Alibaba Cloud

Analytical Queries in MySQL

Øystein Grøvlen

Senior Staff Engineer
Alibaba Cloud

Alibaba Cloud

Analytical Queries in MySQL

Disk-Bound Queries

How to Tune Your Complex Queries

Hash Join

MySQL Wishlist

Alibaba Cloud

OLTP vs OLAP

On-Line Transactional Processing OLTP	On-Line Analytical Processing OLAP	Hybrid transactional/analytical proc. HTAP
Short transactions Look-up by key Single-table queries	Complex queries Table/index scans Multi-table joins	Mix
Performance Metric: Throughput	Performance Metric: Response Time	Both
Row Store	Column Store	MariaDB Column Store MySQL Analytics Service TiDB

Alibaba Cloud

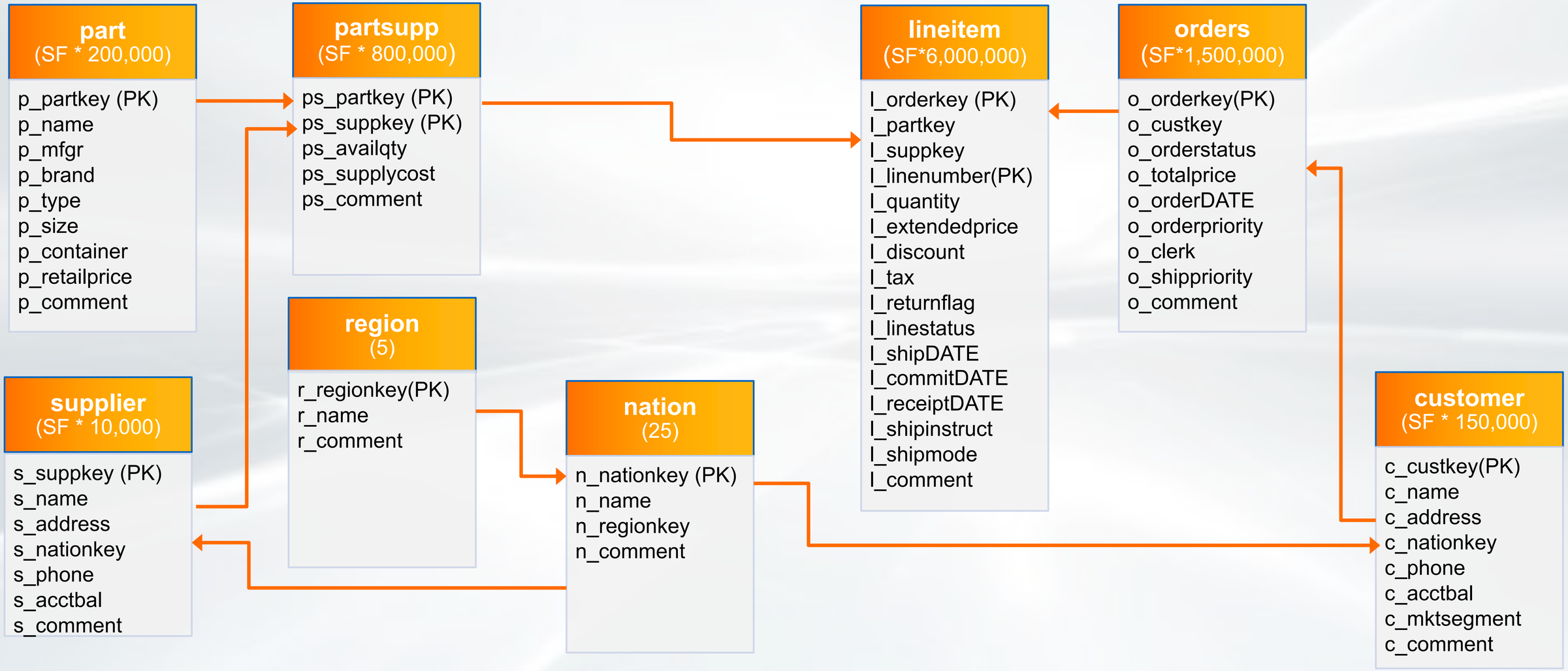
TPC-H vs DBT-3

- TPC-H
  - Decision-support benchmark specification
  - 22 complex queries
  - Uniform data distribution
  - https://www.tpc.org/tpch
- TPC-DS
  - “Snow-storm” schema
  - 99 queries
  - Non-uniform data distribution
  - Models data maintenance
  - https://www.tpc.org/tpcds

- DBT-3
  - Open-source implementation of TPC-H
  - Load test, power test, throughput test
  - Implementations for MySQL, PostgreSQL, SAPdb
  - Does not always follow TPC-H specification
    - Indexes are only allowed on foreign keys and date columns
    - MySQL: covering index, rewrites query
    - PostgreSQL: Adds lots specific indexes, REAL instead of NUMERIC
  - https://sourceforge.net/projects/osdldbt/files/dbt3/

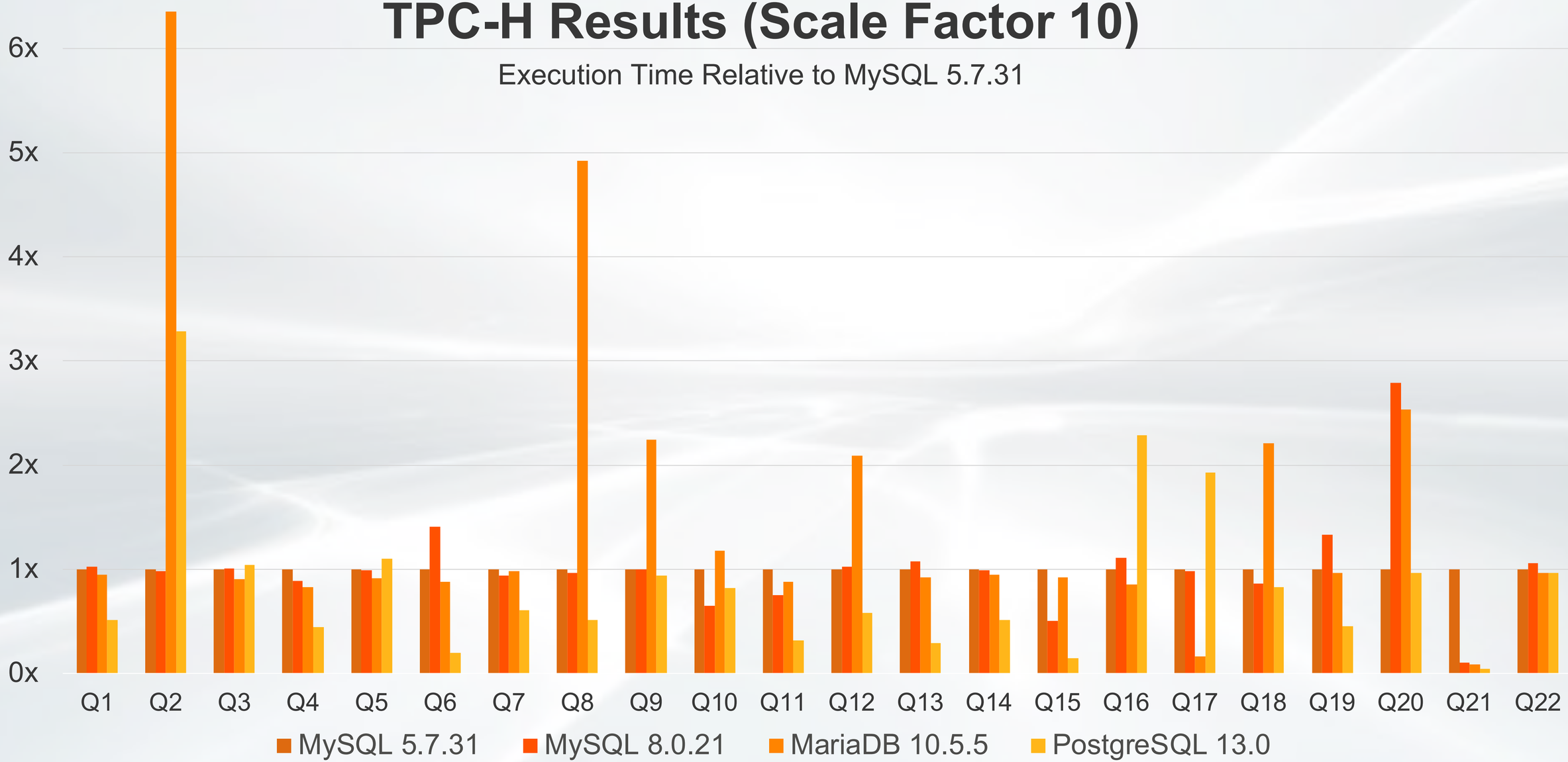
Alibaba Cloud

TPC-H Schema



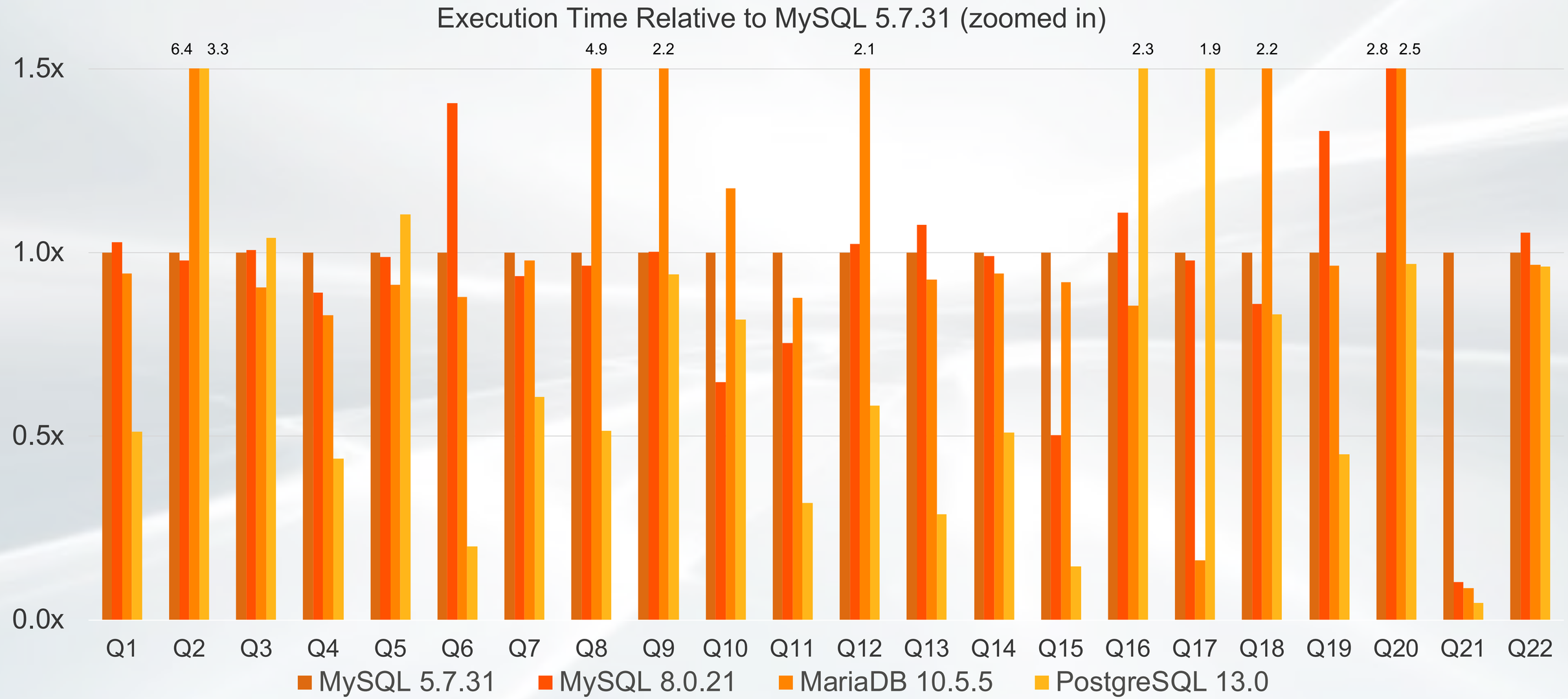
Alibaba Cloud

TPC-H Results (Scale Factor 10)



