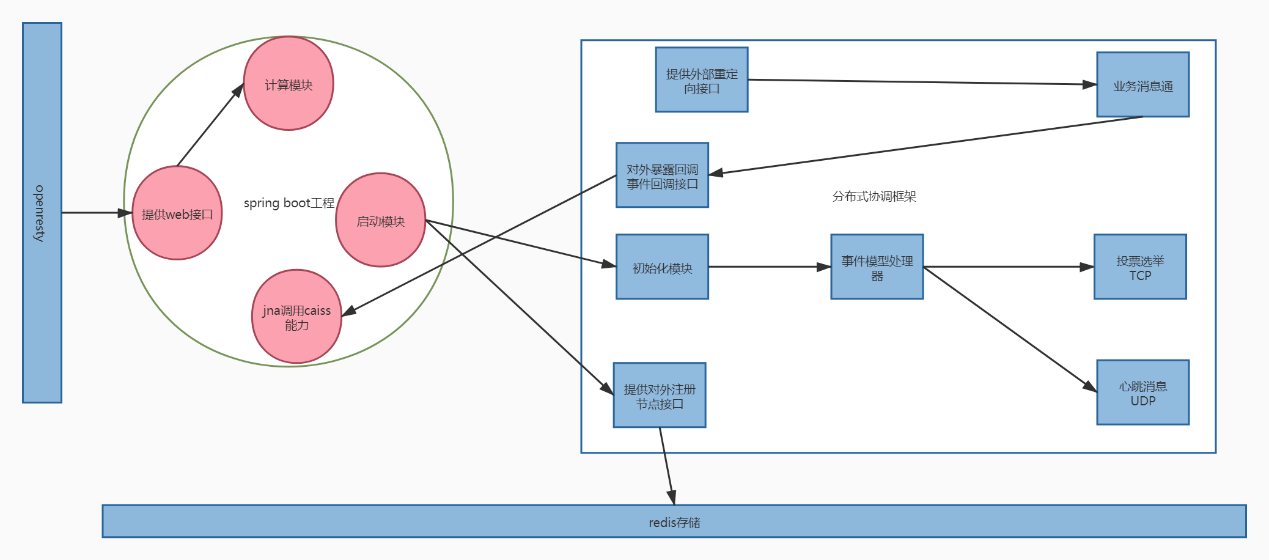
caiss-server伪代码

# 整体架构



# caiss服务层代码

## 启动模块

* 启动模块依赖springboot的启动方式，容器是依赖外部的tomcat容器
* 初始化的时候初始化话web模块，将controller层托管spring容器中（springboot中原有能力）
* 调用cluster-agent服务的初始化方法 提供本机ip和端口port

## Web模块

* 实现caiss的查询接口，创建请求唯一id，提供参数为（关键词）
* 提供路由后请求接口（唯一id，关键词）

## 计算模块

* 实现模块路由，根据请求的模型进行分类，路由到正确的节点，调用cluster-agent的接口（需要传输请求的唯一id和web请求参数，设置同步接口和异步接口，超时时间），若异步接口需要注册回调接口
* 接口调用后将结果汇聚进行排序，返回给前端

## 算法通信模块

* 用jna调用caiss算法层接口

# cluster-agent集群代理服务

cluster-agent:定位负责业务节点状态状态维护和openresty进行联动。cluster-agent负责将业务节点标记为主节点，从节点。维护每个节点的心跳，投票选举主节点以及控制请求路由。

不依赖任何框架，只依赖redis

## 初始化模块

## 分发模块

* 对外部提供注册回调地址
* 提供接口进行重定向项，支持同步和异步接口。若为异步接口，则必须注册回调地址（提供接口也可传输回调地址）
* Socket通信到其他节点，调用caiss-server接口获取结果
* 将结果返回给初始请求节点

## 心跳模块

同业务存储类型中进行心跳检测

* 初始化心跳模块初始化udp服务端口，启动定时任务每5s发一次请求，心跳只在同存储类型中发送
* 每个节点需要记录最后一次其他节点心跳记录，接收到别的节点更新进磁盘
* 若发送心跳失败，重试3次。进入某节点主管宕机，更新值至redis。若redis中已经有超过半数的觉得是宕机，及进入客观宕机状态。进入投票选举模块

## 投票选举

同业务存储类型中进行投票选举

* 从节点进行投票，互相发送当前写事务id和epoch，比较发送的epoch+事务id和当前的比较，以及ip和端口做比较，若大于则回写对方的事务id+（epoch+1）ip和端口。若id一致并且epoch是可接收的 则比较ip和port。若一致则记录一票
* 大于等于半数是更新redis（分布式锁）

发送内容，epoch（朝代），事务id（写事务id），启动的时候注册的ip:port

epoch作用：对票据进行是否有效判断，比如其他节点进行下一轮投票时，收到上一轮的票据数据。每进行一轮epoch+1.投票期间epoch不变

事务id：是写事务id，标识每次写入的事务，id是递增

Ip:port：标识投票对象

每个节点存储存储需要投票的ip:port

开始投票步骤

1. 创建客户端发送本次epoch+id 和 ip:port。（epoch+1）
2. 当接收到别的节点投票后

判断epoch是否和epoch保持一致，至少大于等于本地存储epoch。

Id比较, 若收到id大于本地id，将收到id存储如内存，并发送该id+收到ip:port

若id（收） = id（本），若ip:port一致，则投票+1。若不等于，继续存储ip:port。

并继续发送的本机节点的票据。校验是否有票据超过半数，若超过则注册主节点，发送epoch-id-ip:port-投票结束标识