Text Search in 8.4?+

- WIP: Phrase Search
  - Algebra for text queries
- Fast approximate statistics based on GIN index
- Prefix search support (GIN partial match)
- Middleware for text search configurations

Oleg Bartunov, Teodor Sigaev, Mikhail Prokhorov
Moscow University, SAI, Russia

Phrase Search

- What is a phrase ?
  - It's tsquery
    - 'a b c'::tsquery
  - Ordering is important
    - 'a b c' != 'a c b'
  - Distance between words is important
    - 'a b x c' != 'a b c'

Phrase Search

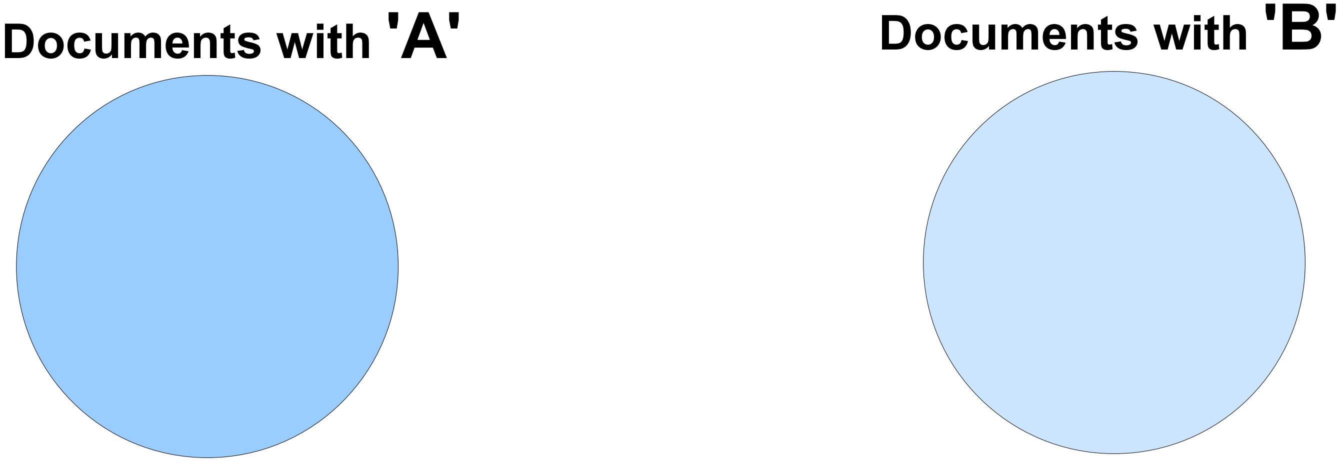
- What is a phrase ?
  - It's tsquery
    - 'a b c'::tsquery
  - Ordering is important
    - 'a b c' != 'a c b'
  - Distance between words is important
    - 'a b x c' != 'a b c'

Phrase Search

- Why there is no phrase search support ?
  - There are already support for boolean operations
  - There is positional information for each lexeme

Motivation for Algebra

- Existing operators defined at *document* level
  - 'A & B'::tsquery means intersections of two sets

