Дмитрий Кремер
МИА “Россия сегодня”

PGDAY'17 RUSSIA
КОНФЕРЕНЦИЯ ПО БАЗАМ ДАННЫХ

Модели разделяемой памяти в PostgreSQL

РОССИЯ СЕГОДНЯ

Содержание:

- Процессы в PostgreSQL
- IPC – межпроцессное взаимодействие
- Распределение структур данных между локальной и разделяемой памятью
- Устройство Buffer cache
- Механизмы (API's) разделяемой памяти в *NIX-системах
- Классическая модель System V
- Поиски разделяемой памяти
- Вызов mmap()
- Динамическая разделяемая память
- Инструменты для работы с памятью

Процессы в PostgreSQL

Часть вывода команды ps auxf | grep ^postgres

```
/opt/postgresql-9.4.12/bin/postgres -D /data/pgsql/9.4
 \_ postgres: logger process
 \_ postgres: checkpointer process
 \_ postgres: writer process
 \_ postgres: wal writer process
 \_ postgres: autovacuum launcher process
 \_ postgres: archiver process   last was 0000000100000E5400000049
 \_ postgres: stats collector process
 \_ postgres: wal sender process replica 192.168.12.87(58523) streaming E54/4A5B2E78
```

Процессы в PostgreSQL - 2

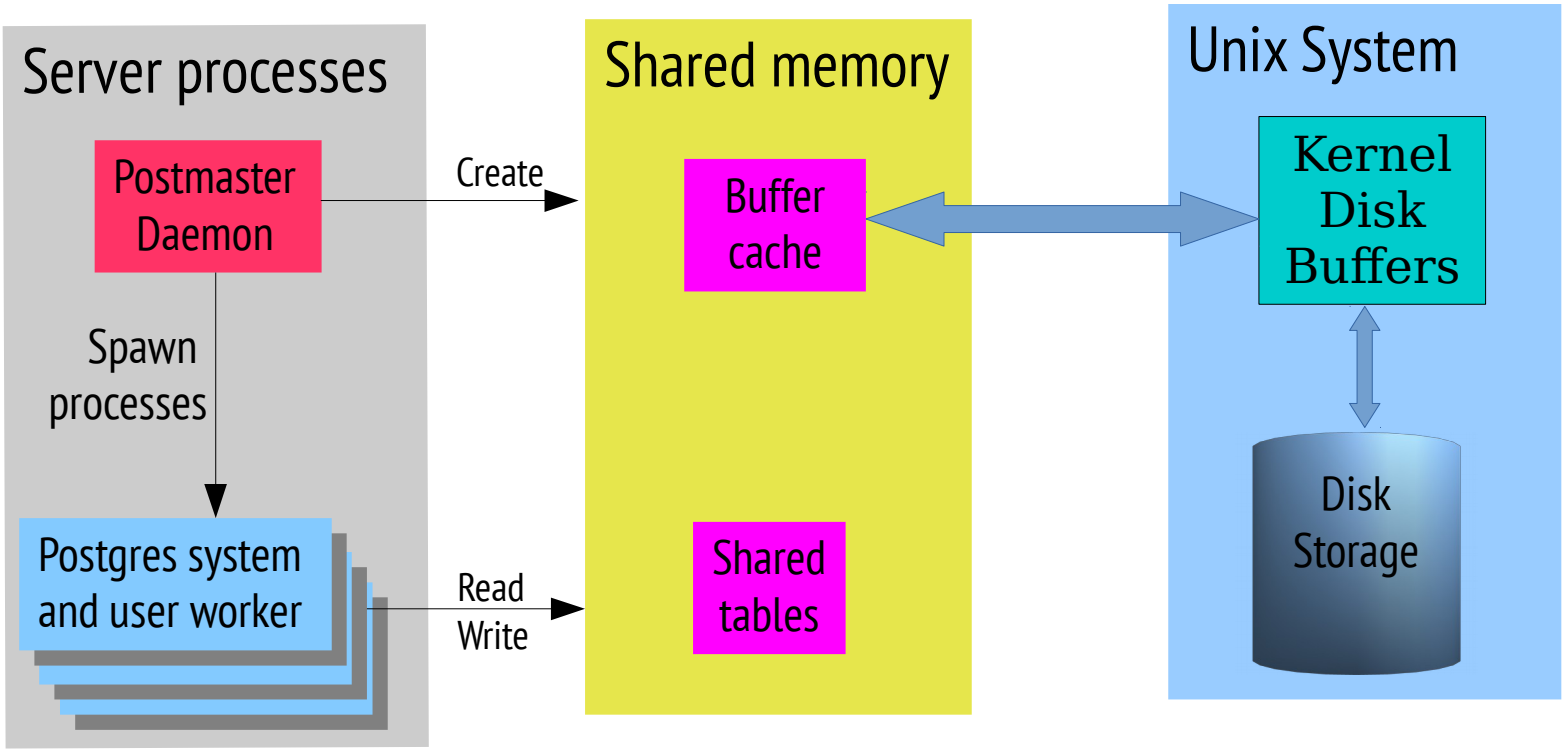
Основной процесс PostgreSQL – postmaster.

