

インターネットを渡り歩いていると次のようなコメントをよく見かける。

①Access は 3、4 人で使う分にはいいけど、それ以上で使うのなら SQL Server だ。
--

②Access は SQL Server より遅い。

などなど。実は両方とも間違いとも言い切れないが、Access や SQL Server を正しく理解した上での発言かどうかは大いに疑問が残る。

弊社のお客様を例にとっても Access 単独で作成されたシステムだが 4 台以上のクライアントで利用しているシステムは数多くある。中には 10 台以上のクライアントが接続された販売管理システムもあるし、複数のマーケット店を統括する本店などでは 4 台のクライアントから毎日 2 時間程度で 1000 枚もの仕入伝票を入力しているところもあるがまったくもって問題はない。

しかしながら弊社のお客様にはデータベースシステムに SQL Server、クライアントには Access mdb または Access adp のプログラムファイルを配置した連携システムもある。こういった話をすると余計に Access と SQL Server の選択に迷いが出てきそうだがそれが現実である。

そもそも両者を同列に比較すること自体がおかしい。なぜなら、Access はデータベース管理能力とアプリケーション開発機能の両面を持つが、SQL Server にはアプリケーション開発機能はないからだ。SQL Server は他の開発ツールで作成されたアプリケーションから呼び出される形で利用される。

とはいえ、一般のものからすると両者の違いが気になるのも事実だろう。そこで今回はいろいろな角度から両者の比較を行ってみることにした。

結論から先に言うと、実は接続されているクライアント台数はそれほど問題ではない（ただしクライアント数が 10 台前後での話）。問題となるのは、

ネットワークのトラフィックだ。

次の表を参照されたい。

データベースシステム	データベースエンジンが動作する場所
Access※	クライアント側
SQL Server	サーバー側

※ただし、ここでは Access のMDB ファイルを使用したケースをいう

こういう風になると、

『えっ？Accessのデータだってサーバーにおいて利用できるはずだけど・・・』という声が聞こえてくるが、ここで問題としているのはデータの置き場所ではなく、そのデータの中から検索や更新などの処理を行っている作業、すなわちデータベースエンジンが稼働している場所だ。

たとえば、次のような実験を行ってみる。

実験：サーバーに配置された 10000 レコードのデータから 1 レコードの値を検索する処理

実験結果は下記の通り。

Access の場合

サーバーに置かれた全レコードをクライアントにダウンロードしながら、該当レコードが無いかどうかをクライアントの CPU（目の前のパソコンという意味）で調べはじめる。

※この検索プログラムのことを JET エンジンなどと呼ぶこともある。

たった 1 レコードの答えを得るためにすべてのデータが一度はネットワークを流れて目の前のパソコンにまでやってくるということになる。

SQL Server の場合（ただし、ストアードプロシージャを利用しない場合）

SQL Server はクライアントに配置されたアプリケーションから呼び出される形で利用される。クライアント側のアプリケーションは VB や Access ADP、またはその他の開発言語で作られたりする。

クライアントのアプリケーションは検索の条件文（SQL 文）をサーバーで待ち構える SQL Server に投げる。この条件文はわずか数十から数百文字程度。そしてこれを受け取った SQL Server はサーバー側の CPU で検索処理を行う。

答えが見つかったらその結果だけ（1 レコード）をクライアントに送り返す。

※ストアードプロシージャを利用する場合は、SQL をサーバー側に名前を付けて保存しておき、クライアント側からはその名前呼び出して利用する。

すなわち、

ネットワーク上に流れるデータ量にその違いが最も表れる！！

のだ。

このネットワーク上のデータの混み具合を『トラフィック』というが、実はこのトラフィックがシステムのレスポンスに大きく関係しているのだ。

※トラフィックはなにもシステムに関係する PC だけで決まるものではない。そのネットワークに接続されたすべての PC でどのような処理を行っているかで決まる。この件に関しては後述する。

