PAF

...Une brique dans la mare de la HA

QUI SUIS-JE ?

- Jehan-Guillaume de Rorthais
- aka. ioguix
- utilise PostgreSQL depuis 2007
- dans la communauté depuis 2008
- @dalibo depuis 2009

AU MENU

- intro sur la HA
- intro sur Pacemaker
- pourquoi PAF
- Fonctionnalités de PAF

HAUTE DISPONIBILITÉ

- déjà pas mal de projets existants (la mare)
- pourquoi un de plus ?

RAPPELS SUR LA HA

- High Availability, aka. Haute disponibilité
- Intégrée dans un Plan de Continuité de Service
- En gros, on double tout
- Bascule automatique ou non ?

HA service != HA donnée

un PCA ne nécessite pas forcément de bascules auto. Des procédures manuelles peuvent aussi être mises en œuvre.

mais si le métier l'exige, alors il faut y passer.

BASCULE AUTO: TECHNIQUEMENT

Complexe à mettre en œuvre:

- comment détecter une vraie défaillance ?
- le maître ne répond plus ?
- est-il vraiment éteint ?
- est-il vraiment inactif ?
- comment réagir à une panne réseau ?
- comment éviter les split brain ?

BASCULE AUTO: APPRENTISSAGE

La route est (presque) droite mais la pente forte:

- plusieurs problèmes/mécaniques à comprendre: quorum, fencing, watchdog, ...
- configuration pointue
- maintenances complexes

- documenter, documenter, documenter
- tester, tester, tester

Si vous ne voulez pas de complexité, n'en faites pas.

---

true

PACEMAKER

Apprentissage difficile…

…Toute résistance est futile.

nombreux outils existant pour la HA/PgSQL

certains ne connaissaient juste pas l'existance de Pacemaker

Aucun n'arrive à son panel fonctionnel

GÉNÉRALITÉS SUR PACEMAKER

- est **LA** référence de la HA sous Linux
- est un "Cluster Resource Manager"
- supporte les fencing, quorum et watchdog
- multi-service, avec dépendances, ordre, contraintes, règles, etc
- s'interface avec n'importe quel service
- chaque service a son **Resource Agent**
- RA stateless ou multi-state
- API possibles: script OCF, upstart, systemd, LSB

ARCHI PACEMAKER

MÉCANIQUE DU CRM

- sorte d'automate
- 4 états: arrêté, démarré, slave ou master
- calcul de transitions pour passer d'un état à l'autre
- API classique (eg. systemd): start, stop, monitor(status)
- API OCF: start, stop, promote, demote, monitor, **notify**
- cas d'une ressource multi-state:

Cette mécanique est importante. L'agent existant ne la respecte pas car PostgreSQL ne supportait pas réellement le demote. Le code est donc complexe pour rien.

Nous allons dans la slide suivant causer du notify

ACTION NOTIFY

- seulement possible avec RA OCF
- déclenchés avant et après chaque action
- permet au RA d'effectuer des opérations

INFOS DU NOTIFY

Variables transmises au RA lors d'un 'pre-promote':

```
active    => [ ],
inactive  => [
               { rsc => 'pgsqld:2', uname => 'srv1' },
               { rsc => 'pgsqld:0', uname => 'srv2' },
               { rsc => 'pgsqld:1', uname => 'srv3' }
             ],
master    => [ ],
slave     => [ ],
promote   => [ { rsc => 'pgsqld:0', uname => 'srv1' }],
demote    => [ ],
start     => [
               { rsc => 'pgsqld:0', uname => 'srv1' },
               { rsc => 'pgsqld:1', uname => 'srv3' },
               { rsc => 'pgsqld:2', uname => 'srv2' }
             ],
stop      => [ ],
type      => 'pre',
operation => 'promote'
```

l'exemple ici ressemble au démarrage du cluster.

Les différentes combinaisons ici permettent de détecter un recover d'esclave, de maitre, un move (switchover), etc.

---

true

MASTER SCORE

- pondère quel slave promouvoir en master
- un slave doit avoir un score positif pour être promu
- pas de promotion si aucun score
- positionné à la discrétion du RA et/ou de l'administrateur

HISTOIRE

- conférence sur Pacemaker/pgsql en 2012
- traumatisme…
- …et une personne vient causer repmgr
- il fallait faire quelque chose…PAF débuté en 2015
- entre temps d'autres projets autour de Pacemaker
- auteurs: Maël Rimbault, moi même

le RA pgsql existant est:

- très fastidieux
- cours plusieurs lapins
- difficile à maintenir
- procédure lourde
- configuration lourde,
- détails d'implémentation visibles, …

OBJECTIFS

- garder Pacemaker: il fait tout et bien à notre place
- se concentrer sur notre métier: PostgreSQL
- coller à l'API OCF, à Pacemaker et les respecter
- garder une configuration du RA **SIMPLE**
- ne supporter **QUE** le multi-state
- ne supporter **QUE** la Streaming Replication
- avoir un code compréhensible et documenté

… PAF n'est qu'une brique (RA OCF pour Pacemaker)

API OCF donne accès à toute la puissance de Pacemaker: start, stop, promote, demote, monitor, notify

true

VERSIONS

Deux versions pour les attraper tous!