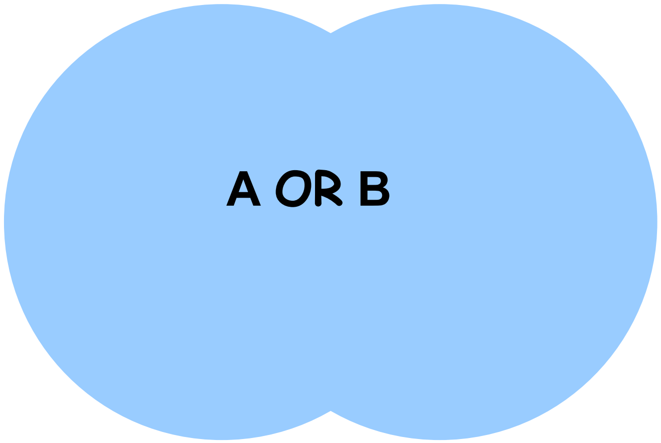


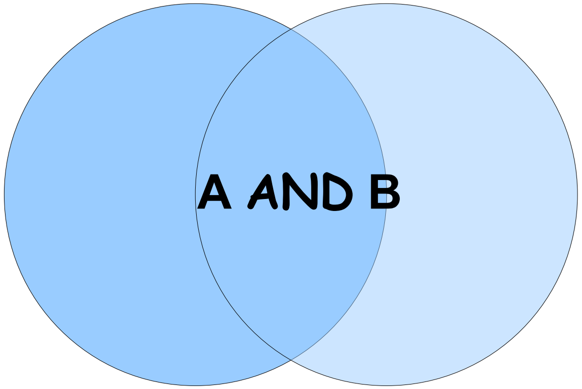
Documents with **'A'**

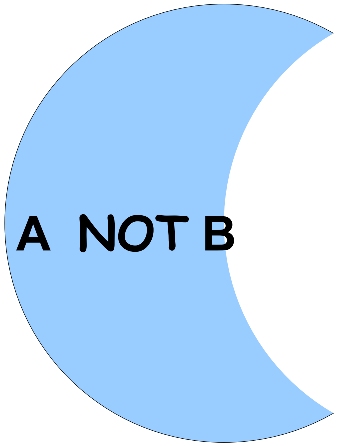
Documents with **'B'**

Motivation for Algebra

- Operators AND, OR, AND NOT work with sets

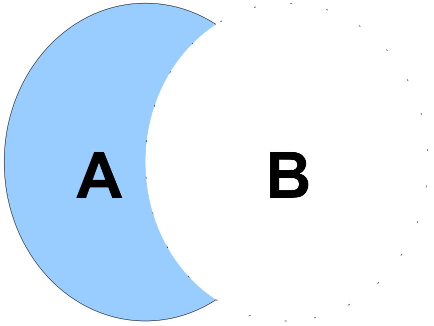






Motivation for Algebra

- Phrase Search requires operation at *lexeme* level — operator BEFORE ($)
- Different semantics - A NOT B
  - Normal search:
    Document with A and
    at the same time without B



  - Phrase search:
    «A $ X» (X is anything, except B)

Motivation for Algebra

- Phrase can be very complex
  - Even simplest phrase can be transformed to a complex expression.

    to_tsquery('nb', '**telefonsvarer**') =>
    'telefonsvarer' | 'telefon' & 'svar'

  - to_tsquery('footballklubber $ SMTH') => '((football & klubber) | (foot & ball & klubber)) $ SMTH' -
    hard, but it's not the hardest case

Motivation for Algebra

- Phrase can be constructed by a program, or manually using casting (SMTH::tsquery)

«A $ (B $!(C $!D))»

- We need well-defined algebra for operations: & | ! $
- Backward compatibility !

Motivation for Algebra

- We introduced «generalized» phrase

  a $[n] b

- Operator BEFORE ($[n]) guarantees
  - *An order* of operands — a BEFORE b
  - Distance between operands,
    default is 1

