XUNSEARCH 10分钟入门

Created by: hightman

2012年11月22日

关于 XUNSEARCH

概况

Xunsearch 一个是以 GPL 协议开源发布的高性能、全功能的全文检索解决方案，并针对中文深度优化和处理，用于帮助开发者针对海量数据快速建立搜索引擎。

Xunsearch 采用结构化分层设计，包含后端服务器和前端开发包两大部分。后端是用 C/C++ 基于 Xapian (读 /zap-ian/) 搜索库、SCWS 中文分词、libevent 等开源库开发，借鉴了 nginx 的多进程多线程混合工作方式，是一个可承载高并发的高性能服务端。前端则是使用流行的脚本语言编写的开发工具包 (即 SDK)，API 简单清晰上手容易，并附带全中文的示例代码、文档和辅助工具，目前只支持 PHP 语言。

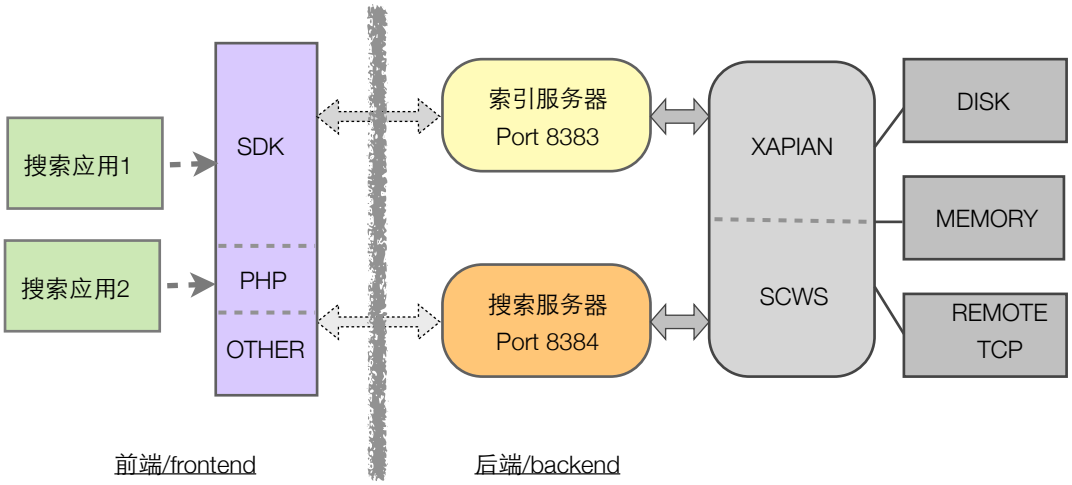
特色优势

- 海量数据下高速搜索响应。单库最多支持 40 亿条数据，在 500 万张网页1.5TB 数据中，非缓存检索时间约 0.5 秒。
- 专为搜索而自主开发的 scws 中文分词，支持复合分词、自定义补充词库，保障查全率、准确率。
- 健壮稳定的后端守护程序，内置缓存池与线程池用于保障性能。
- 索引接口齐全易用，支持实时搜索，支持任何数据库源 (不局限于 SQL)。
- 极低的开发难度，具备规范的中文文档，示范代码，辅助工具。
- 除通用搜索引擎功能外，还内置支持拼音检索、分面搜索、相关搜索、同义词搜索、搜索纠错建议等专业功能。
- 与 Lucene/Sphinx 等相比，xunsearch 提供了更丰富且必需的功能，开发难度更低，开发周期更短。

应用领域

- 目前后端服务器只支持 UNIX (含Linux/BSD/MacOS等) 操作系统，前端开发包只支持 PHP 语言。
- Xunsearch 可以帮助您建立门户/垂直搜索/论坛搜索/WEB站内搜索/文档文献资料搜索等。

架构简图



安装

服务端

前端的 Xunsearch PHP-SDK 与服务端通讯协同工作，要想使用 xunsearch 搜索就必须先安装服务端，目前只支持 UNIX 类型的操作系统 (含 Linux/BSD/MacOS在内) 以源码方式编译安装，暂不支持 Windows。因此也要求您的服务器上你必须装有 gcc、make 等软件包编译安装工具。

1. 下载&解压安装包：http://www.xunsearch.com/download/xunsearch-full-latest.tar.bz2

2. 强烈推荐用 $HOME/xunsearch 或 /usr/local/xunsearch 作为安装目录 (以下简称 $prefix)。无论您是首次安装 xunsearch 还是升级新版本，均只要直接执行我们提供的安装脚本，输入安装目录然后耐心等待即可。

```
cd xunsearch-full-1.3.3 ; sh setup.sh
```