Его основные функции:

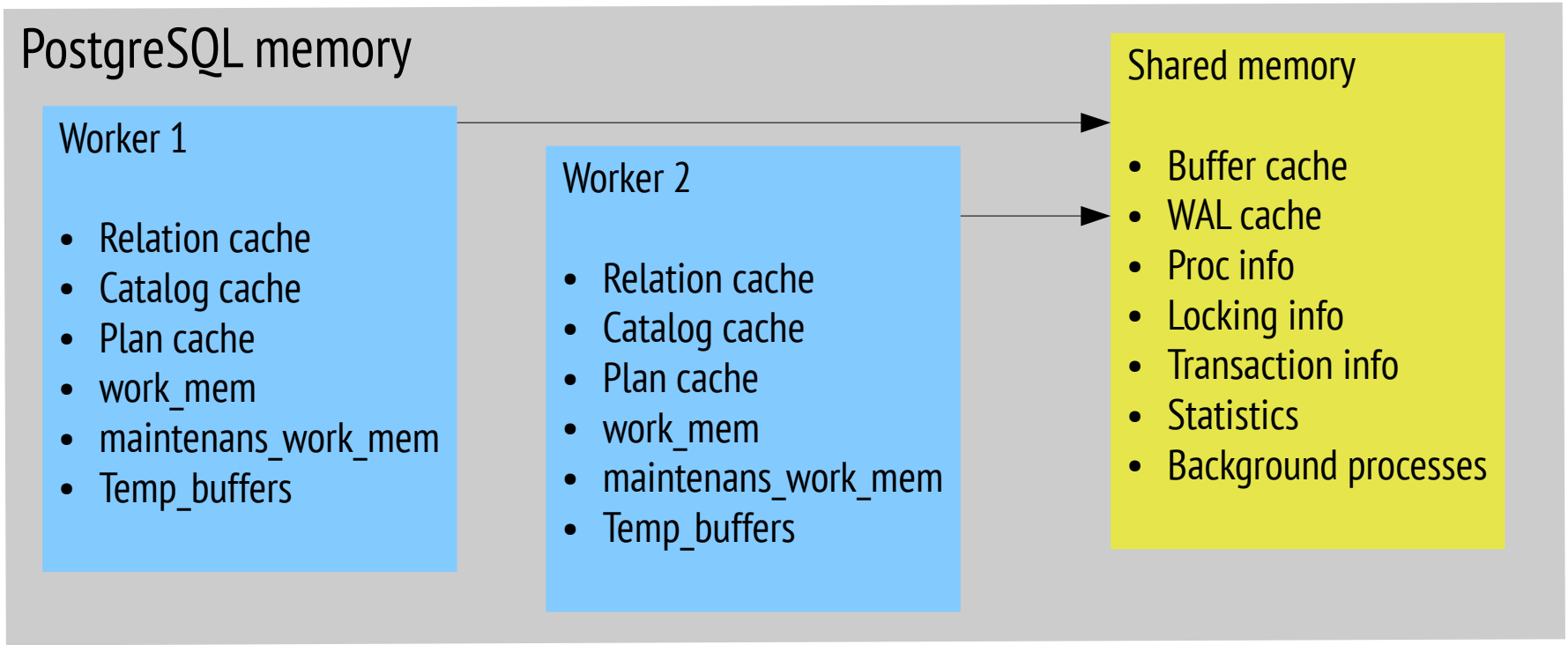
- Порождение (fork) служебных и пользовательских процессов (workers)
- Приём новых соединений
- Управление межпроцессным взаимодействием
- **Выделение и управление общими структурами памяти**

