

Azure IR Connectivity issue with DW intermittently

Last updated by | Veena Pachauri | Mar 9, 2023 at 12:00 AM PST

Contents

- [Issue](#)
- [Suggestion](#)
- [More detail](#)
- [Server side investigation:](#)
 - [SQL error code:](#)
 - [Possible root cause from server side:](#)
 - [Correlation information:](#)

Issue

Azure IR Connectivity issue with DW intermittently

[{"Code":22301,"Message":"Failure happened on 'Source' side.
ErrorCode=SqlOperationFailed;Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Message=A database operation failed with the following error: 'A transport-level error has occurred when receiving results from the server. (provider: TCP Provider, error: 0 - An existing connection was forcibly closed by the remote host.);Source=,'Type=System.Data.SqlClient.SqlException] }

Failure happened on 'Source' side. ErrorCode=UserErrorFailedFileOperation, Errors=[{Class=20,Number=121,State=0,Message=A transport-level error has occurred when receiving results from the server. (provider: TCP Provider, error: 0 - The semaphore timeout period has expired.)}]

Suggestion

Before you start to do investigation, please check the global wide to see any similar issue, whether exceeding the SLA, if yes, we need to engage the Azure Network Team to help us as it may relate to Outage.

```
ExecutionApiCall
| where TIMESTAMP > ago(5d)
//| where SubscriptionId contains "XXXX"
//| where ApiName == "GetState"
| where Status !in ("Canceled", "InProgress")
//and ExecutionType == "SqlServerStoredProcedure"
and (IntegrationRuntime contains "DefaultIntegrationRuntime" or IntegrationRuntime contains "AzureIR" )
| extend succeeded = iff(Status == "Succeeded", 1, 0)
//| extend failed = iff(Status == "Failed", 1, 0)
| extend failed_tcp = iff(Status == "Failed" and ErrorMessage contains "A connection was successfully established with the server", 1, 0)
| summarize sum(failed_tcp), sum(succeeded) by datetime_part("Day",TIMESTAMP)
| order by sum_failed_tcp desc nulls last
```

Column1	sum_failed_tcp	sum_succeeded	
13	83	65303029	
12	76	65031863	
14	40	63393115	
15	7	59404355	
11	5	63937384	
16	0	2613030	

If not, please confirm customer's SLA per Subscription Level.

```

ExecutionApiCall
//| where TIMESTAMP > ago(2d)
| where SubscriptionId contains "XXXXXX"
//| where ApiName == "GetState"
and Status !in ("Canceled", "InProgress")
//and ExecutionType == "SqlServerStoredProcedure"
and (IntegrationRuntime contains "DefaultIntegrationRuntime" or IntegrationRuntime contains "AzureIR" )
| extend succeeded = iff(Status == "Succeeded", 1, 0)
//| extend failed = iff(Status == "Failed", 1, 0)
| extend failed_tcp = iff(Status == "Failed" and ErrorMessage contains "A connection was successfully established with the server", 1, 0)
| summarize sum(failed), sum(failed_tcp), sum(succeeded) by DataFactoryName, PipelineName
| order by sum_failed_tcp desc nulls last

```

We guarantee that at least 99.9% of the time, all activity runs will initiate within 4 minutes of their scheduled execution times.

https://azure.microsoft.com/en-us/support/legal/sla/data-factory/v1_2/ 

If the SLA meet, we should explain to customer RCA might not be found, and set customer expectation with retry and timeout setting as workaround, however, we still need to follow the steps below to move forward to help our customer as much as possible.

Verify Azure IR performance

Sometimes if the Windows OS performance is poor, like high CPU, it will cause TCP connection not getting established. Hence we could start from ADF side to review Azure IR node performance during issue time range to ensure the TCP disconnect is not caused by any Azure IR performance issue.

