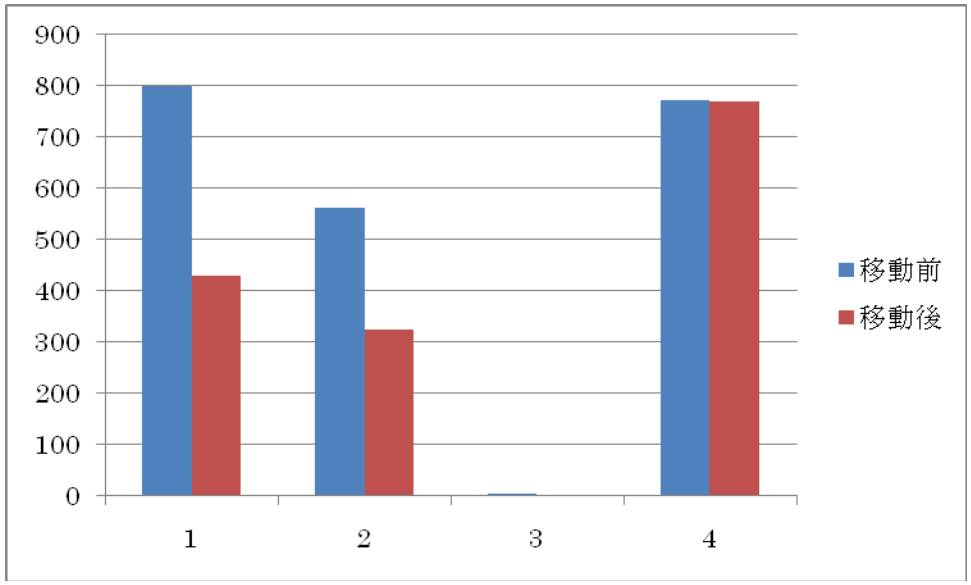


図 2-13　オブジェクト移動後

オブジェクトの移動前後でトランザクションが一時的に止まる状況が発生しました。ただし、オブジェクトを RamSan-400 へ移動した後では停止時間は短くなっています。

2.3.3.4　　4 つのトランザクションの比較

4 つのトランザクションの応答時間は、オブジェクトを RamSan-400 に移動すると、最大で 2 分の 1 に、平均的には 1.4 倍の 1 に短縮しました。



① Customer Registration（顧客の作成）
② Browse Products（製品の表示）
③ Browse Orders（注文の確認）
④ Process Orders（発注）

図 2-14　各トランザクションの応答時間

2.3.3.5　OS ツールでのパフォーマンス測定

ディスク構成	CPU Idle	IOBusy	
		Disk Array	RamSan-400
オブジェクト移動前	57.8%		
オブジェクト移動後	61.4%		

表 2-22　IOSTAT によるパフォーマンス比較

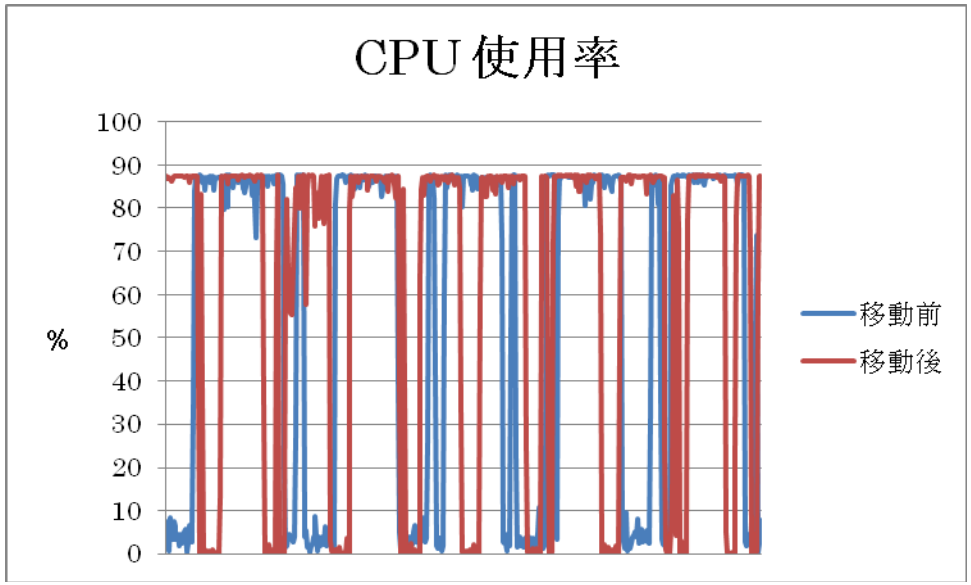


図 2-15　CPU 使用率の推移

IOSTAT では CPU の使用率の変動が激しかったので、参考データとなるが 3.6 ポイント使用率が向上しています。

2.3.3.6　考察

ハードウェアは同一であるがOSによってOracle10g Databaseのパフォーマンスに相当の違いが見られます。総トランザクション数で比較すると移動前では、Windowsが1,180,471に対してLinuxは681,119と60%、移動後ではWindowsが1,968,168に対してLinuxは959,000と移動後では50%となっています。

2GB（うち、1GBがOracle10g Databaseへ割当て）のメモリサイズの場合、スワップの発生やテンポラリファイルへのアクセスが他のプラットホームに比べて頻繁に行なわれるので、この結果はWindowsとLinuxのメモリ管理の違いが顕著に出たものと推測されます。

従って、メモリを増設することよって、違った結果が得られたと考えられます。

2.3.4 PRIMEQUEST 520（Red Hat Enterprise Linux AS V4U4）

2.3.4.1　パフォーマンス改善効果

オブジェクトの移動によって、1 時間あたりの総トランザクション数は 123 万から 298 万と 175 万の増加、1 トランザクションの平均応答時間は 292.3ms から 122.0ms と 170ms 短縮し、約 2.4 倍の改善効果が見られました。

総トランザクション数および改善効果は今回の検証の中では最も良い値を得ることができました。

	一時間の総トランザクション数	平均応答時間(ミリ秒)
オブジェクト移動前	1,230,395	292.3
オブジェクト移動後	2,984,856	122.0
パフォーマンス改善効果	2.4 倍	2.4 倍

表 2-23　パフォーマンス改善効果

2.3.4.2　Statspack 解析

Top 5 Timed Events

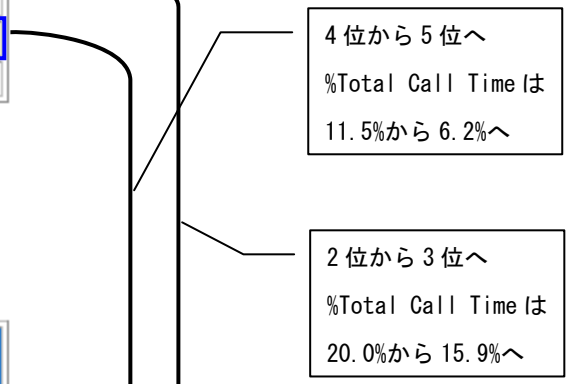
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log file switch (checkpoint incomplete)	82,943	74,906	903	41.8	Configuration
db file sequential read	629,531	35,764	57	20.0	User I/O
buffer busy waits	67,561	28,596	423	16.0	Concurrency
log file sync	401,654	20,993	52	11.7	Commit
enq: HW - contention	3,598	9,193	2,555	5.1	Configuration

表 2-24　オブジェクト移動前

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
log file switch (checkpoint incomplete)	57,053	50,367	883	44.6	Configuration
buffer busy waits	192,515	18,835	98	16.7	Concurrency
db file sequential read	867,174	17,945	21	15.9	User I/O
enq: HW - contention	4,119	7,372	1,790	6.5	Configuration
log file sync	841,055	6,950	8	6.2	Commit

表 2-25　オブジェクト移動後



- % Total Call

Top5 Timed Events の log file sync は 4 位(11,7%)から 5 位(6.2%)、Db file sequential read は 2 位 (20%) から 3 位 (15.9%) になりました。

