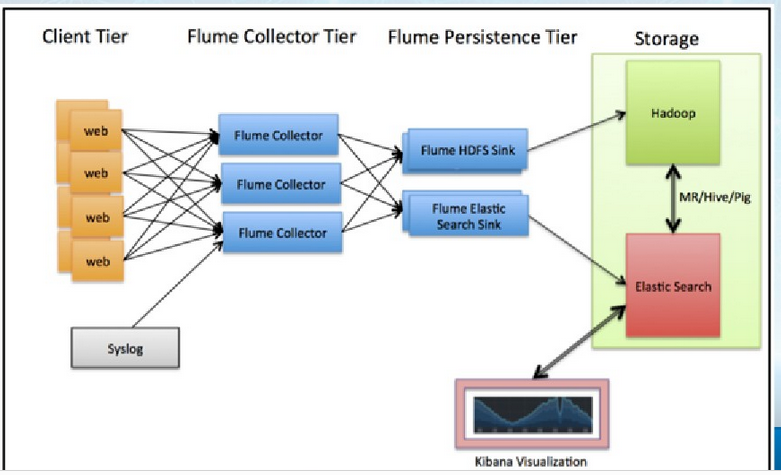
# 对比

## Elasticsearch简介



上图为hadoop和大数据的融合

Elasticsearch是一个实时的分布式搜索和分析引擎。它可以帮助你用前所未有的速度去处理大规模数据。

它可以用于全文搜索，结构化搜索以及分析，当然你也可以将这三者进行组合。

Elasticsearch是一个建立在全文搜索引擎 Apache Lucene™ 基础上的搜索引擎，可以说Lucene是当今最先进，最高效的全功能开源搜索引擎框架。

但是Lucene只是一个框架，要充分利用它的功能，需要使用JAVA，并且在程序中集成Lucene。需要很多的学习了解，才能明白它是如何运行的，Lucene确实非常复杂。

Elasticsearch使用Lucene作为内部引擎，但是在使用它做全文搜索时，只需要使用统一开发好的API即可，而不需要了解其背后复杂的Lucene的运行原理。

当然Elasticsearch并不仅仅是Lucene这么简单，它不但包括了全文搜索功能，还可以进行以下工作:

分布式实时文件存储，并将每一个字段都编入索引，使其可以被搜索。

实时分析的分布式搜索引擎。

可以扩展到上百台服务器，处理PB级别的结构化或非结构化数据。

这么多的功能被集成到一台服务器上，你可以轻松地通过客户端或者任何你喜欢的程序语言与ES的RESTful API进行交流。

Elasticsearch的上手是非常简单的。它附带了很多非常合理的默认值，这让初学者很好地避免一上手就要面对复杂的理论，

它安装好了就可以使用了，用很小的学习成本就可以变得很有生产力。

随着越学越深入，还可以利用Elasticsearch更多高级的功能，整个引擎可以很灵活地进行配置。可以根据自身需求来定制属于自己的Elasticsearch。

### 使用案例：

维基百科使用Elasticsearch来进行全文搜做并高亮显示关键词，以及提供search-as-you-type、did-you-mean等搜索建议功能。

英国卫报使用Elasticsearch来处理访客日志，以便能将公众对不同文章的反应实时地反馈给各位编辑。

StackOverflow将全文搜索与地理位置和相关信息进行结合，以提供more-like-this相关问题的展现。

GitHub使用Elasticsearch来检索超过1300亿行代码。

每天，Goldman Sachs使用它来处理5TB数据的索引，还有很多投行使用它来分析股票市场的变动。

但是Elasticsearch并不只是面向大型企业的，它还帮助了很多类似DataDog以及Klout的创业公司进行了功能的扩展。

### Elasticsearch的优缺点:

#### 优点

Elasticsearch是分布式的。不需要其他组件，分发是实时的，被叫做”Push replication”。

Elasticsearch 完全支持 Apache Lucene 的接近实时的搜索。

处理多租户（multitenancy）不需要特殊配置，而Solr则需要更多的高级设置。

Elasticsearch 采用 Gateway 的概念，使得完备份更加简单。

各节点组成对等的网络结构，某些节点出现故障时会自动分配其他节点代替其进行工作。

#### 缺点

还不够自动（不适合当前新的Index Warmup API）

## Solr简介

Solr（读作“solar”）是Apache Lucene项目的开源企业搜索平台。其主要功能包括全文检索、命中标示、分面搜索、动态聚类、数据库集成，以及富文本（如Word、PDF）的处理。Solr是高度可扩展的，并提供了分布式搜索和索引复制。Solr是最流行的企业级搜索引擎，Solr4 还增加了NoSQL支持。

