Výkonnostní archeologie

Tomáš Vondra, GoodData

tomas.vondra@gooddata.com / tomas@pgaddict.com

@fuzzycz, http://blog.pgaddict.com


PostgreSQL
Up and Running
O'REILLY
Booking.com
OpenFest2013
O'REILLY

Jak se změnil výkon PostgreSQL za několik posledních verzí?

7.4 vyšla 2003, tj. cca 10 let

(překvapivě) ošidná otázka

- během vývoje se většinou dělají “parciální” testy
  - srovnání dvou verzí / commitů
  - zaměřené na konkrétní část kódu / vlastnost
- komplexnější benchmarky pro srovnání dvou verzí
  - obtížné “zkombinovat” (různý hardware, ...)
- aplikační výkon (ultimátní benchmark)
  - podléhá (pravidelným) upgradům hardwaru
  - růst datových objemů, evoluce aplikace (nové fičury)

(poněkud) neférová otázka

- vývoj probíhá oproti aktuálně dostupnému hardwaru
  - Kolik RAM jste měli v serveru před 10 lety?
  - Kdo z vás měl před 10 lety SSD/NVRAM disky?
  - Kdo z nás měl stroje s 8 CPU?
- některé rozdíly jsou důsledkem těchto změn (algoritmy)

Každopádně vyšší výkon na aktuálním hardwaru je fajn ;-)

Pojďme si zabenchmarkovat!

Pokud se bojíte čísel nebo grafů, asi byste radši měli odejít hned.

Bude tu spousta obojího ...

http://blog.pgaddict.com

http://planet.postgresql.org

82,71% statistik na internetu je vycucaných z prstu ...

... přísahám že ty moje to nejsou!

Benchmarky (přehled)

- pgbench (TPC-B)
  - “transakční” benchmark
  - operace pracují s malými počty řádek (přístup přes primární klíče)
- TPC-DS (náhrada TPC-H)
  - “warehouse” benchmark
  - dotazy drtící spousty dat (aggregace, joiny, ROLLUP/CUBE, ...)
- fulltext benchmark (tsearch2)
  - primárně o vylepšeních GIN/GiST indexů
  - platí pro další aplikace používající GIN/GiST (geo, ...)

Použitý hardware

HP DL380 G5 (2007-2009)

- 2x Xeon E5450 (each 4 cores @ 3GHz, 12MB cache)
- 16GB RAM (FB-DIMM DDR2 667 MHz), FSB 1333 MHz
- 6x10k RAID10 (SAS) @ P400 with 512MB write cache
- S3700 100GB (SSD)

Workstation i5 (2011-2013)

- 1x i5-2500k (4 cores @ 3.3 GHz, turbo 3.9 GHz, 6MB cache)
- 8GB RAM (DIMM DDR3 1333 MHz)
- S3700 100GB (SSD)

pgbench

TPC-B “transakční” benchmark

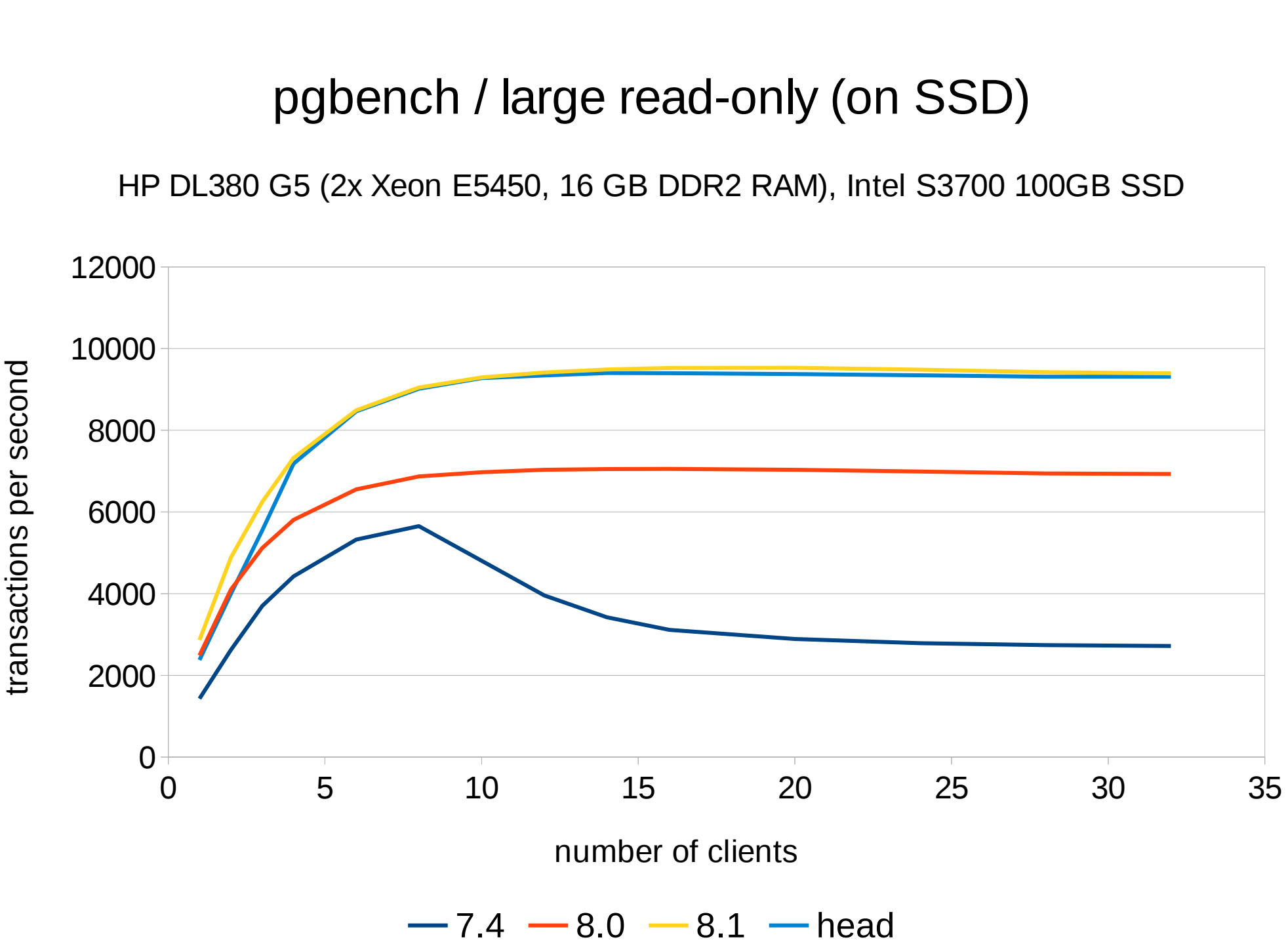
pgbench

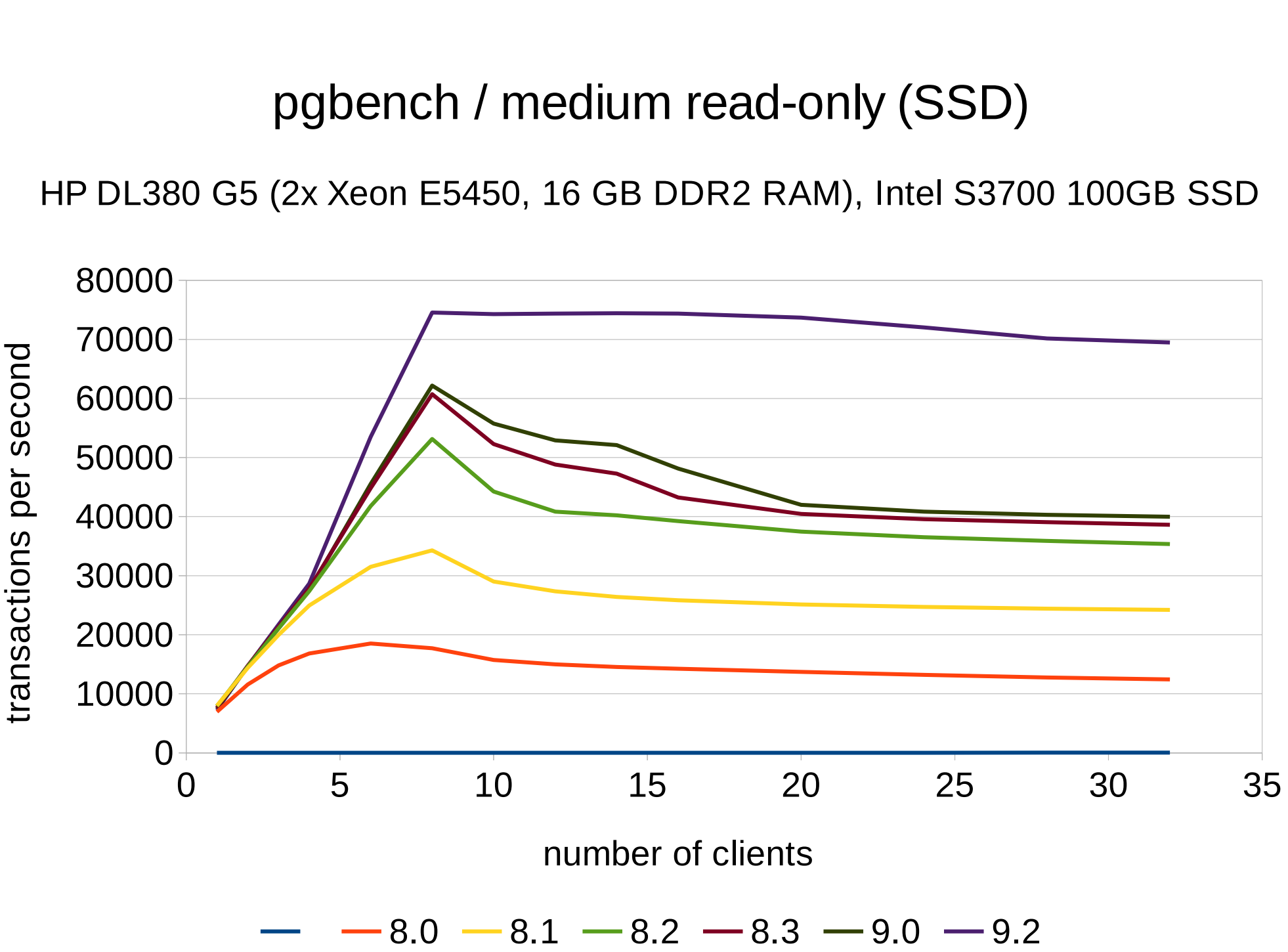
- tři velikosti datasetů
  - malý (150 MB)
  - střední (~50% RAM)
  - velký (~200% RAM)
- dva módy
  - read-only a read-write
- rozsah klientů (1, 2, 4, ..., 32)

