Supervision de PostgreSQL

Annick Battais (Géosciences Rennes - CNRS)

09/11/2017

GEOSCIENCES Rennes

Politique de supervision

- Pourquoi ?
- Pour qui ?
- Quels critères ?
- Quels outils ?

La supervision – pourquoi ?

- Améliorer les performances
- Améliorer l'applicatif
- Anticiper/Prévenir les incidents
- Réagir vite en cas de crash

La supervision – pour qui ?

- Développeur
  - Comprendre et optimiser les requetes
- Dba
  - Performance, surveillance, mise à jour
- Système
  - Performance, qualité de service

La supervision : quels critères?

- Exemples - Matériel
  - Carte raid
  - Barette mémoire
- Exemples – Système d'exploitation
  - Charge cpu : commande top
  - Entrées/sorties disque : top
  - Espace disque : df
  - Temps de réponse : à partir d'une requète test sous psql, time ...

La supervision : quels critères ?

- Exemples - Applicatifs
  - Tomcat : mémoire allouée
  - Autres ....
- Exemples - Bases de données
  - Nombre de connexions (ps)
  - Requêtes lentes
  - Utilisation du cache

La supervision avec postgres

- De façon occasionnelle : si on peut intervenir tout de suite
- De façon automatique : remontée d'alerte et conserver les informations
- Traces
- Statistiques d'activité
- Mais postgres ne sait pas garder ces informations

