# 设计文档

## 项目背景

基于已知关键词的问答库信息，实现一个基于关键词搜索匹配的智能客服服务

## 设计目标和方案设计

### 实现根据关键词的匹配搜索功能

搜索问题的常用解决方案是建立倒排索引，业界通用方案采用ES/solr构建分布式集群服务支撑复杂的查询语句和更高的并发量，然而对于此需求引入搜索集群太重了，权衡考虑采用lucene工具包实现。考虑到数据量较小，索引文件放在内存中维护，不落磁盘。

### 统计问题点击次数

需要维护一个计数器，记录各个问题的访问个数。设计考虑方案Redis，单线程原子操作，天然支持计数器功能。

### 支持高并发

设计上减少I/O环节，题库缓存到内存，redis计数可以考虑异步写入（目前按同步写入）。

整体应用模块无状态，支持水平扩展。

### Docker镜像部署

Redis使用官方；问答库基于mysql官方库和初始化数据脚本打包成单独镜像；

应用打包使用maven的Docker构建插件

## 架构设计和模块说明

### 负载均衡+反向代理

负载均衡可以用nginx；选用容器化部署方案可以引入swarm或者kubernetes等编排工具进行管理

### 业务逻辑

根据前端请求，返回候选问题和特定问题答案。为简化实现，把业务逻辑和搜索服务整合到一个模块。

### 搜索服务（可选）

如果业务场景改变，需要支持更大的数据量和更复杂的查询需求，考虑引入搜索服务。

### 计数器

如前文所述，这里选用redis做计数存储。

### 试题库管理（可选）

针对试题库的维护，考虑独立一套原子服务，如果涉及题库更新，通过消息队列通知搜索服务更新索引

## 关键技术细节

### 排序策略实现

根据需求，排序需要先根据匹配词个数，再根据点击次数做降序排序。

Lucene默认策略是根据搜索词与文档相关度进行打分，降序排列。根据默认策略，不同关键词的权重可能不同，需要替换掉打分公式中部分参数。

点击数排序则需要维护一个NumericDocValuesField类型字段，在相同打分下，点击数高的排前面。这个功能做完全实时意义不太大，设计成30秒定时更新一次索引。