# Dubbo使用手册

目录

[Dubbo使用手册 1](#_Toc474236254)

[1 安装 3](#_Toc474236255)

[2 配置 3](#_Toc474236256)

[3 dubbo-admin 3](#_Toc474236257)

[4 参考资料 6](#_Toc474236258)

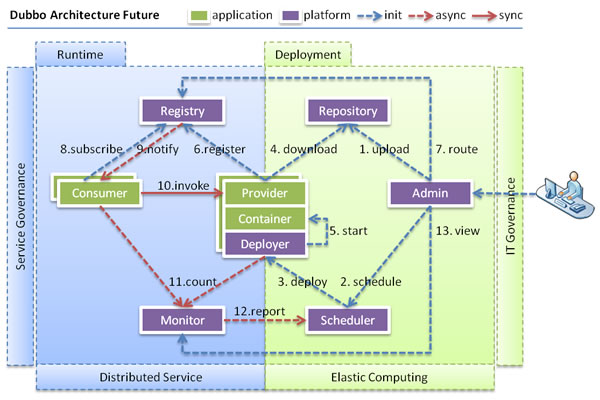
# 背景

解耦

# 架构&原理

s

服务升级流程



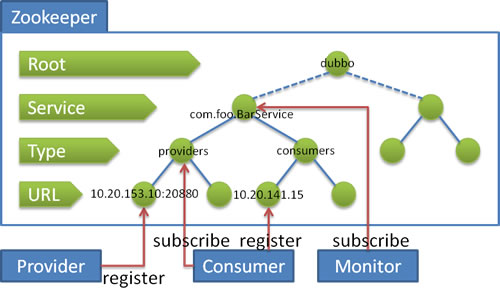
注意这里的Elastic Computing

升级流程和普通升级流程无差异

# 安装

## 注册中心

这里选取基于zookeeper的注册中心，关于zookeeper安装使用，请参考《**Zookeeper使用手册**》



# 配置

## CODI\_HOME/dubbo.properties

|  |
| --- |
| # Dubbo properties  #  # 应用名  dubbo.application.name.provider=codi-superman-provider  dubbo.application.name.consumer=codi-superman-consumer  #【注册中心地址】  dubbo.registry.address=zookeeper://120.55.85.150:2181?backup=120.55.85.150:2182,120.55.85.150:2183  #【分组】用于隔离环境  dubbo.registry.group=dubbo-dev  # 调用协议地址  dubbo.protocol.name=dubbo  dubbo.protocol.port=28080  #【最大线程数】本地开发时可以小一些，比如100，线上适当调大以适应线上机器，如200  dubbo.service.max.thread.threads.size=100 |

红色部分请**按实际项目需求**进行修改

## spring/spring-dubbo-provider.xml

将此xml导入spring-application.xml中

|  |
| --- |
| <beans>  …  <!--dubbo-->  <import resource="./spring-dubbo-provider.xml"/>  <beans/> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd">  <dubbo:application name="${dubbo.application.name.provider}" logger="slf4j"/>  <dubbo:registry address="${dubbo.registry.address}" group="${dubbo.registry.group}"/>  <dubbo:protocol name="${dubbo.protocol.name}"  port="${dubbo.protocol.port}"  threads="${dubbo.service.max.thread.threads.size}"/>  <dubbo:provider/>  <bean id="userService" class="com.codi.yourpackage.biz.service.impl.UserServiceImpl"/>  <dubbo:service interface="com.codi.yourpackage.api.service.UserService" ref="userService"/>  <!--基于注解方式-->  <!--<dubbo:annotation package="com.spy.dubbo.provider.service.impl"/>-->  </beans> |

默认情形下只需要添加需要发布的服务

## spring/spring-dubbo-consumer.xml

将此xml导入spring-application.xml中

|  |
| --- |
| <beans>  …  <!--dubbo-->  <import resource="./spring-dubbo-consumer.xml"/>  <beans/> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd">  <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->  <dubbo:application name="${dubbo.application.name.consumer}"/>  <dubbo:registry address="${dubbo.registry.address}" group="${dubbo.registry.group}"/>  <dubbo:consumer/>  <!-- 生成远程服务代理，可以和本地bean一样使用userService -->  <dubbo:reference id="userService" interface="com.codi.yourpackage.api.service.UserService"/>  <dubbo:annotation package="com.spy.dubbo.consumer"/>  </beans> |