3. 安装完毕后，您就可以通过自带的脚本 ($prefix/bin/xs-ctl.sh) 启动/关闭 xunsearch 服务端了。用法举例：

```
$prefix/bin/xs-ctl.sh start          # 默认启动，绑定本地的 8383/8384 端口
$prefix/bin/xs-ctl.sh -b inet start # 绑定全部 IP，适合 SDK/服务端 不同服务器的情况
$prefix/bin/xs-ctl.sh stop           # 停止服务器，若启动时指定了 -b inet 此处也必须指定
```

4. 没错，安装就是这么简单。特别提示，搜索的所有索引数据将被保存到 $prefix/data 目录，因此如果您希望数据目录另行安排，请采用软连接形式确保 $prefix/data 链至真实数据目录。此外，如果服务端启动时使用了 -b inet 参数，那么请借助 iptables 或其它防火墙工具进行保护，xunsearch 本身出于性能考虑不做其它验证处理。

PHP-SDK

PHP-SDK 的代码默认包含在服务端安装目录中，即 $prefix/sdk/php 。目录结构如下：

```
|-- doc/                    -- HTML 格式的文档、API手册
|-- app/                    -- 搜索项目 ini 文件的默认存储目录
|-- lib/XS.php              -- 搜索库唯一文件，所有搜索相关功能均必须引入此文件
\-- util/                   -- 辅助工具目录
    |-- RequireCheck.php    -- 检测您的 PHP 环境是否符合 xunsearch 运行条件
    |-- Quest.php           -- 搜索测试工具
    \-- Indexer.php         -- 索引管理工具
```

1. 如果您的搜索应用和服务端在同一机器，则无需其它操作，只需在开发时直接引入 $prefix/sdk/php/lib/XS.php 即可。

2. 如果您的搜索应用和服务端不在同一机器，则请复制 $prefix/sdk/php 目录到相应的搜索应用服务器，同时出于安全考虑，建议不要放到 WEB 可访问的目录。

3. 使用 SDK 中的 util 工具要求您的 php(cli) 位于可执行文件默认搜索路径中 (即用 which php 可以检测到)，如不在请做好软链接至 /usr/local/bin/php。

检测运行环境

Xunsearch 要求 PHP 是 5.2.0 及以上版本，强烈推荐使用 5.3.x 系列的 PHP。请在安装完毕后直接执行 $prefix/sdk/php/util/RequireCheck.php 看输出即可。如果您的终端编码不是 UTF-8 请在调用时加上 -c gbk 参数。

DEMO 服务器

某些情况下，部分用户未能自己部署安装服务端，而又想体验 xunsearch。因此，我们从发布 1.3.3 版本起，提供了一台供用户测试的 DEMO 服务器。用户无需安装服务端，直接下载 PHP-SDK 就可以开发测试。特别提示，DEMO 服务器只用于测试目的，并会不定期重置数据。

单独下载 PHP-SDK

如果您没有安装服务端，想直接体验 DEMO 服务器的，您只要从下面地址下载解压 SDK 压缩包即可。
http://www.xunsearch.com/download/xunsearch-sdk-latest.zip

解压后得到 xunsearch-sdk 目录，相应的搜索库文件为 xunsearch-sdk/php/lib/XS.php

服务器地址

索引服务器：demo.xunsearch.com 端口 9393
搜索服务器：demo.xunsearch.com 端口 9394

开始

开发流程

- 为便于讲解说明，假定 PHP-SDK 代码目录为 $sdk。
- 分析搜索需求，设计搜索应用必需的字段。
- 编写项目配置文件，项目配置 ini 文件存放在 $sdk/app 目录。
- 引入 $sdk/lib/XS.php 进行搜索功能和界面开发，借助 $sdk/util/*.php 工具进行测试或调试。

认识对象

- XS -- 搜索项目总对象，所有相关操作均基于此对象及子方法。
- XSDocument -- 搜索结果或索引文档，包括一组字段及值，相当于 SQL 表中的一条记录。
- XSIndex -- 索引管理，通过 XS 对象的 index 属性取得。
- XSSearch -- 搜索功能，通过 XS 对象的 search 属性取得。
- XSException -- 异常类型，必须捕捉此异常以判断操作是否正确，例：

```
require '$sdk/lib/XS.php';  // 引入 xunsearch sdk
try {
    $xs = new XS('demo');  // demo 为项目名称，配置文件是：$sdk/app/demo.ini
    // ... 此外为其它 XSIndex/XSSearch 的相关功能代码
} catch (XSException $e) {
    echo $e . "\n" . $e->getTraceAsString() . "\n"; // 发生异常，输出描述
}
```

编写配置文件

推荐使用我们的在线工具编写：http://www.xunsearch.com/tools/iniconfig　demo项目的配置如下：

创建索引

获取 XSIndex 对象

```
require '$sdk/lib/XS.php';
try {
    $xs = new XS('demo'); // 创建 XS 对象，项目名称为：demo
    $index = $xs->index;  // 获取索引对象
    // ... 在此编写过索引处理代码 ...
} catch (XSException $e) { }
```

