
Informix Dynamic Server Migration Methodology



Version Number:1.03 (日本語)

目次

1 . 紹介編	5
1 . 1 はじめに	5
1 . 2 Informix Dynamic Server Ver. 7.xの利点	6
1 . 3 検討課題	7
2 . 新機能編	8
2 . 1 表とインデックスのパーティショニング(表分割)	8
2 . 2 アフィニティ (AFFINITY)	8
2 . 3 並列データ問合せ (PDQ)とPDQPRIORITY	8
2 . 4 ライト・スキャン(先読み)	8
2 . 5 ハッシュ結合	9
2 . 6 64ビットサポート	9
2 . 7 SQLの書き直しとビュー	9
2 . 8 ハイ・パフォーマンス・ローダー (HPL)	9
2 . 9 ユーティリティの追加	9
3 . 機能変更編	10
3 . 1 非同期通信	10
3 . 2 ROW ID	10
3 . 3 一時表	10
3 . 4 データ分散の統計情報	11
3 . 5 並列ソート	11
3 . 6 ソート領域	11
3 . 7 キー値ロック	11
3 . 8 繰り返し読み込み (Repeatable Read)のロックタイプ	12
3 . 9 構文の違い	12
4 . プロジェクト作業編	13
4 . 1 マイグレーションのための参考資料	13
4 . 2 旧バージョンの環境分析	13
4 . 3 過去のデータ・バックアップおよびアンロード	14
4 . 4 データベース設計とスキーマの変更	14
4 . 5 ハードウェアとOSの変更	15
4 . 6 システム設計(カーネル・パラメータ)	15

4 . 7	カーネルA I O	15
4 . 8	新たなO Sのチェック	16
4 . 9	環境変数	16
4 . 10	新しいインフォミックス製品のインストール	17
4 . 11	データベース接続	18
4 . 12	\$ INFORMIXDIR/etc/sqlhostsの編集	19
4 . 13	ONCONF I G構成パラメータの編集	19
4 . 14	Informix Dynamic Serverの初期化	19
4 . 15	データのロード	20
4 . 16	アプリケーションの起動	21
4 . 17	パフォーマンスのモニターとチューニング	21
4 . 18	同じ条件でのテストと評価	22
4 . 19	問題の発見と修正	22
4 . 20	トラブルシューティング	23
4 . 21	最後に	23
付録A	- モニター分析ツールの一覧	24
付録B	- 新旧のコマンドの比較	26
付録C	- 移植チェックリスト	27
特記事項		28

お願い

本書および本書で紹介する製品をご使用になる前に、本書末尾の「特記事項」に記載されている情報をお読みください。

本書には、IBM の専有情報が含まれています。その情報は、使用許諾条件に基づき提供され、著作権により保護されています。本書に記載される情報には、いかなる製品の保証も含まれていません。また、本書で提供されるいかなる記述も、製品保証として解釈すべきではありません。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックslashと表示されたり、バックslashが円記号と表示されたりする場合があります。

1. 紹介編

1.1 はじめに

近年、お客様やパートナー様より「**Informix Dynamic Server をアップグレードする必要はあるのですか?**」という質問が多く寄せられることから、弊社ではこのアップグレードに伴う技術作業に関連する問題を概説した以下の資料を用意しました。本書はOnLine Ver. 5.x等からInformix

Dynamic Server Ver.7.xへの移行に伴い、起こり得る課題に関連したトピックを説明するものです。今回の変更は大規模なものであり、多数の要素について考慮する必要があります。この資料は以下のパートに別れています。

1 . 紹介編

この章では、Informix Dynamic Server Ver7.Xの利点と注意点についてのポイントを示します。

2 . 新機能編

Informix Dynamic Server Ver.7.xは OnLine Ver. 5.xと比べると数多くの機能が向上されています。特に新しい機能についての概要を述べてあります。

3 . 機能変更編

Informix Dynamic Server Ver.7.xは OnLine Ver. 5.xと比べると多くの機能が変更されています。特に重要と思われる変更点について述べてあります。

注意：新機能および機能変更編は共にその概要をわかりやすく伝える事が目的です。その他数多くの新機能や機能変更の詳細については下記等のガイド類を参照してください。

4. プロジェクト作業編

この章では実際のマイグレーションに必要なタスクをステップ順に記述しています。

注意：Informix Dynamic Server Ver.7.x では数多くのマニュアルを取り揃えています。移行作業にあたっては、特に以下のガイドから必ず各詳細をチェックしてください。

『INFORMIX Migration Guide』	(日本語版あり)
『UNIX Products Migration Guide』	(日本語版あり)
『INFORMIX Installation Guide』	(日本語版あり)
『INFORMIX Dynamic Server Administrator's Guide 1/2』	(日本語版あり)
『INFORMIX Dynamic Server Performance Guide』	(日本語版あり)
『Guide to GLS Functionality』	(日本語版あり)

付録A - モニター分析ツールの一覧 以下付録です。

付録B - 新旧のコマンドの比較

付録C - 移植チェックリスト

1.2 Informix Dynamic Server Ver.7.xの利点

まず、OnLine Ver. 5.xのお客様がInformix Dynamic Server Ver.7.xにアップグレードするには2つの大きな理由があります。

1. 機能の強化
2. パフォーマンスの向上

Informix Dynamic Server Ver.7.xを使用すれば、機能の強化は確実に保証されます。パフォーマンスは、特定の状況下でのみ実現されます。Informix Dynamic Server Ver.7.x でパフォーマンスが向上するかどうかは、少なくとも3つの要素によって決まります。

- (1)ハードウェアのタイプ
- (2)アプリケーションのタイプ
- (3)パフォーマンス・チューニングの実行

(1) ハードウェアのタイプ

OnLine Ver. 5.xは、シングルプロセッサのコンピュータ向けにアーキテクチャが設計され、最適化されています。一方でInformix Dynamic Server Ver.7.xは、OLTPのユーザ数が多く、特に大きな並列問合せをも実行するSMPコンピュータ向けにアーキテクチャが設計され、最適化されています。このアーキテクチャには、次のような3つの注意点があります。

- **注意その1**：シングルプロセッサで“軽い”OLTP(ユーザ数150以下)を実行する OnLine Ver. 5.x は、非常に効率性が高く、チューニングを行わない限り Informix Dynamic Server Ver.7.x にアップグレードしても必ずパフォーマンス向上が実現するとは限りません。
- **注意その2**：特に高速 RISC CPU の場合、シングルプロセッサのパフォーマンスがローエンド SMP (CPU2~4) のパフォーマンスとほぼ同じことがあります。これは Informix のパフォーマンスがハードウェアのプロセッサおよびアーキテクチャに大きく依存するためです。
- **注意その3**：適切なチューニングを行えば、Informix Dynamic Server Ver.7.x は一部の“軽い”OLTPシステムで OnLine Ver. 5.x と匹敵するか、これより高いパフォーマンスを発揮する場合があります。

(2) アプリケーションのタイプ

次のような条件のいずれかが該当する場合は、作業負荷に対応するためにInformix Dynamic Server Ver.7.xをお勧めいたします：

- データウェアハウスまたは意思決定支援(DSS)
- ユーザ数が多い OLTP
- 複雑な OLTP

(3) パフォーマンス・チューニング

OnLine Dynamic Server 7は、OnLine Ver. 5.xより機能が優れており、チューニングしやすいのですが、お客様がパフォーマンスの向上を実感する為には、データベース管理とパフォーマンス・チューニングに対する投資を増やすことが必要となります。

1.3 検討課題

表分割、並列処理（PDQ）あるいは先読みなどの新機能が使用できなければ、大幅なパフォーマンスの向上は期待できません。インフォミックスの競争優位であるスケーラビリティの向上でさえ、より大きなハードウェアへの移行が全くなければある程度までしか活用できません。Informix Dynamic Server Ver.7.x 新機能にはある程度のオーバーヘッドが必要になり、これはユーザが小人数（20人から30人）の場合に顕著に現れます。過去のシステムで既にボトルネックの問題が生じている場合、特にCPUの使用率に問題がある場合（ハードウェア、つまりディスク、CPU、メモリ等を拡充しない限り）またはシングルプロセッサのマシンでパフォーマンスが低下する可能性があります。これはInformix Dynamic Server Ver.7.xでは新しいアーキテクチャを採用しているため、より多くのリソースが必要となるからです。

