



# Способы диагностики PostgreSQL

Владимир Бородин

Ильдус Курбангалиев


# Где взять презентацию?



[yadi.sk/d/KSIYFBiyhis2s](https://yadi.sk/d/KSIYFBiyhis2s)

# Взрыв мониторинга





Кто виноват?  
Что делать?

Р. Чернышевский и А. Герцен

# Во что упёрлись?

## Системные ресурсы:

- › Процессор/память/диски/сеть

## Внутренние процессы базы:

- › Heavy-weight locks
- › Всё остальное

# Процессор

```
1 [|||||199.0%]
2 [|||||198.4%]
3 [|||||199.5%]
4 [|||||199.5%]
5 [|||||197.4%]
6 [|||||197.9%]
7 [|||||198.9%]
8 [|||||197.4%]
9 [|||||198.5%]
10 [|||||199.5%]
11 [|||||199.5%]
12 [|||||199.0%]
13 [|||||199.0%]
14 [|||||199.5%]
15 [|||||199.5%]
16 [|||||199.0%]
17 [|||||198.4%]
18 [|||||199.5%]
19 [|||||197.4%]
20 [|||||199.0%]
21 [|||||198.4%]
22 [|||||199.5%]
23 [|||||199.0%]
24 [|||||199.0%]
25 [|||||1100.0%]
26 [|||||199.0%]
27 [|||||199.5%]
28 [|||||199.5%]
29 [|||||199.5%]
30 [|||||199.5%]
31 [|||||199.0%]
32 [|||||1100.0%]
Mem [|||||11836/129027MB]
Swp [|||||0/16377MB]

Tasks: 633 total, 187 running
Load average: 254.63 130.43 54.73
Uptime: 22 days, 22:19:17

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
27701 postgres 20 0 4650M 199M 193M S 12.0 0.2 0:27.96 postgres: mops mopsdb localhost(44563)
27829 postgres 20 0 4649M 172M 168M R 12.0 0.1 0:13.33 postgres: mops mopsdb localhost(50946)
```



# ДИСК

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sdb	18087.00	265.00	4208.00	199.00	87.09	1.81	41.31	2.40	0.55	0.14	60.10
sdc	17990.00	182.00	4144.00	172.00	86.46	1.38	41.68	2.73	0.63	0.16	68.50
sdd	18186.00	178.00	3942.00	165.00	86.44	1.34	43.77	2.26	0.55	0.15	61.50
sda	47857.00	249.00	4515.00	86.00	204.46	1.32	91.60	14.37	3.09	0.22	100.00

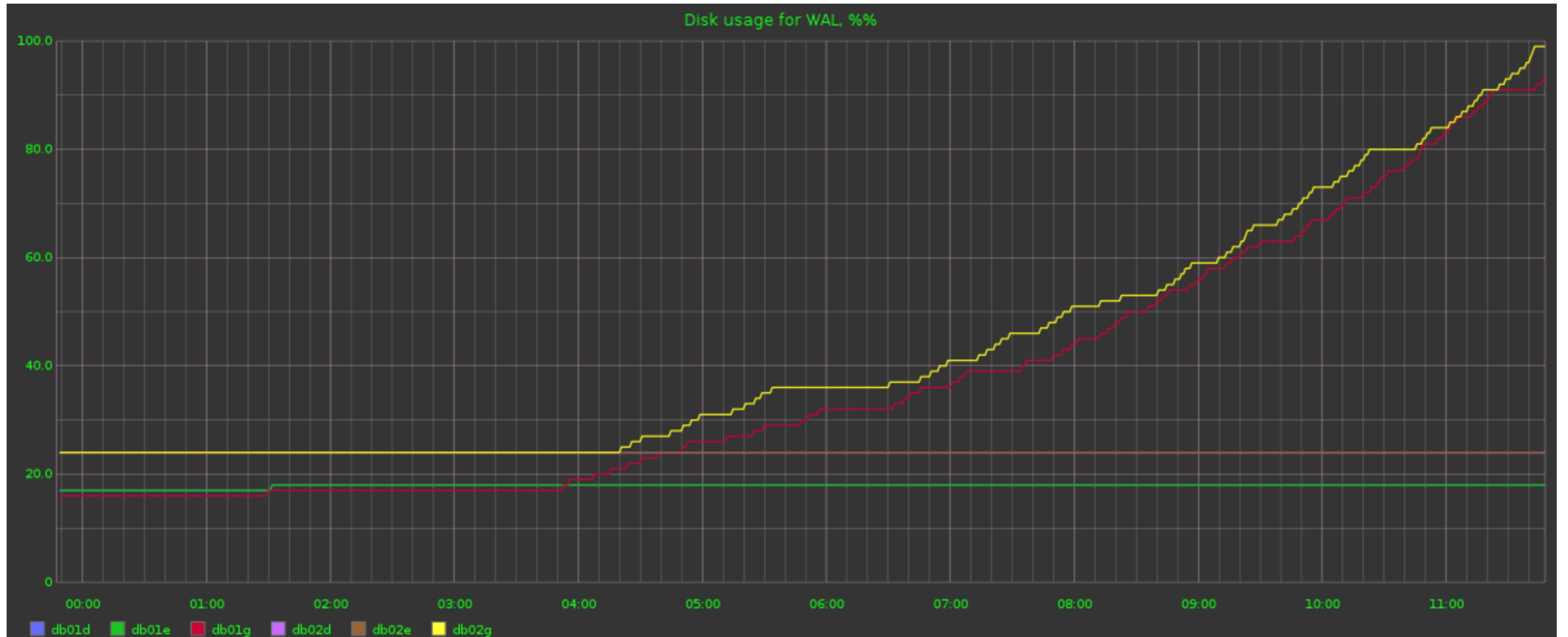
  

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sdb	12322.00	257.00	1996.00	219.00	55.93	1.86	53.43	1.15	0.52	0.17	37.60
sdc	39309.00	375.00	3106.00	201.00	165.20	2.25	103.70	3.86	1.17	0.26	84.40
sdd	12112.00	285.00	2128.00	292.00	55.62	2.25	48.98	1.37	0.57	0.17	42.20
sda	48884.00	377.00	3087.00	61.00	203.26	1.75	133.37	7.52	2.44	0.32	99.90

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sdb	2247.00	616.00	458.00	364.00	10.57	3.83	35.86	0.64	0.78	0.12	9.90
sdc	59343.00	322.00	2672.00	235.00	242.25	2.18	172.20	6.13	2.06	0.34	99.80
sdd	2151.00	212.00	432.00	345.00	10.09	2.18	32.33	0.37	0.48	0.12	9.50
sda	53644.00	673.00	2154.00	123.00	217.96	3.11	198.84	4.66	2.01	0.44	99.60

# Диск





# Сеть

smcdb01d.disk.yandex.net, itype=common



# Память

- › Swap
- › OOM

# Heavy-weight locks

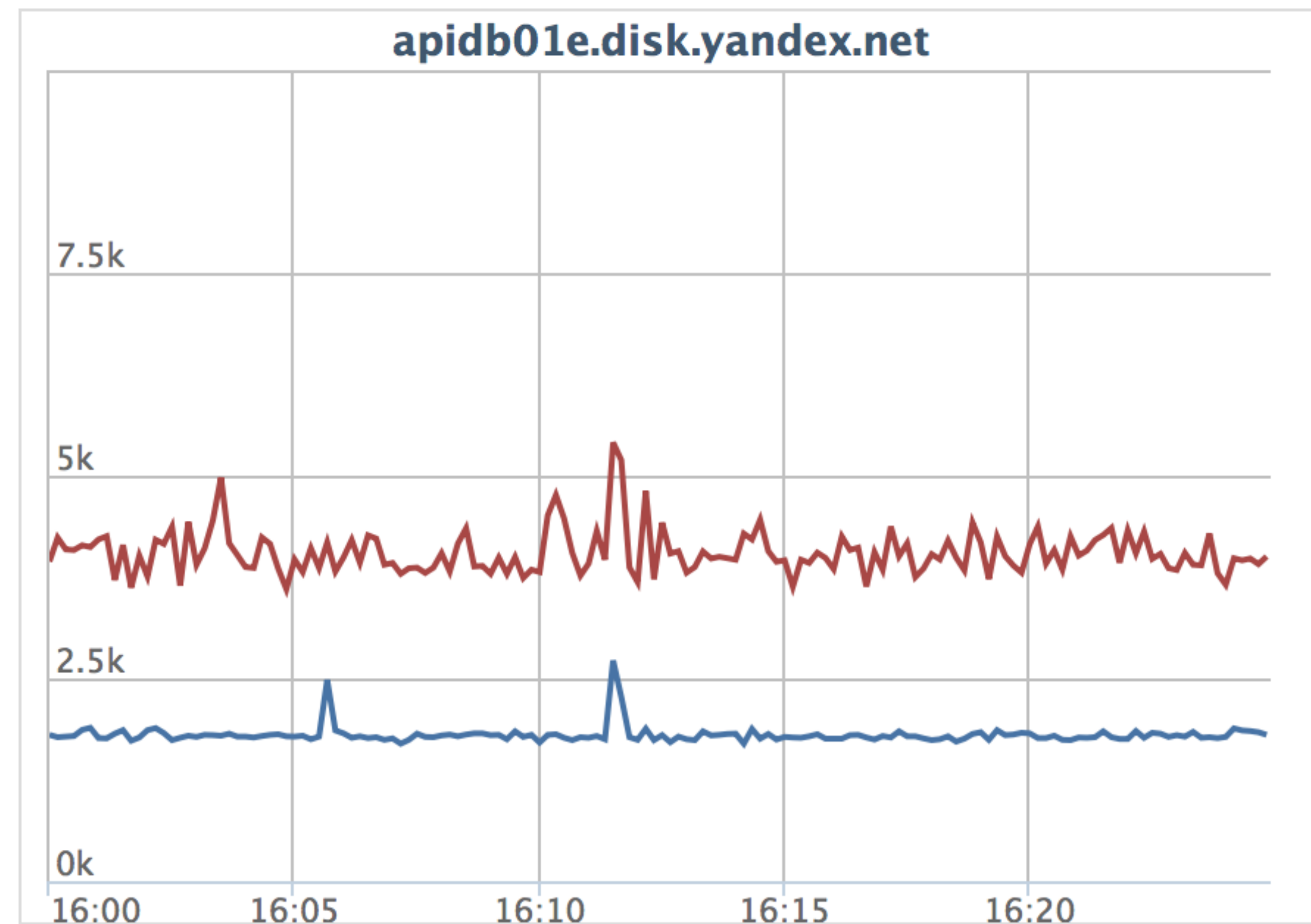
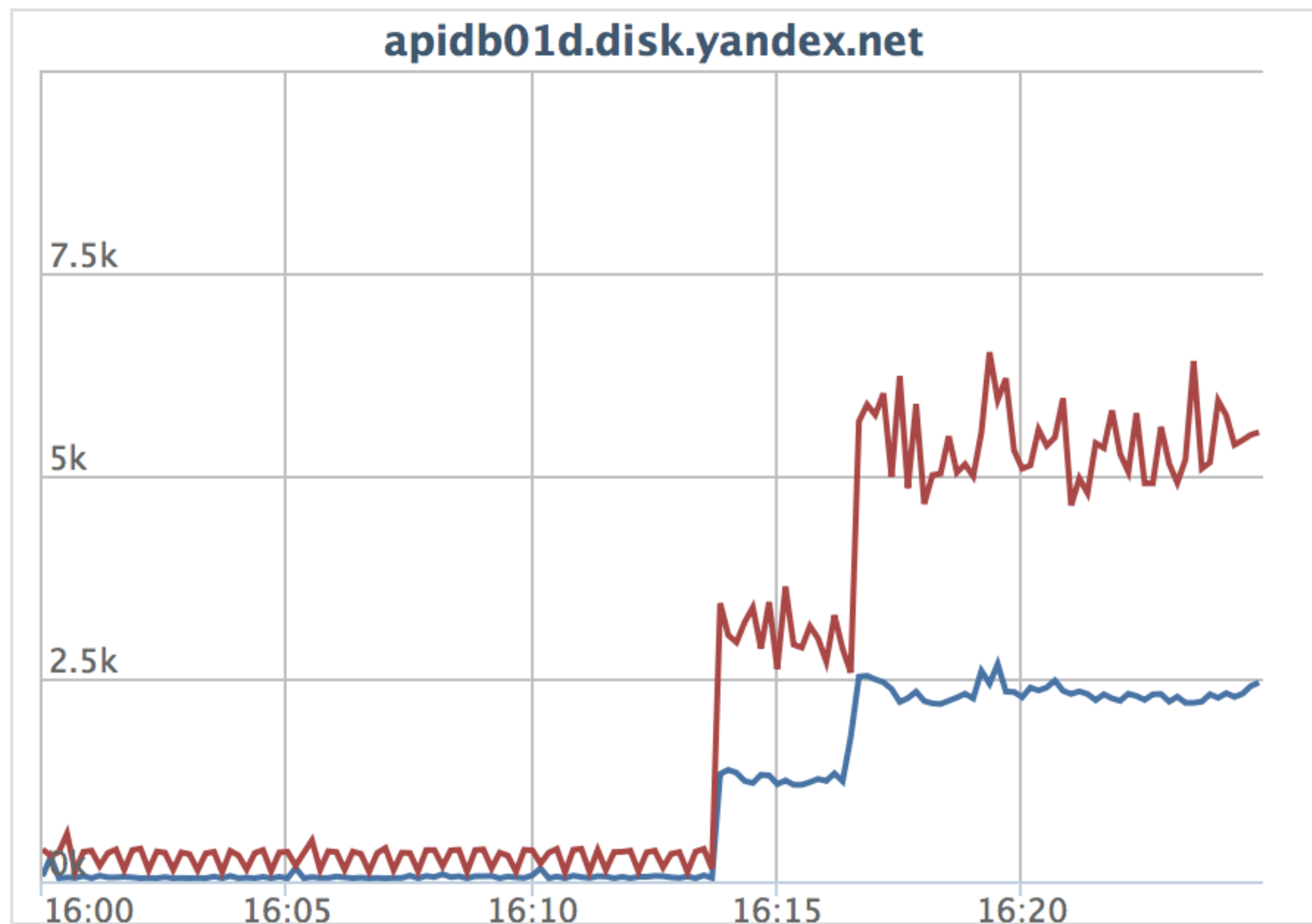
```
smcdb01g/postgres M # \i .sql/locks_flat.sql
-[ RECORD 1 ]-----+-----
waiting_locktype | tuple
waiting_table    | 17389
waiting_query    | SELECT * from databases_078 WHERE user_id = $1 AND app = $2 AND dbId = $3 FOR UPDATE
waiting_mode     | AccessExclusiveLock
waiting_pid      | 30617
other_locktype   | relation
other_table      | 17389
other_query      | UPDATE p_data_078 SET rev = $1, content = $2 WHERE handle = $3 AND collection_id = $4 AND record_id = $5
other_mode       | RowShareLock
other_pid        | 29912
other_granted    | t
-[ RECORD 2 ]-----+-----
waiting_locktype | tuple
waiting_table    | 17389
waiting_query    | SELECT * from databases_078 WHERE user_id = $1 AND app = $2 AND dbId = $3 FOR UPDATE
waiting_mode     | AccessExclusiveLock
waiting_pid      | 30617
other_locktype   | relation
other_table      | 17389
other_query      | SELECT * from databases_078 WHERE user_id = $1 AND app = $2 AND dbId = $3 FOR UPDATE
other_mode       | RowShareLock
other_pid        | 31215
```

