

modb.pro

# PostgreSQL 15部分新特性预览

云和恩墨 彭冲



01

参数相关

02

开发相关

03

并行特性



# 01

## 参数相关



# GUC参数增加内部标识管理

- ❑ EXPLAIN: EXPLAIN 命令中包含的参数
- ❑ NO\_SHOW\_ALL: SHOW ALL命令中排除的参数
- ❑ NO\_RESET\_ALL: RESET ALL 命令中排除的参数
- ❑ NOT\_IN\_SAMPLE: 不包含在 postgresql.conf模板中的参数
- ❑ RUNTIME\_COMPUTED: 运行时计算的参数

```
postgres=# select distinct flags
postgres=# from (select pg_settings_get_flags(name)
postgres=# as flags from pg_settings) as x;
          flags
-----
{EXPLAIN}
{}
{EXPLAIN,NOT_IN_SAMPLE}
{NOT_IN_SAMPLE,RUNTIME_COMPUTED}
{NOT_IN_SAMPLE}
{NO_RESET_ALL,NOT_IN_SAMPLE}
(6 rows)
```

```
postgres=# select name,setting from pg_settings where
pg_settings_get_flags(name)='{NOT_IN_SAMPLE}';
```

name	setting
-----+-----	
allow_in_place_tablespaces	off
allow_system_table_mods	off
application_name	psql
backtrace_functions	
block_size	8192
...	

```
postgres=# select name,setting from pg_settings where
pg_settings_get_flags(name)='{NOT_IN_SAMPLE,RUNTIME_COMPUTED}';
```

name	setting
-----+-----	
data_checksums	off
data_directory_mode	0700
shared_memory_size	143
shared_memory_size_in_huge_pages	72
wal_segment_size	16777216
(5 rows)	

## □ 参数的set和alter system操作可授权

```
postgres=# grant set on parameter log_checkpoints to test;
GRANT
postgres=# grant alter system on parameter archive_mode to test;
GRANT
postgres=# select has_parameter_privilege ('test','log_checkpoints','set');
 has_parameter_privilege 
-----
 t
(1 row)

postgres=# select has_parameter_privilege ('test','log_checkpoints','alter system');
 has_parameter_privilege 
-----
 f
(1 row)

postgres=# select has_parameter_privilege ('test','archive_mode','alter system');
 has_parameter_privilege 
-----
 t
(1 row)
```



# 新增参数权限系统表

## □ 新增pg\_parameter\_acl系统表

```
postgres=# select parname,paracl from pg_parameter_acl ;
parname | paracl
-----+-----
log_checkpoints | {postgres=sA/postgres,test=s/postgres}
archive_mode | {postgres=sA/postgres,test=A/postgres}
(2 rows)
```



# 自定义变量限制

- 已安装扩展插件名称不允许作为自定义变量前缀，否则在扩展插件加载时会自动删除该变量

```
psql (15beta1)
Type "help" for help.

postgres=# show postgres_fdw.t1;
 postgres_fdw.t1
-----
 abc
(1 row)

postgres=# select * from remote_t1 ;
WARNING:  invalid configuration parameter name "postgres_fdw.t1", removing it
DETAIL:  "postgres_fdw" is now a reserved prefix.
 id | val
----+-----
(0 rows)

postgres=# show postgres_fdw.t1;
ERROR:  unrecognized configuration parameter "postgres_fdw.t1"
postgres=#
postgres=# set postgres_fdw.t1 to 'aaa';
ERROR:  invalid configuration parameter name "postgres_fdw.t1"
DETAIL:  "postgres_fdw" is a reserved prefix.
```



# 02

## 开发相关







# public模式安全增强

- ❑ public模式create权限默认被回收
- ❑ public模式owner变为pg\_database\_owner

```
psql (14.2)
Type "help" for help.

postgres=# \dn+ public

      List of schemas
  Name | Owner | Access privileges | Description
-----+-----+-----+-----
 public | postgres | postgres=UC/postgres+ | standard public schema
      |      | =UC/postgres |
```

← PostgreSQL v14

PostgreSQL v15 →

```
psql (15beta1)
Type "help" for help.

postgres=# \dn+ public

      List of schemas
  Name | Owner | Access privileges | Description
-----+-----+-----+-----
 public | pg_database_owner | pg_database_owner=UC/pg_database_owner+ | standard public schema
      |      | =U/pg_database_owner |
```



# NULL可设置是否允许重复

□ 唯一约束和索引可设置NULL值distinct/not distinct

## PostgreSQL v14使用部分表达式限制重复NULL

```
CREATE UNIQUE INDEX idx_abc ON abc ((a2 is null))  
WHERE a2 IS NULL;
```

```
postgres=# create table abc(a2 varchar(10));  
CREATE TABLE  
postgres=# CREATE UNIQUE INDEX idx_abc  
postgres-# ON abc ((a2 is null)) WHERE a2 IS NULL;  
CREATE INDEX  
postgres=# insert into abc values(null);  
INSERT 0 1  
postgres=# insert into abc values(null);  
ERROR:  duplicate key value violates unique constraint "idx_abc"  
DETAIL:  Key ((a2 IS NULL))=(t) already exists.
```

