パフォーマンス&チューニングシリーズ

SAP Sybase IQ 16.0 SP03

ドキュメント ID： DC00283-01-1603-01
改訂： 2013 年 11 月

目次

対象読者

このドキュメントは、パスワードの向上のために SAP® Sybase® IQ を設定するデータベース管理者、データベース設計者、および開発者を対象にしています。

パフォーマンスに関する考慮事項

パフォーマンスは、通常、応答時間とスループットで測定されます。適正な設計を行い、適切なインデックス付け方式を選択することによって、パフォーマンスを最大に向上させることができます。

応答時間とは、1 つのタスクが完了するまでにかかる時間のことです。応答時間は、次の項目の影響により変化します。

- 競合の軽減と待機時間 (特にディスク I/O 待機時間) の短縮
- より高速なコンポーネントの使用
- リソースに必要な時間の短縮 (同時実行性の向上)

スループットは、一定の時間にどれだけの作業量が完了したかを表します。スループットは、通常、1 秒あたりのトランザクション数で表されますが、1 分、1 時間、1 日などの単位で測定する場合もあります。

パフォーマンスを最大に向上させるには、正しく設定されているシステムで SAP Sybase IQ を実行し、適正な設計を行い、適切なインデックス付け方式を選択します。

その他、ハードウェアやネットワークを分析することによって、インストール環境のボトルネックを特定できます。

ハードウェア設定

SAP Sybase IQ のパフォーマンスに影響を与えるハードウェアの問題について説明します。

使用可能な CPU 数の設定

-iqnumbercpus 起動スイッチを設定して、使用できる CPU 数を指定します。このパラメータは、リソース計画を目的として CPU の物理的な数を上書きします。

-iqnumbercpus スイッチは、以下でのみ使用することをおすすめします。

- Intel® CPU が搭載されている、ハイパースレッディングが有効なマシンで、**-iqnumbercpus** を、実際のコア数に設定する
- オペレーティングシステムのユーティリティを使って、SAP Sybase IQ を、マシン内の CPU のサブセットに制限しているマシン

その他の情報
『管理：データベース』の「データベースサーバの実行」>「コマンドラインスイッチ」>「パフォーマンスを制御するコマンドラインオプション」>「メモリオプション」>「CPU 数のスイッチ」

プロセススレッドモデル

SAP Sybase IQ では、最大限のパフォーマンスを得るために、オペレーティングシステムのカーネルスレッドを使用します。デフォルトでは、SAP Sybase IQ はシステム上の CPU の数に基づいたスレッド数を割り当てます。

ライトウェイトプロセスは、カーネルでサポートされるコントロールの基本となるスレッドです。オペレーティングシステムによって、どのライトウェイトプロセス (LWP) をどのプロセッサでいつ実行するかが決定されます。オペレーティングシステムはユーザスレッドのことは関知しませんが、ユーザスレッドが待機中か実行可能かは認識しています。

オペレーティングシステムのカーネルによって、LWP が CPU リソース上にスケジューリングされます。この場合、LWP のスケジューリングクラスと優先度を使用します。各 LWP は、カーネルによって個別にディスパッチされ、個別のシステム呼び出しを実行し、個別のページフォルトを発生させ、マルチプロセッサシステム上では並列に実行します。

高度にスレッド化された単一のプロセスが、すべての SAP Sybase IQ ユーザの処理を実行します。データベースサーバは、接続によって実行される処理の種類、使用可能な合計スレッド数、さまざまなオプションの設定に基づいて、それぞれ異なるカーネルスレッド数を各ユーザ接続に割り当てます。

スレッド不足エラー
クエリ処理に必要なサーバスレッドが不足している場合、SAP Sybase IQ は次のエラーを生成します。

```
Not enough server threads available for this query
```

この状況は、すぐに解消される場合もあります。他のクエリが完了してからクエリを発行すると、使用可能なスレッドが増えるため、クエリが成功する場合があります。状況が持続する場合は、サーバを再起動し、より多くの SAP Sybase IQ スレッドを指定する必要があります。接続数に対して **-iqmt** に設定されている値が小さすぎる可能性もあります。

スレッド使用を管理するための SAP Sybase IQ オプション

- 最大スレッド数を設定するには、サーバ起動オプション **-iqmt** を使用します。デフォルト値は接続数と CPU 数によって計算され、通常、デフォルト値をそのまま使用できます。
- 内部実行スレッドのスタックサイズを設定するには、サーバ起動オプション **-iqtss** を使用します。通常はデフォルト値で十分ですが、複雑なクエリを実行したときに、スタックの深さがこの制限を超えていることを示すエラーが返された場合は、値を増やします。
- ユーザ 1 人あたりに使用するスレッド数の最大値を設定するには、`SET OPTION MAX_IQ_THREADS_PER_CONNECTION` コマンドを使用します。`SET OPTION MAX_IQ_THREADS_PER_TEAM` は、スレッドのチームで使用可能なスレッド数を設定し、単一の操作が割り当てられるスレッド数 (システムリソースの量) を制限します。
- 特定の操作に使用するリソースの量を制御する場合にも、これらのオプションを使用します。たとえば、**INSERT**、**LOAD**、**BACKUP DATABASE**、または **RESTORE DATABASE** のコマンドを発行する前にこのオプションを設定できます。

ネットワークパフォーマンス

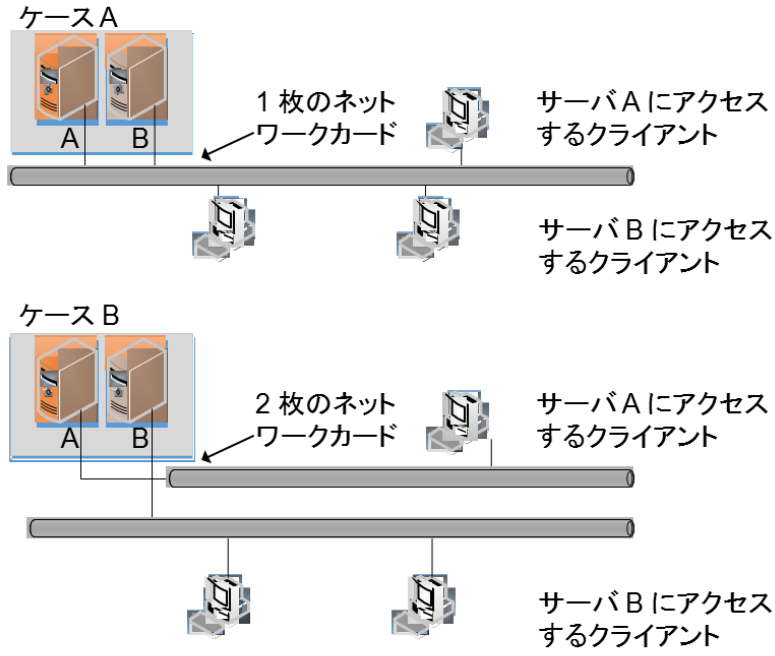
環境を少し変更するだけで、ネットワークパフォーマンスの問題を解決できることがあります。

ネットワークスループットを改善するには、複数のネットワークアダプタを用意します。サービスレベルアグリーメントに基づき、ユーザをクラス別で異なるネットワークに割り当てることができます。

ケース A (下図を参照) では、2 つの異なるデータベースサーバにアクセスするクライアントが 1 枚のネットワークカードを使用しています。このため、サーバ A とサーバ B にアクセスするクライアントは、ネットワーク上とネットワークカードで競合します。ケース B では、サーバ A にアクセスするクライアントとサーバ B にアクセスするクライアントが別々のネットワークカードを使用しています。

異なるマシンをデータベースサーバにすると、さらにパフォーマンスが向上します。異なるデータベースのヘビーユーザを異なるマシンに分けることもできます。

図 1 : ヘビーネットワークユーザの分離



少量のデータを小さなパケットに入れる
ネットワーク上で少量のデータを送信する場合は、デフォルトのネットワークパケットサイズを小さいまま使用します (デフォルトは 512 バイトです)。**-p** サーバ起動オプションは、最大パケットサイズを指定するために使用します。クライアントアプリケーションを使用してパケットサイズを設定できます。

大量のデータを大きなパケットに入れる
大量のデータを送受信するアプリケーションが多い場合は、デフォルトのネットワークパケットサイズを大きくします。転送の数は少なくなりますが、データ転送量は多くなります。

サーバレベルのプロセス
サーバレベルで、できるかぎり多くのデータをフィルタします。

サーバ設定

メモリの概要

SAP Sybase IQ がメモリを割り付ける方法を理解することは、システムで最高のパフォーマンスを実現するのに役立ちます。

サーバメモリ

SAP Sybase IQ によって、バッファ、トランザクション、データベース、およびサーバのヒープメモリが割り付けられます。共有メモリも使用できますが、非常に少量です。

オペレーティングシステムレベルでは、SAP Sybase IQ サーバメモリはヒープメモリで構成されます。ほとんどの場合、SAP Sybase IQ で使用されるメモリがヒープメモリか共有メモリかを気にする必要はありません。メモリ割り付けは、すべて自動的に処理されます。SAP Sybase IQ を実行する前に、オペレーティングシステムカーネルが共有メモリを使用するように正しく構成されていることを確認してください。

ほとんどのオペレーティングシステムは、ファイルシステムバッファリングに使用できるメモリの多くを使用します。使用するオペレーティングシステムのバッファリングポリシーを理解して、メモリの過度の割り付けを回避してください。

SAP Sybase IQ のメインバッファキャッシュとテンポラリバッファキャッシュに使用するメモリと、SAP Sybase IQ メモリオーバヘッド、オペレーティングシステムとその他のアプリケーションに使用するメモリの合計が、システムの物理メモリを超えないようにしてください。

参照：

- 必須メモリ (10 ページ)
- キャッシュメモリ (11 ページ)
- ラージメモリ (12 ページ)
- IQ ページサイズ (13 ページ)
- 連結メモリ (13 ページ)

必須メモリ

オペレーティングシステムや他のアプリケーションで使用する物理メモリ量が判明したら、SAP Sybase IQ が必要とする残りのメモリ量を計算します。

ローパーティションとファイルシステム
UNIX のようなシステムの場合、ローパーティションではなくファイルシステムを使用するデータベースには、オペレーティングシステムによるファイルバッファリング処理のために残りのメモリの 30% がさらに必要になります。Windows では、`OS_FILE_CACHE_BUFFERING = 'OFF'` (新しいデータベースのデフォルト) に設定し、ファイルシステムキャッシュを無効にしてください。

マルチユーザのデータベースアクセス
マルチユーザがデータベースをクエリする場合、SAP Sybase IQ には「アクティブ」ユーザ 1 人あたり約 10MB のメモリが必要です。アクティブユーザとは、同時にデータベースにアクセスしたり、データベースに対して問い合わせを行ったりするユーザのことです。たとえば、SAP Sybase IQ に接続しているユーザが 30 人でも 、アクティブにデータベースを同時に使用しているユーザは 10 人ほどしかいないことがあります。

スレッドスタックのメモリ
スレッドの処理には、少量のメモリが必要です。使用する処理スレッドが多くなるにつれ、必要なメモリも多くなります。**-iqmt** サーバスイッチは、スレッド数を制御します。**-iqtss** サーバスイッチと **-gss** サーバスイッチは、各スレッドに割り付けられたスタックメモリの容量を制御します。IQ スタックに割り付けられたメモリの総量は、次の式で計算された値とほぼ同じになります。 (**-gn** * (**-gss** + **-iqtss**)) + (**-iqmt** * **-iqtss**)

ユーザの数に比例して、スレッドの処理に必要なメモリは増加します。**-gn** スイッチは、データベースサーバが同時に実行できるタスクの数 (ユーザ要求とシステム要求の両方) を制御します。**-gss** スイッチは、これらのタスクを実行するサーバ実行スレッドのスタックサイズをある程度制御します。SAP Sybase IQ は、これらのワーカスレッドのスタックサイズを、次の式を使用して計算します。 (**-gss** + **-iqtss**)

スレッドの合計数 (**-iqmt** と **-gn** の合計) が、現在のプラットフォームで使用できるスレッド数を超えないようにします。

その他のメモリ使用
すべてのコマンドとトランザクションが、ある程度のメモリを使用します。これまで説明してきた要因の他に、メモリを大量に使用する操作には次のものがあります。

- バックアップ。バックアップに使用される仮想メモリの量は、データベース作成時に指定された **IQ PAGE SIZE** によって決まります。この値はおよそ 2 * CPU の数 * 20 * (IQ PAGE SIZE/16) です。プラットフォームによっては、**BACKUP DATABASE** コマンドの BLOCK FACTOR を調整するとバックアップのパフォーマンスが向上する場合がありますが、BLOCK FACTOR を増やすとメモリの使用量も増加します。
- データベースの検証と修復。データベース全体を検証すると、**sp_iqcheckdb** プロシージャは処理を開始する前に、すべてのテーブル、テーブルのそれぞれのフィールドとインデックスを開きます。テーブルの数、テーブル内のカラムとインデックスの累積数によって、**sp_iqcheckdb** に必要な仮想メモリの量は大幅に異なります。必要なメモリ量を制限するには、**sp_iqcheckdb** オプションを使用して 1 つのインデックスまたはテーブルを検証または修復します。
- リークブロックの削除。リーク削除操作でも、すべてのテーブル、ファイル、インデックスを開く必要があるため、データベース全体を検証するときに **sp_iqcheckdb** が使用するのと同じ容量の仮想メモリを使用します。テンポラリバッファキャッシュを使用して、使用ブロックを追跡します。

参照：

- サーバメモリ (9 ページ)
- キャッシュメモリ (11 ページ)
- ラージメモリ (12 ページ)
- IQ ページサイズ (13 ページ)
- 連結メモリ (13 ページ)

キャッシュメモリ

パフォーマンスを最適化するために、IQ メインバッファとテンポラリバッファのキャッシュにできるだけ多くのメモリを割り付けます。ロード、クエリ、およびアプリケーションに対応できるようにデフォルトを変更します。

メインバッファキャッシュおよびテンポラリバッファキャッシュのデフォルトのキャッシュサイズは、それぞれ 64MB です。キャッシュサイズは、サーバ起動オプションである **–iqmc** (メインバッファキャッシュ) および **–iqtc** (テンポラリキャッシュ) で制御されます。これらの起動オプションは、サーバが実行されている間だけ有効なため、サーバを再起動するたびに指定する必要があります。

ラージメモリ所要量は、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 です。IQ のメインストアとテンポラリストアに十分なメモリを確保するために、起動パラメータ **–iqlm**、**–iqmc**、および **–iqtc** の値をそれぞれ、利用可能な総物理メモリの 3 分の 1 の量に設定してください。

参照：
- サーバメモリ (9 ページ)
- 必須メモリ (10 ページ)
- ラージメモリ (12 ページ)
- IQ ページサイズ (13 ページ)
- 連結メモリ (13 ページ)

ラージメモリ

–iqlm 起動パラメータは、SAP Sybase IQ がオペレーティングシステムに対して動的に要求できるラージメモリの最大量を指定します。

一部のロード操作では、デフォルトの 2 GB よりも多くのメモリが必要になることがあります。デフォルトのメモリ量よりも多くのメモリが必要な場合は、**- iqlm** 起動オプションを使用して、SAP Sybase IQ が OS に動的に要求できるメモリの量を増やします。**–iqlm** は、サーバを起動するコマンドまたは設定ファイルの一部として使用し、スイッチとして指定します。

ラージメモリの割り付け
原則として、ラージメモリ所要量は、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 を表します。IQ のメインストアとテンポラリストアに十分なメモリを確保するために、起動パラメータ **–iqlm**、**–iqtc**、および **–iqmc** の値をそれぞれ、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 の量に設定してください。

ほとんどの場合、SAP Sybase IQ プロセスがスワップアウトされないようにするため、総物理メモリの 80% を SAP Sybase IQ に割り付けます。同じシステム上で稼働する他のプロセスを考慮して、実際のメモリ割り付けを調整してください。たとえば、コア数が 32、利用可能な総物理メモリが 128 GB のマシンでは、100 GB (総メモリ 128 GB のおよそ 80%) を SAP Sybase IQ プロセスに割り付けます。上記の原則に従って、**–iqlm**、**–iqtc**、および **–iqmc** パラメータの値をそれぞれ 33 GB に設定します。

参照：
- サーバメモリ (9 ページ)
- 必須メモリ (10 ページ)
- キャッシュメモリ (11 ページ)
- IQ ページサイズ (13 ページ)
- 連結メモリ (13 ページ)

