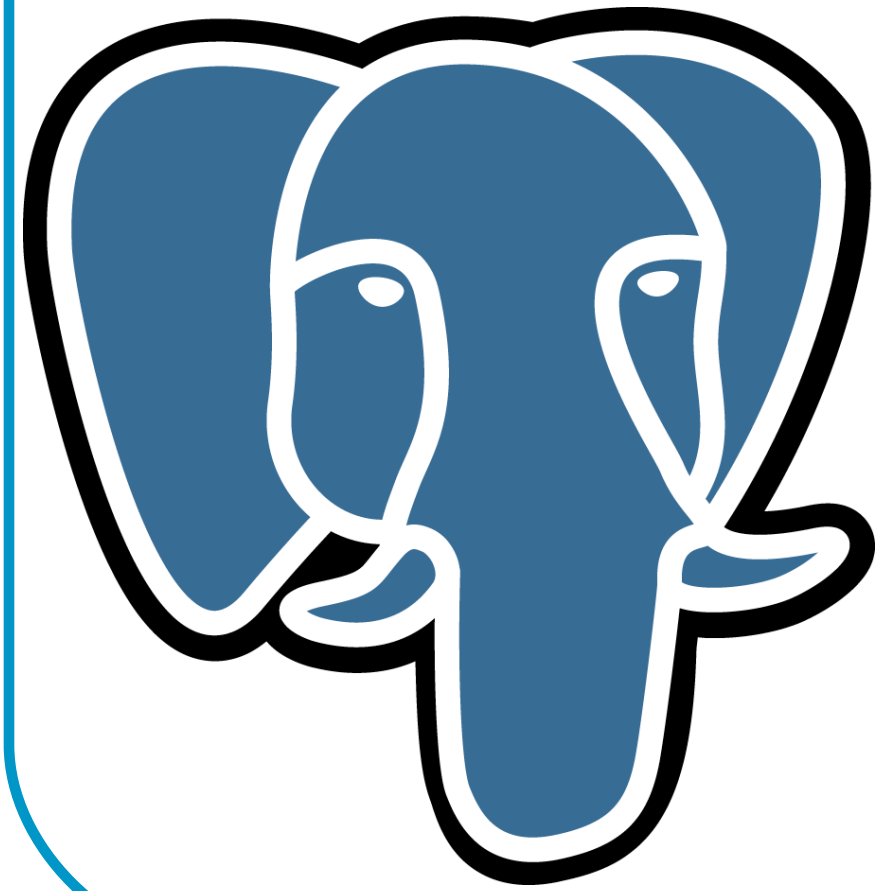
Benchmarking und Performancetesting von und mit PostgreSQL



Stefan Kaltenbrunner
stefan@kaltenbrunner.cc
http://www.kaltenbrunner.cc/blog

PGDay.EU 2010

Stuttgart, Germany

Die Datenbank ist langsam...

- *Was heisst ”langsam”?*
- *Ist es wirklich die Datenbank?*
- *Warum erst seit dem neuen Softwarerelease?*
- *Welche Hardware brauchen wir?*

Was heisst langsam?

- ”Der Server ist überlastet”
  - IO/Disks
  - CPU
  - Speicherauslastung
- ”Die Seite lädt langsam”
  - Langsam oder nur manchmal langsam?"
  - Was ist die Referenz für ”langsam”?

Können wir...?

- 1 Million Zeilen pro Sekunde laden
- 10000 Pageviews pro Sekunde abhandeln?
- Was passiert wenn wir wirklich mal abheben?

Die Basis...

- RAM
  - Memtest86+
  - Sysbench --test=memory
- CPU
  - openssl speed (manchmal)
  - sysbench –test=cpu
  - Stress (*)

Die Basis...

