Gestion du cache en lecture / écriture

PGDAY France 2014

Cédric Villemain cedric@2ndQuadrant.fr

6 Juin 2014

Cédric Villemain

PostgreSQL Expertise, Développement, Support, Formation

Sponsor Platine de PostgreSQL

- **8.1**, PITR
- **8.2**, Warm Standby
- **8.3**, `pg_standby`
- **9.0**, Hot Standby
- **9.1**, Synchronous Replication
- **9.2**, Cascading Replication
- **9.3**, `Event Triggers`, *Refactoring*
- **9.4**, Replication Slot / Réplication Logique

Quel cache ? !

- cache applicatif / logiciel
- cache système
- cache matériel

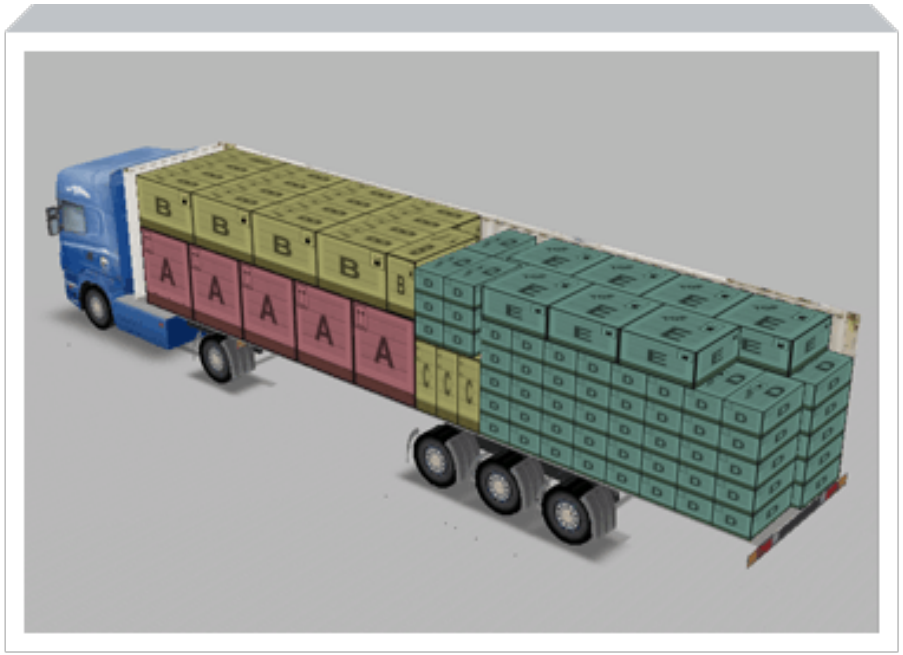
Et encore

- cache en lecture
- cache en écriture

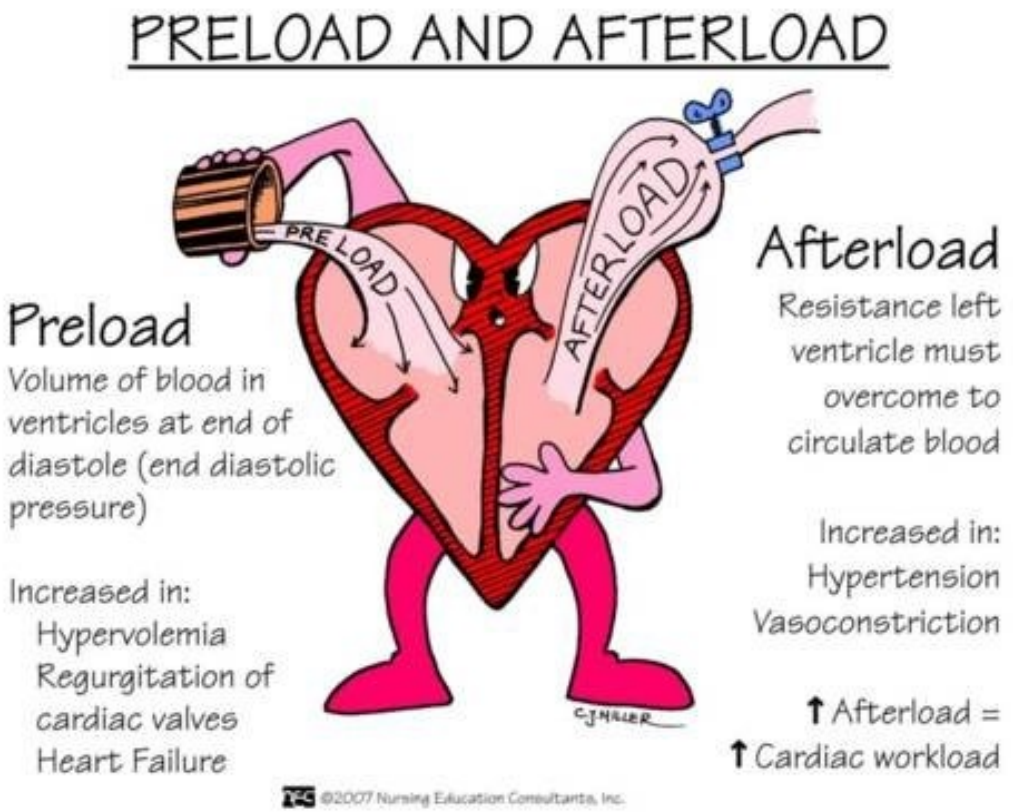
Cache en lecture

Cache en lecture

- PostgreSQL
- OS
- SAN
- RAID
- Disque dur



Cache en écriture



Cache PostgreSQL

Quelques paramètres

- effective_cache_size
- random_page_cost

Quelques outils

- clock sweep
- background writer
- checkpoint writer

Cache du Système d'exploitation / hardware

- meminfo
- NUMA

Cache du Système d'exploitation / hardware

- CFQ
- Anticipatory
- **Deadline**
- **NOOP**

API POSIX / API kernel

- POSIX_FADVISE
- mincore
- (fincore)
- linux, BSD*, UNIX like
- autre API pour windows

pg_buffercache

- Examine le contenu du cache PostgreSQL
- Snapshot
- Analyse

Exemples d'appel

```
SELECT b.*
FROM pg_buffercache b
JOIN pg_class c ON (b.relfilenode = c.relfilenode)
WHERE c.relname = 'pgbench_accounts';

SELECT usagecount, count(*),
       round(
             count(*) * 100
           / sum(count(*)) over ()
      )::text || ' %' as "%"
FROM pg_buffercache
GROUP BY usagecount
ORDER by 1 ;
```

pg_prewarm

- PostgreSQL API
- Utilise POSIX_FADVISE
- Charge 1 bloc à la fois
- prefetch | read | buffer

Exemples d'appel

```
SELECT pg_prewarm('ma_table')