Solr是用Java编写、运行在Servlet容器（如 Apache Tomcat 或Jetty）的一个独立的全文搜索服务器。 Solr采用了 Lucene Java 搜索库为核心的全文索引和搜索，并具有类似REST的HTTP/XML和JSON的API。Solr强大的外部配置功能使得无需进行Java编码，便可对 其进行调整以适应多种类型的应用程序。Solr有一个插件架构，以支持更多的高级定制。

因为2010年 Apache Lucene 和 Apache Solr 项目合并，两个项目是由同一个Apache软件基金会开发团队制作实现的。提到技术或产品时，Lucene/Solr或Solr/Lucene是一样的。

### Solr的优缺点

#### 优点

Solr有一个更大、更成熟的用户、开发和贡献者社区。

支持添加多种格式的索引，如：HTML、PDF、微软 Office 系列软件格式以及 JSON、XML、CSV 等纯文本格式。

Solr比较成熟、稳定。

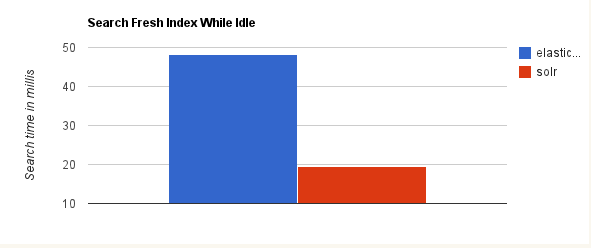
不考虑建索引的同时进行搜索，速度更快。

#### 缺点

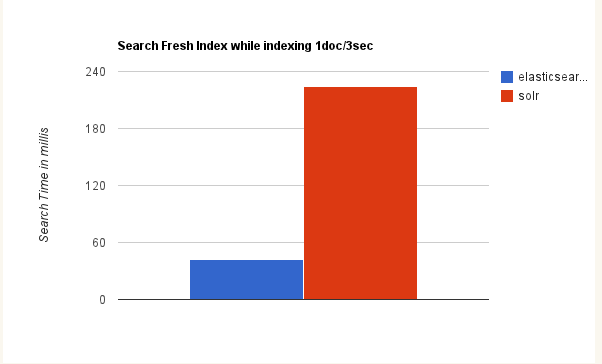
建立索引时，搜索效率下降，实时索引搜索效率不高。

## Es与solr的对比

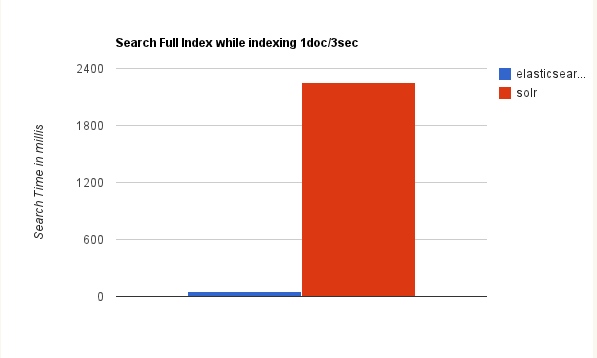
当单纯的对已有数据进行搜索时，Solr更快：



当实时建立索引时, Solr会产生io阻塞，查询性能较差, Elasticsearch具有明显的优势：



随着数据量的增加，Solr的搜索效率会变得更低，而Elasticsearch却没有明显的变化：



### 比较总结

1. 二者安装都很简单；
2. Solr 利用 Zookeeper 进行分布式管理，而 Elasticsearch 自身带有分布式协调管理功能;
3. Solr 支持更多格式的数据，而 Elasticsearch 仅支持json文件格式；
4. Solr 官方提供的功能更多，而 Elasticsearch 本身更注重于核心功能，高级功能多有第三方插件提供；
5. Solr 在传统的搜索应用中表现好于 Elasticsearch，但在处理实时搜索应用时效率明显低于 Elasticsearch。

