1. Heap Statistics

Learn more about Heap Statistics

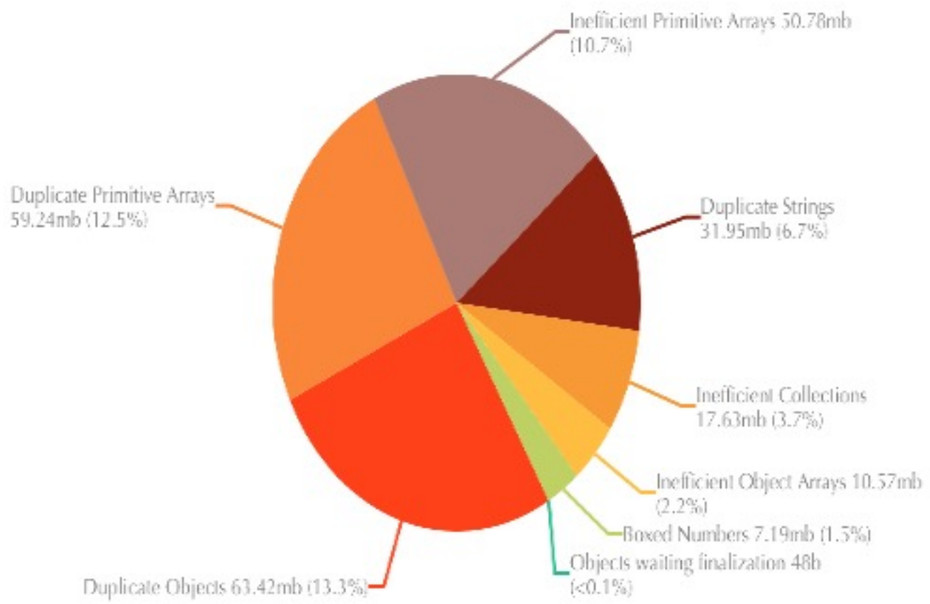
Total Size : 475.21mb

Class Count : 16,656

Object Count : 8,836,636

Thread Count : 69

Memory wasted: 240.78mb (51%)



2. What's in your Memory (by class)?

Learn more about What's in Memory

Class	Percentage	Size	Count
int[]	11.5%	54.56mb	35,125

byte[]	11.2%	53.42mb	59,609
char[]	10.5%	49.86mb	976,307
String	9.6%	45.8mb	489,589
j.u.RegularEnumSet$EnumSetIterator	7.9%	37.31mb	1,222,636
j.l.StringBuilder	7.8%	37.16mb	231,917

Show all records »

3. Large objects

Learn more about Large Objects

Name	Percentage	Size
Unreachable (garbage) objects	95.6%	454.35mb
Java Static java.lang.ApplicationShutdownHooks .hooks	1.5%	6.95mb
Java Static sun.launcher.LauncherHelper .scloader	0.5%	2.27mb
Java Static ch.qos.logback.core.util.ExecutorServiceUtil .THREAD_FACTORY	0.4%	1.72mb
Java Static com.hazelcast.spi.impl.operationservice.Operations .THIS_CLASS_LOADER	0.3%	1.62mb
.. and 20008 more objects retaining 6.77mb (1.4%)		

Show all records »

4. Object Headers

Learn more about Object Headers

Object Header Size	Total size of all headers
12b	101.13mb (21.3%)

Top Object Headers

Class	Percentage	Total header size	Avg obj size	Count
j.u.RegularEnumSet$EnumSetIterator	2.9%	13.99mb	32	1,222,636
char[]	2.4%	11.17mb	133	976,307
String	1.2%	5.6mb	24	489,589
java.util.concurrent.locks.AbstractQueuedSynchronizer$Node	1.0%	4.91mb	32	428,866
j.u.TreeMap$Entry	1.0%	4.6mb	40	401,807

Show all records »

How to fix excessive Object headers?

To see our recommendations, please purchase Enterprise Edition.

5. Duplicate Strings

Learn more about Duplicate Strings

Total Strings	Unique Strings	Wasted Memory
489,589	61,693	31.95mb (6.7%)

Top Duplicate Strings

Duplicate String	Percentage	Wasted	Count
""	0.3%	1.63mb	43,595
"java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject@1ee29264"	0.3%	1.38mb	7,248
"application"	0.2%	1,019.88kb	16,319
"hostname"	0.2%	892.01kb	16,312
"ebname"	0.2%	891.95kb	16,311

Show all records »

Who is holding Duplicate Strings?

Object Tree	Percentage	size
Unreachable (garbage) objects	2.5%	12,350K
java.lang.management.ThreadInfo.lockName	1.4%	6,876K
io.micrometer.core.instrument.ImmutableTag.key	0.4%	1,847K
j.u.StringJoiner.emptyValue	0.3%	1,600K
String[]	0.3%	1,295K

Show all records »

How to fix Duplicate Strings?

