



technology services group

Leveraging HPI and Lucene for Full Text Documentum Search

OpenContent

HPI

High Performance Interface

Author:

George Steimer
georges@tsgrp.com

Last Updated:

September 1, 2009

Table of Contents

1	OVERVIEW.....	3
2	HIGH LEVEL INTEGRATION.....	3
3	LUCENE INDEXING.....	5
4	HPI SEARCH AND DOCUMENT SECURITY.....	5
5	POSSIBLE UPDATE SCENARIOS.....	7
5.1	DOCUMENT PERMISSIONS CHANGE IN DOCUMENTUM	7
5.2	USER IS ADDED OR REMOVED FROM A GROUP OR ACL.....	8
6	SUMMARY.....	8

1 Overview

Many of our clients have expressed interest in using an alternative to Documentum's built in full-text search engine on the 5.3 and D6.x platforms, FAST. With Documentum moving away from the FAST platform in the upcoming Enterprise Search Server (ESS) layer, many of our clients are looking for ways to leverage Lucene now, rather than waiting for ESS.

According to the Apache Lucene website (lucene.apache.org), Lucene is "a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform." Lucene is distributed by Apache as open source, available for free download.

The High Performance Interface (HPI), which can be obtained as free open source from www.tsgrp.com, is a streamlined web interface for TSG's OpenContent web services layer. Among other things, HPI can be used to search a content management system, including Documentum. HPI's configurable nature and user-friendly interface make it a best-of-breed solution for searching a CMS repository or file system cache.

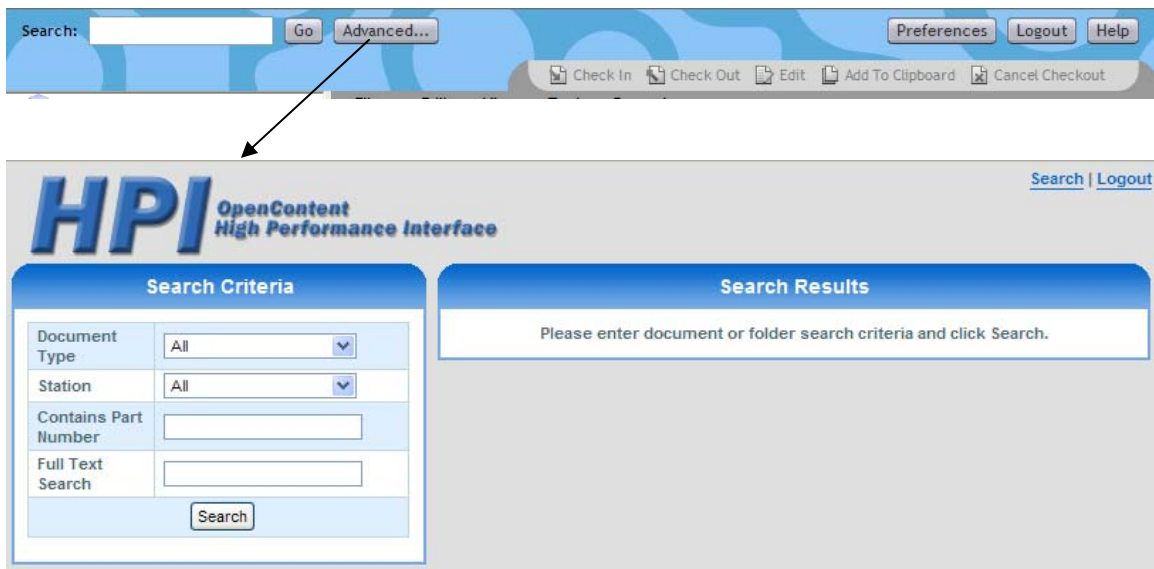
This whitepaper will explore how HPI and Lucene can be used to replace FAST-based full text searching in Documentum 5.3 and above.

2 High Level Integration

For some applications, Webtop is the main interface that users rely on to search for documents. However, HPI's search module has proven itself to be much more user friendly and configurable. The HPI search module can be launched directly from Webtop. Although users could access HPI directly, using this approach may be useful for users that are accustomed to using Webtop.