Traces

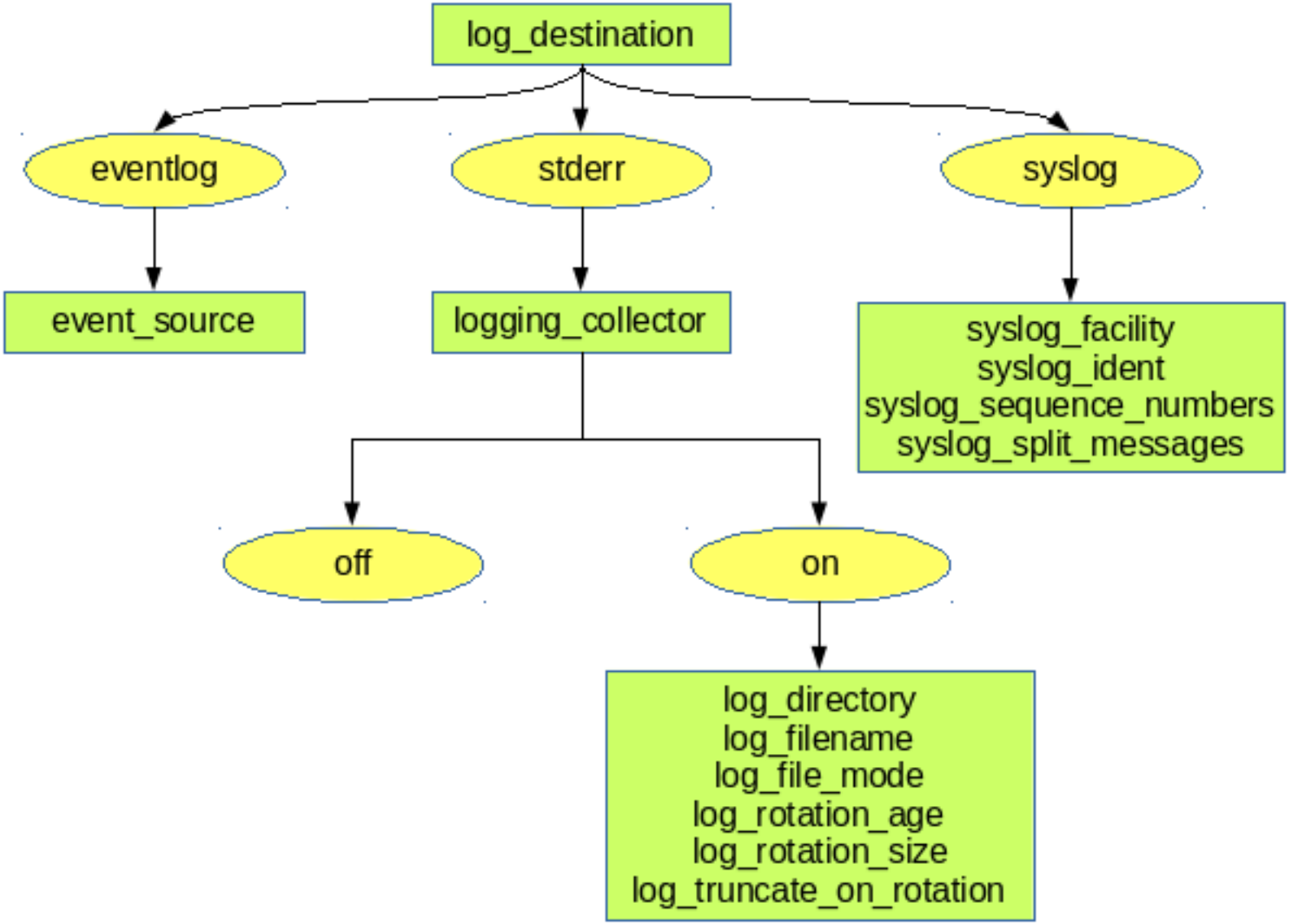
Configuration

- Où tracer
- Quel niveau de traces
- Tracer les requêtes

Récupération

Outils

Traces : où tracer ?



Traces : parametres de postgresql.conf

logging_collector = on

log_directory = 'pg_log' et redémarrer le serveur

Niveau de traces

- log_min_messages : niveau de traces (panic, fatal, log, error et warning)
- log_min_error_statement (defaut=erreur) : niveau de trace d'une requête
- log_error_verbosity

Tracer les requêtes

- log_min_duration_statement : si on veut tracer les t(r) > x
- log_statement
- log_duration

Traces : parametres de postgresql.conf

Tracer les requêtes :

- log_connections, log_disconnections : connexion des utilisateurs
- log_autovacuum_min_duration : si tps(autovacuum)
- log_checkpoints : trace le début et fin des checkpoints
- log_lock_waits : trace l'attente sur des verrous
- log_temp_files : trace la création de fichiers temporaires

Autres paramètres :

- log_line_prefix : ajout de préfixe à une trace
- lc_messages = 'C' pour mettre en anglais
- log_timezone : pour horodatage des traces

Traces : parametres de postgresql.conf

Ce qui est intéressant de récupérer :

- Durée d'exécution (log_statement, log_duration, log_min_duration_statement)

  ==> outils : pgbadger

- Messages de panic (PANIC: could not write to "....)

  ==> envoi d'une alerte par tail_n_mail

- Rechargement de la configuration (LOG: received SIGUP, ...)

  ==> envoi d'une alerte par tail_n_mail

- Fichiers temporaires, créés éventuellement lors d'un tri et lorsque work_mem ne permet pas de tout faire en mémoire

  ==> envoi d'une alerte par tail_n_mail

Outils

- Beaucoup d'outils existent
- Deux types :
  - en temps réel
  - rétro-analyse
- Nous verrons les plus connus/intéressants :
  - PgBadger
  - logwatch

Pgbadger

- Script perl de rétro-analyse
- https://github.com/dalibo/pgbadger/releases
- Traite les journaux applicatifs
- Recherche des informations
  - sur les requêtes et leur durée d'exécution
  - sur les connexions et sessions
  - Sur les checkpoints, verrous, autovacuum ...
- Génère une sortie html

Pgbadger

- Configuration minimale
  - log_destination = syslog ou stderr ou csvlog
  - log_min_duration_statement = 0
  - lc_messages = 'C'
  - log_line_prefix = '%t [%p]:[%l-1] user=%u,db=%d '
- Configuration de base
  - log_connections, log_disconnections
  - log_checkpoints, log_lock_waits, log_temp_files
  - log_autovacuum_min_duration
- Configuration temporaire
  - log_min_duration_statement

