



FOSDEM

MariaDB 10 - The Spider Storage Engine (a sharding plugin for MySQL/MariaDB)

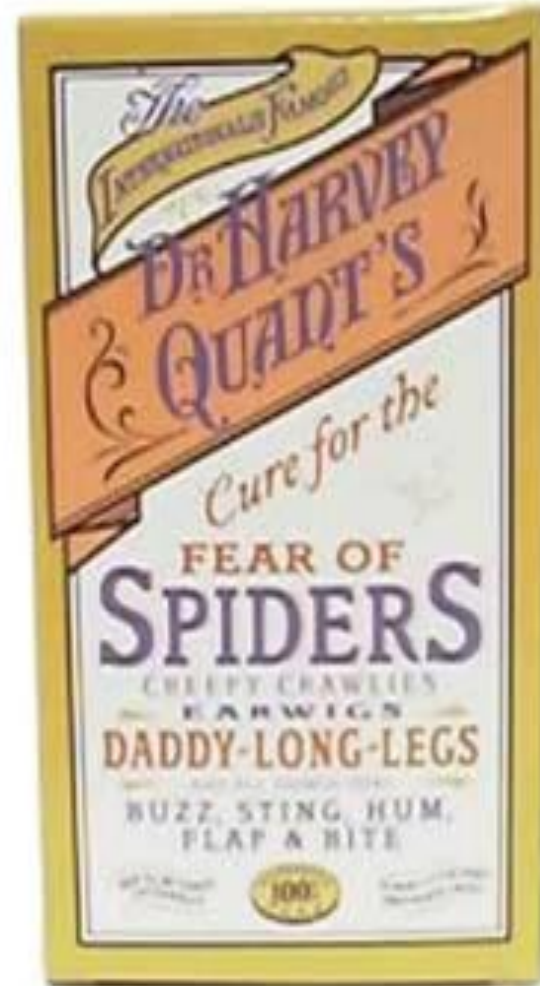
Stéphane Varoqui <stephane@skysql.com>

Colin Charles <byte@mariadb.com>

Introduction

- ❑ Fear of Databases
- ❑ Fear of Sharding
- ❑ Fear of Clustering
- ❑ Fear of Map/Reduce
- ❑ Fear of Spiders

TAKE A PILE !



What is sharding?



- ❑ **“Sharding” is breaking the database down into pieces**
- ❑ **Replication scales for reads, but what about writes?**
- ❑ **Horizontal partitioning is what SPIDER provides**
- ❑ **Storage engine, developed by Kentoku Shiba, associates a partition with a remote server**
- ❑ **Transparent to user, independent from application**

