ServerLoop Hang

Last updated by | Dandan Zhang | Mar 30, 2021 at 10:08 AM PDT

There are an unbounded number of causes for ServerLoop hang. The problem is diagnosed by finding that the ServerLoop (which runs as the main thread in PostmasterMain) is not sending out a heartbeat (which should occur after 60 seconds has passed since the last heartbeat).

LogicalServerName	LastSandboxTimestamp	ClusterName	NodeName	process_id	AppTypeName	AppName	code_package_version	HangDuration
blackbox3	2021-02-03T14:24:54.9369494	tr9.indiasouth1-a.worker.database.windows.net	DB.27	53148	Worker.PAL.PG	d85a9dcb4513	13.1.20201214.4-orcshr-a9250702	13:48:36.4096159
relcare-uat-database	2021-02-03T14:24:55.328309	tr9.indiasouth1-a.worker.database.windows.net	DB.27	57704	Worker.PAL.PG	e2314e235673	13.1.20201214.4-orcshr-a9250702	13:49:41.4582269
anpad-spell	2021-02-03T14:39:54.0296193	tr4487.eastus2-a.worker.database.windows.net	DB.17	52912	Worker.PAL.PG	edf060611cea	13.1.20201214.4-orcshr-a9250702	14:00:58.5949239
u9ed98ee2fa6	2021-02-03T14:39:54.1314251	tr515.westeurope1-a.worker.database.windows.net	DB.32	48168	Worker.PAL.PG	ed30d8d7d0ee	13.1.20201214.4-orcshr-a9250702	06:44:26.2338785
xr-prod-db-server	2021-02-03T14:44:51.8843456	tr8640.eastus1-a.worker.database.windows.net	DB.4	40052	Worker.PAL.PG	d2b0cdc524db	13.1.20201214.4-orcshr-a9250702	17:51:13.6570173

Verify the Symptom:

Servers exhibiting this issue are located using the following query. Use the queries below to see that the server is still exhibiting the problem:

- First query shows that the server is still having issue.
- The second query verifies that there are no new connections being made. Use the HangDuration value from the first query to check if there are any new connections since the hang started.

```
MonRdmsPgSqlSandbox
| extend IsHeartBeat = (text contains "[postmaster.c][ServerLoop] heartbeat")
| summarize LastHeartbeatTimestamp = maxif(originalEventTimestamp, IsHeartBeat), LastSandboxTimestamp =
max(originalEventTimestamp) by ClusterName, NodeName, AppTypeName, AppName, LogicalServerName, process_id
code_package_version
| extend HangDuration = LastSandboxTimestamp - LastHeartbeatTimestamp
| where HangDuration > 180min
| project LogicalServerName, LastSandboxTimestamp, ClusterName, NodeName, process_id, AppTypeName, AppNam
code_package_version, HangDuration
| where LastSandboxTimestamp > ago(30m)
| order by LastSandboxTimestamp asc , LogicalServerName asc
```

LogicalServerName	LastSandboxTimestamp	ClusterName	NodeName	process_id	AppTypeName	AppName	code_package_version	HangDuration
blackbox3	2021-02-03 19:44:55.6210379	tr9.indiasouth1-a.worker.database.windows.net	DB.27	53148	Worker.PAL.PG	d85a9dcb4513	13.1.20201214.4-orcshr-a9250702	19:08:37.0937044
relcare-uat-database	2021-02-03 19:49:46.9311422	tr9.indiasouth1-a.worker.database.windows.net	DB.27	57704	Worker.PAL.PG	e2314e235673	13.1.20201214.4-orcshr-a9250702	19:14:33.0610601

```
MonLogin
| where logical_server_name == "blackbox3"
| where originalEventTimestamp > ago(time(19:08:37.0937044))
| summarize count(), min(originalEventTimestamp), max(originalEventTimestamp) by is_success, is_user_error,
 error, ['state'], result, event, AppTypeName
```

is_success	is_user_error	error	state	result	event	AppTypeName	count_	min_originalEventTimestamp	max_originalEventTimestamp

[NO ROWS]

NOTE: There might be MonLogin rows for new Host.PG connections, and there may be rows for failed connections or for closed connections from either or both Host.PG and Gateway.PG. None of those entries

indicate that new connections are being accepted to the backend. If there are exceptions in this step, e.g., entries from Worker.PAL.PG, contact RBartel for assistance.

