
PROFESSIONAL
Postgres
Отказоустойчивость —
с чем ее едят?
Teodor Sigaev
www.postgrespro.ru

О чем это я?

Технические требования. О чем они. По крайней мере у нас. О кластере.

Встроенный-встроеннный

Но ведь есть Patroni, Stolon, etcd et cetera

Patroni, Stolon, etcd et cetera

- Надо устанавливать и мониторить - непростота
- Внешняя же штука — пострес о себе знает всяко больше чем что-то внешнее
- Не похоже на zero-administration. А хочется вообще zero-configuration

Встроенный кластер

Встроенный отказоустойчивый кластер, репликация

Отсутствие новых администрируемых единиц, типа etcd, zookeeper, patroni etc. Это не значит, что нельзя использовать другие пакеты, библиотеки и тд. (ну не собираемся мы разрабатывать начиная с ассемблера)

Ограничим себя

Отказоустойчивость vs катастрофоустойчивость

Простоту юзерам!

- master: initdb ...
- slave: initdb --cluster PGCONN
  [--synctype T]
  [--priority ID]

SQL Интерфейс

- список нод / конфигурация кластера типа числа синхронных нод
- отставание
- команда на переключение мастера
- остановка кластера при числе нод меньше заданного
- временная остановка реплики (как минимум необходима для плавного минорного апгрейда)
- исключение реплики
- логирование изменений кластера и просмотр логов (?!)

Новый мастер

- Внешний выбор
- В порядке приоритета
  - Что-то рафтоподобное все равно нужно, что бы договориться (например, что мы вообще переключаемся)

Угрозы

- Не защищаемся от угроз, защиту от которых обеспечивается только фенсингом (почему? - см отказоустойчивость)
- Выход из строя мастера (питание, внутренняя ошибка, место на диске)
- Стоит ли рассматривать выход нескольких нод сразу? (самый неприятный — мастер + реплика)

Синхронность

- Асинхронная реплика
- Синхронная реплика
  - Получение
  - Сохранение
  - Применение

Реплика fail

- Выбрать новую «любимую жену»
- Встает вопрос конфигурации — новая синхронная реплика?

Мастер fail

- Выбираем новый мастер тем или иным способом
  - Просто по ID?
  - Выбор зависит от синхронности, если все асинхронны — то самый обновившийся
  - Мб ограничение — только единственная реплика может быть асинхронной, если реплик несколько — хотя бы одна должна быть синхронной
  - Отключение реплики (Антон!)
- Страшный сон — split brain
  - Запрет на работу при недостатке живых нод или при отстутствии синхронных реплик

Временные ограничения

- На принятие решения - минута
- С момента принятия решения
  - на синхронную реплику < 1мин
  - на асинхронную не лимитируется, поскольку не известно ее отставание

Поддерживаемые конфигурации - 1

2 ноды. Мастер + слейв

- Нет проблем с выбором нового мастера, есть проблемы с принятием решения.
- Очевидные проблемы split-brain (выбор админа)
- Производительность vs непотеря. Или [a]синхронность чего-бы-то-небыло

Поддерживаемые конфигурации - 2

2 + 1 ноды. Мастер + слейв + советник

- Неочевидные проблемы split-brain
- Гарантированное согласие за бесконечное время или достижимое согласие за конечное?

Поддерживаемые конфигурации - 3

>2 ноды. Мастер + слейвы

- Принятие решения
- Выбор нового мастера
- Неочевидные проблемы split-brain
- Гарантированное согласие за бесконечное время или достижимое согласие за конечное?

А что с приложением?

- Не хотим зависеть от окружения
- Прокси — внешний (а резервирование?)
- Умное приложение (sql интерфейс)
- Несколько адресов в libpq
- Iptables (не все решает...)

Упрощения

- При промоуте на асинхронную реплику дополучение записей с мастера производится только при плановом переключении. При аварийном переключении дополучение не производится
- Восстановление временно отключенной ноды может быть произведено только если на мастере есть необходимые валы. иначе только перезаливкой. Вообще говоря, можно путем восстановления из бекапа, но штатным путем это будет не в первой версии
- Если она бывший мастер - то восстановление в качестве реплики без перезалития представляется слишком трудозатратным в первой версии

Упрощение

Linux-only пока

2+1 -

Postgres PROFESSIONAL

Спасибо за внимание!

teodor@postgrespro.ru

www.postgrespro.ru