

Deployment of Solr Cores with Zookeeper

Deployment of Solr Cores with Zookeeper



This page is a design for [SOLR-1724](#) which has not yet been released.

If you are interested in the officially supported approach to using Solr with [ZooKeeper](#) (aka [SolrCloud](#)), please see the Solr Reference Guide section, [SolrCloud](#).

Architecture

Zookeeper may be used as a distributed filesystem to write which Solr servers should be running which cores. Ephemeral nodes are intentionally not used, except for the core status updates. Zookeeper is used as a transactionally redundant filesystem, not a system for maintaining connections to various servers. This is best left to dedicated monitoring services. GSON is the JSON library used to serialize and deserialize objects to and from the JSON format.

Code

The Jira issue for this project is located at: <https://issues.apache.org/jira/browse/SOLR-1724>

The Solr Cloud branch is used in conjunction with this issue. The Solr Cloud SVN location is: <http://svn.apache.org/repos/asf/lucene/solr/branches/cloud>

Supported File Types

Zipped cores are the standard because they are easier to manage, download, and transfer across the network.

- Zipped core accessible via HDFS
- Zipped core accessible via HTTP

Zookeeper Filesystem

Cores

Each "cores" file is written to Zookeeper and is of the form cores_N, where N is the version number. This is purposefully similar to the segment infos files written by Lucene. The cores file is stored in the JSON format.

Contents of the cores file:

Name	Type	Description
name	string	The name of the core which is the same as the ZK filename
version	long	version of the core which should match with the version in the filename
array	coresinfo	An array of core infos

Each cores info contains:

Name	Type	Description
host	string	The host
name	string	The core name
instanceDir	string	Optional instanceDir
configFile	string	Optional solrconfig.xml path
schemaFile	string	Optional schema.xml path
dataDir	string	Optional data directory path
url	string	Optional URL from which to download and install the core from. It will be unzipped to the instance dir

Host

Each Solr server (aka host or [CoreContainer](#)) must report to Zookeeper which cores it has installed. Each host file is of the form host_version. It is the responsibility of each Solr host/server to match the state of the cores_N file. Meaning, each Solr server must install the cores defined for it and after successful install, write the hosts file out to Zookeeper.

Contents of a host file:

Name	Type	Description
name	string	Same as the filename
version	long	Version encoded into the filename
array	hostinfo	Array of host infos

Each host info contains:

Name	Type	Description
name	string	Core name
instanceDir	string	The instance directory
configFile	string	The solrconfig.xml path
schemaFile	string	The schema.xml path
dataDir	string	The data directory path
size	long	The size in bytes of the core
lastModified	long	The last modified of the core

Sample Directory Layout

There are 2 cores files in this sample directory layout. Under /production/hosts several host files have been written. Actually, all of the necessary hosts files have been written indicating that for example cores_1 and cores_2 operational definitions have completed on each Solr server.

The /production/hosts/status directory contains ephemeral node files for each core operation currently in progress. The status directory may be read by the cores Zookeeper manager user interface to inform the administrative user of the state of the operations.

/production/cores_1

/production/cores_2

/production/hosts/servera_1

/production/hosts/serverb_1

/production/hosts/serverc_1

/production/hosts/serverd_1

/production/hosts/servera_2

/production/hosts/serverb_2

/production/hosts/serverc_2

/production/hosts/serverd_2

/production/hosts/status/serverc_2_septblogs

/production/hosts/status/servera_2_julyblogs

/production/hosts/status/serverd_2_augustblogs

CoreController

Core deploy client that lives inside a [CoreContainer](#). It listens for events on a given path, finds it's hostname in the latest cores file by version. Each cores file is like Lucene's segment infos file which describes the set of segments that make up the current index. The cores file defines the set of cores that should be installed on a given Solr host.

A default root path must be defined, for the unit tests /production is used.

Libraries

- GSON - <http://code.google.com/p/google-gson/>
- Hadoop hadoop-0.20.2-dev-test.jar and 0.20.2-dev-test.jar
- Commons lang - commons-lang-2.4.jar

TODO

- Allow other Solr core management operations such as swap, rename

- Completing the status updates
- A web based administrative console that includes the ability to upload the next desirable state of the Solr cluster, view the status of the currently executing core operations, and more