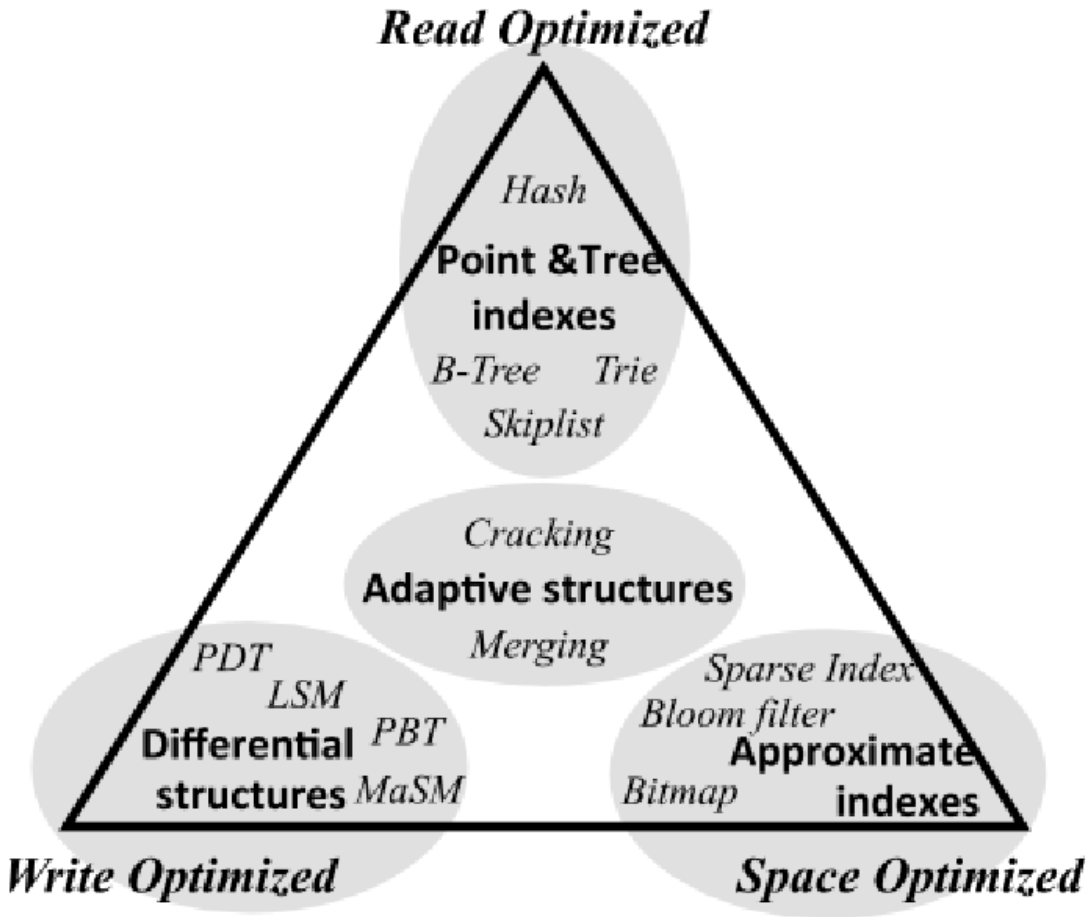
PostgreSQL Глазами обычного программиста

Eugene Klimov aka Slach

Обычный программист

По каким граблям ходят разработчики всех СУБД ;) RUM теорема

R (read) U (update) M (memory) overheads



Под какой тип нагрузки оптимизирован PostgreSQL?

Изначально для OLTP - т.е. «часто» пишем и «часто читаем», «малыми порциями» данных

https://ru.wikipedia.org/wiki/OLTP

Forks ;)

OLAP + MPP («тяжелые» выборки с GROUP BY, ORDER BY и JOIN) + MPP

https://www.citusdata.com - коммерческие инсталяции

https://www.postgres-xl.org (https://github.com/syndbg/docker-postgres-xl-debian)

http://www.timescale.com time-series data

http://www.agensgraph.com graph database

https://www.pipelinedb.com streaming sql

Быстрая batch вставка через COPY
https://postgrespro.ru/docs/postgrespro/10/sql-copy

https://github.com/ossc-db/pg_bulkload

Вопросы залу

Чем thread отличается от process?

Чем LEFT JOIN отличается от INNER JOIN?

Чем sharding отличается от partioning?

Как PostgreSQL «хранит» данные?

