**保密级别：■普通 □秘密 □机密**

**Hydra安装手册**

部 门： \_\_\_\_\_\_大数据平台\_\_\_\_\_\_\_

编写人： \_\_\_\_\_\_\_\_\_jd-bdp\_\_\_\_\_\_\_\_

核准人： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日 期： \_2013\_年\_\_6\_\_月

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **版本号** | **修订日期** | **修订概述** | **修订人** | **审核人** | **批准人** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[1. 简介 3](#_Toc356981490)

[1.1. 部署简介 3](#_Toc356981491)

[1.2. 部署要求 3](#_Toc356981492)

[1.2.1. 硬件要求 3](#_Toc356981493)

[1.2.2. 软件要求 4](#_Toc356981494)

[2. 源码获取 4](#_Toc356981495)

[3. 导入工程 4](#_Toc356981496)

[4. 源码打包 5](#_Toc356981497)

[5. 安装部署 5](#_Toc356981498)

[5.1. Services 5](#_Toc356981499)

[5.2. Hydra-client 5](#_Toc356981500)

[5.3. Hydra-manager 5](#_Toc356981501)

[5.4. Hydra-collector 6](#_Toc356981502)

[5.5. Hydra-web 7](#_Toc356981503)

[5.6. 数据库表 7](#_Toc356981504)

[6. 最佳实践 8](#_Toc356981505)

# 简介

Hydra分为“内部使用版”和“外部开放版”。不同的版本，架构和部署上存在细微差异，本文档只介绍“外部版”的安装步骤，仅供参考。

在安装部署的过程中难免会遇到各种问题，请在了解maven和git等基准上来顺利安装和部署Hydra系统。

## 部署简介

角色说明：

* Services

业务系统，一整套的分布式服务集群系统，现阶段为Dubbo框架下的服务集群系统，

本系统提供了两个exmple场景，可以在源码中hydra-exmple获得

* Hydra-client

打包的为Jar包，和具体的某个应用部署在一起，共享应用JVM，为Hydra的接入层

* Hydra-manager

打包为tar.gz/zip包格式，负责所有业务跟踪点的协调和管理

* Hydra-collector

打包为tar.gz/zip包格式，负责跟踪数据的搜集，并落地到数据库中

* Hydra-web

打包为war包格式，负责从数据库中读取数据和前端展现

* Mysql-DB

用于存储跟踪数据

* Tomcat-Contailer

为Hydra-web提供servlet容器。

## 部署要求

### 硬件要求

1+套业务系统集群机

1套zookeeper单点或集群机

1台机器部署Hydra-manager

1+台机器部署Hydra-Collector

1台机器部署Hydra-web

1台数据库服务器

上述机器可以根据实际情况，做部署上合理规划。

### 软件要求

* Dubbo

Hydra是基于alibaba的dubbo框架基准上做的服务跟踪系统，理论上原有的Dubbo框架服务群中所有应用不需要额外的配置，皆可以平滑的接入Hydra系统。

* Zookeeper

各个服务点依赖于zookeeper来读取Hydra-manager和Hydra-collector获取数据交互路由点，来完成跟踪数据的推送和跟踪的控制。

* Mysql-DB

跟踪数据的持久化存储。

* Tomcat-contailer

前端web应用容器

# 源码获取

|  |
| --- |
| cd ~  git clone [https://github.com/jd-bdp/hydra.git](https://github.com/alibaba/dubbo.git) hydra  git checkout -b hydra-1.x  git checkout master |

# 导入工程

|  |
| --- |
| cd ~/hydra  mvn idea:idea  Eclipse -> Menu -> File -> Import -> Exsiting Projects to Workspace -> Browse -> Finish:  Context Menu -> Run As -> Java Application:  Edit Config:  hydra-example/hydra-exmple-exp1/src/main/resources/dubbo-global.properties  hydra-example/hydra-exmple-exp2/src/main/resources/dubbo-global.properties |

# 源码打包

|  |
| --- |
| cd hydra  mvn install -Dmaven.test.skip=true |

# 安装部署

## Services

Hydra默认使用了hydra-exmple中的两个应用场景来做，你可以在hydra-test/hydra-test-integration打包中获得应用场景。

hydra-test-intergration 分为windows版和linux版(默认)，见如下打包方法。

|  |  |
| --- | --- |
| window | mvn package -Pruntime-env-windows |
| linux | mvn package -Pruntime-env-linux |