Quick view at MariaDB 10



❑ Per Table Local Storage Engines

InnoDB, TokuDB, LevelDB, OQGraph

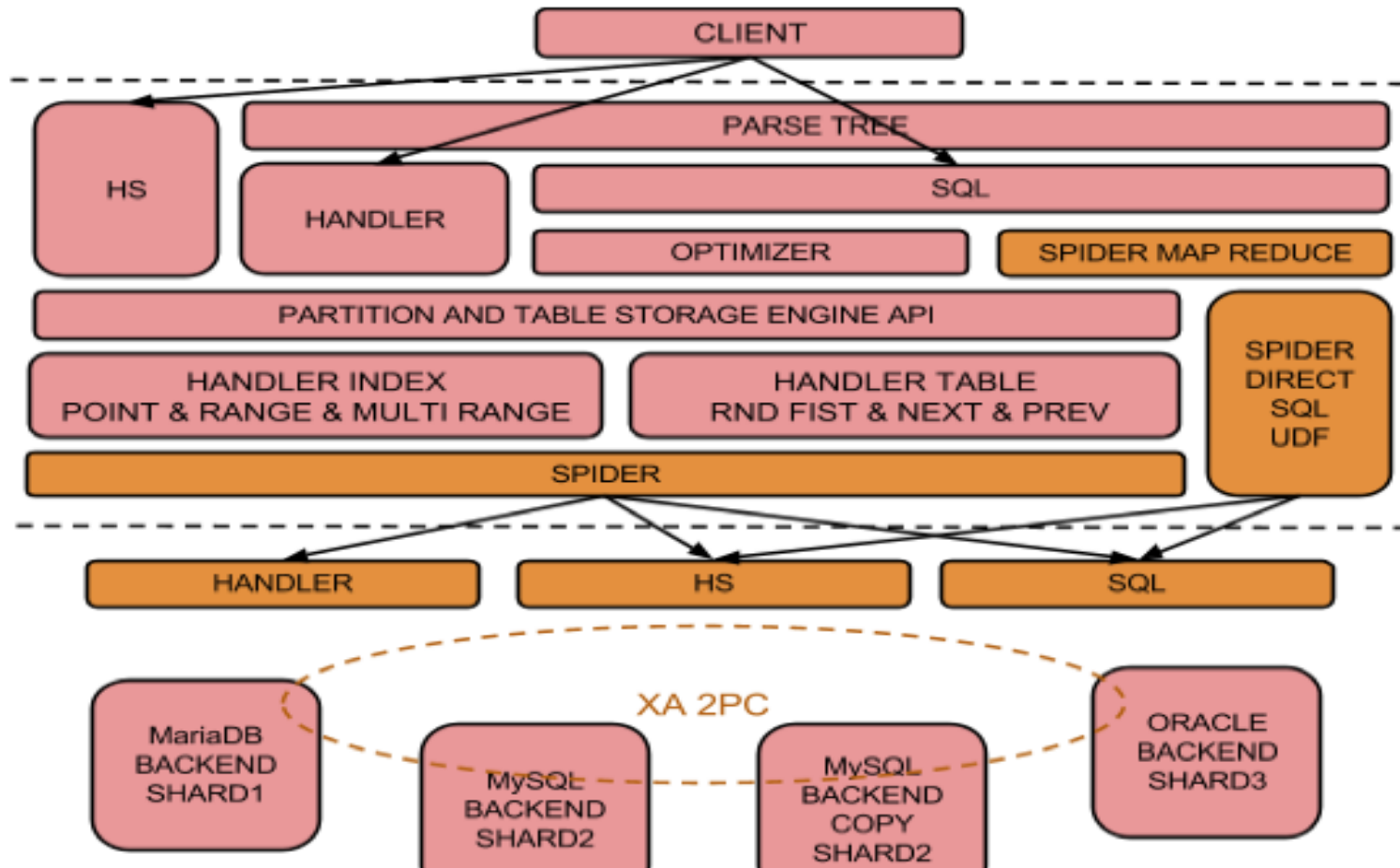
❑ Per Table Sharding and Federation

Spider CONNECT, FederatedX, Cassandra, HBase, Mroonga, SphinxSE

❑ Per Table Replication

Multi-source, parallel, filtering

Spider - It's a Storage Engine



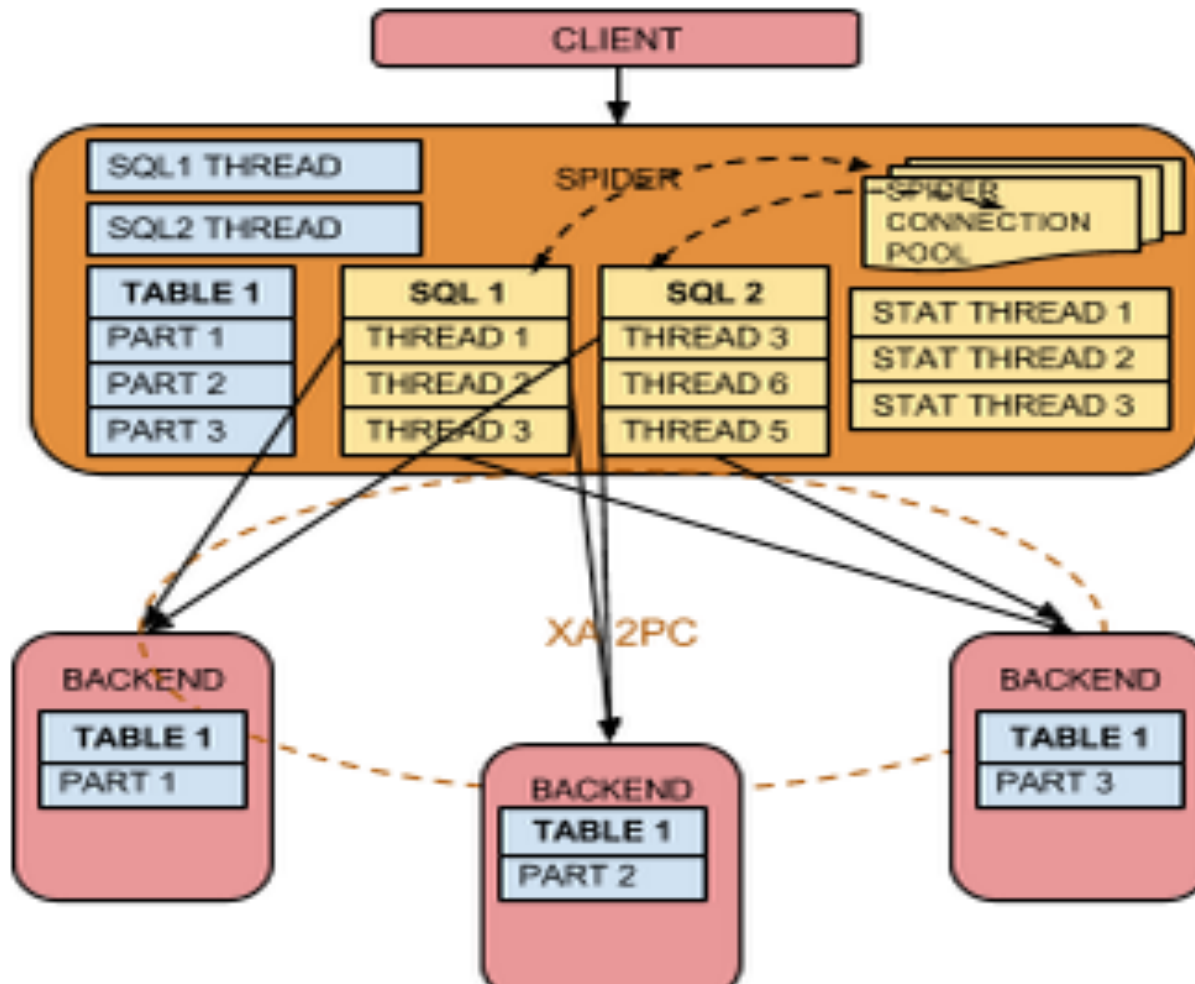
ha_spider.cc SEMI TRX



additional_lock analyze append_condition_sql_part append_delete_all_rows_sql_part append_delete_sql_part
append_direct_update_set_hs_part append_direct_update_set_sql_part append_dup_update_pushdown_sql_part
append_hint_after_table_sql_part append_increment_update_set_sql_part append_insert_sql_part append_insert_values_hs_part
append_insert_values_sql_part append_into_sql_part append_key_column_value_sql_part append_key_order_for_direct_order_limit_with_alias_sql_part
append_key_order_for_merge_with_alias_sql_part append_key_order_with_alias_sql_part append_key_select_sql_part append_key_where_hs_part
append_key_where_sql_part append_limit_hs_part append_limit_sql_part append_match_select_sql_part append_match_where_sql_part
append_minimum_select_sql_part append_multi_range_cnt_sql_part append_select_lock_sql_part append_select_lock_where_sql_part
append_table_select_sql_part append_tmp_table_and_sql_for_bka append_union_all_end_sql_part append_union_all_start_sql_part
append_update_columns_sql_part append_update_set_sql_part append_update_set_sql_part append_update_set_sql_part
append_update_sql_part append_values_connector_sql_part append_values_terminator_sql_part auto_repair auto_repair
bas_ext bulk_req_exec bulk_tmp_table_created bulk_tmp_table_end_bulk_insert bulk_tmp_table_insert bulk_tmp_table_rnd_init
bulk_tmp_table_rnd_next bulk_update_row check_access_kind check_and_error check_and_repair check_and_start_bulk_update
check_crdr check_direct_order_limit check_error_mode check_error_mode check_error_mode check_error_mode check_error_mode
check_insert_dup_update_pushdown check_item_type_sql check_pre_call check_select_column bool check_select_column bool
rndcheck update_columns_sql_part clear_handler_opened clone close close_opened_handler cmp_ref cond cond cond cond cond
create_bulk_access_link delete_all_rows delete_bulk_access_link delete_row delete_table direct_delete_row direct_delete_row
direct_update_rows direct_update_rows_init disable_indexes drop_tmp_tables enable_indexes end_bulk_delete end_bulk_delete
end_bulk_update estimate_rows_upper_bound exec_bulk_update external_lock extra_ft end_ft_init ft_init ft_init ft_init ft_init
get_auto_increment get_error_message get_table handler_opened index_end index_first index_first_internal index_handler_init
index_init index_last index_last_internal index_next index_next_same index_prev index_prev index_prev index_prev index_prev
index_read_last_map_internal index_read_map index_read_map_internal index_type info info_push is_bulk is_bulk is_bulk is_bulk
is_crashed is_fatal_error keys_to_use_for_scanning mk_bulk_tmp_table_and_bulk_start multi_range_read_info multi_range_read_info
multi_range_read_info_const multi_range_read_init multi_range_read_next multi_range_read_next_first multi_range_read_next_first
multi_range_read_next_first multi_range_read_next_first multi_range_read_next_first multi_range_read_next_first multi_range_read_next_first
open optimize position pre_direct_delete_rows pre_direct_delete_rows_init pre_direct_update_rows pre_direct_update_rows
pre_ft_read pre_index_end pre_index_first pre_index_init pre_index_last pre_index_read_last_map pre_index_read_last_map
pre_read_multi_range_first pre_read_range_first pre_rnd_end pre_rnd_init pre_rnd_next pre_write_row print print print print
push_back_hs_upds read_multi_range_first read_multi_range_first_internal read_multi_range_next read_multi_range_next
read_multi_range_next read_multi_range_next read_multi_range_next read_multi_range_next read_multi_range_next read_multi_range_next
read_range_first_internal read_range_next read_time reappend_limit_sql_part records records_in_range records_in_range records_in_range
rename_table repair reset reset_auto_increment reset_first_link_idx reset_hs_keys reset_hs_sql reset_hs_sql reset_hs_sql
reset_hs_upds reset_sql_sql reuse_tmp_table_and_sql_for_bka rm_bulk_tmp_table rnd_end rnd_handler_init rnd_handler_init
rnd_next_internal rnd_pos scan_time set_clone_searched_bitmap set_error_mode set_first_link_idx set_handler set_handler
set_insert_to_pos_sql set_order_pos_sql set_order_to_pos_sql set_searched_bitmap set_select_column_mode set_select_column_mode
set_where_to_pos_sql sql_is_empty sql_is_filled_up start_bulk_delete start_bulk_insert start_bulk_update start_bulk_update
support_bulk_access_hs support_bulk_update_sql support_multi_split_read_sql support_use_handler_sql sync sync sync sync sync
sync_from_clone_source_base table_cache_type table_flags table_type truncate ucheck_partitioned umax umax umax umax umax umax
umax supported key_part length umax supported key_parts umax supported keys umax supported keys umax supported keys