# Heavy-weight locks

[https://wiki.postgresql.org/wiki/Lock\\_dependency\\_information](https://wiki.postgresql.org/wiki/Lock_dependency_information)

Vertical Mini Normalize values

disk\_dataapi\_db-tps



# pg\_stat\_statements

```
xivastore01e/xivadb # select md5(query), calls, total_time, rows, shared_blks_hit, shared_blks_read, (total_time/calls) as avg_time from pg_stat_statements order by avg_time desc limit 5;
```

md5	calls	total_time	rows	shared_blks_hit	shared_blks_read	avg_time
7e890be5ed231cb74a299b3e2e191361	91419	271908102.101001	91419	10643251	1907559	2974.30623941413
5dfd6d18b450b330f871e6ce73dd25f9	200393	275770221.584	200393	17478312	2224273	1376.14697910606
e1a1f6d1a8c094bba4531ee54e15f3e9	1	160.043	0	1	0	160.043
e2696fc881f13cadca929a90bae14bb2	1	108.132	49	960	0	108.132
a1176415f94a938456184c753b7f12d6	1	20.963	2	4	0	20.963

(5 rows)

Time: 0.724 ms

```
xivastore01e/xivadb # select query from pg_stat_statements where md5(query) in ('7e890be5ed231cb74a299b3e2e191361', '5dfd6d18b450b330f871e6ce73dd25f9');
```

```
-----  
SELECT n.local_id, n.event_dt  
FROM xiva.notifications n  
WHERE n.uid = i_uid  
AND n.sid = service_id  
AND n.hash = i_hash  
select local_id::int8, event_ts::int8, is_duplicate::bool from xiva.add_notification($1::int8,$2::text,$3::timestampz,$4::_text,$5::text,$6::int4,$7::bytea,$8::text)
```

(2 rows)

Time: 0.721 ms

```
xivastore01e/xivadb #
```



# pg\_stat\_kcache

```
rpopdb01d/ropopdb M # select queryid, system_time from pg_stat_kcache() order by system_time desc limit 5;
```

queryid	system_time
3325915325	803784.053420192
3564306043	33858.9768130009
3183006759	33788.4303960018
686754901	21555.116499599
3140366899	19707.7809633186

(5 rows)

Time: 8.484 ms

```
rpopdb01d/ropopdb M # select queryid, query from pg_stat_statements where queryid in (3325915325, 3564306043);
```

queryid	query
3325915325	select uidl from rpop.rpop_imap_uidls where folder_id=\$1::int8
3564306043	select r::text from rpop.uidl_list(\$1::int8) r

(2 rows)

Time: 22.389 ms

```
rpopdb01d/ropopdb M # █
```

```

pgload02e/postgres M # \! ls .sql
index_bloat.sql  io_reads_total.sql  io_writes_total.sql  locks_recursive.sql  queries_avg.sql  reset_stat.sql  view_blocking
_tree.sql
io_reads_avg.sql  io_writes_avg.sql  locks_flat.sql      memstat.sql          queries_total.sql  table_bloat.sql

```

```
pgload02e/postgres M # \i .sql/queries_total.sql
```

datname	queryid	total_time	calls	memory_hit	disk_read	disk_write	user_time	system_time
dbproto_jsonb	3967941575	205797351.37	30754846	4293 GB	937 GB	13 GB	4373.01	2515.12
dbproto_jsonb	3575574542	131458416.83	98507273	10219 GB	4202 GB	187 GB	19151.37	11634.62
dbproto_jsonb	3960329493	121142901.69	198725157	6430 GB	2674 GB	11 GB	14667.69	8866.33
dbproto_jsonb	3215553188	98258414.89	40380400	611 GB	2003 GB	18 GB	7906.49	5183.64
dbproto_jsonb	2161962844	68431661.37	31016354	539 GB	1724 GB	5549 MB	5856.81	3778.26
dbproto_jsonb	1977969162	51025947.56	262679876	9938 GB	1806 GB	12 GB	8801.76	5718.26
dbproto_jsonb	3700549953	42273303.10	85885159	1909 GB	1510 GB	3649 MB	6526.37	3316.94
dbproto_jsonb	892401191	33194657.70	31825565	3390 GB	1125 GB	32 GB	4329.29	2553.39
dbproto_jsonb	646519451	31921295.21	12286621	94 GB	0 bytes	0 bytes	597.53	743.97
dbproto_jsonb	2362146957	23397719.65	31825567	2141 GB	864 GB	29 GB	3024.83	1707.74

```
(10 rows)
```

```
Time: 6.803 ms
```

```
pgload02e/postgres M # SELECT query FROM pg_stat_statements WHERE queryid IN (3967941575, 3575574542);
                                query
```

```
-----
UPDATE hierarchy SET path=?, huniq=?, parent=? WHERE (_id=? AND uid=?);
INSERT INTO resources (uid, parent, hid, folder, _id, data, name) VALUES (?, ?, ?, ?, ?, ?, ?);
```

```
(2 rows)
```

```
Time: 4.058 ms
```

```
pgload02e/postgres M # █
```

```
SELECT datname, queryid, round(total_time::numeric, 2) AS total_time, calls,  
       pg_size_pretty((shared_blks_hit+shared_blks_read)*8192 - reads) AS memory_hit,  
       pg_size_pretty(reads) AS disk_read, pg_size_pretty(writes) AS disk_write,  
       round(user_time::numeric, 2) AS user_time, round(system_time::numeric, 2) AS  
system_time  
  
FROM pg_stat_statements s  
  
     JOIN pg_stat_kcache() k USING (userid, dbid, queryid)  
  
     JOIN pg_database d ON s.dbid = d.oid  
  
WHERE datname != 'postgres' AND datname NOT LIKE 'template%'  
  
ORDER BY total_time DESC LIMIT 10;
```



```
SELECT datname, queryid, round(total_time::numeric, 2) AS total_time, calls,  
       pg_size_pretty((shared_blks_hit+shared_blks_read)*8192 - reads) AS memory_hit,  
       pg_size_pretty(reads) AS disk_read, pg_size_pretty(writes) AS disk_write,  
       round(user_time::numeric, 2) AS user_time, round(system_time::numeric, 2) AS  
system_time  
  
FROM pg_stat_statements s  
     JOIN pg_stat_kcache() k USING (userid, dbid, queryid)  
     JOIN pg_database d ON s.dbid = d.oid  
  
WHERE datname != 'postgres' AND datname NOT LIKE 'template%'  
  
ORDER BY total_time DESC LIMIT 10;
```

```

pgload02g/dbproto_jsonb M # SELECT datname, queryid,
    pg_size_pretty(shared_hit) as shared_hit,
    pg_size_pretty(int8larger(0, (shared_read - reads))) as page_cache_hit,
    pg_size_pretty(reads) as physical_disk_read
FROM (
    SELECT userid, dbid, queryid,
        shared_blks_hit * 8192 as shared_hit,
        shared_blks_read * 8192 AS shared_read
    FROM pg_stat_statements
) s
JOIN pg_stat_kcache() k USING (userid, dbid, queryid)
JOIN pg_database d ON s.dbid = d.oid WHERE datname = 'dbproto_jsonb'
ORDER BY reads desc LIMIT 10;

```

datname	queryid	shared_hit	page_cache_hit	physical_disk_read
dbproto_jsonb	3960329493	73 GB	0 bytes	90 GB
dbproto_jsonb	2161962844	3101 MB	0 bytes	13 GB
dbproto_jsonb	3215553188	3053 MB	0 bytes	4950 MB
dbproto_jsonb	1977969162	88 GB	35 GB	3034 MB
dbproto_jsonb	4133035681	2072 kB	0 bytes	228 kB
dbproto_jsonb	855151222	48 kB	0 bytes	128 kB
dbproto_jsonb	2241815758	88 kB	0 bytes	80 kB
dbproto_jsonb	651713302	0 bytes	0 bytes	60 kB
dbproto_jsonb	1409572049	96 kB	8192 bytes	8192 bytes
dbproto_jsonb	30601438	2976 kB	0 bytes	0 bytes

(10 rows)

Time: 1.498 ms

pgload02g/dbproto\_jsonb M #



```

pgload02g/dbproto_jsonb M # SELECT datname, query,
    shared_hit *100 / int8larger(1,shared_hit + shared_read) as shared_buffer_hit,
    (shared_read - reads) *100 / int8larger(1,shared_hit + shared_read) as system_cache_hit,
    reads *100 / int8larger(1,shared_hit + shared_read) as physical_disk_read
FROM (
    SELECT userid, dbid, queryid, query,
        shared_blks_hit * 8192 as shared_hit,
        shared_blks_read * 8192 AS shared_read
    FROM pg_stat_statements
) s
JOIN pg_stat_kcache() k USING (userid, dbid, queryid)
JOIN pg_database d ON s.dbid = d.oid WHERE datname = 'dbproto_jsonb' AND userid = 47162
ORDER BY physical_disk_read desc;