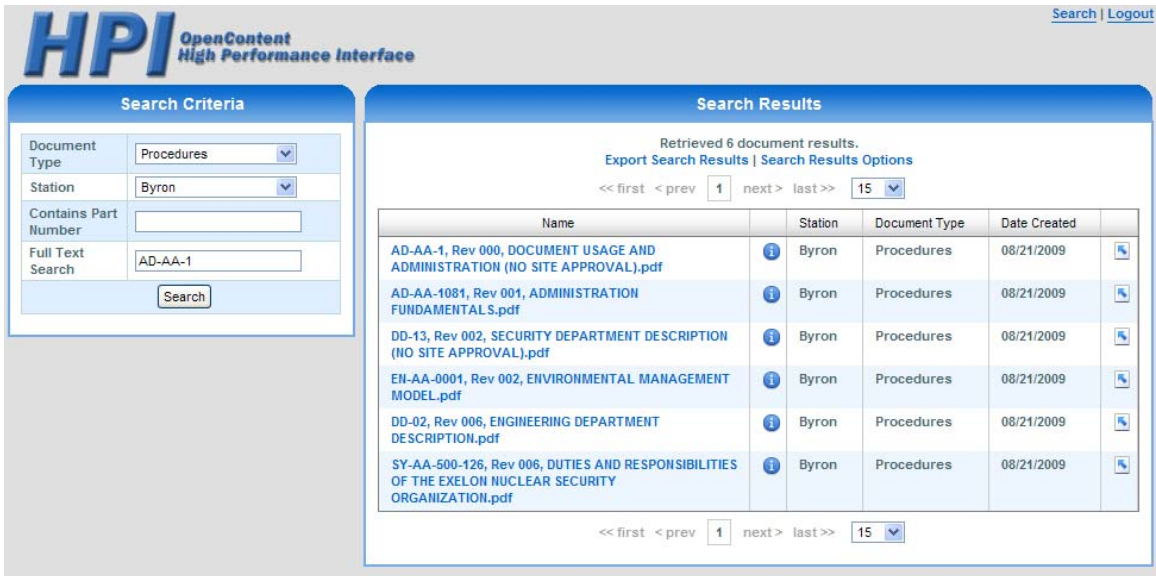
Launching HPI from Webtop

HPI's search interface can be integrated with Webtop when users click on the Advanced Search button, or through a menu option in the Webtop toolbar:



After loading HPI, the user can search Documentum by using HPI's intuitive search form on the left hand side. From a user perspective, searching based on Document metadata or the full text

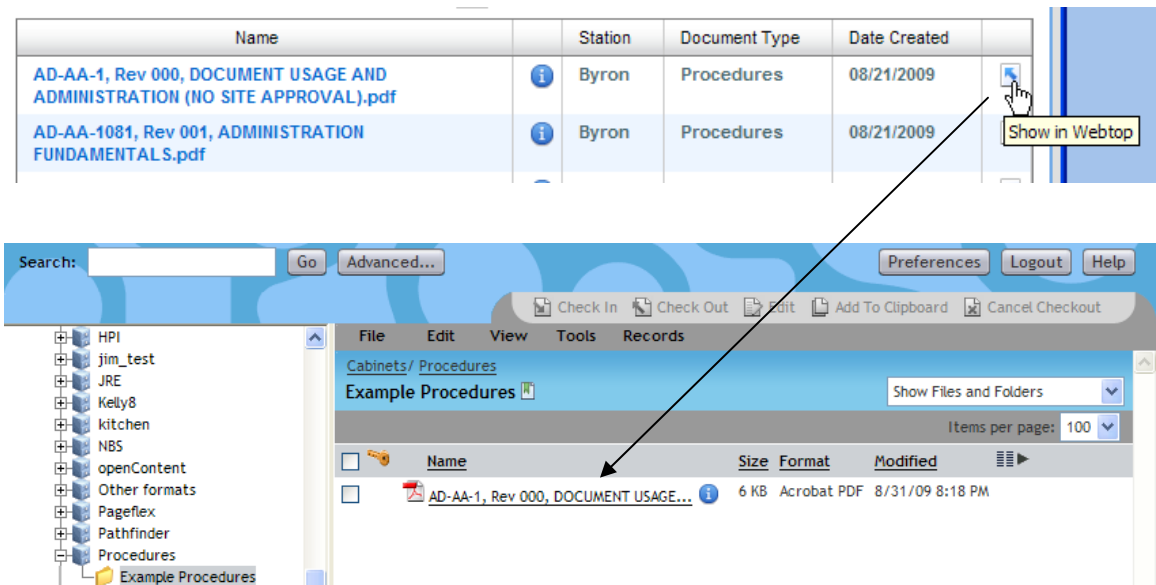
index is seamless. In the above example, if the user chooses to search on Document Type or Station, HPI will search on document metadata. If the user chooses to search on the “Contains Part Number” or “Full Text Search” boxes, a full text query is run against Lucene:



Users can click the document name link to view the document, or the “i” icon to view document properties. Users can page and sort search results by any column, and can show or hide columns as needed. Additionally, an export to Excel feature is included to allow for a local download of the search results.

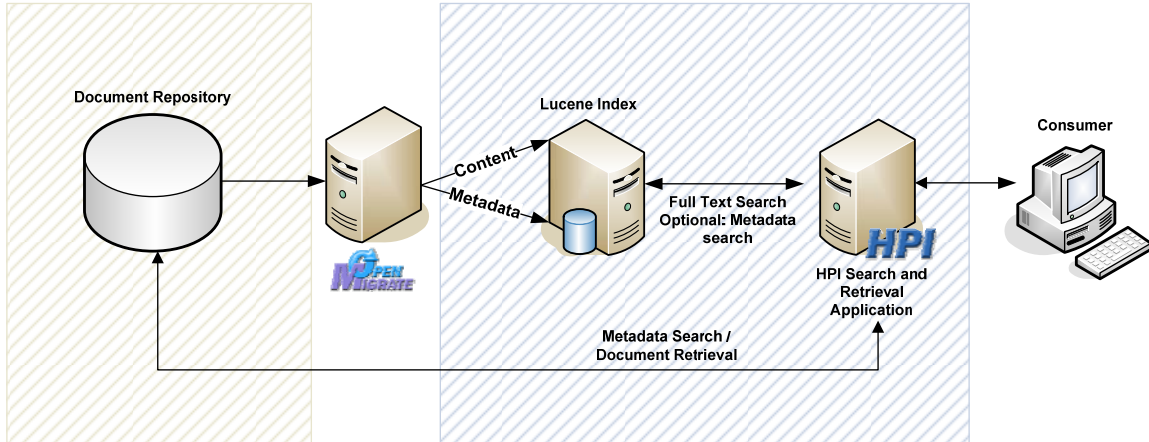
Returning to Webtop from HPI

Users may want to work on a document in Webtop after searching in HPI. For example, the user may want to edit the document, update properties, etc. To accommodate, HPI can provide a link for each document to send the user back to Webtop to work with the document:



3 Lucene Indexing

Documents will be placed in Lucene by using TSG's OpenMigrate. OpenMigrate will be run on a schedule (typically every 3 minutes) to push, update or delete documents and metadata in Lucene:



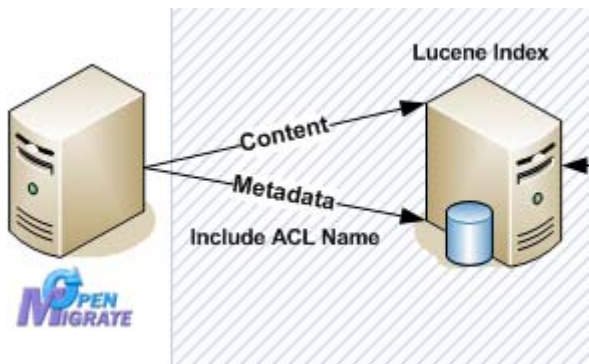
In the diagram above, documents and metadata are synchronized in the Lucene index by OpenMigrate. As in TSG's Consumer Interface Suite (CIS) whitepaper, OpenMigrate can be used to ensure that documents are stored in the index appropriately over time. After the first index is created, OpenMigrate will add, remove and update documents in the Lucene index based on changes in Documentum.

