MySQL and OpenStack Deep Dive (Juno)

Peter Boros, Percona
Jay Pipes, Mirantis

www.mirantis.com

April, 2015

Diving into MySQL and OpenStack

PERCONA

MIRANTIS
Pure Play OpenStack

- Our setup
- Analyzing MySQL query loads
- Takeaways
- Changes from last openstack summit
  - Last time we examined Icehouse, this time Juno

Objectives

- Examine the database communication and load in an OpenStack environment
- Identify database-layer bottlenecks at various levels of concurrency
- Understand the query load placed against the database server during typical requests to Nova and Neutron

Overview

PERCONA

MIRANTIS
Pure Play OpenStack

- Multiple AWS instances running infrastructure and OpenStack components
- Ansible playbooks used to set up both under and overcloud
  - http://github.com/percona/openstack_ansible
- Unmodified OpenStack Juno packages
- Unmodified Percona packages

Deployment

- Different challenges than most deployment tools
- Controller nodes need to perform
- Some things are not needed which may be important to others
  - Actually starting a VM
  - The tenant network to be actually reachable

Architecture

- Each node runs haproxy locally
  - Which knows where each service is located
- This is a benchmark, no haproxy checks
  - Checks can become a serious overhead, everything is checking everything practically

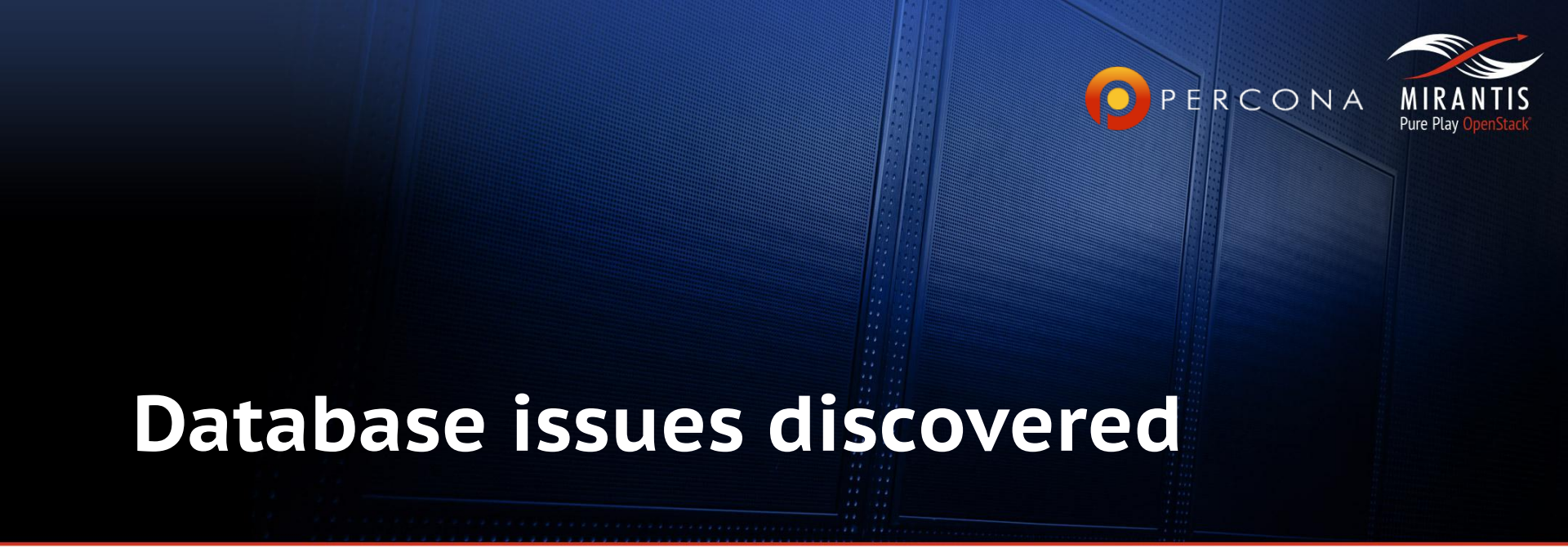
The (first) setup

AWS Instances

PERCONA

MIRANTIS
Pure Play OpenStack

- 7 AWS instances
  - 1 controller nodes
  - 1 compute worker node
  - 1 network node
  - 3 PXC nodes
  - 1 Rally benchmark node
- c3.8xlarge instance type except compute
  - 24 cores / 60GB RAM / SSD ephemeral drives