- Данные разбиты на «страницы» 8Kb (можно менять при компиляции)
- Внутри страниц «строки» (кортежи) целиком
- Страницы с данными кортежей «не упорядочены» (запись быстрее)
- Отдельные «страницы» с «вторичными индексами»
- Отдельные TOAST таблицы для «больших полей» (BLOB, JSONB, TEXT любые типы с TOAST > 2kb)
- Карты свободного пространства *_fsm файлы

Подробнее https://postgrespro.ru/docs/postgrespro/10/storage

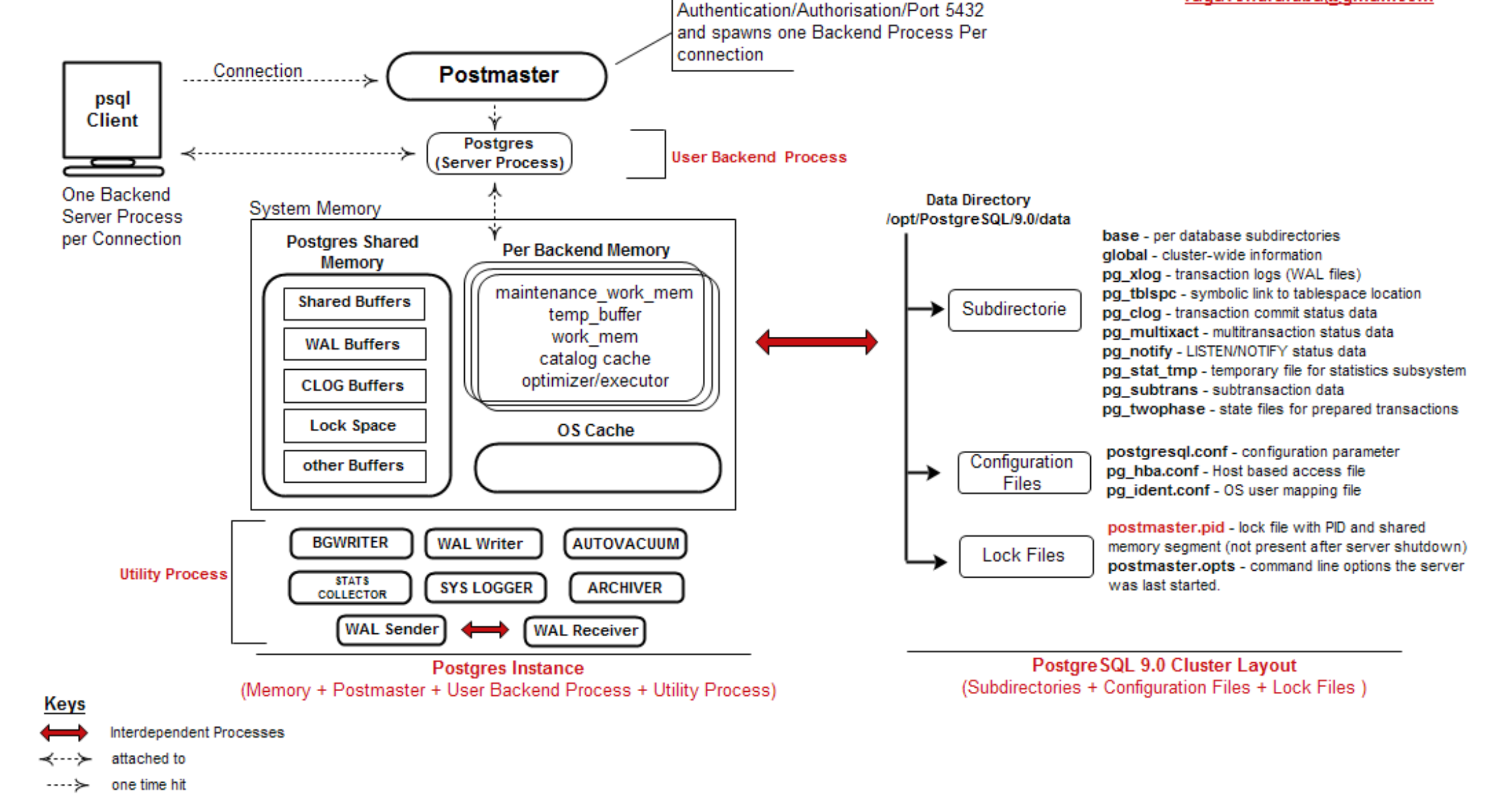
https://postgrespro.ru/docs/postgrespro/10/sql-cluster

Как PostgreSQL «пишет» данные?

- Любое изменение сначала делаются в памяти и потом максимально быстро старается записаться на диск
- INSERT пишется в WAL buffer -> затем fsync на WAL log (на каждый коммит) + создает новый tuple в shared buffers (помечает страницы как dirty)
- DELETE читает страницы с tuple в shared buffers, ставит tx_max=txid, помечает как dirty
- UPDATE = DELETE + INSERT
- Shared buffers – обходятся отдельными процессами bgwriter и checkpointer и изменения фиксируются на диск
- Отдельно для высвобождения «свободных» страниц, используется autovacuum процесс

Как PostgreSQL «читает» данные?

