

MySQL あれこれ

注意事項

- いろんな所からパクってます
- 勘違いが含まれている可能性があります
 - 時間がなくて、きちんと調べてません!!

自己紹介

- 名前: となか
 - @ftnk, id:ftnk
- 仕事: サーバやネットワークなどインフラ
 - プログラマが気にしないことを気にしたり
 - パフォーマンスがでなければ、プログラム直せと
思いつつサーバ構成でなんとかしたり
 - いろんなスクリプト書いたり読んだり
- 静岡ITPro勉強会の代表
 - 12月18日(土)は合同忘年会です!!

MySQL の特徴

- プラグイン形式のストレージエンジン
- GPL と商用のデュアルライセンス
- レプリケーション標準装備
- 便利なSQL独自拡張

代表的なストレージエンジン

- MyISAM
- InnoDB
- Memory
- Archive
- Blackhole
- Spider, VP
- Groonga

MyISAM

- デフォルトのストレージエンジン
- トランザクションがない
- クラッシュ時に破損の可能性
- 更新性能がスケールしない
- 全文検索機能あり
 - 日本語非対応

InnoDB

- トランザクションに対応
- オールマイティ
- InnoDB Plugin でパフォーマンスが向上
 - 2倍ちかくパフォーマンスが上がることも
 - MySQL 5.5 からはInnoDB Plugin が標準
- 全文検索なし

memory

- メモリ上にデータを保存
- 既存データの読み込み専用で利用

Archive

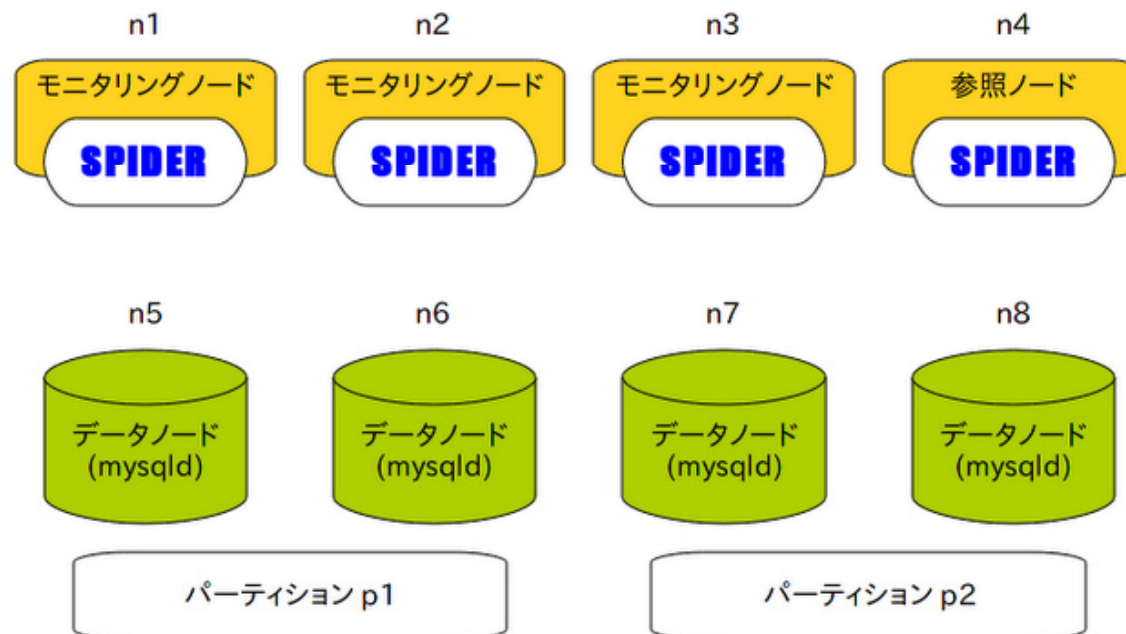
- Select と insert のみ可能
- Log などの追記と参照のみで十分なデータ用

Blackhole

- 書き込み専用のストレージエンジン
 - 書き込み最速
 - /dev/null
- Master-Slave なレプリケーションで、Masterをblackholeにしてbinlogの書き出しを優先

Spider, VP

- Spider = データベースの水平分割
 - HA機能あり
- VP = データベースの垂直分割



Groonga

- 日本語全文検索用ストレージエンジン
 - Senna の後継
 - 参照lockが不要
- Groonga サーバ
 - HTTPやmemcached binaryプロトコル、gqtp(独自プロトコル)でクライアントと通信可能