Solr 是传统搜索应用的有力解决方案，但 Elasticsearch 更适用于新兴的实时搜索应用。

## 具体版本实例对比

**API**

|  |  |  |
| --- | --- | --- |
| 功能 | Solr 6.2.1 | Elastic Search 5.0 |
| 格式 | XML,CSV,JSON | JSON |
| HTTP REST API | 支持 | 支持 |
| 二进制API | 支持（SolrJ） | 支持[传输客户端，Thrift（通过插件）] |
| 是否支持JMX | 支持 | 不支持（ES的特定信息通过REST API传输） |
| 官方客户端库 | Java | Java, Groovy, PHP, Ruby, Perl, Python, .NET，javascript（官方库列表） |
| Community client libraries | PHP, Ruby, Perl, Scala, Python, .NET, Javascript, Go, Erlang, Clojure | Clojure, Cold Fusion, Erlang, Go, Groovy, Haskell, Java, JavaScript, .NET, OCaml, Perl, PHP, Python, R, Ruby, Scala, Smalltalk, Vert.x Complete list |
| 3rd-party product integration（open-source） | Drupal, Magento, Django, ColdFusion, Wordpress, OpenCMS, Plone, Typo3, ez Publish, Symfony2, Riak (via Yokozuna) | Drupal, Django, Symfony2, Wordpress, CouchBase |
| 3rd-party product integration (commercial) | DataStax Enterprise Search, Cloudera Search, Hortonworks Data Platform, MapR | SearchBlox, Hortonworks Data Platform, MapR etc  Complete list |
| Output | JSON, XML, PHP, Python, Ruby, CSV, Velocity, XSLT, native Java | JSON, XML/HTML (via plugin) |

**Infrastructure**

|  |  |  |
| --- | --- | --- |
| 功能 | Solr 6.2.1 | ElasticSearch 5.0 |
| Master-slave replication | 支持（**Only in non-SolrCloud.** In SolrCloud, behaves identically to ES.） | 不支持（Not an issue because shards are replicated across nodes） |
| Integrated snapshot and restore | 文件系统 | Filesystem, AWS Cloud Plugin for S3 repositories, HDFS Plugin for Hadoop environments, Azure Cloud Plugin for Azure storage repositories |

**Indexing**

|  |  |  |
| --- | --- | --- |
| 功能 | Solr 6.2.1 | ElasticSearch 5.0 |
| Data Import | DataImportHandler - JDBC, CSV, XML, Tika, URL, Flat File | [DEPRECATED in 2.x] Rivers modules - ActiveMQ, Amazon SQS, CouchDB, Dropbox, DynamoDB, FileSystem, Git, GitHub, Hazelcast, JDBC, JMS, Kafka, LDAP, MongoDB, neo4j, OAI, RabbitMQ, Redis, RSS, Sofa, Solr, St9, Subversion, Twitter, Wikipedia |
| ID field for updates and deduplication | 支持 | 支持 |
| DocValues | 支持 | 支持 |
| Partial Doc Updates | 支持（with stored fields） | 支持（with \_source field） |
| Custom Analyzers and Tokenizers | 支持 | 支持 |
| Per-field analyzer chain | 支持 | 支持 |
| Per-doc/query analyzer chain | 不支持 | 支持 |
| Index-time synonyms | 支持 | 支持（Supports Solr and Wordnet synonym format） |
| Query-time synonyms | 支持（especially via hon-lucence-synonyms） | 不支持（Technically, yes, but practically no because multi-word/phrase query-time synonyms are not supported. See ES docs and hon-lucene-synonyms blog for nuances.） |
| Multiple | 支持 | 支持 |
| Near-Realtime Search/Indexing | 支持 | 支持 |
| Complex documents | 支持 | 支持 |
| Schemaless | 支持（4.4+） | 支持 |
| Multiple document types per schema | 不支持（One set of fields per schema,one schema per core） | 支持 |
| Online schema changes | 支持Schemaless mode or via dynamic fields | 支持Only backward-compatible changes |
| Apache Tika integration | 支持 | 支持 |
| Dynamic fields | 支持 | 支持 |
| Field copying | 支持 | Via multi-fields |
| Hash-based deduplication | 支持 | Murmur plugin or ER plugin |

**Searching**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |