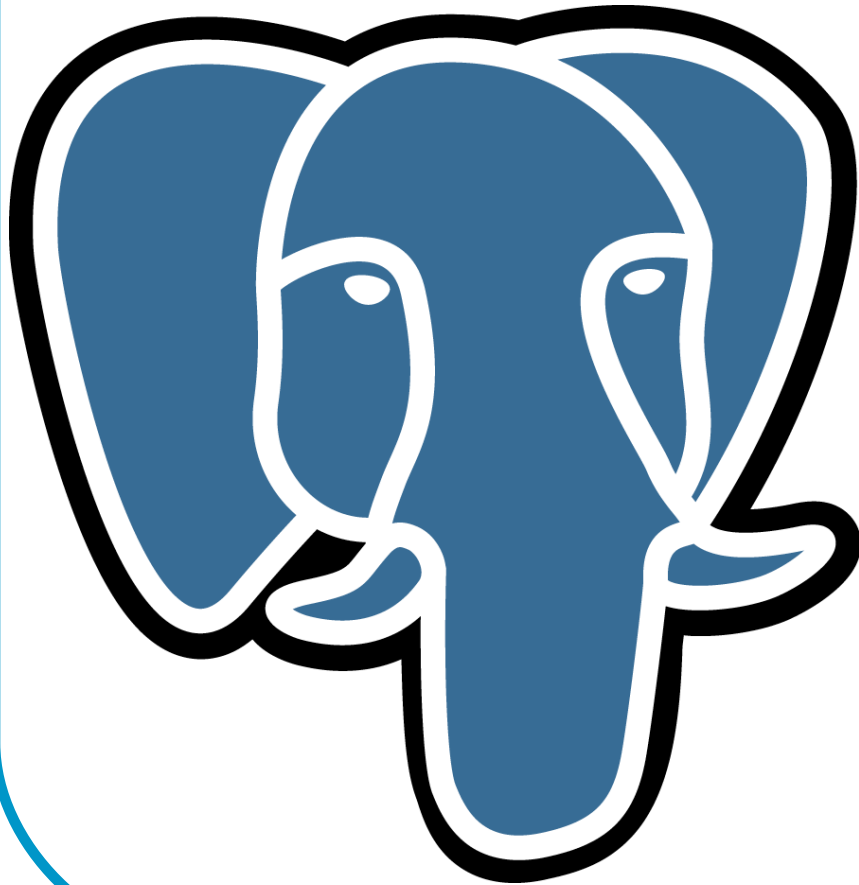


# Building search.postgresql.org



Magnus Hagander  
magnus@hagander.net

FOSDEM  
Brussels  
February 2008

# Why?

- Previous search.postgresql.org:
  - AspSeek, custom version
  - Required special C++ compiler version
  - Frontend was C++ CGI
  - High load on dedicated server
- Other out-of-the-box didn't work
- Good "dogfooding" of tsearch/GIN

# What?

- Indexes
  - Main website (~13k pages)
  - Community sites (~1k pages)
  - List archives (~620k pages)
  - ~6000 searches / day

# Built from

- PostgreSQL 8.3 (originally 8.2)
- PHP and the [www.postgresql.org](http://www.postgresql.org) framework
- Shared hosting server

# Results

- Full integration with framework
- Normal search times well below 1 second
  - But slightly slower than ASPSeek
- Indexing load almost zero
- More relevant hits
- Search: 270 lines of PHP code  
Indexer: <1000 lines of PHP code

# Context-aware indexing

- We know *what the pages look like*
  - Much more than just the URL and HTML
- For lists: sender, subject, time sent, etc
  - Can add indexing weights
- For website(s): remove framework

# Context-aware indexing

- We know *when* to index pages
  - No need to index data that hasn't changed
- List data *never* changes
  - Index current month only (for gaps)
- Static website mirror available
  - Check dates in filesystem
- Normal crawler for community sites

# Custom FTI configuration

- CREATE TEXT SEARCH CONFIGURATION pg  
(COPY = pg\_catalog.english );
- CREATE TEXT SEARCH DICTIONARY  
english\_ispell (  
    TEMPLATE = ispell,  
    DictFile = english,  
    AffFile = english,  
    StopWords = english);
- CREATE TEXT SEARCH DICTIONARY pg\_dict (  
    TEMPLATE = synonym,  
    SYNONYMS = pg\_dict);



# Custom FTI configuration

- ALTER TEXT SEARCH CONFIGURATION pg  
ALTER MAPPING FOR  
    asciiword, asciihword, hword\_asciipart,  
    word, hword, hword\_part  
WITH pg\_dict, english\_ispell, english\_stem;
- ALTER TEXT SEARCH CONFIGURATION pg  
DROP MAPPING FOR email, url, url\_path, sfloat, float;
- ALTER DATABASE search  
SET default\_text\_search\_config = 'public.pg';

# Basic indexing

- Fetch data (http or filesystem)
- Fix broken encodings (iconv)
- Apply regexp(s) to extract important data
- Insert in database

INSERT INTO messages

(list, year, month, msgnum, date, subject, author, txt, fti)

VALUES

(\$listid, \$year, \$month, \ \$1, to\_timestamp(\ \$2), \ \$3, \ \$4, \ \$5,  
setweight(to\_tsvector(\ \$6), 'A') || to\_tsvector(\ \$7))

# Searching

- Simple @@ matching wrapped in pl/pgsql
- How to deal with many hits
  - Processing/sorting too slow
  - LIMIT works, but hit count is lost, can't do paging!
  - Middle ground: LIMIT 1000, Count actual rows
  - **Always** use gin\_fuzzy\_search\_limit (20k)

# Thank you!

All code at  
<https://pgweb.postgresql.org>

## Questions?