



Transparent sharding database virtualization engine (DVE)

Amrith Kumar
Peter Boros
2015-04-15



Preparing MySQL servers

2

- DVE will run on port 3306 (the standard MySQL port)
- You must have mysql running on a different port
- DVE's user should have super privileges
 - It needs to create databases
 - In the example we use root

Pointing DVE to the catalog server

3

```
# cd /opt/tesora/dve/server/  
# ./bin/dveconfig.sh  
> locate catalog jdbc:mysql://10.20.20.11:3307 root root  
> delete catalog (optional)  
> setup localhost catalog 1 1  
> set groupService Hazelcast (only needed for multiple DVE)
```

After this, DVE can be started

Starting DVE

4

```
root@ip-10-20-20-12:/opt/tesora/dve/server# ./bin/dvelauncher  
start
```

Starting Tesora (TM) Database Virtualization Engine (R) ...

Product files = /opt/tesora/dve/server

Log files = /opt/tesora/dve/server/logs

Config files = /opt/tesora/dve/server/config

... Tesora (TM) Database Virtualization Engine (R) started (pid 22982).

MySQL client configuration

5

- In `.my.cnf` protocol should be TCP

```
root@ip-10-20-20-12:~# mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 47
```

```
Server version: 5.5.10 Tesora (TM) Database Virtualization  
Engine (R), \
```

```
Version 1.7.0-Devel (ip-10-20-20-12)
```

MySQL client configuration

6

- In .my.cnf protocol should be TCP

```
root@ip-10-20-20-12:~# mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 47
```

```
Server version: 5.5.10 Tesora (TM) Database Virtualization  
Engine (R), \
```

```
Version 1.7.0-Devel (ip-10-20-20-12)
```