Т.е. вся разделяемая память PostgreSQL-кластера ассоциирована с процессом postmaster

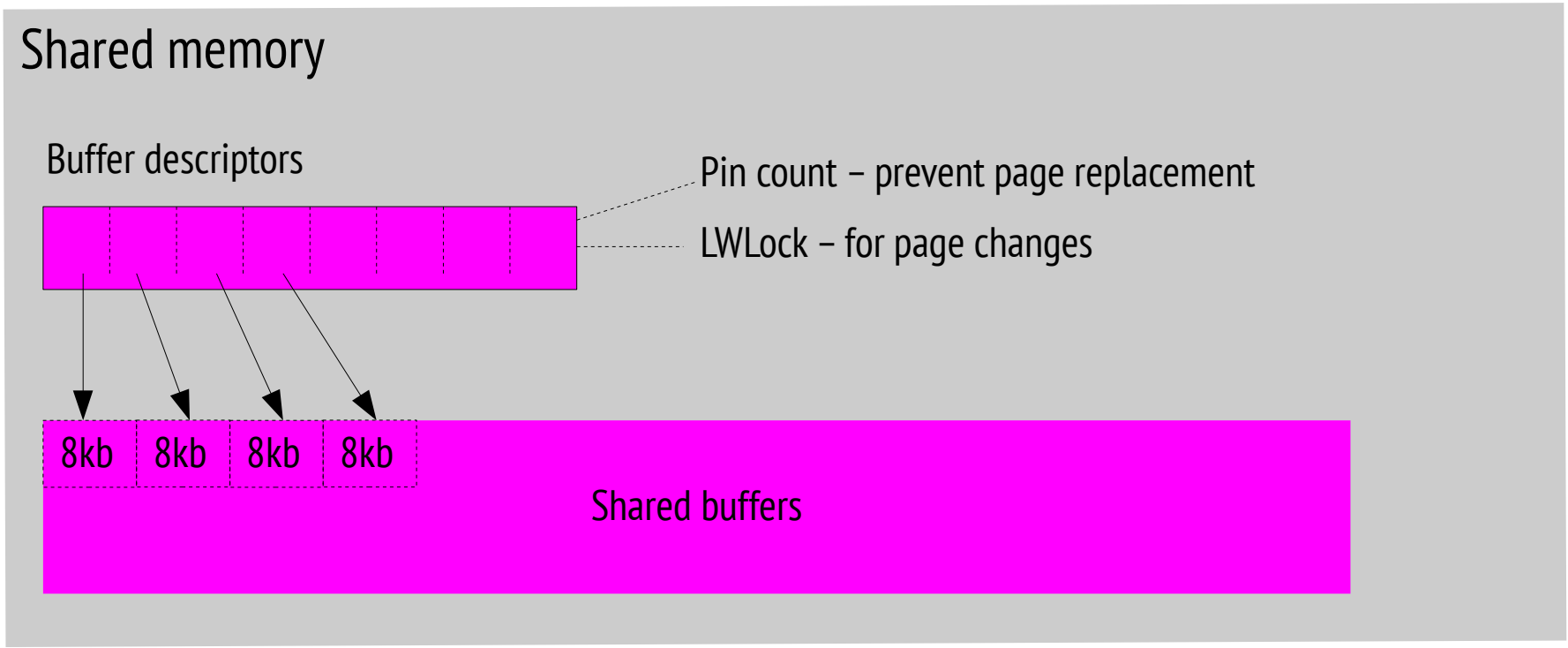
