記事
Toshihiko Minamoto · 2021年1月21日 9m read

Caché データベースブロックの内部構造、パート 1

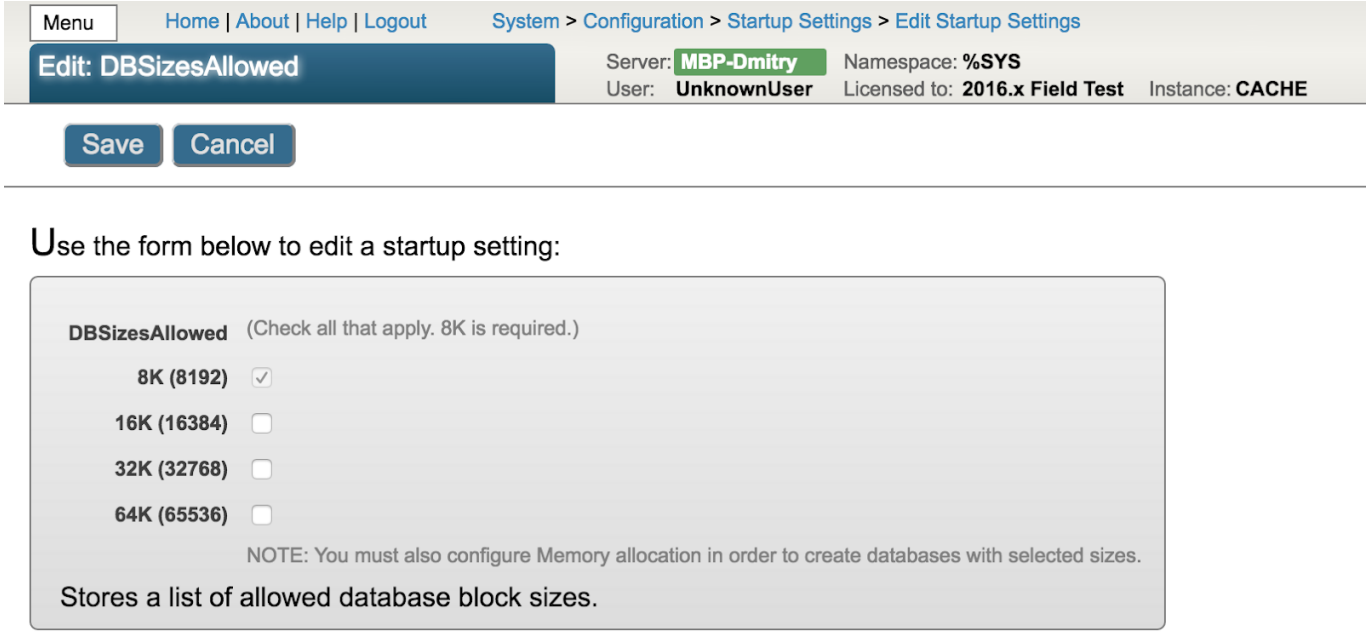
InterSystems Caché のグローバルは、デベロッパーにとって非常に便利な機能を提供します。
しかし、グローバルが高速な上に効率が良いのはなぜでしょう？

理論

基本的に、Caché データベースとは、データベースと同じ名前を持ち、CACHE.DAT ファイルを含んだカタログのことです。 Unix システムでは、このデータベースを普通のディスクパーティションにすることもできます。

Caché のデータはすべてブロックとして保管され、バランスド B* ツリーとして整理されます。 基本的にすべてのグローバルがツリーに保管されると考えると、グローバルのサブスクリプトはツリーの枝を意味する一方で、グローバルのサブスクリプトの値はツリーの葉として保管されると言えます。 バランスド B* ツリーと通常の B ツリーの違いは、前者の枝には $Order と $Query の両関数を使ってサブスクリプト (この記事ではグローバル) のイテレーションをスピーディに実行するのに役立つ適切なリンクがあり、ツリーの幹に戻る必要がないという点です。

