

Testing Computation Pushdown in Distributed Database Systems

Jinsheng Ba¹, Zuming Jiang², Zhendong Su³

ETH Zurich¹, The University of Hong Kong²

ETH zürich



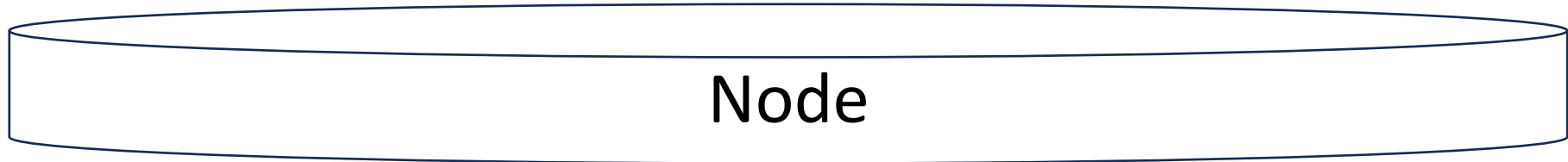
香 港 大 學

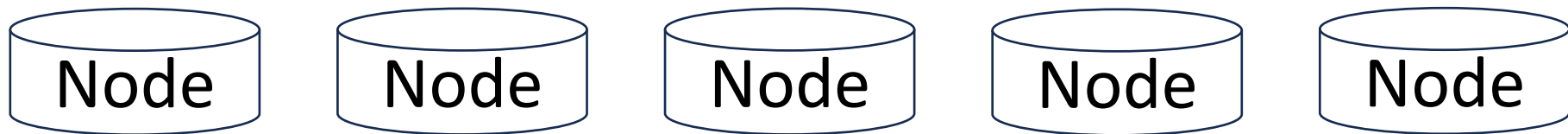
THE UNIVERSITY OF HONG KONG



CockroachDB





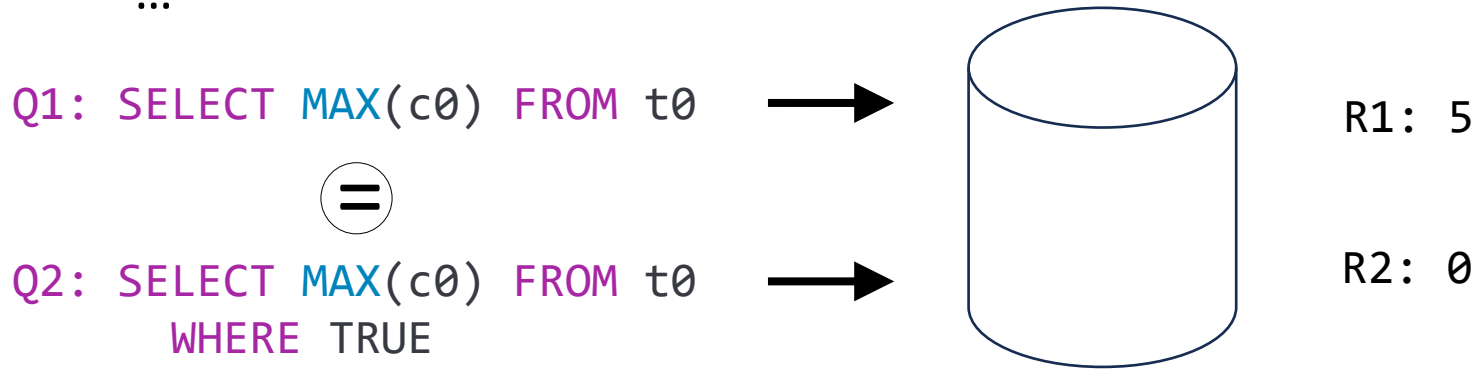


How to find bugs in distributed database systems?

Why can't existing methods test distributed database systems?

