HighLatencyLogin

Last updated by | Holger Linke | Mar 8, 2023 at 12:20 AM PST

Contents

Issue

The customer is consistently unable to login to Azure SQL Database. The error messages reported on the customer's side could be 40613 "not currently available" or other, connectivity-related messages.

Investigation / Analysis

The symptoms on the telemetry show as "HighLatencyLogin" in ASC, with `OutageReasonLevel1` as "Unplanned" and `OutageReasonLevel2` as "HighLatencyLogin". The reason appears to be related to blocking on LCK_M_SCH_S locks.

ASC

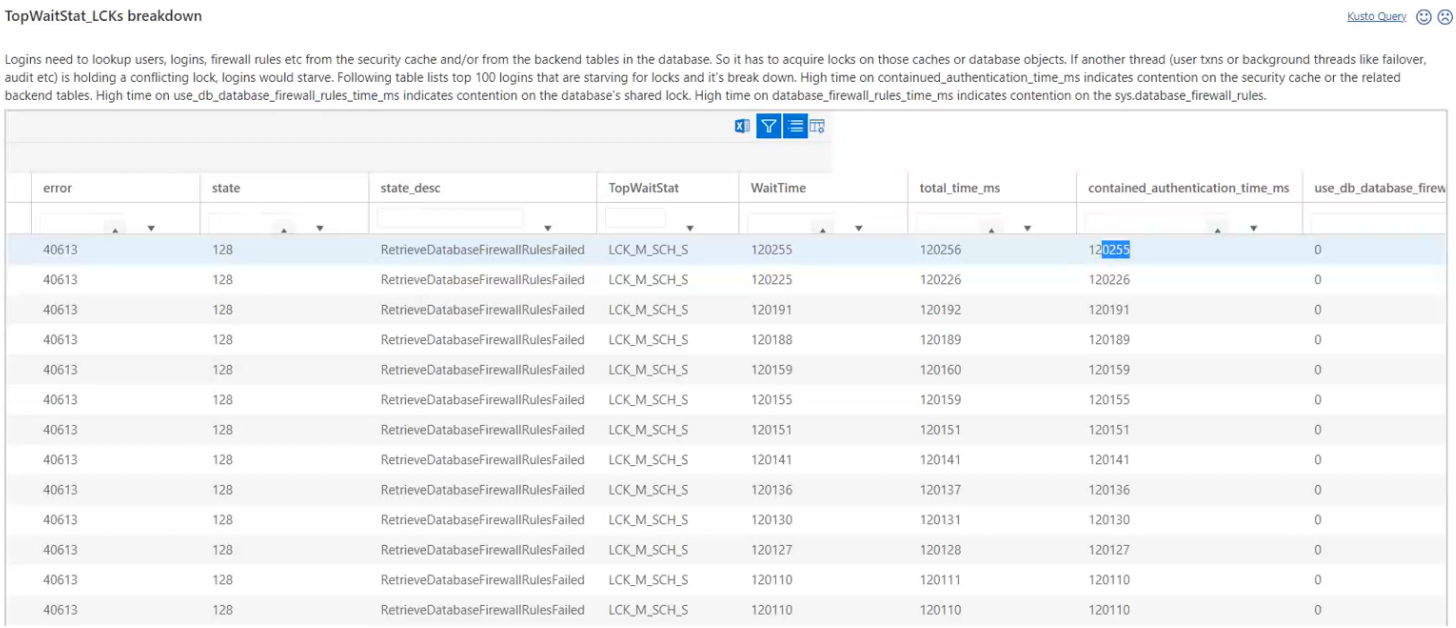
Login Outage details

In **ASC -> Downtime Reasons -> Login Outages**, you will see OutageReasonLevel1 as "Unplanned", OutageReasonLevel2 as "HighLatencyLogin" and OutageReasonLevel3 as "LCK_M_S" or "LCK_M_SCH_M" or "LCK_M_SCH_S".

