**分布式调度框架架构设计说明书**

**员工：陈宝祥**

**工号：608344**

**所在项目组：大数据组**

目录

[一、 简介 4](#_Toc5799)

[1. 编写目的 4](#_Toc17044)

[2. 文档范围 4](#_Toc24782)

[二、 架构表示方式 4](#_Toc5580)

[三、 架构设计目标与约束 5](#_Toc9921)

[1. 设计目标 5](#_Toc1862)

[2. 0层架构 6](#_Toc509)

[3. 一层架构 6](#_Toc31155)

[4. 关键功能需求 7](#_Toc25842)

[5. 关键质量需求 8](#_Toc20030)

[6. 开发策略 8](#_Toc27315)

[四、 用例视图 8](#_Toc16376)

[1. 概述 8](#_Toc25899)

[2. 关键用例 9](#_Toc22460)

[五、 逻辑架构 10](#_Toc9727)

[1. 概述 10](#_Toc9218)

[2. 逻辑分层 10](#_Toc8072)

[六、 进程视图 11](#_Toc31866)

[1. 概述 11](#_Toc9862)

[2. 角色进程视图 11](#_Toc5999)

[七、 实施视图 12](#_Toc8824)

[1. 概述 12](#_Toc10044)

[代码工程视图 13](#_Toc10696)

[八、 部署视图 13](#_Toc10321)

[1. 概述 13](#_Toc12319)

[2. 部署方案视图 13](#_Toc3240)

# 简介

## 编写目的

本文档全面与系统地表述了分布式调度框架的系统构架，并通过使用多种视图来从不同角度描述本系统的各个主要方面，以满足相关涉众（运维、开发人员等）对本系统的不同关注点和需求。本文档记录并表述了系统架构的设计人员对系统构架方面做出的重要决策。

本文档的预期阅读人员为项目经理、程序设计人员、测试人员和其他有关的工作人员。

## 文档范围

本软件架构说文档适合于分布式调度系统的总体应用架构。

# 架构表示方式

本软件架构设计文档以一系列的视图（View）来表示系统的软件构架，主要包括用例视图、逻辑视图、进程视图、部署视图、实施视图等，每个视图拥有一个或多个模型（Model）。并围绕相关视图来描述系统的基本结构、组成机制与工作原理等。本软件架构设计文档还将系统的构架机制描述也放在了逻辑视图之下。本文档主要使用统一建模语言（UML）来充当相关模型的表达语言。

# 架构设计目标与约束

描述构架设计最主要目标就是满足关键系统功能需求和质量约束，这些功能需求和质量要求对软件构架有重大的影响，并决定了构架的设计。本节同时还列明影响构架的其他相关因素，如软件的复用策略、使用商业构件、设计与实施的策略等。

## 设计目标

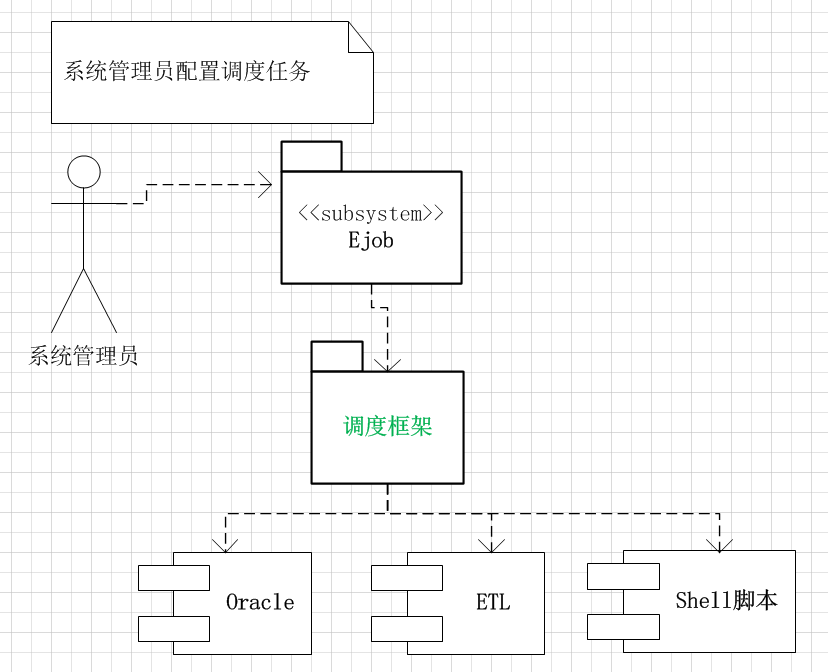
1）支持任务依赖配置(配置文件或提供界面)