Database issues discovered

Other bottlenecks

PERCONA

MIRANTIS
Pure Play OpenStack

- Default number of workers of neutron-server is 1
  - api_workers = NUM_CPU * 2
  - rpc_workers = NUM_CPU
- Keystone is a single process daemon (Icehouse)
  - Run it as a WSGI application within Apache mod_wsgi
  - https://bugs.launchpad.net/rally/+bug/1379876
- Keystone in Juno is multi-process by default
- Queue on the network node

Some not so weird setup

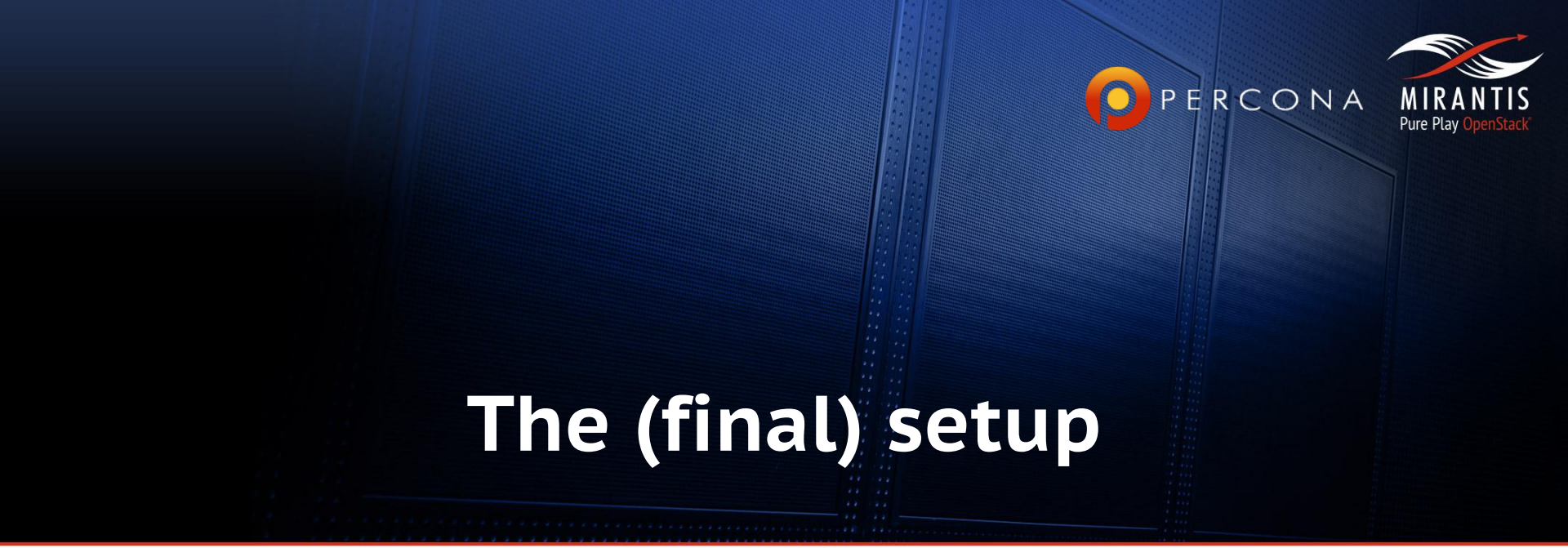
- Increase the number of file descriptors
- For lots of instances, you need a big subnet
  - Just following the guide /24 becomes small very fast
  - Even /16 became small quite fast with the Juno tests
- You have to change the port of the base services, so the local haproxies can bind to the original port
- Configure database connection pools

