[draft]Repickup caused duplicate run

Last updated by | Ruoyu Li | Dec 8, 2022 at 2:04 AM PST

Contents

Issue

Sometimes customer reported that they see **duplicate** data got inserted to their copy activity sink.

Analysis

Simply run below query you can quickly discover it's a repickup issue.

```
TaskHostingEvent
| where ActivityId =~ 'f9acf728-9d45-4352-b3b1-41bc31fef1af' or JobId == '09286bd9-cfcb-1e4c-8019-f4d5585449ec
| project-reorder TIMESTAMP,TraceComponent,LogId,Message,ActivityId,JobId,Detail
```

In short, if you see MessageReset, you can say it's a duplicate run issue.

TIMESTAMP	TraceComponent	LogId	Message
2022-11-21 06:14:54.0127717	StorageQueueAccess	JobEntityCreatedInAzureTable	null
2022-11-21 06:14:54.0518832	StorageQueueAccess	JobCreated	null
2022-11-21 06:18:08.2234712	TaskManagementService	PulledOffNewTask	TaskMaxRetryCount:10, TaskCurrentRetryCount:1
2022-11-21 06:18:08.2236030	TaskManagementService	TaskPickUpLatency	null
2022-11-21 06:28:08.4348080	TaskManagementService	PulledOffNewTask	TaskMaxRetryCount:10, TaskCurrentRetryCount:2
2022-11-21 06:28:08.4348929	TaskManagementService	TaskPickUpLatency	null
2022-11-21 06:29:03.2242264	TaskManagementService	KeepAliveStart	null
2022-11-21 06:29:03.2324401	TaskManagementService	KeepAliveEnd	MessageReset
2022-11-21 06:29:16.6772074	TaskManagementService	KeepAliveStart	null
2022-11-21 06:29:16.6962210	TaskManagementService	KeepAliveEnd	KeepAliveSucceeded
2022-11-21 06:29:43.4136337	TaskManagementService	KeepAliveStart	null
2022-11-21 06:29:43.4228566	TaskManagementService	KeepAliveEnd	MessageReset
2022-11-21 06:29:56.9081914	TaskManagementService	KeepAliveStart	null
2022-11-21 06:29:56.9306381	TaskManagementService	KeepAliveEnd	KeepAliveSucceeded
2022-11-21 06:30:23.6084541	TaskManagementService	KeepAliveStart	null
2022-11-21 06:30:23.6169389	TaskManagementService	KeepAliveEnd	MessageReset

```
if (taskEntity.WorkerId != workerId)
                        {
                            return KeepAliveMessageResult.MessageReset;
                        }
```

Note: MessageNotExist like below is ignorable, as the next keepAlive can keep it alive.

TIMESTAMP	TraceComponent	LogId	Message
2022-11-21 06:42:05.5670888	TaskManagementService	KeepAliveStart	null
2022-11-21 06:42:05.5807938	TaskQueueStorageOperation	KeepAliveMessageNotExistErrorDetail	request context MessageId: c8b06f5a-bd1e-4c4c-85d2-3694f374b609, StorageException... at Microsoft.WindowsAzure.Storage.Shared.Protocol.HttpResponseParsers.ProcessExpe... at Microsoft.WindowsAzure.Storage.Queue.CloudQueue.<>c__DisplayClass36.<Updat... at Microsoft.WindowsAzure.Storage.Core.Executor.Executor.EndGetResponse[T](IAsync... --- End of inner exception stack trace --- at Microsoft.WindowsAzure.Storage.Core.Executor.Executor.EndExecuteAsync[T](IAsyn... at Microsoft.WindowsAzure.Storage.Core.Util.AsyncExtensions.<>c__DisplayClass7.<Cr... --- End of stack trace from previous location where exception was thrown ---
2022-11-21 06:42:05.5809737	StorageQueueAccess	KeepAliveMessageNotExistErrorDetail	Context is null
2022-11-21 06:42:05.5810631	TaskManagementService	KeepAliveEnd	MessageNotExist
2022-11-21 06:42:45.7522767	TaskManagementService	KeepAliveStart	null

```
else if (applied == KeepAliveMessageResult.MessageNotExist)
{
    // This message is already gone in the queue db but still on our list holding resource, keep d
    if (null != taskInfo)
    {
        logger.WriteKeepAliveMessageNotExistLog(taskInfo.TaskId, taskInfo.QueueId);

        //cancel the task when lost keep alive over 10 mins and the task won't be pickup by anothe
        //cancel will cancel the executor, report state and delete task from TM
        if (ShouldCancelTaskWhenLostKeepAlive(taskInfo))
        {
            CancelWhenLostKeepAlive(taskInfo);
        }
    }
    else
    {
        logger.WriteKeepAliveMessageNotExistLog(Guid.Empty, Guid.Empty);
    }
}
```

This copy ran on SHIR with 2 nodes.

AgentGroupId
562ba51e-1f65-4a90-b788-55f8400b24a3
a011b119-4849-491b-9809-3095b8722777