IQ ページサイズ

IQ ページサイズとバッファキャッシュサイズは、データベースのメモリ使用とディスク I/O スループットに影響します。

SAP Sybase IQ は、ページサイズ単位で I/O を実行します。データベースを作成するときに、カタログストアと IQ ストアに別々のページサイズを指定します。テンポラリストアのページサイズは、IQ ストアと同じです。

カタログストアのページサイズは、パフォーマンスに実質的な影響を与えません。デフォルト値 4096 バイトで十分です。IQ ページサイズによって、データベースのデフォルト I/O 転送ブロックサイズと最大データ圧縮の 2 つのパフォーマンスの要因が決まります。SAP Sybase IQ は、データをすべて圧縮します。データ圧縮の量は、IQ ページサイズによって決定されます。

メモリの節約
マシンに十分なメモリが搭載されていない場合は、メモリを増やしてバッファキャッシュサイズを小さくします。 ただし、バッファキャッシュを小さくしすぎると、バッファの不足によって、データのロードまたはデータの問い合わせが非効率的になったり、完了できなくなったりすることがあります。

注意： ページサイズは変更できません。ページサイズによって、一部のデータベースオブジェクトのサイズの上限と LOB 機能を使用できるかどうかが決まります。

参照：
- サーバメモリ (9 ページ)
- 必須メモリ (10 ページ)
- キャッシュメモリ (11 ページ)
- ラージメモリ (12 ページ)
- 連結メモリ (13 ページ)

連結メモリ

HP と Solaris のプラットフォームでは、指定した量のメモリを連結メモリとして指定できます。連結メモリは、物理メモリにロックされた共有メモリです。カーネルはこのメモリを物理メモリからページアウトできません。

連結メモリプール
HP と Solaris のプラットフォームでは、指定した量のメモリを連結メモリとして指定できます。 連結メモリは、物理メモリにロックされた共有メモリです。 カーネルはこのメモリを物理メモリからページアウトできません。

他のアプリケーションが同じマシン上で同時に実行されている場合は、連結メモリによって SAP Sybase IQ のパフォーマンスが向上することがあります。ただし、連結メモリを SAP Sybase IQ 専用に割り付けると、そのメモリはマシン上の他のアプリケーションから利用できなくなります。

これらの UNIX プラットフォームにのみ連結メモリのプールを作成するには、**-iqwmem** コマンドラインスイッチを指定して、連結メモリの MB 数を指定します (Solaris 以外のプラットフォームで **-iqwmem** を設定するには、**root** ユーザの権限が必要です)。64 ビットプラットフォームでは、マシンの物理メモリのみが **-iqwmem** の上限となります。

たとえば、14GB のメモリを搭載するマシンで、10GB の連結メモリを確保するとします。そのためには、次のように指定します。

```
-iqwmem 10000
```

注意： **-iqwmem** は、連結メモリに指定する余裕がメモリにある場合にのみ使用します。メモリが十分にないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。

- Solaris では、**-iqwmem** を指定すると、常に連結メモリが有効になります。
- HP では、サーバを root ユーザで起動した場合に、**-iqwmem** を指定すると連結メモリが有効になります。root ユーザ以外のユーザでサーバを起動した場合は、非連結メモリが有効になります。この動作は、将来のバージョンで変更される可能性があります。

他のアプリケーションとデータベースの影響
サーバに使用されるメモリは、すべてのアプリケーションとデータベースに使用されるメモリプール内のメモリです。複数のサーバまたは複数のデータベースを同時に同じマシン上で実行したり、他のアプリケーションを実行したりしている場合は、サーバが要求するメモリ量を減らす必要があります。

また、UNIX コマンド `ipcs -mb` を発行して、実際のセグメント数を表示することもできます。

HP のメモリ問題のトラブルシューティング
HP-UX では、maxdsiz_64bit カーネルパラメータの値を調べます。このパラメータは、64 ビット HP プロセッサ上で SAP Sybase IQ が使用できる仮想メモリの量を制限します。推奨値については、『インストールおよび設定ガイド』を参照してください。

参照：
- サーバメモリ (9 ページ)
- 必須メモリ (10 ページ)

- キャッシュメモリ (11 ページ)
- ラージメモリ (12 ページ)
- IQ ページサイズ (13 ページ)

チューニングオプション

より高速なクエリ実行を可能にするチューニングオプションがあります。

一般的な使用のための最適化

USER_RESOURCE_RESERVATION オプションを設定して、現在のユーザ数を考慮してメモリ使用を調整します。

SAP Sybase IQ は、開いたカーソルの数を追跡して、メモリを割り付けます。特定の状況においては、USER_RESOURCE_RESERVATION オプションによって、製品を使用していると SAP Sybase IQ で思われる現在のカーソル数の最小値を調整し、テンポラリキャッシュから割り付けるメモリをさらに節約できます。

このオプションは、必須の場合にのみ設定します。このオプションを設定する場合は、保守契約を結んでいるサポートセンタに連絡してください。

参照：

- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

ユーザが多数存在する場合の最適化

ユーザが多数の場合、起動パラメータを調整します。

表 1 : サーバ起動オプション

パラメータ	説明
-gm	接続のデフォルト数を設定する。使用法: **-gm** *#_connections_to_support* この値は、サーバがサポートする接続の合計数を表すが、すべての接続が同時にアクティブなわけではない。
-iqgovern	指定すると、一度に実行されるクエリの最大数が制限される。**-iqgovern** の制限を超えるユーザがクエリを発行した場合は、アクティブなクエリのいずれかが完了するまで、新しいクエリはキューイングされる。使用法: **-iqgovern** *#_ ACTIVE_ queries_to_support* **-iqgovern** の最適な値は、クエリの性質、CPU の数、バッファキャッシュのサイズによって異なる。デフォルト値は 2**numCPU*+ 10。接続ユーザ数が多い場合は、このオプションを 2**numCPU*+ 4 に設定するとスループットが向上する場合がある。
-gn	複数のユーザが実行する場合に、カタログストアと接続の要求を処理するために使用される実行スレッドの数を設定する。使用法: **-gn** *number of tasks (both user and system requests) that the database server can execute concurrently* **-gn** の適正値は、**-gm** の値によって決まる。**start_iq** ユーティリティが **-gn** を計算し、値を適切に設定する。**-gn** の設定値が小さすぎると、サーバが正しく機能しなくなることがある。**-gn** は、480 以下に設定することが推奨される。
-c	カタログストアキャッシュサイズを設定する。使用法: **-c** *catalog_store_cache_size* カタログのキャッシュサイズは、スキーマサイズとオブジェクト数に大きく左右される。カタログストアバッファキャッシュは、カタログストアの汎用メモリプールでもある。MB 単位で指定するには、**-c** *nM* の形式を使用する。たとえば、**-c** *64M* と指定する。 ユーザが 1000 人以下の場合は、**-c** を 16MB 以上に設定します。ユーザが 2000 人以下の場合は、**-c** を 48MB (デフォルト) に設定します。

パラメータ	説明
-cl and -ch	カタログストアのキャッシュサイズの上限 (**-ch**) と下限 (**-cl**) を設定する。*-cl minimum cache size* **-ch** *maximum cache size* 標準のカタログキャッシュサイズが小さすぎる場合は、**-cl** パラメータと **-ch** パラメータを設定する。 **-c** を **-ch** または **-cl** と同じ設定ファイルまたはコマンドラインで使用しないこと。関連情報については、**-ch cache-size** オプションを参照。
-iqmt	処理スレッドの数を設定する。**-gm** 設定に対して、小さすぎる値を **-iqmt** に設定すると、スレッドが不足することがある。

注意： カタログストアのキャッシュサイズを明示的に制御するには、設定ファイル (`.cfg`) またはサーバ起動時のコマンドラインで、次のいずれかを実行する必要があります。ただし、両方は実行しないでください。

- **-c** パラメータを設定する
- **-ch** パラメータと **-cl** パラメータを使用して、カタログストアのキャッシュサイズの特定の上限と下限を設定する

参照：

- 一般的な使用のための最適化 (15 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

同時クエリの制限

-iqgovern スイッチを設定して、特定のサーバでの同時クエリ数を指定します。これは、ライセンスによって規制される接続数とは異なります。

-iqgovern には最適値があり、その値であれば、最適なスループットを実現するために正しい数の同時クエリアクセスを提供します。**-iqgovern** がこのしきい値を超

えて設定された場合は、競合またはリソース不足が発生して、すべての要求が遅くなります。

-iqgovern スイッチを指定することによって、SAP Sybase IQ はディスクへのバッファデータのページングを最適化し、メモリの過剰使用を防止できます。**-iqgovern** のデフォルト値は (2 x CPU 数) + 10 です。場合によっては、いろいろな値を試して最適な値を見つける必要があります。アクティブな接続が多数あるサイトの場合は、**-iqgovern** を多少低めに設定してみてください。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

クエリテンポラリ領域の制限

`QUERY_TEMP_SPACE_LIMIT` を設定して、クエリが拒否される前のテンポラリ領域の予測最大容量を指定します。

`QUERY_TEMP_SPACE_LIMIT` オプションを設定すると、予測されるテンポラリ領域の使用率が指定されたサイズを超える場合に、クエリが拒否されます。 デフォルトでは、クエリによるテンポラリストアの使用率に制限はありません。

SAP Sybase IQ は、クエリの解析に必要なテンポラリ領域を推定します。推定が現在の `QUERY_TEMP_SPACE_LIMIT` 設定を超えると、SAP Sybase IQ は次のエラーを返します。

```
Query rejected because it exceeds total space resource limit
```

このオプションを 0 (デフォルト) に設定すると、制限がないため、テンポラリ領域の条件によってクエリが拒否されることはありません。

接続ごとのテンポラリストアの実際の使用率を制限するには、クエリを含むすべての DML 文に `MAX_TEMP_SPACE_PER_CONNECTION` オプションを設定します。`MAX_TEMP_SPACE_PER_CONNECTION` は、文によるテンポラリストアの実際の実

行時の使用率をモニタして制限します。接続が MAX_TEMP_SPACE_PER_CONNECTION オプションで設定された割り当てを超えた場合は、エラーが返され、現在の文はロールバックされます。

参照：
- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

返されるローによるクエリの制限

QUERY_ROWS_RETURNED_LIMIT オプションの値を設定して、オプティマイザが、大量の結果セットを返す可能性のあるクエリを拒否しないようにします。

QUERY_ROWS_RETURNED_LIMIT オプションを設定すると、クエリオプティマイザは、大量のリソースを消費する可能性のあるクエリを拒否します。 クエリからの結果セットがこのオプションの値を超えると推定される場合、クエリオプティマイザはクエリを拒否し、次のメッセージが表示されます。

```
Query rejected because it exceed resource: Query_Rows_Returned_Limit
```

このオプションは、大量のリソースを消費するクエリのみを拒否するように設定します。

参照：
- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)

- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

カーソルのスクロールの禁止

大量の結果セットを返すクエリのテンポラリストアノードを除去して、パフォーマンスを向上させます。

ホスト変数を宣言せずにカーソルのスクロールを使用すると、SAP Sybase IQ は、クエリ結果をバッファするテンポラリストアノードを作成します。これは、テンポラリストアバッファキャッシュとは異なります。テンポラリストアノードを使用すると、アプリケーションが結果セットを検索するときに前方および後方に効率的にスクロールできます。

ただし、クエリが出力に大量 (数百万) のローを返す場合、およびアプリケーションによって行われるスクロールのほとんどが前方スクロールの場合は、テンポラリストアノードのメモリ要件によってクエリのパフォーマンスが低下する可能性があります。パフォーマンスを向上させるには、次のコマンドを発行してテンポラリストアノードを除去します。

```
SET TEMPORARY OPTION FORCE_NO_SCROLL_CURSORS = 'ON'
```

注意： アプリケーションが後方スクロールを行うことが多い場合、FORCE_NO_SCROLL_CURSORS オプションを ON に設定すると、クエリのパフォーマンスが実際に低下することがあります。これは、テンポラリキャッシュが存在しないため、後方スクロールのたびに SAP Sybase IQ によるクエリの再実行が強制されるからです。

アプリケーションで後方スクロールがほとんど行われない場合は、FORCE_NO_SCROLL_CURSORS = 'ON' を永続的な PUBLIC オプションに設定します。メモリの節約になるため、クエリのパフォーマンスが向上します。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)

- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

カーソル数の制限

MAX_CURSOR_COUNT オプションを設定して、単一の接続が、使用できるメモリまたは CPU のリソースを大量に使用しないようにします。

MAX_CURSOR_COUNT オプションは、1 つの接続が同時に使用できるカーソルの最大数を制限します。デフォルトの値は 50 です。このオプションを 0 に設定すると、カーソル数は無制限になります。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

文の数の制限

MAX_STATEMENT_COUNT オプションを設定して、1 つの接続が作成できる準備文の数を制限します。

MAX_STATEMENT_COUNT オプションは、1 つの接続が同時に使用できる準備文の最大数を制限します。デフォルト数 (50) を超える準備文を任意の接続に対して同時にサポートする必要がある場合は、MAX_STATEMENT_COUNT オプションをより大きな値に設定できます。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)

- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

キャッシュページのプリフェッチ

BT_PREFETCH_MAX_MISS オプションを設定して、プリフェッチメモリの動作を制御します。

BT_PREFETCH_MAX_MISS オプションは、特定のクエリでページのプリフェッチを継続するかどうかを決定します。HG インデックスを使用するクエリの実行速度が予想より遅い場合は、このオプションの値を徐々に増やしてみます。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

プリフェッチされるローの数の制御

PrefetchRows パラメータと PrefetchBuffer パラメータを設定して、特定の条件下でカーソルのパフォーマンスを向上させます。これは、ODBC 接続ダイアログ上、または .odbc.ini ファイル内に設定できるクライアントオプションです。

プリフェッチは、相対位置 1 または相対位置 0 のみをフェッチするカーソルのパフォーマンスを向上させます。2 つの接続パラメータを使用して、カーソルプリ

フェッチのデフォルトを変更できます。PrefetchRows (PROWS) は、プリフェッチされるローの数を設定します。PrefetchBuffer (PBUF) は、プリフェッチされたローを格納するために、この接続で使用できるメモリを設定します。プリフェッチするローの数を増やすと、次の特定の条件ではパフォーマンスが向上する可能性があります。

- アプリケーションが数回の絶対フェッチで数多くのロー (数百ロー以上) をフェッチする場合
- アプリケーションがローを大量にフェッチし、かつ、クライアントとサーバが同じマシン上にあるか高速ネットワークで接続されている場合
- クライアント／サーバ通信がダイヤルアップリンクやワイドエリアネットワークなどの低速ネットワークで行われている場合

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)
- キャッシュパーティションの最適化 (25 ページ)

ファイルシステムバッファリングの制御

一部のファイルシステムでは、ファイルシステムバッファリングのオンとオフを切り替えることができます。ファイルシステムバッファリングをオフにすると、通常、ページングが減り、パフォーマンスが向上します。

既存のデータベースの IQ メイン DB 領域のファイルシステムバッファリングを無効にするには、次の文を発行します。

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING = OFF
```

既存のデータベースの IQ テンポラリ DB 領域のファイルシステムバッファリングを無効にするには、次の文を発行します。

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING_TEMPDB = OFF
```

このオプションは、PUBLIC ロールにのみ設定できます。変更を有効にするには、データベースを停止し、再起動します。

マルチプレックスデータベースは、ダイレクト I/O ファイルシステムデバイスをサポートしていません。マルチプレックス機能には、共有ロー記憶領域が必要だからです。ダイレクト I/O パフォーマンスオプションは、シンプレックスデータベースに対してのみサポートされています。

このダイレクト I/O パフォーマンスオプションは、Solaris UFS、Linux、Linux IBM、AIX、Windows ファイルシステムでのみ有効です。 このオプションは HP-UX と HP-UXi には影響しません。また、ローディスク上に構築されたデータベースにも影響はありません。 Linux では、ダイレクト I/O はカーネルバージョン 2.6.x でサポートされます。

Linux カーネルバージョン 2.6 および AIX でダイレクト I/O を有効にするには、環境変数 IQ_USE_DIRECTIO を 1 に設定します。Linux カーネルバージョン 2.6 および AIX では、ダイレクト I/O はデフォルトで無効になっています。
IQ_USE_DIRECTIO は、Solaris と Windows には影響しません。

注意：

- SAP Sybase IQ は、Linux カーネルバージョン 2.4 でダイレクト I/O をサポートしていません。Linux カーネルバージョンで `IQ_USE_DIRECTIO` 環境変数を設定すると、SAP Sybase IQ サーバは起動しません。エラー "`Error: Invalid Block I/O argument, maybe <pathname> is a directory, or it exceeds maximum file size limit for the platform, or trying to use Direct IO on unsupported OS`" が報告されます。

