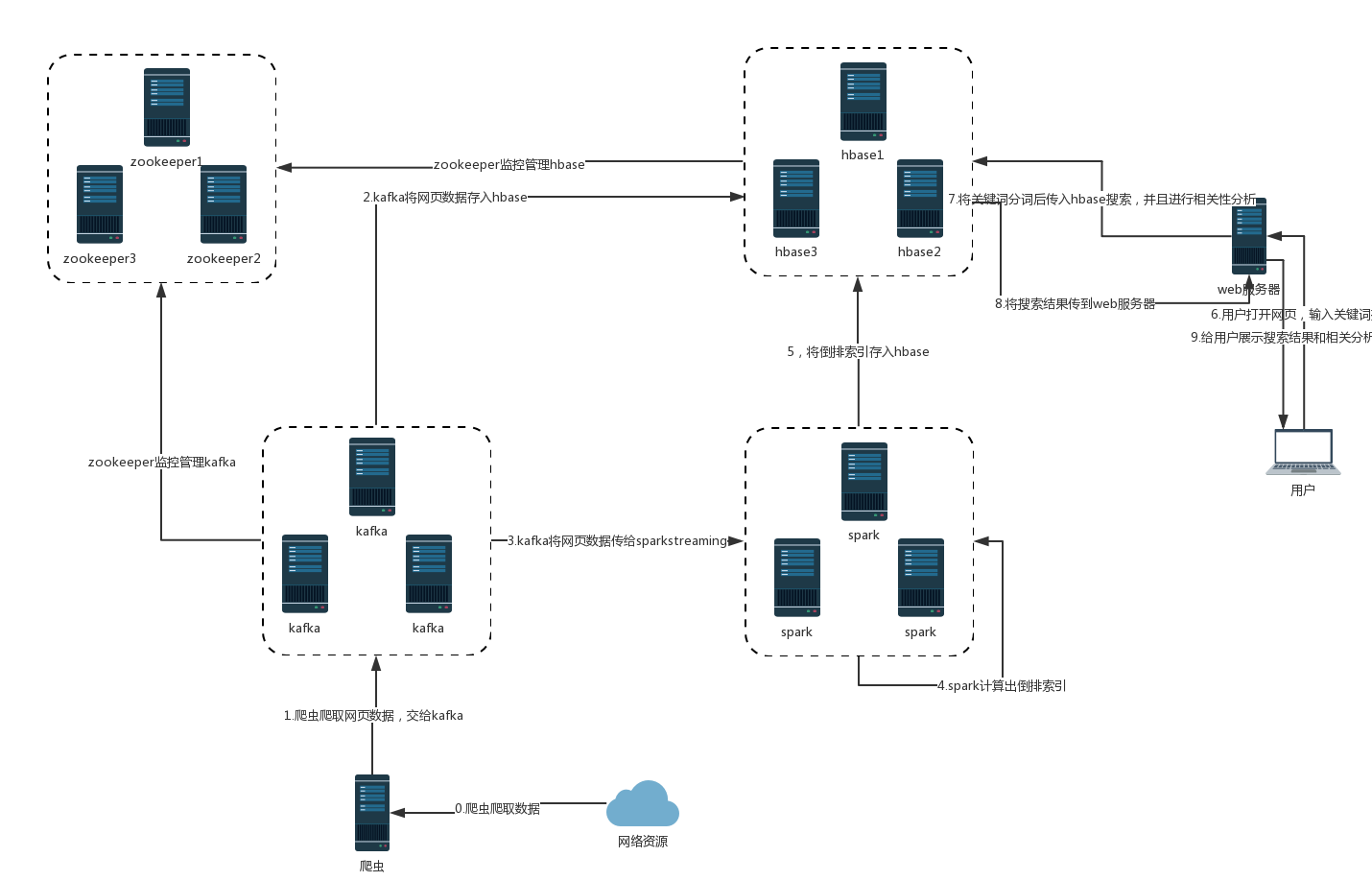
**Spark实验报告**

1. 集群说明

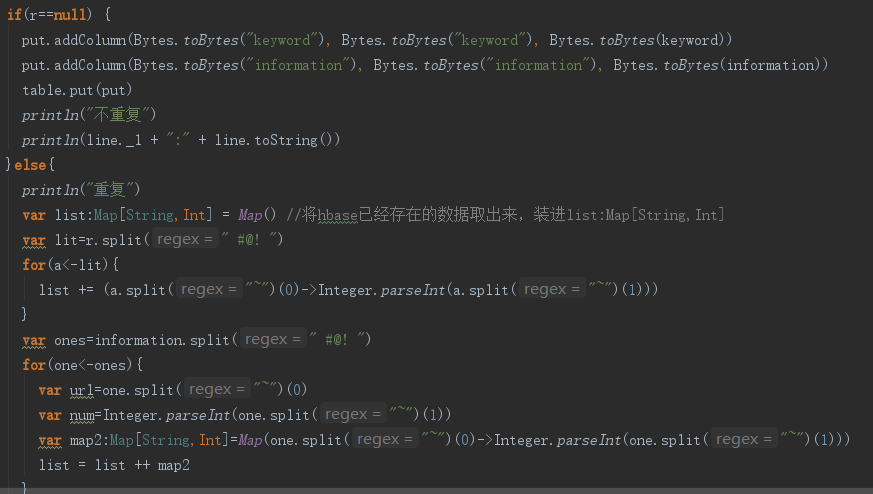


Ps: zookeeper监控管理hbase，zookeeper监控管理kafka

1. 爬虫爬取数据
2. 爬虫爬取网页数据，交给kafka
3. kafka将网页数据存入hbase
4. kafka将网页数据传给sparkstreaming
5. spark计算出倒排索引
6. 将倒排索引存入hbase
7. 用户打开网页，输入关键词搜索
8. 将关键词分词后传入hbase搜索，并且进行相关性分析
9. 将搜索结果传到web服务器
10. 给用户展示搜索结果和相关分析

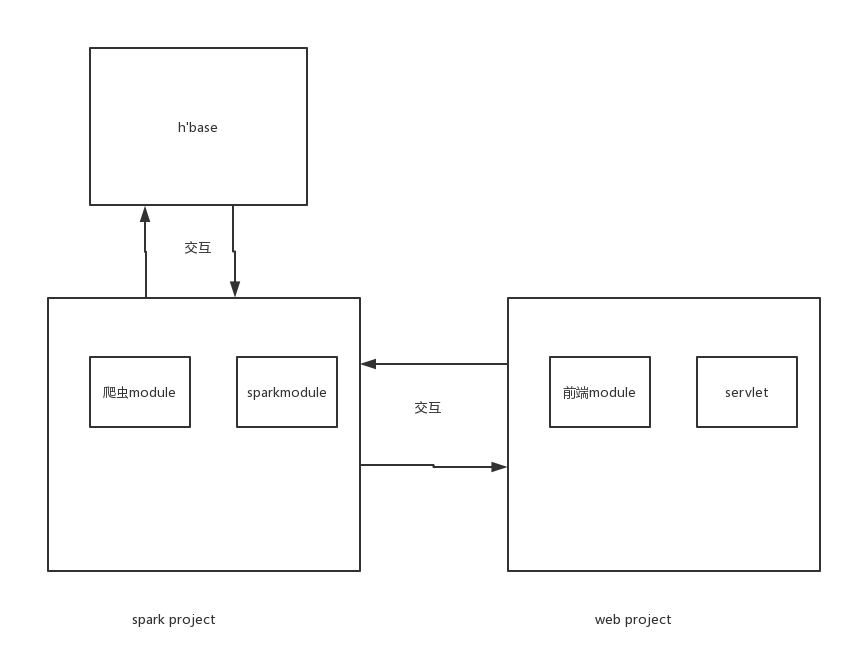
实际搭建服务器集群中，将spark和kafka集群搭建在了一起，爬虫服务器和web服务器在同一服务器上的不同工程中。

1. 算法说明
2. 本次实验采用的搜索方法为将网站标题等文字进行分词，用分得的词作为key做倒排索引，用户查询的时候，用hbase的接口进行快速搜索
3. 相关分析：本次爬取的网页主要是视频网页，额外增加一个“猜你喜欢”功能，用来对用户输入的关键词进行相关性分析，原理是根据用户搜索的最最近的几个视频的制作人，配音，演员等关键词，再次进入hbase进行搜索，返回结果并呈现。
4. 当将要存入的关键词存入倒排索引表时，如果该关键词已存在与hbase表中,就将表中的value取出，分成各个url装入map中，将新爬取的关键词也装入一个map中，将两个map合并（map合并的过程是 如果存在不同的key则将所有出现的key和对应的都存入，如果存在相同的key则，新的value覆盖掉旧的value），这样就不会丢失原来的url，也能进行实时更新

