



Asynchronous Query Execution

Alexander Rubin
April 12, 2015

About Me

Alexander Rubin, Principal Consultant, Percona

- Working with MySQL for over 10 years
 - Started at MySQL AB, Sun Microsystems, Oracle (MySQL Consulting)
 - Worked at Hortonworks (Hadoop company)
 - Joined Percona in 2013

Problem Set

Problem 1: Reporting query takes too long

- Use 1 CPU core only!
- Does not take advantage of my big server!

Problem Set

Problem 2: Pagination is slow

SELECT ... ORDER BY ... LIMIT 10

– Works very fast

SELECT COUNT(1) ... ORDER BY

– Is really slow

Problem Set

Problem 3: A query slows down page load

```
INSERT INTO page_log VALUES (...)
```

- Only used for internal logs
- Makes all pages load slow

Problem Set

Customers unhappy



Answer

“async”
query
execution

Agenda

- Splitting 1 query into N threads in the code
- “Bash” script example
- PHP asynchronous code example

Problem 1: pagination query

```
mysql> select FlightDate, Carrier, origin, dest, ActualElapsedTime  
-> from ontime  
-> where origin = 'SFO'  
-> order by FlightDate desc limit 10;
```

FlightDate	Carrier	origin	dest	ActualElapsedTime
2013-10-31	B6	SFO	FLL	316
2013-10-31	B6	SFO	FLL	307
2013-10-31	B6	SFO	JFK	298
2013-10-31	B6	SFO	AUS	201
2013-10-31	B6	SFO	LGB	84
2013-10-31	B6	SFO	LGB	78
2013-10-31	B6	SFO	BOS	313
2013-10-31	B6	SFO	BOS	315
2013-10-31	B6	SFO	BOS	336
2013-10-31	B6	SFO	JFK	343

```
10 rows in set (0.00 sec)
```

Problem 1: pagination query

```
mysql> select count(*)  
-> from ontime  
-> where origin = 'SFO';
```

```
+-----+  
| count(*) |  
+-----+  
| 3433692 |  
+-----+
```

1 row in set (1.52 sec)

Problem 1: pagination query

```
mysql> explain select count(*) from ontime where origin = 'SF0'\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: ontime
         type: ref
possible_keys: airport_date
          key: airport_date
        key_len: 6
         ref: const
        rows: 6735366
  Extra: Using where; Using index
1 row in set (0.00 sec)
```

Problem 1: pagination query

```
mysql> explain select FlightDate, Carrier, origin, dest,
ActualElapsedTime
from ontime where origin = 'SFO'
order by FlightDate desc limit 10\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: ontime
         type: ref
possible_keys: airport_date
          key: airport_date
    key_len: 6
         ref: const
        rows: 6735366
     Extra: Using where
```

Problem 1: pagination query

```
mysql> select SQL_CALC_FOUND_ROWS FlightDate, Carrier, origin, dest,
ActualElapsedTime from ontime where origin = 'SFO' order by FlightDate desc
limit 10;
```

FlightDate	Carrier	origin	dest	ActualElapsedTime
2013-10-31	B6	SFO	FLL	316
2013-10-31	B6	SFO	FLL	307
2013-10-31	B6	SFO	JFK	298
2013-10-31	B6	SFO	AUS	201
2013-10-31	B6	SFO	LGB	84
2013-10-31	B6	SFO	LGB	78
2013-10-31	B6	SFO	BOS	313
2013-10-31	B6	SFO	BOS	315
2013-10-31	B6	SFO	BOS	336
2013-10-31	B6	SFO	JFK	343

10 rows in set (23.06 sec)

```
mysql> select found_rows();
```

found_rows()
3433692

Problem 1: Solution

- Run the main query (LIMIT 10, 0.00 sec) first
- Run the second query (COUNT, 2 sec) after
 - Asynchronously
- Update the COUNT on top of the report
 - Javascript comes handy

Problem 2: reporting query

Which airlines have maximum delays for the flights inside continental US during the business days from 1988 to 2009?

Problem 2: reporting query

```
SELECT
    min(yeard), max(yeard), Carrier, count(*) as cnt,
    sum(ArrDelayMinutes>30) as flights_delayed,
    round(sum(ArrDelayMinutes>30)/count(*),2) as rate
FROM ontime
WHERE
    DayOfWeek not in (6,7) and OriginState not in
('AK', 'HI', 'PR', 'VI')
    and DestState not in ('AK', 'HI', 'PR', 'VI')
    and flightdate < '2010-01-01'
GROUP by carrier
HAVING cnt > 100000 and max(yeard) > 1990
ORDER by rate DESC
```


Problem 2: reporting query

min(yeard)	max(yeard)	Carrier	cnt	flights_delayed	rate
2003	2009	EV	1454777	237698	0.16
2006	2009	XE	1016010	152431	0.15
2006	2009	YV	740608	110389	0.15
2003	2009	B6	683874	103677	0.15
2003	2009	FL	1082489	158748	0.15

...

