How to investigate a memory leak with CP4BA 23.0.1?

By NICOLAS PEULVAST

https://community.ibm.com/community/user/automation/blogs/nicolas-peulvast/2022/10/25/how-to-investigate-a-memory-leak

Target audience: CP4BA user with Cluster Administrator role
Estimated duration: 30 minutes

This blog post idea comes from the experience of a Memory Leak occurring on our Pre-Production of our ODM on Cloud offering, the 10th of May 2022.

The goal is to propose a solution to have some insight or idea of the possible root cause of a Memory Leak, occurring in the Cloud Pak for Business Automation.

Table of Contents

On this article, we are focusing on the ADS Runtime Pod but this solution works for all Pods based on the official Liberty base image.

Tips: in order to know if a Pod of the Cloud Pak is using the Liberty base image, you can run the following command with your own pod name:

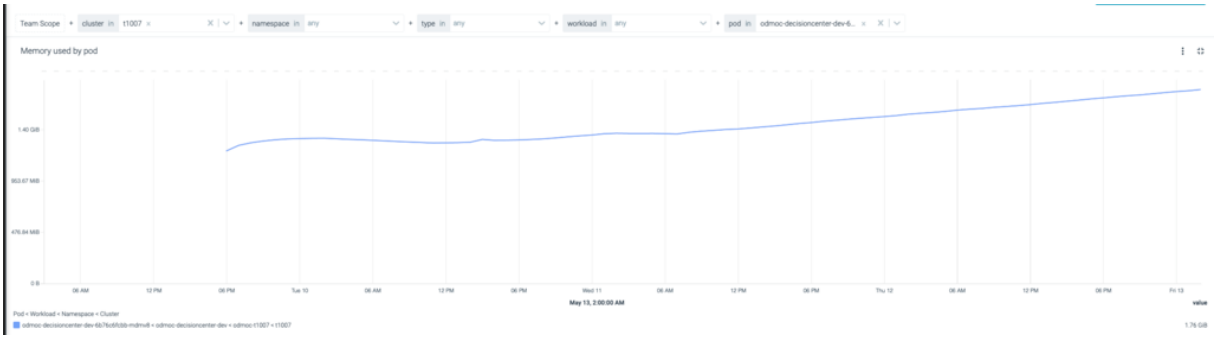
```
> oc exec -ti dba2301ads-ads-rest-api-6757765dd7-sf6w8 -- [ -d "/opt/ibm/wlp/usr/servers" ] && echo "yey" || echo "nope"
```

If `yey` appears, then you can continue this article.

If `nope` appears, then this article will not work for you ... but you can read it anyway! ;)

Context

The visible evidence of a Memory Leak is an increasing consumption of Memory across hours or days, sometime very slowly.



In our case, there was no special activity registered on the Pre-Production environment during the leak.

Procedure for the Heap Dump

As the memory consumption increases regularly, do two snapshots, separated at least from 3 hours or more if possible. You do not have to reach the OutOfMemory on your pod during this timeframe.

1- First log on on the Cluster using the `oc login` command.

2- Get the name of your Pod that revealed the memory leak.

```
> oc get pods | grep ads-runtime-service
dba2301ads-ads-runtime-service-8698b98f78-6vwkw 1/1 Running  911 (8d ago)       13d
dba2301ads-ads-runtime-service-8698b98f78-gz4k6 1/1 Running  911 (8d ago)       13d
```

Use the correct name for the other commands ...