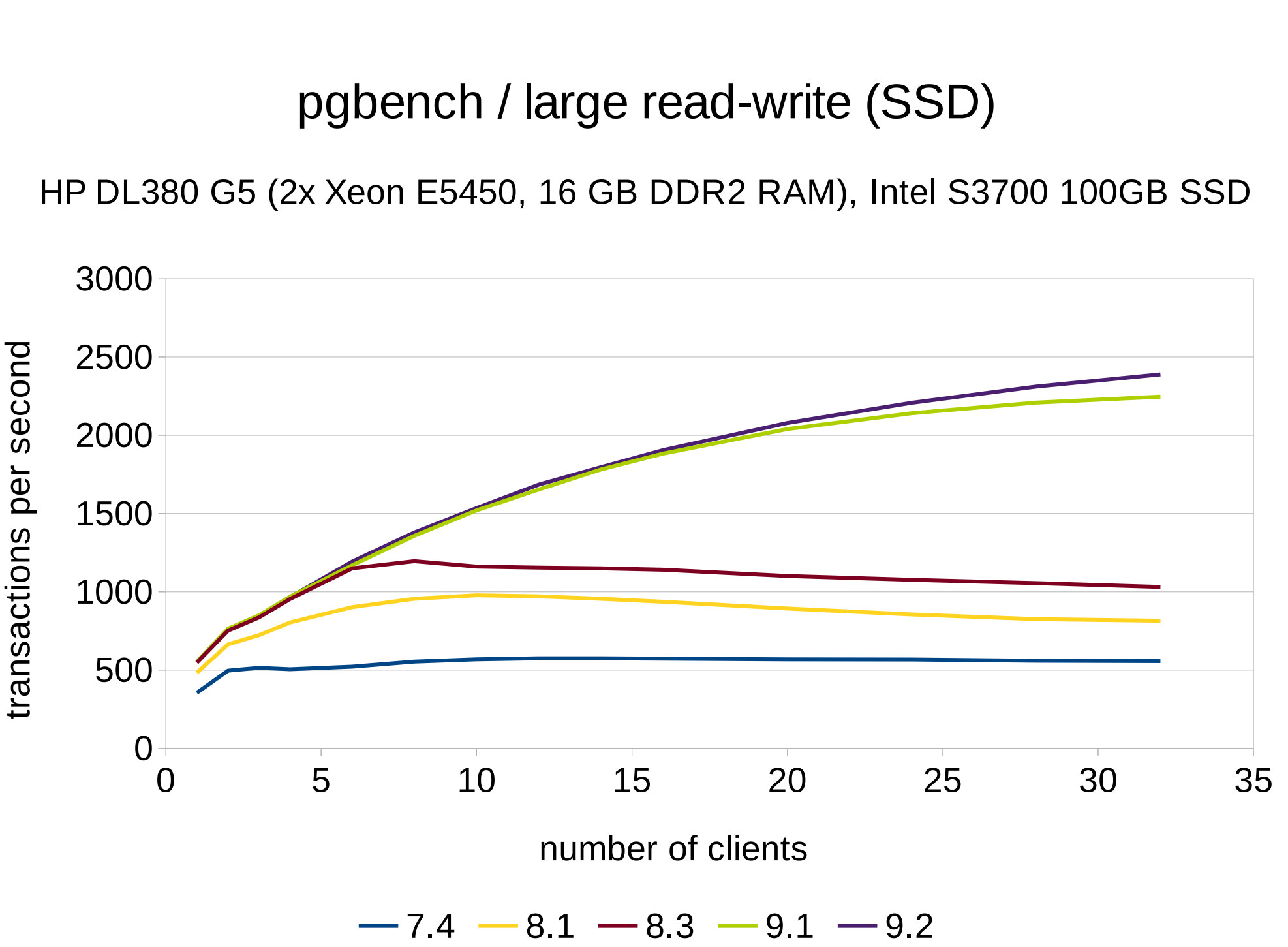
pgbench

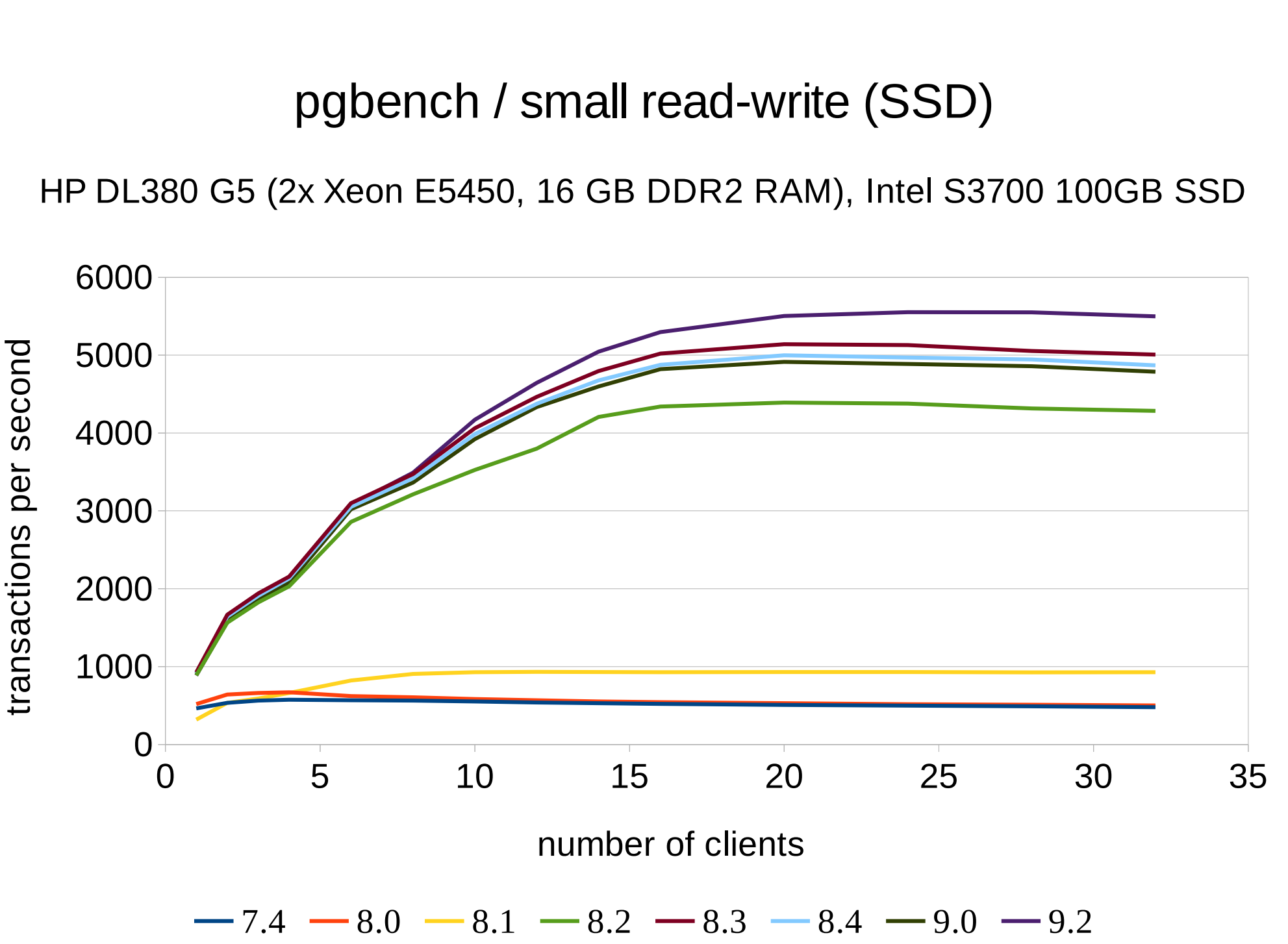
- tři velikosti datasetů
  - malý (150 MB) <– problémy se zámky, etc.
  - střední (~50% RAM) <– CPU bound
  - velký (~200% RAM) <– I/O bound
- dva módy
  - read-only a read-write
- rozsah klientů (1, 2, 4, ..., 32)

```
BEGIN;

    UPDATE accounts SET abalance = abalance + :delta
     WHERE aid = :aid;

    SELECT abalance FROM accounts WHERE aid = :aid;

    UPDATE tellers SET tbalance = tbalance + :delta
     WHERE tid = :tid;

    UPDATE branches SET bbalance = bbalance + :delta
     WHERE bid = :bid;

    INSERT INTO history (tid, bid, aid, delta, mtime)
    VALUES (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);

END;
```


pgbench / large read-only (on SSD)
HP DL380 G5 (2x Xeon E5450, 16 GB DDR2 RAM), Intel S3700 100GB SSD
transactions per second
12000
10000
8000
6000
4000
2000
0
0
5
10
15
20
25
30
35
number of clients
7.4
8.0
8.1
head


pgbench / medium read-only (SSD)
HP DL380 G5 (2x Xeon E5450, 16 GB DDR2 RAM), Intel S3700 100GB SSD
transactions per second
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
30
35
number of clients
8.0
8.1
8.2
8.3
9.0
9.2

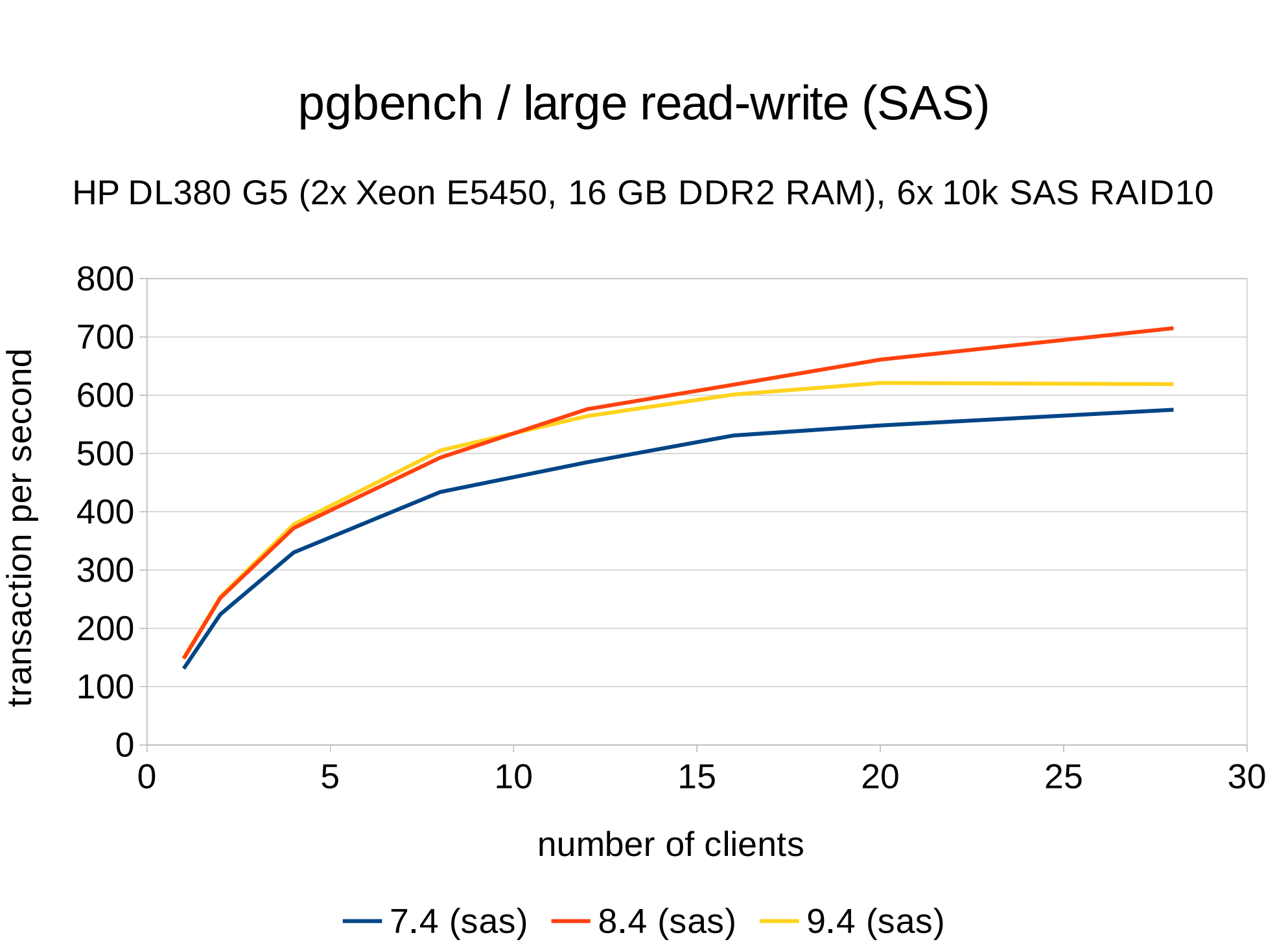

pgbench / large read-write (SSD)
HP DL380 G5 (2x Xeon E5450, 16 GB DDR2 RAM), Intel S3700 100GB SSD
transactions per second
3000
2500
2000
1500
1000
500
0
0
5
10
15
20
25
30
35
number of clients
7.4
8.1
8.3
9.1
9.2


pgbench / small read-write (SSD)
HP DL380 G5 (2x Xeon E5450, 16 GB DDR2 RAM), Intel S3700 100GB SSD
transactions per second
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
number of clients
7.4
8.0
8.1
8.2
8.3
8.4
9.0
9.2

Co rotační disky?

