

# 第九届PostgreSQL中国技术大会

## 2019 PostgreSQL Conference China

开源驱动 自主研发

主办:  PostgreSQL中文社区

协办: 

⌚ 2019年11月29日-30日

◎ 北京维景国际大酒店



# 第九届PostgreSQL中国技术大会

2019 PostgreSQL Conference China

开源驱动 自主研发

⌚ 2019年11月29日-30日

◎ 北京维景国际大酒店

主办 :  PostgreSQL中文社区

协办 :  ITpub

## 80%的问题

---

CPU高

IO高

SQL慢

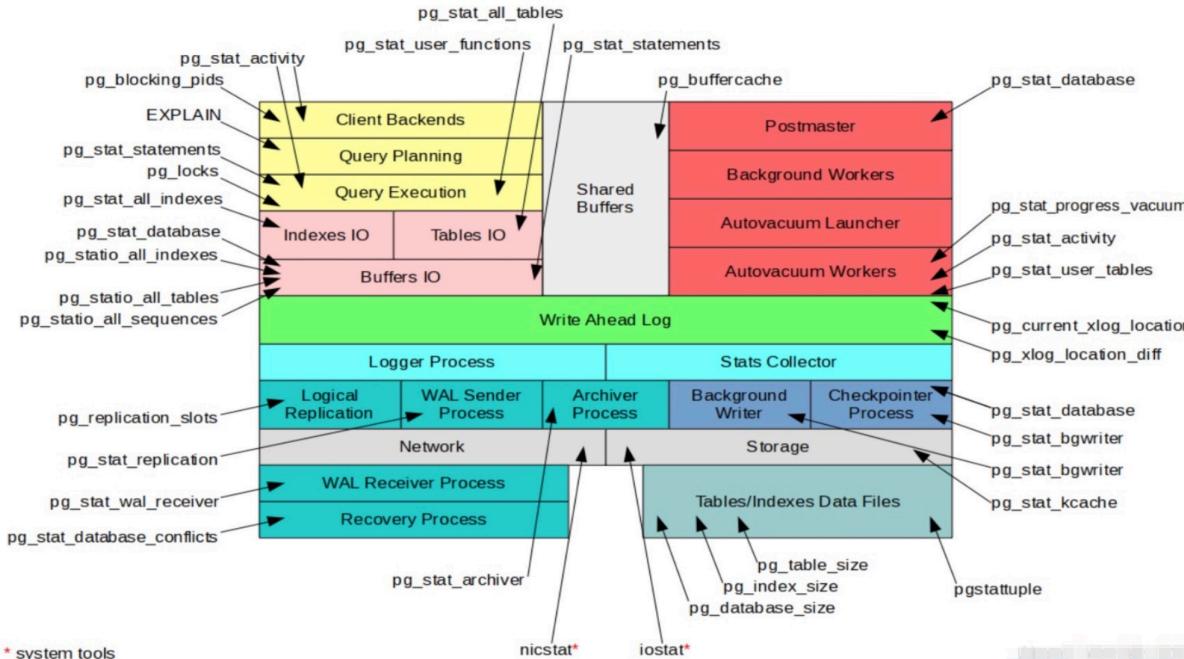
## 解决问题 – CPU相关命令

- top
- mpstat
- pidstat
- perf

## 解决问题 – IO相关命令

- iostat
- iotop
- pidstat

## 解决问题 – 数据库相关视图



# 解决问题 – 执行计划

```
EXPLAIN SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
        QUERY PLAN
```

执行计划代价

```
Hash Join (cost=540.00..439429.44 rows=10003825 width=27)
  Hash Cond: (m.receiver_id = u2.id)
    -> Hash Join (cost=270.00..301606.84 rows=10003825 width=23)
      Hash Cond: (m.sender_id = u1.id)
        -> Seq Scan on messages m (cost=0.00..163784.25 rows=10003825)
        -> Hash (cost=145.00..145.00 rows=10000 width=4)
          -> Seq Scan on users u1 (cost=0.00..145.00 rows=10000 width=4)
    -> Hash (cost=145.00..145.00 rows=10000 width=4)
      -> Seq Scan on users u2 (cost=0.00..145.00 rows=10000 width=4)
```

