

База данных PostgreSQL: восстановление состояния на заданный момент времени

Вадим Исаканов, инженер Southbridge

Резервирование на базе WAL

Write-Ahead Logging (WAL) - Журнал опережающей записи

Изменения в файлах с данными (с таблицами и индексами) записываются только сохранения изменений были занесены в журнал.

В случае краха можно восстановить базу с помощью журнала (восстановление с помощью наката, или REDO)

<https://postgrespro.com/docs/postgresql/10/wal-intro>
<http://postgresql.ru.net/manual/wal-intro.html>

Восстановление на базе WAL

WAL-логи позволяют восстановить состояние базы на заданный момент времени:

```
[root@node1 ~]# ll /var/lib/pgsql/10/data/pg_wal/
total 950280
-rw----- 1 postgres postgres      305 Oct 19 08:22 000000010000000600000009A.00000028.backup
-rw----- 1 postgres postgres 16777216 Oct 19 08:46 0000000100000007000000022
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000023
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000024
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000025
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000026
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000027
-rw----- 1 postgres postgres 16777216 Oct 19 08:47 0000000100000007000000028
-rw----- 1 postgres postgres 16777216 Oct 19 08:48 0000000100000007000000029
-rw----- 1 postgres postgres 16777216 Oct 19 08:48 000000010000000700000002A
-rw----- 1 postgres postgres 16777216 Oct 19 08:48 000000010000000700000002B
-rw----- 1 postgres postgres 16777216 Oct 19 08:48 000000010000000700000002C
```

Восстановление на базе WAL

Восстановление на базе WAL-логи выполняется этапами:

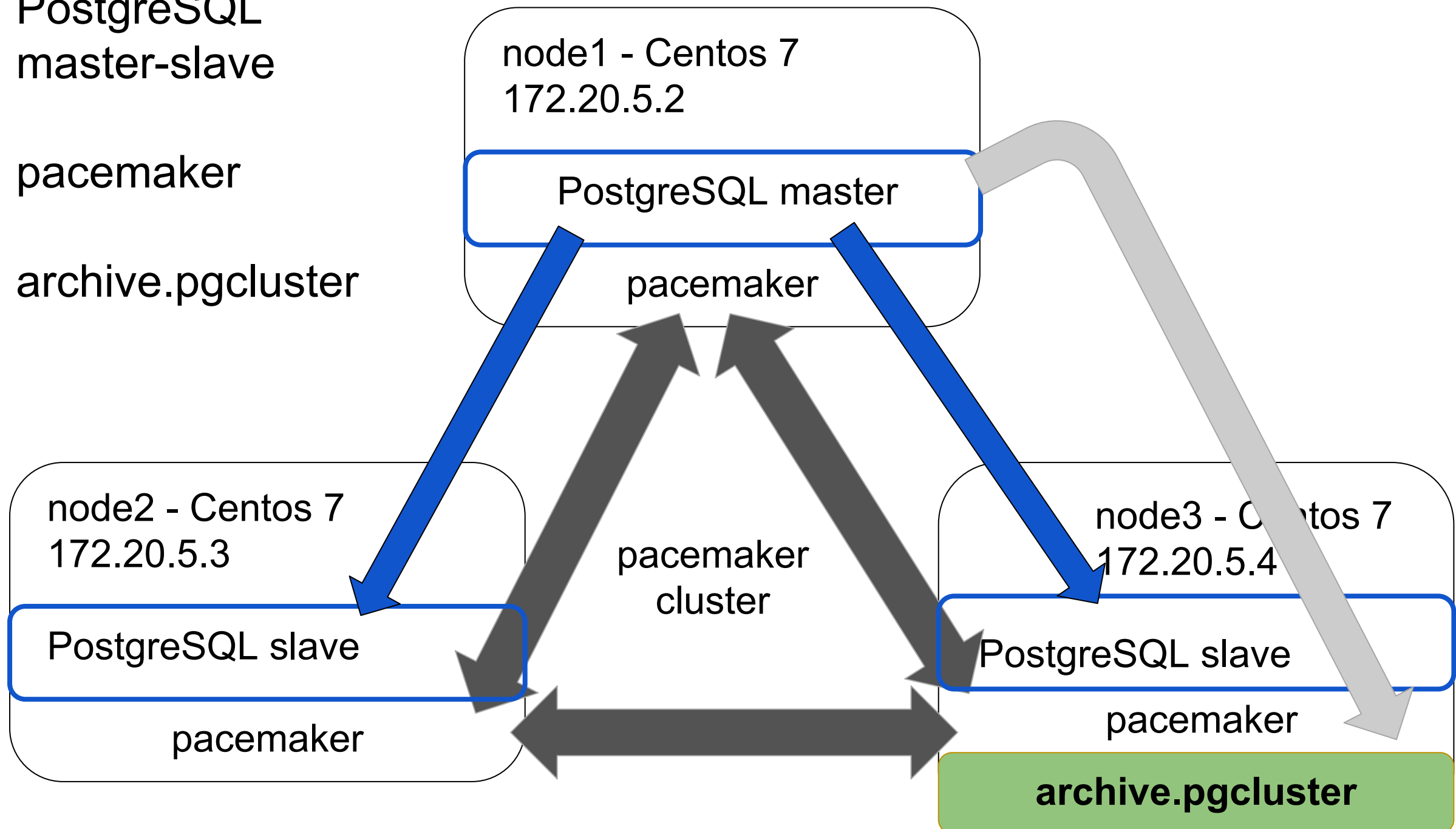
- восстановление полной физической копии
`/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data start`
`rm -rf /var/lib/pgsql/10/data/`
`tar -xvC /var/lib/pgsql/10/data/ archive/base.tar.gz`
- восстановление на базе WAL с момента создания копии до заданного времени
`restore_command = 'cp archivedir/%f %p'`
- запуск PostgreSQL, применение WAL-логов
`/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data start`

