SR 3-23363317521 : mysql router performance issues

Severity	2-Significant
Escalation Status	Never Escalated
Last Updated	Nov 20, 2020 12:55 PM (Friday)
Bug Reference	No Related Bugs
Status	Customer Working
Opened	Jun 19, 2020 4:28 PM (5+ months ago)
Attachments	cpu_info.txt, net_dump2_app01p.partaa, net_dump2_app01p.partab, net_dump2_app02p.partaa, net_dump2_app02p.partab, net_dump2_app02p.partac, net_dump2_app03p.partaa, net_dump2_app03p.partab, net_dump2_app04p.partaa, net_dump2_app04p.partab, net_dump2_app04p.partac, net_dump2_app05p.partaa, net_dump2_app05p.partab, net_dump2_app06p.partaa, net_dump2_app06p.partab, net_dump2_app06p.partac, net_dump2_app07p.partaa, net_dump2_app07p.partab, net_dump2_app07p.partac, net_dump2_app08p.partaa, net_dump2_app08p.partab, net_dump2_app08p.partac, net_dump2_app09p.partaa, net_dump2_app09p.partab, net_dump2_app10p.partaa, net_dump2_app10p.partab, net_dump2_app10p.partac, net_dump2_app10p_autorename1.partaa, net_dump2_dbc01p.partaa, net_dump2_dbc02p.partaa, net_dump2_dbc02p.partab, net_dump2_dbc02p.partac, net_dump2_dbc03p.partaa, net_dump2_dbc03p.partab, net_dump2_dbc03p.partac, net_dump_app01p.partaa, net_dump_app01p.partab, net_dump_app01p.partac, net_dump_app02p.partaa, net_dump_app02p.partab, net_dump_app02p.partac, net_dump_app03p.partaa, net_dump_app03p.partab, net_dump_app03p.partac, net_dump_app04p.partaa, net_dump_app04p.partab, net_dump_app04p.partac, net_dump_app05p.partaa, net_dump_app05p.partab, net_dump_app05p.partac, net_dump_app06p.partaa, net_dump_app06p.partab, net_dump_app07p.partaa, net_dump_app07p.partab, net_dump_app07p.partac, net_dump_app08p.partaa, net_dump_app08p.partab, net_dump_app08p.partac, net_dump_app09p.partaa, net_dump_app09p.partab, net_dump_app09p.partac, net_dump_app10p.partaa, net_dump_app10p.partab, net_dump_app10p.partac, net_dump_app10p_autorename1.partaa, net_dump_app10p_autorename1.partab, net_dump_app10p_autorename1.partac, net_dump_dbc01p.partaa, net_dump_dbc02p.partaa, net_dump_dbc02p.partab, net_dump_dbc02p.partac, net_dump_dbc03p.partaa, net_dump_dbc03p.partab, net_dump_dbc03p.partac, net_dump_dbc03p.partad, router-slavevms-500threads-mysqlonly.png, router-slavevms-500threads.png, slavevms-500threads-

			mysqlonly.png, slavevms-500threads.png, testfileupload, testfileupload_autorename1
Related Articles	No Related Articles	Related SRs	No Related SRs
Support Identifier	21922144		
Account Name	UCL		
Primary Contact	roy bhurtha	Alternate Contact	
System		Host	No Related Hosts
Product	MySQL Router		
Product Version	8.0		
Operating System	Linux x86-64	OS Version	Red Hat Enterprise 7

Problem Description

hello,

We are running a 3 node innodb cluster using mysql router.

Performance is significantly slower using the router than when we do not use it.

What can be done to improve the performance of the router?

Thanks,

Roy

History

Update from Customer — **R.BHURTHA@UCL.AC.UK**- Nov 20, 2020 12:55 PM (Friday)

Hello,

We are still performing tests, please leave the call open for a further week.

Thanks,

Roy

Update from Customer — **R.BHURTHA@UCL.AC.UK**- Oct 20, 2020 9:25 AM (1+ month ago)

Hello Shawn,

We have discovered that named server caching was not enabled. On enabling this, there is a noticeable improvement in performance. We will run more tests and update you.

In the meantime, please keel the call open.

Roy

ODM Action Plan — **Oracle Support**- Oct 9, 2020 8:06 PM (1+ month ago)

Hello Roy,

Sorry for the delay. I had some internal Oracle stuff come up and it took away a fair chunk of my usual time to spend handling service requests instead of managing.

I don't see anything that suggests those two options are mutually exclusive (listen to sockets or listen on ports). You can do both or either.

If you enable sockets, I do not see a need to disable TCP ports. You can if you want to but I don't see that it is required.

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer — **R.BHURTHA@UCL.AC.UK**- Oct 8, 2020 12:31 PM (1+ month ago)

Hi Shawn,

For option(1), do I need to set --conf-skip-tcp and --conf-use-sockets

Thanks,

Roy

ODM Action Plan — **Oracle Support**- Oct 7, 2020 2:26 PM (1+ month ago)

Hello Roy,

You only need to do some configuration changes. You do not need to do any installing.

You need to start by telling your cluster nodes to identify themselves (and as a side effect, each other) using their IP addresses. To do that, you may need to manually set the --report-host option on each node. That option wasn't designed for InnoDB Cluster to use this way but they did it anyway. That option was designed to allow humans to tag their MySQL servers with "friendly" names so that they were easier to recognize in MySQL Enterprise Monitor. For example, hosts used in one department may be named for fictional beasts (Grendel, Hydra, Chimera, ...) while others could be named for authors (Twain, Clarke, Byron, Yeats, ...). That field was meant to hold those nicknames.

Now, because that option was hijacked by a different team for another role, you need to be sure that what you set for --report-host is a network-navigable value. If you don't put an address here, it means you need to rely on your DNS system to convert an alias (like one of those names I mentioned) or a FQDN into an address. This is what the nodes will use to identify themselves to the other nodes in the group and how the group/cluster will identify its members to any Routers that connect to it.

That sets up your metadata so that the IC cluster is a collection of IP addresses (not names).