AS number_of_block_prewarmed;

-- ou plus spécifique

SELECT pg_prewarm(
       'ma_table', 'buffer', 'main',
       0, 10)
AS number_of_block_prewarmed;
```

pgfincore

- PostgreSQL API : accès fichier
- Kernel API : mincore / fincore
- Utilise POSIX_FADVISE
- Charge 1 fichier ou 1 bloc à la fois
- prefetch seulement

pgfincore

- Snapshot mémoire
- Chargement du snapshot
- Export du snapshot

pgfincore

- Analyse mémoire cache
- lecture
- écriture
- Stratégie read_ahead

Exemples d'appel

```
SELECT * FROM pgfadvise_dontneed('ma_table');
SELECT * FROM pgfadvise_willneed('ma_table');
SELECT * FROM pgfadvise_random('ma_table');
SELECT * FROM pgfadvise_sequential('ma_table');
SELECT * FROM pgfadvise_normal('ma_table');
```

Exemples d'appel

```
CREATE TABLE pgfincore_snapshot AS
  SELECT 'pgbench_accounts'::text as relname, *,
         now() as date_snapshot
    FROM pgfincore('pgbench_accounts', true);
-- possible de coupler avec pg_buffer_cache et
-- sur serveur primaire ou secondaire:
SELECT databit
FROM pgfincore_snapshot s,
LATERAL pgfadvise_loader(
                s.relname, s.segment,
                true, -- charge bloc
                true, -- décharge bloc
                s.databit)
WHERE relname = 'pgbench_accounts' ;
```

Options

- Pré-chauffage shared_buffers : pg_prewarm
- Préchauffage cache système : pg_prewarm et pgfincore
- Modification read-ahead : pgfincore
- Analyse / snapshot cache : pgfincore et pg_buffercache

Concrètement

- Simple pré-chauffage : pg_prewarm
- Restauration cache (avant/après pg_dump) : pgfincore
- Redémarrage serveur : pgfincore
- Failover / switchover : pgfincore

En cours de développement

- Amélioration de mincore (linux)
- Ajout de fincore (linux)
- Amélioration de pgfincore

Des questions ?

Maintenant, ou plus tard, n'hésitez pas !