Verifying waitstats

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:28 AM PST

Contents

Verifying waitstats

Issue

This is a "How To" TSG that provides you with Kusto telemetry queries and DMV queries for identifying wait stats.

Investigation / Analysis

The queries listed in this section return the following column values:

column	content
wait_type	Name of the wait type. See Types of DB Waits and Types of OS waits for additional information and a list of values.
waiting_tasks_count	Number of waits on this wait type. This counter is incremented at the start of each wait.
wait_time_ms	Total wait time for this wait type in milliseconds. This time is inclusive of signal_wait_time_ms.
max_wait_time_ms	Maximum wait time on this wait type.
signal_wait_time_ms	Difference between the time that the waiting thread was signaled and when it started running.

Kusto Telemetry

Verify waitstats using the following Kusto queries:

Worker thread waits

```
// Worker thread waits
let startTime = datetime(2022-09-27 14:00:00Z);
let endTime = datetime(2022-09-27 16:00:00Z);
let srv = "servername";
let db = "databasename";
MonWorkerWaitStats
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where LogicalServerName =~ srv or logical_server_name =~ srv
| where logical_database_name =~ db
| order by originalEventTimestamp asc
| project originalEventTimestamp, NodeName, LogicalServerName, logical_database_name, AppName, session_id, err
| limit 1000
```

Sample output:

originalEventTimestamp	Nod...	LogicalSe...	logical_database...	AppName	session_id	error_state	query_hash	query_plan_hash	text_data
2022-09-27 14:27:47.60...	DB.61	weholgerl	AdventureWorks	ac4fb573f27d	76	52	4885638715773847243	6211063923851286534	<waitstats><wait name="LCK_M_IU" requests="1" time="7145" signalTime="0" maxTime="7145"/><wait n
2022-09-27 14:31:12.33...	DB.61	weholgerl	AdventureWorks	ac4fb573f27d	76	52	4885638715773847243	6211063923851286534	<waitstats><wait name="LCK_M_IU" requests="1" time="173276" signalTime="0" maxTime="173276"/><w
2022-09-27 15:38:01.29...	DB.61	weholgerl	AdventureWorks	ac4fb573f27d	76	52	4885638715773847243	6211063923851286534	<waitstats><wait name="LCK_M_IU" requests="1" time="4003915" signalTime="0" maxTime="4003915"/>

Statement-level waits

```
let startTime = datetime(2022-09-27 14:00:00Z);
let endTime = datetime(2022-09-27 16:00:00Z);
let srv = "servername";
let db = "databasename";
MonWiQdsWaitStats
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where LogicalServerName =~ srv
| where database_name  =~ db
//| where query_hash in ("0xB662B6B62D36C008")
| where max_query_wait_time_ms > 1000  // reduce noise
| project TIMESTAMP, originalEventTimestamp, exec_type, statement_type,  query_hash, query_id, query_plan_hash
| order by TIMESTAMP asc
```

Sample output:

TIMESTAMP	exec_type	statement_type	query_hash	...	query_plan_ha...	p..	is_...	que...	wait_category	total_query_wait_time_ms	avg_query_wait_time_ms	min_query_wait_time_ms	max_query_wait_time_ms
2022-09-27 14:33:29....	0	x_estypSelect	0x018C617DBDDCFB52	..	0x80BB0673D...	;...	0	0	BUFFERIO	1386	1386	1386	1386
2022-09-27 14:33:29....	0	x_estypSelect	0x4C9F3C246E348B17	..	0x20AB037B5...	;...	0	0	BUFFERIO	1464	1464	1464	1464
2022-09-27 14:33:29....	0	x_estypUpdate	0xEBB4FDCB3A8B5B4F	..	0x5559462BB...	;...	1	0	UNKNOWN	5137	2568,5	0	5137
2022-09-27 14:33:29....	0	x_estypUpdate	0xEBB4FDCB3A8B5B4F	..	0x5559462BB...	;...	1	0	BUFFERIO	2877	1438,5	0	2877
2022-09-27 14:33:30....	3	x_estypSelect	0x43CD4686DA9886CB	..	0x563221BB2...	;...	1	0	LOCK	180422	90211	7145	173276

Top 20 average wait types overview

```
// Top 20 average wait types overview
let startTime = datetime(2022-09-27 14:00:00Z);
let endTime = datetime(2022-09-27 16:00:00Z);
let srv = "servername";
let db = "databasename";
MonDmCloudDatabaseWaitStats
| where end_utc_date > startTime
| where end_utc_date < endTime
| where LogicalServerName =~ srv
| where database_name =~ db
| extend delta_wait_time_ms_History = delta_wait_time_ms * 1.0 / delta_waiting_tasks_count
| summarize avg(delta_waiting_tasks_count), max(delta_waiting_tasks_count), avg(delta_signal_wait_time_ms), av
| extend avg_waiting_tasks_count = toint(avg_delta_waiting_tasks_count)
| extend max_waiting_tasks_count = max_delta_waiting_tasks_count
| extend avg_signal_time_ms = toint(avg_delta_signal_wait_time_ms)
| extend avg_wait_time_ms = toint(avg_delta_wait_time_ms_History)
| extend max_wait_time_ms = toint(max_max_wait_time_ms)
| where max_wait_time_ms > 10  // reduce noise
| project wait_type, avg_waiting_tasks_count, max_waiting_tasks_count, avg_signal_time_ms, avg_wait_time_ms, m
| top 20 by avg_wait_time_ms desc
```