Nova fun facts and logs

PERCONA MIRANTIS Pure Play OpenStack

- Nova parameters
  - compute_driver = nova.virt.fake.FakeDriver
  - ram_allocation_ratio = 100000
  - cpu_allocation_ratio = 100000

```
2015-04-11 17:43:08.023 24020 AUDIT nova.compute.resource_tracker [-
  ] Auditing locally available compute resources
2015-04-11 17:43:15.967 24020 AUDIT nova.compute.resource_tracker [-
  ] Total physical ram (MB): 800000, total allocated virtual ram
  (MB): 770048
2015-04-11 17:43:15.968 24020 AUDIT nova.compute.resource_tracker [-
  ] Free disk (GB): 598497
2015-04-11 17:43:15.968 24020 AUDIT nova.compute.resource_tracker [-
  ] Total usable vcpus: 1000, total allocated vcpus: 0
```



The (final) setup

http://ceilingcat.ninja

AWS Instances

- 30 AWS instances
  - 5 controller node (we want more in the future)
  - 20 compute worker nodes
  - 1 network + queue node
  - 3 PXC nodes
  - 1 Rally benchmark node
- c3.8xlarge instance type except compute
  - 24 cores / 60GB RAM / SSD drives
- m3.medium
  - 1 cores / 3.75GB RAM

Isolating the database layer

PERCONA

MIRANTIS
Pure Play OpenStack

- Enable Nova's fake virt driver
- 12 compute nodes to simulate more realistic compute_nodes table updates and reservation/claim queries
- Quota driver set to QuotaDBDriver (Icehouse)
  - Set quotas to -1
- Keystone UUID tokens
  - UUID tokens hammer the Keystone DB more anyway :)

What we tested

Check we're ready

PERCONA

MIRANTIS
Pure Play OpenStack

```
root@ip-10-10-10-107:~# rally deployment check
keystone endpoints are valid and following services are available:
+----------+----------------+-----------+
| services | type           | status    |
+----------+----------------+-----------+
| cinder   | volume         | Available |
| cinderv2 | volumev2       | Available |
| glance   | image          | Available |
| heat     | orchestration  | Available |
| heat-cfn | cloudformation | Available |
| keystone | identity       | Available |
| neutron  | network        | Available |
| nova     | compute        | Available |
+----------+----------------+-----------+
```

Check the scenario config file

Generated by Ansible from template:

https://github.com/percona/openstack_ansible/blob/master/roles/rally_node/templates/boot_and_delete_server.json.j2

```
root@ip-10-10-10-107:~# cat /var/tmp/boot_and_delete_server.json
{
    "NovaServers.boot_and_delete_server": [
        {
            "args": {
                "flavor": {
                    "name": "m1.tiny"
                },
                "image": {
                    "name": "cirros-0.3.2-x86_64"
                },
                "nics": [ {
                    "net-id": "28a314dc-321e-4baf-9e01-1c299d2b91fc"
                        } ] },...
```

Populated with ID of a shared network created for the benchmark tenants

Scenario file (cont'd)

```
...
        "runner": {
            "type": "constant",
            "times": 1000,
            "concurrency": 24
        },
        "context": {
            "users": {
                "tenants": 50,
                "users_per_tenant": 10
            },...
```

Number of cores on the benchmarking node

Created by Rally in its pre-processing steps

Scenario file (cont'd)

```
...
                "quotas": {
                  "nova": {
                    "instances": -1,
                    "cores": -1,
                    "ram": -1
                  },
                  "neutron": {
                    "port": -1
                  }
                }

            }
        }
    ]
}
```

Note: we found no perceivable difference between setting to -1 and setting to 100,000

Running the benchmark

```
root@ip-10-10-10-107:~# rally -v task start /var/tmp/boot_and_delete_server.json
```

```
root@ip-10-10-10-150:~# rally task list
+--------------------------------------+----------------------------+---------+--------+-----+
| uuid                                 | created_at                 | status  | failed | tag |
+--------------------------------------+----------------------------+---------+--------+-----+
| 46bb408f-b94b-40a7-8a03-ab03f763d5bb | 2014-10-31 19:23:22.457154 | running | False  |     |
+--------------------------------------+----------------------------+---------+--------+-----+
root@ip-10-10-10-150:~# rally task report 46bb408f-b94b-40a7-8a03-ab03f763d5bb --out
20k_1c_1d_boot_server.html
```

