# 应用阿里云搜索服务构建图书馆站内搜索引擎

王　爽　陈俊杰　肖　铮　黄国凡
(厦门大学图书馆　厦门 361005)

【摘要】利用云搜索服务已成为新的站内搜索技术方向。厦门大学图书馆采用阿里云服务重构站内搜索,将云搜索封装为独立的搜索引擎。网站数据经过预处理后提交、生成索引,传递搜索字符串到云搜索引擎即可使用云服务,实现搜索及结果呈现。评测结果表明,阿里云站内搜索与原有站内搜索相比,在检索效率和功能等多方面有显著提升。

【关键词】站内搜索　云服务　云搜索　阿里云
【分类号】TP393

## The Application of Aliyun Search Cloud Service to Build Search Engine for Library Sites

Wang Shuang　Chen Junjie　Xiao Zheng　Huang Guofan
(Xiamen University Library, Xiamen 361005, China)

【Abstract】The application of cloud search service becomes a new search technical direction. Xiamen University Library chooses Aliyun service to rebuild site search, and packages the cloud search as an independent search engine. After pre-processing, the site data is submitted and index files are generated, then the search strings are passed to the cloud search engine. Base on the cloud search service, searching finished and results displayed. Evaluation results show that Aliyun cloud search, compared with the old search engine, is significantly improved in search efficiency and functionality, etc.

【Keywords】Site search　Cloud service　Cloud search　Aliyun

云计算近年来不断发展,已从抽象的概念演变为用户身边实实在在的产品与服务。在云环境下,图书馆借助云服务的支持,基于云计算随时取用、按需支付、平台无关、易部署易扩展的特点[1],进一步提升资源建设与服务能力,已经是一种不可回避的选择[2]。

## 1　现有站内搜索技术分析

2002 年,罗良道[3]调查了国内 10 所重点高校图书馆应用站内搜索的情况,当时只有 4 所高校的图书馆网站提供站内搜索入口。随着图书馆各类服务的开展,网络资源的持续增长,大中型图书馆的网站内容日益丰富,站内搜索已成为提升用户体验的重要应用。目前这 10 所图书馆中有 7 所已在其网站提供了站内搜索入口。笔者选取这 7 所高校图书馆,在站内搜索技术平台方面与厦门大学图书馆列表比较,如表 1 所示。

由表1可以看出,目前图书馆站内搜索大多基于本地数据库或通用搜索引擎。此外,曹强[5]提出基于Lucene

收稿日期: 2013-04-18
收修改稿日期: 2013-05-30

表1　8所高校图书馆站内搜索技术平台一览

| 站点名称 | 站内搜索平台 |
|---|---|
| 北京大学图书馆 | Google |
| 清华大学图书馆 | Google |
| 浙江大学图书馆 | 本地数据库 |
| 上海交通大学图书馆 | Google |
| 南京大学图书馆 | 本地数据库 |
| 复旦大学图书馆 | 本地数据库 |
| 武汉大学图书馆 | Baidu |
| 厦门大学图书馆 | 阿里云搜索[4] |

的站内搜索实现方式;2012年和2013年两大云服务提供商(Amazon[6]和阿里云[7])相继发布了全新概念的"云搜索"服务[8,9]之后,利用云搜索服务成为一种新的站内搜索技术方向。以上4种站内搜索技术方案的比较如表2所示:

表2　4种站内搜索技术方案的比较

| 站内搜索技术方案 | 通用 | 实时 | 服务器资源 | 选择方案时需要考虑的因素 |
|---|---|---|---|---|
| 本地数据库全文搜索 | 否 | 是 | 需要 | 根据实际需求对不同数据库、不同表、不同字段建立不同的索引。搜索过程中只能采用LIKE的形式,即只能进行字段的"完全匹配",无法实现进一步的搜索智能化。 |
| 基于通用搜索引擎 | 是 | 否 | 不需要 | 实现快速,无需耗费服务器及带宽资源[10]。但由于其机制是由爬虫按照一定频率抓取网站数据更新索引,数据更新滞后。 |
| 基于开源搜索引擎(如Lucene) | 是 | 是 | 需要 | Lucene只是一个搜索引擎工具包,将其封装成搜索引擎平台需要进行后续开发,技术门槛较高。索引、查询、交互功能的实现均需本地服务器资源的支撑。 |
| 云搜索 | 是 | 是 | 不需要 | 索引和查询这两部分功能交由云服务器实现,本地只需处理用户交互,部署简便,开发量小,功能强大。 |

值得一提的是,云搜索由厂商利用自身的搜索技术对搜索服务进行高度封装,之后以服务的形式提供给开发者使用,其部署相对简便——只需开发者创建一个搜索域,将数据按规范整理上传,云平台可自动提供所需的技术资源并部署高度优化的搜索索引,还可根据被搜索的数据规模、查询率的变化情况等无缝伸缩——客户可以变更搜索参数、调整搜索相关性,并可随时应用新设置而无需重新上传数据。因其通用、实时、且无需服务器资源的优点,云搜索将成为图书馆构建站内搜索平台的全新选择。

## 2　阿里云搜索服务构建图书馆站内搜索的可行性分析

### 2.1　厦门大学图书馆站内搜索现状及需求

厦门大学图书馆新版主页于2012年12月正式发布。新版主页的栏目分类设置、内容组织方式与旧版主页有很大的不同,一些零散的子系统未能有效地组织于搜索引擎中,导致用户无法快速找到所需资源。

站内搜索需要检索的数据种类很多,包括主页栏目、商业数据库、自建数据库、免费资源、下载文档、图书馆通告等各类信息,这些信息分散存储于数据库表中,不同的表结构各异,尚有部分内容是以文件形式存放,如果采用本地数据库的全文搜索针对这些数据进行检索,不仅检索效率不高,易造成漏检,而且检索结果也无法按照相关度进行排序和推荐,无法实现为用户提供快速、正确、全面、有效的检索结果的目的。针对多样性数据内容,开发一个检索效率高,结果全面,并可进行相关度排序的搜索引擎是较为迫切的需求。

### 2.2　可行性分析

目前提供云搜索服务的有Amazon及阿里云两大云服务提供商,这两家厂商实现云搜索的技术思路、实现方式都高度相似,用户的选择主要应考虑服务质量、访问速度、费用等方面,而在这些方面Amazon CloudSearch[11]处于劣势,原因如下:

(1)中文分词是中文搜索的一大关键技术,Amazon CloudSearch主要面向欧美市场,经试用其针对中文并未进行相应优化。

(2)鉴于国内网络的国际出口带宽限制,国内访问国外网站速度较慢,而Amazon CloudSearch并未专门针对国内网络提速。

(3)阿里云搜索针对数据量少于10万的用户采取免费策略,同时不限制访问次数,而Amazon CloudSearch只提供30天的免费试用,之后将根据搜索实例、上传文档、IndexDocuments请求、数据传输的实际情况收取相应费用[12]。

厦门大学图书馆站内搜索有两个特点:数据量小,目前厦门大学图书馆主页的数据仅有2 223条;用户对于搜索结果的精确度要求较低,主要目的是导向目标栏目及子系统,因此对于搜索引擎的排序、优化无需过多干预。阿里云搜索现有的功能已经能够满足以上需求,其成熟的云服务保证了搜索引擎的稳定、快速;用户无须提供搜索引擎服务器资源,也无需关注搜索的实现细节;开发工作主要集中于整理数据、解析搜索结果即可,开发量小、难度低。经综合比较,采用阿里云搜索服务实现图书馆站内搜索是具创新意义的最佳选择。

