

Reducing memory consumption: from 1.5GB to 73MB

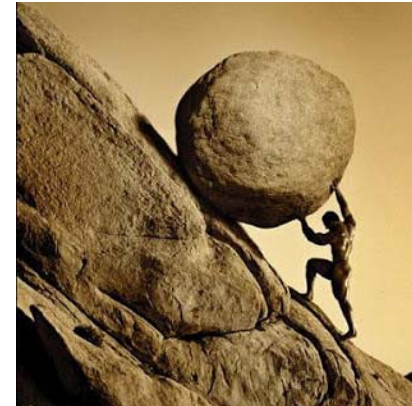
Andres Abad Rodriguez
CERN

Based on a true story....

- <http://plumbr.eu/blog/reducing-memory-consumption-by-20x>
- Read via @javahispano

Initial situation

- Crazy usage of memory > 1.5GB
 - Java application reporting OutOfMemoryError messages
 - Not memory leaks (Checked with **Plumbr**)



First issue: wrong library selection

- XMLBeans introduced a heavy XML schema

```
public class Person {  
    private String id;  
    private Date dateOfBirth;  
    private String forename;  
    private String surname;  
}
```

XMLBeans to cache
a 1.3 million people
database



**1.5GB of
heap
needed!!**

Solution: change the caching to use a simpler structure
java.util.HashMap<Long, Person> (*Id* as key and *Person*
object as value)

New memory usage: 1.5GB → 214MB

Second issue: improving structure

- As the keys in the Map were essentially numbers...could we improve this somehow?

Yes, by replacing the HashMap with a more optimized Map (TLongObjectHashMap from Trove collections)

Trove collections*: LGPL library. High performance collections in Java

New memory usage: 214MB → 143MB

* <http://trove.starlight-systems.com>

Third issue: redundant class data

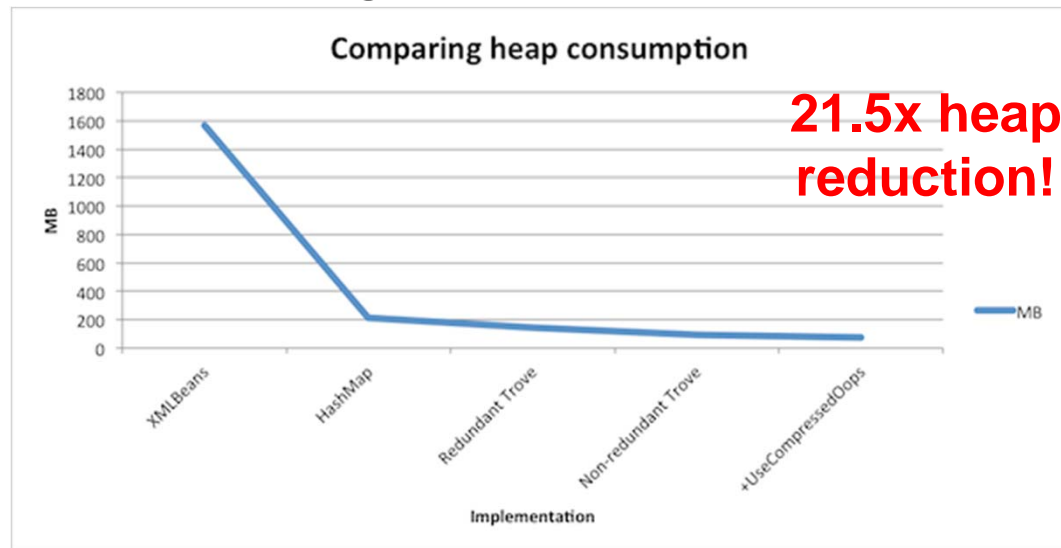
- Redundant piece of information in the class
 - Date Of Birth was actually encoded in the ID
 - Easy to calculate based on the ID
 - **Solution:** Remove the duplicated field

```
public class Person {  
    private String id;  
    private String forename;  
    private String surname;  
}
```

New memory usage: 143MB → 93MB

Fourth issue: improving the JVM config

- Old 64bit JVM running which did not compress ordinary object pointers* by default
 - Activated using `-XX:+UseCompressedOops`



New memory usage: 93MB → 73MB

* <https://wikis.oracle.com/display/HotSpotInternals/CompressedOops>

Lessons learned

- Benchmark the libraries
- Always ask yourself if there is a simpler structure that fits
- Remove redundancies
- Use the correct parameters in the compiler and in the virtual machine

Improvement results can be impressive!!!



That's all Folks!