- IO-Subsystem
  - dd
  - bonnie++
  - sysbench –test=fileio
  - /src/tools/fsync/test_fsync.c

Die Basis...

```
mastermind@minibrain:~ $ test_fsync
Loops = 1000

Simple write:
        8k write                    184399.779/second

Compare file sync methods using one write:
        open_datasync 8k write        95.214/second
        open_sync 8k write            19.756/second
        8k write, fdatasync           80.475/second
        8k write, fsync               19.297/second

Compare file sync methods using two writes:
        2 open_datasync 8k writes     41.255/second
        2 open_sync 8k writes          9.738/second
        8k write, 8k write, fdatasync 80.500/second
        8k write, 8k write, fsync     19.636/second

Compare open_sync with different sizes:
        open_sync 16k write           19.860/second
        2 open_sync 8k writes          9.737/second
```

Die Basis - Datenbank

- shared_buffers
  - 20-25%, aber...
- work_mem/maintenance_work_mem
  - Was ist meine Last?
- WAL tuning
  - wal_buffers, checkpoint_segments, *fsync*
- Planner tuning
  - effective_cache_size, page_costs

Die Basis #2

- Connection Pooling
  - Verbindungsaufbau ist teuer
  - Volltextsuche
  - Weniger ist mehr (Cores*2!)
- Ausgewogene Hardware
  - RAM, Disk, CPU
- Monitoring
  - Was passiert eigentlich?