1 Dissolve the group
2 Rename your nodes using their addresses
3 Reform the group using their addresses

While this is a staged process, it does not require uninstalling or reinstalling anything. However --report-host is not a dynamic variable, it will require a mysqld restart to change.
https://dev.mysql.com/doc/refman/8.0/en/replication-options-replica.html#sysvar_report_host

That will allow you to then bootstrap a Router to point to the reformed cluster. Since the cluster self-identifies as a collection of IP addresses, Router won't need to use a DNS server to figure out where a name is on your network. If you know the map of names->addresses, you don't even need to do the bootstrap, just edit the configuration the last bootstrap created and substitute the correct IP addresses for the names. Once you have edited the configuration file, simply restart Router.

Then, the last place to fix is how you call Router from your application. Since you want to have Router located on the same host as your application, you would be using the loopback address or through a local socket.

(the --socket option for Router)
https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-conf-options.html#option_mysqlrouter_socket

Do you have any questions about the process I just described?

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Oct 7, 2020 1:11 PM (1+ month ago)
Hi Shawn,

If we wish to implement option(3) would we have to uninstall and then reinstall the router?

Thanks,

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Oct 7, 2020 10:43 AM (1+ month ago)
Hello Shawn,

We will be testing these changes this week and will update the call.

Thanks,

Roy

ODM Action Plan **Oracle Support**- Sep 25, 2020 3:28 PM (1+ month ago)
Hello Roy,

I believe there are still several things you can to do improve performance. Here are the first three that popped to mind.

1) Install Router on the application server (just like you needed to install a Connector/* library) and connect to it via a local socket. This skips an entire TCP/IP leg in your message path (even if that leg may be through the loopback address).

2) Setup and use a connection pool at your application to hand off existing sessions from one application thread to the next. Please note: you cannot easily share one active session with more than one active thread at a time. Our client-server protocol is stateful and cannot interleave active commands. Each application thread still needs its own session unless you can internally block threads from using a shared session pointer while the another thread is using it (this is pretty tricky to program properly because each session can also only have one active transaction). The simplest solution is to not try to share sessions and to allow each app thread to own its own session during that block of work.

Example: it's time to render a web page. That page needs data from 4 tables. Wait in your code until you have decided which commands to execute, get a session (live or from a pool), run those queries, cache those results, release that session, then process the results into HTML or other content.

That kind of "burst" pattern where all the database work is performed as fast as practical in your application keeps idle sessions to a minimum at the database. It also minimizes the time that any locks need to exist within your data which improves total concurrency. It does not ask the database to wait for user input (while managing an active transaction) and it doesn't persist that transaction while other unrelated operations (like streaming content to a browser) is happening.

3) Take the DNS system out of the picture by only using IP addresses for your nodes and your user accounts. Doing #1 eliminates DNS from the "app->router" leg of the journey automatically.

===

Then, remember that any changes you make to your data (while using a Group-Replication-based topology) requires all the nodes in the group to "certify" that transaction. If both tests (with and without Router) are to the same GR or InnoDB Cluster (IC) topology, then this extra time per transaction is part of your baseline measurement. If you are comparing IC (through Router) to an asynchronous replicating pair (without Router) then we have not yet taken that extra time into consideration.

Alistair tried to explain your layout and I tried to explain how I could read what he wrote different ways (I apologize for addressing that reply to you instead of him). Could either of you clear up my confusion?

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Update from Customer **R.BHURTHA@UCL.AC.UK**- Sep 25, 2020 11:19 AM (1+ month ago)

Hi Shawn,

I updated the routing straegy to first_available which has improved performance slightly.

However, the cluster still performs faster without the router.

Are there any other settings we can investigate.

Thanks,

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Sep 7, 2020 9:50 AM (2+ months ago)

Hii Shawn,

Thanks for the update. We are looking into modifying the routing plan 'first-available' as suggested and will update you shortly.

Roy

ODM Action Plan **Oracle Support**- Aug 26, 2020 6:28 PM (2+ months ago)

Hello Roy,

I'll presume you have the application logic in place (for now) to safely and consistently identify (across all of your application servers or direct-connecting users) which instance is in charge of which rows on which tables at every single moment.

For now, I'm just trying to translate your physical description into a mental image. I know that in a GR group or IC cluster of more than 2 nodes that you cannot have just one RO (read-only) node. I'm interpreting what you said to mean that each group in each DC operates in single-primary mode (one RW node - the Primary, one or more RO nodes - the Secondaries).

Please excuse the crudity of this models as I cant build them to scale or paint them. (This help system doesn't even have monospaced fonts)

"We have a 2 datacentre design running active-active, with a 3 node InnoDB cluster (1 RO in each DC)" ## If I interpret you to have meant "1 RW in each...") I get this picture.

```
DC1 DC2
==================== || ==============================
Primary Primary
| |
+ -Secondary 1 +- Secondary 1
| |
+- Secondary 2 +- Secondary 2
```

To me, the term "active-active" means that both DC1-Primary and DC2-Primary are active targets for writes from the clients that connect to them and that you have connected those nodes in a replication loop (which is outside of the normal Group Replication stuff).

```
DC1_primary --------> DC2_primary
^ |
\ --------------------------+
```

Is that what you meant by "active-active" ?

Or does the model look more like this?

```
DC1 DC2
==================== || ==============================
Primary ------------------------------------ Secondary 2
|
+ -Secondary 1
```

Or is the Primary in a different location than one of those two Secondaries entirely?

And, shall I presume that each of the larger DC's contains a set of one or more application servers and that you would strongly prefer that when that app server

needs to bounce through Router that it would prefer a LOCAL (within the DC) target if practical? If that is your preference, you and I need to wait. There is an internal project to implement a "preference system" like that but it is nowhere near ready for even testing.

The closest we can do today is to modify the order in which you list the nodes then choose "first-available" (if you are doing manual routing. If you are letting GR or IC drive the routing tables, I'm fairly sure we don't have that level of control yet).

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **A.SPARK@UCL.AC.UK**- Aug 26, 2020 3:23 PM (2+ months ago)
> I need to ask (for comparison purposes), without Router between your applications and the InnoDB Cluster, how did your applications choose which secondary to use? What was your criteria to determine which Secondary to talk to? Your strategy could influence the next generation of Router planning.

We have a 2 datacentre design running active-active, with a 3 node InnoDB cluster (1 RO in each DC) so without router we have hardcoded the hostname of the RO node which is within the same DC as the application servers in that DC. (equal number of app servers in each DC & significantly increased latency in connecting when mysql goes cross-DC).