## 3　实现方案

### 3.1　图书馆网站数据源与阿里云搜索字段的对应

笔者对厦门大学图书馆主页数据进行分析整理，将数据分为网站栏目、数据库、免费资源、馆员资料、下载文档 5 种类型，前 4 种类型数据可直接读取数据库内容生成 JSON[13] 文件，下载文档分散在不同类型的内容中，需要把所有包含下载链接的数据选取出来，并剔除文本内容，只保留下载链接的地址及文件题名，再生成相应的 JSON 文件。阿里云提供资讯模板、小说模板、游戏、APP 模板和论坛模板，根据厦门大学图书馆网站数据的特点和检索需要，笔者最终选择资讯类作为生成 JSON 文档的模板。该模板支持全文检索、标题检索；支持类型、类别过滤；支持创建时间、更新时间、点击等各种维度的排序；相关性方面除了全文匹配度外，对时效性要求高，同时还会参考点击、关注等因素，并预留 Boost（站长自定义加分项）字段方便调节该资讯排序情况[14]。最终确定的文档字段如表 3 所示：

表 3　阿里云搜索资讯类型字段与厦门大学图书馆站内搜索对应字段

| 阿里云搜索资讯类型字段说明 | | 厦门大学图书馆站内搜索对应字段说明 | |
|---|---|---|---|
| 字段名 | 类型 | 对应网站字段 | 对应字段说明 |
| Id | STRING | 自建 ID | ID 号或者网页名 |
| Title | TEXT | 网页标题 | |
| Body | TEXT | 网页内容 | |
| Type_id | UINT16 | 分类 | 厦门大学图书馆主页一级分类 |
| cat_id | UINT16 | 分面筛选 | 解决同一文档属于多个分类的问题。如数据库公告，即可在公告信息也可在资源分类下 |
| url | STRING | 网页 URL | |
| Source | TEXT | 网页来源 | 为今后集成其他网页做准备 |
| create_timestamp | UINT32 | 网页发布时间 | |
| update_timestamp | UINT32 | 网页更新时间 | |
| hit_num | UINT32 | 网页点击数 | 有则添加 |
| Boost | INT8 | 权重 | 数值越大排名越前 |
| Integer_1 | UINT32 | 二级分类 | 厦门大学图书馆主页二级分类。如 1 = 中文数据库 2 = 外文数据库 |
| Integer_2 | UINT32 | 文档类型 | 1 = 网页　2 = doc　3 = ppt　4 = pdf　5 = jpg　6 = rar　7 = exe |
| Tag | TAG | 别名 | 网页别名，数据库别名 |

需要注意的是，JSON 文档需要以 UTF－8 格式进行编码，根据 JSON 数据格式要求，在生成文档时需要对数据内容进行预处理，如剔除没有检索意义的 HTML 标记；去掉内容的制表符 Tab，字符“\”转换成“/”等。