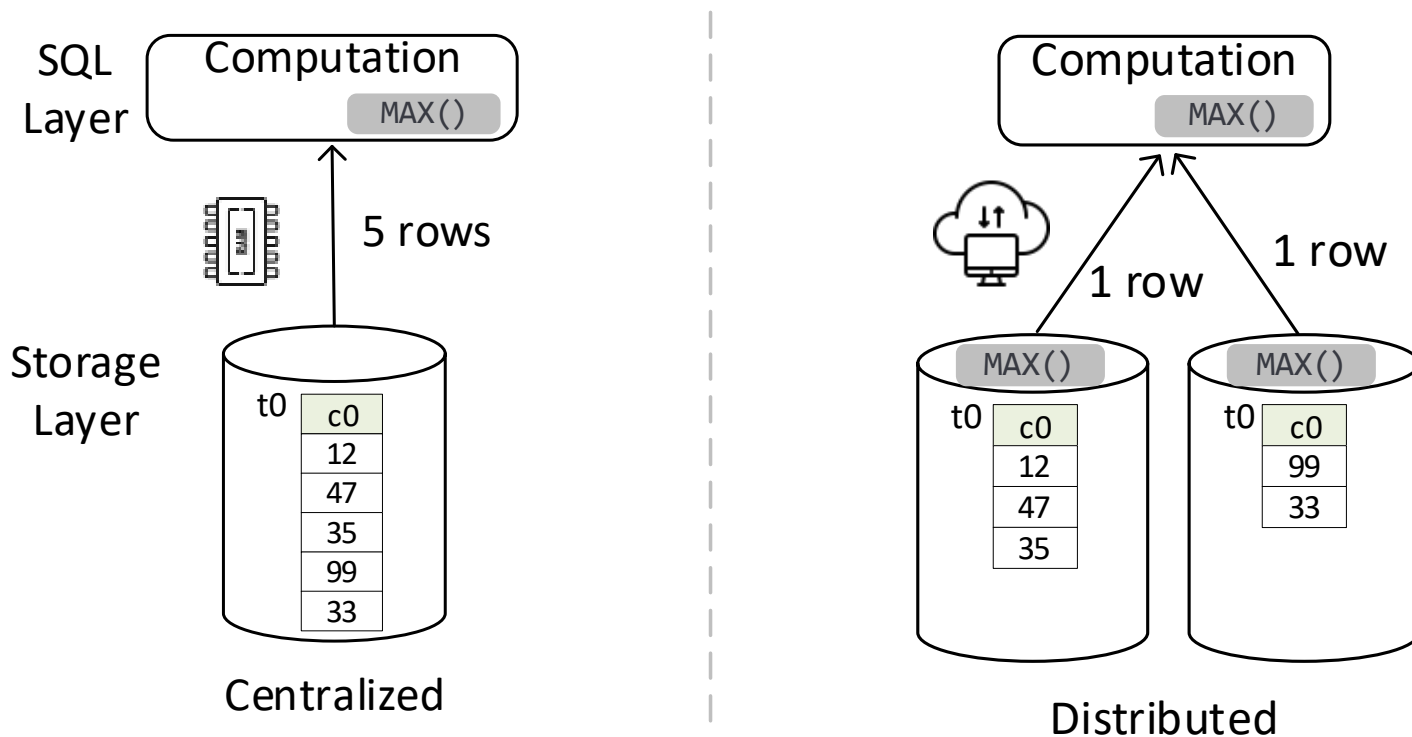
Existing Database Testing Methods

- TLP
- NoREC
- ...



