Batched Nested Loop Joins

Tanuj Nayak
Feb 3, 2023

YugabyteDB Friday Tech Talks

YFTT

Talks by Engineers for Engineers

yugabyteDB

Joins

Let’s make pen-pals!

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball

Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

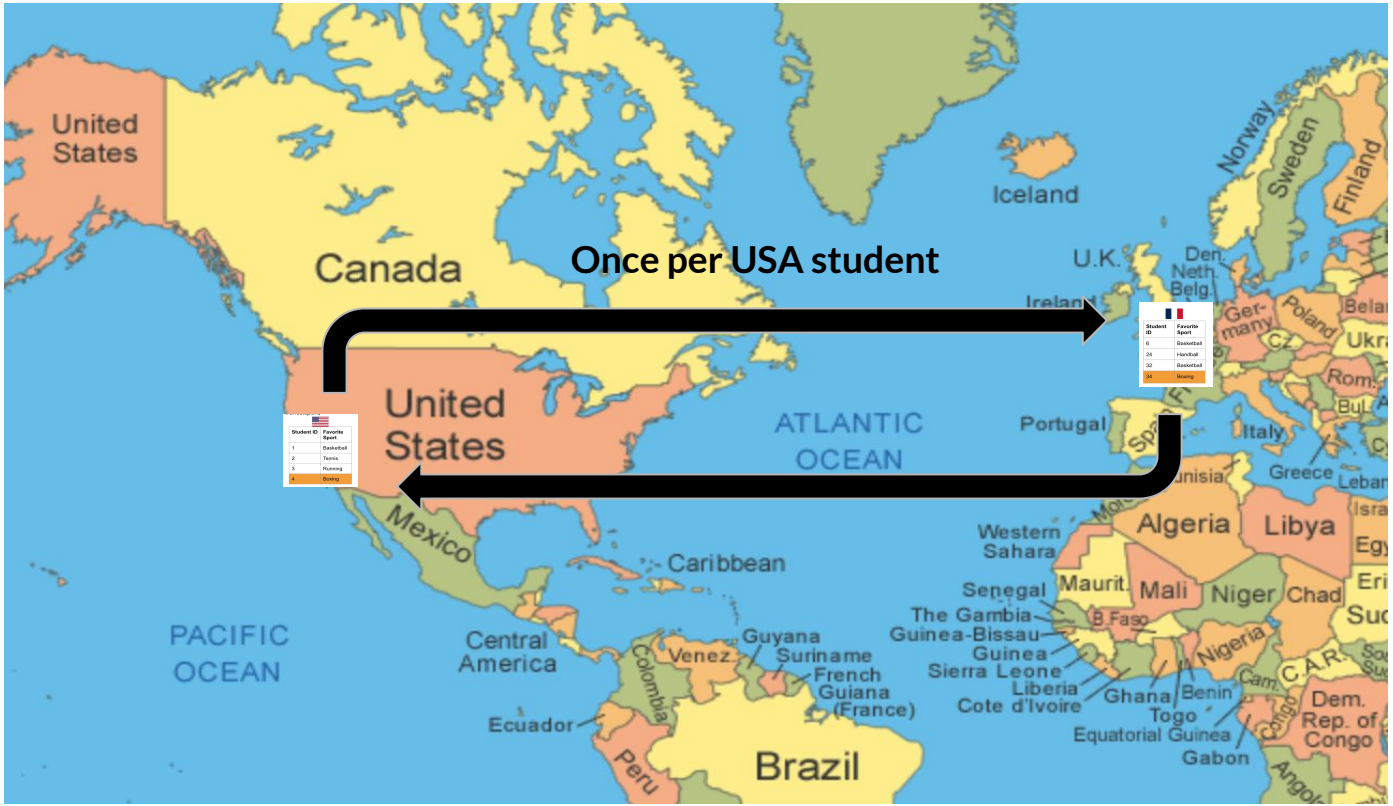
Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball
1	34	Boxing

So What's The Issue?