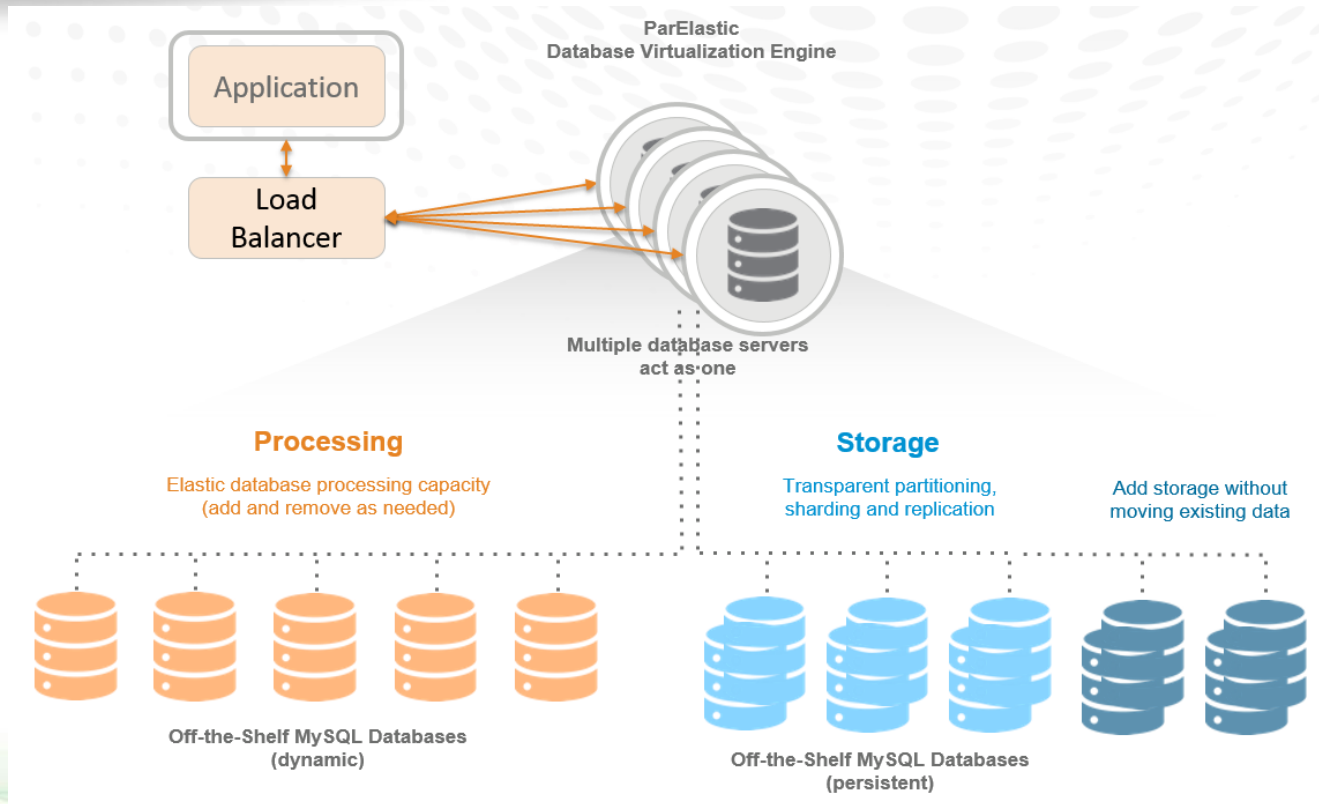

What exactly is Tesora DVE?

7

- Automatic transparent sharding
 - Parallel query planning and execution
 - Horizontally scalable
 - Elastic
-
- Fully Open Source

DVE Terminology

8



Adding a first shard

9

- Create a single-node persistent site

```
mysql> create persistent site sb1 URL='jdbc://10.20.20.13:3307'
```

```
User='root' Password='root';
```

```
Query OK, 0 rows affected (0.11 sec)
```

Adding a first shard

10

- Create a single-node persistent site

```
mysql> create persistent site sb1 URL='jdbc://10.20.20.13:3307'
```

```
User='root' Password='root';
```

```
Query OK, 0 rows affected (0.11 sec)
```

- And a persistent group

```
mysql> create persistent group sb_group add sb1;
```

```
Query OK, 0 rows affected (0.05 sec)
```

Checking it twice

11

- Checking persistent sites

```
mysql> show persistent sites\G
***** 1. row *****
Persistent_Site: sb1
  HA_Type: Single
  Master_Url: jdbc:mysql://10.20.20.13:3307
***** 2. row *****
Persistent_Site: sitel
  HA_Type: Single
  Master_Url: jdbc:mysql://10.20.20.11:3307?useUnicode=
(truncated)
```

- Checking persistent groups

```
mysql> show persistent groups;
```

+-----+-----+	
Name	Latest_generation
+-----+-----+	
DefaultGroup	3
InformationSchemaGroup	1
sb_group	4
SystemGroup	2
+-----+-----+	

```
4 rows in set (0.08 sec)
```

Create a template

13

```
mysql> create template sysbench xml='<?xml version="1.0"?>
    '> <template name="sysbench">
    '>     <requirement>
    '>         <declaration>create range if not exists id_range
(integer) persistent group #sg#
    '>         </declaration>
    '>     </requirement>
    '>
    '>     <tabletemplate range="id_range" match="sbtest.*" model="Range">
    '>         <column>id</column>
    '>     </tabletemplate>
    '> </template>';
```

Query OK, 0 rows affected (0.17 sec)

How data is distributed

14

- Many ways to distribute data
 - Broadcast
 - Deterministic
 - Range
 - Map
- Specifying distribution
 - Through DDL
 - Through templates

Data distribution and elasticity

15

- You can add sites to a group
- Data will still be distributed
- How?
 - Broadcast tables [trivial]
 - Distributed tables [hard]

Dynamic data distribution

16

- Multiple patents (8478790, 8386532, ...)
- Relies on a novel data structure
- Moving from Yes/No → No/Maybe

Create schema and table

17

```
mysql> create database sbtest default persistent
group sb_group using template sysbench;
Query OK, 1 row affected (0.12 sec)
mysql> create range sb_id (integer) persistent
group sb_group;
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> show ranges;
```

Range	Persistent_Group	Signature
sb_id	sb_group	int

1 row in set (0.02 sec)

```
# sysbench --test=sysbench/tests/db/oltp.lua --
mysql-host=10.20.20.12 --mysql-user=root --mysql-
password=root --oltp-table-size=10000 --oltp-
tables-count=1 --mysql-table-options="persistent
group sb_group range distribute on (id) using
sb_id" prepare
```

Adding a shard

18

```
mysql> alter persistent group sb_group add generation sb2;  
Query OK, 1 row affected (0.12 sec)
```

OR

```
mysql> create persistent group sb_group2 add sb1, sb2;  
Query OK, 2 row affected (0.14 sec)  
mysql> create range sb_id2 (integer) persistent group sb_group2;  
Query OK, 0 rows affected (0.04 sec)
```

In the later case, move the table to the new persistent group.

We tried to do the second one with pt-online-schema-change.

pt-online-schema-change doesn't work...

19

- `SELECT @@server_id` is not supported
- Explain output is different, which is used by the chunking mechanism
- `<=>` operator is not supported (used by triggers)
- `INSERT IGNORE` is not supported
- Default load monitoring doesn't work because there is no `Threads_running` at the proxy level.

None of those are non-addressable issues. Either in `pt-online-schema-change` or DVE.

Speaking of EXPLAIN

20

```
mysql> explain select * from sbtest1 where id=3000\G
***** 1. row *****
      Type: SELECT
    Target_group: NULL
    Target_table: NULL
    Target_dist:
Target_index_hints: NULL
      Other: found distribution key match
      SQL: SELECT `sbtest1`.`id` AS s1i0_6,`sbtest1`.`k` AS
s1k1_7,`sbtest1`.`c` AS s1c2_8,`sbtest1`.`pad` AS s1p3_9 FROM
`sbtest1` WHERE `sbtest1`.`id` = 3000
1 row in set (0.01 sec)
```

See where the data comes from

21

```
mysql> select id,k,@dve_sitename from sbtest1 where id between 3000 and 3010;
```

id	k	@dve_sitename
3000	4517	sb1
3002	5011	sb1
3004	4957	sb1
3006	5028	sb1
3008	5428	sb1
3010	5023	sb1
3001	5021	sb2
3003	4995	sb2
3005	4961	sb2
3007	5014	sb2
3009	4979	sb2

```
11 rows in set (0.02 sec)
```


What DVE does

22

- Receive Query, Parse Query, Validate Syntax, Generate Query Parse Tree
- Identify tables (views), joins, aggregations, restrictions, sorts, ...
- Consult metadata
 - Where is the data
 - What restrictions
 - What indices can be used
 - What joins, aggregations, sorts, ... being done
- Make MPP Query Plan and get MySQL to execute.
- MySQL returns results
- Simple 🤖

What DVE does

23

```
Select count (*)
```

```
From CUSTOMER;
```

```
count (*)
```

```
-----
```

```
2771
```

```
(1 row affected)
```

```
PROVISION 1 DYNAMIC SITE
```

```
ON DYNAMIC SITE CREATE TEMP TABLE T1 ( C int );
```

```
ON ALL STORAGE SITES
```