Horizon

PERCONA

MIRANTIS
Pure Play OpenStack

All Hypervisors

Hypervisor Summary

VCPU Usage
Used 30,053 of 20,000

Memory Usage
Used 14.7TB of 15.3TB

Disk Usage
Used 29.3TB of 11.4PB

The workload

PERCONA

MIRANTIS
Pure Play OpenStack

- Rally boot_server and boot_and_delete_server scenario
  - boot_server 30k nodes
  - boot_and_delete_server with 3k nodes afterwards
- Most realistic of production workloads for database interactions
- Good mix of read and write queries
- Good stress of lock-heavy queries
- Had to modify rally to skip cleanup

Skipping cleanup in rally

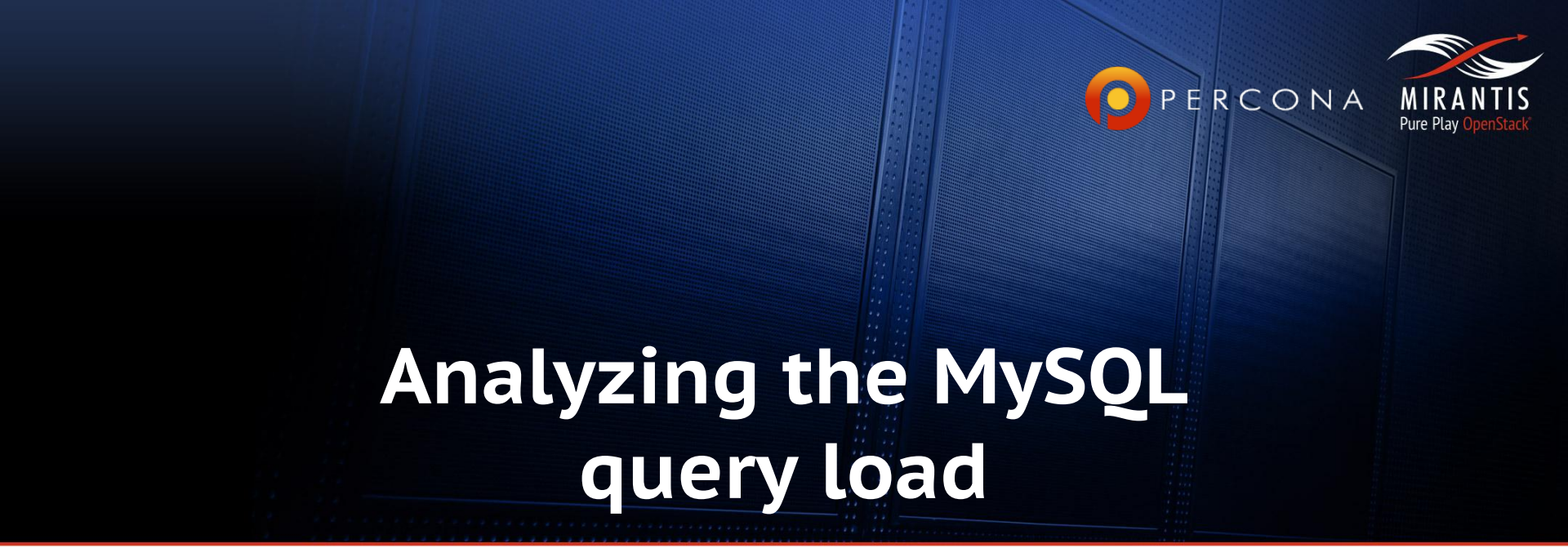
PERCONA

MIRANTIS
Pure Play OpenStack

```
- @base.scenario(context={"cleanup": ["nova"]})
+ @base.scenario(context={"cleanup": [""]})
  def boot_server(self, image, flavor, auto_assign_nic=False,
  **kwargs):
```

Some quirks

- All tenants in Rally scenario use same share network
  - Limitation in way Rally assigns networks during boot
  - https://review.openstack.org/#/c/103306/ coming soon



Analyzing the MySQL query load

pt-query-digest

pt-query-digest is a sophisticated but easy to use tool for analyzing MySQL queries. It can analyze queries from MySQL slow, general, and binary logs, as well as SHOW PROCESSLIST and MySQL protocol data from tcpdump. By default, the tool reports which queries are the slowest, and therefore the most important to optimize. More complex and custom-tailored reports can be created by using options like *--group-by*, *--filter*, and *--embedded-attributes*.

