# 流复制异步同步性能测试

# 环境准备

## 主库参数设置

wal\_level = replica

archive\_mode = on

archive\_command = 'test ! -f /data01/pgsql/archivedir/%f && cp %p /data01/pgsql/archivedir/%f'

hot\_standby=on

restore\_command=

异步

alter system set synchronous\_standby\_names='';

alter system set synchronous\_commit='remote\_write';

select pg\_reload\_conf();

同步

alter system set synchronous\_standby\_names='';

alter system set synchronous\_commit='remote\_write';

select pg\_reload\_conf();

## 备库设置

alter system set primary\_conninfo='user=replica password=123456 channel\_binding=disable host=192.168.1.30 port=5432 sslmode=disable sslcompression=0 sslsni=1 ssl\_min\_protocol\_version=TLSv1.2 gssencmode=disable krbsrvname=postgres target\_session\_attrs=any application\_name=node2';

## 流复制

pg\_basebackup -F p -X stream -v -P -h 192.168.1.30 -p 5432 -U replica -D $PGDATA -R -C --slot pg31

# 测试用例

## **测试表**

create table user\_info

(userid int,

engname text,

cnname text,

occupation text,

birthday date,

signname text,

email text,

qq numeric,

crt\_time timestamp without time zone,

mod\_time timestamp without time zone

);

create table user\_session

(userid int,

logintime timestamp(0) without time zone,

login\_count bigint default 0,

logouttime timestamp(0) without time zone,

online\_interval interval default interval '0'

);

create table user\_login\_rec

(userid int,

login\_time timestamp without time zone,

ip inet

);

create table user\_logout\_rec

(userid int,

logout\_time timestamp without time zone,

ip inet

);

## 初始化数据

insert into user\_info (userid,engname,cnname,occupation,birthday,signname,email,qq,crt\_time,mod\_time)

select generate\_series(1,20000000),

'digoal.zhou',

'德哥',

'DBA',

'1970-01-01'

,E'公益是一辈子的事, I\'m Digoal.Zhou, Just do it!',

'digoal@126.com',

276732431,

clock\_timestamp(),

NULL;

insert into user\_session (userid) select generate\_series(1,20000000);

alter table user\_info add constraint pk\_user\_info primary key (userid);

alter table user\_session add constraint pk\_user\_session primary key (userid);

## 业务函数

create or replace function f\_user\_login

(i\_userid int,

OUT o\_userid int,

OUT o\_engname text,

OUT o\_cnname text,

OUT o\_occupation text,

OUT o\_birthday date,

OUT o\_signname text,

OUT o\_email text,

OUT o\_qq numeric

)

as $BODY$

declare

begin

select userid,engname,cnname,occupation,birthday,signname,email,qq

into o\_userid,o\_engname,o\_cnname,o\_occupation,o\_birthday,o\_signname,o\_email,o\_qq

from user\_info where userid=i\_userid;

insert into user\_login\_rec (userid,login\_time,ip) values (i\_userid,now(),inet\_client\_addr());

update user\_session set logintime=now(),login\_count=login\_count+1 where userid=i\_userid;

return;

end;

$BODY$

language plpgsql;

## 测试

使用pgbench进行压力测试, 发现瓶颈并合理优化

## pgbench用到的登陆脚本

[postgres@30p\_pg test]$ cat login.sql

\set userid random(1,20000000)

select userid,engname,cnname,occupation,birthday,signname,email,qq from user\_info where userid=:userid;

insert into user\_login\_rec (userid,login\_time,ip) values (:userid,now(),inet\_client\_addr());

update user\_session set logintime=now(),login\_count=login\_count+1 where userid=:userid;

同步

### 压力测试

pgbench -M prepared -r -c 8 -f /home/postgres/test/login.sql -j 8 -n -T 180 -h 127.0.0.1 -p 5432 -U postgres postgres

### **压力测试结果**

### **瓶颈分析与优化**

压力测试中查看数据库服务器的iostat -x

异步

### 压力测试

pgbench -M prepared -r -c 8 -f /home/postgres/test/login.sql -j 8 -n -T 180 -h 127.0.0.1 -p 5432 -U postgres postgres

### 压力测试结果

### 瓶颈分析与优化

压力测试中查看数据库服务器的iostat -x