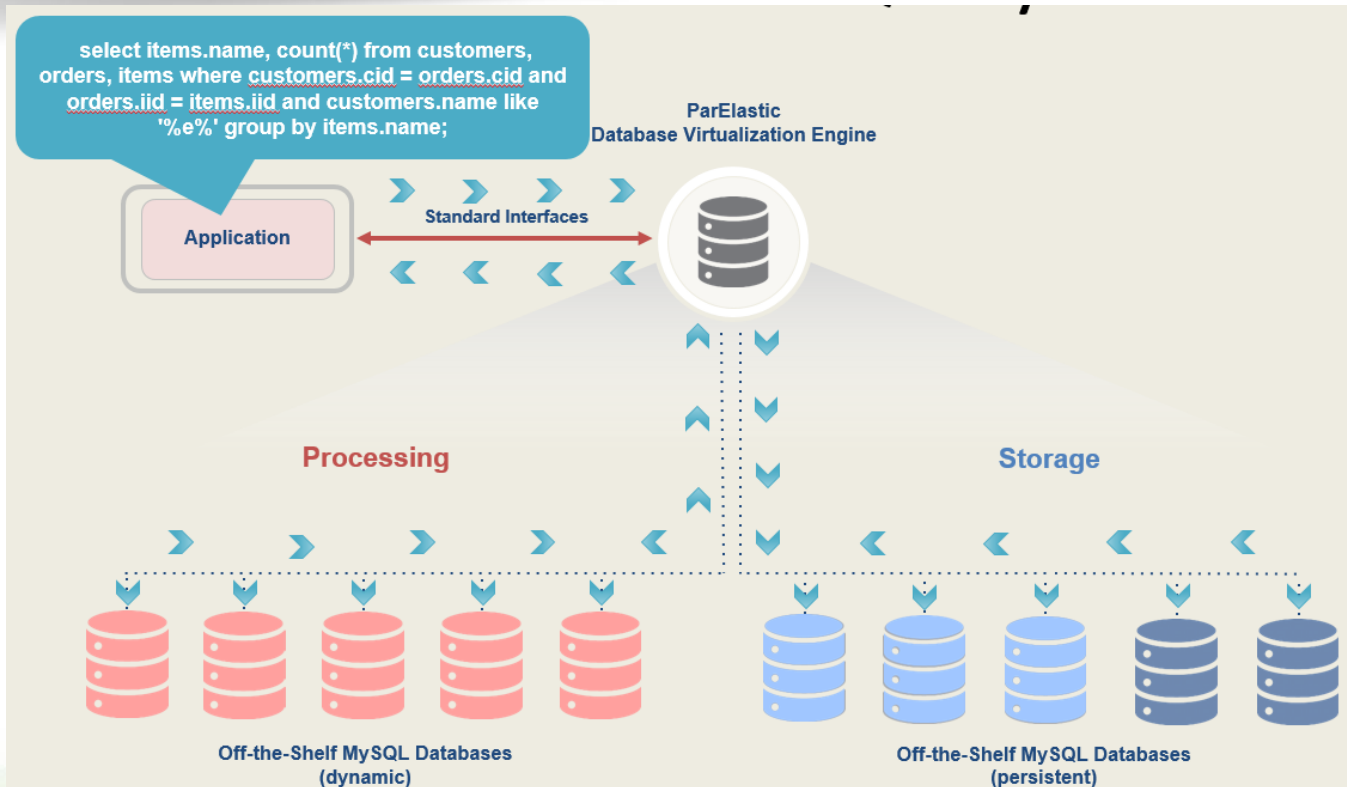
```
SELECT COUNT(*) FROM CUSTOMER  
AND REDISTRIBUTE TO T1
```

```
ON DYNAMIC SITE
```

```
SELECT SUM(C) FROM T1;
```