IPC – межпроцессное взаимодействие



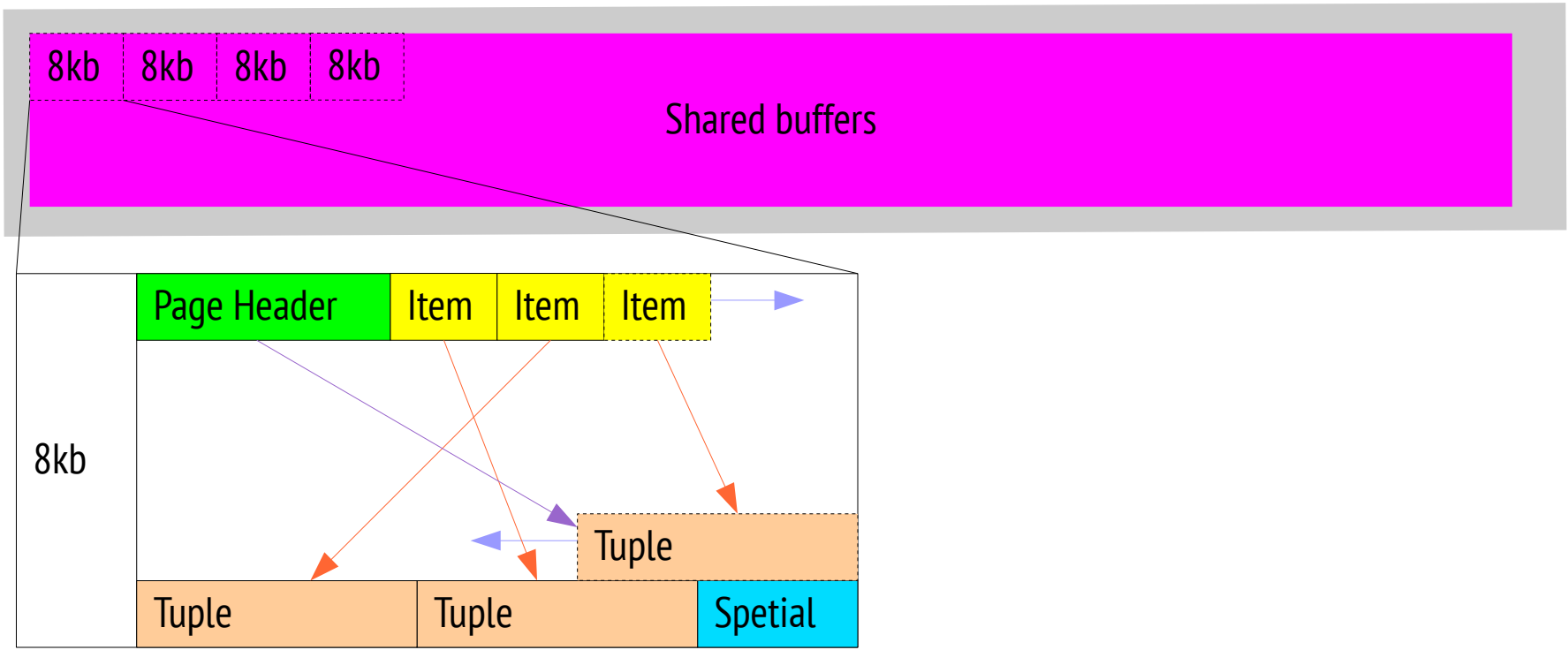
Распределение структур данных между локальной и разделяемой памятью



