

# Arthas

## 一、基本信息介绍

Arthas 是一款命令行诊断工具，用于诊断Java应用程序问题，提供了包括\*累加载信息\*，\*方法执行耗时\*，\*线程堆栈\*，\*内存分析\*等功能。

「小组件」无法展示，请点击[「https://miworkpro.mioffice.cn/doc」](https://miworkpro.mioffice.cn/doc)查询原私有化文档，查看该部分内容

用户发出指令请求，通过 arthas客户端 推送到 arthas服务端，此时建立连接，服务端向客户端返回视图信息，客户端向服务端推送指令。对于每一条指令，如查看某一方法返回值，用户会先推送指令，然后服务端与指定进程建立连接(`loadAgent`)，然后使用 ASM 修改指定类的某一方  
法，动态修改 `.class`，做功能增强，随后方法的返回值信息将形成一定格式返回到 arthas服  
务端，服务端返回到 arthas客户端 做展示。

- **ASM**: 字节码操作类库，可以动态修改 `.class` 文件，修改Java类功能。
- `VirtualMachine#loadAgent` : 将一个进程挂载到目标进程

## 二、基本用法

- `cls` : 清屏
- `help` : 常看指令
- `cat` : 打印文件内容
- `echo` : 输出
- `grep` : 过滤
- `history` : 历史
- `keymap` : 快捷键
- `pwd` : 返回当前工作目录

• **quit** : 退出

• **reset** : 重置所有类

## 2.1 看板与监控类

### 2.1.1 dashboard

#### 实时数据看板

ID	NAME	GROUP	PRIORITY	STATE	%CPU	DELTA_TIME	TIME	INTERRUPTED	DAEMON
2	Reference Handler	system	10	WAITING	0.0	0.000	0:0.015	false	true
3	Finalizer	system	8	WAITING	0.0	0.000	0:0.000	false	true
4	Signal Dispatcher	system	9	RUNNABLE	0.0	0.000	0:0.000	false	true
5	Attach Listener	system	5	RUNNABLE	0.0	0.000	0:0.015	false	true
387	arthas-timer	system	5	WAITING	0.0	0.000	0:0.000	false	true
391	arthas-NettyHttpTelnetBootstrap-3-1	system	5	RUNNABLE	0.0	0.000	0:0.031	false	true
392	arthas-NettyWebsocketTtyBootstrap-4-1	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
393	arthas-NettyWebsocketTtyBootstrap-4-2	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
394	arthas-shell-server	system	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
395	arthas-session-manager	system	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
398	arthas-NettyHttpTelnetBootstrap-3-2	system	5	RUNNABLE	0.0	0.000	0:0.156	false	true
403	arthas-command-execute	system	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
Memory									
heap	used	total	max	usage	GC	gc.ps_scavenge.count	28		
ps_eden_space	546M	1689M	7202M	7.59%		gc.ps_scavenge.time(ms)	230		
ps_survivor_space	378M	1194M	2623M	14.42%		gc.ps_marksweep.count	4		
ps_old_gen	34M	35M	35M	99.98%		gc.ps_marksweep.time(ms)	445		
nonheap	133M	468M	5401M	2.48%					
code_cache	229M	238M	-1	96.15%					
metaspace	66M	67M	240M	27.84%					
compressed_class_space	145M	153M	-1	95.06%					
direct	17M	18M	1024M	1.66%					
mapped	8K	8K	-	400.00%					
Runtime									
os.name					Windows 11				
os.version					10.0				
java.version					1.8.0_422				
java.home					C:\Users\mi\.jdks\corretto-1.8.0_422\jre				
systemload.average					-1.00				
processors					22				
timestamp/uptime					Wed Feb 19 17:06:34 CST 2025/399s				
内存状态									
系统信息									

### 2.1.2 memory

#### 内存情况

Memory	used	total	max	usage
heap	786M	1689M	7202M	10.93%
ps_eden_space	618M	1194M	2623M	23.57%
ps_survivor_space	34M	35M	35M	99.98%
ps_old_gen	133M	468M	5401M	2.48%
nonheap	230M	239M	-1	96.05%
code_cache	67M	67M	240M	28.01%
metaspace	146M	153M	-1	94.97%
compressed_class_space	17M	18M	1024M	1.66%
direct	8K	8K	-	400.00%
mapped	8K	8K	-	0.00%