增删改

```
$doc = new XSDocument(array(  // 创建 XSDocument
    'pid' => 123, // 主键字段，必须指定
    'subject' => '测试文档标题', 'message' => '测试文档内容',
    'chrono' => time()
));
$index->add($doc);  // 添加文档，不检测便索引库内是否已有同一主键数据
$index->update($doc); // 更新文档，若有同主键数据则替换之
$index->del('123'); // 删除主键值为 234 的文档
$index->del(array('123','456')); // 删除主键值为 123 及 456 的文档
```

清空索引

当搜索字段文件变更或出现严重不同步时，建议清空索引。

```
$index->clean(); // 一执行立即生效
```

索引同步

出于性能优化设计，上面所看到的索引操作都是异步操作。也就是说您通过 PHP 调用的 API 返回后，索引变动是先保存在服务端的队列中，由服务端根据负荷延迟一并写入索引库。这个时间差我们控制在合理范围内，通常是几秒钟时间。但如果您在批量更新后希望立即同步，可以这样：

```
$index->flushIndex();
```

使用搜索

获取 XSSearch 对象

```
require '$sdk/lib/XS.php';
try {
    $xs = new XS('demo'); // 创建 XS 对象，项目名称为：demo
    $index = $xs->search;  // 获取搜索对象
    // ... 在此编写过搜索处理代码 ...
} catch (XSException $e) { }
```

搜索语法

- 查询语句和流行的搜索引擎相似，通过空格把搜索词、句连接起来即可，字段检索使用 field:XXX 的格式。
- 允许使用 AND/OR/NOT/XOR 等显式地布尔关系组合，可以使用小括号 () 包围表达优先级。
- 支持使用双引号对较长搜索词进行精确匹配，要求字段设计时勾选"精确"项。

```
$search->search('杭州 西湖'); // 搜索同时包含这2个词的结果
$search->search('杭州 OR 西湖'); // 搜索包含其中一个词的结果
$search->search('subject:杭州 西湖'); // 包括西湖并且标题包含杭州的结果
```

获取结果

- 设置数量及偏移

```
$search->setLimit(5, 15); // 设置最多返回 5条，并跳过前 15条，即返回第 16-20 条结果
```

- 获取搜索结果

```
$docs = $search->setQuery('测试')->search(); // 搜索 '测试'
foreach ($docs as $doc) {
    $subject = $search->highlight($doc->subject); // 高亮处理标题
    echo $doc->rank() . '. ' . $subject . ' [' . $doc->percent() . '%] - ' . date('Y-m-d') . "\n";
    echo $doc->message . "\n\n";
}
```

- 获取搜索结果数量 (估算值)

```
$count = $search->getLastCount(); // 获取最后一次 $search->search() 的匹配数量
$count = $search->count('测试'); // 直接检索包含 '测试' 的数量
```

搜索日志

关于日志

系统内部会自动记录并分析搜索关键词日志，通过日志衍生出相关的扩展功能。日志同样是异步更新的并且延迟较大，如需要强制刷新请调用以下指令或索引 API。

```
php $sdk/util/Indexer.php -p demo --flush-log  # 通过辅助工具刷新日志
$index->flushLogging(); // 通过索引 API 更新
```

热门搜索

通过 XSSearch::getHotQuery 方法获取热门搜索词，返回的数组以关键词为键名，搜索指数为值。

```
$words = $search->getHotQuery(); // 获取前 6个总热门搜索词
$words = $search->getHotQuery(6, 'lastnum'); // 获取前 10个上周热门词
```

相关搜索

通过 XSSearch::getRelatedQuery 方法获取热门搜索词，返回相关搜索词组成的数组。

```
$words = $search->getRelatedQuery(); // 获取前 6个和最近一次 setQuery() 相关的搜索词
$words = $search->getRelatedQuery('测试', 10); // 获取 10个和 '测试' 相关的搜索词
```

搜索纠错

由于输入过快或拼音输入中文很容易出现错误，XSSearch::getCorrectedQuery 方法返回纠正后的关键词组成的数组。

```
$docs = $search->setQuery('侧试')->search(); // 正常进行搜索误打的 '侧试'
$corrected = $search->getCorrectedQuery(); // 尝试修正
if (count($corrected) > 0) { // 有修正词则列出
    echo "您是不是要找：\n";
    foreach ($corredted as $word) echo $word . "\n";
}
```

搜索建议

类似常见搜索引擎那样，当用户在输入框键入少量字、拼音、声母时进行智能补全，可以节省用户的输入时间。

```
$words = $search->getExpandedQuery('c'); // 返回 array('测试')
$words = $search->getExpandedQuery('测'); // 返回 array('测试')
$words = $search->getExpandedQuery('cs'); // 返回 array('测试')
```