2/10/2021 10:47:12 PM	2/10/2021 10:47:27 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:59.635000
2/10/2021 10:48:42 PM	2/10/2021 10:49:27 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:59.635000
2/10/2021 10:51:12 PM	2/10/2021 10:51:27 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:59.635000
2/10/2021 10:52:12 PM	2/10/2021 10:53:57 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.146000
2/10/2021 10:54:42 PM	2/10/2021 10:55:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 10:56:13 PM	2/10/2021 10:56:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 10:57:13 PM	2/10/2021 10:57:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 10:58:43 PM	2/10/2021 10:59:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 11:01:13 PM	2/10/2021 11:01:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	LongLogins	login_time_ms	{"package":"sqlserver","No
2/10/2021 11:02:13 PM	2/10/2021 11:02:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	LongLogins	login_time_ms	{"package":"sqlserver","No
2/10/2021 11:03:13 PM	2/10/2021 11:03:58 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:02:00.013000
2/10/2021 11:05:13 PM	2/10/2021 11:05:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:59.970000
2/10/2021 11:06:13 PM	2/10/2021 11:06:28 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:59.970000
2/10/2021 11:07:14 PM	2/10/2021 11:07:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 11:08:44 PM	2/10/2021 11:09:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.159000
2/10/2021 11:11:14 PM	2/10/2021 11:11:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.115000
2/10/2021 11:12:14 PM	2/10/2021 11:12:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.115000
2/10/2021 11:13:14 PM	2/10/2021 11:13:59 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.115000
2/10/2021 11:15:14 PM	2/10/2021 11:15:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.115000
2/10/2021 11:16:14 PM	2/10/2021 11:16:29 PM	prod-brazilsouth1-a	srlive000001	prd_live_cbccampogrande	Unplanned	HighLatencyLogin	TopWaitStat_LCK_M_SCH_S	WaitTime: 00:01:50.115000

Top Waitstats locks

In **ASC -> Connectivity -> Data Explorer -> [anypackage_0_0,anypackage_26078, TopWaitStat_LCK] - Slow Logins, SNI ReadTimeout**, check "TopWaitStat_LCKs breakdown" for wait type details during login.