Top5 Timed Events 内の順位に大きな変化はありませんでしたので、ここでは、File IO Stats に注目することにします。

データファイルの平均※Reads/sec は 350 から 665、Writes/sec は 203 から 464 に上がりました。Undotbs01 の平均 Writes/s は 82 から 165 に増えました。

RamSan-400 に置いたデータファイル (SCOTT_SSD) で増加した分だけでなく、SCOTT_DATA でも増加しているので移動効果があったことがわかります。

※移動前は SCOTT_DATA＋SCOTT_INDEX、移動後は SCOTT_DATA＋SCOTT_INDEX+SCOTT_SSD

File IO Stats

- ordered by Tablespace, File

Tablespace	Filename	Reads	Av Reads/s	Av Rd(ms)	Av Blks/Rd	Writes	Av Writes/s	Buffer Waits	Av Buf Wt(ms)
SCOTT_DATA	/u02/etfc/pqdb1/SCOTT_DATA.DBF	388,404	217	58.35	1.00	89,768	50	30,401	870.78
SCOTT_INDEX	/u02/etfc/pqdb1/SCOTT_INDEX.DBF	237,929	133	54.91	1.00	273,772	153	27,810	60.63
SYSAUX	/u02/etfc/pqdb1/sysaux01.dbf	1,454	1	3.87	1.00	2,252	1	1	20.00
SYSTEM	/u02/etfc/pqdb1/system01.dbf	1,453	1	8.36	1.00	190	0	0	0.00
UNDOTBS1	/u02/etfc/pqdb1/undotbs01.dbf	28	0	0.71	1.00	147,383	82	9,338	120.36
USERS	/u02/etfc/pqdb1/users01.dbf	19	0	0.00	1.00	19	0	0	0.00

表 2-26　オブジェクト移動前

File IO Stats

- ordered by Tablespace, File

Tablespace	Filename	Reads	Av Reads/s	Av Rd(ms)	Av Blks/Rd	Writes	Av Writes/s	Buffer Waits	Av Buf Wt(ms)
SCOTT_DATA	/u02/etfc/pqdb1/SCOTT_DATA.DBF	444,523	343	28.46	1.00	183,991	142	30,542	480.22
SCOTT_INDEX	/u02/etfc/pqdb1/SCOTT_INDEX.DBF	133,158	103	38.19	1.00	203,038	156	539	45.94
SCOTT_SSD	/u02/rsfc/pqdb1/SCOTT_SSD.DBF	284,288	219	0.68	1.00	214,813	166	136,075	28.94
SYSAUX	/u02/etfc/pqdb1/sysaux01.dbf	5,578	4	4.29	1.16	2,366	2	0	0.00
SYSTEM	/u02/etfc/pqdb1/system01.dbf	951	1	7.34	1.03	154	0	0	0.00
UNDOTBS1	/u02/etfc/pqdb1/undotbs01.dbf	34	0	0.00	1.00	213,732	165	25,683	25.80
USERS	/u02/etfc/pqdb1/users01.dbf	34	0	0.00	1.00	34	0	0	0.00

表 2-27　オブジェクト移動後

- トランザクション当たりの Wait

	Txn/min.	Log file switch		Log file sync		Db file sequential read	
		Wait/min.	Wait/txn（10^{-3}）	Wait/min.	Wait/txn（10^{-3}）	Wait/min.	Wait/txn（10^{-3}）
移動前	19,845.1	2,583.0	130.2	723.9	36.48	1,233.2	62.1
移動後	48,142.8	2,398.4	49.8	331.0	6.87	854.5	17.7
改善効果	2.43	2.61		5.31		3.50	

表 2-28　トランザクション当たりの Wait の比較

オブジェクトを移動する前のトランザクション当たりの Wait は、log file sync は 5.31 倍掛かっていました。Db file sequential read は 3.50 倍掛かっていました。32GB と潤沢なメモリを積んでいる分、他のプラットホームと比べると、log file sync の効果は低くなっているようですが、それでもトランザクション数は 2.4 倍の効果が得られています。

2.3.4.3　SwingBench のキャプチャー画面

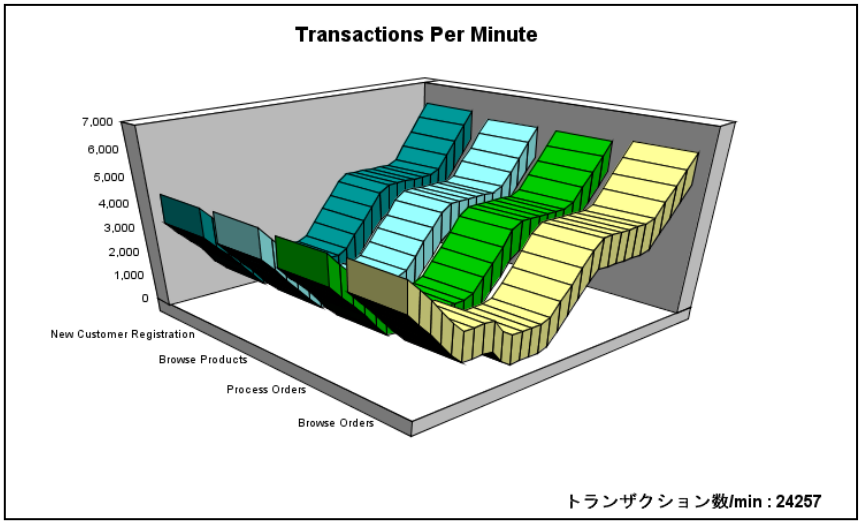


図 2-16　オブジェクト移動前

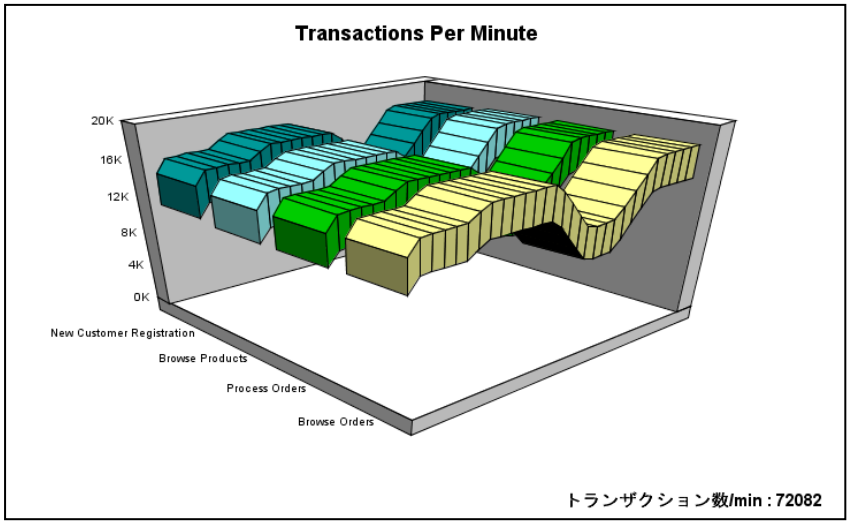
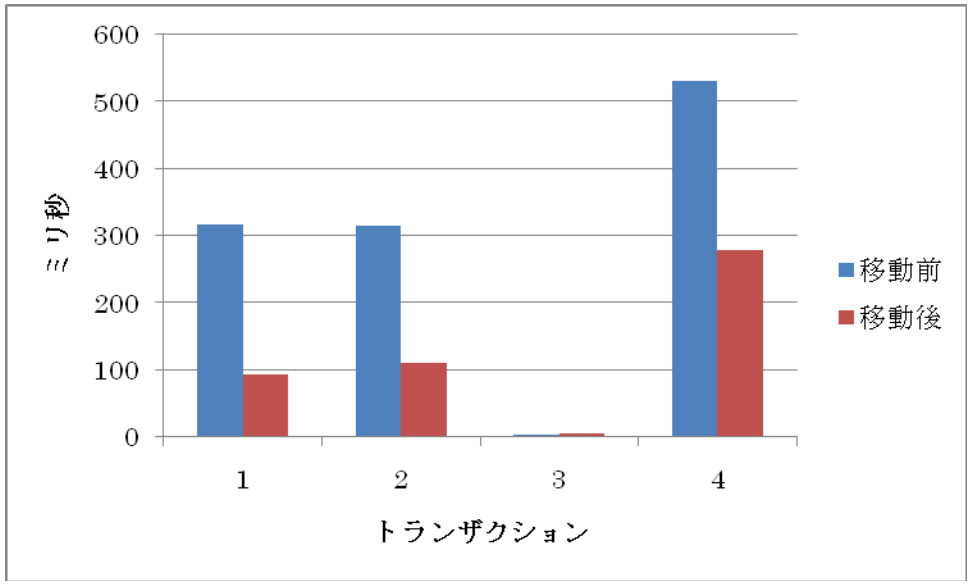


