

# What MySQL can learn from PostgreSQL (and some vice versa)



The PostgreSQL Company

Joshua D. Drake

# Before we get started...

- Who am I?
  - Major contributor to PostgreSQL.Org
    - PostgreSQL SPI Liaison (basically Treasurer)
    - Fund raising contact ([jdrake@postgresql.org](mailto:jdrake@postgresql.org))
  - Lead consultant Command Prompt, Inc.
    - All kinds of fun database stuff
  - President and Director U.S. PostgreSQL Association
    - [www.postgresql.us](http://www.postgresql.us)
  - Retiring (Director, Software in the Public Interest)
  - Known throughout community as JD or Linuxpoet

# What's the same?

- Let's not twiddle with the obvious
  - MySQL and PostgreSQL both:
    - Have large communities
    - Have large feature sets
    - Are SQL based
    - Have Good users
    - Have Bad users
    - Have Arrogant asses
    - Have Super geeks

# What's different?

Lots of stuff... Let's start with something simple



No, this was not the MySQL booth at OSCON 2007 but it may as well have been.

# Where was MySQL?

- MySQL AB was present with on average 1 employee in the booth
  - Old marketing material
  - No excitement
  - No reason to stop by
- The MySQL community wasn't present

# How was PostgreSQL at OSCON?

- Active
- Lively
- Full of discussion
  - Technical
  - Community
  - Advocacy
  - Education

# Sea of blue, army of smurfs!

PostgreSQL Booth, OSCON 2007



# Advocacy Efforts

- MySQL AB (now Sun) does not advocate. They promote, they sell.
  - The community is a second class citizen
    - MySQL AB has announced that Enterprise Customers will get features that the community will not.
    - MySQL does not actively engage the community for engineering efforts.
- Does the MySQL community advocate?

# PostgreSQL Advocates!

- In the first quarter of 2008 there are already 7 planned PostgreSQL community conferences
  - East – Maryland (done)
  - PG UK 2008 (done)
  - PDXPUG Day @ OSCON
  - LWEPG Day @ LinuxWorld
  - West – Portland
  - PGDay.IT – Italy
  - PGCon EU – TBA



# MySQL User Conferences?

None...  
(that I know of)

(We are at a MySQL AB conference, run by  
O'Reilly)

# So what?

There is nothing wrong with corporate conferences but for a community to be truly sustainable, the community must have its own ecosystem.

# What makes a community?

- Members/Users
- Review of other communities (versus MySQL)
  - Ubuntu:
    - The most popular Linux distribution in only 4 years
      - Driven by rabid, helpful and friendly community members
  - PostgreSQL:
    - Highly active in all areas
      - Driven by all walks of technical life. Engineers, hackers, consultants, end users, professional developers, advocacy and educational folks.

# What makes a community part two

Long before anyone else, (~2000?), came the Japanese

<http://www.postgresql.jp/>

In 2005, came the French

<http://www.postgresqlfr.org/>

In 2006, came SPI

<http://www.spi-inc.org/>

In 2007, came the Italians

<http://www.itpug.org/>

In 2008, the regions got it together

<http://www.postgresql.eu>

<http://www.postgresql.us>

# What makes a community part three

- Must a legal structure exist?
  - No
    - Must useful for larger and mature communities
      - Enables proper financial capabilities
        - Corporate sponsorship
        - Enabling community members
          - Sponsoring talks
          - Creating grants
      - Enables logistical support
        - Swag purchases
        - Address
        - A home base, or H.Q.

# Community Infrastructure

- PostgreSQL has defined community leads

# Community Infrastructure part two

- Advocacy: Josh Berkus
- User Group Liaison : Selena Deckelmann
- Fund Raising: Joshua D. Drake
- WWW Team: Dave Page
- Sysadmins: Marc Fournier
- Head Buddha (unofficial): Tom Lane
- Win32 Lead: Magnus Hagander

# Community infrastructure part three

- Why are community leads important?
  - Defined points of contact
  - Defined points of accountability
  - Provide stewardship through the meritocracy

# Co-opetition

- What is Co-Opetition?
  - Competition
    - The community thrives because companies compete
  - Cooperation
    - The community thrives because companies who are competing, also cooperate.



*Enterprise***DB**<sup>TM</sup>

# MySQL AB has no Co-opetition

Only Competition

# Co-opetition part two

- Only successful with companies understand Open Source
  - PostgreSQL is a meritocracy
    - Contributors gain influence through their merits
    - Companies can earn influence through the sponsorship (or employment) of contributors
      - Financial sponsorship does not gain influence
  - Only works when there is more than one company
    - Must not be in direct competition

# Competition and community

- Recognizing the value of the community
  - The community is the real stock holder in Open Source
  - To be truly successful as an open source project (with commercial participation) the commercial participation must be a servant to the community

# Leveraging Co-opetition

- Truly successful communities have multiple entities creating the software
  - Linux
    - Redhat, SUSE, Canonical, TurboLinux, Xandros
  - PostgreSQL
    - Command Prompt, EDB, Fujitsu, NTT, Sun, Truviso, Unisys
  - MySQL
    - MySQL AB
- Without diversification, project suffers

# Downside to Co-opetition

- MySQL has mostly (all?) Open Source product so the community benefits from all resources
- PostgreSQL has resources allocated in lots of directions. Many closed source and not a benefit to the community.
- Competition can sometimes forget the cooperation directive.

# The feature game

- MySQL adds features more quickly than PostgreSQL due to its willingness to add features to stable releases
- PostgreSQL adds features **only** in major releases causing 12-14 month breaks between feature sets

# The feature game part two

- Because of the MySQL model, new features appear quickly
- PostgreSQL does not practice release early, release often
  - (PostgreSQL still releases on average 3x faster than closed source databases)

# The feature game part three

- PostgreSQL adds features based on:
  - Correctness
  - Maintainability
  - Portability
  - Stability
- Downside is a slower development cycle with large sets of features appearing all at once
- Upside is, out the door PostgreSQL is always more stable, scalable and predictable.

# The feature game part four

- MySQL adds features based on:
  - Buzzwords
  - Perceived demand
  - Usefulness
- Downside is an unstable development model
  - New features appearing in stable releases (after stable release).
  - Features being enhanced (not fixed) in stable releases.
- Upside, MySQL has mindshare

# The Right Way

- Depends on goals
  - If the goal is customers
    - MySQL is the 'World's Most Popular Open Source Database'
    - Microsoft has the 'World Most Popular Operating System'
  - If the goal is community
    - PostgreSQL provides a technically superior (for most workloads), highly scalable, business and open source friendly database.
    - PostgreSQL has a vibrant and active community create a stable ecosystem

# If I ran Sun

- Open source everything, no second class citizens
- Sell support contracts (profit)
- Engineer Sun MySQL appliances (profit)
- Engineer Sun MySQL NDB clusters (profit)
- Adhere to standards (increased marketshare and respect)
- Immediately fix the development model (increased stability)
- Make Sun MySQL a servant to the MySQL community (respect)
- Support the creation of community lead conferences, user groups and workshops (increased community, marketshare, respect and profit)

# Questions?

- I can answer technical questions
- I can answer community questions