3- Then, do a snapshot of your Heap using the following command line:

```
> oc exec -ti dba2301ads-ads-runtime-service-8698b98f78-6vwkw -- /opt/ibm/wlp/bin/s
erver dump --include=heap
Defaulted container "parsing-service" out of: parsing-service, tls-init (init)

Dumping server defaultServer.
```

```
Server defaultServer dump complete in /opt/ibm/wlp/output/defaultServer/defaultServ
er.dump-23.07.13_09.18.42.zip.
```

If you do not specify a server name, defaultServer is used. However, in some cases, you must specify the name of the server.

```
> oc exec -ti dba2301ads-ads-runtime-service-8698b98f78-6vwkw -- /opt/ibm/wlp/bin/s
erver dump ads-runtime --include=heap
Defaulted container "runtime-service" out of: runtime-service, tls-init (init), fol
der-prepare-container-ads (init)

Dumping server ads-runtime.
Server ads-runtime dump complete in /opt/ibm/wlp/output/ads-runtime/ads-runtime.dum
p-23.07.13_09.18.42.zip.
```

4- Copy the generated file from the pod to your local disk to ease the analysis.

```
> oc cp dba2301ads-ads-runtime-service-8698b98f78-6vwkw:/opt/ibm/wlp/output/default
Server/defaultServer.dump-23.07.13_09.18.42.zip defaultServer.dump-22.10.25_15.22.2
9.zip
Defaulted container "runtime-service" out of: runtime-service, tls-init (init)
tar: Removing leading `/' from member names
```

or

```
> oc cp dba2301ads-ads-runtime-service-8698b98f78-6vwkw:/opt/ibm/wlp/output/ads-run
time/ads-runtime.dump-23.07.13_09.18.42.zip ads-runtime.dump-22.10.25_15.22.29.zip
Defaulted container "runtime-service" out of: runtime-service, tls-init (init), fol
der-prepare-container-ads (init)
tar: Removing leading `/' from member names
```

5- Finally remove this important Zip file from the Pod to be sure that the Pod is not using too much disk storage and having the risk to be evicted from the Cluster.

```
> oc exec -ti dba2301ads-ads-runtime-service-8698b98f78-6vwkw -- rm /opt/ibm/wlp/ou
tput/defaultServer/defaultServer.dump-23.07.13_09.18.42.zip
Defaulted container "runtime-service" out of: runtime-service, tls-init (init)
```

or

```
> oc exec -ti dba2301ads-ads-runtime-service-8698b98f78-6vwkw -- rm /opt/ibm/wlp/ou
tput/ads-runtime/ads-runtime.dump-23.07.13_09.18.42.zip
```

```
Defaulted container "runtime-service" out of: runtime-service, tls-init (init), fol
der-prepare-container-ads (init)
```

Tip for Ephemeral Storage

To do the Snapshot, one must be careful to stay in the pre-defined ephemeral storage limits from the Pod, if defined.

If an eviction of your Pod occurs during the Snapshot, then you can remove this constraint.

1- First, scale down the CP4BA Operator:

```
> oc scale deployment ibm-cp4a-operator --replicas=0
deployment.apps/ibm-cp4a-operator scaled
```

And if needed, all sub-operator as well.

```
> oc scale deployment ibm-ads-operator --replicas=0
deployment.apps/ibm-ads-operator scaled
```

2- Finally remove the Ephemeral Storage constraint using the following command (we suppose that the jq command is available on your machine):

```
> oc get deployment dba2301ads-ads-runtime-service -o json | jq 'walk(if type=="obj
ect" then del(."ephemeral-storage") else . end)' | oc replace --force -f -
deployment.apps "dba2301ads-ads-runtime-service" deleted
deployment.apps/dba2301ads-ads-runtime-service replaced
```

When the Snapshot are done, you must scale up the CP4BA Operator.

```
> oc scale deployment ibm-cp4a-operator --replicas=1
deployment.apps/ibm-cp4a-operator scaled
```

And if needed, all sub-operator as well.

```
> oc scale deployment ibm-ads-operator --replicas=1
deployment.apps/ibm-ads-operator scaled
```