## PostgreSQL v15

```
UNIQUE NULLS DISTINCT  
UNIQUE NULLS NOT DISTINCT
```

```
postgres=# create table abc (  
postgres(# a2 varchar(10),  
postgres(# unique NULLS NOT DISTINCT (a2)  
postgres(# );  
CREATE TABLE  
postgres=# insert into abc values(null);  
INSERT 0 1  
postgres=# insert into abc values(null);  
ERROR:  duplicate key value violates unique constraint "abc_a2_key"  
DETAIL:  Key (a2)=(null) already exists.
```

## □ 数据库日志增加jsonlog格式

```
{
  "timestamp":"2022-04-20 10:24:47.710 CST",
  "pid":3383,
  "session_id":"625f67e3.d37",
  "line_num":12,
  "session_start":"2022-04-20 09:54:43 CST",
  "txid":0,
  "error_severity":"LOG",
  "message":"checkpoint complete: wrote 4 buffers (0.0%); 0 WAL file(s)
added, 0 removed, 0 recycled; write=0.303 s, sync=0.135 s, total=0.809
s; sync files=4, longest=0.127 s, average=0.034 s; distance=0 kB,
estimate=114 kB",
  "backend_type":"checkpointer",
  "query_id":0
}
```

## log\_destination= 'jsonlog'

```
{
  "timestamp":"2022-04-20 10:36:26.125 CST",
  "user":"postgres",
  "dbname":"postgres",
  "pid":3881,
  "remote_host":"[local]",
  "session_id":"625f71aa.f29",
  "line_num":1,
  "ps":"idle",
  "session_start":"2022-04-20 10:36:26 CST",
  "vxid":"3/2",
  "txid":0,
  "error_severity":"LOG",
  "message":"statement: select\n  relname,\n  relkind\nfrom\npg_class\nlimit 1;",
  "application_name":"psql",
  "backend_type":"client backend",
  "query_id":0
}
```



# jsonlog与日志分析

## □ 使用file\_fdw映射本地日志文件

```
create foreign table pg_log_5(  
  jsonstr text  
  ) server local_file_server  
  options (program 'cat /opt/pgdata1500/log/pg_log_5.json |jq -cMR');
```

## □ 使用物化视图及json构造函数转换json类型

```
create materialized view mv_pg_log_5  
as select json(trim(jsonstr,'')) as jsonlog from pg_log_5;
```

## □ 使用json函数进行日志分析

```
select * from mv_pg_log_5 where jsonlog is not json;  
select distinct json_query(jsonlog::jsonb,'$.error_severity')  
from mv_pg_log_5;
```

<https://www.modb.pro/db/411473>



# Merge into命令

## □ 新增Merge命令根据单表匹配修改另一个表(insert\update\delete)

匹配则更新，不匹配插入

```
merge into a_merge a using (select b.aid,b.name,b.year from b_merge b) c on (a.id=c.aid)
when matched then update set year=c.year when not matched then insert
values(c.aid,c.name,c.year);
```

仅匹配更新

```
merge into a_merge a using (select b.aid,b.name,b.year from b_merge b) c on (a.id=c.aid)
when matched then
update set year=c.year;
```

不匹配插入

```
merge into a_merge a using (select b.aid,b.name,b.year from b_merge b) c on (a.id=c.aid)
when not matched then
insert values(c.aid,c.name,c.year);
```

二次匹配

```
merge into a_merge a using (select b.aid,b.name,b.year,b.city from b_merge b) c on
(a.id=c.aid)
when matched and c.city != '江西' then
update set name=c.name
when not matched and c.city = '江西' then
insert values(c.aid,c.name,c.year);
```



# Numeric数据类型增强

PostgreSQL v14 → Oracle的number(p, s) 类型，根据精度(p)和小数位数(s)的不同，分别对应多种情况。

□ 当小数位数 (s) > 0时

- $0 < s \leq p$  推荐numeric(p,s)

- $p < s$  推荐numeric(s,s)

□ 当小数位数 (s) < 0时

- 当  $p + |s| \leq 4$  时，使用integer或者smallint

- 当  $4 < p + |s| \leq 9$  时，使用integer

- 当  $9 < p + |s| \leq 18$  时，使用bigint

- 当  $p + |s| > 18$  时，使用numeric(p + |s|, 0)