```

datname	query	shared_buffer_hit	system_cache_hit	physical_disk_read
dbproto_jsonb	SELECT * from resources where uid=? AND parent=? AND name=? LIMIT ?;	71	24	4
dbproto_jsonb	UPDATE resources SET name=?, parent=? WHERE (_id=? AND uid=?);	86	10	3
dbproto_jsonb	UPDATE hierarchy SET path=?, huniq=?, parent=? WHERE (_id=? AND uid=?);	87	10	1
dbproto_jsonb	INSERT INTO resources (uid, parent, hid, folder, _id, data, name) VALUES (?, ?, ?, ?, ?, ?, ?);	92	7	0
dbproto_jsonb	INSERT INTO resources (folder, _id, uid, parent, name) VALUES (?, ?, ?, ?, ?);	93	6	0
dbproto_jsonb	SELECT * FROM resources WHERE (uid=? AND parent=?);	96	3	0
dbproto_jsonb	SELECT * from resources where _id=? AND uid=? LIMIT ?;	97	1	0
dbproto_jsonb	INSERT INTO hierarchy (path, huniq, _id, uid, parent) VALUES (?, ?, ?, ?, ?);	89	10	0
dbproto_jsonb	SELECT * from hierarchy where huniq=? AND uid=? LIMIT ?;	98	1	0
dbproto_jsonb	SELECT * from hierarchy where _id=? AND uid=? LIMIT ?;	98	1	0
dbproto_jsonb	SELECT * from resources where hid=? LIMIT ?;	87	12	0

(11 rows)

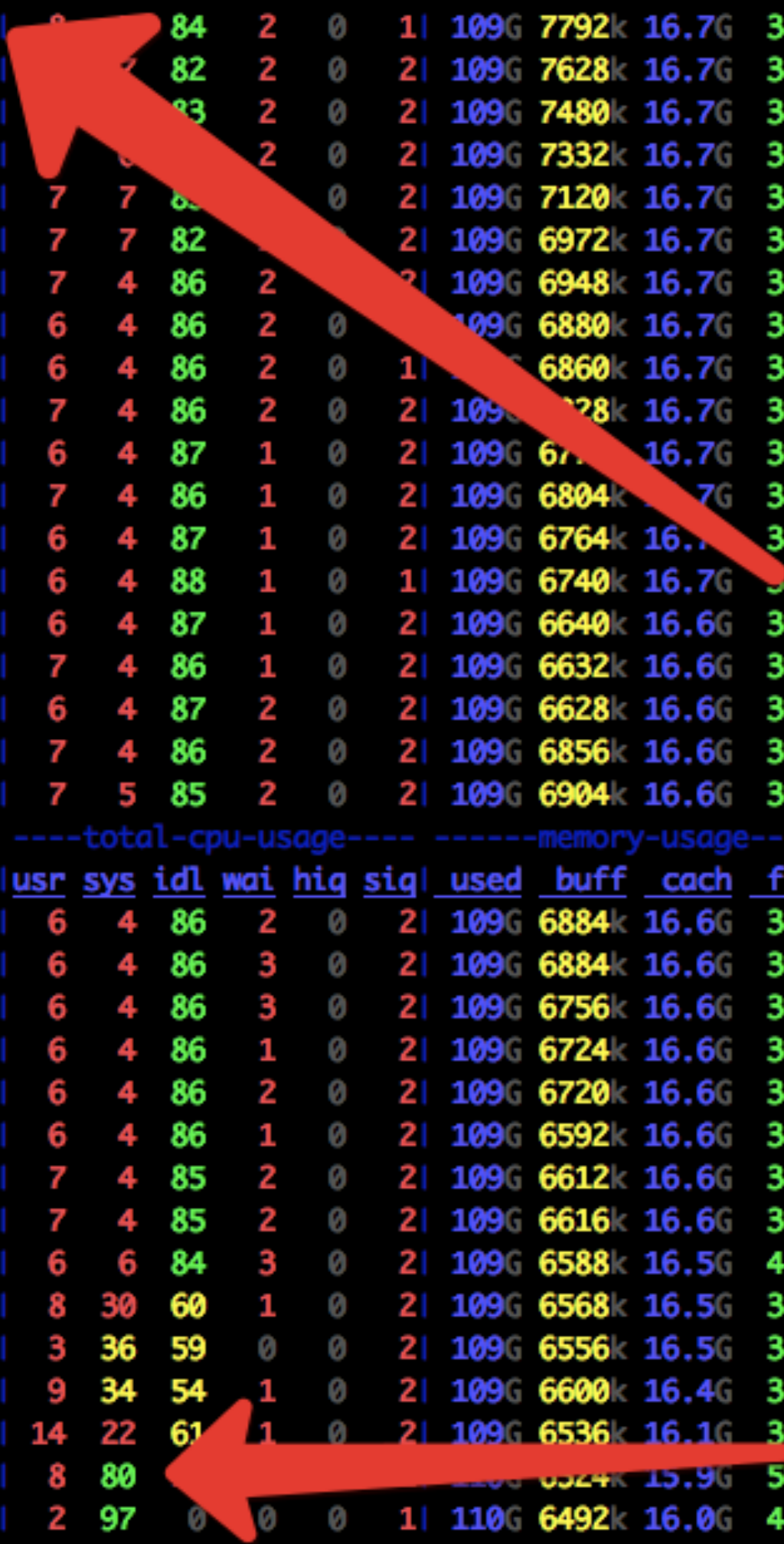
Time: 11.946 ms

pgload02g/dbproto\_jsonb M # █



# Проблемы сложнее

Сильное увеличение нагрузки на I/O приводит к потреблению всего процессорного времени в system



date/time	read	writ	read	writ	usr	sys	idl	wai	hiq	siq	used	buff	cach	free	rcv	send	1m
22-10 12:38:43	22M	37M	755k	9252k	6	1	92	1	0	0	109G	7800k	16.7G	389M	780k	19M	6.35
22-10 12:38:44	174M	176M	35.2k	33.0k	8	1	84	2	0	1	109G	7792k	16.7G	381M	791k	20M	6.57
22-10 12:38:45	279M	259M	58.1k	46.7k	7	1	82	2	0	2	109G	7628k	16.7G	387M	862k	45M	6.57
22-10 12:38:46	282M	253M	58.8k	46.5k	7	1	83	2	0	2	109G	7480k	16.7G	384M	849k	42M	6.57
22-10 12:38:47	286M	247M	59.8k	44.2k	7	1	82	2	0	2	109G	7332k	16.7G	371M	783k	39M	6.57
22-10 12:38:48	267M	218M	57.2k	40.1k	7	7	82	2	0	2	109G	7120k	16.7G	380M	807k	42M	6.57
22-10 12:38:49	245M	236M	53.6k	48.0k	7	7	82	2	0	2	109G	6972k	16.7G	384M	703k	36M	6.60
22-10 12:38:50	274M	270M	56.3k	49.9k	7	4	86	2	0	2	109G	6948k	16.7G	388M	803k	40M	6.60
22-10 12:38:51	254M	268M	53.4k	52.2k	6	4	86	2	0	1	109G	6880k	16.7G	382M	751k	36M	6.60
22-10 12:38:52	245M	253M	52.7k	48.8k	6	4	86	2	0	1	109G	6860k	16.7G	380M	791k	40M	6.60
22-10 12:38:53	253M	236M	55.0k	46.1k	7	4	86	2	0	2	109G	6728k	16.7G	390M	705k	33M	6.60
22-10 12:38:54	263M	238M	58.2k	49.0k	6	4	87	1	0	2	109G	6712k	16.7G	392M	722k	37M	7.11
22-10 12:38:55	269M	243M	57.2k	48.7k	7	4	86	1	0	2	109G	6804k	16.7G	389M	710k	34M	7.11
22-10 12:38:56	280M	269M	59.1k	53.7k	6	4	87	1	0	2	109G	6764k	16.7G	385M	769k	36M	7.11
22-10 12:38:57	245M	257M	52.7k	51.9k	6	4	88	1	0	1	109G	6740k	16.7G	386M	680k	33M	7.11
22-10 12:38:58	267M	256M	56.2k	52.7k	6	4	87	1	0	2	109G	6640k	16.6G	392M	718k	38M	7.11
22-10 12:38:59	275M	238M	60.4k	45.0k	7	4	86	1	0	2	109G	6632k	16.6G	391M	575k	25M	11.5
22-10 12:39:00	272M	232M	59.0k	46.5k	6	4	87	2	0	2	109G	6628k	16.6G	392M	742k	40M	11.5
22-10 12:39:01	272M	247M	59.1k	49.9k	7	4	86	2	0	2	109G	6856k	16.6G	390M	654k	33M	11.5
22-10 12:39:02	271M	249M	58.0k	49.9k	7	5	85	2	0	2	109G	6904k	16.6G	391M	708k	35M	11.5
22-10 12:39:03	259M	260M	55.1k	50.9k	6	4	86	2	0	2	109G	6884k	16.6G	381M	661k	33M	11.5
22-10 12:39:04	273M	244M	57.0k	48.0k	6	4	86	3	0	2	109G	6884k	16.6G	390M	678k	33M	11.4
22-10 12:39:05	283M	244M	58.8k	47.0k	6	4	86	3	0	2	109G	6756k	16.6G	392M	638k	30M	11.4
22-10 12:39:06	260M	240M	56.6k	48.5k	6	4	86	1	0	2	109G	6724k	16.6G	393M	776k	38M	11.4
22-10 12:39:07	228M	282M	47.9k	55.1k	6	4	86	2	0	2	109G	6720k	16.6G	387M	769k	34M	11.4
22-10 12:39:08	216M	258M	44.9k	49.6k	6	4	86	1	0	2	109G	6592k	16.6G	389M	737k	32M	11.4
22-10 12:39:09	249M	279M	49.4k	52.9k	7	4	85	2	0	2	109G	6612k	16.6G	386M	664k	33M	12.6
22-10 12:39:10	243M	242M	52.6k	48.0k	7	4	85	2	0	2	109G	6616k	16.6G	390M	779k	37M	12.6
22-10 12:39:11	249M	212M	55.1k	42.4k	6	6	84	3	0	2	109G	6588k	16.5G	419M	737k	36M	12.6
22-10 12:39:12	245M	214M	53.7k	42.6k	8	30	60	1	0	2	109G	6568k	16.5G	388M	492k	21M	12.6
22-10 12:39:13	252M	182M	60.2k	41.8k	3	36	59	0	0	2	109G	6556k	16.5G	381M	565k	28M	12.6
22-10 12:39:14	277M	240M	57.9k	46.8k	9	34	54	1	0	2	109G	6600k	16.4G	370M	261k	10M	12.8
22-10 12:39:15	231M	272M	48.5k	44.5k	14	22	61	1	0	2	109G	6536k	16.1G	395M	844k	30M	12.8
22-10 12:39:16	163M	129M	36.8k	31.0k	8	80	0	0	0	1	110G	6524k	15.9G	503M	879k	53M	12.8
22-10 12:39:17	166M	142M	40.2k	36.4k	2	97	0	0	0	1	110G	6492k	16.0G	484M	197k	7092k	12.8
22-10 12:39:18	198M	170M	48.4k	41.3k	6	89	4	0	0	1	110G	6308k	16.0G	387M	251k	9782k	12.8
22-10 12:39:19	195M	139M	46.1k	31.6k	6	93	0	0	0	1	110G	6308k	16.0G	376M	424k	12M	16.7



# Perf

```
Samples: 3M of event 'cycles', Event count (approx.): 894845079549
60.42% postgres      [.] s_lock
 8.09% postgres      [.] LWLockAcquire
 7.03% postgres      [.] LWLockRelease
 5.46% postgres      [.] PinBuffer
 2.80% postgres      [.] heap_page_prune_opt
 2.67% postgres      [.] hash_search_with_hash_value
 2.15% postgres      [.] heap_hot_search_buffer
 1.25% postgres      [.] UnpinBuffer
 0.93% postgres      [.] HeapTupleSatisfiesMVCC
 0.36% libc-2.12.so   [.] __memcmp_sse4_1
 0.35% postgres      [.] _bt_next
 0.33% [kernel]        [k] _spin_lock
 0.29% postgres      [.] CheckForSerializableConflictOut
 0.29% postgres      [.] ReadBuffer_common
 0.24% postgres      [.] hash_any
 0.23% postgres      [.] HeapTupleIsSurelyDead
 0.23% postgres      [.] heapgetpage
 0.21% postgres      [.] get_hash_value
```

# Perf

```
Samples: 2M of event 'cycles', Event count (approx.): 815539111912
```

Overhead	Shared Object	Symbol
71.00%	[kernel]	[k] _spin_lock_irqsave
0.67%	postgres	[.] AllocSetAlloc
0.63%	postgres	[.] SearchCatCache
0.62%	postgres	[.] GetSnapshotData
0.51%	postgres	[.] base_yyparse
0.45%	[kernel]	[k] schedule
0.40%	postgres	[.] hash_search_with_hash_value
0.40%	[kernel]	[k] update_curr
0.39%	[kernel]	[k] _spin_lock
0.35%	[kernel]	[k] prepare_to_wait
0.33%	[kernel]	[k] select_idle_sibling
0.29%	libc-2.12.so	[.] __strcmp_sse42
0.28%	[kernel]	[k] task_rq_lock

| We need to go deeper

Кобб

# SystemTap

- › Можно посмотреть очень многое
- › Требует пересборки PostgreSQL
- › Быстро star не написать, а иметь готовые на все случаи жизни не получится
- › Не стоит использовать в бою