### 3.2　图书馆网站原始文档的提交

图书馆网站数据提交到阿里云搜索服务器的过程如图 1 所示：

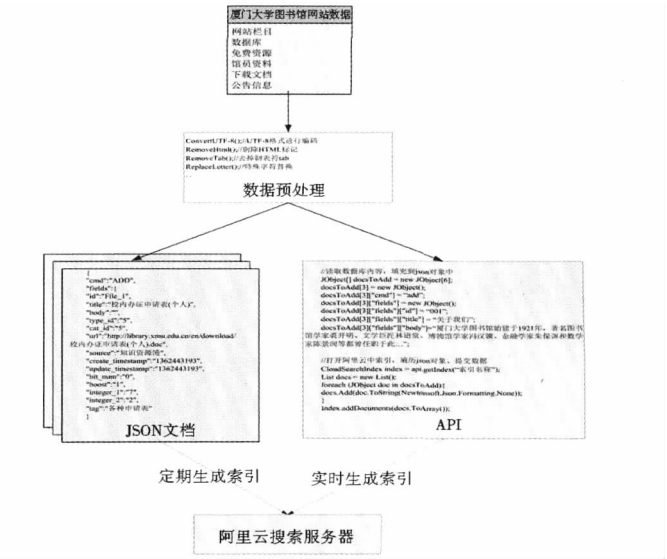


图 1　图书馆网站数据提交到阿里云搜索服务器的过程

### 3.3　将阿里云搜索结果嵌入到图书馆系统

厦门大学图书馆“站内搜索”的功能定位是整合馆内所有自建系统的数据，因此笔者将“阿里云搜索”封装为一个独立的仅提供搜索服务的小系统，实现搜索及搜索结果的呈现。其他系统只需要传递搜索字符串到该系统即可使用搜索服务，用户得到搜索结果后再引导到各个子系统，如图 2 所示：

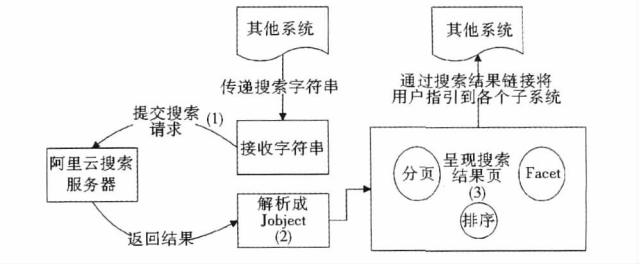


图 2　“阿里云搜索”封装后嵌入图书馆系统

采用这种实现方式的优势在于各个子系统无需分别开发阿里云的调用模块，可以极大节省开发成本。具体实现要点如下：

（1）接收参数，提交请求

如提交的链接为：

http://x.x.x.x/aliyun/s.aspx?q=民国期刊

**该链接默认传递的参数“q”是检索关键字，除此之外还可以传递“page”（页码）、“pageSize”（每页大小）。搜索子系统在解析链接参数后调用 API 进行检索：**

```
var index = api.getIndex(indexName); //打开相应索引
results = index.search("q=" + query, page.ToString(), pageSize.ToString(), sort, filter, facet, null); //调用搜索 API 获得搜索结果
```

**（2）解析返回数据**

**阿里云返回的标准 JSON 数据在 ASP.NET 平台使用 Newtonsoft.json[15] 来解析。它将返回的字符串解析成 JSON 对象，呈现数据时能够直接以访问键值对（Key－Value Pair，KVP）[16] 的方式快捷地定位信息：**

```
JObject sr = JObject.Parse(results);
```

**在实现要点（1）中调用 API 时传入 facet 参数，该参数用于指定执行 facet 的字段并设置该 facet 需要返回的数据。根据字段定义，设定 cat_id 为执行 facet 的字段，并要求 API 返回每个 cat_id 分类的搜索结果统计信息：**

```
NameValueCollection facet = new NameValueCollection();
  // 初始化 facet
facet.Add("facet_key", "cat_id");
  //定义执行 facet 的字段为 cat_id
facet.Add("facet_fun", "sum(hit_num)#count()");
  //要求 API 返回搜索结果中根据 cat_id 进行分类统计的个数
```

**通过上述设置，API 返回的数据不仅包含基本的搜索结果，还包含 facet 相关数据。**

**（3）呈现数据**

**在搜索结果页面，数据呈现包含 4 个部分，具体布局如图 3 所示：**

图 3　厦门大学图书馆站内搜索结果页面布局

①分　类

用户的二次搜索实质是进一步限定搜索范围。根据实现要点（2）解析得到的 facet 数据可将搜索结果的分类统计信息呈现出来，用户点击相应链接会触发程序在一次搜索的基础上传递一个附加的分类号作为过滤器，链接如下：

http://x.x.x.x/aliyun/s.aspx?q=民国期刊&page=1&pageSize=10&c=5

程序接收参数 c 作为过滤器：

```
string filter = null;
if (Request.QueryString["c"] != null) {
  cat = Request.QueryString["c"].ToString();
  filter = "cat_id=" + cat;
}
JObject rs = sh.Search(q, page, pageSize, orderbyTime, filter);
  // 在搜索中加入过滤器 filter
```