単純な移植作業によるアプリケーション移行のみでは、ある程度以上のパフォーマンス向上は望めません。また新しいインフォミックス製品のバージョンに変更する場合、特にSEからInformix Dynamic Server Ver.7.xに変更する場合は、お客様の運用/管理作業が確実に増えることが予想されます。この際、Informix Migration Guide等の資料が準備されておりますので、そちらを参照してください。

2. 新機能編

Informix Dynamic Server Ver.7.xは OnLine Ver. 5.xと比べると数多くの機能が向上されています。中には全く新しい機能と考えられるものもあります。

2.1 表とインデックスのパーティショニング（表分割）

表およびインデックスを複数のDB領域に分割（パーティショニング）して作成する事ができるようになりました。表データのパーティショニングにはラウンドロビン(fragment by round robin) と表現式(キーレンジとも言う; fragment by expression)による方法が、インデックスのパーティショニングには表現式のみが使用できます。

また、表作成でパーティショニングを指定して、インデックス作成時にパーティショニングの指定や配置するDB領域を指定しないと自動的に表と同じパーティショニングを適用します。そのため表のパーティショニングでラウンドロビンを指定した場合は、その表のインデックスを分散するのに必ず表現式の指定をするか、インデックスの作成をする別のDB領域を指定しなければなりません。

2.2 アフィニティ (AFFINITY)

CPU仮想プロセスを特定の物理CPUに貼り付ける（バインド）アフィニティ機能も一部のOSバージョンで追加されました。インフォミックスのエンジンはアフィニティを利用してバインドされたCPUの処理能力を効率的に利用する事ができます。ただし別のCPUが数多く空いている場合に、限られた数のバインドCPUを待つことは効果をあげません。

2.3 並列データ問合せ（PDQ）とPDQPRIORITY

並列処理を効率的に行う並列データ問合せ（PDQ）機能が追加されました。バッチ処理やDSS処理をPDQで実行する際、CPUやメモリの優先度を宣言するためにPDQPRIORITYが用意されています。環境変数 PDQPRIORITYで指定する方法と、SQL文の『set pdqpriority値』で指定する方法の2通りあり、SQL文での指定が優先します。SQL文で値を変更すると、次の『set pdqpriority値』で変更するまでその値が継続しますので、ある特定の処理だけPDQPRIORITYの値を変更して実行させたい時には、一連の処理完了後に値を『set pdqpriority default』で変更前の値に戻す必要があります。

2.4 ライト・スキャン（先読み）

通常、バッファ・キャッシュは頻繁にアクセスされるページの再使用に依存するOLTPタイプのアプリケーション向けに最適です。ディスクへのI/Oの数を減らせるのであれば、特定のユーザに対してバッファをロックすることでオーバーヘッドが生じてもそれだけの価値があります。ただし大量の情報スキャンが必要な問合せを実行すると、一度しかスキャンされないページでバッファ・キャッシュがいっぱいになり、グローバル・キャッシングの意味がなくなります。

ライト・スキャン（先読み）を使用すればこの問題が軽減できます。これは表やインデックスでの順次読出し等で構成パラメータRA_PAGESで指定するページ分を高速に先読みします。読み込まれる表がメモリに入り切らなくなると、データのスキャンにローカル・バッファが使用されます。この機能を利用するには他のユーザとの衝突を避けるために、ISOLATIONレベルが単純読み込みでなければなりません。

2.5 ハッシュ結合

大量データの結合にはハッシュ表を作成・利用するハッシュ結合が採用されます。ハッシュ結合は通常の結合方式よりはるかに速いのですが、リソースの面に課題があります。処理速度を速めるには、高速のディスクスキャンと十分なメモリがなければ一時表を使用せずに結合を実行することができません。またこれはかなりCPUを消費するプロセスですから、他のユーザの処理速度が落ちる可能性があります。リソースが問題にならないDSS環境ではハッシュ結合により大幅なスピードアップをはかることができます。この詳細は“set explain on”の出力ファイル(sqexplain.out)を見れば個々の表に使用された結合方法が分かります。その後、onconfigファイルのOPTCOMPINDの値をチェックしてください。

2.6 64ビットサポート

64ビットRISCチップOSを販売しているベンダーもあります。64ビットサポート製品は現在、順次、Informix Dynamic Server Ver.7.xで採用されていきます。このサポートは基本的にメモリ管理の領域に限られています。つまりバッファを4GBまで大幅に増やしてキャッシングを効率化させたり、超大型DSSメモリを構成してハッシュ結合を効率化することができます。しかし、チャンク・サイズとディスク上のオフセットが2GBまでの限界は継続して適用されません。

2.7 SQLの書き直しとビュー

SQL文を調べる場合、パーサーは述語を論理的に分析して、より良い問合せパスを提供できるようになりました（たとえば、WHERE句が、WHERE *tab1.a=tab2.b* AND *tab1.a>1000*の場合）。ここで、*tab2.b>1000*の方が高いパフォーマンスを実現する有効なフィルターであると結論することもできます。“set explain on”の出力ファイル(sqexplain.out)を見れば個々の表に使用されたアクセス・パスとフィルターがチェックできます。

この方法で問合せを書き直す以外にも、コンテキストによっては、表のビューを結合されたSELECTとして書き直して、一時表が生成されない1つのSELECT文を作成することも可能です。

Ver. 5.xからInformix Dynamic Server Ver.7.xへ移行する際には、大きな表のフラグメンテーションが可能かどうかを考慮する必要があります。

2.8 ハイ・パフォーマンス・ローダー（HPL）

Informix Dynamic Server Ver.7.2x以降ではハイ・パフォーマンス・ローダー(HPL)が使用できるのでデータのロード時間を大幅に短縮できます。HPLはスペースが小さい場合にディスクを切り替えて順番に表領域を使用します。またファイル・システム、インデックス表領域を並列にアンロードすることもできます。HPLはデータがフラグメンテーション化され、ロード・ファイルが複数のファイル・システムに分割されている場合に、最適なパフォーマンスを発揮します。このため、OnLine Ver. 5.xからInformix Dynamic Server Ver.7.xへ移行する際には、大きな表のフラグメンテーションが可能かどうかを考慮する必要があります。

2.9 ユーティリティの追加

以下のユーティリティが追加されました。

onspaces -d DB領域名	-pパス名	-oオフセット	空のチャンクを削除
onspaces -m DB領域名	-pパス名	-o	ミラーリングの追加
onspaces -r DB領域名			ミラーリングを削除

onparams -d DB領域名 -sサイズ LOGSIZEパラメータと異なるサイズの論理ログを作成

3.機能変更編

Informix Dynamic Server Ver.7.xは OnLine Ver. 5.xと比べると多くの機能が変更されています。

3.1 非同期通信

非同期通信は、OnLine Ver. 5.xとInformix Dynamic Server Ver.7.xの最も重要な違いです。これはディスクからの読み出しを非同期で実行します。OnLine Ver. 5.xでは多数のユーザに対応すると、メモリやプロセス・リソースが大量に消費され、スケジューリングをOSに依存していました。負荷が大きいシステムの場合、これらのリソースの大部分が使用されずに事実上はシステムから排除されます。つまりメモリはページ・アウトされ、プロセスはユーザが何かを実行するまでスリープ状態になります。ユーザ当たりのサーバ・プロセスとシステムコールを中断すれば、OSは比較的容易に対応できます。

