Web サイトのバックエンドとしての PostgreSQL

株式会社サンブリッジテクノロジーズ
(http://www.sunbridge.com/tech/)
開発本部 プロジェクトマネージャ
酒井 勉

概要

無償データベースの PostgreSQL を使用してシステムを構築する事例が増えてきている。どのような分野に PostgreSQL が適用できて、商用データベースとのすみ分けはどうなっていくのか。本プレゼンテーションでは、PostgreSQL の適用分野として「Web アプリケーションのバックエンド」を想定し、商用データベースとの比較を交えながら、アプリケーション開発と運用の両側面から PostgreSQL 採用のメリット、デメリットを検討する。

Linux 普及との類似性

1990 年代後半になって、実運用に耐えうるオペレーティングシステムとして Linux がにわかに脚光を浴びた。Linux ディストリビューションを開発、販売する会社が乱立し、ハードウェアベンダはこぞって Linux サポートを打ち出したのは記憶に新しい。当時、Linux を使用してシステムを構築するメリットと言われていたのは、「ソースが公開されているので、いざとなれば自分でバグ修正や機能アップしながら使用できる」、「ＰＣサーバ向けの商用オペレーティングシステムよりもメンテナンス性も安定性も高い」、「オペレーティングシステムが無償なのでシステムの価格が抑えられる」等であり、これらの評価は無償のオープンソースデータベースである PostgreSQL にもそっくり当てはまると考えられる。

Linux はそれまでの商用オペレーティングシステムを完全に置き換えてしまったわけではなく、いわゆる「Linux 向き」の適用分野でのソリューションとして安定した地位を獲得してきている。例えば、Web サーバや Ftp サーバ、Mail サーバ等、複雑かつ高価なパッケージソフトウエアを必要とせず、その反面、パフォーマンスと可用性が安価に提供できることが要求される分野での基本コンポーネントとして、また、柔軟にシステムを改変でき、対応アーキテクチャも多いことから組み込み系のオペレーティングシステムとしてである。

同様に、PostgreSQL も、商用データベースを完全に置き換えてしまう夢のデータベースではなく、特定の分野において商用データベースよりも適していると認知され普及していくものと推測される。弊社のこれまでのシステム構築経験上から、PostgreSQL は Web サイトのバックエンドストレージとして非常に適したデータベースであると考える理由を以下で説明する。

Web サイト構築に必要なコンポーネント

ここでは Web サイトの裏側について簡単に説明する。まず、ブラウザからの http によるリクエストを処理する Web サーバと呼ばれるコンポーネントがあり、静的なドキュメントの要求であればここまでで完結する。ユーザが入力した情報を処理する必要があれば、perl や php、java 等で記述された Web アプリケーションが適切な計算を行い、結果を Web サーバ経由でユーザに提示する。シンプルなシステムであれば、これだけのコンポーネントでも、例えば、簡単な会員登録、ショッピングサイトや掲示板等、様々なサービスを提供することが可能である。このようにして作成されたアプリケーションは、ユーザからの情報をローカルのファイルに保存することになるが、ここで問題が発生する。まず、ファイルシステムの構造にもよるが、一般的にファイルの数が増えてくるとアクセスに必要な時間が急激に増加する。サイトが反映すればするほどユーザへの応答時間が増加し、サービスの質は下がるがシステム的に解決することはできない。ハードウェアの性能をあげる事で解決しようとしても、２倍、３倍のパフォーマンスを得るためには多額の投資が必要で

ある。同一のシステムを追加してこの問題を解決しようとする手法に「ミラーリング」があるが、頻繁に更新される情報を複数のシステム間で同期させておく事は技術的に困難である。そのような場合には、データはデータで別システムに保存し、複数の Web サーバから参照、更新するのが一般的であり、データの保存には RDBMS が使われる事が多い。通常データベースサーバは直接アクセスされることはないので、普通にブラウザから Web サイトをアクセスしている限り、ユーザがデータベースサーバの存在を意識することはないが[1]、ある程度複雑なシステムの裏側ではデータベースが使用されていると考えてほぼ間違いない。

システム選定のポイント

Linux 普及以前では Web サーバのプラットフォームとして Microsoft Windows または商用 Unix(特に Sun Solaris) を使用するのが一般であった。オペレーティングシステムが決まると、アプリケーションを稼動させる仕組みもほぼ決まってしまう。Windows では Web サーバに IIS を使用し、ASP や ColdFusion でアプリケーションを記述するのが定番である。データベースは Oracle や MS SQL Server が使用されることが多い。Unix の場合、Web サーバは Apache が大半ではあるが、アプリケーションについては柔軟性があり、C や perl で記述された CGI や Java Servlet 等が良く見られる。データベースは Oracle の場合が多い。

データベースを選定する際に、システム発注側からは実績とサポートを重視して商用 RDBMS を指定するケースが多い。オペレーティングシステムとして、無償、無保証の Linux を歓迎するにもかかわらず、データベースについてはサポートがないと心配というのは少々矛盾しているように思われる。そもそも商用データベースとはいえ、Linux 上での稼動実績が多いのかどうかは疑問である。特に、Web 等の不特定多数ユーザからピーク時には一気にアクセスが集中するような使われ方で Linux と商用データベースを組み合わせているシステムはどれほどあるのだろうか。大規模なアクセスを想定するのであれば、商用 UNIX と商用データベースを組み合わせるであろうし、逆に、安価でそこそこの安定性があれば良い、という要件であれば Linux と PostgreSQL を組み合わせる方が納得のいくシステム選定である。ただし、後者には PostgreSQL に実装されている機能で十分システムが構築できるという前提が含まれており、商用データベースでしか実現できないような機能が必須であるならば商用データベースを採用し、商用オペレーティングシステム上で稼動させるべきである。

---

[1] データベースサーバで発生したエラーがそのままユーザに返されてしまうサイトが良くあるが、システムの構成がわかると特定の製品のセキュリティホールをついて不正アクセスされる可能性があるので好ましくない。

Web サイトに適した PostgreSQL の機能

まず Web アプリケーションは一体どのような作りになっているのか、弊社の構築・移行実績から考察したいと思う。Web サイトのシステムコンポーネントは、前述の通り、外部に公開された(時には複数の)Web サーバと、外部からは直接アクセスができないようにしたデータベースサーバで構成する事が多く、一箇所のホスティングセンター内でシステムが完結している場合がほとんどである。例えば業務システムのように支社間で独立したデータベースを分散して持っていて、同期を取りながら更新する等の高度な要求はほとんど考えられない。社内業務用の既存システムと接続するということも考えられるが、特にプログラム的にデータベース同士を結合せず、業務システムでバッチ処理によりデータをエクスポートして Web サイト側に持ってきているという例が多々見られる。

また、データベースに格納されるデータについては、圧倒的に文字列型が多い。これは、Web で実現されているサービス、例えば、ユーザ登録、ユーザ認証、商品検索、アンケート、掲示板などで使用されるデータ形式を考えると当然と思われる。ここで問題となるのは、多くの RDBMS では文字列型のカラムを定義する際に、同時に最大文字長を指定する必要があるということである。テーブル設計をしていて、「住所」や「商品名」、「フィードバックコメント」には最大何文字とればよいのかは非常に悩ましい問題である。商用データベースの Oracle を例に取ると、可変長の文字列型を格納するために適したデータ型は以下の３つが用意されている。

- VARCHAR2
- LONG
- CLOB

最初の VARCHAR2 は、可変長ではあるものの、最大 4000 バイトという制限がある。これは全角に直すと 2000 文字であり、住所や商品名でこの長さを超えることはまずないと思われるが、商品の詳細説明文や、フリーのコメント、掲示板の書き込みなどを保存するには不十分なケースもある。このような場合は、複数のレコードに連番を振って、アプリケーション側で対処することが求められる。次の LONG は最大 2G-1 バイトまでの文字型データを格納できるが、１テーブルに最大１カラムしか使用できず、また LONG のカラムには LIKE 検索が使用できないなど各種の制限があり、VARCHAR2 のように気軽に SQL が書けないという問題がある[2]。CLOB は Oracle8.0 から導入された比較的あたらしいデータ型で、最大 4G-1 バイトまで格納可能でかつ１テーブルに複数カラム持たせることが可能だが、CLOB データへのアクセスには LOB ロケータというポインタを経由する必要があり、アプ

[2] これは、そのような商用 RDBMS を使用してアプリケーションを開発するエンジニアが苦労する問題であって、そのような使いにくいデータ型しか用意していない RDBMS が問題であると言っているわけではない。