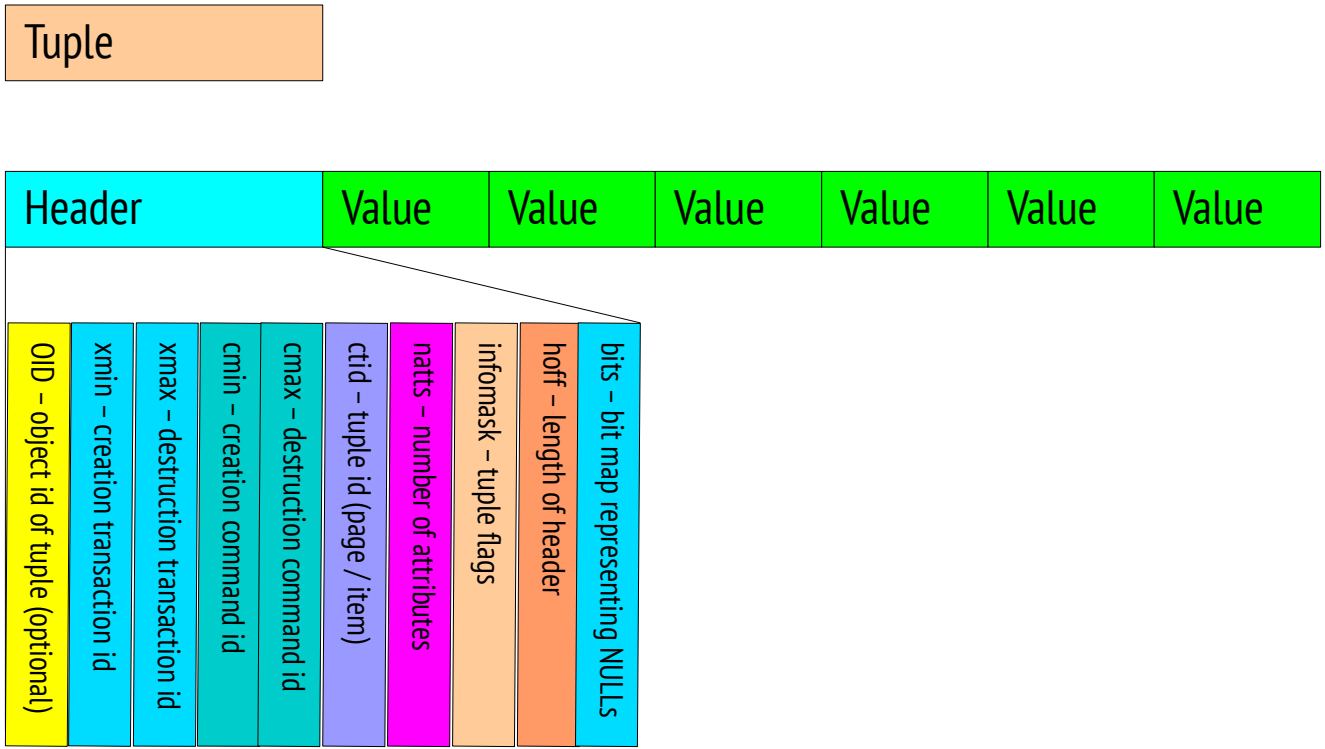
Устройство Buffer cache



Устройство Buffer cache - 2



Устройство Buffer cache - 3



Настройка shared buffers в postgresql.conf

shares_buffers = between 25% and 40% RAM if RAM > 1Gb

ATTENTION!!! Larger settings for shared_buffers usually require a corresponding increase in **max_wal_size**

huge_pages = try / on /off

Настройка Huge pages в Linux

Указываем нужное нам значение shared_buffers и стартуем кластер с выключенными huge pages

