Alfresco 5.0.x and Solr4

Basic best practices for configuring Solr4 with Alfresco 5.0.x

Last update: July 9th 2015

Table of Contents

Introduction

Alfresco One 5.0 uses Solr4 as its search engine to provide advanced features including Filtered Search, Suggester and more.

Solr4's resource requirements can exceed those of Solr1 (included in Alfresco 4.x). This document aims to present some detail around a few configuration options that can improve performance and resource usage for Solr4 in a typical environment.

All the findings presented in this document have arisen through the usage of Alfresco One as Alfresco's own document repository and our experience of running Alfresco Cloud.

Index Volume Administration

Size

Solr4's indexes can be significantly larger than those for Solr1 previously.

For a repository with 10 million documents of varying size and complexity, an index size of around 200GB is not abnormal. Later we will see some configuration changes which can dramatically reduce that size, but a volume that houses large indexes needs careful administration.

Recommendations:

1. For the volume that holds the indexes (alf_data/solr4/index) you should maintain twice as much free space as the current total size of the indexes

During index building and merging operations, and during index backups, the system can temporarily take up more than double the current index size. The extra space is an absolute requirement to make sure that intermittent problems do not occur during merges and backup.

2. Set the backup directory to a different volume

Alfresco backs up the Solr indexes every morning by default. These are saved by default to alf_data/solr4Backup . To reduce the risk of the backups taking up space required by the indexes themselves, it makes sense to set the backups to a different volume. (cf. http://docs.alfresco.com/5.0/tasks/solr-backup.html)

3. Compress the index backups

A simple cron job to compress the files created by the backup will save space on the backup volume.

Speed

As with Solr1, a fast disk subsystem is essential if search and indexing performance is to be maintained as the number of documents in the repository grows.

Recommendations:

1. Your index files (e.g. /alf_data/solr4/index) should be on the fastest disk subsystem that you have available.

The index volumes can be under heavy read/write load, and slower disk access times will adversely affect search performance, in the same way that slower disk write times will adversely affect indexing times.

Striped SSDs are recommended.

2. The index content files can be stored on less expensive, slower disks

When indexing a document, Solr4 asks the repository for a copy of the text extracted from the document. Transformations are particularly expensive operations for the repository, so to reduce the load, Solr4 caches a copy of the extracted text, compressed, in a separate directory (solr4/content).

It is less critical for this directory to be on the fastest possible disk. The content is written and read less often that the index files, and can tolerate slower, cheaper disk subsystems

System processor requirements

Search and Indexing can be processor–intensive operations. Whilst most systems will differ in their exact requirements and specifications, there are a couple of basic rules-of-thumb that can help the running of a performant system.

Core Pool Size

The core pool size dictates how many threads are used for performing the indexing of nodes.

Recommendation:

1. Set the Core Pool Size to half the number of CPUs available

In the solrcore.properties file in solr4/workspace-SpacesStore/conf set the value of the property *alfresco.corePoolSize* to half the number of CPUs available to the system that Solr is running on.

Suggester CPU sizing

If you decide to use the Search Suggestions functionality (see below for further discussion of Suggester), make sure that your system is adequately sized to cope with it.

Recommendation:

1. If you decide that your use-case requires Suggester, make sure that your Solr system has at least 4 CPUs

The additional workload of building and serving the suggestions is suited for a multi-core architecture. Having at least 4 CPUs/cores available to Solr will help mitigate performance issues in other areas of search and indexing

Solr4 Configuration Optimisation

Out-of-the-box, the Solr4 configuration is standardised and is suitable for a test or demonstration system. This may not represent the best configuration for larger repositories, or systems with specific use-cases.

Some easy configuration changes are discussed next. Some of these affect additional functionality that has been brought in to version 5.0.x and so come with adverse side-effects that need to be understood. Others are changes that were discovered post-release, and may be set differently by default in future versions.

Suggester configuration

'Search Suggestions' is new functionality in Alfresco One 5.0. When a user starts to type a query into the standard search box in Share, they receive real-time suggestions for the search based on what they have typed so far.

The Suggester functionality can significantly increase the load on the server, in terms of both CPU required (at search time) and disk space required (to store the extra information needed).

Recommendations:

1. Disable Suggester for the Archive store

Unless you have a specific requirement to be able to use the Suggester functionality for the Archive store (where deleted documents are kept before removal) it makes sense to disable the functionality.

To do this, edit the following section in /solr4/archive-SpacesStore/conf/schema.xml

```
<!-- Suggestion -->
    <field name="suggest"              type="text_shingle"    indexed="true"
omitNorms="true"  stored="false"  multiValued="true" />
```

and change *indexed="true"* to *indexed="false"*

2. Determine whether or not your use-case requires the new Suggester functionality for the standard search. If it is not required, disabling it may help the performance of the system and reduce the size of the indexes

To disable it, make the same change as above, but in the file /solr4/workspace-SpacesStore/conf/schema.xml

MaxShingleSize and OmitPositions

Two configuration changes that have shown to help in the reduction of index sizes on Solr4 relate to the MaxShingleSize and OmitPositions parameters. Both of these parameters may be used in the future, or may be used in highly customised environments.

For the moment, it makes sense to change these parameters from their out-of-the-box defaults if there is no customised requirement for them.

Recommendations:

1. Set OmitPositions=true

To do this, edit /solr4/workspace-SpacesStore/conf/schema.xml and edit the section:

<!-- Suggestion -->

<field name="suggest" type="text_shingle" indexed="true" omitNorms="true" stored="false" multiValued="true" />

Add the parameter *OmitPositions="true"*

2. Set MaxShingleSize to 2

To do this, find *maxShingleSize="3"* in the schema.xml and change the value from 3 to 2

Socket timeout configuration

If the repository holds very large documents which require indexing, (e.g. PDFs of hundreds of MB each), there can be issues of timing, whereby Solr can timeout a connection to the repository.

To remedy this, simply make sure that the socketTimeout specified for Solr is higher than the repository socket timeout.

Recommendation:

1. In archive-SpacesStore/conf/solrcore.properties and workspace-SpacesStore/conf/solrcore.properties check the property *alfresco.socketTimeout*

   In the relevant configuration file for your App Container where Alfresco is running (e.g. server.xml if you are using Tomcat) check the socket timeout for incoming http/https connections. I.e. for Tomcat, check the value for connectionTimeout in the relevant "Connector" section.

The first value should be greater than the second. Typical example values are 360000 and 20000 respectively.

Example figures

The above recommendations were arrived at following internal production usage of Alfresco One and Alfresco Cloud for over a year.

Here are some metrics for one of the systems in use, to give you an idea of scale. Bear in mind that these are the figures from only one system, and every system will have a different usage pattern and metric profile. Hopefully these figures will give some indication of an installation's predicted size, and provide at least a ballpark figure for sizing.

- Number of documents in the repository: approx. 10,000,000
- /solr4/content directory size : 110GB
- Memory available to Solr (Tomcat) : 25GB
- Memory total in system : 30GB
- Total index size
    - Before the recommendations in this document : 202GB
    - After the recommendations : 96GB
- Time to complete a full re-index : 4 days