  LOAD TABLE 入力ファイルがデータファイルに対するシンボリックリンクである場合は、Linux カーネルバージョン 2.6 で、同じエラーが発生してダイレクト I/O が失敗することがあります。
- Solaris には、ファイルシステムバッファキャッシュのサイズを制限するカーネルパラメータはありません。時間の経過とともにファイルシステムバッファキャッシュが大きくなり、バッファキャッシュページを置き換えるため、オペレーティングシステムに過度のページングアクティビティが発生し、パフォーマンスが低下します。Solaris では、可能であればデータベースにローデバイスを使用します。
- Windows では、ファイルシステムよりもアプリケーションを優先させるページングアルゴリズムを使用できます。これは、SAP Sybase IQ パフォーマンスの向上に役立ちます。

参照：

- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)

キャッシュパーティションの最適化

CACHE_PARTITIONS 値を変更すると、マルチ CPU 構成でロードまたはクエリのパフォーマンスが向上することがあります。

SAP Sybase IQ では、システム上の CPU の数に応じて、バッファキャッシュのキャッシュパーティションの数が自動的に計算されます。マルチ CPU 構成でロードまたはクエリのパフォーマンスが予想より悪い場合は、CACHE_PARTITIONS データベースオプションの値を変更するとパフォーマンスが向上することがあります。

バッファは、キャッシュの LRU (Least Recently Used) 側の終端に近づくと、ウォッシュマーカを越えます。SAP Sybase IQ は最も古いページ (ウォッシュマーカを越えたページ) をディスクに書き出して、そのページが占有していたキャッシュ領域を再利用できるようにします。スイーパスレッドと呼ばれる SAP Sybase IQ 処理スレッドのチームが、最も古いバッファを一掃し (書き出し) ます。

データのページをキャッシュに読み込む必要がある場合、SAP Sybase IQ は LRU バッファを取り込みます。バッファがまだ「ダーティな」(変更された) 状態の場合は、先にバッファをディスクに書き込む必要があります。モニタの **-cache** レポートの [Gdirty] カラムは、LRU バッファをダーティな状態で取り込んだために、SAP Sybase IQ がそのバッファを使用する前に書き出す必要があった回数を示します。

通常、SAP Sybase IQ では [Gdirty] の値が 0 に維持されます。この値が 0 より大きい状態が長時間続く場合は、スイーパスレッドの数とウォッシュマーカを制御するデータベースオプションを調整する必要があります。

その他の情報

- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「CACHE_PARTITIONS オプション」
- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「SWEEPER_THREADS_PERCENT オプション」

- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「WASH_AREA_BUFFERS_PERCENT オプション」

参照：
- 一般的な使用のための最適化 (15 ページ)
- ユーザが多数存在する場合の最適化 (16 ページ)
- 同時クエリの制限 (17 ページ)
- クエリテンポラリ領域の制限 (18 ページ)
- 返されるローによるクエリの制限 (19 ページ)
- カーソルのスクロールの禁止 (20 ページ)
- カーソル数の制限 (21 ページ)
- 文の数の制限 (21 ページ)
- キャッシュページのプリフェッチ (22 ページ)
- プリフェッチされるローの数の制御 (22 ページ)
- ファイルシステムバッファリングの制御 (23 ページ)

入出力の分散

ディスクストライピング、ランダムファイルディスクアクセス、および順次ファイルディスクアクセスを使用して、入出力 (I/O) を分散します。

ローデバイス

UNIX のようなオペレーティングシステムでは、データベースまたは DB 領域をローデバイスまたはファイルシステムファイルに作成できます。

ディスクパーティションは、通常、ファイルシステムモード (たとえば、UFS ファイルシステムを通して) またはローモードの 2 つのモードでアクセスされます。ローモードではバッファを使用しない I/O を行い、通常、読み取りまたは書き込みのシステム呼び出しごとにデバイスに対するデータ転送を行います。UNIX のデフォルトのファイルシステムである UFS は、バッファを使用する I/O システムであり、バッファにデータを蓄積してからバッファ全体を一度に転送します。

データベースまたは DB 領域をローデバイスまたはファイルシステムファイルに作成できます。SAP Sybase IQ は指定されたパス名から、それがローパーティションかファイルシステムファイルかを自動的に判断します。ローパーティションは任意のサイズに設定できます。

参照：

- ディスクストライピング (27 ページ)
- 内部ストライピング (28 ページ)
- ランダムファイルアクセスおよび順次ファイルアクセス (29 ページ)
- トランザクションログとメッセージログ (30 ページ)

ディスクストライピング

複数のディスクにわたるデータのストライピングは、優れたパフォーマンスを得るために重要な手法です。

ディスクストライピングは、システム内のさまざまな場所で実行することができ、RAID ハードウェアまたはソフトウェアの一部として実行されることがよくあります。以下に例を示します。

- ディスクアレイやコントローラなどのデバイスレイヤで実行する。
- オペレーティングシステムや、Veritas などの専用デバイス管理ソフトウェアで実行する。
- アプリケーションで実行する。

デフォルトでは、SAP Sybase IQ は、DB 領域内のすべてのファイルにわたってページを内部的にストライプするので、パフォーマンス向上のためにソフトウェアレベルまたはハードウェアレベルでストライピングを追加する必要はありません。もちろん、データベースへの記憶域冗長性の実装の一環として、追加のストライピングが必要になることもあります。たとえば、RAID-5 が使用されている場合です。

記憶域冗長性を備えた SAP Sybase IQ で最高のパフォーマンスを実現するには、簡易ミラーリングまたは "RAID-1" を使用します。上記のように、SAP Sybase IQ は、DB 領域内の 2 ディスクミラーセットのすべてにわたってデータを分散します。

ほとんどの SAP Sybase IQ データベースは、コストが原因でミラーリングを使用せず、冗長性を実現するためには、RAID-5 または同様の RAID レベルで実装されます。RAID-5 では、適切なチャンクサイズ (1 枚のディスクに対し、次のディスクに移る前にどれだけのデータが書き込まれるか) を選択することが、システムにパフォーマンス上の大きな影響を与えます。RAID-5 では、書き込みのオーバヘッドが大きいからです。アプリケーションで、頻度または緊急性の高いロード、更新、または削除を行う場合、またはテンポラリ DB 領域 I/O にクエリが頻繁に行われる場合は、チャンクサイズを小さくして、SAP Sybase IQ データベースページのサイズの 25-50% の範囲にすると、最高のパフォーマンスが得られる可能性が高いと思われます。アプリケーションで行われるのがほとんど読み込みで、書き込みアクティビティはほとんど行われないのであれば、チャンクサイズを大きくして、SAP Sybase IQ ページサイズの 75 ～ 100% の範囲にすると、最高のパフォーマンスが得られる可能性が高いと思われます。

SAP Sybase IQ は通常、複数の読み込みをプリフェッチしようとするか、または複数の書き込みを並列的にフラッシュしようとするので、たとえアクティブなクエリが 1 つだけの場合でも、小さなチャンクサイズを使用して各ページの読み込みまたは書き込みを数多くのディスクに分散することには、利点はほとんどなく、パフォーマンスが損なわれるのが普通です。

RAID を使用している場合、最高のパフォーマンスは通常、ハードウェア (コントローラや配列など) ベースの RAID を使用することによって実現されます。ソフトウェアベースの RAID ツールもうまく動作しますが、サーバの CPU に余分なパフォーマンスロードが少し追加される場合があります。

参照：
- ローデバイス (26 ページ)
- 内部ストライピング (28 ページ)
- ランダムファイルアクセスおよび順次ファイルアクセス (29 ページ)
- トランザクションログとメッセージログ (30 ページ)

内部ストライピング

ディスクストライピングでは、複数のディスクスピンドルを使用して高速な並列ディスク書き込みを行います。

SAP Sybase IQ では、サードパーティ製のソフトウェアを使用せずにディスクストライピングを可能にするオプションが用意されています。サードパーティ製のソフトウェアとハードウェアによるディスクストライピングを使用している場合は、次の説明に従う必要はありません。`CREATE DBSPACE` コマンドに `STRIPING ON` オプションを指定することにより、ディスクストライピングを有効にできます。

DB 領域の作成時にデフォルトのストライピングを変更するために使用する構文は、次のとおりです。

```
SET OPTION "PUBLIC".DEFAULT_DISK_STRIPING = { ON | OFF }
```

すべてのプラットフォームで `DEFAULT_DISK_STRIPING` オプションのデフォルト値は `ON` です。ディスクストライピングが `ON` の場合、入力データは、使用可能な領域があるすべての DB 領域に分散されます。ディスクストライピングが `OFF` の場合は、論理ファイルの先頭から DB 領域 (ディスクセグメント) に格納され、一度に 1 つのディスクセグメントが格納されます。

`DEFAULT_DISK_STRIPING` の値を変更すると、ストライピングの優先を指定しないすべての後続の `CREATE DBSPACE` 操作に影響を与えます。

ディスクストライピングが ON の場合、`ALTER DBSPACE DROP` コマンドを使用して DB 領域からファイルを削除できます。ただし、DB 領域を削除する前に `sp_iqemptyfile` ストアドプロシージャを使用して、DB 領域内のすべての

データを再配置します。ディスクストライピングでは、データが複数のファイルに分散されるため、sp_iqemptyfile プロセスには、多数のテーブルとインデックスの再配置が必要になることがあります。sp_iqdbspaceinfo ストアドプロシージャと sp_iqdbspace ストアドプロシージャを使用して、DB 領域上に存在するテーブルとインデックスを確認します。

参照：

- ローデバイス (26 ページ)
- ディスクストライピング (27 ページ)
- ランダムファイルアクセスおよび順次ファイルアクセス (29 ページ)
- トランザクションログとメッセージログ (30 ページ)

ランダムファイルアクセスおよび順次ファイルアクセス

ランダムアクセスファイル専用のディスクドライブ数、およびこれらのファイルに対して実行される 1 秒あたりの操作数を増やすことによって、ランダムアクセスファイルに関連するパフォーマンスを向上させることができます。

ランダムファイルには、IQ ストア、テンポラリストア、カタログストア、プログラム (SAP Sybase IQ 実行ファイル、ユーザおよびストアドプロシージャ、アプリケーションなど)、オペレーティングシステムファイルのランダムファイルがあります。

一方、順次アクセスファイルに関連するパフォーマンスは、専用ディスクドライブに格納し、他のプロセスとの競合をなくすことによって向上させることができます。順次ファイルには、トランザクションログやメッセージログファイルがあります。

ボトルネックを防止するには:

- ランダムディスク I/O を順次ディスク I/O から分離する。また、パフォーマンスを最大にするために、DB 領域ごとに 1 つの物理デバイス (ディスクまたは HW RAID セット) から 1 つのパーティションのみを使用する。
- 他のデータベースまたは他の I/O 集約型アプリケーションの I/O から SAP Sybase IQ データベース I/O を分離する。
- データベースファイル、テンポラリ DB 領域、トランザクションログファイルをデータベースサーバと同じ物理マシン上に配置する。

参照：

- ローデバイス (26 ページ)
- ディスクストライピング (27 ページ)
- 内部ストライピング (28 ページ)
- トランザクションログとメッセージログ (30 ページ)

トランザクションログとメッセージログ

トランザクションログおよびメッセージログのサイズを管理して、ディスク領域を節約します。

トランザクションログファイルには、リカバリ情報と監査情報が含まれています。データベースファイルの断片化を防ぎ、メディア障害から保護するには、トランザクションログを、データベースそのものとは別のデバイスまたはパーティションに保管してください。

トランザクションログファイルでは、時間の経過に伴って、大容量のディスク領域が消費されることがあります。トランザクションログを定期的にトランケートして、ディスク領域を節約します。

ログをトランケートするには:

1. サーバを停止します。
2. **start_iq** コマンドまたは `.cfg` ファイルの一部として **–m** パラメータを使用して、サーバを起動します。
3. サーバを停止し、**–m** パラメータを使用しないで再起動します。

–m スイッチを永続的に設定したままにしないでください。**–m** を設定すると、データベースファイルを含むデバイスのメディア障害に対して無防備な状態になります。**–m** は、サーバの再起動後に `.cfg` から削除します。トランザクションログファイルを移動したり、ファイル名を変更したりするには、トランザクションログユーティリティ (**dblog**) を使用します。

警告！ SAP Sybase IQ のトランザクションログファイルは、多くのリレーショナルデータベースのトランザクションログファイルとは異なります。なんらかの理由で (ログファイルではなく) データベースファイルが失われた場合は、データベースが失われます。ただし、バックアップを正しく実行している場合は、データベースを再ロードできます。

メッセージログ

SAP Sybase IQ はエラー、状態、挿入通知の各メッセージを含むすべてのメッセージをメッセージログファイルに記録します。このファイルのサイズを制限して、ディスク領域を節約します。

サイトによっては、メッセージログファイルが急速に増大することがあります。このファイルのサイズを制限するには:

- ファイルの最大サイズを設定し、アクティブなメッセージログがいっぱいになったときにログファイルをアーカイブする
- `NOTIFY_MODULUS` データベースオプションの設定を増やす

- LOAD TABLE、INSERT、CREATE INDEX の各文で NOTIFY パラメータを使用して、通知メッセージを OFF に設定する
- **-iqmsgsz** スイッチを使用して、メッセージログのサイズを制限する

その他の情報

- 『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」>「start_iq サーバオプション」>「-iqmsgsz iqsrv16 サーバオプション」
- 『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」>「start_iq サーバオプション」>「-m iqsrv16 サーバオプション」
- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「NOTIFY_MODULUS オプション」
- 『リファレンス：文とオプション』の「SQL 文」>「CREATE INDEX 文」
- 『リファレンス：文とオプション』の「SQL 文」>「INSERT 文」
- 『リファレンス：文とオプション』の「SQL 文」>「LOAD 文」

.

参照：
- ローデバイス (26 ページ)
- ディスクストライピング (27 ページ)
- 内部ストライピング (28 ページ)
- ランダムファイルアクセスおよび順次ファイルアクセス (29 ページ)

パフォーマンスのモニタリング

使用可能なリソースをシステムが最大限に利用しているかどうかを確認するために使用できるツールです。

データベースプロファイリングプロシージャ

データベース使用量統計を返すストアドプロシージャです。

表 2 : データベースプロファイリングプロシージャ

名前	説明
sp_iqconnection	接続およびバージョンについての情報を表示する。この情報には、テンポラリ DB 領域を使用しているユーザ、バージョンを有効にしているユーザ、各接続が SAP Sybase IQ 内で行っている作業、接続ステータス、データベースバージョンステータスなどが含まれる。使用法: `sp_iqconnection [ connhandle ]` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqconnection プロシージャ」を参照。
sp_iqcontext	接続ごとに、現在実行されている文に関する情報を追跡して表示する。使用法: `sp_iqcontext [ connhandle ]` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqcontext プロシージャ」を参照。
sp_iqcheckdb	現在のデータベースの妥当性を確認する。オプションで、DB 領域またはデータベースの割り付けの問題を解決する。使用法: `sp_iqcheckdb 'mode target [ … ] [ resources resource-percent ]'` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqcheckdb プロシージャ」を参照。
sp_iqdbstatistics	最後に実行された **sp_iqcheckdb** の結果をレポートする。使用法: `sp_iqdbstatistics` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqdbstatistics プロシージャ」を参照。
sp_iqdbsize	現在のデータベースのサイズを表示する。使用法: `sp_iqdbsize([ main ] )` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqdbsize プロシージャ」を参照。

名前	説明
sp_iqspaceinfo	データベース内の各オブジェクトによる領域の使用状況を表示する。使用法: `sp_iqspaceinfo [‘main	[table table-name	index index-name] [...] ‘]` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqspaceinfo プロシージャ」を参照。
sp_iqstatus	データベースのその他のステータス情報を表示する。使用法: `sp_iqstatus` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqstatus プロシージャ」を参照。
sp_iqtablesize	現在のデータベース内の各オブジェクトが使用しているブロック数と、オブジェクトが置かれている DB 領域の名前を表示する。使用法: `sp_iqtablesize ( table_owner.table_name )` 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「システムストアドプロシージャ」>「sp_iqtablesize プロシージャ」を参照。

参照：

- イベントプロファイリングプロシージャ (33 ページ)
- 主要パフォーマンス指標 (34 ページ)
- バッファキャッシュのパフォーマンス (36 ページ)

イベントプロファイリングプロシージャ

イベントプロファイリングプロシージャは、ストアドプロシージャ、関数、イベント、トリガのパフォーマンス統計を返します。

表 3 : イベントプロファイリングプロシージャ

名前	説明
sa_server_option	Interactive SQL でデータベースのプロファイリングオプションを設定する。使用法: `CALL sa_server_option ( 'procedureprofiling', 'ON')`