図 2-17　オブジェクト移動前

オブジェクトの移動前後でトランザクションの変動は安定するようになっています。これは、ログファイルの処理の影響が少なくなっているものと思われます。

2.3.4.4　4 つのトランザクションの比較

4 つのトランザクションの応答時間は、オブジェクトを RamSan-400 に移動すると、最大で 3 分の 1 に、平均的には 2 分の 1 以上短縮しました。



① Customer Registration（顧客の作成）
② Browse Products（製品の表示）
③ Browse Orders（注文の確認）
④ Process Orders（発注）

図 2-18　各トランザクションの応答時間

2.3.4.5　OS ツールでのパフォーマンス測定

ディスク構成	CPU Idle	IOBusy	
		Disk Array	RamSan-400
オブジェクト移動前	57.5%		
オブジェクト移動後	49.8%		

表 2-29　IOSTAT によるパフォーマンス比較

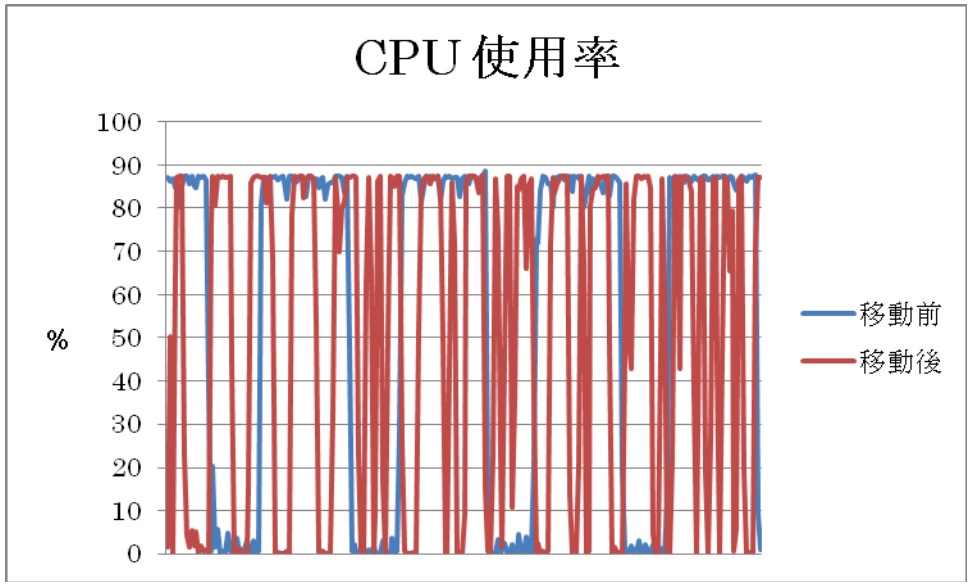


図 2-19　CPU 使用率の推移

IOSTAT では CPU の使用率の変動が激しかったので、参考データとなるが 7.7 ポイント使用率が向上しています。

2.3.4.6　考察

このシステムでは、メモリサイズが 32GB と潤沢に積んでおり、移動前の CPU の使用率も 42.5%とリソースの使用率も良いといえますが、RamSan-400 にオブジェクトを移動することでトランザクション数が今回の検証の中では最も多くなり、また改善効果も 2.4 倍と最大でした。

つまり、このシステムではこれ以上 CPU やメモリを増やしても効果はほとんどなく、I/O の改善が必須だったことが分かります。

log file sync 以外にも Redo ログファイルに関する待機イベントの改善はかなり見られ、log file parallel write は 5 倍速くなっており、これも、全体のパフォーマンス向上に繋がったと考えられます。

2.4 Oracle 10g Database ベンチマークまとめ

2.4.1 結果一覧

改善効果	CPU 数 メモリ	OS	移動前 TXN 移動後 TXN	評価
SPARC Enterprise M4000	4CPU 8Core 16 ｽﾚｯﾄﾞ 8GB	Solaris 10 11/06	745K 1,525K	● 移動前後でトランザクション数は 780K 増加し 2 倍の改善効果が見られました。 ● log file Switch は 3.8 倍、log file sync は 137 倍、Db fileSeqRD は 1.3 倍の改善が見られました。 ● 移動前の(I/O 待ちによる)トランザクションの停止がなくなりました。 ● CPU の平均使用率は 3.3%から 6.3%に上がりましたが、DiskArray の使用率は 100%近くで RamSan-400 の使用率は 10%と、未だ、改善の余地があります。
PRIMERGY RX600 S2	8CPU 2GB	Windows Server 2003R2 EE	1,180K 1,968K	● 移動前後でトランザクション数は 788K 増加し 1.7 倍の改善効果が見られました。 ● log file sync は 77 倍、Db fileSeqRD は 1.3 倍の改善が見られました。 ● トランザクションの停止は移動前後で見られませんでした。 ● CPU の平均使用率は 15%から 35%と 20 ポイント上がりました。DiskArray の使用率は 100%近く、RamSan-400 の使用率は 45%と I/O が分散されています。
		RHEL AS V4U4	681K 959K	● 移動前後でトランザクション数は 278K 増加し 1.4 倍の改善効果が見られました。 ● log file Switch は 1.6 倍、log file sync は 178 倍、Db fileSeqRD は 1.8 倍の改善が見られました。 ● トランザクションの停止は移動前後で見られましたが、移動後の方が停止時間は短くなっています。 ● CPU の平均使用率は 42%から 39%と 3 ポイント上がりました。
PRIMEQUEST 520	8CPU 32GB	Win2003	未検証	——
		RHEL AS V4U4	1,230K 2,985K	● 移動前後でトランザクション数は 1,755K 増加し 2.4 倍とこの検証では最大の改善効果が見られました。 ● log file Switch は 2.6 倍、log file sync は 5.3 倍、Db fileSeqRD は 3.5 倍の改善が見られました。 ● トランザクションの変動は移動後の方が少なくなり安定しています。 ● CPU の平均使用率は 43%から 50%と 7 ポイント上がりました。

表 2-30　結果一覧

2.4.2 考察

この検証ではDisk ArrayをBUSYは100%の状態にしてベンチマークを実施しました。

そのため、RamSan-400にオブジェクトを移動させても、改善効果は1.4～2.4倍と意外と低いものに見えるようですが、これは、Disk ArrayのBUSY率は100%のままで、ここが依然システムのボトルネックなってしまった結果だと考えられます。

しかも、今回は、リハーサル（予備テスト）時にStatsPackで判明したWaitの上位を占めた13GBのオブジェクト（Redoログ、Undoファイル、索引の一部）をRamSan-400に移動したに過ぎず、この作業に費やした時間は1時間にも満たない※ものでした。

こういった簡単な作業にもかかわらず、直ぐに2倍のパフォーマンスアップが実現できるということは、RamSan-400が非常に即効性の高いソリューションであるといえ、事実、log file Syncを見れば最高で178倍というI/Oの向上が図られていることからも、そのことが証明されています。