Slow query log event in Percona Server


```
# Time: 141031 14:39:25.542062
# User@Host: cinder[cinder] @ controller [10.10.10.104]  Id:   380
# Schema: cinder  Last_errno: 0  Killed: 0
# Query_time: 0.000191  Lock_time: 0.000042  Rows_sent: 0  Rows_examined: 1  Rows_affected: 1
# Bytes_sent: 52  Tmp_tables: 0  Tmp_disk_tables: 0  Tmp_table_sizes: 0
# InnoDB_trx_id: F7B6
# QC_Hit: No  Full_scan: No  Full_join: No  Tmp_table: No  Tmp_table_on_disk: No
# Filesort: No  Filesort_on_disk: No  Merge_passes: 0
#   InnoDB_IO_r_ops: 0  InnoDB_IO_r_bytes: 0  InnoDB_IO_r_wait: 0.000000
#   InnoDB_rec_lock_wait: 0.000000  InnoDB_queue_wait: 0.000000
#   InnoDB_pages_distinct: 2
SET timestamp=1414766365;
UPDATE services SET updated_at='2014-10-31 14:39:25', report_count=318 WHERE services.id = 1;
```

Slow query log event in Percona Server

PERCONA

MIRANTIS
Pure Play OpenStack

```
$ pt-query-digest  --filter 'print JSON::XS::encode_json($event)' --no-report single_event.log | json_pp
{
   "InnoDB_IO_r_bytes" : "0",
   "Bytes_sent" : "52",
   "Query_time" : "0.000191",
   "pos_in_log" : 0,
   "timestamp" : "1414766365",
   "InnoDB_IO_r_ops" : "0",
   "Rows_sent" : "0",
   "arg" : "UPDATE services SET updated_at='2014-10-31 14:39:25'
            , report_count=318 WHERE services.id = 1\n",
   "Tmp_table_on_disk" : "No",
   "Full_scan" : "No",
   "Merge_passes" : "0",
   "bytes" : 93,
   "Last_errno" : "0",
   "Filesort_on_disk" : "No",
   "QC_Hit" : "No",
   "ts" : "141031 14:39:25",
   "fingerprint" : "update services set updated_at=?,
report_count=? where services.id = ?",
   "InnoDB_IO_r_wait" : "0.000000",
   "Lock_time" : "0.000042",
   "InnoDB_trx_id" : "F7B6",
   "user" : "cinder",
   "InnoDB_rec_lock_wait" : "0.000000",
   "InnoDB_pages_distinct" : "2",
   "Tmp_disk_tables" : "0",
   "host" : "controller",
   "Tmp_tables" : "0",
   "db" : "cinder",
   "cmd" : "Query",
   "Full_join" : "No",
   "Rows_affected" : "1",
   "Filesort" : "No",
   "ip" : "10.10.10.104",
   "Tmp_table_sizes" : "0",
   "Tmp_table" : "No",
   "Killed" : "0",
   "InnoDB_queue_wait" : "0.000000",
   "Rows_examined" : "1"
}
```

pt-query-digest /path/to/slow.log ...

PERCONA MIRANTIS Pure Play OpenStack

- Transactions with the most rows affected
  - --group-by InnoDB_trx_id --order-by Rows_affected:sum > digest_trx_rows_affected.txt
- Transactions with the highest statement count
  - --group-by InnoDB_trx_id --order-by Query_time:cnt > digest_trx_most_stmts.txt
- Largest statements by rows affected
  - --order-by Rows_affected:sum > digest_stmt_rows_affected.txt

pt-query-digest /path/to/slow.log ...

