徹底比較 PostgreSQL vs MySQL

2011/2/25
PostgreSQL Conference2011 にて
日本 PostgreSQL ユーザ会
株式会社あすか
宗近 龍一郎

アジェンダ

- 自己紹介
- PostgreSQL と MySQL の市場シェア
- PostgreSQL/MySQL の最新情報
- レプリケーション
- Memcache
- ベンチマーク
- チューニング
- まとめ

自己紹介

- 日本 PostgreSQL ユーザ会　関西支部前支部長
- 株式会社あすか　取締役副社長
- PostgreSQL 以外の活動
  - 日本 PHP ユーザ会発起人
  - 日本 Asterisk ユーザ会
  - いくつもの OSS のユーザ会活動に関わってきました (関わっています)
- ハンドル名は「まいパパ」

PostgreSQL と MySQL の市場シェア

- 実は PostgreSQL の日本国内でのシェアは、 2009 年に MySQL に抜かれてました

	2007 年	2008 年	2009 年
PostgreSQL	46.9%	53.0%	51.9%
MySQL	44.8%	51.4%	60.5$

- 出展
  - 第 3 回オープンソースソフトウェア活用ビジネス実態調査 (2009 年度調査)
    - http://www.ipa.go.jp/software/open/ossc/seika_1004.html
  - @ＩＴ： Web を使って商用 DB の情報をどんどん集めよう！
    - http://www.atmarkit.co.jp/fdb/rensai/dbwatch2010/dbwatch20100 9_02.html

なんで PostgreSQL が MySQL にシェアを逆転されたのか

- 私なりの解釈ですが :)
  - ウェブサービスを展開する多くの企業が MySQL を採用した
  - レプリケーションの実装が遅かった (バージョン 9 からの対応)
  - MySQL よりも動作が遅いと思われていた
  - LAMP に比べて、 LAPP のゴロが悪い ?! ^^;
- 日本 PostgreSQL ユーザ会の方々は大変頑張っておられるので、けっして力不足というわけではないです m(__)m

PostgreSQL と MySQL を利用する OSS の比較

- 海外由来のものは MySQL が先で PostgreSQL への対応はどちらかというと遅れていた
  - Mediawiki(Wiki)　MySQL/PostgreSQL
  - Drupal(CMS)　　MySQL/PostgreSQL
  - 島根県版 CMS(CMS)　PostgreSQL
  - Redmine(プロジェクト管理)　MySQL/PostgreSQL/SQLite
  - 9arrows(プロジェクト管理)　PostgreSQL
  - Moodle(e ラーニング)　　MySQL/PostgreSQL/Oracle
  - Scalix(Web メール)　　PostgreSQL
- しかし最近のものになると海外の OSS も積極的に PostgreSQL に対応している。

PostgreSQL は MySQL に比べて劣っているのか

- 以前は MySQL より遅いと言われていたが、 8.1 くらいから処理速度が向上している
- JIS2004(第 3 、第 4 水準 JIS) は 8.3 から対応している
  - MySQL は 6.0 から正式対応
  - CJK の対応の良さは石井さんのおかげ
- ストレージエンジンが 1 つなので、 MySQL のように迷うことが無い
  - 私は最近ストレージエンジンでトラブったことがあります ^^;
- BSD ライセンスなので商用の製品等に組み込んでの再配布が可能 (MySQL は基本的に商利用はライセンスを購入)

MySQL に比べて改善してほしい点

- pgpool 以外は分散ソリューションが弱いかも
  - MySQL は MySQL Cluster や MySQL Proxy があり、導入も比較的しやすい
  - PostgreSQL には PL/Proxy とか PgBouncer というソリューションもあるが、あまり事例を聞かない。
- レプリケーションも多段で行えるようになればなあ。。
  - 蓄積されているデータを分析に使いたい

PostgreSQL 最新情報

- 最新バージョンは 2010 年 9 月にリリースされた PostgreSQL9.0
  - レプリケーション (ストリーミングレプリケーション) 機能の追加
  - 64Bit Windows のサポート等
  - VACUUM 処理の高速化
- 以下のサイトを参考に
  - http://lets.postgresql.jp/documents/technical/9.0/1

