記事
Toshihiko Minamoto · 2022年3月1日 5m read

2021.2 SQL 機能スポットライト - スマートサンプリング & テーブル統計の自動化

これは、適応性とパフォーマンスに優れた SQL エクスペリエンスを提供する 2021.2 SQL 強化機能に関する連載第 2 回目の記事です。
この記事では、前の記事で説明したランタイムプランの選択機能の主要な入力である **テーブル統計の収集におけるイノベーション**に焦点を当てます。

皆さんには次のことを何度もお伝えしてきました:

『Tune tableを実行しましょう！』

TUNE TABLE SQL コマンドまたは$SYSTEM.SQL.Stats.Table ObjectScript API を通じてテーブルをチューニングすることは、IRIS SQL が適切なクエリプランをはじき出すのに役立つテーブルデータの統計情報を収集することです。
これらの統計情報には、テーブル内のおおよその行数など、オプティマイザーが JOIN の順序（通常、最も小さいテーブルから始めるのが最も効率的です）などを決定する上で役立つ重要な情報が含まれています。
クエリのパフォーマンスについて InterSystems サポートに寄せられる多くの問い合わせは、TUNE TABLE を実行してもう一度試すだけで解消されます。このコマンドを実行することで、既存のプランが無効になり、次の呼び出しによって新しい統計が得られるためです。
サポートへの問い合わせから、こういったユーザーがテーブルの統計情報を収集していなかった理由が 2 つわかりました。テーブル統計について知らなかった、または本番システムでチューニングを実行した際のオーバーヘッドの余裕がなかったという理由です。 2021.2 では、この 2 つの理由に対処しました。

ブロックレベルのサンプリング

2 つ目の理由から始めましょう。統計を収集するコストです。 テーブル統計を収集するには多大な I/O が必要となるため、*テーブル全体をスキャンしているのであれば*、オーバーヘッドも高まります。 API ではすでに、行の一部のみサンプリングすることをサポートしていましたが、この操作にはかなりのコストがかかるというご意見をいただいていました。 2021.2 では、マスターマップグローバルをループすることでランダムな行を選択するのではなく、その下の物理ストレージにすぐにアクセスしてそのグローバルにカーネルが実際のデータベースブロックのランダムなサンプルを取得するように変更しました。
サンプリングされたこれらのブロックから、それらが保存する SQL テーブル行を推論し、通常のフィールド単位の統計情報構築ロジックに対応します。

これを大規模なビールフェスティバルへの参加に例えると、すべての通路を歩き、いくつかの醸造所のブースを選んでそれぞれのボトルをカートに入れるのではなく、単に主催者に依頼してランダムなボトルが入った木箱を渡してもらうので歩かなくても済む、というものです。
（実際のビール試飲会では、歩き回ったほうが適切ですが  ）。 酔いを醒ますために、以下に、ブロックベースのアプローチ（青い十字）に対する今までの行ベースのアプローチ（赤い十字）をプロットした単純なグラフを示しています。 これは、一部のお客様が TUNE TABLE を実行することを警戒している巨大なテーブルについては大きなメリットがあることを表しています。

ブロックサンプリングの制限はあまりありませんが、最も重要なのは、デフォルトのストレージマッピングでないテーブル（例: %Storage.SQL

を使用してグローバル構造をカスタムマッピングしているテーブル）では使用できないことです。
このような場合には、過去に機能していた方法である、行ベースのサンプリングに戻ります。

自動チューニング

オーバーヘッドに関する認識の問題が片付いたところで、お客様が TUNE TABLE を実行していなかったもう 1 つの理由について考えましょう。その存在を知らなかったという理由です。 それについて文書化することもできました（また、ドキュメントを改善する余地が常にあることは認識しています）が、この非常に効率的なブロックサンプリングは、私たちが長年求めてきたことを実行する機会であると捉えました。すべてを自動化するということです。 2021.2 からは、統計がまったく提供されていないテーブルに対してクエリを準備する場合、最初に上記のブロックサンプリングメカニズムを使用してそれらの統計を収集し、クエリプランニングに統計を使用し、以降のクエリで使用できるように、テーブルメタデータに保存します。

仰々しく聞こえるかもしれませんが、上記のグラフは、GB
サイズのテーブルでは、この統計収集の作業がわずか数秒で開始していることがわかります。 不適切なクエリプランを使用してそのようなテーブルをクエリしている場合（適切な統計がないため）は、事前に簡易サンプリングを実行するよりもはるかにコストがかかる可能性があります。 もちろん、これは、ブロックサンプリングを使用できるテーブルのみに行い、行ベースのサンプリングのみをサポートする特殊なテーブルの場合は、（残念ながら）統計なしで対処するしかありません。

他の新機能と同様に、皆さんの最初の体験に関する感想とフィードバックをお送りください。 この分野では、テーブルの使用状況に基づいて統計を最新の状態に維持するなど、自動化に関するアイデアは他にもありますが、そういった機能をラボ外部の使用体験に基づくものにしたいと考えています。

#SQL #リレーショナルテーブル #InterSystems IRIS

ソースURL:
https://jp.community.intersystems.com/post/20212-sql-%E6%A9%9F%E8%83%BD%E3%82%B9%E3%83%9D%E3%83%83%E3%83%88%E3%83%A9%E3%82%A4%E3%83%88-%E3%82%B9%E3%83%9E%E3%83%BC%E3%83%88%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AA%E3%83%B3%E3%82%B0-%E3%83%86%E3%83%BC%E3%83%96%E3%83%AB%E7%B5%B1%E8%A8%88%E3%81%AE%E8%87%AA%E5%8B%95%E5%8C%96