PERCONA

MIRANTIS
Pure Play OpenStack

- Queries that write the most
  - --filter '$event->{fingerprint} !~ m/^select|^set|^commit|^show|^admin/i' --order-by Query_time:cnt --limit 20 > writes-digest.txt
- Queries generating row lock wait
  - --order-by InnoDB_rec_lock_wait:sum > digest-rec_lock_waits.txt

Profile based on time spent in query

PERCONA

MIRANTIS
Pure Play OpenStack

```
# Profile
# Rank Query ID           Response time   Calls  R/Call V/M   Item
# ==== ================== =============== ====== ====== ===== ============
#    1 0xF125BE340342755E 8883.0208 91.2%   1787 4.9709  2.01 SELECT
ipavailabilityranges ipallocationpools
#    2 0x606E846001C42B44  131.4699  1.4%   1432 0.0918  0.25 SELECT ports
#    3 0x4F4306396B18E4FC  113.7931  1.2%   6155 0.0185  1.40 SELECT ports
#    4 0x04866480D062DCEC   74.5814  0.8%   1659 0.0450  0.00 SELECT ports
#    5 0x36F4C83C6705BB3C   71.3762  0.7%   1290 0.0553  0.60 SELECT ports
# MISC 0xMISC             462.9782  4.8% 885151 0.0005   0.0 <160 ITEMS>
```

Profile based on lock wait timeout

PERCONA

MIRANTIS
Pure Play OpenStack

```
# Profile
# Rank Query ID           Response time   Calls  R/Call V/M   Item
# ==== ================== =============== ====== ====== ===== ============
#    1 0xF125BE340342755E 8883.0208 91.2%   1787 4.9709  2.01 SELECT
ipavailabilityranges ipallocationpools
# MISC 0xMISC             779.0427  8.0% 434076 0.0018   0.0 <157 ITEMS>
```

Profile based on lock wait timeout

PERCONA

MIRANTIS
Pure Play OpenStack

```
# Query 1: 35.74 QPS, 177.63x concurrency, ID 0xF125BE340342755E at byte 776833596
# This item is included in the report because it matches --limit.
# Scores: V/M = 2.01
# Time range: 2015-04-11 09:57:01 to 09:57:51
# Attribute    pct   total     min  max  avg  95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count          0    1787
# Exec time     91    8883s   133us      10s       5s   10s       3s   6s
# Lock time     98    8882s    48us      10s       5s   10s       3s   6s
# Rows sent     0   1.75k     1    1    1    1    0    1
# Rows examine    0   3.49k       2    2    2    2    0    2
# Rows affecte    0      0    0    0    0    0    0    0
# Bytes sent  0 768.84k     439  441  440.57  420.77     0  420.77
# Tmp tbl size    0      0    0    0    0    0    0    0
# Query size  0 757.38k     434  434  434  434       0    434
# InnoDB:
# pages distin    0   7.37k       3    7    4.22 5.75 1.15 4.96
# rec lock wai  99   8882s       0    10s       5s   10s       3s   6s
```

Problematic query

PERCONA

MIRANTIS
Pure Play OpenStack

```
SELECT ipavailabilityranges.allocation_pool_id AS ipavailabilityranges_allocation_pool_id,
ipavailabilityranges.first_ip AS ipavailabilityranges_first_ip,
ipavailabilityranges.last_ip AS ipavailabilityranges_last_ip
FROM ipavailabilityranges
INNER JOIN ipallocationpools ON ipallocationpools.id = ipavailabilityranges.
allocation_pool_id
WHERE ipallocationpools.subnet_id = '25ffb478-400e-4cf8-b2ad-525f3b175932'
LIMIT 1 FOR UPDATE;
```

```
+----+-------------+----------------------+------+------------------+-----------+---------
+------------------------------+------+--------------------------+
| id | select_type | table                | type | possible_keys    | key       | key_len | ref
     | rows | Extra                   |
+----+-------------+----------------------+------+------------------+-----------+---------
+------------------------------+------+--------------------------+
|  1 | SIMPLE      | ipallocationpools    | ref  | PRIMARY,subnet_id | subnet_id | 111     | const
     |    1 | Using where; Using index |
|  1 | SIMPLE      | ipavailabilityranges | ref  | PRIMARY           | PRIMARY   | 110     | neutron.
ipallocationpools.id |    1 | Using index              |
+----+-------------+----------------------+------+------------------+-----------+---------
+------------------------------+------+--------------------------+
```