Проверяем:

```
$ head -1 $PGDATA/postmaster.pid
24267
$ grep ^VmPeak /proc/24267/status
VmPeak:  17556252 kB
$ grep ^Hugepagesize /proc/meminfo
Hugepagesize:       2048 kB
$ echo $((17556252/2048))
8572
```

Устанавливаем параметры

```
$ sysctl -w vm.nr_hugepages=8572
$ sysctl -w hugetlb_shm_group = 498
```

◄—— группа, которой принадлежит системный пользователь postgres

Проверяем:

```
$ grep Huge /proc/meminfo
```

Механизмы (API's) разделяемой памяти в *NIX-системах

System V

- Классический механизм управления разделяемой памятью. Поддерживается большинством *NIX систем
- Память делится между **несвязными** процессами.

Shared mapping - mmap(2)

- Shared **anonymous** mapping. Память делится между **связными** процессами. (Через fork())
- Shared **file** mapping. Память делится между **несвязными** процессами.

POSIX shared memory

- Память делится между **несвязными** процессами без нагрузки лишней нагрузки на файловую систему и ввод/вывод.
- Полагается полее простым и эффективным, чем старые API

Классическая модель System V – параметры ядра

Параметр	Описание	Рекомендуемые значения
SHMMAX	Максимальный размер сегмента разделяемой памяти (в байтах)	не меньше 1 КБ (больше, если запускается много копий сервера)
SHMMIN	Минимальный размер сегмента разделяемой памяти (в байтах)	1
SHMALL	Общий объём доступной разделяемой памяти (в байтах или страницах)	если в байтах, то же, что и SHMMAX; если в страницах, то ceil(SHMMAX/PAGE_SIZE)
SHMSEG	Максимальное число сегментов разделяемой памяти для процесса	требуется только 1 сегмент, но значение по умолчанию гораздо больше
SHMMNI	Максимальное число сегментов разделяемой памяти для всей системы	как SHMSEG плюс потребность других приложений
SEMMNI	Максимальное число идентификаторов семафоров (т. е., их наборов)	как минимум ceil((max_connections + autovacuum_max_workers + max_worker_processes + 5) / 16)
SEMMNS	Максимальное число семафоров для всей системы	ceil((max_connections + autovacuum_max_workers + max_worker_processes + 5) / 16) * 17 плюс потребность других приложений
SEMMSL	Максимальное число семафоров в наборе	не меньше 17
SEMMAP	Число записей в карте семафоров	
SEMVMX	Максимальное значение семафора	не меньше 1000 (по умолчанию оно обычно равно 32767; без необходимости менять его не следует)

Классическая модель System V - 2

Пример скрипта по настройке параметров разделяемой памяти

```
#!/bin/bash
# simple shmsetup script
page_size=$(getconf PAGE_SIZE)
phys_pages=$(getconf _PHYS_PAGES)
shmall=$(expr $phys_pages / 3)
shmmax=$(expr $shmall \* $page_size)
echo kernel.shmmax = $shmmax
echo kernel.shmall = $shmall
```

```
$ ./shmsetup >> /etc/sysctl.conf
$ sysctl -p
```

Классическая модель System V - 3

IPC, управление межпроцессорным взаимодействием

Утилиты ipcs и ipcrm используются для просмотра и освобождения разделяемых ресурсов

Семафоры:

$ ipcs -l

------ Semaphore Limits --------
max number of arrays = 6000 // SEMMNI
max semaphores per array = 2500 // SEMMSL
max semaphores system wide = 32000 // SEMMNS
max ops per semop call = 32 // SEMOPM
semaphore max value = 32767 //SEMVMX

Желательно установить сообразно количеству подключений

$ sysctl kernel.sem = 2500 32000 32 6000

------ Shared Memory Limits --------
max number of segments = 4096 // SHMMNI
max seg size (kbytes) = 16777216 // SHMMAX
max total shared memory (kbytes) = 16777216 // SHMALL
min seg size (bytes) = 1 // SHMMIN