To see our recommendations, please purchase Enterprise Edition.

6. Inefficient collections

Learn more about Inefficient Collections

Total Collections	Inefficient collections	Wasted Memory

310,070

268,004

17.63mb (3.7%)

Top inefficient collections

Problem	Percentage	Wasted
91% of j.u.HashMap contains no elements	0.8%	3.97mb
18% of java.util.concurrent.ConcurrentHashMap contains 1 element only	0.5%	2.51mb
45% of j.u.ArrayList contains 2 - 4 elements only	0.5%	2.5mb
41% of j.u.ArrayList contains no elements	0.3%	1.64mb
89% of j.u.TreeSet contains no elements	0.3%	1.33mb

Show all records »

Who is holding Inefficient Collections?

Object Tree	Percentage	size
j.u.HashMap$EntrySet.this$0	0.6%	3,059K
Unreachable (garbage) objects	0.6%	3,024K
org.springframework.web.reactive.result.method.HandlerMethodArgumentResolverComposite.argumentResolverCache	0.4%	1,972K
Unreachable (garbage) objects	0.3%	1,283K
j.u.HashMap$EntryIterator.this$0	0.1%	699K

Show all records »

How to fix Inefficient Collections?

To see our recommendations, please purchase Enterprise Edition.

7. Inefficient Object Arrays

Learn more about Inefficient Object Arrays

Total Object Arrays	Inefficient Object Arrays	Wasted Memory
463,354	277,695	10.57mb (2.2%)

Top inefficient Object Arrays

Problem	Percentage	Wasted
60% of String[] contains no elements	0.8%	3.98mb
5% of Object[] contains half empty elements	0.2%	1,001.07kb
1% of Object[] contains no elements	0.2%	906.87kb

23% of String[] declared with 1 length	0.2%	805.36kb
7% of Object[] contains 1 element only	0.2%	803.06kb

Show all records »

Who is holding Inefficient Object Arrays?

Object Tree	Percentage	size
com.hazelcast.internal.metrics.impl.MetricDescriptorImpl.tags	0.6%	2,721K
com.hazelcast.internal.metrics.impl.MetricDescriptorImpl.tags	0.3%	1,275K
Unreachable (garbage) objects	0.2%	1,096K
Object[]	0.2%	913K
j.u.Spliterators$ArraySpliterator.array	0.2%	813K

Show all records »

How to fix Inefficient Object Arrays?

To see our recommendations, please purchase Enterprise Edition.

8. Inefficient Primitive Arrays

Learn more about Inefficient Primitive Arrays

Total Primitive Arrays	Inefficient Primi Arrays	Wasted Memory
1,091,968	147,618	50.78mb (10.7%)

Top inefficient Primitive Arrays

Problem	Percentage	Wasted
9% of byte[] contains lot of 0s	4.1%	19.25mb
5% of char[] contains no elements	2.4%	11.54mb
1% of char[] contains lot of 0s	2.2%	10.23mb
2% of int[] contains no elements	1.2%	5.73mb
42% of int[] contains lot of 0s	0.3%	1.21mb

Show all records »

Who is holding Inefficient Primitive Arrays?

Object Tree	Percentage	size
java.nio.HeapCharBuffer.hb	2.1%	9,984K
com.fasterxml.jackson.core.json.UTF8JsonGenerator._charBuffer	1.9%	9,456K
com.fasterxml.jackson.core.json.UTF8JsonGenerator._outputBuffer	1.9%	9,419K

Unreachable (garbage) objects	1.4%	6,678K
Unreachable (garbage) objects	0.9%	4,154K

Show all records »

How to fix Inefficient Primitive Arrays?

To see our recommendations, please purchase Enterprise Edition.

9. Boxed Numbers

Learn more about Boxed Numbers

Total Boxed Objects	Wasted Memory
398,039	7.19mb (1.5%)

Top Boxed Numbers

Object	Percentage	size	count
j.l.Long	1.1%	5.15mb	269,900
j.l.Integer	0.3%	1.6mb	105,137

java.lang.Integer	0.3%	1.8MB	100,187

Who is holding Boxed Numbers?

Object Tree	Percentage	size
Unreachable (garbage) objects	1.0%	5,074K
Unreachable (garbage) objects	0.3%	1,476K
Unreachable (garbage) objects	<0.1%	280K

How to fix Boxed Numbers?

To see our recommendations, please purchase Enterprise Edition.

