



Диагональное масштабирование PostgreSQL



Что такое PostgreSQL



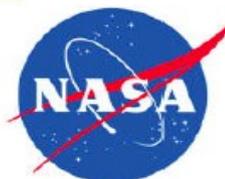
PostgreSQL – это свободно распространяемая объектно-реляционная СУБД (ORDBMS)

Поддержка ANSI SQL (1992...2011),
а также NoSQL (key-value, JSON, JSONB)

Web: <http://www.postgresql.org>

Лицензия: BSD, MIT-like

Кто использует



Кроссплатформенность vs Oracle/MSSQL

PostgreSQL поддерживает все виды Unix, включая Linux, FreeBSD, Solaris, HP-UX, Mac OS X, а также MS Windows, а также разнообразные архитектуры.

Конкурентная работа при большой нагрузке vs MySQL

PostgreSQL использует многоверсионность (MVCC) для обеспечения надежной и быстрой работы в конкурентных условиях под большой нагрузкой.

Независимость vs MySQL:

PostgreSQL не принадлежит ни одной компании, он развивается международным сообществом, в том числе и российскими разработчиками.

Превосходная поддержка vs Oracle/MSSQL:

Сообщество PostgreSQL предоставляет квалифицированную и быструю помощь. Коммерческие компании предлагают свои услуги по всему миру.

Расширяемость vs MySQL

Расширяемость PostgreSQL позволяет добавлять новую функциональность, например:

- * новые несоставные типы данных, без остановки сервера и своими силами.
- * бэкграунд воркеры
- * функции на языках программирования C/Python/Perl

Доступность vs Oracle/MSSQL:

PostgreSQL распространяется под лицензией BSD, которая не накладывает никаких ограничений на коммерческое использование и не требует лицензионных выплат. Вы можете даже продавать PostgreSQL под своим именем!

Независимость vs MySQL:

PostgreSQL не принадлежит ни одной компании, он развивается международным сообществом, в том числе и российскими разработчиками.

Предсказуемость vs MySQL:

PostgreSQL не принадлежит ни одной компании, он развивается международным сообществом, в том числе и российскими разработчиками.



Олег Бартунов, Федор Сигаев, Александр Коротков

Внесли вклад в PostgreSQL:

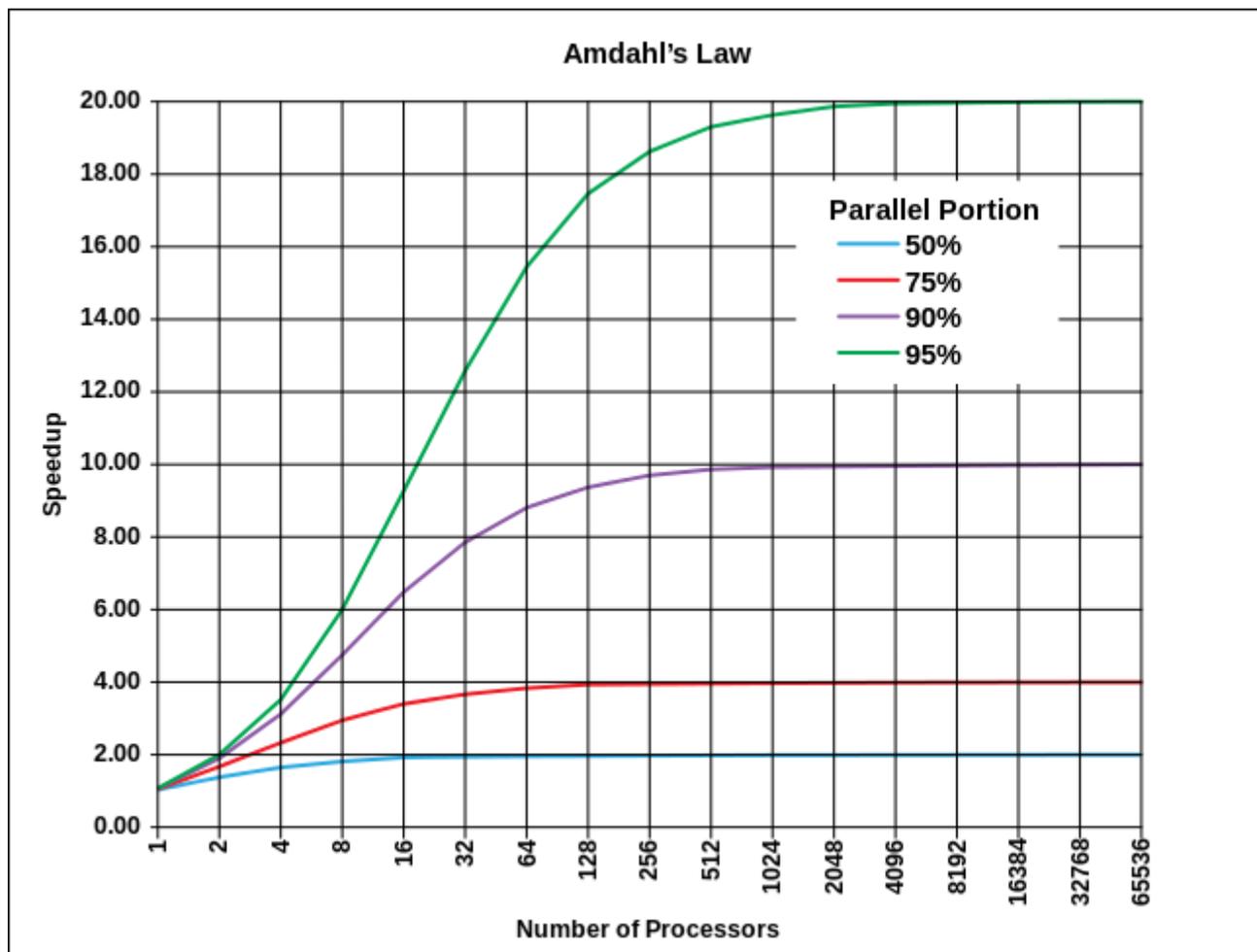
- Поддержка локализации
- Полнотекстовый поиск
- Слабоструктурированные данные
- Новые методы индексации
- Различные расширения

- Российский вендор PostgreSQL в России
 - Поддержка, разработка, консалтинг, обучение
- Тему технологической независимости СУБД пропагандируем с 2011 г.
- Члены международного сообщества
 - Спонсоры международных конференций (Канада, Австрия, Бразилия)
 - PgConf.Russia - крупнейшая в мире конференция по PostgreSQL
 - В направлениях, где мы ведем разработку, PostgreSQL является лидером* среди РСУБД
 - геоинформационные системы, слабоструктурированные данные, полнотекстовый поиск, расширяемость
 - Более 20 докладов на международных конференциях
- Все российские ключевые международно признанные разработчики PostgreSQL работают в нашей компании
- В нашей команде 4 кандидата наук: 3 – по PostgreSQL и технологиям БД