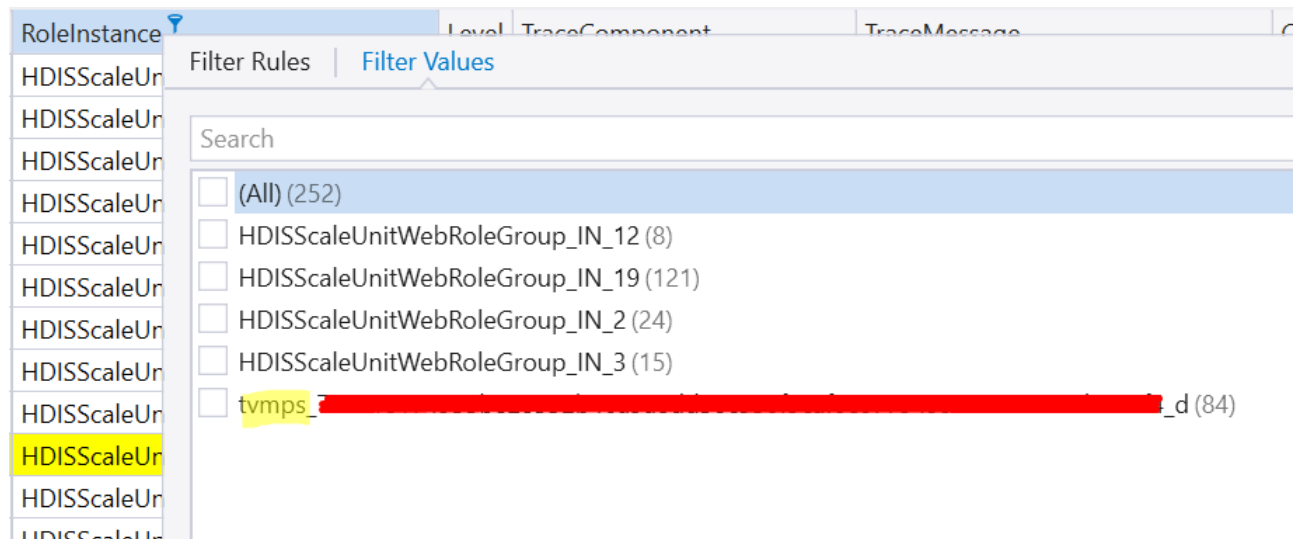
1. Get the role Instance name from CustomLogEvent table result.

```

CustomLogEvent
| where * contains "<activity_run_ID>"
| project TIMESTAMP, Role, RoleInstance, Level, TraceComponent, TraceMessage, trimlog(Message)
| order by TIMESTAMP asc

```

You could get the role instance name of Azure IR node from "RoleInstance" column



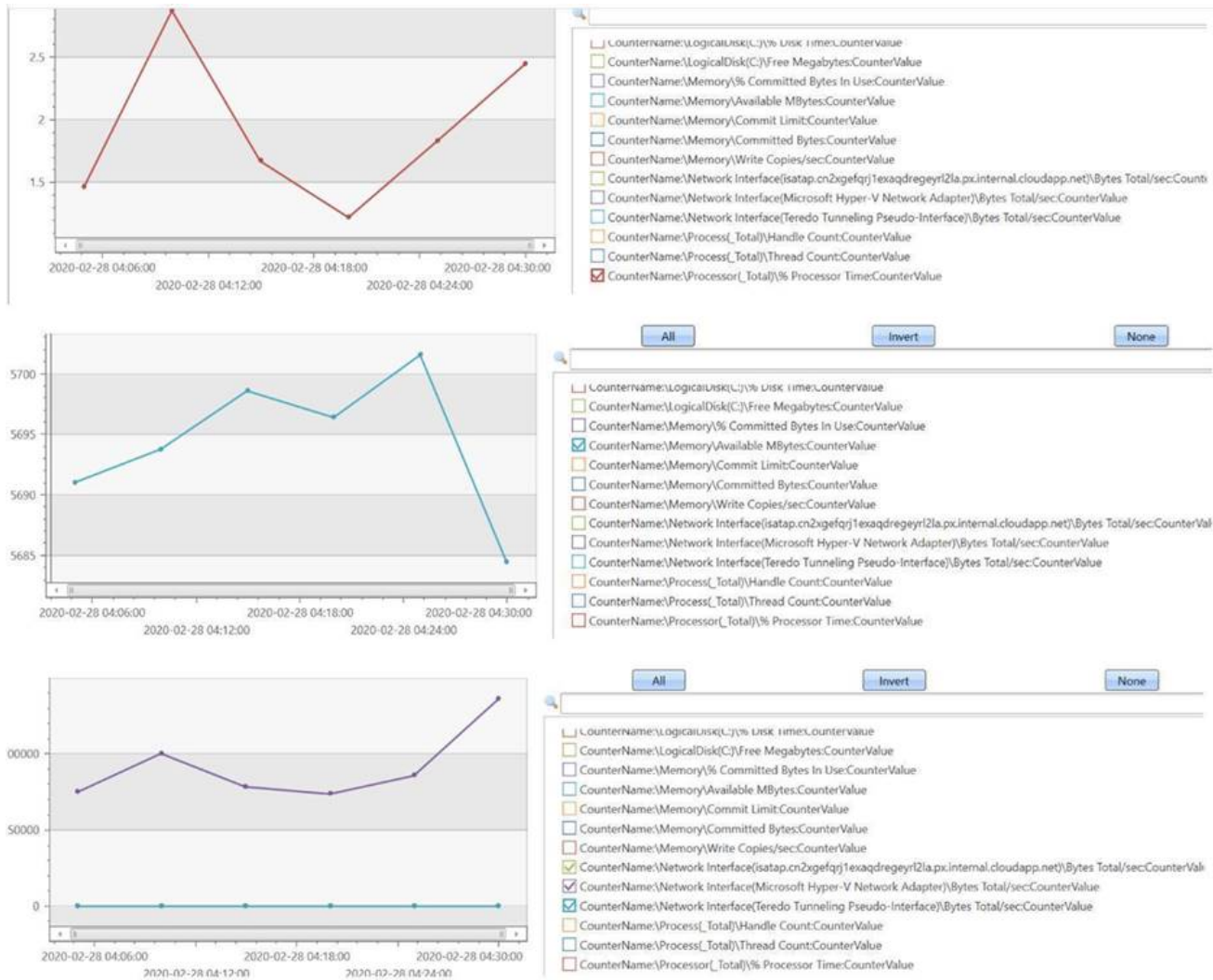
2. Review performance on that Azure IR instance during issue time frame.

```

CounterFiveMinuteNodeRollup
| where RoleInstance == "tvmps_XXXXXXX"
| where TIMESTAMP > datetime(...start_time) and TIMESTAMP < datetime(...end_time)
| project TIMESTAMP, CounterName, CounterValue
| render timechart

