



执行命令 jcmd 进程pid help GC.class_stats后可以看到在执行命令jcmd 进程pid GC.class_stats的时候可以在命令后谜案添加的参数有哪些，在控制台中可以看到有个参数为-help。

class_stats 字段说明

我们可以使用命令jcmd 进程pid GC.class_stats -help查看执行命令jcmd 进程pid GC.class_stats得到的结果中每一个列名的含义

InstSize:

Size of each object instance of the Java class.

Java 类每个对象实例的大小。

InstCount:

Number of object instances of the Java class.

Java 类的对象实例数。

InstBytes:

This is usually (InstSize * InstNum). The only exception is java.lang.Class, whose InstBytes also includes the slots used to store static fields. InstBytes is not counted in ROAll, RWAll or Total.

InstSize * InstNum, 所有类实例的总大小。包含静态字段的大小，但是不包含 java.lang.Class 类。此数据不计入 ROALL、RWALL、Total（在本pdf最后可以看到）中。

Mirror:

Size of the Klass::java_mirror() object.

KlassBytes:

Size of the InstanceKlass or ArrayKlass for this class. Note that this includes VTab, ITab, OopMap.

K_secondary_supers:

Number of bytes used by the Klass::secondary_supers() array.

VTab:

Size of the embedded vtable in InstanceKlass.

ITab:

Size of the embedded itable in InstanceKlass.

OopMap:

Size of the embedded nonstatic_oop_map in InstanceKlass.

IK_methods:

Number of bytes used by the InstanceKlass::methods() array.

IK_method_ordering:

Number of bytes used by the InstanceKlass::method_ordering() array.

IK_default_methods:

Number of bytes used by the InstanceKlass::default_methods() array.

IK_default_vtable_indices:

Number of bytes used by the InstanceKlass::default_vtable_indices()

array.

IK_local_interfaces:

Number of bytes used by the InstanceKlass::local_interfaces()
array.

IK_transitive_interfaces:

Number of bytes used by the InstanceKlass::transitive_interfaces()
array.

IK_fields:

Number of bytes used by the InstanceKlass::fields() array.

IK_inner_classes:

Number of bytes used by the InstanceKlass::inner_classes() array.

IK_signers:

Number of bytes used by the InstanceKlass::signers() array.

class_annotations:

Size of class annotations.

类注解的大小

class_type_annotations:

Size of class type annotations.

class type 的注解大小

fields_annotations:

Size of field annotations.

字段注解大小

fields_type_annotations:

Size of field type annotations.

field type 注解大小

methods_annotations:

Size of method annotations.

方法注解大小

methods_parameter_annotations:

Size of method parameter annotations.

方法参数注解大小

methods_type_annotations:

Size of methods type annotations.

method type 注解大小

methods_default_annotations:

Size of methods default annotations.

方法默认注解大小

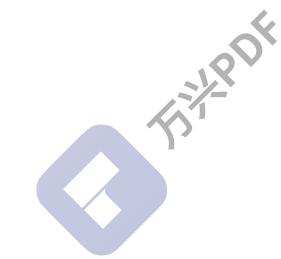
annotations:

Size of all annotations.

注解总数量

Cp:

Size of InstanceKlass::constants().



CpTags:

Size of InstanceKlass::constants()->tags().

CpCache:

Size of InstanceKlass::constants()->cache().

CpOperands:

Size of InstanceKlass::constants()->operands().

CpRefMap:

Size of InstanceKlass::constants()->reference_map().

CpAll:

Sum of Cp + CpTags + CpCache + CpOperands + CpRefMap.

MethodCount:

Number of methods in this class.

方法数量

MethodBytes:

Size of the Method object.

方法对象的大小

ConstMethod:

Size of the ConstMethod object.

ConstsMethod 对象大小

MethodData:

Size of the MethodData object.

MethodData 对象大小

StackMap:

Size of the stackmap_data.

stackmap 大小

Bytecodes:

Of the MethodBytes column, how much are the space taken up by bytecodes.
在 MethodBytes 列中，占用字节码的大小

MethodAll:

Sum of MethodBytes + Constmethod + Stackmap + Methoddata.

方法的总大小

ROAll:

Size of all class meta data that could (potentially) be placed in read-only memory. (This could change with CDS design).

可以放入只读内存的所有类元数据大小

RWAll:

Size of all class meta data that must be placed in read/write memory. (This could change with CDS design).

必须放入读写内存的类元数据大小

Total:

ROAll + RWAll. Note that this does NOT include InstBytes..

ROALL 和 RWALL 的总和，但是不包含 InstBytes

