Errorlog queries (Managed Instance)

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:42 PM PDT

```
//*******************************************************************************************
// Errorlog & dumps crashes
//*******************************************************************************************
// E.01
// Dumps & Crashes for primary nodes only, secondary nodes are ignored
let buffer = 15m; // we take dumps that have occured this much before/after we've seen the DB on the node
let clPhysicalDbGuid = toscalar
(
    MonDmRealTimeResourceStats
    | where AppTypeName == 'Worker.CL'
    | where LogicalServerName == '{LogicalServerName}' and database_name =~ '{LogicalDatabaseName}'
    | summarize arg_max(TIMESTAMP, physical_database_guid)
    | project physical_database_guid
);
let dbNameForHadrReplicaStates = iif(isnotempty(clPhysicalDbGuid), clPhysicalDbGuid,
'{LogicalDatabaseName}');
let PrimaryReplicas = MonDmDbHadrReplicaStates
| where LogicalServerName =~ "{LogicalServerName}" and logical_database_name  =~
dbNameForHadrReplicaStates
| where is_primary_replica == 1
| extend isB = (AppName startswith 'b')
| order by isB asc, TIMESTAMP asc nulls first
| extend prevB = prev(isB), nextB = next(isB)
| extend prevNodeName = prev(NodeName), nextNodeName = next(NodeName)
| extend isFirst= (NodeName != prevNodeName) or (isB != prevB)
| extend isLast=(NodeName != nextNodeName) or (isB != nextB)
| where isFirst or isLast
| extend EndTime=next(TIMESTAMP)
| extend StartTime=TIMESTAMP
| where isFirst == true
| extend codeversion=extract("^(.*)-WASD", 1, code_package_version)
| project ClusterName, NodeName, AppName, StartTime, EndTime, code_package_version, codeversion;
let AllDumps = MonSqlDump
| where LogicalServerName =~ "{LogicalServerName}"
| where CompleteFileName endswith ".dmp"
| summarize  TIMESTAMP=min(TIMESTAMP) by NodeName, AppName, CompleteFileName
| order by TIMESTAMP asc
| project TIMESTAMP, NodeName, AppName, CompleteFileName;
PrimaryReplicas
| join kind= inner (
   AllDumps
) on NodeName, AppName
| where TIMESTAMP between ((StartTime - buffer) .. (EndTime + buffer))
| project TIMESTAMP, NodeName, AppName, CompleteFileName
| top 100000 by TIMESTAMP desc

// E.02
// Error Summary
AlrSQLErrorsReported
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}"  and   AppName =~ "{AppName}"
| where error_number == 701
| summarize error_count=count() by bin(TIMESTAMP, 5min), error_no=strcat(error_number, ""), NodeName
| render timechart

AlrSQLErrorsReported
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and AppName =~ "{AppName}"
| where error_number in (3314,911,9001)

// E.03
// Errorlog (filtered)
MonSQLSystemHealth
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and AppName =~ "{AppName}" and NodeName =~ "
{NodeName}"
| where event== "systemmetadata_written"
| where message notcontains "[INFO]" and message notcontains "[TDE] SetDbeState" and message notcontains
"Log was backed up" and message notcontains "created in FSLOG"
| where message notcontains "accepting vlf header" and  message notcontains "Log writer started sending"
and message notcontains "DbMgrPartnerCommitPolicy::SetSyncAndRecoveryPoint"
| where message notcontains "Started XE session" and message notcontains "
[CFabricReplicaManager::GetHadrSessionTimeout] HADR Session Timeout"
| where message notcontains "[CFabricReplicaManager::GetHadrPingFrequency] HADR Ping Frequency" and
message notcontains "Zeroing completed on"
| where message notcontains "HADR_FQDR_XRF: Sending the extended recovery forks to secondary" and
message notcontains "Cleaning up conversations for"
| where message notcontains "[CFabricReplicaManager::Start] Registered service type" and message notcontains
"Backup" and message notcontains "Recovery of database"
| where message notcontains "[DynamicDeactivationTask::DoDbOperation]
DynamicFileAllocationDetectionUnderLockTask was called"
| where message notcontains "No Changes detected from manager notification"
| where  strlen(extract(@"Backup(.*)\s+processed",1,message))<=0
| where  strlen(extract(@"Backup(.*)\s+LSN",1,message))<=0
| where  strlen(extract(@"Backup(.*)\s+Estimated",1,message))<=0
| where  strlen(extract(@"Zeroing(.*)\s+from page",1,message))<=0
| where  strlen(extract(@"Zeroing(.*)\s+from page",1,message))<=0
| where  strlen(extract(@"Backup(.*)\s+WITH DIFFERENTIAL",1,message))<=0
| where  strlen(extract(@"\[DecryptCurrDEK\] (.*)\s+Logical Database Name",1,message))<=0
| where  strlen(extract(@"(Received|Sending) page request (from|to) (.*)\s+for Undo or APR for
page",1,message))<=0
| extend msg_TIMESTAMP = extract(@"(20[0-9]{2}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}.[0-9]{2})", 1, message)
| extend msg_TIMESTAMP = todatetime(msg_TIMESTAMP)
