

OpenSource DataBase
三つ巴討論会 memo
MySQL vs PG vs Firebird
2005-03-26
#####

非公開情報をネタに話し合おう!

1 使いにくい点、ダメな点

1.1 MySQL

- version のどれを使う
 - 3.23 :メンテ終わり
 - 4.0 :メンテおわりつつ
 - 4.1 : main stream
 - 5.0 : 精力的に更新
- 4.1 が!!!
 - UTF8 にした
 - ユーザーの知らないうちに自動変換が勝手にかかる
 - cp932, eucjpmsは 5.0から

4.0 までは自動変換ない。

1.2 FireBird

- 2000年OpenSource化
- 1983 年当時の仕様をひきずる
- SQLパーサーのオーバーヘッドが大きい
- OpenSource 活動がいま一歩
- QA は良い
- 開発者は他のDBを嫌う傾向がある
- 日本語はSJISとUnicodeのマッピングが問題
- UCS4将来対応予定

1.3 postgresql

- 遅い
 - Oよりは? case by case. 平均でやや遅い
 - 出来ないとき、困難な時の逃げ道はあるか?(大量データの読み込みの時はcommitなしでやるとか。 専用のローダーが無いので欲しい)
 - O は逃げ道が多い
- 速くするにはなにかを犠牲にするのかも
- インデックスを張ってロードすると遅い(それはMySQLも同じ)

- InnoDB は postgres より遅いというのは本当?
- Super-Smack のある結果では、InnoDB の方が数倍早かった。
- シナリオ次第か? ->要検証
- UPDATE,SELECT中に、全スキャンを行うようなテストだと、MySQL は遅くなったが、postgresはあまり変わらなかったという結果も->要追試

- FireBird はメモリーの大きく取れるように改造しないと、こういったベンチのテストには不向きか?
- RAM disk つかっちゃうという回避策もある

2 他の注意点

2.1 MySQL の注意点

- JDBC はだめ。prepare()の実装がだめで、prepareを使うと通常の statement に比べてとてつもなく遅くなる。
- Java でのテストでは、MySQL は悪い結果を出すことになるだろう。
- > ごめん、間違い。使い方を誤ると遅くなるけど、正しく使えば prepare() は statement より速いか同じぐらい。フィールドの型と、Java側の型を同じにしないといけない

2.2 postgres

- クライアントが接続してくるとプロセスがフォーク。重い
- RPM(等、distributionについているパッケージ)の問題
- locale の問題、ソートしても結果むちゃくちゃ。
- ソースから入れるのが回避策
- ホスティングの場合は(向こうが用意しているものを使わざるを得ず)、あきらめる
- MySQL も似たもの。distributer は latin1 で用意しているから文字化け発生確立高い。
- また、ライセンス変更で古いバージョンを採用しているdistributionが多い
- FireBird はまだ distribution に入っていない(Borlandのライセンスの問題, win 上のユーザーが多いなどの理由から)

2.3 FireBird

- fork モデルとスレッドモデルがある。
- case by case で速さが違う
- 組み込み用途が多くて、OpenSource に取り組む人材の不足
- 開発はVC++が多い

3 日本語の扱い

3.1 postgres

- 複数のエンコーディングはできない
- DB 単位で変えられる

3.2 FireBird

- フィールド単位でエンコーディングを変更できる
- FireBird のデフォルトはキャラクタセットなし
- > すばらしい。
- 接続時の指定でキャラクタセットを変えることが可能
- cp932 <-> Unicode の問題はある

3.3 MySQL

- フィールド単位で変えられる
- 4.1 以上では、日本語の扱いに難点が多い

回避策：以下のどれか

- 4.1 以上はあきらめる
- unicode でいく
- サーバーとクライアントのキャラクタセットは同じにする
- サーバーとクライアントのキャラクタセットの食い違いに注意
- 4.1 からは、クライアントはクライアント自身が持つキャラクタセットでサーバーにつながに行く。これが文字破壊の原因の一つ。
- 自動変換がじゃま
- > あった方が便利という意見もある
- postgres は EUC のサーバーなら EUC のクライアントでつなげばいい
- postgres ではエンコードには疑問に感じていない。自動変換して欲しい(希望)
- postgres のコードマップはかなり前から日本人が作っている
- MySQL は最近自動変換が加わったから混乱している。4.0 までは無かった。いままでと変わったから混乱しているだけかも。

4 バージョン

4.1 MySQL

- MySQL の細かいリビジョンが多すぎて、知らない人間には判断できない
- バージョンによつての機能の違いが分からない
- 3.23 replication, InnoDB
- 4.0 マルチキャラクタセット, query cache
- 4.1 Sub Query, GIS, SSL, utf8
- 5.0 stored, view
- 5.1 trigger

-- 開発版をだすのが混乱の元か?

-- 4.1 以上は、資産がない人にはOK(これから新たに始めるなら4.1以上でOK)。4.0までの資産を活かしたい場合はあまりおすすめできない。

- MySQLは3.23は安定だけど、その中のこの機能は α とか、というのはちょっと混乱(いいのか?)。

- 追加した機能についてはバグはあるけど、stable リリース後にたまに機能追加をやることがあった

- マニュアルの記述がバージョンぐちゃぐちゃに書かれており、どのバージョンについてのことか分かりにくい。

- 日本語マニュアルは4.1.1で止まっている

- バージョンアップ早すぎない? 1,2か月でupgradeって...

-- いや、3.21時代は1日とかで変わったこともあったし、だいたい週間で変わっていることが昔(ver.3で)は多かった。それに比べれば今はあまり変わらないね。

4.2 PostgreSQL

- postgres のバージョンがよく上がる

- 8.0 (少数点1位、2桁目)までがメジャー番号。機能追加を表す。

-- この数字が違う場合は互換性は無くてもいいというポリシー

--- 割り切り

-- バージョンが変わると、データをdump してリストアする必要がある

--- MySQL は何の作業も要らない。前のバージョンのファイルはそのまま使える

-- 3桁目(ex. 8.0.1) はバグフィックスを示す

-- 古いクライアントは新しいサーバーにつながる

--- サーバーがクライアントを識別して動きを変える

- postgresの日本語マニュアルはすぐにでる。ちゃんと追従している。

-- postgres リリースと一緒に日本語マニュアルが出せる状態

-- メジャー upgrade の時は変更は多いけど。

4.3 FireBird

FireBird のマニュアルは今作業中。

FireBirdのQA作業

- QA用システムがある

-- お手製。

- スポンサー企業のエンジニアのフィードバックもある

- borlandユーザー会も手伝っている

- なので、開発は5人ぐらいだけど、テストする人間は多い

- リビジョンアップなしの仕組みを考えている最中