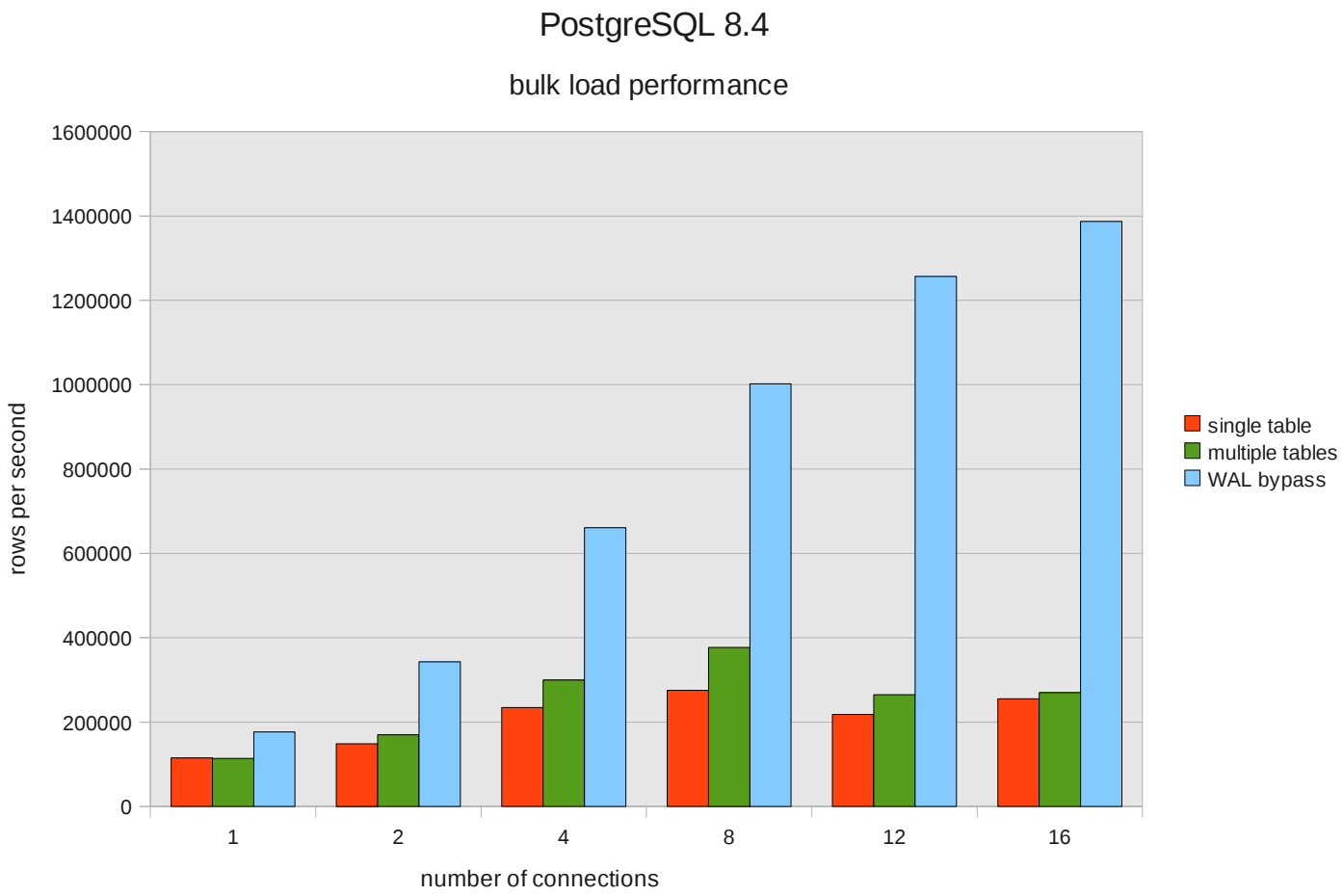
Workload #1 - COPY

- COPY
  - Batchprocessing
  - CSV Import
  - Backup/Restore
  - Desaster Recovery
- Was geht?
  - 1 TB/h ist machbar, aber...

Workload #1 – COPY

- Hardware
  - RAM +
  - IO ++
  - CPU +++
- Software
  - ”basic tuning”
  - paralleles laden
  - Keine Indizes (wenn möglich), TRUNCATE