<https://simply.name/postgresql-and-systemtap.html>

# GDB

```
# yum -y install postgresql94-debuginfo
# cat /usr/local/yandex/gdb_bt_cmd

bt

# fgrep gdb /usr/bin/bt

cmd = 'gdb -batch -x /usr/local/yandex/gdb_bt_cmd
/usr/pgsql-9.4/bin/postgres %d' % pid

#
```



# GDB

```
#0  0x00007fdacd68bf27 in semop () from /lib64/libc.so.6
#1  0x00000000000061fe27 in PGSemaphoreLock (sema=0x7fdb02f84b0, interruptOK=0 '\000')
at pg_sema.c:421
#2  0x0000000000006769ba in LWLockAcquireCommon (l=0x7fdacf400560, mode=LW_SHARED) at
lwlock.c:626
#3  LWLockAcquire (l=0x7fdacf400560, mode=LW_SHARED) at lwlock.c:467
#4  0x00000000000065db61 in BufferAlloc (smgr=0x1b30770, relpersistence=112 'p',
forkNum=MAIN_FORKNUM, blockNum=21261, mode=RBM_NORMAL, strategy=0x0,
hit=0x7fff20e86eff "") at bufmgr.c:591
#5  ReadBuffer_common (smgr=0x1b30770, relpersistence=112 'p', forkNum=MAIN_FORKNUM,
blockNum=21261, mode=RBM_NORMAL, strategy=0x0, hit=0x7fff20e86eff "") at bufmgr.c:340
#6  0x00000000000065e55e in ReadBufferExtended (reln=0x7fdacf08ede0,
forkNum=MAIN_FORKNUM, blockNum=21261, mode=<value optimized out>, strategy=<value
```

# GDB

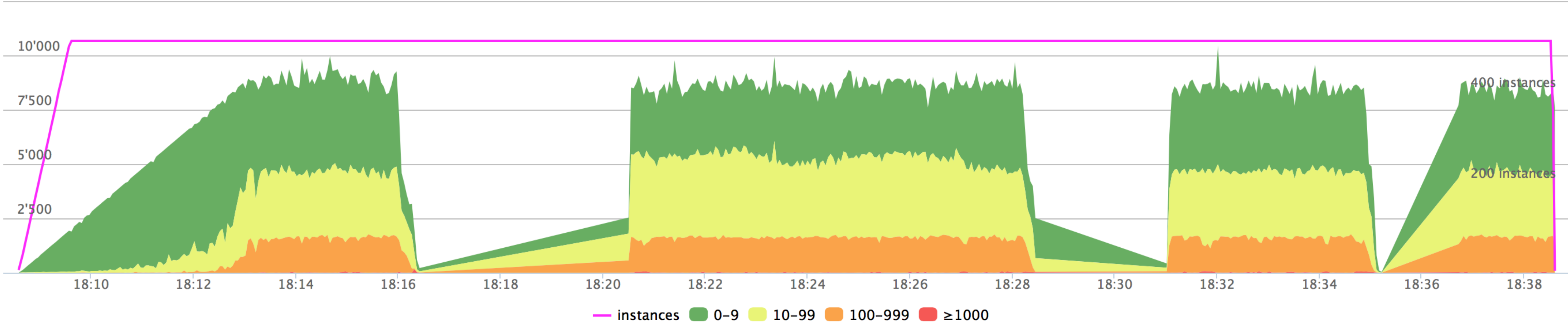
- › <https://github.com/dev1ant/misc/blob/master/bt>
- › Не складывает production
- › Пакеты \*-debuginfo есть во всех репозиториях страны
- › С точностью до строчки кода понятно, что происходит

# Отладка проблемы

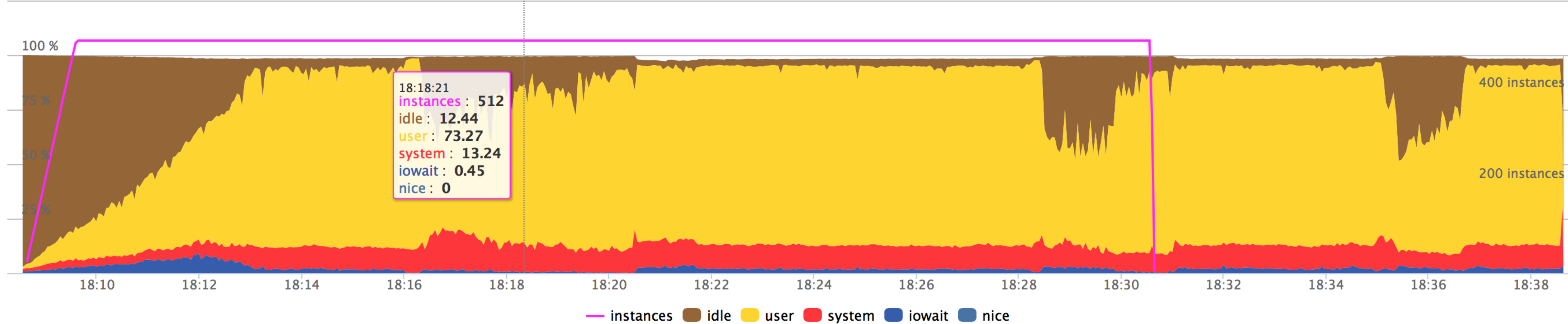


# Пример 1

Распределение времен ответа



Процессор @ pglload03g.mail.yandex.net





# Пример 1

Samples: 1M of event 'cycles', Event count (approx.): 392337478115

60.38%	postgres	[.] s_lock
2.67%	postgres	[.] PinBuffer
1.84%	postgres	[.] UnpinBuffer
1.79%	postgres	[.] XLogInsert
1.53%	postgres	[.] LWLockRelease
1.47%	postgres	[.] LWLockAcquire
1.17%	postgres	[.] rb_left_right_iterator
1.10%	postgres	[.] ginPostingListDecodeAllSegments
1.08%	postgres	[.] LockBuffer
0.98%	postgres	[.] ginMergeItemPointers
0.84%	postgres	[.] ginCompressPostingList
0.81%	postgres	[.] gintuple_get_key
0.77%	[kernel]	[k] _spin_lock
0.70%	rsync	[.] 0x0000000000001fde6
0.69%	postgres	[.] ReadBuffer_common
0.48%	libc-2.12.so	[.] memcpy
0.46%	postgres	[.] AllocSetAlloc
0.43%	postgres	[.] PageIndexTupleDelete
0.43%	[kernel]	[k] try_atomic_semop

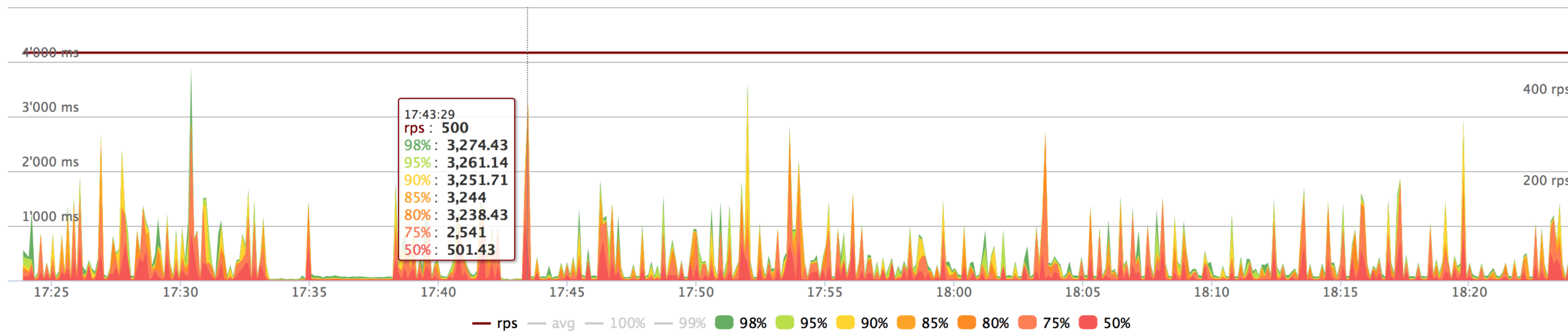
# Пример 1

```
ALTER INDEX mail.i_box_uid_lids SET (FASTUPDATE=OFF);
```

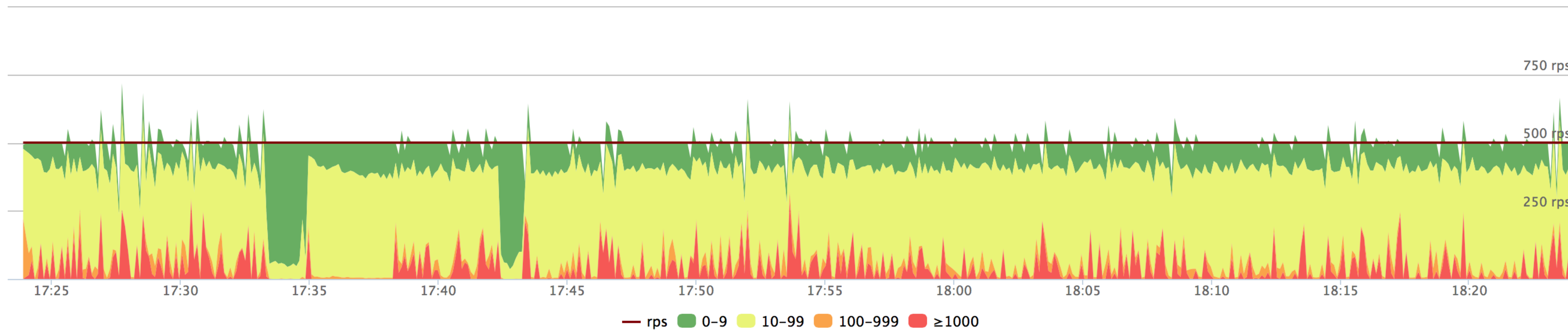
[postgresql.org/docs/current/static/gin-implementation.html](https://www.postgresql.org/docs/current/static/gin-implementation.html)

# Пример 2

Квантили времен ответа



Распределение времен ответа





# Пример 2

```
Samples: 475K of event 'cycles', Event count (approx.): 84273407321
```

5.12%	[kernel]	[k] compaction_alloc
4.03%	[unknown]	[.] 0x00007fec0f3be68b
2.80%	postgres	[.] XLogInsert
1.81%	[kernel]	[k] _spin_lock
1.55%	libc-2.12.so	[.] __strcoll_l
1.53%	libpython2.6.so.1.0	[.] 0x00000000000090d1e
1.44%	[kernel]	[k] _spin_lock_irqsave
1.31%	[kernel]	[k] copy_user_generic_string
1.19%	postgres	[.] hash_search_with_hash_value
1.08%	postgres	[.] SearchCatCache
1.07%	postgres	[.] vac_cmp_itemptr
1.00%	postgres	[.] AllocSetAlloc
0.82%	[kernel]	[k] page_fault
0.79%	libpython2.6.so.1.0	[.] PyEval_EvalFrameEx



# Пример 2

0x00007fb53eebfe97 in semop () from /lib64/libc.so.6

#0 0x00007fb53eebfe97 in semop () from /lib64/libc.so.6

#1 0x00000000000061fe27 in PGSemaphoreLock (sema=0x7fb757757430, interruptOK=0 '\000') at pg\_sema.c:421

#2 0x0000000000006769ba in LWLockAcquireCommon (l=0x7fb540c000a0, mode=LW\_EXCLUSIVE) at lwlock.c:626

#3 LWLockAcquire (l=0x7fb540c000a0, mode=LW\_EXCLUSIVE) at lwlock.c:467

#4 0x00000000000065ebde in StrategyGetBuffer (strategy=0x0, lock\_heId=0x7ffffb44cdebe "\001\001\030\001") at freelist.c:134

#5 0x00000000000065de25 in BufferAlloc (smgr=0x2a34090, relpersistence=112 'p', forkNum=MAIN\_FORKNUM, blockNum=1072861, mode=RBM\_NORMAL, strategy=0x0, hit=0x7ffffb44cdf2f "") at bufmgr.c:648