たとえば、上記の実験を 10 台の PC から同時に行ったらどうだろう？

Access の場合、

10000 レコード×10 台=100000 レコード

以上のデータが同時にネットワーク上に流れることになる。

一方、SQL Server では

クエリ数十バイト×10 台+結果の 1 レコード×10 台=数百バイト～数千バイト

程度がネットワーク上を流れることとなるのだ。

以上のようにトラフィックを考慮すると断然 **SQL Server** を利用したクライアントサーバー方式が有利であることが分かるが、1つの **CPU**（最近では複数の **CPU** を有するサーバーも見かけるようになったが）を複数のクライアントが同時に利用するので負荷が集中するということも一応頭に入れておきたい。

ここで見方を変えて、たとえば販売管理システムを 10 台のクライアントで利用することを考えてみよう。

このプログラムで同時に行われる処理といえば、

- ・ 売上の同時入力
- ・ 得意先マスターや商品マスターなどの同時参照
※これらマスターの同時入力はほとんど考慮する必要はない
- ・ 売上台帳の検索や集計処理、または参照

などが考えられる。

Windows クライアントにはキャッシュメモリーが機能している。キャッシュメモリーとはハードディスクのデータを一度読みに行くとその内容が特定のメモリーに複写され、二度目以降からはハードディスクではなくメモリーがアクセスされる仕組みをいう。これによりデータの読み出し速度は飛躍的に高速となる。

だが、レコード追加作業においてはキャッシュメモリーは無関係だ。レコード追加を行うためにはその都度サーバーのデータが直接処理されることとなるのだ。

1 枚の伝票入力にどれだけの時間が必要とされるのかを考えてみるとよい。それはどんなにすばやいデータ入力ができたとしても 10 秒以上は必要となる。仮に 5 明細の伝票入力だとすれば、1 台のクライアントからは 2 秒に 1 レコード（1 明細）のレコード追加となる。すなわち、10 台で高速に伝票入力が行われたとしても 1 秒間にたった 5 レコードの追加となるわけである。

データベースシステムであればどんなものであっても 1 秒間に数十から数百レコードの追加処理が可能だ。だから、クライアント 10 台程度のデータ入力作業であったとしても、それほど問題にはならないというわけだ。

売上入力画面では得意先マスターや商品マスターを参照することが多いが、このような場合にもキャッシュメモリーが利用される。よって、マスターの参照が同時に行われたとしてもそれほど問題にはならないということがお分かりであろう。キャッシュメモリーがヒットすれば、ネットワークのトラフィックに影響はないのだ。

では、売上台帳の参照はどうだろうか？

通常売上データは膨大なものだ。すべての得意先の何年分ものデータが蓄積されているので場合によっては数百メガバイトとなることもありえるだろう。そういった膨大なデータを参照するとキャッシュメモリー上には展開し切れなくなる。だから、売上台帳の参照が複数のクライアントから頻繁に行われるとしたらネットワークのトラフィックは確実に上がっていく。さらにいえば、画面上に過去の売上データをブラウズ（サブフォームなどでスクロールしながらデータを確認すること）できるような処理を作ったとしたら、それは

いくら **SQL Server** であったとしても例外ではなく理屈は同じで、ネットワーク上にそれらのデータは流れてくるのだ。

売上入力画面には得意先マスターや商品マスターを参照するコンボボックスが必ずといっていいほど存在する。コンボボックスを開いてそこに得意先マスターが数百件ブラウズできるということは、サーバーからデータをダウンロードしなければ絶対に表示できない。もちろん、**SQL Server** を利用したシステムであっても同じだ。

実は、**Access** 販売管理システムに 10 台のクライアントがぶら下がっていても問題がないというのはネットワークのトラフィックがそれほど上がっていないからだ。

前述の、4 台クライアントで 2 時間に 1000 枚の伝票入力するシステムでは入力画面上にコンボボックスをほとんど置いていない。台帳類の参照もない。なぜなら、トラフィックの問題をはじめから予測していたからだ。

ちなみに、**Access** などでコンボボックスのデータがセットされるのはフォームが開かれたときだ。すなわち、フォームの開け閉めを頻繁に行うとトラフィックは上がる。だから開け閉めをできるだけ少なくさせることも重要である。

