2ndQuadrant

Professional PostgreSQL

PgFincore and the OS Page Cache

Cédric Villemain

<cedric@2ndQuadrant.fr>

http://www.2ndQuadrant.fr/

pgDay

12/07/10, Stuttgart

License

- Creative Commons:
  - Attribution-Non-Commercial-Share Alike 2.5
  - You are free:
    - to copy, distribute, display, and perform the work
    - to make derivative works
  - Under the following conditions:
    - Attribution. You must give the original author credit.
    - Non-Commercial. You may not use this work for commercial purposes.
    - Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one.

Why did I wrote PgFincore ?

Why did I wrote PgFincore ?

- Main Database is about the RAM size
- PostgreSQL share Server ressources

Why did I wrote PgFincore ?

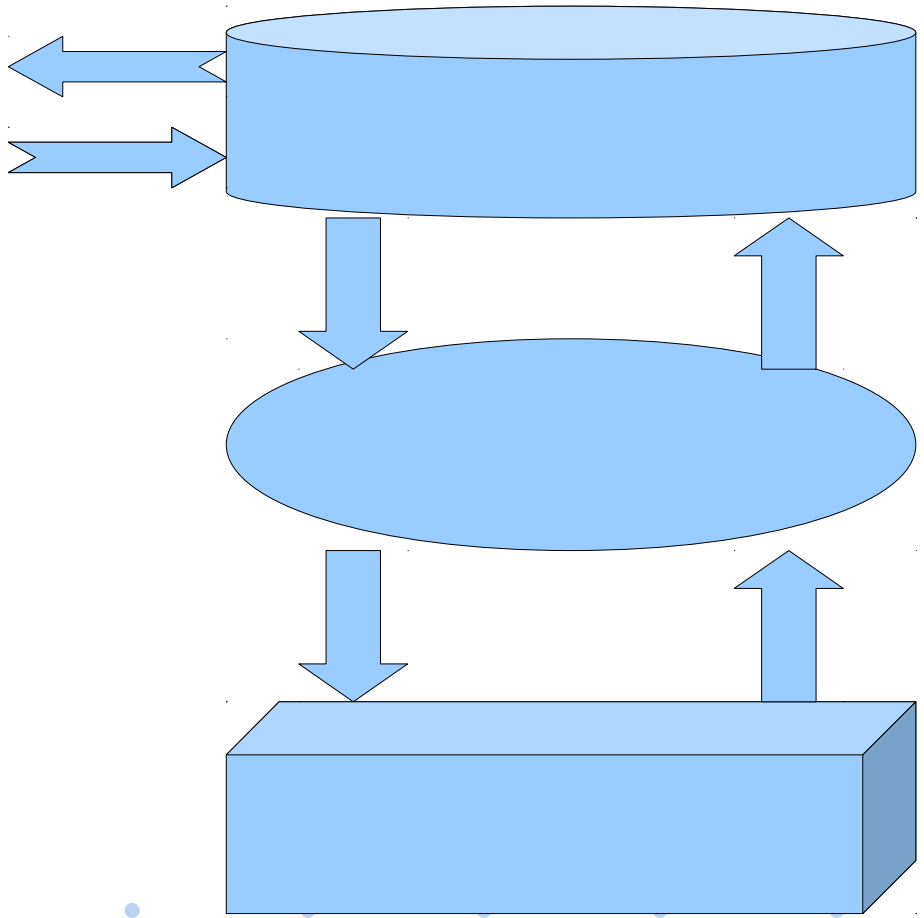
- Main Database is about the RAM size
- PostgreSQL share Server ressources
- Keep good TPS when Server reboot

Why did I wrote PgFincore ?

- Main Database is about the RAM size
- PostgreSQL share Server ressources
- Keep good TPS when Server reboot
- All about IO analysis

SELECT a, b, c FROM foo;

- PostgreSQL
- OS
- Hardware



PostgreSQL Buffer Cache

- Shared Memory

PostgreSQL Buffer Cache

- Shared Memory
- Simple LRU List

PostgreSQL Buffer Cache

- Shared Memory
- Simple LRU List
- Effective Cache Size
  - Mackert and Lohman approximation

The OS Page Cache

- Keep Logical Content

The OS Page Cache

- Keep Logical Content
  - 4kb
  - mmap + mincore

The OS Page Cache

- Keep Logical Content
- Complex LRU list

The OS Page Cache

- Keep Logical Content
- Complex LRU list
  - Double LRU list
  - Some piece of FIFO
  - Larger pin counter