默认情况下只需添加使用的服务接口

# 开发使用

http://dubbo.io/User+Guide-zh.htm

文档介绍的很详细

# 服务治理

## dubbo-admin（使用）

由于dubbo-admin默认情况下使用jdk7，当使用jdk8时报如下错误：

Bean property 'URIType' is not writable or has an invalid setter

因此，重新构建dubbo-admin

|  |
| --- |
| 1、webx的依赖改为3.1.6版；  <dependency>  <groupId>com.alibaba.citrus</groupId>  <artifactId>citrus-webx-all</artifactId>  <version>3.1.6</version>  </dependency>  2、添加velocity的依赖，我用了1.7；  <dependency>  <groupId>org.apache.velocity</groupId>  <artifactId>velocity</artifactId>  <version>1.7</version>  </dependency>  3、对依赖项dubbo添加exclusion，避免引入旧spring  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <version>${project.parent.version}</version>  <exclusions>  <exclusion>  <groupId>org.springframework</groupId>  <artifactId>spring</artifactId>  </exclusion>  </exclusions>  </dependency>  4、webx已有spring 3以上的依赖，因此注释掉dubbo-admin里面的spring依赖  <!--<dependency>-->  <!--<groupId>org.springframework</groupId>-->  <!--<artifactId>spring</artifactId>-->  <!--</dependency>-->  确定war包解压后lib目录没有spring 3 以下的依赖就行。然后运行正常了。 |

dubbo-admin默认没有启用分组功能，这里启用分组功能

dubbo\dubbo-admin\src\main\resources\META-INF\spring\dubbo-admin.xml



在${user.home}/dubbo.properties中添加分组信息

dubbo.registry.group=dubbo

dubbo\dubbo-admin\src\main\webapp\WEB-INF\dubbo.properties

unzip dubbo-admin-2.5.4-SNAPSHOT.war –d dubbo-admin

/opt/tomcat-dubbo-admin/webapps/dubbo-admin/WEB-INF

dubbo.registry.address=zookeeper://cloud1.cd121.cc:2181?backup=cloud2.cd121.cc:2181,cloud3.cd121.cc:2181