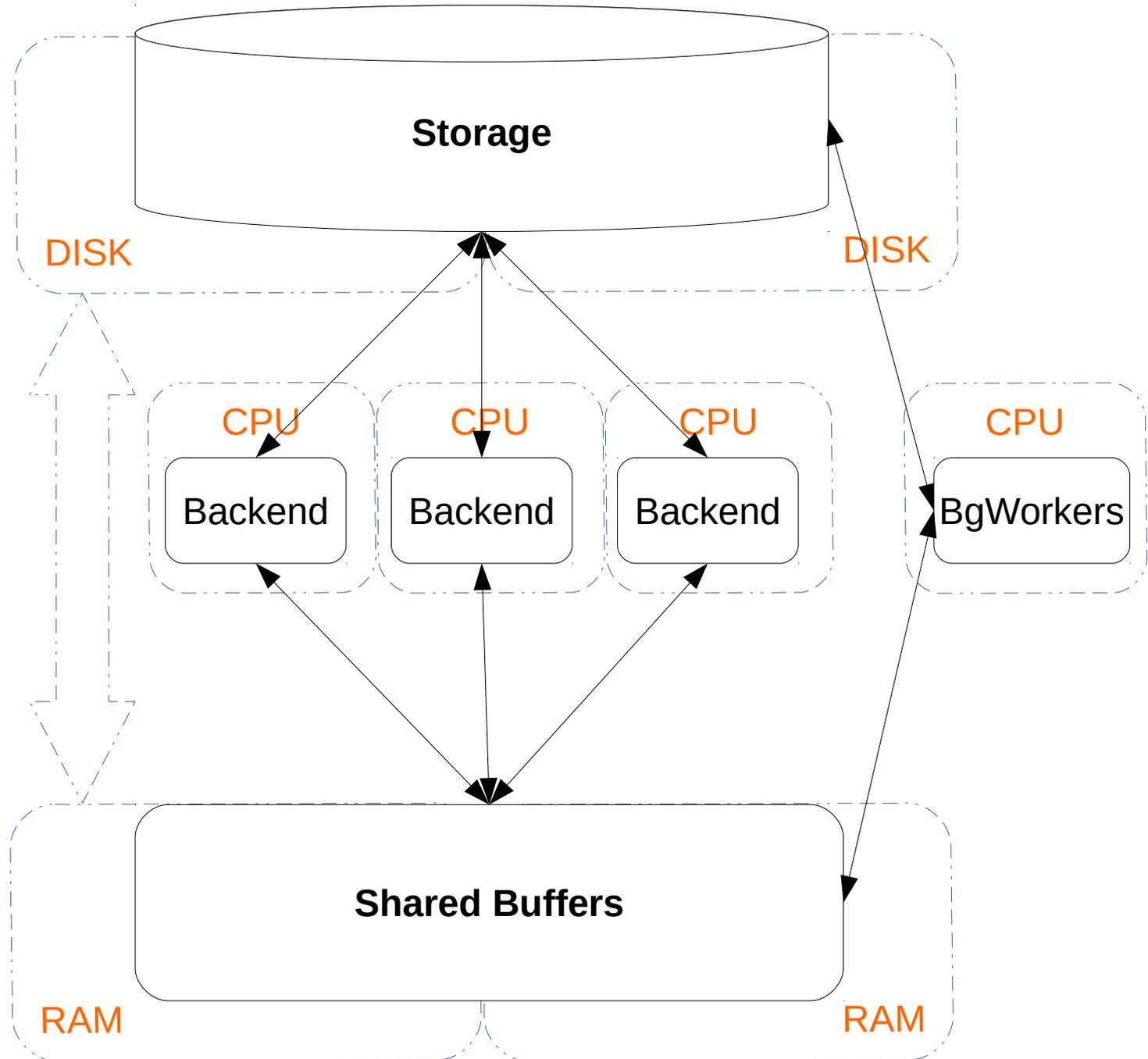
МАСШТАБИРУЕМОСТЬ *

* Это способность увеличивать свою производительность при добавлении ресурсов

Закон Амдала