Threading Model

It's a per table connection pool

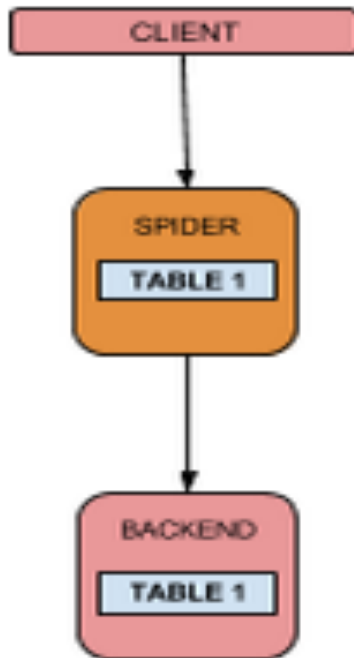


It's already a proxy but there is more

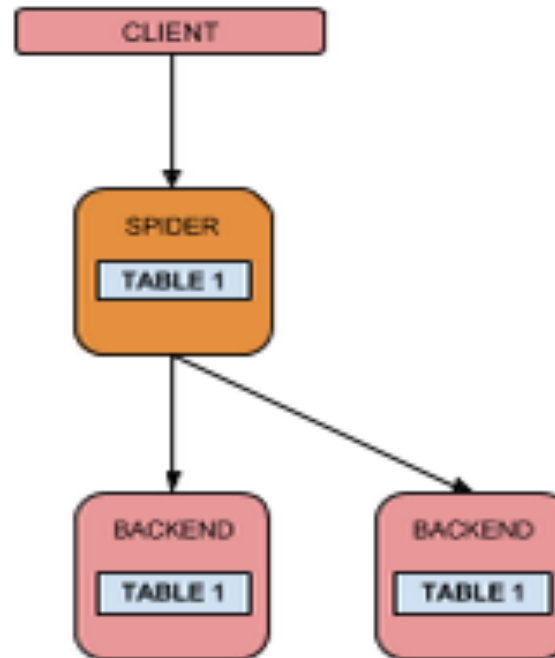
Maintain consistency between replicants in 2PC



A - FEDERATION



B - FEDERATION & HA



Federation Howto



```
spider1 << EOF
CREATE SERVER backend
  FOREIGN DATA WRAPPER mysql
OPTIONS (
  HOST '192.168.0.202',
  DATABASE 'test',
  USER 'skysql',
  PASSWORD 'skyvodka',
  PORT 5054
);

CREATE TABLE test.sbtest
(
  id int(10) unsigned NOT NULL AUTO_INCREMENT,
  k int(10) unsigned NOT NULL DEFAULT '0',
  c char(120) NOT NULL DEFAULT '',
  pad char(60) NOT NULL DEFAULT '',
  PRIMARY KEY (id),
  KEY k (k)
) ENGINE=spider COMMENT='wrapper "mysql",srv "backend"';
SELECT * FROM test.sbtest LIMIT 10;
EOF
```

FEATURES - Execution Flow



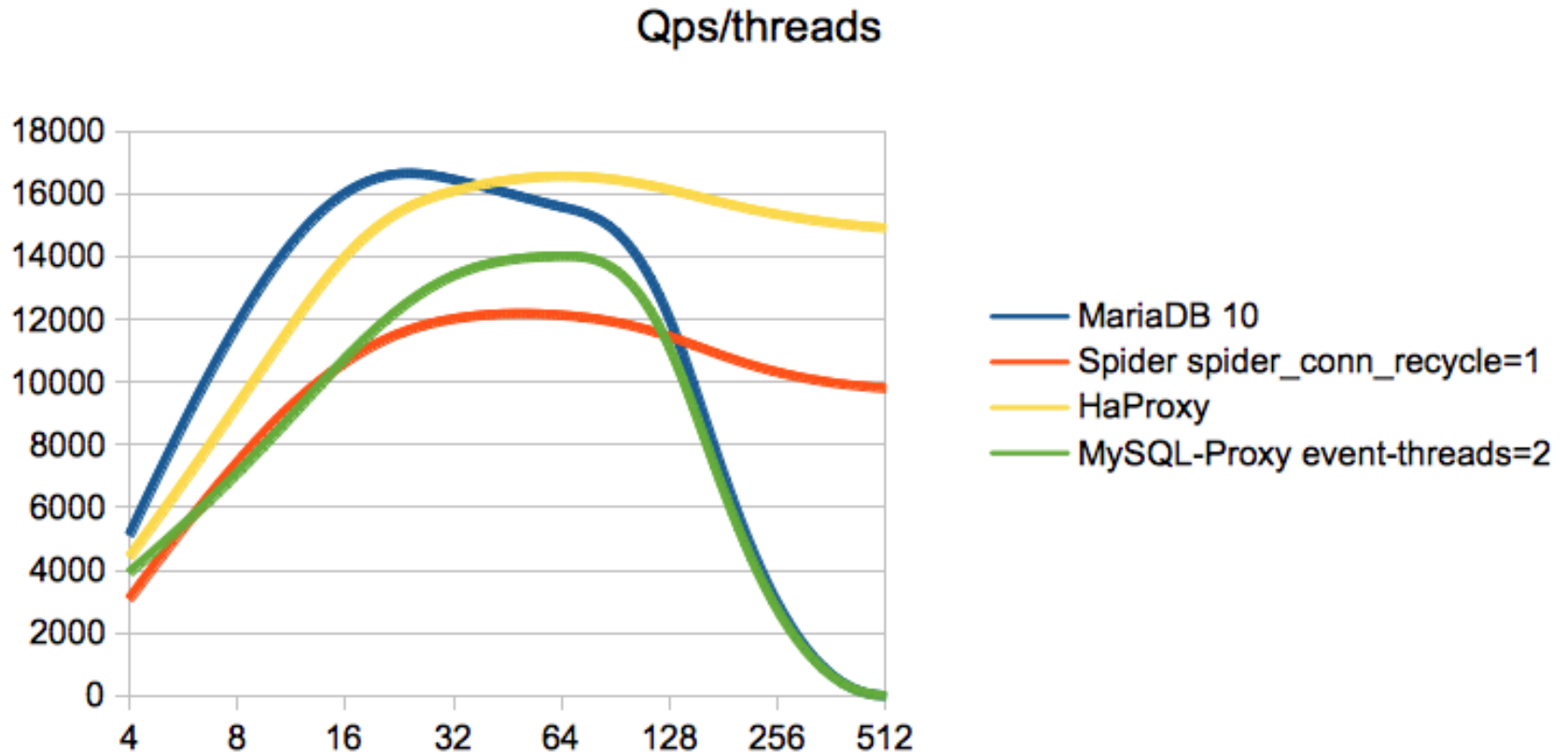
Execution Control	Spider	10	RC
Configuration at table and partition level, settings can change per data collection	Yes	Yes	Yes
Configurable empty result set on errors. For API that does not have transactions replay	Yes	Yes	Yes
Query Cache tuning per table of the on remote backend	Yes	Yes	Yes
Index Hint per table imposed on remote backend	Yes	Yes	Yes
SSL connections to remote backend connections	Yes	Yes	Yes
Table definition discovery from remote backend	Yes	F(*)	?
Direct SQL execution to backend via UDF	Yes	Yes	Yes
Table re synchronization between backends via UDF	Yes	Yes	Yes
Maintain Index and Table Statistics of remote backends	Yes	Yes	Yes
Can you Independent Index and Table Statistics	No	Yes	Yes
Maintain local or remote table increments	Yes	Yes	Yes
LOAD DATA INFILE translate to bulk inserting	Yes	Yes	Yes