Alibaba Cloud

TPC-H Results (Scale Factor 10)



Alibaba Cloud

Configuration Details

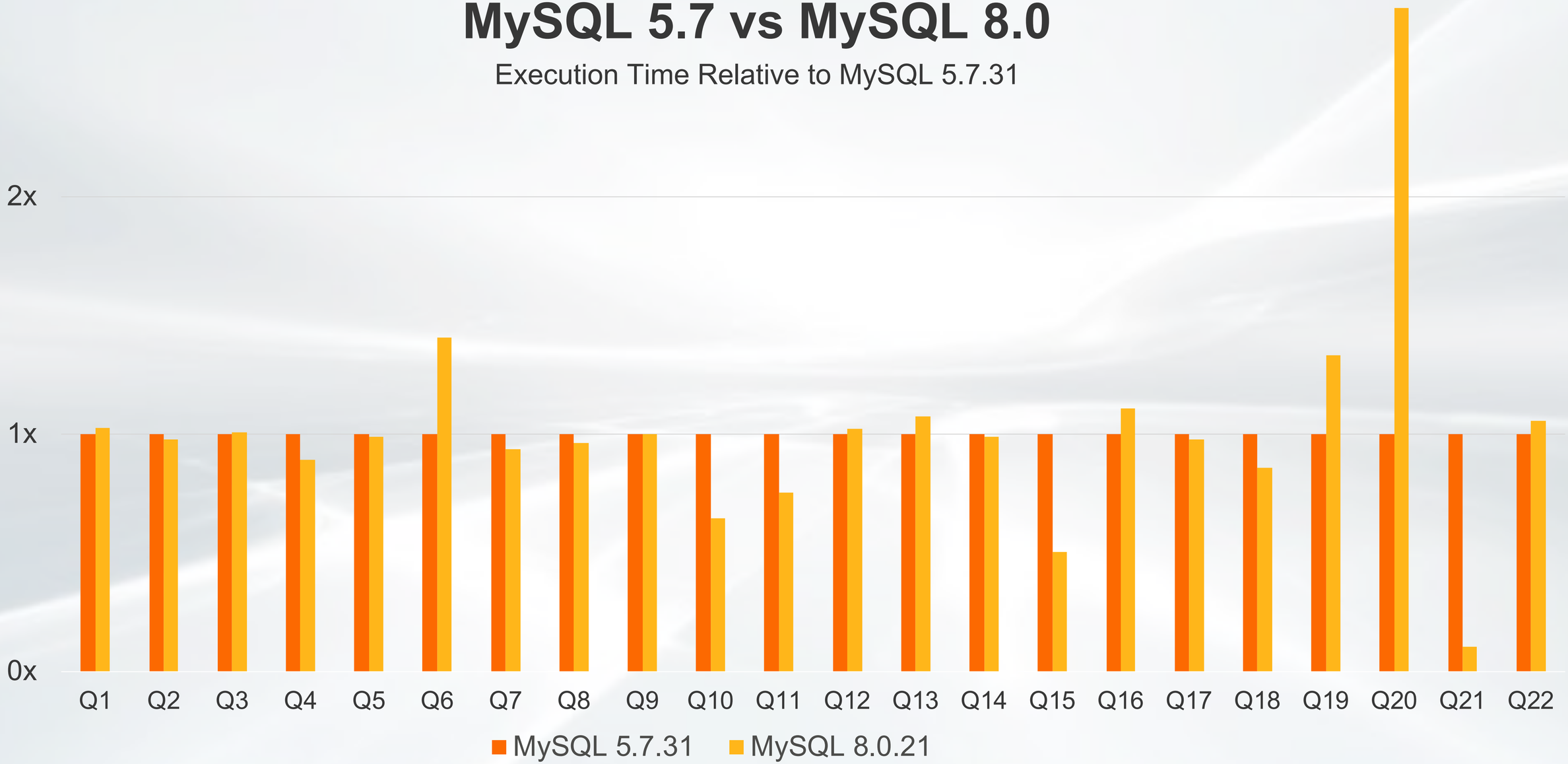
- CPU:
  - 2 Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz (Broaldwell), 16 physical cores each
- Buffer pool: 32 GB (All data in memory)
  - MySQL/MariaDB: `innodb_buffer_pool_size`
  - PostgreSQL: `shared_buffers`, `effective_cache_size`
- Working memory: 64 MB
  - MySQL/MariaDB: `join_buffer_size`, `read_rnd_buffer_size`
  - PostgreSQL: `work_mem`
- Other
  - MySQL/MariaDB: `innodb_adaptive_hash_index = off`
  - PostgreSQL: `max_parallel_workers_per_gather = 0`

For details, check
https://github.com/ogrovlen/tpchcomp

Alibaba Cloud

MySQL 5.7 vs MySQL 8.0

Execution Time Relative to MySQL 5.7.31



Alibaba Cloud

MySQL 8.0 vs MySQL 5.7

- Faster Queries:
  - Cost-model for in-memory data access: Q21 (–89%)
  - Semi-join for EXISTS: Q4 (–11%), Q21 (–9%)
  - Rewrite to CTE: Q15 (–50%)
  - New TempTable engine: Q10 (–35%), Q11 (–25%)
  - Unknown: Q18 (–14%)
- Slower queries:
  - Changed query plan: Q6 (+41%), Q20 (+180%)
  - Default character set (Latin1 vs UTF8): Q19 (+33%)
  - New plan (anti-join): Q16 (+11%) (The old plan is even slower)

Alibaba Cloud

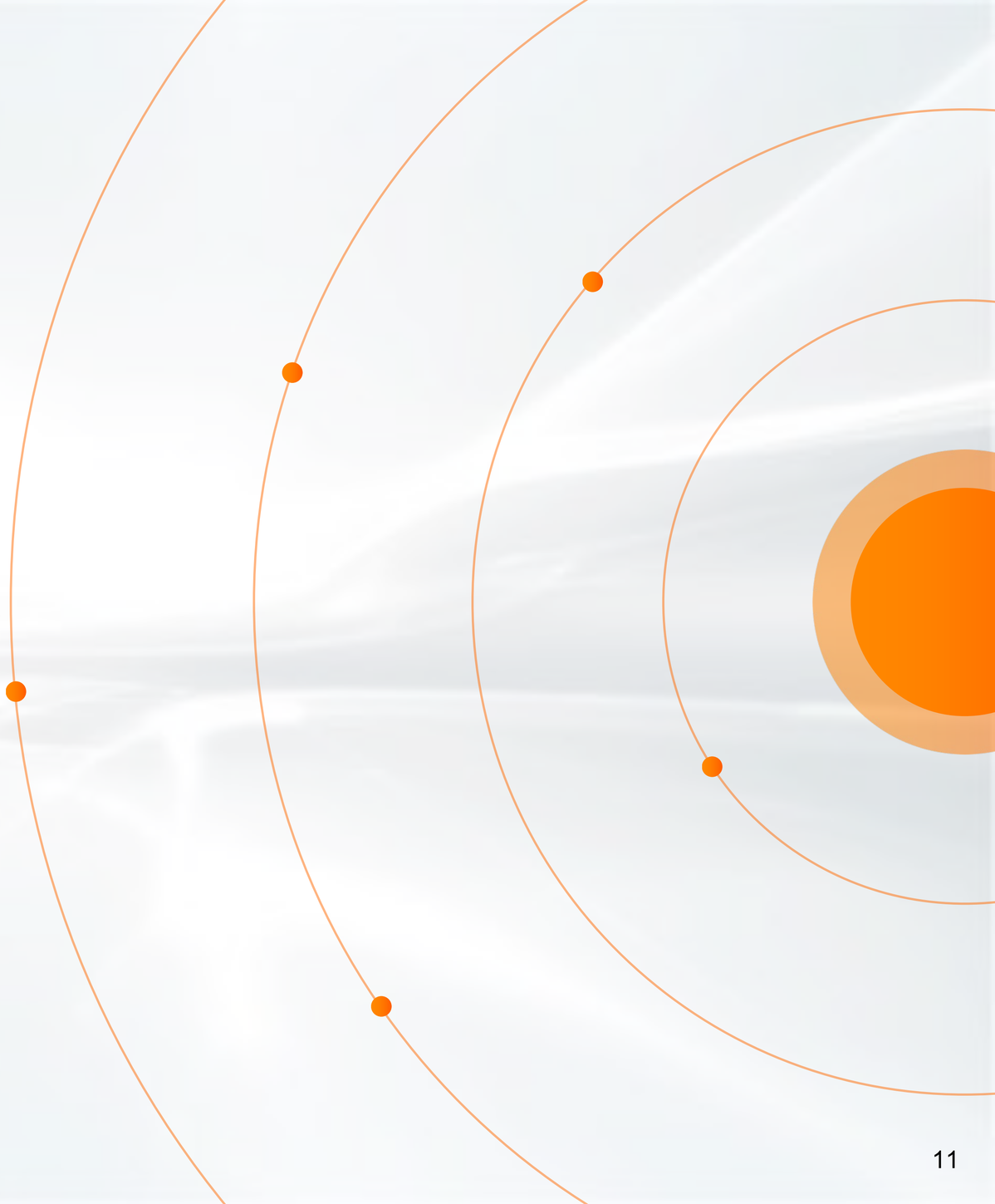
Analytical Queries in MySQL

Disk-Bound Queries

How to Tune Your Complex Queries

Hash Join

MySQL Wishlist



Alibaba Cloud

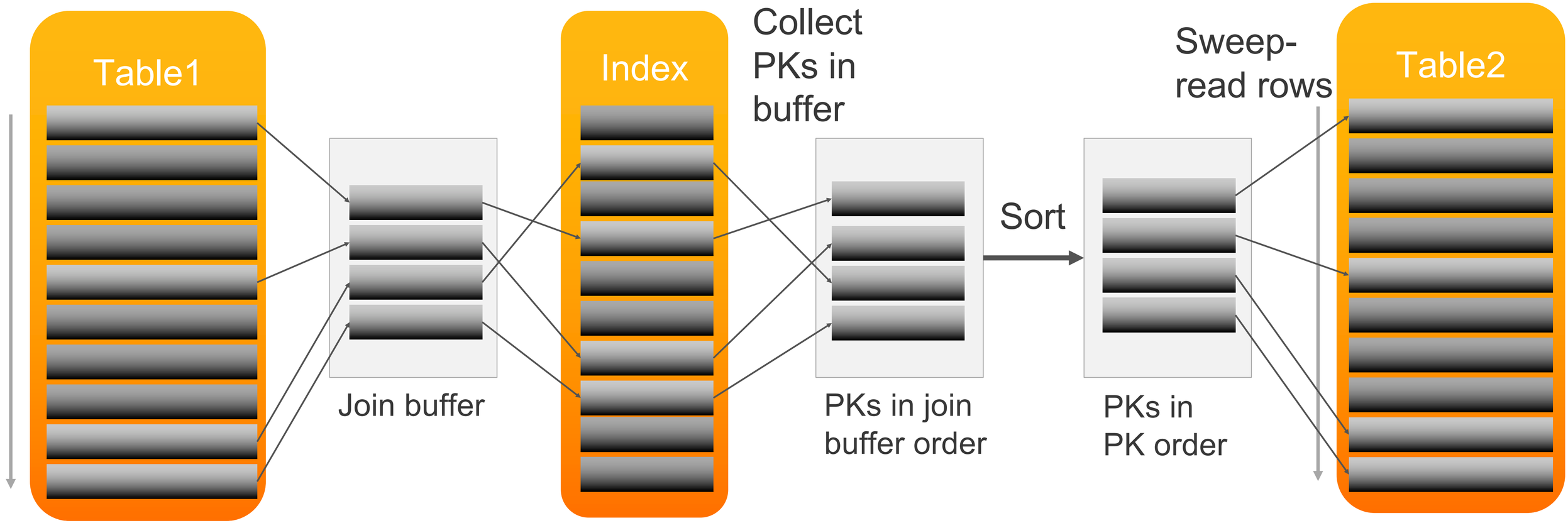
Disk-Bound Queries

- Disk:
  - Intel SSD DC P4510 Series, PCIe 3.1 x4, NVMe Interface
- Configuration settings:
  - `innodb_buffer_pool_size = 1G`
  - `innodb_flush_method = O_DIRECT`
  - `innodb_old_blocks_time = 0`
  - `read_rnd_buffer_size = 64M`
  - `join_buffer_size = 64M`
  - `optimizer_switch = 'batched_key_access=on,mrr_cost_based=off'`

Alibaba Cloud

Batched Key Access (BKA)

MRR Applied to Join Buffering



Alibaba Cloud

BKA, Example

TPC-H Q13 (Customer Distribution Query)

```
SELECT c_count, COUNT(*) AS custdist
FROM (
     SELECT c_custkey, COUNT(o_orderkey) AS c_count
     FROM customer LEFT OUTER JOIN orders
       ON c_custkey = o_custkey AND o_comment NOT LIKE '%express%requests%'
     GROUP BY c_custkey
   ) AS c_orders
GROUP BY c_count
ORDER BY custdist DESC, c_count DESC;
```

No BKA

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	22500404	100.00	Using temporary; Using filesort
2	DERIVED	customer	NULL	index	PRIMARY,i_c_nationkey	PRIMARY	4	NULL	1500000	100.00	Using index
2	DERIVED	orders	NULL	ref	i_o_custkey	i_o_custkey	5	dbt3_sf10.customer.c_custkey	15	100.00	Using where

BKA

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	22500404	100.00	Using temporary; Using filesort
2	DERIVED	customer	NULL	index	PRIMARY,i_c_nationkey	i_c_nationkey	5	NULL	1500000	100.00	Using index; Using temporary
2	DERIVED	orders	NULL	ref	i_o_custkey	i_o_custkey	5	dbt3_sf10.customer.c_custkey	15	100.00	Using where; Using join buffer (Batched Key Access)

Alibaba Cloud

BKA – Reduced Disk I/O

TPC-H Q13 (Customer Distribution Query)

```
SELECT event_name, count_read, FORMAT_PICO_TIME(avg_timer_read) “Avg Read Time”,
                               FORMAT_BYTES(sum_number_of_bytes_read)”Bytes Read”
  FROM performance_schema.file_summary_by_event_name
  WHERE event_name='wait/io/file/innodb/innodb_data_file';
```

No BKA

event_name	count_read	Avg Read Time	Bytes Read
wait/io/file/innodb/innodb_data_file	7070389	117.03 us	107.89 GiB

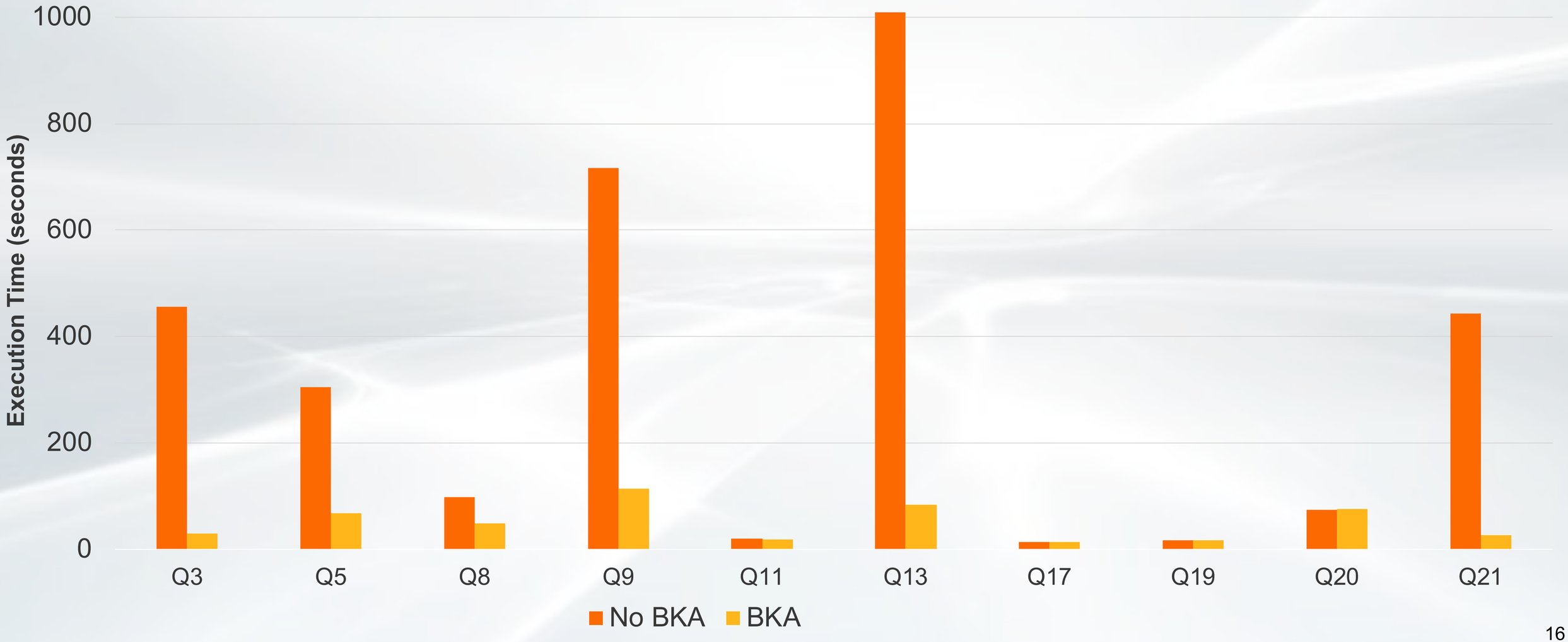
BKA

event_name	count_read	Avg Read Time	Bytes Read
wait/io/file/innodb/innodb_data_file	664932	19.08 us	10.15 GiB

Alibaba Cloud

BKA Improvement

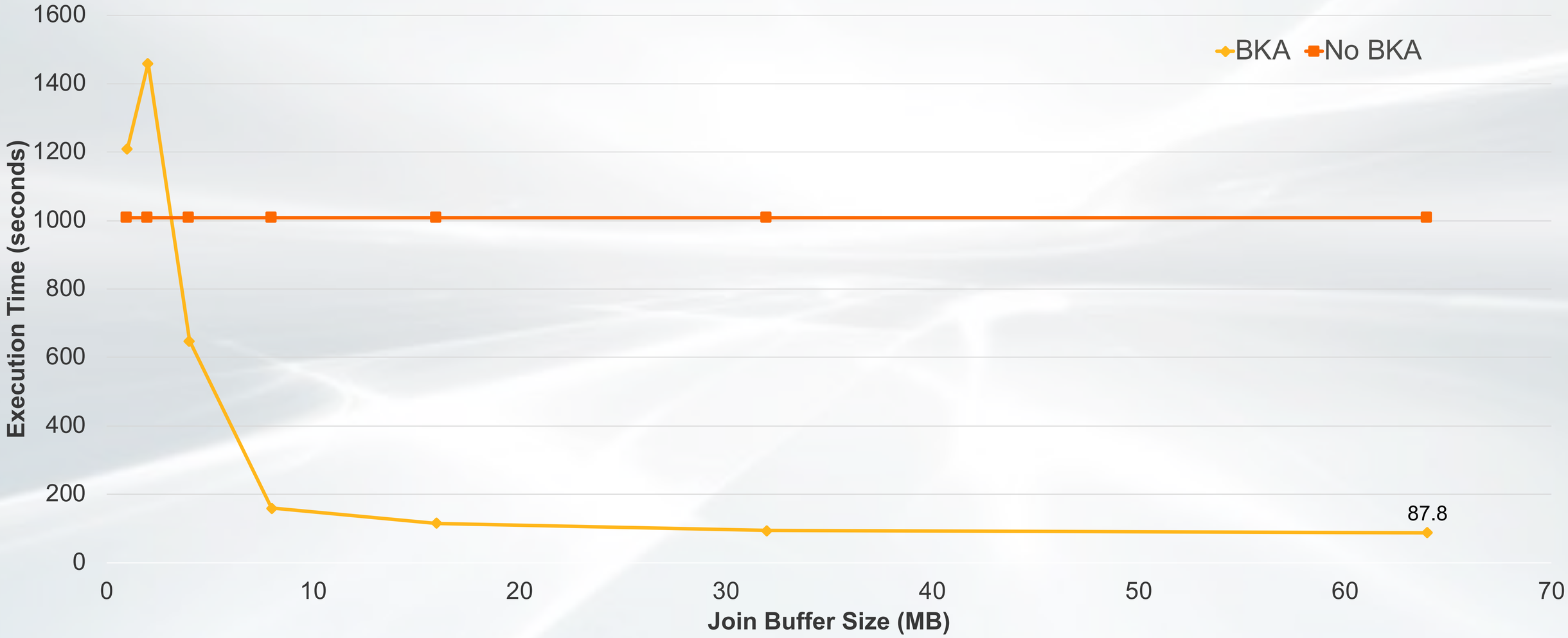
TPC-H queries that will use BKA



Alibaba Cloud

Effect of Join Buffer Size

TPC-H Q13 (Customer Distribution Query)



