
# システム・リソースの計画と管理

Version 2024.1
2024-06-03

InterSystems Corporation　1 Memorial Drive　Cambridge MA 02142　www.intersystems.com

システム・リソースの計画と管理
InterSystems IRIS Data Platform　Version 2024.1　2024-06-03


インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください：

InterSystems Worldwide Response Center (WRC)
Tel:　+1-617-621-0700
Tel:　+44 (0) 844 854 2917
Email:　support@InterSystems.com

# 目次



図一覧

# システム・リソースの計画と管理

InterSystems SQL でアプリケーションを開発および実行する場合、自由に使用できるシステム・リソースについて考慮し、これらを管理して、システムのパフォーマンスを最適化する必要があります。このページは、特定のワークロードに必要なリソースを見積もったり、ワークロードで利用可能なリソースを最適化したりするのに役立ちます。

インスタンスを導入する前に、次のセクションを参照して、最適なパフォーマンスの実現に必要なリソースを見積もることができます。

- メモリの計画
- ストレージの計画
- InterSystems IRIS の CPU サイジングおよび拡張

次のセクションでは、導入時または導入後の共有メモリの構成プロセスについて説明します。

- InterSystems IRIS 共有メモリの構成

次のセクションで説明するとおり、リソースの使用量を定期的に監視することで、パフォーマンスを最適化できます。

- メモリ使用量の確認と監視
- データの圧縮とストレージ・コストの削減

# 1 メモリの計画

このセクションでは、システム・メモリ・リソースの見積もりおよび計画方法の決定についてのガイドラインを示します。メモリ・リソースについて特に注意が必要な特定のワークロード・シナリオについても説明します。これらのガイドラインを使用して他の基準（企業の標準やクラウドの予算制限など）に基づいて選択したシステムが作業負荷要件に対処するのにほぼ十分であるかどうかを評価したり、それらの要件に基づいて必要なシステムを計画したりすることができます。

メモリ計画の目的は、各ホストで実行されるすべてのエンティティに、すべての通常の動作環境下で十分なメモリをホスト上で提供することです。これは、パフォーマンスと可用性の両方における重要な要素です。メモリ要件を見積もったら、導入中または導入後に適宜共有メモリを構成する必要があります。運用中に定期的に実際のメモリ使用量を確認および監視し、必要に応じて調整する必要があります。

重要　Linux システムで十分な物理メモリを構成しておらず、たびたび限界付近に達している場合、ライト・デーモンや CSP サーバ・プロセスなど、長時間実行される InterSystems IRIS プロセスのうち、通常動作で多くのメモリにアクセスするものを、OOM killer が誤って問題の原因と判断し、それらを終了させるリスクがあります。これにより InterSystems IRIS インスタンスの障害が生じ、その後の開始時にクラッシュ回復が必要となります。ただし、OOM killer を無効にすることはお勧めできません。この安全メカニズムにより、メモリの不足時にオペレーティング・システムのクラッシュを回避し、ユーザが介入して InterSystems IRIS を通常動作に戻す機会が与えられるためです。この問題を回避するには、OOM killer が機能することがないよう、十分な物理メモリを構成することをお勧めします（詳細は、"インターシステムズ製品のプロセス・メモリ" を参照してください）。

## 1.1 メモリ要件の見積もり

以下にメモリの見積もりの概要を示します。

通常、InterSystems IRIS インスタンスをホストするサーバには、以下に示すようにメモリのコンシューマが主に 4 つあります。

- ファイル・システム・キャッシュ、およびページ・ファイルまたはスワップ領域を含む、オペレーティング・システム。
- InterSystems IRIS とそれをベースとするアプリケーション以外の、実行中のアプリケーション、サービス、およびプロセス
- InterSystems IRIS およびアプリケーションのプロセス

  InterSystems IRIS はプロセスベースです。アプリケーションの実行中にオペレーティング・システムの統計を調べると、InterSystems IRIS の一部として多数のプロセスが実行されていることがわかります。
- 以下のものを含む、InterSystems IRIS 共有メモリ
  - ディスク読み取りを最小限に抑えるためにデータがキャッシュされる、データベース・キャッシュ (グローバル・バッファ・プールとも呼ばれます)。そのサイズはパフォーマンスの主要な要素となります。
  - ディスク読み取りを最小限に抑えるためにコードがキャッシュされる、ルーチン・キャッシュ。
  - 共有メモリ・ヒープ。ここから共有メモリが自動的に割り当てられ、さまざまなインスタンスの目的には手動で割り当てられます。
  - その他の共有メモリ構造

  可能な限りのパフォーマンスを実現するには、すべて通常の動作環境の下で、InterSystems IRIS 共有メモリのこれらすべてのコンシューマを物理 (システム) メモリ内に保持する必要があります。

  重要　仮想メモリとそれを使用するためのメカニズム (スワップ領域やページングなど) は、メモリ容量の一時的な問題が発生した場合にシステムが動作を継続できるようにするため重要ですが、最優先すべきは (リソースが許すならば)、通常の動作条件下で仮想メモリをまったく使用しなくても済むように十分な物理メモリを含めることです。

ごく一般的なガイドラインとして、InterSystems IRIS ベースのアプリケーションをホストするシステムに対して、CPU コアごとに 4 ～ 8 GB のシステム・メモリを割り当てることをお勧めします。例えば、16 コアのシステムには、最低でも 64 GB、できれば 128 GB の RAM を用意する必要があります。

注釈　このコア数には、Intel のハイパースレッディング (HT) や IBM の同時マルチスレッディング (SMT) などのスレッドは含めないでください ("InterSystems IRIS プラットフォームでの一般的なパフォーマンス強化" を参照)。つまり、例えば、IBM AIX 論理パーティション (LPAR) に 8 つのコアが割り当てられている場合、SMT-4 が有効になっていて、論理プロセッサが 32 個のように見えても、その LPAR に割り当てられる RAM の合計の計算は 4 ～ 8 GB * 8 = 32 ～ 64 GB となります。

以下のコンポーネントのメモリ要件の概算を加算することで、より具体的な見積もりを得ることができます。

- オペレーティング・システム (ファイル・システム・キャッシュ、およびページ・ファイルまたはスワップ領域を含む)、および InterSystems IRIS 以外でインストールされているすべてのプログラムで必要な量。

  システムで実行される他のエンティティとプロセスのメモリ要件は大きく異なります。可能であれば、InterSystems IRIS とコホストするソフトウェアが使用するメモリを現実的に見積もってください。

  スワップ領域やページ・ファイルの場合、構成に必要なメモリの量はOSによって異なります。Windows の場合、Microsoft では RAM の総量の 1.5 倍を構成することを推奨しています。Linux の場合は、2 ～ 4 GB を構成することを計画します。