デフォルトで、データベースファイルの各ブロックのサイズは 8,192 バイトに固定されています。
既存のデータベースのブロックサイズは変更できません。
新しいデータベースを作成する場合は、保管するデータ型に合わせて、16kB、32 kB、または 64 kB をブロックサイズに選択できます。
但し、すべてのデータはブロック毎に読み取られることを覚えておきましょう。つまり、1 バイトの値を 1 つだけリクエストする場合でも、システムはリクエストされたデータブロックの前にいくつかのブロックを読み取ります。 また、Caché はデータを再利用するためにデータベースブロックをメモリに格納するグローバルバッファを使うことと、ブロックサイズと同じサイズとグローバルバッファを使用することを覚えておきましょう。 対応するブロックサイズのグローバルバッファがシステムにない場合は、そのブロックサイズの既存のデータベースをマウントしたり、新しいデータベースを作成したりはできません。
実際使用するブロックサイズでメモリサイズを定義することをおすすめします。 データベースブロックよりも大きいバッファブロックを使用することは可能ですが、その場合、各バッファブロックには 1 つのデータベースブロック、もしくはさらに小さいブロックしか保管されません。



この画像では、8 kB のグローバルバッファのメモリがそれぞれ 8 kB のブロックで構成されるデータベースに割り当てられています。
このデータベースのブロックのうち、空でないものは、それぞれのマップが 62,464 個のブロック (それぞれ 8 kB のブロック) を定義するかたちでマップに定義されています。

ブロックタイプ

システムは複数のブロックタイプに対応しています。
各レベルにおいて、ブロックの適切なリンクは同じタイプのブロックまたはデータの終わりを意味する Null ブロックにポイントしている必要があります。

- **タイプ 9**: グローバルカタログのブロック。 通常このブロックでは、既存のすべてのグローバルがそれぞれのパラメーターと併せて説明されます。このパラメーターには、グローバルのサブスクリプトのコレーション (照合順序) が含まれています。大切なパラメーターの 1 つで、グローバル作成後には変更できません。
- **タイプ 66**: 高レベルポインタのブロック。
  このブロックの上に置けるのは、グローバルカタログのブロックのみです。
- **タイプ 6**: 低レベルポインタのブロック。 このブロックの上に置けるのは高レベルポインタのブロックのみで、これより下に置けるのはデータブロックのみです。
- **タイプ 70**: 高レベルポインタと低レベルポイントの両方で構成されるブロック。 このブロックは、対応するグローバルに保管される値の数が少ないため、複数のレベルのブロックを設ける必要がない場合に使用されます。
  このブロックは通常、グローバルカタログのブロックと同様、データブロックにポイントします。
- **タイプ 2**: 比較的大きなグローバルを保管するポインターのブロック。 値を複数のデータブロックに均等に割り当てるには、ポインタのブロックにレベルを追加すると便利です。
  このブロックは通常、ポインタのブロックの間に置かれます。
- **タイプ 8**: データブロック。 通常このブロックには、1 つのノードの値ではなく、複数のグローバルノードの値が保管されます。
- **タイプ 24**: ラージストリングのブロック。 1 つのグローバルの値がブロックサイズよりも大きい場合、その値はラージストリングの特殊ブロックに記録される一方で、データブロックのノードにはラージストリングのブロック一覧へのリンクが合計サイズと共に保管されます。
- **タイプ 16**: マップブロック。
  このブロックは、未割り当てのブロックに関する情報を保管するようにデザインされています。

典型的な Caché データベースの最初のブロックには、データベースファイルそのものに関するサービス情報、2 つ目のブロックには複数のブロックからなるマップが含まれます。 最初のカタログブロックは 3 番目 (ブロック #3) に位置付けされ、1 つのデータベースに複数のカタログブロックを保管できます。