- Из текста запроса SQL определяются индексы и таблицы из которых будет произведено чтение
- Чтение идет с диска «страницами» (8kb) и кешируется сначала в файловой системе затем в shared buffers (если страница уже в памяти, то нет смысла читать ее второй раз)
- Чтение из secondary индекса «по значению» = O(log n) операций чтения где n общее кол-во «страниц» (страницы индексов тоже кешируются в shared buffer)
- Для «страниц данных» даже если вам нужно 1 поле из 100, прочитается вся строка



http://www.interdb.jp/pg/ - отличное описание внутренностей PostgreSQL

Как PostgreSQL обрабатывает много запросов «одновременно»?

- Postmaster процесс делает fork отдельного Backend Process на каждый коннект, чтобы не плодить fork, можно оптимизировать подключения через pgBouncer (mode=transaction) и через задание max_connections
- Один процесс обрабатывает одновременно 1 запрос
- Все процессы взаимодействуют с shared buffers через buffer manager
- Другие фоновые процессы (vacuum, checkpoint) порождаются postmaster по мере необходимости

Как «мониторить» PostgreSQL?

- https://github.com/postgrespro/mamonsu
- https://github.com/cybertec-postgresql/pgwatch2
- http://okmeter.io

DISASTER триггеры
- репликация running status, replication lag
- 90% connection usage
- High idle_in_transaction connections rate
- low shared buffers cache hit rate
- deadlocks
- high CPU\Memory usage
- high iowait, high disk utilization
- **pg_stat_statements**
- **pg_vacuum_progress**

Как «отлаживать» PostgreSQL «на живую»?

- SELECT * FROM pg_stat_activity;
- SELECT * FROM pg_stat_statements;
- SELECT * FROM pg_vacuum_progess;
- SELECT * FROM pg_stat_user_tables;

https://github.com/dataegret/pg-utils

https://github.com/NikolayS/postgres_dba

http://pginsight.io/

https://github.com/joyent/pgsqlstat (надо собирать с DTrace)

Как обновлять «схему данных» в PostgreSQL?

Тяжело

- ADD COLUMN xxx DEFAULT Null
- CREATE INDEX CONCURENTLY
- ALTER FUNCTION … боль ;(https://github.com/trustly/fdiff

http://quantumdb.io – Java mirror таблицы + триггеры и переписывание запросов в JDBC драйвере

Можно попробовать нагородить что то поверх

https://github.com/yandex/pgmigrate + https://github.com/ankane/pgsync

или

pglogical + https://github.com/enova/pgl_ddl_deploy

Как обновлять версию PostgreSQL

- Downgrade – очень больно, оно вам точно нужно? ;)
- pg_upgrade с простоем для 8.x, pg_upgradecluster для Debian
- Основной алгоритм «без простоя» для 9.4+ – логическая репликация + pglogical + failover proxy (pgbouncer) + pgrepup + pgl_ddl_deploy
- pglogical Не работает если нет primary или unique index, pglogical не реплицирует DDL
- https://hunleyd.github.io/posts/Upgrading-PostgreSQL-from-9.4-to-10.3-with-pglogical/
- https://github.com/Slach/pgrepup
- https://github.com/enova/pgl_ddl_deploy

DEMO stand (пока не работает)

- https://bitbucket.org/production_databases/pgsql_live_upgrade

Как делать backup в PostgreSQL?

pg_dump (базы до 10-20Gb)

- https://github.com/moorereason/autopgsqlbackup
- https://github.com/pgexperts/pgbackup

pg_basebackup + wal archiver

- https://github.com/wal-g/wal-g (пока только S3)
- https://github.com/ohmu/pghoard (не умеет incremental backup)
- https://github.com/wal-e/wal-e (заброшен, не умеет со slave)
- https://pgbackrest.org (Perl ;)

Standalone

- https://www.pgbarman.org/
- https://github.com/2ndquadrant-it/barman/issues/21 (Opensource это Люди ;)

Как делать «отказоустойчивость» в MySQL?

Master-slave

- https://repmgr.org/ (ставьте 4ю версию)
- https://github.com/zalando/patroni (нужен pgbouncer)
- https://github.com/sorintlab/stolon (встроенный proxy)
- https://github.com/nanopack/yoke (нужен pgbouncer)

Master-Master

- haproxy + pglogical + генерируемые ключи
- Postgres-XL, CitusData

Как делать «sharding» в MySQL?

Лучше всего делать руками в приложении (особенно решардинг)

- https://www.citusdata.com/blog/2018/01/10/sharding-in-plain-english/ (Primary KEY нужно генерировать самим)

- https://github.com/postgrespro/pg_shardman (нет resharding)

- https://plproxy.github.io/ - pgPL/SQL удаленный вызов функций

Вопросы из зала?

Со мной можно связаться
bloodjazman@gmail.com
https://t.me/bloodjazman