2）支持与Ejob集成，任务监控和周期调度交由Ejob去做

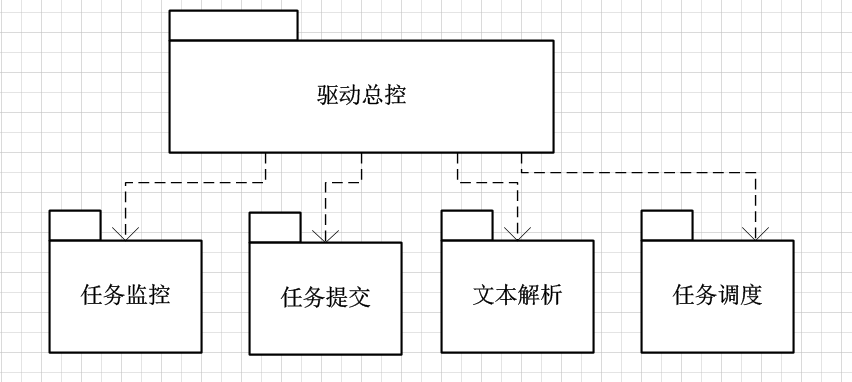
3）支持与yarn等资源调度框架集成

4）质量属性：服务线性扩展、高可用、性能等;

## 0层架构



## 一层架构



## 关键功能需求

* 驱动总控模块

驱动总控模块是调度系统的核心执行模块，负责各模块间的协调工作，主要提供1）调用文本解析2）、提供启动监控；3）提供调用任务调度4）提供调用任务提交5）提供命令行参数解析功能

* 文本解析模块

1、提供解析xml文本功能；并将解析后的文本信息写入zk；

2、提供文本变化监听功能，当文本发生变化时，动态刷新调度任务信息；

3、提供任务信息合并功能，支持多个配置任务合并为一个；

4、提供注解方式注入任务信息；

* 任务调度器

1、提供将任务根据依赖关系，切分为不同的阶段的功能；

2、提供任务依赖性检测功能，避免任务死循环；

3、提供每一个阶段内，任务并发执行切分功能；

* 任务提交器

1、提供不同的运行模型进行提交（集群模式，本地模式）

2、提供资源测试功能（每个任务所占的cpu，内存等资源监测）

* 任务监控模块

1、提供对每个任务和阶段进行监控的功能；

2、提供每个阶段提交时，失败重试等容错功能；

3、提供每个阶段内，单个任务失败等；

## 关键质量需求

由于分布式调度模块涉及到跑批任务的调度运行，因此要求稳定、安全、便捷，易于管理和操作。

* 调度速度：不超过10秒；
* 可扩展性：要求支持shell,存储过程,java程序等调用；
* 可靠性：平均故障间隔时间不低于200小时。

## 开发策略

* **软件复用策略**

系统才用EDA事件驱动架构进行API的解耦操作，各种API间的调用通过事件传递；

# 用例视图

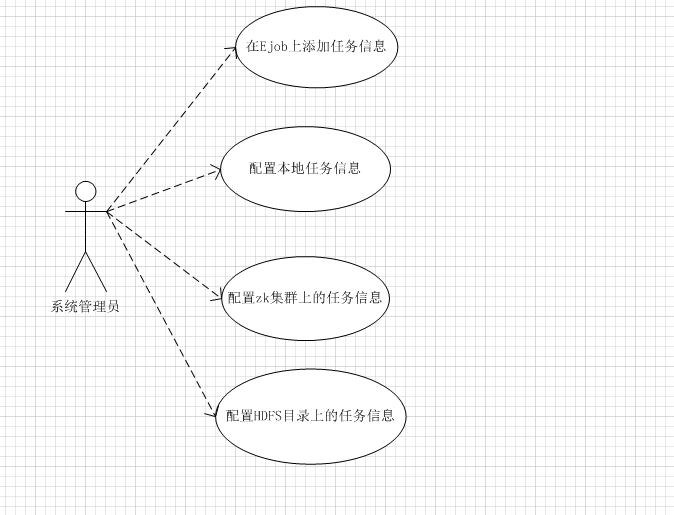
## 概述

用例视图从用户使用的角度描述系统构架的基本外部行为特性，通常包含业务用例模型与系统用例模型。业务用例模型不适用于本系统，这里只关注系统用例。这里选取了用例模型中对系统构架的内容产生重大影响的应用场景与用例集合，这些用例代表了系统主要的核心功能，决定了系统构架的基本组成元素。有些用例强调或决定了构架的某些具体然而重要的细节，通常也可以列在本节内，总之所列的用例集合应基本覆盖系统构架的主要方面。