Informix Dynamic Server Ver.7.xの場合事実上サーバは1つで、すべての接続を潜在的にアクティブであると想定して処理しなければなりません。またI/Oブロックを発生させずに処理しなければなりません。この方法では小規模から中規模程度のシステムでOnLine Ver. 5.xよりもリソースへの負担が大きくなります。例えばあるシステムに接続ユーザが100人いて、うち作業をするのが10人しかいないとします。OnLine Ver. 5.xでは、90のプロセスがI/Oコールをブロックされて休眠状態となり、I/Oのブロックを実行しているプロセスが10あることになります。Informix Dynamic Server Ver.7.x (シングルCPU) の場合、1つのプロセスが100すべての接続を管理します。これは(プラットフォームによって異なりますが)100の接続を非ブロック・モードで選択/ポーリングしなければなりません。これを実行すればOnLine Ver. 5.xと比較して当然コストが高くなります。

どこで線を引くかはシステムによって異なりますが、この場合のユーザの作業についてはOnLine Ver. 5.xよりも Informix Dynamic Server Ver.7.xの方が効率は低くなります。小規模システムを慎重に構成すればこのような影響を軽減できますが、お客様側で予想される作業負荷がInformix Dynamic Server Ver.7.xに適切なものであることを慎重に確認しなければなりません。

3.2 ROW ID

ROW IDはInformix Dynamic Server Ver.7.xから削除されてはいません。ただし、場合によってはROW IDの使用を控えた方がよい場合があります。通常の表が分割されない場合、ROW IDはOnLine Ver. 5.xの時と同じように機能します。ただし表分割するとROW IDは無効になり、表をWITH ROWIDSで作成しない限り機能しません。この場合、WITH ROWIDSを使用するとROW IDと全く同じような特殊な一意の列が表に追加されます。ところが、エンジンは以前には使用されなかった余分なインデックスを表で保持しなければならなかったため、パフォーマンスが低下します。これを解決する正しい方法は、適切な主キーを使用することです。これは、Informix Dynamic Server Ver.7.x のみならずCDRやIDS/AD、IDS/XP (Ver.8.x) 等の新機能が追加されるにつれ、ますます重要になっていく予定です。

3.3 一時表

一時表の最適化は、ログなしで利用できる一時表領域があるInformix Dynamic Server Ver.7.xの方が優れています。OnLine Ver. 5.xでは、一時表をWITH NO LOGで明示的に作成した場合にのみログしませんでした。Informix Dynamic Server Ver.7.xでは、ログされない一時表領域をその他のタイプの表(たとえば、SUB SELECTなど)に使用することができます。また、複数の一時表がある場合、SELECT INTO TEMP temp WITH NO LOGを使用すると、一時表がフラグメント化されることにも注目してください。

また、Informix Dynamic Server Ver.7.xでは、一時表の名前は、現行データベース・サーバーのその他の永続表と異なる名前にする必要があります。そうしないと、一時表を作成したセッション内で、同じ名前を持つその他の永続表にアクセスしようとしても、一時表へのアクセスが優先されます。

3.4 データ分散の統計情報

コスト・ベースのオプティマイザは、問合せのコストを正確に割り出すために必要な統計情報をすべて保持していなければなりません。ただし、これにはデータ分散の統計情報が適切に構成されている必要があります。これにはUPDATE STATISTICSコマンドを（一定間隔の運用で）実行します。その場合、例えば次のように実行できます。表にUPDATE STATISTICS LOW、インデックスにUPDATE STATISTICS MEDIUM、インデックスの主キー列にUPDATE STATISTICS HIGHを利用する方法も可能です。

3.5 並列ソート

PSORT_NPROCS環境変数はInformix Dynamic Server Ver.7.xでも引き続き使用されますが、プロセスではなくスレッドを参照します。非常に多くのpsortスレッドが割り当てられますから、PSORT_NPROCとインデックス作成スレッドの関係には注意してください。メモリや一時領域に余裕がないと、リソースの問題が生じる可能性があります。

3.6 ソート領域

ソートにはファイル・システム領域が使用されなくなり、今後はソート当たりデフォルトでメモリ5MBのソート領域を割り当てます。こうなると、一時表領域の一時DB領域へのあふれが生じます。一時パーティションは、特定のユーザに関連するとは報告されませんが、onstat -tオプションを使用すれば一時DB領域の使用を追跡することができます。一時表領域には多数のスクラッチ領域が割り当てられます。

一時表とソートファイル

	Ver5.x	Ver7.x
一時ソートファイル	1) PSORT_DBTEMPに指定したディレクトリ順 2) PSORT_DMTEMPがない場合はDBTEMPが指定する1つのディレクトリ 3)上記環境変数がない場合は /tmp	1)PSORT_DBTEMP 2)DBSPACETEMP環境変数 3)DBSPACETEMP構成パラメータ 4) /tmp
暗黙の一時表 select info temp	ルートDB領域	1)DBSPACETEMP環境変数 2)DBSPACETEMP構成パラメータ
明示的な一時表 create temp table	指定したDB領域またはデータベースの作成先DB領域	1)DBSPACETEMP環境変数 2)DBSPACETEMP構成パラメータ 3)同左

3.7 キー値ロック

OnLine Ver. 5.x ではインデックス・キーの削除/挿入時にB-treeについて隣接キーロックを使用していましたが、そのため削除した時には別のユーザによる挿入が、また挿入した時には別のユーザによる削除が隣接キーロックの排他ロックにより実行できませんでした。

Informix Dynamic Server Ver.7.x のキー値ロックでは、削除の際に1バイトの削除フラグの値を変更してキー値に排他ロックをかけ、別のユーザによる挿入を妨げません。また挿入の際にはキー値に排他ロックをかけ、別のユーザによる削除が可能です。

3.8 繰返し読み込み (Repeatable Read) のロックタイプ

Repeatable Readでのselect文では検索対象のインデックス・キーにS R (共有繰返し読み込み)ロックをかけ、delete文では削除するキーにまたinsert文では現行のキーにX R (排他繰返し読み込み)をかけます。

別のユーザによる追加の処理では、インデックス・キーを挿入する次のキーにS RロックやX Rロックの存在を確認し、Repeatable Readを実行中のユーザに対する追加処理の影響をなくします。

動作	現行キーのロック	次キーのチェック
select (CR)	S ロック	しない
select (CS)	S ロックのテスト	しない
select (RR)	S R ロック	しない
insert (CR)	X ロック	X R or S R ロックのテスト
insert (RR)	X ロック X R ロック	X R or S R ロックのテスト X R or S R ロックのテスト
delete (CR)	X ロック	しない
delete (CS)	X ロック	しない
delete (RR)	X R ロック	しない

OnLine Ver. 5.xで作成されたインデックスのうち、カタログ表のインデックスはInformix Dynamic Server Ver.7.x の起動時に自動変換で『削除フラグ』が追加されますが、ユーザ作成のインデックスはインデックスの再作成またはupdate statistics実行時によって削除フラグが追加されます。

3.9 構文の違い

Informix Dynamic Server Ver.7.xにはSQL構文に多数の追加がなされていますが、バージョンの違いによる大きな問題が発生する可能性はほとんどありません。アプリケーションの変更が必要になるような大きな変更があれば、リリースノートに記載されています。ただし旧構文の中には、新しい内部機能の使用を妨害したり、パフォーマンスの向上を阻むものもあります。一方でパフォーマンスを向上させるような新しい構文は(アプリケーションの変更が許されるのであれば)考慮の価値があるでしょう。これに該当するのはアプリケーションに修正を加えることなく変更できる場合のストアードプロシジャやSQL等です。

また新しい構文はフロントエンド製品がエンジンと同調している場合に限り、フロントエンド側で使用できることに注意してください。これにはエンジンと同じバージョンのESQL/C等が該当します。その他のフロントエンド・プログラムの場合は、言語パーサーにとって旧バージョンから既知の文を作成して実行しなければなりません。

4. プロジェクト作業編

この章では実際のマイグレーションに必要なタスクをステップ順に記述しています。

4.1 マイグレーションのための参考資料

Informix Dynamic Server Ver.7.xでは以下のマイグレーション資料を取り揃えています。以下の移行作業にあたっては必ず詳細を参照してください。