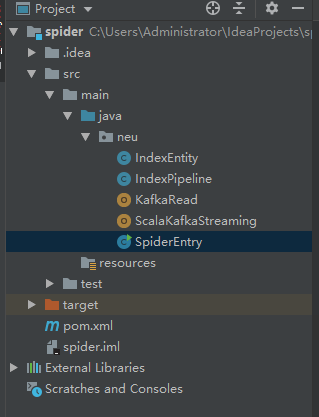


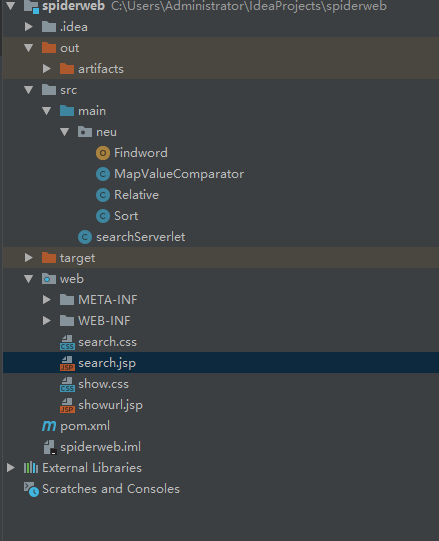


1. 程序结构



Spark程序和web程序分别创建在两个project中，能有效避免包覆盖和环境冲突

爬虫端：

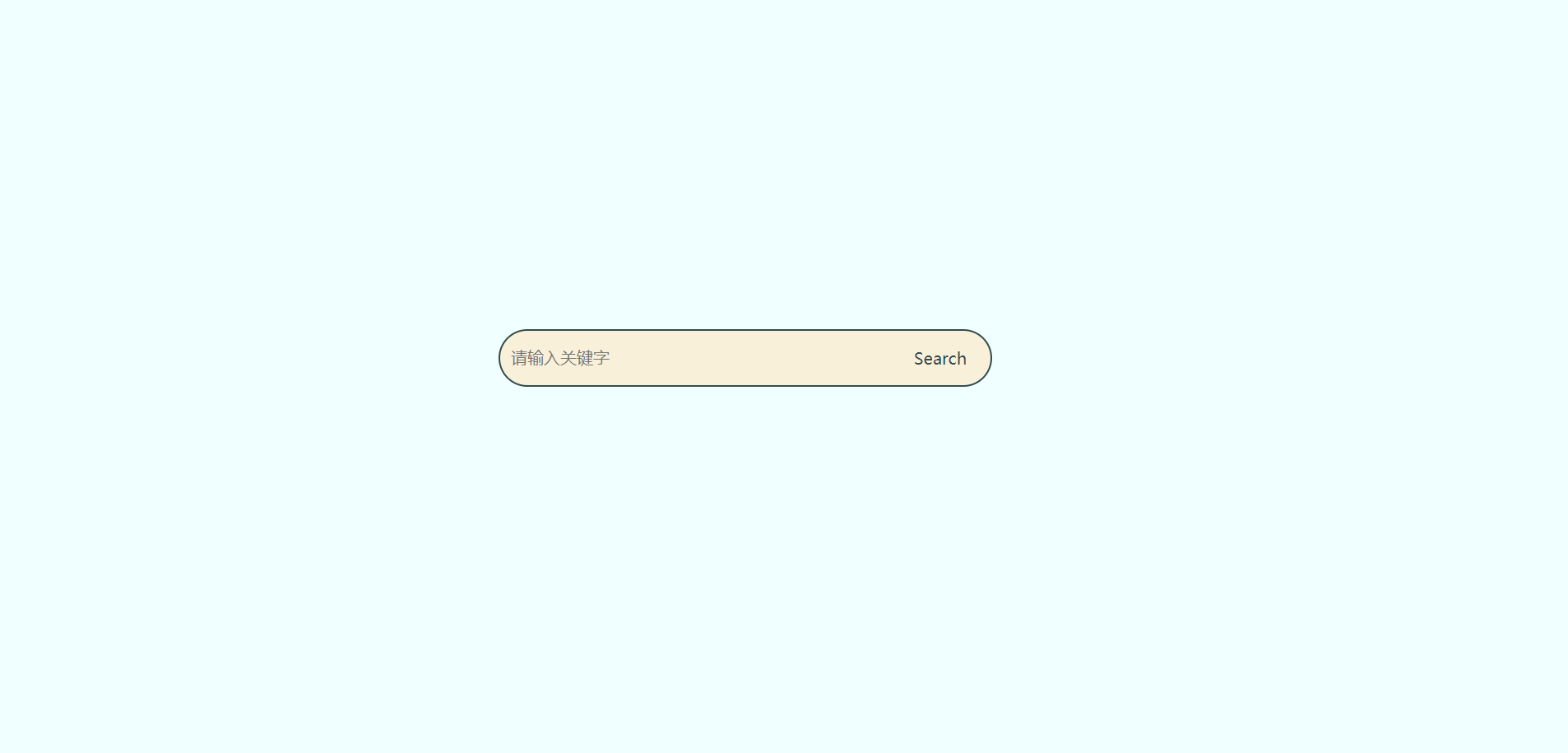
Web端：

1. 运行说明
2. 在后台服务器爬虫一直在运行，kafka在实时监听传入的网页数据，spark streaming在不断的更新索引，对hbase进行更新。
3. 当服务器运行时，需要先与hbase建立联系，这个过程稍慢，建立以后，用户再进行搜索时，速度将非常迅速。
4. 相关性分析是对关键词的搜索结果进行分析，而不是用户的关键词