  前述したように、スワップやページングはパフォーマンスの低下を招くため、メモリ容量の一時的な問題 (メモリ・カードの故障など) により必要となった場合にのみ実行する必要があります。さらに、システムにより仮想メモリが使用された場合、さらに深刻な状況を回避するため迅速なアクションを取ることができるようオペレータに通知するアラートを構成する必要があります。

注釈　強くお勧めしているラージ・ページおよびヒュージ・ページが構成されると、InterSystems の共有メモリ・セグメントは物理メモリに固定され、スワップされることはなくなります (″ラージ・ページおよびヒュージ・ページの構成″ を参照)。

- InterSystems IRIS プロセスで必要な量。

  一般に、プロダクションで実行される InterSystems IRIS プロセスの数は 1000 以下で、通常、それぞれが 12 ～ 16 MB を消費します。したがって、ほとんどの場合、12 ～ 16 GB あれば十分です。ただし、実行される InterSystems IRIS プロセスの数やメモリのニーズは大きく異なり、アプリケーションでの全体的なプロセス数やプロセス当たりのメモリ・パーティション・サイズの要件について評価すると、このコンポーネントに対してより大きな見積もりが必要であると示されることがあります。

- InterSystems IRIS インスタンスの**共有メモリ**に必要な量。

  ほとんどのインスタンスの共有メモリの割り当ては、それぞれのサイジングの一般的なガイドラインを提供する次の表に示す構成パラメータで決まります。

| パラメータ | サイズの特定対象 | システム・メモリ ≤ 64 GB の場合の割り当て | システム・メモリ > 64 GB の場合の割り当て |
|---|---|---|---|
| globals | データベース・キャッシュ (グローバル・バッファ・プール) | 合計システム・メモリの 50% | 合計システム・メモリの 70% |
| routines | ルーチン・キャッシュ | 256 MB 以上 | 512 MB 以上 |
| gmheap | 共有メモリ・ヒープ * | 256 MB 以上 | 384 MB 以上 |
| jrnbufs | ジャーナル・バッファ | 64 MB (既定) | 64 MB (既定) |

  * gmheap 設定は KB 単位で指定されますが、比較しやすいように、ここでは MB 単位で示されています。

  これらの各パラメータの初期設定を決定したら、次の式を使用して、インスタンスの合計共有メモリ要件を見積もることができます。(MaxServers + MaxServerConn によって、インスタンスから、およびインスタンスへ、それぞれ許容される ECP 接続の最大数が指定されます。既定値は、前者が 2、後者が 1 です。)

  globals*1.08 + routines*1.02 + gmheap (MB 単位) + jrnbufs + (MaxServers + MaxServerConn)*2 + 300

  例えば、ECP 接続パラメータは既定の設定のままの (分散キャッシュ・クラスタに含まれていないため) 128 GB の RAM を有するシステムでは、提示されているサイジング・ガイドラインにより必要な合計共有メモリは次のようになります。

  (128 GB*0.7*1.08) = 97 GB + (522 + 384 + 64 + 6 + 300 = 1276 MB) = 合計 98 GB

128 GB のシステムでは、InterSystems IRIS の見積もりは、プロセッサに 16 GB、共有メモリに 97 GB で合計 113 GB、残りの 15 GB がその他のシステム用となります。

注釈　アプリケーションで多数の SQL クエリを使用する場合、または並列デジャーナリングを有効にすることを計画している場合は、追加メモリを gmheap に割り当てる必要があります。詳細は、″並列クエリ処理の構成″ の ″共有メモリの考慮事項″、および ″並列デジャーナリングのシステム要件″ を参照してください。

実際のプロセス・メモリおよび共有メモリの使用量は、上記の見積もりとは異なる可能性があります。特に次のことに留意してください。

- データベース・キャッシュ (globals) の場合、既定のデータベース・ブロック・サイズの 8 KB に乗数 1.08 が適用されます。この係数は、ブロック・サイズが大きい場合は小さくなります。データベース・キャッシュを許可し、これを複数のブロック・サイズに割り当てる場合、効果的な乗数はより小さくなります。
- 共有メモリ・ヒープのサイズは、ホストの CPU の数に応じて自動的に増加される可能性があります。

"メモリ使用量の確認と監視" では、インスタンスの実際のプロセスおよび共有メモリの使用量の確認方法について説明しています。

## 1.2 ラージ・ページおよびヒュージ・ページの構成

サポートされていれば、ラージ・メモリ・ページおよびヒュージ・メモリ・ページの使用は、次のとおり、パフォーマンスにおいて非常に有益であり、強くお勧めします。

- IBM AIX® ― ラージ・ページの使用は、特に 16GB を超える共有メモリ (データベース・キャッシュ、ルーチン・キャッシュ、および共有メモリ・ヒープの合計) を構成する場合、強くお勧めします ("メモリ要件の見積もり" を参照)。

  既定では、ラージ・ページが構成されていると、システムはメモリ割り当てで自動的にラージ・ページを使用します。ラージ・ページに共有メモリを割り当てることができない場合、共有メモリは標準 (スモール) ページに割り当てられます。ただし、memlock パラメータを使用すれば、ラージ・ページをさらにきめ細かく制御できます。

- Linux (すべてのディストリビューション) ― 物理 (ベア・メタル) サーバまたは仮想サーバには、可能であれば静的なヒュージ・ページ (2MB) を強くお勧めします。InterSystems IRIS 共有メモリ・セグメントに対して静的ヒュージ・ページを使用すると、CPU の平均使用率がアプリケーションによっておよそ 10 ～ 15% 削減されます。

  既定では、ヒュージ・ページが構成されると、InterSystems IRIS は開始時にヒュージ・ページで共有メモリのプロビジョニングを試みます。十分な領域がない場合、InterSystems IRIS は標準のページに戻り、割り当てられていたヒュージ・ページ領域が孤立して、システム・ページングが生じる可能性があります。ただし、memlock パラメータを使用すれば、この動作を制御することができ、ヒュージ・ページの割り当てに失敗した場合は開始時に失敗します。

- Windows ― ページ・テーブル・エントリ (PTE) のオーバーヘッドを削減するには、ラージ・ページを使用することをお勧めします。

  既定では、ラージ・ページが構成されると、InterSystems IRIS は開始時にラージ・ページで共有メモリのプロビジョニングを試みます。十分な領域がない場合、InterSystems IRIS は標準のページに戻ります。ただし、memlock パラメータを使用すれば、この動作を制御することができ、ラージ・ページの割り当てに失敗した場合は開始時に失敗します。