```

You could review below performance counters for CPU/Memory/Network.



If the performance is fine, please follow the steps below:

- 1, Please engage the SQL Engineer to work with you and confirm any issue from DW side.
- 2, If nothing wrong with DW perf or others, please confirm DW Connection policy per [Connection policy](#).
- 3, If it is the proxy mode, you need to share the gateway IP ([how to get DW gateway IP](#)) to DW PG/CSS to check any block there.
- 4, If it is Redirect mode, we need to share the Azure IR IP range and VM name to CSS Azure Network Team to figure out more detail, however, still need DW's team collaboration.

Kusto table CustomLogEvent has a column RoleInstance which contains VM name.

Examples, Azure IR (Azure Batch) - tvmps_XXXXXXXXXXXXXXXXXXXX9501cb915_d WebService VM - HDISXXXXXXXXXXXXXXXXXXXX_IN_52

- 5, In case of above suggestions could not help you, you need to build the Self-IR with dedicated IR at the same region as Azure IR to repro with netmon trace enabled 24*7 until the problematic time by collaborating with Azure Network Team. If could, engaging the DW PG see the possibility to enable the same network trace on their side.

On the other side, as confirmed with SQL DW PG, direct copy is pretty fragile and not the recommended best practice for egressing large amount of data from DW. We should try to get the customer to move to CETAS / Polybase approach.

More detail

[Troubleshoot Azure Data Factory connectors - Azure Data Factory | Microsoft Docs](#)

Cause analysis

If the error message contains the string "SQLException", SQL Database throws an error indicating some specific operation failed.

Recommendation

If the SQL error is not clear, try to alter the database to the latest compatibility level '150'. It can throw the latest version SQL errors. For more information, see the [documentation](#). For more information about troubleshooting SQL issues, search by SQL error code in [Database engine errors](#). For further help, [contact Azure SQL support](#).

Actually, we have public doc above to explain that we need to work with SQL Support team to move forward.


Server side investigation:**SQL error code:**

You can check below wiki to understand the SQL error code:

https://supportability.visualstudio.com/AzureSynapseAnalytics/_wiki/wikis/AzureSynapseAnalytics/442317/SQL-Error-Codes

Possible root cause from server side:

You can work with synapse team by checking below possible root cause:

Possible cause	Detailed cause	Kusto(SQL)	Related wiki/TSG
Database is in pause status	Pause and resume	dw_engine_restart_version_slo("appname") Monlogin	
	maintenance	Monmanagement	https://supportability.visualstudio.com/AzureSynapseAnalytics/_v/Unavailable-Scheduled-Maintenance
	scale	Monmanagement dw_engine_restart_version_slo("appname") dw_engine_dms_topology_changes('@appname')	
	failover	dw_engine_dms_topology_changes('@appname') WinFabLogs	
	Other connection errors	Monlogin; MonSQLSystemHealth; MonDwEngineLogs;	Azure Synapse Analytics: https://supportability.visualstudio.com/AzureSynapseAnalytics/_v/Troubleshoot-Availability-and-Connection-Errors AzureSQLDB: https://supportability.visualstudio.com/AzureSQLDB/Troubleshoot-DB-Availability-and-Connection-Errors
Gateway node issue	Gateway restart	Monlogin; MonDwUserSessionConnections; Service Fabric	DWCN_TSG0016: A transport-level error has occurred when send
Throttling	Maximum Concurrent open sessions: 1024	Monlogin; MonDwEngineLogs	Doc: https://docs.microsoft.com/en-us/azure/synapse-analytics/s-capacity-limits#workload-management 