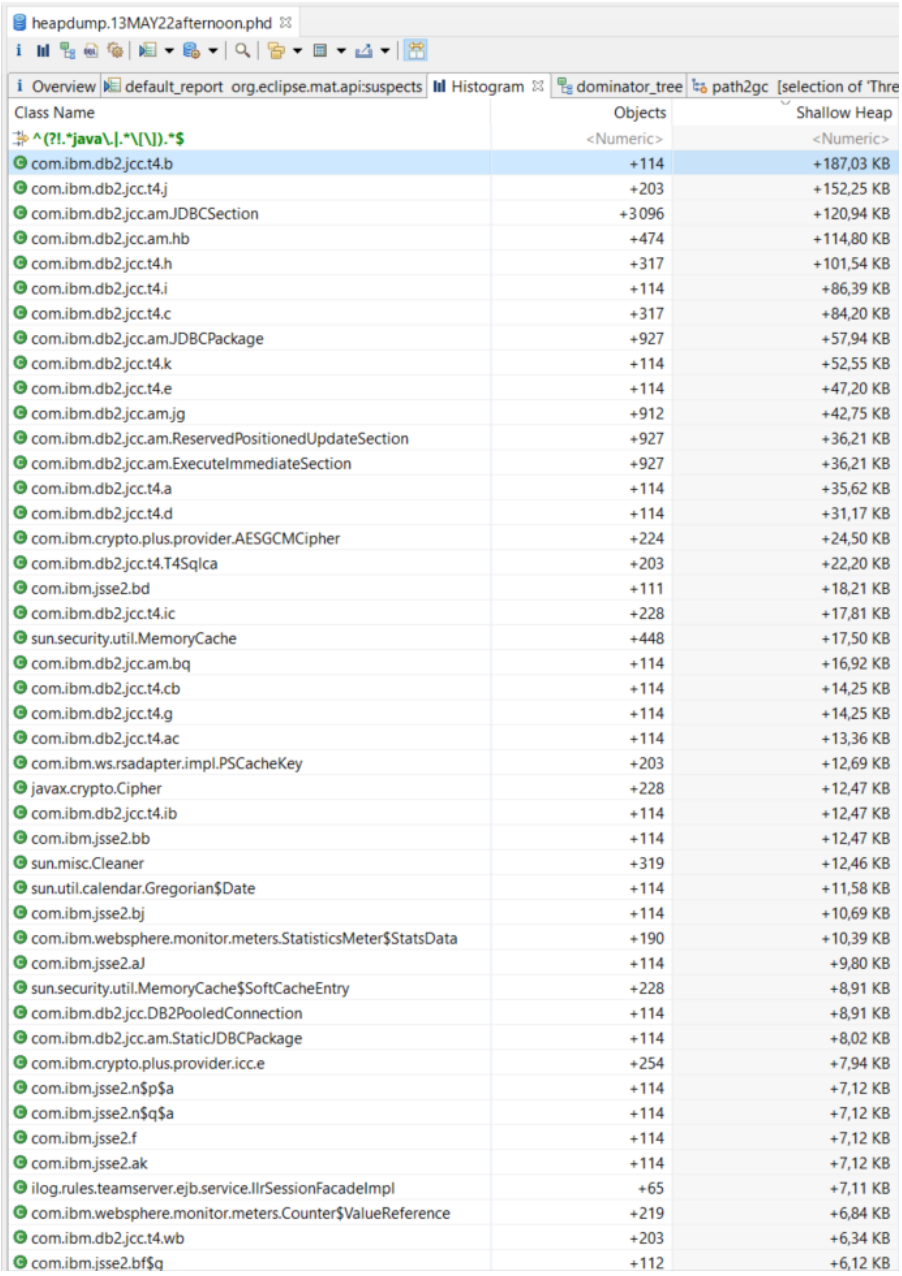
Analysis of the Heap Dumps

At the end of the Procedure described above, you'll obtain 2 Zips of Heap Dump that we have to compare to find the growing objects/classes.