Problematic query

PERCONA

MIRANTIS
Pure Play OpenStack

```
mysql> select * from ipallocationpools;
+--------------------------------------+--------------------------------------+------------+----------------+
| id                                   | subnet_id                            | first_ip   | last_ip        |
+--------------------------------------+--------------------------------------+------------+----------------+
| 7240da3e-826a-48d5-ba1f-efbbc56d49f0 | e55cd523-63e2-49d4-9826-d2a69cbad60d | 10.11.0.10 | 10.11.254.254  |
| ffbb76cf-71cf-457e-a6ef-a90e50ca2962 | 25ffb478-400e-4cf8-b2ad-525f3b175932 | 172.16.0.2 | 172.31.255.254 |
+--------------------------------------+--------------------------------------+------------+----------------+

mysql> select * from ipavailabilityranges;
+--------------------------------------+---------------+----------------+
| allocation_pool_id                   | first_ip      | last_ip        |
+--------------------------------------+---------------+----------------+
| 7240da3e-826a-48d5-ba1f-efbbc56d49f0 | 10.11.0.11      | 10.11.254.254  |
| ffbb76cf-71cf-457e-a6ef-a90e50ca2962 | 172.16.82.254 | 172.31.255.254 |
+--------------------------------------+---------------+----------------+
```

Errors during benchmarks

PERCONA

MIRANTIS
Pure Play OpenStack

Resource <Server: rally_novaserver_gxbgrkswpscgamvj> has ERROR status:

From nova show:

```
| fault | {"message": "Multiple security groups found
matching 'default'. Use an ID to be more specific.",
"code": 409, "created": "2015-04-07T13:19:16Z"} |
```