TopWaitStat_LCKs breakdown

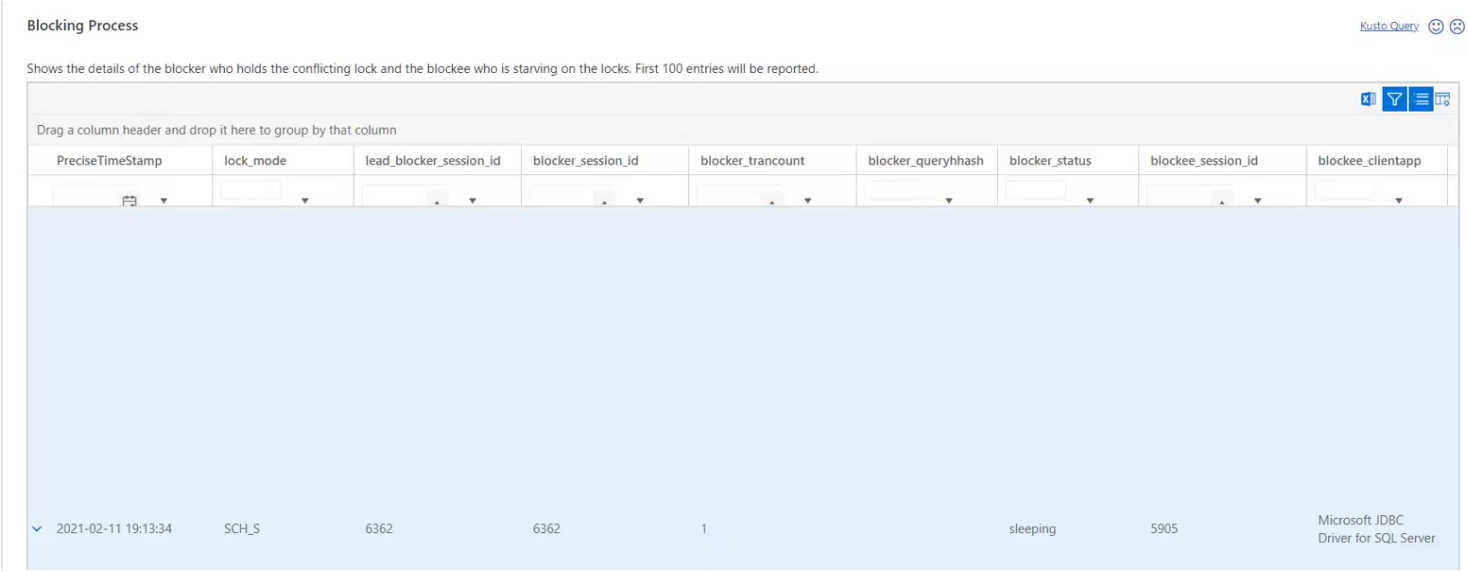
Kusto Query

Logins need to lookup users, logins, firewall rules etc from the security cache and/or from the backend tables in the database. So it has to acquire locks on those caches or database objects. If another thread (user txns or background threads like failover, audit etc) is holding a conflicting lock, logins would starve. Following table lists top 100 logins that are starving for locks and it's break down. High time on contained_authentication_time_ms indicates contention on the security cache or the related backend tables. High time on use_db_database_firewall_rules_time_ms indicates contention on the database's shared lock. High time on database_firewall_rules_time_ms indicates contention on the sys.database_firewall_rules.

error	state	state_desc	TopWaitStat	WaitTime	total_time_ms	contained_authentication_time_ms	use_db_database_firew
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120255	120256	120255	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120225	120226	120226	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120191	120192	120191	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120188	120189	120189	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120159	120160	120159	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120155	120159	120155	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120151	120151	120151	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120141	120141	120141	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120136	120137	120136	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120130	120131	120130	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120127	120128	120127	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120110	120111	120110	0
40613	128	RetrieveDatabaseFirewallRulesFailed	LCK_M_SCH_S	120110	120110	120110	0

Blocking process and head blocker

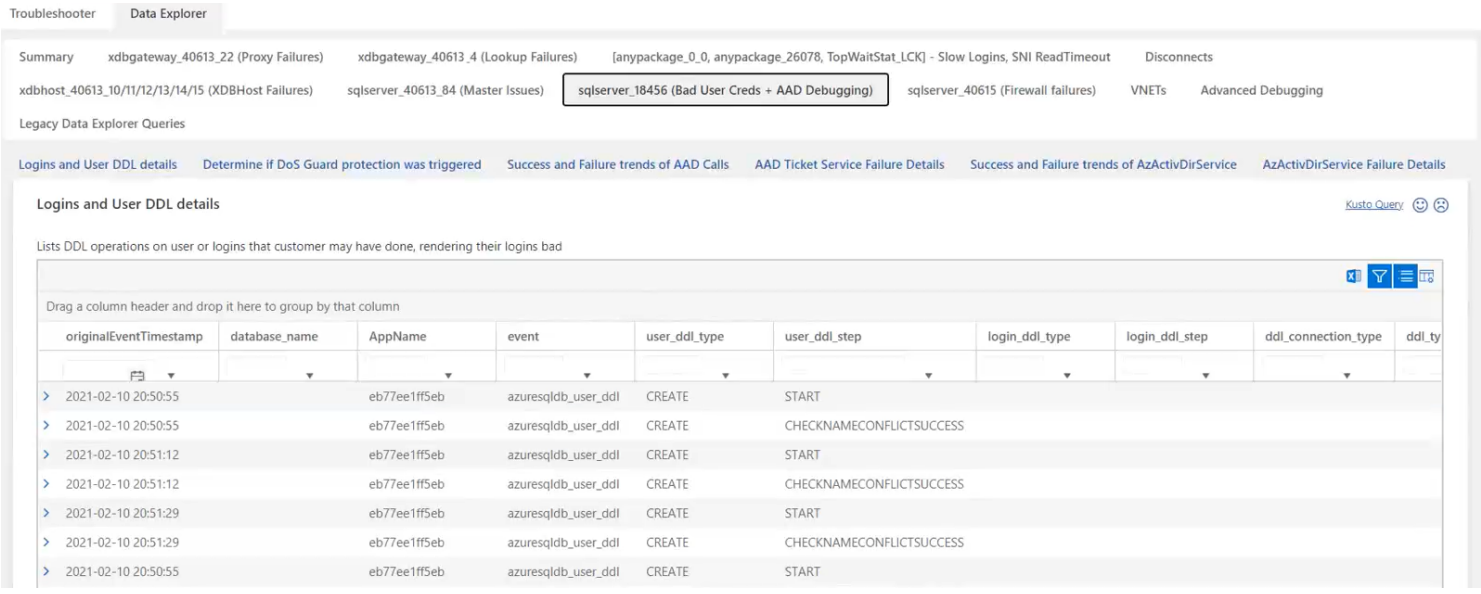
In **ASC -> Connectivity -> Data Explorer -> [anypackage_0_0,anypackage_26078, TopWaitStat_LCK] - Slow Logins, SNI ReadTimeout**, check "Blocking Process" to understand the blocking header is background process or client process.