Oracle10g Databaseではその構造上、幾らメモリやCPUを追加したとしても、高い頻度のI/Oアクセスの発生が避けられないRedoログに代表されるファイルが幾つか存在します。

非公式なデータとはなりますが、30GB程度のデータベースを全てRamSan-400上に置いた場合は、12.7倍という改善効果が見られています。また、今回、検証したプラットホームの中でハードウェアリソースが8CPU（使用率は移動前で42.5%）、メモリが32GBと優れていたPRIMEQUEST520が、移動前後のトランザクション数が最も多く、かつ、改善効果が2.4倍と最も良かったことから見ても、ボトルネックとなっているファイルやオブジェクトを見つけ出し、より適切なものをRamSan-400に移動すれば、今回の検証結果よりも更にパフォーマンスが改善し、サーバリソースも十分に活用されることようになることは容易に想像できます。

システムのI/Oボトルネックを解消する唯一のソリューション、それが半導体ディスクRamSan-400なのです。

※テスト時間とファイルの実移動時間は除く

3　インターオペラビリティ

表 3-1 の通り今回検証したプラットフォームにおいて接続性は問題ありませんでした。

プラットホーム	OS	HBA	HBA Driver	ディスクの認識		パーティション作成		ファイルシステム作成	
				コマンド	結果	コマンド	結果	コマンド	結果
SPARC Enterprise M4000	Solaris 10 11/06	4Gbps SE0X7F11F×2	FJSVpfcaV40L00	dmesg Iostat –E	○	format	○	newfs	○
PRIMERGY RX600 S2	Windows Server 2003 R2EE	2Gbps Emulex LP1050	7-1.10A4 x64 lpfc_2.6_ioctl_module-2.0.15-1	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬｰ)	○	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞｨｽｸ管理)	○	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞｨｽｸ管理)	○
	RHEL AS V4U4		6-1.11A3 64bit IA64	dmesg cat /proc/scsi/scsi	○	fdisk	○	mkfs –ext3	○
PRIMEQUEST 520	Windows Server 2003 R2EE	4Gbps Emulex LPe11002-M4	FJSVpfcaV40L00	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞﾊﾞｲｽﾏﾈｰｼﾞｬｰ)	○	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞｨｽｸ管理)	○	ｺﾝﾋﾟｭｰﾀの管理 (ﾃﾞｨｽｸ管理)	○
	RHEL AS V4U4		7-1.10A4 x64	dmesg cat /proc/scsi/scsi	○	fdisk	○	mkfs –ext3	○

表 3-1 RamSan-400　インターオペラビリティ結果

3.1　SPARC Enterprise M4000（Solaris10）

3.1.1 ディスクの認識

3.1.1.1　dmesg

```
m4000r2 console login: /pci@1,700000/fibre-channel@0 (fjpfca0):
        INFO : AL link up (private loop) . alpa=0x1, 4Gbps.
/pci@1,700000/fibre-channel@0 (fjpfca0):
        found target. target_id=0x0 port_id=0xef wwn=210000e000a81caa
/pci@0,600000/pci@0/pci@9/fibre-channel@0 (fjpfca1):
        INFO : NPORT link up (connected point-to-point. id=0x2, 4Gbps.)
/pci@0,600000/pci@0/pci@9/fibre-channel@0 (fjpfca1):
        found target. target_id=0x0 port_id=0x1 wwn=21010020c2054293
```

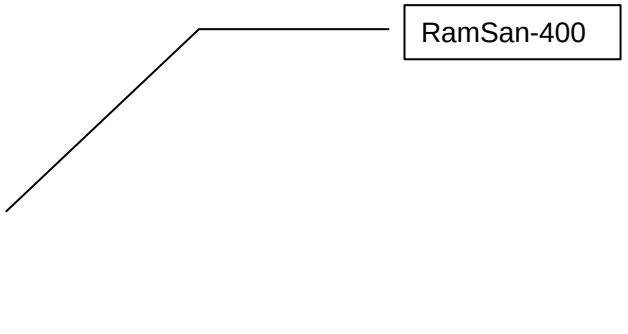


図 3-1　dmesg での RamSan の認識の確認

3.1.1.2　iostat-E

```
sd21      Soft Errors: 2 Hard Errors: 0 Transport Errors: 0
Vendor: FUJITSU  Product: E400A           Revision: 0000 Serial No:
Size: 209.71GB <209706811392 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 2 Predictive Failure Analysis: 0
```

```
sd23      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: TMS      Product: FC65            Revision: 2.7n Serial No:
Size: 34.36GB <34359738368 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
```

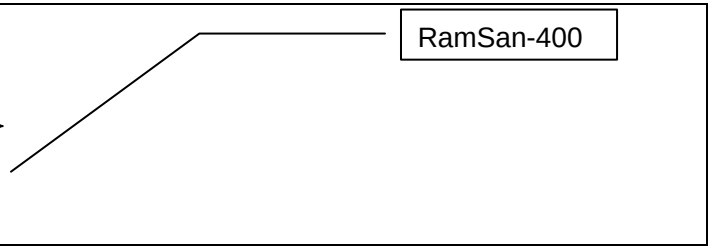


図 3-2　iostat コマンドの結果

3.1.2 パーティションの作成

```
bash-3.00# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>  00200000
          /pci@0,600000/pci@0/pci@8/pci@0/scsi@1/sd@0,0
       1. c0t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>  00200002
          /pci@0,600000/pci@0/pci@8/pci@0/scsi@1/sd@1,0
       2. c1t0d0 <FUJITSU-E400A-0000 cyl 24998 alt 2 hd 64 sec 256>
          /pci@1,700000/fibre-channel@0/sd@0,0
       3. c2t0d0 <TMS-FC65-2.7n cyl 16381 alt 2 hd 128 sec 32>
          /pci@0,600000/pci@0/pci@9/fibre-channel@0/sd@0,0
```

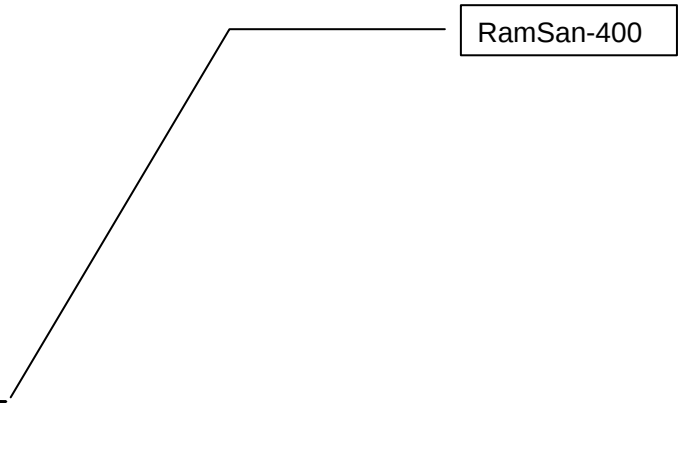


図 3-3　foramt コマンドの結果

3.1.3 ファイルシステムの作成

```
bash-3.00# newfs /dev/rdsk/c2t0d0s2
newfs: 新しいファイルシステム /dev/rdsk/c2t0d0s2 を作成しますか: (y/n)? y
警告: 最終シリンダで 2048 セクタが割り当てられません。
/dev/rdsk/c2t0d0s2:      全セクター数: 67096576 (シリンダ数: 10921、トラック数:
48、セクター数: 128)
        32762.0MB、683 シリンダグループ (16 c/g, 48.00MB/g, 5824 i/g)
スーパーブロックのバックアップの位置 (fsck -F ufs -o b=# のため) :
 32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
シリンダグループの初期化:
............
スーパーブロックのバックアップの位置 (最後の 10 シリンダグループのため):
 66162848, 66261280, 66359712, 66458144, 66556576, 66655008, 66753440,
 66851872, 66950304, 67048736
```

