Title: Java memory usage much higher than heap+nonheap

Post Body:

I have a Java application (oracle JDK 8u191) with SpringBoot version 2.1.1.RELEASE and I'm experiencing a weird memory usage (don't thinks this is related to spring).

The sum of the HEAP and NONHEAP values provided by the application is much lower the real memory usage reported by Docker.

I know that Java has others memory areas than HEAP and NONHEAP, but I don't know how to measure them and if the current behavior is normal.

Curently my applications is reporting the following (after 3 days running in production):

```
              Init      Used      Committed      Max      HEAP:      250M      79M      250M      800M      NONHEA
```

The actuator endpoint `/metrics` for springBoot reports the same memory usage.

But the docker stats commands reports this (inside the container `smem` reports the same value for the java process):

`MEM USAGE / LIMIT 750.2MiB / 1.172GiB`

The difference between the heap+nonheap and the current used memory are `299,2mb`.

I was not expecting this 300mb the first time we put this service in production it caused a lot of OOM until we found the rigth configuration.

This 'unregistered memory usage' seems to only go up and never shrink, even after GC.

If I restart my aplication the difference between the heap+nonheap withe the real usage is about 100mb.

My Dockerfile starts the application this way:

```
ENTRYPOINT ['java', '-Xms250m', '-Xmx800m', '-XX:+UseG1GC', '-Djava.security.egd=file:/dev/./urandom', '-jar',
'/app.jar']
```

I also have a scheduled rotine that runs `System.gc()` every hour to shrink back the commited memory because this service have small spikes over the day and I want unused memory to be free for another services.

I want to know if this behavior is normal. And how can I measure what is using so much memory? visualvm and jconsole only show the heap and nonheap. Seems there is no metrics in springBoot actuator to measure this too.

How much memory should I left free in my container for this type of usage? I couldn't find any helpful article about it.

Edit1:

After 7 days, the difference between (heap+nonheap)-total became 550mb, leaving my container without any memory left.

Edit2:

Things starting making sense after digging into this articles:

Java process memory usage deviation from jcmd committed memory

Where do these java native memory allocated from?

Java process memory usage (jcmd vs pmap)

And indeed we are using a third party library with reports of memory leaks: `compile('axis:axis-wsdl4j:1.5.1')`

I'll try upgrade to Axis2 and will post the results here.

Accepted Answer: None
Highest Rated Answer: None