聊聊postgre的Replication

作者	时间	QQ技术交流群
perrynzhou@gmail.com	2021/12/11	672152841

存储内核技术交流

微信扫描二维码，关注我的公众号

开源存储问题解答社区:https://github.com/perrynzhou/deep-dive-storage-in-china

CAP理论

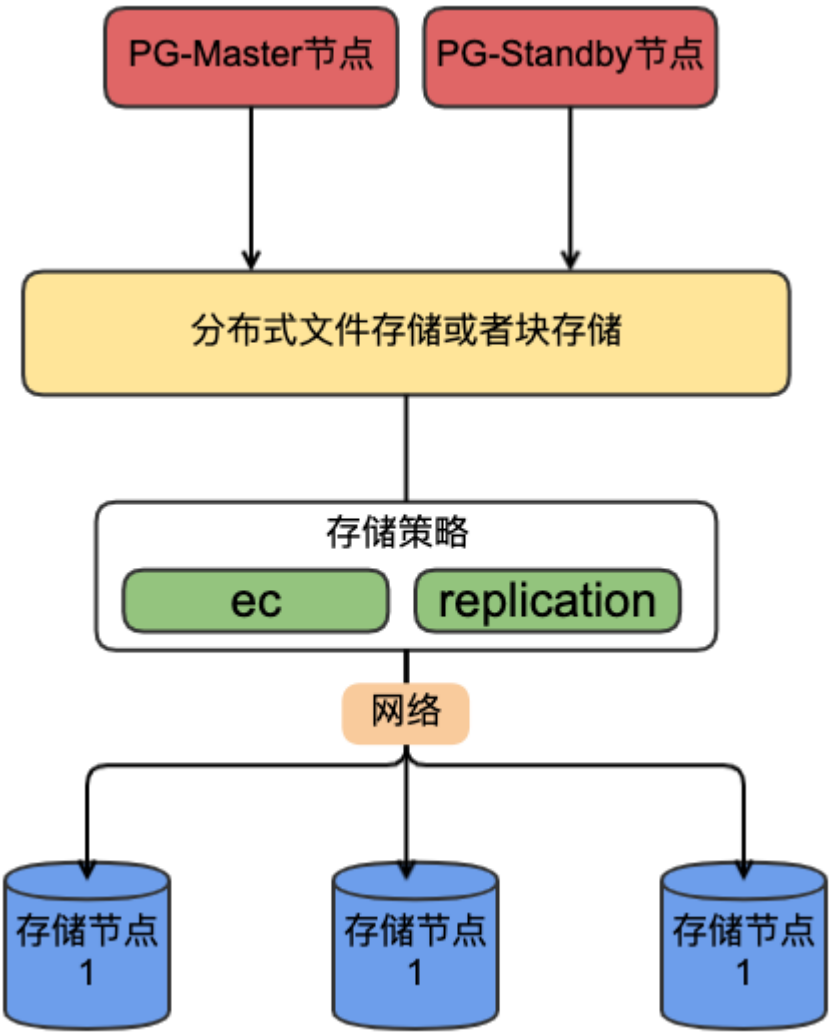
- `consistency`:在整个集群角度来看，每个节点是看到的数据一致的；不能出现集群中节点出现数据不一致的问题
- `vailability`:集群中节点，只有有一个节点能提供服务
- `partitioning`:集群中的节点之间网络出现问题，造成集群中一部分节点和另外一部分节点互相无法访问