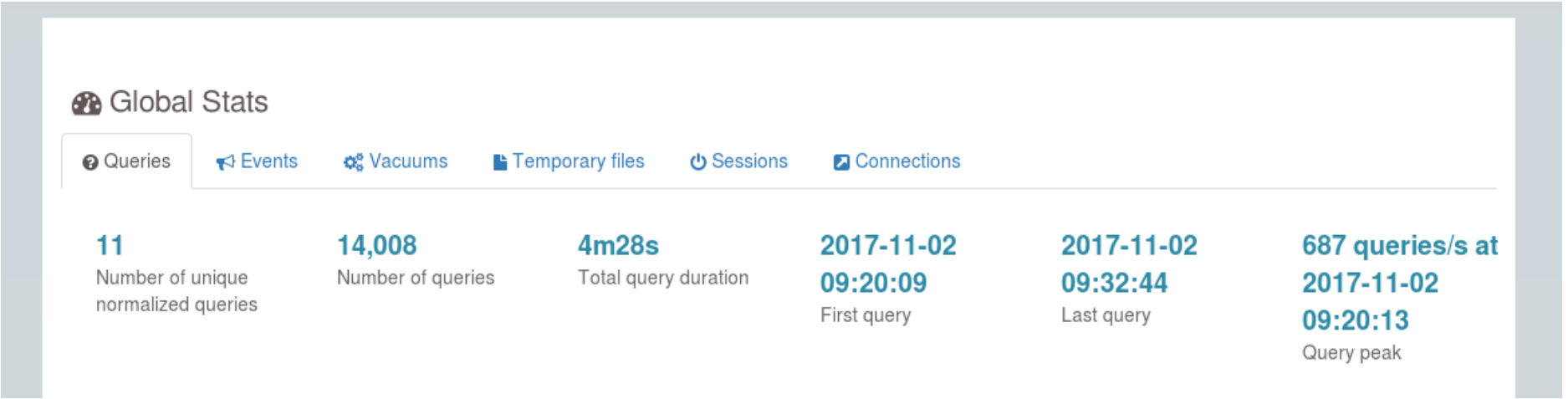
Pgbadger

- Options
  - --outfile
  - --begin
  - --end
  - --dbname
  - --dbuser
  - --dbclient
  - --appname
  - --prefix

Pgbadger : exemple 1

Statistiques générales sur les requêtes

- Nombre de requêtes normalisées/nombre de requêtes total
- durée totale d'exécution



Pgbadger : exemple 2

Statistiques générales sur les requêtes

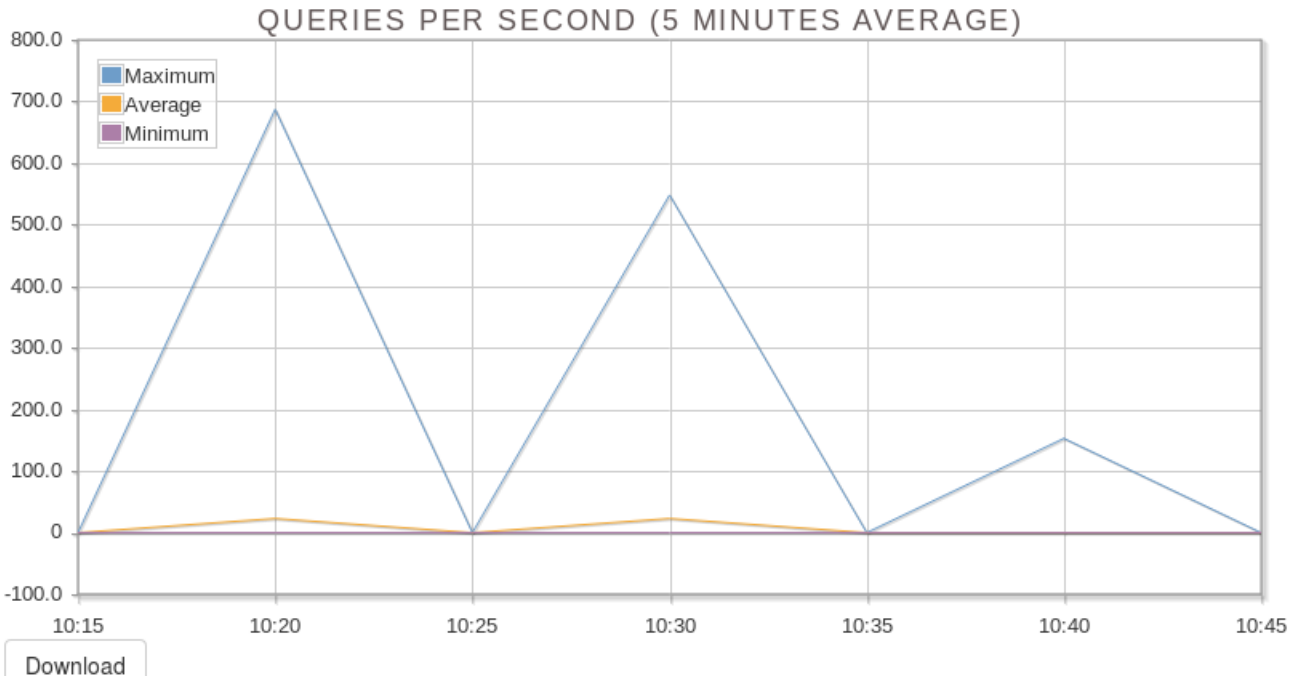
- Nombre moyen de requêtes

SQL Traffic

KEY VALUES

687 queries/s
Query Peak

2017-11-02 09:20:13
Date



logwatch

- Script perl
  - http://sourceforge.net/projects/logwatch
  - https://github.com/dalibo/pgsql_logwatch/
  - A installer manuellement