Sample output:

wait_type	avg_waiting_tasks_count	max_waiting_tasks_count	avg_signal_time_ms	avg_wait_time_ms	max_wait_time_ms
LCK_M_IU	1	1	0	1394779	4003915
WAIT_ON_SYNC_STATISTICS_REFRESH	2	4	0	437	2322
PAGEIOLATCH_EX	1	1	0	22	22
RESOURCE_GOVERNOR_IDLE	2	2	0	11	12
WRITELOG	3	7	0	9	15
PREEMPTIVE_XHTTP	15	15	0	9	47
PAGEIOLATCH_SH	353	703	17	7	39
ASYNC_NETWORK_IO	443	443	14	4	204
SOS_SCHEDULER_YIELD	148	931	5	0	12

Show all wait stats

```
// Show all wait stats
// can filter on wait_times and wait_types as needed
let startTime = datetime(2022-09-27 14:00:00Z);
let endTime = datetime(2022-09-27 16:00:00Z);
let srv = "servername";
let db = "databasename";
MonDmCloudDatabaseWaitStats
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where server_name =~ srv
| where database_name =~ db
| where delta_wait_time_ms > 10 or delta_max_wait_time_ms > 10  // reduce noise
//| where wait_type in ("HADR_SYNC_COMMIT", "CXCONSUMER", "CXSYNC_PORT")
| extend avg_wait_time_per_task_ms = round(1.0 * (delta_wait_time_ms) / delta_waiting_tasks_count, 2)
| project TIMESTAMP, PreciseTimeStamp, NodeName, AppName, LogicalServerName, database_id, database_name, wait_
| sort by PreciseTimeStamp asc nulls last
```

Sample output:

TIMESTAMP	PreciseTimeStamp	NodeName	AppName	LogicalServerName	database_id	database_name	wait_type	delta_waiting_tasks_count	delta_wait_time_ms	delta_max_wait_time_ms	delta_signal_wait_time_ms	avg_wait_time_per_task_ms
2022-09-27 14:26:02.4222792	2022-09-27 14:26:02.4222792	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	WRITELOG	7	26	0	0	3,71
2022-09-27 14:26:02.4222792	2022-09-27 14:26:02.4222792	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	WAIT_ON_SYNC_STATISTICS_REFRESH	4	5126	2322	0	1281,5
2022-09-27 14:31:02.4413666	2022-09-27 14:31:02.4413666	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	LCK_M_IU	1	7145	7145	0	7145
2022-09-27 14:31:02.4413666	2022-09-27 14:31:02.4413666	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	MEMORY_ALLOCATION_EXT	8932	14	0	0	0
2022-09-27 14:31:02.4413666	2022-09-27 14:31:02.4413666	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	WAIT_ON_SYNC_STATISTICS_REFRESH	1	28	0	0	28
2022-09-27 14:36:02.4605097	2022-09-27 14:36:02.4605097	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	LCK_M_IU	1	173277	166131	0	173277
2022-09-27 15:41:02.7090049	2022-09-27 15:41:02.7090049	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	LCK_M_IU	1	4003915	3830639	0	4003915
2022-09-27 15:41:02.7090049	2022-09-27 15:41:02.7090049	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	MEMORY_ALLOCATION_EXT	9164	15	0	0	0
2022-09-27 15:41:02.7090049	2022-09-27 15:41:02.7090049	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	IO_COMPLETION	5	20	8	0	4
2022-09-27 15:41:02.7090049	2022-09-27 15:41:02.7090049	DB.61	ac4fb573f27d	weholgerl	9	AdventureWorks	WRITELOG	1	15	6	0	15

Show graph of wait type and wait time

```
// show graph of wait type and wait time
let startTime = datetime(2022-09-27 14:00:00Z);
let endTime = datetime(2022-09-27 16:00:00Z);
let srv = "servername";
let db = "databasename";
MonDmCloudDatabaseWaitStats
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where LogicalServerName =~ srv
| where database_name =~ db
| summarize sum(delta_wait_time_ms) by bin(end_utc_date, 5min), wait_type
| where sum_delta_wait_time_ms > 10 // reduce noise
| sort by end_utc_date asc nulls last
| project end_utc_date, wait_type, sum_delta_wait_time_ms
| render timechart
```

DMV sys.dm-db-wait-stats

This needs to be run by the customer in the affected database.

```
select * from (
  SELECT
    [wait_type],
    [wait_time_ms],
    ([wait_time_ms] - [signal_wait_time_ms]) AS [resource_time_ms],
    [signal_wait_time_ms],
    [waiting_tasks_count],
    100.0 * [wait_time_ms] / SUM ([wait_time_ms]) OVER() AS [percentage]
    FROM sys.dm_db_wait_stats
    WHERE [waiting_tasks_count] > 0
) as t1
where t1.percentage > 0.01
order by t1.percentage desc;
```