### 1.2.1 IBM AIX® でのラージ・ページの構成

AIX は複数のページ・サイズ (4 KB、64 KB、16 MB、および 16 GB) をサポートしています。4 KB ページと 64 KB ページの使用は、InterSystems IRIS にとって透過的です。InterSystems IRIS が 16 MB のラージ・ページを使用するようにするには、そのラージ・ページを AIX で構成する必要があります。AIX は、構成済みのラージ・ページまたはヒュージ・ページの数を需要に基づいて自動的に変更することはありません。現時点では、InterSystems IRIS は 16 GB のヒュージ・ページを使用しません。

ラージ・ページは高性能な環境でのみ構成します。ラージ・ページに割り当てられたメモリは、ラージ・ページのみが使用できるようになるためです。

既定では、ラージ・ページが構成されていると、システムはメモリ割り当てで自動的にラージ・ページを使用します。ラージ・ページに共有メモリを割り当てることができない場合、共有メモリは標準 (スモール) ページに割り当てられます。ラージ・ページの詳細な制御については、"memlock" を参照してください。

AIX では、ラージ・ページは以下のように vmo コマンドを使用して構成します。

```
vmo -r -o lgpg_regions=<LargePages> -o lgpg_size=<LargePageSize>
```

<LargePages> は予約するラージ・ページの数を指定し、<LargePageSize> は、ハードウェアでサポートされるラージ・ページのサイズをバイト単位で指定します。

注釈　動的 LPAR (Logical PARtitioning : 論理区画) をサポートするシステムでは、-r のオプションを省略して、システムを再起動することなく動的にラージ・ページを構成できます。

例えば、以下のコマンドは、1 GB のラージ・ページを構成します。

```
# vmo -r -o lgpg_regions=64 -o lgpg_size=16777216
```

ラージ・ページを構成したら、bosboot コマンドを実行してこの構成をブート・イメージに保存します。システムの起動後、以下の vmo コマンドを使用して固定されたメモリ用にこのラージ・ページを有効にします。

```
vmo -o v_pinshm=1
```

ただし、memlock=64 の場合、`vmo -o v_pinshm=1` は必要ありません。

### 1.2.2 Linux でのヒュージ・ページの構成

十分なヒュージ・ページが使用できる Linux プラットフォームでは、InterSystems IRIS の共有メモリ・セグメントがヒュージ・ページ・プールから割り当てられます。ヒュージ・ページを使用すると、ページ・テーブルの領域を節約することでメモリを節約できます。また、InterSystems IRIS を共有メモリ・セグメント内にロックして、そのページがページ・アウトされないようにすることができます。InterSystems IRIS をホストするシステムでは、ほとんどの状況でヒュージ・ページを使用することをお勧めします。

Linux システムの既定のメモリ・ページ・サイズは 4 KB です。最新の Linux ディストリビューションには、ヒュージ・ページ (システム構成に応じて、2 MB または 1 GB のメモリ・ページ・サイズ) のオプションが含まれています。ヒュージ・ページが構成されていると、InterSystems IRIS はメモリの割り当て時に自動的にヒュージ・ページを使用します。

重要　2.6.38 カーネルを使用する一部の Linux ディストリビューションには、透過的ヒュージ・ページ (THP) が導入されていて、ヒュージ・ページの作成、管理、および使用が自動化されます。ただし、THP は InterSystems IRIS に割り当てられたメモリの大半を構成する共有メモリ・セグメントを処理しません。また、実行時にメモリ割り当ての遅延を引き起こすことがあり、パフォーマンスに影響する可能性があります。この影響は、ジョブまたはプロセスの作成頻度が高いアプリケーションでは顕著に表れます。このような理由から、InterSystems IRIS をホストするすべてのシステムで THP を無効化するようにお勧めします。このトピックの詳細は、InterSystems Developer Community の ″Linux Transparent huge pages and the impact to InterSystems IRIS″ を参照してください。

Linux でヒュージ・ページを構成するには、以下の手順に従います。

1. ステータスを確認します。

   /proc/meminfo にはヒュージ・ページに関する情報が記録されています。既定では、ヒュージ・ページは割り当てられていません。既定のヒュージ・ページ・サイズは 2 MB です。以下はその例です。

   ```
   HugePages_Total:     0
   HugePages_Free:      0
   HugePages_Rsvd:      0
   Hugepagesize:     2048 KB
   ```

2. ヒュージ・ページ数を変更します。

   ページ数のシステム・パラメータを直接変更できます。例えば、2,056 ページのヒュージ・ページを割り当てるには以下のように指定します。

   ```
   # echo 2056 > /proc/sys/vm/nr_hugepages
   ```

   注釈　また、以下のように sysctl(8) を使用することもできます。

   ```
   # sysctl -w vm.nr_hugepages=2056
   ```

   ヒュージ・ページは連続したページとして割り当てる必要があります。また、割り当て後には再起動が必要になることがあります。したがって、割り当てを確実なものとして恒久的な変更とするには、以下の手順に従います。

a. /etc/sysctl.conf ファイルに以下の行を入力します。

```
echo "vm.nr_hugepages=2056" >> /etc/sysctl.conf
```

b. システムを再起動します。

c. 再起動後に、meminfo を検証します。例えば、以下のようになります。

```
[root woodcrest grub]# tail -4 /proc/meminfo
HugePages_Total:  2056
HugePages_Free:   2056
HugePages_Rsvd:      0
Hugepagesize:     2048 KB
```

3. InterSystems IRIS でのヒュージ・ページの使用を検証します。

InterSystems IRIS を起動すると、割り当てられた共有メモリの量が、例えば以下のようなメッセージで表示されます (これは messages.log ファイルに記録されています)。

```
Allocated 3580MB shared memory: 3000MB global buffers, 226MB routine buffers
```

ヒュージ・ページで使用可能なメモリ量は、割り当てられる共有メモリの合計量より多いことが必要です。このメモリ量が不足していると、ヒュージ・ページは使用できません。例えば、共有メモリが 3580 MB で、ヒュージ・ページ・サイズが 2 MB の場合、少なくとも 1791 のヒュージ・ページを構成する必要があります。

注釈　未使用のヒュージ・ページを他のコンポーネントで使用することはできないため、HugePages_Total は、できるだけ共有メモリ使用量に近い値に設定することをお勧めします。

### 1.2.3 Windows でのラージ・ページの構成

InterSystems IRIS がそのメモリを Windows システムで大きなページとして割り当てることができるようにすることをお勧めします。これにより、メモリおよびページング容量をより効率的に使用できます。そのためには、InterSystems IRIS の実行に使用する Windows ユーザ・アカウントに Windows の“メモリ内のページのロック”(SELockMemory) 特権を付与します。この特権により、InterSystems IRIS はメモリをラージ・ページとして要求できるようになります。起動時のメモリ割り当ての詳細な制御については、″memlock″ を参照してください。

