OSDN Technical Tuning Program

Open Source Database ~ Security & Tuning

(1 hour)

Riotaro OKADA
riotaro@tunebiz.net

- はじめに

- UNIX は 米国およびその他の国における米国X/Open, Inc.の登録商標です。
- Linux はLinus Torvalds の米国およびその他の国における登録商標あるいは商標です。
- X Window System は、X Consortium, Inc.の商標です。
- Red Hat は、米国 Red Hat Software, Inc. の登録商標です。
- その他、記載されている会社名、製品名は各社の登録商標または商標です。
-

WHAT'S TUNEBiZ

- メンバーによるアクティブな活動
  - 日本Linux協会発起人 現ボード
  - 日本Apacheユーザ会core
  - 日本PHPユーザ会世話人
  - 日本PostgreSQLユーザ会
  - Project BLUE世話人
  - SI-Linux世話人
  - 日本UNIXユーザ会
  - IAJ セキュリティ部会
  - Linux JF (Japanese FAQ)
  - www.linux.or.jp Webmaster
  - Recruit Allabout.com Linux
- R&D / 構築/ 支援
  - Internet アーキテクチャ
    - Linux/Apache/Ruby,Perl,PHP/ PostgreSQL/MySQL
  - Webビジネス
    - 情報整理,戦略支援
  - Security
    - 構築,iDC運用,運用設計
    - クラックレスキュー
  - プロフェッショナルサービス
    - 1回10万円のコンサルティングメニュー
    - 1回30万円のセキュリティ監査メニュ

など -> riotaro@tunebiz.net

Securing Internet Architecture

- I What is Security,Tuning
- II Tuning SQL
- III Tuning Infra

Ⅰ What is Security,Tuning

セキュリティとは?

チューニングとは?

セキュリティとチューニング

セキュリティ[security]

- (1) 安全。防犯。安全保障。
- (2)（有価）証券。

チューニング[tuning]

- (1) ラジオ・テレビ放送などで、周波数を同調させ、特定の放送局を選択すること。
- （2）音程を正確に合わせる。音合わせをする。（音叉）
  - - 大辞林より

セキュアなWEBサイト

- 安定性
  - 高可用性
  - セキュリティ
- 性能
  - 最大毎秒接続数
  - 最大転送バイト数・スループット
  - RTT(roundtrip time)

→ 構築段階・運用段階・障害対応

II Tuning SQL

開発者のための
より良いSQL文の基礎

SQL最適化原則

- より速く
  - Indexをフル活用する
    - シーケンシャルサーチを極力減らす
    - 効果のあがるindex
- より明確に
  - 副問合わせ
  - 結合

Indexの効果

- 「実行計画」
  - indexを使えるSQLか
  - 「コスト」(使用最大タプルなど)が軽減されるか

  → シーケンシャル

  → index

  explain [SQL文] シミュレーションテストで確認
    - cf. PostgreSQL オフィシャルマニュアル p155

使われていないIndex

- 条件カラムに計算式がある
  - select id,name,price from tbl_product where price*1.25 < price
- 複合インデックス
  - 複合indexがid,nameである場合の二項目以降の条件
    - select id,name,price from tbl_product where name =‘okdt’
  - その順で条件に使われる場合はIndexが使われる
    - select id,name,price from tbl_product where id<100 and name =‘okdt’

使われていないIndex

- NULLかどうかを確認する条件のとき
  - select id,name,price from tbl_product where name is null
- !=演算子もindexを使わない
  - select id,name,price from tbl_product where name != null
- 文字列の先頭から比較しない検索条件
  - select id,name,price from tbl_product where name like '%okdt'
- Not in 演算子
  - select id,name,price from tbl_product where price not in(1, 1000)

使われていないIndex

- 副問合わせ（ネスト）より結合
  - × select tbl_sales.id, tbl_sales.amount from tbl_sales
    where tbl_sales.id =
    (select tbl_product.id from tbl_product
    where tbl_product.stock>10)
  - ○ select tbl_sales.id, tbl_sales.amount from tbl_sales,tbl_product
    where tbl_sales.id = tbl_product.id
    and tbl_product.stock>10)

SQLのためのデータクリーニング（PHP）

- ereg_replace(“[^0-9]”,””,$data)
  - クリーニングの基本
- addslashes($data)
  - addcslashes (string str, string charlist)
  - クオート（single,double）, バックスラッシュ、NULL文字などをスラッシュ付加
- quotemeta($data)
  - .+?[^](*)$ などをquoteする
- escapeshellcmd($data)
  - メタ文字をエスケープする