## 关键用例

* 关键的系统参与者

1. 系统管理员的用例



游客的用例说明

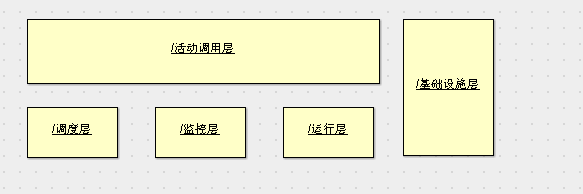
|  |  |
| --- | --- |
| **用例名称** | **简单描述** |
| **在Ejob上添加任务信息** | 根据用户输入，在Ejob建立调度任务，定时调度； |
| **配置本地任务信息** | 在本地目录下，配置任务信息（包括任务ID，依赖等），然后保存在本地任务等待框架解析； |
| **配置ZK上的任务信息** | 在ZK上编辑任务的信息，由调度框架进行解析读取； |
| **配置HDFS上的任务信息** | 在HDFS目录下，编辑任务的调度信息，调度框架进行读取解析； |

# 逻辑架构

## 概述

逻辑视图从系统内在逻辑结构的角度描述系统的基本结构与动态行为，通常包括分析模型（Analysis Model）、设计模型（Design Model）以及数据模型（Data Model）等。设计模型说明了系统的组成元素、组织架构和关系，并描述了各组成元素的协作以及状态转换关系等；本节将分别在系统层次结构模型中描述系统的层次组织结构；在主要的包和子系统中说明系统的具体组成。

## 逻辑分层



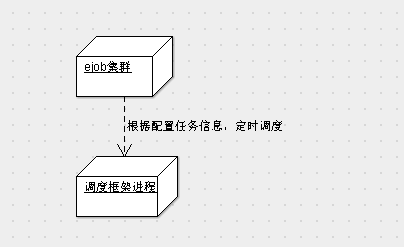
# 进程视图

## 概述

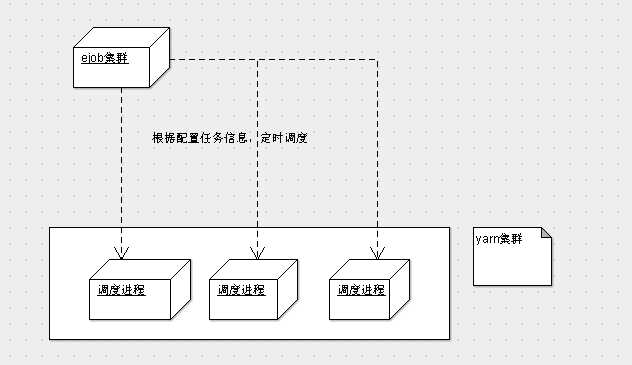
进程视图从系统运行时刻的角度，描述系统划分为进程、线程的结构，及其动态关系。模型主要说明不同系统角色之间的创建、交互和消息通讯关系等。

## 角色进程视图

本地模式：



集群模式：

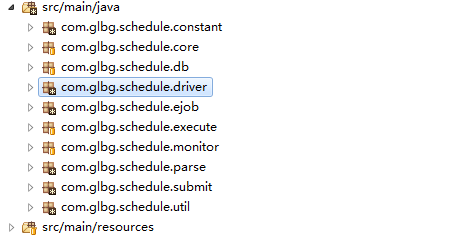


# 实施视图

## 概述

本部分从编译与构建的角度，描述系统实施构件的组织结构与依赖关系（主要是编译依赖）。模型包括实施子系统和构件结构，及其依赖关系。同时还表达了逻辑视图中各个包和类分配到实施视图中的子系统和构件的映射关系。

## 代码工程视图



其中：

Core:调度模块；

Db：数据库操作相关模块；

Driver:驱动总控模块；

Ejob:跟Ejob集成相关的事情；

Execute：任务执行模块；

Monitor：任务监控模块；

Parse：文本解析模块；

Submit：任务提交模块；

Util：各种工具类；

# 部署视图

## 概述

从系统软硬件物理配置的角度，描述系统的网络逻辑拓扑结构。模型包括各个物理节点的硬件与软件配置，网络的逻辑拓扑结构，节点间的交互和讯关系等。同时还表达了进程视图中的各个进程具体分配到物理节点的映射关系。

## 部署方案视图