1.大型网站高并发访问时候，如何处理

部署多个web应用实例 tomcat

负载均衡 建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

Tomcat可以配置负载均衡，F5 BIG-IP，Nginx

Redix nosql中的一种

缓存 用到的挺多

分布式系统，

做企业级产品，做小系统时候，不用拆分，企业级的不会拆分，如OA。

做服务，产品时候要把大的应用，拆分成小的系统

子系统之间的调用 有webservice、rest接口、[HttpInvoker](http://lot-of-mine.iteye.com/blog/1020491)、RMI，Hessian，Burlap，Httpinvoker等

传统的任务调度

timer 时间调度框架 简单的业务逻辑

**Quartz**，**Quartz**表达式 定时器，复杂的业务逻辑，处理大数据有一定得局限性

•Timer

•ScheduledExecutor

•开源工具包 Quartz

•开源工具包 JCronTab

多线程，部署多个实例，做个任务池

服务器集群？

多线程、多实例可能存在的问题

多线程的两种模式

no sleep 1,2 处理完成之后，抓数据，3继续处理它自己的数据

sleep 1,2,3都完成之后再去抓数据

拆表

按时间拆

按类型拆

按区域拆

查询路由

迁移hist表

业务表的数据迁移到hist表

最近的时间可以查业务表

spring中的几个重要接口

ApplicationListener <ContextRefreshedEvent>监听spring容器加载完触发

ApplicationContextAware设置ApplicationContext

InitializingBean类初始化完触发

factoryBean

在spring开始启动的时候就执行某方法的三种方案

1. 在bean的配置文件中，指定init-method=”方法名”
2. 实现InitializingBean接口，在类初始化完成后触发
3. 实现ApplicationListener <ContextRefreshedEvent>监听spring容器加载完触发

初始化properties文件中，供spring读取

Pamirs-schedule的设计目的

1. 让批量任务或者不断变化的任务，能够被动态的分配到多个主机的JVM中，不同的线程组中并行执行。
2. 所有的任务能够不重复，不遗漏的快速处理
3. 任务处理服务器可以动态的增加和减少
4. 可以通过域来隔离不同的执行环境
5. 可以任意指定调度的时间区间

TaskType任务类型是任务调度分配处理的单位。

ScheduleServer任务处理器

TaskQueue任务队列是对任务进行的分片划分。

TaskDealBean任务处理类是业务系统进行数据处理的实现类，要求实现Schedule的接口IScheduleTaskDealMulti或者IScheduleTaskDealSingle的方法。

OwnSign运行区域，是对运行环境的划分，进行调度任务和数据隔离。例如，开发环境、预测环境、预发环境、生成环境。

运行方式：

调度信息基于数据库存储的模式

调度信息基于zooKeeper存储的模式

使用案例

计费系统，账务系统，订购系统，物流系统，运单系统。

目的：设计一个任务调度系统

提高代码的重用性，利用redis缓存技术读取提高性能。

订单系统：高峰期一小时50万订单量如何完成补全任务？

传统的任务调度框架：

单线程处理任务，单点任务处理存在宕机风险，处理性能低下。

多线程多任务需要处理的问题

1. 怎么保证每个实例处理的数据是唯一的
2. 如果一个实例宕机，它处理的数据如何分配给其它实例
3. 多线程如何分配数据，避免数据的不重复执行。

解决方案，引入队列模型，