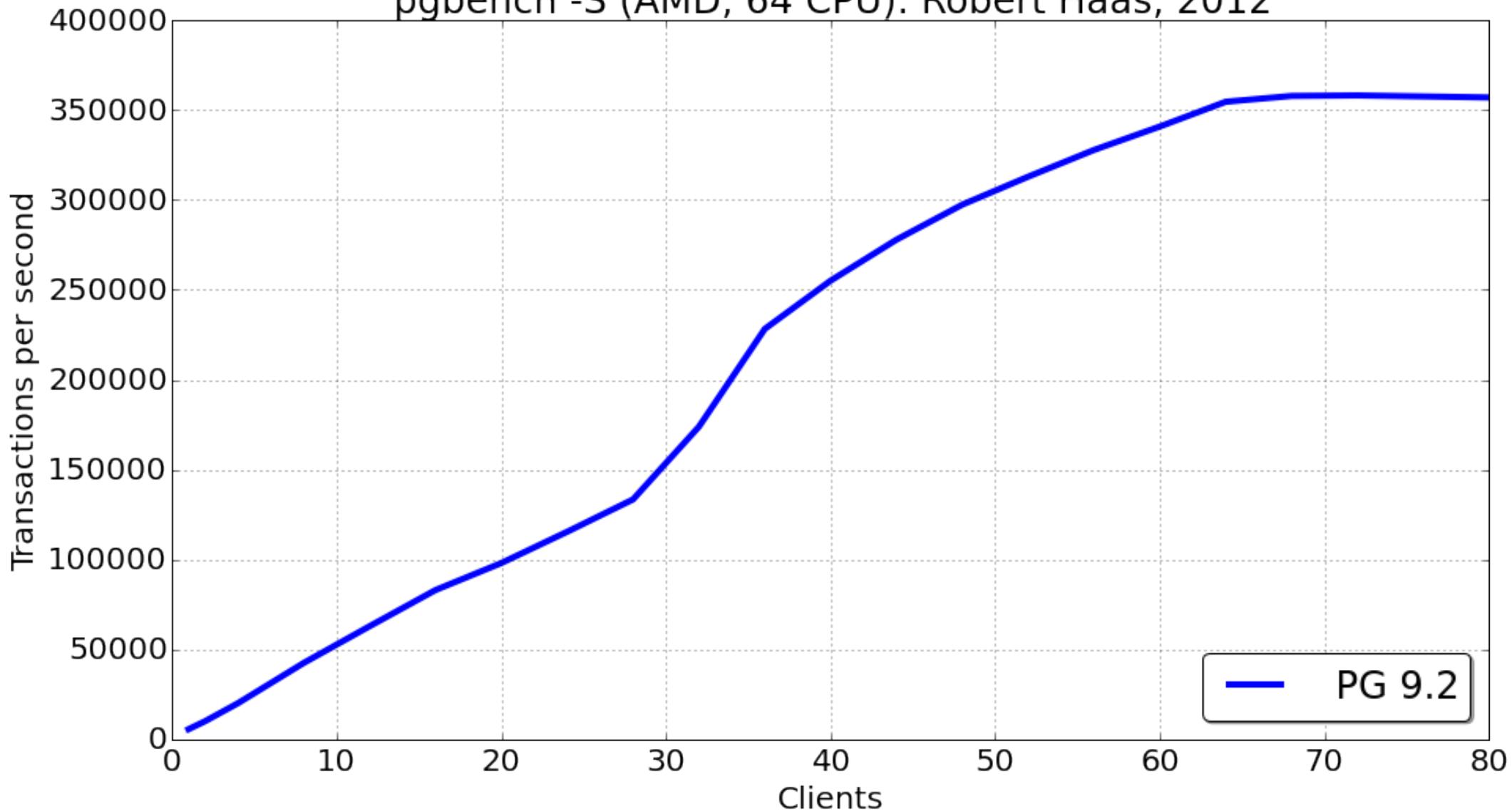


Архитектура с точки зрения ОС



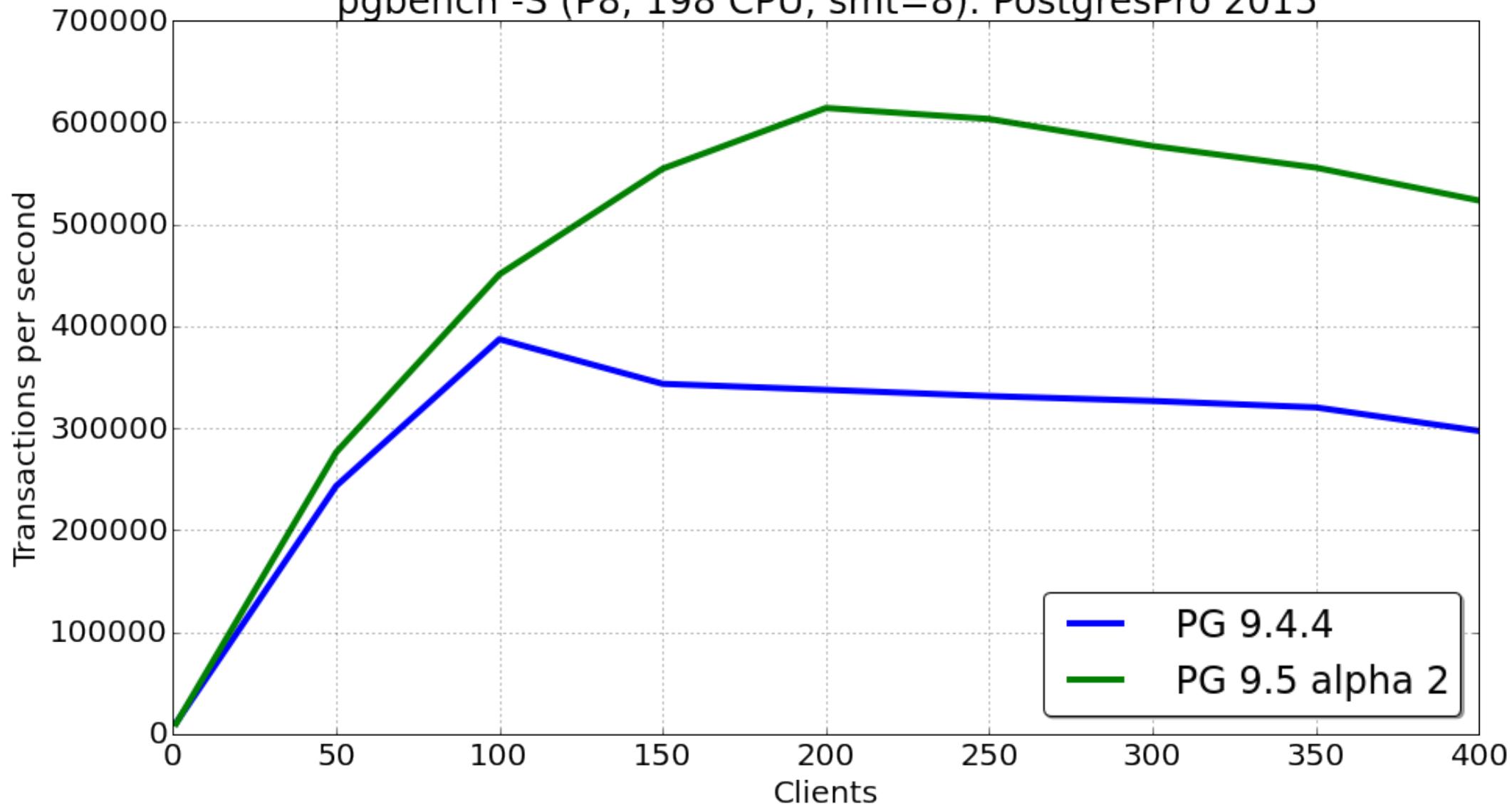
Масштабирование 9.2 (select-only)

