SolrTerminology

This is the place to define terms that are specific to Solr or that have meanings in the Solr community that differ from their meanings in the general populace.

Also see the official Solr Glossary in the Reference Guide

When adding to this page, please list terms in alphabetical order.

SolrCloud

One of the most confusing aspects of Solr terminology is in the difference between collections, shards, replicas, cores, and config sets. These terms have specific meanings with reference to SolrCloud.

- ***Collection***: A complete logical index in a SolrCloud cluster. It is associated with a config set and is made up of one or more shards. If the number of shards is more than one, it is a distributed index, but SolrCloud lets you refer to it by the collection name and not worry about the shards parameter that is normally required for DistributedSearch.
- ***Config Set***: A set of config files necessary for a core to function properly. Each config set has a name. At minimum this will consist of solrconfig.xml (SolrConfigXml) and schema.xml (SchemaXml), but depending on the contents of those two files, may include other files. This is stored in Zookeeper. Config sets can be uploaded or updated using the upconfig command in the command-line utility or the bootsrap_confdir Solr startup parameter.
- ***Core***: This is discussed in the General list (below) as Solr Core. One difference with SolrCloud is that the config it uses is in Zookeeper. With traditional Solr, the core's config will be in the conf directory on the disk.
- ***Leader***: The shard replica that has won the leader election. Elections can happen at any time, but normally they are only triggered by events like a Solr instance going down. When documents are indexed, SolrCloud will forward them to the leader of the shard, and the leader will distribute them to all the shard replicas.
- ***Replica***: One copy of a shard. Each replica exists within Solr as a core. A collection named "test" created with numShards=1 and replicationFactor set to two will have exactly two replicas, so there will be two cores, each on a different machine (or Solr instance). One will be named test_shard1_replica1 and the other will be named test_shard1_replica2. One of them will be elected to be the leader.
- ***Shard***: A logical piece (or slice) of a collection. Each shard is made up of one or more replicas. An election is held to determine which replica is the leader. This term is also in the General list below, but there it refers to Solr cores. The SolrCloud concept of a shard is a logical division.
- ***Zookeeper***: This is a program that helps other programs keep a functional cluster running. SolrCloud requires Zookeeper. It handles leader elections. Although Solr can be run with an embedded Zookeeper, it is recommended that it be standalone, installed separately from Solr. It is also recommended that it be a redundant ensemble, requiring at least three hosts. Zookeeper can run on the same hardware as Solr, and many users do run it on the same hardware.

General

- ***Auto-warming***: What Solr does when it opens a new cache, and seeds it with key/val pairs based on the "top" keys from the old instance of the cache
- ***Constraint***: A viable method of limiting a set of objects (*)
- ***Core***: See Solr Core below.
- ***DisMax***: Typically a reference to the DisMaxQParserPlugin but in older contexts may be referring to the DisMaxRequestHandler.
- ***Facet***: A distinct feature or aspect of a set of objects; "a way in which a resource can be classified" (*)
- ***Field Collapsing***: A specific use case of Result Grouping where the groups are dictated by the value of a field.
- ***Filter***: depending on context, may be:
  1. Another word for "Constraint"
  2. The "fq" param which constrains the results from a query without influencing the scores.
  3. Specifically referring to the Lucene "Filter" class
- ***NRT***: Near Real Time: This refers to the general concept of wanting document updates to be "immediately" visible to search clients.
- ***Request Handler***: A Solr component that processes requests. For example, the DisMaxRequestHandler processes search queries by calling the DisMax Query Parser. Request Handlers can perform other functions, as well.
- ***QTime***: The elapsed time (in milliseconds) between the arrival of the request (when the SolrQueryRequest object is created) and the completion of the request handler. It does not include time spent in the response writer formatting/streaming the response to the client.
- ***Query Parser***: A Solr component that parses the parameters and search terms submitted in a search query.
- ***Searcher***: In Solr parlance, the term "Searcher" tends to refer to an instance of the SolrIndexSearcher class. This class is responsible for executing all searches done against the index, and manages several caches. There is typically one Searcher per SolrCore at any given time, and that searcher is used to execute all queries against that SolrCore, but there may be additional Searchers open at a time during cache warming (in which and "old Searcher" is still serving live requests while a "new Searcher" is being warmed up).
- ***Shard***: A distributed index is partitioned into "shards". Each shard corresponds to a Solr core and contains a disjoint subset of the documents in the index.
- ***Slop***: As in "phrase slop": the number of positions two tokens need to be moved in order to match a phrase in a query.
- ***Solr Core***: Also referred to as just a "***Core***". This is a running instance of a Lucene index along with all the Solr configuration (SolrConfigXml, SchemaXml, etc...) required to use it. A single Solr application can contain 0 or more cores which are run largely in isolation but can communicate with each other if necessary via the CoreContainer. From a historical perspective: Solr initially only supported one index, and the SolrCore class was a singleton for coordinating the low-level functionality at the "core" of Solr. When support was added for creating and managing multiple Cores on the fly, the class was refactored to no longer be a Singleton, but the name stuck.
- ***Solr Home Dir***: Also referred to as the "***Solr Home Directory***" or just "***Solr Home***" this is the main directory where Solr will look for configuration files, data, and plugins. Knowing which directory to use as the Solr Home is the one piece of information that Solr must either assume (the default

is "./solr") or be configured using some mechanism beyond Solr's normal configuration files. An example Solr Home is included in Solr releases and contains a README.txt explaining the directory structure. For more information on ways to override the default Solr Home, please read SolrIn stall.

- ***Static warming***: What users can do using newSearcher and firstSearcher event listeners to force explicit warming actions to be taken when one of these events happens – frequently it involves seeding one or more caches with values from "static" queries hard-coded in the solrconfig.xml