6 x 10k SAS drives (RAID 10)

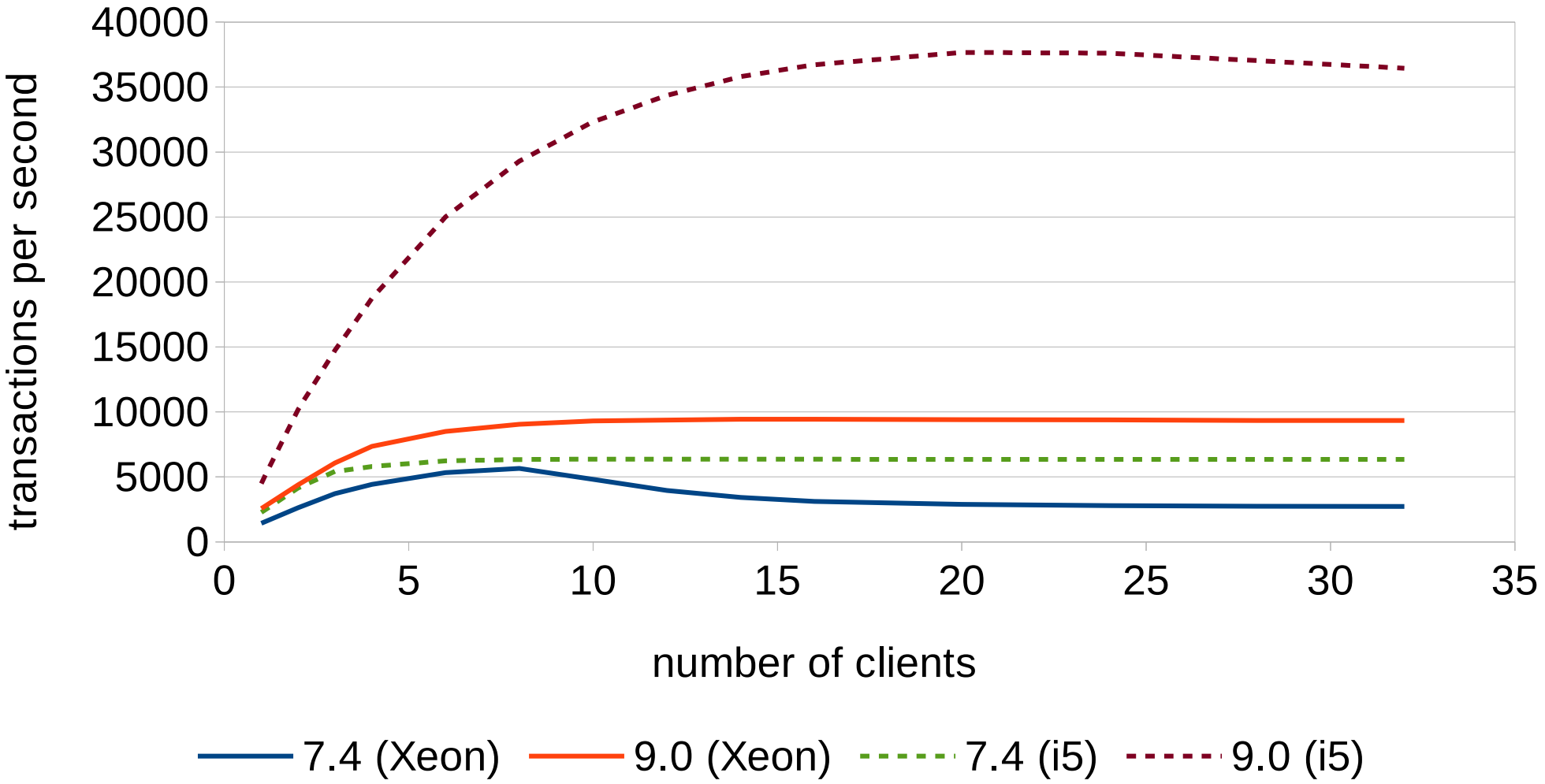
P400 with 512MB write cache


pgbench / large read-write (SAS)
HP DL380 G5 (2x Xeon E5450, 16 GB DDR2 RAM), 6x 10k SAS RAID10
transaction per second
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
number of clients
7.4 (sas)
8.4 (sas)
9.4 (sas)

No a co ta i5-2500k mašina?

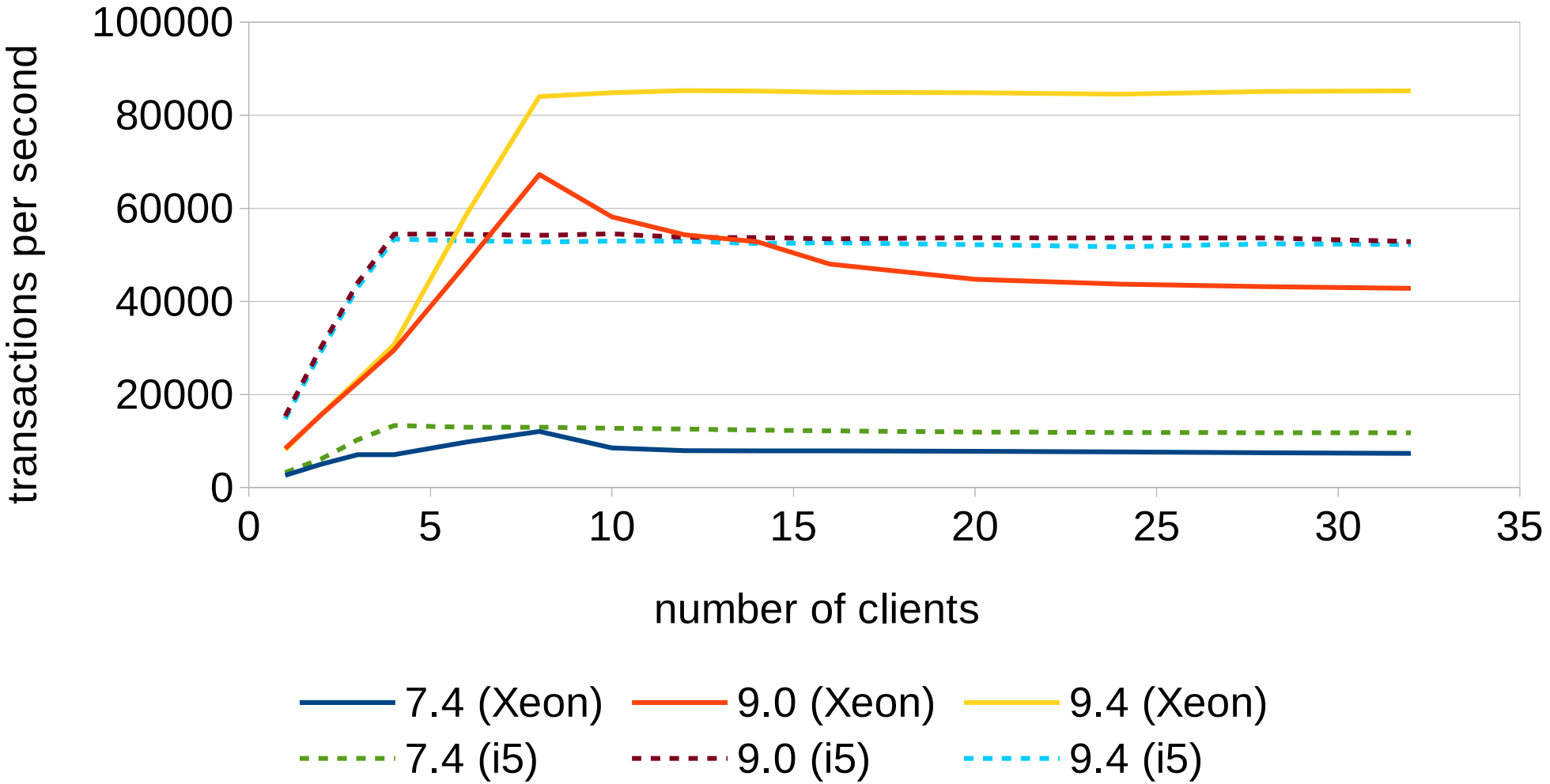
pgbench / large read-only (Xeon vs. i5)

2x Xeon E5450 (3GHz), 16 GB DDR2 RAM, Intel S3700 100GB SSD
i5-2500k (3.3 GHz), 8GB DDR3 RAM, Intel S3700 100GB SSD



