# arthas 使用教程

作者：Gakkiyomi2019
原文链接：https://ld246.com/article/1617011164563
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20171222.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="arthas-使用教程">arthas 使用教程</h2>
<blockquote>
Arthas 是 Alibaba 开源的 Java 诊断工具，通过这个工具我们可以快速并准确的定位问题。
这里会简单介绍使用 arthas 进行 Troube Shooting，详细安装流程与深入的操作命令参数请自
查阅官方文档学习。:pray:
</blockquote>
官方文档: <a href="https://ld246.com/forward?goto=https%3A%2F%
Farthas.aliyun.com%2Fdoc%2Findex.html" target="_blank" rel="nofollow ugc">https://arthas.
liyun.com/doc/index.html</a>
<h2 id="资源监控">资源监控</h2>
arthas 提供了 CPU，内存，运行环境等资源信息面板,并且可以设置刷新时间
<h3 id="命令">命令</h3>
<h4 id="dashboard">dashboard</h4>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/03/image-87b8067c.png?imageView2/2/interlace/1/format/jpg"></
>
<h4 id="heapdump">heapdump</h4>
dump java heap, 类似 jmap 命令的 heap dump 功能。
查看 java 堆内存的对象实例情况
<code>heapdump /arthas-output/nap.hprof</code> 将堆的使用情况 dump 到 /arthas-o
tput/nap.hprof 文件
可以使用 jdk 自带的 <code>VisualVM</code> 来查看
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/03/image-fa22cc7e.png?imageView2/2/interlace/1/format/jpg"></
>
也可以使用 IntelliJ IDEA 自带的 HPROF 内存查看器 查看 输入 Action <code>Open profiler
napshot</code>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/03/image-b5ed4494.png?imageView2/2/interlace/1/format/jpg"><
p>
<h4 id="Profiler">Profiler</h4>
<blockquote>
arthas 支持监控一段时间的火焰图来统计性能热点。
</blockquote>
<code>profiler start</code> 启动搜集 CPU 的使用情况
<code>profiler stop</code> 从 start 到 stop 这个时间段的 CPU 火焰图
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2021/03/image-3ae478b4.png?imageView2/2/interlace/1/format/jpg"></
>
火焰图里，X 轴越长,代表使用的越多,Y 轴是调用堆栈信息。当前收集的是
么类型的数据，比如 cpu 那么 x 轴长度越大,占用的 cpu 资源就越多。
<h4 id="thread">thread</h4>
<blockquote>
查看当前线程信息，查看线程的堆栈
</blockquote>
<table>
<thead>
<tr>
<th>参数名称</th>
<th>参数说明</th>
</tr>
</thead>

<tbody>
<tr>
<td>id</td>
<td>线程 id</td>
</tr>
<tr>
<td>[n:]</td>
<td>指定最忙的前 N 个线程并打印堆栈</td>
</tr>
<tr>
<td>[b]</td>
<td>找出当前阻塞其他线程的线程</td>
</tr>
<tr>
<td>[i <code><value></code>]</td>
<td>指定 cpu 使用率统计的采样间隔，单位为毫秒，默认值为 200</td>
</tr>
<tr>
<td>[--all]</td>
<td>显示所有匹配的线程</td>
</tr>
</tbody>
</table>
查看当前最吃资源的线程

/b3logfile.com/file/2021/03/image-1a574aff.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="方法调用时间追踪">方法调用时间追踪</h2>
<h3 id="命令-">命令</h3>
<h4 id="trace">trace</h4>
<blockquote>
使用 trace 命令可以针对某一类的某一方法进行方法调用时间的追踪,精确定位到问题所在。 
使用建议： 
<ol>
<li>首先使用 thread 命令查看当前高占用线程堆栈定位到方法</li>
<li>然后使用 trace 监控方法进行用时追踪</li>
</ol>
</blockquote>

/b3logfile.com/file/2021/03/image-99aeabcc.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="在线热更新">在线热更新</h2>
arthas 允许我们修改一个正在运行的字节码文件，也允许我们从外部加载一个新的 claas 文件来盖旧的。这样我们可以实现在线热部署，这样可以临时性的解决客户生产环境产生的问题,例如空指针常 :joy:。 
<h3 id="命令--">命令</h3>
jad/mc/retransform 组合可实现热更新
<h4 id="jad">jad</h4>
反编译代码，将已加载的字节码文件反编译到容器内文件夹。现场操作人员可以直接修复问题(果问题够简单)
<code>jad --source-only net.skycloud.cmdb.resource.api.rest.ResourceSearchController
gt; /tmp/ResourceSearchController.java</code>

/b3logfile.com/file/2021/03/image-4d8bf68d.png?imageView2/2/interlace/1/format/jpg"></

>
修改反编译出来的 class 文件 (除法分母不能为 0)
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2021/03/image-d2c05e85.png?imageView2/2/interlace/1/format/jpg"></>
<h4 id="mc">mc</h4>
Memory Compiler/内存编译器，编译 <code>.java</code> 文件生成 <code>.class</code>。 这也是我们可以从 外部加载 java 文件(打补丁) 的基础。
若前方技术支持无法解决，可以让后端伙伴进行 bug 修复，然后传输修复后的 java 文件来以解眉之急。不需要重新生成镜像。
<code>mc /tmp/ResourceSearchController.java -d /tmp</code>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2021/03/image-72adb642.png?imageView2/2/interlace/1/format/jpg">
<h4 id="retransform">retransform</h4>
加载外部的 <code>.class</code> 文件，retransform(重新转换) jvm 已加载的类。
<code>retransform /tmp/net/skycloud/cmdb/resource/api/rest/ResourceSearchControllr.class</code>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2021/03/image-6bc39bcd.png?imageView2/2/interlace/1/format/jpg"></>
<h5 id="查看已更新的类">查看已更新的类</h5>
<code>retransform -l</code>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2021/03/image-c15e45a5.png?imageView2/2/interlace/1/format/jpg"></>
<h5 id="恢复原来的类">恢复原来的类</h5>
这两个命令一起执行
<code>retransform --deleteAll</code>
<code>retransform --classPattern net.skycloud.cmdb.resource.api.rest.ResourceSearchCntroller</code>
<h2 id="Troube-Shooting">Troube Shooting</h2>
作为在现场的工作人员需要至少需要掌握以上命令来进行诊断。troube shooting 流程如下。</>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2021/03/image-d3427379.png?imageView2/2/interlace/1/format/jpg"></>