$$a\ \$[n]\ b == a\ \&\ b\ \&\ (\exists\ i,j : pos(b)_i - pos(a)_j = n)$$

$$a\ \$\quad b == a\ \&\ b\ \&\ (\exists\ i,j : pos(b)_i - pos(a)_j = 1)$$

Operations

- $a \ \$[n] \ b = b \ \$[-n] \ a$
- $!(a \ \$[n] \ b) = \, !a \mid \, !b \mid (\forall \ i,j : pos(b)_i - pos(a)_j \ != n \ )$
- $!!(a \ \$[n] \ b) = a \ \$[n] \ b$
- $a \ \$ \ !b = a \ \& \ (\exists \ i,j : \, !pos(b)_i - pos(a)_j = 1)$
- $!a \ \$ \ b = b \ \& \ (\exists \ i,j : pos(b)_i - \, !pos(a)_j = 1 \ )$
- $!a \ \$ \ !b = (\exists \ i,j : \, !pos(b)_i - \, !pos(a)_j = 1 \ )$
- $a \ \$ \ (b \mid c) = a \ \$ \ b \ \mid \ a \ \$ \ c$
  $(b \mid c) \ \$ \ a = b \ \$ \ a \ \mid \ c \ \$ \ a$
- $a \ \$ \ (b \ \& \ c) = b \ \& \ c \ \& \ (a \ \$ \ b \ \mid \ a \ \$ \ c);$