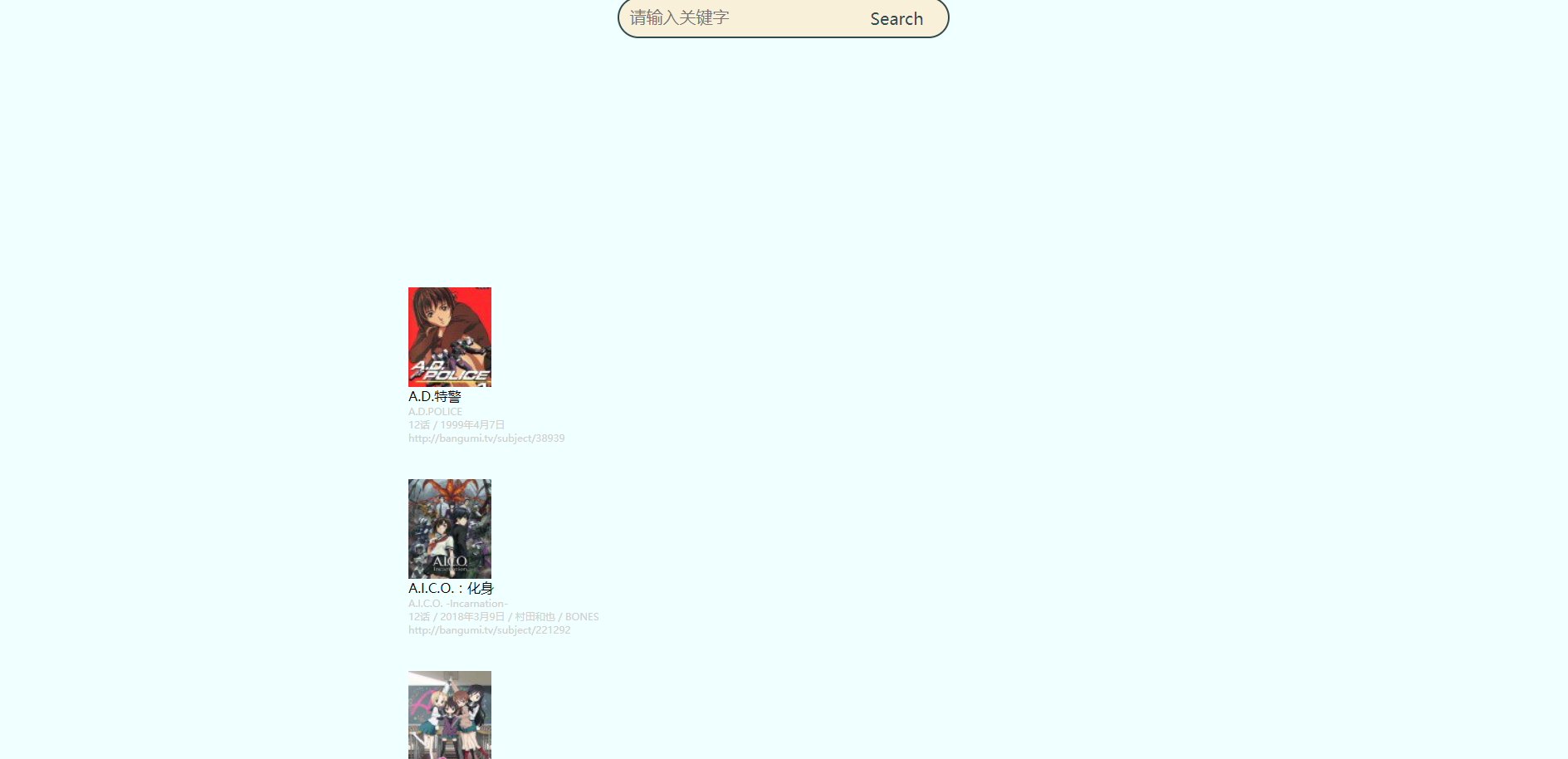




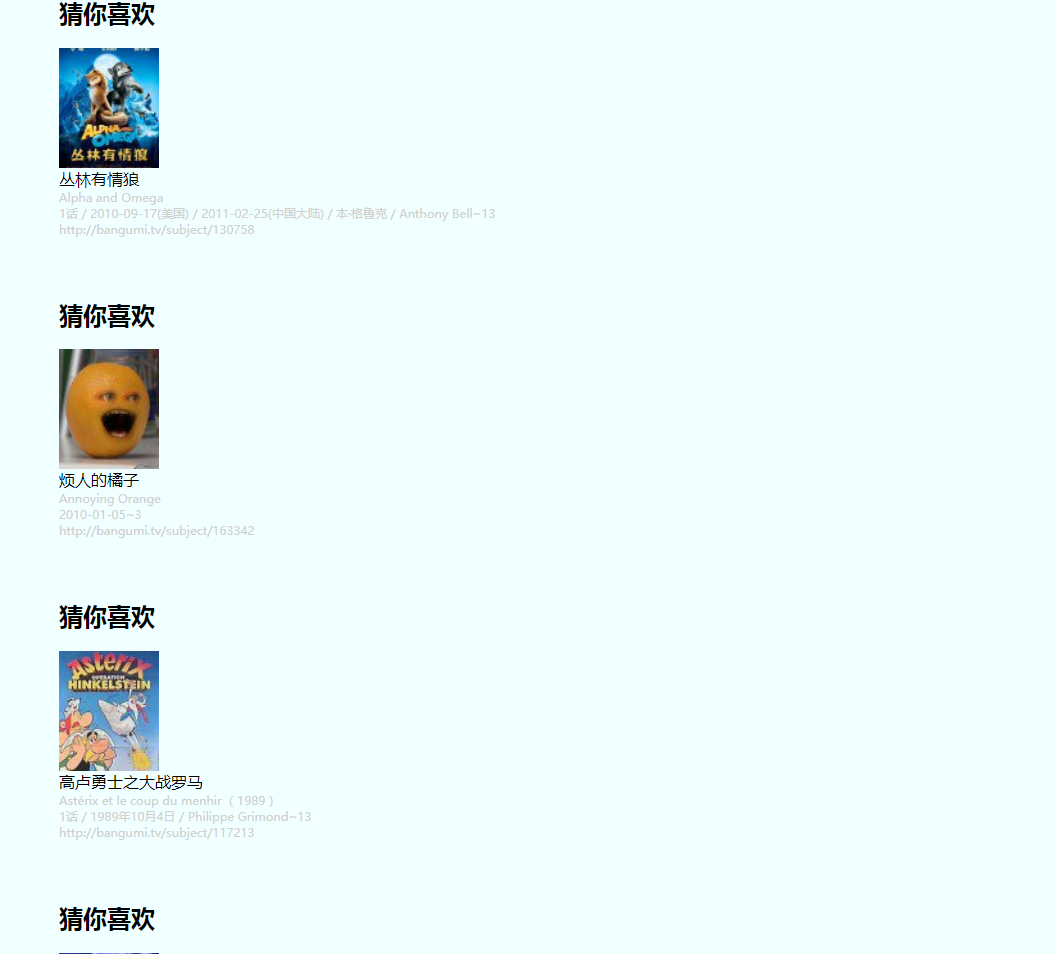
用户进入搜索界面：



输入关键词“A”后进行查询：



关联分析结果：



并且此时搜索栏缩小居上，方便进行多次搜索。

点击搜索结果，进入页面：