図 3-4　newfs コマンドの実行

3.2　PRIMERGY RX600-S2（Windows Server 2003）

3.2.1 ディスクの認識

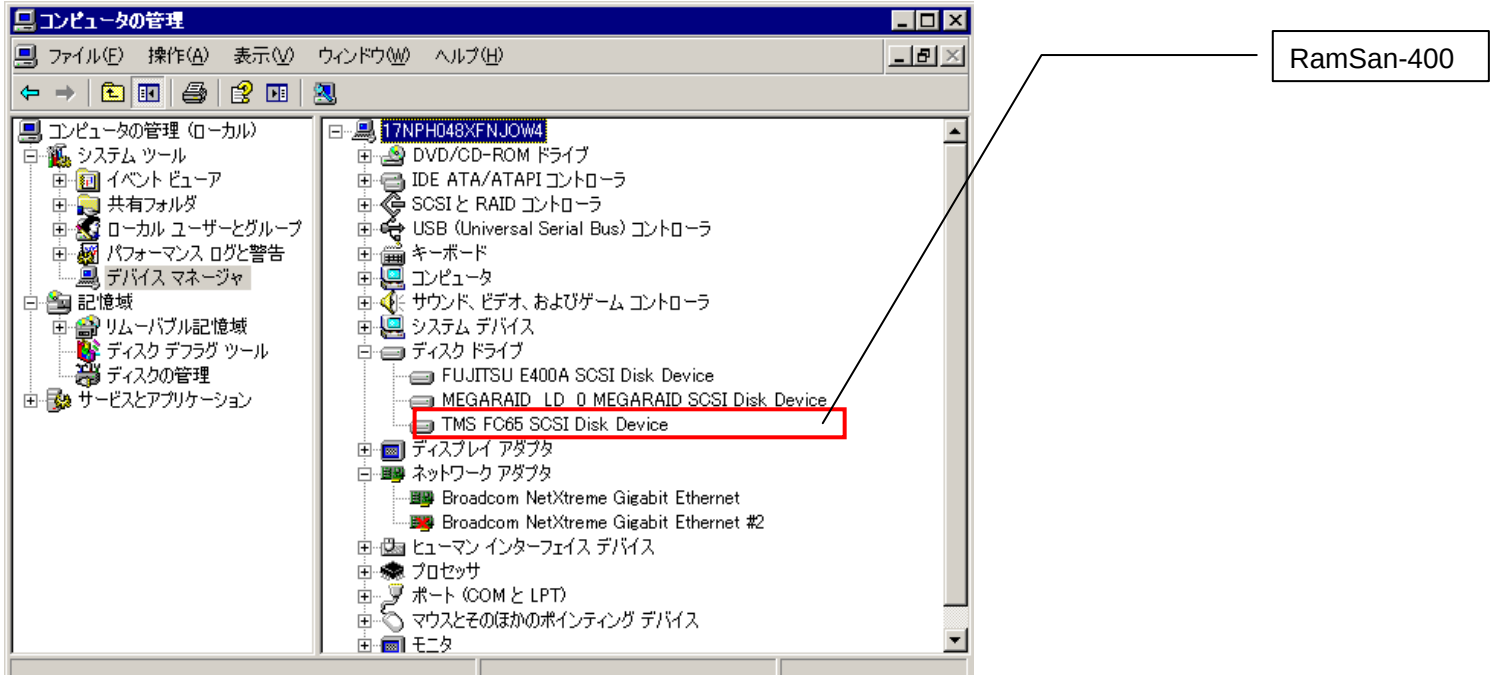


図 3-5　デバイスマネージャー

3.2.2 パーティションの作成とファイルシステムの作成

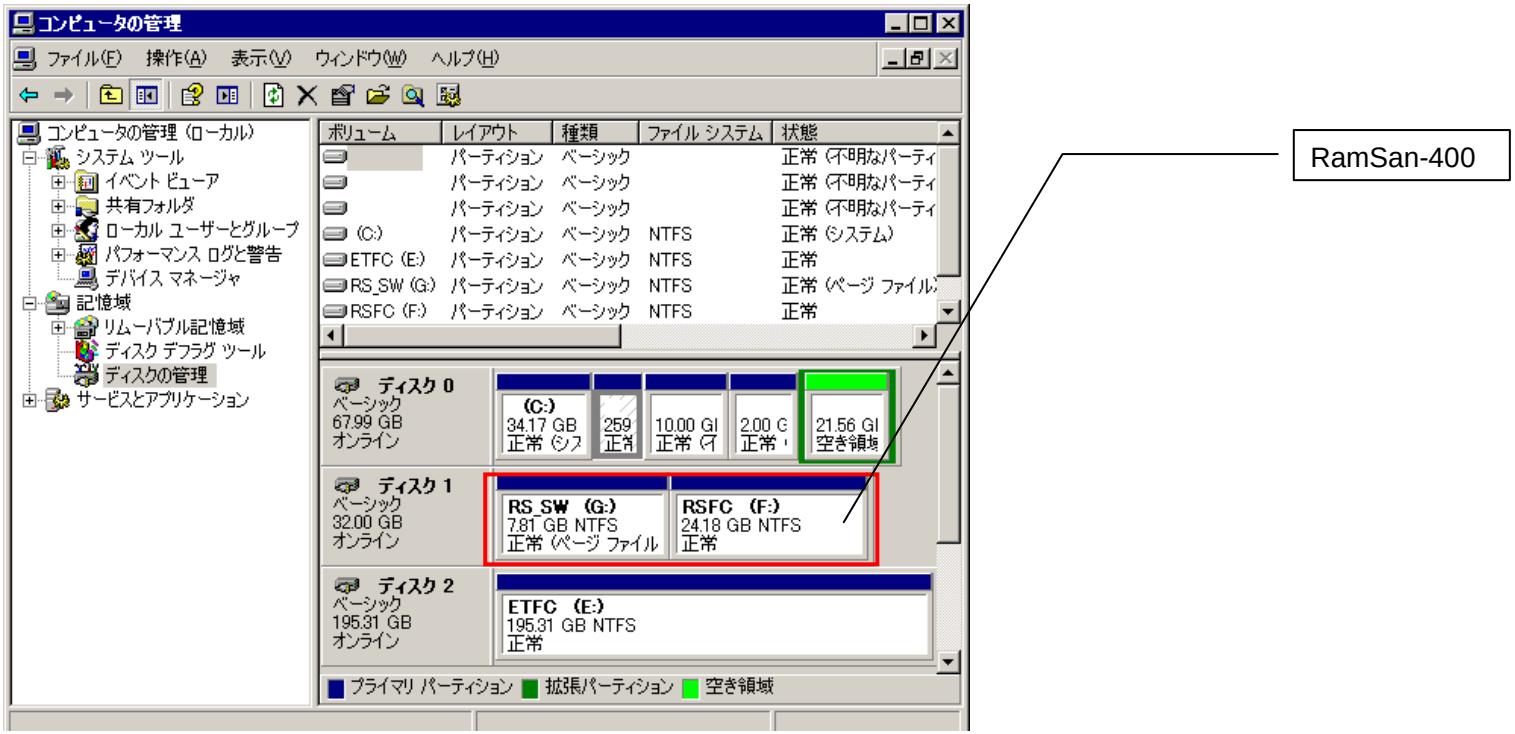


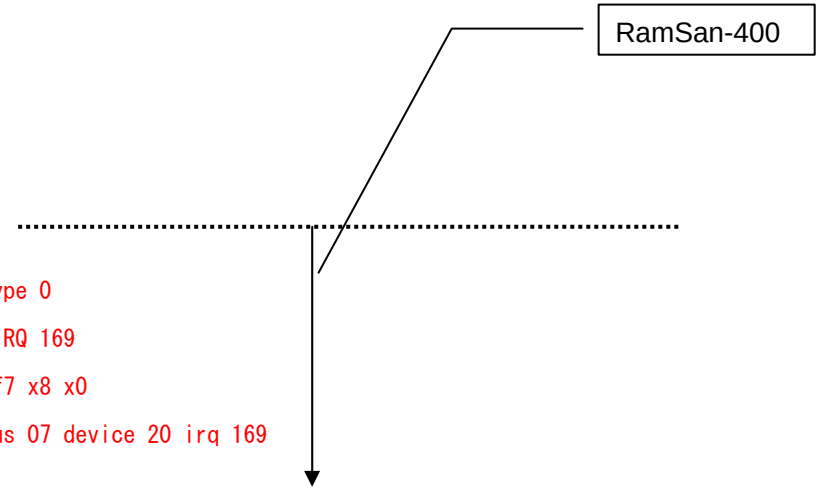
図 3-6 ディスクの管理

3.3　PRIMERGY RX600-S2（Red Hat Enterprise Linux AS V4U4）

3.3.1 ディスクの認識

3.3.1.1　dmesg

```
lpfc 0000:02:04.0: 0:1303 Link Up Event x1 received Data: x1 x1 x8 x2
scsi2 : Emulex LP1050 2Gb PCI-X Fibre Channel Adapter on PCI bus 02 device 20 irq 169
  Vendor: FUJITSU   Model: E400A            Rev: 0000
  Type:   Direct-Access                     ANSI SCSI revision: 03
SCSI device sdb: 409600000 512-byte hdwr sectors (209715 MB)
SCSI device sdb: drive cache: write back
SCSI device sdb: 409600000 512-byte hdwr sectors (209715 MB)
SCSI device sdb: drive cache: write back
 sdb: unknown partition table
Attached scsi disk sdb at scsi2, channel 0, id 0, lun 0
Attached scsi generic sg3 at scsi2, channel 0, id 0, lun 0,  type 0
ACPI: PCI interrupt 0000:07:04.0[A] -> GSI 16 (level, low) -> IRQ 169
lpfc 0000:07:04.0: 1:1303 Link Up Event x1 received Data: x1 xf7 x8 x0
scsi3 : Emulex LP1050 2Gb PCI-X Fibre Channel Adapter on PCI bus 07 device 20 irq 169
  Vendor: TMS       Model: FC65             Rev: 2.7n
  Type:   Direct-Access                     ANSI SCSI revision: 04
SCSI device sdc: 67108864 512-byte hdwr sectors (34360 MB)
```



```
SCSI device sdc: drive cache: write back
