



# **Performance Probleme aufspüren**

**Oberhausen, 2013**

**Hans-Jürgen Schönig**

# Gründe für schlechte Performance



1. Dumme Anfragen
  - das passiert häufiger als man denkt
2. Suboptimale PostgreSQL Parameter
3. Schlechte Indizierung
4. Elende VACUUM Policy
5. Langsame Abfragen
6. HINWEIS: Die Liste ist bei Weitem nicht komplett

# 1. Dumme Abfragen (1):



- Das machen keine Anwesenden ;)

Das gilt alles nur für Personen, die nicht auf dieser Konferenz sind ;)

- einige wenige Abfragen können ein ganzes System killen.
- Performance Probleme werden viel zu oft von völlig unnötigen Codepfaden verursacht

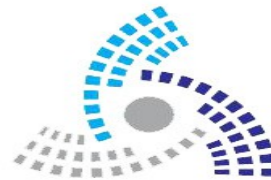
# 1. Dumme Abfragen (2):



- Was ist eigentlich eine dumme Abfrage?
  - Stellen wir uns eine Seite vor, die die Suche nach Produkten ermöglicht.
  - Passende Produkte werden gelistet
  - Offenbar war aber jemand etwas übermotiviert ...

# 1. Dumme Abfragen (3):

- Was ist an dieser Seite so dumm?



Searchinatrix

test



24.345.210 RESULTS

[Test Synonyms, Test Antonyms |...](#) Diese Seite übersetzen

[thesaurus.com/browse/test](https://thesaurus.com/browse/test)

Synonyms for **test** at Thesaurus.com with free online thesaurus, antonyms, and definitions. Dictionary and Word of the Day.

[test - Wiktionary](#) Diese Seite übersetzen

[en.wiktionary.org/wiki/test](https://en.wiktionary.org/wiki/test)

01.09.2013 - A challenge, trial. 2012 March-April, Colin Allen, "Do I See What You See?", American Scientist, volume 100, number 2, page 168 : Numerous experimental ...

[Test Your Vocabulary](#) Diese Seite übersetzen

[www.testyourvocab.com](http://www.testyourvocab.com)

Test your English vocabulary size, and measure how many words you know. For children, adults and EFL/ESL learners.

[TypingTest.com - Free Typing Test &...](#) Diese Seite übersetzen

[www.typingtest.com](http://www.typingtest.com)

TypingTest.com offers a free online Typing Test and exciting keyboarding games. Test your typing speed now!

[No WACA Test in 2014-15 cricket season |...](#) Diese Seite übersetzen

[www.news.com.au/national-news/western-australia/no-waca-test-in...](http://www.news.com.au/national-news/western-australia/no-waca-test-in...) - Vor 9 Stunden  
DEPTH has been dropped as a Test match venue after Cricket Australia announced its

# 1. Dumme Abfragen (4):



- Wo geht der Speed verloren?
  - Produkte zu finden ist keine große Sache ...
  - Was ist mit der grauen Zahl am rechten Eck?
  - Wird sich eigentlich irgend jemand Seite 15.287 ansehen wollen?
  - Macht es einen Unterschied, ob es 18.435.394 oder 24.345.210 Matches gibt?

# 1. Dumme Abfragen (5):



- Was bedeutet das in der Realität?

```
test=# SELECT count(*) FROM t_test WHERE id < 10;
count
-----
      9
(1 row)
Time: 0.669 ms
```

```
test=# SELECT count(*) FROM t_test;
count
-----
10000000
(1 row)

Time: 908.966 ms
```

**!!! Die erste Abfrage ist 1.300 mal schneller !!!**

# 1. Dumme Abfragen (6):



- Das ist nur ein Beispiel:
  - Klarerweise gibt es viele Möglichkeiten, Fehler zu machen
  - Performance Killer sind vielfältig
  - Wenn Sie die GUI unter Ihrer Kontrolle haben:
    - => Nutzen Sie es zu Ihrem Vorteil
    - => Wollen Sie wirklich 980.000 Abfragekombinationen erlauben?