Добавляем роль archive.pgcluster

PostgreSQL
master-slave

pacemaker

archive.pgcluster



Установка, шаг 0

Ноды:

node1.pgcluster 172.20.5.2/24 Начальный мастер
node2.pgcluster 172.20.5.3/24 Реплика
node3.pgcluster 172.20.5.4/24 Реплика | **archive.pgcluster**

[all]

cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.20.5.2 node1.pgcluster node1

172.20.5.3 node2.pgcluster node2

172.20.5.4 node3.pgcluster node3 **archive.pgcluster**

172.20.5.5 **cluster1.pgcluster** cluster1

172.20.5.6 **cluster2.pgcluster** cluster1

Установка, шаг 1

Установка ПО

[all]

```
yum install epel-release -y  
yum install mailx lbzip2 -y
```

Настройка хранилища

[archive|node3]

```
useradd -d /srv/walbackup walbackup  
useradd -G walbackup,postgres -d /srv/basebackup basebackup  
echo 'PASSWORD0' | passwd --stdin basebackup  
echo 'PASSWORD1' | passwd --stdin walbackup  
chmod g+rwX /var/run/postgresql  
chmod g+rx /srv/walbackup
```

Установка, шаг 2

Поместим в систему скрипты для бэкапа и очистки.

[archive]

```
mkdir -p /srv/southbridge/bin
cd /srv/southbridge/bin && { curl \
  -O "https://raw.githubusercontent.com/nbw74/scripts/trunk/backup/pgsql/sb-pgsql-basebackupwrapper.sh" \
  -O "https://raw.githubusercontent.com/nbw74/scripts/trunk/backup/pgsql/sb-pgsql-prunewal.sh" \
  ; chmod a+x sb-pgsql-basebackupwrapper.sh sb-pgsql-prunewal.sh \
  ; cd -; }
```

Аутентификационная информация для pg_basebackup:

[archive]

```
sudo -iu basebackup
echo 'cluster2.pgcluster:5432:replication:replicator:REPLICATOR_PASSWORD' >> .pgpass
chmod 0600 .pgpass
```


Установка, шаг 3

Настроим crontab

[archive]

```
sudo -iu basebackup crontab -e
```

```
04 00 * * */4 /srv/southbridge/bin/sb-pgsql-basebackupwrapper.sh --pg-version=10 --depth=8  
cluster2.pgcluster
```

```
sudo -iu walbackup crontab -e
```

```
*/19 * * * * /srv/southbridge/bin/sb-pgsql-prunewal.sh --max-used=86 --rm-count=100 --min-age=34
```