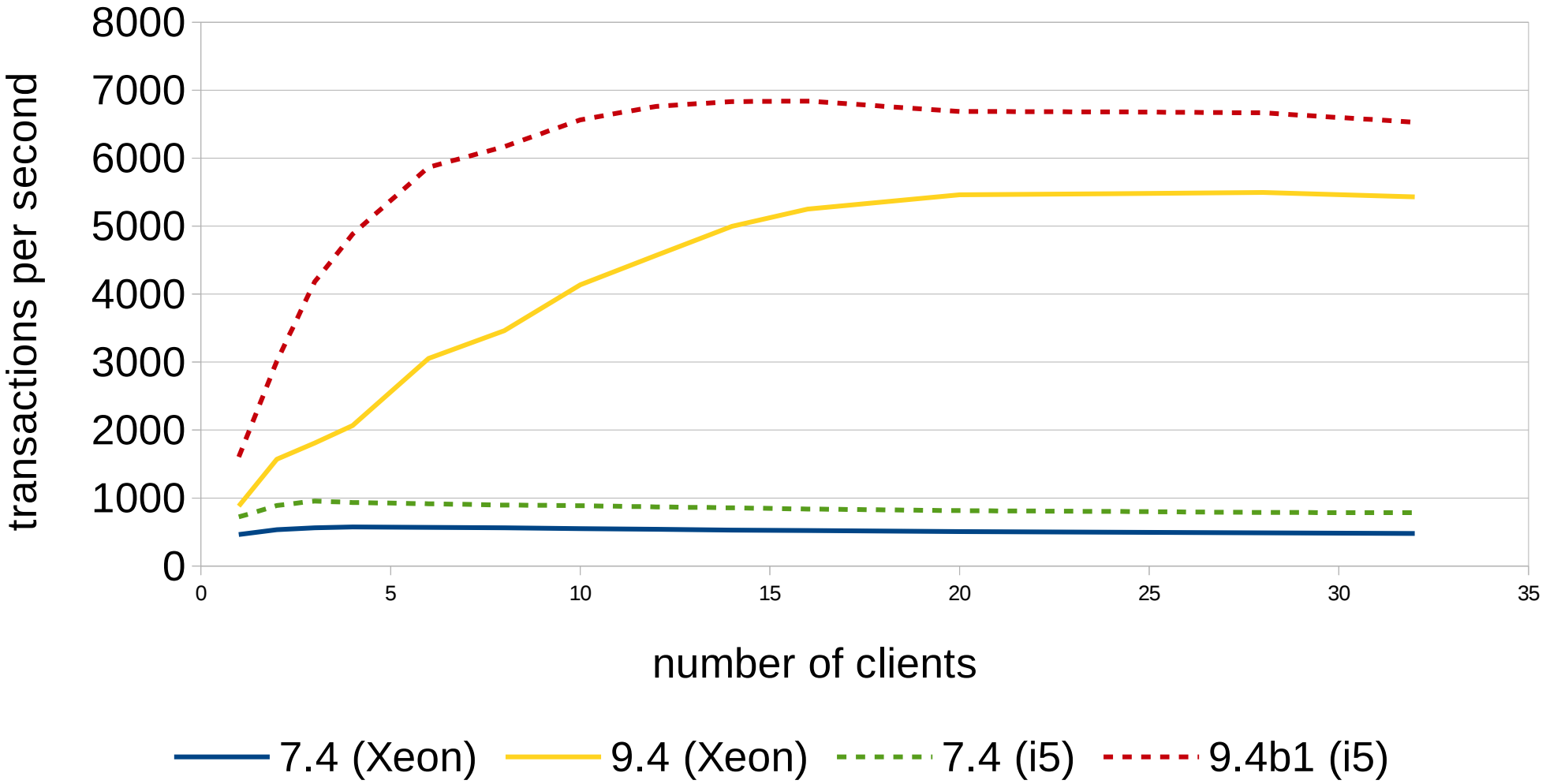
pgbench / small read-only (Xeon vs. i5)

2x Xeon E5450 (3GHz), 16 GB DDR2 RAM, Intel S3700 100GB SSD
i5-2500k (3.3 GHz), 8GB DDR3 RAM, Intel S3700 100GB SSD

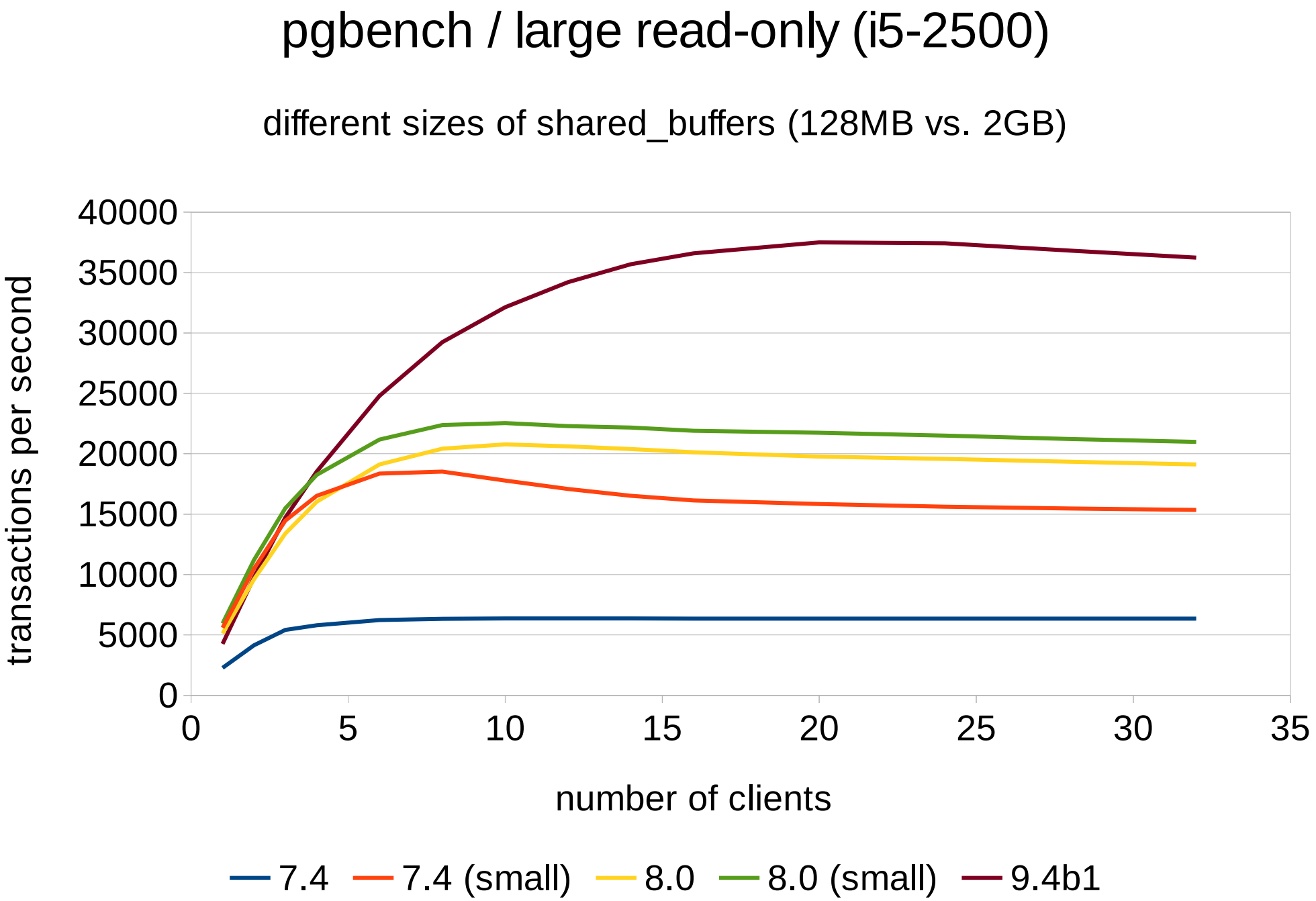


pgbench / small read-write (Xeon vs. i5)

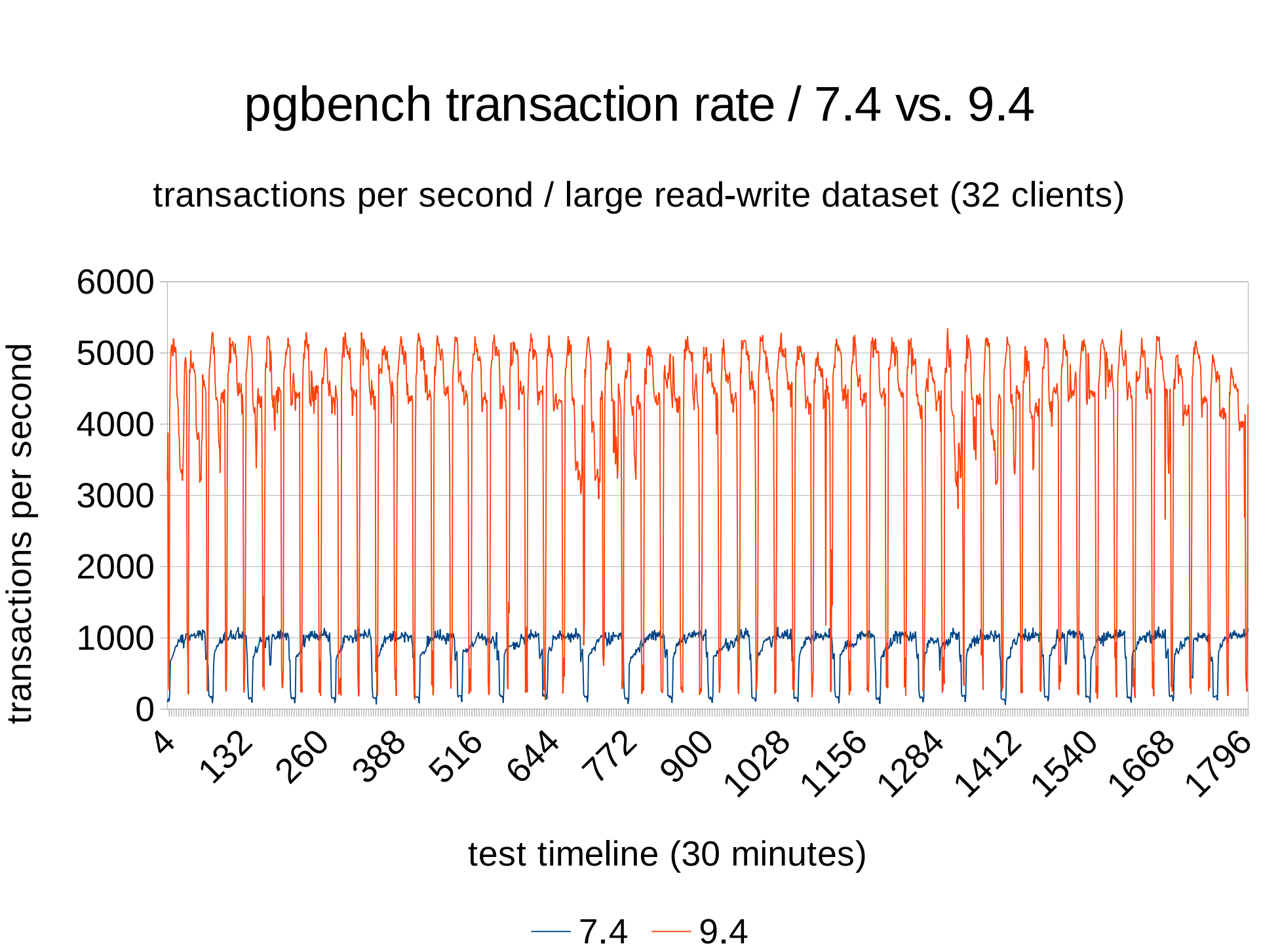
2x Xeon E5450 (3GHz), 16 GB DDR2 RAM, Intel S3700 100GB SSD
i5-2500k (3.3 GHz), 8GB DDR3 RAM, Intel S3700 100GB SSD



Legendy říkají že starší verze lépe fungují s nižšími paměťovými limity (shared_buffers etc.)


pgbench / large read-only (i5-2500)
different sizes of shared_buffers (128MB vs. 2GB)
transactions per second
40000
35000
30000
25000
20000
15000
10000
5000
0
0
5
10
15
20
25
30
35
number of clients
7.4
7.4 (small)
8.0
8.0 (small)
9.4b1

transactions per second vs. latence


pgbench transaction rate / 7.4 vs. 9.4
transactions per second / large read-write dataset (32 clients)
transactions per second
6000
5000
4000
3000
2000
1000
0
4
132
260
388
516
644
772
900
1028
1156
1284
1412
1540
1668
1796
test timeline (30 minutes)
7.4
9.4

pgbench transaction latency / 7.4 vs. 9.4

pgbench / shrnutí

- značná vylepšení
- vylepšené zamykání
  - lepší škálování na velké počty jader (64 ...)
- mnoho dalších optimalizací
  - výrazné zrychlení i na malém počtu klientů
- lessons learned
  - frekvence procesoru není míra výkonu
  - počet jader není míra výkonu