I did suggest that we should think about having a 5 or 7 node cluster when we take this to prod for the impact of a DC being lost not being as significant. It would be nice if router could somehow be aware of which RO nodes are in which DC and try those first before going cross-DC (hence my comment about network response time awareness)

We'll have a look at first-available / next-available and come back to you.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Aug 26, 2020 8:52 AM (2+ months ago)
Thanks for the update Shawn.

The user application, is configured to send transactions to the read/write server. In addition, it will pass select queries to q nominated read-only server.

The connections to these servers are defined by hostname and port.

Roy

ODM Action Plan **Oracle Support**- Aug 17, 2020 5:47 PM (3+ months ago)
Hello Alistair,

Thank you. This was exactly the types of patterns I have been unsuccessful at teasing out of the data.

Going through your points individually:

>
> From the last TCP dump of MySQL router, the following settings were changed but had no impact on performance/throughput in the end:
1)MySQL traffic is encrypted over TLSv1.2 with Router and ordinarily isn't. This seems to be linked to https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-conf-options.html#option_mysqlrouter_ssl_mode which defaults to PREFERRED, we should set this to DISABLED and run another test
>

Correct.
Router is a proxy. It can sit within a DMZ. So the corporate communication requirements may need all external connections (client program <-->Router) to be encrypted but that could be relaxed on the inside of that boundary (Router <--> Database server) to avoid the overhead of TLS. That is why that configuration option exists.

This negotiation will add time to each connection attempt. If your applications have very short-lived sessions (and require a very lightweight login cycle) then disabling TLS is definitely going to help. Another option to consider is to have a connection pool. This will cache a set of pre-connected (already logged in) sessions between your application and the Router and that would reduce how many new sessions need to be created over the course of an "operating period" (however long your application is running).

>
> Now that TLS encryption is disabled, it makes seeing what's going on a bit easier.

> From this latest dump, what I'm seeing is:
> A) Regular re-negotiation of the authentication method https://dev.mysql.com/doc/refman/8.0/en/caching-sha2-pluggable-authentication.html
> We deferred dealing with switching to caching_sha2_password but seems like it's required.
>

You should be seeing this same negotiation for your non-Router sessions, too. The choice about which authentication plugin to use (which handshake method will happen) is determined by the definition of the Account (the username@hostpattern combination) at the server. Without TLS, the short version of this essentially looks like....

(client) (server)
connect to the port
respond with initial packet
identify user
<collects the name being presented and the host from where it is connecting>
<compares to the Accounts>
<finds most host-specifc Account from that list>

<reads the authentication_plugin and TLS requirements>
respond with TLS or skip to authentication handshake (if TLS is not required)

There are a LOT of details I left out of that summary (like "capabilities" negotiations....) but that is the general process. With router in the mix (and leaving out some notes) you have this flow:L

(client) (router) (cluster metadata) (server)
<get current change set>
<-- change set data

connect
<- <initial packet>
identify user
<- <TLS if required>
<determine which server to use>

connect -->
<-- respond with initial packet
identify user -->
<TLS and account determination>
<-- TLS or authentication handshake

Summary: there's another communication hop and potentially a second TLS negotiation involved in the establishment of the proxied connection.

>
> B) Round-robin routing strategy seems inherently flawed, as connection is constantly being re-negotiated with new servers. Need to get to a place where MySQL Router only changes server if it's performing poorly / not responding or the metadata says it's not in the pool anymore. That's really a feature request for Oracle but there are no doubt more settings Roy can tweak to improve that in the meantime. The TTL was one of them, there are others.
>

This needs to be broken into separate parts:
a) The connection is not being renegotiated. Connections negotiate once for the duration of that session. What you are very likely seeing is your application generating a lot of short-lived sessions.

b) Router's algorithms do not measure "load" on the backend machine. In a multi-node Cluster (example - nodes: F, G, and H where F is Primary, G and H are secondary), all read-write requests will bounce to a primary, all read-only requests will head to a secondary unless you are using round-robin-with-fallback.
https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-conf-options.html#option_mysqlrouter_routing_strategy

For an affinity-based Routing plan, we offer first-available : Let's say that the list was ordered as (G,H) and G was down (or too busy to respond in time). Router would send that new connection to H. When the next new connection arrives G is tested again. If it can negotiate with G, then that's where the proxied session will live.

This does not happen with next-available. When that algorithm is in use, if G dies - it is removed from the list (forgotten) until Router restarts or the list is regenerated. And the list will be regenerated (by InnoDB Cluster) each time a node changes its status inside the Cluster. So if the list comes back as (H, G) then the connections stay with H. If the list comes back as (G,H) then new sessions will start routing to G again and H will be there to catch the overflow.

So the "not responding" condition is something we can use inside of Router for a decision. "Performing poorly" would only be visible with a connection attempt timeout. Based on that, have you tried "next-available" ? (Router does not inspect packets, it's a simple connect-and-forward arrangement)

>
> You need to implement something more ProxySQL like in terms of load balancing where connections are persistent, nodes in the cluster are monitored for query execution time, network response time, not having been booted from the cluster and load balance according to that rather than stupidly switching between them all, that adds ZERO value, I'm sorry but round-robin for this is a really daft approach.
>
> If there are any other settings you can think of to optimise the traffic distribution and reduce the amount of renegotiation that would be much appreciated.
>

We left the choice of pooling sessions to the end user. We left our Router as "dumb" as practical to make its performance as predictable as practical. Collecting and analyzing performance metrics (such as determining the statistical spreads of response times for the same command digest from each backend) are beyond its current capabilities because it does NOT inspect the command packets it Routes. It facilitates the client's connection to an appropriate node then gets as far as it can out of the way.

I need to ask (for comparison purposes), without Router between your applications and the InnoDB Cluster, how did your applications choose which secondary to use? What was your criteria to determine which Secondary to talk to? Your strategy could influence the next generation of Router planning.