- Heap: 堆内存

\* ps\_eden\_space : Eden区是年轻代的一部分，对象首先在 Eden 区分配，满时会触发 MinorGC，存活的对象分配到 Survivor 区。

- \* `ps_survivor_space` : Survivor区也是年轻代的一部分，分为From和To两个区域，在 `MinorGC` 时，对象会在这两个区域内移动，当对象年龄增长，会移动到老年代
- \* `ps_old_gen` : 用于存储生命周期较长的对象，当年轻代对象存活时间超过一定阈值时，会晋升为老年代，老年代大小较大，满的时候会触发 `Full GC`

- No Heap: 非堆内存

- \* `metaspace` : 元空间，用于存储类的元数据，类的结构和方法，不属于堆
- \* `compressed_class_space` : 压缩类空间
- \* `code_cache` : 代码缓存，用于存储JIT编译器编译的热点代码
- \* `direct` : 直接内存，通常用于 `NIO` 的 `ByteBuffer` 操作
- \* `mapped` : 映射内存，通过内存映射加载到JVM的文件或者其他资源

## 2.1.3 JVM

### JVM虚拟机信息

## 2.1.4 thread

### 线程信息

ID	NAME	GROUP	PRIORITY	STATE	%CPU	DELTA_TIME	TIME	INTERRUPTED	DAEMON
2	Reference Handler	system	10	WAITING	0.0	0.000	0:0.015	false	true
3	Finalizer	system	8	WAITING	0.0	0.000	0:0.000	false	true
4	Signal Dispatcher	system	9	RUNNABLE	0.0	0.000	0:0.000	false	true
5	Attach Listener	system	5	RUNNABLE	0.0	0.000	0:0.015	false	true
387	arthas-timer	system	5	WAITING	0.0	0.000	0:0.000	false	true
391	arthas-NettyHttpTelnetBootstrap-3-1	system	5	RUNNABLE	0.0	0.000	0:0.031	false	true
392	arthas-NettyWebSocketTtyBootstrap-4-1	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
393	arthas-NettyWebSocketTtyBootstrap-4-2	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
394	arthas-shell-server	system	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
395	arthas-session-manager	system	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
398	arthas-NettyHttpTelnetBootstrap-3-2	system	5	RUNNABLE	0.0	0.000	0:0.281	false	true
403	arthas-command-execute	system	5	RUNNABLE	0.0	0.000	0:0.031	false	true
23	commons-pool-evictor	main	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
25	logback-1	main	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
27	com.alibaba.nacos.client.identify.watc	main	5	TIMED_WAITIN	0.0	0.000	0:0.187	false	true
29	com.alibaba.nacos.client.Worker	main	5	TIMED_WAITIN	0.0	0.000	0:0.046	false	true
30	com.alibaba.nacos.client.Worker	main	5	WAITING	0.0	0.000	0:0.000	false	true
31	com.alibaba.nacos.client.Worker	main	5	WAITING	0.0	0.000	0:0.000	false	true
32	nacos.publisher-com.alibaba.nacos.comm	main	5	WAITING	0.0	0.000	0:0.000	false	true
34	com.alibaba.nacos.client.remote.worker	main	5	WAITING	0.0	0.000	0:0.000	false	true
35	com.alibaba.nacos.client.remote.worker	main	5	TIMED_WAITIN	0.0	0.000	0:0.375	false	true
36	grpc-nio-worker-ELG-1-1	main	5	RUNNABLE	0.0	0.000	0:0.015	false	true
38	grpc-nio-worker-ELG-1-2	main	5	RUNNABLE	0.0	0.000	0:0.546	false	true
53	com.alibaba.nacos.client.Worker	main	5	WAITING	0.0	0.000	0:0.000	false	true
55	com.alibaba.nacos.client.Worker	main	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	true
58	Abandoned connection cleanup thread	main	5	TIMED_WAITIN	0.0	0.000	0:0.031	false	true
59	Catalina-utility-1	main	1	TIMED_WAITIN	0.0	0.000	0:0.078	false	false
60	Catalina-utility-2	main	1	WAITING	0.0	0.000	0:0.125	false	false
61	container-0	main	5	TIMED_WAITIN	0.0	0.000	0:0.000	false	false
62	com.alibaba.nacos.client.Worker	main	5	WAITING	0.0	0.000	0:0.000	false	true

