**Sor与mysql全文索引比较**

**一.solr提供了好几种目录工厂类，分别如下：** 1：solr.StandardDirectoryFactory,这是一个基于文件系统存储目录的工厂，它会试图选择最好的实现基于你当前的操作系统和Java虚拟机版本。  
 2：solr.SimpleFSDirectoryFactory,适用于小型应用程序，不支持大数据和多线程。  
 3：solr.NIOFSDirectoryFactory,适用于多线程环境，但是不适用在windows平台（很慢），是因为JVM还存在bug。  
 4：solr.MMapDirectoryFactory,这个是solr3.1到4.0版本在linux64位系统下默认的实现。它是通过使用虚拟内存和内核特性调用mmap去访问存储在磁盘中的索引文件。它允许lucene或solr直接访问I/O缓存。如果不需要近实时搜索功能，使用此工厂是个不错的方案。  
 5：solr.NRTCachingDirectoryFactory,此工厂设计目的是存储部分索引在内存中，从而加快了近实时搜索的速度。  
 6：solr.RAMDirectoryFactory,这是一个内存存储方案，不能持久化存储，在系统重启或服务器crash时数据会丢失。且不支持索引复制。

**二.下面我们对比solr全文索引与mysql全文索引的优势**

1.速度方面

我们在数据库中存放了200万条数据，分别使用mysql ngram（ngram\_token\_size=1）中文分词器建立mysql全文索引和使用solr.NRTCachingDirectoryFactory建立索引，测试结果表明，在查询字段一直，分页条件一样的情况下mysql查询速度为1.5s左右时，solr的查询速度为0.1s左右，如果solr使用solr.RAMDirectoryFactory的情况下，查询速度会再提高10倍左右，但solr.RAMDirectoryFactory建立的索引只存在内存中，而不存在磁盘中，服务重启后需要重新建立

另外，mysql的全文索引使用场所特别受限，这主要是因为其对中文分词的支持太差，要想支持中文，需要设置ngram\_token\_size=1，这样的话也增加了索引的分词数，占用更多磁盘

1. 关于架构的可扩展性问题

由于solr是单独部署在tomcat下,其可以单独使用一台机器来部署，索引全部存储到部署机上，而mysql全文索引是与数据库文件是存在一起的，这样会加大mysql服务器的压力，降低其查询必能,同时solr可以做集群，进一步分摊压力，提交并发能力

1. 搜索的准确性，由于mysql是通过ngram参数来设置中文分词的，要想支持一个字符的搜索，必须设置ngram\_token\_size=1,即使这样，当我查询内容中包含“衣服”字段的记录时，mysql可能返回的记录第一条包含“衣”，第二第包含“服”,第三条包含 “衣服 ”，而solr会将包含“衣服”的记录首先放到第一条，这正是我们想要的搜索准确性
2. 高亮支持，solr可以设置内容中匹配的字符高亮显示，而mysql无法做到