MySQL 最新情報

- 最新バージョン (安定版) は、 2010 年 12 月にリリースされた、 5.5
  - 準同期レプリケーションのサポート
  - パフォーマンスの向上 (従来から 2 ~ 3 倍)
  - ただ、コンパイルに cmake を必要とするので以前のバージョンより導入が難しいかも ^^;
- 新機能の解説については nippondanji さん (MySQL の中の人) のサイトがよくまとまっています
  - http://nippondanji.blogspot.com/2010/04/mysqlmysql-554.html

MySQL に関する情報

- ORACLE 社は今後も MySQL を維持、発展させていくことを約束
  - 「 MySQL に関する 10 力条」を 2010 年に発表
  - Oracle Enterprise Manager が 2011 年に MySQL に対応
- ただ、 2010 年 11 月からサポート料金が大幅に上がっている
  - http://journal.mycom.co.jp/news/2010/11/08/015/index.html
  - コミュニティエディションはそのまま無償で使えます。

PostgreSQL と MySQL のレプリケーションの比較

- PostgreSQL でのレプリケーションはまだ実装されたばかりの機能なので、若干制約は多いかな。
- データの整合性確保は PostgreSQL の方が上。
  - スレーブ上での更新は不可
- 以下のブログ「 PostgreSQL 雑記」がよくまとまっています。
  - http://postgresql.g.hatena.ne.jp/pgsql/20100704

PostgreSQL のレプリケーション①

- PostgreSQL のレプリケーションは、イメージ的には、マスターからスレーブへプッシュするような方式
- 多段の構成が組めない
- ただしデータの一貫性では、 MySQL より勝る
  - リカバリ処理中の参照クエリーを許す

PostgreSQL のレプリケーション②

- 導入方法は InterDB の鈴木さん (しくみ分科会の鈴木さん) の資料がわかりやすくまとまっています
  - http://www.interdb.jp/techinfo/pg_sr/sr01.html
- マスターを止めずにレプリケーションが可能
- データベース全体のレプリケーションとなる
  - データベースの一部だけ (特定のデータベースだけをレプリケーションする) とかをレプリケーションすることはできない
- 現バージョン (9.0) では少ないながらも遅延が発生
  - 遅延が発生しないバージョンは次期 (9.1) から

MySQL のレプリケーション①

- MySQL のレプリケーションは、イメージ的には、スレーブがマスターへ自分が適用していないクエリーを取りにいくような方式
- スレーブを多段で構成することが可能
- スレーブの状態取得は MySQL の方がきめ細かく参照できる
  - 遅延時間の確認等
  - 運用でカバーすることを想定したためか

MySQL のレプリケーション②

- 導入方法は色々なところで紹介されているため今回は特にお話しません :)

  10 年以上もたって「枯れた」機能なので

- マスターを止めずにレプリケーションを行うことも可能だが、一時的に停止して行う方がやりやすいと思います
- たすきがけでマルチマスターの構成も組めますが、これは注意して行ってください

MySQL の準同期レプリケーション (5.5 での新機能)

- 今までの非同期レプリケーションでは、マスターが自分のログファイルにクエリーを記録した時点でデータを commit するため、スレーブに伝わらないうちに、マスターが死んでしまうと、スレーブが更新していないクエリーが発生することがあった
- 準同期レプリケーションでは、この点を改善しスレーブ側がクエリーの受け取りを完了するまで、マスターが commit を行わない。従って不整合が起こる可能性が少なくなる
- この点を利用して、 MySQL を冗長化しフェールオーバーさせる仕組みを作ることができる
  - keepalived 等を利用して

MySQL Proxy

- MySQL Proxy とは MySQL サーバのロードバランスを行うソフトウェア
- 例えば、 MySQL の読み込みクエリーを分散させるような場合に利用する
- 書き込みを分散させるような場合は、後述の MySQL Cluster と組み合わせる
- ORCALE 社にあるバイナリファイルを持ってきてコピーし、後は設定ファイルを書くだけなので比較的導入は楽

