Introduction to Solr

Apache^{TM} Solr is a search server built on top of Apache Lucene^{TM}, an open source, Java-based, information retrieval library. Solr is designed to drive powerful document retrieval or analytical applications involving unstructured data, semi-structured data or a mix of unstructured and structured data. It also has secondary support for limited relational, graph, statistical, data analysis or storage related use cases. Since Solr is Apache 2.0 licensed open source software designed for extensibility, it gives you the freedom to adapt or optimize it for almost any commercial or non-commercial use case.

Solr's query syntax and parsers offer support for everything from the simplest keyword searching through to complex queries on multiple fields and faceted search results. Collapsing and clustering results offer compelling features for e-commerce and storefronts. Streaming expressions allow you to conduct analytics on an entire corpus, a subset matching a query, or a random sample from a set of documents. Powerful math expressions build on streaming expressions to provide the backbone for advanced analysis and predictive analytics use cases.

Advanced relevancy tuning is also supported; Solr provides access to almost all of Lucene's text analysis features including tokenization, stemming, synonyms and much more, allowing you to tune relevancy based on knowledge of your users and your domain. Solr even allows for customization of relevancy via machine learning using the Learning To Rank feature.

Queries are transmitted to Solr via HTTP 1.1 or 2.0 requests and the response is typically a list of structured document descriptors. In the classic example, 10 descriptors are returned, each including a URL to locate the document (often rendered as "10 blue links"). However, Solr can go far beyond document locators and many other types of document metadata might also be included. Flexible schema configurations allow nearly any type of metadata to be associated with a document indexed in Solr. The schema elements page of the indexing guide has more details on these options.

JSON is the default response format, but it could also be XML, CSV, optimized binary, or (with customization) any format you desire. This means that a wide variety of clients will be able to use Solr. Such clients might be web applications, browsers, rich client applications, or mobile devices. Any platform capable of HTTP can talk to Solr. Several Client APIs are provided for use in common programming languages.

In addition to providing a network accessible engine for Lucene based document retrieval, Solr provides the ability to scale beyond the limitations of a single machine. Indexes can be sharded and replicated for performance and reliability, using either one of two Solr Cluster Types. One type of cluster requires no supporting infrastructure, and instances are managed directly by administrators. The second type uses Apache Zookeeper^{TM} to coordinate management activities across the cluster.