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Minor EDIT ---
When I wrote this "...The choice about which authentication plugin to use (which handshake method will happen) is determined by the definition of the Account (the username@hostpattern combination) at the server. Without TLS, the short version of this essentially looks like.... " I meant "Without Router" not "Without TLS"
-- Shawn

Update from Customer **A.SPARK@UCL.AC.UK**- Aug 17, 2020 2:38 PM (3+ months ago)
Hi Shawn,

Here's my reading of both waves of tcp dumps:

From the last TCP dump of MySQL router, the following settings were changed but had no impact on performance/throughput in the end:
1)MySQL traffic is encrypted over TLSv1.2 with Router and ordinarily isn't. This seems to be linked to https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-conf-options.html#option_mysqlrouter_ssl_mode which defaults to PREFERRED, we should set this to DISABLED and run another test
2)Connection is renegotiated very frequently
This would seem to linked to the metadata for the servers in the cluster being purged every 1/2 second https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-conf-options.html#option_mysqlrouter_ttl the TTL is ridiculously low in 8.0.12+(we are on 8.0.20) so if 1) doesn't sort this out then we could increase it to 20 but this is a double edged sword as it means that failover in case of a faulty primary node would take longer

Now that TLS encryption is disabled, it makes seeing what's going on a bit easier.

From this latest dump, what I'm seeing is:
A) Regular re-negotiation of the authentication method https://dev.mysql.com/doc/refman/8.0/en/caching-sha2-pluggable-authentication.html
We deferred dealing with switching to caching_sha2_password but seems like it's required.

B) Round-robin routing strategy seems inherently flawed, as connection is constantly being re-negotiated with new servers. Need to get to a place where MySQL Router only changes server if it's performing poorly / not responding or the metadata says it's not in the pool anymore. That's really a feature request for Oracle but there are no doubt more settings Roy can tweak to improve that in the meantime. The TTL was one of them, there are others.

New Relic defaults to saying PHP does something even when it's network stuff in between, so the MySQL time it records is purely the query execution time, none of the talking between servers.

These dumps repeatedly show that by going via MySQL router, the connections are not persistent and being renegotiated every few seconds, in my mind this will be the largest factor in the poor performance we are seeing. Queries pile up rather than being executed when under heavy load.

You need to implement something more ProxySQL like in terms of load balancing where connections are persistent, nodes in the cluster are monitored for query execution time, network response time, not having been booted from the cluster and load balance according to that rather than stupidly switching between them all, that adds ZERO value, I'm sorry but round-robin for this is a really daft approach.

If there are any other settings you can think of to optimise the traffic distribution and reduce the amount of renegotiation that would be much appreciated.

Thanks
Alistair

(The Moodle guy at UCL)

ODM Action Plan **Oracle Support**- Aug 14, 2020 4:22 PM (3+ months ago)
Hello Roy,

I am disappointed that I need to report that I have made scant little progress. I have resolved your captures back into readable text and I have done some exploring. I have spent what is probably WAY too much time trying to get a modern graphical tool (Wireshark) onto the linux image Oracle data security rules need me to use as a secondary workstation (your capture files were bigger than the free disk space I could create on my primary Windows workstation).

I'll ditch those efforts and resume my attempts to compare those files via terminal-based tooling.

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Aug 14, 2020 4:10 PM (3+ months ago)
Hello,

We will be uploading 2 trace files shortly which are the from the latest tests. These are from one DB server and one application server.

Please let me know if you have made any progress.

Thanks

Update from Customer **R.BHURTHA@UCL.AC.UK**- Aug 14, 2020 11:39 AM (3+ months ago)
Hello,

Can we have an update please?

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Aug 7, 2020 9:19 AM (3+ months ago)
Hi Shawn,

Did you get anywhere with the trace files? Can we have an update please?

Thanks for your help so far.

Roy

ODM Action Plan **Oracle Support**- Jul 20, 2020 6:47 PM (4+ months ago)

Hello Roy,

I am back. You asked about expectations... This is a generic set of network operations that we are trying to review. Using a set of crude ASCII art diagrams you have two basic scenarios:

(direct connect)
[[your application]] -> (firewall stuff) -> TCP stack -> local NIC (optionally with a trip to a DNS server) -> you local network -> the receiving NIC -> mysqld

(connection via router)

[[your app]] -> (firewall stuff) -> TCP stack
|
Router -> (firewall stuff) -> TCP stack -> local NIC ->.... the rest of the message path...

That trip from the application to Router could either hit the TCP loopback (127.0.0.1 or ::1) and avoid a trip to the external IP or it could hit the external IP then appear as an incoming message packet (the sender is the receiver). In which case it would look like this:

+----------+ via external IP address
v |
[[your app]] -> (firewall stuff) -> TCP stack ----+
|
(firewall stuff)
|
Router -> (firewall stuff) -> TCP stack -> local NIC ->.... the rest of the message path...

Or... you could be talking directly to your Router via a socket:

[[your app]] === /socket/ Router -> (firewall stuff) -> TCP stack -> local NIC ->.... the rest of the message path...

Each time you establish a TCP session/connection via Router, you are using at least 2 ephemeral ports and one fixed port: One from your app to the TCP stack, one fixed port by Router to listen on, and another from Router out to MySQL. Each time you close that session, Router will close its end of the client-server link to your mysqld. A direct connection would only require one ephemeral port per session.

If you are churning those connections quickly, the TIME_WAIT timer (built into TCP) may be blocking the ability of Router to have quick access to another ephemeral port. That is one problem I will be trying to identify (an exhaustion of the pool of ephemeral ports). You could resolve that by inserting a connection pool between your application and Router (usually, these are built into the client-side libraries). That would keep those sessions alive long enough to be reused by another application thread and avoid the TCP churn.

When you are forming the client-server connection from Router to your mysqld, I expect that to be as fast as it was for your non-Router communication chain.

I do not expect the local TCP connection from your app to Router to require a huge additional delay (a difference I would be looking for).

I do expect some time within Router to buffer the command and copy it into the other session to send to the server (that should be a relatively stable and small delay added to each command as that is pretty much a block data copy from one memory location to another).

I should be able to detect if you are using the loopback address or not to talk to Router.

I expect the same commands (I will be looking for equivalents) to be replied to by MySQL as quickly in either situation (directly from Router to MySQL or directly from your app to MySQL).

I will be looking for any handling delays introduced by buffering and copying the results from the Router return leg through to the internal application return leg. There are two parts to each result: the state of the operation (did it work? if not, which error did it hit?) and any data that the command may have generated.

If by way of timing analysis we can determine that the time spent within Router is the significant portion of the slowdown, then I clearly have something to show my developers. If the delays are mostly OUTSIDE of Router (based on the time it takes for a packet to enter router, be copied to the other buffer, then re-exit the other end of the pipe) then that's something environmental.

Please recognize: I am not a network specialist. I know what I shared just from the general knowledge I have picked up over the years. This is the same level of general knowledge that would permit me to explain how a plane physically flies but I am neither a pilot nor a certified mechanic. I will not be fast or efficient in this analysis because I am not familiar with the tools I will need to use. I have an expectation based on an understanding how the data flows within the system for which patterns of messages I should be looking for but it will be a trial-and-error attempt for me to find comparable situations in the two streams. If you have a local resource that can do this analysis in parallel, they will probably be much faster than I due to their expertise in this domain of knowledge.

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 20, 2020 4:35 PM (4+ months ago)

Hi Ligaya,

If Shawn is still out please request an available engineer to look at this.

Also, the issue is now urgent for us, so can you escalate the priority of the call?

Thanks,

Roy

ODM Action Plan **Oracle Support**- Jul 17, 2020 3:07 PM (4+ months ago)
Hello,

My name is Ligaya and I am currently tasked with ensuring MySQL issues are handled correctly.

Shawn will not be in today but is expected back on Monday. Due to the large amount of data on the issue and Shawn's deep knowledge of your problem, would it be acceptable to you to wait until Shawn is back?

If you need a reply before that, please let us know so we can try and find an available engineer to work on the issue. Please be aware that it may take time to find an available engineer to work on the issue.

Respectfully,
Ligaya

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 17, 2020 2:54 PM (4+ months ago)
Hi Shawn,

We will make a start on going through these two logs also.

Can you advise if there is a set config pattern, performance pattern, behaviour/latency that we should expect to see.

Thanks,

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 17, 2020 9:23 AM (4+ months ago)
Hi Shawn,

Can I ask you to look at just two logs for the time being.

moodle/net_dump_app02p is a trace without the router between app02 and the RO db's
moodle2/net_dump2_app02p is a trace with the router between app02 and the RO db's

Thanks,

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 16, 2020 10:12 AM (4+ months ago)
Hi Shawn,

The file upload completed on Friday, apologies for more arriving after I had emailed you. They were uploaded by the server admins and I didn't realise the load hadn't completed.

We performed traces for the scenarios requested and provided the names/ip addresses of the application servers and db servers so I wasn't expecting you to ask us to mine the logs and search for patterns as this was not mentioned in the request.

I will need to check if the logs still exist and if there is a team available to scan through them.

However I would appreciate it if you would take a look at the logs.

We have less than two week to decide if we can go ahead with the Innodb Cluster so this is now a priority for us.

Thanks,

Roy

ODM Action Plan **Oracle Support**- Jul 15, 2020 5:49 AM (4+ months ago)
Hello Roy,

Every time I thought you had finished, you attached another set of files. I'm still not sure you are done uploading data.

I also asked you to highlight what you found during your own analysis (to avoid duplicating any efforts). You never provided that summary. In particular, you know your application and its command patterns. I expected you (or your team or the admin that generated these captures) to have isolated and compared the same command sequences with and without router in the communication chain.

That is my starting point.

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region

Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 14, 2020 10:37 AM (4+ months ago)
Hi Shawn,

Have you had a chance to look at the tcp dumps sent last week?

Thanks,

Roy

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 10, 2020 9:47 AM (4+ months ago)
Upload to TDS successful for the file net_dump2_app10p_autorename1.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 10:01 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc02p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 9:17 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc02p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 9:03 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc02p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 9:03 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc02p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 8:42 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc03p.partad.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 8:25 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc03p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 8:22 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc03p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 8:03 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc02p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 7:27 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc03p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 7:24 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc03p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 7:08 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc01p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:46 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app04p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:45 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app08p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:41 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app10p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:27 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_dbc03p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:24 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc03p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:22 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc02p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:22 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app07p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:18 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app06p.partac.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 9, 2020 6:18 PM (4+ months ago)
hi Shawn,

The output uploaded is from a tcpdump from each application server connection to the cluster.
There are 10 files per test. The first 10 files are from the 10 application servers that the router is installed on. These app servers are connecting to 3 db servers in the cluster.
The ip addresses of the 10 application servers are

mdl-app01p.ad.ucl.ac.uk 10.28.80.171
mdl-app02p.ad.ucl.ac.uk 10.36.80.175
mdl-app03p.ad.ucl.ac.uk 10.28.80.172
mdl-app04p.ad.ucl.ac.uk 10.36.80.174
mdl-app05p.ad.ucl.ac.uk 10.28.80.185
mdl-app06p.ad.ucl.ac.uk 10.36.80.186
mdl-app07p.ad.ucl.ac.uk 10.28.80.189

mdl-app08p.ad.ucl.ac.uk 10.36.80.190
mdl-app09p.ad.ucl.ac.uk 10.28.80.191
mdl-app10p.ad.ucl.ac.uk 10.36.80.192

The DB Servers and ip addresses are

mdl-dbc01p.ad.ucl.ac.uk 10.29.80.112
mdl-dbc02p.ad.ucl.ac.uk 10.37.80.113
mdl-dbc03p.ad.ucl.ac.uk 10.29.80.114

There is no sensitive data on theses files. Unfortunately we cannot re-upload as the files have been deleted from the servers.

Thanks,

Roy

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:18 PM (4+ months ago)
Upload to TDS successful for the file net_dump_dbc01p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:16 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app05p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:07 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app02p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:01 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app07p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:01 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app10p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:00 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app06p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 6:00 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app08p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app04p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app02p.partab.

ODM Action Plan **Oracle Support**- Jul 9, 2020 5:45 PM (4+ months ago)
Hello Roy,

I tried to slow you down while you were uploading data. You must have missed my update.

What kinds of captures are these? What tool should I use to look at them? Are these parts of compressed files? How should I assemble them?

What did your analysis reveal? Where should I look to see what you saw? Please, at least guide me on how to replicate your analysis of the data you collected.

Why are there so many different series of file names? Why was just one of them not enough to prove your case?

Which address:port combinations equat to which nodes in the communication streams you captured?

I have not tried to look at your data, yet, because if that data contains HIPAA, electronic health records, or other forms of sensitive information (like payment details), you did not tag your uploads appropriately by uploading them to the proper subfolder (the "Uploading PI/ePHI Data" tab of that same document you were following has those instructions). I would need to ask you to remove your data and re-upload it with the proper tagging. Does this data contain sensitive information (like health records or payment details)?

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:27 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app03p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:19 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app01p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:18 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app09p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:01 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app09p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:01 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app07p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:00 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app10p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:00 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app06p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 5:00 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app08p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 4:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app03p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 4:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app05p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 4:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app04p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 4:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app02p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 9, 2020 4:57 PM (4+ months ago)
Upload to TDS successful for the file net_dump2_app01p.partaa.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 9, 2020 12:04 PM (4+ months ago)
Hi,

Please see attached log files for load test without Mysql Router.

We will upload the log files for loads using Mysql Router shortly.

Roy

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 11:46 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p_autorename1.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 11:31 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p_autorename1.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 11:02 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app06p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 10:30 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p_autorename1.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 10:03 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app09p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:59 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app03p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:53 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app08p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:52 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app04p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:38 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app04p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:38 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app08p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:37 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app03p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 9:36 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app09p.partab.

ODM Action Plan **Oracle Support**- Jul 8, 2020 8:51 PM (4+ months ago)
Hello Roy,

At first I thought you were sending me a single capture in 10 parts but you still had some parts to upload (3,4,6, and 9) now I see that you are sending in multiple streams in multiple parts lettered aa-ac (so far).

You will need to point me to the specific locations you identified in your own analysis about where to look, which systems sit where in the topology, and explain why you need ALL of these streams to demonstrate the issue.

How many parts are there to each stream?

Regards,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:44 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app07p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:43 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app05p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:37 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app04p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:37 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app06p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:36 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app03p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:35 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app09p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:24 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:10 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app02p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:10 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:09 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app07p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:09 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app05p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:08 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app01p.partac.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:08 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app02p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 8:07 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app01p.partab.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:10 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app10p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:10 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app07p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:09 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app08p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:09 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app05p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:08 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app02p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 7:07 PM (4+ months ago)
Upload to TDS successful for the file net_dump_app01p.partaa.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 5:37 PM (4+ months ago)
Upload to TDS successful for the file testfileupload_autorename1.

Update from Customer **B.WATTS@UCL.AC.UK**- Jul 8, 2020 4:59 PM (4+ months ago)
Upload to TDS successful for the file testfileupload.

Notes **Oracle Support**- Jul 6, 2020 12:38 PM (4+ months ago)
Hello roy -

Thanks for your update.
From the previous doc please check the "Uploading Very Large Files" tab.

Thank you very much.
Best Regards,
Antonio Correia (on Behalf of SHAWN)
MySQL Support, EMEA

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 6, 2020 12:20 PM (4+ months ago)
Hi Antonio,

I dont see any detail on how to use FTPS & HTTPS to MOS File Upload service.

Can you advise please?

Roy

Notes **Oracle Support**- Jul 6, 2020 10:59 AM (4+ months ago)
Hello roy -

Please review the next doc:

How to Upload Files to Oracle Support (Doc ID 1547088.2)

Thank you very much.
Best Regards,
Antonio Correia (on Behalf of SHAWN)
MySQL Support, EMEA

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jul 6, 2020 10:34 AM (4+ months ago)
Hi Shawn,

We are collating the network traces. It looks as though these files (unzipped) will exceed be between 10G - 14G. Do you have an ftp site that I can upload these files to?

Thanks,

Roy

ODM Action Plan **Oracle Support**- Jun 26, 2020 5:34 PM (5+ months ago)
Hello Roy,

Thank you for the New Relic documentation. It helps me to put your numbers into their proper context.

I'm not the only engineer watching this service request. One of the others (who works the same hours as the developers who maintain Router) has shared our minimal progress with them. They have requested a tcp dump (or some other kind of network trace) to be able to track the timings of a page's processing trips through Router (both outbound and inbound from both sides: application and database server) and compare those to the same trips between the application and the database without Router. A common tool for this on many platforms is Wireshark but there are other tools available to capture this kind of packet-level information.

Please work with your admins or networking people to try to capture a representation of the MySQL-based network traffic of the same page with and without Router in the mix. I'm hoping that you have an isolated test system that you could execute some read-only operations from against your live production data to collect this timing info. That way, the system tracing the calls through Router won't be flooded with the full set of production queries (which could make it harder to find similar cases).

If a test web server talking to a test database shows the same delays, that would be ideal.

One theory under consideration is that Router may be slower because it is having issues actually talking to some of the nodes so those connection attempts are timing out. Router then switches to another node (times out again) then finally finds a node it can talk to. The timings and exchanges in the trace would show us that in operation (if that is the situation). If it is some other situation, those traces should guide us to somewhere else to look.

Thank you,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 26, 2020 3:04 PM (5+ months ago)
Hi Shawn,

You are correct in your understanding of how the ports have been confugired.

This may answer your question of how the NewRelic monitor records the metrics we sent .

https://docs.newrelic.com/docs/understand-dependencies/distributed-tracing/get-started/how-new-relic-distributed-tracing-works

Roy

ODM Action Plan **Oracle Support**- Jun 24, 2020 6:47 PM (5+ months ago)
Hi Roy,

I will admit, I am weak in PHP. But I noticed that your properties specify two different ports. One in an array then again within the array for an instance.

(with indention)
```
$CFG->dboptions = array(
'dbpersist' => false,
'dbsocket' => false,
'dbport' => '6446',
'dbhandlesoptions' => false,
'dbcollation' => 'utf8mb4_unicode_ci',
'readonly' => [
'instance' => [
'dbhost' => '127.0.0.1',
'connecttimeout' => '5',
'latency' => '2',
'dbport' => '6447',
],
]
);
```

I interpret that as a set of defaults (for a default connection -a read-write connection-) that points to 127.0.0.1:6446 and an override that points to 127.0.0.1:6447 for read-only connections. If I'm reading that properly, this looks ideal for a Router. Did I get that correct?