行数评估

## 10%的问题

网络类问题

如何分析和定位



- 1.了解TCP原理
- 2.抓包、wireshark分析
- 3.了解PG的前端协议



## 【案例】网络延迟不同插入性能差异很大

```
time=0.951 ms
time=0.951 ms
time=0.953 ms
time=0.948 ms
time=0.960 ms
time=0.957 ms
time=0.953 ms
time=0.961 ms
time=0.961 ms

istics ---
oss, time 26824ms
s
```

```
- PointManager save points size=182, cost = 519
- PointManager save points size=1001, cost = 3630
- PointManager save points size=1001, cost = 3297
- PointManager save points size=1001, cost = 3488
- PointManager save points size=1001, cost = 2843
```

Ping延迟1ms , 1000条插入耗时4s

## 【案例】网络延迟不同插入性能差异很大

```
; t
[]
[]
[0]
[]
[]

; free -m
shared  buff/cache   available
      0        1622       1486

; ping rdspglocation.pg.rds.aliyuncs.com
.25.231) 56(84) bytes of data.
lcmp_seq=1 ttl=102 time=0.080 ms
lcmp_seq=2 ttl=102 time=0.071 ms
lcmp_seq=3 ttl=102 time=0.087 ms
lcmp_seq=4 ttl=102 time=0.088 ms

save points size=1001, cost = 562
save points size=1001, cost = 539
` save points size=1001, cost = 569
save points size=919, cost = 547
save points size=1001, cost = 584
```

Ping延迟0.1ms，1000条插入耗时500ms



## 【案例】网络延迟不同插入性能差异很大

tcpdump -i eth0 port 3433 -s 0 -w t.cap

No.	Time	d	Source	Destination	Protocol	Length	Info
96874	2019-03-21 17:18:10.483756	0.000200s	[REDACTED]	[REDACTED]	PGSQL	108 >P/B/D/E/S	
96876	2019-03-21 17:18:10.485492	0.001736s	[REDACTED]	[REDACTED]	TCP	56 3433 → 43560 [ACK] Seq=642993 Ack=334222 Win=5247 Len=0	
97217	2019-03-21 17:18:11.070551	0.585059s	[REDACTED]	[REDACTED]	PGSQL	132 <1/2/T/D/C/Z	
97221	2019-03-21 17:18:11.075424	0.004873s	[REDACTED]	[REDACTED]	PGSQL	108 >P/B/D/E/S	
97223	2019-03-21 17:18:11.077150	0.001726s	[REDACTED]	[REDACTED]	TCP	56 3433 → 43560 [ACK] Seq=643069 Ack=334274 Win=5247 Len=0	
97334	2019-03-21 17:18:11.337959	右击 解码 0	[REDACTED]	[REDACTED]	PGSQL	132 <1/2/T/D/C/Z	
97335	2019-03-21 17:18:11.338059	0	[REDACTED]	[REDACTED]	PGSQL	61 >S	
97336	2019-03-21 17:18:11.339866	0	[REDACTED]	[REDACTED]	TCP	56 3433 → 43560 [ACK] Seq=643145 Ack=334279 Win=5247 Len=0	
97341	2019-03-21 17:18:11.369750	0	[REDACTED]	[REDACTED]	PGSQL	62 <Z	
97342	2019-03-21 17:18:11.369802	0	[REDACTED]	[REDACTED]	PGSQL	192 >B/E/S	
97344	2019-03-21 17:18:11.371531	0	[REDACTED]	[REDACTED]	TCP	56 3433 → 43560 [ACK] Seq=643151 Ack=334279 Win=5247 Len=0	

Frame 97223: 56 bytes on wire (448 bits),  
Linux cooked capture  
Internet Protocol Version 4, Src: 172.16.  
Transmission Control Protocol, Src Port:

作为过滤器应用  
准备过滤器  
对话过滤器  
对话着色  
SCTP  
追踪流



## 【案例】网络延迟不同插入性能差异很大

B代表bind， d代表Describe， e代表execute

Frame	Timestamp	Source	Destination	Type	Length
38800	5.114633			PGSQL	7254
38801	5.114643			PGSQL	1016
38802	5.114699			PGSQL	5814
38803	5.115865			TCP	60
38810	5.115874			TCP	3433
38812	5.115877			TCP	3433
38815	5.115905			PGSQL	10134
38826	5.117374			TCP	3433
38827	5.117393			PGSQL	20214
38837	5.118604			TCP	3433
38844	5.119786			TCP	3433
38847	5.119804			PGSQL	23094
38855	5.121153			TCP	3433
38856	5.121172			PGSQL	11574
38857	5.121183			PGSQL	13014
38864	5.122462			TCP	3433
38865	5.122480			PGSQL	25974
38875	5.123852			TCP	3433

```
Frame 38800: 7254 bytes on wire (58032 bits), 7254 bytes captured (58032 bits)
  ► Ethernet II, Src: Xensource [00:16:3e:0a:bc:89], Dst: PostgreSQL [00:00:00:00:00:00] (Broadcast)
  ► Internet Protocol Version 4, Src Port: 26008, Dst Port: 5432, Seq: 53222, Ack: 184984, Len: 7200
  ► Transmission Control Protocol, Src Port: 26008, Dst Port: 3433, Seq: 53222, Ack: 184984, Len: 7200
    Type: Bind
    Length: 656
```



**【案例】** 网络延迟不同插入性能差异很大

C代表command complete



## 【案例】网络延迟不同插入性能差异很大

结论	解法
单行插入1条网络消耗时间就达到2ms，1000条insert语句，消耗在网络的rt时间就达到2s。	使用insert into values(),(),()的方式批量进行发送，避免网络的多次交互。