リケーション開発の難易度があがる。つまり、Oracle を使用した場合、4000 バイトを超えるかもしれない文字列データを保存するためには、アプリケーション開発者が苦労しながらなんとか解決するしか方法がない。
一方、PostgreSQL には、独自の text 型というデータ型があり、バージョン 7.0 までは 1 レコードあたり約 8000 バイトの制限があったものの、7.1 以降では text 型 1 フィールドあたり最大 1GB の可変長文字列型データを格納することができる[3]。このとき導入された TOAST (The Oversized Attribute Storage Technique) と呼ばれる技術により、8000 バイトに収まりきらないデータは必要なら圧縮しながら別テーブルに格納する方式になっているため、1 レコードあたりのデータサイズについては実質上制限がなくなった。Web アプリケーションで扱われるデータに文字列型が多いことを考えると、柔軟かつ容易にアクセスできる機能を提供している点で、開発者にとっては PostgreSQL を採用する大きな理由となる[4]。

また、Web アプリケーションのもうひとつの特徴として、ブラウザからのリクエストが 1 リクエスト単位で切れてしまうことがあげられる。専用アプリケーションであれば、常にサーバに対して 1 つコネクションを確立し、クライアント/サーバ間で現在どのような状態なのかという情報を共有しながら動作することが可能なのだが、Web アプリケーションでは、もうそれ以上クライアントからのリクエストはないのか、単に次の要求まで間隔が空いているのかの区別をすることは難しい。クライアント/サーバ間のコネクションを保持しておける前者の場合、データベースに対する検索を一度だけ発行しておいて、n レコード前進/後退といった指示をすることにより対象レコードを切り替える RDBMS の機能を使用できるのだが、Web アプリケーションでは、リクエスト毎にクエリを発行し、必要なレコードだけを抽出するという効率の悪いオペレーションが必須となる。PostgreSQL では、対象レコードのうち、先頭から n レコードを飛ばして、m レコードだけを抽出する、といったオペレーションを記述するために、OFFSET、LIMIT という独自拡張された SQL 文がある。例として、商品テーブルに対して商品名で検索をして 21～30 件目の商品名を取り出すには、以下の SQL となる。

```
SELECT id, name, price FROM item_list
        WHERE name LIKE '特選%'
        ORDER BY id OFFSET 20 LIMIT 10;
```

この機能を使用すれば、アプリケーション側に特別の工夫もなく、「1 画面に n 件表示」や、

---

[3] そもそも「可変長」データなのだから最大文字長の指定が不要な方が当然とは思うのだが、標準の SQL はそのようになっていない。
[4] あるサイトのテーブル設計書をお客様に見せたときに、「データ長はドキュメントに記載しないのですか？」と驚かれたことがある。

「m件目から表示」といった機能が容易に作成できる。再び Oracle を例にとって上記のと同等の検索文を記述すると、以下のようになる。

```
SELECT * FROM
        (SELECT id, name, price FROM item_list
                WHERE name LIKE '特選%'
                ORDER BY id)
        WHERE ROWNUM <= 30
MINUS
SELECT * FROM
        (SELECT id, name, price FROM item_list
                WHERE name LIKE '特選%'
                ORDER BY id)
        WHERE ROWNUM < 21;
```

この SQL は

1) 検索対象をサブクエリとして記述し 1 から 30 番目までのレコードを取り出す。
2) 同様に 1 から 20 番目のレコードを取り出す。
3) 上記 1) の検索結果から 2) の検索結果の差分をとることで 21 番目から 30 番目のレコードを取り出す。

という意味であるが、記述内容が非常に複雑になり、直感的になにをしようとしているのかわかりにくい。[5]

では次に、Web アプリケーションではデータベースのどのような機能が使われる事が多いのか、弊社の関与した案件[6]から得た統計的情報をご紹介したい。以下にあげた機能は、商用データベースでも PostgreSQL でもサポートされている RDBMS の特徴的な機能である。

対象サンプル数: 9 システム
使用されている機能:

- ✧ 検索/更新 9
- ✧ ビュー 4

---

[5] Oracle の ROWNUM は、ORDER BY でレコードが並べ替えられる前のレコード番号を指すので、サブクエリを使わずに上記の SQL を記述すると期待したレコードと違う結果が返される。

[6] 弊社でシステム構築したもの、他社製システムのトラブル等を解析したものを含む。データベースは、PostgreSQL 及び各種商用 RDBMS が使用されている。

- ユーザ定義関数　　3
- トリガ　　3
- 外部キー　　1

上記を見てわかる通り、大多数の Web サイトでは、RDBMS としての(ある程度以上)高度な機能は使用せず、単なる検索付き、保存機能付き巨大配列としてしか扱っていない事がわかる。言い換えると、それで十分サービスが提供できるレベルのアプリケーションが構築できてしまうということであり、少なくとも機能要件の側面から見れば、高機能、高性能、高価格の商用データベースを採用する必要性はあまり感じられない。むしろ、Web サイトのバックエンドとしてのデータベースには、機能そのものよりも、安定性、高速性、メンテナンスのしやすさといった要素が求められる。