Distributed joins

Distributed joins



Joins

- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT

Joins

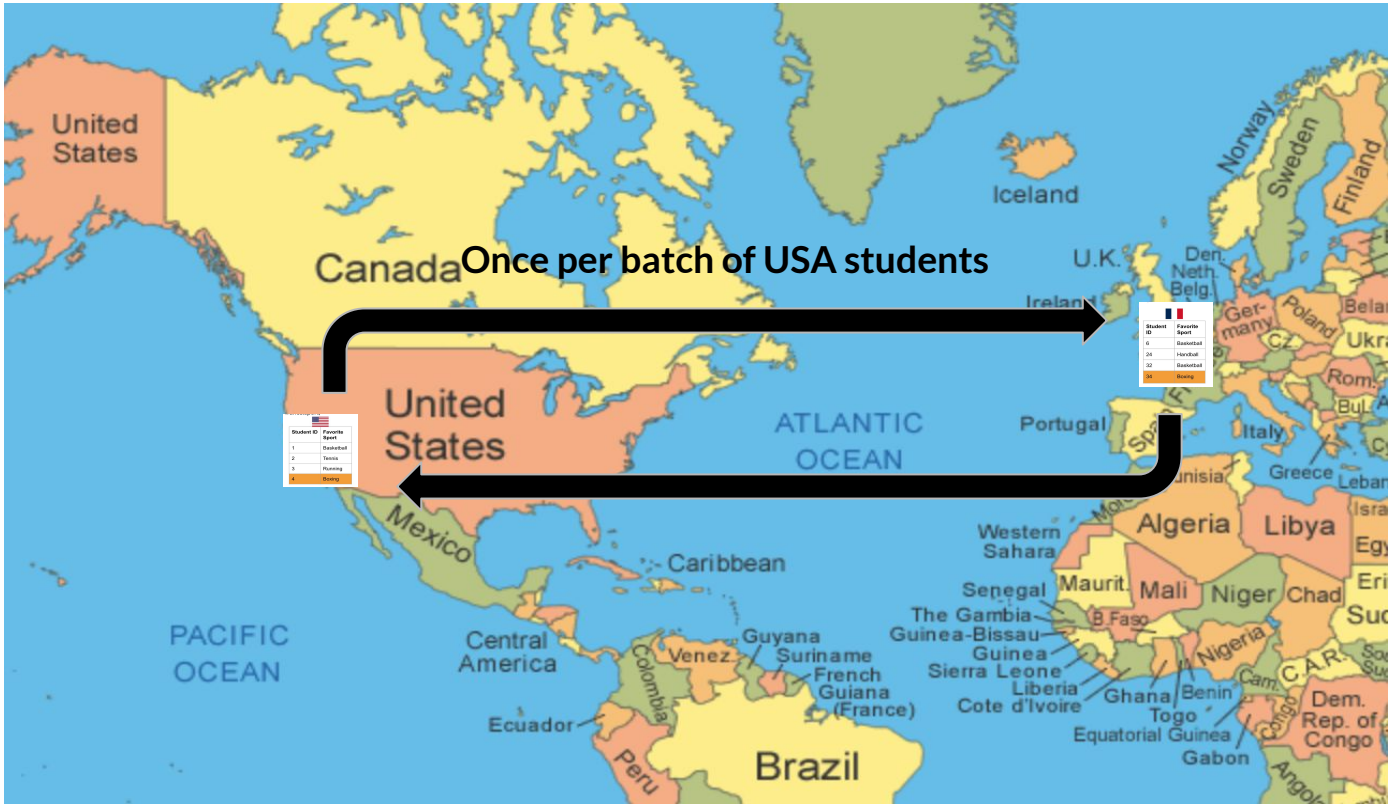
- Let's make pen-pals!
- SELECT usa.id, france.id FROM usa, france WHERE usa.sport = france.sport;

Student ID	Favorite Sport
1	Basketball
2	Tennis
3	Running
4	Boxing

Student ID	Favorite Sport
6	Basketball
24	Handball
32	Basketball
34	Boxing

USA ID	FR ID	SPORT
1	6	Basketball
1	32	Basketball
4	34	Boxing

Distributed joins



Join batching in action

```
create table t1(a int, b int, primary key(a asc, b asc));

create table t2(p int, q int, primary key(p desc, q desc));

insert into t1 select i, i+1 from generate_series(1,100000) i where i % 7 = 0;

insert into t1 select i, i-1 from generate_series(1,100000) i where i % 7 = 0;

insert into t2 select i, i+1 from generate_series(1,100000) i where i % 13 = 0;
```

Join batching in action

```
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                                QUERY PLAN
-------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..2619.28 rows=15384 width=16)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = t2.p)
(4 rows)
```

Join batching in action

```
yugabyte=# set yb_bnl_batch_size to 1024;
SET
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                                QUERY PLAN
-------------------------------------------------------------------------------
 YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16)
   Join Filter: (t1.a = t2.p)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
(5 rows)
```