获得tar.gz包或者zip包后，将服务分布式部署到不同的机器上，以模拟应用场景,

一下介绍场景一的部署方法，场景二的部署方法类似。

* 打包

|  |
| --- |
| cd ~/hydra/modules/hydra-test/hydra-test-integration  mvn package -Dmaven.test.skip=true |

* 部署

|  |
| --- |
| scp -r target/\*.tar.gz username@ip:dirname |

* 配置

|  |
| --- |
| cd basedir/conf  vi \*exp1.properties |

* 启动

|  |
| --- |
| cd basedir/bin  cd exp1  ./startC.sh  ./startB.sh  sh startA.sh  cd ..  sh startTrigger-exp1.sh start |

* 停止

|  |
| --- |
| cd basedir/bin  sh startTrigger-exp1.sh stop  All.sh stop |

* 输出

|  |
| --- |
| cd basedir/log  tail -f \*.log |

## Hydra-client

|  |
| --- |
| cd ~/hydra/modules/hydra-client  mvn package -Dmaven.test.skip=true |

将打包好的jar包放在应用的classpath下，启动应用即可。

不用管这步，前面打包好了

## Hydra-manager

* 打包

|  |
| --- |
| cd ~/hydra/modules/hydra-manager  mvn package -Dmaven.test.skip=true |

* 部署

|  |
| --- |
| scp -r target/\*.tar.gz username@ip:dirname |

* 配置

|  |
| --- |
| cd basedir/conf |

* 启动

|  |
| --- |
| cd basedir/bin  sh manager.sh start |

* 停止

|  |
| --- |
| cd basedir/bin  sh manager.sh stop |

* 输出

|  |
| --- |
| cd basedir/log  tail -f manager.log |

## Hydra-collector

* 打包

|  |
| --- |
| cd ~/hydra/modules/hydra-collector  mvn package -Dmaven.test.skip=true |

* 部署

|  |
| --- |
| scp -r target/\*.tar.gz username@ip:dirname |

* 配置

|  |
| --- |
| cd basedir/conf |

* 启动

|  |
| --- |
| cd basedir/bin  sh collector.sh start 这里改为collector-mysql.sh start |

* 停止

|  |
| --- |
| cd basedir/bin  sh collector.sh stop |

* 输出

|  |
| --- |
| cd basedir/log  tail -f collector.log |

## Hydra-web

* 打包

|  |
| --- |
| cd ~/hydra/modules/hydra-web  mvn package -Dmaven.test.skip=true |

* 部署

|  |
| --- |
| scp -r target/\*.war username@ip:$TOMCAT\_WEBAPPS |

* 配置

|  |
| --- |
| cd $TOMCAT\_CONF  vi \*.ini ... |

* 启动

|  |
| --- |
| startup.sh |

* 停止

|  |
| --- |
| shutdown.sh |

* 输出

|  |
| --- |
| tail -f $LOG\_DIR/\*.log |

## 数据库表

sql语句为：[table-mysql\initTable.sql](file:///E:\\hydra_workspace\\hydra\\docs\\table-mysql\\initTable.sql)

# 最佳实践

以下演示安装样例：

1. 部署zk单点或集群环境，以保证获得最佳SOA，zk的部署请参照官方文档。
2. 部署应用系统ABC，只需要部署hydra-test-integration模块打包的tar.gz包，拷贝三份分布式部署。
3. 部署一个触发器Trigger，以激活服务的调用。
4. 部署一个Manager，以管理各个跟踪点的跟踪上下文。
5. 部署一个或者多个Collector消费机集群，以搜集来自Hydra-client推送过来的跟踪数据。
6. 部署一个web应用，已提供给前端展现应用系统服务上下文。

Example场景说明：

有三个服务应用A、B、C和一个触发RPC调用的应用Trigger，服务调用关系为A-B-C，

每隔500s触发一个调用，持续时间为1天，最终能在前端展现效果如下截图。

部署地址

|  |  |  |
| --- | --- | --- |
| 角色 | ip | port |
| ZK | 192.168.200.110-112 | 2181~ |
| A | 192.168.200.110 | 20990 |
| B | 192.168.200.111 | 20991 |
| C | 192.168.200.112 | 20992 |
| Trigger | 192.168.200.113 | - |
| Manager | 192.168.228.81 | 20890 |
| Collector | 192.168.228.81-82 | 20889 |
| Web | 192.168.228.81 | 8080 |
| MySql-DB | 192.168.228.81 | - |

效果图：