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：ITPUB

2019年11月29日-30日

北京维景国际大酒店

80%的问题

CPU高

IO高

SQL慢

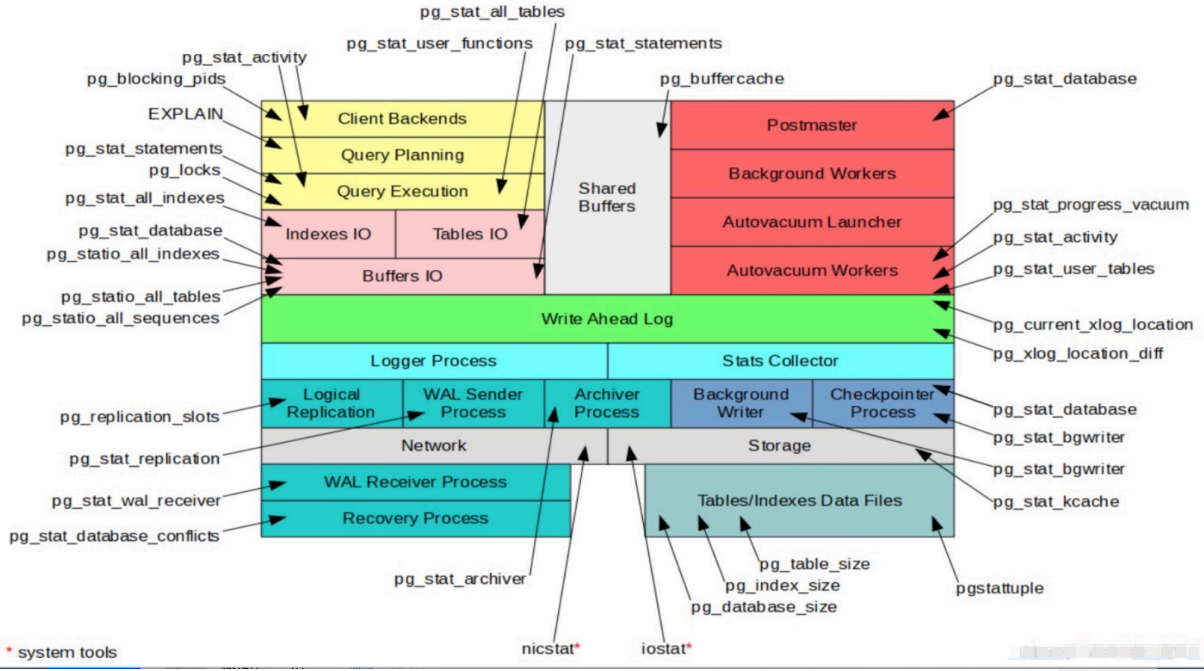
解决问题 – CPU相关命令

- 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

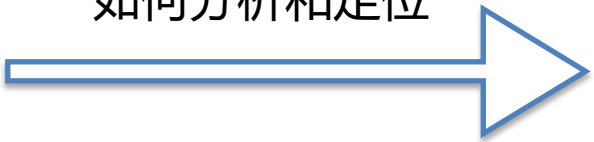
```
EXPLAIN SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
                                QUERY PLAN
-------------------------------------------------------------------------------------------
 Hash Join  (cost=540.00..439429.44 rows=10003825 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)
```