Sample output:

Results | Messages | Execution plan

	wait_type	wait_time_ms	resource_time_ms	signal_wait_time_ms	waiting_tasks_count	percentage
1	SOS_SCHEDULER_YIELD	8172	0	8172	369	64.924128068642249
2	PAGEIOLATCH_SH	4054	4047	7	62	32.207833478986255
3	ASYNC_NETWORK_IO	250	250	0	74	1.986176213553666
4	PREEMPTIVE_HTTP_REQUEST	44	44	0	38	0.349567013585445
5	MEMORY_ALLOCATION_EXT	43	43	0	34100	0.341622308731230
6	WRITELOG	24	24	0	3	0.190672916501151

DMV sys.dm-os-wait-stats

This needs to be run by the customer in the affected database. Similar information is also available in ASC.

```
select * from (
  SELECT
    [wait_type],
    [wait_time_ms],
    ([wait_time_ms] - [signal_wait_time_ms]) AS [resource_time_ms],
    [signal_wait_time_ms],
    [waiting_tasks_count],
    100.0 * [wait_time_ms] / SUM ([wait_time_ms]) OVER() AS [percentage]
    FROM sys.dm_os_wait_stats
    WHERE [wait_type] NOT IN (
      N'BROKER_EVENTHANDLER', N'BROKER_RECEIVE_WAITFOR', N'BROKER_TASK_STOP', N'BROKER_TO_FLUSH', N'BROKER_TRA
      N'CHECKPOINT_QUEUE', N'CHKPT', N'CLR_AUTO_EVENT', N'CLR_MANUAL_EVENT', N'CLR_SEMAPHORE',
      N'DBMIRROR_DBM_EVENT', N'DBMIRROR_EVENTS_QUEUE', N'DBMIRROR_WORKER_QUEUE', N'DBMIRRORING_CMD',
      N'DIRTY_PAGE_POLL', N'DISPATCHER_QUEUE_SEMAPHORE', N'EXECSYNC', N'FSAGENT',
      N'FT_IFTS_SCHEDULER_IDLE_WAIT', N'FT_IFTSHC_MUTEX',
      N'HADR_CLUSAPI_CALL', N'HADR_FILESTREAM_IOMGR_IOCOMPLETION', N'HADR_LOGCAPTURE_WAIT', N'HADR_NOTIFICATIO
      N'KSOURCE_WAKEUP', N'LAZYWRITER_SLEEP', N'LOGMGR_QUEUE', N'ONDEMAND_TASK_QUEUE', N'PWAIT_ALL_COMPONENTS_
      N'QDS_PERSIST_TASK_MAIN_LOOP_SLEEP', N'QDS_ASYNC_QUEUE', N'QDS_SHUTDOWN_QUEUE', N'QDS_CLEANUP_STALE_QUER
      N'RESOURCE_GOVERNOR_IDLE', N'REQUEST_FOR_DEADLOCK_SEARCH', N'RESOURCE_QUEUE', N'SERVER_IDLE_CHECK',
      N'SLEEP_BPOOL_FLUSH', N'SLEEP_DBSTARTUP', N'SLEEP_DCOMSTARTUP', N'SLEEP_MASTERDBREADY', N'SLEEP_MASTERMD
      N'SNI_HTTP_ACCEPT', N'SP_SERVER_DIAGNOSTICS_SLEEP', N'SQLTRACE_BUFFER_FLUSH', N'SQLTRACE_INCREMENTAL_FLU
      N'WAIT_FOR_RESULTS',N'WAITFOR', N'WAITFOR_TASKSHUTDOWN',N'WAIT_XTP_HOST_WAIT', N'WAIT_XTP_OFFLINE_CKPT_N
      N'XE_DISPATCHER_JOIN', N'XE_DISPATCHER_WAIT', N'XE_TIMER_EVENT')
    AND [waiting_tasks_count] > 0
) as t1
where t1.percentage > 0.01
order by t1.percentage desc;
```

Sample output:

Results | Messages | Execution plan

	wait_type	wait_time_ms	resource_time_ms	signal_wait_time_ms	waiting_tasks_count	percentage
1	SOS_WORK_DISPATCHER	3598966497	3598585659	380838	576865	86.719936661501096
2	XE_LIVE_TARGET_TVF	181464687	181464445	242	3086	4.372534775207472
3	HADR_FABRIC_CALLBACK	90018579	90017441	1138	12053	2.169068668948582
4	VDI_CLIENT_OTHER	14643221	14643134	87	916	0.352839960777318
5	PWAIT_EXTENSIBILITY_CLEANUP_TASK	90602617	0	90602617	303	2.183141525256116
6	PVS_PREALLOCATE	90639173	90639079	94	902	2.184022370912012
7	POPULATE_LOCK_ORDINALS	83404463	83404457	6	4	2.009696326619211

Public Doc Reference

- sys.dm_db_wait_stats (Azure SQL Database)
- sys.dm_os_wait_stats (Transact-SQL)

Internal Doc Reference

- Blocking
- Troubleshooting Blocking

How good have you found this content?