jstat 是 jdk 自带的一个命令行工具，用于监控 java 虚拟机的统计信息，

通常可用它查看指定 java 进程的 gc 情况

jstat 的用法

要查看 jstat 的使用说明，可执行 jstat -help 命令，输出结果如下

```
Usage: jstat -help|-options
     jstat -<option> [-t] [-h<lines>] <vmid> [<interval> [<count>]]

 <option>     An option reported by the -options option
 <vmid>       Virtual Machine Identifier. A vmid takes the following form:
                 <lvmid>[@<hostname>[:<port>]]
             Where <lvmid> is the local vm identifier for the target
             Java virtual machine, typically a process id; <hostname> is
             the name of the host running the target Java virtual machine;
             and <port> is the port number for the rmiregistry on the
             target host. See the jvmstat documentation for a more complete
             description of the Virtual Machine Identifier.
 <lines>      Number of samples between header lines.
 <interval>   Sampling interval. The following forms are allowed:
                 <n>["ms"|"s"]
             Where <n> is an integer and the suffix specifies the units as
             milliseconds("ms") or seconds("s"). The default units are "ms".
 <count>      Number of samples to take before terminating.
 -J<flag>     Pass <flag> directly to the runtime system.
```

-option

jstat 支持多个选项（option），可通过 jstat -options 查看所有的选项

每个选项的功能说明如下

选项	显示信息说明
class	class loader 行为的统计信息
compiler	HotSpot JIT (Just In Time) 编译器行为的统计信息
gc	堆上垃圾回收行为的统计信息

选项	显示信息说明
gccapacity	分代容量及对应空间占用的统计信息
gccause	垃圾回收统计信息，和 -gcutil 类似，区别是带有最后一次及当前垃圾回收事件的原因
gcnew	新生代行为的统计信息
gcnewcapacity	新生代容量及对应空间占用的统计信息
gcold	老年代和永久代行为的统计信息
gcoldcapacity	老年代容量的统计信息
gcutil	垃圾回收的统计信息汇总
printcompilation	HotSpot 编译器统计信息
gcmetacapacity	元空间的容量统计信息
gcpermcapacity	永久代容量的统计信息

这些选项的输出列并不相同，要查看每一个选项的输出列含义，可以使用 man jstat 查看 jstat 的 man page

-t 在第一列显示时间戳，它是自虚拟机启动之后的时间

-h lines 每 lines 行输出后显示一次表头，默认值是 0，仅在第一行显示。例如 -h 3 表示每 3 行输出一次表头。

vmid 虚拟机标志。格式为 <lvmid>[@<hostname>[:<port>]]，其中 lvmid 通常是进程 ID，hostname 是虚拟机所在的主机名，port 是目标主机的远程注册端口。

interval 采样间隔，即每隔多长时间输出一次统计结果，格式为 <n>["ms"|"s"]，默认单位为 ms。

count 采样次数，即输出多少次统计结果。不设置采样间隔时，默认只输出一次。

-J flag 传递给 java 虚拟机的参数。例如 -J-Xms48m 设置虚拟机的初始堆内存为 48m。

示例

如下是一些示例，监控的是本地 local 上的 java 虚拟机。

查看垃圾回收汇总信息

使用 -gcutil 选项，查看 lvmid 为 21891 的 java 虚拟机垃圾回收信息，每隔 250 ms 输出一次统计信息，共输出 7 次。执行如下命令：

```
jstat -gcutil 21891 250 7
```

输出结果为：

```
  S0     S1     E      O      P     YGC     YGCT    FGC    FGCT     GCT
 12.44   0.00  27.20   9.49  96.70    78    0.176     5    0.495    0.672
 12.44   0.00  62.16   9.49  96.70    78    0.176     5    0.495    0.672
 12.44   0.00  83.97   9.49  96.70    78    0.176     5    0.495    0.672
  0.00   7.74   0.00   9.51  96.70    79    0.177     5    0.495    0.673
  0.00   7.74  23.37   9.51  96.70    79    0.177     5    0.495    0.673
  0.00   7.74  43.82   9.51  96.70    79    0.177     5    0.495    0.673
  0.00   7.74  58.11   9.51  96.71    79    0.177     5    0.495    0.673
```

其中每一列的含义为：

列	描述
S0	Survivor 0 的空间占用百分比
S1	Survivor 1 的空间占用百分比
E	Eden 区的空间占用百分比
O	老年代（Old）的空间占用百分比
P	永久代（Permanent）的空间占用百分比
YGC	YGC 次数
YGCT	YGC 占用时间
FGC	Full GC 次数

列	描述
FGCT	Full GC 占用时间
GCT	GC 占用的总时间

从中可以看出

1. 在第 3 次统计和第 4 次统计之间 YGC 次数从 78 增加到 79，表示出现了一次 YGC
2. YGC 消耗时间为 0.177 - 0.176 = 0.001 s
3. Eden 区和 S0 区的存活对象复制到了 S1 区，Survivor 区的空间占用从 12.44% 降到了 7.74%
4. 部分对象晋升到老年代（Old），老年代的空间占用从 9.49% 上升到 9.51

查看新生代的统计信息

使用 -gcnew 选项，查看 lvmid 为 21891 的 java 虚拟机的新生代的垃圾统计信息，每 3 行输出后显示一次表头，共输出 250 次统计信息。执行如下命令：

```
jstat -gcnew -h3 21891 250
```

输出结果为：

```
S0C    S1C    S0U    S1U   TT MTT  DSS      EC       EU     YGC    YGCT
 64.0   64.0    0.0   31.7 31  31   32.0    512.0    178.6   249    0.203
 64.0   64.0    0.0   31.7 31  31   32.0    512.0    355.5   249    0.203
 64.0   64.0   35.4    0.0  2  31   32.0    512.0     21.9   250    0.204
S0C    S1C    S0U    S1U   TT MTT  DSS      EC       EU     YGC    YGCT
 64.0   64.0   35.4    0.0  2  31   32.0    512.0    245.9   250    0.204
 64.0   64.0   35.4    0.0  2  31   32.0    512.0    421.1   250    0.204
 64.0   64.0    0.0   19.0 31  31   32.0    512.0     84.4   251    0.204
S0C    S1C    S0U    S1U   TT MTT  DSS      EC       EU     YGC    YGCT
 64.0   64.0    0.0   19.0 31  31   32.0    512.0    306.7   251    0.204
```

其中每一列的含义为：

列	描述
S0C	当前 Survivor 0 区的容量（KB）

列	描述
S1C	当前 Survivor 1 区的容量（KB）
S0U	当前 Survivor 0 区的使用情况（KB）
S1U	当前 Survivor 1 区的使用情况（KB）
TT	晋升老年代的阈值
MTT	晋升老年代的最大阈值
DSS	期望的 survivor 区大小
EC	当前 Eden 区的容量（KB）
EU	Eden 区的占用情况（KB）
YGC	YGC 事件次数
YGCT	YGC 占用的时间

这里可以看出：

1. 在第 2 次和第 3 次统计之间 YGC 次数从 249 增加到 250，表示出现了一次 YGC
2. YGC 消耗时间为 0.204 - 0.203 = 0.001 s
3. 垃圾收集器发现，S0 区的空间占用将会超过 DSS，因此会把对象晋升到老年代（该输出并未显示），TT 值从 31 降到了 2