TPC-DS

“Decision Support” benchmark

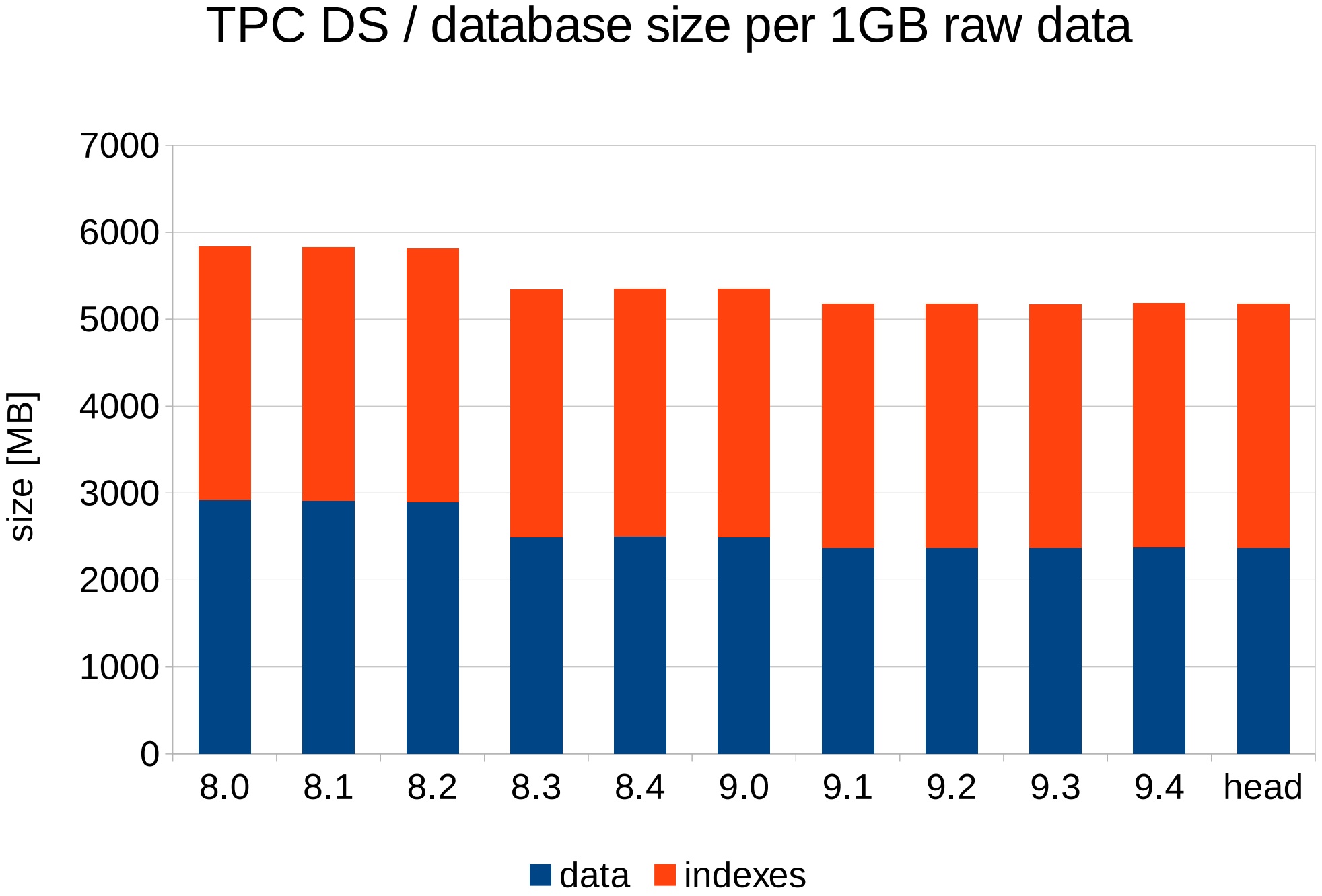
(aka “Data Warehouse” benchmark)

TPC-DS

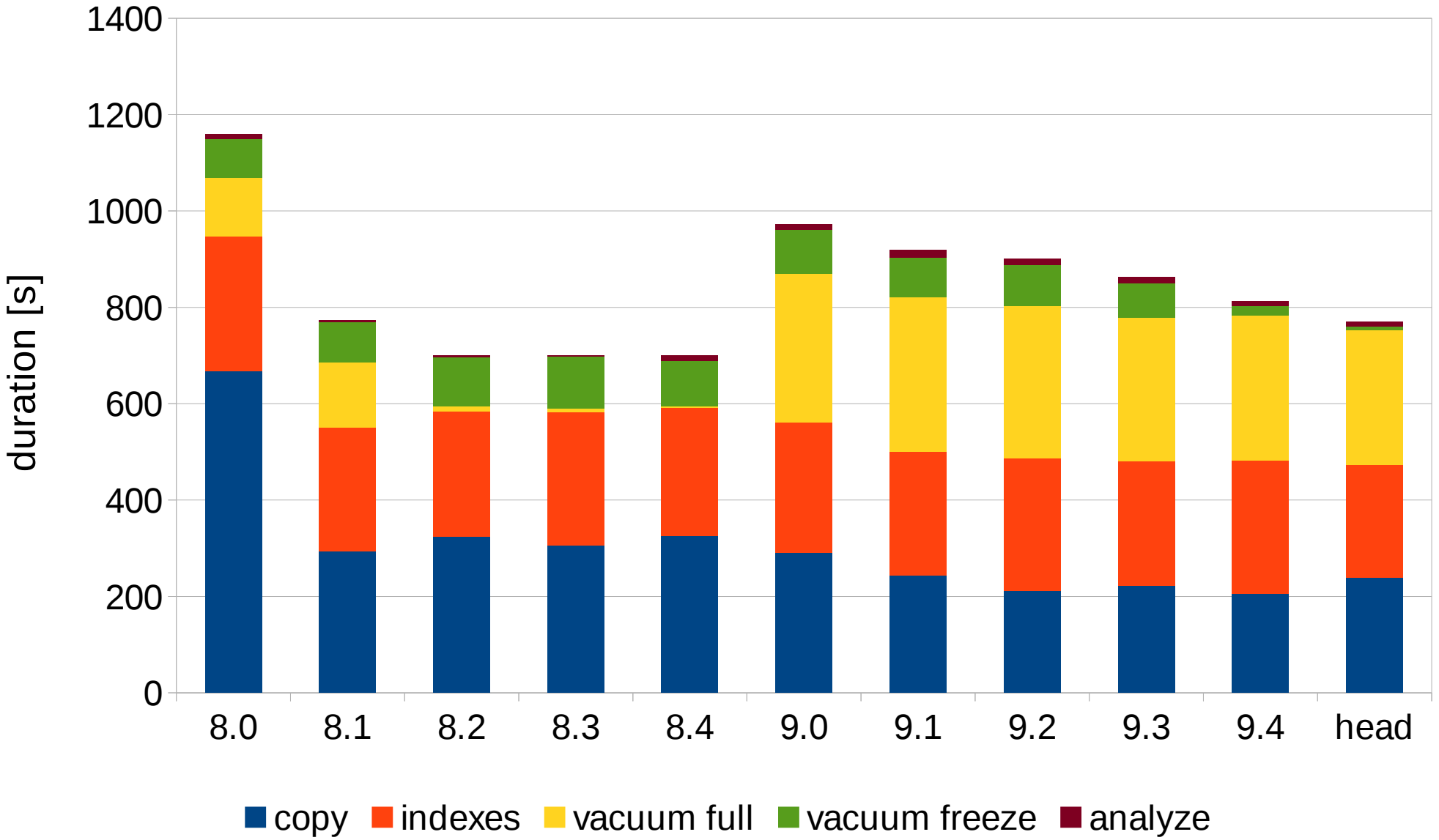
- orientováno na analytiku / warehousing
  - dotazy drtící velké objemy dat (GROUP BY, JOIN)
  - neuniformní rozdělení dat (realističtější než TPC-H)
- definováno 99 šablon dotazů (TPC-H jen 22)
  - některé rozbité (padá generátor)
  - některé zatím nepodporované (ROLLUP/CUBE)
  - 41 dotazů pro >= 7.4
  - 61 dotazů pro >= 8.4 (CTE, Window functions)

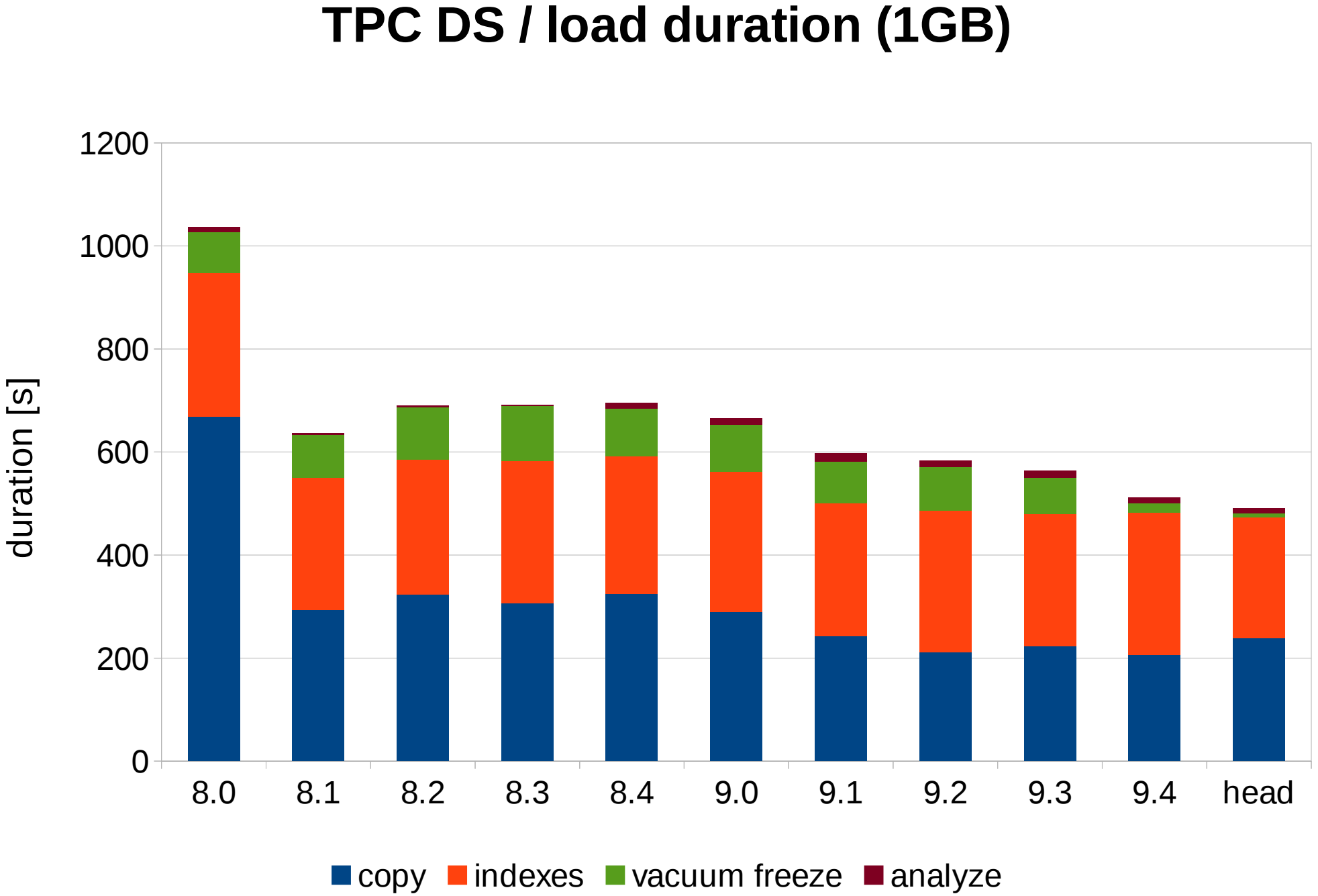
TPC-DS

- **1GB and 16GB datasets (raw data)**
  - 1GB nedostatečný pro publikaci, 16GB je nestandardní (dle TPC)
- **ale i tak je to zajímavé ...**
  - většina produkčních databází se vejde do 16GB
  - ukazuje to trendy (do jisté míry aplikovatelné na větší DB)
- **schéma**
  - víceméně defaultní (standard compliance FTW!)
  - stejné pro všechny verze (indexy na FK/join keys, pár dalších)
  - nepochybně možno dál zoptimalizovat