Alibaba Cloud

Analytical Queries in MySQL

Disk-Bound Queries

How to Tune Your Complex Queries

Hash Join

MySQL Wishlist

Alibaba Cloud

How to Tune Your Complex Queries

- Add optimizer hints
  - `INDEX()`, `JOIN_ORDER()`, etc
- Adjust configuration settings
  - (e.g., `innodb_buffer_pool_size`, `join_buffer_size`)
- Improve quality of statistics:
  - Add histograms
  - Increase sampling (`innodb_stats_persistent_sample_pages`, `histogram_generation_max_mem_size`)
- Rewrite queries
- Add more indexes
- Force hash join by “hiding” index

Alibaba Cloud

Optimizer Hints – Ignore Index

TPC-H Q6 (Forecasting Revenue Change Query)

```
SELECT SUM(l_extendedprice * l_discount) AS revenue
FROM lineitem
WHERE l_shipdate >= '1995-01-01' AND l_shipdate < DATE_ADD('1995-01-01',INTERVAL '1' YEAR)
  AND l_discount BETWEEN 0.08 - 0.01 AND 0.08 + 0.01 AND l_quantity < 24;
```

MySQL 5.7

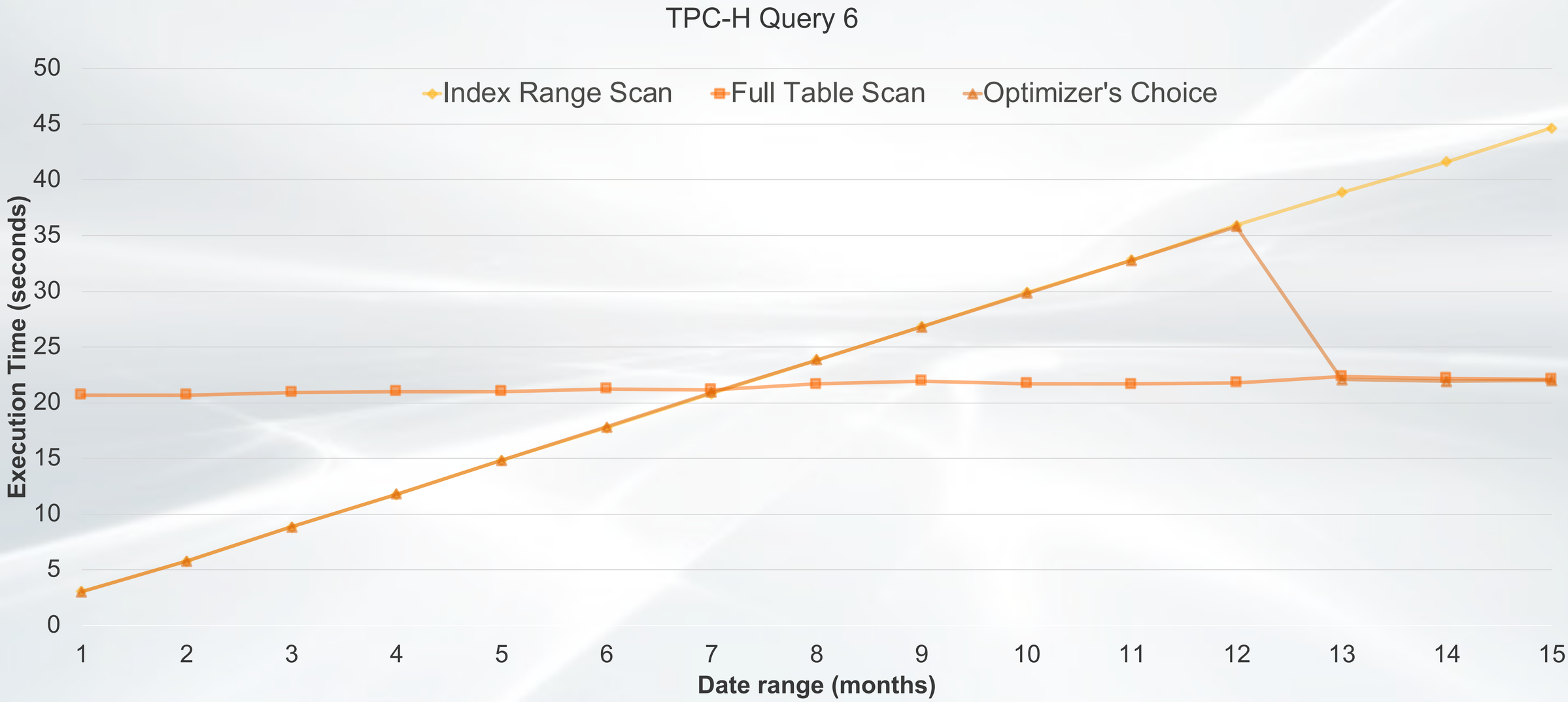
```
+----+-------------+----------+------------+------+---------------+------+---------+------+----------+----------+-------------+
| id | select_type | table    | partitions | type | possible_keys | key  | key_len | ref  | rows     | filtered | Extra       |
+----+-------------+----------+------------+------+---------------+------+---------+------+----------+----------+-------------+
|  1 | SIMPLE      | lineitem | NULL       | ALL  | i_l_shipdate  | NULL | NULL    | NULL | 59986052 |     1.06 | Using where |
+----+-------------+----------+------------+------+---------------+------+---------+------+----------+----------+-------------+
```

MySQL 8.0

```
+----+-------------+----------+------------+-------+---------------+--------------+---------+------+----------+----------+-----------------------------------+
| id | select_type | table    | partitions | type  | possible_keys | key          | key_len | ref  | rows     | filtered | Extra                             |
+----+-------------+----------+------------+-------+---------------+--------------+---------+------+----------+----------+-----------------------------------+
|  1 | SIMPLE      | lineitem | NULL       | range | i_l_shipdate  | i_l_shipdate | 4       | NULL | 17117650 |    12.51 | Using index condition; Using where |
+----+-------------+----------+------------+-------+---------------+--------------+---------+------+----------+----------+-----------------------------------+
```

Alibaba Cloud

Full Table Scan vs Index Range Scan

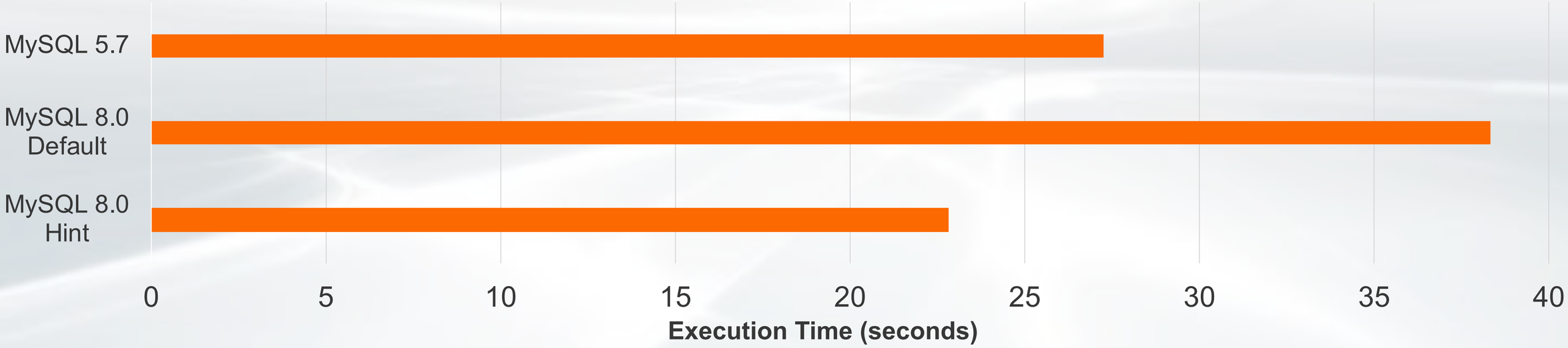


Alibaba Cloud

Optimizer Hints – Ignore Index

TPC-H Q6 (Forecasting Revenue Change Query)

```
SELECT /*+ NO_INDEX(lineitem i_l_shipdate) */ SUM(l_extendedprice * l_discount) AS revenue
FROM lineitem
WHERE l_shipdate >= '1995-01-01' AND l_shipdate < DATE_ADD('1995-01-01',INTERVAL '1' YEAR)
  AND l_discount BETWEEN 0.08 - 0.01 AND 0.08 + 0.01 AND l_quantity < 24;
```



Alibaba Cloud

Optimizer Hints – Change Join Order

TPC-H Q20 (Potential Part Promotion Query)

```
SELECT s_name, s_address
FROM supplier, nation
WHERE s_suppkey IN (
      SELECT ps_suppkey
      FROM partsupp
      WHERE ps_partkey IN ( SELECT p_partkey FROM part WHERE p_name LIKE 'dodger%' )
        AND ps_availqty > (
            SELECT 0.5 * SUM(l_quantity)
            FROM lineitem
            WHERE l_partkey = ps_partkey AND l_suppkey = ps_suppkey
              AND l_shipdate >= '1994-01-01'
              AND l_shipdate < DATE_ADD( '1994-01-01', INTERVAL '1' YEAR)
        )
  )
  AND s_nationkey = n_nationkey AND n_name = 'INDIA'
ORDER BY s_name;
```