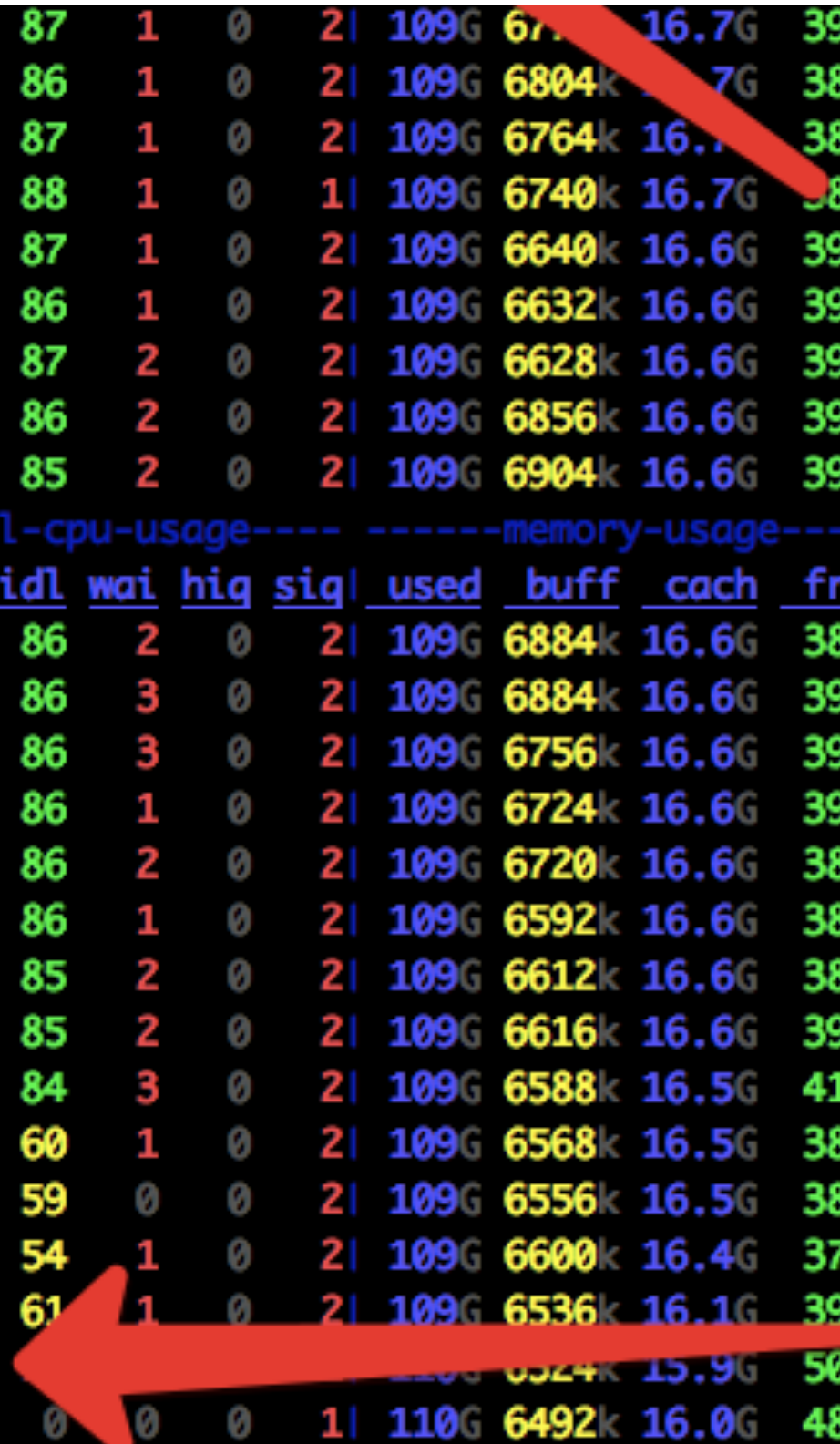
# Пример 2

```
# echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

[postgresql.org/message-id/55356496.9060607@aule.net](https://postgresql.org/message-id/55356496.9060607@aule.net)



# Пример 3



date/time	read	writ	read	writ	usr	sys	idl	wai	hiq	sigl	used	buff	cach	free	rcv	send	1m	5m	15m
22-10 12:38:54	263M	238M	58.2k	49.0k	6	4	87	1	0	2	109G	674k	16.7G	392M	722k	37M	7.11	7.13	8.00
22-10 12:38:55	269M	243M	57.2k	48.7k	7	4	86	1	0	2	109G	6804k	16.7G	389M	710k	34M	7.11	7.13	8.00
22-10 12:38:56	280M	269M	59.1k	53.7k	6	4	87	1	0	2	109G	6764k	16.7G	385M	769k	36M	7.11	7.13	8.00
22-10 12:38:57	245M	257M	52.7k	51.9k	6	4	88	1	0	1	109G	6740k	16.7G	386M	680k	33M	7.11	7.13	8.00
22-10 12:38:58	267M	256M	56.2k	52.7k	6	4	87	1	0	2	109G	6640k	16.6G	392M	718k	38M	7.11	7.13	8.00
22-10 12:38:59	275M	238M	60.4k	45.0k	7	4	86	1	0	2	109G	6632k	16.6G	391M	575k	25M	11.5	8.04	8.29
22-10 12:39:00	272M	232M	59.0k	46.5k	6	4	87	2	0	2	109G	6628k	16.6G	392M	742k	40M	11.5	8.04	8.29
22-10 12:39:01	272M	247M	59.1k	49.9k	7	4	86	2	0	2	109G	6856k	16.6G	390M	654k	33M	11.5	8.04	8.29
22-10 12:39:02	271M	249M	58.0k	49.9k	7	5	85	2	0	2	109G	6904k	16.6G	391M	708k	35M	11.5	8.04	8.29
----system---- -dsk/total- --io/total- ----total-cpu-usage---- -----memory-usage----- -net/total- ---load-avg---																			
date/time	read	writ	read	writ	usr	sys	idl	wai	hiq	sigl	used	buff	cach	free	rcv	send	1m	5m	15m
22-10 12:39:03	259M	260M	55.1k	50.9k	6	4	86	2	0	2	109G	6884k	16.6G	381M	661k	33M	11.5	8.04	8.29
22-10 12:39:04	273M	244M	57.0k	48.0k	6	4	86	3	0	2	109G	6884k	16.6G	390M	678k	33M	11.4	8.07	8.30
22-10 12:39:05	283M	244M	58.8k	47.0k	6	4	86	3	0	2	109G	6756k	16.6G	392M	638k	30M	11.4	8.07	8.30
22-10 12:39:06	260M	240M	56.6k	48.5k	6	4	86	1	0	2	109G	6724k	16.6G	393M	776k	38M	11.4	8.07	8.30
22-10 12:39:07	228M	282M	47.9k	55.1k	6	4	86	2	0	2	109G	6720k	16.6G	387M	769k	34M	11.4	8.07	8.30
22-10 12:39:08	216M	258M	44.9k	49.6k	6	4	86	1	0	2	109G	6592k	16.6G	389M	737k	32M	11.4	8.07	8.30
22-10 12:39:09	249M	279M	49.4k	52.9k	7	4	85	2	0	2	109G	6612k	16.6G	386M	664k	33M	12.6	8.37	8.40
22-10 12:39:10	243M	242M	52.6k	48.0k	7	4	85	2	0	2	109G	6616k	16.6G	390M	779k	37M	12.6	8.37	8.40
22-10 12:39:11	249M	212M	55.1k	42.4k	6	6	84	3	0	2	109G	6588k	16.5G	419M	737k	36M	12.6	8.37	8.40
22-10 12:39:12	245M	214M	53.7k	42.6k	8	30	60	1	0	2	109G	6568k	16.5G	388M	492k	21M	12.6	8.37	8.40
22-10 12:39:13	252M	182M	60.2k	41.8k	3	36	59	0	0	2	109G	6556k	16.5G	381M	565k	28M	12.6	8.37	8.40
22-10 12:39:14	277M	240M	57.9k	46.8k	9	34	54	1	0	2	109G	6600k	16.4G	370M	261k	10M	12.8	8.48	8.43
22-10 12:39:15	231M	272M	48.5k	44.5k	14	22	61	1	0	2	109G	6536k	16.1G	395M	844k	30M	12.8	8.48	8.43
22-10 12:39:16	163M	129M	36.8k	31.0k	8	80	0	0	0	1	110G	6524k	15.9G	503M	879k	53M	12.8	8.48	8.43
22-10 12:39:17	166M	142M	40.2k	36.4k	2	97	0	0	0	1	110G	6492k	16.0G	484M	197k	7092k	12.8	8.48	8.43
22-10 12:39:18	198M	170M	48.4k	41.3k	6	89	4	0	0	1	110G	6308k	16.0G	387M	251k	9782k	12.8	8.48	8.43
22-10 12:39:19	195M	139M	46.1k	31.6k	6	93	0	0	0	1	110G	6308k	16.0G	376M	424k	12M	16.7	9.37	8.72
22-10 12:39:20	309M	211M	71.8k	48.6k	3	91	5	0	0	2	110G	6312k	15.9G	378M	357k	13M	16.7	9.37	8.72
22-10 12:39:21	224M	174M	54.0k	40.2k	6	77	16	0	0	1	110G	6308k	15.9G	375M	307k	13M	16.7	9.37	8.72
22-10 12:39:22	258M	238M	54.1k	46.7k	11	54	31	1	0	2	110G	6300k	15.7G	370M	462k	19M	16.7	9.37	8.72



# Пример 3

```
Samples: 3M of event 'cycles', Event count (approx.): 894845079549
60.42% postgres      [.] s_lock
 8.09% postgres      [.] LWLockAcquire
 7.03% postgres      [.] LWLockRelease
 5.46% postgres      [.] PinBuffer
 2.80% postgres      [.] heap_page_prune_opt
 2.67% postgres      [.] hash_search_with_hash_value
 2.15% postgres      [.] heap_hot_search_buffer
 1.25% postgres      [.] UnpinBuffer
 0.93% postgres      [.] HeapTupleSatisfiesMVCC
 0.36% libc-2.12.so   [.] __memcmp_sse4_1
 0.35% postgres      [.] _bt_next
 0.33% [kernel]       [k] _spin_lock
 0.29% postgres      [.] CheckForSerializableConflictOut
 0.29% postgres      [.] ReadBuffer_common
 0.24% postgres      [.] hash_any
 0.23% postgres      [.] HeapTupleIsSurelyDead
 0.23% postgres      [.] heapgetpage
 0.21% postgres      [.] get_hash_value
```



# Пример 3