Join batching in action

```
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                              QUERY PLAN
-----------------------------------------------------------------------
 Nested Loop  (cost=0.00..2619.28 rows=15384 width=16)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = t2.p)
(4 rows)
```

```
yugabyte=# set yb_bnl_batch_size to 1024;
SET
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                                  QUERY PLAN
-------------------------------------------------------------------------------
 YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16)
   Join Filter: (t1.a = t2.p)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
(5 rows)
```

Join batching in action

```
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                              QUERY PLAN
-----------------------------------------------------------------------
 Nested Loop  (cost=0.00..2619.28 rows=15384 width=16)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = t2.p)
(4 rows)
```

```
yugabyte=# set yb_bnl_batch_size to 1024;
SET
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                                 QUERY PLAN
-----------------------------------------------------------------------------
 YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16)
   Join Filter: (t1.a = t2.p)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
(5 rows)
```

Join batching in action

```
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                              QUERY PLAN
-----------------------------------------------------------------------
 Nested Loop  (cost=0.00..2619.28 rows=15384 width=16)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = t2.p)
(4 rows)
```

```
yugabyte=# set yb_bnl_batch_size to 1024;
SET
yugabyte=# explain select * from t1,t2 where t1.a = t2.p;
                                  QUERY PLAN
-------------------------------------------------------------------------------
 YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16)
   Join Filter: (t1.a = t2.p)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8)
         Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
(5 rows)
```

Join batching in action

```
yugabyte=# set yb_bnl_batch_size to 1;
SET
yugabyte=# explain (analyze, dist) select * from t1,t2 where t1.a = t2.p;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..2619.28 rows=15384 width=16) (actual time=5.369..1540.282 rows=2196 loops=1)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8) (actual time=2.923..7.759 rows=7692 loops=1)
         Storage Table Read Requests: 8
         Storage Table Execution Time: 2.835 ms
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8) (actual time=0.191..0.191 rows=0 loops=7692)
         Index Cond: (a = t2.p)
         Storage Index Read Requests: 1
         Storage Index Execution Time: 0.174 ms
 Planning Time: 0.174 ms
 Execution Time: 1540.624 ms
 Storage Read Requests: 7700
 Storage Write Requests: 0
 Storage Execution Time: 1339.648 ms
 Peak Memory Usage: 14 kB
(14 rows)
```

Join batching in action

```
yugabyte=# explain (analyze, dist) select * from t1,t2 where t1.a = t2.p;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16) (actual time=7.050..88.505 rows=2196 loops=1)
   Join Filter: (t1.a = t2.p)
   ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8) (actual time=2.401..3.503 rows=7692 loops=1)
         Storage Table Read Requests: 8
         Storage Table Execution Time: 2.330 ms
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8) (actual time=3.047..9.755 rows=274 loops=8)
         Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
         Storage Index Read Requests: 1
         Storage Index Execution Time: 2.814 ms
 Planning Time: 0.917 ms
 Execution Time: 89.047 ms
 Storage Read Requests: 16
 Storage Write Requests: 0
 Storage Execution Time: 24.844 ms
 Peak Memory Usage: 1012 kB
(15 rows)
```

Pros of Batched Nested Loop Joins

- **Lower memory usage**
  - Hash joins require the entirety of one table in memory
  - Merge joins require input tables to be sorted
- **Good for join filters that have high selectivity**
  - Great if your inner table has a small fraction of useful data

Watch out for sorting!

- Normal nested loop joins preserve the ordering of its outer table

```
yugabyte=# set yb_bnl_batch_size to 1;
SET
yugabyte=# select * from t1,t2 where t1.a = t2.p limit 20;
   a   |   b   |   p   |   q
-------+-------+-------+-------
 99918 | 99917 | 99918 | 99919
 99918 | 99919 | 99918 | 99919
 99827 | 99826 | 99827 | 99828
 99827 | 99828 | 99827 | 99828
 99736 | 99735 | 99736 | 99737
 99736 | 99737 | 99736 | 99737
 99645 | 99644 | 99645 | 99646
 99645 | 99646 | 99645 | 99646
 99554 | 99553 | 99554 | 99555
 99554 | 99555 | 99554 | 99555
 99463 | 99462 | 99463 | 99464
 99463 | 99464 | 99463 | 99464
 99372 | 99371 | 99372 | 99373
 99372 | 99373 | 99372 | 99373
 99281 | 99280 | 99281 | 99282
 99281 | 99282 | 99281 | 99282
 99190 | 99189 | 99190 | 99191
 99190 | 99191 | 99190 | 99191
 99099 | 99098 | 99099 | 99100
 99099 | 99100 | 99099 | 99100
(20 rows)
```