## 2.1.5 getstatic

## 获取类的静态方法

## 代码块

## 1 getstatic class\_module field

- 推荐使用Ognl表达式

## 2.1.6 vmtool

JVM工具

- **查询内存对象**

## 代码块

```
1   vmtool --action getInstances --className java.lang.String --limit 1000 | grep  
'MI'
```

```

[arthas@23606]$ vmtool --action getInstances --class Name.java.lang.String --limit 4000 | grep "MI"
@String["resultCode":200,"errorCode":0,"serviceInfo":{"name":"providers:com.mi.oa.hr.feedback.api.service.FeedbackRPCService::","groupName":"DEFAULT_GROUP","clusters":"","cacheMillis":60000,"hosts":[{"ip":"10.167.179.93","port":20880,"weight":1.0,"healthy":true,"enabled":true,"ephemeral":true,"clusterName":"DEFAULT","serviceName":"DEFAULT_GROUP@providers:com.mi.oa.hr.feedback.api.service.FeedbackRPCService::","marked":false,"metadata":{"side":"provider","service.name":"ServiceBean:com.mi.oa.hr.feedback.api.service.FeedbackRPCService","methods":"uploadFile,sendToLark,sendToMeego,getFileByFdsObjectByName,save,screenshot","release":"2.7.24-XIAOMI","deprecated":false,"dubbo":"2.0.2","pid":1,"interface":"com.mi.oa.hr.feedback.api.service.FeedbackRPCService","generic":false,"retries":1,"path":"com.mi.oa.hr.feedback.api.service.FeedbackRPCService","protocol":"dubbo","meta-data-type":remote,"application":feedback,"dynamic":true,"service.filter":errorcodeExceptionFilter,"exception":exception,"category":providers,"anyhost":true,"timestamp":1736133948779}],"valid":true,"instanceHeartBeatTimeout":15000,"ipDeleteTimeOut":30000,"instanceHeartBeatInterval":5000}],"lastRefTime":174002831986,"checksum":,"allIPs":false,"reachProtectionThreshold":false,"env":,"valid":true,"useSpecifiedURL":false,"success":true}],
@String[Instance{instanceId="null", ip="10.167.179.93", port=20880, weight=1.0, healthy=true, enabled=true, ephemeral=true, clusterName='DEFAULT', serviceName='DEFAULT_GROUP@providers:com.mi.oa.hr.feedback.api.service.FeedbackRPCService::', metadata={side=provider, service.name=ServiceBean:com.mi.oa.hr.feedback.api.service.FeedbackRPCService, methods=uploadFile,sendToLark,sendToMeego,getFileByFdsObjectByName,save,screenshot, release=2.7.24-XIAOMI, deprecated=false, dubbo=2.0.2, pid=1, interface=com.mi.oa.hr.feedback.api.service.FeedbackRPCService, generic=false, retries=1, path=com.mi.oa.hr.feedback.api.service.FeedbackRPCService, protocol=dubbo, meta-data-type=remote, application=feedback, dynamic=true, service.filter=errorcodeExceptionFilter, -exception, category=providers, anyhost=true, timestamp=1736133948779}}],
@String[{side=provider, service.name=ServiceBean:com.mi.oa.hr.feedback.api.service.FeedbackRPCService, methods=uploadFile,sendToLark,sendToMeego,getFileByFdsObjectByName,save,screenshot, release=2.7.24-XIAOMI, deprecated=false, dubbo=2.0.2, pid=1, interface=com.mi.oa.hr.feedback.api.service.FeedbackRPCService, generic=false, retries=1, path=com.mi.oa.hr.feedback.api.service.FeedbackRPCService, protocol=dubbo, meta-data-type=remote, application=feedback, dynamic=true, service.filter=errorcodeExceptionFilter, -exception, category=providers, anyhost=true, timestamp=1736133948779}],
@String[{"name":"providers:com.mi.oa.hr.feedback.api.service.FeedbackRPCService::","groupName":"DEFAULT_GROUP","clusters":"","cacheMillis":60000,"hosts":[{"ip":"10.167.179.93","port":20880,"weight":1.0,"healthy":true,"enabled":true,"ephemeral":true,"clusterName":"DEFAULT","serviceName":"DEFAULT_GROUP@providers:com.mi.oa.hr.feedback.api.service.FeedbackRPCService::","marked":false,"metadata":{"side":"provider","service.name":"ServiceBean:com.mi.oa.hr.feedback.api.service.FeedbackRPCService","methods":"uploadFile,sendToLark,sendToMeego,getFileByFdsObjectByName,save,screenshot","release":"2.7.24-XIAOMI","deprecated":false,"dubbo":"2.0.2","pid":1,"interface":"com.mi.oa.hr.feedback.api.service.FeedbackRPCService","generic":false,"retries":1,"path":"com.mi.oa.hr.feedback.api.service.FeedbackRPCService","protocol":"dubbo","meta-data-type":remote,"application":feedback,"dynamic":true,"service.filter":errorcodeExceptionFilter,"exception":exception,"category":providers,"anyhost":true,"timestamp":1736133948779}],"valid":true,"instanceHeartBeatTimeout":15000,"ipDeleteTimeOut":30000}],"lastRefTime":1740028441939,"checksum":,"allIPs":false,"reachProtectionThreshold":false,"valid":true}],
@String[2.7.24-XIAOMI]