## 10%的问题

数据库内核  
相关问题

需要的技能

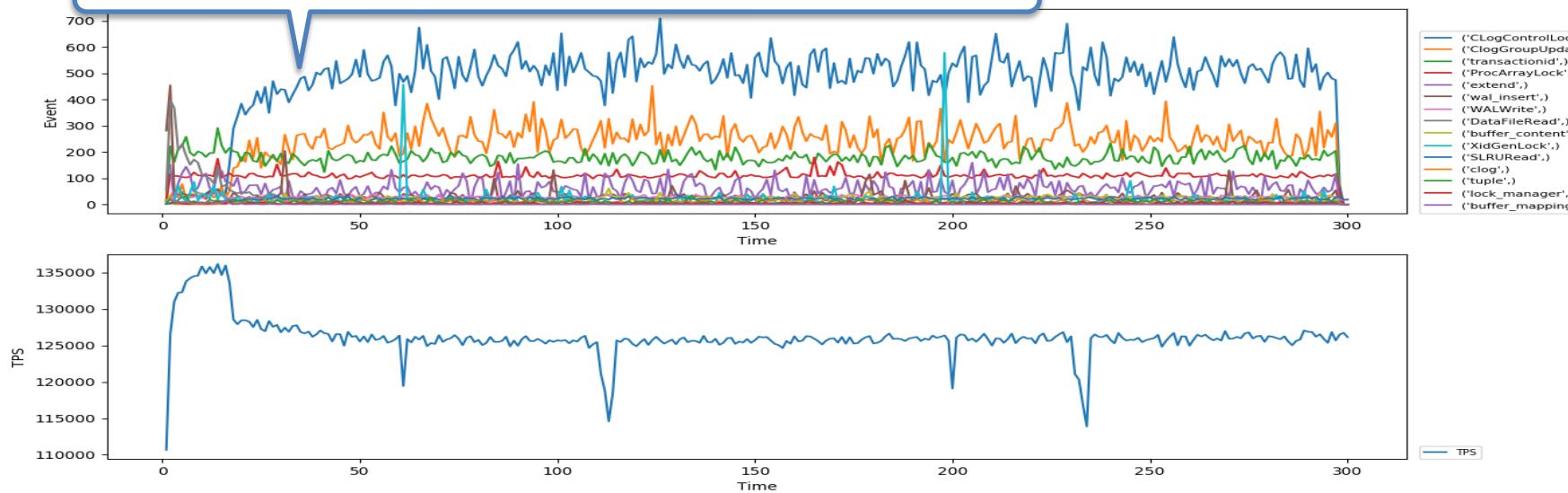
1.能够阅读内核代码

2.gdb,pstack,systemtap,  
perf等



## 【案例】vacuum为何造成TPS抖动

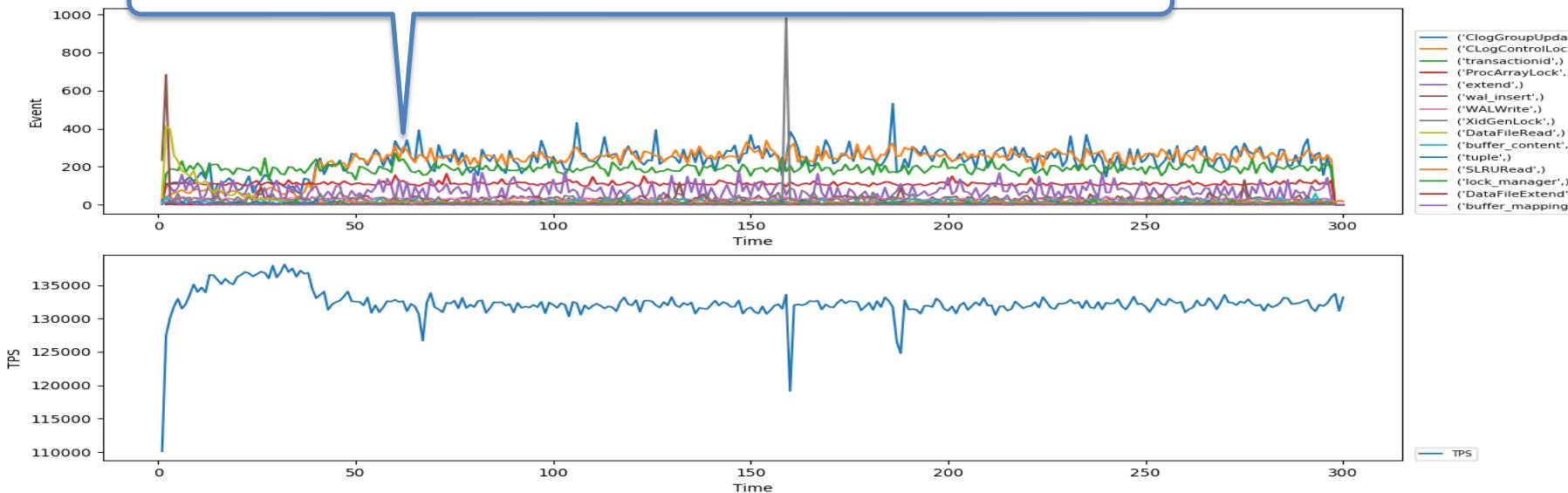
ClogControlLock的争抢是产生TPS抖动的原因





## 【案例】vacuum为何造成TPS抖动

调整clog buffer大小后，ClogControlLock的争抢明显降低



## 【案例】内存异常增长

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1539	root	20	0	2132m	378m	7652	S	136.8	0.1	94525:34	/usr/local/rds/bifrost_ag/bifrost_rt_uploader -c
41993	pg250772	20	0	66.0g	39g	6.6g	R	99.9	7.8	3:54.51	postgres: 1st pga [56327] INSERT