执行计划代价

行数评估

10%的问题

网络类问题



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

stics ---
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

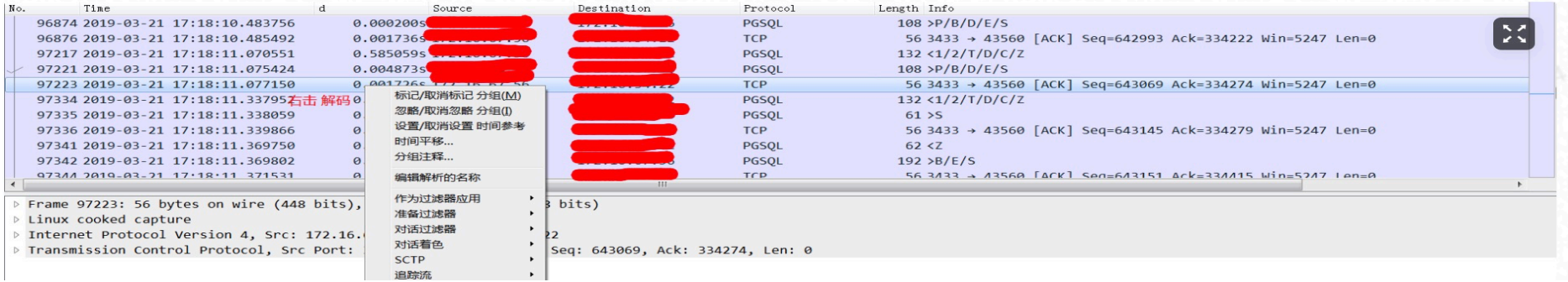
【案例】网络延迟不同插入性能差异很大

```
; t
2]
3]
10]
4]
7]
                                                                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

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

;  ping rdspglocation.pg.rds.aliyuncs.com
5.25.231) 56(84) bytes of data.
64 bytes from 100.25.25.231 (100.25.25.231): icmp_seq=1 ttl=102 time=0.080 ms
64 bytes from 100.25.25.231 (100.25.25.231): icmp_seq=2 ttl=102 time=0.071 ms
64 bytes from 100.25.25.231 (100.25.25.231): icmp_seq=3 ttl=102 time=0.087 ms
64 bytes from 100.25.25.231 (100.25.25.231): icmp_seq=4 ttl=102 time=0.088 ms
```