И для проверки запустим вручную скрипт бэкапа

[archive]

```
sudo -iu basebackup
```

```
/srv/southbridge/bin/sb-pgsql-basebackupwrapper.sh --pg-version=10 --depth=8 cluster2.pgcluster  
--debug
```

```
[root@node3 ~]# ll /srv/basebackup/cluster2.pgcluster/  
total 8  
drwx----- 20 basebackup basebackup 4096 Oct 18 19:11 2018-10-18T1911  
drwx----- 20 basebackup basebackup 4096 Oct 19 06:28 2018-10-19T0627
```

Настройка архивации WAL

Сделаем пользователю postgres беспарольный вход по SSH на archive.pgcluster как пользователю walbackup

[all]

```
sudo -iu postgres  
ssh-keygen -t ed25519 -N '' -f $HOME/.ssh/id_ed25519  
ssh-copy-id -i $HOME/.ssh/id_ed25519.pub walbackup@archive.pgcluster
```

Поместим на ноды кластера скрипт архивации

[all]

```
mkdir -p /srv/southbridge/libexec  
curl 'https://raw.githubusercontent.com/nbw74/roles/trunk/postgresql/files/pgsql-archive-command.sh' -o /srv/southbridge/libexec/pgsql-archive-command.sh  
chmod a+x /srv/southbridge/libexec/pgsql-archive-command.sh
```

■ Настроим PostgreSQL

Редактируем конфиг

[all]

```
vi /var/lib/pgsql/10/data/postgresql.conf
```

```
archive_mode    = on
archive_command = '/srv/southbridge/libexec/pgsql-archive-command.sh -V
10 -H archive.pgcluster -D
    pgcluster -p %p -f %f'
archive_timeout = 10
```