Once the index is created, HPI can be used to search the Lucene full text index, and optionally, the metadata stored in Lucene. Alternatively, HPI can query Documentum for metadata searches. In either case, when a user requests to view the document, HPI will pull the document contents from Documentum.

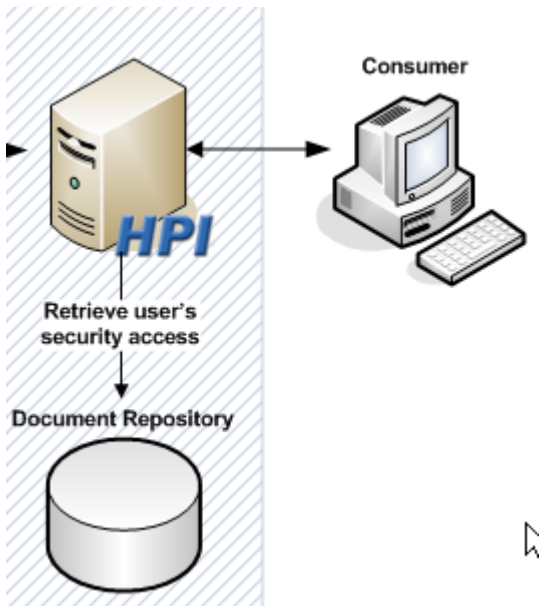
4 HPI Search and Document Security

As outlined in the previous section, HPI will be running full text searches through Lucene rather than Documentum. This brings up the question of document security. Since all documents can potentially be loaded into Lucene, we need to insure that full text search can integrate with Documentum security settings without adversely affecting search performance.

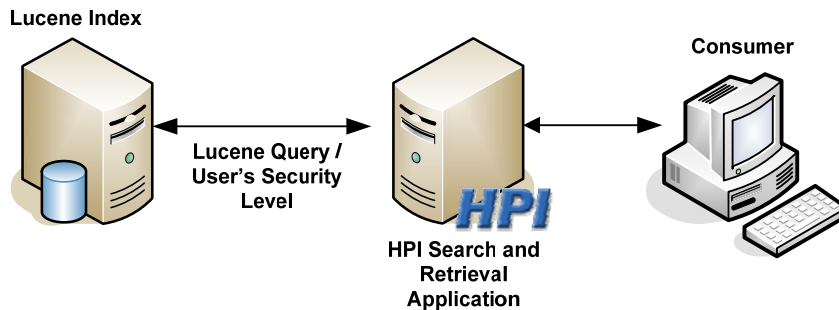
The first step to integrate with Documentum security is to pull the document ACL name into the Lucene index during the load of document metadata:



Once the ACL name is stored as metadata in the Lucene index, we can integrate the user's security access information along with the search data. First, when HPI loads, we can retrieve the user's security access from Documentum:



Note that this query will only be performed the first time the user enters HPI. For the remainder of the HPI browser session, the user's security level can be combined with the results from the Lucene index to filter out documents for which the user does not at least have READ permissions.



There are a few different ways that the security filter could be applied. In one scenario, we could simply append the user's ACL information at the end of each query. For example, appending the Lucene query equivalent of: `and document_acl in ('acl_one', 'acl_two')`. Another option would be to filter through the results set after Lucene returns and remove the results based on a list of the user's ACLs stored in memory. Note that this second option does not include a round trip to Documentum for each result. Further research is needed to determine which method is a better fit for each client. Both scenarios will work, but if the number of search results returned in a typical query is high, while the number of ACLs a user has READ access on is low, filtering through the Lucene query will perform better. If the typical search returns a relatively low number of results, and the number of ACLs that a typical user has at least READ access on is high, returning all results and then filtering the list in memory will perform better.