PBXT

- ログベースのストレージエンジン
 - ACID
 - トランザクション
 - MVCC
(マルチ・バージョン・コンカレンシー・コントロール)
 - 行レベルロック
- XAMPPに組み込まれている

XtraDB

- Percona が開発しているストレージエンジン
 - Percona は、いろんなパッチをあてた独自の MySQL のバイナリを配布している
- マルチCPU環境でよりよく動作するように設計

Aria (旧 Maria)

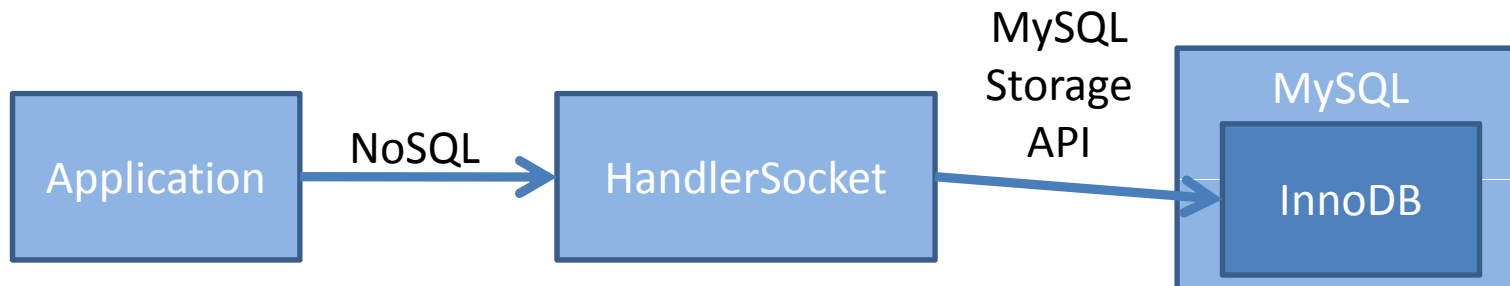
- 次世代のトランザクション対応ストレージエンジンを目指して開発中
- 現状はクラッシュセーフなMyISAM

RethinkDB

- SSD に最適化されたストレージエンジン
- ロックフリー
- バックアップ・スキーマの変更がオンラインで可能
- 全ての操作がデータベースへの追記
 - データベース = ログ
 - 自動ロールバック

HandlerSocket

- DeNAが開発した daemon plugin
- NoSQL のプロトコルを受け入れる
- MySQL のストレージAPI を使って、ストレージに直接
 - MySQLのSQL解析部分などをバイパスすることで、パフォーマンスを大幅に向上
 - 750,000 query/sec



レプリケーション

- 非同期レプリケーション
- セミ同期レプリケーション
- 遅延レプリケーション

MySQL のレプリケーション

- Master
 - Application -> mysqld -> SQL のパース (SQL thread)
 - -> data storage
 - -> binlog
- Slave
 - mysqld
 - > binlog 取得して relay binlog に書き込み (IO thread)
 - > SQL のパース (SQL thread)
 - -> data storage
 - -> binlog

非同期レプリケーション

- MySQL 5.1 までのレプリケーション
 - mixiの日記とかで「書き込みが完了しました。反映に時間がかかることがあります。」とか表示されるアレ
- MySQL 5.5 ではsemi-syncレプリケーションが導入
 - Masterからアプリケーションへの応答をレプリケーションが完了するまで遅延させることで、MasterとSlaveの一貫性を保つ。
 - Masterからアプリケーションへの応答を遅延させるので、当然、パフォーマンスは落ちる

セミ同期レプリケーション

- MySQL 5.5 で導入
- Masterからアプリケーションへの応答をレプリケーションが完了するまで遅らせることで、MasterとSlaveの一貫性を保つ。
- Masterからアプリケーションへの応答を遅らせるので、当然、パフォーマンスは落ちる

遅延レプリケーション

- Master からのレプリケーションを設定した時間だけ遅延させるレプリケーション
 - 障害や誤操作によってMasterのデータが破損した場合、レプリケーションでSlaveのデータも破損する可能性がある
 - 遅延しておけば、Slave のデータが破損する前にレプリケーションを止めることが可能

宣伝

- 静岡Developers勉強会
 - 11月27日 三島
 - 12月18日静岡 ペガサート
- 合同忘年会
 - 12月18日 18:00 から 坐 和民
- 静岡ITPro勉強会
 - 2月26日 ペガサート
 - 運用・監視のフレームワーク