  $(b \ \& \ c) \ \$ \ a = b \ \& \ c \ \& \ (b \ \$ \ a \ \mid \ c \ \$ \ a)$

Recursive definition

$(a\ \$[n]\ b)\ \$[m]\ c = (a\ \$[n]\ b)\ \&\ c\ \&$

$(\exists i,j: \mathrm{posL}(c)_j - \mathrm{posR}(\text{“ab”})_i = m) \Rightarrow$

$(a\ \$[n]\ b)\ \&\ c\ \&\ (\exists i,j: \mathrm{pos}(c)_j - \mathrm{posR}(\text{“ab”})_i = m) \Rightarrow$

$(a\ \&\ b\ \&\ (\exists k,l: \mathrm{pos}(b)_k - \mathrm{pos}(a)_l = n))\ \&$

$c\ \&\ (\exists i,j: \mathrm{pos}(c)_j - \mathrm{posR}(\text{“ab”})_i = m) =$

$= a\ \&\ b\ \&\ c\ \&$

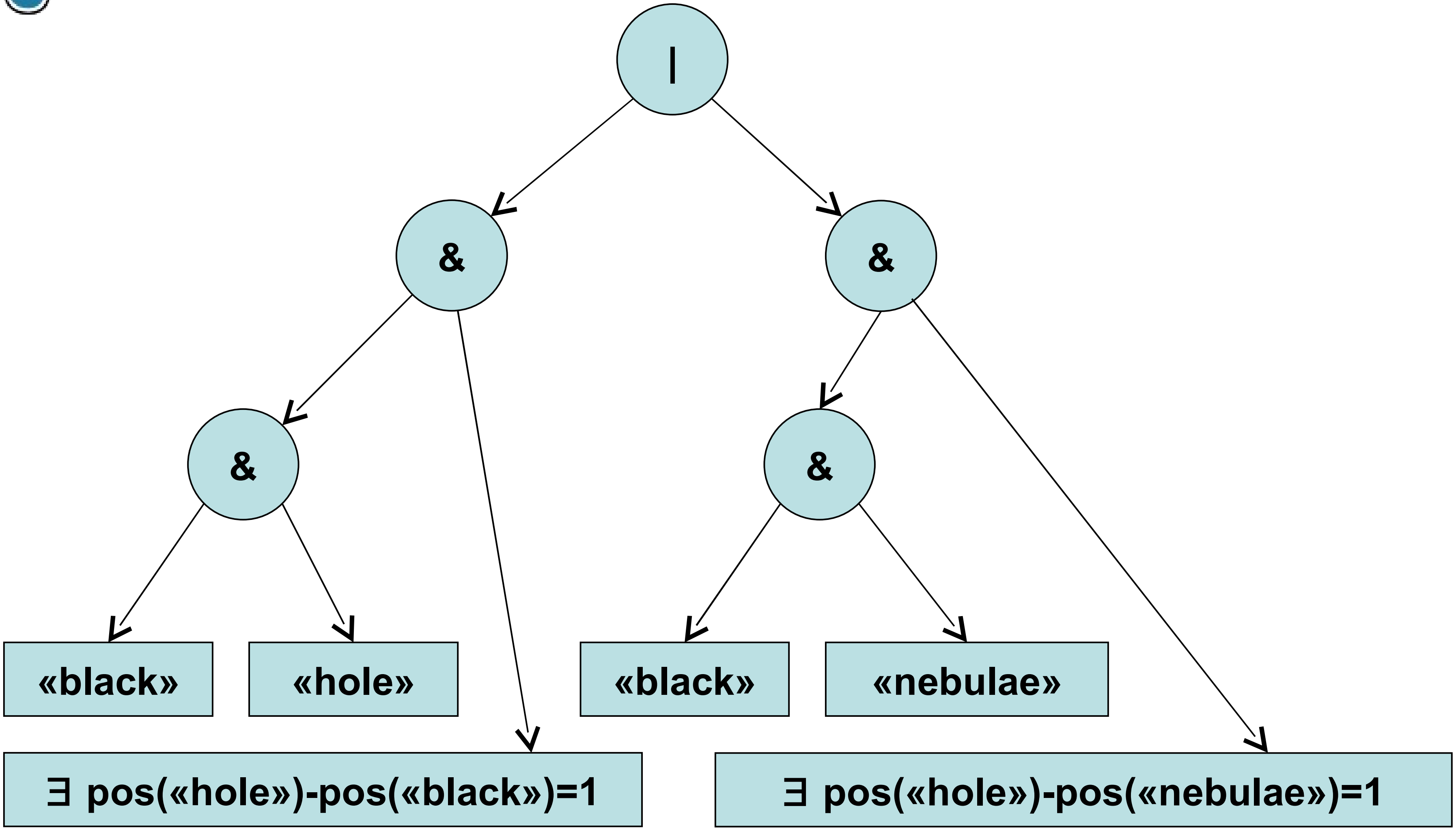
$(\exists k,l: \mathrm{pos}(b)_k - \mathrm{pos}(a)_l = n)\ \&\ (\exists j: \mathrm{pos}(c)_j - \mathrm{pos}(b)_k = m) =$