名前	説明

sa_procedure_profile	データベースプロシージャ、関数、イベント、またはトリガ内の各行について、実行時間を返す。使用法:
	`sa_procedure_profile`
	`[ filename`
	`[, save_to_file ] ] )`

sa_procedure_profile_summary	プロシージャ、関数、イベント、トリガすべての実行時間の要約を示す。使用法:
	`sa_procedure_profile_summary`
	`[ filename`
	`[, save_to_file ] ] )`

その他の情報
『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「カタログストアドプロシージャのアルファベット順リスト」

- sa_server_option システムプロシージャ
- sa_procedure_profile システムプロシージャ
- sa_procedure_profile_summary システムプロシージャ

参照：
- データベースプロファイリングプロシージャ (32 ページ)
- 主要パフォーマンス指標 (34 ページ)
- バッファキャッシュのパフォーマンス (36 ページ)

主要パフォーマンス指標

SAP Control Center 内に統計の集合を設定して、サーバ上の主要パフォーマンス指標 (KPI) をモニタリングします。KPI の値は、グループ化されて集合となり、SCC のモニタに表示されます。

主要パフォーマンス領域には、SAP Sybase IQ サーバ、マルチプレックスサーバ、および論理サーバが含まれています。

SAP Sybase IQ のサーバ統計
さまざまなサーバステータスおよび使用量統計です。

表 4 : SAP Sybase IQ のサーバ統計

主要パフォーマンス領域	使用量統計
SAP Sybase IQ の可用性統計	リソースの状態、CPU 使用量、メモリ割り付け、キャッシュ使用、およびアクティブな接続。
概要統計	サーバステータス、CPU 使用量、メモリ割り付け、およびアクティブな接続。
接続統計	アクティブ、使用可能、再開、ロールバック、サスペンド中のユーザおよびノード間の各接続。また、1 分あたりの接続および接続切断の平均数も表示する。
DB 領域および DB 領域ファイルの統計	DB 領域および DB 領域ファイルのサイズと使用可能なパーセンテージ。
ストア入出力統計	ストア入出力統計は、1 秒あたりのストアの読み込みおよび書き込みの回数を特定する。
キャッシュ統計	メインバッファ、カタログ、およびテンポラリキャッシュの統計。
操作および要求の統計	待機中操作、アクティブ操作、および合計操作の平均、最小、最大、および合計。
ネットワーク統計	ネットワーク統計は、ネットワークのアクティビティを示す。
トランザクション統計	サーバ上に現在あるトランザクションの詳細。

マルチプレックス、およびノード関連統計
マルチプレックス、およびマルチプレックスサーバのノード関連の統計です。

表 5 : マルチプレックス、およびノード関連統計

主要パフォーマンス領域	
マルチプレックスの可用性	マルチプレックスノード別の可用性統計。
マルチプレックスステータス	マルチプレックスのステータス。
マルチプレックスノードのプロパティ	各マルチプレックスノードのロール、ステータス、およびフェイルオーバ状態。
マルチプレックスリンクの可用性	セカンダリノードとコーディネータの間のノード間通信。

論理サーバ統計
論理サーバ、および論理サーバのノード関連の統計です。

表 6 : 論理サーバ統計

主要パフォーマンス領域	使用量統計
論理サーバの可用性	論理サーバのステータス。
論理サーバエンジン統計	CPU 使用量、接続、および接続統計。
論理サーバ接続	使用可能な接続の数の平均、最小、最大、および合計。
論理サーバトランザクション	トランザクションの平均、最小、最大、および合計。また、ロードトランザクションの平均数および最小数も表示する。
論理サーバキャッシュ統計	カタログバッファ、テンポラリバッファ、およびメインバッファのキャッシュの平均、最小、および最大のキャッシュ使用統計。
論理サーバの操作および要求の統計	待機中操作およびアクティブ操作の平均、最小、最大、および合計。

その他の情報
詳細については、SCC またはサイト http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680 の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

参照：
- データベースプロファイリングプロシージャ (32 ページ)
- イベントプロファイリングプロシージャ (33 ページ)
- バッファキャッシュのパフォーマンス (36 ページ)

バッファキャッシュのパフォーマンス

バッファキャッシュのパフォーマンスは、全体的なパフォーマンスにとって重要な要因です。IQ UTILITIES 文は、バッファキャッシュの統計を収集するキャッシュモニタを起動します。キャッシュモニタからの出力を使用して、メインバッファキャッシュとテンポラリバッファキャッシュのメモリ割り付けを細かく調整します。

このチェックリストを確認して、通常の範囲を超えているキャッシュ動作を切り離します。

表 7 : バッファキャッシュモニタのチェックリスト

統計	正常な動作	調整が必要な動作	推奨される対応策
HR% (Cache hit rate)	90% 以上。 GARRAY、BARRAY、Bitmap (bm)、hash object、sort object、variable-length btree (btreev)、fixed-length btree (btreef)、bit vector (bv)、dbext、dbid、vdo、store、checkpoint block (ckpt) などの個別の内部データ構造体の場合、クエリの実行中はヒット率が 90% を上回る。最初は 90% を下回る場合がある。プリフェッチが機能し始めると (PF または PrefetchReqs > 0)、ヒット率が徐々に上昇して 90% を超える。	プリフェッチが機能した後もヒット率が 90% を下回る。 **注意 :** オブジェクトによってはプリフェッチを行わないものがあるため、これらのヒット率は一般に低くなります。	**-iqmc** と **-iqtc** を調整し、メインとテンポラリのキャッシュサイズのバランスをとり直してみる。 `PREFETCH_THREADS_PERCENT` オプションを調整し、プリフェッチスレッド数を増やしてみる。

統計	正常な動作	調整が必要な動作	推奨される対応策
Gdirty (Grabbed Dirty)	適度なキャッシュサイズ (10GB 未満) が設定されたシステムでは 0。	GDirty > 0 **注意：** スイーパスレッドがアクティブになるのは、ダーティページの数がウォッシュエリアの一定の割合に達した場合に限ります。GDirty/GrabbedDirty が 0 より大きく、I/O 率 (Writes) が低い場合は、単にシステムに軽い負荷がかかっていると考えられるため対応策は不要です。	SWEEPER_THREADS_PERCENT オプション (デフォルトは 10%) または WASH_AREA_ BUFFERS_ PERCENT オプション (デフォルトは 20%) を調整し、ウォッシュエリアのサイズを増やす。
BWaits (Buffer Busy Waits)	0	> 0 の状態が持続し、複数のジョブが同じバッファで衝突していることを示している。	I/O 率 (Writes) が高い場合は、キャッシュのスラッシングが原因で Busy Waits が起きていると考えられる。キャッシュレポートでヒット率を調べて、メインとテンポラリのキャッシュのバランスをとり直す必要があるかどうかを確認する。 メインキャッシュとテンポラリキャッシュのうち、BWaits が一貫して 0 を上回っているほうに、さらにメモリを割り付ける。 ほぼ同一の多数のクエリを同時にバッチジョブで開始している場合は、開始時刻をずらしてみる。ほとんど同一のクエリには、INSERT、UPDATE、DELETE、SELECT など、データに触れてバッファを使用するものが含まれる。

統計	正常な動作	調整が必要な動作	推奨される対応策
LRU Waits (デバッグレポートでは、LRUNum TimeOuts percentage)	20% 以下	> 20%。これは重大な競合問題が起きていることを示す。	オペレーティングシステムのパッチレベルやその他の環境設定を確認する。これはオペレーティングシステムの問題の可能性が高い。
IOWait (IONum-Waits)	10% 以下	> 10%	ディスクエラーや I/O リトライを調べる。
FLWait (FLMu-texWaits)	20% 以下	> 20%	DB 領域の設定を確認する。 データベース領域が不足しかかっていないか？ DISK_STRIPING が ON になっているか？ **sp_iqcheckdb** が 15% を超える断片化を報告していないか？
HTWait (BmapHTNum-Waits) MemWts (Mem-NtimesWaited) (PFMgrCond-VarWaits)	10% 以下	> 10%	Sybase 製品の保守契約を結んでいるサポートセンタに問い合わせる。

統計	正常な動作	調整が必要な動作	推奨される対応策
CPU time (デバッグレポートでは CPU Sys Seconds、CPU Total Seconds)	CPU Sys Seconds < 20%	CPU Sys Seconds > 20% CPU Total Seconds も低い使用率を示しており、システムがビジー状態になるだけのジョブがある場合は、キャッシュがスラッシングしているか、並列度が失われていると考えられる。	**-iqgovern** を調整し、実行できる同時クエリの合計数を減らす。 キャッシュレポートでヒット率と I/O 率を調べて、キャッシュがスラッシングしていないかどうかを確認する。また、cache_by_type (またはデバッグ) レポートでハッシュオブジェクトのヒット率を調べて、ハッシュオブジェクトがスラッシングしていないかどうかを確認する：ヒット率が 90% 未満で I/O 率 (Writes) が高くないか？ 試行された並列処理をクエリプランで確認する。十分なスレッドが使用可能だったか？ システムに大量の CPU が搭載されているか？マルチプレックス構成などの対策が必要な場合もある。
InUse% (Buffers in use)	起動時以外は 100% かそれに近い値	100% 未満	バッファキャッシュが大きすぎる可能性がある。 **-iqmc** と **-iqtc** を調整し、メインとテンポラリのキャッシュサイズのバランスをとり直してみる。
Pin% (Pinned buffers)	< 90%	> 90 ～ 95%。システムがバッファ不足状態に危険なほど近づいていることを示す。この状態になるとトランザクションがロールバックされる。	メインとテンポラリのキャッシュサイズのバランスをとり直してみる。 バッファキャッシュサイズのバランスをとり直すことができない場合は、**-iqgovern** を減らして、同時に実行されるジョブの数を制限してみる。

統計	正常な動作	調整が必要な動作	推奨される対応策
Free threads (ThrNumFree)	Free > Resrvd	空きスレッドの数が予約済みの数まで減少している場合は、システムのスレッドが不足していると考えられる。	次のいずれかを試してみる。 **-iqmt** を設定してスレッドの数を増やす。 スレッド関連の次のオプションの値を減らす：MAX_IQ_THREADS_ PER_CONNECTION, MAX_IQ_THREADS_ PER_TEAM. USER_RESOURCE_ RESERVATION を設定し、クエリエンジンのリソース割り付けを制限する。 **-iqgovern** を設定し、ジョブの数を制限する。
FlOutOfSpace (デバッグのみ)	0。このストアのフリーリストが満杯でないことを示し、割り付けられていないページが使用可能。	1。このストア (メインまたはテンポラリ) が全部割り付けられていることを示す。	ストアに DB 領域を追加する。

注意： あるキャッシュが他のキャッシュよりもかなり多くの I/O を実行している場合、キャッシュ割り付けの 10% ほどの少量のメモリをそのキャッシュに再割り当てします (増やします)。再割り付けが終了したら、負荷を再実行し、パフォーマンス上の変化をモニタリングします。問題が継続する場合は、メモリの再割り付けを繰り返します。

その他の情報
『リファレンス：文とオプション』の「SQL 文」>「IQ UTILITIES 文」

参照：
- データベースプロファイリングプロシージャ (32 ページ)
- イベントプロファイリングプロシージャ (33 ページ)
- 主要パフォーマンス指標 (34 ページ)

マルチプレックスパフォーマンス

パフォーマンスを向上させ、ディスク領域をさらに有効に活用するためにシステムを調整します。

マルチプレックスの各サーバは、独自のホスト上にある場合と、ホストを他のサーバと共有している場合があります。複数のサーバが同じシステム上にある場合、作業負荷の処理にかかる CPU 時間は、単一の組み合わされたサーバの場合とほとんど変わりません。しかし、独立した複数のサーバでは、単一の組み合わされたサーバより多くの物理メモリが必要になります。これは、各サーバが使用するメモリを他のサーバが共有できないからです。

マルチプレックスディスク領域の管理

現在のトランザクションで定期的にコミットを発行することにより、書き込みサーバで古いバージョンのテーブルが削除され、ディスクブロックが解放されます。`auto_commit` オプションを指定すると、バージョンの蓄積が最小限に抑えられるため、領域を最小限に抑えることができます。

ユーザがいずれかのサーバで、古いバージョンのテーブルを必要とするトランザクションを実行している間は、SAP Sybase IQ はその古いバージョンのテーブルを削除できません。このため、マルチプレックスデータベースでテーブルの更新とクエリが同時に発生すると、SAP Sybase IQ が大量のディスク領域を消費することがあります。消費される領域の量は、データとインデックスの性質および更新の頻度によって決まります。

クエリする必要がなくなった古いバージョンを書き込みサーバが削除できるようにすれば、ディスクブロックを解放できます。古いテーブルバージョンをリカバリできるように、すべてのサーバのユーザ全員が現在のトランザクションを定期的にコミットする必要があります。これで、サーバは稼働し続けることができ、すべての機能を利用できます。`sp_iqversionuse` ストアドプロシージャを使用して、リモートサーバで使用されているバージョンを表示できます。

論理サーバのリソースの管理

論理サーバを使用することにより、マルチプレックスリソースの使用を最も効果的に管理できます。論理サーバを使用してアプリケーションごとに異なるマルチ

プレックスサーバのセットを割り当て、アプリケーションの個々のパフォーマンス要件を達成します。

マルチプレックスでは、各接続は単一の論理サーバコンテキストの下で動作します。クエリをマルチプレックスサーバに発行すると、接続の論理サーバの構成に応じて、その実行は 1 つまたは複数のマルチプレックスサーバに分散されます。論理サーバに割り当てられているリソースを動的に調整し、処理を行うアプリケーションの変化するニーズを満たすために、論理サーバに対してマルチプレックスサーバの追加または削除を行います。

クエリ負荷の分散

SAP Sybase IQ は、分散クエリ処理を通じて部分的な負荷分散を提供するか、または (サードパーティ製ソフトウェアを必要とする) ネットワーククライアントを通じて全面的な負荷分散を提供します。

分散クエリ処理
DQP (分散クエリ処理) は、マルチプレックスサーバ上で要件を満たしている適格なクエリに対して、自動的に発生します。

マルチプレックスサーバで、MIPC (マルチプレックスプロセス間通信接続) を確立しておく必要があります。現在の接続の論理サーバで、少なくとも他に 1 つのメンバノードが利用できるようになっている必要があります。共有テンポラリ DB 領域で、書き込み可能ファイルが利用できるようになっている必要があります。

ネットワーククライアントとの負荷分散
ネットワーククライアントを使用して、マルチプレックスクエリサーバ間でクエリ負荷を分散するには、プール内のマシンにクライアント接続をディスパッチできる中間システムが必要となります。

この方法を使用するには、クライアントシステムで、中間ロードバランスシステムの IP アドレスとポート番号および汎用サーバ名を指定し、**VerifyServerName** 接続パラメータを **NO** に設定した特別な ODBC DSN を作成します。クライアントがこの DSN を使って接続すると、ロードバランサは負荷が最も少ないと判断したマシンに対して接続を確立します。

注意： サードパーティ製ソフトウェアが必須です。**VerifyServerName** は、この方法が機能するようにするだけです。

その他の情報
『管理：データベース』の「付録：接続パラメータと通信パラメータリファレンス」>「ネットワーク通信のパラメータ」>「VerifyServerName 通信パラメータ [Verify]」

『管理：マルチプレックス』の「分散クエリ処理」

スキーマ設計

優れたデータベースパフォーマンスを実現するための第一歩は、優れたデータベース設計です。応答時間を短縮し、クエリ結果が早く得られるようにするには、開発時に時間をとって、設計機能をスキーマに組み込みます。

インデックス処理

SAP Sybase IQ に対する選択およびソリューションのインデックス処理。

参照：

- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

インデックスのヒント

正しいカラムインデックスタイプを選択して、クエリをより高速に実行します。

SAP Sybase IQ では、いくつかのインデックスが自動的に設定されます。射影を最適化する 1 つのインデックスがすべてのカラムに対して設定され、`UNIQUE`、`PRIMARY KEYS`、`FOREIGN KEYS` に対して `HG` インデックスが設定されます。これらのインデックスはいくつかの目的には役立ちますが、特定のクエリをできるだけ迅速に処理するには別のインデックスが必要となります。

INDEX_ADVISOR

`INDEX_ADVISOR` は、1 つまたは複数のカラム上にインデックスを追加で設定することによりクエリが高速に処理される可能性がある場合に、メッセージを生成します。

インデックスアドバイザをアクティブにするには、INDEX_ADVISOR オプションを ON に設定します。メッセージはクエリプランの一部として出力されます。クエリプランが有効になっていない場合は、メッセージログ (.iqmsg) に単独のメッセージとして出力されます。出力の形式は OWNER.TABLE.COLUMN となります。

