

Adaptive Server Anywhere で大きな IN リストを使用する場合の考慮点

この文書では、Adaptive Server Anywhere で大きな IN リストを使用する場合の考慮事項と、代わりにテンポラリ・テーブルが役に立つ場合を説明します。

フォーム `table.columnA IN (< item 1 >, < item 2 >)` の IN リスト述部は、SQL クエリによく見られます。IN リストは結果セットの条件を指定する簡単な方法で、フォーム `table.columnA = < item1 > OR table.columnA = < item2 >` の一連の OR 条件と同等です。実際、IN リスト要素一式が大きくならなければ、Adaptive Server Anywhere オプティマイザは、意味的に同等なので、一連の OR 条件を IN リストに自動的に変換します。この変換が実行されると、Adaptive Server Anywhere が IN リスト要素の処理に最適化を適用しやすくなります。同様に、Adaptive Server Anywhere は、述部が同一の左辺式を参照する場合は、2 つの IN リストをマージします。

大きな IN リスト述部 (特にアプリケーション・プログラムにより生じるもの) は、下記のような理由で問題があります。

- アプリケーション (クライアント) 中での構成に時間とメモリがかかる
- SQL 文が非常に大きくなる (クライアント/サーバ環境ではネットワークを通じてその SQL 文を渡さなければいけない)
- Adaptive Server Anywhere サーバ内で大量の RAM がその文に使用され、他の要求の効率に影響を与えることがある

例として、250,000 のリテラル定数 (整数) で形成される単体の IN リスト述部を持つ簡単なクエリを検査します。各定数が 5 桁 (バイト) だと仮定すると、SQL 文のテキストだけでおおよそ 1.25 MB になります。

ネットワークを通じて Adaptive Server Anywhere サーバに文が送信されると、最適化の前に構文解析と分析が行われます。文の構文解析は SQL 要求内で異なるトークンを表現するために内部データ構造の構築が必要です。この例のクエリでは、構文解析ツリー表現で IN リストのリテラル定数要素ごとに 40 バイトが必要です。これにより要求の同時実行のためにおおよそ 10MB の追加 RAM が必要になり、Adaptive Server Anywhere バッファ・プールで使用可能なメモリから 10MB が差し引かれることになります。

構文解析後、Adaptive Server Anywhere は要求のためにオプティマイザ・データ構造を構築します。また、オプティマイザによって代替案として選択された実行プランも構築します。幸い、IN リスト述部とその他の大部分の述部はこれら 2 つのフェーズ間で共有されるため、構築する必要があるのは、各 IN リスト述部のコピー 1 つのみです。ただし、Adaptive Server Anywhere データベース・サーバでは、IN リスト要素ごとに 88 バイトが必要なので、追加と

して 22 MB のデータベース・キャッシュが使用されます。合計すると、この文では約 33 MB のキャッシュが必要です。キャッシュに制限がある、または多数の同時接続がある環境では、このフォームのクエリはサーバのメモリ不足を引き起こしたり、使用可能なバッファ・プール領域量の制限により他のクエリのパフォーマンスに影響を与えたりすることがあります。

大きな IN リストを使用する代替の方法としては、その接続のテンポラリ・テーブルへ IN リストをロードし、次にクエリを再構成して、元のクエリのテーブルとテンポラリ・テーブルをジョインします。この方法では、Adaptive Server Anywhere がメモリの中ですべての IN リスト値を表す必要がありません。代わりに、Adaptive Server Anywhere ではネスト・ループ・ジョインやハッシュ・ジョインなどのハイパフォーマンスでメモリ効率のよいクエリ処理演算子を使用して、テンポラリ・テーブルに保存された値に基づくクエリ結果セットの条件を指定します。この 2 つの方法のトレードオフには、IN リスト値の数を指針として使用します。この数はシステム定義パラメータに依存しますが、テンポラリ・テーブルの使用を決定するには、経験的に IN リスト要素数が 100 から 1000 の間の場合となるでしょう。