

松尾 悦郎

株式会社ミガロ.

システム事業部 システム1課

RPGでパフォーマンスを制御 — 順次読み込みの方法と Delphi/400やJACi400との連携

どのようなシステムにも存在する一覧照会について
RPG プログラムで処理速度を制御するテクニックと
それを Delphi や JACi400 で連携する方法を紹介する。



略歴

1979年6月16日生れ
2002年広島大学理学部卒
2006年株式会社ミガロ、入社
2006年6月システム事業部配属

現在の仕事内容

主に JACi400 を使った Web アプリケーションの開発を担当しており、システムの要件定義から納品・フォローまでを行っている。

- はじめに
- 基本的な一覧照会
- 実践的な一覧照会
- 順次読み込みの手法
- Delphi/400プログラムとの連携
- JACi400プログラムとの連携
- 最後に

はじめに

今日、私たちが生活する社会にはたくさんの業種があり、いろいろなシステムが稼働している。同じ業種でも、企業によっては、全く違うアプリケーションが使われていることもある。それは、それぞれの企業の業務や取り組みが異なるので、仕方のないことであり当然である。1つとして同じシステムはない、と言っても過言ではないかもしれない。

このように多種多様なシステムが存在しているわけであるが、システム開発者が共通して考慮し、頭を抱えるポイントがある。それは「処理速度（パフォーマンス）」だ。「人間がコンピュータの応答時間に我慢できるのは3秒以内」という説を聞いたことがあるが、今では高速化が進み、検索ボタンを押して、表示されるまで10秒かかる照会プログラムはユーザーに使われない。一方で、ユーザーからの要望は難易度を増し、便利な機能を次々と求められるのである。

そこで今回は、どのようなシステムに

も存在する一覧照会の機能について、IBMのRPGプログラムで処理速度を制御するテクニックと、それをDelphiやJACi400（Webシステム）で連携する方法を紹介したいと思う。

基本的な一覧照会

詳細に入る前に、一覧照会プログラムについて、確認の意味で簡単にパターン別の仕組みを説明する。

A 5250のアプリケーション

サブファイルを使用し、対象のレコードをこのサブファイルに書き込む。画面の制約があるため、1画面で表示できない内容はPAGEDOWNやPAGEUPで表示する。

B Delphi/400のGUIアプリケーション

Delphi/400を使用したアプリケーションであれば、ワークファイル（中間ファイル）を使用し、対象レコードをいったんRPGでこのワークファイルに書き

出す。それをDelphi/400で画面に表示し、スクロールさせて全件を表示させる。

簡単な仕様であれば、RPGを使わずに、SQLで抽出することもある。

C Webアプリケーション

JACi400を使用したアプリケーションであれば、内部テーブルを使用し、対象レコードをいったんRPGでこの内部テーブルに保持する。その後、Webサーバーに対象のデータを送信し、JACi400を介してHTMLで作成された画面に表示させる。これも、スクロールさせて全件を表示させる。

以上が基本的な仕組みで、いずれもRPGで対象レコードを抽出するものである。

実践的な一覧照会

基本的な一覧照会プログラムの仕組みを確認したところで、いよいよ詳細な内容に入って行く。当然のことだが、実際

の場面ではいろいろと考慮しなければならない。代表的なものが対象レコード数である。

対象となるレコードが100件や200件であれば気にしなくてもよいかもしれないが、企業によっては、対象レコードが1万件や10万件、さらには100万件と膨大になることがほとんどで、この点を無視すると実際に使える（ユーザーが使用する）ものにはならない。

この場合、制御するポイントをまとめると、以下の2点となる。

●処理速度に関する、パフォーマンスの制御

検索を開始してから画面に表示し終えるまでの時間

●表示件数に関する、パフォーマンスの制御

1回の処理で画面に表示できる件数

表示件数での制御は、特に Web アプリケーションで考慮する必要がある。それは Web アプリケーションの場合、一度に通信できるデータの量を考慮する必要が出てくるからである。

(Web アプリケーションであれば一度に表示する件数を決め、ページを分けて次のレコードを参照するのが一般的)