What DVE does

24

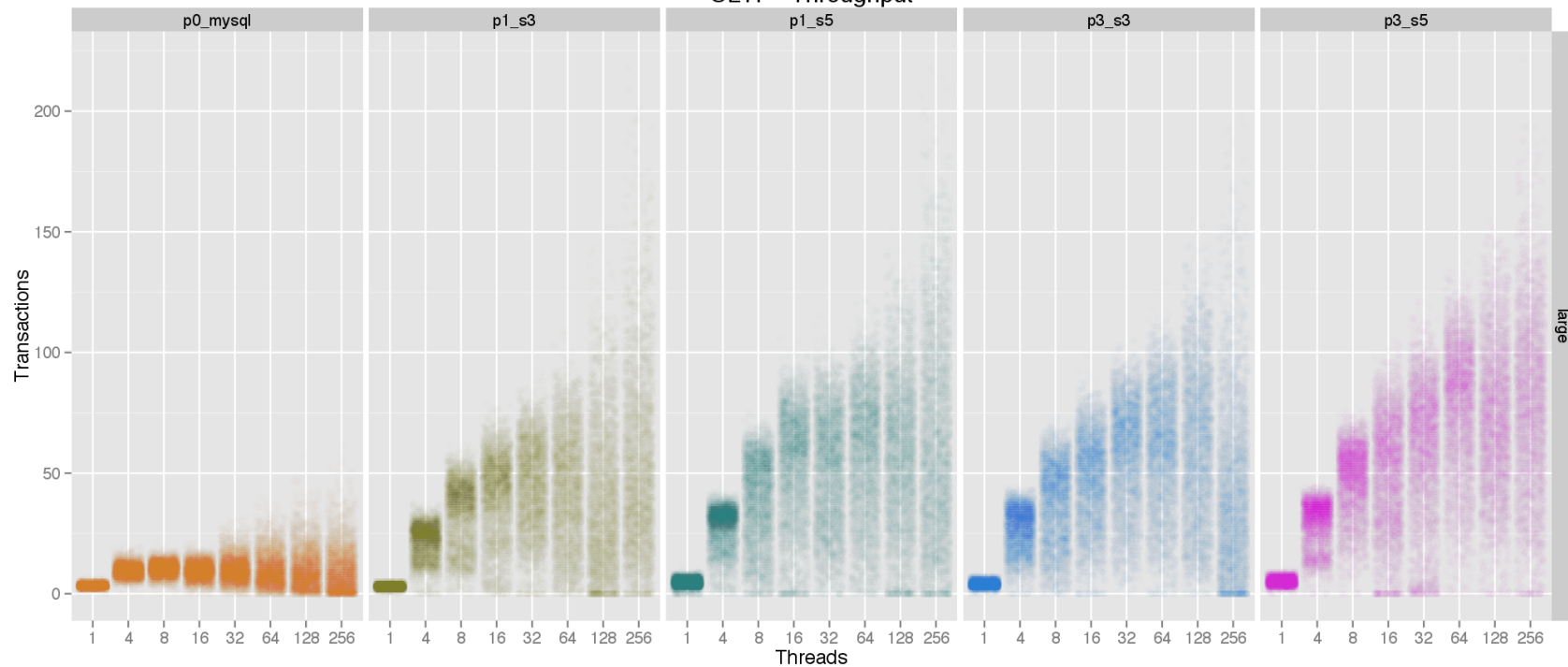


- Transparent sharding, parallel query execution
 - Complex queries including cross site joins, aggregations, sorts, ...
- Full support for views
- Referential Integrity (constraints)
- Elasticity
- Fully Open Source

DVE throughput (sysbench OLTP)

26

OLTP - Throughput



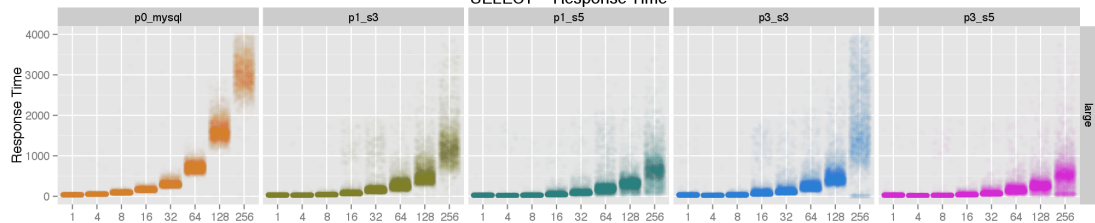
Architecture • p0_mysql • p1_s3 • p1_s5 • p3_s3 • p3_s5

DVE throughput (sysbench OLTP)

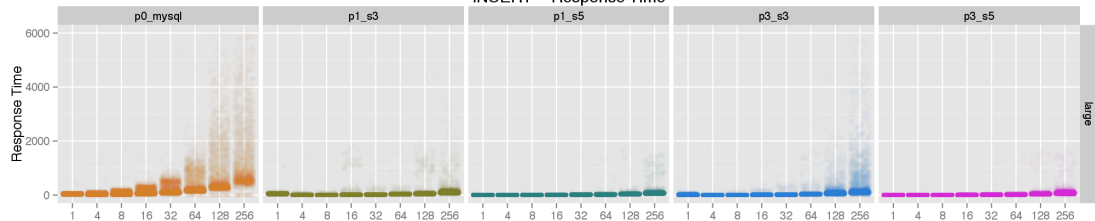
27

SELECT,INSERT,UPDATE - Response Time

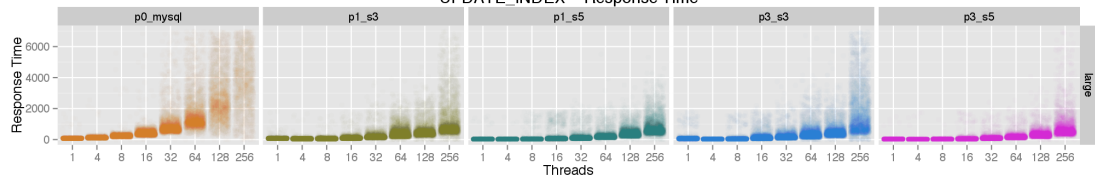
SELECT - Response Time



INSERT - Response Time



UPDATE_INDEX - Response Time



Architecture • p0_mysql • p1_s3 • p1_s5 • p3_s3 • p3_s5



Q&A



PERCONA
LIVE



Thank you



PERCONA
LIVE