So, those would be the two ports I worry about getting scanned. Most intrusion scanners ignore loopback address messages, some do not.

Did you make any headway determining how your charting system determined when to stop the timer on the PHP area and start the timer for the MySQL area? I can imagine something like this....

<enter your PHP page> start the PHP timer

<send a command to MySQL> start the MySQL timer.
<receive the result> stop the MySQL timer
... repeat the last two steps for every command the page needs to execute ...

<complete the PHP page> stop the PHP timer

In that scenario, the MySQL values would change by only a bit but the overall timer (for the entire page) would keep ticking.

====

If I think through this using the numbers from /mod/forum/view.php .

* Your non-Router average time for that page was only 912ms. Your through-Router average time was about 12.5s

* The rough-estimate average displacement of the two MySQL response charts was about 150ms (max).

If I presume we get the same average increase per call (150ms), then with the total time difference of (12500-912=) 11588 ms we can generate an estimate of how many calls that page performs.

In order for a 150ms/call change in response time to add up to 11588ms, you would need that page to do a smidge more than 77 database calls per invocation. That's a lower bound because my divisor was a max value. If the actual time differential were smaller than 150ms, the value would be higher than 77.

Is that a reasonable estimates of how many MySQL commands that page (/mod/forum/view.php) would usually execute per invocation? I'm asking as a sanity check.

However, if we had 77 calls per page without the router, then each call would need at most 912/77= 11.8ms. That is WAY lower than the average time per call in the "no Router" chart you showed me. The local minima in the cycle was only down to about 85ms. So, I think the sanity check may be failing.

====

I chose that page because your monitoring system gave that page the most weight (it thought this page was the most impactful on your system). So, I presumed it would have a dominating effect on the charts.

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 24, 2020 4:26 PM (5+ months ago)

Hi Shawn,

In response to (1) here is the relevant section from the application config file, I will come back to you re. (2)

```
<?php
unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype = 'mysqli';
$CFG->dblibrary = 'native';
$CFG->dbhost = '127.0.0.1';
$CFG->dbname = '';
$CFG->dbuser = '';
$CFG->dbpass = '';
$CFG->prefix = '';
$CFG->dboptions = array(
'dbpersist' => false,
'dbsocket' => false,
'dbport' => '6446',
'dbhandlesoptions' => false,
'dbcollation' => 'utf8mb4_unicode_ci',
'readonly' => [
'instance' => [
'dbhost' => '127.0.0.1',
'connecttimeout' => '5',
'latency' => '2',
'dbport' => '6447',
],
]
```

);

ODM Action Plan — **Oracle Support**- Jun 23, 2020 5:03 PM (5+ months ago)

Hello Roy,

Thank you for the focused charts.

While the periodic cycling (+-30% at worst around a visual average) about every 3 minutes is interesting, I'm focusing on the peaks.

In the direct-connect chart, your longest response times were around 125ms. The local minima were around 85ms.

In the through-Router chart, your peak response time was around 275ms, the local maxima (in the cycles) were roughly 210-250 increasing over time. The local minima were variable between 160 and 195ms.

This, by itself, does not explain the change in application performance from sub-second to 10+ seconds per page.

(example) For the page /mod/forum/view.php
/* without */
mean: 912ms - samples: 2.2s 2.2s 2.1s

/* with Router */
mean: 12.5s - samples: 43s 41.8s 41.5s

I think we need to look in two places:

1) the operating system

The Router may operate on the same port as the server (3306) or a different port. Either way, there could be applications trying to scan/sanitize/block unwanted traffic through the local router port (or socket). Do you have any antivirus, antimalware, port scanners, or other processes operating on this host that would interject itself between your application and your Router or between the Router and its server?

2) Your connection properties

This feels like a longshot but there may be some slight difference in how you tell your PHP library to connect to the backend Router (so that it can proxy your requests to the cluster you want to use) that is creating a conflict in the exchange of information between those two layers. Can you post how you connect from your application in both situations? (you should blot out or hide any user names or passwords. I'm only interested in the other settings)

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Update from Customer — **R.BHURTHA@UCL.AC.UK**- Jun 23, 2020 4:25 PM (5+ months ago)

Upload to TDS successful for the file router-slavevms-500threads-mysqlonly.png.

Update from Customer — **R.BHURTHA@UCL.AC.UK**- Jun 23, 2020 4:25 PM (5+ months ago)

Upload to TDS successful for the file slavevms-500threads-mysqlonly.png.

ODM Action Plan — **Oracle Support**- Jun 23, 2020 3:36 PM (5+ months ago)

Hello Roy,

Those are very interesting charts. After looking at the first chart, I expected the second to be dominated by the deep yellow of the MySQL color. It wasn't. The second chart is being overwhelmed by the blue area (color coded as PHP).

###################

(a rough ASCII approximation for anyone else who finds this SR later - not to scale)

These are constantly varying stacked line charts

Legend: P = PHP area, M = MySQL area, R = Redis area.

/* without router */
|---- P-----|---M----|--R--| (250ms/110ms/80ms rough visual average over the majority of the chart)

/* with router */
|---------- P----------|--M---|R| (4s/200ms/?ms the Redis line is too thin to estimate)

###################

Do you know how this chart derives its numbers? How does it tell what is "PHP" time and what is "MySQL" time? Where (within the message passing or processing layers) does the breakout occur? (In other words, where does one timer pause while the other starts measuring)

Is it possible for you to re-generate those same graphs for the same times but ONLY include the MySQL portion (the deeper yellow strip between the PHP and the Redis areas)?

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 23, 2020 10:06 AM (5+ months ago)
Hi Shawn,

There was no difference in connection pooling with/without the router.

I have attached metrics taken from the application server that shows the difference in performance end-users are experiencing.

You can see an increase in response time, for example, from 912ms to 12.5 secs.

Thanks,

Roy

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 23, 2020 10:02 AM (5+ months ago)
Upload to TDS successful for the file router-slavevms-500threads.png.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 23, 2020 10:01 AM (5+ months ago)
Upload to TDS successful for the file slavevms-500threads.png.

ODM Action Plan **Oracle Support**- Jun 22, 2020 7:45 PM (5+ months ago)
Hello Roy,

The analysis changes only slightly with multiple application servers and multiple cores. I'll work through the math to see if this still appears sane.