dubbo.admin.root.password=Codi1234 #管理员账户

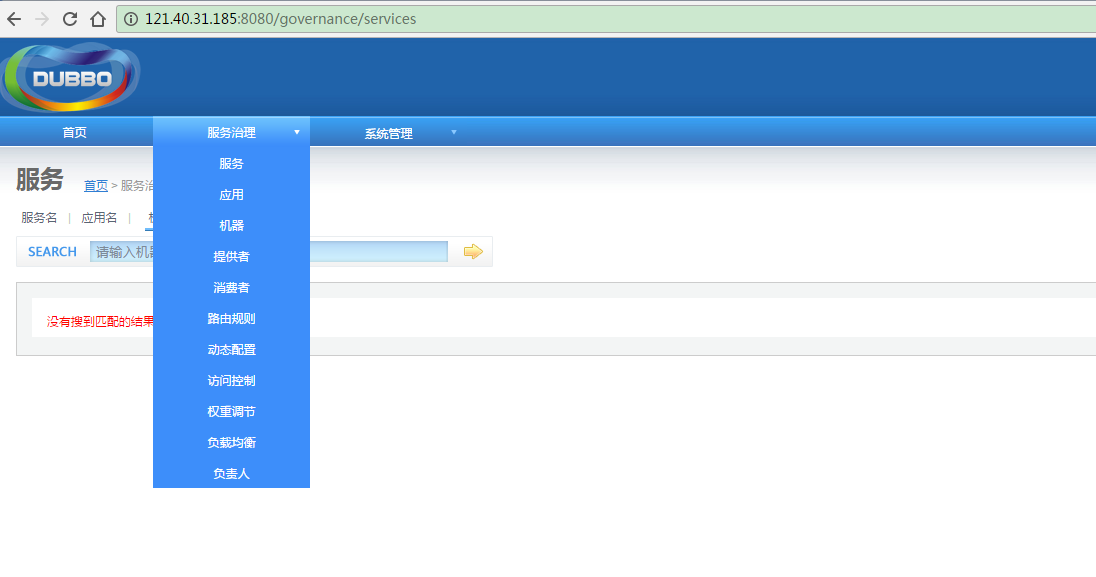
dubbo.admin.guest.password=Codi1234

启动

sh start.sh

<http://121.40.31.185:8080/>

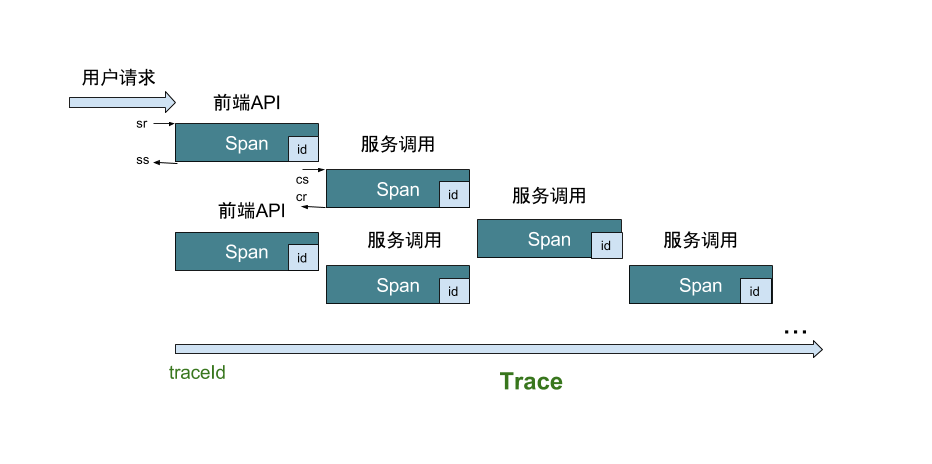




# 服务跟踪

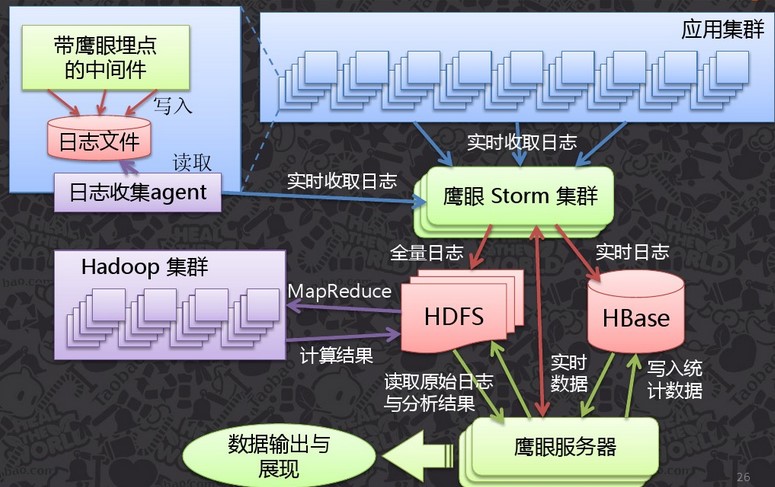