②排　序

搜索结果的排序采用两种方式，相关度及记录创建时间。阿里云默认的排序方式是相关度，采用相关度排序无需多余设置，而采用创建时间逆序则需要设置：

```
string[] sort = new string[]{"-create_timestamp"}; //设置排序的字段为创建时间
results = index.search(cq, page.ToString(), pageSize.ToString(), sort, filter, facet, null); // 在搜索中加入 sort 排序
```

③分　页

根据页面布局设计，分页部分只显示当前页的前后各三页及上一页、下一页的导航链接。API 返回结果中包含该检索字符串在整个库中的搜索结果数。通过提取这个数量，并配合页面的 pageSize 参数即可计算出总页数，最后提取其前后各三页的链接呈现在页面中。

```
{"result":{"total":28,}} //包含搜索结果数量的 json
```

如分页链接：

http://x.x.x.x/aliyun/s.aspx?q=民国期刊&page=2&pageSize=10

从这里可以看出，翻页的操作只是修改页码（page）部分的参数，实质是重新发送一次搜索请求。

④搜索结果

这部分是页面主体，用于将搜索结果的明细呈现给用户。笔者采用 ASP.NET 的 Repeater 控件实现数据呈现，将实现要点（2）中解析得到的 JObject 传递给相应的 Repeater 控件，布局则由页面端通过 DIV＋CSS 来实现。

```
JArray resultItems = rs["result"]["items"] as JArray;
if (resultItems != null) {
resultRp.DataSource = resultItems;
resultRp.DataBind();
}
```

**（4）其他高级功能的实现**

**除了上述功能外，阿里云还支持高级字段筛选、多索引合并检索等高级功能。**

①高级字段筛选

字段定义中的 INT 或者 UINT 类型字段都能够用于筛

选。实现方式是在调用搜索 API 时传入相应的 filter,如“filter = user_id = 5 AND price > 20”表示仅返回 user_id = 5 并且 price > 20 的文档。

②多索引合并检索

如果针对每个站点都建立了索引,则需要跨站搜索时即可利用多索引合并搜索的特性。如果是多个相同模板索引,引擎处理多个索引的方式与单个索引没有区别。如果是不同模板的索引,则引擎会根据相关性交替返回不同模板的结果。

## 4　阿里云搜索应用效果评估

**厦门大学图书馆于 2013 年 2 月 25 日启动基于阿里云搜索服务的站内搜索升级项目,3 月 7 日提交数据测试,3 月 18 日项目正式上线,快捷简便地构建了图书馆主页的站内搜索引擎,很好地弥补了原站内搜索的不足,满足了读者快速准确获取网站内容的需求。阿里云站内搜索与原有站内搜索在检索效率和功能方面的比较如表 4 所示:**

表 4　厦门大学图书馆原站内搜索与阿里云站内搜索的比较

| 比较项目 | 原站内搜索 | 阿里云站内搜索 |
|---|---|---|
| 检索时间 | 0.98 秒 | 0.017 秒 |
| 检索结果(数量) | 22 条 | 23 条 |
| 支持分类聚合 | 否 | 是 |
| 搜索服务使用量 | 56 次/日 | 100 次/日 |

(注:检索词为 CNKI)。

## 5　结　语

**阿里云搜索服务高质量、易扩展、可高度定制化的特点是厦门大学图书馆采用其构建站内搜索引擎的主要因素。随着阿里云搜索服务自身技术与功能的不断完善,图书馆可借此实现更多功能,满足更为个性化的需求。阿里云搜索服务目前处于内测阶段,提供全免费服务,未来阿里云服务的收费标准、服务的稳定性及安全性将是笔者关注的重点。**

## 参考文献:


[1] 刘炜. 图书馆需要一朵怎样的云[J]. 大学图书馆学报,2009,27(4):2-6. (Liu Wei. How Libraries Uprising with the Cloud Computing[J]. *Journal of Academic Libraries*,2009,27(4):2-6.)