- 1.1: jusqu'à EL6 et Debian 7
- ...ou jusqu'à Pacemaker 1.12/corosync 1.x
- 2.0: depuis EL7 et Debian 8
- ... ou depuis Pacemaker 1.13/Corosync 2.x

Pas eu d'annonce officielle de la 1.0 car test et workshop interne.

Le second cycle de développement couvre un certain nombre de point soulevés lors des workshop.

La version 2.0 va pas tarder à être publiée.

La 1.1 est moins prioritaire mais devrait suivre peu de temps après.

CONFIGURATION

- system_user
- bindir
- datadir (oops)
- pgdata
- pghost
- pgport
- recovery_template
- start_opts

CONFIGURATION

Agent `pgsql` historique dans Pacemaker:

- pgctl
- start_opt
- ctl_opt
- psql
- pgdata
- pgdba
- pghost

- pgport
- pglibs
- monitor_user

CONFIGURATION

Encore ?

- monitor_password
- monitor_sql
- config
- pgdb
- logfile
- socketdir
- **stop_escalate**
- **rep_mode**
- **node_list**
- restore_command

CONFIGURATION

C'est pas fini…

- archive_cleanup_command
- recovery_end_command
- master_ip
- repuser
- primary_conninfo_opt
- **restart_on_promote**
- **replication_slot_name**
- tmpdir
- **xlog_check_count**
- **crm_attr_timeout**

---

true

Et les derniers pour la route:

- stop_escalate_in_slave
- check_wal_receiver

…et ne parlons pas du code en lui même, de sa lisibilité, du langage inapproprié, etc.

Par ailleurs, les dev de Pacemaker discutent de l'ajout de librairies/modules pour d'autres langages

---

SON SAVOIR FAIRE

...on ouvre le chˆWcapot.

ENTRAILLES

- écrit en perl
- demote = stop + start (= slave)
- mécanique d'élection en cas de failover
- détecte les types de transitions grâce aux notify (recovers et move)

depuis la 9.3, un demote fiable est possible.

backporté jusqu'à en 9.1 à priori.

Détecter les transitions a été un gros travail en 1.1 et 2.0, mais ça nous permet de faire des trucs assez cool.

Par exemple...

RECOVER D'UN STANDBY

Transition: `stop` -> `start`

Démo recover d'un slave:

Your browser does not support the video tag.

- 3 srv
- master on srv1
- master IP on srv1
- slaves on srv2 & srv3
- continuous writes being replicated
- show the master scores
- warn: the master might not detect a slave failure (depend on monitor calls)

---

true

RECOVER D'UN MASTER

Transition: `demote` -> `stop` -> `start` -> `promote`

Démo recover master:

Your browser does not support the video tag.

Don't forget to show that after rever, the slave hook on the master immediately

---

FAILOVER ET ÉLECTION

Démo failover avec élection:

Your browser does not support the video tag.

Failover2:

- based on score, failover is supposed to promote on srv2
- but in reality, srv3 is the best slave
- look at the log on srv2!!
- srv1 is getting fenced
- srv2 changes the scores
- after the failover, srv3 changes the score again because srv2 is not connected

---

VOUS PENSIEZ EN AVOIR FINI ?

ÉCHANGE DE RÔLE

- uniquement en 2.0
- le standby désigné valide les données reçues du master
- le standby annule la promotion si intérgation de l'ancien master non garanti

---

Démo échange des rôles:

Your browser does not support the video tag.

---

true

OÙ

«Mais où est donc passé ce chien ?»

- site officiel: http://dalibo.github.io/PAF/
- code: https://github.com/dalibo/PAF
- releases: https://github.com/dalibo/PAF/releases
- support: https://github.com/dalibo/PAF/issues

- docs et quick start dispos sur le site officiel
- support et rapport de bug sur la page d'issue
- pas de mailing list pour le moment

DES QUESTIONS ?

- recover d'un master (qui d'autre le fait ?)
- recover d'un slave (qui d'autre le fait ?)
- échange des rôles d'un slave (qui pourrait me donner le vocabulaire correct ?)
- repmgr le fait…mais à coup de pg_rewind ou rsync
- repmgr le fait…mais que sur un cluster à deux nœuds
- repmgr le fait…mais pensez à couper repmgrd avant

---

true