10. Duplicate Objects

Learn more about Duplicate Objects

Total Duplicate Objects	Wasted Memory
2,177,296	63.42mb (13.3%)

Types of Duplicate Objects

Object	Percentage	Wasted	Duplicate Count
j.u.RegularEnumSet$EnumSetIterator	7.9%	37.31mb	1,222,633
java.util.concurrent.locks.AbstractQueuedSynchronizer$Node	2.8%	13.09mb	428,827
com.hazelcast.internal.metrics.impl.MetricsDictionary$Word	1.9%	9.09mb	397,047
j.u.AbstractList$Itr	0.8%	3.93mb	128,789

Top Duplicate Objects

Duplicate Object	Percentage	Wasted	Count
j.u.RegularEnumSet$EnumSetIterator (unseen : 0, lastReturned : 0, this$0 : j.u.RegularEnumSet@80f31aa0)	6.1%	28.95mb	948,722
java.util.concurrent.locks.AbstractQueuedSynchronizer$Node (waitStatus : 0, prev : null, next : null, thread : null, nextWaiter : null)	2.7%	13.03mb	426,947
j.u.RegularEnumSet$EnumSetIterator (unseen : 0, lastReturned : 1, this$0 : j.u.RegularEnumSet@80f31a20)	1.5%	7.31mb	239,448
j.u.AbstractList$Itr (cursor : 1, lastRet : 0, expectedModCount : 0, this$0 : j.u.Arrays$ArrayList@81326080)	0.8%	3.65mb	119,724
j.u.RegularEnumSet$EnumSetIterator (unseen : 0, lastReturned : 0, this$0 : j.u.RegularEnumSet@80f31698)	0.2%	1.05mb	34,466

Show all records »

Who is holding Duplicate Objects?

Object Tree	Percentage	size
Unreachable (garbage) objects		51.880K

unreachable (garbage) objects	10.7%	51,889K
{j.u.TreeMap}.values	1.9%	9,305K
j.u.Collections$UnmodifiableCollection$1.i	0.8%	3,741K

How to fix Duplicate Objects?

To see our recommendations, please purchase Enterprise Edition.

11. Duplicate Primitive Arrays

Learn more about Duplicate Primitive Arrays

Total Duplicate Arrays	Wasted Memory
342,162	59.24mb (12.5%)

Types of Duplicate Arrays

Array Type	Percentage	Wasted	Duplicate Count
char[]	5.8%	27.74mb	241,047

byte[]	5.7%	27.23mb	48,603
int[]	0.5%	2.33mb	31,778
short[]	0.3%	1.38mb	18,109
long[]	0.1%	549.88kb	2,502

Show all records »

Top Duplicate Arrays

Duplicate Array	Percentage	Wasted	Count
byte[8000] (123, 34, 115, 116, 97, 116, 117, 115, 34, 58, 34, 85, 80, 34, 125, 0, 0, 0, 0, 0, ...)	1.9%	9.23mb	1,208
char[4000] (, ...)	1.9%	9.23mb	1,208
char[69] (j, a, v, a, ., u, t, i, l, ., c, o, n, c, u, r, r, e, n, t, ...)	1.0%	4.52mb	29,640
byte[2493] (120, 1, -99, 88, 7, 123, 27, -57, 21, -100, -69, -61, 1, 60, 20, 118, 74, -106, -61, 56, ...)	0.9%	4.34mb	1,814
char[16] (, , , , , , , , , , , , , , ,)	0.5%	2.28mb	49,801

Show all records »

Who is holding Duplicate Arrays?

Object Tree	Percentage	size
Unreachable (garbage) objects	4.8%	23,324K
com.fasterxml.jackson.core.json.UTF8JsonGenerator._outputBuffer	1.9%	9,448K
com.fasterxml.jackson.core.json.UTF8JsonGenerator._charBuffer	1.9%	9,448K

java.io.ByteArrayOutputStream.buf	0.8%	3,744K
j.u.AbstractMap$SimpleImmutableEntry.value	0.8%	3,742K

Show all records »

How to fix Duplicate Arrays?

To see our recommendations, please purchase Enterprise Edition.

12. Objects waiting for Finalization

Learn more about Objects waiting for Finalization

Wasted Memory

48b (<0.1%)

What are the objects waiting for finalization?

To see objects waiting for finalization, click here.

How to fix objects waiting for finalization?

To see our recommendations, please purchase Enterprise Edition.

13. Threads

Learn more about Threads

To view all threads stacktrace click here.

14. Heap settings

Learn more about Heap Settings

No major recommendations.

15. System Properties

Learn more about System Properties

Key	Value
PID	3134
awt.toolkit	sun.awt.X11.XToolkit

awt.toolkit	sun.awt.X11.XToolkit
com.sun.management.jmxremote	
com.sun.management.jmxremote.authenticate	false
com.sun.management.jmxremote.port	9191

Show all records »