```

- 执行表达式

`getInstances` 之后会把数据绑定在 `instances` 变量上，可以通过执行表达式计算

代码块

```
1  vmtool --action getInstances --className java.lang.String --limit 1000 --
```

```
express 'instances[0]'
```

- 强制GC

## 代码块

```
1   vmtool --action forceGc
```

ID	NAME	OWNER	PRIORITY	STATE	AUTO	DELTATIME	LATE	INTERVAL	DURATION
117	dubbo-client-idleCheck-t	main	5	TIMED_WAITING	0.31	0.015	0:5.296	false	true
2	Reference Handler	system	18	WAITING	0.0	0.000	0:0.000	false	true
3	Finalizer	system	8	WAITING	0.0	0.000	0:0.000	false	true
4	Signal Dispatcher	system	9	RUNNABLE	0.0	0.000	0:0.000	false	true
5	Attach Listener	system	5	RUNNABLE	0.0	0.000	0:0.015	false	true
142	arthas-timer	system	5	WAITING	0.0	0.000	0:0.000	false	true
142	arthas-NettyTelnetBoo	system	5	RUNNABLE	0.0	0.000	0:0.031	false	true
142	arthas-NettyWebsocketB	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
142	arthas-NettyWebsocketTtyB	system	5	RUNNABLE	0.0	0.000	0:0.000	false	true
143	arthas-shell-server	system	5	TIMED_WAITING	0.0	0.000	0:0.000	false	true
143	arthas-session-manager	system	5	TIMED_WAITING	0.0	0.000	0:0.000	false	true
143	arthas-NettyTelnetBoo	system	5	RUNNABLE	0.0	0.000	0:0.984	false	true
149	arthas-command-execute	system	5	TIMED_WAITING	0.0	0.000	0:0.593	false	true
284	arthas-NettyHttpTelnetBoo	system	5	RUNNABLE	0.0	0.000	0:0.031	false	true
20	Timer-for-arthas-dashboard	system	5	RUNNABLE	0.0	0.000	0:0.002	false	true
23	com.cns-pool-evictor	main	5	TIMED_WAITING	0.0	0.000	0:0.000	false	true
25	logback	main	5	WAITING	0.0	0.000	0:0.000	false	true
27	com.alibaba.nacos.client.main	main	5	TIMED_WAITING	0.0	0.000	0:0.265	false	true
29	com.alibaba.nacos.client.main	main	5	TIMED_WAITING	0.0	0.000	0:0.000	false	true
30	com.alibaba.nacos.client.main	main	5	TIMED_WAITING	0.0	0.000	0:0.093	false	true
31	com.alibaba.nacos.client.main	main	5	WAITING	0.0	0.000	0:0.000	false	true
32	nacos.publisher-com.aliba	main	5	WAITING	0.0	0.000	0:0.000	false	true
Memory	used	total	max	usage	GC				
heap	173M	1674M	7282M	2.41%	gc.ps_scavenge.count	34			
ps Eden space	20M	761M	2526M	0.83%	gc.ps_scavenge.time(ms)	392			
ps survivor space	8K	80384K	88384K	0.00%	gc.ps_marksweep.count	5			
ps old gen	152M	834M	5401M	2.83%	gc.ps_marksweep.time(ms)	819			
nonheap	234M	244M	-1	95.90%					
code cache	70M	71M	240M	29.35%					
metaspace	147M	155M	-1	94.83%					
direct	17M	18M	1824M	1.67%					
mapped	1K	1K	-						
Runtime									
os.name					Windows 11				
os.version					10.0				
java.version					1.8.0_422				
java.home					C:\Users\mi\.jdk\corretto-1.8.0_422\jre				
system.load.average					-1.00				
processors					22				
timestamp/uptime					Thu Feb 28 11:27:15 CST 2025/2526s				