- If it is a background process, check "Lead Blockers Activity Log" to see if this background process has anything logged in the ErrorLog which will help you understand the details. This usually happend during failover, where our backend hold the lock and blocks the login.
- If it is a client process, check with the customer why this client is holding locks for this long. Mitigation should be discussed with customer to kill this session or fix this long running session issue.

Logins and user DDL details

If it is a client process and a contained login is taking long, please go to **ASC -> Connectivity -> Data Explorer -> sqlserver_18456(Bad User Creds +AAD Debugging)** and check "Logins and User DDL details". If customer is modifying the user like create without finish, then it will hold the lock and cause blocking on contained user login.



Kusto

MonLogin

From MonLogin, you can see long login with total time more than 14s due to long wait time on schema lock in message column. The error and stat could be different.

```
let srv = "servername";
let db = "databasename";
let startTime = datetime(2023-02-14 04:00:00Z);
let endTime = datetime(2023-02-14 18:00:00Z);
let timeRange = ago(1d);
MonLogin
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
//| where  TIMESTAMP >= timeRange
| where logical_server_name =~ srv
| where database_name =~ db
| where package == "sqlserver"
| where total_time_ms > 14000
| where error > 0
| limit 100
| project originalEventTimestamp, LogicalServerName, database_name, AppName, package, error, state, total_time
```

MonBlockedProcessReportFiltered

From MonBlockedProcessReportFiltered, you can summarize the blocking count by lock_mode and confirm that the blocking is on schema lock and corresponds to the time when the issue occurred.

By playing with the 3 commented lines and removing the "summarize" line, you can also narrow down to specific blocked process reports and analyze their details. Typically, the blocking schema lock is on the security cache or another backend tables - see sample output further below.

```
let srv = "servername";
let db = "databasename";
let startTime = datetime(2023-02-14 04:00:00Z);
let endTime = datetime(2023-02-14 18:00:00Z);
let timeRange = ago(1d);
MonBlockedProcessReportFiltered
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
//| where  TIMESTAMP >= timeRange
| where logical_server_name =~ srv
| where database_name =~ db
| project TIMESTAMP, LogicalServerName, logical_server_name, AppName, AppName_Ex, event, database_id, logical_
//| where lock_mode == "SCH_M" or lock_mode == "SCH_S"
//| project TIMESTAMP, lock_mode, blocked_process_filtered
//| limit 100
| summarize count(), min(TIMESTAMP), max(TIMESTAMP) by lock_mode
```

Sample output:

Sample Telemetry for LCK_M_SCH_M and LCM_M_SCH_S wait on Security Cache

```
MonLogin
| where TIMESTAMP > datetime(2020-05-08 13:00) and TIMESTAMP < datetime(2020-05-08 14:00)
| where logical_server_name =~ "saral-appdbserver-a13b66de"
| where database_name =~ "saral_axdb_03a68bfc50632543"
| where is_success == 0
| project originalEventTimestamp, error, ['state'], is_user_error, database_firewall_rules_time_ms,message
```