PostgreSQL v15 →

- numeric(p, s) 类型  
scale可以是负数  
或者比precision大

```
psql (15beta1)
Type "help" for help.

postgres=# select '1234'::numeric(4,-2);
 numeric
-----
      1200
(1 row)
```



# 03

## 并行特性





# v15之前并行特性

	14	13	12	11	10	9.6	9.5
Parallel query execution on remote databases	Yes	No	No	No	No	No	No
Foreign data wrapper query parallelism	Yes	No	No	No	No	No	No
Query parallelism for RETURN QUERY	Yes	No	No	No	No	No	No
Parallelized VACUUM for Indexes	Yes	Yes	No	No	No	No	No
Parallelized CREATE INDEX for B-tree indexes	Yes	Yes	Yes	Yes	No	No	No
Parallel hash joins	Yes	Yes	Yes	Yes	No	No	No
Parallel B-tree index scans	Yes	Yes	Yes	Yes	Yes	No	No
Parallel bitmap heap scans	Yes	Yes	Yes	Yes	Yes	No	No
Parallel merge joins	Yes	Yes	Yes	Yes	Yes	No	No
Parallel full table scans (sequential scans)	Yes	Yes	Yes	Yes	Yes	Yes	No
Parallel JOIN, aggregate	Yes	Yes	Yes	Yes	Yes	Yes	No
Parallel query	Yes	Yes	Yes	Yes	Yes	Yes	No
Parallel vacuumdb jobs	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel restore	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel pg_dump	Yes	Yes	Yes	Yes	Yes	Yes	Yes



## □ 新增distinct操作并行

```
postgres=# select version();
               version
-----
PostgreSQL 15beta1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36), 64-bit
(1 row)

postgres=# explain (costs off) select distinct a from t1;
               QUERY PLAN
-----
Unique
->  Sort
    Sort Key: a
    ->  Gather
        Workers Planned: 2
        ->  HashAggregate
            Group Key: a
            -> Parallel Seq Scan on t1
(8 rows)
```



# postgres\_fdw下推及并行提交

## □ push down CASE expressions

```
psql (14.2)
Type "help" for help.

postgres=# explain (verbose, costs off)
postgres=# select * from remote_t1
postgres=# where id >
postgres=#       case
postgres=#         when val > 50 then 0
postgres=#         else 75
postgres=#       end;
                    QUERY PLAN
-----
Foreign Scan on public.remote_t1
Output: id, val
Filter: (remote_t1.id > CASE WHEN (remote_t1.val > 50) THEN 0 ELSE 75
5 END)
Remote SQL: SELECT id, val FROM public.local_t1
```

```
psql (15devel)
Type "help" for help.

postgres=# explain (verbose, costs off)
postgres=# select * from remote_t1
postgres=# where id >
postgres=#       case
postgres=#         when val > 50 then 0
postgres=#         else 75
postgres=#       end;
                    QUERY PLAN
-----
Foreign Scan on public.remote_t1
Output: id, val
Remote SQL: SELECT id, val FROM public.local_t1 WHERE ((id > (CASE W
HEN (val > 50) THEN 0 ELSE 75 END)))
(3 rows)
```

## □ 支持parallel commit

```
postgres=# create server local_pg_server2 foreign data wrapper postgres_fdw
options ( port '1500', dbname 'postgres' , parallel_commit 'true');
CREATE SERVER
```



# 服务端工具并行及压缩增强

- ❑ pg\_basebackup支持对LZ4和Zstandard算法的压缩及解压
- ❑ pg\_basebackup支持--compress选项控制压缩算法  
`--compress=[{client|server}-]METHOD[:DETAIL]`
- ❑ pg\_basebackup文本格式支持服务端压缩后客户端存储前进行解压。
- ❑ pg\_receivewal支持--compress选项以及LZ4压缩



# 服务端工具并行及压缩增强

## □ PostgreSQL 15 pg\_basebackup服务端压缩并行备份对比

Compression	Size -Ft (GB)	Time -Ft	Time -Fp
none	3.8	17.0	18.1
gzip	1.5	234.9	233.1
lz4	2.0	31.5	35.1
zstd	1.3	56.1	59.1
zstd:workers=2	1.3	26.4	29.5
zstd:workers=4	1.3	15.0	21.8
zstd:workers=6	1.3	11.5	23.0
zstd:workers=8	1.3	9.9	22.9
zstd:workers=12	1.3	10.3	20.3
zstd:workers=16	1.3	10.0	20.5
zstd:workers=20	1.3	10.2	20.3
zstd:workers=24	1.3	10.1	21.0

-Ft/-Fp --compress=none

-Ft/-Fp --compress=server-gzip

-Ft/-Fp --compress=server-lz4

-Ft/-Fp --compress=server-zstd

-Ft/-Fp --compress=server-zstd:2

<http://rhaas.blogspot.com/2022/05/parallel-server-side-backup-compression.html>





感谢聆听!