archive_timeout = 10 - в учебных целях

Настройка кластера и проверка

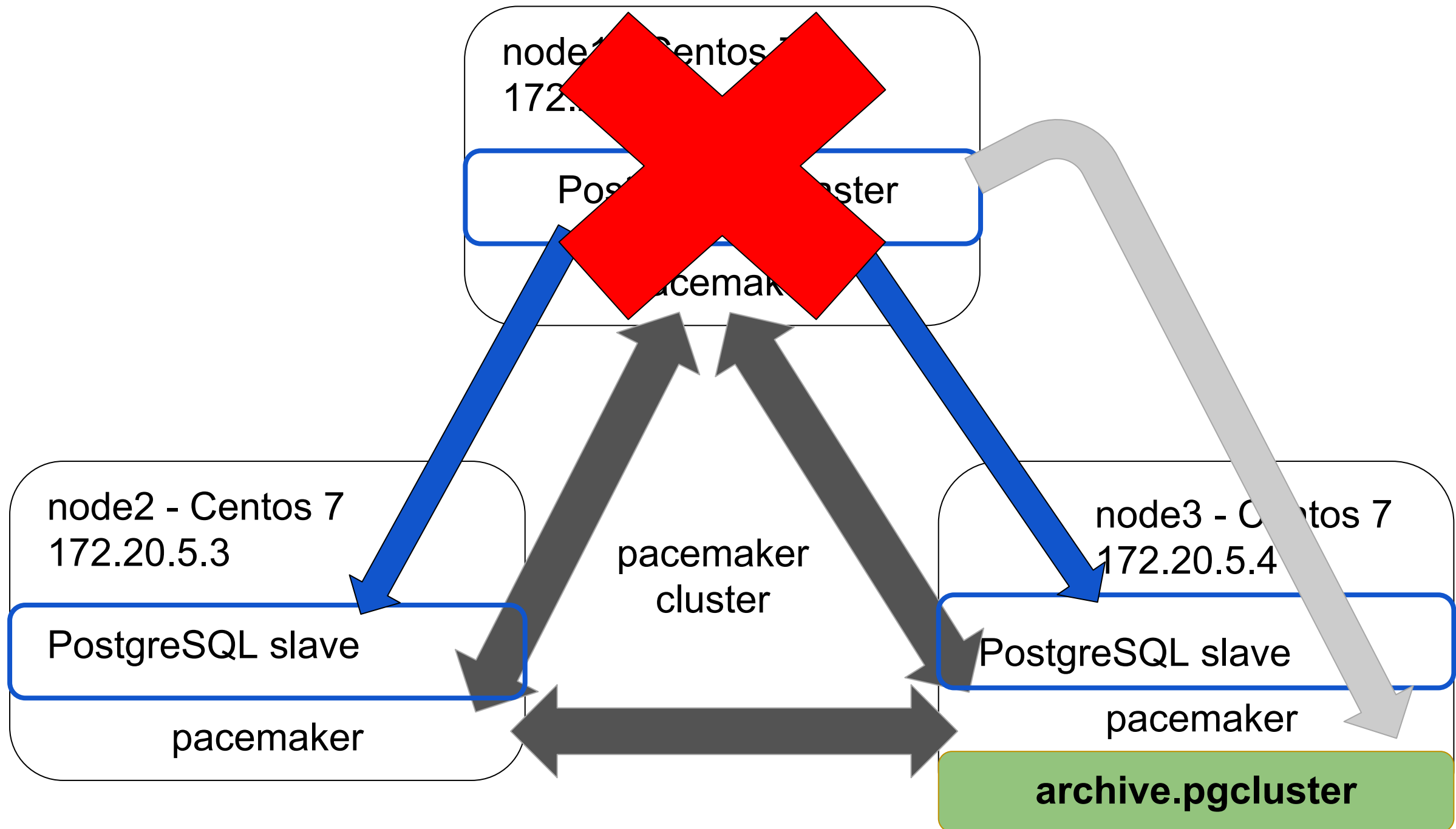
[master]

```
pcs property set maintenance-mode=true  
sleep 2  
sudo -iu postgres /usr/pgsql-10/bin/pg_ctl -D  
    /var/lib/pgsql/10/data restart  
sleep 2  
pcs property set maintenance-mode=false
```

После применения настроек расетакер архивация будет запущена:

```
[root@node3 ~]# ll /srv/walbackup/pgcluster/  
total 7160  
-rw----- 1 walbackup walbackup 882 Oct 19 07:22 000000010000000040000002C.bz2  
-rw----- 1 walbackup walbackup 5674 Oct 19 07:22 0000000100000000400000039.bz2  
-rw----- 1 walbackup walbackup 4025 Oct 19 07:22 000000010000000040000003A.bz2  
-rw----- 1 walbackup walbackup 4013 Oct 19 07:22 000000010000000040000003B.bz2  
-rw----- 1 walbackup walbackup 3979 Oct 19 07:22 000000010000000040000003C.bz2  
-rw----- 1 walbackup walbackup 4008 Oct 19 07:22 000000010000000040000003D.bz2
```


Тесты. Возвращаем в строй “выпавший” master



■ Тесты. Восстановление бэкапа:

По шагам:

- настраиваем `recovery.conf`
- отключаем в `postgresql.conf` архивирование WAL
- восстанавливаем на базе WAL с момента создания копии до заданного времени
`restore_command = 'cp archivedir/%f %p'`
- запуск PostgreSQL, применение WAL-логов
`/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data start`

Тесты. Восстановление бэкапа на заданный момент времени

Подготовительный этап -- выставление нужных прав файлам архива WAL

Подключаемся как basebackup

```
sudo -iu basebackup
```

```
cd cluster2.pgcluster/2022-06-17T0400/
```

```
sed -r -i 's|^(\archive_.*)|# \1|g' postgresql.conf
```

Создаем recovery.conf, указывая recovery_target_time

Запускаем Postgres

```
/usr/pgsql-10/bin/pg_ctl -D $PWD start
```

Ждем окончания запуска и применения WAL-логов

<https://gitlab.slurm.io/red/slurm/blob/master/lecture/6.pgsql-backup-and-restore-lecture.pdf>

РЕД
СЛЁРМ

+


Southbridge

slurm.io