originalEventTimestamp	error	state	is_user_error	database_firewall_rules_time_ms	message
2020-05-08 13:55:22.7140345	26078	1	0	0	<waitstats><wait name="LCK_M_SCH_M" requests="1" time="92171"
2020-05-08 13:55:22.7140857	26078	1	0	0	<waitstats><wait name="LCK_M_SCH_M" requests="1" time="90862"
2020-05-08 13:55:22.7142380	26078	1	0	0	<waitstats><wait name="LCK_M_SCH_M" requests="1" time="92088"

```
<blocked-process-report monitorLoop="552266">
<blocked-process>
  <process id="process2a8c78f8108" taskpriority="0" logused="0" waitresource="OBJECT: 60:615673241:4 " waittime="175635" ownerId="4661625448" transactionname="MDView"
lasttranstarted="2020-05-27T02:00:07.057" XDES="0x2a8b6589b8" lockMode="Sch-S" schedulerid="5" kpid="2936" status="suspended" spid="258" sbid="0" ecid="0" priority="0" trancount="0"
lastbatchstarted="2020-05-27T02:00:06.657" lastbatchcompleted="2020-05-27T02:00:06.640" lastattention="1900-01-01T00:00:00.640" clientapp="FinancialReporting" hostname="filtered"
hostpid="6196" loginname="filtered" isolationlevel="read committed (2)" xactid="0" currentdb="60" currentdbname="db_d365opsprod_hpeuat_mr_20200521_06452519_6847"
lockTimeout="4294967295" clientoption1="671088672" clientoption2="128056">
    <executionStack>
     <frame queryhash="0x7ec601e5e9838108" queryplanhash="0x84a65e8c1500feb2" line="2" stmtstart="4" stmtend="1356"
sqlhandle="0x020000007d97cf30d1664cf6fc167908cc3934afa2475db0000000000000000000000000000000000000000"/>
    </executionStack>
    <inputbuf>
filtered   </inputbuf>
  </process>
 </blocked-process>
 <blocking-process>
  <process status="suspended" waitresource="METADATA: database_id = 1 SECURITY_CACHE($hash = 0x0:0x18), lockPartitionId = 0" waittime="2094" spid="623" sbid="0" ecid="0" priority="0"
trancount="0" lastbatchstarted="2020-05-27T01:59:13.813" lastbatchcompleted="2020-05-27T01:59:13.813" lastattention="2020-05-27T01:58:51.510" clientapp="FinancialReporting"
hostname="filtered" hostpid="8300" loginname="filtered" isolationlevel="read committed (2)" xactid="0" currentdb="60" currentdbname="db_d365opsprod_hpeuat_mr_20200521_06452519_6847"
lockTimeout="4294967295" clientoption1="671219744" clientoption2="128056">
    <executionStack>
     <frame queryhash="0x5079197c16545b0b" queryplanhash="0x09dd8316273cc9ab" line="5" stmtstart="204" stmtend="746"
sqlhandle="0x03003c00e02cac296bbbb50018aa000001000000000000000000000000000000000000000000000000000000000000"/>
     <frame queryhash="0x0000000000000000" queryplanhash="0x0000000000000000" line="10" stmtstart="786" stmtend="942"
sqlhandle="0x03003c008a53f32385045600b0ab000001000000000000000000000000000000000000000000000000000000000000"/>
     <frame queryhash="0x0000000000000000" queryplanhash="0x0000000000000000" line="56" stmtstart="4888" stmtend="5064"
sqlhandle="0x03003c00d4ab884ee0045600b0ab000001000000000000000000000000000000000000000000000000000000000000"/>
     <frame queryhash="0x0000000000000000" queryplanhash="0x0000000000000000" line="138" stmtstart="11264" stmtend="11392"
sqlhandle="0x03003c006263a04c97045600b0ab000001000000000000000000000000000000000000000000000000000000000000"/>
    </executionStack>
    <inputbuf>
filtered   </inputbuf>
  </process>
 </blocking-process>
</blocked-process-report>
```