いずれも一度に全件処理をするのではなく、順次に処理をする考え方である。それでは、これらのポイントをどのように RPG で実装するか、その方法を以降で具体的に説明していく。

順次読み込みの手法

これまでの5250アプリケーションの開発で工夫できる順次読み込みの仕組みを用いて、Delphi/400やJACi400(Web)で活用するための実装方法を紹介している。

処理速度のパフォーマンス制御

前述した通り、100万件のデータを一度に抽出して画面に表示すると、処理に時間がかかってしまうのは簡単に想像できると思う。この場合、1回の検索処理の最大時間を設定することで対応する。

つまり、全体の抽出処理を複数回に分割してしまうのだ。分割することでユーザーのストレスを軽減する。と同時に、

必要なデータを確認したら、以降の処理をせずに終了することもできる。業務効率が上がるはずだ。

さっそく、実際のプログラミングのポイントの説明しよう。

①設定時間の計算

処理の開始ポイントを決め、その時点でのタイムスタンプを内部で保持しておく。この際、開始ポイントは、プログラム起動時でも、ファイルのREAD開始時でもかまわないので、開発しやすいようにルールを決めておく。

そして、繰り返しでREADするたびにタイムスタンプを取得して、開始ポイントと比較するのである。比較した結果、設定時間を超えていなければ処理を継続し、時間を超えていれば処理をいったん中止すればよい。非常に簡単だ。

②読み込み処理を終了するタイミング

設定時間を過ぎたら処理を中止するのだが、ここで次回のために、続きの開始ポイントを保持しておかなければならない。また、その開始ポイントは、読み込んだ最後のキーを持って、次回開始時のキーを持ってもいい。ただし、保持するタイミングに気をつけなければならない。

読み込んでいるファイルが1つで、キーがUNIQUEであれば、そのまま終了しても問題はない。しかし、複数ファイルを参照し、見出しと明細のような組み合わせで処理している場合は、設定時間を過ぎたからといって即時処理を中止すると、本来対象として表示すべきレコードが漏れてしまったりする。

このような場合は、見出しファイルのキーに紐づく明細ファイルの内容をすべて処理し終えてから、読み込み処理を終了しなければならない。

以上のことを考慮して、処理終了のタイムスタンプを取得する場所が、適切なタイミングになるようにうまく設定する。

③次回開始キーの保持

最後に、次回開始キーを保持する方法である。キーテーブルを作成し、そこに書き込む。

読み込み処理終了時にキーテーブルに対して更新し、続きの読み込み処理を行

う時に、キーテーブルを参照し、読み込み開始ポイントをセットするのである。

この場合、一覧照会画面を閉じる時にキーテーブルをクリアするか、画面起動時にクリアする処理を忘れないようにしてほしい。

以上が、処理速度のパフォーマンス制御のためのポイントである。【図1】

その他の細かいこと、例えば未処理のレコードがあるのか、全件処理を終えているのかを表示するなど、便利な機能も実装していただきたい。

表示件数のパフォーマンス制御

次に、表示件数（データ量）において、パフォーマンスを制御する方法について説明しよう。

これは、Webアプリケーションの一般的な仕様で、インターネットの検索ページを思い出してもらおうとイメージしやすいと思う。【図2】

よく見かけるのは、1ページに20件の検索結果が表示され、画面の一番下に残りの検索結果がページ番号として表示されており、そのページ番号をクリックすると画面が変わり、別の20件の検索結果が表示されるというものである。これは、一度に通信できるデータ量を考慮しており、さらには安定したレスポンスを提供することができる。

ここでは、JACi400を使ったWebアプリケーションを例に、表示件数のパフォーマンス制御のためのプログラミングポイントを説明する。

①レコードの抽出

JACi400の場合、プログラムはすべてRPGで組まれることになる。一覧照会は、内部テーブルを使って処理をする。

そのため、あらかじめ一覧表示用の外部記述ファイルを作成しておき、その記述を参照するように内部テーブルを宣言する。そして、その内部テーブルに演算命令の「OCUR」を使用して、抽出したレコードをすべて保管する。

