

## 使用jvisualvm监控Spark作业

jvisualvm工具JDK自带的一个监控工具，该工具是用来监控java运行程序的cpu、内存、线程等的使用情况，并且使用图表的方式监控java程序、还具有远程监控能力，不失为一个用来监控java程序的好工具。

同样，我们可以使用jvisualvm来监控Spark应用程序（Application），从而可以看到Spark应用程序堆，线程的使用情况，从而根据这些数据去优化您的程序。废话少说，下面来说说怎么配置，使得我们可以使用jvisualvm来监控程序：

- 1、在\$SPARK\_HOME/conf目录下配置spark-default.conf文件，加入如下配置：

```
spark.driver.extraJavaOptions -Dcom.sun.management.jmxremote.port=1099  
-Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false
```

jmx默认的是1099端口，上面没有使用到ssl；并且不需要登录即可建立Jmx连接。

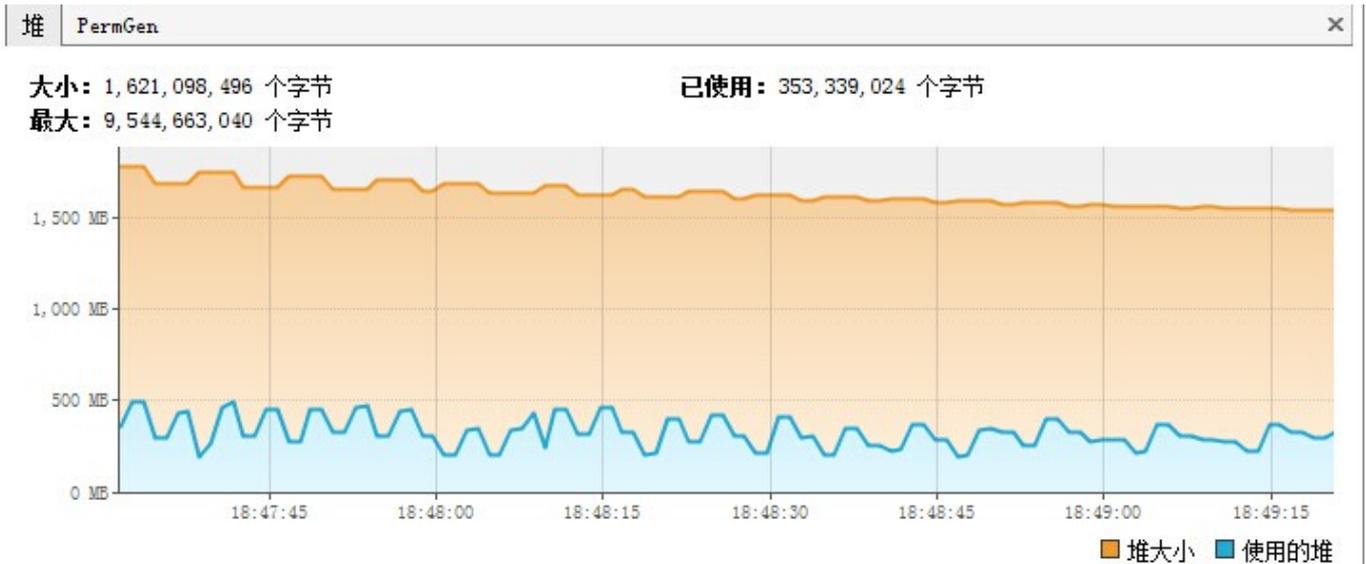
在前面的文章（[《Spark Metrics配置详解》](#)）中，我们介绍了Spark的几种Metrics，其实这里我们也可以使用Metrics来配置JMX，我们只要在\$SPARK\_HOME/conf目录下配置metrics.properties，如下：

```
*.sink.jmx.class=org.apache.spark.metrics.sink.JmxSink
```

这个在1099端口上开启JMX，目前貌似不支持端口的配置，以及SSL、authenticate等配置，其他的都和上面的一样。

- 2、启动你的Spark应用程序
- 3、打开jvisualvm.exe监控你的Spark应用程序

在JDK安装的bin目录下有个jvisualvm.exe，双击它，然后进行配置，依次选择 文件->添加JMX连接，然后会弹出一个对话框，在连接文本框里面填上你Driver机器的地址和端口，比如::1099。然后就可以看到如下的画面：



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：[iteblog\\_hadoop](https://www.iteblog.com)

里面可以看到很多关于你Spark应用程序的系信息，比如JVM配置和系统参数（其实这些在你应用程序的4040端口看到这些信息）。

在Java安装目录下的bin文件夹还有个jconsole.exe程序，这个使用和jvisualvm.exe使用差不多，只不过jvisualvm.exe图形漂亮很多。当然，还有很多工具可以监控Java程序的，比如jprofiler，这个功能更强大，但是配置起来也比较麻烦，而且这款软件是收费的。关于如何使用我就不介绍了。

**本博客文章除特别声明，全部都是原创！**  
**原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。**  
**本文链接: [【】（）](#)**