FEATURES - Performance

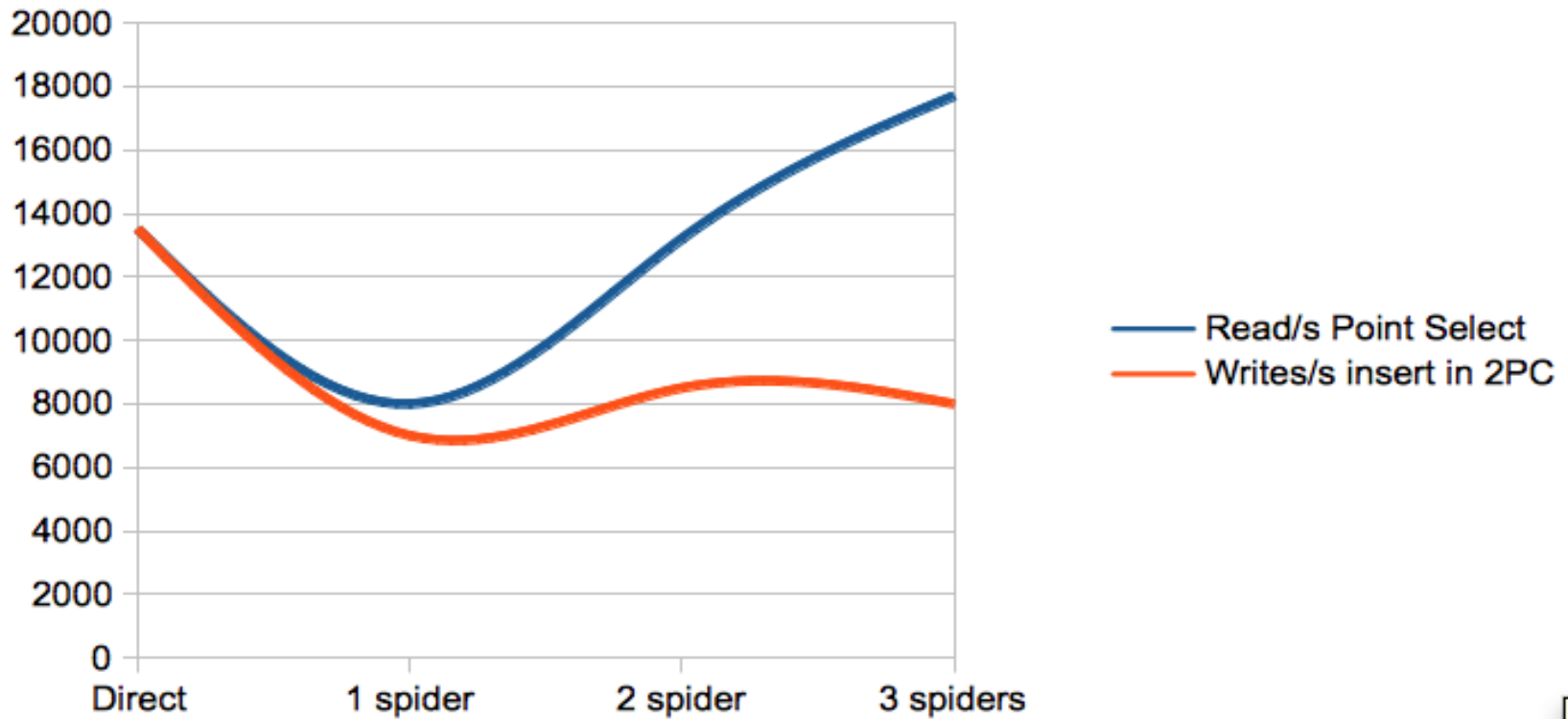


Features	Spider	10	RC
Engine Condition Pushdown	Yes	F(*)	Yes
Concurrent partition scan	Yes	No	Yes
Batched key access	Yes	P(*)	Yes
Block hash join	No	Yes	Yes
HANDLER backend propagation	Yes	F(*)	Yes
HANDLER backend translation from SQL	Yes	F(*)	Yes
HANDLER OPEN cache per connection	No	No	Yes
HANDLER use prepared statement	No	No	Yes
HANDLER_SOCKET protocol backend propagation	Yes	No	Yes
HANDLER_SOCKET backend translation from SQL	No	No	No
Map reduce for ORDER BY ... LIMIT	Yes	Yes	Yes
Map reduce for MAX & MIN & SUM & GROUP BY	Yes	No	Yes
Batch multiple WRITES in auto commit to reduce network round trip	Yes	Yes	Yes
Relaxing backend consistency	Yes	Yes	Yes

