Saving Bandwidth

The case of missing metadata

Georgi Kodinov
Team Lead, MySQL SrvGen Team
Agenda

➢ Network footprint of a SELECT
➢ How not to send the stuff you don’t need
➢ The outcome of our efforts
Let’s Start With an Example

- CREATE TABLE mysqlers (pk_column INTEGER PRIMARY KEY, first_name VARCHAR(20), last_name VARCHAR(20), years_in_mysql INTEGER, joining_date DATE);
- INSERT INTO mysqlers VALUES(1, 'Georgi', 'Kodinov', 12, '2006-03-13');
- SELECT * FROM mysqlers;
Network Footprint Of a SELECT
Gathering and making sense of it
The Response Summary

1 count packet

5 column metadata packets

1 OK/EOF packet

1 result row packet
Network Footprint of mysql_query(): Details

• Request: 81 bytes on the wire
  – 27 bytes TCP
  – 23 bytes COM_QUERY
  – 22 bytes SQL query text

• Reply: 443 bytes on the wire
  – 389 bytes TCP
  – 32 bytes MySQL (packet # and length)
  – LPacket1: 1 b: Meta count
  – LPacket2: 60 b: Meta of pk_column
  – LPacket3: 62 b: Meta of first_name
  – LPacket4: 60 b: Meta of last_name
  – LPacket5: 70 b: Meta of years_in_mysql
  – LPacket6: 66 b: Meta of joining_date
  – LPacket7: 31 b: Row Data
  – LPacket8: 7 b: OK/EOF
Network Footprint of mysql_query()
How Not To Send
The Stuff You Don’t Need
First Naïve Attempt

```sql
mysql> select @@session.resultset_metadata;
+---------------+
| @@session.resultset_metadata |
| FULL          |
+---------------+
1 row in set (0.14 sec)

mysql> SET @@session.resultset_metadata=NONE;
ERROR 3640 (HY000): The client doesn't support optional metadata transfer
```
A Sample SELECT Program

We can deal with no metadata

SSL off so we can trace

We don’t need metadata!

Check lack of metadata

Data still here
The Outcome Of Our Efforts
A Sample SELECT Program: it works!

But How about the Network?
A Very Different Picture

106 Bytes !
It used to be 443 !

The metadata sub-packets are gone !
The “Before” And “After” Image: 320% Lost!
It Was a True Team Effort!

• “Thank you” goes to:
  – Davi Arnaut @ Twitter
  – Facebook
  – Ramil Kalimulin
  – The MySQL team @ Oracle
Tools Used

• Wireshark
  – https://www.wireshark.org/

• Oracle MySQL 8.0 client and server

• Protocol Doxygen in MySQL 8.0
Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.