Hardware Read Cache

- Physical Cache
- Got one Open-Source ?
- Useless, prefer Write Cache

PostgreSQL Current Features

- Hit/miss ratio
- Tablespace
- Prefetch Buffers
- Synchronous Seq Scan
- Buffers Ring Limit

PostgreSQL Current Issues

- Monitoring real disk activity
- Seq Scan
- OS Restart
- PostgreSQL Restart
- Switchover

Getting Stats – Restoring State

- Get stats per segment of table or index

Getting Stats – Restoring State

- Get stats per segment of table or index
- Restore the OS Page Cache state for a table or index

Tools to do the job

- mmap/mincore

Tools to do the job

- mmap/mincore
- posix_fadvise

Impacts and Limits

- More syscall

Impacts and Limits

- More syscall
- Memory mapping

Impacts and Limits

- More syscall
- Memory mapping
- posix_fadvise implementation
  - POSIX_FADV_NOREUSE ← it had been deactivated
  - ~~POSIX_FADV_WILLNEED ← does not work with already in core memory blocks (up to linux 2.6.??)~~

PgFincore Functions - DBA

- Debug
  - Set client_min_messages to DEBUG1; -- or DEBUG5

PgFincore Functions - DBA

- Debug
- pgsysconf()
  - Number of free pages
  - Page Size

PgFincore Functions - DBA

- Debug
- pgsysconf()
- pgmincore('table_foo')
- pgfadv_WILLNEED('table_foo')
- pgfadv_DONTNEED('table_foo')

PgFincore Functions - Usefull !

- pgmincore_snapshot('table_foo')
- pgfadv_WILLNEED_snapshot('table_foo')

PgFincore Functions - Useless ?

- pgfadv_NORMAL('table_foo')
- pgfadv_SEQUENTIAL('table_foo')
- pgfadv_RANDOM('table_foo')

Some Uses Cases : preload

```
cedric=# select * from pgfadv_WILLNEED('pgbench_accounts');
        relpath         | block_size | block_disk | block_free
------------------------+------------+------------+------------
 base/16385/168683      |       4096 |     262144 |       4195
 base/16385/168683.1    |       4096 |     262144 |       3918
 base/16385/168683.2    |       4096 |     262144 |       3885
 base/16385/168683.3    |       4096 |      66028 |       4166
(4 lignes)

Temps : 18395,987 ms
```

```
cedric=# select * from pgmincore('pgbench_accounts') ;
        relpath         | block_size | block_disk | block_mem | group_mem
------------------------+------------+------------+-----------+-----------
 base/16385/168683      |       4096 |     262144 |     31090 |     13667
 base/16385/168683.1    |       4096 |     262144 |     93131 |     11138
 base/16385/168683.2    |       4096 |     262144 |    126301 |      8425
 base/16385/168683.3    |       4096 |      66028 |     66023 |         6
(4 lignes)

Temps : 111,543 ms
```