SCSI device sdc: 67108864 512-byte hdwr sectors (34360 MB)
SCSI device sdc: drive cache: write back
 sdc: unknown partition table
Attached scsi disk sdc at scsi3, channel 0, id 0, lun 0
Attached scsi generic sg4 at scsi3, channel 0, id 0, lun 0,  type 0
```

図 3-7　dmesg での RamSan の認識の確認

3.3.1.2　cat /proc/scsi/scsi

```
[root@localhost ~]# cat /proc/scsi/scsi
Attached devices:
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor: FUJITSU  Model: E400A            Rev: 0000
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi3 Channel: 00 Id: 00 Lun: 00
  Vendor: TMS      Model: FC65             Rev: 2.7n
  Type:   Direct-Access                    ANSI SCSI revision: 04
```

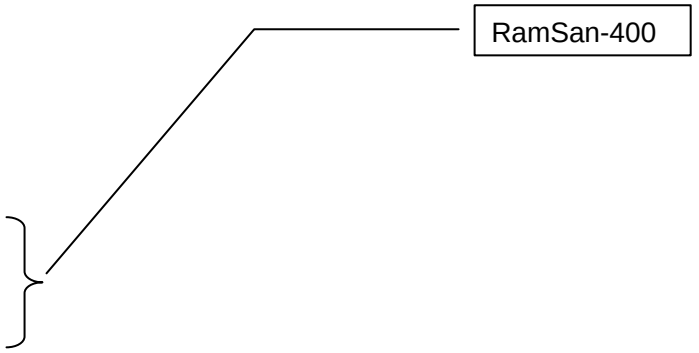


図 3-8　/proc/scsi/scsi の結果

3.3.2 パーティションの作成

```
[root@localhost ~]# fdisk /dev/sdc
Command (m for help): p
Disk /dev/sdc: 34.3 GB, 34359738368 bytes
64 heads, 32 sectors/track, 32768 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1       32768    33554416   83  Linux
```

図 3-9　fdisk コマンドの実行

3.3.3 ファイルシステムの作成

```
[root@localhost ~]# mkfs -t ext3 /dev/sdc1
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
4194304 inodes, 8388604 blocks
419430 blocks (5.00%) reserved for the super user
```

```
First data block=0
Maximum filesystem blocks=4294967296
256 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 36 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override
```

