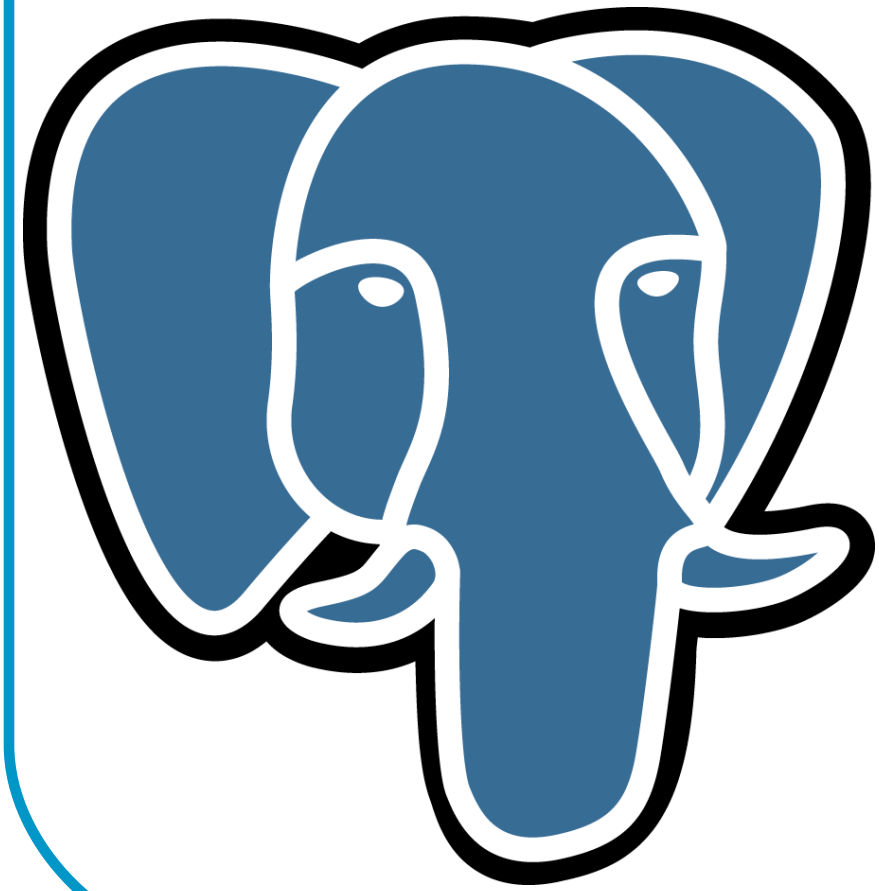
# Building search.postgresql.org



Magnus Hagander
magnus@hagander.net

PGCon 2008

Ottawa, Canada
May 2008

Search for: postgresql Search

List: pgsql-hackers

Post date: within last year

Sort by: Rank

Based on your search term, we recommend the following links:
» http://sql-info.de/postgresql/postgres-gotchas.html
» http://www.commandprompt.com/ppbok
» http://www.php.net/manual/en/ref.pgsql.php

## Results 1-20 of 251.

Searching in 146,944 pages took 0.75321 seconds. Site search powered by PostgreSQL 8.3.

Result pages: 1 2 3 4 5 6 7 8 9 10 11 12 13 Next

1 . Re: compiling postgres in winxp [2.8]
Posted 2007-11-30 17:13:01-08 by <mac_man2005@hotmail.it>.
**pg**_file_write" contrib\adminpack\adminpack.c: In function `**pg**_file_write':
contrib\adminpack\adminpack.c:129: error
http://archives.postgresql.org/pgsql-hackers/2007-12/msg00003.php

2 . Re: [COMMITTERS] pgsql: Most recent Postgres version is 8.2.6, per report from Robert [2.1]

# What?

- Indexes
  - Main website (~14k pages)
    - Every 4 hours
  - Community sites (~1.5k pages)
    - Incrementally every 4 hours
  - List archives (~680k pages)
    - Every 30 mins
- Searches
  - ~7000 searches / day