SPIDER - Read Only Sysbench



SPIDER - POINT UPDATE & SELECT

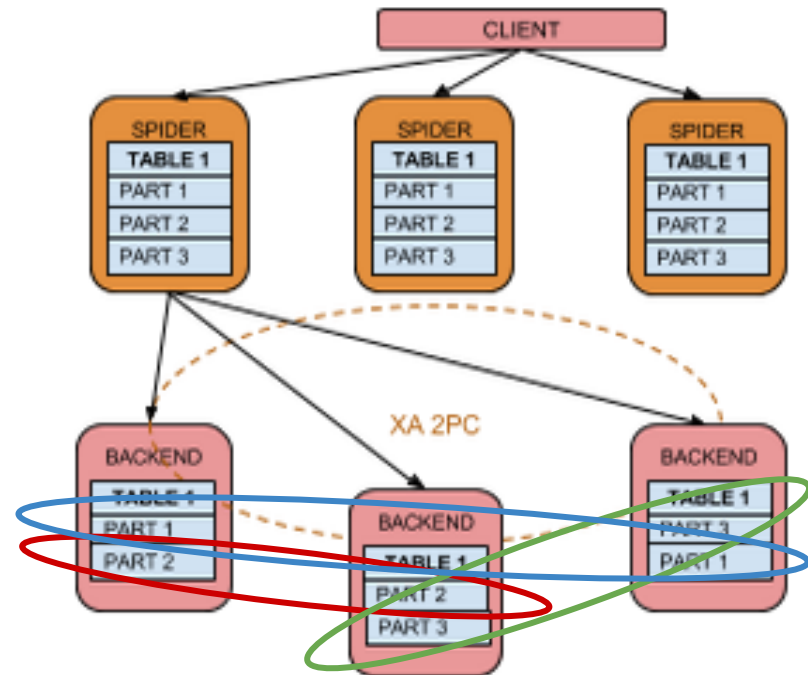
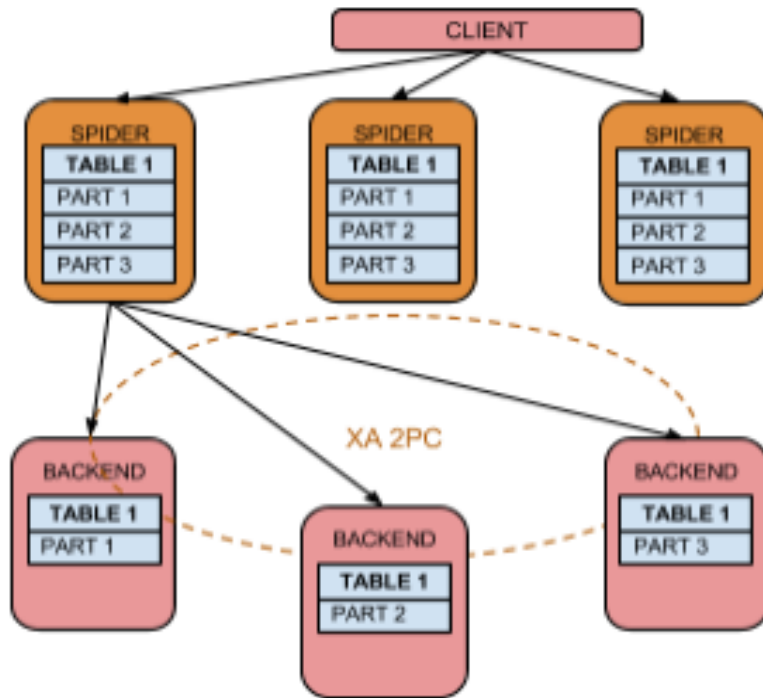


Spider - A Sharding and Clustering Solution



C - SHARDING

D - SHARDING & HA



3 Node Groups

HOWTO - Sharding & HA



```
CREATE TABLE backend.sbtest
(
  id int(10) unsigned NOT NULL AUTO_INCREMENT,
  k int(10) unsigned NOT NULL DEFAULT '0',
  c char(120) NOT NULL DEFAULT '',
  pad char(60) NOT NULL DEFAULT '',
  PRIMARY KEY (id),
  KEY k (k)
) ENGINE=spider COMMENT='wrapper "mysql", table "sbtest"'
PARTITION BY KEY (id) (
  PARTITION pt1 COMMENT = 'srv "backend1 backend2_rpl" mbk "2", mkd "2", msi "5054",
link_status "0 0"',
  PARTITION pt2 COMMENT = 'srv "backend2 backend1_rpl" mbk "2", mkd "2", msi "5054",
link_status "0 0" ');

CREATE SERVER mon
  FOREIGN DATA WRAPPER mysql
OPTIONS(
  HOST '192.168.0.201',
  DATABASE 'backend',
  USER 'skysql',
  PASSWORD 'skyvodka',
  PORT 5054
);

INSERT INTO `mysql`.`spider_link_mon_servers` VALUES
('%','%','%',5054,'mon',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,0,NULL,NULL);

SELECT spider_flush_table_mon_cache();
```