$= a\ \&\ b\ \&\ c\ \&\ (\exists j,k,l: \mathrm{pos}(b)_k - \mathrm{pos}(a)_l = n\ \&\ \mathrm{pos}(c)_j - \mathrm{pos}(b)_k = m) =$

$= a\ \$[n]\ b\ \$[m]\ c$

$a\ \$[n]\ (b\ \$[m]\ c) = a\ \$[n]\ b\ \$[m]\ c$ (as above)

Query: «black» $ («hole» | «nebulae») ==>
«black» $ «hole» | «black» $ «nebulae»

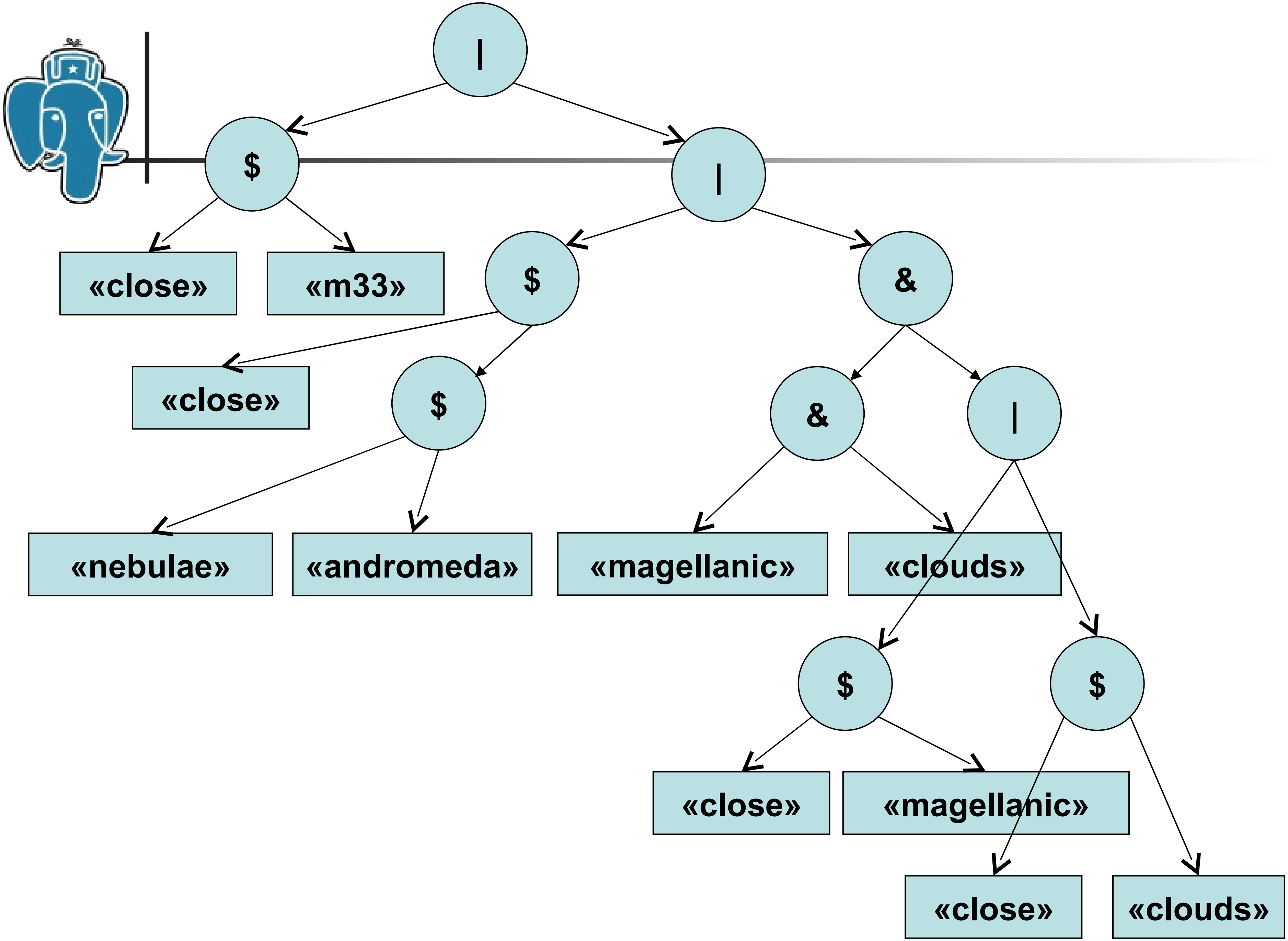


Example

Query: «close» $ «galaxies»

After dictionary: «close» $ («m33» |
(«andromeda» $ «nebulae» | («magellanic» & «clouds»))

```
Phrase:  «close» $ «m33» |
         ( «close» $ («andromeda» $ «nebulae» )) |
         ( «magellanic» & «clouds» &
            ( «close»  $ «magellanic» |  «close»  $ «clouds»)
         )
```


|
$
|
«close»
«m33»
$
&
«close»
$
&
|
«nebulae»
«andromeda»
«magellanic»
«clouds»
$
$
«close»
«magellanic»
«close»
«clouds»

Phrase Search

- Possible extensions
  - #[n] — soft $[n], order doesn't important
  - a<$[n] b — at most n words between operands
  - a $[n]> b — at least n words between operands
  - And so on ...

Partial Match for GIN

- Prefix search for a text search
- Improve performance **LIKE '%foo%'**
  - It's not a full text search
  - Btree index (text_pattern_ops) can improve
    - LIKE '%FOO'
    - LIKE 'FOO%'

Partial Match: Wildspeed

Index all permutations of string !

```
=# select permute('hello');
                   permute
  --------------------------------------
   {hello$,ello$h,llo$he,lo$hel,o$hell}
```

'$' is used for visualization, we use \0

LIKE '%l%' => ~ 'l*'
LIKE 'h%o' => ~ 'o$h*'
LIKE '%o'   => ~ 'o$*'
LIKE 'h%'   => ~ 'h*$