Cons of Batched Nested Loop Joins

- Batched nested loop joins do not preserve the ordering of its outer table

```
yugabyte=# set yb_bnl_batch_size to 1;
SET
yugabyte=# select * from t1,t2 where t1.a = t2.p limit 20;
   a   |   b   |   p   |   q
-------+-------+-------+-------
 99918 | 99917 | 99918 | 99919
 99918 | 99919 | 99918 | 99919
 99827 | 99826 | 99827 | 99828
 99827 | 99828 | 99827 | 99828
 99736 | 99735 | 99736 | 99737
 99736 | 99737 | 99736 | 99737
 99645 | 99644 | 99645 | 99646
 99645 | 99646 | 99645 | 99646
 99554 | 99553 | 99554 | 99555
 99554 | 99555 | 99554 | 99555
 99463 | 99462 | 99463 | 99464
 99463 | 99464 | 99463 | 99464
 99372 | 99371 | 99372 | 99373
 99372 | 99373 | 99372 | 99373
 99281 | 99280 | 99281 | 99282
 99281 | 99282 | 99281 | 99282
 99190 | 99189 | 99190 | 99191
 99190 | 99191 | 99190 | 99191
 99099 | 99098 | 99099 | 99100
 99099 | 99100 | 99099 | 99100
(20 rows)
```

```
yugabyte=# set yb_bnl_batch_size to 1024;
SET
yugabyte=# select * from t1,t2 where t1.a = t2.p limit 20;
   a   |   b   |   p   |   q
-------+-------+-------+-------
 86723 | 86722 | 86723 | 86724
 86723 | 86724 | 86723 | 86724
 86814 | 86813 | 86814 | 86815
 86814 | 86815 | 86814 | 86815
 86905 | 86904 | 86905 | 86906
 86905 | 86906 | 86905 | 86906
 86996 | 86995 | 86996 | 86997
 86996 | 86997 | 86996 | 86997
 87087 | 87086 | 87087 | 87088
 87087 | 87088 | 87087 | 87088
 87178 | 87177 | 87178 | 87179
 87178 | 87179 | 87178 | 87179
 87269 | 87268 | 87269 | 87270
 87269 | 87270 | 87269 | 87270
 87360 | 87359 | 87360 | 87361
 87360 | 87361 | 87360 | 87361
 87451 | 87450 | 87451 | 87452
 87451 | 87452 | 87451 | 87452
 87542 | 87541 | 87542 | 87543
 87542 | 87543 | 87542 | 87543
(20 rows)
```

Cons of Batched Nested Loop Joins

- Batched nested loop joins do not preserve the ordering of its outer table

```
yugabyte=# explain analyze select * from t1,t2 where t1.a = t2.p order by t2.p desc;
                                                                 QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=0.00..2700.20 rows=15384 width=16) (actual time=45.056..6708.822 rows=2196 loops=1)
   ->  Index Scan using t2_pkey on t2  (cost=0.00..850.12 rows=7692 width=8) (actual time=38.408..73.873 rows=7692 loops=1)
   ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8) (actual time=0.809..0.810 rows=0 loops=7692)
         Index Cond: (a = t2.p)
 Planning Time: 0.211 ms
 Execution Time: 6709.560 ms
 Peak Memory Usage: 8 kB
(7 rows)
```

```
yugabyte=# explain analyze select * from t1,t2 where t1.a = t2.p order by t2.p desc;
                                                                    QUERY PLAN
--------------------------------------------------------------------------------------------------------------------------------------------
 Sort  (cost=2033.27..2071.73 rows=15384 width=16) (actual time=350.783..350.930 rows=2196 loops=1)
   Sort Key: t1.a DESC
   Sort Method: quicksort  Memory: 217kB
   ->  YB Batched Nested Loop Join  (cost=0.00..963.38 rows=15384 width=16) (actual time=85.134..349.825 rows=2196 loops=1)
         Join Filter: (t1.a = t2.p)
         ->  Seq Scan on t2  (cost=0.00..769.20 rows=7692 width=8) (actual time=41.830..47.262 rows=7692 loops=1)
         ->  Index Scan using t1_pkey on t1  (cost=0.00..0.22 rows=2 width=8) (actual time=35.034..35.416 rows=274 loops=8)
               Index Cond: (a = ANY (ARRAY[t2.p, $1, $2, ..., $1023]))
 Planning Time: 1.947 ms
 Execution Time: 351.694 ms
 Peak Memory Usage: 1214 kB
(11 rows)
```

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YugabyteDB Friday Tech Talks

YFTT

Talks by Engineers for Engineers

yugabyteDB