Public Databases

Efficiences of Scale and ccTLDs

ccTLD Workshop 2004

October 11, 2004
Bangkok, Thailand

Hervey Allen

Outline

- Why consider using a database
- What is a “public database”
- A few database choices
- PostgreSQL and MySQL
- Types of data to store
- Building zone files from a database
- Summary
- Some more resources
- A real world example at APNIC
  - Champika

Why use a database?

Look at this from the viewpoint of database use vs. spreadsheets:

- Multi-user access.
- Easy to extend.
- Keep access to your data secure.
- Maintaining data integrity.
- Relational queries.
- Speed and available complexity of queries.

What's the problem?

As in, what types of problems are we trying to solve or avoid?

- Large zone file maintenance.
- Customer accounting.
- Customer service and tracking.
- Making sure that your data is correct.
- Keep your data secure:
  - Customer records.
  - Accounting records.

Multi-user access

A flat file (spreadsheet) can only be accessed by one person at a time.

As your organization grows you may have multiple people needing access to update records (aliases, mx records, A records, etc.).

Multi-user access means better customer service and better efficiency.

Easy to extend

I.E. - Multiple users accessing zone file information via a database:

- Now you can create a programatic interface to generate your zone file.
- Zone file can be generated at regular intervals without human intervention.
- Database can ensure that data entered is unique to create correct zone files.

Maintaining data integrity

You want to know that your data is not corrupt and you want to keep it that way.

- A well-designed database can help "force" your organization to enter in correct data.
- A database can verify data relations and integrity of your data.
- Databases have many tools for backup, recovery, cleanup, and data checking.

Relational Queries

This is something that you cannot do in a spreadsheet. Queries are limited.

- A relational database lets you create multiple tables with records.
- You can view your data in many different ways.
- Finding relations, querying for them, and getting results is an *extremely* powerful feature of relational databases.

Speed and complexity of queries

A well-designed database allows for extremely fine-grained queries on very large sets of data. These queries are:

- Fast!
- You can mathematically guarrantee the correctness of queries using boolean logic.
- You can guarrantee completeness of results.
- And, did I say the queries were “fast!”…

Public databases

By “public databases” we mean:

- Database software that is available under “free” licenses.
- Database systems developed in a public forum.
- Commerical databases must be purchased.
- Commercial databases require you to pay for newer versions.
- Both public and commercial databases have support contracts that you can pay for.
- Public databases have a legacy of user community support that is very effective.

Some Database choices

Public databases:

– **MySQL:** http://www.mysql.org/

– **PostgreSQL:** http://www.postgresql.com/

– **Mini SQL:** http://www.hughes.com.au/

Some “not” Public databases:

– **IBM's DB2:** http://www.ibm.com/db2/

– **Oracle:** http://www.oracle.com/

MySQL and Postgresql

Religious wars have been started over the question, “Which is better?”

PostgreSQL POWERED **versus** MySQL

One general opinion (imho) goes like this:

Postgresql has more advanced database features and is more complete while MySQL has a huge developed base of applications, is easier to use, and is very fast for small to medium-sized db's.

MySQL and Postgresql cont.

- Both databases are available for Linux and FreeBSD.
- Both are free.
- Both have tools for administering them graphically.
  - pgAdmin for postgresql
  - MySQL Administrator (beta)
  - Lots more for both, including web-based tools.
- Both can be accessed from your favorite programming language.
- Both are used to create dynamic web sites.

SQL and Some Tools

SQL = Structured Query Language

Command line and tools:

- mysql
  - phpMyAdmin (web)
  - mysql-administrator (beta)
- psql
  - pgphpAdmin (web)
  - pgAdmin

SQL and Some Tools

"LAMP"

- Linux, Apache, MySQL, Php

"FAMP"

- FreeBSD, Apache, MySQL, Php

"LAPP"

- Linux, Apache, PostgreSQL, Php

"FAPP"

- FreeBSD, Apache, PostgreSQL, Php

Types of data to store

Customer:

- Accounting records
- Transactions
- Support

Zone file:

- Domain records

Relations:

- Customer
- Domains

Building zone files from a database

- Your choice of language:
  - php
  - perl
  - C, C++
- Loop through all records (ensures completeness).
- Built dynamically and you can still be accessing your zone and customer data at the same time.

Summary

Scaling your registry operation can be difficult without the use of a database:

- Large choice of database software and tools.
- If well-designed your life as a registry becomes much easier.
- If well-designed your customer's experience will be all that much better.

More resources

- http://www.mysql.com/
- http://www.posgresql.com/
- O'Reilly books (http://www.oreilly.com/)
  - http://www.onlamp.com/
- http://www.php.net/
- The SANOG mailing list (sanog@sanog.org)

A real world example at APNIC

Champika Wijayatunga from APNIC will show an example of a database system using registry data in a practical manner.

Sample SQL queries on data will be shown as well.