基本术语

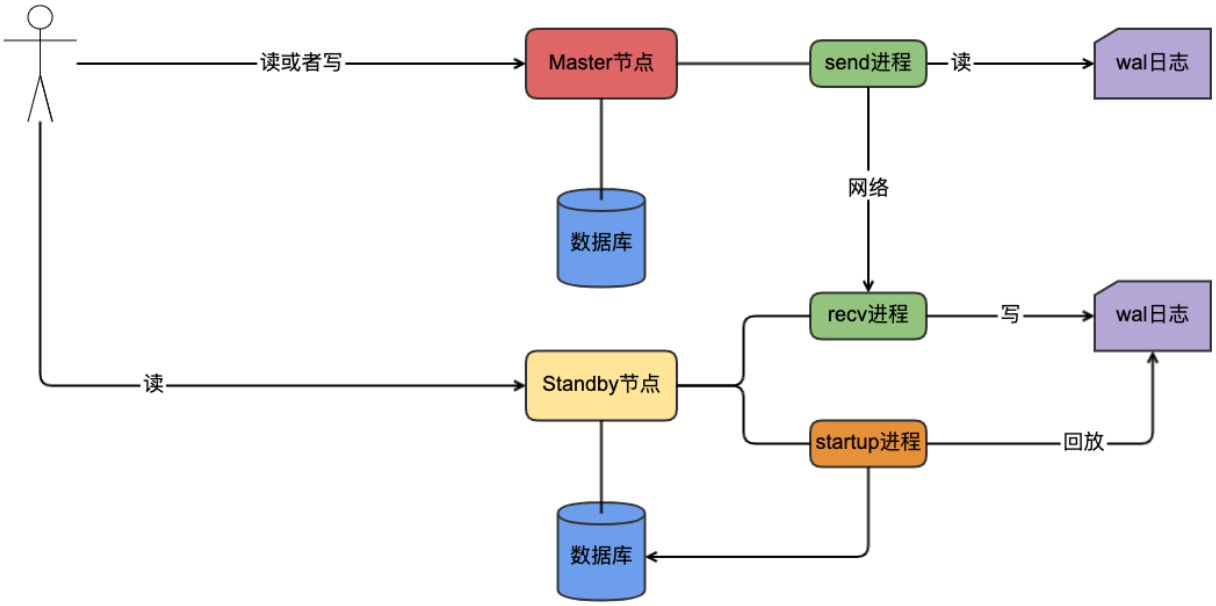
- `Master节点`:提供数据写的服务节点
- `Standby节点`:根据主节点(master节点)数据更改，这些更改同步到另外一个节点(standby节点)
- `Warm Standby节点`：可以提升为master节点的standby节点
- `Hot Standby节点`：主要提供读服务的standby节点

PostgreSQL支持的Replication方案

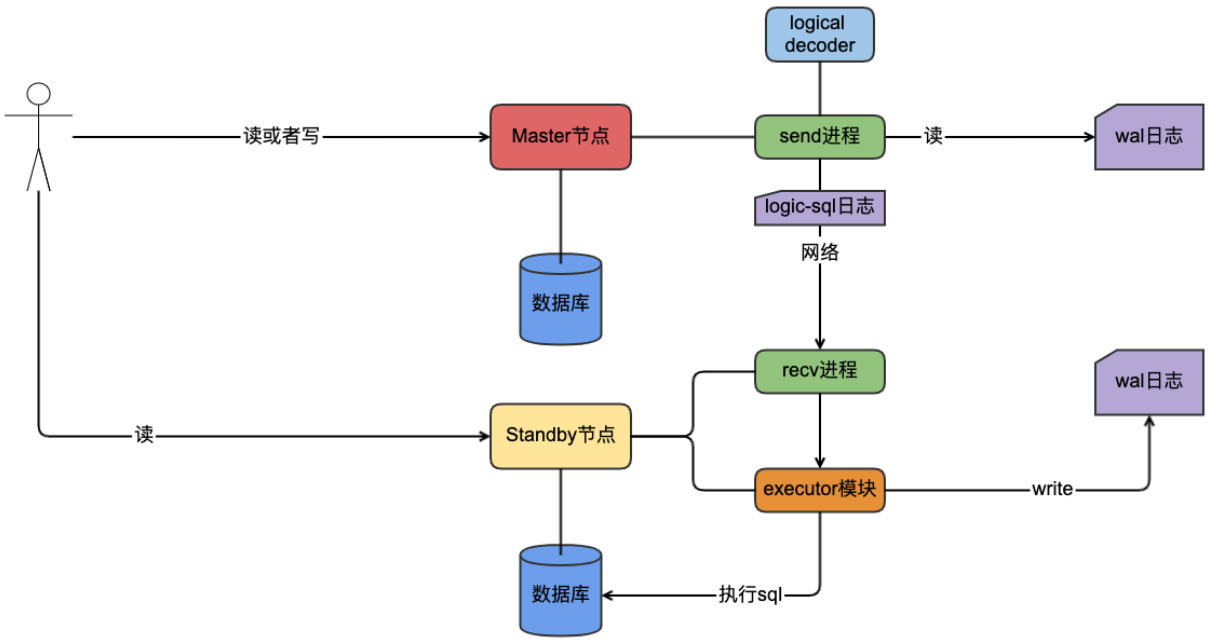
- 基于文件或者磁盘Replication:这种方式采用共享磁盘或者共享NAS方式，采用了存储计算分离的方式，如果采用这样的方式，PostgreSQL是计算节点，底层的是一个分布式块存储或者分布式文件存储。这样的好处很明显，只需要保证计算层的高可用即可，但是弊端也很明显由于底层是分布式存储，PG性能取决于分布式文件存储。如果底层的分布式存储做的足够健壮，数据基本不会丢失