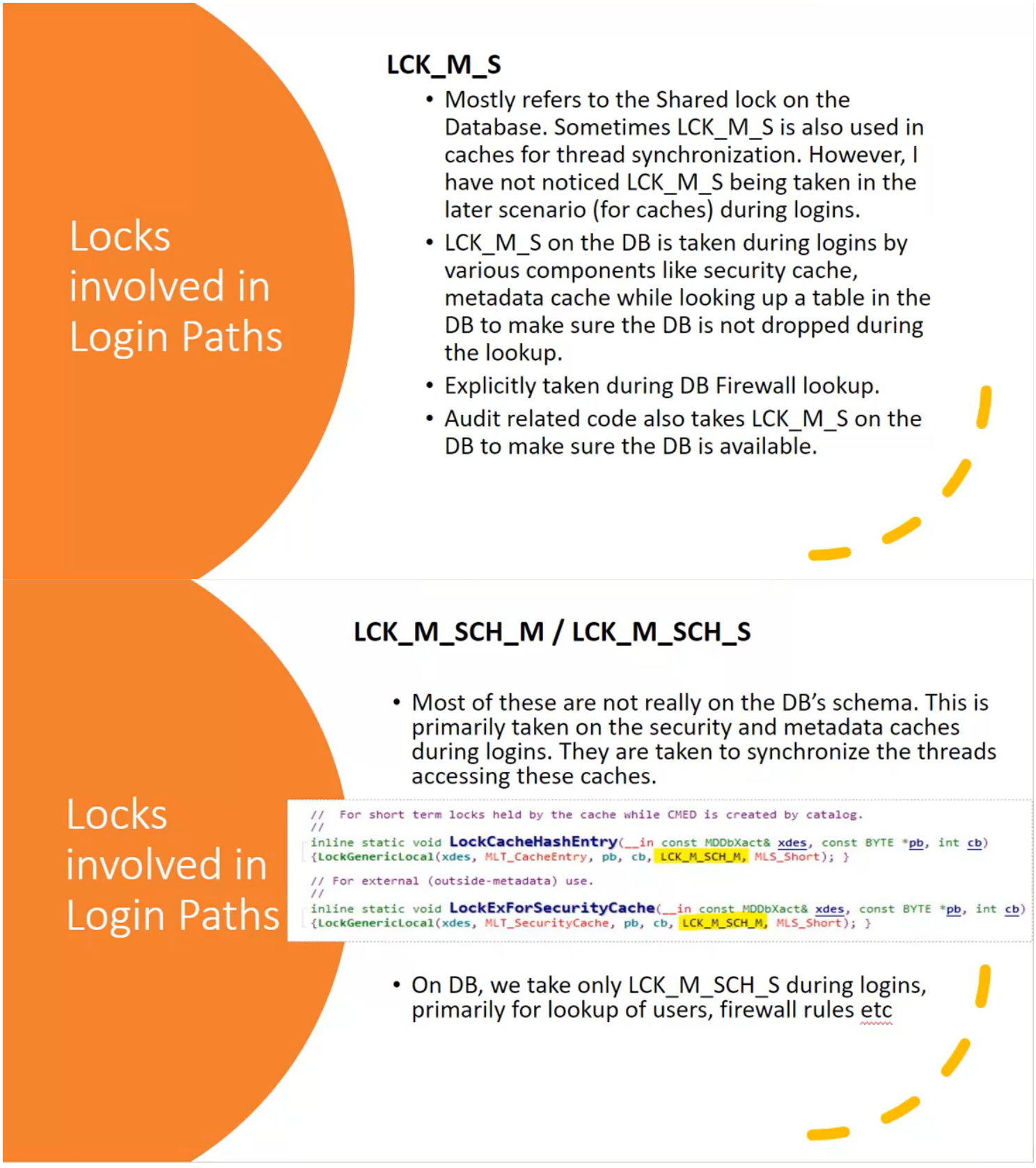
Mitigation

If the head blocker is a background process, it could have occurred during a failover where our backend holds the lock on purpose to block the logins. It should be a quick operation and resolve itself once the failover is complete. If it is occurring for longer and/or is still happening, you should open an IcM and get it investigated further.

If the head blocker is a client process, then check with the customer why this client is holding locks for this long. Mitigation should be discussed with customer to kill this session or fix this long running session issue. Check the blocked process report for open transactions on the blocking header to see why transaction is running too long.

More Information

Watch 'Troubleshooting long login issues due to locks' | Microsoft Stream



How good have you found this content?