Not supported by histograms

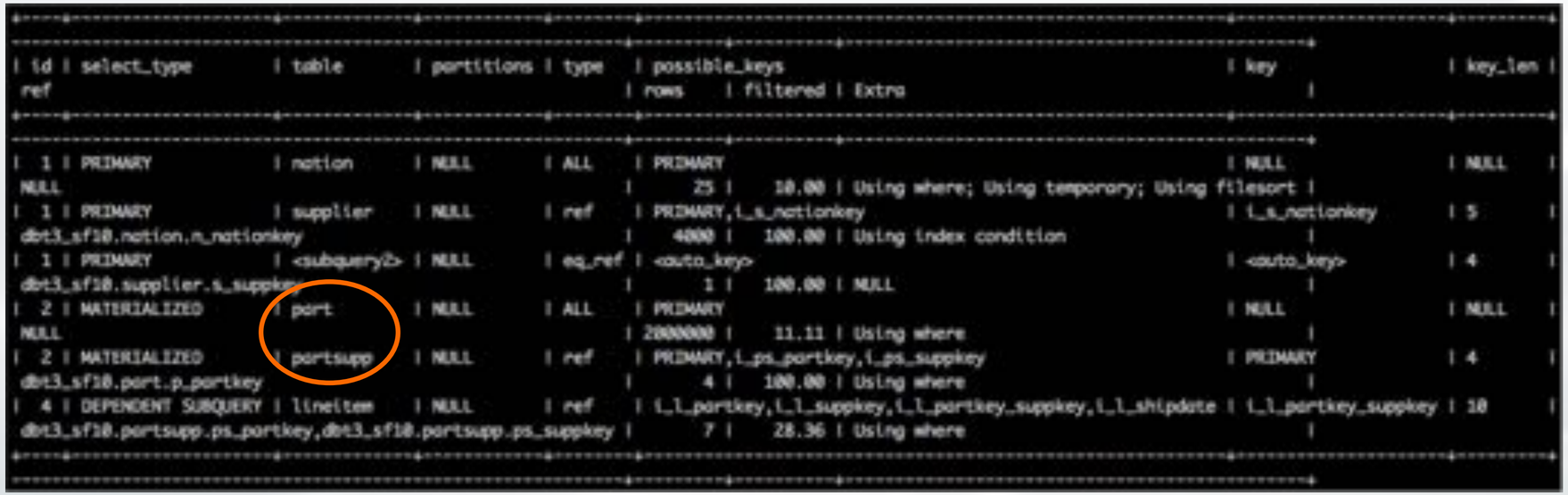
Cost of subqueries not considered

Alibaba Cloud

Optimizer Hints – Change Join Order

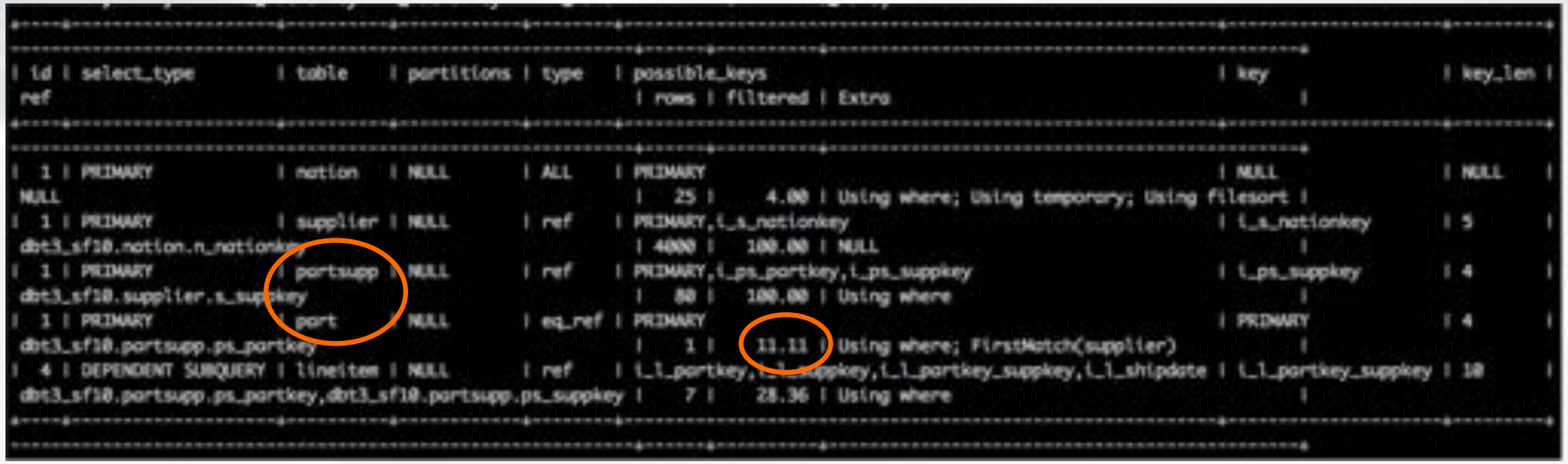
TPC-H Q20 (Potential Part Promotion Query)

MySQL 5.7



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	nation	NULL	ALL	PRIMARY	NULL	NULL	NULL	25	10.00	Using where; Using temporary; Using filesort
1	PRIMARY	supplier	NULL	ref	PRIMARY,i_s_nationkey	i_s_nationkey	5	dbt3_sf10.nation.n_nationkey	4000	100.00	Using index condition
1	PRIMARY	<subquery2>	NULL	eq_ref	<auto_key>	<auto_key>	4	dbt3_sf10.supplier.s_suppkey	1	100.00	NULL
2	MATERIALIZED	part	NULL	ALL	PRIMARY	NULL	NULL	NULL	2000000	11.11	Using where
2	MATERIALIZED	partsupp	NULL	ref	PRIMARY,i_ps_partkey,i_ps_suppkey	PRIMARY	4	dbt3_sf10.part.p_partkey	4	100.00	Using where
4	DEPENDENT SUBQUERY	lineitem	NULL	ref	i_l_partkey,i_l_suppkey,i_l_partkey_suppkey,i_l_shipdate	i_l_partkey_suppkey	10	dbt3_sf10.partsupp.ps_partkey,dbt3_sf10.partsupp.ps_suppkey	7	28.36	Using where

MySQL 8.0



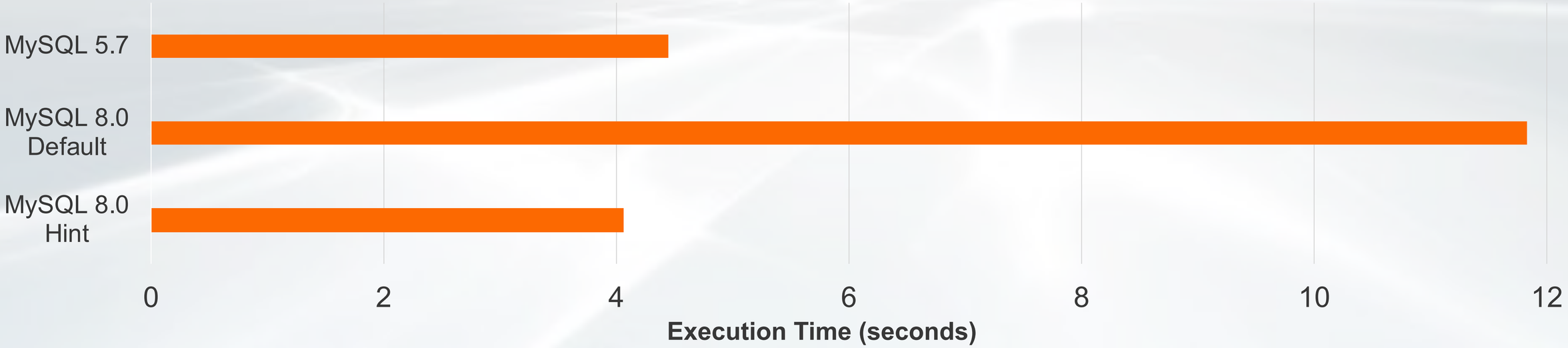
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	nation	NULL	ALL	PRIMARY	NULL	NULL	NULL	25	4.00	Using where; Using temporary; Using filesort
1	PRIMARY	supplier	NULL	ref	PRIMARY,i_s_nationkey	i_s_nationkey	5	dbt3_sf10.nation.n_nationkey	4000	100.00	NULL
1	PRIMARY	partsupp	NULL	ref	PRIMARY,i_ps_partkey,i_ps_suppkey	i_ps_suppkey	4	dbt3_sf10.supplier.s_suppkey	80	100.00	Using where
1	PRIMARY	part	NULL	eq_ref	PRIMARY	PRIMARY	4	dbt3_sf10.partsupp.ps_partkey	1	11.11	Using where; FirstMatch(supplier)
4	DEPENDENT SUBQUERY	lineitem	NULL	ref	i_l_partkey,i_l_suppkey,i_l_partkey_suppkey,i_l_shipdate	i_l_partkey_suppkey	10	dbt3_sf10.partsupp.ps_partkey,dbt3_sf10.partsupp.ps_suppkey	7	28.36	Using where

Alibaba Cloud

Optimizer Hints – Change Join Order

TPC-H Q20 (Potential Part Promotion Query)

```
SELECT /*+ JOIN_ORDER(part, partsupp) */ s_name, s_address
FROM supplier, nation
WHERE s_suppkey IN (
      SELECT ps_suppkey
      FROM partsupp
      WHERE ps_partkey IN ( SELECT p_partkey FROM part WHERE p_name LIKE 'dodger%' )
        AND …
```



Alibaba Cloud

MySQL Statistics – Improve Accuracy

- MySQL:
  - Index statistics (number of distinct values):
    - `innodb_stats_persistent_sample_pages` should be increased (default 20)
    - In these tests, it was set higher than max. number of leaf pages of any table (1,000,000)
    - Consider setting `innodb_stats_auto_recalc = OFF`, and run `ANALYZE TABLE` manually
    - Will usually change slowly, so do not need to be update very often
    - Always run `ANALYZE TABLE` manually after large batch insert/delete/updates!
  - Histograms (8.0):
    - Created on all columns used in conditions
    - Default settings used for `histogram_generation_max_mem_size` and number of buckets
    - Some histograms may have to be frequently updated (e.g., date/time columns)

Alibaba Cloud

Statistics

- MariaDB:
  - Index statistics (number of distinct values):
    - Similar to MySQL
  - Histograms:
    - Created automatically by `ANALYZE TABLE` (since 10.4)
    - Highest accuracy by default (since 10.4)
- PostgreSQL:
  - Use default settings (single column NDV, MCV, histograms, physical correlation)
  - Tried increasing `default_statistics_target`: Lots of changed query plans, but no general improvement
  - More advanced statistics available (Multivariate statistics)

Alibaba Cloud

Rewriting Queries

- MySQL does some automatic rewrites/transformation
  - Merge derived tables (MySQL 5.7)
  - EXISTS-subqueries to IN-subqueries (MySQL 8.0.16)
  - Scalar subquery to join with derived table (MySQL 8.0.21)
  - Push condition down to derived table/CTE (MySQL 8.0.22)
- Manual rewrites
  - Use CTE (TPC-H Q15 https://mysqlserverteam.com/mysql-8-0-improved-performance-with-cte/)
  - Subqueries to derived tables (TPC-H Q18 http://oysteing.blogspot.com/2016/12/improved-query-performance-by-using.html)
  - Use window functions instead of subqueries for aggregation