```
# grep '^#4 ' /tmp/bt | awk '{print $2, $4, $NF}' | sort
| uniq -c | sort -rn
    126 0x000000000000065db61 BufferAlloc bufmgr.c:591
     67 0x000000000000065e03a BufferAlloc bufmgr.c:760
     43 0x00000000000005c8c3b pq_getbyte pqcomm.c:899
     39 0x000000000000065dd93 BufferAlloc bufmgr.c:765
      6 0x00000000000004b52bb RecordTransactionCommit
xact.c:1194
     4 0x000000000000065da0e ReadBuffer_common bufmgr.c:476
     1 ReadBuffer_common re1persistence=112 bufmgr.c:340
     1 exec_eval_expr expr=0x166e908, pl_exec.c:4796
```

# Пример 3

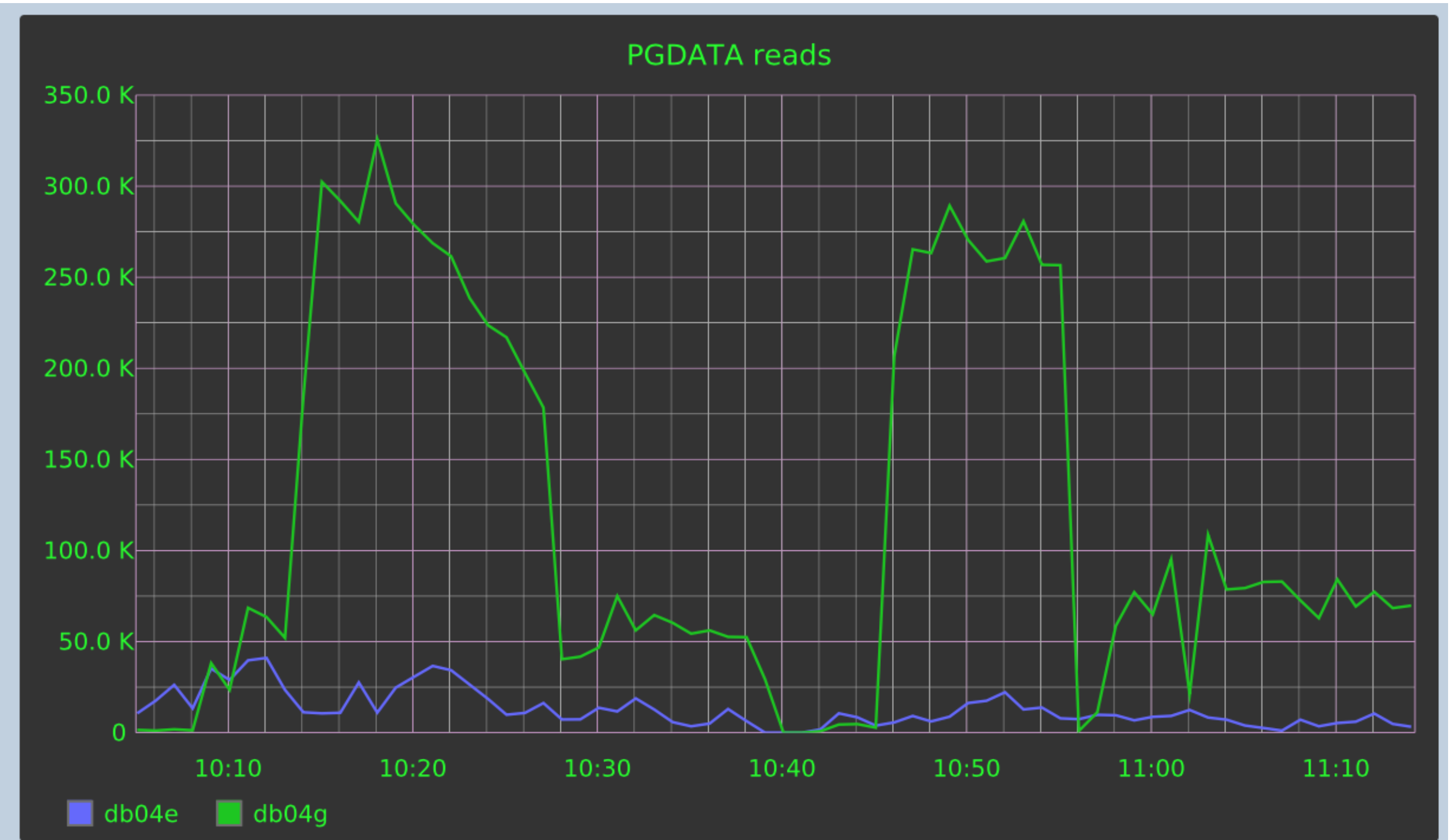
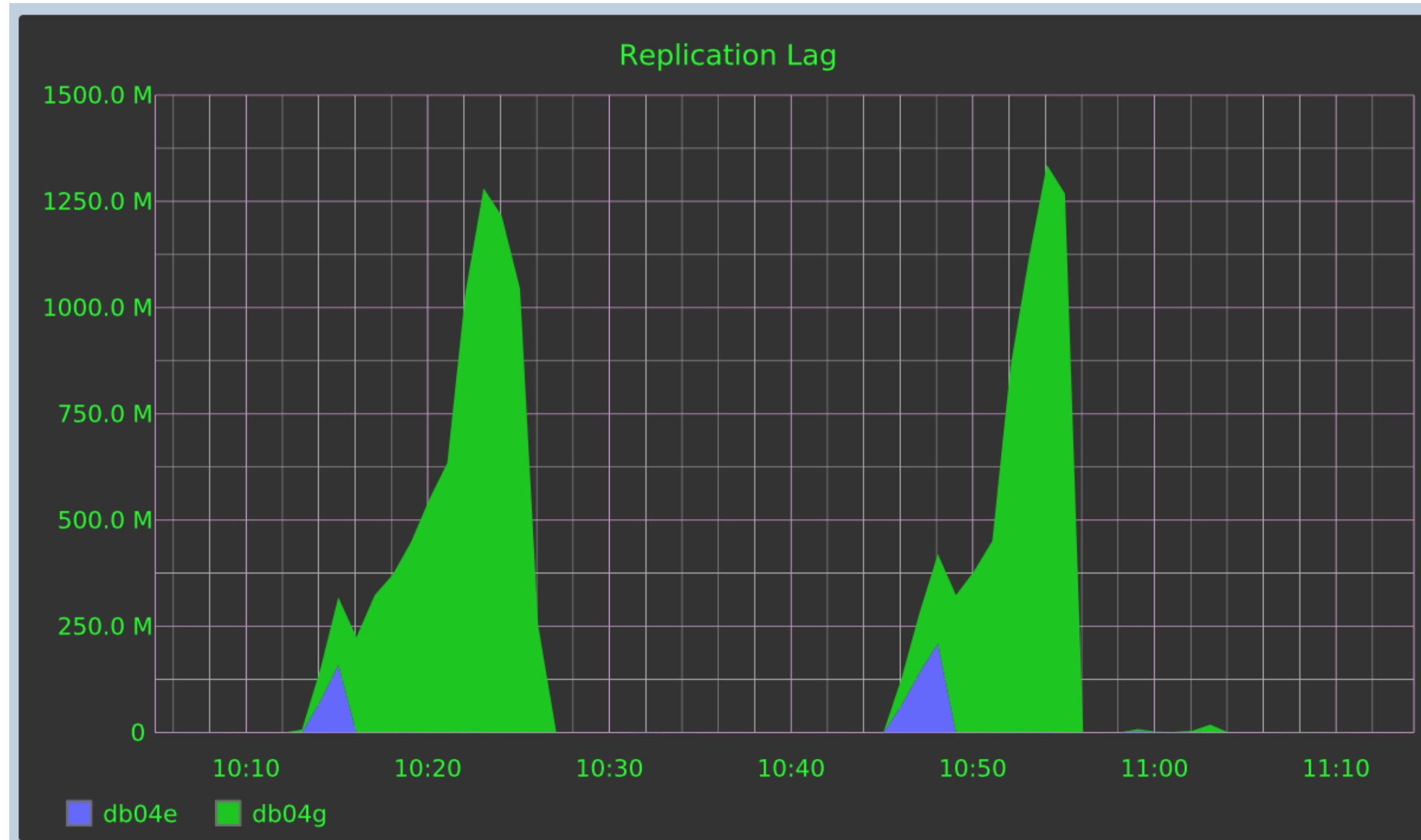
- › Уменьшение shared\_buffers
- › src/include/storage/lwlock.h

```
-#define NUM_BUFFER_PARTITIONS 16
```

```
+#define NUM_BUFFER_PARTITIONS 128
```

- › [git.postgresql.org/pg/commitdiff/d72731a7](https://git.postgresql.org/pg/commitdiff/d72731a7)
- › [git.postgresql.org/pg/commitdiff/1dcfb8da](https://git.postgresql.org/pg/commitdiff/1dcfb8da)
- › [wiki.postgresql.org/wiki/Shared\\_Buffer\\_Improvements](https://wiki.postgresql.org/wiki/Shared_Buffer_Improvements)

# Пример 4



<https://clck.ru/9Xffq>

# Пример 4

Total DISK READ: 490.42 M/s | Total DISK WRITE: 3.82 M/s

TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO	COMMAND
3316	be/4	postgres	492.34 M/s	0.00 B/s	0.00 %	39.91 %	postgres: startup process
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	init
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]

Samples: 110K of event 'cycles', Event count (approx.): 15973213676

29.63%	[kernel]	[k]	copy_user_generic_string
6.71%	postgres	[.]	hash_search_with_hash_value
5.10%	postgres	[.]	mdnblocks
3.89%	[kernel]	[k]	put_page
3.82%	postgres	[.]	_mdfd_getseg
1.96%	[kernel]	[k]	__mem_cgroup_commit_charge
1.70%	[kernel]	[k]	list_del
1.63%	postgres	[.]	LWLockAcquire



# Пример 4

```
0x00007f54a71444c0 in __read_nocancel () from /lib64/libc.so.6
#0 0x00007f54a71444c0 in __read_nocancel () from /lib64/libc.so.6
#1 0x000000000065d2f5 in FileRead (file=<value optimized out>, buffer=0x7f53ac0dba20 ";9",
amount=8192) at fd.c:1286
#2 0x000000000067acad in mdread (reIn=<value optimized out>, forknum=<value optimized out>,
blocknum=12063036, buffer=0x7f53ac0dba20 ";9") at md.c:679
#3 0x0000000000659b4e in ReadBuffer_common (smgr=<value optimized out>, relpersistence=112 'p',
forkNum=MAIN_FORKNUM, blockNum=12063036, mode=RBM_NORMAL_NO_LOG, strategy=0x0, hit=0x7fff898a912f "")
at bu
fmgr.c:476
#4 0x000000000065a61b in ReadBufferWithoutRelcache (rnode=..., forkNum=MAIN_FORKNUM,
blockNum=12063036, mode=<value optimized out>, strategy=<value optimized out>) at bufmgr.c:287
#5 0x00000000004cfb78 in XLogReadBufferExtended (rnode=..., forknum=MAIN_FORKNUM, blkno=12063036,
mode=RBM_NORMAL_NO_LOG) at xlogutils.c:324
#6 0x00000000004a3651 in btree_xlog_vacuum (lsn=71744181579864, record=0x1e49dc0) at nbtxlog.c:522
#7 btree_redo (lsn=71744181579864, record=0x1e49dc0) at nbtxlog.c:1144
#8 0x00000000004c903a in StartupXLOG () at xlog.c:6827
#9 0x000000000062f8bf in StartupProcessMain () at startup.c:224
#10 0x00000000004d3e9a in AuxiliaryProcessMain (argc=2, argv=0x7fff898a98a0) at bootstrap.c:416
#11 0x000000000062a99c in StartChildProcess (type=StartupProcess) at postmaster.c:5146
```

# Пример 4

- › <https://clck.ru/9Xfx6>
- › <https://commitfest.postgresql.org/5/233/>
- › Секционирование больших insert-only таблиц

Светлое будущее



# Какие ожидания нужно мониторить?

- › Locks (heavyweight)
- › LWLocks (lightweight locks)
- › Latch
- › Network
- › Storage (IO)



# Какие недостатки у текущих инструментов?

- › Часто нельзя использовать в production
- › Они разрозненные и их много
- › Тяжело настраиваются и устанавливаются

# pg\_stat\_wait

- › Profiling
- › Трассировка в файл
- › История ожиданий

# Требования

- › Минимальный overhead
- › Возможность просмотра в онлайн-режиме
- › Высокая точность
- › Один инструмент для всех типов ожиданий

# pg\_stat\_wait\_profile

```
b1=# SELECT * FROM pg_stat_wait_profile ORDER BY wait_count DESC;
```

pid	class_id	class_name	event_id	event_name	wait_time	wait_count
10781	4	Latch	0	Latch	90939504509	18217
11085	4	Latch	0	Latch	51834	9209
11085	5	Network	1	WRITE	65355	576
11085	3	Storage	0	READ	2273	188
11085	5	Network	0	READ	90932631592	93
10779	3	Storage	1	WRITE	1440	65
10782	3	Storage	0	READ	1040	43
10780	3	Storage	1	WRITE	367	15
5829	5	Network	1	WRITE	11	3
6025	5	Network	1	WRITE	10	2
11085	1	LWLocks	41	BufferPartitionLock	1355	2
11085	3	Storage	1	WRITE	9	2
5829	5	Network	0	READ	462	2



# pg\_stat\_wait\_history

```
b1=# SELECT * FROM pg_stat_wait_history WHERE class_id != 4 LIMIT 2;
```

pid	sample_ts	class_id	class_name	event_id	event_name	wait_time
p1	p2	p3	p4	p5		
11085	2015-07-10 10:37:41.321489-04	3	Storage	1	WRITE	3520
1663	16384	16458	0	0		
11085	2015-07-10 10:37:09.390442-04	5	Network	1	WRITE	7815
0	0	0	0	0		

> class\_id - класс ожидания (lock, lwlock, io, latch, or network)

> event\_id - событие

> p1 .. p5 - параметры ожидания

# Трассировка

Даёт большой overhead, используйте только для отдельных сессий

```
terminal 1: $ psql b1
```

```
terminal 2:
```

```
$ ps ax | grep postgres
```

```
<...>
```

```
11085 ?          Ss          0:00 postgres: postgres b1 [local] idle
```

```
$ psql b1 -c "select pg_start_trace(11085, '/tmp/f.trace')"
```

```
terminal 1:
```

```
b1=# CREATE TABLE t1 AS SELECT i, i*10 AS i1 FROM generate_series(1,10) i;  
SELECT 10
```

# Трассировка

terminal 2:

```
$ tail -f /tmp/f.trace
```

```
stop 2015-07-10 10:03:35.603458-04 Network
```

```
start 2015-07-10 10:03:35.603464-04 Network READ 0 0 0 0 0
```

```
stop 2015-07-10 10:03:44.099587-04 Network
```

```
start 2015-07-10 10:03:44.100401-04 Storage READ 1663 16384 1259 2 0
```

```
stop 2015-07-10 10:03:44.100424-04 Storage
```

```
start 2015-07-10 10:03:44.102549-04 Network WRITE 0 0 0 0 0
```

```
stop 2015-07-10 10:03:44.102573-04 Network
```

```
start 2015-07-10 10:03:44.102582-04 Network READ 0 0 0 0 0
```

```
stop 2015-07-10 10:05:33.029975-04 Network
```

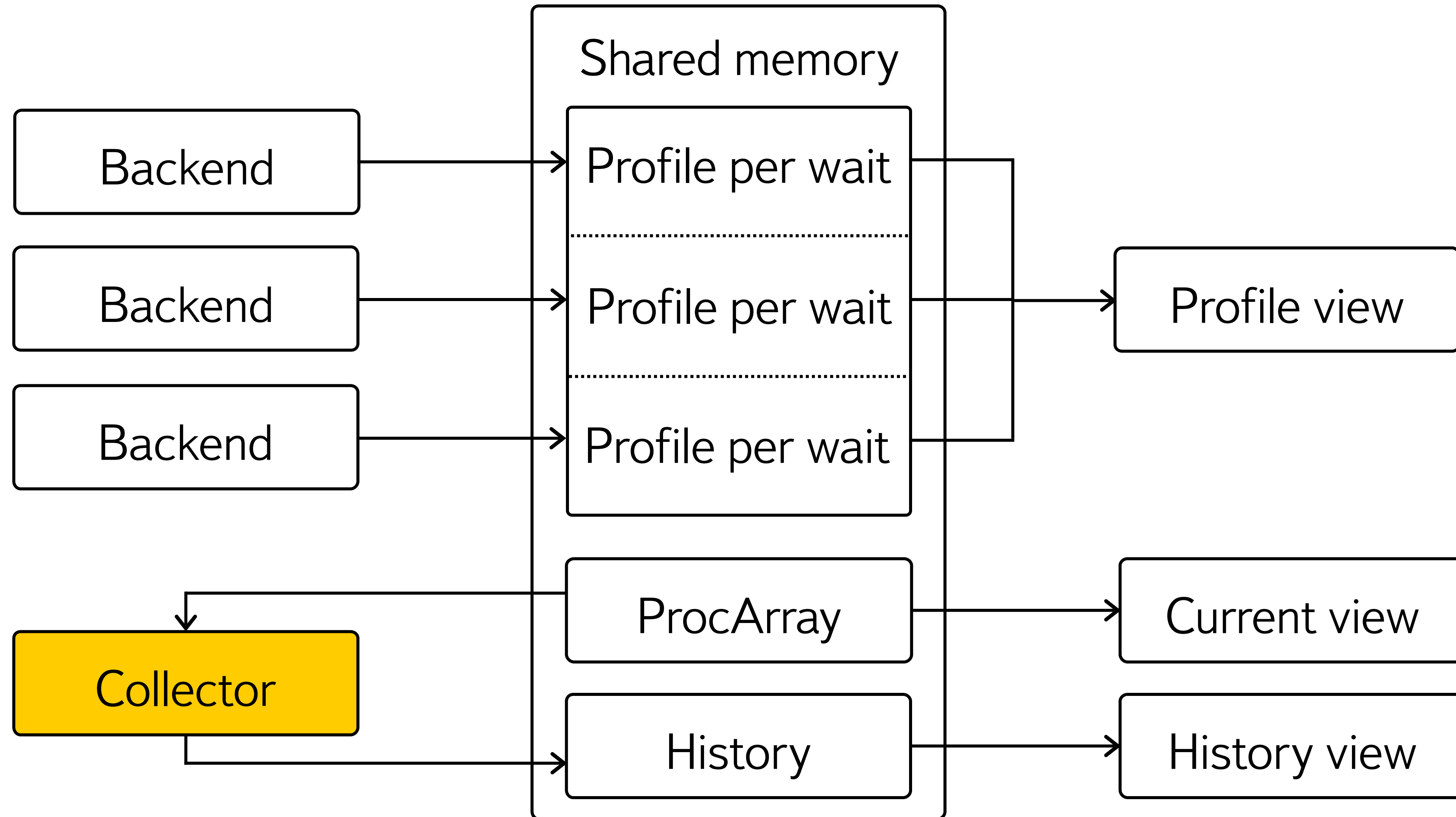
```
start 2015-07-10 10:05:33.030205-04 Storage READ 1663 16384 2691 0 28
```

```
stop 2015-07-10 10:05:33.030233-04 Storage
```

```
start 2015-07-10 10:05:33.030246-04 Storage READ 1663 16384 1255 0 50
```

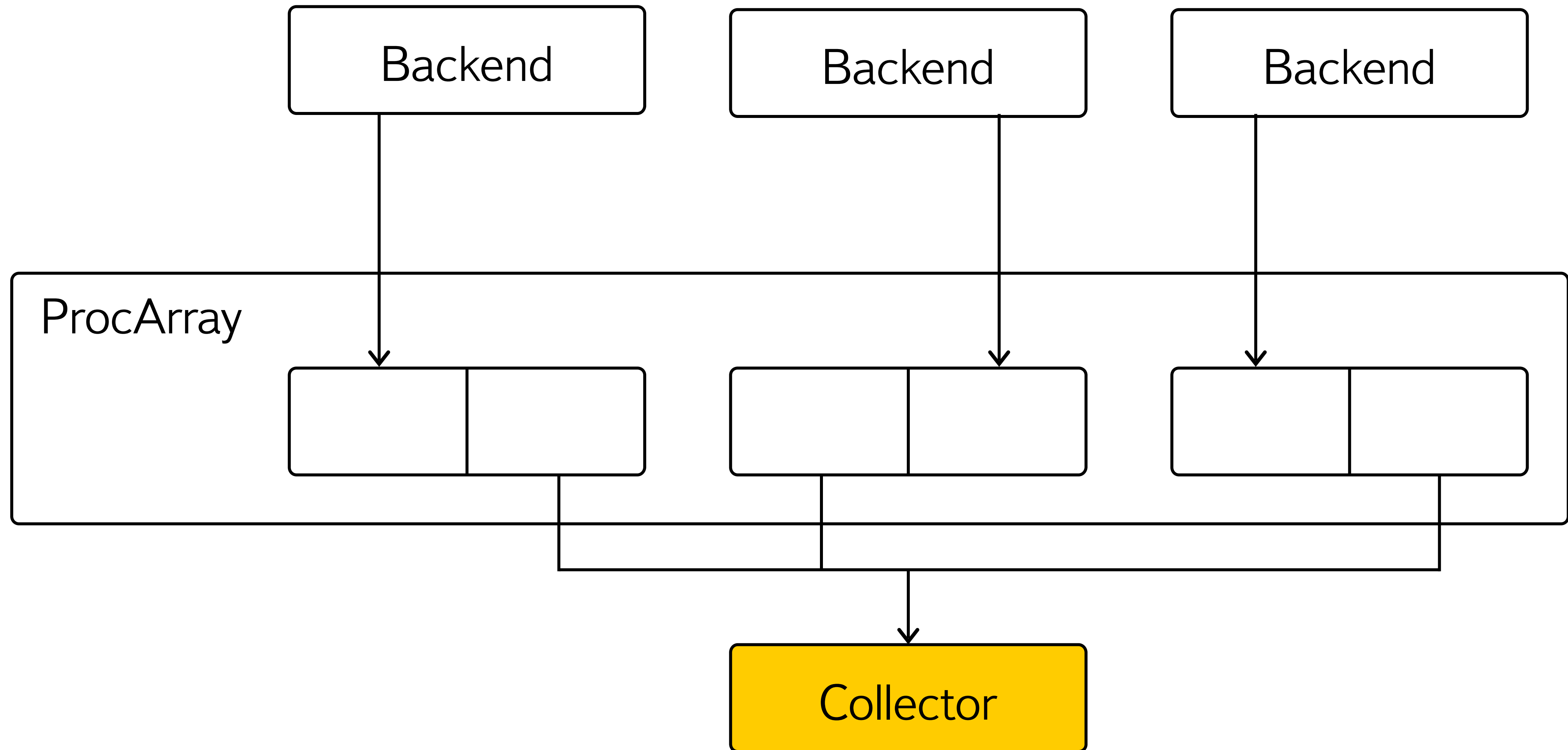
```
stop 2015-07-10 10:05:33.03026-04 Storage
```

# Как устроен pg\_stat\_wait





# Сбор истории



# Overhead

Тестовая конфигурация:

- › Intel(R) Xeon(R) CPU X5675@3.07GHz, 24 cores
- › RAM 24 GB
- › pgbench -S 500 ~ 1.6 Gb

Избегаем влияния дисковой подсистемы:

- › fsync off
- › tmpfs

# Monitoring off

```
[ildus@1-i-kurbangaliev postgrespro]$ pgbench -S b1 -c 96 -j 4 -T 300
starting vacuum...end.
transaction type: SELECT only
scaling factor: 500
query mode: simple
number of clients: 96
number of threads: 4
duration: 300 s
number of transactions actually processed: 39349816
latency average: 0.732 ms
tps = 131130.859559 (including connections establishing)
tps = 131153.752204 (excluding connections establishing)
```

# Monitoring on

```
[ildus@1-i-kurbangaliev postgrespro]$ pgbench -S b1 -c 96 -j 4 -T 300
starting vacuum...end.
transaction type: SELECT only
scaling factor: 500
query mode: simple
number of clients: 96
number of threads: 4
duration: 300 s
number of transactions actually processed: 39172607
latency average: 0.735 ms
tps = 130574.626755 (including connections establishing)
tps = 130600.767440 (excluding connections establishing)
```



# Open source

- › <https://github.com/postgrespro/postgres>
- › <https://clck.ru/9XsnG>
- › <https://clck.ru/9XsnE>

Немного примеров



# Было

```
0x00007f10e01d4f27 in semop () from /lib64/libc.so.6
#0 0x00007f10e01d4f27 in semop () from /lib64/libc.so.6
#1 0x00000000000061fe27 in PGSemaphoreLock (sema=0x7f11f2d4f430, interruptOK=0 '\000') at
pg_sema.c:421
#2 0x0000000000006769ba in LWLockAcquireCommon (l=0x7f10e1e00120, mode=LW_EXCLUSIVE) at
lwlock.c:626
#3 LWLockAcquire (l=0x7f10e1e00120, mode=LW_EXCLUSIVE) at lwlock.c:467
#4 0x000000000000667862 in ProcArrayEndTransaction (proc=0x7f11f2d4f420, latestXid=182562881)
at procarray.c:404
#5 0x0000000000004b579b in CommitTransaction () at xact.c:1957
#6 0x0000000000004b6ae5 in CommitTransactionCommand () at xact.c:2727
#7 0x0000000000006819d9 in finish_xact_command () at postgres.c:2437
#8 0x000000000000684f05 in PostgresMain (argc=<value optimized out>, argv=<value optimized
out>, dbname=0x21e1a70 "xivadb", username=<value optimized out>) at postgres.c:4270
#9 0x000000000000632d7d in BackendRun (argc=<value optimized out>, argv=<value optimized out>)
at postmaster.c:4155
#10 BackendStartup (argc=<value optimized out>, argv=<value optimized out>) at postmaster.c:
3829
```

# Стало