## 理论基础

Google Dapper论文， 参考中文 https://bigbully.github.io/Dapper-translation/



## 业界Trace系统：

### 阿里的Eagle eye

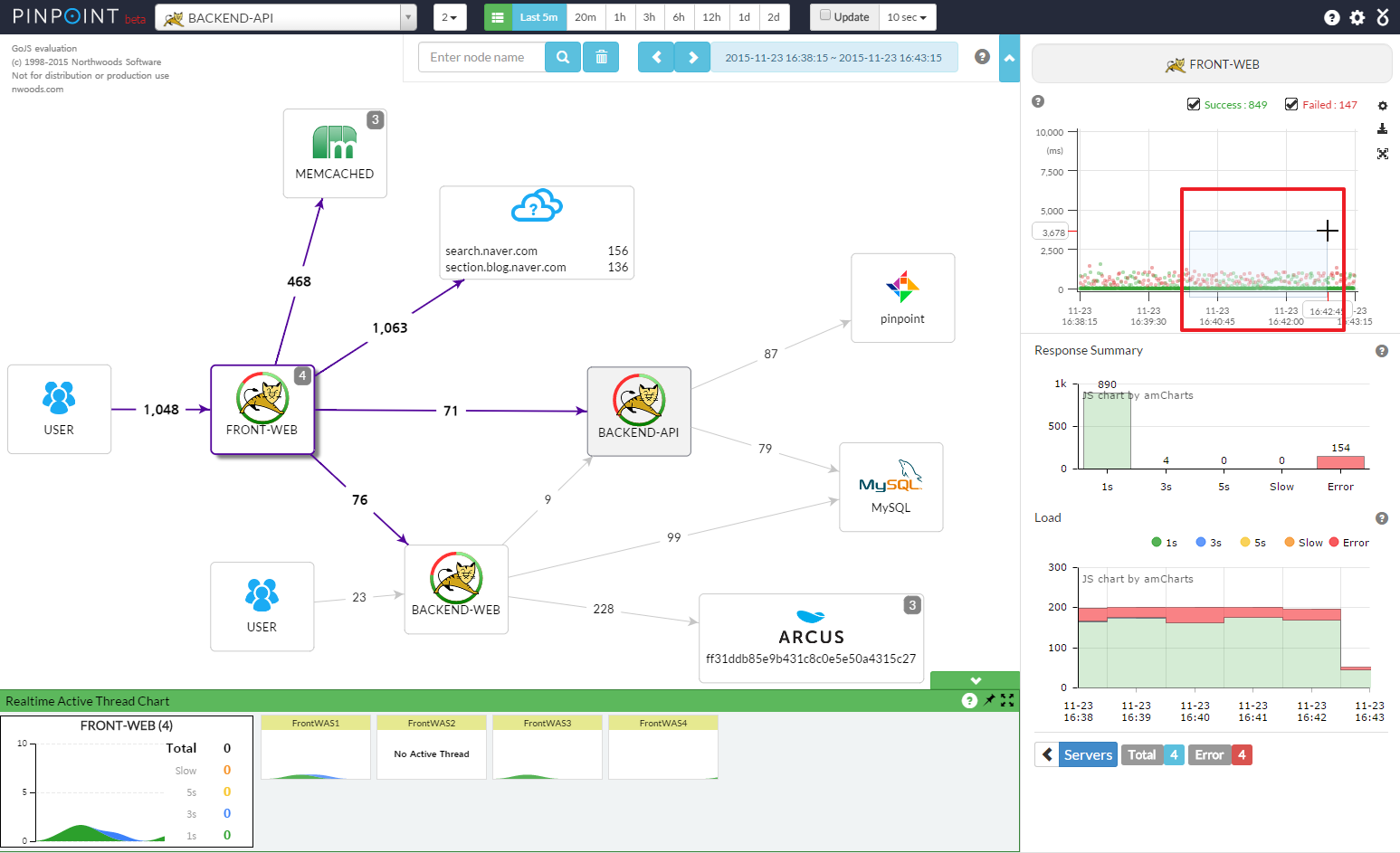


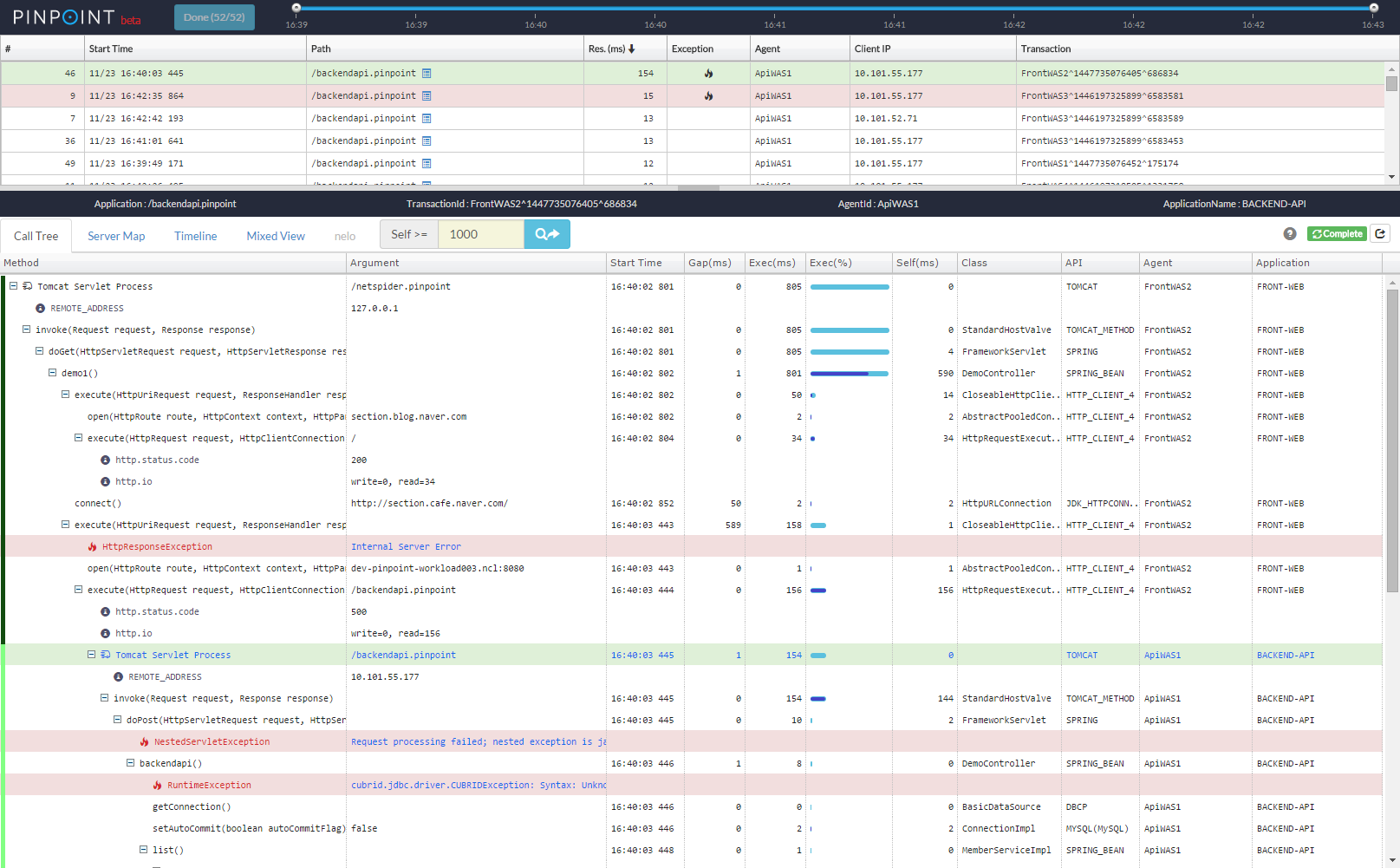
不开源

### Twitter的Zipkin

参考 6.3

### Pinpoint



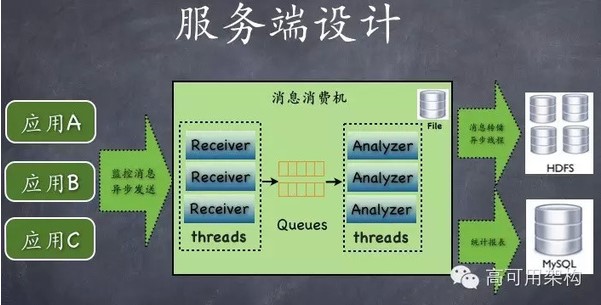
s

<https://github.com/naver/pinpoint>