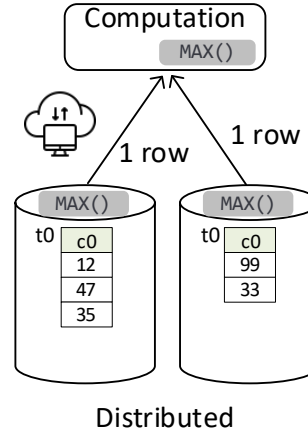
Q1 = Q2, so a bug exists if R1 != R2

SELECT MAX(c0) FROM t0



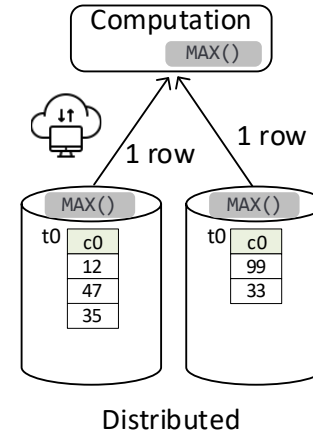
Pushdown Optimization

Q1: SELECT MAX(c0) FROM t0



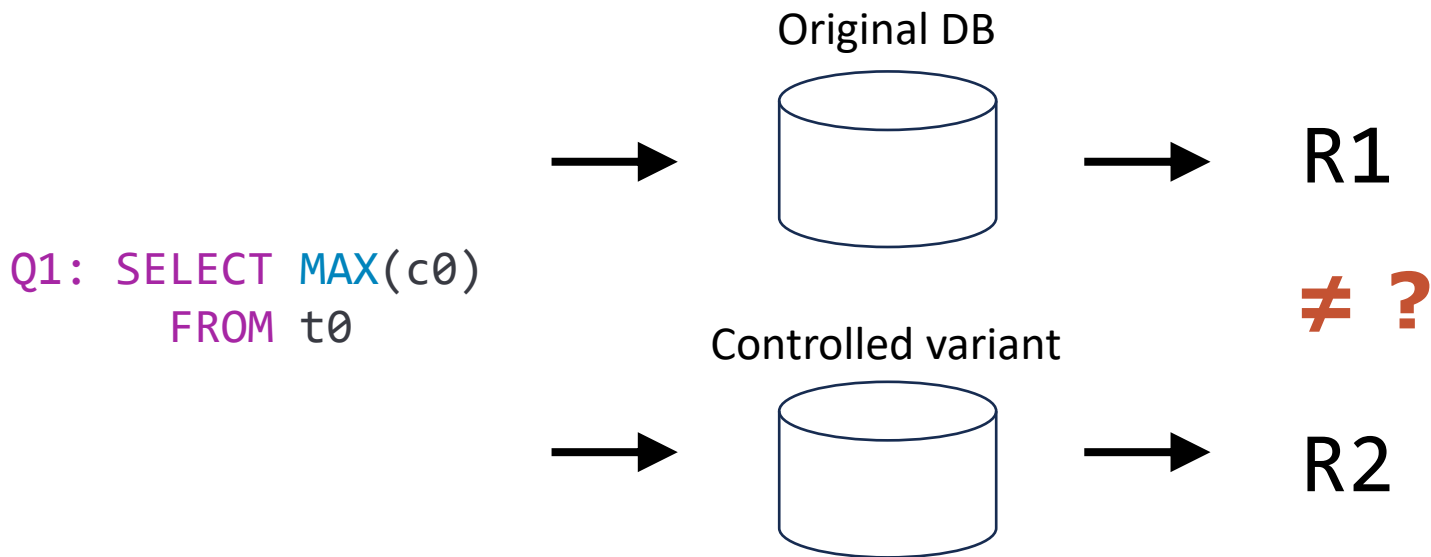
Q2: SELECT MAX(c0) FROM t0
WHERE TRUE

Same faulty pushdown path



Pushdown depends on data storage, network,
and other environmental factors.

Our Idea



Construct a DB variant to control pushdown

How to Construct Variants?

DBMS	Aggregate	Distinct	Expression	Function	GroupBy	Join	Topk	Union	Sum
CockroachDB	✓	✓	✓	✓	✓	✓	✓	✓	8
TiDB	✓	✓	✓	✓	✓	✓	✓	✓	8
YugabyteDB	✓	✓	✓	✓	X	X	X	X	4

```
yb_can_pushdown_max(...) {  
    ...  
    return true;  
}
```

All studied pushdown optimizations are controlled by such scheduling functions

Scheduling Function Identification

Static Matching



`*push*(...)`



Functional Validation



Run official
test cases



Semantic Checking



Inspect function
semantics

YugabyteDB:

62

26

13

How Accurate It Is

DBMS	All	Pushdown
CockroachDB	12	12
TiDB	26	25
YugabyteDB	13	13
Sum:	51	50

98 % identified functions are scheduling functions

Pushdown

DBMS	Aggregate	Distinct	Expression	Function	GroupBy	Join	Topk	Union	Sum
CockroachDB	✓	✓	✓	✓	✓	✓	✓	✓	8
TiDB	✓	✓	✓	✓	✓	✓	✓	✓	8
YugabyteDB	✓	✓	✓	✓	✗	✗	✗	✗	4

We do not miss any known pushdown optimizations

Testing

src/postgres/src/backend/optimizer/path/indxpath.c:

```
bool yb_can_pushdown_distinct(...) {  
    ...  
    return true;  
}
```

→ 0

```
bool yb_can_pushdown_distinct(...) {  
    ...  
    return true;  
    + return false;  
}
```

→ 5

≠



=



SELECT MAX(c0)
FROM t0

Challenge: It is too slow to modify each scheduling function and recompile

YugabyteDB: 1-3 hours on 16 core CPUs!

Recompilation is not needed

Testing Process

```
while(true) {  
    write_memory(var1, 1);  
    r1 = execute_db();  
    r2 = execute_variant();  
    if (r1 != r2) {  
        log(a bug);  
    }  
}
```

Database Process

```
yb_can_pushdown_distinct(...) {  
    ...  
    if (shared_memory(var1) == True)  
        return true;  
    else  
        return false;  
}
```

Shared memory

Var1 = 1

Evaluation

- 13 logic bugs
- Cover 5 out of the 8 pushdown operators
- 9 bugs Major/High severity

DBMS	Bug ID	Category	Operator	Status
CockroachDB	145108	Logic	Expression	Confirmed
CockroachDB	148279	Logic	Expression	Confirmed
CockroachDB	145888	Crash	-	Fixed
CockroachDB	146065	Crash	-	Fixed
TiDB	60586	Logic	Expression	Fixed
TiDB	60624	Logic	Function	Confirmed
TiDB	60630	Logic	Expression	Confirmed
TiDB	60664	Logic	Function	Confirmed
TiDB	60674	Logic	-	Confirmed
TiDB	60841	Logic	Function	Confirmed
TiDB	60958	Logic	Join	Confirmed
TiDB	60960	Logic	Function	Confirmed
TiDB	61664	Logic	Aggregate	Confirmed
TiDB	61683	Crash	-	Confirmed
TiDB	61684	Crash	-	Fixed
TiDB	61694	Crash	-	Confirmed
TiDB	61715	Crash	-	Fixed
TiDB	61735	Crash	-	Fixed
YugabyteDB	26620	Logic	Aggregate	Confirmed
YugabyteDB	26717	Logic	Distinct	Fixed
YugabyteDB	27031	Logic	Expression	Fixed
YugabyteDB	26753	Crash	-	Confirmed
YugabyteDB	27009	Crash	-	Fixed
YugabyteDB	27029	Crash	-	Confirmed
YugabyteDB	27059	Crash	-	Confirmed
Sum:				25

Bug Example

```
CREATE TABLE orders(order_id INT, order_date DATE);
CREATE INDEX i0 ON orders(order_date, order_date ASC);
INSERT INTO orders
    VALUES(1, '2025-08-20'), (2, '2025-08-21'), (3, '2025-08-25');
SELECT DISTINCT 'exist' FROM orders WHERE orders.order_date<'2025-09-01';
-- {exist} without pushdown, {} with pushdown
```

```
src/postgres/src/backend/optimizer/path/indxpath.c
-4757,6 +4757,7  yb_can_pushdown_distinct(...) {
--         return true;  // with pushdown
++         return false; // without pushdown
```

Bug Example

-----EET-----

```
SELECT DISTINCT 'a' FROM t0 WHERE t0.c0<5; -- {}  
SELECT DISTINCT 'a' FROM t0 WHERE ((t0.c0 = 100) AND NOT (t0.c0 =  
100) AND (t0.c0 = 100 IS NOT NULL)) OR (t0.c0 < 5); -- {}
```

-----TLP-----

```
SELECT DISTINCT 'a' FROM t0; -- {}  
SELECT DISTINCT 'a' FROM t0 WHERE t0.c0<5 IS TRUE UNION ALL SELECT  
DISTINCT 'a' FROM t0 WHERE t0.c0<5 IS FALSE UNION ALL SELECT  
DISTINCT 'a' FROM t0 WHERE t0.c0<5 IS NULL; -- {}
```

Conclusion



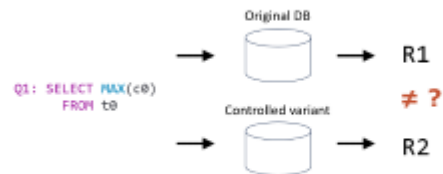
How to find bugs in distributed database systems?

`SELECT MAX(c0) FROM t0`



Pushdown Optimization

Our Idea



Construct a DB variant to control pushdown

Scheduling Function Identification



Pushdown

DBMS	Aggregate	Distinct	Expression	Function	Groupby	Join	Topk	Union	Sum
CockroachDB	✓	✓	✓	✓	✓	✓	✓	✓	8
TiDB	✓	✓	✓	✓	✓	✓	✓	✓	8
YugabyteDB	✓	✓	✓	✓	X	X	X	X	4

We do not miss any known pushdown optimizations

Recompilation is not needed