```
xivadb01e/postgres M # SELECT * FROM pg_stat_wait_history WHERE class_name NOT IN ('Network', 'Latch') LIMIT 20;
```

pid	sample_ts	class_id	class_name	event_id	event_name	wait_time	p1	p2	p3	p4	p5
17316	2015-07-07 15:38:25.011413+03	1	LWLocks	4	ProcArrayLock	46	0	0	0	0	0
10685	2015-07-07 15:38:24.920067+03	1	LWLocks	4	ProcArrayLock	55	0	0	0	0	0
5822	2015-07-07 15:38:25.062193+03	1	LWLocks	4	ProcArrayLock	66	0	0	0	0	0
29168	2015-07-07 15:38:25.011413+03	1	LWLocks	4	ProcArrayLock	70	0	0	0	0	0
16983	2015-07-07 15:38:25.072456+03	1	LWLocks	4	ProcArrayLock	23	0	0	0	0	0
7905	2015-07-07 15:38:25.072456+03	1	LWLocks	4	ProcArrayLock	25	0	0	0	0	0
24608	2015-07-07 15:38:25.011413+03	1	LWLocks	4	ProcArrayLock	14	0	0	0	0	0
19582	2015-07-07 15:38:24.909923+03	1	LWLocks	4	ProcArrayLock	54	0	0	0	0	0
17101	2015-07-07 15:38:25.011413+03	1	LWLocks	4	ProcArrayLock	35	0	0	0	0	0
28237	2015-07-07 15:38:24.920067+03	1	LWLocks	4	ProcArrayLock	56	0	0	0	0	0
2603	2015-07-07 15:38:25.001273+03	1	LWLocks	4	ProcArrayLock	8	0	0	0	0	0
8257	2015-07-07 15:38:25.072456+03	1	LWLocks	4	ProcArrayLock	52	0	0	0	0	0
11843	2015-07-07 15:38:24.980917+03	1	LWLocks	4	ProcArrayLock	9	0	0	0	0	0
19845	2015-07-07 15:38:25.072456+03	1	LWLocks	4	ProcArrayLock	2	0	0	0	0	0
28237	2015-07-07 15:38:24.930211+03	1	LWLocks	4	ProcArrayLock	35	0	0	0	0	0
12631	2015-07-07 15:38:25.062193+03	1	LWLocks	4	ProcArrayLock	63	0	0	0	0	0
24591	2015-07-07 15:38:25.062193+03	1	LWLocks	4	ProcArrayLock	61	0	0	0	0	0
17778	2015-07-07 15:38:24.899773+03	1	LWLocks	4	ProcArrayLock	17	0	0	0	0	0
2213	2015-07-07 15:38:24.909923+03	1	LWLocks	4	ProcArrayLock	4	0	0	0	0	0
20087	2015-07-07 15:38:24.980917+03	1	LWLocks	4	ProcArrayLock	68	0	0	0	0	0

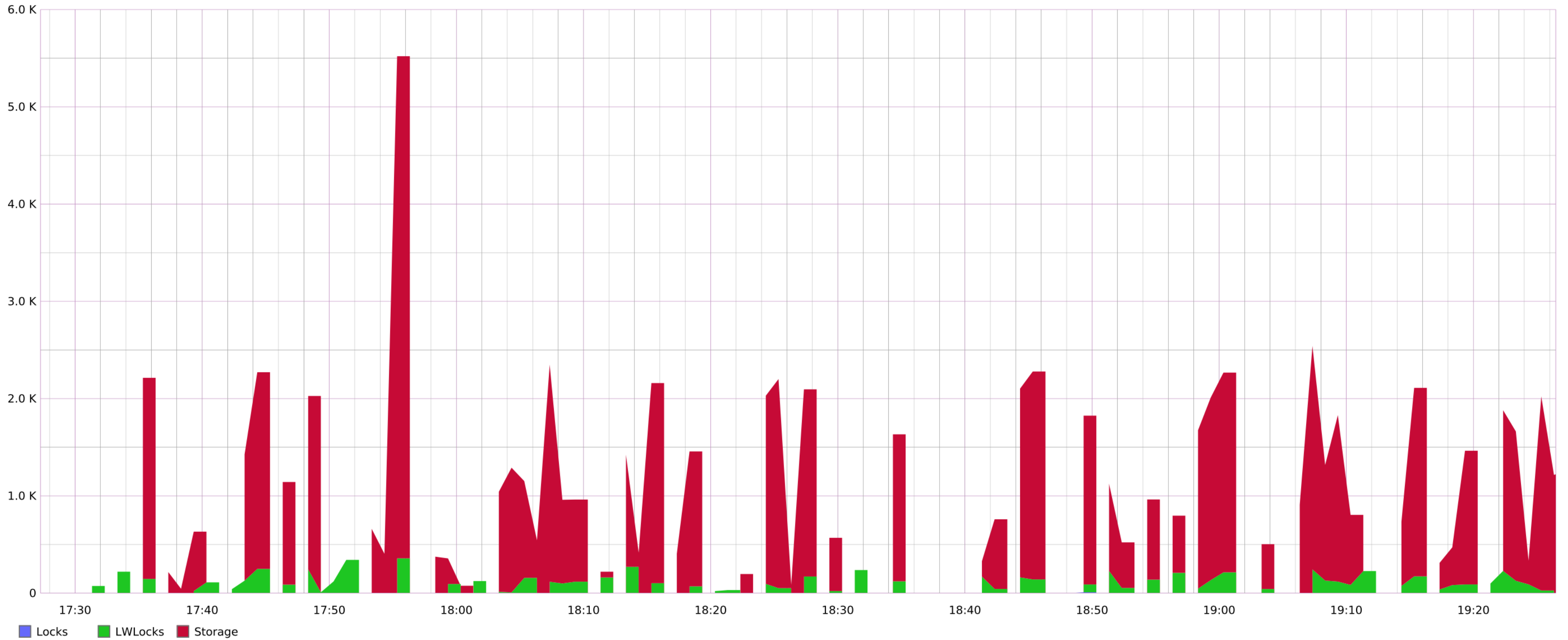
(20 rows)

Time: 6.458 ms

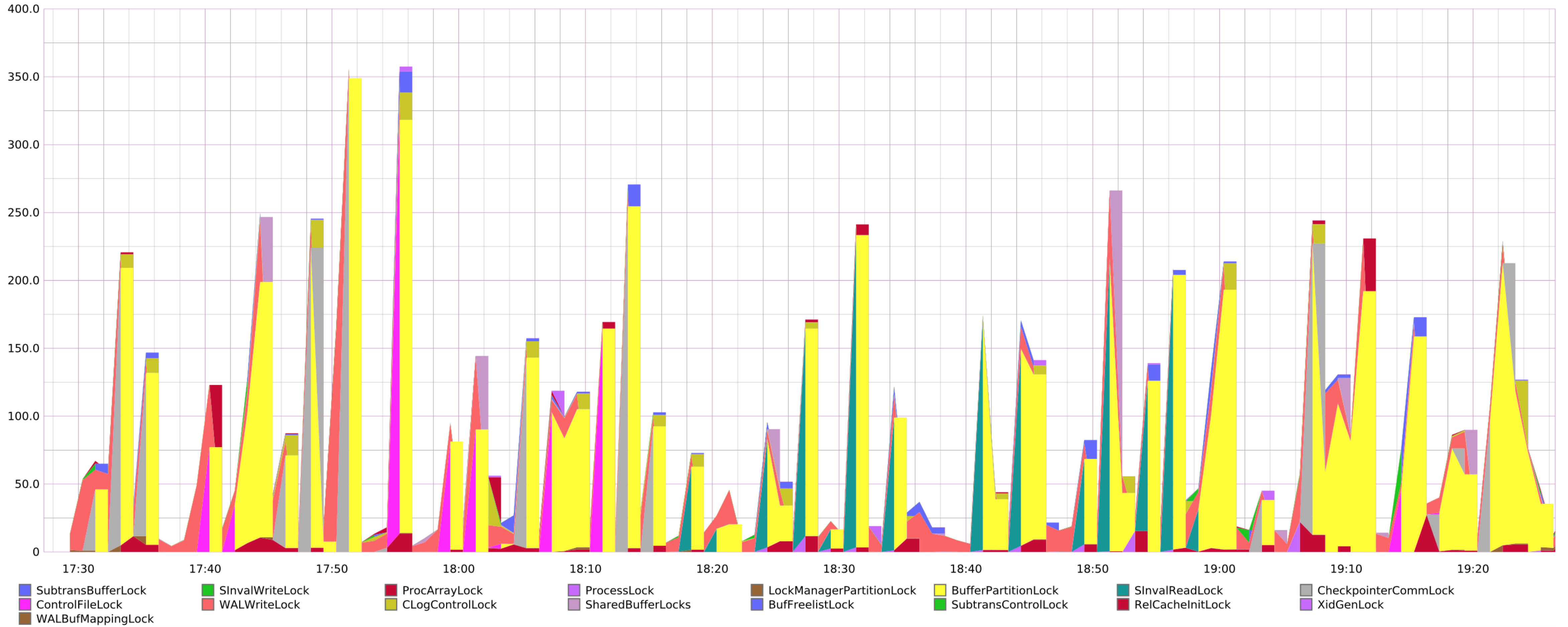
```
xivadb01e/postgres M # █
```



Wait time by classes (ms)



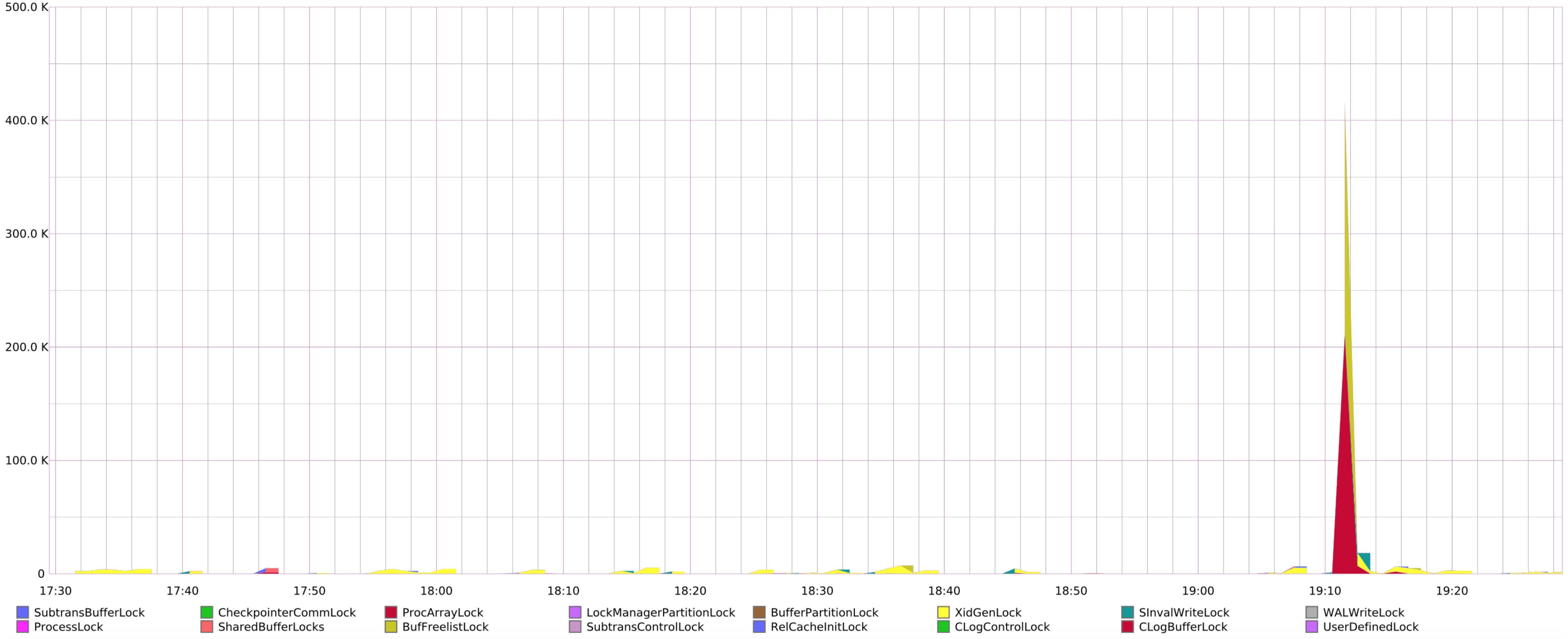
Wait time for LWLocks (ms)



Wait time by classes (ms)



Wait time for LWLocks (ms)





Вопросы?

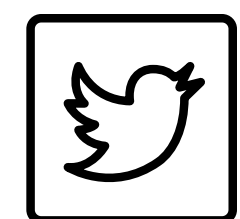
# Контакты

Владимир Бородин

Системный администратор



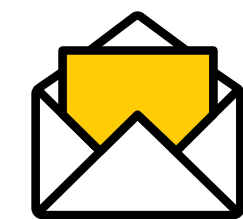
[d0uble@yandex-team.ru](mailto:d0uble@yandex-team.ru)



@man\_brain

Ильдус Курбангалиев

Разработчик



[i.kurbangaliev@postgrespro.ru](mailto:i.kurbangaliev@postgrespro.ru)