Alibaba Cloud

Use Window Functions Instead of Subqueries for Aggregation

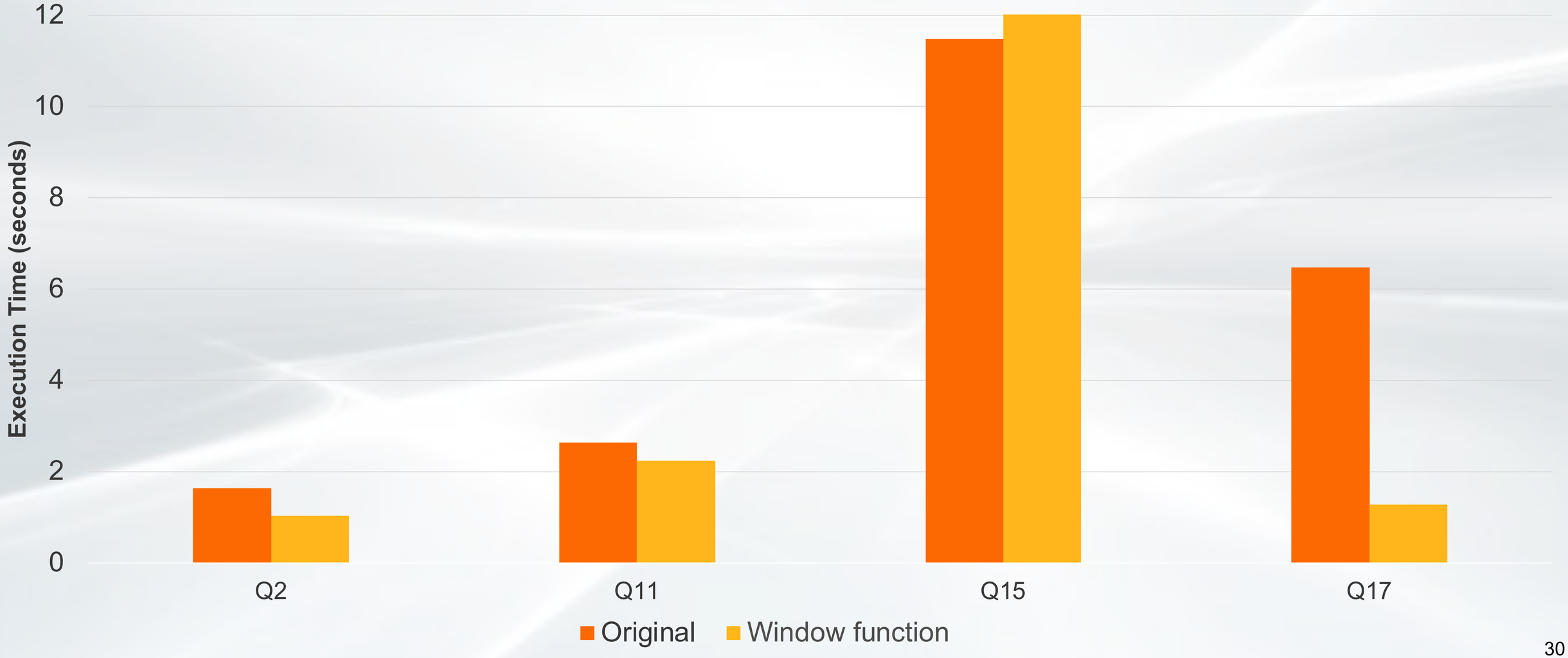
TPC-H Q17 Small-Quantity-Order Revenue Query

```
SELECT SUM(l_extendedprice) / 7.0 AS avg_yearly
FROM lineitem, part
WHERE p_partkey = l_partkey AND p_brand = 'Brand#11' AND p_container = 'SM CAN'
  AND l_quantity
          < (SELECT 0.2 * AVG(l_quantity) FROM lineitem WHERE l_partkey = p_partkey);
```

```
WITH win AS (
    SELECT l_extendedprice, l_quantity,
           AVG(l_quantity) OVER (PARTITION BY p_partkey) avg_l_quantity
    FROM lineitem, part
    WHERE p_partkey = l_partkey AND p_brand = 'Brand#11' AND p_container = 'SM CAN'
)
SELECT SUM(l_extendedprice) / 7.0 AS avg_yearly
FROM win
WHERE l_quantity < 0.2 * avg_l_quantity;
```

Alibaba Cloud

Window Functions Instead of Subqueries for Aggregation



Alibaba Cloud

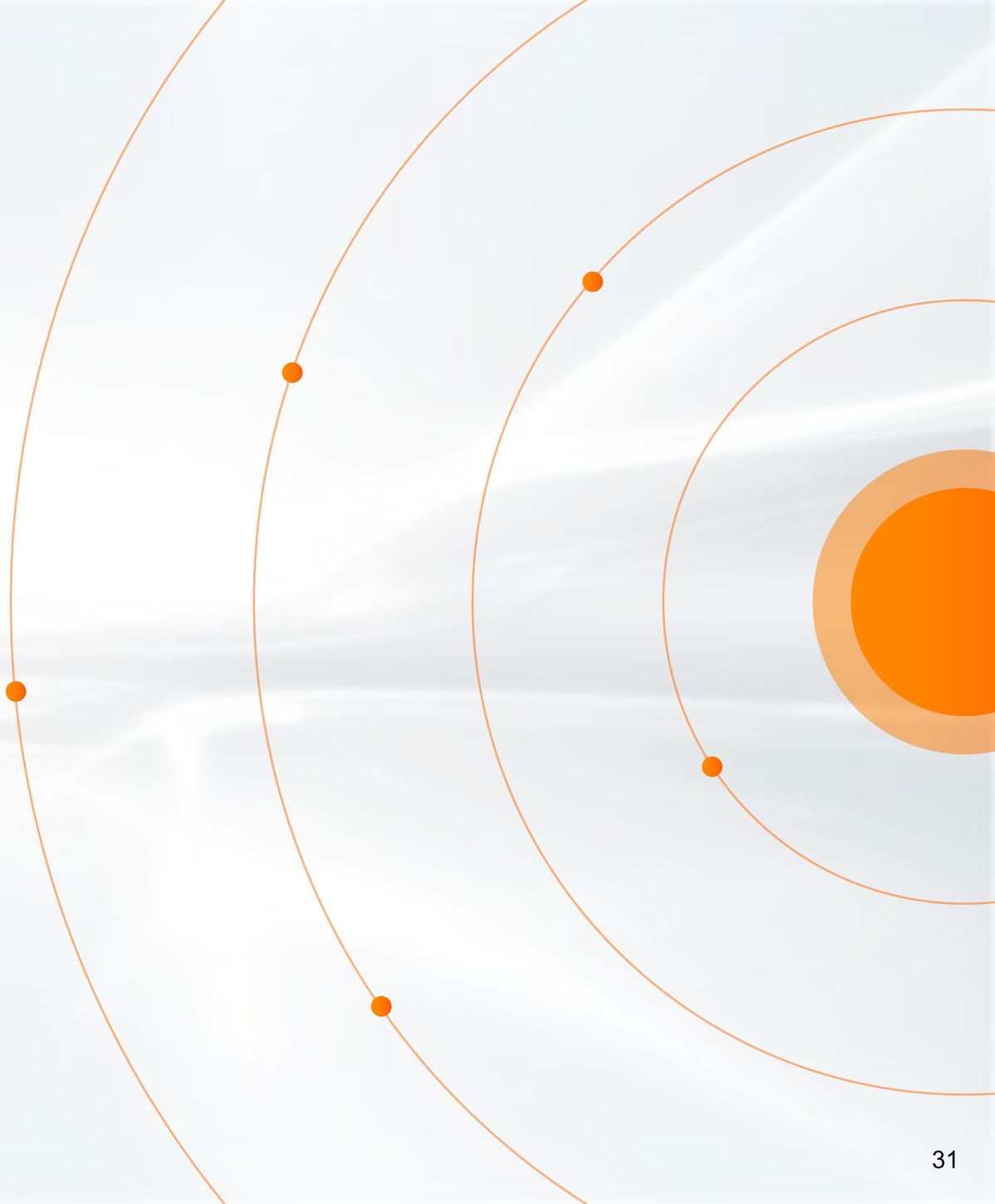
Analytical Queries in MySQL

Disk-Bound Queries

How to Tune Your Complex Queries

Hash Join

MySQL Wishlist



Alibaba Cloud

Force Hash Join by “Hiding” Index

- MySQL always prefer indexed nested loops join over hash join
- Can get hash join by telling optimizer to ignore index
- Histograms are important for hash join!
  - Range estimates from index are not used when index is ignored

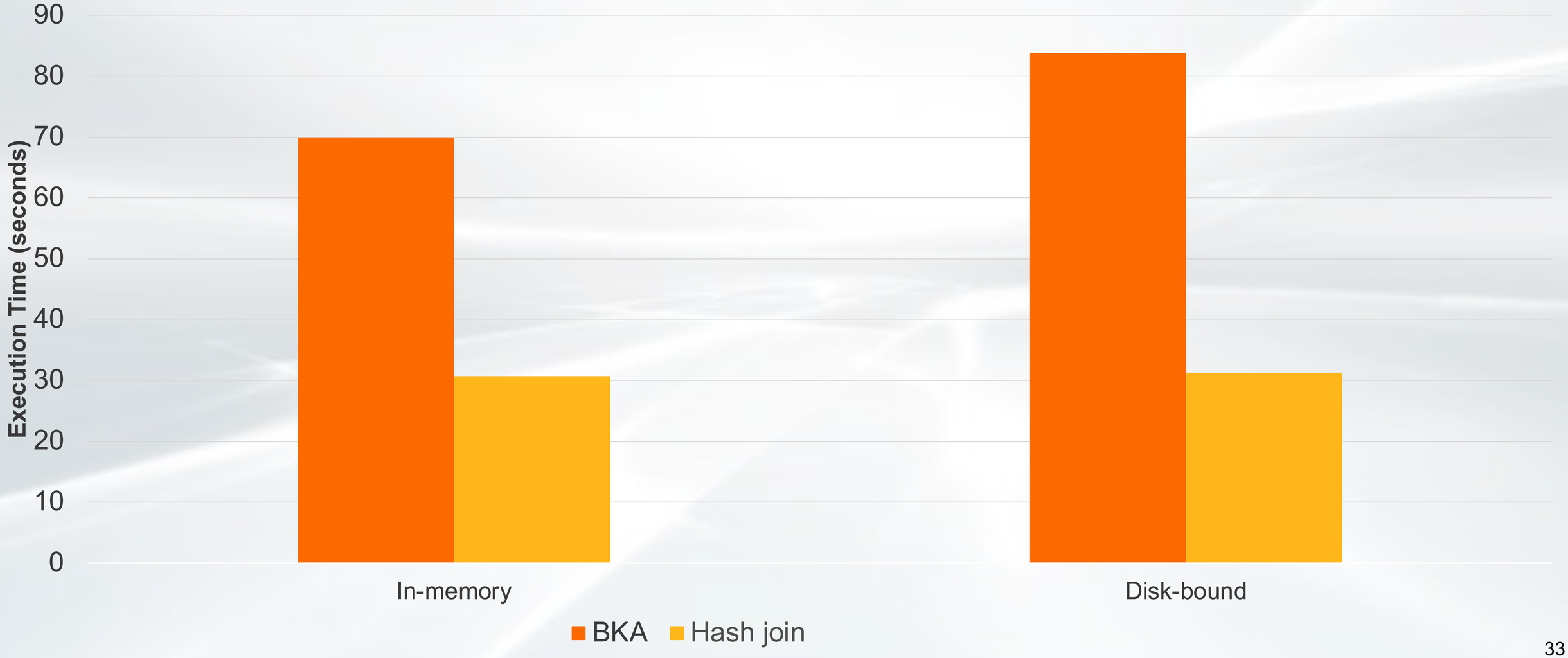
TPC-H Q13

```
SELECT c_count, COUNT(*) AS custdist
FROM (
    SELECT /*+ NO_INDEX(orders i_o_custkey) */ c_custkey, COUNT(o_orderkey) AS c_count
    FROM customer LEFT OUTER JOIN orders
      ON c_custkey = o_custkey AND o_comment NOT LIKE '%express%requests%'
    GROUP BY c_custkey
  ) AS c_orders
GROUP BY c_count
ORDER BY custdist DESC, c_count DESC;
```

Alibaba Cloud

Hash join vs BKA

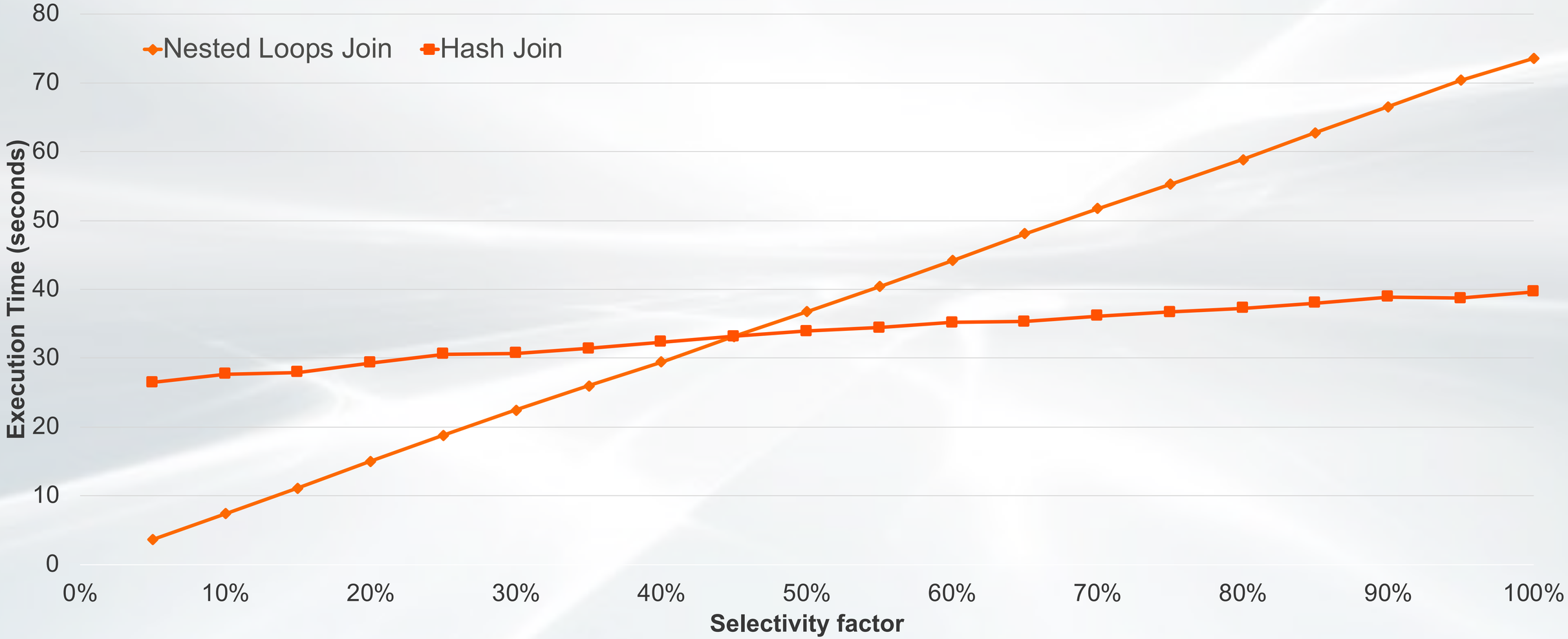
TPC-H Q13 (Customer Distribution Query)



Alibaba Cloud

When to Use Hash Join?

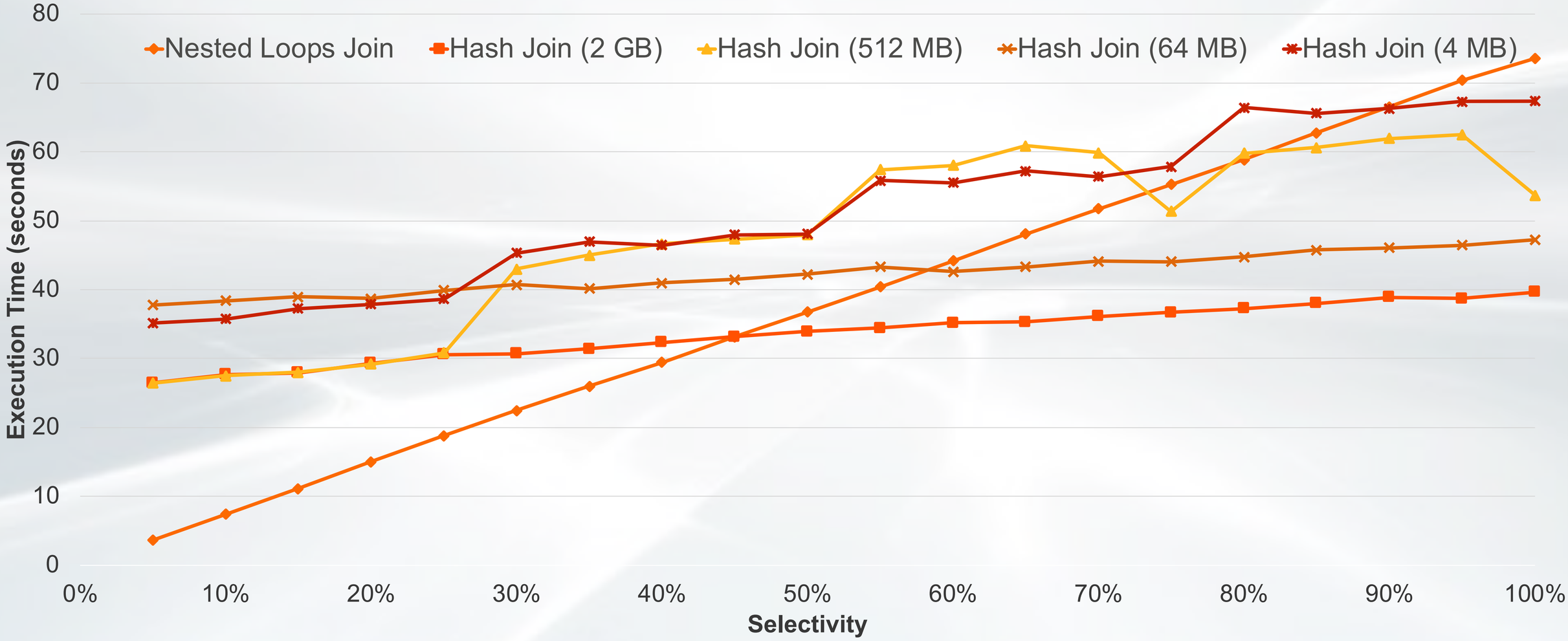
SELECT COUNT(*) FROM orders JOIN lineitem ON l_orderkey = o_orderkey WHERE o_custkey <= 1500000 * *selectivity*



Alibaba Cloud

Impact of Join Buffer Size

SELECT COUNT(*) FROM orders JOIN lineitem ON l_orderkey = o_orderkey WHERE o_custkey <= 1500000 * *selectivity*



Alibaba Cloud

Is Join Buffer Big Enough for In-Memory Hash Join?

Check `performance_schema.memory_summary_global_by_event_name`

```
TRUNCATE performance_schema.memory_summary_global_by_event_name;
< Run query >
SELECT event_name, count_alloc, format_bytes(sum_number_of_bytes_alloc) "Total usage",
     high_count_used, format_bytes(high_number_of_bytes_used) "Max. usage"
FROM performance_schema.memory_summary_global_by_event_name
WHERE event_name LIKE 'memory/sql/hash_join';
```

```
set join_buffer_size = 64*1024*1024;
```

event_name	count_alloc	Total usage	high_count_used	Max. usage
memory/sql/hash_join	595	1.51 GiB	19	69.25 MiB

```
set join_buffer_size = 2048*1024*1024;
```

event_name	count_alloc	Total usage	high_count_used	Max. usage
memory/sql/hash_join	27	1.73 GiB	27	1.73 GiB

Alibaba Cloud

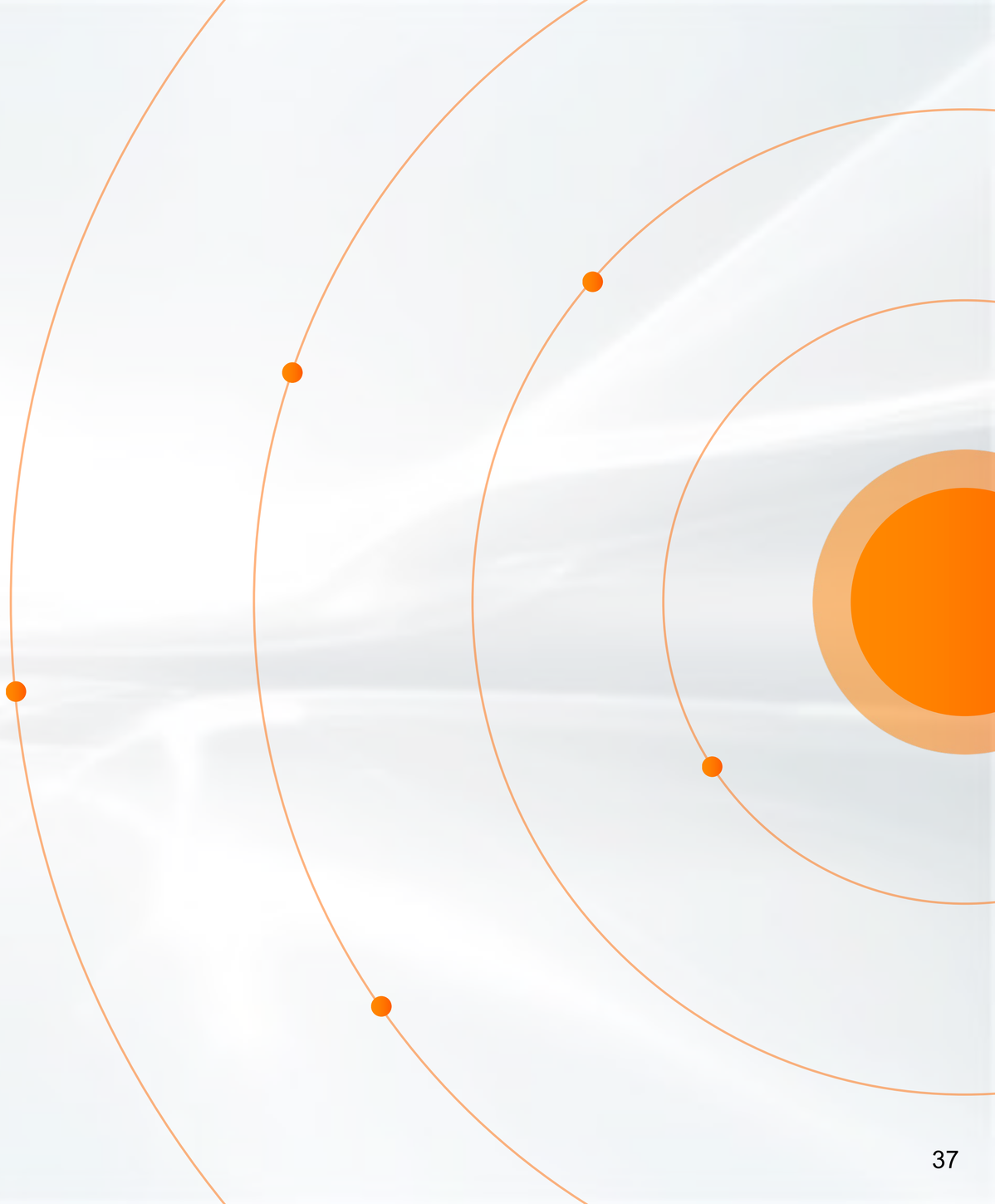
Analytical Queries in MySQL

Disk-Bound Queries

How to Tune Your Complex Queries

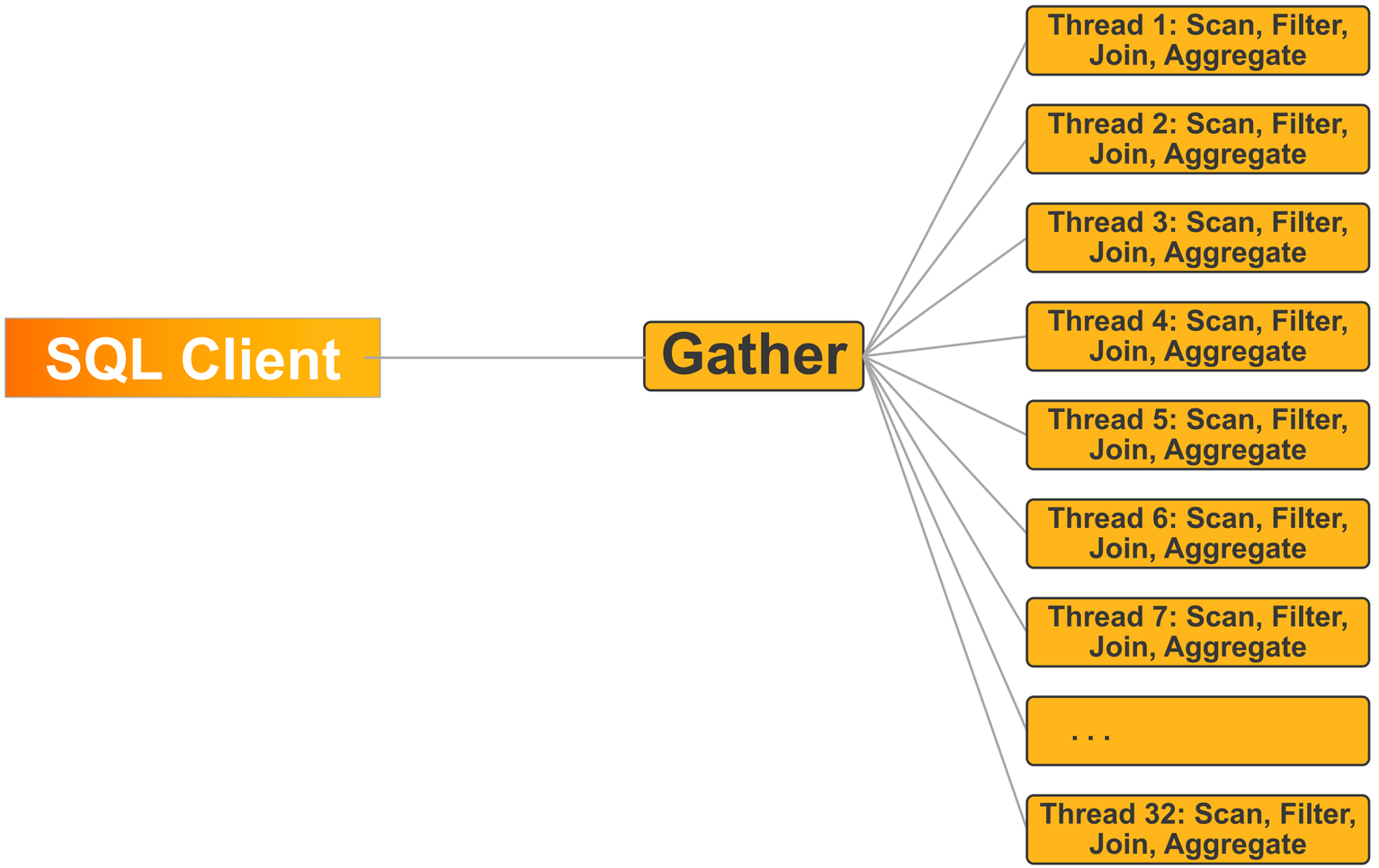
Hash Join

MySQL Wishlist



Alibaba Cloud

POLARDB Parallel Query

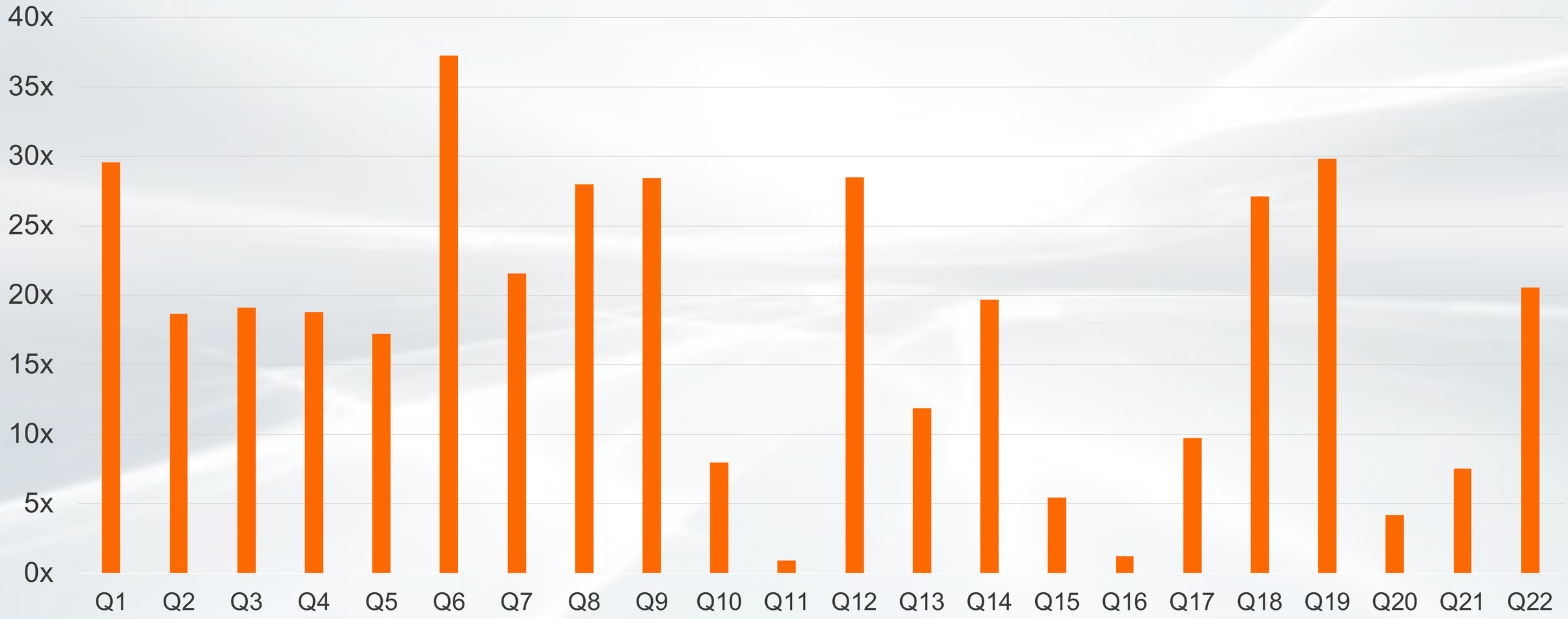


With 32 parallel threads, each thread does < 2% of the work

Alibaba Cloud

Scale-up, POLARDB Parallel Query

TPC-H Scale Factor 10, 32 physical cores



Alibaba Cloud

MySQL Wishlist for Complex Queries

- Implemented in POLARDB:
  - Parallel queries
  - Parallel hash join
  - Hash-join-aware optimizer
  - HASH_JOIN hint
  - Join cardinality based on histograms
- Other wishes:
  - Heap-based tables !!!
  - Improved cost model for index vs table scan (Bug#99994)
  - Fix inaccuracies in InnoDB range estimates (Bug#73386)
  - Cost model for BKA
  - Use histograms to estimate LIKE prefixes
  - Improve concurrency for secondary index scans (Bug#74280/Bug#74283)
  - Statistics on column correlation
  - Statistics on average column length

Alibaba Cloud

Thank you