注釈　InterSystems IRIS が既定の SYSTEM アカウントで動作している場合は、既定で“メモリ内のページのロック”特権が付与されています。

ラージ・ページを使用しているときに InterSystems IRIS を再起動する場合は、構成されているすべてのメモリが割り当てられるようにするために、通常は Windows の再起動も必要になります。起動時に、構成されているすべてのメモリ容量を割り当てることができない場合は、少量のメモリで起動が試行されます。このとき、大きなページを使用できる場合とできない場合があります。メッセージ・ログで InterSystems IRIS の最新の起動状況を確認することで、実際に割り当てられているメモリをチェックできます。

## 1.3 並列デジャーナリングのシステム要件

並列デジャーナリングを構成して、複数のアップデータ・ジョブが並列でレコードを適用できるようにすることができます。並列デジャーナリングを有効にすると、アップデータ、リーダ、プリフェッチャのジョブを含む、多くのプロセスを同時に実行することができます。アップデータ・ジョブの既定の数は、パフォーマンスを最適化するように、システム・リソースに基づいて決定されます。この機能はスループットを向上させ、ミラーリング (″並列デジャーナリングの構成″を参照) やジャーナルのリストア (″^JRNRESTO を使用したジャーナル・ファイルからのグローバルのリストア″ を参照) で使用されます。

注釈　並列デジャーナリングにより複数のアップデータが同じデータベース内で別のグローバルを記録できるため、少数のデータベースで作業する場合でもスループットを向上させることができます。

次の表を使用して、最大 16 までの任意の数のアップデータに必要なリソースを見積もることができます。

| CPU の最小数 | 最小データベース・キャッシュ・サイズ | 最小空き gmheap | アップデータ数 |
|---|---|---|---|
| 8 | 0 GB | 0.4 GB | 2 |
| 9 | 0 GB | 0.6 GB | 3 |
| 12 | 0 GB | 0.8 GB | 4 |
| 15 | 160 GB | 1.0 GB | 5 |
| 18 | 192 GB | 1.2 GB | 6 |
| 21 | 224 GB | 1.4 GB | 7 |
| 24 | 256 GB | 1.6 GB | 8 |
| 27 | 288 GB | 1.8 GB | 9 |
| 30 | 320 GB | 2.0 GB | 10 |
| 33 | 352 GB | 2.2 GB | 11 |
| 36 | 384 GB | 2.4 GB | 12 |
| 39 | 416 GB | 2.6 GB | 13 |
| 42 | 448 GB | 2.8 GB | 14 |
| 45 | 480 GB | 3.0 GB | 15 |
| 48 | 512 GB | 3.2 GB | 16 |

注釈　　システムが搭載する CPU の数の決定については、"CPU (プロセッサ)" を参照してください。

アップデータを追加で有効にする場合、より多くのメモリを gmheap に割り当てる必要が生じる可能性が非常に高くなります。既定では、gmheap は、グローバル・バッファに割り当てられるメモリの 3% に構成されます。アップデータの追加ごとに、追加で 200 MB を割り当てる必要があります。

# 2 ストレージの計画

現在、従来の磁気回転方式の HDD デバイスから SSD および PCIe フラッシュ・ベース・デバイスに至るまで多くのストレージ技術が利用されています。さらに、複数のストレージ・アクセス技術には、NAS、SAN、FCoE、直接接続、PCIe、ハイパー・コンバージド・インフラストラクチャを備えた仮想ストレージなどがあります。

ご使用のアプリケーションに最適なストレージ技術は、アプリケーション・アクセス・パターンで決まります。例えば、ランダム読み取りが大部分であるアプリケーションの場合、SSD またはフラッシュ・ベース・ストレージが理想的なソリューションです。また、書き込みが非常に多いアプリケーションの場合、従来の HDD デバイスが最適です。

このセクションでは、以下を含む、アプリケーションに最適なストレージ・ソリューションのガイドラインとベスト・プラクティスを示します。

- ストレージの構成
- ストレージの接続
- ファイル・システムの分離
- バッファ型入出力と直接入出力

- noatime マウント・オプション

## 2.1 ストレージの構成

ストレージ・アレイを取り巻く状況は、技術的特徴、機能、およびパフォーマンスのオプションにおいて絶えず変化していて、複数のオプションが最適なパフォーマンスおよび復元可能性を InterSystems IRIS にもたらします。このセクションでは、InterSystems IRIS の最適なパフォーマンスおよびデータの復元可能性を実現するための一般的なベスト・プラクティスのガイドラインを示します。アプリケーションの入出力パターンによって、最高のソリューションをもたらすストレージ RAID レベルおよび構成をストレージ・ベンダと共に決定できます。さまざまなファイル・システムがサポートされ、構成に応じて最適化されます。

可能な場合には、ファイル・タイプのブロック・サイズと同等のブロック・サイズを使用することが最良です。ほとんどのストレージ・アレイは指定されたボリューム用に使用できるブロック・サイズが低く制限されていますが、ファイル・タイプのブロック・サイズをできる限り近くできます。例えば、ストレージ・アレイ上の 32KB または 64KB のブロック・サイズは、通常、8KB ブロック形式の IRIS.DAT ファイルを効果的にサポートする実用的なオプションです。ここでの目標は、アプリケーションのニーズに基づいてストレージ・アレイの入出力が過剰または無駄にならないようにすることです。

重要　パフォーマンスと復元可能性を最適化するために、ジャーナル・ファイルは別個のに配置する必要があります。

以下の表は、InterSystems IRIS のインストール環境内でのストレージの入出力の一般的な概要を示しています。

| 入出力の種類 | タイミング | 方法 | 留意事項 |
|---|---|---|---|
| データベースの読み取り (ほぼランダム) | インタフェースおよびユーザの処理ごとに継続的 | インタフェース・クエリまたはユーザの処理によってディスク入出力が開始され、データを読み取り | データベースの読み取りは、デーモンが処理する Web ページ、SQL クエリ、または直接ユーザ処理によって実行されます。 |
| データベースの書き込み (順序付けされるが非連続で実行) | 約 80 秒ごとまたは保留中の更新がデータベース・キャッシュのしきい値のパーセントに達したとき (先に基準を満たした方) | データベース・ライト・デーモン (8 個のプロセス) | データベースの書き込みは、ライト・デーモンと呼ばれる一連のデータベース・システム・プロセスによって実行されます。ユーザ処理によってデータベース・キャッシュが更新され、トリガ (時間またはデータベース・キャッシュのパーセントが一杯になったとき) によってライト・デーモンを使用してディスクの更新が実行されます。通常、更新レートに応じて、書き込みサイクル中に数 MB から数 GB 書き込む必要があることを想定しています。 |
| WIJ 書き込み (連続) | 約 80 秒ごとまたは保留中の更新がデータベース・キャッシュのしきい値のパーセントに達したとき (先に基準を満たした方) | データベース・マスタ・ライト・デーモン (1 個のプロセス) | ライト・イメージ・ジャーナル (WIJ) は、データベースの書き込みサイクル中のシステム障害から物理データベース・ファイルの整合性を保護するために使用されます。書き込みは、約 256KB のサイズごとに行われます。 |
| ジャーナル書き込み (連続) | ジャーナル・データの 64KB ごとまたは 2 秒ごと、または ECP またはアプリケーションの同期要求 | データベース・ジャーナル・デーモン (1 個のプロセス) | ジャーナル書き込みは連続で、4KB から 4MB までサイズが変化します。1 秒あたり数十回の書き込みから、ECP および個別のアプリケーション・サーバを使用した非常に大きい配置の場合には 1 秒あたり数千回の書き込みまでになる場合があります。 |

ストレージのボトルネックは、データベースのシステム・パフォーマンスに影響する最も一般的な問題の 1 つです。一般的な誤りは、十分な数の個別ディスクを確保して 1 秒あたりに予期される入出力操作 (IOPS) をサポートするのではなく、データ容量のみに対してストレージをサイズ調整することです。

| 入出力の種類 | 平均応答時間 | 最大応答時間 | 留意事項 |
|---|---|---|---|
| データベースのランダム読み取り (キャッシュ処理なし) | <=1 ms | <=2 ms | データベース・ブロックは固定 8KB、16KB、32KB、または 64KB (ホストでのデータベース・キャッシュは大きいため、ディスクへのほとんどの読み取りはキャッシュされません)。 |
| データベースのランダム書き込み (キャッシュ処理あり) | <=1 ms | <=2 ms | すべてのデータベース・ファイルの書き込みは、ストレージ・コントローラのキャッシュ・メモリによってキャッシュされることが想定されます。 |
| ジャーナル書き込み | <=1 ms | <=2 ms | ジャーナル書き込みは連続で、4KB から 4MB までサイズが変化します。 |

これらの数値はガイドラインとして提供しており、任意のアプリケーションで理想的なパフォーマンスを実現するためには、許容範囲およびしきい値がこれらの数値よりも高いまたは低い場合があります。これらの数値および入出力の水準は、ご使用のストレージ・ベンダとの議論の開始点として使用してください。

## 2.2 ストレージの接続

このセクションでは、ストレージ・エリア・ネットワーク (SAN) およびネットワーク接続ストレージ (NAS) に関する考慮事項について説明します。

### 2.2.1 SAN ファイバ・チャンネル

各ホストから SAN スイッチまたはストレージ・コントローラに複数のパスを使用します。1 つのカードの障害から保護するために複数の HBA を使用すると保護のレベルが上がりますが、最小の推奨事項は、少なくとも 1 つのデュアルポート HBA を使用することです。

ストレージ・アレイ・レイヤに復元可能性をもたらすには、ストレージ・コントローラの障害から保護し、ファームウェアの更新などのアクティビティのためのメンテナンス期間でさえも継続してアクセスを提供するために、アクティブ/アクティブ構成またはアクティブ/パッシブ構成のいずれかのデュアル・コントローラを備えたアレイをお勧めします。

冗長性を実現するために複数の SAN スイッチを使用する場合、推奨される一般的な方法は、各スイッチを個別の SAN ファブリックにし、不具合のある構成の変更を 1 つのスイッチにとどめ、両方のスイッチに影響してすべてのストレージ・アクセスが妨げられることがないようにすることです。

### 2.2.2 ネットワーク接続ストレージ (NAS)

これは、一般的に 10GB のイーサネットが使用可能であり、最高のパフォーマンスを実現するためには、10GB のスイッチおよびホスト・ネットワーク・インタフェース・カード (NIC) をお勧めします。

専用のインフラストラクチャを用意して、LAN の通常のネットワーク・トラフィックからトラフィックを分離することもお勧めします。こうすることによって、ホストとストレージの間の NAS のパフォーマンスを予測できます。

ジャンボ・フレームのサポートを含めて、ホストとストレージの間の十分な通信を提供することも必要です。

多くのネットワーク・インタフェース・カード (NIC) は TCP オフロード・エンジン (TOE) をサポートします。TOE のサポートは、普遍的に利点があると見なされているわけではありません。使用可能なサイクル (またはその欠如) に対するオーバーヘッドおよびゲインは、サーバの CPU によって大きく変化します。また、TOE のサポートは役に立つ期間が制限されます。これは、指定された NIC の TOE パフォーマンス・レベルにすぐにシステム処理能力が追いつくまたは多くの場合追い越すためです。

## 2.3 ファイル・システムの分離

パフォーマンスと復元可能性を高めるために、InterSystems IRIS 用に少なくとも 4 つの独立したファイル・システムを備えて以下をホストすることをお勧めします。

1. インストール・ファイル、実行可能ファイル、およびシステム・データベース (既定では、ライト・イメージ・ジャーナル (WIJ) ファイルを含む)
2. データベース・ファイル (およびオプションで WIJ)
3. プライマリ・ジャーナル・ディレクトリ
4. 代替ジャーナル・ディレクトリ

さらに、それとは別に WIJ ファイル専用のファイル・システムを構成に追加することもできます。このファイルは、既定で install-dir¥mgr ディレクトリに作成されます。該当するファイル・システムに、WIJ を最大サイズまで増加できるだけの十分な空き容量があることを確認してください。WIJ の詳細は、"データ整合性ガイド" の "ライト・イメージ・ジャーナリングとリカバリ" の章を参照してください。

注釈　UNIX®、Linux、および macOS プラットフォームでは、**/usr/local/etc/irissys** は InterSystems IRIS レジストリ・ディレクトリであるため、ローカル・ファイル・システム上に配置する必要があります。

重大なディスク障害が発生してデータベース・ファイルを破損した場合、バックアップからのリカバリにおいてジャーナル・ファイルが重要な要素になります。したがって、データベース・ファイルおよび WIJ が使用するデバイスから独立したストレージ・デバイスにプライマリ・ジャーナル・ディレクトリおよび代替ジャーナル・ディレクトリを配置する必要があります(WIJ の損傷によってデータベースの整合性が損なわれる可能性があるため、ジャーナルは WIJ から独立させる必要があります)。 代替ジャーナル・デバイスを使用すると、プライマリ・ジャーナル・デバイスでのエラーの後でもジャーナリングを続行できるため、プライマリ・ジャーナル・ディレクトリおよび代替ジャーナル・ディレクトリも相互に独立したデバイスに配置する必要があります。運用上の理由から、これらの異なるデバイスを同じストレージ・アレイ上の異なる論理ユニット (LUN) にすることもできますが、一般的には独立性が高いほど好ましいです (別々の物理ドライブを強く推奨)。別々のジャーナル・ストレージについての詳細は、"InterSystems IRIS データ整合性ガイド" の "ジャーナリング" の章にある "ジャーナリングの最善の使用方法" を参照してください。