pgbench -S (AMD, 64 CPU). Robert Haas, 2012

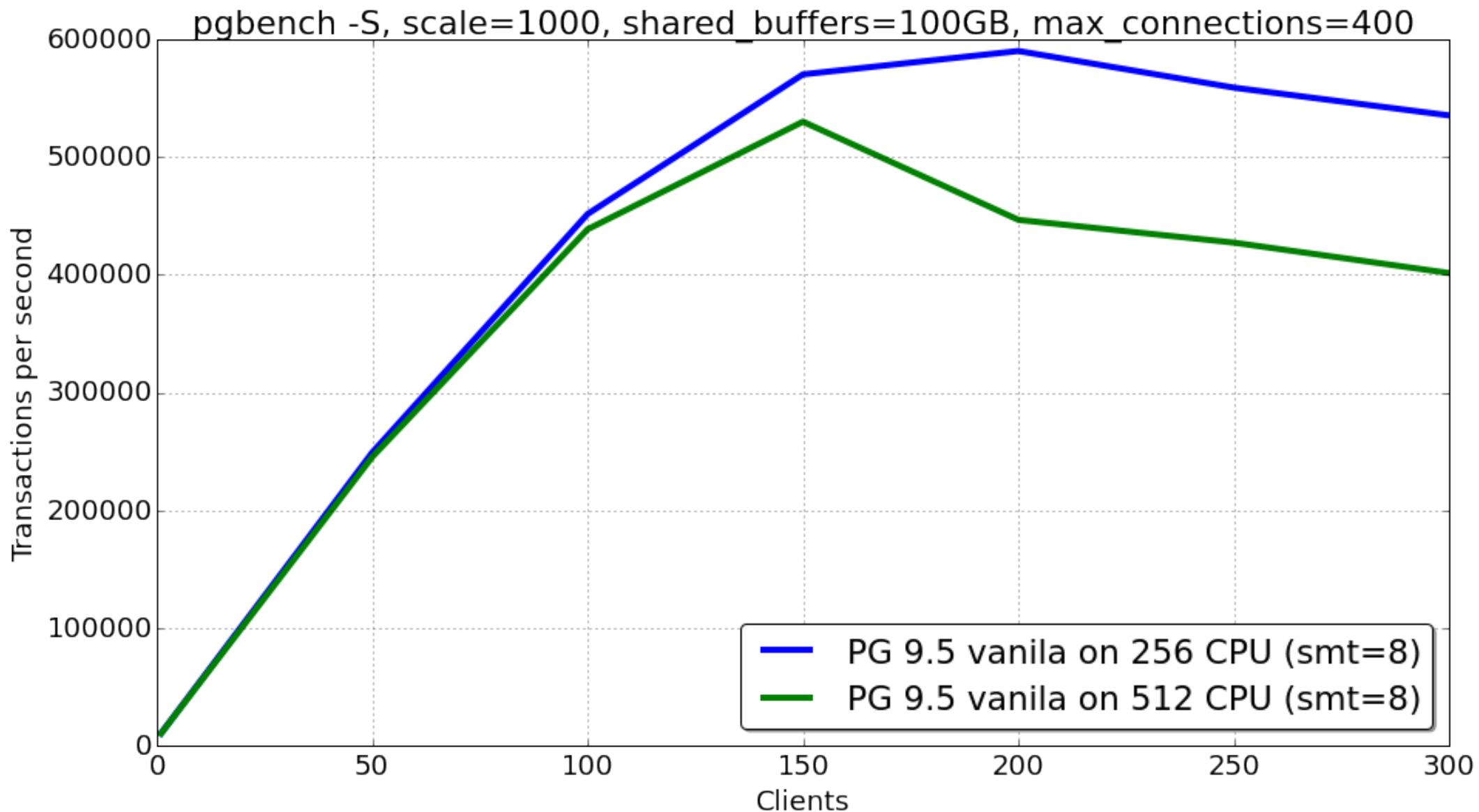


Производительность 9.4-9.5 (select-only)

pgbench -S (P8, 198 CPU, smt=8). PostgresPro 2015

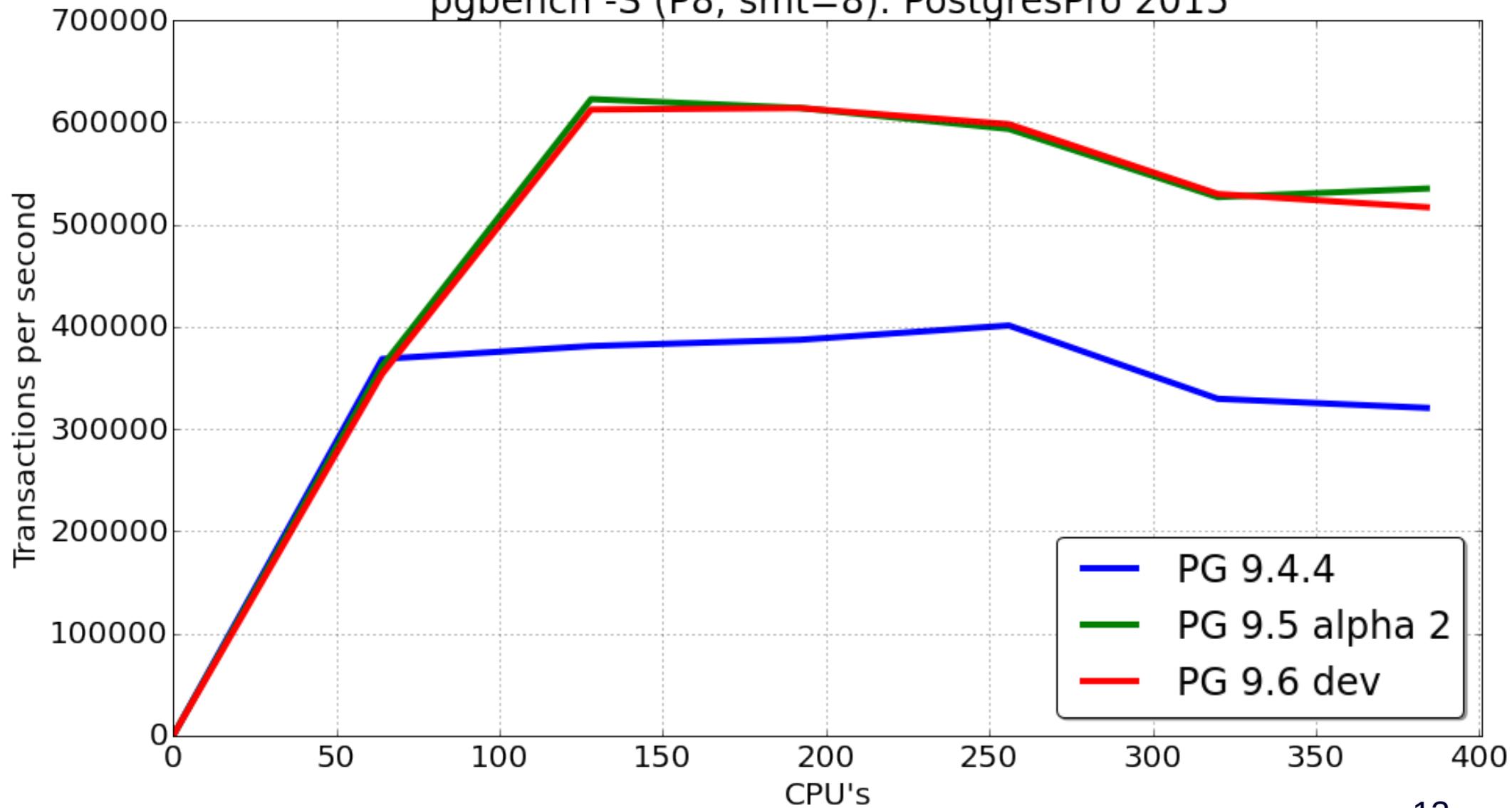


Масштабирование 9.5



Масштабирование 9.4-9.5 (select-only)

pgbench -S (P8, smt=8). PostgresPro 2015



* Semaphores

- + Простота реализации (Posix)
- Низкая эффективность

* Атомарные операции (Fetch And Add, ...)