PostgreSQL は、Web アプリケーションを構築するのに十分な機能を備えており、Web でよく使用する機能では商用 RDBMS よりも開発者にとって使いやすいことを見てきたが、次に、運用、保守について考察したいと思う。

PostgreSQL では pg_dump というコマンドでテーブルを指定するか、pg_dumpall というコマンドで全テーブルを対象としてバックアップする事ができる。この際、対象はコマンドが発行された時点でのスナップショットとなるため、特にサービスを停止する必要がないのは Web アプリケーションで使用する利点である。しかし、前回からの差分だけバックアップしたい、または、アーカイブログから「xx 時 yy 分のこのトランザクションまで」リストアしたい、といった要求には現在のところ応えられない。できる限りこまめにバックアップをとるか、アプリケーション側で特別な工夫をする等でカバーしなければならないのだが、逆に、細かなリストアにも対応できるかわりに専門のトレーニングを受けたエンジニアが何時間もかけて復旧作業にあたらなければいけない[7]、というのも Web サイトの運営者側にとってはあまり現実的ではない。企業の業務システムならば専任のデータベースエンジニアを雇って作業にあたらせることもあるが、Web サイトの運営側にとっては、そのような人員を社内に確保しておくのが難しいからである。

また、PostgreSQL には、特有の VACUUM というコマンドがあり、運用上の問題となる場合があった。PostgreSQL では更新されたデータは追記され、更新前データには無効のマークをつけるアーキテクチャを採用しているため、頻繁に更新が発生するアプリケーションを使用するとパフォーマンスが劣化し、データベースファイルが増大する。不要なデータをクリーンアップするために VACUUM コマンドを定期的に実行しなければならないのだが、バージョン 7.1 までは VACUUM 実行中はクライアントプログラムが待たされるという制限があったため、Web サイトの定期メンテナンス時にサービスを停止して行う等、運

[7] しかもその結果リストアに失敗しました、というのもよくある話なので、理論的に復旧する機能があるということと、実際に復旧ができるということは別問題である。

用上の注意が必要であった。なお、2002/02/06 にリリースされたバージョン 7.2 では、7.1 からの変更点として「VACUUM の動作が変更され、実行中でもロックがかかることがなくなったため、サービスが長時間待たされることがなくなった」とあるので、現時点で弊社では実際に試してはいないが、可用性が向上していることが伺える。

結論

ここまでの論旨を以下のようにまとめてみた。

- ある程度複雑なサービスを提供する Web サイトにはデータベースサーバが必須である。
- PostgreSQL には、Web アプリケーション開発のしやすさで商用データベースに勝っている点もある。
- 実際の Web アプリケーションで使用されている RDBMS の機能は、さほど高度ではなく、PostgreSQL がサポートしている機能で十分である。
- 運用、保守については現状では PostgreSQL には機能が足りない面もあるが、逆に管理は容易である。

開発者にとって PostgreSQL は Web サイトのバックエンドとしては非常に扱いやすく、サイト運営者側にとっても商用データベースと比較してライセンス料、サポート料を抑えられるメリットがある。運用、保守についてどの程度まで妥協できるかが、商用 RDBMS にするか PostgreSQL にするかどうかの判断基準になると思われる。

冒頭で Linux 普及との類似性について述べたが、最後に PostgreSQL の今後の普及についての考えを述べて締めくくりたいと思う。

Linux では、大手ハードウェアベンダが自社製品でのサポートを打ち出したのに対し、PostgreSQL に関してはあまりそのような状況になっていない[8]。これは、ハードウェア+オペレーティングシステム+データベースでは所詮「ソリューション」とはならず、結局はその上のシステム構築は構築会社が行うため、構築会社側から見れば個別に自分たちで各コンポーネントを調達してインテグレートする事に比べてさほどメリットがないことも一因であると感じている。逆に、ハードウェアの選定や Linux のインストール、PostgreSQL のソースからのビルド、インストール、設定ができないか面倒なようであれば、そのような構築会社にシステム開発、運用を依頼するのは少々ためらわれるのではないだろうか。オープンソースのメリットを活かし、社内でソースコードを触れるくらいのスキルを持った開発会社が Web サイト等のコンポーネントとして積極的に採用するなどで、PostgreSQL の益々の普及に期待したい。

---

[8] SRA と IBM が AIX と PostgreSQL の組み合わせで販売協力している。