ジャーナル・ディレクトリと WIJ ディレクトリは、インストール時には構成されません。これらのディレクトリを InterSystems IRIS のインストール後に変更する方法の詳細は、"InterSystems IRIS データ整合性ガイド" の "ジャーナル設定の構成" を参照してください。

InterSystems IRIS では、データベース・ディレクトリでのシンボリック・リンクの使用はサポートされていません。

注釈　現在のストレージ・アレイ (特に、SSD/フラッシュ・ベースのアレイ) では、前に推奨した種類の独立性は必ずしも可能ではありません。このような技術を使用する場合、パフォーマンスと復元可能性を高めるために、ストレージ・ベンダの推奨事項を参照して、これに従ってください。

## 2.4 バッファ型入出力と直接入出力

一般に、ほとんどのオペレーティング・システムでは 2 つの異なる入出力方式が用意されています。

- バッファ型入出力 – オペレーティング・システムが読み取りおよび書き込みをキャッシュします。プログラム (InterSystems IRIS など) またはユーザが別途指定しない限り、これが通常の動作モードです。
- 直接入出力、同時入出力、またはバッファなし入出力 – 読み取りおよび書き込みがオペレーティング・システムのキャッシュ処理をバイパスします。一部の InterSystems IRIS ファイルでより良いパフォーマンスおよびスケーラビリティ特性が得られます。

利用可能な場合、InterSystems IRIS は直接、同時、またはバッファなし入出力を利用します。ファイル・システムおよびマウント・オプションを選択する際は、ファイルのタイプに応じて、次の点に注意する必要があります。

### ジャーナル・ファイル

一部のプラットフォームには、このリリース用のドキュメント "インターシステムズのサポート対象プラットフォーム" の "サポートされているファイル・システム" に記載のとおり、最適なパフォーマンスを実現するための直接入出力または同時入出力のマウント・オプションの使用に関して特別な推奨事項があります。その他のプラットフォームでは、InterSystems IRIS はジャーナル・ファイルには適切に、自動的に直接入出力を使用します。特別な考慮事項は必要ありません。

### データベース (IRIS.DAT ファイル)

InterSystems IRIS は独自のデータベース・キャッシュを使用するため、データベース・ファイルに対してオペレーティング・システム・レベルのバッファ処理を行うことに優位性はありません。データベース・キャッシュに十分な容量を割り当ててください。また、お使いのプラットフォームでの直接入出力の最適な使用方法について確認してください。

- Linux ― InterSystems IRIS は、データベース・ファイルに直接入出力を使用します。

  これは、VxFS ファイル・システムを使用している場合、cio マウント・オプションを使用して同時入出力のファイル・システムをマウントすることによってオーバーライドできます。

- macOS ― InterSystems IRIS は、データベース・ファイルにバッファ型入出力を使用します。

- Windows ― InterSystems IRIS は、データベース・ファイルにバッファなし入出力を使用します。

- IBM AIX ― InterSystems IRIS は、cio ファイル・システムのマウント・オプションが使用されているかどうかにかかわらず、データベース・ファイルに同時入出力を使用します。

  注釈　AIX では、InterSystems IRIS の実行中に、データベース・ファイルを読み取るために外部コマンドが使用されるという特殊な構成では、その外部コマンドが失敗する可能性があります。これは、同時入出力のために、データベース・ファイルが InterSystems IRIS によって開かれているためです。一例を挙げると、高度なバックアップまたはスナップショット・ユーティリティがあるにもかかわらず、cp コマンドを使用して外部バックアップを実行する場合です。cio オプションを使用してファイル・システムをマウントすると、すべてのプログラムが同時入出力でファイルを開くように強制されるため、この問題が解決します。

### インストール・ファイルと実行可能ファイル

InterSystems IRIS は、これらのファイルにバッファ型入出力を使用します。直接入出力マウント・オプションをサポートするプラットフォームでは、これらのファイルにバッファ型入出力を使用することが推奨されます。

### 外部アプリケーション・ファイルおよびストリーム

通常、外部ファイルを使用するアプリケーションは、バッファされたファイル・システムに配置されたファイルを利用します。

## 2.5 noatime マウント・オプション

通常、このオプションを使用できる場合、ファイル・アクセス時間の更新を無効にすることをお勧めします。これは通常、さまざまなファイル・システムで noatime マウント・オプションを使用して実行できます。

# 3 InterSystems IRIS の CPU サイジングおよび拡張

InterSystems IRIS は、システムの全体的な CPU 処理能力を最大限に活用するように設計されています。すべてのプロセッサやプロセッサ・コアが同様であるわけではありません。クロック速度、コアあたりのスレッド数、プロセッサのアーキテクチャなど、表面上の差異があるだけでなく、仮想化の影響も異なります。



## 3.1 CPU の基本的なサイジング

アプリケーションはそれぞれ大幅に異なるため、CPU リソース要件の測定方法として、アプリケーションのベンチマーキングおよび負荷テストや、既存のサイトから収集したパフォーマンス統計ほど優れたものはありません。ベンチマーキングも既存の顧客パフォーマンス・データも使用できない場合は、以下のいずれかの計算から始めます。

- 100 人のユーザごとに 1 ～ 2 つのプロセッサ・コア
- 1 秒あたり 200,000 件のグローバル参照ごとに 1 つのプロセッサ・コア

重要　これらの推奨値は、アプリケーション固有のデータを使用できない場合の開始点にすぎず、使用しているアプリケーションに適しているとは限りません。アプリケーションのベンチマーキングと負荷テストを実施して、正確な CPU 要件を確認することが非常に重要です。

## 3.2 コアの数と速度のバランス

CPU コアの速度と数のどちらを選択するかについては、以下の点を考慮してください。

- アプリケーションが使用するプロセスが多いほど、並行処理と全体のスループットを高めるうえで、コア数の増加の効果が大きくなります。
- アプリケーションが使用するプロセスが少ないほど、高速のコアの効果が大きくなります。

例えば、きわめて多くのユーザが単純なクエリを同時に実行するようなアプリケーションでは、コア数が多い方が効果的ですが、比較的少数のユーザが計算集約型のクエリを実行するようなアプリケーションでは、数が少なくても高速のコアの方が効果的です。複数のプロセスがすべてのコアで同時に実行されているときにリソースの競合がないことを前提とすると、理論上は、高速のコアが多数あれば、どちらのアプリケーションにも効果的です。"メモリ要件の見積もり" に記載しているように、プロセッサ・コアの数は、サーバにプロビジョニングするメモリを見積もる際の 1 つの要素であるため、コア数を増やすと、追加のメモリが必要になる可能性があります。

## 3.3 CPU の仮想化に関する考慮事項

プロダクション・システムは、ベンチマークと現在の顧客サイトにおける測定に基づいてサイジングされます。共有ストレージを使用する仮想化では、ベア・メタルと比べて CPU オーバーヘッドがほとんど増加しないため、ベア・メタルの監視から仮想 CPU の要件をサイジングするのが有効です。

注釈　ハイパー・コンバージド・インフラストラクチャ (HCI) の導入については、HCI ストレージ・エージェントまたはアプライアンスのオーバーヘッドに対応するために、ホストレベルの CPU 要件の見積もりに 10% を加算してください。

個々の VM に最適なコア数を決定する際には、高可用性を実現するために必要なホストの数とコストおよびホスト管理のオーバーヘッドの最小化の間のバランスを取ります。コア数を増やすことによって、後者に反することなく、前者の要件を満たすことができる可能性があります。

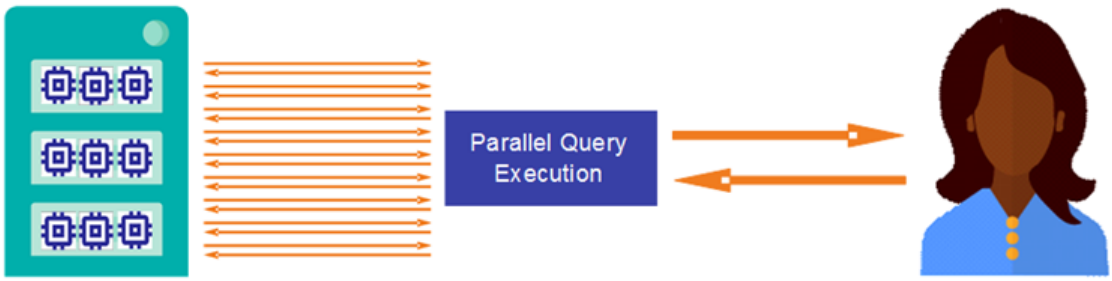
仮想 CPU の割り当てには、以下のベスト・プラクティスを適用してください。

- プロダクション・システム、特にデータベース・サーバは使用率が高いと想定されるため、最初に、物理 CPU と仮想 CPU の間で同等の仮定に基づいてサイジングする必要があります。物理 CPU が 6 つ必要であれば、仮想 CPU も 6 つ必要であると仮定します。
- パフォーマンスの最適化に必要な数を超える仮想 CPU を割り当てないでください。多数の仮想 CPU を仮想マシンに割り当てることもできますが、使用されていない仮想 CPU を管理するために (通常は小さな) パフォーマンスのオーバーヘッドが生じる可能性があります。ここで重要なのは、システムを定期的に監視して、仮想 CPU が正しく割り当てられていることを確認することです。

## 3.4 クエリの並列実行でのコア数の活用

CPU コアを追加してアップグレードすると、クエリの並列実行と呼ばれる InterSystems IRIS の機能を使用することで、向上した処理能力を最も効果的に活用できます。

図 1: クエリの並列実行




クエリの並列実行は、CPU コアごとに 1 つのプロセスを生成する、CPU の使用率を最大化するための柔軟なインフラストラクチャを基盤とし、大規模な集約を行う分析作業負荷など、データ量が多い場合に最も効果的です。

# 4 InterSystems IRIS 共有メモリの構成

前のセクションで説明した共有メモリ割り当ては、導入時に上記パラメータを構成マージ・ファイルに含めることで構成マージ機能を使用して実行します。以下に例を示します。

```
[config]
globals=0,0,5000,0,0,0
gmheap=165888
jrnbufs=64
routines=151
```

注釈　gmheap の値は KB 単位、その他は MB 単位です。globals の値内の複数のフィールドは、さまざまなデータベース・ブロック・サイズに対するデータベース・キャッシュのサイズを指定します。一般的には、ここで示すように、8 KB のブロック・サイズのキャッシュのみが指定されます。

ファイル内の設定は、導入時に既定の構成パラメータ・ファイル (CPF) にマージされるため、インスタンスの初回起動時には、指定されたパラメータ値と適宜割り当てられた共有メモリを使用して起動されます。(構成マージは、自動導入では大変便利で、異なるマージ・ファイルを適用することで、同じソースから異なる構成のインスタンスを導入することができます。)

導入時に構成マージを使用してこれらの割り当てが行われない場合、割り当ては導入の直後に以下のようにして指定できます (いつでも指定できるわけではありません)。

- iris merge コマンドを実行して構成マージを使用します。
- 管理ポータルを使用します。手順は次の場所に記載されています。
  - データベース・キャッシュとルーチン・キャッシュ ― "システム管理ガイド" の "データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て"
  - 共有メモリ・ヒープ ― "システム管理ガイド" の "共有メモリ・ヒープの構成"
  - ジャーナル・バッファ ― "データ整合性ガイド" の "ジャーナル設定の構成"
- 目的に合った ObjectScript クラスを使用します。クラス・リファレンス、および "構成パラメータ・ファイル・リファレンス" のパラメータのエントリを参照してください。ここで説明したメモリ・パラメータの場合、Config.config クラスになります。
- インスタンスの **iris.cpf** ファイル (install-dir/**mgr** ディレクトリにあります) を編集し、前のセクションで説明したパラメータの値を変更します。

これらのいずれかの方法で必要なすべての変更を行った場合、それらを適用するには、インスタンスを再起動します。

# 5 メモリ使用量の確認と監視

プロダクション・システムにおけるパフォーマンスの問題は、多くの場合、アプリケーションに必要なメモリが不足していることが原因です。1 つ以上の InterSystems IRIS インスタンスをホストするサーバにメモリを追加すると、データベース・キャッシュ、ルーチン・キャッシュ、共有メモリ、またはいくつかの組み合わせに、より多くのメモリを割り当てることができます。データベース・キャッシュが小さすぎて作業負荷の作業セットを保持できない場合は、クエリがディスクにフォールバックされて、必要なディスク読み取りの回数が大幅に増加し、深刻なパフォーマンスの問題が生じます。そのため、これが、メモリを追加する主な理由となることがよくあります。並列ジャーナリングやクエリの並列処理を使用する場合などの特定の状況では、共有メモリ・ヒープおよびルーチン・キャッシュを増やすことも役立つ場合があります。

インスタンスの起動の最後に、インスタンスの共有メモリ割り当ての概要を示す、以下のようなメッセージがメッセージ・ログに書き込まれます。これには、"InterSystems IRIS 共有メモリの構成" の例の値が組み込まれています。

```
11/06/21-10:59:37:513 (91515) 0 [Generic.Event] Allocated 5682MB shared memory using Huge Pages
11/06/21-10:59:37:514 (91515) 0 [Generic.Event] 5000MB global buffers, 151MB routine buffers,
  64MB journal buffers, 289MB buffer descriptors, 162MB heap, 6MB ECP, 9MB miscellaneous
```

注釈　バッファ記述子という用語は、グローバル、ルーチン、およびジャーナル・バッファに関連する制御構造を表します。"メモリ要件の見積もり" に記載したように、ヒープの数は、自動調整されるため、**gmheap** で指定した値より大きくなる可能性があります。

構成マージを使用して必要な共有メモリ割り当てで導入を行う場合、これらのメッセージを確認することで、意図した割り当てになっているかを確認できます。導入に続いて共有メモリを割り当てる場合は、変更後に開始する再起動に続いて、これらの割り当てを確認できます。

システムがテスト環境またはプロダクション環境で稼働したら、InterSystems IRIS 内の実際のメモリ使用状況を以下のように確認できます。

- インスタンスの共有メモリ使用状況を表示するには、管理ポータルの [システム使用状況モニタ] ページに移動し ([**システムオペレーション**] → [**システム使用**] の順に選択します)、[**共有メモリヒープ使用状況**] ボタンをクリックして、[共有メモリヒープ使用状況] ページを表示します。このページに表示される情報については、"監視ガイド" の "共有メモリ・ヒープ使用状況" を参照してください。

- InterSystems IRIS プロセスによる最大メモリ使用状況を大まかに見積もるには、実行プロセスのピーク数に、プロセスあたりの最大メモリ (bbsiz) の既定の設定である 262.144 MB をかけます。ただし、この設定を “無制限” を表す -1 に変更している場合 (″プロセス当たりの最大メモリの設定″ を参照) (これはインターシステムズがほとんどのプロダクション・システムに推奨する設定です)、これらのプロセスに必要な最大メモリ使用状況を見積もるには、より詳細な分析が必要です。InterSystems IRIS プロセスが使用するメモリの詳細は、″インターシステムズ製品のプロセス・メモリ″ を参照してください。

重要 システムが通常のプロダクション・ワークロード下にあるときに、システム・モニタによって、「Updates may become suspended due to low available buffers」というアラートや、「Available buffers are getting low (25% above the threshold for suspending updates)」という警告が生成される場合は、データベース・キャッシュ (グローバル・バッファ・プール) の大きさが十分でないため、キャッシュを増やしてパフォーマンスを最適化する必要があります。

# 6 データの圧縮とストレージ・コストの削減

ストレージ・コストは、特にクラウド・インフラストラクチャを使用している場合、アプリケーションのオーバーヘッドの大部分を占めることがよくあります。慎重に監視し、定期的にデータを圧縮することにより、これらのコストを削減できます。InterSystems IRIS では、データを圧縮するいくつかの方法を提供していますが、それぞれ利点とリスクが異なります。以下の表をガイドとして使用して、個々のアプリケーションに最も適した方法を決定することができます。

| 方法 | 利点 | リスク |
| --- | --- | --- |
| ジャーナルの圧縮 | ・ ストレージを即座に大幅に削減します。<br>・ ジャーナル暗号化に適合します。<br>・ 毎日のジャーナル・ボリュームは頻繁にデータベースの合計より大きくなるため、5:1 を超える圧縮率が可能です。 | ・ ジャーナル・リストアに必要なオーバーヘッドが増大する可能性があります。<br>・ ジャーナル・ファイルの内容によっては、圧縮率が低くなる場合があります。 |
| データベース内ストリーム圧縮 | ・ 長いテキストまたはバイナリ・ストリームの圧縮に適しています。<br>・ 暗号化データベースで機能します。<br>・ 空き容量の圧縮および削除を使用して、.DAT ファイルのサイズを縮小できる場合があります。 | ・ ストリーム圧縮は、ストリーム・データ型に対してのみ機能します。<br>・ 新規ストリームのみを圧縮します。<br>・ コードに対する更新または変更が必要となる場合があります。 |
| データの削除 | ・ データ・ボリュームが、削除された量だけ直接削減されます。<br>・ 空き容量の圧縮および削除を使用して、.DAT ファイルのサイズを縮小できる場合があります。 | ・ データは永久に削除されます。<br>・ .DAT レベル以外のデータの削除は複雑です。 |
| データのアーカイブ | ・ データには引き続きアクセスできます。<br>・ データはよりコストの低いストレージ・オプションに移行できます。<br>・ 空き容量の圧縮および削除を使用して、.DAT ファイルのサイズを縮小できる場合があります。 | ・ アーカイブされたデータにアクセスを試みる際に、遅延が大きくなったり、パフォーマンスが低下することがあります。<br>・ .DAT レベル以外のデータのアーカイブは複雑です。 |
| グローバルの圧縮 | ・ グローバル・データはより少数のブロックに統合され、データベースの空き容量が増加します。<br>・ スペースを節約しつつ、データを維持できます。<br>・ 空き容量の圧縮および削除を使用して、.DAT ファイルのサイズを縮小できる場合があります。 | ・ グローバル・ブロック密度が高いと、ランダムな挿入のパフォーマンスが大幅に低下する場合があります。<br>・ ランダムな挿入またはランダムな削除を伴うグローバルは、定期的に圧縮する必要がある場合があります。<br>・ ミラーリング時は、グローバルの圧縮は、両方のメンバで実行する必要のあるI/O負荷の高い操作です。 |

| | | |
|---|---|---|
| ストレージ・レベルの圧縮 | · 自動的に行われ、最も小さな労力で済みます。<br>· ミラー・メンバに同じストレージ・アレイを使用している場合、重複排除機能には大きな利点があります。<br>· ストレージ・レベルの暗号化に適合します。 | · ストレージ・レベルの圧縮はベンダから提供されるため、追加の費用がかかる場合があります。<br>· すべての追加費用がストレージの節約を上回る損益分岐点を考慮する必要があります。<br>· 暗号化データベースは、ストレージ・レベルの圧縮の利点を無効にする可能性があります。<br>· .DAT ファイルのサイズは変わりません。 |

データベース管理は、ストレージのオーバーヘッドを削減する重要なツールです。このトピックの詳細は、"ローカル・データベースの管理" を参照してください。