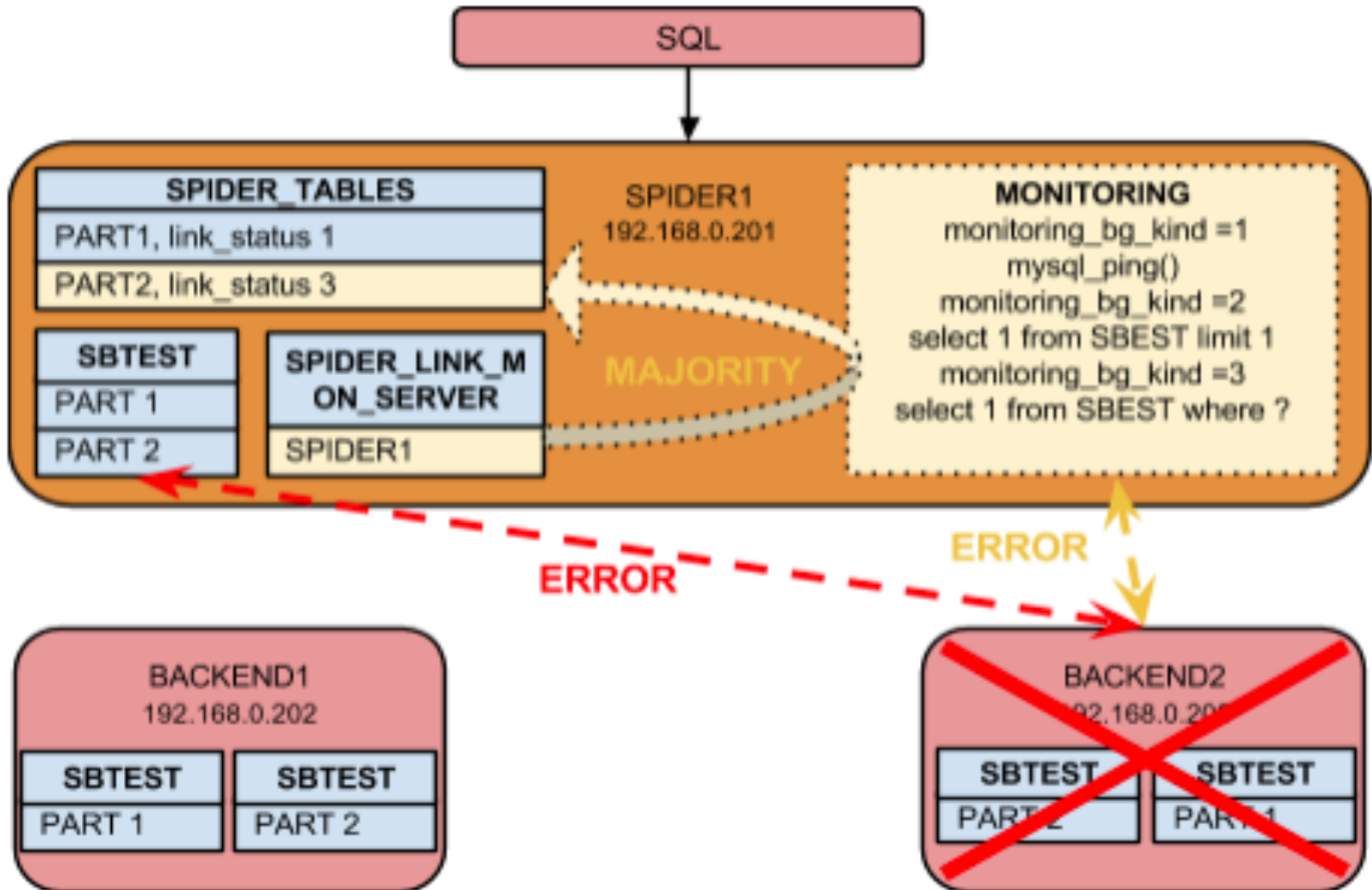
FEATURES - HA



Clustering and High Availability	Spider	10	RC
Commit, Rollback transactions on multiple backend	Yes	Yes	Yes
Multiplexing to a number of replicas using xa protocol 2PC	Yes	Yes	Yes
Split brain resolution based on a majority decision, failed node is remove from the list of replicas	Yes	Yes	Yes
Enable a failed backend to re-enter the cluster transparently	No	No	No
Synchronize DDL to backend, table modification, schema changes	No	No	No
Synchronize DDL to other Spider	No	No	No
Transparent partitioning	No	No	No
Heterogenous Backends			
MariaDB and MySQL database backend	Yes	Yes	Yes
Oracle database backend, if build from source against the client library 'ORACLE_HOME'	Yes	Yes	Yes
Local table attachment	Yes	Yes	Yes

HOWTO - Sharding & HA

Node failure



HOWTO - Sharding & HA

Reintroducing failed node



```
ALTER TABLE backend.sbtest
ENGINE=spider COMMENT='wrapper "mysql", table "sbtest"'
PARTITION BY KEY (id)
(
PARTITION pt1 COMMENT = 'srv "backend1 backend2_rpl" mbk "2", mkd "2", msi "5054", link_status "2 0"',
PARTITION pt2 COMMENT = 'srv "backend2 backend1_rpl" mbk "2", mkd "2", msi "5054", link_status "0 2" '
) ;

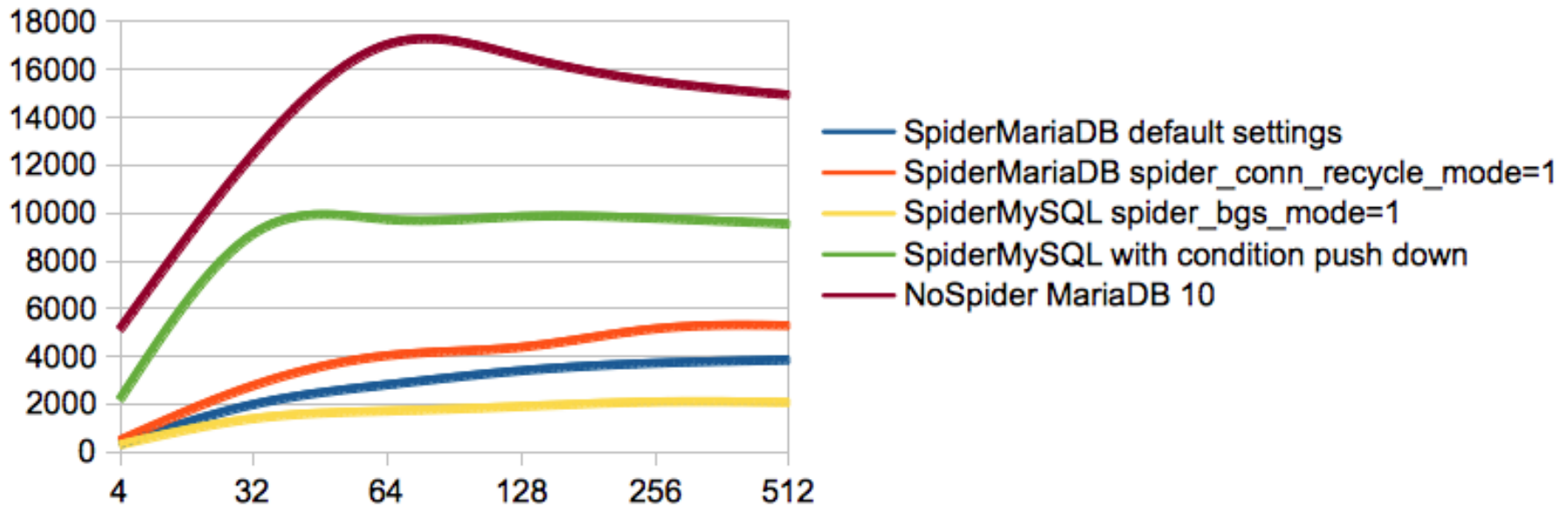
select spider_copy_tables('backend.sbtest#P#pt1','0','1');
select spider_copy_tables('backend.sbtest#P#pt2','1','0');

ALTER TABLE backend.sbtest
ENGINE=spider COMMENT='wrapper "mysql", table "sbtest"'
PARTITION BY KEY (id)
(
PARTITION pt1 COMMENT = 'srv "backend1 backend2_rpl" mbk "2", mkd "2", msi "5054", link_status "0 0"',
PARTITION pt2 COMMENT = 'srv "backend2 backend1_rpl" mbk "2", mkd "2", msi "5054", link_status "0 0" '
) ;
```