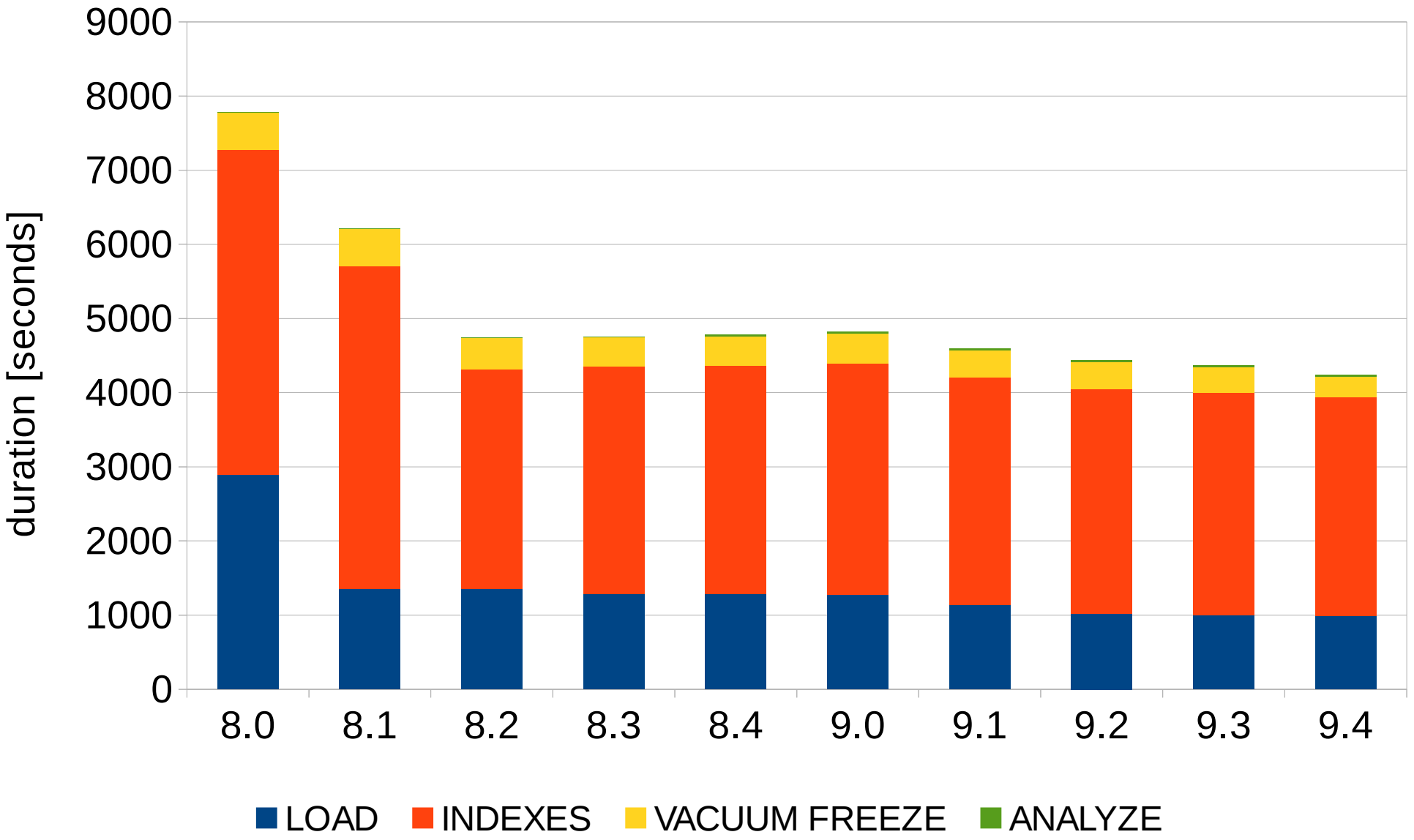

TPC DS / database size per 1GB raw data
size [MB]
7000
6000
5000
4000
3000
2000
1000
0
8.0
8.1
8.2
8.3
8.4
9.0
9.1
9.2
9.3
9.4
head
data
indexes

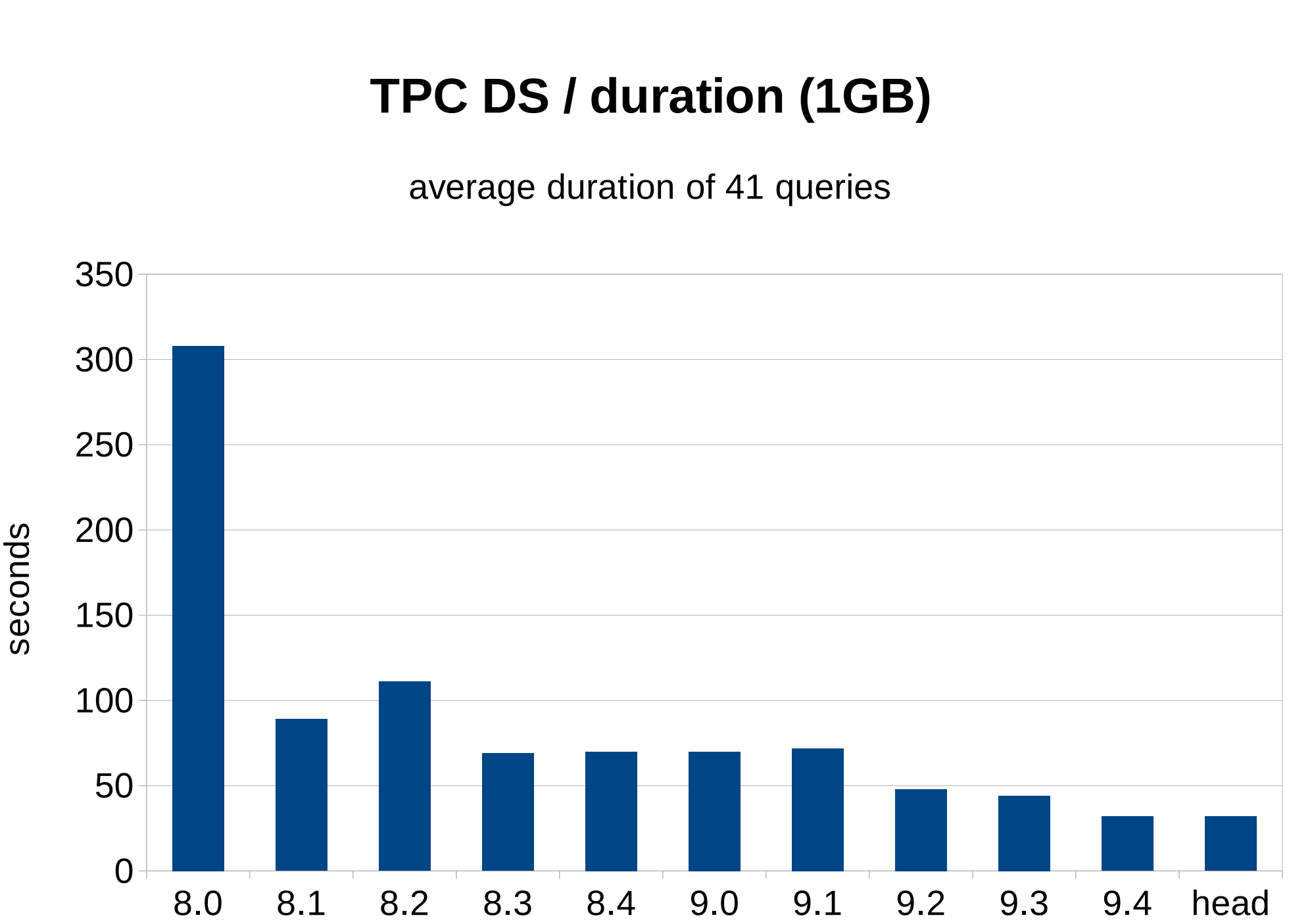
TPC DS / load duration (1GB)



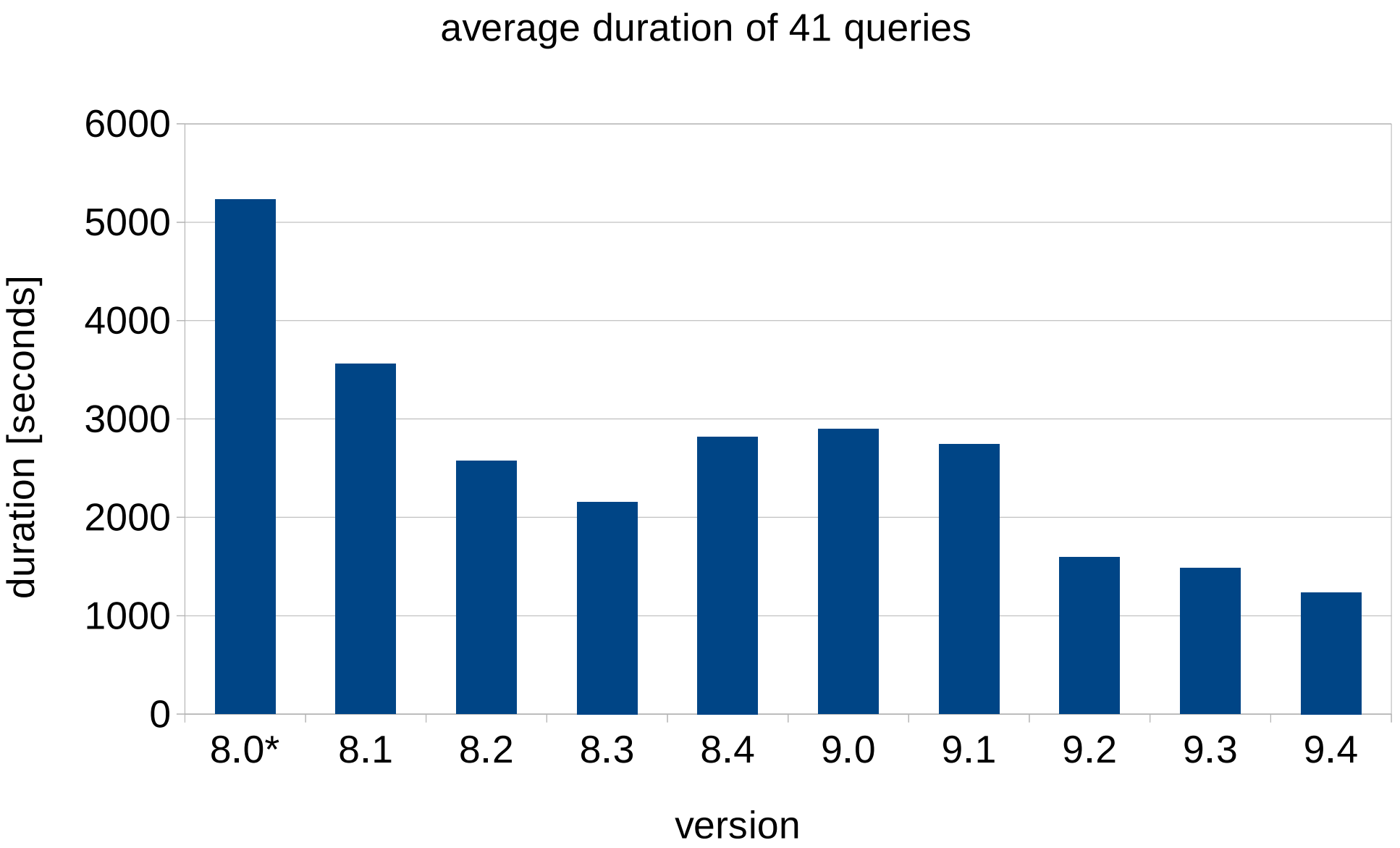

TPC DS / load duration (1GB)
duration [s]
1200
1000
800
600
400
200
0
8.0
8.1
8.2
8.3
8.4
9.0
9.1
9.2
9.3
9.4
head
copy
indexes
vacuum freeze
analyze