使用工具

为了便于用户开发调试，我们在 $sdk/util 目录提供了一套辅助工具。输出结果默认为 UTF-8 编码，如果发生乱码请测试在所有命令后加上 -c gbk 以修正编码。

util.Indexer

支持批量导入索引、清空索引、刷新提交、同义词管理等，详细请参见 php util/Indexer.php --help。

```
php util/Indexer.php demo --clean  # 清空 demo 项目的索引数据
php util/Indexer.php demo --flush  # 刷新未写入的索引队列
# 导入 MySQL 的 dbname.tbl_post 表到 demo 项目并采用平滑重建方式
php util/Indexer.php demo --rebuild --source="mysql://root:pass@localhost/dbname" --sql="SELECT * FROM tbl_post"
```

util.Quest

功能齐全的综合搜索测试工具，详细请参见 php util/Quest.php --help。

```
php util/Quest.php demo 测试  --limit 3  # 搜索 demo 项目中包含 "测试" 的数据，并最多只返回 3条
php util/Quest.php --suggest demo cs  # 列出以"cs" 开头的搜索建议
php util/Quest.php --correct demo 侧试  # 列出 "侧试" 的修正词
```

util.Logger

搜索日志管理，支持删除、修改、刷新、清空等功能，详细请参见 php util/Logger.php --help。

```
php util/Logger.php demo --flush  # 刷新 demo 项目搜索日志
php util/Logger.php demo  # 查看 demo 项目的热门搜索词
php util/Logger.php demo 测试  # 查看 demo 项目和 "测试" 相关的搜索词
php util/Logger.php demo --clean  # 清空 demo 项目搜索日志
php util/Logger.php demo --del="word1,word2"  # 删除搜索日志中的 word1 和 word2
php util/Logger.php demo --put=word  # 添加 word 到 demo 项目搜索日志中
```

util.SearchSkel

该工具读取并分析项目配置文件 (NAME.ini)，然后生成一个通用适合 WEB 访问的搜索代码骨架。您可以在此基础上补充和修改代码即可，能大幅提升开发效率。详细请参见 php util/SearchSkel.php --help。

```
php util/SearchSkel.php demo /path/to/web  # 生成 demo 项目的搜索骨架，生成结果在  /path/to/web/demo
```

搜索骨架代码

```
|-- search.php          -- 搜索入口页面，可放入 web 直接访问用于测试
|-- search.tpl          -- 搜索结果输出模板文件
\-- suggest.php         -- 提取搜索建议，通过 jQuery.AutoComplete 插件调用
```

使用SCWS

SCWS 分词已经被内置包含到 xunsearch 服务端，因此用户也可以直接通过 SDK 代码使用 scws 分词功能，中文分词是计算机中文语言处理的重要环节。

分词设置

在 SDK 中必须选创建 XS 对象再创建 XSTokenizerScws 对象，在此基础上使用 scws 分词。

```
$xs = new XS('demo');  // 创建初始 XS 项目，否则无法定位服务端会抛出异常
$tokenizer = new XSTokenizerScws;  // 创建分词对象实例
$tokenizer->setIgnore(true);  // 让返回的分词结果忽略标点符号
$tokenizer->setDuality(true);  // 对分词结果中的连续单字做二元组合
$tokenizer->setMulti(3); // 设置复合分词方案：0x01-长词切为短词，0x02-单字二元，0x04-重要单字，0x08-全部单字
```

获取分词结果

返回的结果数组每个元素包含：该词在文本中的位置(off)，词性(attr)，词内容(word)。

```
$text = "迅搜(xunsearch)是优秀的开源全文检索解决方案";
$words = $tokenizer->getResult($text); // 返回分词结果
print_r($words);  // 打印分词结果
```

提取重要词

根据指定的词性，从给定文本中根据词语的重要性、出现率提取重要词列表，返回词中的 times 表示出现次数。

```
$text = "迅搜(xunsearch)是优秀的开源全文检索解决方案";
$tops = $tokenizer->getTops($text, 5, 'n,v,vn'); // 提取前 5 个重要词，要求词性必须是 n 或v 或 vn
print_r($tops);  // 打印重要词列表
```

更多

如果您在使用或学习 xunsearch 中，有任何意见或建议都可以告诉我们。

联系方式

- **QQ群** 14413875
- **电子邮箱** support@xunsearch.com
- **交流论坛** http://bbs.xunsearch.com

常用网址

- **官方网站** http://www.xunsearch.com
- **代码仓库** https://github.com/hightman/xunsearch
- **下载地址** http://www.xunsearch.com/download
- **在线文档** http://www.xunsearch.com/doc/php

截图

搜索首页

搜索结果

搜索建议

搜索纠错