24 rows in set (15 min 56.40 sec)

Problem 2: potential solution

```
#!/bin/bash
fn="./res$$$.txt"
for c in '9E' 'AA' 'AL' 'AQ' 'AS' 'B6' 'CO' 'DH' 'DL' 'EA' 'EV' 'F9'
'FL' 'HA' 'HP' 'ML' 'MQ' 'NW' 'OH' 'OO' 'PA' 'PI' 'PS' 'RU' 'TW' 'TZ'
'UA' 'US' 'WN' 'XE' 'YV'
do
    sql=" select min(yeard), max(yeard), Carrier, count(*) as cnt,
sum(ArrDelayMinutes>30) as flights_delayed,
round(sum(ArrDelayMinutes>30)/count(*),2) as rate FROM ontime WHERE
DayOfWeek not in (6,7) and OriginState not in ('AK', 'HI', 'PR',
'VI') and DestState not in ('AK', 'HI', 'PR', 'VI') and flightdate <
'2010-01-01' and carrier = '$c'"
    mysql -uroot ontime -e "$sql" >> "$fn" &
done
wait

sort -n $fn|uniq
```

Problem 2: potential solution

```
$ time ./airline_par.sh > /airline_par_res.txt
real    8m13.323s
user    0m0.064s
sys     0m0.068s
```

Problem 2: potential solution

```
$ head airline_par_res.txt
```

min(yeard)	max(yeard)	Carrier	cnt	flights_delayed	rate
1988	1988	AL	265654	26291	0.10
1988	1988	PS	32052	1367	0.04
1988	1989	PI	551858	56122	0.10
1988	1990	EA	579546	55616	0.10
1988	1991	PA	206841	19465	0.09
1988	2001	TW	2659963	280741	0.11
1988	2005	HP	2607603	235675	0.09

Cpu0	:	14.1%us,	1.7%sy,	0.0%ni,	79.5%id,	4.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu1	:	11.9%us,	3.9%sy,	0.0%ni,	82.1%id,	2.1%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu2	:	14.7%us,	1.3%sy,	0.0%ni,	80.0%id,	4.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu3	:	15.5%us,	1.7%sy,	0.0%ni,	79.1%id,	3.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu4	:	14.9%us,	1.3%sy,	0.0%ni,	81.5%id,	2.3%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu5	:	17.4%us,	2.0%sy,	0.0%ni,	76.6%id,	4.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu6	:	13.2%us,	1.3%sy,	0.0%ni,	84.1%id,	1.3%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu7	:	11.7%us,	1.3%sy,	0.0%ni,	84.3%id,	2.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu8	:	16.8%us,	1.7%sy,	0.0%ni,	78.8%id,	2.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu9	:	16.1%us,	2.0%sy,	0.0%ni,	79.5%id,	2.3%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu10	:	15.9%us,	1.7%sy,	0.0%ni,	79.7%id,	2.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu11	:	18.3%us,	2.0%sy,	0.0%ni,	77.0%id,	2.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu12	:	8.3%us,	1.7%sy,	0.0%ni,	89.4%id,	0.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu13	:	7.6%us,	1.3%sy,	0.0%ni,	90.4%id,	0.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu14	:	6.6%us,	0.3%sy,	0.0%ni,	92.4%id,	0.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu15	:	8.6%us,	1.3%sy,	0.0%ni,	89.4%id,	0.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu16	:	15.0%us,	0.3%sy,	0.0%ni,	84.3%id,	0.3%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu17	:	2.7%us,	0.3%sy,	0.0%ni,	95.7%id,	1.3%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu18	:	24.1%us,	1.3%sy,	0.0%ni,	74.6%id,	0.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu19	:	11.4%us,	0.3%sy,	0.0%ni,	87.3%id,	1.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu20	:	7.4%us,	1.0%sy,	0.0%ni,	90.6%id,	1.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu21	:	13.3%us,	0.7%sy,	0.0%ni,	85.4%id,	0.7%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu22	:	6.3%us,	0.7%sy,	0.0%ni,	92.1%id,	1.0%wa,	0.0%hi,	0.0%si,	0.0%st
Cpu23	:	6.3%us,	1.7%sy,	0.0%ni,	91.0%id,	1.0%wa,	0.0%hi,	0.0%si,	0.0%st

Problem 3: Asynchronous PHP

```
# Run queries in parallel:
foreach ($all_links as $linkid => $link) {
    $link->query("SELECT something FROM tableN WHERE ", MYSQLI_ASYNC);
}
$processed = 0;
do {
    $links = $errors = $reject = array();
    foreach ($all_links as $link) {
        $links[] = $errors[] = $reject[] = $link;
    }
    # loop to wait on results
    if (!mysqli_poll($links, $errors, $reject, 60)) {
        continue;
    }
    foreach ($links as $k=>$link) {
        if ($result = $link->reap_async_query()) {
            $res = $result->fetch_row();
            # Handle returned result
            mysqli_free_result($result);
        } else die(sprintf("MySQLi Error: %s", mysqli_error($link)));
        $processed++;
    }
} while ($processed < count($all_links));
```

Problem 3: Asynchronous PHP

<http://www.percona.com/blog/2013/03/06/accessing-xtradb-cluster-nodes-in-parallel-from-php-using-mysql-asynchronous-calls/>

Questions?



Thank you!

Blog: <http://www.arubin.org>