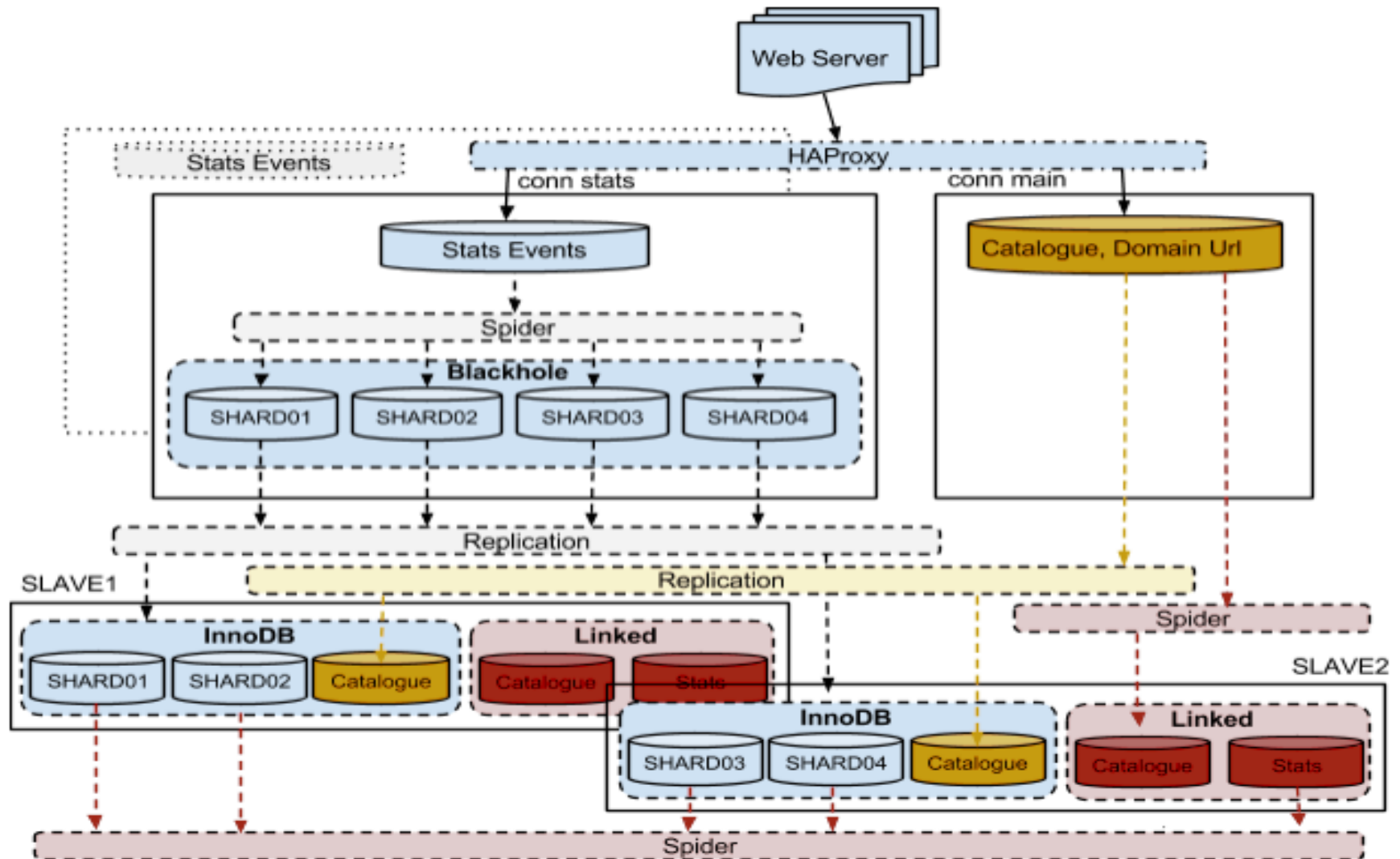
Read Only Sysbench



Qps/Threads



SPIDER - Asynchronous Writes at Scale



WRITE Performance settings



```
default-storage-engine=MyISAM
skip-innodb
skip_name_resolv
back_log=1024
max_connections = 1024
table_open_cache = 4096
table_definition_cache = 2048
max_allowed_packet = 32K
binlog_cache_size = 32K
max_heap_table_size = 64M
thread_cache_size = 1024
query_cache_size = 0
expire_logs_days=4
progress_report_time=0
Binlog_ignore_db=ccmstats
```

```
spider_use_handler=1
spider_sts_sync=0
spider_remote_sql_log_off=1
spider_remote_autocommit=0
spider_direct_dup_insert=1
spider_local_lock_table=0
spider_support_xa=0
spider_sync_autocommit=0
spider_sync_trx_isolation=0
spider_crd_sync=0
spider_conn_recycle_mode=1
spider_reset_sql_alloc=0
```

Double Partitioning



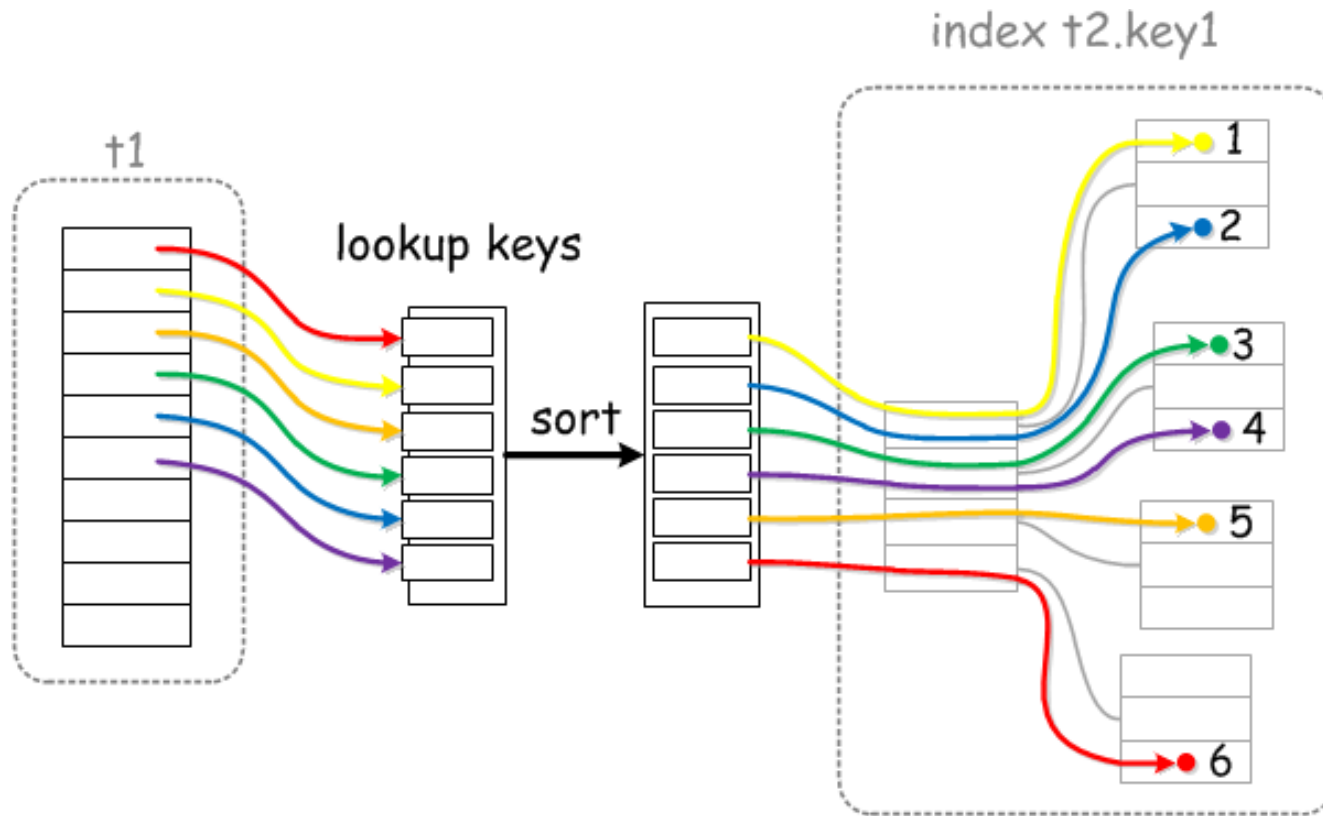
- ❑ Spider split per server
- ❑ Internal split per time line range on TokuDB