| project PreciseTimeStamp, msg_TIMESTAMP, NodeName, message
| order by PreciseTimeStamp asc nulls last , msg_TIMESTAMP asc nulls last

MonSQLSystemHealth
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and   AppName =~ "{AppName}" and NodeName =~ "
{NodeName}"
| project PreciseTimeStamp, message
| where message contains "stack"
//| where message !contains ("DecryptCurrDEK")
| order by PreciseTimeStamp asc nulls last

// E.04
// Errorlog (aggregate)
MonSQLSystemHealth
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and   AppName =~ "{AppName}" and NodeName =~ "
{NodeName}"
| where event=="systemmetadata_written"
| where message notcontains "[INFO]" and message notcontains "[TDE] SetDbeState" and message notcontains
"Log was backed up" and message notcontains "created in FSLOG"
| where message notcontains "accepting vlf header" and  message notcontains "Log writer started sending"
and message notcontains "DbMgrPartnerCommitPolicy::SetSyncAndRecoveryPoint"
| where message notcontains "Started XE session" and message notcontains "
[CFabricReplicaManager::GetHadrSessionTimeout] HADR Session Timeout"
| where message notcontains "[CFabricReplicaManager::GetHadrPingFrequency] HADR Ping Frequency" and
message notcontains "Zeroing completed on"
| where message notcontains "HADR_FQDR_XRF: Sending the extended recovery forks to secondary" and
message notcontains "Cleaning up conversations for"
| where message notcontains "[CFabricReplicaManager::Start] Registered service type" and message notcontains
"Backup" and message notcontains "Recovery of database"
| where  strlen(extract(@"Backup(.*)\s+processed",1,message))<=0   //  Backup(b394e9a7-1ae7-48e6-b2d6-
0725e6b3e7a7): 10 percent (405405696/4053934080 bytes) processed
| where  strlen(extract(@"Backup(.*)\s+LSN",1,message))<=0   //  Backup(b394e9a7-1ae7-48e6-b2d6-
0725e6b3e7a7): Last LSN: 11817791:18192:1
| where  strlen(extract(@"Backup(.*)\s+Estimated",1,message))<=0   //  Backup(b394e9a7-1ae7-48e6-b2d6-
0725e6b3e7a7): Estimated total size = 4665073664 bytes (data size = 0 bytes, log size = 4665073664 bytes)
| where  strlen(extract(@"Zeroing(.*)\s+from page",1,message))<=0   //  Zeroing  ... from page
| where  strlen(extract(@"Backup(.*)\s+WITH DIFFERENTIAL",1,message))<=0
| where  strlen(extract(@"\[DecryptCurrDEK\] (.*)\s+Logical Database Name",1,message))<=0
| where  strlen(extract(@"Backup(.*)\s+Copying",1,message))<=0
| extend msg = case (strlen(extract(@" spid[0-9]{1,9}[s]{0,1}\s+([^$]*)",1,message))>0, extract(@" spid[0-9]{1,9}
[s]{0,1}\s+([^$]*)",1,message), message)
| summarize min(TIMESTAMP), max(TIMESTAMP), count() by msg
| order by count_ desc nulls last

MonSQLSystemHealth
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and AppName =~ "{AppName}" and NodeName =~ "
{NodeName}"
| where event=="systemmetadata_written"
| where message !contains "DecryptCurrDEK"
//| where message contains "recovery complete"
| extend msg_TIMESTAMP = extract(@"(20[0-9]{2}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}.[0-9]{2})", 1, message)
| project PreciseTimeStamp , NodeName, msg_TIMESTAMP, message
| order by msg_TIMESTAMP asc, PreciseTimeStamp asc nulls last

 // E.05
// error of interest
MonSQLSystemHealth
| where TIMESTAMP > datetime({StartTime}) and TIMESTAMP < datetime({EndTime})
| where LogicalServerName =~ "{LogicalServerName}" and   AppName =~ "{AppName}" and NodeName =~ "
{NodeName}"
| where event=="systemmetadata_written"
| where message contains "I/O requests taking er than 15 seconds" or message contains "Wait for replica
catchup for"
  or message contains "Error:" or message contains "is not currently available"
  or message contains "[DbrSubscriber] Switching role to primary"
  or message contains "is changing roles from"
  or message contains "A connection timeout has occurred"
  or message contains "DBR Subscriber is now invoking switch role"
  or message contains "CHadrTransportReplica State change from"
  or message contains "latch"
  or message contains "dbg"
  or message contains "yield"
  or (message contains "701" and message contains "Error" )
  or message contains "FAIL_"
  or message contains "limit for the database"
| extend msg_TIMESTAMP = extract(@"(20[0-9]{2}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}.[0-9]{2})", 1, message)
| extend msg_TIMESTAMP = todatetime(msg_TIMESTAMP)
| project PreciseTimeStamp, msg_TIMESTAMP, NodeName, message
| order by PreciseTimeStamp asc nulls last, msg_TIMESTAMP asc nulls last

// E.06
// Console error output
MonNodeTraceETW
| where TIMESTAMP >ago(40h)
| where Message contains "{AppName}"
| project TIMESTAMP, NodeName, Message, Pid
| summarize min(TIMESTAMP), max(TIMESTAMP), any(Message) by NodeName, Pid
| order by NodeName asc , min_TIMESTAMP asc
```

How good have you found this content?