1条insert语句，用了30多GB内存？



## 【案例】内存异常增长

```
gdb -p pid
```

```
(gdb) p MemoryContextStats(TopMemoryContext)
```

```
TopMemoryContext: 2952216 total in 150 blocks; 264624 free (15 chunks); 2687592 used
pgstat TabStatusArray lookup hash table: 24576 total in 2 blocks; 2432 free (3 chunks);
22144 used
TopTransactionContext: 8192 total in 1 blocks; 7960 free (1 chunks); 232 used
    TopTransactionContextResetAtSPLCommit: 0 total in 0 blocks; 0 free (0 chunks); 0 used
Record information cache: 24576 total in 2 blocks; 15072 free (5 chunks); 9504 used
Function stat entries: 24576 total in 2 blocks; 15056 free (4 chunks); 9520 used
TableSpace cache: 8192 total in 1 blocks; 2312 free (0 chunks); 5880 used
Type information cache: 24488 total in 2 blocks; 2840 free (0 chunks); 21648 used
Operator lookup cache: 24576 total in 2 blocks; 10976 free (5 chunks); 13600 used
RowDescriptionContext: 8192 total in 1 blocks; 7112 free (0 chunks); 1080 used
MessageContext: 2097152 total in 9 blocks; 535280 free (1 chunks); 1561872 used
Operator class cache: 8192 total in 1 blocks; 776 free (0 chunks); 7416 used
smgr relation table: 122880 total in 4 blocks; 57776 free (15 chunks); 65104 used
TransactionAbortContext: 32768 total in 1 blocks; 32728 free (0 chunks); 40 used
Portal hash: 8192 total in 1 blocks; 776 free (0 chunks); 7416 used
PortalMemory: 8192 total in 1 blocks; 7880 free (0 chunks); 312 used
    PortalHeapMemory: 1024 total in 1 blocks; 816 free (0 chunks); 208 used
    ExecutorState: 516096 total in 6 blocks; 90376 free (0 chunks); 425720 used
        printtup: 0 total in 0 blocks; 0 free (0 chunks); 0 used
        ExprContext: 0 total in 0 blocks; 0 free (0 chunks); 0 used
        ExprContext: 0 total in 0 blocks; 0 free (0 chunks); 0 used
    .....
ExprContext: 0 total in 0 blocks; 0 free (0 chunks); 0 used
    ExprContext: 36754817256 total in 1182406 blocks; 18814651376 free (5829232 chunks);
17940165880 used
        ExprContext: 0 total in 0 blocks; 0 free (0 chunks); 0 used
        ExprContext: 8192 total in 1 blocks; 5680 free (2 chunks); 2512 used
    Relcache by OID: 1040384 total in 7 blocks; 471976 free (14 chunks); 568408 used
CacheMemoryContext: 42814696 total in 27 blocks; 3960632 free (0 chunks); 38854064 used
```

可以看出在ExecutorState  
过程中ExprContext占用了  
大量的内存

## 【案例】内存异常增长

查看执行计划，发现  
是hashagg

```
pgsql=# explain SELECT id, now(), jsonb_object_agg(group_id, exe_id) FROM profile_tag_supplier_lead_1 GROUP BY id;
QUERY PLAN
HashAggregate  (cost=2045882.39..2045885.39 rows=200 width=48)
  Group Key: profile_tag_supplier_lead_1.id
  -> Append  (cost=0.00..1588274.59 rows=91521560 width=24)
      -> Seq Scan on profile_tag_supplier_lead_1 (cost=0.00..0.00 rows=1 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g75960 (cost=0.00..3361.91 rows=193691 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g75984 (cost=0.00..81.75 rows=4675 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g76008 (cost=0.00..81.75 rows=4675 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g74016 (cost=0.00..8.28 rows=428 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g74024 (cost=0.00..21930.54 rows=1263754 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g70912 (cost=0.00..75.22 rows=4322 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g46672 (cost=0.00..3154.24 rows=181624 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g70928 (cost=0.00..2933.44 rows=169044 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g46696 (cost=0.00..16642.52 rows=958952 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g46688 (cost=0.00..3154.24 rows=181624 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g74040 (cost=0.00..79.77 rows=4577 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g70944 (cost=0.00..322.69 rows=18569 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g76072 (cost=0.00..22089.52 rows=1272952 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g74048 (cost=0.00..21399.31 rows=1233131 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g70984 (cost=0.00..76.60 rows=4360 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g74992 (cost=0.00..3386.38 rows=195138 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g78176 (cost=0.00..8.12 rows=412 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g78184 (cost=0.00..1223.97 rows=70497 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g72112 (cost=0.00..15.71 rows=871 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g72984 (cost=0.00..2199.13 rows=126713 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g73056 (cost=0.00..21935.00 rows=1264000 width=24)
      -> Seq Scan on profile_tag_supplier_lead_g76088 (cost=0.00..81.75 rows=4675 width=24)
```