『INFORMIX Migration Guide』	(日本語版あり)
『UNIX Products Migration Guide』	(日本語版あり)
『INFORMIX Installation Guide』	(日本語版あり)
『INFORMIX Dynamic Server Administrator's Guide 1』	(日本語版あり)
『INFORMIX Dynamic Server Administrator's Guide 2』	(日本語版あり)
『INFORMIX Dynamic Server Performance Guide』	(日本語版あり)
『Guide to GLS Functionality』	(日本語版あり)

4.2 旧バージョンの環境分析

INFORMIXシステムの環境設定をする項目は極めてたくさんあります。特定のハードウェア、OS、インフォミックスのバージョン等を設定して最大限に活用したとしても、OnLine Ver.5では高いパフォーマンスが得られ、Informix Dynamic Server Ver.7.xでは得られないというような事態になる可能性があります。

この事態を避けるため、既存の環境を技術担当者と一緒にチェックリスト（付録Cを参照）に記入してください。チェックリストの内容によっては、さらに追加の調査を推奨します。その場合はさらに詳細な状況の分析を行うべきでしょう。これには次のものが含まれます：

- 旧バージョンによるマシンの使用率の分析と管理資料（UNIX分析ツール等による）
- 一定負荷（ユーザ数、並列オペレーション）で運用時の各処理の応答時間
- バックアップの現状とその応答時間の資料

■ 既知の問題についての情報

特にアプリケーションとデータベースについて、すでにボトルネックが生じていないかどうかを判断することが重要です。このような点は、移行後まず一番にテストしてください。新しい環境で再生し、テストできるように、過去の環境の分析結果と統計情報（実行時間 real や “ set explain on;”からのsqexplain.out）等を用意しておくことをお勧めします。

4.3 過去のデータ・バックアップおよびアンロード

小さなデータベース（表が100以下で4GB以下）ならば、たいていは“dlexport -ss”を使用してディスクにアンロードできます。アンロードされると、自動的にスキーマ・ファイルが作成されます。巨大なデータベースの場合はこの方法は使えません。まず、“dbschema -ss”を使用してスキーマ・ファイルを作成してください。次に、“unload to filename select * from tablename”を使用して個々の表毎にアンロードします。OnLine Ver.6.x より前のバージョンでは“-ss”を指定できません。大きなデータベースの場合は、バックアップ・コピーを作成するための追加のディスク・スペースを用意し、表からファイルをアンロードした方がよいでしょう。追加ディスクで完全なバックアップを必ず作成してください。（特に本番系システムの場合）移植後ただちにプロダクション・オペレーションを開始するのであれば、新しいディレクトリに新しいバージョンをインストールして、旧バージョンで緊急作業ができるように並行業務の可能性も考慮しておかなければなりません。

4.4 データベース設計とスキーマの変更

Informix Dynamic Server Ver.7.xには、うまく設計されたOnLine Ver. 5.xデータベースでも Informix Dynamic Server Ver.7.x環境では最大限のパフォーマンスを発揮しなくなるような、いくつかの変更と拡張が加えられています。

新しい環境で複数のフラグメントに対するアクセスの分散化が可能かつ有効であれば、使用頻度が高い大きな表に対して表分割を考慮してください。これには最適なディスク入出力の均一な分散を可能にするだけの十分なディスクを必要とします。また表現式による表分割をおこなう場合、式の定義はあまり複雑なものにしないようにしてください。表分割を使用する場合は、複数の物理的ディスクに分散されたディスクスペースが必要になります。その時、1つのディスクに複数のフラグメントを作成するのは良くありません。異なるディスクレイアウトを検討してください。

表ごとに初期エクステント・サイズを決定します。OnLine Ver.6.x 以上では、“dbschema -ss...”または“dbexport -ss”で、自動的にエクステントの現在の状況がわかり、それを利用できます。OnLine Ver.6.xより前のバージョンの場合は、マニュアルに記載された式により初期エクステント・サイズを割り出すか、“tbcheck -pe”を使用して文書化する必要があります。時間的な余裕がない場合のみ、レコード行の長さをすべてのインデックスの長さに足し、その数に行数を乗じ、（注：データはページ単位に格納されます）その結果の10%から20%で（極めて簡易な計算ですが）エクステント・サイズを割り出すことができます。“oncheck -pT databasename:tablename”を使用して、エクステントを割り出します。さらに追加エクステント（追加サイズ）も設定する必要があります。初期エクステント・サイズを8（大きな表の場合は16）で割ることが、同じく簡単な方法です。

過去のシステムの場合には、SQLの最適化を実行することを推奨します。よく複合インデックスの最初のキーについて有効ではない列が見つかります。このようなインデックスは、可能であればまずプログラム開発者またはシステム管理者と話し合ってから、再構築し直してください（結果をチェックするには、set explain on;"からのsqexplain.out を使用できます）。最終的に次のような情報をチェックおよび考慮してください。

- 各表のエクステント・サイズと追加サイズ
- DB領域への表およびインデックスの割当て（ディスク・レイアウト）
- 表およびインデックスの表分割基準の定義
- BLOB データを表と一緒に格納しない場合の、BLOB領域へのBLOB データ（データタイプ TEXT と BYTE）の割当て

4.5 ハードウェアとOSの変更

アップグレードと同時にハードウェアの大きな変更が行われると、インフォミックスの変更とシステム全体の変更を切り分けるのが大変難しくなります。このような状況では、変更以前に細かな測定が行われていて、ハードウェアの変更から予測される変化をすべて数量化できるレベルまで理解されていない限り、変化の善し悪しを比較することは不可能です[証明は非常に難しい]。ディスク速度を測るddやネットワーク・パフォーマンスを示すnetstat、あるいは競合を測定するrsamラッチテストなど、OSツールがあれば、少なくともパフォーマンスの問題を生じるようなOSの変更を認識するに足りる情報は得られるはずです。

4.6 システム設定 (カーネル・パラメータ)

カーネル・パラメータをチェックしてください。機種やUNIXのバージョンにもよりますが通常は以下のカーネル・パラメータが関与していることが多いと思われます。UNIXシステム管理者の助けを借りてチェックしてください。

- 共有メモリのパラメータ (SHMMAX、SHMMIN、SHMALL、SHMMNI、SHMSEG、SHMBRK)
- セマフォ処理のためのパラメータ (SEMMNI、SEMMNS、SEMMSL)
- システム全体でのプロセス数、ユーザ数、ユーザ当たりのオープン・ファイル数、システム全体でのオープン・ファイル数
- システムの最大メモリ (MAXMEM)
- ファイル・サイズの上限 (特にロウ・デバイスではなくクックド・ファイルの場合)

OSリソースの管理の重要性が高まったことで、場合によってはカーネルの限界に注意を払う必要があります。リリースノートには、足掛かりとして使用できるカーネル・パラメータのサンプル構成が含まれていますがこれはOSの以前のバージョンには適用できても、大きなサーバや特殊なニーズを持つサーバには適用できないかもしれません。リリースノートには、その構成のマシンでポートとQAテストが実行されたことしか記載されておらず、厳密な限界テストが実施されたとは書かれていない場合があります。この場合にカーネルの限界を調べる場合にはシステム管理者およびOSベンダーへの問合せをお勧めいたします。

4.7 カーネルAIO

リリースノートに掲載された「カーネル非同期I/O (KAIO)」に関する情報をチェックしてください。この情報はプラットフォーム固有です。KAIOをアクティブ (使用) または非アクティブ (不使用) にするのに環境変数 (KAIOONまたはKAIOOFF) を設定することが可能です。この環境変数を設定する必要がある場合は、関連するプロフィールを更新しなければなりません。Informix Dynamic Server 7.xのリリースノートの最後に出てくる調整可能なパラメータもチェックして、必要であれば設定してください。

注意：前記のアフィニティや新しいツール (例：Onperf、HPL、DB/Cockpit等) に関する情報もリリースノートに含まれています。またリリースノートをもとに、Informix Dynamic Serverが利用するプラットフォームにおいてサポートしている、記憶媒体やバックアップ・ツールについて調べてください。