Ping延迟0.1ms，1000条插入耗时500ms

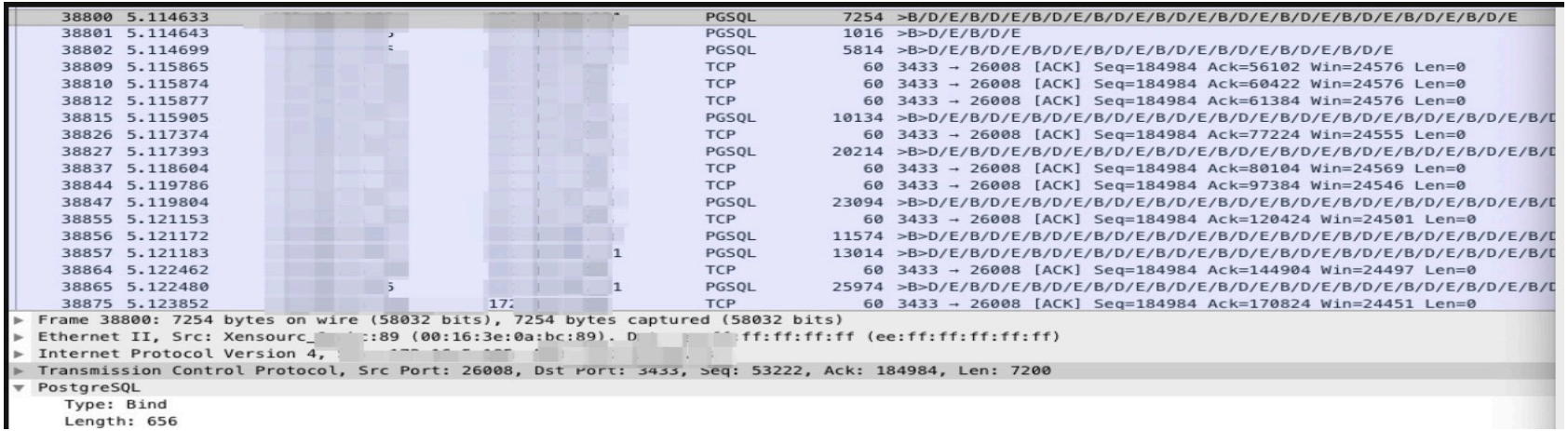
【案例】网络延迟不同插入性能差异很大

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



【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



【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

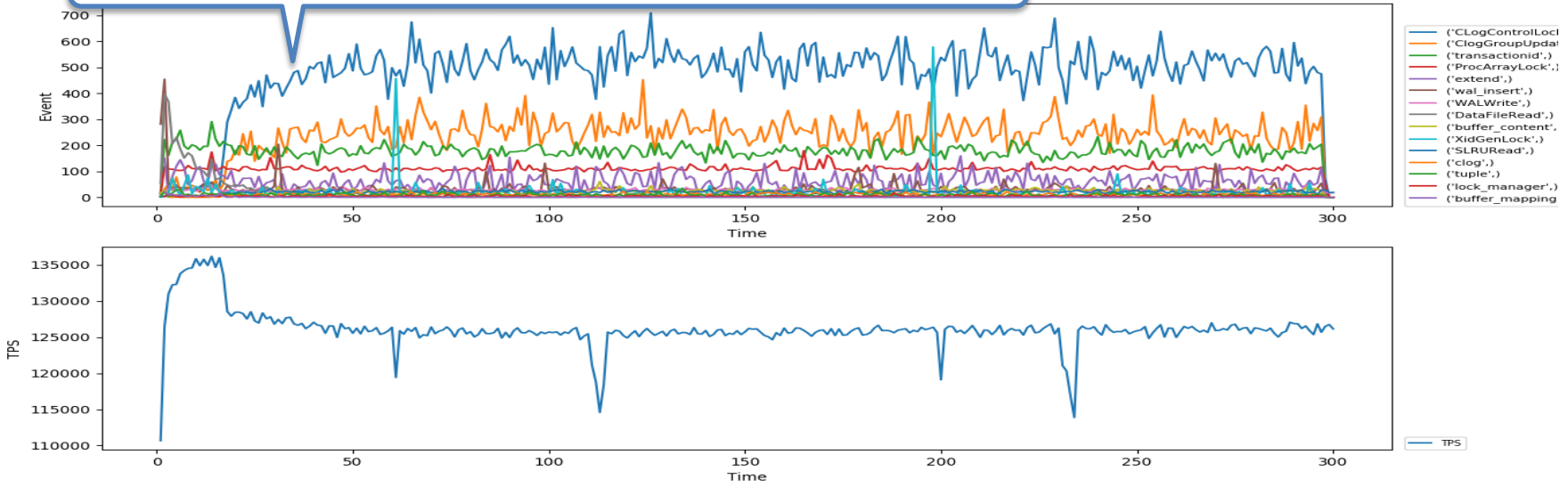
需要的技能

1.能够阅读内核代码

2.gdb,pstack,systemtap,perf等

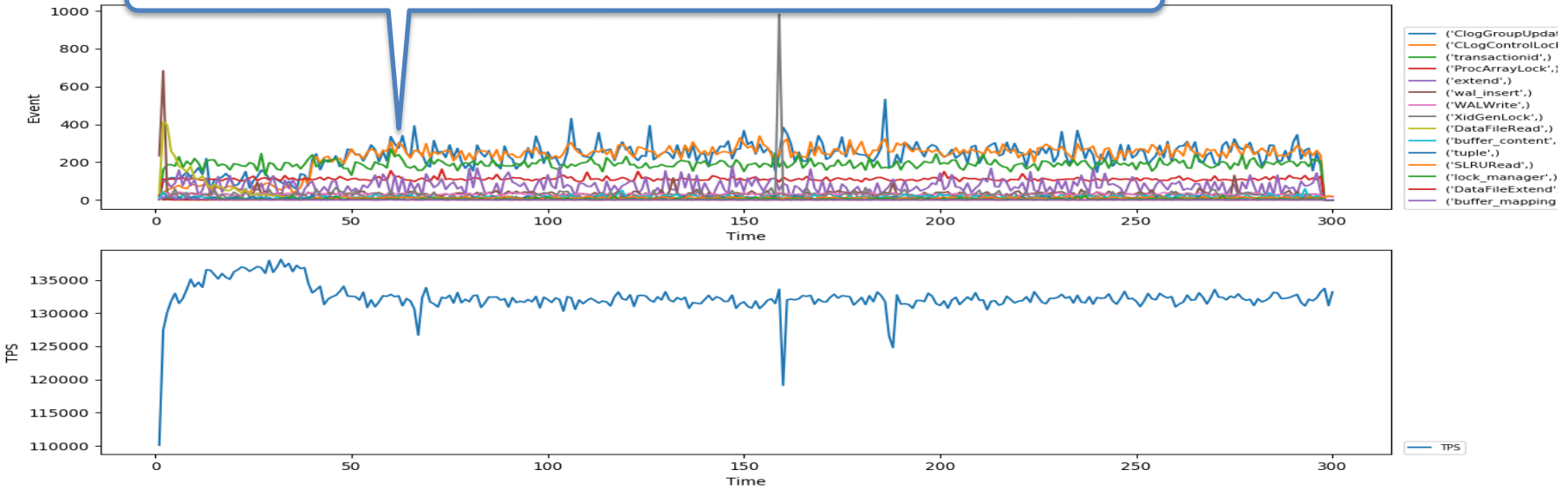
【案例】vacuum为何造成TPS抖动

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: lst pga [redacted]0[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过程下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

```
pga=# 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

```
pga=# set enable_hashagg =off;
SET
pga=# 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_g76073_id on profile_tag_supplier_lead_g76073  (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;
```

THANKS