## 2. PostgreSQL Parameter:



- Häufig vorkommende Fehler
  - schlechte work\_mem Settings
  - viel zu viele Checkpoints
  - seltsame shared\_buffers Einstellungen
  - unüberlegte Flushing Policy
  - zu intensives Logging

## 2. PostgreSQL Parameters:



- shared\_buffers anpassen

- die folgenden Werte können Sinn machen:

- => 25% vom RAM

- => nicht zu viel verwenden (maximal 16 GB)

- => kein Gesetz ist in Stein gehauen

## 2. PostgreSQL Parameter:



- work\_mem Settings:
  - work\_mem kontrolliert:
    - Sortieren, etc.
    - GROUP BY
    - IN Statements
  
    - GIN mit FASTUPDATE
    - walsender (für Tuplestores)
    - etc.

## 2. PostgreSQL Parameter:



### - work\_mem Settings: Ein Beispiel ...

```
test=# CREATE TABLE t_test AS SELECT * FROM generate_series(1, 1000000) AS x;  
SELECT 1000000
```

```
test=# explain analyze SELECT * FROM t_test ORDER BY x;  
QUERY PLAN
```

```
-----  
Sort (cost=141431.84..143931.84 rows=1000000 width=4) (actual time=1438.083..1676.490 rows=1000000 loops=1)  
  Sort Key: x  
  Sort Method: external sort Disk: 13688kB  
-> Seq Scan on t_test (cost=0.00..14425.00 rows=1000000 width=4) (actual time=0.056..618.101 rows=1000000 loops=1)  
Total runtime: 1789.480 ms  
(5 rows)
```

```
test=# SET work_mem TO '1 GB';  
SET
```

```
test=# explain analyze SELECT * FROM t_test ORDER BY x;  
QUERY PLAN
```

```
-----  
Sort (cost=114082.84..116582.84 rows=1000000 width=4) (actual time=397.446..487.880 rows=1000000 loops=1)  
  Sort Key: x  
  Sort Method: quicksort Memory: 71452kB  
-> Seq Scan on t_test (cost=0.00..14425.00 rows=1000000 width=4) (actual time=0.010..116.191 rows=1000000 loops=1)  
Total runtime: 567.224 ms  
(5 rows)
```

## 2. PostgreSQL Parameter:



- Flushing macht einen realen Unterschied:

- 10.000 Transactions:

INSERT INTO test VALUES (1);

- Default Setup:

4.6 sec (Mac OS X, SSD)

7.5 sec (Mac OS X, mechanische Disks)

1.48 min (Linux, mechanische Disks)

=> schummeln zahlt sich aus

## 2. PostgreSQL Parameter:



- Flushing macht einen realen Unterschied:

- 10.000 Transactions:

```
INSERT INTO test VALUES (1);
```

- “Entspanntes” Flushen:

```
SET synchronous_commit TO off;
```

1.1 sec (Mac OS X, SSD)

Ähnliche Ergebnisse auf allen Systemen mit  
ähnlicher Hardware

## 2. PostgreSQL Parameter:



### - Flushing:

- Jede Transaktion muss "ON COMMIT" auf die Disk

  - => Datenverlust wird damit ausgeschlossen

  - => Flushen ( fsync() ) kann teuer sein

- Durability Anforderungen sollten überdacht werden

  - => Flushes minimieren

  - => Durability ist nicht immer ein Thema

## 2. PostgreSQL Parameter:



### - "Checkpointing to death":

LOG: checkpoints are occurring too frequently  
(13 seconds apart)

HINT: Consider increasing the configuration  
parameter "checkpoint\_segments".

- Zu häufige Checkpoints führen zu Performance  
Problemen

=> checkpoint\_segments sollten erhöht werden

=> pump checkpoint timeout kann erhöht werden



## 2. PostgreSQL Parameter:



### - Intensives Logging

- Man sollte nicht jedes Mini-Statement loggen
- Kurze Statements zu loggen kann massiven Overhead mit sich bringen
- `log_min_duration_statement` hilft, die Flut an Logs ein wenig in den Griff zu bekommen

## 3. Schlechte Indizierung:



- Indexierung ist eines der WICHTIGSTEN Themen
  - VIELE Performance Probleme entstehen durch fehlende Indexes
  - 3 unerschöpfliche Quellen für Probleme:
    - => Fehlende Indices
    - => Falsche Indices
    - => Zu viele Indices