Partial Match: wildspeed

750,000 words, average length is 8 characters, time in ms

```
              |  h%   | hel% |  h%o | %l% | %lll% |  %l |  %lll | %ll%o |
--------------+------+------+------+-----+-------+-----+-------+-------+
wildspeed     | 28.0 |  1.1 |  1.1 | 434 |  0.7  | 426 |   0.7 |    18 |
Btree/seqscan |  8.5 |  1.0 |  8.6 | 415 |  408  | 407 | 404.0 |   404 |
```

```
CREATE INDEX ... USING btree (w text_pattern_ops) :  3.175 seconds
CREATE INDEX ... USING gin (w2 wildcard_ops)      :  1 hour 10 minutes
```

Prefix search

The popular request for the text search

```
SELECT 'superstar on party'::tsvector @@ 'super:*' AS yes;
   yes
  -----
    t

SELECT 'supernovae:1A sky:2B'::tsvector @@ 'super:A*' AS ye
   yes
  -----
    t
```

Prefix Search

- Based on partial match algorithm in GIN
- Syntax — use flag '*'
  - 'abc:*'::tsquery - search documents with words 'abc*'
- Prefix search comes for free, no special actions required !
- Dictionary API supports prefix flag

Prefix search

- tsquery @@ to_tsquery('supernova:a* & stars')
  - Find **supernova*** in **titles**

```
=# select count(*) from papers where fts @@
                                             to_squery('supernova:a* & stars');
 count
-------
   838
(1 row)
=# select count(*) from papers where fts @@
                                           to_tsquery('supernova:a & stars');
 count
-------
   835
(1 row)
```

Fast approximate statistics

- Gevel extension — GiST/GIN indexes explorer (http://www.sai.msu.su/~megera/wiki/Gevel)
- **Fast** — uses only GIN index (no table access)
- **Approximate** — no table access, which contains visibility information, approx. for long posting lists
- Statistics looks good for mostly **read-only** data

Fast approximate statistics

- Top-5 most frequent words (463,873 docs)

```
=# SELECT * FROM gin_stat('gin_idx') as t(word text, ndoc
int)  order by ndoc desc limit 5;

  word  |  ndoc
--------+--------
 page   | 340858
 figur  | 240366
 use    | 148022
 model  | 134442
 result | 129010
(5 rows)
Time: 520.714 ms
```

Fast approximate statistics

- gin_stat() vs ts_stat()

```
=# select * into stat from ts_stat('select fts from papers') order by
ndoc desc, nentry desc,word;

...wait....

=# SELECT a.word, b.ndoc as exact, a.estimation as estimation,
round ( (a.estimation-b.ndoc)*100.0/a.estimation,2)||'%' as error
FROM (SELECT * FROM gin_stat('gin_x_idx') as t(word text, estimation
int)  order by estimation desc limit 5 ) as a, stat b
WHERE a.word = b.word;

  word  | exact  | estimation | error
--------+--------+------------+-------
 page   | 340430 |     340858 | 0.13%
 figur  | 240104 |     240366 | 0.11%
 use    | 147132 |     148022 | 0.60%
 model  | 133444 |     134442 | 0.74%
 result | 128977 |     129010 | 0.03%
(5 rows)
Time: 550.562 ms
```

Middleware for FTS configuration

- Dictionaries should be able to specify how interpret their output. For example, dictionary returns (a,b) . Possible interpretations:
  - (a,b) -> a & b
  - (a,b) -> a | b
  - (a,b) -> a $[n] b - 'b' follows 'a', n words max
  - (a,b) -> a #[n] b - soft $ (no order)
  - Etc.

Middleware for FTS configuration

- Option for dictionary to return also an original word
- Manage how word is processed by a stack of dictionaries
  - Stop if recognized — current behaviour
  - Process and continue — filters
    - Use case: accent removal problem (wrong highlighting) — cannot be solved by using function to_tsvector(remove_accent(document))

Text Search in 8.4?+

- This work is supported by
  - jfg://networks — over-blog.net
  - EnterpriseDB

Oleg Bartunov, Teodor Sigaev, Mikhail Prokhorov
Moscow University, SAI, Russia