To do that, do a diff using the Eclipse Heap Dump Analyzer named "Memory Analyzer (MAT)" : https://www.eclipse.org/mat/

You'll probably need an additional extension for the IBM JDK => https://www.ibm.com/support/pages/eclipse-memory-analyzer-tool-dtfj-and-ibm-extensions

In our case, the difference reveals the following Class Names:



heapdump.13MAY22afternoon.phd

Overview | default_report org.eclipse.mat.api:suspects | Histogram | dominator_tree | path2gc [selection of 'Thre

Class Name	Objects	Shallow Heap
^(?!.*java\.|.*\[\]).*$	<Numeric>	<Numeric>
com.ibm.db2.jcc.t4.b	+114	+187,03 KB
com.ibm.db2.jcc.t4.j	+203	+152,25 KB
com.ibm.db2.jcc.am.JDBCSection	+3 096	+120,94 KB
com.ibm.db2.jcc.am.hb	+474	+114,80 KB
com.ibm.db2.jcc.t4.h	+317	+101,54 KB
com.ibm.db2.jcc.t4.i	+114	+86,39 KB
com.ibm.db2.jcc.t4.c	+317	+84,20 KB
com.ibm.db2.jcc.am.JDBCPackage	+927	+57,94 KB
com.ibm.db2.jcc.t4.k	+114	+52,55 KB
com.ibm.db2.jcc.t4.e	+114	+47,20 KB
com.ibm.db2.jcc.am.jg	+912	+42,75 KB
com.ibm.db2.jcc.am.ReservedPositionedUpdateSection	+927	+36,21 KB
com.ibm.db2.jcc.am.ExecuteImmediateSection	+927	+36,21 KB
com.ibm.db2.jcc.t4.a	+114	+35,62 KB
com.ibm.db2.jcc.t4.d	+114	+31,17 KB
com.ibm.crypto.plus.provider.AESGCMCipher	+224	+24,50 KB
com.ibm.db2.jcc.t4.T4Sqlca	+203	+22,20 KB
com.ibm.jsse2.bd	+111	+18,21 KB
com.ibm.db2.jcc.t4.ic	+228	+17,81 KB
sun.security.util.MemoryCache	+448	+17,50 KB
com.ibm.db2.jcc.am.bq	+114	+16,92 KB
com.ibm.db2.jcc.t4.cb	+114	+14,25 KB
com.ibm.db2.jcc.t4.g	+114	+14,25 KB
com.ibm.db2.jcc.t4.ac	+114	+13,36 KB
com.ibm.ws.rsadapter.impl.PSCacheKey	+203	+12,69 KB
javax.crypto.Cipher	+228	+12,47 KB
com.ibm.db2.jcc.t4.ib	+114	+12,47 KB
com.ibm.jsse2.bb	+114	+12,47 KB
sun.misc.Cleaner	+319	+12,46 KB
sun.util.calendar.Gregorian$Date	+114	+11,58 KB
com.ibm.jsse2.bj	+114	+10,69 KB
com.ibm.websphere.monitor.meters.StatisticsMeter$StatsData	+190	+10,39 KB
com.ibm.jsse2.aJ	+114	+9,80 KB
sun.security.util.MemoryCache$SoftCacheEntry	+228	+8,91 KB
com.ibm.db2.jcc.DB2PooledConnection	+114	+8,91 KB
com.ibm.db2.jcc.am.StaticJDBCPackage	+114	+8,02 KB
com.ibm.crypto.plus.provider.icc.e	+254	+7,94 KB
com.ibm.jsse2.npa	+114	+7,12 KB
com.ibm.jsse2.nqa	+114	+7,12 KB
com.ibm.jsse2.f	+114	+7,12 KB
com.ibm.jsse2.ak	+114	+7,12 KB
ilog.rules.teamserver.ejb.service.IlrSessionFacadeImpl	+65	+7,11 KB
com.ibm.websphere.monitor.meters.Counter$ValueReference	+219	+6,84 KB
com.ibm.db2.jcc.t4.wb	+203	+6,34 KB
com.ibm.jsse2.bf$q	+112	+6,12 KB

And in this case, you can notice that the DB2 Driver is responsible (com.ibm.db2.*) and also that the Crypto layer (sun.security.*, com.ibm.crypto.* and javax.crypto.*) is also included in that issue.

You can either find the corresponding issue in your custom code or find a possible reason directly in the web, such as https://www.ibm.com/support/pages/apar/IJ39517 or involve the team in charge of this packages (ODM team, DB2 team or JDK team).