図 3-10　mkfs コマンドの実行

3.4　PRIMEQUEST 520（Windows Server 2003）

3.4.1 ディスクの認識

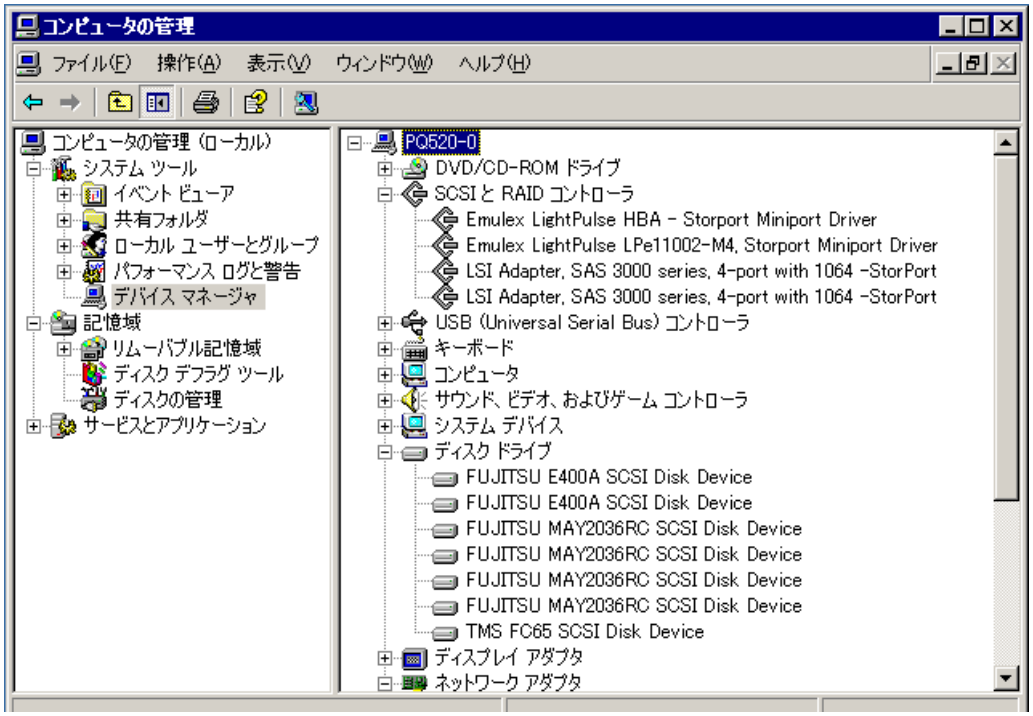


図 3-11　デバイスマネージャー

3.4.2 パーティションの作成とファイルシステムの作成

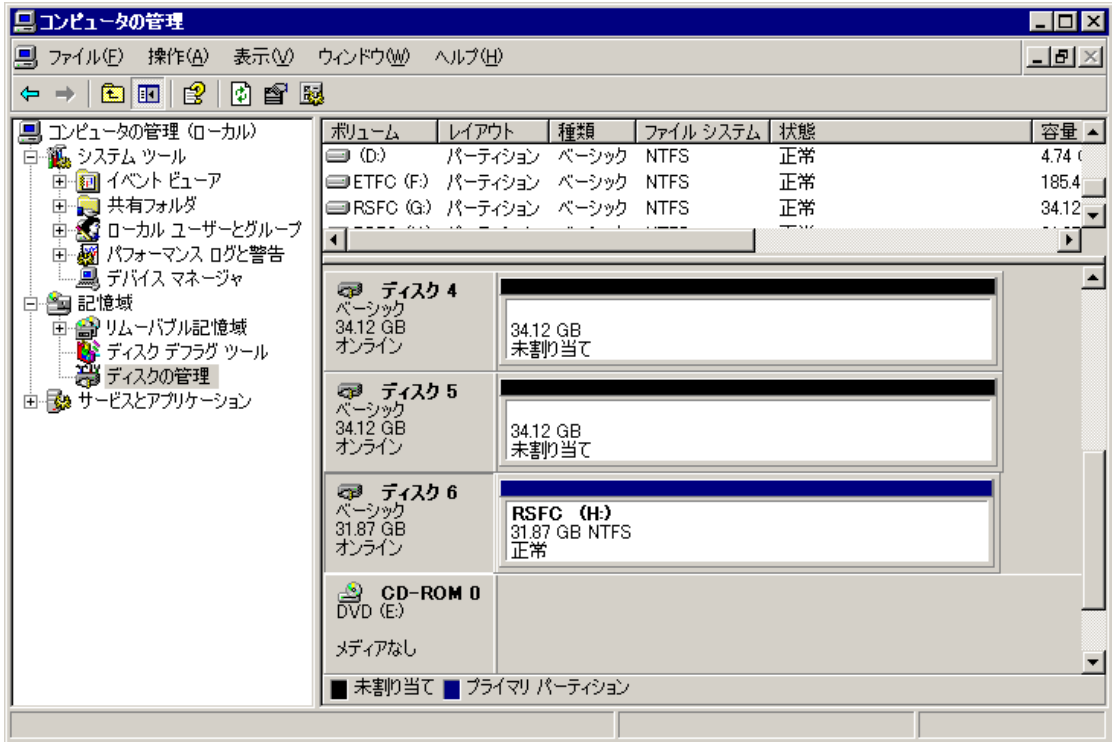


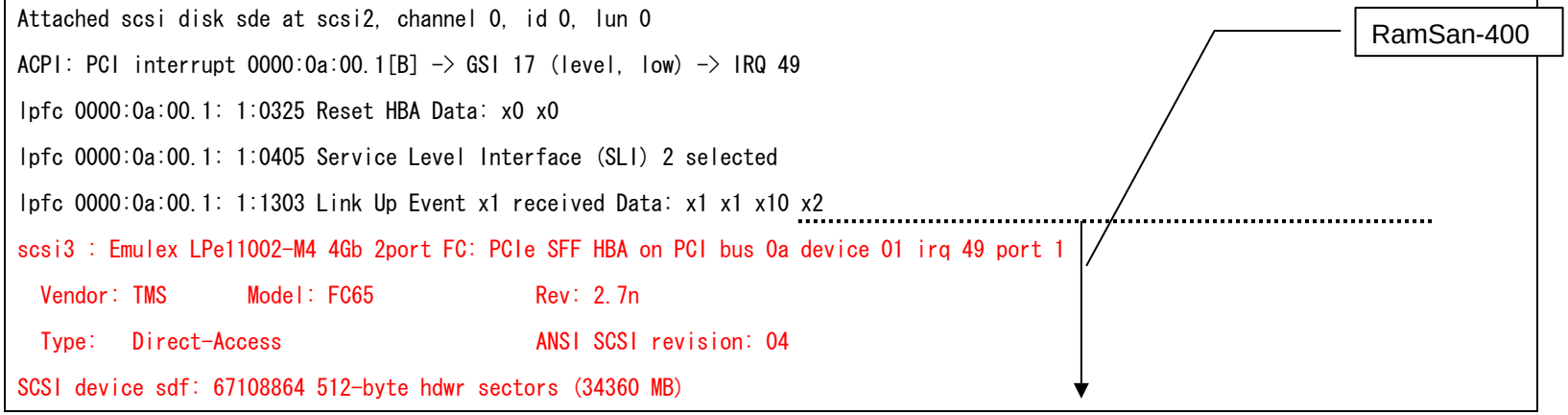
図 3-12　ディスクの管理

3.5　PRIMEQUEST 520（Red Hat Enterprise Linux AS V4U4）

3.5.1 ディスクの認識

3.5.1.1　dmesg

```
scsi2 : Emulex LPe11002-M4 4Gb 2port FC: PCIe SFF HBA on PCI bus 0a device 00 irq 48 port 0
  Vendor: FUJITSU   Model: E400A             Rev: 0000
  Type:   Direct-Access                      ANSI SCSI revision: 03
SCSI device sde: 409600000 512-byte hdwr sectors (209715 MB)
SCSI device sde: drive cache: write back
SCSI device sde: 409600000 512-byte hdwr sectors (209715 MB)
SCSI device sde: drive cache: write back
 sde: sde1
Attached scsi disk sde at scsi2, channel 0, id 0, lun 0
ACPI: PCI interrupt 0000:0a:00.1[B] -> GSI 17 (level, low) -> IRQ 49
lpfc 0000:0a:00.1: 1:0325 Reset HBA Data: x0 x0
lpfc 0000:0a:00.1: 1:0405 Service Level Interface (SLI) 2 selected
lpfc 0000:0a:00.1: 1:1303 Link Up Event x1 received Data: x1 x1 x10 x2
scsi3 : Emulex LPe11002-M4 4Gb 2port FC: PCIe SFF HBA on PCI bus 0a device 01 irq 49 port 1
  Vendor: TMS       Model: FC65              Rev: 2.7n
  Type:   Direct-Access                      ANSI SCSI revision: 04
SCSI device sdf: 67108864 512-byte hdwr sectors (34360 MB)
```



```
SCSI device sdf: drive cache: write back
SCSI device sdf: 67108864 512-byte hdwr sectors (34360 MB)
SCSI device sdf: drive cache: write back
 sdf: sdf1 sdf2 sdf3 sdf7
```

図 3-13　dmesg での RamSan の認識の確認

3.5.1.2　cat /proc/scsi/scsi

```
[root@localhost ~]# cat /proc/scsi/scsi
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor: FUJITSU  Model: E400A           Rev: 0000
  Type:   Direct-Access                   ANSI SCSI revision: 03
Host: scsi3 Channel: 00 Id: 00 Lun: 00
  Vendor: TMS      Model: FC65            Rev: 2.7n
  Type:   Direct-Access                   ANSI SCSI revision: 04
```

図 3-14　/proc/scsi/scsi

3.5.2 パーティションの作成

```
[root@localhost ~]# fdisk /dev/sdf
Command (m for help): p
Disk /dev/sdf: 34.3 GB, 34359738368 bytes
64 heads, 32 sectors/track, 32768 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdf1               1       32768    33554416   83  Linux
```

図 3-15　fdisk コマンドの実行

3.5.3 ファイルシステムの作成

```
[root@localhost ~]# mkfs -t ext3 /dev/sdf1
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
4194304 inodes, 8388604 blocks
419430 blocks (5.00%) reserved for the super user
```

```
First data block=0
Maximum filesystem blocks=4294967296
256 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 36 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override
```

図 3-16　mkfs コマンドの実行

4　付録

4.1　半導体ディスク装置 RamSan-400 のご紹介

4.1.1 RamSan-400 とは？

RamSan-400 は、米テキサスメモリーシステム社が開発・販売している半導体ディスク装置で、国内では弊社システムクリエイトが代理店として販売・サポートを行っています。

従来のディスクドライブを搭載したストレージ装置では、ヘッドのシークやプラッタの回転などの機械的な動作を伴うために、特にランダム I/O に関しては ms オーダーの遅延が発生してしまい、システムのパフォーマンス低下の一因となっておりました。

RamSan-400 は、一時記憶領域として DDR2 SDRAM メモリを搭載しており機械的な遅延は発生しませんので、ランダムアクセスにおいては、ディスクドライブベースのストレージと比較して数倍～数百倍の I/O アクセスが実現できます。