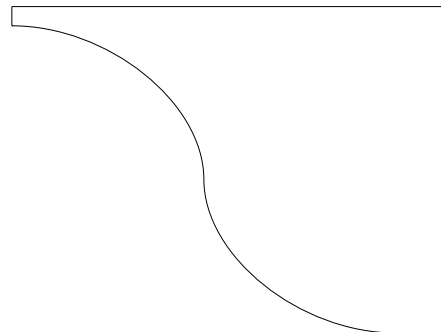
## 3. Schlechte Indizierung:

- Fehlende Indices finden:

- Die wichtigste Query eures Lebens ...

```
SELECT      relname, idx_scan, seq_scan, seq_tup_read
FROM        pg_stat_user_tables
ORDER BY    seq_tup_read DESC;
```

- Im Problemfall gibts einen “Hockey Stick” ...



**Todeszone**

**“Ok”**

### 3. Schlechte Indizierung:



- Ein absolutes Klassiker ...

- Wo wirds wohl klemmen?

```
test=# \d t_user
      Table "public.t_user"
  Column      | Type      | Modifiers
-----+-----+-----
 id           | integer   | not null
 email        | text      |
 password     | text      |
Indexes:
 "t_user_pkey" PRIMARY KEY, btree (id)
```

## 3. Schlechte Indexierung:



- Zu viele Indices:
  - Jeder Index Update ist I/O intensiv
  - Meistens ist das “Random I/O”
    - => Auf mechanischen Disks sehr sehr teuer
  - Performance Verlust kommt meistens schleichend
- **Unnötige Indices finden**
  - `pg_stat_user_indexes` kann helfen

## 4. Elende VACUUM Policy:



- Eine schlechte VACUUM Policy kann ein echtes Problem werden:
  - Es entsteht "Table Bloat"
  - Table Bloat hat unerwünschte Seiteneffekte
  - Platz wird unnötig verschwendet

## 4. Elende VACUUM Policy (2):



- Table Bloat finden:

- pgstattuple hilft dabei, eine Tabelle zu analysieren

```
test=# CREATE EXTENSION pgstattuple;
CREATE EXTENSION
test=# \x
Expanded display is on.
test=# SELECT * FROM pgstattuple('pg_class');
-[ RECORD 1 ]-----+-----
table_len          | 73728
tuple_count        | 315
tuple_len          | 57495
tuple_percent      | 77.98
dead_tuple_count   | 18
dead_tuple_len     | 2952
dead_tuple_percent | 4
free_space         | 9208
free_percent       | 12.49
```

## 4. Elende VACUUM Policy:



- Index bloat finden

- Dabei kann die selbe Extension verwendet werden

```
test=# SELECT * FROM pgstatindex('pg_class_oid_index');
```

```
-[ RECORD 1 ]-----+-----  
version          | 2  
tree_level       | 1  
index_size       | 24576  
root_block_no    | 3  
internal_pages   | 0  
leaf_pages       | 2  
empty_pages      | 0  
deleted_pages    | 0  
avg_leaf_density | 43.23  
leaf_fragmentation | 0
```



## 5. Langsame Abfragen:



- Schnelle vs langsame Abfragen:

- In vielen Fällen verursachen "Slow Queries" die meiste Load

- Viele kurze Abfragen können ebenfalls zum Problem werden

=> `pg_stat_statements` erzeugt entsprechende Statistiken

=> muss via `shared_preload_libraries` aktiviert werden

## 5. Langsame Abfragen:



### - Eine Übersicht ...

```
test=# CREATE EXTENSION pg_stat_statements;
CREATE EXTENSION
test=# \d pg_stat_statements
      View "public.pg_stat_statements"
  Column          |      Type          | Modifiers
-----+-----+-----
userid            | oid                 |
dbid              | oid                 |
query             | text                |
calls             | bigint              |
total_time        | double precision    |
rows              | bigint              |
shared_blks_hit   | bigint              |
shared_blks_read  | bigint              |
shared_blks_dirtied | bigint              |
shared_blks_written | bigint              |
local_blks_hit    | bigint              |
local_blks_read   | bigint              |
local_blks_dirtied | bigint              |
local_blks_written | bigint              |
temp_blks_read    | bigint              |
temp_blks_written | bigint              |
blk_read_time     | double precision    |
blk_write_time    | double precision    |
```

Any question?



Herzlichen Dank für die Aufmerksamkeit ...

Any question?