②内部テーブルから、表示するレコードを取り出す

対象レコードをすべて内部で保管してしまうと、次は、画面に表示するレコードのみを抽出する。

図1

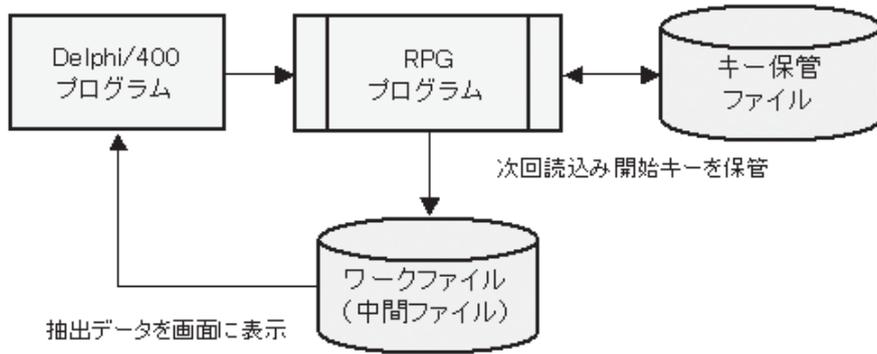


図2

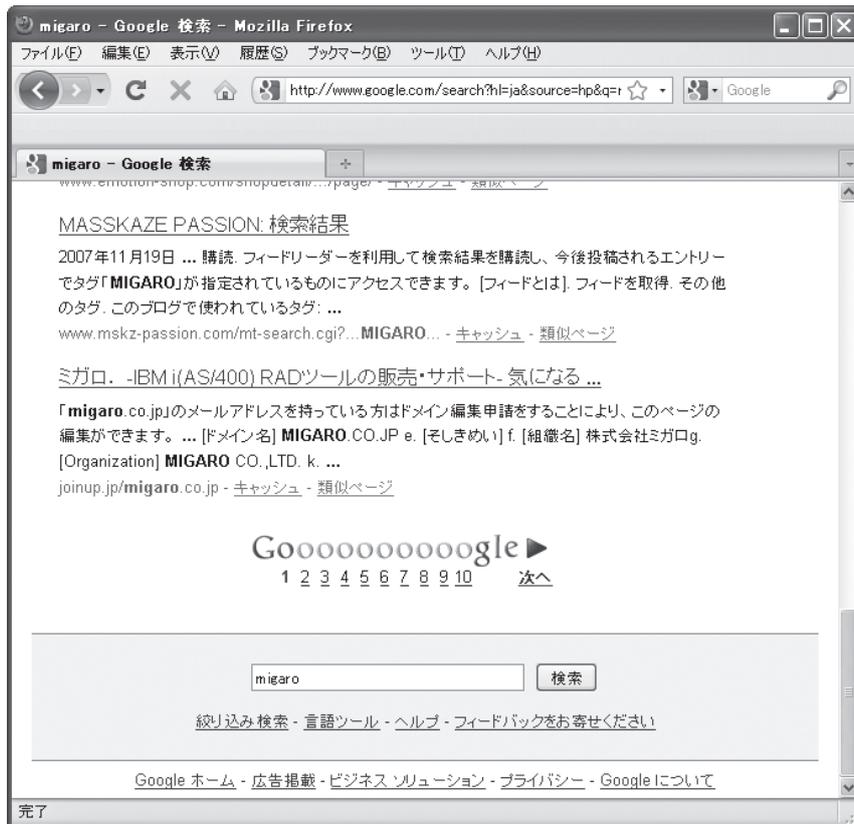
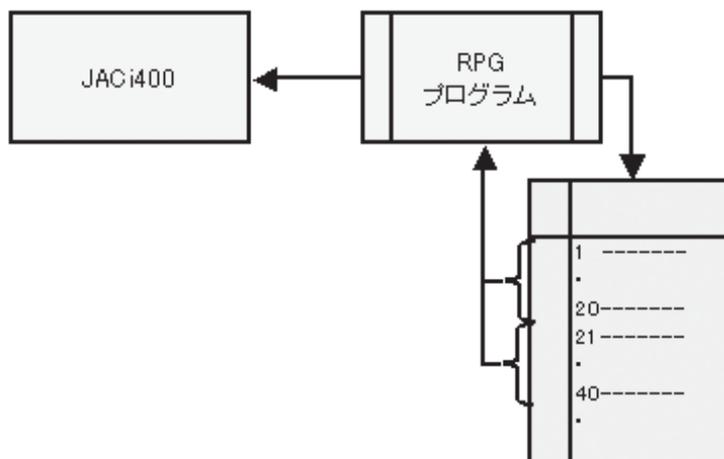