- + Эффективная поддержка со стороны компиляторов (gcc/clang, xlc, msvc)
- Изоляция одного значения, более сложные алгоритмы

* Spin lock

- + Полный контроль алгоритма
- Эффективна изоляция только небольшого куска кода.

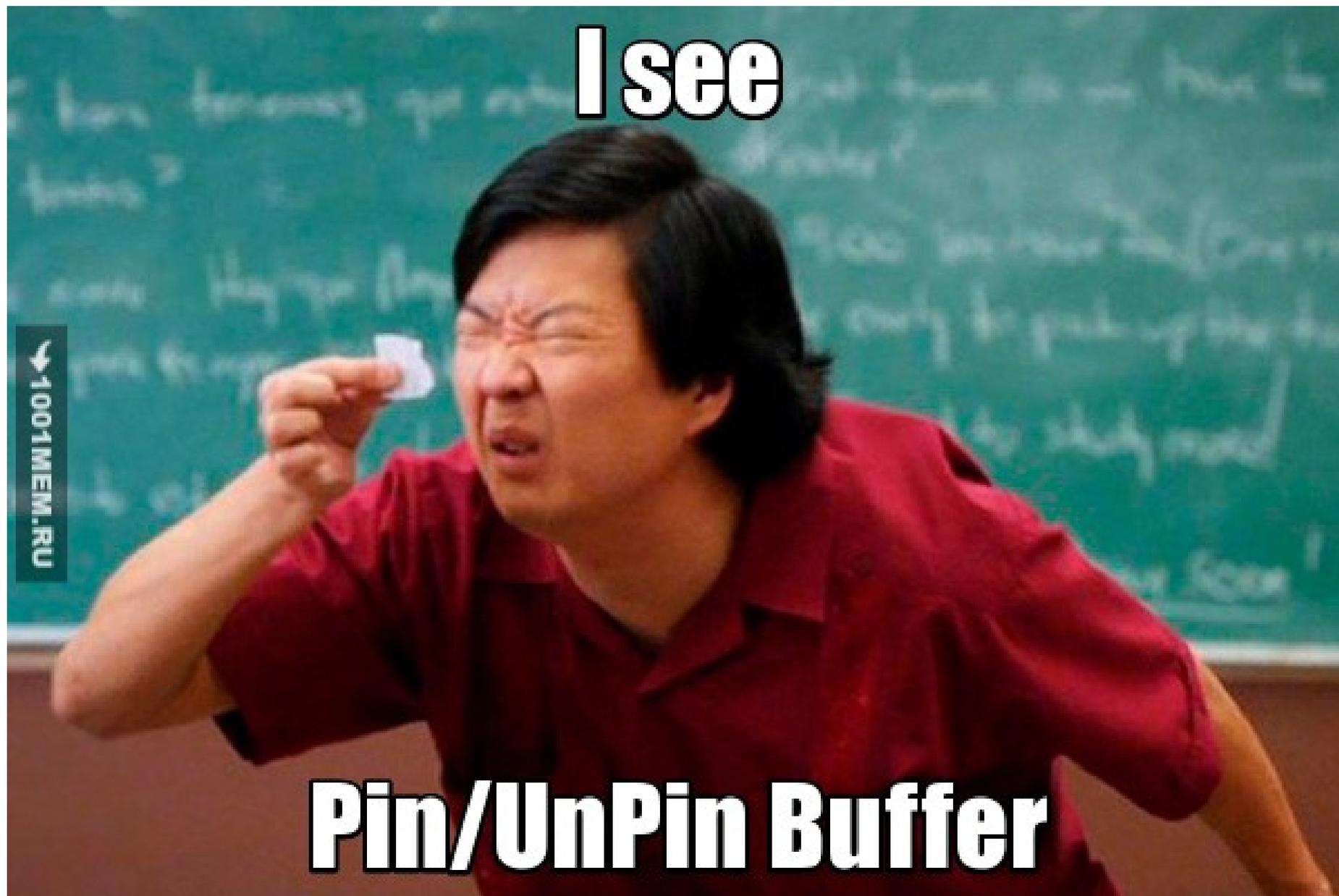
perf top

33.48%	postgres	[.] s_lock
2.51%	postgres	[.] GetSnapshotData
1.82%	postgres	[.] PinBuffer