5 Possible Update Scenarios

The following sections overview document and user security update scenarios that could occur in a production system.

5.1 Document Permissions Change in Documentum

In a production application, a document could be updated in Documentum by changing the security ACL. The following scenarios could occur:

Document Given Less Permissions

1. OpenMigrate runs at time T, updating the Lucene index.
2. At time T+1, document A has its security changed to an ACL that restricts access on the document. Let's say that the document is now only visible to dmadm and all other users have no access.
3. At time T+2, a user executes a full text search that returns document A in the result set. The user will be able to see document A in the result set, but clicking on the document to view contents will not allow user to read the document. This is because the contents are streamed directly from Documentum, where the ACL is checked in real time.
4. At time T+3, OpenMigrate runs, updating the Lucene index.
5. At time T+4, a user executes a full text search that contains text in document A. Document A does not return in the result set.

In the above scenario, the maximum amount of time that the document is returned from Lucene in the search results is the same as the schedule that the OpenMigrate task is on. In most cases, OpenMigrate runs every 3 minutes.

Document Given More Permissions

1. At time T, document B has its security set to an ACL that restricts access on the document to all users but dmadm.
2. OpenMigrate runs a time T+1, updating the Lucene index.
3. At time T+2, document B has its security changed to an ACL that gives READ access to all users.
4. At time T+3, a user executes a full text search that contains text in document B. The document will not be included in the user's search results.
5. At time T+4, OpenMigrate runs, updating the Lucene index.
6. At time T+5, a user executes a full text search that contains text in document B. Document B now returns correctly in the result set.

As with the above scenario, the max time that the user will not see document B in search results will be the OpenMigrate schedule interval – typically 3 minutes.

Closing the Gaps

To close the gaps in the above scenarios, a TBO could be added in Documentum to kick off the OpenMigrate job as soon as a document is saved in Documentum. This approach should be analyzed in more detail in the client environment to determine if making this approach would be detrimentally taxing on Documentum, OpenMigrate and/or Lucene.

5.2 User is Added or Removed from a Group or ACL

Search results could also be affected if a user is added or removed from an ACL or a group while the user is in the middle of an HPI session. Since ACLs typically have groups set on them rather than individual users, the following scenario runs through a group change example:

1. At time T, a user enters HPI to execute full text searches. At the time the user enters HPI, the system gathers the user's security access permissions.
2. At time T+1 the user is removed from Group A and added to Group B.
3. At time T+2, the user searches for documents. In the result set, Document X is returned, which is only readable by users in Group A. The logged in user will be able to see Document X in the result set, but as in the previous section, if the user tries to view the document, the document contents will not be displayed to the user.
4. At time T+3, the user enters a full text search that contains text in Document Y, which is only readable by users in Group B. The user will not see document Y in the search results.
5. At time T+4, the user exits HPI
6. At time T+5, the user returns to HPI. At this point, all search results will behave properly.

In the above scenario, since the design outlined above retrieves the users security permissions upon system access, if the user's group assignments are changed, those changes will not be reflected until the user exists the system and re-enters.

Closing the Gap

Closing this gap is more difficult than with document permissions changing. Instead of checking the user's permissions only once, the system could run this check each time a query is run. While this solution will certainly close the gap, the performance hit would be significant. Alternatively, an integration could be built in to notify HPI whenever a group change is made in Documentum, but the cost of this integration may negate the benefits. An alternative no-cost solution would be to make group changes when users are not in the system.

6 Summary

For clients that are looking for an alternative to FAST or Verity based full text search engines integrated into Documentum, Lucene combined with OpenMigrate and HPI provides a compelling alternative with no software costs. Additionally, EMC has announced that their upcoming ESS product will use Lucene as a search engine rather than FAST. The design outlined in this whitepaper allows clients to move away from FAST without waiting to upgrade to the ESS platform.