- Analyser les journaux de nombreux services et produire un rapport synthétique
- A exécuter périodiquement
- Envoie un mail s'il détecte certains motifs

logwatch

- Options
  - --logfile : les fichiers à traiter
  - --service, filtre sur le service à traiter
  - --detail, niveau de detail du rapport
  - --print, affiche le rapport sur la sortie standard
  - --save, sauvegarde du rapport dans un fichier
  - --mailto, pour envoyer le rapport

logwatch

- /usr/sbin/logwatch --detail Med --service postgresql --range All
- /usr/sbin/logwatch

```
--detail Med \
--service postgresql \
--range Yesterday \
--output mail \
--mailto admin@mydom.com \
--format html
```

Autres outils

- tail_n_mail (EndPointCorporation)
  - Outil externe écrit en perl
  - Analyser le contenu des journaux applicatifs
  - Envoyer un message si détection de certains motifs
  - A installer manuellement
- Sondes pour nagios
  - check_postgres
  - check_pgactivity (développé par dalibo, donne plus d'infos que check_postgres)

Statistiques d'activité

- Tracer l'activité en cours

  track_activities = on

- S'assurer que les requêtes ne sont pas tronquées
  - track_activity_query_size (par défaut 1024)
- track_counts
  - Pour récupérer des informations sur les nombre de lignes (lues,insérées ..) et blocs lus par postgreSQL
- track_io_timing
  - Pour chronométrer des operations sur disque

Statistiques d'activité

- track_functions
  - Pour récupérer des statistiques sur les procédures stockées
- stats_temp_directory
  - Pour déplacer le fichier des stats (voir le défaut dans postgresql.conf : global.stat)
  - Le collecteur modifie ce fichier quand les autres processus lui fournissent des stats

Statistiques d'activité

- Informations intéressantes
  - sur l'activité
  - sur l'instance
  - sur les bases
  - sur les tables
  - sur les index
  - sur les fonctions

Statistiques d'activité

- Nombre de connexions par base

```
SELECT datname, numbackends FROM pg_stat_database;
SELECT datname, count(*) FROM pg_stat_activity GROUP BY 1;
```

- Taille des bases

```
SELECT datname, pg_database_size(oid) FROM pg_database;
```

- Nombre de verrous

```
SELECT d.datname, count(*) FROM pg_locks l
JOIN pg_database d ON l.database=d.oid
GROUP BY d.datname ORDER BY d.datname;
```

Statistiques d'activité

- Beaucoup d'outils existent pour exploiter les statistiques
- Les plus connus/intéressants sont :
  - Munin : récupère les stats toutes les 5 minutes et crée des pages html ou png statiques (graphe, pas d'alerte)
  - Nagios : utilise les sondes check_postgres ou check_pgactivity (alerte, pas de graphe)
  - OPM (dalibo) : Ppen postgresql monitoring stocke les stats dans une base de données (graphe, sonde pour nagios)
  - pg_stat_statements : contrib de postgres, collecte de stats sur les requêtes exécutées
  - PoWA (dalibo) effectue des captures des statistiques collectées par pg_stat_statements et observe en temps réel les requêtes

Conclusion

- Un système est perenne si bien supervisé
- Supervision automatique
  - Configurer les traces
  - Configurer des statistiques
  - Mettre en place des outils d'historisation

Bibliographie

Postgresql, Architecture et notions avancées – Guillaume Lelarge

Documentation PostgreSQL 9.6.5
https://docs.postgresql.fr/9.6/pg96.pdf

Formation Dalibo – DBA4 - PostgreSQL Performances
https://cloud.dalibo.com/p/exports/formation/manuels/formations/dba4/dba4-pdf.pdf

Formation Dalibo – module H1 Supervision
https://cloud.dalibo.com/p/exports/formation/manuels/modules/h1/h1-pdf.pdf