200 Billions records in 2x1T

```
CREATE TABLE `wt_ptest_results` (  
  `splitlot_id` int(10) unsigned NOT NULL DEFAULT '0',  
  `ptest_info_id` smallint(5) unsigned NOT NULL DEFAULT '0',  
  `run_id` mediumint(7) unsigned NOT NULL DEFAULT '0',  
  `flags` binary(1) NOT NULL DEFAULT '\0',  
  `value` float DEFAULT NULL,  
  PRIMARY KEY (`splitlot_id`, `ptest_info_id`, `run_id`)  
) ENGINE=SPIDER DEFAULT CHARSET=latin1  
COMMENT='wrapper "mysql", table "wt_ptest_results"  
PARTITION BY KEY (splitlot_id)  
(PARTITION PARTSRV1 COMMENT = 'srv "SERVER1"' ENGINE = SPIDER,  
PARTITION PARTSRV2 COMMENT = 'srv "SERVER2"' ENGINE = SPIDER);
```

```
CREATE TABLE `wt_ptest_results` (  
  `splitlot_id` int(10) unsigned NOT NULL DEFAULT '0',  
  `ptest_info_id` smallint(5) unsigned NOT NULL DEFAULT '0',  
  `run_id` mediumint(7) unsigned NOT NULL DEFAULT '0',  
  `flags` binary(1) NOT NULL DEFAULT '\0',  
  `value` float DEFAULT NULL,  
  PRIMARY KEY (`splitlot_id`, `ptest_info_id`, `run_id`)  
) ENGINE=TokuDB ROW_FORMAT=tokudb_lzma DEFAULT  
CHARSET=latin1  
PARTITION BY RANGE (SPLITLOT_ID)  
(PARTITION FIRSTPART VALUES LESS THAN (1400100000),  
PARTITION D1400100000 VALUES LESS THAN (1400199999),  
PARTITION D1400200000 VALUES LESS THAN (1400299999),  
PARTITION D1400300000 VALUES LESS THAN (1400399999),  
PARTITION D1400400000 VALUES LESS THAN (1400499999),  
...)
```

BKA join

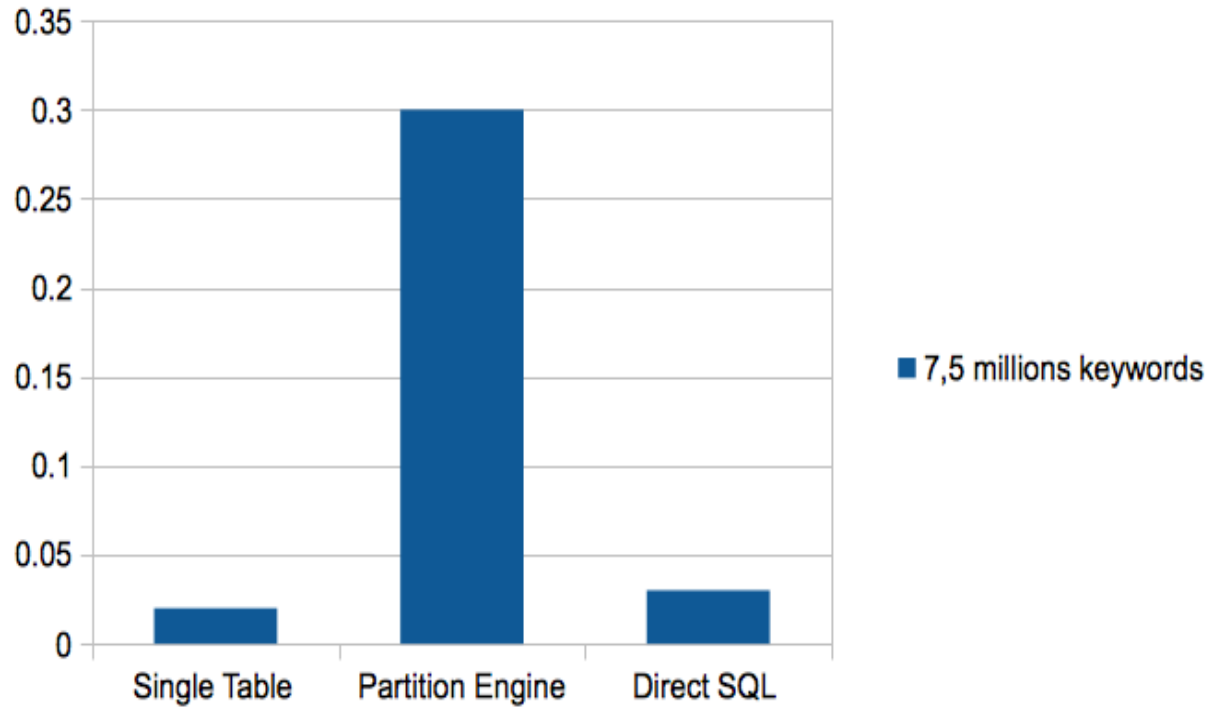


With *Batched Nested Loops Join* and *Key-Ordered* retrieval, index lookups are done in one "sweep"

DIRECT SQL



Latency for 1000 Key Point Access



Spider - MAP REDUCE Direct SQL



```
CREATE TEMPORARY TABLE `res` (  
  `keyword_crc64` bigint(20) unsigned NOT NULL,  
  `date` date NOT NULL DEFAULT '0000-00-00',  
  `idUrl` int(10) unsigned NOT NULL,  
  `keyword` varchar(128) NOT NULL DEFAULT '',  
  `idDomaine` tinyint(3) unsigned NOT NULL DEFAULT '0',  
  `nb` mediumint(8) unsigned NOT NULL DEFAULT '0',  
  `id` bigint(20) unsigned NOT NULL DEFAULT '0'  
) ENGINE=MEMORY DEFAULT CHARSET=latin1;
```

```
SELECT spider_bg_direct_sql('SELECT * FROM stats_url_kw_cur s WHERE s.id IN  
(241448386253908686)', 'res', concat('host "', host, '"', port '"', port,  
"', user '"', username, '"', password '"', password, '"', database '"',  
tgt_db_name, '"')) a FROM  
mysql.spider_tables where  
db_name = 'commentcamarche' and table_name like 'stats_url_kw_cur#P#pt%';
```

- ❑ cch parameter index for multi channel , // searches
- ❑ Tranparent for SUM , COUNT , MAX , MIN using spider_casual_read>=1, spider_bgs_mode=>1

Debugging RC



BUILD server

```
cmake . -DBUILD_CONFIG=mysql_release -DCMAKE_BUILD_TYPE=Debug -DWITH_VALGRIND=ON
```

RUN debug

```
mysqld --debug=S:T:t:r:p:n:L:i:F:f:D:d,info,error,query,qcache,my,exit,general,where:O,  
/tmp/mysqld.trace
```

```
valgring mysqld ...
```

Full backend log available from spider node

```
SET GLOBAL general_log=ON
```

```
SET GLOBAL spider_general_log=on
```

ROADMAP



- ❑ Node recovery with multi-source
- ❑ DDL synchronization
- ❑ Auto sharding

RESOURCES



❑ Documentation

<https://mariadb.com/kb/en/spider/>

❑ Engine Condition Push Down

http://spiderformysql.com/downloads/spider-3.1/mariadb-10.0.7-partition_cond_push.tgz

❑ Stephane Varoqui <stephane@skysql.com>

❑ Colin Charles <colin@mariadb.org>

❑ Sergey Vojtovich <svoj@mariadb.org>

(integration into MariaDB) + Alexander Barkov
<bar@mariadb.org> (build, +groonga)

❑ Commerical Support: Spiral Arms - Kayoko Goto <kayoko.goto@spiral-arms.com>

❑ Kentoku Shiba's blog: <http://wild-growth.blogspot.com/>

Who uses Spider



- ❑ 104 Job Bank (largest job search site in Taiwan - Google PageRank 8)
- ❑ Kadokawa Corporation (publications, media group)
- ❑ MicroAd (one of the largest ad networks in Japan)
- ❑ Sansan (card management app for teams)
- ❑ teamLab
- ❑ CCM Benchmark (French online media group with properties in 6 languages)

- ❑ Few editors



Q&A
Thanks