------ Messages: Limits --------
max queues system wide = 32768 // MSGMNI
max size of message (bytes) = 65536 // MSGMAX
default max size of queue (bytes) = 65536 // MSGMNB

Классическая модель System V и современный PostgreSQL (9.3+)

```
$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x6c0145c0 6488064    zabbix     600        219056     7
0x0052daf1 7602177    postgres   600        56         4
```

Выделено ВСЕГО 56 БАЙТ!

Начиная с версии 9.3, PostgreSQL практически НЕ ИСПОЛЬЗУЕТ System V

Читаем исходный код

src/backend/storage/ipc/shmem.c

The shared memory is created by a postmaster and is inherited by each backend via fork() (or, in some ports, via other OS-specific methods).

(a) There are three kinds of shared memory data structures available to POSTGRES: fixed-size structures, queues and hash tables.
(b) During initialization, each module looks for its shared data structures in a hash table called the "Shmem Index". If the data structure is not present, the caller can allocate a new one and initialize it. If the data structure is present, the caller "attaches" to the structure by initializing a pointer in the local address space.
(c) In standard Unix-ish environments, individual backends do not need to re-establish their local pointers into shared memory, because they inherit correct values of those variables via fork() from the postmaster.
(d) memory allocation model: shared memory can never be freed, once allocated.

Ищем разделяемую память

```
$ ps -u postgres o pid,cmd f | grep postgre | grep -v local

24267 /opt/postgresql-9.4.12/bin/postgres -D /data/pgsql/9.4
24269  \_ postgres: logger process
27271  \_ postgres: checkpointer process
27272  \_ postgres: writer process
27273  \_ postgres: wal writer process
27274  \_ postgres: autovacuum launcher process
27275  \_ postgres: archiver process    last was 0000000100000E670000000D
27276  \_ postgres: stats collector process
27302  \_ postgres: wal sender process replica 192.168.12.87(55843) streaming
E67/EDEBF58

$ pmap  24267
```

Ищем разделяемую память - 2

```
24267:   /opt/postgresql-9.4.12/bin/postgres -D /data/pgsql/9.4
0000000000400000   5484K r-x--  /opt/postgresql-9.4.12/bin/postgres
0000000000b5a000     52K rw---  /opt/postgresql-9.4.12/bin/postgres
0000000000b67000    336K rw---    [ anon ]
000000000141d000    404K rw---    [ anon ]
0000003401e00000    128K r-x--  /lib64/ld-2.12.so
.....................................................................................................................
00007fd1aa707000      4K r----  /lib64/libnss_files-2.12.so
00007fd1aa708000      4K rw---  /lib64/libnss_files-2.12.so
00007fd1aa712000     72K rw-s-  /dev/shm/PostgreSQL.1999361057
00007fd1aa724000     28K r-x--  /opt/postgresql-9.4.12/lib/pg_stat_statements.so
00007fd1aa72b000   2048K -----  /opt/postgresql-9.4.12/lib/pg_stat_statements.so
00007fd1aa92b000      4K rw---  /opt/postgresql-9.4.12/lib/pg_stat_statements.so
00007fd1aa92c000  96836K r----  /usr/lib/locale/locale-archive
00007fd1b07bd000     24K rw---    [ anon ]
00007fd1b07c3000   1316K r-x--  /usr/lib64/libxml2.so.2.7.6.#prelink#.HKkJRX (deleted)
00007fd1b090c000   2044K -----  /usr/lib64/libxml2.so.2.7.6.#prelink#.HKkJRX (deleted)
00007fd1b0b0b000     40K rw---  /usr/lib64/libxml2.so.2.7.6.#prelink#.HKkJRX (deleted)
00007fd1b0b15000      4K rw---    [ anon ]
00007fd1b0b1e000      4K rw-s-    [ shmid=0x118001 ]
00007fd1b0b1f000      4K rw---    [ anon ]
00007fd1b0c00000 17430528K rw-s-  /anon_hugepage (deleted)
00007ffd39ad8000     84K rw---    [ stack ]
00007ffd39b24000      4K r-x--    [ anon ]
ffffffffff600000      4K r-x--    [ anon ]
```