- 基于wal的物理Replication：postgresql支持物理复制,其原理的就是先把Master节点的热备，然后传输到standby节点，在standby节点恢复;最后master不断的发送数据变更wal日志给standby节点，standby节点不断的接受wal日志，然后进行apply。物理复制是针对所有的Master节点上的databaase.由于wal是基于page的级别的，standby节点应用比较快，开销小。在物理复制中，Master节点会运行多个wal send进程;Standby节点会运行多个wal recv进程和startup进程，send是master发送wal日志的进程;recv进程是standby节点接受wal日志的进程,startup进程是standby节点apply wal日志的进程。



- 基于SQL的逻辑Replication:基本原理是应用端发出更改请求,master不断的产生日志，紧接着master的send进程读取wal日志，然后经过decode模块进行解析wal日志转换为类似于sql的方式发送给standby的recv进程，recv进程接受到sql日志，发送给standby的execute模块进行解码成为sql语句,然后执行sql语句，产生wal日志。



Replication实践

物理复制

- 准备两个PG实例

```
// 主节点 ip=127.0.0.1,port = 5432
// 从节点 ip=127.0.0.1,port = 5433
[perrynzhou@CentOS8-Dev /postgres]$ ps -ef|grep -v grep|grep postgre
perrynz+   13955       1  0 15:22 ?        00:00:00
```

```
/usr/local/postgres/bin/postgres -D /postgres/data1
perrynz+   13957   13955  0 15:22 ?        00:00:00 postgres: checkpointer
perrynz+   13958   13955  0 15:22 ?        00:00:00 postgres: background
writer
perrynz+   13959   13955  0 15:22 ?        00:00:00 postgres: walwriter
perrynz+   13960   13955  0 15:22 ?        00:00:00 postgres: autovacuum
launcher
perrynz+   13961   13955  0 15:22 ?        00:00:00 postgres: archiver
perrynz+   13962   13955  0 15:22 ?        00:00:00 postgres: stats
collector
perrynz+   13963   13955  0 15:22 ?        00:00:00 postgres: logical
replication launcher
perrynz+   13966       1  0 15:22 ?        00:00:00
/usr/local/postgres/bin/postgres -D /postgres/data2
perrynz+   13968   13966  0 15:22 ?        00:00:00 postgres: checkpointer
perrynz+   13969   13966  0 15:22 ?        00:00:00 postgres: background
writer
perrynz+   13970   13966  0 15:22 ?        00:00:00 postgres: walwriter
perrynz+   13971   13966  0 15:22 ?        00:00:00 postgres: autovacuum
launcher
perrynz+   13972   13966  0 15:22 ?        00:00:00 postgres: stats
collector
perrynz+   13973   13966  0 15:22 ?        00:00:00 postgres: logical
replication launcher
```

- 在主节点创建复制账户和备份主节点

```
// 主库创建数据库用户
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5432 -U perrynzhou
psql (14rc1)
Type "help" for help.

// 复制槽很重要，防止主库过早清理Wal
// 删除槽位信息 select pg_drop_replication_slot('slot_name');

postgres=# select * from
pg_create_physical_replication_slot('db1_repl_slot');
   slot_name    | lsn
---------------+-----
 db1_repl_slot |
(1 row)

postgres=# select slot_name, slot_type, active, wal_status from
pg_replication_slots;
   slot_name    | slot_type | active | wal_status
---------------+-----------+--------+------------
 db1_repl_slot | physical  | f      |
(1 row)
```

```
// 备份主库
$ /usr/local/postgres/bin/pg_basebackup --pgdata /postgres/master_backup --
format=p \
   --write-recovery-conf --checkpoint=fast --label=mffb --progress \
   --host=127.0.0.1  --port=5432 --username=perrynzhou
166886/166886 kB (100%), 1/1 tablespace

// 停止从库
/usr/local/postgres/bin/pg_ctl -D /postgres/data2/ -l pg_logfile2 stop
// 删除从库数据库
rm -rf /postgres/data2 && mv /postgres/master_backup  /postgres/data2

 // 添加配置到从库的postgresql.conf
primary_conninfo = 'host=127.0.0.1 port=5432 user=perrynzhou
password=zhoulin'
primary_slot_name = 'db1_repl_slot'
```