テーブル設計のポイント

- フィールドの使用頻度
  - 頻度の高いものはテーブルの前のほうに
- 更新フィールド、参照フィールド
  - 更新フィールドは隣接させたほうが
- 可変長フィールド、NULL許可フィールド
  - 固定長フィールドの後ろに置いたほうが
- 同意のフィールドのデータ形式統一
  - テーブル間：SQL文の要件から統一（型をあわせる）
- 日付・時刻フィールド
  - 問合わせフィールドにはコスト高い

III Tuning Infra

データベースシステムの
ための
インフラチューニング

インフラチューニングのポイント

- テーブル容量/Indexは大切に。
- メモリリソースの活用
  - カーネル共有メモリ
  - DBメモリオプション
    - 共有メモリ
    - ソートメモリ
    - キーメモリ
- スワップを最低限に

テーブル容量/Indexは大切に。

- テーブル容量の最適化
  - Vacuum
    - PostgreSQLのみの思想
    - 「毎晩行え」だそうな。
- indexの乱用
  - disk,メモリスペースの浪費につながりかねない
    - indexを別ディスクに置いてパラレルアクセス配置
  - 更新頻度が高いテーブルでは逆効果
    - index更新分のオーバーヘッド
  - 予想検索結果が総データ量の10%以下に抑える工夫を

カーネルによる共有メモリ

- SHMMAX（最大共有メモリセグメントサイズ）
  - Redhat 7.1 32M
  - Miracle 2.0 64M（128Mメモリの場合）
    - （どうもメモリサイズを見ているらしい）
  - Linux kernel
    - # echo 134217728 >/proc/sys/kernel/shmmax
    - /etc/sysctl.conf
      - kernel.shmmax= 134217728

DBメモリオプション

- 共有メモリバッファ
  - テーブルリソースのキャッシュに主に影響がある
  - 設定
    - PostgreSQL
      - 起動時の –Bオプション
      - postgresql.conf
        - shared_buffers=
    - MySQL
      - 起動時の –O table_cache が近い効果あり
  - サイズ決定のポイント
    - アクセス頻度の高いテーブルのサイズ
    - swapの起きないぎりぎりのサイズ

DBメモリオプション

- ソートメモリ(sort_mem)
  - ソート＆マージのためのテンポラリメモリ
    - ORDER BY,CREATE INDEX,MERGE JOINで使用
  - 設定
    - PostgreSQL
      - 起動時 –S オプション
    - MySQL
      - 起動時 –O sort_buffer=1M
  - サイズ決定のポイント
    - アクセス頻度の高いテーブルのサイズ
    - swapの起きないぎりぎりのサイズ

DBメモリオプション

- キーメモリ
  - キーアクセスのキャッシュに主に影響がある
    - メモリーがせつなくても確保しておくとイイ
  - 設定
    - MySQL
      - -O key_buffer_size

インフラチューニング

- Linux Kernel 利用可能リソースを増やす
  - メモリーを増やす
  - Echo 16384 > /proc/sys/fs/file-max
  - Echo 49152 > /proc/sys/fs/inode-max
- httpd.conf forkを減らす
  - MinSpareServers 50
    MaxSpareServers 150
    MaxRequestsPerChild 3000000
    MaxClient 256

  これだけでおよそ4倍の性能向上になるケースも

Update運用の重要性

- メジャーバージョンアップ
  - PostgreSQL7.2は魅力的。要コンバート
- 被害はほとんどがよく知られたバグのアタック
  - ramen,hidden line
  - RPC, linuxconf, … 通常サービスの不正利用
  - Buffer Overflow
    - bind 8.2.2-P4 以前
  - DoS
    - wu-ftpd 2.6 未満・ syslogd
- UPDATE運用
  - Debian系は apt-get
  - RedHatはup2dateを開発

リソース

- PostgreSQLオフィシャルマニュアル
  - 日本PostgreSQLユーザ会
- MySQLマニュアル
  - 日本MySQLユーザ会
- PHPマニュアル
  - 日本PHPユーザ会
- Apacheマニュアル
  - 日本Apacheユーザ会

エントロピー高い（ぼそっ）

Thank You

Please feel free to mail me

riotaro@tunebiz.net