4.8 新たなOSのチェック

場合によっては、Informix Dynamic Server Ver.7.x製品がOnLine Ver. 5.xと同じOSバージョン（SunOSとSolarisなど）では動作しないことがあります。こういう場合には、OSの違いを十分に理解して、Informix Dynamic Serverの変化と関連付けることが極めて重要になります。新しいOSがまだ不完全で、前のOSと同じパフォーマンスを実現するまでには多くの環境設定と修正が必要になることもあります。

また、現行プラットフォームに歩調を合わせる努力は大切ですが、そのためには、通常ベンダーに新しいOSに以前のOSとの下位互換性があるかどうかを問い合せます。下位互換性があればほとんど、「新しい」バージョンを利用することが可能です。しかしこの構成が最適化されておらず、新しいOSで以前のバージョンのバイナリが動くことによってパフォーマンスを低下させるような可能性もあります。また当然、OSの新機能は以前のOSのバイナリにはアクセスできません。また、ソフトウェアやOSにさらなる改善を行う前に修正が必要とされるバグ（機能の変更も）が現れることもあります。

さらにエンジンが複雑になるにつれて、特に複数の接続や、大量の共有メモリ、非同期オペレーションなどの処理に関して、システムコールへの依存度が高まってきています。極端にいつてしまえば、このような限界について全くテストされていないOSが数多くあり、Informix Dynamic Server Ver.7.xは期待どおり動作してくれない可能性もあります。障害のタイプは必ずしも、明らかなプロセスの終了や内部で報告されるエラーとは限らず、ハングアップやパフォーマンスの低下などにも起こり得ます。

最後に**サーバー固有のリリースノート(マシンノートとも言う。\$INFORMIXDIR/release にあります)に必要なパッチの情報を必ず参照してください。**必要に応じて最新情報をベンダーに問い合わせて、確実にパッチを提供してもらうようように心掛けてください。OSの問題による影響を最も受けやすいコードはネットワーク・コード、共有メモリ使用率、ディスクI/Oが多く報告されております。この場合、ベンダー固有のシステム管理ツールを十分に理解しておけばとても役立ちます。

4.9 環境変数

OnLine Ver. 5.xに適用された環境変数の中には、Informix Dynamic Server 7.xでは使用されないものがあります。ユーザ環境を調べて、インフォミックス関連の変数がすべて適切かどうか確認してください。エンジンのパフォーマンスを最大限に活用するには、環境変数を変える事が必要な場合があります。リリース・ノートを調べて、ユーザ・インフォミックスの環境に適応させてください。Informix Dynamic Server 7.xでは、環境変数の設定は、\$INFORMIXDIR/etc/informix.rcファイルを介してグローバルに行われるか、ユーザー毎の環境変数設定ファイルや ~user/.informixファイルを使用してユーザごとに行われます。これに該当するのはローカル・ユーザだけで、ネットワーク・クライアントには該当されないことに注意してください。最低限のユーザinformixの環境変数の設定は以下の通りです。またいくつかの種類の環境変数についての概要を以下に示します。

Informix Dynamic Server 7.xで必要最低限の環境変数

環境変数	値	例
INFORMIXDIR	Ver7.2xのインストール・ディレクトリ	/usr/informix/7.2x
PATH	\$INFORMIXDIR/binを含める	\${INFORMIXDIR}/bin:\${PATH}
INFORMIXSERVER	サーバ名	June
ONCONFIG	構成パラメータ・ファイル	onconfig.June

Informix Dynamic Server 7.x のアプリケーションで重要な環境変数

TREM	使用する端末の種類	必須
TERMCAP TERMINFO	\$INFORMIXDIR/etcの termcap または terminfo	I-SQL や I-4GL を使用する時
INFORMIXTREM	TERMCAP、TERMINFO のどちらを使用するか	I-SQL や I-4GL を使用する時
DB_LOCALE	データベースの内部コード	日本語データを格納する時
SERVER_LOCALE	ログ・ファイルやメッセージの出力に使用するコード	日本語表示が必要な時
CLIENT_LOCALE	クライアント・アプリが使用するコード	日本語表示が必要な時
ESQLMF	ESQL/C で複数バイトをコパ ール時の前処理方法	I-SQL や I-4GL を使用する時
GLS8BITSYS	ファイル名に複数バイトが使用できるかどうか	メッセージログ、診断情報ファイル、dbexport の指定ファイル名、ESQLソースファイル名
DBAPICODE	Ver5 で使用していた API コードを指定する	Ver5 との互換性のため
DBCODESET	Ver5 で使用していた内部コードを指定する	Ver5 との互換性のため

Informix Dynamic Server 7.x のアプリケーションで便利な環境変数

INFORMIXSERVER	サーバ名	DBPATH より優先
DBPATH	//サーバ名://別のサーバ名	サーバ・エントリの検索順
PDQPRIORITY	0, 1 ~ 100, default, high, low, off のどれか	並列処理の度合の設定
INFORMIXSQLHOSTS	\$INFORMIXDIR/etc/sqlhosts 以外のパス名	標準以外のファイルを使用したい場合
PSORT_NPROC	0 または 数値	ソート・スレッドの数、下記参照

PSORT_NPROCS	表のソート	インデックスの作成
環境変数の設定なし	1 スレッド	CPU が 1 つ : フラグメントあたり 1 スレッド
0	3 スレッド	CPU が 複数 : フラグメントあたり 2 スレッド
値	指定した値を上限値とする	フラグメントごとに指定した値

Informix Dynamic Server 7.x では提供されない環境変数

SQLEXEC	DBATOWCHAR	JIOCODE	DBASCIIIBC
DBMONEYSYSCALE	DBKSTOWHIRA	KINSOKUBOLCHAR	DBCS3BC
DBDOW	DBKTOWHIRA	KINSOKUEOLCHAR	
DBERA	DBWHITESP	KJMAPLEVEL	

OnLine Ver. 5.x で使用していた環境変数が Ver. 7 で継続使用できるもの

DBINFXRC	スタートアップ・ファイルのパス名	デフォルトは \$INFORMIXDIR/etc/infrmixenv.rc
DBMONEY	MONEY 型の書式	
DBTIME	DATETIME 型の書式	

OnLine Ver. 5.x とは異なる指定となる Ver. 7の環境変数

DBDATE	DATE型の書式で元号を使用する時	年の指定がGY2またはEY2
DBFORM	フォームで使用するコードセットの指定	環境変数CLIENT_LOCALEと同じ値にすること
DBLANG	INFORMIX製品で表示されるセッティングやフォームで使用するエリア名	日本語表示 例: ja_jp.sjis 詳細下記参照 英語表示 en_us.8859-1

日本語内部コードの変更での注意点

Ver5.x	指定方法	Ver7.x	Ver7.xのスキーマで必要な作業	注意点
Sjis	デフォルト	sjis	なし	なし
		sjis-s	なし	JIS X 0212符号は変換されない
		ujis	半角カナの列定義の変更	半角カナに2倍の領域が必要
Sjis-s	DBCDESET	sjis	なし	なし
		sjis-s	なし	なし
		ujis	半角カナの列定義の変更	半角カナに2倍の領域が必要
Ujis	DBCDESET	sjis	半角カナの列定義の変更	半角カナに2倍の領域が必要
		sjis-s	半角カナの列定義の変更	半角カナに2倍の領域が必要
		ujis	なし	なし

4.10 新しいインフォミックス製品のインストール

可能であれば、新製品は旧製品と一緒に別々のディレクトリにインストールしてください。つまり、重大な問題が生じて、必要であれば旧バージョンを使用して作業を続行できるようにしておきます。手順は以下の通りです。