## Force GC 过程做了什么？

1. 标记阶段：通过GC Roots标记所有存活对象，遍历整个堆内存
  2. 清理阶段：通过标记清除算法，将所有未被标记的对象从内存清除
  3. 压缩阶段：对老年代进行压缩，将存活的对象移动到一侧，释放出连续内存空间
  4. 后续处理：整堆分区，重构RSet，清除DCQ，更新新生代

- 中断线程

## 代码块

```
1   vmtool --action interruptThread -t 1
```

## 2.2 监控

### 2.2.1 watch

方便观察指定方法的调用情况，返回值、异常、入参

1. `sm CLASS_NAME METHOD_NAME`：确定方法是否存在
2. `options unsafe true`：如果想要修改 `java.*` 下的类，请标记unsafe，**很危险不要轻易尝试**
3. `reset CLASS_NAME`：切记，监控完后必须要重置
4. 监控构造函数：`watch demo.CLASS <init>`

```
[arthas@32168]$ watch com.mi.oa.hr.workbench.api.web.BusinessController expatriateList -x 2
```

- `{params, target, returnObj}`：参数，对象，返回值

参数名称	参数说明
<code>class-pattern</code>	类名表达式匹配
<code>method-pattern</code>	函数名表达式匹配
<code>express</code>	观察表达式，默认值： <code>{params, target, returnObj}</code>
<code>condition-express</code>	条件表达式
<code>[b]</code>	在**函数调用之前**观察
<code>[e]</code>	在**函数异常之后**观察
<code>[s]</code>	在**函数返回之后**观察
<code>[f]</code>	在**函数结束之后**（正常返回和异常返回）观察
<code>[E]</code>	

	开启正则表达式匹配，默认为通配符匹配
[x:]	指定输出结果的属性遍历深度，默认为 1，最大值是 4
[m <arg>]	指定 Class 最大匹配数量，默认值为 50。长格式为 [maxMatch <arg>]。

## 2.2.2 Ognl 表达式

### Ognl 表达式

#### 代码块

```
1 ognl '@expression'
```

```
[arthas@032168]$ ognl '@com.mi.oa.hr.workbench.common.constant.CommonConstants@USER_LOCATION_MAP.entrySet()'
@EntrySet[
    @Node[xukexin=CHN003],
    @Node[xiaoqian3=CHN004],
    @Node[xiedan=CHN004],
    @Node[zengli3=CHN003],
    @Node[longzhaoxia=CHN003],
    @Node[xieyin1=CHN002],
    @Node[zhangyun=CHN006],
    @Node[yangwei12=CHN004],
]
```

## 2.3 编译与反编译

### 2.3.1 反编译

#### 反编译

1 jad class

```
[arthas@32168]$ jad com.xiaomi.oa.hr.hrod.boot.request.HpcConditionDeptCompany

ClassLoader:
+-sun.misc.Launcher$AppClassLoader@18b4aac2
 +-sun.misc.Launcher$ExtClassLoader@e25b2fe

Location:
/C:/Users/mi/.m2/repository/com/xiaomi/oa/hr/hrod-api-spring-boot-starter/1.0.73-SNAPSHOT/hrod-api-spring-boot-starter-1.0.73-20231017.092806-1.jar

/*
 * Decompiled by CFR 0.152 (Fabricated byne).
 */
package com.xiaomi.oa.hr.hrod.boot.request;

public class HpcConditionDeptCompany {
    private String deptId;
    private String cityCode;
```

## 文章引用

- 工作六年，我学会了用Arthas来辅助我的日常工作
- Arthas内存分析工具介绍与示例
- Arthas官方文档
- Ognl 语法基础教程

## 案例介绍

- Arthas解决线上字符乱码问题