## 【案例】内存异常增长

- 关闭HashAggregate后，查看执行计划

```
pgsql# set enable_hashagg =off;
SET
pgsql# explain SELECT id, now(), jsonb_object_agg(group_id, exe_id) FROM profile_tag_supplier_lead_1 GROUP BY id;
                                         QUERY PLAN
-----
GroupAggregate  (cost=84.31..6584902.68 rows=200 width=48)
  Group Key: profile_tag_supplier_lead_1.id
    -> Merge Append  (cost=84.31..6127291.88 rows=91521560 width=24)
        Sort Key: profile_tag_supplier_lead_1.id
          -> Sort  (cost=0.01..0.02 rows=1 width=24)
              Sort Key: profile_tag_supplier_lead_1.id
                -> Seq Scan on profile_tag_supplier_lead_1  (cost=0.00..0.00 rows=1 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g75960_id on profile_tag_supplier_lead_g75960  (cost=0.42..5017.78 rows=193691 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g75984_id on profile_tag_supplier_lead_g75984  (cost=0.28..125.41 rows=4675 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g76008_id on profile_tag_supplier_lead_g76008  (cost=0.28..125.41 rows=4675 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g74016_id on profile_tag_supplier_lead_g74016  (cost=0.27..14.69 rows=428 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g74024_id on profile_tag_supplier_lead_g74024  (cost=0.43..32802.74 rows=1263754 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g70912_id on profile_tag_supplier_lead_g70912  (cost=0.28..117.11 rows=4322 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g46672_id on profile_tag_supplier_lead_g46672  (cost=0.42..4707.78 rows=181624 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g70928_id on profile_tag_supplier_lead_g70928  (cost=0.42..4391.08 rows=169044 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g46696_id on profile_tag_supplier_lead_g46696  (cost=0.42..24069.70 rows=958952 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g46688_id on profile_tag_supplier_lead_g46688  (cost=0.42..4707.78 rows=181624 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g74040_id on profile_tag_supplier_lead_g74040  (cost=0.28..123.94 rows=4577 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g70944_id on profile_tag_supplier_lead_g70944  (cost=0.29..486.82 rows=18569 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g76072_id on profile_tag_supplier_lead_g76072  (cost=0.43..33013.71 rows=1272952 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g74048_id on profile_tag_supplier_lead_g74048  (cost=0.43..32047.39 rows=1233131 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g70984_id on profile_tag_supplier_lead_g70984  (cost=0.28..118.68 rows=4360 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g74992_id on profile_tag_supplier_lead_g74992  (cost=0.42..5045.49 rows=195138 width=24)
    -> Index Scan using uk_profile_tag_supplier_lead_g78176_id on profile_tag_supplier_lead_g78176  (cost=0.27..14.45 rows=412 width=24)
```

## 【案例】内存异常增长

- 内存使用明显降低

```
top - 23:48:58 up 442 days, 3:47, 2 users, load average: 2.96, 2.92, 2.64
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.0%us, 2.2%sy, 0.0%ni, 94.6%id, 0.1%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 529140580k total, 205389516k used, 323751064k free, 5120960k buffers
Swap: 0k total, 0k used, 0k free, 159875280k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
32643 pg250772 20 0 33.2g 8.0g 7.9g R 90.6 1.6 6:31.98 postgres: pg2507721 pga [local] EXPLAIN
```

```
> pg2507721@a69h13312:~ (ssh)
-> Index Scan using uk_profile_tag_supplier_lead_g46685_id on profile_tag_supplier_lead_g46685 (cost=0.42..4706.49 rows=181538 width=24)
-> Index Scan using uk_profile_tag_supplier_lead_g46709_id on profile_tag_supplier_lead_g46709 (cost=0.42..4706.49 rows=181538 width=24)
-> Index Scan using uk_profile_tag_supplier_lead_g70073_id on profile_tag_supplier_lead_g70073 (cost=0.28..125.41 rows=4675 width=24)
-> Index Scan using uk_profile_tag_supplier_lead_g70977_id on profile_tag_supplier_lead_g70977 (cost=0.43..28057.04 rows=1080974 width=24)
pga=# set enable_hashagg =on;
SET
pga=# explain analyze SELECT id, now(), jsonb_object_agg(group_id, exe_id) FROM profile_tag_supplier_lead_2 GROUP BY id;
^CCancel request sent
ERROR: canceling statement due to user request
pga=# set enable_hashagg =off;
SET
pga=# explain analyze SELECT id, now(), jsonb_object_agg(group_id, exe_id) FROM profile_tag_supplier_lead_2 GROUP BY id;
```

THANK  
S