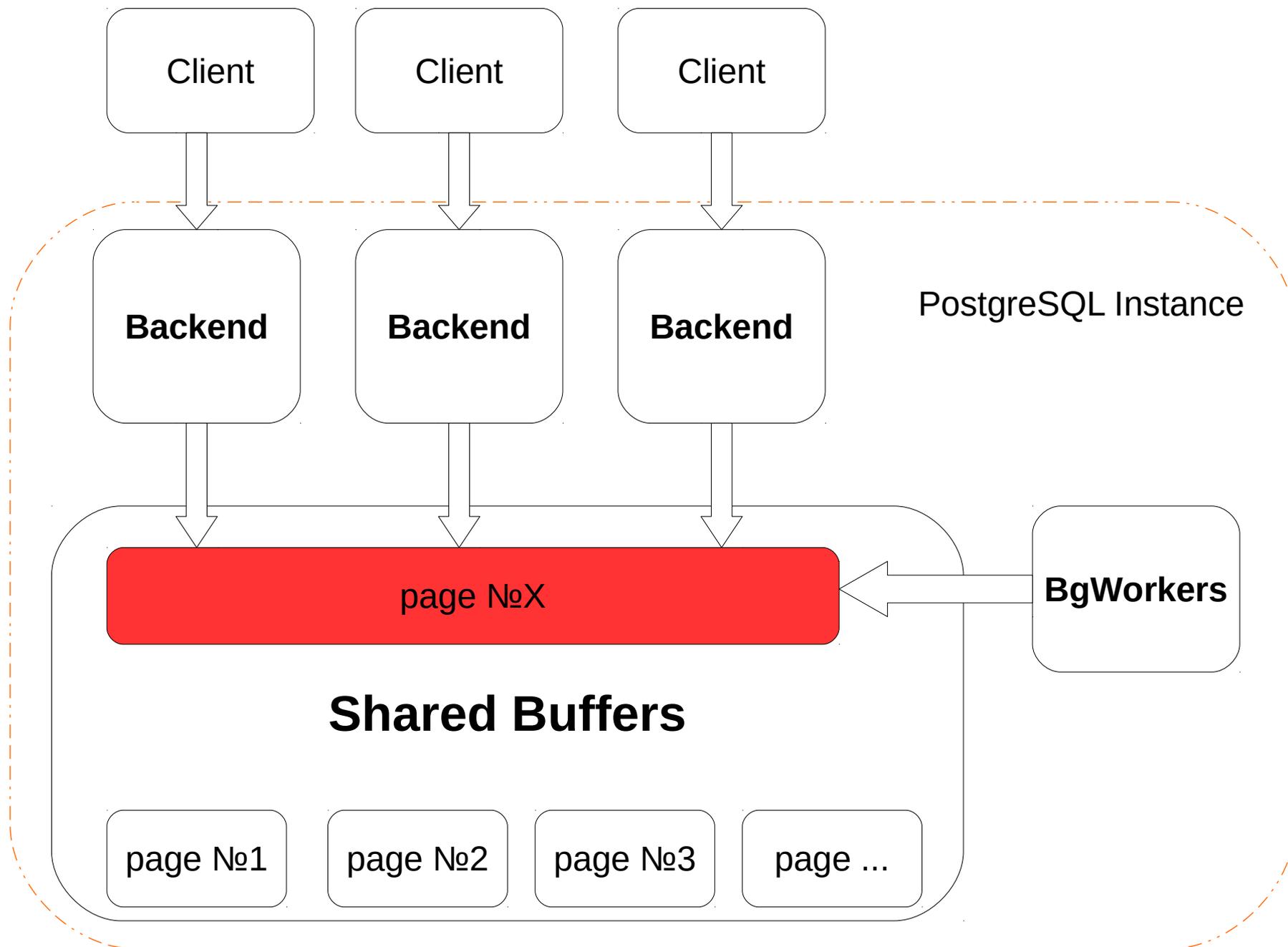
gdb

```
#0 0x00003fffac40a858 in __newselect_nocancel () from /lib64/power8/libc.so.6
#1 0x00000000106105f0 in pg_usleep (microsec=<optimized out>) at pgsleep.c:53
#2 0x00000000103e5f18 in s_lock (lock=0x3fe607980be0, file=0x10718398 "bufmgr.c",
line=<optimized out>) at s_lock.c:110
#3 0x00000000103aea10 in UnpinBuffer (buf=0x3fe607980bc0, fixOwner=1 '\001') at
bufmgr.c:1540
#4 0x00000000103b4910 in ReleaseAndReadBuffer (buffer=<optimized out>,
relation=0x3fe6067073e0, blockNum=<optimized out>) at bufmgr.c:1401
```

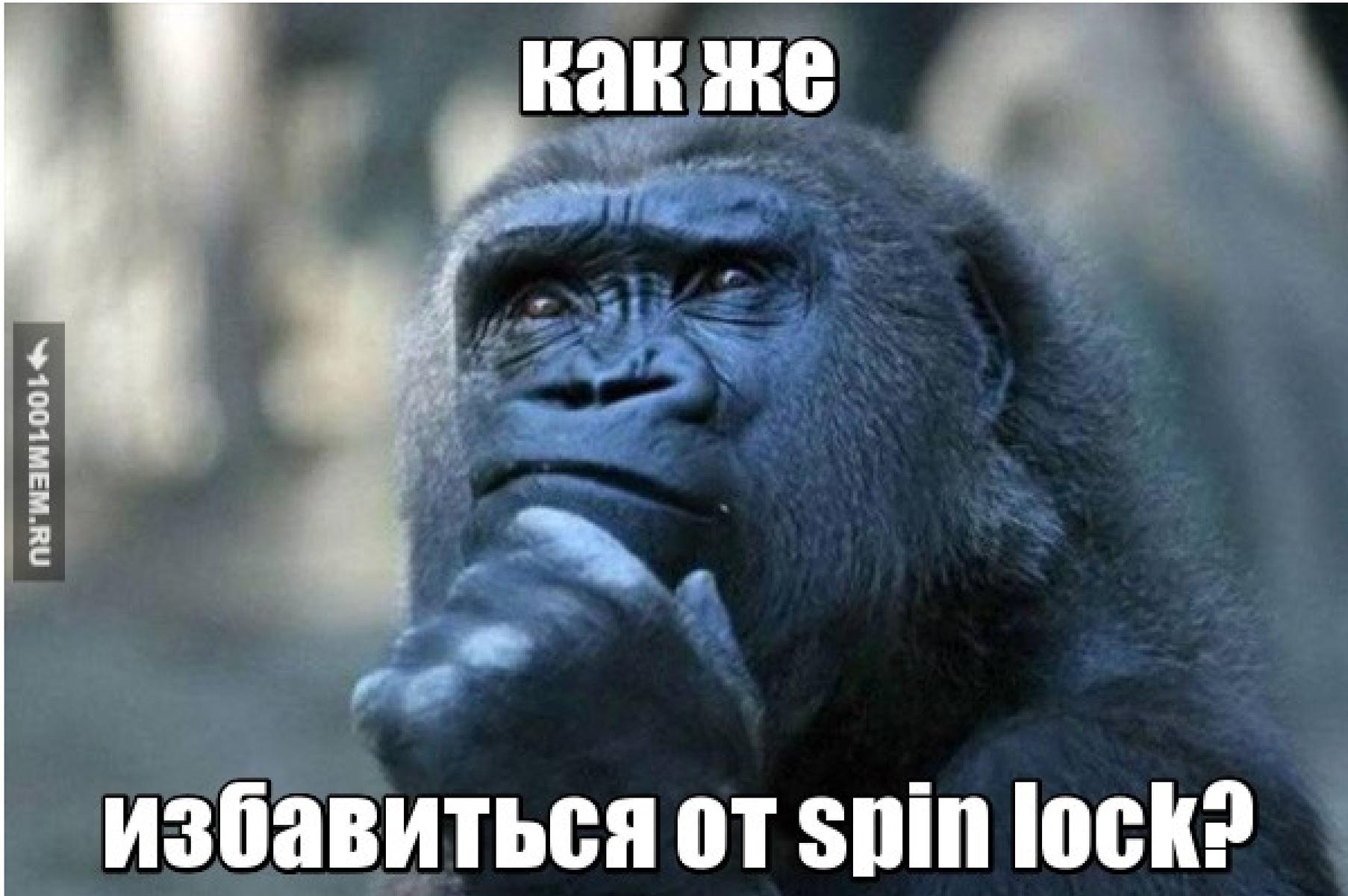
Что мы заметили (1.82% в perf)