MySQL Cluster

- 元々は NDB といい、非共有型のクラスタシステムである
  - 特殊なハードやソフトを必要としない
- １つのデータノードに書き込まれたクエリーは別のデータノードにも反映されるため、前述した、 MySQL Proxy と組み合わせれば書き込み処理を分散できる
- MySQL Server を同梱した RPM パッケージを入手すれば比較的導入は楽です
- MySQL Cluster はコミュニティエディションでも使えます

memcache の話①

- memcached は高性能な分散メモリキャッシュサーバ
  - データベースへの問い合わせ結果を一時的にキャッシュすることで，データベースへのアクセス回数を減らし、ウェブアプリケーションの高速化やスケーラビリティの向上に寄与します
- RDBMS の種類を問わないので、当然 PostgreSQL でも MySQL でも使える
- アクセスできる言語としては、 PHP 、 Java 等多数あり、専用の API が存在する

memcache の話②

- memcache の話についてはしくみ分科会の鈴木さんの資料が詳しいです
  - http://www.atmarkit.co.jp/fdb/rensai/memcached/memcached1.html
- FaceBook 、 Youtube 、 Wikipedia 、 Mixi 等大規模サイトでの実績が多数あります
  - http://www-jp.mysql.com/why-mysql/memcached/
- オンライン系のゲームサイトでもまず使われていると思って良いでしょう :)

pgbench と mysqlbench

- pgbench は PostgreSQL 用の、 mysqlbench は MySQL 用のベンチマークテストツール
- 石井さんの作られた、 pgbench をベースに MyNA(日本 MySQL ユーザ会) 有志が mysqlbench を移植した
- MySQL には MyBench という同じような名前のツールもあるので混同しないように :)

PostgreSQL のチューニング

- PostgreSQL のチューニングは共有バッファのサイズとか、ある程度定石が決まっている
- しくみ分科会の鈴木さんの「 PostgreSQL 完全機能リファレンス」が参考になります
  - http://www.amazon.co.jp/exec/obidos/ASIN/4798014958/bukurojin-22
- 前述した pgbench 等で、ベンチマークを行い適切な値を設定します

MySQL のチューニング

- MySQL の設定ファイル (my.cnf) には 300 以上ものパラメータがあるが実際に設定 (変更) する値はある程度定石が決まっています
- MySQL には、 my.cnf のサンプルがいくつも同梱されているのですが、長く更新されていないものも多いので、実際には書籍等を見ながら適切な値を設定しましょう
- 奥野 (nippondanji) さんの「エキスパートのための MySQL[運用 + 管理] トラブルシューティングガイド」が私のお勧め

PostgreSQL/MySQL の GUI ツールの紹介

- PostgreSQL
  - Pgadmin III
- MySQL
  - MySQL GUI Tools
    - MySQL Administrator/MySQL Query Browser/MySQL Workbench の詰め合わせ
- 両方で使える
  - CSE
  - Navicat(商用)

PostgreSQL/MySQL の使い分け (私見)

- オープンソーシャル等ウェブ系サービスのアプリケーション開発には MySQL が良いと考えられる
  - 情報量が豊富である (サンプルが豊富)
- 業務系等長く使い続けるシステムには PostgreSQL の方が良いと考えられる
  - データベースシステムとして堅牢であること
  - ストアドプロシジャの仕組みが枯れていること

まとめ (というか雑談)

- PostgreSQL 良いとか、 MySQL が良いとかという議論はあまり意味をなさず、ご自分で手馴れて使いやすいとか、利用用途によって適切なものを選択されたら良いと思います
  - JPUG も MyNA も共同して勉強会を開いてお互いの良いところを吸収しあっています
- 昔は、大企業では商用のデータベースしか使わないところも多かったのですが、随分事情は変わってきています

ご静聴ありがとうございました