//workload processed by the server - from the last calculation //

(without router)
14M+15+80+300K = about 14.3M/sec or 6.99e-08 seconds/operation

(with router)
2M+2+30+40K = about 2.04M/sec or 4.90e-07 seconds/operation

Subtracting the fast from the slow gives me about:
(1/2040000)-(1/14300000)=4.203e-07 seconds/operation (as an average difference)

I'll presume (for this run through) that you have 6 application servers, each with 6 cores:

[workload per app server]

Presuming you have 6 app servers, each one generates 1/6 of the average load:

(with router) 14.3M/6 = 2383333 operations/machine-sec = 4.196e-07 machine-seconds/operation

(without router) 2.04M/6 = 340000 op/machine-sec = 2.941e-06 machine-sec/op

Unit math: (operations/sec) / (machines) = operations/(machine * sec) or ops/machine-sec

This means, on average, each operation now takes
(6/2.04M) - (6/14.3M) = 2.941e-06 - 4.196e-07 = 2.522e-06 machine-seconds/op longer.

Again, presuming a clock rate of 2.5GHz and 6 cores/machine

(2.522e-06 machine-sec/op) * (6 cores/machine) = 1.513e-05 core-sec/operation

That is still only 15microseconds.

In clock cycles, that works out to 1.513e-05 * 2.5e09 = 37824 additional clock cycle for each core (which includes calls into the kernel for memory allocations and network support).

Units: (core-sec/op) * (cycles/sec) = core-cycles/op

So even in this situation, that could be reasonable. A lot of it could depend on how much payload your average query actually returns (your workload is SELECT-dominated).

==

When you routed your connections via Router, did you change anything else about how your application connected? For example, were your direct connections using a connection pool but your Router connections were not?

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 22, 2020 6:22 PM (5+ months ago)
Upload to TDS successful for the file cpu_info.txt.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 22, 2020 6:19 PM (5+ months ago)
Hi Shaw,

Thanks for the detailed analysis.

To add to the notes, when the end users tested with the router, the application was timing out on page loads and hanging on some queries.

I have attached CPU details for the application servers and DB servers.

Thanks,

Roy

ODM Action Plan **Oracle Support**- Jun 22, 2020 5:52 PM (5+ months ago)
Hello Roy,

Thank you for the numbers.

"Noticeable performance drops are Network DB throughput (14M/sec to 2M/sec), DML per second (inserts dropped from 15/sec to 2/sec, updated 80/sec to 30/sec), Row writes (300k/sec to 40k/sec)"

I'm presuming that the cost of executing each command at the server and the time spent transiting the network both remain constant meaning the only difference in time should be the extra time spent handling the commands and their results within Router.

(without router)
14M+15+80+300K = about 14.3M/sec or 6.99e-08 seconds/operation

(with router)
2M+2+30+40K = about 2.04M/sec or 4.90e-07 seconds/operation

Subtracting the fast from the slow gives me about:
(1/2040000)-(1/14300000)=4.203e-07 seconds/operation (as an average difference)

Yes, it's an average multiple of about 7:1 (roughly) but the incremental increase in per-operation duration is only 4.203e-07 seconds. That's four-tenths of a microsecond or about 420 nanoseconds

If (for the sake of simplicity) you had one core and your per-core clock speed is about 2.5GHz and if your core is running at 100% of its rated clock. This translates to about 1051 extra clock cycles/command.
(0.000000420 sec/command)*(2500000000 cycles/sec) = 1051 cycles/command

That increase in operation duration seems reasonable for a proxying operation. Router needs to receive the command, buffer it, identify or open a connection, send the command, receive the result, and pass that back to your calling application.

I double-checked my figures but my math appears to be sound. I'm only trying to turn the additional time spent for each operation into a value of physical units (how many more clock cycles is each command taking with Router in the picture?). Those are limited within any computer.

The same basic calculation process can be modified to a multicore system. How many cores are in this machine?

If you spot an error in my math, please let me know.

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Update from Customer **R.BHURTHA@UCL.AC.UK**- Jun 22, 2020 10:48 AM (5+ months ago)
Hi Shawn,

We have a three-node cluster.

The router is installed on all the application servers.

The communication link is using tcp ports 6446 (r/w) and 6447 (ro)

The stats we have inspected are from MySQL Enterprise Monitor.
Noticeable performance drops are Network DB throughput (14M/sec to 2M/sec), DML per second (inserts dropped from 15/sec to 2/sec, updated 80/sec to 30/sec), Row writes (300k/sec to 40k/sec)

Thanks,

Roy

ODM Action Plan **Oracle Support**- Jun 19, 2020 5:40 PM (5+ months ago)
Hello Roy,

Please validate this crude ASCII art diagram....

A
(application) ----------> [[InnoDB Cluster]]
\ /
\ /
---> <Router> ---
B

When your connections are directly from your application to the Primary of the InnoDB Cluster (route A) , you get better performance than if you bounce your database connections through the proxy (route B) that is MySQL Router.

You say that your throughput is "significantly lower" - even that is relative and it depends on a lot of factors. On a per-call basis, what is the timing difference? (I'm trying to quantify the slowdown in an effort to direct my focus to the correct part of the issue).

(some additional background questions to help me get started)

What statistics have you collected? (to avoid me asking for info you may already own)

Where did you install your Router? (it should be on the same host as your application)

What kind of communication link exists between each of those three nodes in the diagram?

Yours,
--
Shawn Green
MySQL Product Support Manager, AMER region
Oracle USA, Inc. - Hardware and Software, Engineered to Work Together.
Office: Blountville, TN

Become certified in MySQL! Visit https://www.mysql.com/certification/ for details.

Customer Problem Description **R.BHURTHA@UCL.AC.UK**- Jun 19, 2020 4:28 PM (5+ months ago)
Customer Problem Description

Problem Summary
---
mysql router performance issues

Problem Description
---
hello,

We are running a 3 node innodb cluster using mysql router.

Performance is significantly slower using the router than when we do not use it.

What can be done to improve the performance of the router?

Thanks,

Roy

Error Codes

Problem Category/Subcategory

MySQL Router Installation and Configuration

Uploaded Files