### 京东的hydra

<https://github.com/odenny/hydra> 2013.3.30停止维护

### 点评的CAT



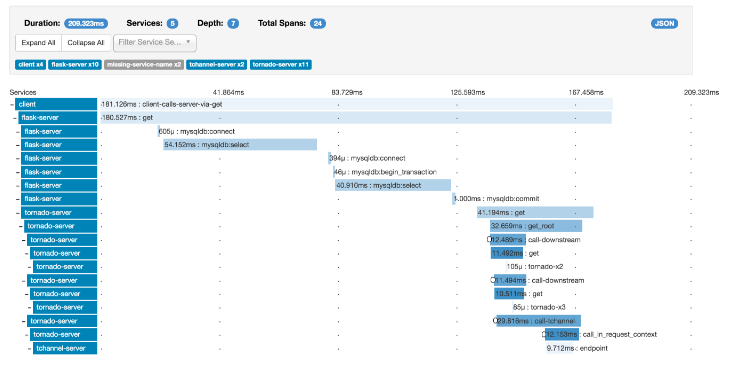
<https://github.com/dianping/cat>

难易程度：Pinpoint < hydra < CAT

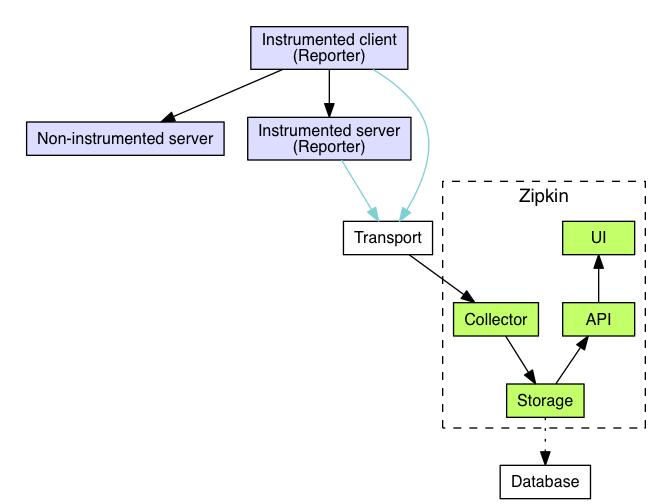
总之，针对这样的需求，一般要求分布式跟踪系统大致满足如下特性：低侵入性，高性能，高可用容错，低丢失率