- ユーザ名 "informix" とグループ "informix" を作成 (ユーザ "informix" はグループ "informix")
- Informix ディレクトリを定義 (環境変数 INFORMIXDIR を設定します) パスを拡張してください (PATH=\$INFORMIXDIR/bin:\$PATH)
- インストールはユーザ名 "root" として実行してください。
- 必ずフロントエンド製品の旧バージョンを先にインストールします
- Informix Dynamic Server 本体の前にすべてのフロントエンド製品をインストールします
- Informix Dynamic Server 本体をインストールします
- 最後に Language Supplement JA Version (GLS) をインストールします。

注意：実作業には必ずInstall Guideを参照してください。

4.11 データベース接続

フロントエンド/バックエンド通信システムがかなり複雑になりました。OnLine Ver. 5.xではパイプ経由の接続だけが使用されていましたが、ネットワーク接続が必要であれば、フロントエンド・パイプ・トラフィックをネットワーク・ソケットに経路指定する中間プロセスが必要になります。Informix Dynamic Server 7.xでは、接続タイプはプラットフォームによって異なりますが、主な接続タイプは次のとおりです。

- ipcshm (共有メモリを通じた IPC) - ローカル接続のみ。
- ipcstr (ストリーム・パイプを通じた IPC) - 複数のローカル接続のみ。
- tlispx (IPX/SPX 接続) - 複数のネットワークおよびループバック。
- soctcp (ソケットを通じた TCP 接続) - 複数のネットワークおよびループバック。
- tlitcp (TLI を通じた TCP 接続) - 複数のネットワークおよびループバック。

サポートはプラットフォームによって異なり、追加のOSオプションが必要になることがあります。ハードウェアとOSの組み合わせが異なると、たとえば共有メモリの実装が弱かったり、使用されるマシン・リソースに法外な費用がかかるなどのさまざまな理由により、以前の接続タイプが最適ではないこともあります

4.12 \$INFORMIXDIR/etc/sqlhostsの編集

OnLine Ver. 5.xでは記述が必要ありませんでしたが、Informix Dynamic Server 7.x から必須になりました。以下の行を追加してください。(tab)はタブ・キーを意味します。

サーバ名(tab)onipcshm(tab)ホスト名(tab)サーバ名

【 例 (ホスト名をfujiとする) 】

```
June    onipcshm          fuji          June
```

4.13 ONCONFIG構成パラメータの編集

以下の手順で構成パラメータ・ファイルを用意して編集してください)

```
informix> cd $INFORMIXDIR/etc
informix> cp onconfig.std $ONCONFIG
informix> vi $ONCONFIG
```

Informix Dynamic Server 7.xが初期化できるために最低限設定しなければならないパラメータとその値を以下に示します。

パラメータ	値	注意
ROOTPATH	ローカルまたはUNIXファイルパス	
MSGPATH	存在するパス/online.log	
TAPEDEV	/dev/null	
LTAPEDEV	/dev/null	
SERVERNUM	0	他のINFORMIXサーバで使用していない値
SERVERNAME	June	サーバ名を記述する

NETTYPE	ipcshm,1,20,CPU	同じシステムのクライアントから接続することを前提
NUMAIOVPS		KAIOでは1か2、AIOは使用ディスク数
DRLOSTFOUND	存在するパス/dr.lostfound	

4.14 Informix Dynamic Server の初期化 (ディスク領域と共有メモリの設定)

ディスクはUNIXシステム管理者と一緒に用意してください。ロウ・デバイスまたはクックド・ファイルのどちらを使用するか、RAIDシステムを利用するか、ボリューム管理ツール (Logical Volume Manager等) を使用するか、モニター用ツール (Informixまたはオペレーティング・システム) を使用するかなどを決めなければなりません。Logical Volume Managerを使用する場合は、論理ボリュームが複数のディスクに分散されていないかどうか確認してください。分散しているとパラレルアクセスは保証できません。

onmonitorのParameterメニューのInitializeか、コマンド oninit -iy のどちらかで初期化します。メッセージファイルを確認して初期化が成功したかを確認してください。さらにユーザー " informix"としてサーバーを起動して、共有メモリのサイズ、共有メモリ・セグメントの数 (ipcsm または onstat -g seg)、メッセージログ (onstat -m) をチェックしてください。次にチャンク、さらにDB領域、BLOB領域をセットアップしてください。

ディスク・ボリュームおよびディスク領域は、お客様側のシステム管理者に作成を依頼してください。システム管理者は、チャンクのシンボリック・リンクをセットアップしなければなりません。多くのDB領域を1つのディスクに入れるとパフォーマンスの改良が難しくなります。一方、ディスクの容量が2Gバイト以上であれば、ディスク当たり複数のロウ・デバイスを作成しなければなりません。この場合にはまず複数のディスクにDB領域を順番に作成してから、また最初のディスクに戻ってください (たとえば、4Gバイトのディスクが3つある場合は、DB領域1と4をディスク1に、DB領域2と5をディスク2に、DB領域3と6をディスク3に入れるとデータのフラッシュ時等でパフォーマンス上有利です)。

重要：チャンクには「サイズ+オフセットで最大2Gバイト」が適用されます。またプラットフォームによっては、オペレーティング・システムがディスクの先頭にヘッダ情報を格納しなければなりません。ディスク全体 (おおよそ2Gバイト) がロウ・デバイスとして定義されているか、またはディスクの先頭からロウ・デバイスが始まる場合、チャンクはオフセット0から始めることができないはずです。(オペレーティング・システムの管理マニュアルまたはLogical Volume Managerの説明を参照してください)。

一時DB領域をセットアップします。サイズはアプリケーション、SQL文 (order by、group by、into temp,...)、ユーザ数、表のサイズ、使用されるインデックスによって決まります。一時DB領域は、インデックス作成や " update statistics"の実行に使用されるので、一時DB領域は常に、少なくともデータベース全体の " update statistics medium"や最大の表にインデックスを作成できるだけの大きさが必要です。できる限り複数のディスクに、一時DB領域を作成した方が良いでしょう。

以上で サーバの初期化が完了し、データベースが作成できる環境になりますが、日本語環境のデータベースにしたい場合は、使用する日本語のコードを明記しなければなりません。英語環境のDBに対しては英語環境で利用しなければなりませんし、SJISコードで作成したデータはSJISコード環境で利用しなければなりません。そのため、OnLine Ver. 5.x のINFORMIX-SQLのアプリケーションがInformix Dynamic Server 7.x のDBにアクセスするためには、OnLine Ver. 5.xの日本語コードと同じコードのInformix Dynamic Server 7.xのDBを作成した方が便利です。Informix Dynamic Server 7.x のDBを利用するためにOnLine Ver. 5.x のINFORMIX-SQLを変更する点がUNIX Products Migration Guideの5-3 ~ 5-7ページに記載されていますので、確認してください。

4.15 データのロード

小さなデータベース(総計4GB以内)は“dbimport”でロードする事をお勧めします。ロードをする前にスキーマ・ファイルを更新しておく必要があります。dbimportはまずデータをロードしてから、インデックスを作成します。

データをロードするには、データがアンロードされた時と同じデータタイプに関する環境変数(DBDATE、DBMONEY、DBFORMAT、GLS日本語環境の変数)を、設定しておかなければなりません。

大量のデータを扱う場合は、エクステント設定のあるスキーマ・ファイルを使用して、まずデータベースを作成する必要があります。個々の表が必要であれば、後から同時並行で“load”コマンドを使用することによりロードする事ができます。

注意：同じディスクへのパラレル・ロードは避けてください。場合によっては同時並行で、インデックスが個々並列に作成されます。さらに制約条件を適切な順序で生成するようにしてください。

注意：大きな表にインデックスを作成するには大量の一時DB領域が必要になります。インデックスをロードまたは作成する間、環境変数PSORT_NPROCSをCPUの最大数に設定し、パラメータDBSPACETEMPもそれに応じて設定しておくことをお勧めいたします。

ロード・プロセスが完了したら、“update statistics”オペレーションを実行してください。通常はデータベース全体で“update statistics medium”を実行すれば十分です。必要であれば、個々の表のインデックスキーとなる列に“update statistics high”を実行することができます。