※プログラム **MDB** ファイルをサーバーではなくクライアント側に置かねばならない

逆に言えば、**SQL Server** を使ったシステムにおいてもシステムの設計が悪いとレスポンスは上がらない。

開発者はシステムの設計時にこれらのことを総合判断して **SQL Server** を利用するかどうかを決定する。

前述のように、トラフィックを考える場合、システムに関係しないクライアントの挙動も把握しておく必要がある。たとえば、最近ではウイルスチェックプログラムをクライアントにインストールすることも多いが、あれも問題を含む。何台ものウイルスチェックプログラムが同じ時間帯にメーカーのサーバーから最新のプログラムをダウンロードするとトラフィックは一挙に跳ね上がる。

ここまでの話を見ていると、やはり **Access** が不利のようにも思えるかもしれないが私は決してそうは思っていない。再度述べるが、**Access** はプログラム開発能力とデータベース管理能力の両面を備えている。この点においてコストパフォーマンスは非常に高い。

たとえば、**SQL Server** をバックエンドに配置し、フロントエンドにはクライアントアプリケーションを用いる方法では **Access** 単体での開発に比べるとどうしてもコストは上がる。また、クライアントアプリケーションの開発を自前でやろうとするなら断然 **Access** に軍配が上がる。データベースの管理面においても **Access** のほうがはるかに簡単で分かりやすい。エンドユーザーが将来自分たちの手でシステムに変更を加えていきたいと考えているのなら断然 **Access** をお勧めする。

逆に、データベースエンジンとして SQL Server を選ぶとしたら次のような理由が考えられる。

データの許容量	Access の最大ファイルサイズは 2GB。SQL Server は TB(テラバイト)
多彩なりカバリー機能	SQL Server にはログファイル(クライアントのアクセスを記録したファイル)管理機能があり、復旧時に利用することでトラブルが発生した直前の状態にまでデータベースを復元することが可能 稼働中でもバックアップをとることが可能
信頼性	接続クライアントが数百台以上という前提で作られているので堅牢 ※AccessMDB もめったに壊れないが・・・ ログファイルがあるので誰が何をしたか追跡できる。 データベースを簡単には持ち出せなくなる データベースのアクセス権限をユーザーごとに細かく設定することができる
拡張性	インターネットや VPN を利用した多彩な利用が可能

Access が遅い遅いと嘆いている方にビットからのアドバイス。

クライアントのメモリーを増やす	キャッシュメモリーを増やすためにもメモリーをできるだけ確保したい
MDB ファイルを時々最適化する	MDB ファイルは使用していると徐々にファイルサイズは大きくなっていく。ファイルを最適化することで、余計な部分が削除され、ファイルサイズは元に戻る。 さらに、最適化を実行するとインデックスファイルも最適化され、たとえば 1.2 ギガのデータでそれまで 2 時間もかかっていた集計作業が最適化後に 10 分で完了したという例もある。 プログラムとデータをともに扱う MDB ファイルで起動時間が長いと感じるならば、定期的な最適化は必須だ
プログラムファイルとデータファイルを分離させる	1つの MDB ファイルでプログラムとデータの両方を受け持っているのであれば、それらを分割して使用せよ。プログラムのメンテナンス性も格段に良くなる。
プログラム MDB(MDE)ファイルをクライアント側に配置する	すべてのクライアントに分割したプログラム用の MDB(MDE)を配置し、データファイル MDB はできるだけ高速なサーバーに配置せよ
プログラム MDB ファイルを MDE に変換する	さらに高速化を期待するのなら MDB を MDE に変換せよ。変換することでファイルサイズは小さくなり、プログラムのレスポンスは上がる。
ネットワーク上の他のクライアントで行われている作業をチェックする	他のクライアントがトラフィックを上げるような処理を行っていないかをチェックすること

さて、Access と SQL Server どちらを選ぶかはその目的で自ずと決まる。
お金が有り余っているのなら話は別だが、オーバースペックだけは気をつけたいものだ。
ビットの独り言：クライアント 10 台以下なら Access で充分だ！！

以上