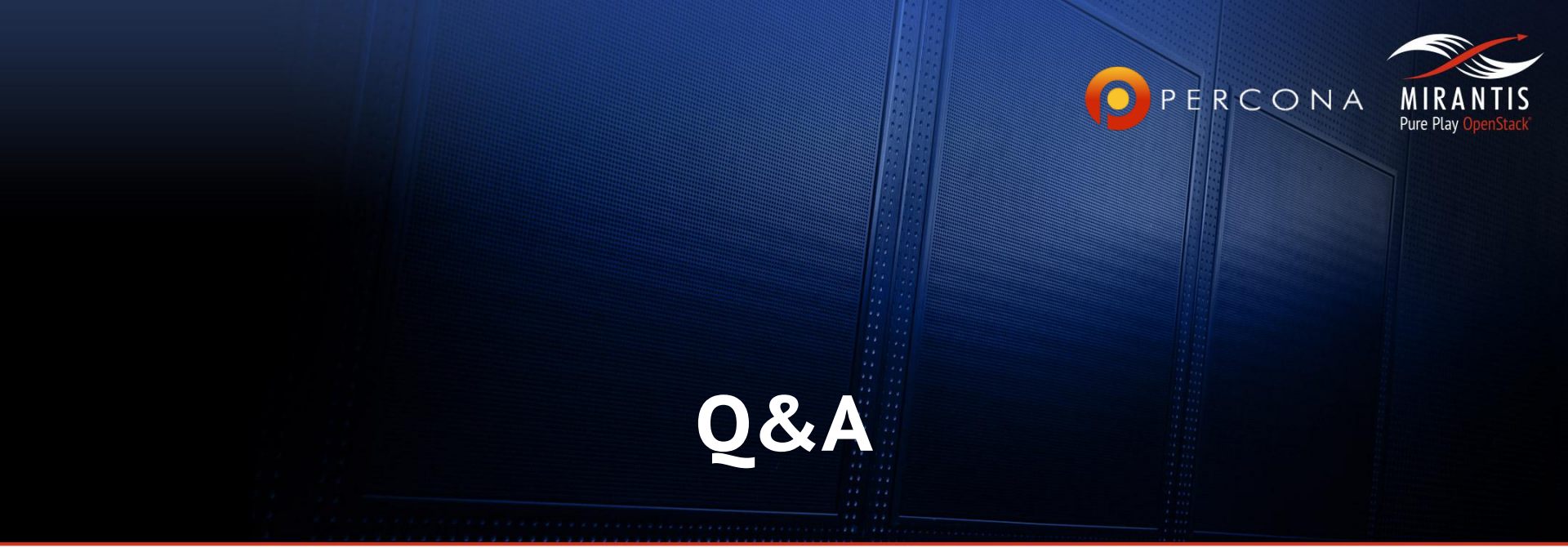
Now some rally reports

Takeaways

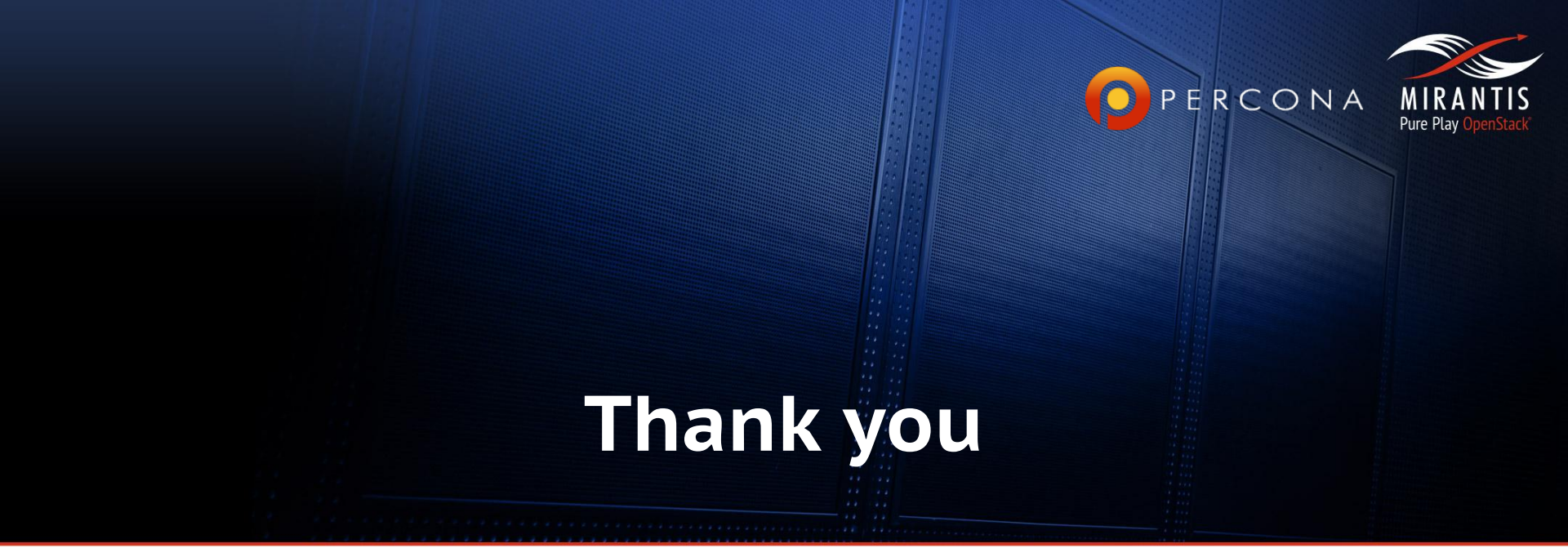
PERCONA

MIRANTIS
Pure Play OpenStack

- The biggest bottleneck at the database level is a database hotspot per tenant network (IP allocation)
- You need to scale the controllers node to see it
  - With 3 active it's somewhat visible already, 5 showed it clearly
  - This will be solved by ip allocation pools in neutron
- All components except neutron are good with using multiple writers
  - wsrep_sync_wait = 1 to read your own writes



Q&A



Thank you

For your time