TPC DS / load duration (16 GB)




TPC DS / duration (1GB)
average duration of 41 queries
seconds
350
300
250
200
150
100
50
0
8.0
8.1
8.2
8.3
8.4
9.0
9.1
9.2
9.3
9.4
head

TPC DS / duration (16 GB)



TPC-DS / shrnutí

- **výrazně rychlejší load dat**
  - většina času se vytváří indexy (paralelizace, RAM)
  - pokud ignorujeme VACUUM FULL (změna implementace v 9.0)
  - mírné snížení velikosti
- **výrazně rychlejší dotazy**
  - značné zrychlení dotazů (~6x)
  - širší použití indexů, index only scany

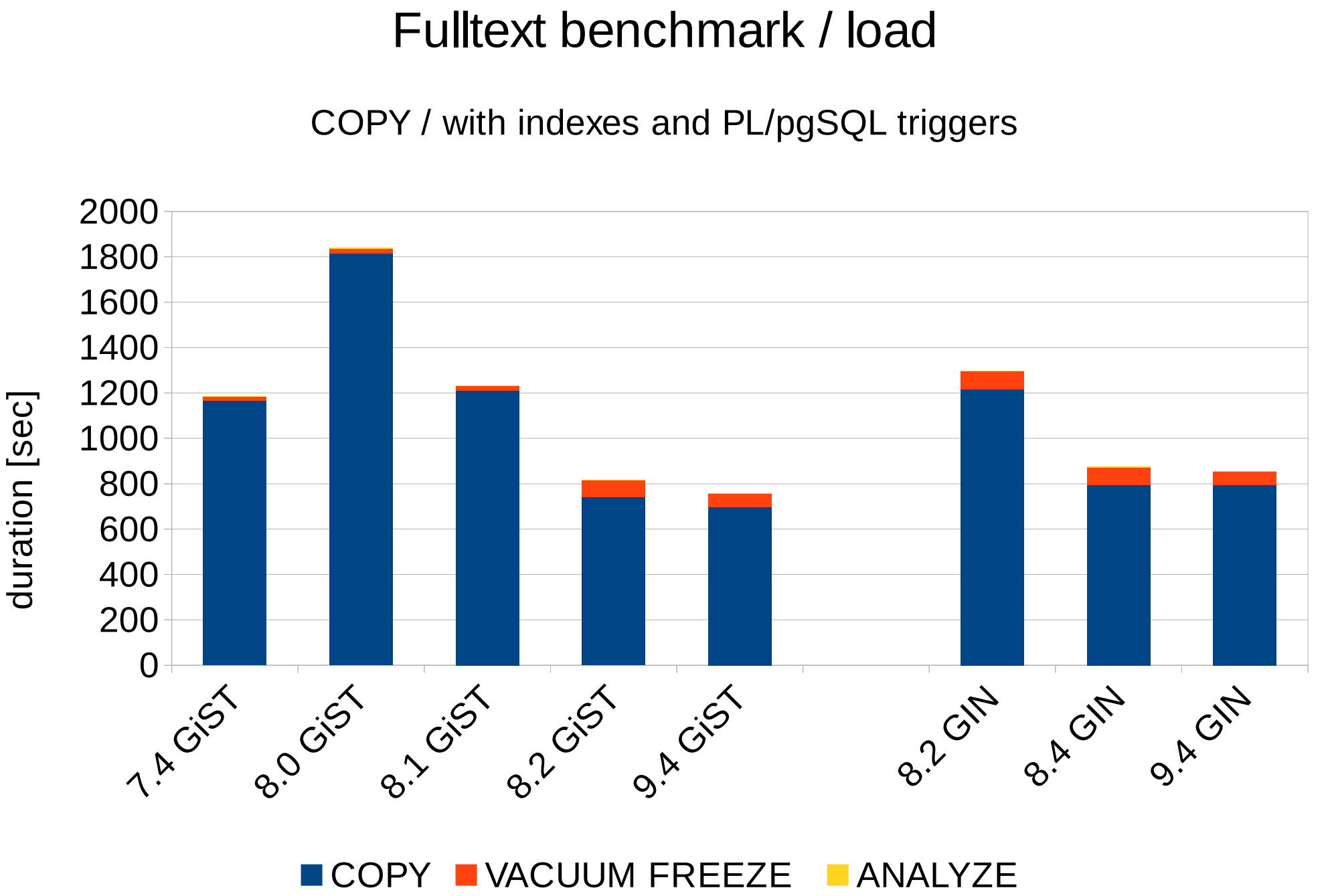
Fulltext Benchmark

testování GIN a GiST indexů
prostřednictvím fulltextu

Fulltext benchmark

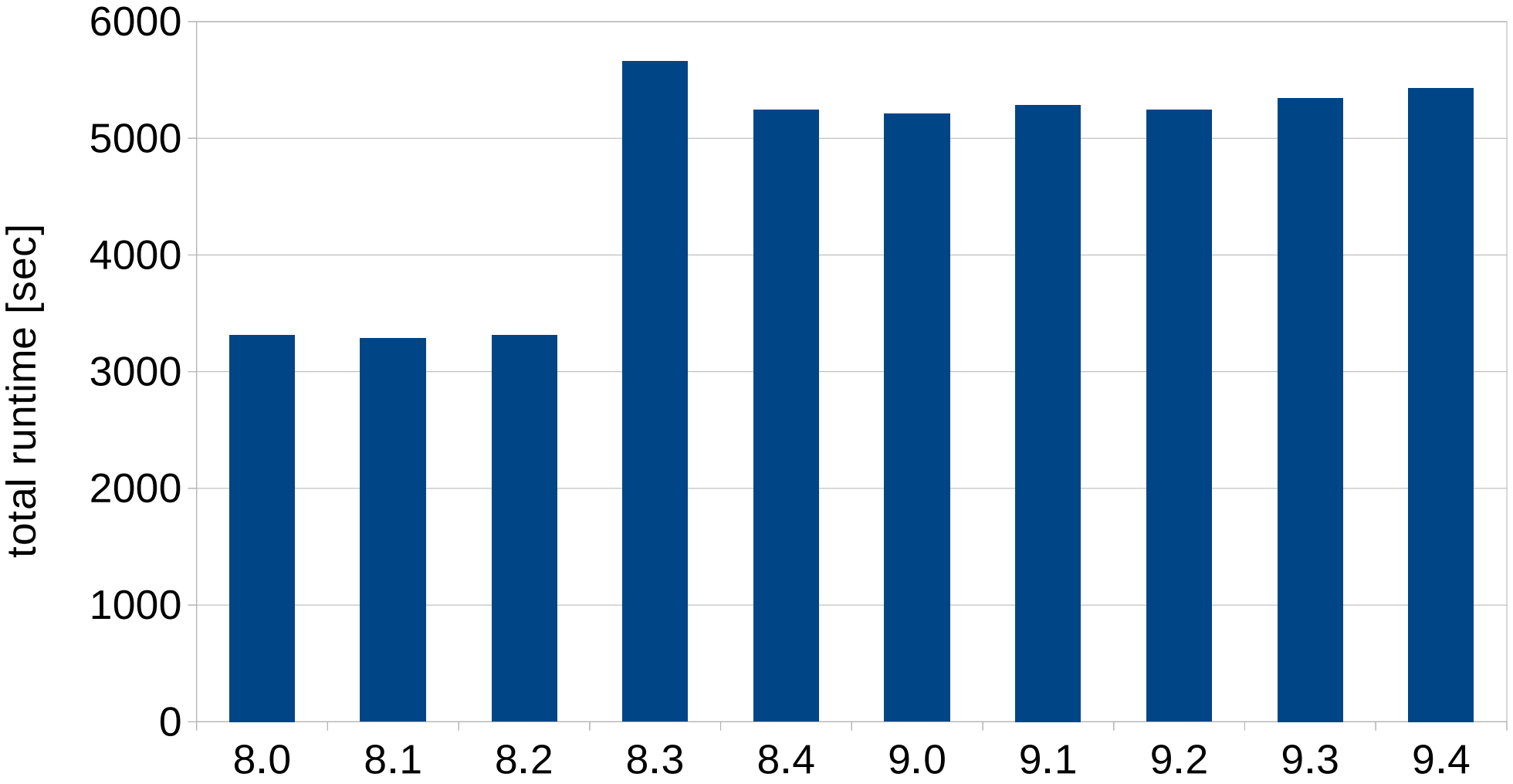
- prohledávání archivů pgsql mailing listů
  - ~1M zpráv, ~5GB dat
- ~33k reálných dotazů (z postgresql.org)
  - syntetické dotazy dávají cca stejné výsledky

```
SELECT id FROM messages
    WHERE body @@ ('high & performance')::tsquery
    ORDER BY ts_rank(body, ('high & performance')::tsquery)
    DESC LIMIT 100;
```


Fulltext benchmark / load
COPY / with indexes and PL/pgSQL triggers
duration [sec]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
7.4 GiST
8.0 GiST
8.1 GiST
8.2 GiST
9.4 GiST
8.2 GIN
8.4 GIN
9.4 GIN
COPY
VACUUM FREEZE
ANALYZE