- 主从配置

```
// 主库 postgresql.conf
port = 5432
max_connections = 100
shared_buffers = 128MB
dynamic_shared_memory_type = posix
wal_level = replica
max_wal_size = 1GB
min_wal_size = 80MB
archive_mode = on
archive_command = 'cp %p /postgres/archive1/%f '
listen_addresses = '*'

//从库postgresql.conf
listen_addresses = '*'
archive_command = 'cp %p /postgres/archive2/%f '
port = 5433
max_connections = 100
shared_buffers = 128MB
dynamic_shared_memory_type = posix
max_wal_size = 1GB
min_wal_size = 80MB
wal_level = replica
hot_standby = on
max_standby_streaming_delay = 30s
hot_standby_feedback = on
primary_conninfo = 'host=127.0.0.1 port=5432 user=perrynzhou
```

```
password=zhoulin'
primary_slot_name = 'db1_repl_slot'
```

- 主从验证

```
// 主库
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5432 -U perrynzhou
psql (14rc1)

postgres=# create table tt1(id int);
CREATE TABLE
postgres=# insert into tt1 values(1);
INSERT 0 1
postgres=#

// 从库
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5433 -U perrynzhou
psql (14rc1)

postgres=# \d
          List of relations
 Schema | Name | Type  |   Owner
--------+------+-------+------------
 public | tt1  | table | perrynzhou
(1 row)

postgres=# \t
Tuples only is on.
postgres=# select * from tt1;
  1

postgres=#
```

逻辑复制

- 主从配置

```
port = 5432
max_connections = 100
shared_buffers = 128MB
dynamic_shared_memory_type = posix
wal_level = logical
max_wal_size = 1GB
min_wal_size = 80MB
archive_mode = on
archive_command = 'cp %p /postgres/archive1/%f '
```

```
listen_addresses = '*'

//从库postgresql.conf
listen_addresses = '*'
archive_command = 'cp %p /postgres/archive2/%f '
port = 5433
max_connections = 100
shared_buffers = 128MB
dynamic_shared_memory_type = posix
max_wal_size = 1GB
min_wal_size = 80MB
wal_level = replica
hot_standby = on
max_standby_streaming_delay = 30s
hot_standby_feedback = on
primary_conninfo = 'host=127.0.0.1 port=5432 user=perrynzhou password=zhoulin'
primary_slot_name = 'db1_repl_slot'
```

- 配置主库和从库

```
// 配置主库
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5432 -U perrynzhou
psql (14rc1)

postgres=# CREATE PUBLICATION my_publication FOR ALL TABLES;
CREATE PUBLICATION
postgres=#

// 配置从库
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5433 -U perrynzhou
psql (14rc1)

postgres=# CREATE SUBSCRIPTION my_subscription CONNECTION 'host=127.0.0.1
port=5432 user=perrynzhou dbname=postgres' PUBLICATION my_publication WITH
(copy_data=false);
NOTICE:  created replication slot "my_subscription" on publisher
CREATE SUBSCRIPTION
```

- 验证

```
// 主库插入数据
$ /usr/local/postgres/bin/psql  -h 127.0.0.1 postgres -p 5432 -U perrynzhou
postgres=# select * from tt2;
 id
----
```

```
(0 rows)

postgres=# insert into tt2 values(100);
INSERT 0 1

// 从库查看数据
[perrynzhou@CentOS8-Dev /postgres]$ /usr/local/postgres/bin/psql  -h
127.0.0.1 postgres -p 5433 -U perrynzhou
postgres=# select * from tt2;
 id
-----
 100
(1 row)

postgres=#

// 主库查看复制槽位信息
postgres=# select slot_name, slot_type, active, wal_status from
pg_replication_slots;
    slot_name     | slot_type | active | wal_status
------------------+-----------+--------+------------
 my_subscription | logical   | t      | reserved
(2 rows)
```