重要：この時点で新しいテープで、使用可能なバックアップ・ツールを使用して、初期レベルのデータ・バックアップを実行してください。

4.16 アプリケーションの起動

移植を実行する前に、アプリケーション移植の責任者を決定する様にしてください。

アプリケーションはすべて事前にコンパイルして、サーバの適切なバージョンにリンクしておかなければなりません。詳細はMigration Guideを参照してください。

本番用システムを移植するのであれば、すべてのアプリケーション機能テストが正常に実行されていることが条件になります。これについてはチェックリストを確認していただく必要があります。

またOnLine Ver. 6.x以降でPDQパラメータを正しく設定するには、意思決定支援(DSS)系問合せ(型、数、頻度)について知っていなければなりません。

4.17 パフォーマンスのモニターとチューニング

パフォーマンスのモニター手段がないと、移行前と移行後の影響を比較したり、結果として起こるチューニングをテストすることは非常に難しくなります。システムの以前のパフォーマンスと比較する材料がないまま、パフォーマンスが低下したと苦情を言うお客様も少なくはないのです。OSとInformix Dynamic Server Ver.7.xはリソース(メモリ、CPU、ディスク)をモニターするさまざまな手段を提供しています。以前の分析で実行された方法を、ここで繰り返してください。後で詳細を評価できるように、結果を文書化しておくことをお勧めします。一般的なツール類に関しては付録Aを参照してください。

お客様がパフォーマンスが低下したと感じる場合、システムが変更されたことに多くは関係があります。この変化には、非常に顕著な場合（ハードウェア/ソフトウェアの変更）と、ごくわずかな場合（表に数行が増えただけでも影響が出る）可能性があります。エンジンのバージョン全体が大きく変更されると、内部の数多くのパラメータが変更される為、その問題が簡単に対処できるような小さな問題なのか、新しいソフトウェアと既存の構成の互換性という大きな問題であるのかを指摘するのが難しくなる場合もあります。

パフォーマンスの問題を調査する際、単に旧バージョンよりも新しいバージョンの方が遅いのだと片付けてしまうのではなく、その他の多数の要素を考慮することが重要です。以下に挙げるポイントは、OnLine 5.xからInformix Dynamic Server Ver.7.xへの変換で発生したものです。その多くは一般的なバージョンアップにも該当するものです。

この場合、パフォーマンス低下の原因を突き止めることができなければ、アプリケーションあるいはSQLを“set explain on”で起動してください。出力ファイル（sqexplain.out）を見れば個々の表に使用されたオプティマイザのアクセス戦略が分かります。またsysmasterデータベースのSMI表“sysptprof”は、たとえばどの表が「順次スキャン」で読み込まれたかなどを調べるのに良い情報源です。通常、ONCONFIGファイルのパラメータOPTCOMPIND、あるいはこれを環境変数として0に設定する（デフォルト値は2）場合があります。こうすれば、オプティマイザは旧バージョンとほぼ同じように動作し、ハッシュ結合ではなく既存のインデックスが使用されます。このことはSMI表sysqqexplainやsysconblockで調べることができます。

Informix Dynamic Server Ver.7.xは、パフォーマンスを微調整する多数の機能を提供しています。どの機能が必要かは、システムおよびアプリケーションの監視結果によって決まります。またインフォミックスのトレーニング・センターではパフォーマンスチューニングに関するトレーニングを実施しております。

ヒント：一度に多数のチューニング・パラメータを変更する事は、基本的に避けてください（パラメータは一回に1つずつ変更してください）。

手順は段階的に進めてください。まず、基本的なチューニングを行ってから、個々のアプリケーションまたはプログラムを微調整してください。環境変数で、一時的に無効にできない構成パラメータは、アプリケーション全体に適した設定にしなければなりません。業務時間外に実行されるバッチ・オペレーションについては、別のONCONFIG構成ファイルを使用することもできます（注：サーバの再起動が必要です）。

4.18 同じ条件でのテストと評価

2つのシステムを比較する場合、バージョンの違いに直接関係のない要素を取り除き、できるだけ同じ状態で比較すべきであることは言うまでもないことです。両者はプラットフォームが異なり、OSも異なることがあります。前述のように、パフォーマンスを向上させるためにはフロントエンド製品を最新バージョンにアップグレードすることが賢明な方法です。実行されるSQL文、マシン負荷なども同じでなければなりません。このような条件が揃わないと、パフォーマンスが明らかに大幅にダウンしている場合以外は、ゼロから始めてパフォーマンスの問題を診断しなければならなくなります。

これらのオペレーションのOnLine Ver. 5.xでの実行時間と、システム構成、予想されるシステム負荷など、Informix Dynamic Server Ver.7.xで同じテストを再現するのに必要なすべての条件を記録しておきます。また、比較できるようにsqexplain.outファイルも取得しておきます。移行後に、OnLine Ver. 5.xでのテスト時とできるだけ近い条件で同じテストを実行してください。問題があれば、ここで明らかになるはずですが、見つかった異常がほんの少力で、テストで問題があった所だけを修正できれば良いのですが、実際のオペレーションでも同じような問題が発生して専門家の手助けが必要になる場合もあります。全般的な問題が見つかった場合には、テスト結果をもとにシステム全体を修正する必要が生じます。

4.19 問題の発見と修正

パフォーマンスの問題を診断する方法を正確に説明するのは難しいのですが、過去に信頼できる結果があれば、後はその結果を比較するだけで済みます。一般的には、まず問合せプラン (SET EXPLAIN ON; の sqexplain.out) を見ます。これは、問題の原因が単にオプティマイザの選択の違いにあることが多いからです。プランが同じであれば、次に物理的なレイアウトを調べて、たとえばデータが再ロードされたなど、何か変更がないかどうか確認してください。単に ONCONFIG 構成パラメータが不十分でチューニングが必要であることが分かる場合もあります。

注意： データ配置を適切に設定すればオプティマイザが判断を誤ることなどは考えにくいものですが、まれに発生します。特定のインスタンスはそれぞれ詳細は異なるはずですが、オプティマイザが I/O コストや CPU 速度などのコストを計算する前提条件の中には一定のものがありますが、常に適切とは限りません。ゆえに誤ったパスが選ばれる可能性があります。問題を突き止めるにはさらなる調査が必要です。

問題を突き止める作業をしていれば、おのずと問題を修正するアクションが分かってきます。これには、問合せの書き直しや、ONCONFIG 構成パラメータの変更、OS パラメータの変更などが含まれます。あるいは、データベースの論理的または物理的デザインに問題があることが分かる場合もあります。

4.20 トラブルシューティング

重要： インフォミックス製品が関与する問題が発生した場合には、お客様が、サポート契約に基づくサポート方法を通じて連絡することが可能です。

4.21 最後に

Informix Dynamic Server Ver.7.x と OnLine Ver. 5.x のパフォーマンスを比較する際に考慮しなければならない領域が、数多くあります。Informix Dynamic Server Ver.7.x システムを適切にセットアップすれば、いくつかの例外を除いて OnLine Ver. 5.x よりも高いパフォーマンスが実現できるはずですが、残念ながら、ハードウェアや OS、アプリケーションの制約により、Informix Dynamic Server Ver.7.x を最適に構成できない場合には、Informix Dynamic Server Ver.7.x が本来の処理速度を発揮できないように思えることもあります。パフォーマンスが遅いケースについてそれぞれ検討していけば改善できる場合もありますが、全般的な問題を突き止めるのは非常に難しいことです。しかし、落ち着いて、その問題が Informix Dynamic Server Ver.7.x にあるか切り分ける必要があります。問題 (OS、アプリケーション、データベース・デザイン) はこれまでも存在していたのに、お客様が Informix Dynamic Server Ver.7.x になって初めてその問題に気づいたケースもあります。

付録 A - モニター分析ツールの一覧

注 詳細はマニュアル等を参照してください

インフォミックスのモニターツール

Informix Dynamic Server Ver.7.xのパフォーマンスを監視するためのツールは多数あります。少なくとも onstat に慣れておくことは最低条件です。

onstat

エンジン内部状態に関する情報が豊富に揃っており強力な武器になります。便利なオプションには次のようなものがあります。

- d: チャンクのディスク入出力
- g iof: db領域当たりのi/oレート
- g ses: 特定のセッションについて非常に有用な情報を提供してくれます。スレッドやメモリの使用率、一時表、sql文が含まれます
- g glo: すべてのvpの状態を監視します。onlineがどれくらいのシステムを使用しているか、10秒ごとにチェックします。
- t(-t): ページ番号と行番号と共に(すべての)表領域を表示します。表ロードの監視や、一時/ソート領域の使用率を監視する場合に便利です。
- p: 先読み統計やバッファ付きi/o使用率など、多数の有用な情報を提供してくれます。バッファの設定はキャッシュ読込み(%cached)95%以上、キャッシュ書込み(%cached) 85%以上?
- g act: アクティブ・スレッドを監視します。大量のcpuリソースを消費していると思われるスレッドを突き止めるのに便利です。
- g rea: 実行キューを監視してcpuのボトルネックを突き止めます。

またテスト実行中に一定間隔で onstat -a を実行するのも良い方法です。

onperf

使用できる場合には、onperf監視ツールは、分かりやすいグラフィカル形式で多様な有益情報を提供してくれます。ディスク使用率はスキャンの詳細を表示するのに効果的で、rsamコールはセッションのアクティビティを知る手掛かりになります。

xtree

長時間実行中のdss形式の問合せを監視するのに非常に便利なツールです。問合せの実行ツリーのパスとスループットを表示してくれます。sqexplain.outファイルと併用すれば、問合せがどのように実行され、問合せ処理が現在どの段階にあるかを正確に知ることができます。ツリーのスキャン部とフィルタ部を監視すれば、異なる問合せパスの処理を正確に比較できます。

SMI表(注: SMI表の列名等の情報は予告無しに変更される場合があります)

sysmasterデータベースには、内部メモリ構造のビューのような表が入っています。onstatを通じて入手可能な情報のほとんどが管理しやすいフォーマットで含まれています。このフォーマットを中心に監視ツール一式を構築できます。有用な表には次のようなものがあります。

sysssqexplain: セッションのsql文を保持します。履歴情報を含める時に便利です。

systabnames: システムの中のすべてのパーティションにアクセスできます。部分名から実際の表名を調べるのに便利です。

sysptprofile: パーティションプロフィール情報。順次スキャンの順番の早い方から表を表示します。

UNIX分析ツール (プラットフォーム固有)

ディスクやCPU、メモリの基本的なシステム使用率を監視するには、OSユーティリティに慣れておくと便利です。OSの特色にもよりますが、sarやiostat、vmstatに加えて、標準のpsやipcsが含まれています。ネットワーク管理ツールも有効です。ベンダーの中には、topやmonitor、glanceなど、もっと素晴らしいツールを提供している所もあります。プロセスをトレースする手段もかなり有効で、特に、パフォーマンス低下の原因がシステム・コールにあると疑われる場合には役に立ちます。デバッガやtraceまたはtrussなどのユーティリティを通じて、システム・コールを数えたりシステム・コールの使用率のプロフィールを作成できることもあります。これらのツールの中にはパブリック・ドメイン・バージョンもあります。

sar

例 sar -d 5 10

長期分析の場合、間隔は5秒以上にすること (例 :60秒など)。

重要なオプション :

- sar -b バッファ・アクティビティ
- sar -c システム・コールの数
- sar -d ディスク・アクティビティ (チャンクの分散、稼働率が常に60%以上でないこと)
- sar -m メッセージおよびセマフォ・アクティビティ
- sar -q 待ち行列
- sar -u CPUの稼働率 (%idleと%wioがいずれも高い場合はI/Oに問題あり
%usr + %sysが100%に近い場合は、CPUの過負荷が考えられる。)
- sar -w コンテキスト・スイッチング、ページング

iostat

比較的長期間にわたってディスク負荷を監視するにはiostatを使用してください。必要であれば、ディスクの分散、断片化を変更してください。

vmstat

mpstat

仮想メモリやCPU、ページングの稼働率を調べるにはvmstatを使用してください。

top

このツールバーは、CPUの消費量が多いほとんどのプロセスを表示します。

プラットフォーム別のバリエーションは次のとおりです :

- glance (HP/UX)
- 3dmon (AIX)

付録B - 新旧のコマンドの比較

	Ver.5.xまで	Ver.7.x
ディスク入出力	lostat sar -d tbatat -d	lostat sar -d stat -d -D onstat -g iof, -g ioq onstat -g ppf select chunknum, fname, reads, pagesread, writes, pagewritten from syschkio c, syschunks k where c.chunknum = k.chknum; select *from sysptprof;
エクステンツ	tbcheck -pe tbcheck -pt	oncheck -pe oncheck -pt select *from sysextents;
共有メモリ	Vmstat tbatat -p tbatat -l tbatat -u	vmstat (pi, po, free, id) onstat -p (cache rate, ovlock) onstat -l (pages/io) onstat -u onstat -m (Checkpoint Intervall)
プロセス		onstat -g glo, -g rea, -g act onstat -g ses <sid> onstat -g sql <sid> select tabname, rowidlk, type,owner, username from syslocks L, sysessions S where L. owner = S.sid order by tabname, owner; (locks)

(注 : SM 表の列名等の情報は、製品の改良等により予告無しに変更される場合があります)

付録C - 移植チェックリスト

会社名： _____ 電話番号： _____

技術担当者： _____ ファックス番号： _____

ハードウェア環境

	旧	新
ハードウェアのタイプ (メーカー)		
オペレーティング・システムのバージョン		
パッチ		
プロセッサの数		
メイン・メモリ		
ハードディスク 数とサイズ DB用の数		
コントローラの数		
RAID/Volume Manager		
バックアップ・メディア		
ネットワーク		
アプリケーション・サーバ		
ミラリング (OS/DB)		
テストプロダクション環境		
特別な考慮事項		

ソフトウェア環境

	旧	新
SE		
Informix Dynamic Server		
C-ISAM		
データベースのサイズ		
表の最大レコード数		
表の数		
ESQL/C		
ESQL/COBOL		
4GL		
NewEra		
CLI		
ODBC		
I-NET		
I-STAR		
ソース・コードは存在するか？		
非インフォミックス製品		
アプリケーションはリリースされているか？		
アプリケーションはテストされているか？		

お客様の期待

(例:既存のボトルネック、時間枠、制限時間、パフォーマンス・ステータス分析、断片化やNLS、PDQなどの新機能の使用)

移植の対象範囲

(例 :アーカイブの担当者は誰か、フロントエンドまたはバックエンドの移植、新しいデータベースの設計など)

実用性、予想所要時間、対処 (担当が記入)

チェックリストの記入者 : _____ 日付 : _____

情報を確認した人 : _____ 日付 : _____

担当SE / コンサルタント: _____

特記事項

本書は米国IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本IBMの営業担当員にお尋ねください。本書でIBM 製品、プログラム、またはサービスに言及していても、そのIBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権(特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務 知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書においてIBM 以外のWeb サイトに言及している場合がありますが、便宜のため記載しただけであり決してそれらのWeb サイトを推奨するものではありません。それらのWeb サイトにある資料は、このIBM 製品の資料の一部ではありません。それらのWeb サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム(本プログラムを含む) との間での情報交換、および(ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります。単に目標を示しているものです。

表示されているIBM の価格はIBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェイスに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従ってIBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

サンプルプログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従ってIBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

サンプルプログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよびibm.com® は、世界の多くの国で登録されたInternational Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれIBM または各社の商標である場合があります。

す。現時点でのIBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、およびPostScript® は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Intel®、Itanium®、およびPentium® は、Intel Corporation または子会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。

Java およびすべてのJava 関連の商標およびロゴはSun Microsystems, Inc.の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、およびWindows NT は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX はThe Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。 _
