Troubleshoot Cisco Unified Intelligence Center (CUIC) Heap OutOfMemory issues because of OSCache

Contents

Introduction
What is OSCache?
What OSCache stores in CUIC?
Why OSCache causing HeapOutOfMemory issues?
How to limit the OSCache capacity?
What should be the cache.capacity value?
What are the limits imposed on cache.capacity on various CUIC releases:

Introduction

This document will describe how to troubleshoot CUIC Out of memory issue; because of oscache.properties.

Contributed by Venu Gopal Sane, Cisco Software Engineer.

What is OSCache?

OSCache is a Java framework developed by OpenSymphony that makes it easy to cache content in Web applications.
In CUIC using hibernate, it is configured to act as second level cache.

What OSCache stores in CUIC?

OScache stores report filters, result datasets, filter collections, filter collection values etc to reduce the IO on underlying database. When you run a report and then choose a couple of values (random)to filter it by (from the valuelist or collections), they too are stored as separate dynamic collections and collectionvalues. CUIC stores these individual dynamic collection values in the database as well as in the cache. These are however temporary in nature. Whenever the report is done running, the purge clears them out periodically.

Why OSCache causing HeapOutOfMemory issues?

At times, we found that the oscache entries had grown because of these collection values, before purge kicks in.
Typically observed that the number of items in the cache was 6-7 times that of total no of collections(including the temporary ones) in the database. **Oscache capacity size is unlimited by default**, so oscache size was growing in heap memory predominantly because of these

temporary collection values.

So following defect raised:
CSCuj26488 - CUIC Out of memory issue; oscache.properties (Found in 9.1(1) and fixed from 10.0(1) FCS onwards)

How to limit the OSCache capacity?

Setting a limit on capacity of the oscache would make it use the LRU implementation for cache and removes Least Recently Used (LRU) entries from cache automatically.

Limit cache.capacity in **/opt/cisco/cuic/cuicsrvr/webapps/cuic/WEB-INF/classes/oscache.properties** to a number instead of keeping it unlimited.

What should be the cache.capacity value?

Any value from 300,000 to 800,000.

Read below for more details:

When we observed the HeapOutOfMemory issues because of OsCache with Customers who has a large number of reports and filters, from heap analysis found following stats (From RBS Customer systems):
- 220000 collection values in database
- 7 times of this in cache i.e around: 1,540,000
- So taken a decision to limit the capacity value around the half of this number, so that cache holds: i.e 800000

That has resolved the OutOfMemory (OOM) issues because of oscache and the same limit is enforced in 10.x FCS releases.

Later on Cisco has not observed any issues with customers on 10.x releases where OutOfMemory issues are solely attributed to OsCache. Yes we do observed OOM issues even in 10.x releases, initially thought that it's because of oscache and further reduced the limit to 300000, but that does not make any difference and finally it is root caused that it is because of Virtual Memory hiking due to LD services being enabled.

So with that observation in mind and as the collection values in cache are not always at the magnitude of 1,540,000 for all customers; also as the OsCache capacity limitation makes use of LRU implementation, it is decided that we can further reduce the oscache limit to 300000.

What are the limits imposed on cache.capacity on various CUIC releases:

Currently, oscache capacity is limited as below:
- 9.1(1) FCS to COP5: Unlimited

- 10.x FCS releases: **800000**
- 9.1(1) COP6: **300000**

- 10.0(1) COP5: **300000**
- 10.5(1) ES03: Planning to limit to 300000 just because of the reasons as stated above and to maintain the uniform value across the releases, otherwise the current limitation of 800000 itself is sufficient enough.

From 11.0(1) onwards oscache is replaced with Hazelcast as second level cache for Hibernate to avoid the issues because of OSCache & Jgroups.