Some Uses Cases : snapshot/restore

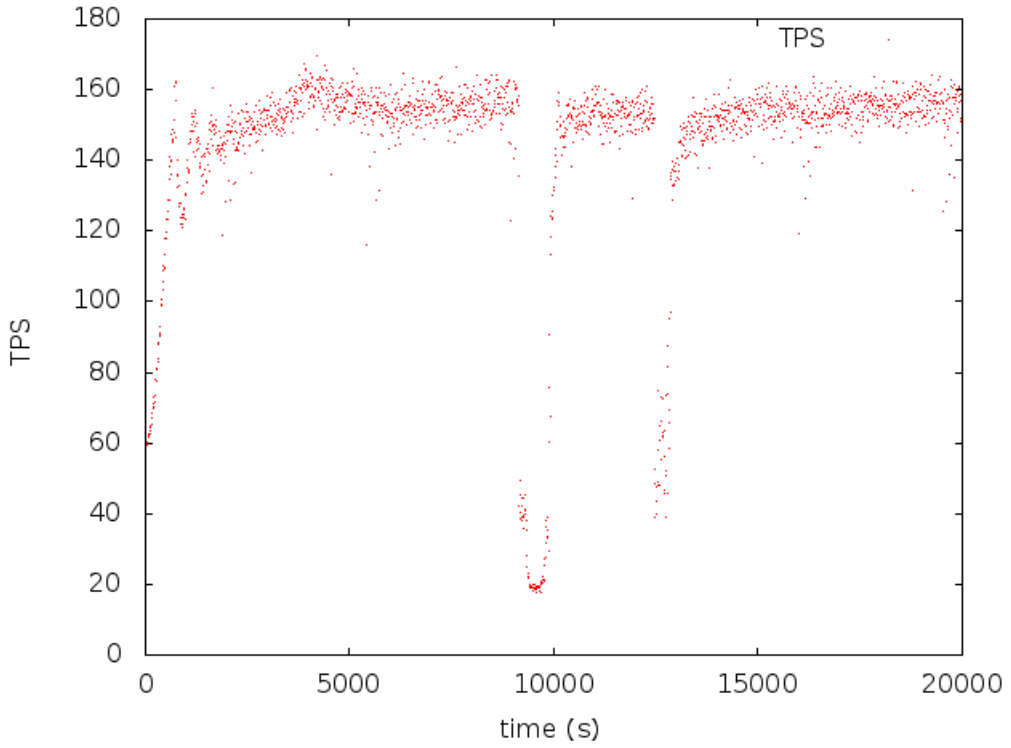
```
cedric=# select * from pgmincore_snapshot('pgbench_accounts');
           relpath           | block_size | block_disk | block_mem | group_mem
-----------------------------+------------+------------+-----------+-----------
 base/16385/168683_mincore   |       4096 |     262144 |     91453 |     16820
 base/16385/168683.1_mincore |       4096 |     262144 |     89426 |     16952
 base/16385/168683.2_mincore |       4096 |     262144 |     89603 |     17075
 base/16385/168683.3_mincore |       4096 |      66028 |     23965 |      4193
(4 lignes)

Temps : 102,307 ms
```

```
cedric=# select * from pgfadv_WILLNEED_snapshot('pgbench_accounts');
           relpath           | block_size | block_disk | block_free
-----------------------------+------------+------------+------------
 base/16385/168683_mincore   |       4096 |     262144 |     230414
 base/16385/168683.1_mincore |       4096 |     262144 |     140948
 base/16385/168683.2_mincore |       4096 |     262144 |      50986
 base/16385/168683.3_mincore |       4096 |      66028 |      26806
(4 lignes)

Temps : 38228,758 ms
```

Some Uses Cases : Monitoring



Some Uses Cases : Monitoring

Some Uses Cases : Performance Boost

- -- run a pgbench

  ./pgbench -S -t 100 -c2

  tps = 38.569719 (excluding connections establishing)
- -- restore buffer cache

  select * from pgfadv_willneed_snapshot('pgbench_accounts');

  select * from pgfadv_willneed_snapshot('pgbench_accounts_pkey');
- -- run a pgbench

  ./pgbench -S -t 100 -c2

  tps = 170.889926 (excluding connections establishing)

Track_disk PostgreSQL branch

- track_disk

```
relname                | pgbench_accounts
heap_blks_hit          | 442
heap_blks_read         | 39838
heap_blks_real_read    | 31023
idx_blks_hit           | 83635
idx_blks_read          | 37372
idx_blks_real_read     | 7112
```

Track_disk PostgreSQL branch

- bypass_os_cache
  - But the readahead is hitting disk before we request blocks
  - Needs snapshots

Ideas

- Auto bypass OS Cache for BIG Seq Scan
- Auto scale prefetch window
- Analyze disk my_table; -- and auto-analyze disk
- Explain analyze disk select a,b,c from my_table;

Future of PgFincore

- Snapshot/Restore store data in a table
- Windows, non-POSIX, BSD port
- Fincore() syscall in Linux kernel
- Mmap vs asyncIO ?
- Make it in PostgreSQL ?!

Thanks

- Andres Freund
- Andrew (RhodiumToad) Gierth

- YOU

References

- PgFincore
  - http://villemain.org/projects/pgfincore
- PostgreSQL patch
  - git://git.postgresql.org/git/users/c2main/postgres.git
  - branch track_disk
- Libprefetch (only literature)
  - http://libprefetch.cs.ucla.edu/
- Fincore – LWN, after first proposal
  - http://lwn.net/Articles/371538/
- Fincore syscall, commented by Andrew Morton
  - http://lwn.net/Articles/371540/

Time to ask

- Questions ?

- cedric@2ndQuadrant.fr