Correlation information:

You can use client connection id of the pipeline activity to check the session id from synapse side.

```
CustomLogEvent
| where * contains "<activity_run_ID>"
| project TIMESTAMP,Role,RoleInstance,Level,TraceComponent,TraceMessage,trimlog(Message)
| order by TIMESTAMP asc
```

TraceMessage	JobId	Message
ExecutionRuntimeReportInProgress	2c3a99df-bd3a-4314-9a18-68405bc603ec	<LogProperties> <Text>Activity run picked up, details: [] </Text> </LogProperties>
ExecutionRuntimeReportInProgress	2c3a99df-bd3a-4314-9a18-68405bc603ec	<LogProperties> <Text>Pickup info is not posted correctly from client </Text> </LogProperties>
ExecutionRuntimeCommon	2c3a99df-bd3a-4314-9a18-68405bc603ec	<LogProperties> <Text>ExecutionControllerBase.GetExecutor - Type:SqlServerStoredProcedure.</Text> </LogProperties>
ExecutionRuntimeReportFinish	2c3a99df-bd3a-4314-9a18-68405bc603ec	<LogProperties> <Text>ExecutionController.ReportState - Finished.</Text> </LogProperties>
SqlConnectionOpen	b9cf7298-4a7a-46fb-94c9-00dc07e4b332	<LogProperties> <Text>Client Connection ID: ebf57c8-98c9-460d-a302-d5ee1cbd7085 </Text> </LogProperties>
UpdateQueryJobInProgress	00000000-0000-0000-0000-000000000000	<LogProperties> <Text>Connected database. Begin execute query.</Text> </LogProperties>
ExecutionSprocExecutor	2c3a99df-bd3a-4314-9a18-68405bc603ec	<LogProperties> <Text>Hit the exception...</Text> </LogProperties>
SqlConnectionClosed	b9cf7298-4a7a-46fb-94c9-00dc07e4b332	<LogProperties> <Text>Client Connection ID: ebf57c8-98c9-460d-a302-d5ee1cbd7085 </Text> </LogProperties>

In DW (SqlAzure) cluster.

Step a. Try to find the client connection in MonLogin using the connection id.

```
MonLogin
| where connection_peer_id =~ "ebf57c8-98c9-460d-a302-d5ee1cbd7085"
| project PreciseTimeStamp, logical_server_name, database_name, AppName, connection_id, connection_peer_id
```

PreciseTimeStamp	logical_server_name	database_name	AppName	connection_id	connection_peer_id
2020-11-11 21:56:18.6407094	symphonyproddwhserver	dwhAzrSympProd	cd95baea1028--0	24F710EB-A99D-421F-B228-E0A1A95676F1	EBF57C8-98C9-460D-A302-D5EE1CBD7085
2020-11-11 21:56:18.9375433	symphonyproddwhserver	dwhAzrSympProd	Worker	FAD41607-3DCC-4ED7-8954-F68922C57171	EBF57C8-98C9-460D-A302-D5EE1CBD7085

Step b. Try to find the session id in MonDwUserSessionConnections using the connection_id we retrieved from MonLogin

```
MonDwUserSessionConnections
| where AppName has "cd95baea1028" and ClientAppName != ''
| where TIMESTAMP between (datetime(2020-11-11 21:50:00)..datetime(2020-11-11 22:12:00))
| where ConnectionId in ("24F710EB-A99D-421F-B228-E0A1A95676F1", "FAD41607-3DCC-4ED7-8954-F68922C57171")
| project PreciseTimeStamp, LogicalServerName, AppName, SessionId, ConnectionId
```

PreciseTimeStamp	LogicalServerName	AppName	SessionId	ConnectionId
2020-11-11 21:56:12.0937515	symphonyproddwhserver	cd95baea1028--0	SID447463	24F710EB-A99D-421F-B228-E0A1A95676F1
2020-11-11 21:56:12.0937515	symphonyproddwhserver	cd95baea1028--0	SID447463	24F710EB-A99D-421F-B228-E0A1A95676F1
2020-11-11 21:59:14.1899298	symphonyproddwhserver	cd95baea1028--0	SID447463	24F710EB-A99D-421F-B228-E0A1A95676F1

How good have you found this content?