Еще один взгляд на архитектуру



s_lock в Pin/UnPinBuffer



Было:

perf top

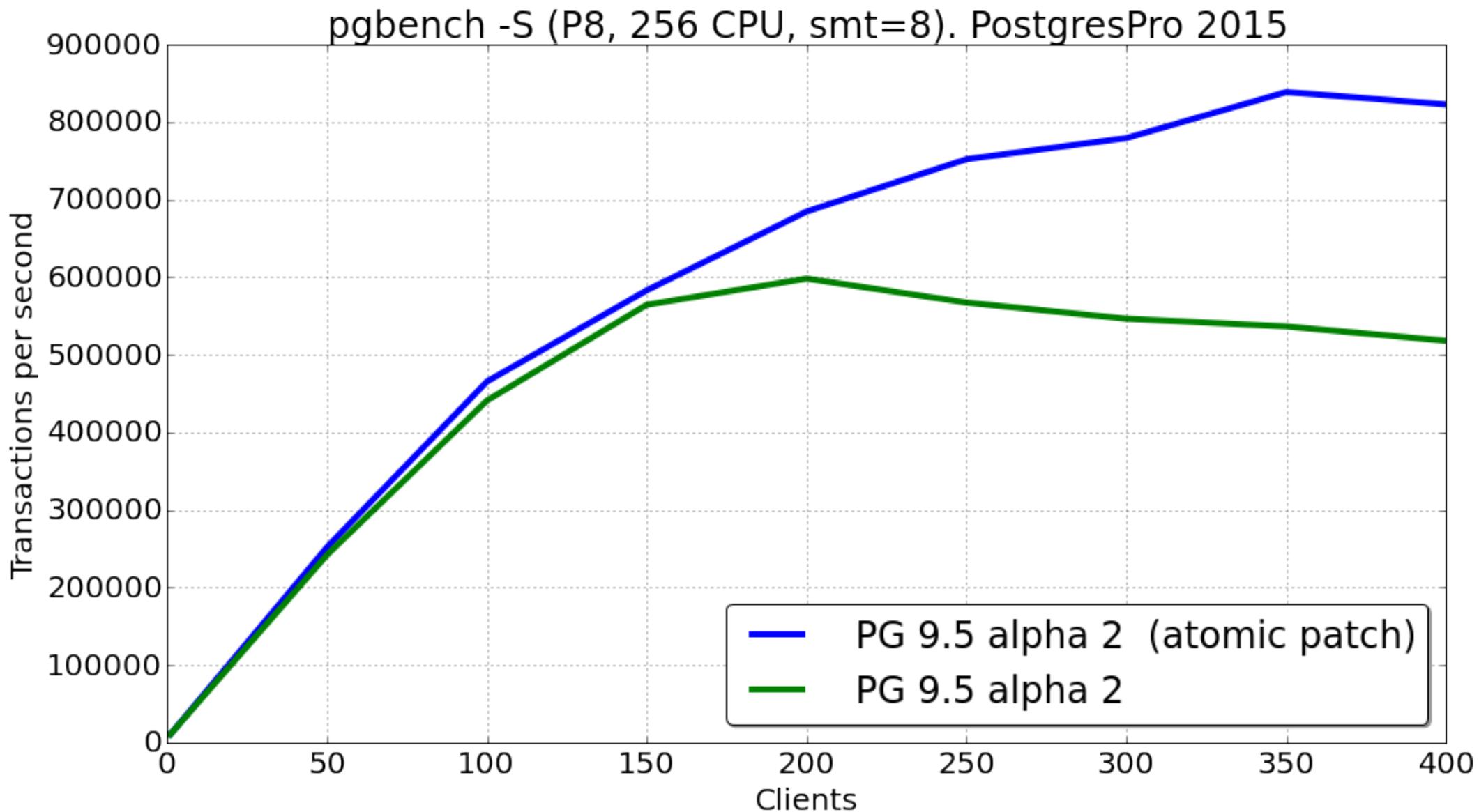
33.48%	postgres	[.] s_lock
2.51%	postgres	[.] GetSnapshotData
1.82%	postgres	[.] PinBuffer

Стало:

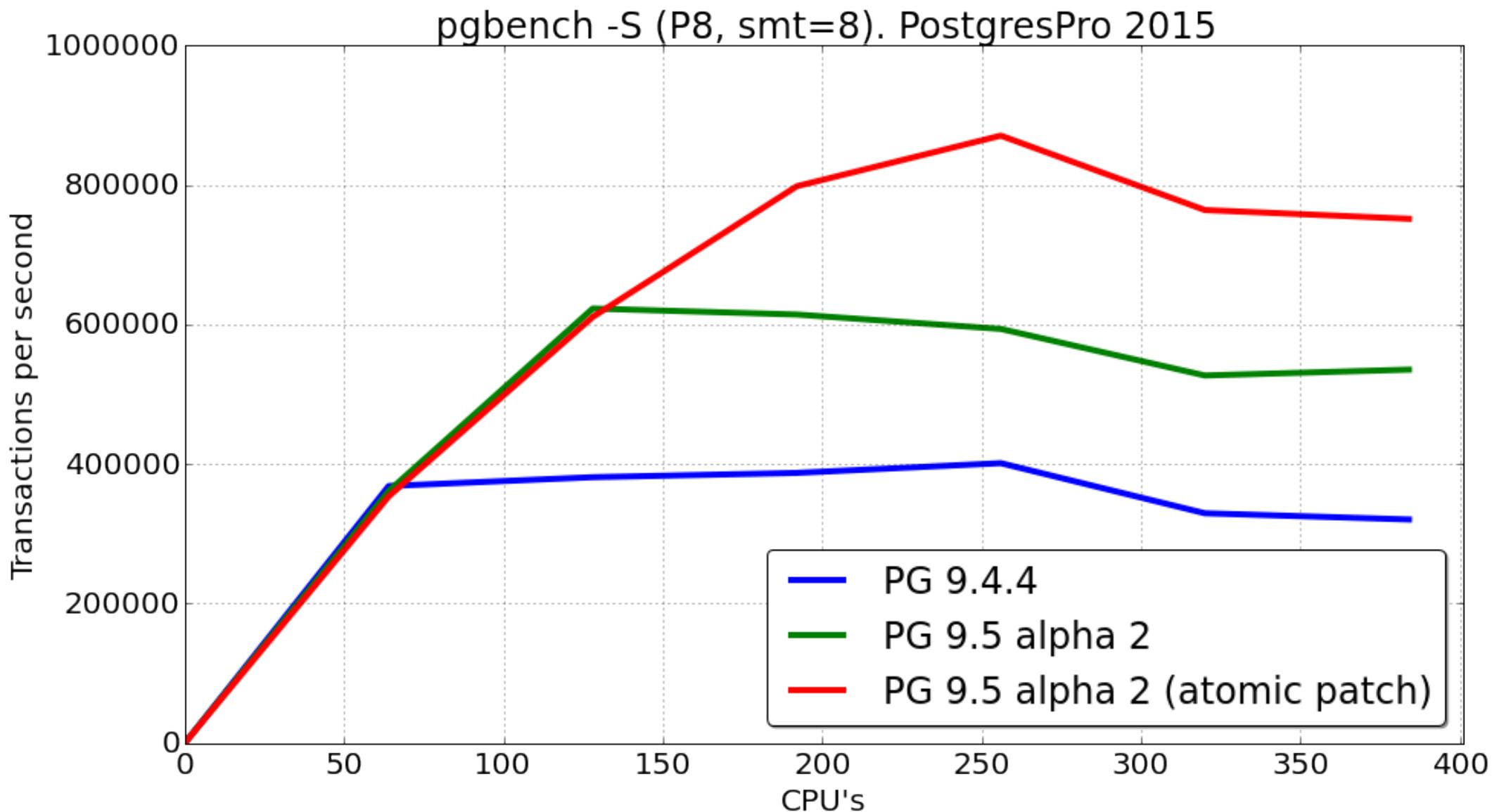
perf top

13.75%	postgres	[.] GetSnapshotData
4.88%	postgres	[.] AllocSetAlloc
2.47%	postgres	[.] LWLockAcquire

Производительность патча



Масштабирование с патчем



Эффективная утилизация ресурсов

Количество одновременно работающих
КЛИЕНТОВ

Сделать новый Buffer Manager*

* оценочно речь идет об увеличении производительности до 2 раз (!!!)

Улучшение алгоритма изоляции транзакций*

* оценочно до +10% производительности

Оптимизация Buffer Manager для NUMA-архитектуры * **

* оценочно до +10-15% производительности

** очень тяжелая задача

Спасибо за внимание!

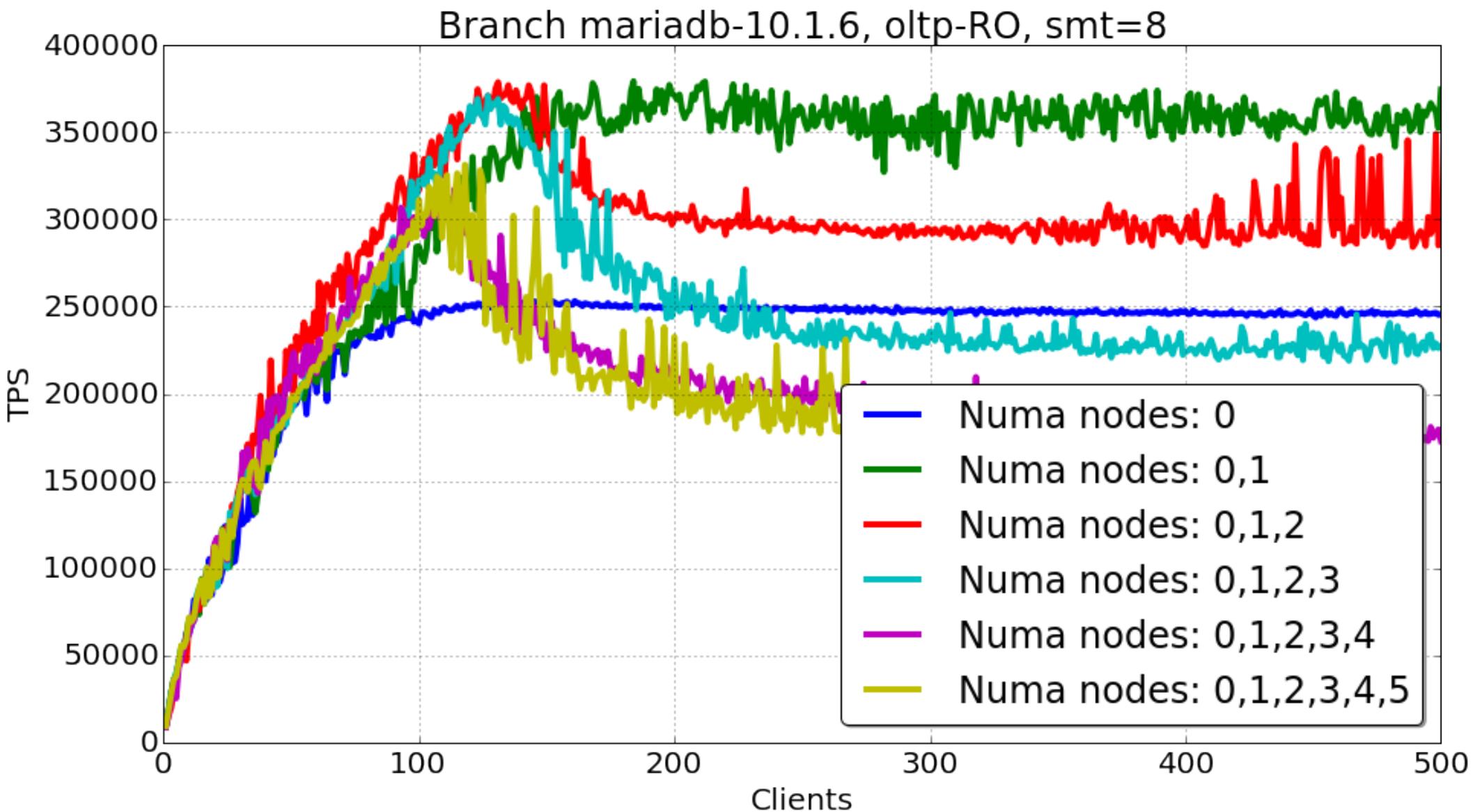


Дмитрий Васильев

Postgres Professional

<http://www.postgrespro.ru/>
info@postgrespro.ru

Секретные слайды



Её под Linux PPC не достать :)