

MongoDBについて

時間があればここを読んでください。

☞ <http://blog.madoro.org/mn/52>

NoSQLについて

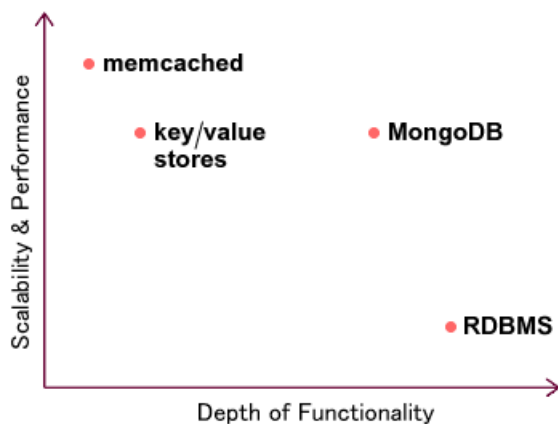
シンプルな機能と高いスケーラビリティを提供する目的で開発された。

☞ <http://journal.mycom.co.jp/articles/2010/04/26/nosql/index.html>

MongoDBの位置づけ

リレーショナルデータベースの機能性とKVSの速度を合わせ持つことを狙いとして開発

☞ <http://gihyo.jp/dev/serial/01/ruby/0031>



採用例

- 一日に60万以上のPVがあるサイト
- 将来的にスケーリングが必要なシステム

MongoDBをMySQLと比べたメリット

- 高速な処理
- スケーラブル
 - 分散データベースが簡単に出来る

導入

最新の v1.6.4 を導入しました。

比較的簡単に導入できます。

利用には、NoSQLのノウハウが必要です。MySQLで出来ることは大体出来るようですが、トランザクション、テーブルjoin、auto incrementは使えません。auto incrementの代わりにユニークなオブジェクトIDを発行します。

動作テスト

- 位置情報評価用にぐるなびの情報(約6万件)をMongoDBに格納しました。
 - 鳥取周辺の飲食情報 999件をGoogleMap上に配置しました。☞ <http://o2.lo/test/mongo/maps.php>
- Jmeterによるレスポンス評価
- 評価データ
 - ☞ <http://labs.infoalive.com/mongodb/212> のぐるなびデータを取得
 - 添付のmongo2mysql.phpを実行
- 評価プログラム
 - MongoDBによる一覧を表示 ☞ <http://o2.lo/test/mongo/maps2.php> .. 添付のmaps2.php
 - MySQLによる一覧を表示 ☞ <http://o2.lo/test/mongo/maps3.php> .. 添付のmaps3.php
- 結果

JMeterを使い、120秒間に1,000スレッド実行するテストを行った。

MySQLの1,000スレッドは後少しのところまでフリーズとなった。また、表示出来ないエラーも発生している。

TYPE	aggregate_report_count	average	aggregate_report_median	aggregate_report_90%_line	aggregate_report_min	aggregate
HTML	1000	82	39	181	25	1184
MongoDB	350	133	92	144	68	1044
MongoDB	1000	156	89	318	60	1909
MySQL	350	589	541	679	465	1704
MySQL	969	60770	42019	140512	10	158336

上記の簡単な評価としては、averageを確認すると良い。

MongoDBが 0.15秒程度で処理出来るのに対し、MySQLは350スレッドで0.6秒、1,000スレッドで60秒(フリーズ)の結果となった。

導入に向けてのライブラリ開発

- Deco MongoDBライブラリの開発
 - Pluginとして実装