図3



1画面に20件を表示する場合、内部テーブルの1番目から20番目までを最初に抽出する。その方法も、テーブルに保管した時と同様に「OCUR」命令を使用し、演算項目1に1から20までを繰り返してセットして、画面に表示する内容として取り出す。

JACi400はデータを取り出してしまえば、画面への送信はあらかじめ準備されているので、新たにロジックを組み込む必要はなく簡単である。

また、画面へデータを送信した後に、ページ番号を選択し他ページに遷移する場合も、内部テーブルの何番目から何番目と指定して取り出すだけなので、やはり簡単である。【図3】

以上が、表示件数のパフォーマンス制御を実装する際のポイントである。

なお、内部テーブルを使用することで、次のようなメリットがあることも強調しておきたい。

- ファイルの読み込みや書き込み処理をしないので、ワークファイル（中間ファイル）に対象レコードを保管するよりも早い。
- 簡単に早く対象レコードを抽出することができるので、ページの画面遷移が早い。

しかし、内部テーブルを使用するとレコードの上限が決まってしまうということもあるので、選択は実際の状況に合わせて判断しなければならない。

Delphi/400 プログラムとの連携

RPGでパフォーマンスの制御ができれば、Delphi/400のプログラムと連携する方法を紹介しておく。

基本は前述したとおり、RPGで作成されたワークファイルからDelphi/400でデータを取得して画面に表示するもので、順次読み込みをする場合は、Delphi/400の画面で「続きの検索」処理を組み込む必要がある。

まず画面に「検索」ボタンと「続きの検索」ボタンを用意する。画面起動時は「検索」ボタンを有効にし、「続きの検索」は使用不可の状態にしておく。検索を実

行するとRPGプログラムを呼び出すのだが、その時にRPGから続きの処理の有無をパラメータでもらうようにする。

例えば、「1」：続き有り、「2」：処理完了、「9」：対象データなしと設定し、この値によって、用意したボタンを制御するのである。続きがあれば「続きの検索」ボタンを有効にし、続きがなく完了であればそのまま使用不可とし、処理完了の内容を表示するといった具合だ。

とてもシンプルな内容なので、ぜひ試していただきたい。

JACi400 プログラムとの連携

次に、JACi400での実装方法を紹介する。

JACi400では前述したとおり、表示件数で制御することが一般的となる。画面で表示する件数を固定するか、画面でユーザーが選択できるようにする。

ユーザーが選択する場合も、プルダウンリストで20件、50件とリストを決めておくほうがよいだろう。その画面の表示件数をもとに、対象の全レコード数からページ数を計算し画面に表示する。表示件数が20件、対象レコードが80件であれば、画面に1から4ページを選択できるようにするのである。

そしてページ番号を選択すると、JACi400ではアクションコード（設定したコード）がRPGに渡されるので、そのコードで内部テーブルから抽出するレコードを判断し、必要なぶんだけ画面に送信するのである。

画面に表示する1レコードの項目数が多い場合は、同じ20レコードでも通信するデータ量が増え、処理に時間がかかってしまうので注意が必要であること補足しておく。

最後に

以上、一覧照会のテクニックとして紹介したが、処理速度（パフォーマンス）に悩む開発者の方々に少しでもヒントになれば幸いである。RPGとDelphi/400、JACi400をうまく連携させて、今後もより使いやすいシステムを提供し紹介していきたいと思う。

現在の仕事内容（詳細）

主にJACi400を使ったWebアプリケーションの開発を担当しており、システムの要件定義から納品・フォローまでを行っている。
以前はRPGプログラムの開発を担当していたこともあり、JACi400の開発フェーズでは、RPGプログラム開発の管理を行っている。また、HAツールである*noMAXの技術サポートも担当している。