## zipkin（使用）

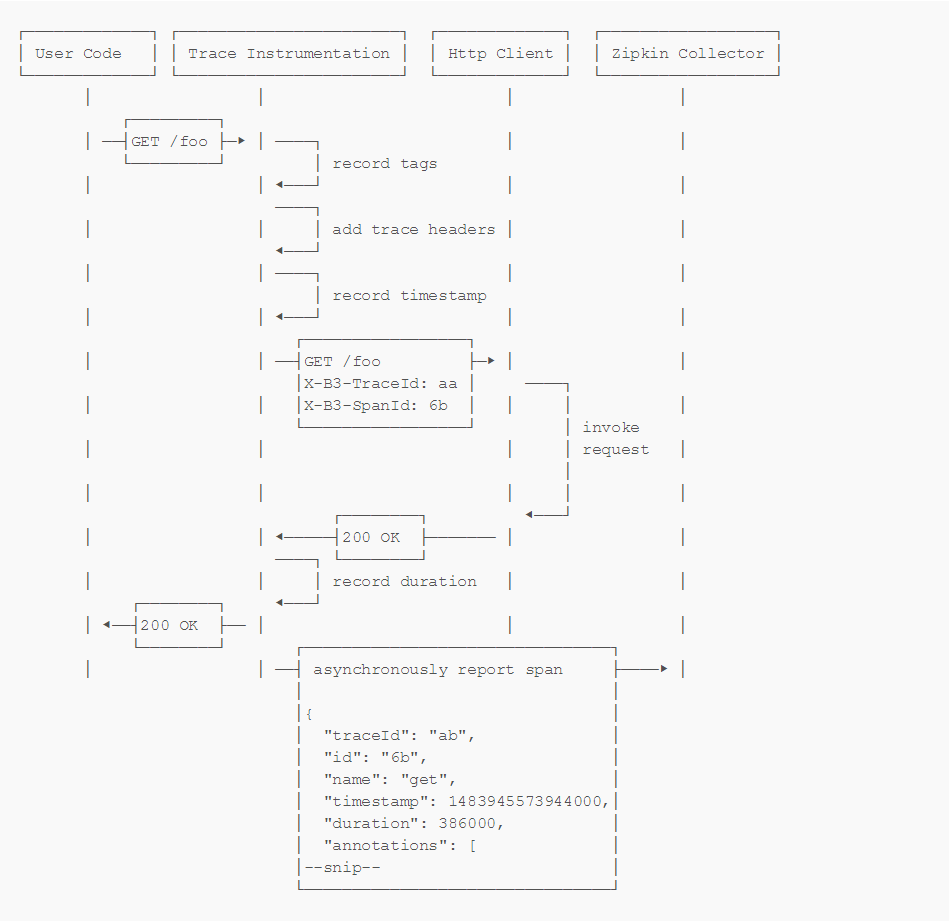
### WebUI



### 架构图



### 流程图



### 缩略词

**span**：标识一次分布式调用，其自身包含了id，parentId(指向上级Span的id)， traceIds，服务名称等重要属性，其应尽量保持精简；

**trace**：标识整个请求链，即一些列Span的组合。其自身的ID将贯穿整个调用链，其中的每个Span都必须携带这个traceId，因此traceId将在整个调用链中传递；

**cs**：客户端发起请求，标志Span的开始；

**sr**：服务端接收到请求，并开始处理内部事务，其中sr - cs则为网络延迟和时钟抖动；

**ss**：服务端处理完请求，返回响应内容，其中ss - sr则为服务端处理请求耗时；

**cr**：客户端接收到服务端响应内容，标志着Span的结束，其中cr - ss则为网络延迟和时钟抖动。

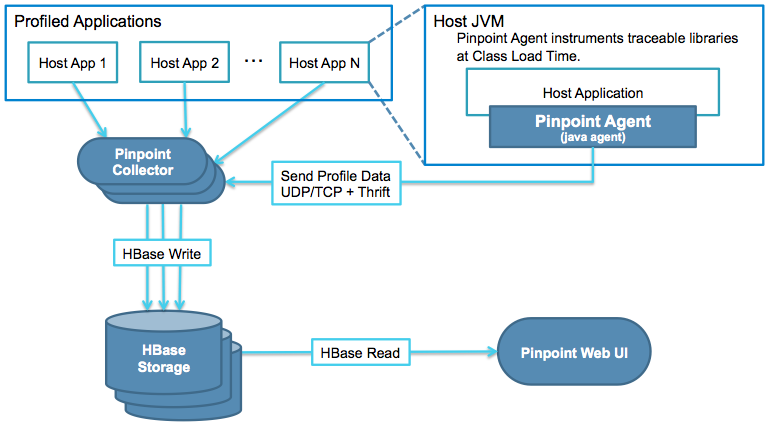