Workload #1 – COPY



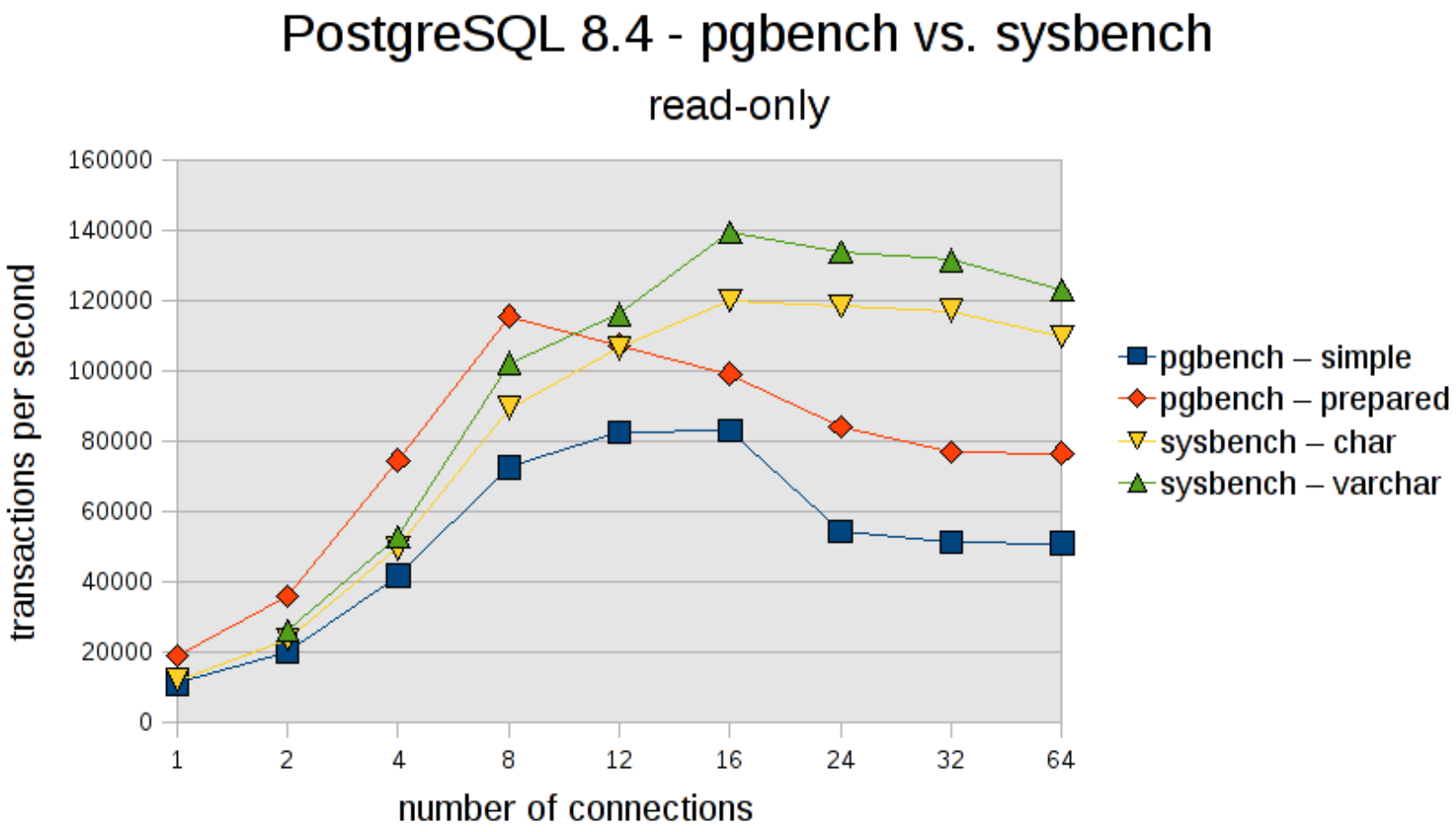
Workload #2 - SELECT

- Hardware
  - RAM++
  - IO
  - CPU++
- Software
  - ”basic tuning”
  - paralleles laden
  - Persistente Verbindungen

Workload #2 - SELECT

- Einfache Abfragen
  - 10000 qps pro core
  - Dual Quadcore Nehalem → ~120000qps
- Software
  - ”basic tuning”
  - paralleles laden
  - Persistente Verbindungen

Workload #2 - SELECT



SELECT vs. CONNECTIONS

```
mastermind@minibrain:~ pgbench -h 127.0.0.1 -C -j 4 -c 4 -S -T 60
starting vacuum...end.
transaction type: SELECT only
scaling factor: 10
query mode: simple
number of clients: 4
number of threads: 4
duration: 60 s
number of transactions actually processed: 26008
tps = 433.418138 (including connections establishing)
tps = 2346.151158 (excluding connections establishing)
```

SELECT vs. CONNECTIONS

```
mastermind@minibrain:~ pgbench -h 127.0.0.1 -j 4 -c 4 -S -T 60
starting vacuum...end.
transaction type: SELECT only
scaling factor: 10
query mode: simple
number of clients: 4
number of threads: 4
duration: 60 s
number of transactions actually processed: 624753
tps = 10411.929275 (including connections establishing)
tps = 10414.993531 (excluding connections establishing)
```

SELECT vs. CONNECTIONS

```
mastermind@minibrain:~$  pgbench -p 5433 -C -j 4 -c 4 -S -T 60
starting vacuum...end.
transaction type: SELECT only
scaling factor: 10
query mode: simple
number of clients: 4
number of threads: 4
duration: 60 s
number of transactions actually processed: 186371
tps = 3106.148596 (including connections establishing)
tps = 7291.458824 (excluding connections establishing)
```

Wie testen?

- OS
  - Filesystem
  - IO-Scheduler
  - RAID-Parameter(Caches!)
  - Buffercache (de)tuning (dirty_background_ratio)
- Datenbank
  - Basis Tuning
  - Kein –enable-casserts!

Wie testen?

- Tools
  - Die eigene Applikation/Workload
  - Pgbench (> 9.0)
  - Sysbench (read-only)
  - Tsung(*)
  - Mark Wongs DBT2/DBT3 tests

Wie testen?

- pgbench
  - -C …. 1 Verbindung/Abfrage
  - -S …. Nur Lesend
  - -f …. ”eigene Workload”
  - -T …. Laufzeit(Checkpoints!)
- Sysbench
  - --oltp-read-only=on

Wie testen?

- Monitoring
  - Munin
  - check_postgres.pl
  - Collectd
  - vmstat
  - sysstat

Die Lehren...

- *Der beste Test ist immer der der eigenen Anwendung!*
- Die Basis muss stimmen
  - Basistests der Hardware/OS-Kombination
  - Basistuning der Applikation/Datenbank
- Standardtools verwenden
- Nur weil Hardware teuer ist...

Danke!

Fragen?