また、RAID 構成されたバックアップ用のディスクドライブと 30 分動作するバッテリも搭載するなど、データ保護に対する機能も標準で装備されています。

RamSan-400 は OS からは通常のファイバーチャネル（または、インフィニバンド）接続のストレージとして認識されますので、OS やアプリケーションに一切手を加えることなく、アクセスの多いデータだけを移動させることで、I/O ボトルネックを解消できる唯一のソリューションです。

図 4-1　RamSan-400 外観(前面)

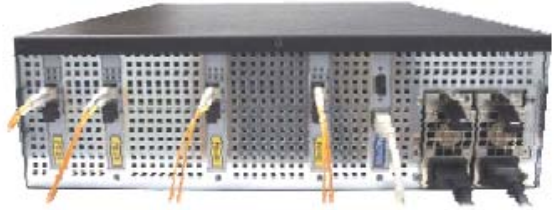


図 4-2　RamSan-400 外観(背面)

4.1.2 RamSan-400 仕様

項目	値
メモリ容量	32／64／96／128GB
IO 性能	400,000IOPS
転送速度（最大）	3.0GB/秒
IO 遅延	14μ秒以下
ホストポート	4Gbps FC×2(標準)～8（最大）and/or　10Gbps InfiniBand x 1(標準)～4（最大）
バックアップディスク	4 台のハードディスク（RAID3；ホットスワップ対応）
バッテリバックアップ	30 分（バッテリ×3 台；冗長性あり）
電源ユニット	2 個（二重化；ホットスワップ対応）
サイズ	3U ラックマウントサイズ

表 4-1　RamSan-400 仕様

4.1.3 RamSan-300

RamSan-400 のエントリモデルとして、メモリが 16GB または 32GB を搭載した RamSan-300 があります。

RamSan-300 は Redo ログなど比較的小さなファイルやオブジェクトの I/O パフォーマンスを改善するのに最適なソリューションです。

詳しくは、弊社ホームページをご覧ください。
http://www.sc-i.co.jp

図 4-3　RamSan-300 外観

4.2　SwingBench について

4.2.1 概要

4.2.1.1　　作者

Dominic Giles（http://www.dominicgiles.com）

18 年間オラクル UK に勤めていたデータベースのスペシャリストです。

4.2.1.2　　Swing Bench とは

SwingBench は簡単な操作で OracleDatabe に負荷を与えるツールです。JAVA ベースで書かれているので、どんなプラットフォーム(1.5JVM)上でも動作します。

SwingBench は３つのフロントエンド(Swingbench、Charbench、Minibench）と 4 つの異なったベンチマークを持っており、自分自身で新たなトランザクションを追加することも可能です。

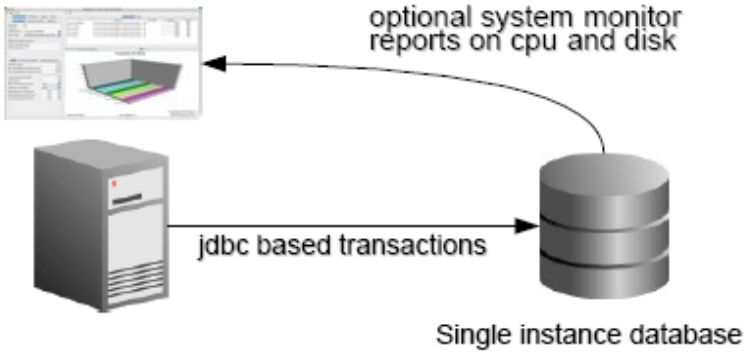


図 4-4　Swing　Bench

4.2.1.3　ベンチマーク

SwingBench は、XML ベースコンフィグレーションファイルによって次の 4 種類の負荷をシミュレートすることができます。

ベンチマーク	Description	Read/Write Ratio	Maximum size
Order Entry	Classic Order Entry Benchmark. (TPC-C like)	60/40	100GB
Calling Circle	Telco based self service application	70/30	100GB
Stress Test	Simple Insert/Update/Delete/Select Benchmark	50/50	―
Sales History	DSS Benchmark	100/0	No Limit

表 4-2　SwingBench 負荷

※上記はデフォルトです。

4.2.1.4　受注シミュレーション（Order Entry）

評価テストでは Swingbench の受注シミュレーション（Order Entry）の負荷を使用しました。これは、受注業務で使用される代表的な 5 種類のトランザクション即ち、顧客の作成、製品の表示、発注、受注処理、および注文の確認で構成されています。

通常では、製品の表示、注文の表示といったトランザクションが過半数を占めるはずですが、今回は、それぞれ、同じだけのトランザクションが来たものとして負荷を 1:1:1:1 に設定しています。

4.2.1.5　実行例と評価方法

図 4-5 は、SwingBench の実行画面の一部で、4 つの帯が 4 種類のトランザクションを、縦軸がトランザクション数/min、横軸が時間（秒）を表します。

トランザクションが安定している場合の帯は水平に近くなり、不安定な場合は波を打ちます。右図は、帯が床面に付いているので、トランザクションが 0（＝停止している）の状態が発生していることが分かります。

実行が終わると、図 4-6 のような XML ファイルが作成され、総トランザクション数などの情報が記述されます。

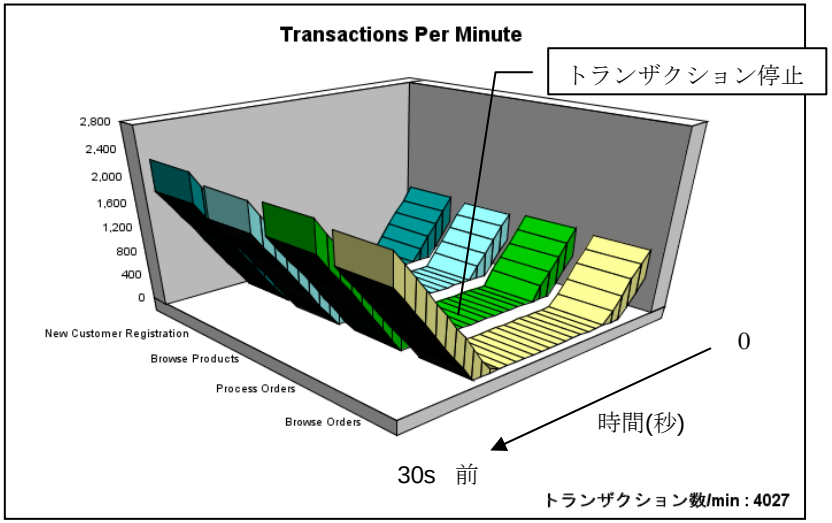


図 4-5　SwingBench 実行例

```
- <Results xmlns="http://www.dominicgiles.com/swingbench">
  - <Overview>
      <BenchmarkName>"Order Entry (PLSQL)"</BenchmarkName>
      <Comment>""</Comment>
      <TimeOfRun>2007/08/24 16:36:10</TimeOfRun>
      <TotalRunTime>1:02:05</TotalRunTime>
      <TotalLogonTime>0:00:02</TotalLogonTime>
      <TotalCompletedTransactions>2188302</TotalCompletedTransactions>
      <TotalFailedTransactions>62</TotalFailedTransactions>
      <AverageTransactionsPerSecond>587.4</AverageTransactionsPerSecond>
      <MaximumTransactionRate>38018</MaximumTransactionRate>
    </Overview>
  - <Configuration>
      <NumberOfUsers>100</NumberOfUsers>
      <MinimumThinkTime>250</MinimumThinkTime>
      <MaximumThinkTime>0</MaximumThinkTime>
      <ConnectString>//10.20.5.166:1521/pgdb1.local</ConnectString>
      <TimingsIn>miliseconds</TimingsIn>
    </Configuration>
  - <DMLResults>
      <SelectStatements>6184620</SelectStatements>
      <InsertStatements>2188535</InsertStatements>
```

図 4-6　実行結果 XML ファイル