### 应用

<https://gitlab.codiruifu.com/codi-product/dubbo-trace>

本工程中详细列举了相关示例、以及接入配置，可快速上手

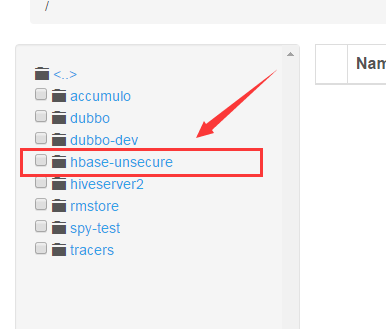
## Pinpoint

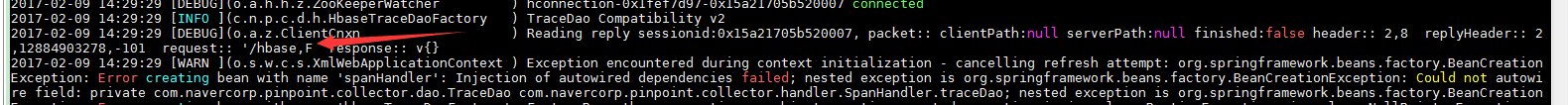
### 架构图



### 应用

必须是/hbase





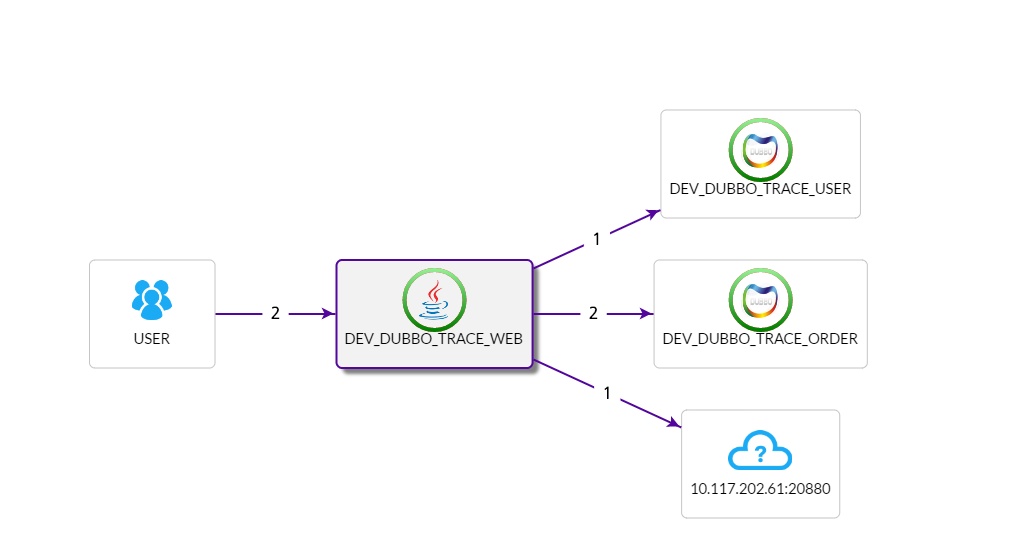
在ambari系统中

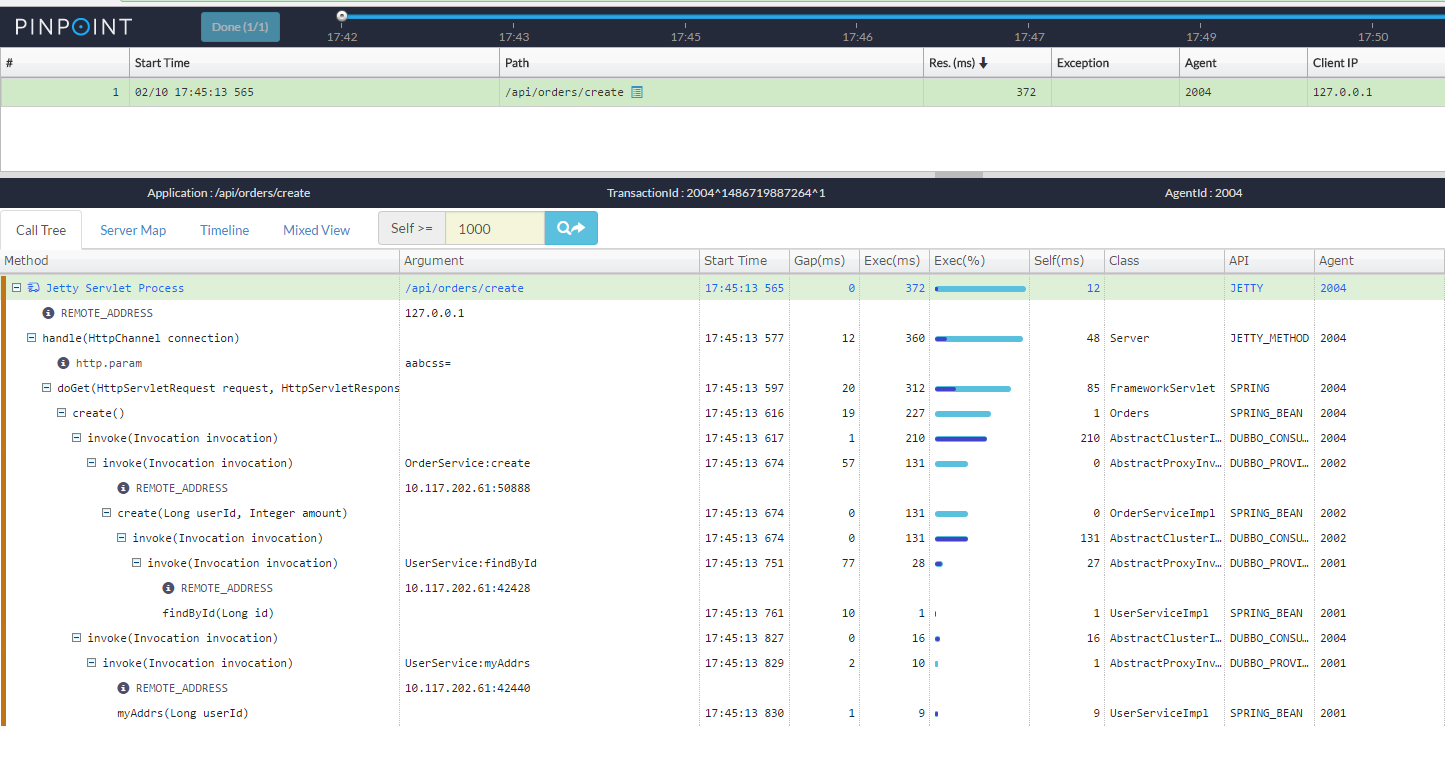


分别启动pinpoint-collector,pinpoint-web

在对于的应用tomcat/bin下添加setenv.sh文件，内容如下

|  |
| --- |
| AGENT\_PATH=/opt/pinpoint/pinpoint-agent  VERSION=1.6.0  AGENT\_ID=1001  APPLICATION\_NAME=DEV\_FUND  CATALINA\_OPTS="$CATALINA\_OPTS -javaagent:$AGENT\_PATH/pinpoint-bootstrap-$VERSION.jar"  CATALINA\_OPTS="$CATALINA\_OPTS -Dpinpoint.agentId=$AGENT\_ID"  CATALINA\_OPTS="$CATALINA\_OPTS -Dpinpoint.applicationName=$APPLICATION\_NAME" |





数据统计有问题

# 参考资料

* <http://dubbo.io/>
* <https://github.com/alibaba/dubbo/tree/master/dubbo-demo>
* <https://bigbully.github.io/Dapper-translation/> dapper中文
* <http://zkread.com/article/1108251.html> （业界Trace系统介绍）
* <https://github.com/ihaolin/dubbo-trace>
* <http://zipkin.io>
* <http://zipkin.io/pages/architecture.html>