[2] 孙坦,黄国彬. 基于云服务的图书馆建设与服务策略[J]. 图书馆建设,2009(9):1-6. (Sun Tan, Huang Guobin. Tactics for the Library Construction and Service Based on the Cloud Service[J]. *Library Development*,2009(9):1-6.)

[3] 罗良道. 高校图书馆 Web 站点站内搜索引擎的调研与建设[J]. 情报科学,2002,20(9):946-948. (Luo Liangdao. The Investigation and Construction of the Website Search Engine in College Library[J]. *Information Science*,2002,20(9):946-948.)

[4] 阿里云搜索[EB/OL]. [2013-05-25]. http://css.aliyun.com/?spm=5176.383518.0.41.G4lGiM. (Aliyun Cloud Search[EB/OL]. [2013-05-25]. http://css.aliyun.com/?spm=5176.383518.0.41.G4lGiM.)

[5] 曹强. 基于 Lucene 的 Web 站点站内全文检索系统的设计与实现[J]. 图书情报工作,2007,51(9):124-126,144. (Cao Qiang. Design and Implementation of a Website Full-text Retrieval System Based on Lucene[J]. *Library and Information Service*,2007,51(9):124-126,144.)

[6] Amazon Web Services[EB/OL]. [2013-05-25]. http://aws.amazon.com/cn/.

[7] 阿里云[EB/OL]. [2013-05-25]. http://www.aliyun.com/. (Aliyun[EB/OL]. [2013-05-25]. http://www.aliyun.com/.)

[8] 36 氪. Amazon 的 AWS 推出云搜索服务 CloudSearch[EB/OL]. [2013-05-25]. http://www.36kr.com/p/99382.html. (36Kr. Amazon AWS Launched CloudSearch Service[EB/OL]. [2013-05-25]. http://www.36kr.com/p/99382.html.)

[9] 36 氪. 搅动搜索市场,阿里巴巴推出自己的搜索服务“阿里云·搜索”[EB/OL]. [2013-05-25]. http://www.36kr.com/p/201388.html. (36Kr. Stir the Search Market, Alibaba Launched Its Own Search Service “Aliyun Search”[EB/OL]. [2013-05-25]. http://www.36kr.com/p/201388.html.)

[10] 徐芳. 基于通用搜索引擎实现站内搜索的二次开发[J]. 现代图书情报技术,2009(5):81-85. (Xu Fang. The Secondary Development of Site Search Based on Common Search Engines[J]. *New Technology of Library and Information Service*,2009(5):81-85.)

[11] Amazon CloudSearch (beta)[EB/OL]. [2013-05-25]. http://aws.amazon.com/cn/cloudsearch/.

[12] Amazon CloudSearch 定价[EB/OL]. [2013-05-25]. http://aws.amazon.com/cn/cloudsearch/pricing/. (Amazon CloudSearch Pricing[EB/OL]. [2013-05-25]. http://aws.amazon.com/cn/cloudsearch/pricing/.)

[13] Introducing JSON[EB/OL]. [2013-05-25]. http://www.json.org/index.html.

[14] 模版[EB/OL]. [2013-05-25]. http://css.aliyun.com/wiki/index.php/模版. (Template[EB/OL]. [2013-05-25]. http://css.aliyun.com/wiki/index.php/模版.)

[15] JSON.NET[EB/OL]. [2013-05-25]. http://json.codeplex.com/.

[16] Key-Value Pair(KVP)[EB/OL]. [2013-05-25]. http://searchenterprisedesktop.techtarget.com/definition/key-value-pair.


(作者 E-mail: fandog@gmail.com)