LF インデックスまたは HG インデックス
カラムが列挙型 FP 記憶領域を使用していない場合、ジョインクエリの WHERE 句で参照されるグループ化カラムの LF インデックスまたは HG インデックスの作成を検討する必要があります。オプティマイザは、最適なクエリプランを作成するために、列挙型の FP インデックスまたは HG/LF インデックスからのメタデータを必要とする場合があります。HAVING 句で非集合カラムが参照される場合、クエリを最適化するには、LF インデックスまたは HG インデックスの使用が有効です。次に例を示します。

```
SELECT c.name, SUM(l.price * (1 - l.discount))
FROM customer c, orders o, lineitem l
WHERE c.custkey = o.custkey
    AND o.orderkey = l.orderkey
    AND o.orderdate >= "1994-01-01"
    AND o.orderdate < "1995-01-01"
GROUP by c.name
HAVING c.name NOT LIKE "I%"
    AND SUM(l.price * (1 - l.discount)) > 0.50
ORDER BY 2 desc
```

インデックスの追加は、記憶領域要件とロード時間の増大につながるため、 クエリパフォーマンスが向上する場合にのみ実行してください。

その他の情報
『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「INDEX_ADVISOR オプション」

参照：
- インデックスを使用する状況と場所 (48 ページ)
- 簡単なインデックス選択基準 (49 ページ)
- HG インデックスのロード (51 ページ)
- マルチカラムインデックス (52 ページ)

インデックスを使用する状況と場所

インデックスは、SAP Sybase IQ 内の基本的なチューニングメカニズムです。インデックスをいつどこで使用すればよいかがわかっていれば、クエリの実行速度を上げることができます。

次の状況では常にインデックスを使用します。

- ジョインカラム (カーディナリティに関係なく HG インデックス)
- 検索可能カラム (カーディナリティにもとづいて HG インデックスまたは LF インデックス)
- DATE、TIME、DATETIME/TIMESTAMP の各カラム (DATE、TIME、DTTM)
  DATE、TIME、DATETIME/TIMESTAMP の各カラムには、カーディナリティに応じて LF インデックスまたは HG インデックスも必要です。
- カラムの使用頻度が高いかどうかが不明な場合は、そのカラムに LF インデックスまたは HG インデックスを配置する。その後、負荷管理を有効にしてインデックスの使用をモニタリングすることができます。
- PRIMARY KEY、UNIQUE CONSTRAINT、UNIQUE HG のうち、適切なインデックスを使用する。そのインデックスによって、インデックス設定されたカラムの一意データに関する追加情報は SAP Sybase IQ に提供されます。
- HNG インデックスまたは CMP インデックスが配置されているカラムには、対応する LF インデックスまたは HG インデックスが必要です。
- クライアントに対してのみデータが返される (射影される) カラムにはインデックスは不要です。

参照：

- インデックスのヒント (47 ページ)
- 簡単なインデックス選択基準 (49 ページ)
- HG インデックスのロード (51 ページ)
- マルチカラムインデックス (52 ページ)

簡単なインデックス選択基準

いくつかの簡単な質問に回答することで、カラムに対して適切なインデックスを選択するために役立ちます。

クエリのことを考えずにデータモデルに最適のインデックスを判断するには、それぞれのカラムについて、以下の簡単な質問を自分にしてみてください。

- カーディナリティは 1500 ～ 2000 を超えているか。
  答えが「はい」の場合は、このカラムに HG インデックスを配置します。そうでない場合は、このカラムに LF インデックスを配置します。
- このカラムには、DATE、TIME、DATETIME、TIMESTAMP いずれかのデータが含まれているか。
  答えが「はい」の場合は、このカラムに DATE、TIME、DTTM いずれかのインデックスを配置します。また、このカラムには LF または HG も配置する必要があります。

- カラムは範囲検索または集合に使用されるか。
  答えが「はい」の場合は、このカラムに HNG インデックスを配置します。また、このカラムには LF または HG も配置する必要があります。この集合にカラム以外のものも含まれている場合は、HNG は適切といえないことがあります。ほとんどの場合、HNG インデックスは不要です。LF インデックスまたは HG インデックスには、集約を実行するために十分な機能以上のものが備わっているからです。これは、DATE、TIME、DATETIME それぞれの型には当てはまりません。
- このカラムは単語検索に使用されるか。
  答えが「はい」の場合は、このカラムに WD インデックスを配置します。LF インデックスも HG インデックスも不要です。もし配置したら、大量の領域を消費することになります。
- このカラムは全文検索に使用されるか。
  答えが「はい」の場合は、このカラムに TEXT インデックスを配置します。LF も HG も不要です。もし配置したら、大量の領域を消費することになります。
- 同じテーブル内の 2 つのカラムをお互いに比較するか (A = B, A < B, A > B, A <= B, A > = B)。
  答えが「はい」の場合は、この 2 つのカラムに CMP インデックスを配置します。
- このカラム、または一連のカラムは、GROUP BY 文または ORDER BY 文で使用するか。
  答えが「はい」の場合は、このカラムに、または GROUP BY 文または ORDER BY 文の中のカラムに HG インデックスを配置します。それぞれのカラムには、対応する HG インデックスまたは LF インデックスも必要です。
- このカラムは、マルチカラムプライマリキー、制約、またはインデックスの一部か。
  答えが「はい」の場合は、マルチカラムインデックス内の各カラムに HG インデックスまたは LF インデックスを配置します。

参照：

- インデックスのヒント (47 ページ)
- インデックスを使用する状況と場所 (48 ページ)
- HG インデックスのロード (51 ページ)
- マルチカラムインデックス (52 ページ)

HG インデックスのロード

HG インデックスは、データのロードおよび削除中の維持費が他のインデックスよりもかかります。HG インデックスのパフォーマンスを左右する主な要因は、HG インデックス構造内のデータの場所、つまり操作の実行頻度が高いか低いかです。

実行頻度の高い HG 操作は、影響を受けるローが、特定のキーの周囲に緊密にグループ化されている操作です。実行頻度の低い操作は、影響を受けることになるローの数が、キー当たりでほんの少しだけという可能性のある操作です。たとえば、データに関する日付は通常、操作が記録された時刻や変更されたデータなどの周囲にグループ化されます。つまり、新規のデータは、HG インデックス構造の一番最後に配置されます。日付 HG インデックス内のデータを削除すると、データは通常、日、週、月などのチャンクで剥がれるため、HG ツリーの先頭から削除されるか、または削除対象としていくつかのキーの周囲に緊密にグループ化されます。これらの操作は、比較的高速です。SAP Sybase IQ は少数のページで動作し、多数のローに影響を与えるからです。

価格、顧客 ID、市区町村、国など、どちらかというと更新頻度の低いデータは、だいぶ異なります。たとえば、"価格設定" データがロードされると、それぞれの値は、インデックスにすでに存在しているすべてのデータにわたって大幅に異なります。株価を追跡するカラムであれば、そのデータを格納する数値フィールドは、高い頻度で更新されます。この変更対象となっているデータは、すでにロードされている値のほぼすべての範囲にわたるからです。影響を受けるそれぞれの行に対して保持する必要のあるインデックスページの量が原因となり、これらの操作は、実行頻度の低い操作に比べて低速です。最悪のシナリオは、ロードまたは削除される EACH ROW に対して、SAP Sybase IQ が 1 ページを読み込むか、または書き込むことが強制されることです。これは、最適とは程遠いものといえますが、SAP Sybase IQ は、HG インデックスのロードおよび削除のフェーズ 2 を並列処理するように設計されており、影響は大幅に軽減されます。

これもしかたのないことですが、データモデルの設計とインデックス処理にはどのような影響が及んでいるでしょうか。SAP Sybase IQ 内の典型的なチューニングおよび最適化は通常、インデックスまたはインデックスの欠如に集約されます。データおよびロードによってインデックスにどのような影響が及ぶかを知ることは、どのインデックスを整備するか、またどのインデックスから抜けるかを判断するうえで、重要な側面です。HG インデックスは、他のインデックスと比較してロードに時間がかかるので、使用と設計の際には注目ポイントとなることがよくあります。確かに、HG インデックスは、クエリのパフォーマンスに役立つこともあります。インデックスを追加することが、クエリに対して少々プラスの影響を与えることもありますが、データのロードに影響を与えることの方が多いものです。これらの状況では、ロードまたは削除に時間がかかるのはなぜか、またそれについて何ができるかを理解することが重要です。

現在ロードされているデータに関しては、新規データの更新頻度が重大な役割を果たしています。顧客 ID の比較的ランダムなカラムは、クエリのパフォーマンスを高速にするためにインデックス処理する必要があり、そのカラムを対象としたインデックスが存在している必要があります。ただし、そのテーブルにはプライマリキーが存在していて、そのキーは、トランザクションの日時を格納している顧客 ID フィールドおよび日付フィールドであるとします。順序がそのままになっていると (customer_id、transaction_date)、ほとんどの場合、データがテーブルからロードまたは削除される頻度は低くなります。ロードされているデータは、トランザクション日付別にロードされます。ただし、顧客 ID カラムはマルチカラムインデックスの先頭にあるため、がい SAP Sybase IQ に対し、HG インデックス構造全体にわたり、すみずみまでデータを触らせることになります。

(transaction_date、customer_id に対する) 単純な変更により、この動作が変更されます。インデックスは、プライマリキーの参照整合性を制御するために、引き続き整備された状態です。カラムの順序は、プライマリキー拡張に際しては、さほど重要ではありません。したがって、カラムの順序を変更しても、下流の工程に悪影響を及ぼすことはありません。この単純な変更により、トランザクション日付別にロードされている新規データはすべて、高い頻度で、HG インデックス構造の最後に強制的に挿入されます。時間の経過に伴い、ロードは整合性を保ちながら実行されます。一般的に、データは常に、HG 構造の最後に進むからです。

マルチカラムインデックスのカラム順序を変更するだけで、パフォーマンスに劇的な影響を与える場合があります。HG インデックスのサイズは、大きく変化しないものです。データの幅は、順序に関係なく同じだからです。変化するものは、データがテーブルにロードされる速度、またはテーブルから削除される速度です。

参照：
- インデックスのヒント (47 ページ)
- インデックスを使用する状況と場所 (48 ページ)
- 簡単なインデックス選択基準 (49 ページ)
- マルチカラムインデックス (52 ページ)

マルチカラムインデックス

現時点では、インデックス作成に際して複数のカラムをサポートしているインデックスは HG、UNIQUE HG、UNIQUE CONSTRAINT、および PRIMARY KEY のみですが、マルチカラムインデックスは GROUP BY 文および ORDER BY 文にも役立ちます。

統計の視点からは、マルチカラムインデックスは、マルチカラムテーブルジョイン内で十分な情報を提供し、オプティマイザは、ジョインの正確な統計、およびそのジョインが多対多であるか 1 対多であるかを認識することができます。オプティマイザは、統計を最適化に使用できるほど十分にスマートでもありますが、実際の作業では個別の HG/LF インデックスを使用しています。オプティマイザは、

ジョインとソートすべてのシナリオのコストを見積もり、その操作にはどのインデックスが最適であるかを判定します。統計は、まさにその時点で役に立ちます。

HG インデックスについて留意すべき点をいくつか示します。

- HG の挿入は最も高価
- インデックスの最後に挿入が行われることを保証しようとする

トランザクション日付やバッチ番号 (連続データ) などの増分データは、一般的にはインデックスリストの先頭に配置します。連続キーを保証しようとするものです。

参照：
- インデックスのヒント (47 ページ)
- インデックスを使用する状況と場所 (48 ページ)
- 簡単なインデックス選択基準 (49 ページ)
- HG インデックスのロード (51 ページ)

ジョインカラム

ジョインの場合は、データ型をできるだけ幅の小さなものにして、ディスク I/O とメモリの必要容量を減らします。

整数比較は文字列比較より所要時間が短いので、ジョインには整数データ型 (できれば符号なし) を使用します。データ型をできるだけ幅の小さなものにすると、ディスク I/O とメモリの必要容量が減るため、パフォーマンスが向上します。ジョインの視点から見ると、HG インデックスは機能が少し多いので、ジョインカラムには、カーディナリティの適切なインデックス (LF または HG) でなく HG インデックスを使用します。これは、LF インデックスと比較したときの HG インデックスのロード所要時間の増加可能性と比較検討する必要があります。

参照：
- インデックス処理 (47 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)

- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

プライマリキー

マルチカラムのプライマリキーでは、プライマリキーで指定された各カラムに、追加の LF インデックスまたは HG インデックスが配置されている必要があります。これは手動で行う必要があります。SAP Sybase IQ では、HG インデックスが複合カラム上に作成されるのみだからです。

UNIQUE constraint、UNIQUE HG、およびプライマリキーは、同一の構造を共有しています。その構造は、ロー ID を格納するために、G-Array なしで HG インデックスを使用しています。可能であれば、テーブルに対してプライマリキーを使用します。これにより、インデックスが使用されていない場合でも、オプティマイザが、クエリパスに関して、より多くの情報にもとづいた決定を行いやすくなります。このインデックス構造では詳細な統計が提供され、オプティマイザは、より優れた選択ができ、データをトラバースするための構造を提供します。

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

外部キー

プライマリキーの場合と同様に、外部キーを使用して、クエリジョインパフォーマンスを向上させます。これにより、テーブルがどのようにジョインされている

かに関する情報がもう 1 つ、またそれらのジョインの背後にある統計が SAP Sybase IQ に与えられます。SAP Sybase IQ では、外部キーカラムに HG インデックスが自動作成されるため、HG インデックスも LF インデックスも追加する必要はありません。外部キーでは、参照先のテーブルにプライマリキーが存在していることが必要です。

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

データ型の適切なサイズ設定

すべてのデータ型、特に文字ベースのデータ型に対して、できるだけ正確なサイズを設定します。

カラムでどのデータ型を使用するかを決定するには、次の要素について考慮します。

- SAP Sybase IQ には、数多くのデータ型が含まれています。アプリケーションに適したデータ型を使用すると、最適なパフォーマンスが得られます。
- HOUR、MINUTE、SECOND の各情報が不要の場合は、DATETIME でなく DATE を使用します。
- データが TINYINT データ型または SMALLINT データ型の中で適合する場合は、INTEGER や BIGINT よりもそちらのデータ型を使用します。
- NUMERIC() または DECIMAL() を定義するときには、記憶領域を過度に割り付けないでください。そのレベルの精度をすべては必要としないデータの場合は、コスト高となる可能性があるからです。

- CHAR() 型と VARCHAR() 型は、デフォルトのフラット FP インデックスで幅が固定されています。唯一の違いは、使用中のバイト数を表すそれぞれの VARCHAR() 行に 1 バイトが追加されることです。

SAP Sybase IQ には、BINARY()、CHAR()、VARCHAR()、VARBINARY() の各データ型でよく見られる大規模な繰り返しパターンを圧縮する圧縮アルゴリズムが含まれています。

参照：
- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

NULL 値

カラムを NULL または NOT NULL として定義すると、オプティマイザの動作がより効率的になります。

NULL または NOT NULL と指定すると、オプティマイザは、データの特性についての情報をもう 1 つ得ることによって、ジョインおよび検索条件を学習効果で推測することができます。NULL データの場合、他のデータベースであれば領域が節約されますが、このデータベースページでは節約されません。ただし、NULL データは、SAP Sybase IQ の圧縮アルゴリズムおよび最適化されたインデックスが原因となって、ディスクに格納されるときに圧縮されます。

- NULL または NOT NULL を常に指定する
- テーブルが作成されるときの Open Client 接続と ODBC 接続のデフォルト動作は異なる

- オプティマイザに対して、ジョインと引数の検索に使用するデータの特性に関する追加情報を提供する

参照：
- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

符号なしのデータ型

場合によっては、符号なしのデータ型を使用することで符号を比較する必要がなくなり、より高速なクエリを作成できます。

すべてのデータが常に 0 以上であり、データの符号が問題にならない場合に、符号なしのデータ型を使用します。符号が格納されないため、カラムの比較で符号を比較する必要がなくなります。これにより、特にキーカラムについてパフォーマンスが向上し、データのジョインと検索の操作で手順が 1 つ少なくなります。

参照：
- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)

- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

LONG VARCHAR と LONG VARBINARY

ラージオブジェクトの保管メカニズムを使用しないでカラム単位の保管を増やすには、VARCHAR() および VARBINARY() を使用します。

通常の場合、開発者とデータベース管理者は、VARBINARY() データと VARCHAR() データは 255 バイトに制限されていると考えています。SAP Sybase IQ は、幅が 32K までの VARCHAR() および VARBINARY() (別名 LONG VARCHAR または LONG VARBINARY) をサポートしています。これによって、より大量のテキストデータまたはバイナリデータを保管することができます。このとき、データ型 BLOB/CLOB または IMAGE/TEXT の、より高度に専門化されたラージオブジェクトの保管メカニズムに移行する必要はありません。

- 適量のテキストデータまたはバイナリデータを格納するために使用できる
- 最大幅は 32K (VARBINARY() では 64K ASCII 16 進)
- WORD および TEXT インデックスは、VARCHAR() データに対して、255 バイトより幅が広いことが許容されている唯一のインデックス
- 記憶域は 256 バイトのチャンクとして割り付けられる
- 257 バイトの文字列には 512 バイトの記憶域が必要
- 511 バイトの文字列にも 512 バイトの記憶域が必要

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

ラージオブジェクトの格納

32K を超える記憶域が必要なデータには、ラージオブジェクトデータ型を使用します。

- ラージオブジェクトデータ型には、ASCII (TEXT/CLOB) データとバイナリ (IMAGE/BLOB) データが格納される。データの各 BLOB/CLOB セルは、1 つ以上のページに格納される。
  - 前提のページサイズは 128K
  - データが 129K の場合、情報の格納に 2 ページ必要
  - データが 1K の場合、データの格納に 1 ページ必要
  - どちらの場合も、ページはブロックサイズの倍数に圧縮される
- バイナリまたはテキストベースのオブジェクトを格納するために使用できる
- LONG BINARY データ型の最大サイズは 6K から無制限に拡大される
- この TEXT インデックスは、存立可能なインデックスのみ
- TEXT インデックスおよびその検索機能によって全検索が可能
- オブジェクト (byte_length64) のサイズを返す特殊関数
- 内容 (byte_substr64) 全体ではなく、オブジェクトの一部を返す特殊関数
- BFILE() 関数によって、バイナリオブジェクトセルの内容を個別のファイルに抽出することが可能

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

テンポラリテーブル

トランザクションコミット全体を通じてデータが永続するようにする場合は、グローバルテンポラリテーブルを作成するとき、またはローカルテンポラリテーブルを宣言するときに、ON COMMIT PRESERVE ROWS オプションを使用します。

テンポラリテーブルロックには以下の 3 種類があります。

- # テーブル
  ```
  CREATE TABLE #temp_table( col1 int )
  ```
- ローカルテンポラリテーブル
  ```
  DECLARE LOCAL TEMPORARY TABLE temp_table ( col1 int )
  ```
  ローカルテンポラリテーブルの動作は、# テーブルの場合とよく似ています。
- グローバルテンポラリテーブル
  ```
  CREATE GLOBAL TEMPORARY TABLE temp_table ( col1 int )
  ```
  グローバルテンポラリテーブルの構造は、接続の変更や再起動を行っても変わりません。

正常なハッシュ (#) テーブルには、ON COMMIT PRESERVE ROWS オプションは不要です。ハッシュテーブル内のデータは常に、トランザクションコミット全体を通じて永続するからです。

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)
- ハッシュの分割 (64 ページ)

パフォーマンス向上のための非正規化

データベースを非正規化することによりパフォーマンスが向上することがありますが、非正規化には、リスクと短所もあります。

非正規化を正しく行うには、アプリケーションに関する十分な知識が必要となるため、パフォーマンスに問題がある場合にのみ非正規化を実行してください。データを最新の状態に保つためにどれだけの作業が必要かを考慮する必要があります。

これは、大量のデータの要約が頻繁に必要とされる意志決定支援アプリケーションと個別にデータ変更を行うトランザクション処理要求との違いを示す良い例です。非正規化を行う場合、特定の処理の効率を向上させるために、他の処理の効率が低下することがあります。

非正規化を行うと、データの整合性に問題が発生する可能性があるため、アプリケーションの設計時に慎重に文書化し、注意する必要があります。

非正規化の決定
環境内のアプリケーションのデータアクセス要件とその実際のパフォーマンス特性を分析します。次の項目について検討します。

- 重要なクエリと予想される応答時間
- 使用するテーブルまたはカラム。1 アクセスあたりのロー数
- 通常のソート順
- 同時予測
- アクセス頻度が最も高いテーブルのサイズ
- 要約を計算するプロセスの有無

参照：
- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)

- ハッシュの分割 (64 ページ)

ロードを高速化するための UNION ALL ビュー

テーブル内のすべてのローに二次的なインデックスを維持するにはコストがかかりすぎる場合、UNION ALL ビューを使用するとロードパフォーマンスが向上することがあります。

SAP Sybase IQ では、日付などでデータを複数のベーステーブルに分けることができます。データは、これらの小さいテーブルにロードします。そして、UNION ALL ビューを使ってテーブルを 1 つの論理的な統一体に結合し、この統一体に対してクエリを実行します。

これによりロードパフォーマンスを改善することができますが、一部のクエリのパフォーマンスに悪影響を与える可能性があります。 単一のベーステーブルに対するクエリと、小さく分割された複数のベーステーブルにわたる UNION ALL ビューに対するクエリのパフォーマンスは、ビューの定義がすべての制約条件を満たしていれば、ほとんどのタイプのクエリでほぼ同じになります。 ただし、一部のクエリタイプ、特に DISTINCT を伴う、または複数のジョインカラムに関連するジョインを伴うクエリを UNION ALL ビューに対して実行した場合、その実行速度は単一の大きなベーステーブルに対して実行した場合に比べると非常に遅くなる可能性があります。 この方法を使用する場合は、アプリケーションのクエリパフォーマンスを低下させても、ロードパフォーマンスを改善する必要があるかどうかを検討するようにしてください。

UNION ALL ビューを作成するには、ベーステーブルを別々の物理テーブルに分割する論理的手段を選択します。 最も一般的な方法は、月ごとに分割することです。 たとえば、第 1 四半期のすべての月を含むビューを作成するには、次のコマンドを入力します。

```
CREATE VIEW
SELECT * JANUARY
UNION ALL
SELECT * FEBRUARY
UNION ALL
SELECT * MARCH
UNION ALL
```

月ごとに、1 つのベーステーブル (この例では JANUARY、FEBRUARY、または MARCH) にデータをロードできます。次の月は、同じカラムと同じインデックスタイプで構成された新しいテーブルにデータをロードします。

注意： UNION ALL ビューに対して INSERT...SELECT を実行することはできません。UNION ALL 演算子は、このリリースでは完全に並列であるわけではありません。 使用すると、クエリの並列処理が制限される場合があります。

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ハッシュの分割 (64 ページ)

UNION ALL ビューを参照するクエリ

UNION ALL ビューを参照するクエリのパフォーマンスを調整するには、JOIN_PREFERENCE オプションを設定します。このオプションは、UNION ALL ビュー間のジョインに影響を与えます。

最適化が効果を発揮するには、UNION ALL ビューのすべてのパーティションにすべてのインデックスが定義されている必要があります。DISTINCT を指定するクエリでは、UNION ALL ビューを使用すると、ベーステーブルを使用するよりも実行速度が遅くなる傾向があります。

SAP Sybase IQ には、UNION ALL ビューに関する次のような最適化が用意されています。

- UNION ALL ビューでの分割 GROUP BY
- UNION ALL ビューへのプッシュダウンジョイン

UNION を分割されたテーブルとして扱えるのは、以下の制約条件がすべて満たされている場合に限られます。

- 1 つまたは複数の UNION ALL のみが含まれる。

- UNION の各アームの FROM 句にテーブルが 1 つだけ含まれており、そのテーブルは物理ベーステーブルである。
- UNION のどのアームにも、DISTINCT、RANK、集合関数、または GROUP BY 句はない。
- UNION の各アームに含まれる SELECT 句の中の各項目は、カラムである。
- 最初の UNION アームの SELECT リスト内のカラムのデータ型のシーケンスが、UNION の後続の各アームにおけるシーケンスと同じである。

参照：
- UNION ALL ビューのパフォーマンス (64 ページ)

UNION ALL ビューのパフォーマンス

クエリで、ORDER BY の前に DISTINCT 演算子を評価するようにします。そうすれば、ソート順が ASC となります。

UNION より下の DISTINCT を評価する最適化は、DESC 順序に適用されません。そのため、ORDER BY が DESC の場合、UNION ALL ビュー内への DISTINCT 演算子のプッシュをはじめとした一部の最適化は適用されません。 たとえば、次のクエリはパフォーマンスに影響を与えます。

```
SELECT DISTINCT state FROM testVU ORDER BY state DESC;
```

このパフォーマンス上の問題を回避するには、クエリで ORDER BY の前に DISTINCT 演算子を評価する必要があります。こうすることにより、ソート順が ASC になり、最適化を適用できるようになります。

```
SELECT c.state FROM (SELECT DISTINCT state
  FROM testVUA) c
ORDER BY c.state DESC;
```

参照：
- UNION ALL ビューを参照するクエリ (63 ページ)

ハッシュの分割

ハッシュテーブルを分割すると、データが論理パーティションに分散されて、並列実行の対象となります。これによって、大きなテーブルおよび分散されたクエリ (PlexQ) に対するジョインパフォーマンスが拡張される場合があります。

新たなジョインアルゴリズムと集約アルゴリズムは、必要とされる中間記憶域およびネットワーク転送の量を減らすことによって、また、テーブルのローをパーティションにセマンティック分割することを活用して並列処理を増やすことによって、ハッシュの分割を利用することができます。データアフィニティによっ

て実現するキャッシュ動作の向上、およびアフィニティにもとづく作業割り付けによって、PlexQ 環境でのスケーラビリティがさらに進みます。

ハッシュ分割されたジョインアルゴリズムまたはハッシュ分割された集約アルゴリズムの使用
ハッシュ分割されたジョインアルゴリズムまたはハッシュ分割された集約アルゴリズムを使用するには、テーブル (複数可) のハッシュ分割キーのカラムがすべて、それらのテーブルの等価ジョイン条件またはグループ化式に使用されていることが不可欠です。追加のジョイン条件またはグループ化式が使用されていても、ハッシュ分割キーの構成要素であるカラムがクエリに使用されていなければ、その分割されたアルゴリズムは不適格です。そのような場合は、他の非分割アルゴリズムが引き続き適格で、非分割テーブルにはこのアルゴリズムが選択されます。一連のテーブルをハッシュ分割する際には、それらのテーブルに対するアプリケーションクエリで使用される最小の一般的カラムセットをカバーしている必要があります。分割のベースには、1 つのカラムですむことがよくあります。分割されたテーブル間でのジョインでは、カラムのデータ型が同一であることが不可欠です。

小さなテーブルがハッシュの分割から受ける利点は、大きなテーブルが受ける利点ほど大きくはありません。ハッシュの分割は、ローが少なくとも 10 億あるテーブルを対象とするように検討してください。

ラージメモリ
一部のロード操作では、デフォルトの 2 GB よりも多くのメモリが必要になることがあります。デフォルトのメモリ量よりも多くのメモリが必要な場合は、**- iqlm** 起動オプションを使用して、SAP Sybase IQ が OS に動的に要求できるメモリの量を増やします。

原則として、ラージメモリ所要量は、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 を表します。IQ のメインストアとテンポラリストアに十分なメモリを確保するために、起動パラメータ **–iqlm**、**–iqtc**、および **–iqmc** の値をそれぞれ、SAP Sybase IQ に割り付けた利用可能な総物理メモリの 3 分の 1 の量に設定してください。

ほとんどの場合、SAP Sybase IQ プロセスがスワップアウトされないようにするため、総物理メモリの 80% を SAP Sybase IQ に割り付けます。同じシステム上で稼働する他のプロセスを考慮して、実際のメモリ割り付けを調整してください。たとえば、コア数が 32、利用可能な総物理メモリが 128 GB のマシンでは、100 GB (総メモリ 128 GB のおよそ 80%) を SAP Sybase IQ プロセスに割り付けます。上記の原則に従って、**–iqlm**、**–iqtc**、および **–iqmc** パラメータの値をそれぞれ 33 GB に設定します。

その他の情報

- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「JOIN_PREFERENCE オプション」
- 『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「AGGREGATION_PREFERENCE オプション」
- 『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」>「start_iq サーバオプション」>「–iqlm iqsrv16 サーバオプション」
- 『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」>「start_iq サーバオプション」>「–iqmc iqsrv16 サーバオプション」
- 『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」>「start_iq サーバオプション」>「–iqtc iqsrv16 サーバオプション」

参照：

- インデックス処理 (47 ページ)
- ジョインカラム (53 ページ)
- プライマリキー (54 ページ)
- 外部キー (54 ページ)
- データ型の適切なサイズ設定 (55 ページ)
- NULL 値 (56 ページ)
- 符号なしのデータ型 (57 ページ)
- LONG VARCHAR と LONG VARBINARY (58 ページ)
- ラージオブジェクトの格納 (59 ページ)
- テンポラリテーブル (60 ページ)
- パフォーマンス向上のための非正規化 (61 ページ)
- ロードを高速化するための UNION ALL ビュー (62 ページ)

トラブルシューティング

パフォーマンスに関する一般的な問題を洗い出して解決します。

パフォーマンス上の問題の切り離し

問題が SAP Sybase IQ に対して、外部的なものか内部的なものかを判断します。

SAP Sybase IQ 外部的問題

- OS、ハードウェア、および記憶域をモニタリングして、ボトルネックや問題がないかを調べる
- 高い CPU 使用、高い CPU システム時間、低い CPU ユーザ時間、高い待機時間を探す
- 10 ミリ秒を超える I/O サービス時間を探す

SAP Sybase IQ 内部的問題

- `INDEX_ADVISOR` を有効にして、欠落しているインデックスを探す
- 問題が 1 つのクエリに発生している場合は、`QUERY_PLAN_AS_HTML` および `INDEX_ADVISOR` を有効にする

参照：

- 診断ツール (67 ページ)
- パフォーマンスに関する一般的な問題 (68 ページ)

診断ツール

オペレーティングシステム上にあって、システムアクティビティを監視するユーティリティです。

これらのユーティリティを使用すると、実行中のプロセス数、ページアウト回数、スワップ回数についての統計を表示できます。この統計によって得た情報を使用して、システムでページングが過度に発生していないかどうかを調査し、必要に応じて調整を行ってください。たとえば、特殊な高速ディスクにスワップファイルを配置します。

OS	ユーティリティ	説明
UNIX	**top**, **topas**	プロセッサアクティビティを継続してリアルタイムで参照する。 Solaris、Linux、HP-UX の各プラットフォームでは、**top** を使用できる。AIX では、**topas** を使用できる。
	ps	プロセスステータスをレポートする。
	vmstat	システムプロセス、メモリ、ページング、ブロック IQ、トラップ、CPU アクティビティに関する情報を表示する。
	iostat -x	ディスクのサブシステム情報を表示する。
	sar	選択された OS アクティビティの結果を、標準出力に書き込む。
Windows	[タスクマネージャ]、[リソースモニタ]	コンピュータのパフォーマンス、実行しているアプリケーション、プロセス、CPU 使用率、および他のシステムサービスに関する詳細情報を提供する。

注意： システムモニタにアクセスするには、[Logical Disk] オブジェクト、PAGEFILE.SYS ファイルが格納されているディスクのインスタンス、[Disk Transfers/Sec] カウンタを選択します。

Windows ページファイルはデータベース DB 領域デバイスとは別のディスクに格納してください。オブジェクト [Memory] と [Pages/Sec] カウンタもモニタリングできます。ただし、この値はソフトフォールトとハードフォールトの両方を含む全メモリフォールトの合計となります。

参照：
- パフォーマンス上の問題の切り離し (67 ページ)
- パフォーマンスに関する一般的な問題 (68 ページ)

パフォーマンスに関する一般的な問題

パフォーマンスに関する一般的な問題に対するソリューションです。

参照：
- パフォーマンス上の問題の切り離し (67 ページ)
- 診断ツール (67 ページ)

ページングとディスクスワッピング

適切なメモリ管理では、ページスワッピングを回避します。オペレーティングシステムファイルの使用を最小限に抑えるには、物理メモリを増やすか、または再割り当てします。

メモリが不足していると、パフォーマンスが著しく低下します。このような場合、使用可能なメモリを増やす必要があります。SAP Sybase IQ に割り付け可能なメモリが多ければ多いほど、パフォーマンスも向上します。

メモリ量には常に一定の制限があるため、データの一部のみがメモリに格納され、残りのデータはディスク上に格納されるという状況が発生します。オペレーティングシステムが、ディスク上のデータを検索して取り出し、メモリ要求に対応する必要がある場合、ページングまたはスワッピングが発生します。適切なメモリ管理では、ページングまたはスワッピングを回避するか、または最小限に抑えます。

最も頻繁に使用されるオペレーティングシステムファイルは、「スワップファイル」です。メモリが消耗している場合、オペレーティングシステムがメモリのページをディスクにスワップして、新しいデータの領域を確保します。スワップされたページを再び呼び出すと、他のページがスワップされて、要求されたメモリページが元に戻ります。ユーザのディスク使用率が高い場合、スワッピングには時間がかかります。スワッピングが起こらないようなメモリ編成にして、オペレーティングシステムファイルの使用を最小限に抑えてください。

SAP Sybase IQ では、物理メモリを最大限利用するために、データベースに対する「すべて」の読み込みと書き込みにバッファキャッシュを使用します。

注意： ディスク上のスワップ領域には、少なくとも物理メモリ全体を収容できるだけのサイズを確保します。 スワップ／ページ領域を複数の高速なディスクにストライピングすることが重要です。

参照：

- インデックスおよびローの断片化 (69 ページ)
- カタログファイル増大 (70 ページ)
- スラッシングとクエリ実行 (71 ページ)

インデックスおよびローの断片化

内部インデックスの断片化は、インデックスページが最大ボリュームまで使用されていないときに発生します。ローの断片化は、ローが削除されると発生します。ページのロー全体を削除した場合、そのページは解放されますが、ページの一部のローが未使用の場合は、未使用領域がディスクに残ります。テーブルに対する DML 操作 (`INSERT`、`UPDATE`、`DELETE`) は、インデックスの断片化を発生させることがあります。

断片化の問題についての情報を取得するには、次のストアドプロシージャを実行します。

- **sp_iqrowdensity** は FP インデックスレベルでのローの断片化を報告します。
- **sp_iqindexfragmentation** は、補助インデックス内の内部断片化を報告します。

出力を調べて、インデックスの再作成、再編成、再構築などの対応策をとるかどうかを判断する必要があります。FP インデックスを補助するために別のインデックスを作成できます。

その他の情報

- 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「sp_iqrowdensity プロシージャ」
- 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「sp_iqindexfragmentation プロシージャ」

参照：

- ページングとディスクスワッピング (69 ページ)
- カタログファイル増大 (70 ページ)
- スラッシングとクエリ実行 (71 ページ)

カタログファイル増大

カタログファイルのサイズが増加するのは正常なことで、その割合はアプリケーションとカタログの内容によって異なります。.db ファイルのサイズがパフォーマンスに影響を与えることはなく、.db ファイル内の空きページが必要に応じて再利用されます。

カタログファイルの増大を最小限に抑えるには、次の方法を使用します。

- CREATE TABLE 文に対しては、IN SYSTEM を使用しないようにする
- システムストアドプロシージャを実行した後で COMMIT 文を発行する
- 長時間実行されるトランザクションの最中に COMMIT 文を発行する

参照：

- ページングとディスクスワッピング (69 ページ)
- インデックスおよびローの断片化 (69 ページ)
- スラッシングとクエリ実行 (71 ページ)

スラッシングとクエリ実行

ハッシュアルゴリズムを伴うクエリ中にスラッシングを制限するには、HASH_THRASHING_PERCENT オプションを調整します。

HASH_THRASHING_PERCENT データベースオプションを調整し、許容するハードディスク I/O の割合を制御します。この割合を超えると、文がロールバックされてエラーが返されます。HASH_THRASHING_PERCENT のデフォルト値は 10% です。HASH_THRASHING_PERCENT の値を大きくするとロールバックするまでのディスクへのページング量が増え、HASH_THRASHING_PERCENT の値を小さくするとこれが減ります。

以前のバージョンの SAP Sybase IQ では実行されていた、ハッシュアルゴリズムを伴うクエリが、デフォルトの HASH_THRASHING_PERCENT の制限に達するとロールバックされるようになります。SAP Sybase IQ はエラー

```
Hash insert thrashing detected
```

またはエラー

```
Hash find thrashing detected
```

を報告します。実行に必要なリソースをクエリに割り当てるには、次の 1 つ以上の対応策を講じてください。

- HASH_THRASHING_PERCENT の値を増やし、ページングの制限を緩和します。
- テンポラリキャッシュのサイズを増やす。システムスラッシングが生じる可能性を回避するため、テンポラリキャッシュのサイズを増やすと、メインバッファキャッシュ割り付けで同じサイズが減ることに注意してください。
- SAP Sybase IQ がこの文の 1 つ以上のハッシュサイズの見積もりを誤っている原因を突き止めて改善します。たとえば、LF インデックスまたは HG インデックスを必要とするすべてのカラムにそれがあるかどうかを確認します。また、複数カラムのインデックスが適切かどうかも検討します。
- データベースオプション HASH_PINNABLE_CACHE_PERCENT の値を減らします。

クエリで起きている可能性のある問題を特定するには、テンポラリデータベースオプション QUERY_PLAN='ON' と QUERY_DETAIL ='ON' を指定してクエリを実行します。クエリプランの見積もりを調査します。

参照：

- ページングとディスクスワッピング (69 ページ)
- インデックスおよびローの断片化 (69 ページ)
- カタログファイル増大 (70 ページ)

クエリと削除

クエリの計画、構築、制御を行うための推奨事項です。

クエリの構築

クエリの構造を改善することにより、クエリの実行速度が向上することがあります。

- サブクエリを含むコマンド文をジョインとして構成することによって、実行速度を高めることができる場合があります。
- GROUP BY 句で複数のカラムをグループ化する場合、可能であれば、カラムに対応するユニークな値を基にして降順にカラムをリストします。 これによって最適なクエリのパフォーマンスが実現されます。
- 追加のカラムを使用して、頻繁に行う計算の結果を格納すると、パフォーマンスを向上させることができます。

ORDER BY クエリパフォーマンスの強化

マルチカラム HG インデックスを使用することにより、ORDER BY クエリのパフォーマンスを強化できます。

マルチカラム HG インデックスを使用して、単一テーブルクエリ内の複数カラムへの参照がある ORDER BY クエリのパフォーマンスを強化できます。 この変更により、ユーザが意識することなく、クエリパフォーマンスが向上します。

ORDER BY 句に複数のカラムが含まれるクエリは、マルチカラム HG インデックスを使用した方が処理速度が向上する可能性があります。たとえばテーブル T にマルチカラムインデックス HG(x,y,z) がある場合、このインデックスは射影の順序付けに使用されます。

```
SELECT abs (x) FROM T
ORDER BY x, y
```

上記の例では、HG インデックスはソート順に x と y を縦方向に射影します。

ROWID() 関数が SELECT リスト式内にある場合、マルチカラム HG インデックスも使用されます。 例を示します。

```
SELECT rowid()+x, z FROM T
ORDER BY x,y,z
```

ROWID() が ORDER BY リストの末尾にあり、ROWID() を除くそのリストのカラムがインデックス内に存在し、順序付けキーが先行する HG カラムの順序に一致す

る場合、マルチカラムインデックスがクエリに使用されます。 次に例を示します。

```
SELECT z,y FROM T
ORDER BY x,y,z,ROWID()
```

参照：

- サブクエリのパフォーマンスの改善 (74 ページ)
- キャッシュ方法の使用 (74 ページ)

サブクエリのパフォーマンスの改善

SUBQUERY_FLATTENING_PREFERENCE と SUBQUERY_FLATTENING_PERCENT を使用して、サブクエリのフラット化を制御します。

サブクエリのフラット化は、オプティマイザがサブクエリの入ったクエリを、ジョインを使用するクエリに書き換える最適化方法です。SAP Sybase IQ によって多くのサブクエリがフラット化されますが、すべてではありません。SUBQUERY_FLATTENING_PREFERENCE と SUBQUERY_FLATTENING_PERCENT を使用して、オプティマイザがこの最適化を使用する状況を制御します。

参照：

- ORDER BY クエリパフォーマンスの強化 (73 ページ)
- キャッシュ方法の使用 (74 ページ)

キャッシュ方法の使用

SUBQUERY_CACHING_PREFERENCE オプションを設定して、相関サブクエリのキャッシュ方法を選択します。

相関サブクエリにはサブクエリ外の 1 つまたは複数のテーブルへの参照が含まれており、参照されるカラムの値が変更されるたびに再実行されます。SUBQUERY_CACHING_PREFERENCE オプションを使用して、相関サブクエリを実行するためのキャッシュ方法を選択します。

参照：

- ORDER BY クエリパフォーマンスの強化 (73 ページ)
- サブクエリのパフォーマンスの改善 (74 ページ)

クエリプランの生成

クエリプランを生成することにより、オプティマイザが作成した実行プランを理解するのに役立ちます。

クエリを実行する前に、クエリオプティマイザはクエリ実行プランを作成します。クエリ実行プランは、データベースサーバがデータベース内の文に関連する情報にアクセスするために使用するステップのセットを表しています。最適化されたばかりかどうか、オプティマイザをバイパスしたかどうか、プランが以前の実行からキャッシュされたかどうかなどに関係なく、文の実行プランの保存と確認が可能です。クエリの実行プランは、元の文で使用される構文と正確に対応するとはかぎりませんが、実行プランに記述された操作は、セマンティック上は元のクエリと同等です。

クエリプランによって、処理段階を表す一連のノードで構成される実行ツリーが生成されます。ツリーの一番下のノードはリーフノードです。各リーフノードは、クエリ内のテーブルを表します。プランの最上部にあるのは、演算子ツリーのルートです。情報はテーブルから上方向に、ジョイン、ソート、フィルタ、格納、集合、サブクエリを表す演算子を通じて流れます。

ロード実行プラン
ロード実行プランは、テーブルにデータを挿入する際にデータベースエンジンが使用するステップの詳細を記述したものです。ロードプランは、クエリ実行プランと同じデータベースおよび出力を使用します。データフローオブジェクト (DFO) のツリーは、ロードの各段階で処理されるローの数を特定します。SQL 文が違えば、生成される DFO ツリーも違う場合があります。また、文が同じでも対象のテーブルの種類が違えば (未分割、範囲分割、ハッシュ分割、ハッシュ範囲分割など)、生成されるツリーが違う場合があります。

クエリプランの生成
クエリプランを生成するには、適切な評価オプションを設定してクエリを実行します。プランのテキスト版は `.iqmsg` ファイルに書き込まれます。HTML 版は、[Interactive SQL のプランビューア]、またはほとんどの Web ブラウザで表示することができます。

注意： クエリプランは、特定のクエリまたはロードの効率を評価するためにのみ使用します。QUERY_PLAN オプションを ON に設定した状態で SAP Sybase IQ を実行すると、パフォーマンスに大きな影響が及ぶことがあります。特に顕著なのは、**INSERT...VALUE** 文のボリュームが増えます。

クエリ評価オプション

適切なオプション設定は、クエリプランを評価するのに役立ちます。

- INDEX_ADVISOR – ON に設定されていると、インデックスアドバイザは、Sybase IQ クエリプランの一部として、または、クエリプランが無効の場合に Sybase IQ メッセージログファイル内の独立したメッセージとして、推奨されるインデックスを表示します。これらのメッセージは、"Index Advisor:" という文字列で始まります。この文字列を検索することで、Sybase IQ メッセージファイルからこれらのメッセージをフィルタできます。このオプションはメッセージを OWNER.TABLE.COLUMN 形式で出力します。このオプションのデフォルト設定は OFF です。

  詳細については、『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムプロシージャ」>「sp_iqindexadvice プロシージャ」を参照してください。
- INDEX_ADVISOR_MAX_ROWS ― このオプションはインデックスアドバイザによって格納されるメッセージの数を制限します。 指定した制限値に到達すると、INDEX_ADVISOR は新しいアドバイスの保存を停止します。 ただし、既存のアドバイスのカウントとタイムスタンプの更新は続行します。
- NOEXEC ― このオプションを ON に設定すると、Sybase IQ によってクエリプランが生成されますが、クエリ全体は実行されません。
  EARLY_PREDICATE_EXECUTION オプションを ON に設定すると、クエリの一部は実行されます。
  EARLY_PREDICATE_EXECUTION を OFF に設定すると、クエリを通常どおり実行する場合に比べてクエリプランが大きく異なる可能性があるため、OFF に設定しないでください。
- QUERY_DETAIL ― このオプションと、QUERY_PLAN または QUERY_PLAN_AS_HTML のいずれかを ON に設定すると、クエリプランの作成時にクエリに関する追加情報が表示されます。 QUERY_PLAN と QUERY_PLAN_AS_HTML が OFF の場合は、このオプションは無視されます。
- QUERY_PLAN ― このオプションが ON に設定されている場合 (デフォルト)、Sybase IQ はクエリについてのメッセージを生成します。
- QUERY_PLAN_TEXT_ACCESS ― このオプションが ON の場合、Interactive SQL クライアントから IQ クエリプランを表示、保存、出力できます。
  QUERY_PLAN_ACCESS_FROM_CLIENT オプションが OFF の場合、クエリプランはキャッシュされないので、クエリプランが関係する他のデータベースオプションが Interactive SQL クライアントのクエリプランの表示に影響を与えることはありません。 このオプションはデフォルトで OFF になっています。

『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「GRAPHICAL_PLAN 関数 [文字列]」と「HTML_PLAN 関数 [文字列]」を参照してください。

- QUERY_PLAN_AFTER_RUN – ON に設定されている場合、クエリの実行が完了するとクエリプランが出力されます。これにより、クエリの各ノードから渡された実際のロー数などの情報をクエリプランに追加できます。このオプションを使用するには、QUERY_PLAN を ON にします。 このオプションはデフォルトで OFF になっています。
- QUERY_PLAN_AS_HTML ― Web ブラウザで表示できるように、HTML 形式のグラフィカルなクエリプランを生成します。HTML 形式では、ノード間にハイパーリンクが設定されるため、.iqmsg ファイルのテキスト形式よりはるかに使いやすくなります。クエリプランのファイル名にクエリ名を含めるには、QUERY_NAME オプションを使用します。このオプションはデフォルトで OFF になっています。
- QUERY_PLAN_AS_HTML_DIRECTORY ― QUERY_PLAN_AS_HTML が ON で、ディレクトリが QUERY_PLAN_AS_HTML_DIRECTORY で指定されている場合、Sybase IQ は指定されたディレクトリに HTML のクエリプランを書き込みます。
- QUERY_PLAN_TEXT_CACHING ― プランをキャッシュできるようにリソースを制御できます。 このオプションが OFF の場合 (デフォルト)、クエリプランは該当するユーザ接続でキャッシュされません。
  QUERY_PLAN_TEXT_ACCESS オプションを OFF にすると、QUERY_PLAN_TEXT_CACHING の設定にかかわらず、そのユーザからの接続ではクエリプランはキャッシュされなくなります。
  『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「GRAPHICAL_PLAN 関数 [文字列]」と「HTML_PLAN 関数 [文字列]」を参照してください。
- QUERY_TIMING ― サブクエリのタイミング統計の収集などクエリエンジンの反復的な機能を制御します。 非常に短い相関サブクエリの場合、各サブクエリを実行するタイミングを合わせる処理のために、全体のパフォーマンスが大幅に低下するため、このオプションは、通常、OFF (デフォルト) にします。
- QUERY_PLAN_MIN_TIME – クエリ実行のしきい値を指定します。このクエリプランが生成されるのは、クエリ実行時間がしきい値 (マイクロ秒単位) を超過した場合のみです。このことは、実行時間がごく短いクエリに対してクエリプランの生成をオフにすることによって、システムパフォーマンスを向上させる場合があります。

注意： クエリプランを生成すると、.iqmsg ファイルに大量のテキストが追加される場合があります。QUERY_PLAN が ON の場合、QUERY_DETAIL も ON である場合は特に、メッセージログラッピングまたはメッセージログのアーカイブを有効にしてメッセージログファイルがいっぱいにならないようにしてください。

参照：
- クエリプランの使用 (78 ページ)

クエリプランの使用

QUERY_PLAN_AS_HTML オプションを設定してクエリプランの HTML 版を生成することにより、Web ブラウザで表示できます。

HTML 版は、Interactive SQL の[プランビューア]で表示することができます。HTML クエリプランでは、ツリーの各ノードが処理の詳細にハイパーリンクされています。任意のノードをクリックし、プラン内をすばやく移動できます。

SQL 関数の GRAPHICAL_PLAN と HTML_PLAN は、IQ クエリプランを文字列結果セットとしてそれぞれ XML 形式と HTML 形式で返します。データベースオプションの QUERY_PLAN_TEXT_ACCESS と QUERY_PLAN_TEXT_CACHING は、新しい関数の動作を制御します。

グラフィカルなクエリプランは、Interactive SQL の[プランビューア]で表示します。テキストプランは[プランビューア]でサポートされておらず、プランタイプはサポートされていませんというエラーメッセージが返されます。SQL 関数の GRAPHICAL_PLAN と HTML_PLAN を使用して、クエリプランを文字列結果として返させます。

その他の情報

- 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「SQL 関数」>「GRAPHICAL_PLAN 関数 [文字列]」
- 『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「SQL 関数」>「HTML_PLAN 関数 [文字列]」
- 『リファレンス：文とオプション』の「データベースオプション」>「QUERY_PLAN_TEXT_ACCESS オプション」
- 『リファレンス：文とオプション』の「データベースオプション」>「QUERY_PLAN_TEXT_CACHING オプション」

参照：
- クエリ評価オプション (76 ページ)

クエリ処理の制御

すべてのユーザが、特定のクエリの処理にかかる時間に制限を設定できます。SET ANY PUBLIC OPTION システム権限を持つユーザは、特定のユーザのクエリ

に他のクエリより高い優先度を与えることや、処理のアルゴリズムを変更し、クエリ処理の速度を操作できます。

クエリの時間制限の設定

MAX_QUERY_TIME オプションを設定することにより、クエリの実行時間を制限できます。クエリの実行時間が MAX_QUERY_TIME より長くなると、SAP Sybase IQ は適切なエラーを表示してクエリを停止します。

注意： SAP Sybase IQ では、小数の *option-value* の設定がすべて整数値にトランケートされます。たとえば、3.8 という値は 3 にトランケートされます。

参照：

- クエリの優先度の設定 (79 ページ)
- クエリ最適化オプションの設定 (80 ページ)
- ユーザ指定の条件ヒントの設定 (81 ページ)
- 負荷のモニタリング (81 ページ)

クエリの優先度の設定

クエリの優先度オプションを設定することにより、クエリ処理にユーザ別の優先度を割り当てることができます。

処理をキューで待機しているクエリは、そのクエリを送信したユーザの優先度、そしてクエリが送信された順序の順に実行されます。優先度の高いクエリがすべて実行されるまで、優先度の低いキューのクエリは実行されません。

次のオプションは、クエリにユーザ別の処理の優先度を割り当てます。

- IQGOVERN_PRIORITY – 処理キューで待機しているクエリに数字の優先度 (1、2、または 3 で、1 が最も高い) を割り当てます。
- IQGOVERN_MAX_PRIORITY – SET ANY SYSTEM OPTION システム権限が付与されたユーザは、ユーザまたはロールの IQGOVERN_PRIORITY に上限値を設定できます。
- IQ_GOVERN_PRIORITY_TIME ― 優先度の高い (優先度 1 の) クエリが、指定した時間より長く -iqgovern キューで待機している場合に、優先度の高いユーザを開始できます。

クエリの優先度を調べるには、sp_iqcontext ストアドプロシージャによって返される IQGovernPriority 属性を確認します。

参照：

- クエリの時間制限の設定 (79 ページ)
- クエリ最適化オプションの設定 (80 ページ)

- ユーザ指定の条件ヒントの設定 (81 ページ)
- 負荷のモニタリング (81 ページ)

クエリ最適化オプションの設定

最適化オプションは、クエリ処理の速度に影響します。

- AGGREGATION_PREFERENCE–集合関数(GROUP BY、DISTINCT、SET)を処理するためのアルゴリズムの選択を制御します。このオプションは、主に内部用として設計されているため、経験のあるデータベース管理者のみが使用してください。
- DEFAULT_HAVING_SELECTIVITY_PPM – クエリ内のすべての HAVING 述部の選択性を設定します。この設定が、HAVING 句によってフィルタされるロー数をオプティマイザが見積もった設定より優先して使用されます。
- DEFAULT_LIKE_MATCH_SELECTIVITY_PPM – たとえば、LIKE '*string* *%string*' (% はワイルドカード文字) のような一般的な LIKE 述部のデフォルトの選択性を設定します。他に選択性に関する情報が提供されておらず、一致文字列が、一連の定数文字と 1 つのワイルドカードで始まっていない場合、オプティマイザはこのオプションを参照します。
- DEFAULT_LIKE_RANGE_SELECTIVITY_PPM – LIKE '*string*%' という形で先行定数 LIKE 述部のデフォルトの選択性を設定します。ここで、一致文字列は一連の定数文字とその後のワイルドカード文字 (%) 1 つで構成されます。選択性に関する情報が提供されていない場合、オプティマイザはこのオプションを参照します。
- MAX_HASH_ROWS ― クエリオプティマイザがハッシュアルゴリズムを使用するときに考慮する最大ロー数の推測値を設定します。 デフォルトは、2,500,000 ローです。たとえば、2 つのテーブル間にジョインがあり、両方のテーブルからジョインに入力されるロー数がこのオプションで設定された値を超えると、オプティマイザはハッシュジョインを選択肢から外します。TEMP_CACHE_MEMORY_MB がユーザあたり 50MB を超えるシステムの場合は、このオプションにさらに大きな値を設定します。
- MAX_JOIN_ENUMERATION ― オプティマイザによる単純化が適用された後で、ジョイン順序を最適化するためのテーブルの最大数を設定します。 通常は、このオプションを設定する必要はありません。

参照：

- クエリの時間制限の設定 (79 ページ)
- クエリの優先度の設定 (79 ページ)
- ユーザ指定の条件ヒントの設定 (81 ページ)
- 負荷のモニタリング (81 ページ)

ユーザ指定の条件ヒントの設定

選択性ヒントは、オプティマイザが適切なクエリ方式を選択するのに役立ちます。

SAP Sybase IQ クエリオプティマイザは、使用可能なインデックスからの情報を使用して、クエリを実行するための適切な方式を選択します。クエリ内の各条件について、オプティマイザはインデックスを使用して条件を実行できるかどうかを決定します。条件を実行できる場合、オプティマイザはインデックスを選択し、そのテーブル上の他の条件に対する順序を決定します。これらの決定で最も重要な要因になるのは、条件の選択性、つまり条件を満たすテーブルローの端数です。

オプティマイザは通常、ユーザの介入なしに、一般的に最適な決定を行います。ただし、状況によっては、オプティマイザが条件の実行前にその選択性を正確に決定できない場合があります。これらの状況は通常、条件が適切なインデックスを使用できないカラムを対象としている場合、または算術演算または関数式が含まれるために条件が複雑すぎてオプティマイザが正確に予測できない場合に発生します。

その他の情報

- 『リファレンス: ビルディングブロック、テーブル、およびプロシージャ』の「SQL 言語の要素」>「ユーザ指定の条件ヒント」

参照：

- クエリの時間制限の設定 (79 ページ)
- クエリの優先度の設定 (79 ページ)
- クエリ最適化オプションの設定 (80 ページ)
- 負荷のモニタリング (81 ページ)

負荷のモニタリング

テーブル、カラム、インデックスの各使用状況をモニタするストアドプロシージャを使用して、クエリのパフォーマンスを向上させます。

最適化のメタデータを提供するため、またユニーク性とプライマリ／外部キーの関係を確保するために、インデックスが作成されることがよくあります。ただし、いったんインデックスが作成されると、インデックスがもたらす利点を数量化するという難題が DBA に発生します。

複数の接続によりアクセスされるか、または長期間アクセスされる必要があるデータを一時的に記憶するために、IQ メインストアにテーブルが作成されることがよくあります。このテーブルは、貴重なディスク領域を継続的に使用しているうちに忘れられてしまう可能性があります。さらに、データウェアハウス内のテーブルの数が多くなりすぎたり、負荷が複雑すぎて手作業で使用状況を分析できなくなったりします。

そのため、使用されていないインデックスとテーブルは、ディスク領域の浪費、バックアップタイムの延長、DML パフォーマンスの低下の原因となります。

SAP Sybase IQ には、指定された負荷の統計情報の収集と分析を行うための各種ツールが用意されています。DBA はクエリで参照されているので維持する必要があるデータベースオブジェクトをすぐに判断できます。使用されていないテーブル、カラム、インデックスを削除して、浪費される領域の低減、DML パフォーマンスの向上、バックアップタイムの短縮を達成できます。

負荷のモニタリングはストアドプロシージャを使用して実現されます。ストアドプロシージャは、収集処理を制御し、テーブルとカラムの使用状況とインデックス情報を詳細に報告します。 これらのプロシージャは INDEX_ADVISOR 機能を補完します。この機能はクエリのパフォーマンスを改善する可能性があるカラムインデックスの追加を推奨するメッセージを生成します。 推奨されたインデックスを追加したら、その使用状況を追跡することにより、保持するだけの価値があるかどうかを判断できます。

参照：

- クエリの時間制限の設定 (79 ページ)
- クエリの優先度の設定 (79 ページ)
- クエリ最適化オプションの設定 (80 ページ)
- ユーザ指定の条件ヒントの設定 (81 ページ)

削除オペレーションの最適化

SAP Sybase IQ では、HG インデックスと WD インデックスが設定されたカラムでの削除オペレーションを処理するために最適なアルゴリズムを選択します。

HG 削除オペレーション

SAP Sybase IQ は、HG (High_Group) インデックスが付いたカラムで削除オペレーションを処理するために、次の 3 つのアルゴリズムから 1 つを選択します。

- スモールデリートは、非常に少数のグループからローを削除するときに最適なパフォーマンスが得られます。 通常は、削除するローが 1 つだけか、HG インデックスが設定されたカラムに等号述部がある場合に選択されます。 スモールデリートアルゴリズムは、HG にランダムにアクセスできます。 最悪の場合、I/O はアクセスされるグループの数に比例します。
- ミッドデリートは、いくつかのグループからローを削除するときに最適なパフォーマンスが得られます。ただし、それらのグループが十分に分散されているか、十分に少なくて、あまり多くの HG ページがアクセスされないことが条件です。ミッドデリートアルゴリズムは、HG へのアクセスを順番に提供しま

す。最悪の場合、I/O はインデックスページ数によって制限されます。 ミッドデリートでは、削除するレコードをソートするというコストが加わります。
- ラージデリートは、多数のグループからローを削除するときに最適なパフォーマンスが得られます。 ラージデリートでは、すべてのローが削除されるまで HG が順番にスキャンされます。 最悪の場合、I/O はインデックスページ数によって制限されます。 ラージデリートは並列処理ですが、並列度はインデックスの内部構造および削除対象のグループの分散度によって制限されます。 HG カラムの範囲述部を使用して、ラージデリートのスキャン範囲を狭めることができます。

HG 削除コスト

削除コストモデルでは、I/O コスト、CPU コスト、使用可能なリソース、インデックスメタデータ、並列度、クエリから使用できる述部など、多数の要素が考慮されます。

HG、LF、enumerated FP のいずれかのインデックスを持つカラムの述部を指定すると、コストが大幅に削減されます。HG コスト計算でラージデリート以外のアルゴリズムを選択するには、削除によって影響を受ける重複しない個別の値の数を判定できる必要があります。個別カウント数は、初めはインデックスグループの数および削除されるローの数より少ないものと見なされます。述部は個別カウント数の改善された見積もりや、正確な見積もりでさえも提供できます。

現在のコスト計算では、ラージデリートにおける範囲述部の効果を考慮していません。そのため、ラージデリートの方が速いケースでミッドデリートが選択されることもあります。そうしたケースでは、必要に応じて強制的にラージデリートアルゴリズムを適用できます。これについては、次の項で説明します。

HG 削除パフォーマンスオプションの使用

HG_DELETE_METHOD オプションを使用すると、HG 削除パフォーマンスを制御できます。

HG_DELETE_METHOD オプションで指定したパラメータの値に対応する削除アルゴリズムが強制的に使用されます。

- 1 = スモールデリート
- 2 = ラージデリート
- 3 = ミッドデリート
- DML_OPTIONS5 = 4 (デリート述部のプッシュを無効にします) デフォルトは 0 — HG ラージデリートへの範囲述部のプッシュを無効にします。

HG_DELETE_METHOD データベースオプションの詳細については、『リファレンス：文とオプション』の「データベースオプション」>「HG_DELETE_METHOD オプション」を参照してください。

参照：

- WD 削除オペレーション (84 ページ)
- TEXT 削除オペレーション (85 ページ)

WD 削除オペレーション

SAP Sybase IQ は、**WD** (Word) インデックスが付いたカラムで削除オペレーションを処理するために、次の 3 つのアルゴリズムから 1 つを選択します。

- スモールデリートは、削除されるローに個別の単語が少数しか含まれておらず、多くの WD ページにアクセスする必要がない場合に、最適なパフォーマンスが得られます。 WD スモールデリートアルゴリズムは、WD へのアクセスを順番に実行します。 最悪の場合、I/O はインデックスページ数によって制限されます。 スモールデリートには、削除するレコード内の単語とレコード ID をソートするというコストが伴います。
- WD のミッドデリートは、WD スモールデリートの変形で、スモールデリートと同じ条件下では便利です。つまり、削除されるローに個別の単語が少数しかない場合です。 WD のミッドデリートでは、削除するレコード内の単語のみをソートします。 このソートは並列処理ですが、並列度は単語数と使用可能な CPU スレッド数によって制限されます。 Word インデックスの場合、通常は、ミッドデリートを使用した方がスモールデリートより高速です。
- ラージデリートは、削除されるローに個別の単語が多数含まれているために、インデックス内の多数の「グループ」にアクセスする必要がある場合に、最適なパフォーマンスが得られます。 ラージデリートでは、すべてのローが削除されるまで WD が順番にスキャンされます。 最悪の場合、I/O はインデックスページ数によって制限されます。 ラージデリートは並列処理ですが、並列度はインデックスの内部構造および削除対象のグループの分散度によって制限されます。

WD 削除コスト

WD 削除コストモデルでは、I/O コスト、CPU コスト、使用可能なリソース、インデックスメタデータ、並列度など、多数の要素が考慮されます。

WD_DELETE_METHOD データベースオプションを使用すると、WD 削除パフォーマンスを制御できます。

WD 削除パフォーマンスオプションの使用

WD_DELETE_METHOD オプションで指定したパラメータの値に対応する削除アルゴリズムが強制的に使用されます。

- 0 = コストモデルで選択されたミッドデリートまたはラージデリート
- 1 = スモールデリート

- 2 = ラージデリート
- 3 = ミッドデリート

WD_DELETE_METHOD データベースオプションの詳細については、『リファレンス：文とオプション』の「データベースオプション」>「WD_DELETE_METHOD オプション」を参照してください。

参照：
- HG 削除オペレーション (82 ページ)
- TEXT 削除オペレーション (85 ページ)

TEXT 削除オペレーション

SAP Sybase IQ は、TEXT インデックスが付いたカラムで削除オペレーションを処理するために、次の 2 つのアルゴリズムから 1 つを選択します。

- スモールデリートは、削除されるローに個別の単語が少数しか含まれておらず、多くの TEXT ページにアクセスする必要がない場合に、最適なパフォーマンスが得られます。 TEXT スモールデリートアルゴリズムは、TEXT へのアクセスを順番に実行します。 最悪の場合、I/O はインデックスページ数によって制限されます。 スモールデリートには、削除するレコード内の単語とレコード ID をソートするというコストが伴います。
- ラージデリートは、削除されるローに個別の単語が多数含まれているために、インデックス内の多数の「グループ」にアクセスする必要がある場合に、最適なパフォーマンスが得られます。 ラージデリートでは、すべてのローが削除されるまで TEXT が順番にスキャンされます。 最悪の場合、I/O はインデックスページ数によって制限されます。 ラージデリートは並列処理ですが、並列度はインデックスの内部構造および削除対象のグループの分散度によって制限されます。

TEXT 削除コスト
TEXT 削除コストモデルでは、I/O コスト、CPU コスト、使用可能なリソース、インデックスメタデータ、並列度など、多数の要素が考慮されます。

TEXT_DELETE_METHOD データベースオプションを使用すると、TEXT 削除パフォーマンスを制御できます。

TEXT 削除パフォーマンスオプションの使用
TEXT_DELETE_METHOD オプションで指定したパラメータの値に対応する削除アルゴリズムが強制的に使用されます。

- 0 = コストモデルで選択されたミッドデリートまたはラージデリート

- 1 = スモールデリート
- 2 = ラージデリート

TEXT_DELETE_METHOD データベースオプションの詳細については、『Sybase IQ の非構造化データ分析の概要』の「TEXT インデックスとテキスト設定オブジェクト」>「TEXT_DELETE_METHOD オプション」を参照してください。

参照：

- HG 削除オペレーション (82 ページ)
- WD 削除オペレーション (84 ページ)

索引

記号

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

S

T

U

W

い

お

か

き

く

さ

し

す

せ

ち

て

と

な

に

ね

は

ひ

ふ

へ

ま

め

ら

り

れ

ろ