この後は、ポインタブロック (枝)、データブロック (葉)、そしてラージストリングのブロックが続きます。 先ほども触れましたが、グローバルカタログのブロックには、データベースやグローバル設定に存在するすべてのグローバルに関する情報が保管されます (そうでない場合は、そのようなグローバルに置かれます)。
この場合、そのようなグローバルを説明するノードには、低レベルの Null ポインタが割り当てられます。
既存のグローバルの一覧は、管理ポータルのグローバルカタログから表示できます。
また、このポータルでは、削除したグローバルをカタログに保存 (照合順序を保存するなど) したり、デフォルトの、またはカスタマイズしたコレ−ションを設定した新しいグローバルを作成したりできます。

一般的に、ブロックのツリーは下の画像のように描くことができます。
赤く表示されているのは、ブロックへのリンクです。

データベースの整合性

Caché の現在のバージョンでは、データベースの最も重大な問題を解決しましたので、データベースのパフォーマンスが低下するリスクは極めて低くなっています。 それでも、定期的に ^Integrity ツールを使って自動整合性チェックを実行することをおすすめします。同ツールは、データベースページもしくはタスクマネージャーのマネジメントポータルを使って、%SYS ネームスペースのターミナルで起動できます。 自動整合性チェックは、デフォルトでセットアップされており、事前に定義もされているので、有効化するだけで実行できます。

整合性チェックでは、OSI モデルの下位層でのリンク検証、ブロックタイプの確認、適切なリンクの分析、グローバルノードと適用される照合順序のマッチングが行われます。
整合性チェックの最中にエラーが出る場合は、%SYS ネームスペースから ^REPAIR ツールを実行できます。 このツールを使えば、どのブロックでも表示できる上に、必要であれば、データベースを修正するなどの変更を施すことができます。

実践

しかし、ここまで紹介した内容は理論にすぎません。
グローバルとそのブロックの実態を見極めるのは未だに容易なことではありません。
現時点で、ブロックを表示するには、上述した ^REPAIR ツールを使う方法しかありません。
下に示すのがこのプログラムの典型的な出力です。

1 年前、私は新しいツールを開発するプロジェクトに着手しました。それは、データベースを破損させるリスクなしにブロックのツリーをイテレーションし、それらのブロックをウェブ UI
に視覚化し、その視覚化されたイメージを SVG または PNG に保存するオプションを提供するというものです。
このプロジェクトは名付けて CacheBlocksExplorer。ソースコードは、Github
からダウンロードしていただけます。

実装した機能には以下の通りです。

- システム内の構成済みのデータベースや単純にマウントされたデータベースを表示する。
- 情報をブロック別、ブロックタイプ別、右ポインタ別、リンク付ノードの一覧別に表示する。
- 前述のレベルよりも低いレベルのブロックにポイントしているノードに関する詳細を表示する。
- ブロックへのリンクを削除することにより、そのブロックを非表示にする
  (そのブロックに保管されているデータを損傷することはありません)。

To-Do リスト:

適切なリンクを表示する: 現在のバージョンでは、適切なリンクはブロックに関する情報の中に表示されていますが、矢印として表示する方がベターでしょう。

ラージストリングのブロックを表示する: 現在のバージョンでは表示されていない。

グローバルカタログのブロックを3 番目のブロックだけでなく、すべて表示する。

ツリー全体も表示したいのですが、何十万個もあるブロックとそれぞれのリンクを一緒にすばやくレンダリングできるライブラリは、まだ見つかっていません。現在のライブラリは、ウェブブラウザーにレンダリングしていますが、そのスピードは Caché が構造全体を読み取るスピードよりもかなり遅いです。

次回の記事では、私の自作ツール Cache Block Explorer を実際に使い、その機能をさらに詳しく説明するほか、使用例をいくつか紹介し、グローバルやブロックに関する実行可能なデータを数多く取得する方法を披露したいと思います。

#システム管理 #データベース #Caché

ソースURL:
https://jp.community.intersystems.com/post/cach%C3%A9-%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%83%96%E3%83%AD%E3%83%83%E3%82%AF%E3%81%AE%E5%86%85%E9%83%A8%E6%A7%8B%E9%80%A0%E3%80%81%E3%83%91%E3%83%BC%E3%83%88-1