Action Items (High Level):

The action steps of this TSG are:

1. First, create a dump so that the RC of the hang can be determined.
2. Confirm that the dump has been uploaded in AzureWatson.
3. Restart the server by killing DkMon.exe.
4. Determine that the server has recovered and is accepting connections. If the server has troubles restarting, then investigate those problems.
5. Copy the Dump URL to the incident. Mitigated the incident and assign to Richard Bartel to resolve the issue.

Step 1: Creating a dump for the running process (this does not restart the server).

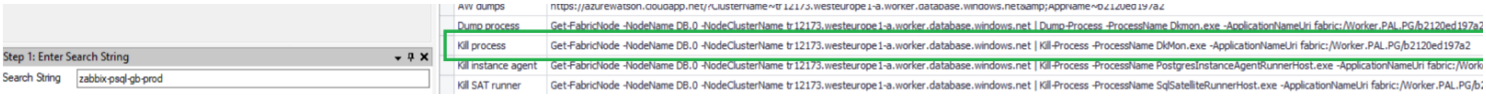
DO NOT USE THE DUMP-PROCESS CAS COMMAND, EVEN IF IT LOOKS LIKE THE COMMAND SUCCEEDS.

Use this link to create dump: SOP0207: Take a Manual CrashDump of PalWorker

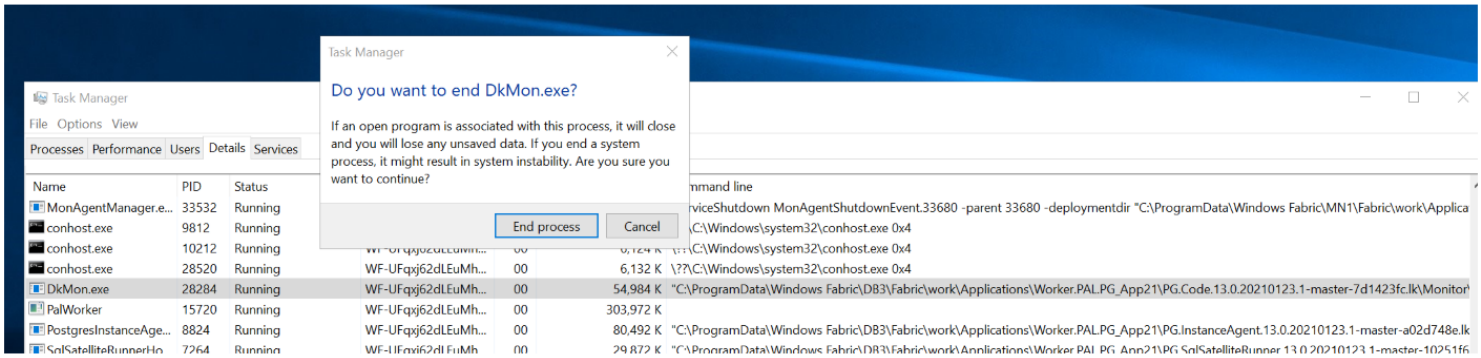
Step 2: Confirm that the dump has been uploaded in AzureWatson. Follow the steps here to locate the Dump in Azure Watson: SOP0208: Finding SqlPal dump in Azure Watson

Please understand that dump upload is not quick, and that the larger the dump the longer it will take Azure Watson to upload the dump. ServerLoop dumps are often large (10-35 GB or more) but not always. E.g., in the Kusto query example shown in SOP0207, the dump was 31 GB and upload took over 40 minutes to complete. In the SOP0208 example, a 850 MB dump was uploaded in 27 seconds.

Step 3: Restart the server by killing DkMon.exe. IMPORTANT: Go back to the "Verify the Symptom" and verify that the server has not been restarted for another incident. If it is still hung, then Kill DkMon.exe using the CAS command.



The Kill-Process CAS command is located in "orcasql/elastic servers and databases.xts"



Step 4: Determine that the server has recovered and is accepting connections. If the server has troubles restarting, then investigate those problems. This step should be self-explanatory. If you need advice on how to proceed, check out these TSGs:

- OSSRDBMS-Availability-005: Server is Unhealthy in Windows Fabric
- TSG: Elastic Server Restart: Too many restarts for postgresql server

Step 5: Copy the Dump URL to the incident. Mitigated the incident and assign to Richard Bartel to resolve the issue. When specifying the DumpURL, please make sure the URL points to a specific dump, and not to a dump bucket.

Use either of these forms: • The DumpUID URL: https://azurewatson.cloudapp.net/dumpuid/b81fbeef-297b-48a2-8c04-f644d973b25f, or • The Dump Details URL: https://azurewatson.cloudapp.net/dumps/3675147330