Summary of EFS/CSE OufOfMemory Problem.

CSE has a new functionality that went live in May 2019. This functionality leverages the current Websphere infrastructure as well as new infrastructure in CalCloud using Apache Webserver and Tomcat App server. The new infrastructure is as shown below.

Machine generated alternative text:
iiii iiiiiiiiiiiiiiiii iiiii iiiii iiiiiiiiiiiiiiiiiiiiiiiii iiiii iiiii iiiii iiiii iiii 
MS - CSE[Non-PROD] 
alCloud - EFS [NON-PROD] 
••••iiii 
••••iiii 
22/80/443 
••••iiiii' 
—calCloud - EFS [PROD & PFT] 
MS - CSEIPROD & PFT] 
80/u3 
••••iiii 
••••iiii 

The functionality is being used by Fresno county only at this point but in the future other counties will take advantage of this new functionality. Shortly after the functionality go-live users reported that they receive a timeout error when using the functionality. This issue can be replicated in PFT2 environment and was resolved by making a change to Apache Webserver and Tomcat App server on the CalCloud side. However, the changes applied in CalCloud (not applied to PROD yet) caused the WAS on OMS side to throw OutOfMemory.

Machine generated alternative text:
End 
users 
- CSEIPROD & PFT] 
OutOfMemory 
—calCloud - EFS [PROD & PFT] 
80/u3 
webs. 
mod_proxy 
18443 
increased heap to 
2GB x 2 JVM 

The test case is to run two iterations where in each iteration 30 concurrent users are using CSE (Online) application. The first iteration passes successfully but causes the heap to increase. None of the heap is release in between iterations. The functionality heavily uses the heap memory due to documents (10MB/session loaded to memory) that are being sent to an external entity (Docusign). During second iteration we start to see Out Of Memory errors. A heap dump was performed and we saw 24MB char objects. We noticed that WAS garbage collection (GC) is not clearing the heap even after more 10 hours. This seems to indicate that there is a memory leak in the application.

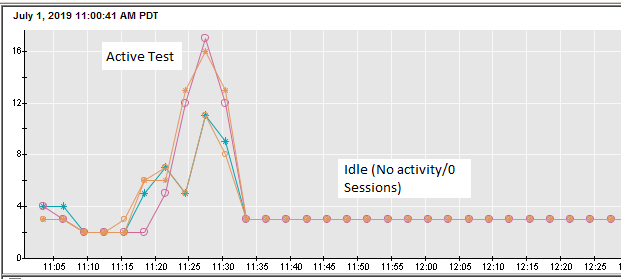
Troubleshooting Steps Performed

**WebSphere Config Changes**

1. modified the GC policy for Online Apps from GENCON to OPTTHRUPUT.
   1. This caused a different timeout that causes formset to be submitted without electronic signature. The out of memory error showed up again on the third iteration.
   2. Heap Analyzer shows the heap being taken by char objects.
   3. Garbage Collection (GC) seems like it is not releasing memory even without CSE activity. The GC decreased down to normal after a period of 12 hours.
2. Changed GC policy from OPTTHRUPUT to OPTAVGPAUSE still observed no change in heap usage, reverted the change to GENCON.
3. Modified threadpool for Online App from min 10 to min 0.

Reason to change: We have noticed 3 threads per app are being used even at 0 active sessions and these threads may be holding the char objects.

* 1. Even after the above change some threads are active at 0 sessions and never goes down below 3.



* 1. Heap usage remained the same, GC did not release the heap.

**IHS Plugin configuration changes:** *Load was not being distributed evenly among 4 apps.*

|  |  |
| --- | --- |
|  |  |
| IHS | LoadWeight from 2 to 20 |
| IHS | MaxConnections from Unlimited to 5 |

1. Load balancer had issues with these configurations and have been reverted.
2. Determine if GC is reclaiming heap memory after threads timeout. Wait 10 minutes between each iteration and ensure application servers aren’t restarted between iterations. Take heap dump manually between iterations.
   1. When the heap dump was taken manually the heap usage is reset and goes down.