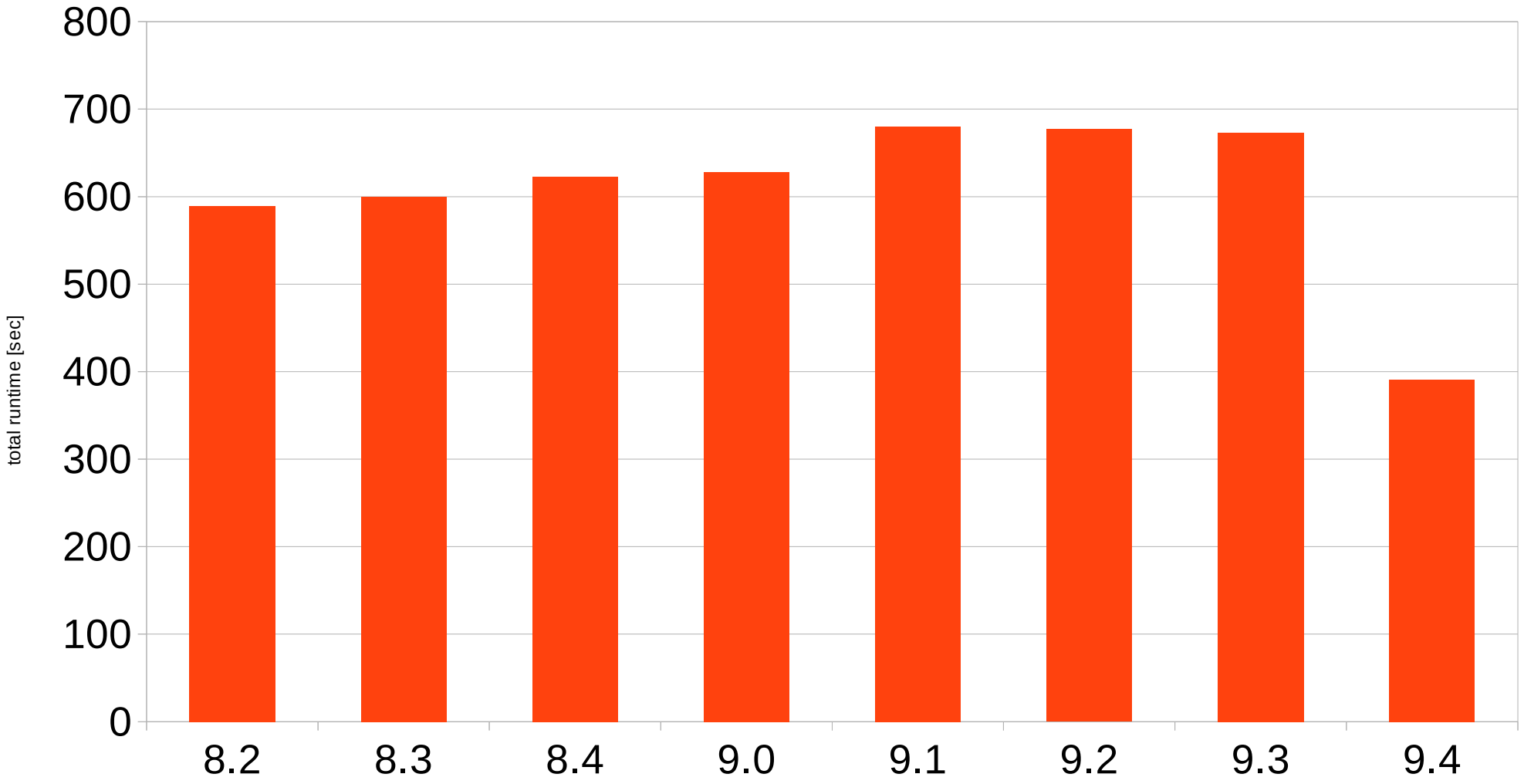
Fulltext benchmark / GiST

33k queries from postgresql.org [TOP 100]

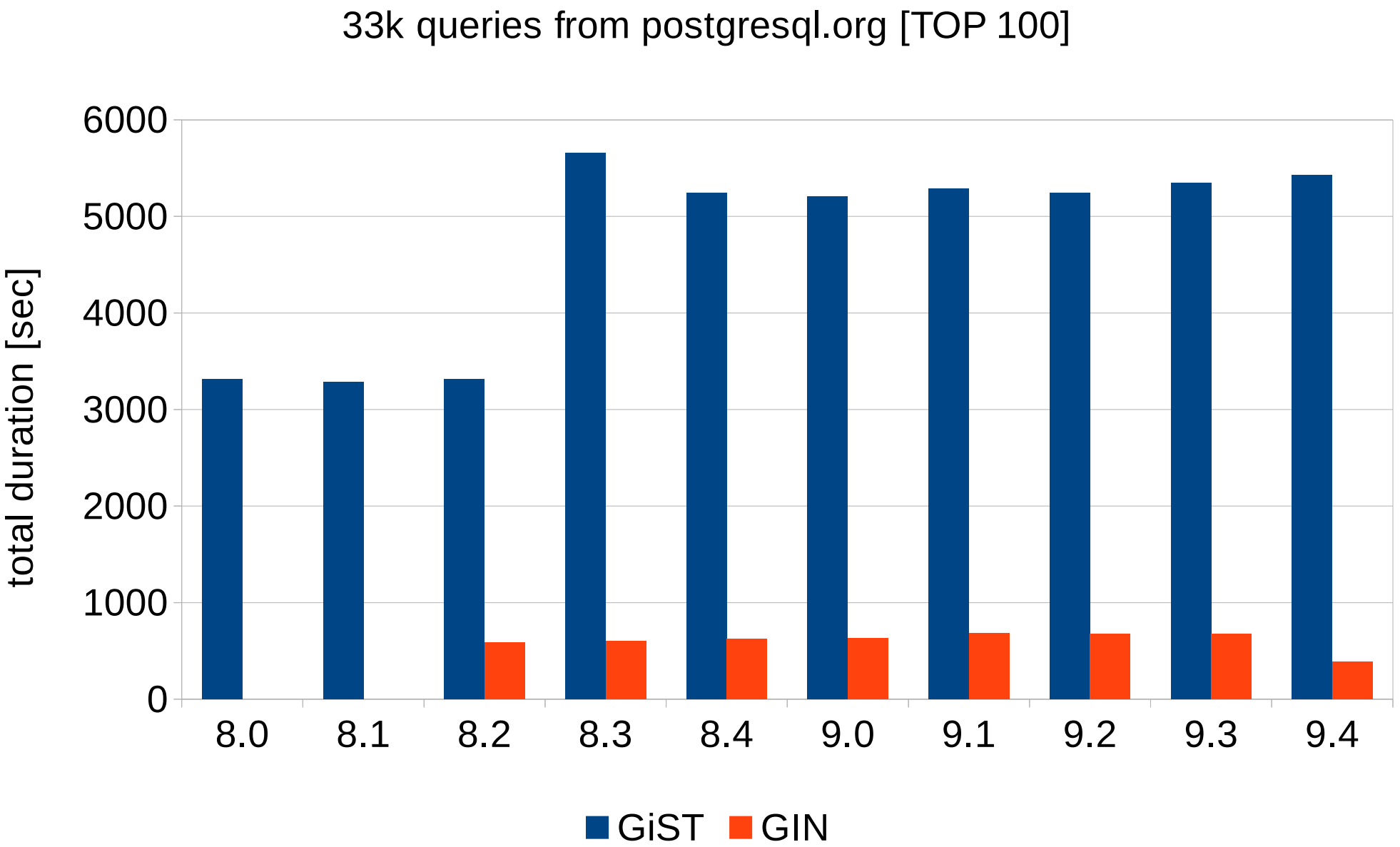


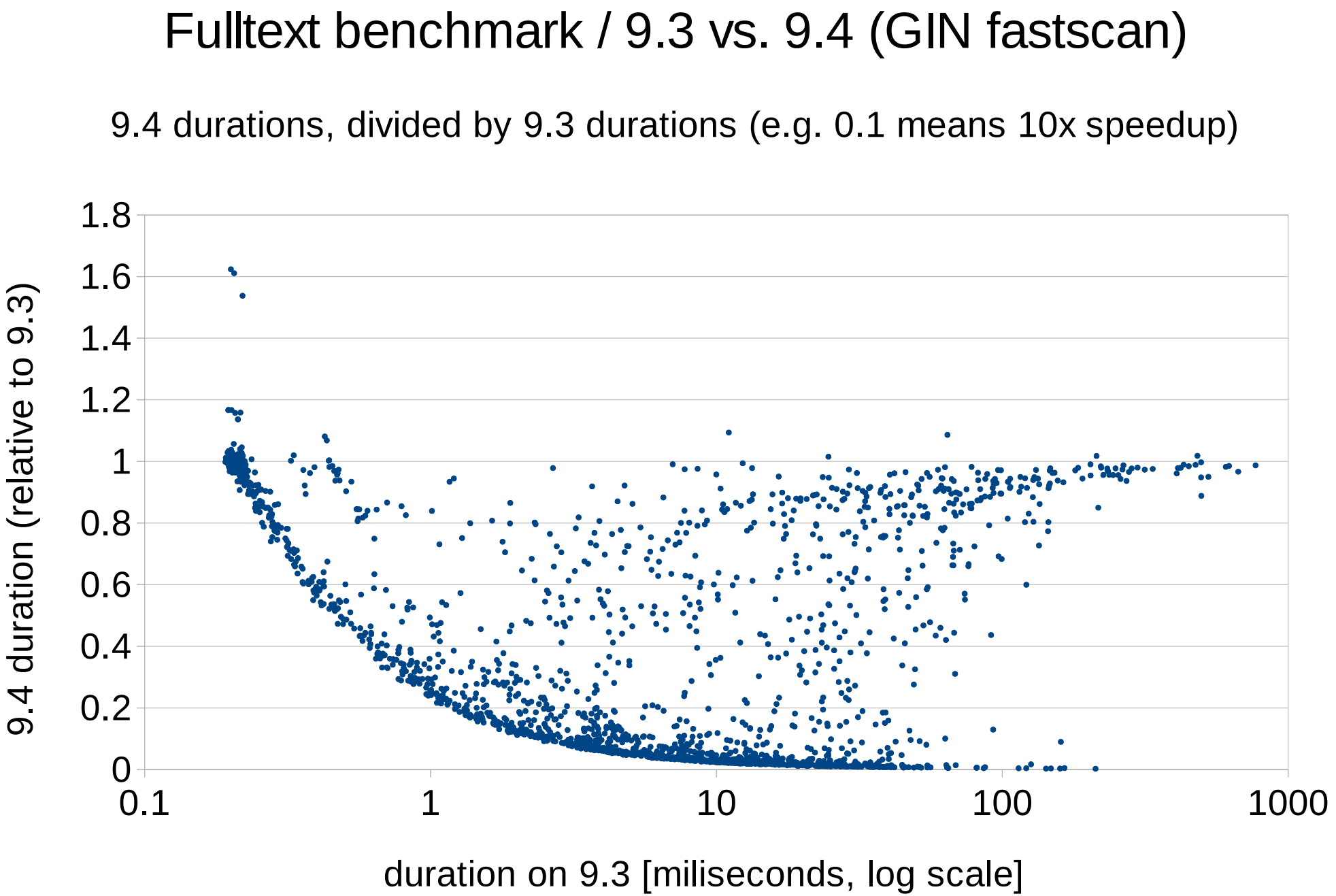
Fulltext benchmark / GIN

33k queries from postgresql.org [TOP 100]



Fulltext benchmark - GiST vs. GIN




Fulltext benchmark / 9.3 vs. 9.4 (GIN fastscan)
9.4 durations, divided by 9.3 durations (e.g. 0.1 means 10x speedup)
9.4 duration (relative to 9.3)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.1
1
10
100
1000
duration on 9.3 [miliseconds, log scale]

Fulltext / shrnutí

- GIN fastscan
  - dotazy kombinující “časté & vzácné”
  - 9.4 skenuje “vzácné” seznam první
  - exponenciální zrychlení pro tyto dotazy
  - ... to je celkem fajn ;-)
- jenom ~5% dotazů se zpomalilo
  - víceméně dotazy pod 1ms (chyby měření)