Механизмы (API's) разделяемой памяти в *NIX-системах

System V

- Классический механизм управления разделяемой памятью. Поддерживается большинством *NIX систем
- Память делится между **несвязными** процессами.

Shared mapping - mmap(2)

- Shared **anonymous** mapping. Память делится между **связными** процессами. (Через fork())
- Shared **file** mapping. Память делится между **несвязными** процессами.

POSIX shared memory

- Память делится между **несвязными** процессами без нагрузки лишней нагрузки на файловую систему и ввод/вывод.
- Полагается полее простым и эффективным, чем старые API

Читаем исходный код

src/backend/storage/port/sysv_shmem.c

As of PostgreSQL **9.3**, *we normally allocate* **only a very small amount of System V shared memory**, *and only for the purposes of providing an interlock to protect the data directory. The real shared memory block is allocated using* **mmap().** *This works around the problem that many systems have very low limits on the amount of System V shared memory that can be allocated. Even a limit of a few megabytes will be enough to run many copies of PostgreSQL without needing to adjust system settings.*

mmap() и анонимные сегменты разделяемой памяти не поддерживается pre-2.4 ядрами Linux

Где ещё посмотреть разделяемую память

```
$grep -B1 -E '^Size: *[0-9]{6}' /proc/24267/smaps

7fd1b0c00000-7fd5d8a00000 rw-s 00000000 00:0c 973975612                  /anon_hugepage
(deleted)
Size:          17430528 kB
```

Динамическая разделяемая память в PostgreSQL

Параметр *dynamic_shared_memory_type* в postgresql.conf может принимать следущие зачения:

- **posix** - управление разделяемой памятью по стандарту POSIX. Для выделения памяти используется системный вызов shm_open()
- **sysv** - управление разделяемой памятью по стандарту System V. Для выделения памяти используется системный вызов shmget()
- **windows** - управление разделяемой памятью в Windows,
- **mmap** - для выделения памяти используется системный вызов mmap(),
- **none** – разделяемая память не используется

Динамическая разделяемая память в PostgreSQL

shared_buffers ≠ dynamic_shared_memory

Параметр dynamic_shared_memory_type не влияет на способ выделения памяти для структур описанных выше.

Динамическая разделяемая память используется для параллельных воркеров. При dynamic_shared_memory_type = none параллелизм работать не будет.

Инструментарий для анализа памяти

Утилиты

- pmap – память процесса
- ipcs – объекты разделяемой памяти System V
- ps – список процессов

Файлы

- /proc/$postmasterPID/maps – расширенная информация из pmap
- /proc/$postmasterPID/smaps – карта памяти с детализацией по каждому диапазону адресов
- /proc/$postmasterPID/status

Список источников

The Linux Programming Interface (published in October 2010, No Starch Press, ISBN 978-1-59327-220-3)

IEEE Std 1003.1TM, 2016 Edition, Standard for Information Technology—®Portable Operating System Interface (POSIX)

https://momjian.us/main/writings/pgsql/inside_shmem.pdf

https://www.postgresql.org/files/developer/tour.pdf

https://www.postgresql.org/docs/current/static/runtime-config-resource.html

https://www.depesz.com/2012/06/09/how-much-ram-is-postgresql-using/

http://man7.org/training/download/posix_shm_slides.pdf

https://www.youtube.com/watch?v=R1U8ayT9PaY&

PGDAY'17 RUSSIA

Спасибо!
Вопросы?

dmitry.kremer@gmail.com
https://github.com/djester