# 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

# Built from

- PostgreSQL 8.3 (originally 8.2+tsearch2)
- PHP and the 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
  - Can filter on metadata
- 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)

```
if ($this->http->RequestURL($url, $lastscan)) {
   ...
   if (!$this->indexer->WantIndexFile($suburl,
         $lastmod,$contenttype)) {
      continue;
   }
   ...
```

# Basic indexing

- Fix broken encodings (iconv)

```
if (is_null($encoding) &&
    preg_match('/<meta.*charset=([^"]+)/is',
                        $pagecontents, $matches) > 0) {
        $encoding = $matches[1];
}
if (is_null($encoding))
   $encoding = "iso-8859-1"; // Default
$str = iconv($encoding, 'utf-8//TRANSLIT',
    $pagecontents);
```

# Basic indexing

- Apply regexp(s) to extract important data

```
if (!preg_match("#
    <!--X-Subject: ([^\n]*) -->.*
    <!--X-From-R13: ([^\n]*) -->.*
    <!--X-Date: ([^\n]*) -->.*
    <!--X-Body-of-Message-->(.*)
    <!--X-Body-of-Message-End-->#s",
    $this->http->responsetext, $matches)) {

...
}
```

# Basic indexing

- Apply regexp(s) to extract important data

```
$subject = substr(html_entity_decode(strip_tags(
    iconv($this->http->encoding, 'utf-8//TRANSLIT',
    $matches[1])),ENT_COMPAT,'utf-8'),0,128);

...

$body = html_entity_decode(strip_tags(iconv(
    $this->http->encoding, 'utf-8//TRANSLIT',
    $matches[4])),ENT_COMPAT,'utf-8');
```

# Basic indexing

- 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)

# Searching

```
CREATE FUNCTION archives_search(...) t AS $$
...
tsq := plainto_tsquery(query);
IF numnode(tsq) = 0 THEN
  det = (NULL, 0, 0, NULL, NULL, NULL, NULL, NULL, NULL);
  RETURN NEXT det;
  RETURN;
END IF;
```

# Searching

```
...
OPEN curs FOR
SELECT list,year,month,msgnum,
    ts_rank_cd(fti,tsq)
FROM messages
WHERE fti @@ tsq AND
  date>COALESCE(firstdate,'1900-01-01')
ORDER BY date DESC LIMIT 1000;
```

# Searching

```
LOOP
  FETCH curs INTO hit;
  IF NOT FOUND THEN
    EXIT;
  END IF;
  hits := hits+1;
  ...
```

# Searching

```
SELECT INTO det lists.name, hit.year, hit.month,
  hit.msgnum, messages.date, messages.subject,
  messages.author, ts_headline(messages.txt,tsq,
    'StartSel="[[[[[[",StopSel="]]]]]]"'),
  hit.ts_rank_cd
FROM messages INNER JOIN lists ON messages.list=lists.id
WHERE messages.list=hit.list AND messages.year=hit.year
    AND messages.month=hit.month AND
    messages.msgnum=hit.msgnum;
END LOOP;
```

# Searching

```
  det=(NULL,NULL,NULL,NULL,NULL);
  det.year=hits;
  det.month=pagecount;
  RETURN NEXT det;
END;
```

# Searching

```
$this->rs = $this->page->pg_query_params(
   "SELECT * FROM archives_search(
      $1,$this->list,$datestr,NULL,$2,$3,$4)",
  array($this->page->query,
        $this->page->firsthit-1,
        $this->page->hitsperpage,
        $this->sort),
  'search');
```

# Seaching

```
$this->totalhits =
   pg_fetch_result($this->rs,
     pg_num_rows($this->rs)-1,
   1);
$this->search_base_count =
   pg_fetch_result($this->rs,
     pg_num_rows($this->rs)-1,
   2);
```

# Future direction

- New custom parser
  - Need to deal with pg_xyz searching
  - Code exists, just a small matter of time
- Maybe use 8.3 pl/pgsql scrollable cursors
- Metadata for ranking results
  - For example, move current docs up
- Fixing up docbot integration
  - Not really related to tsearch today

# Thank you!

All code at

https://pgweb.postgresql.org

Questions?