VIRGINIA POLYTECHNIC INSTITUTE
AND STATE UNIVERSITY

CS 5604 Information Storage and Retrieval

Front-End / User Interface Team

Moeti Masiane & Lawrence Warren

(moeti@vt.edu, lwarren@vt.edu)

Blacksburg, VA 24061

5/4/2016

Instructor

Prof. Edward Fox

Sponsored by

NSF grant IIS 1319578, III: Small: Integrated Digital Event Archiving and Library (IDEAL)

Abstract

With the advent of the information age, there have been numerous advantages for mankind. These advantages have been as a result of the ease of access to information. As the amount of information has grown, the concept of big data has been conceived. Zhang [1] defines big data as data that has high velocity, variety and volume. These attributes of big data are also known as the "3 Vs". Challenges have arisen concerning the storage and retrieval of this information. In this paper we focus on the efficient retrieval and presentation of stored big data. We present *Search*, a system that allows users to search tweet and webpage collections using faceted search. In addition to providing search capabilities, *Search* provides suggestions based on past user searches and similarities between search results and other documents that the system thinks could benefit the user. This capability requires that user activities be logged. In order to do this, user access initially is restricted to registered users who have valid user logins, and each user is assigned to a user management group. User activities are logged and used to provide suggested content. The search interface is implemented using Cloudera's open-source Hue dashboard and the existing search back-end has been implemented using Cloudera's unified platform for big data.

This paper is part of a wider research project whose focus is developing an information retrieval and analysis system in support of the IDEAL (Integrated Digital Event Archiving and Library) project. The search engine will retrieve results relating to tweet and web page data that has been collected by Dr E. Fox and his team of researchers from Virginia Tech. The overall project has been broken into smaller projects and these smaller projects have been assigned to different project teams. This paper's sole focus is the research and development relating to the creation of the front end of the search engine. The front end is responsible for accepting search queries, logging user activities, displaying search results and presenting suggested content based on provided user queries and past user activity.

Currently we have created a Hue interface that retrieves and displays search results based on demo data, both on our virtual machines and on the production cluster. We are waiting for tweet and web page information to be indexed on the cluster so that we can implement a search interface that runs on actual data processed by the rest of the project teams. Future work, which we intend to complete includes storing past user data in a MySQL database, providing the ability for the interface to accommodate both users that have logged into the system and those who have not, and collecting data concerning the links that a user has clicked on in our interface. We have created the database schema and tables for storing user search history on our virtual machines, but are still working on implementing the actual uploading of the user data. Providing access to both logged in users and those who have not logged in is challenging

because Hue does not allow this functionality. However, our proposed solution involves running two concurrent instances of our search interface: one for users with logins and the other for the rest with all users being directed to the page that does not require user registrations and providing a link to the other instance for users with login credentials. We are currently researching solutions to the challenge of keeping track of the links users have clicked on.

Table of Contents

A. Table of Figures

B. Table of Tables

1. Overview

IDEAL is an ongoing project; however, this is the first time there has been a team devoted to creating a front end interface so it goes without saying that there is no base work for this branch of the project. The first order of business was to learn some of the various components we would be dealing with to propose a plan of attack. After learning the SOLR interface and how it interacts with the cluster, we chose to use HUE as our basis because it is inherently made to work with a few of our needs. HUE is a set of web applications that enable you to interact with a CDH cluster and its applications let you browse HDFS and work with Hive and Cloudera Impala queries, MapReduce jobs, and Oozie workflows [2]. It is a lightweight Web server that lets you use Hadoop directly from your browser. HUE is just a 'view on top of any Hadoop distribution' and can be installed on any machine. The current way to play with Hue is often to go on demo.gethue.com or download a Virtual Machine. We did the latter and installed VirtualBox and Cloudera onto our personal machines to allow us to demo any alterations we needed to practice before we tried it on the actual cluster. Our main focus was to explore and implement ways to refine the faceted search capabilities and exploring options to enhance the user experience. A faceted search refers to a way to explore large amounts of data by displaying summaries about various partitions of the data and later allowing to narrow the navigation to a specific partition [5]. Its deployment impacts all other search patterns and the information architecture as a whole. The infrastructure for faceted navigation can enable a tighter relationship between searching and browsing. It can shape the structure and navigation of the entire site or application.

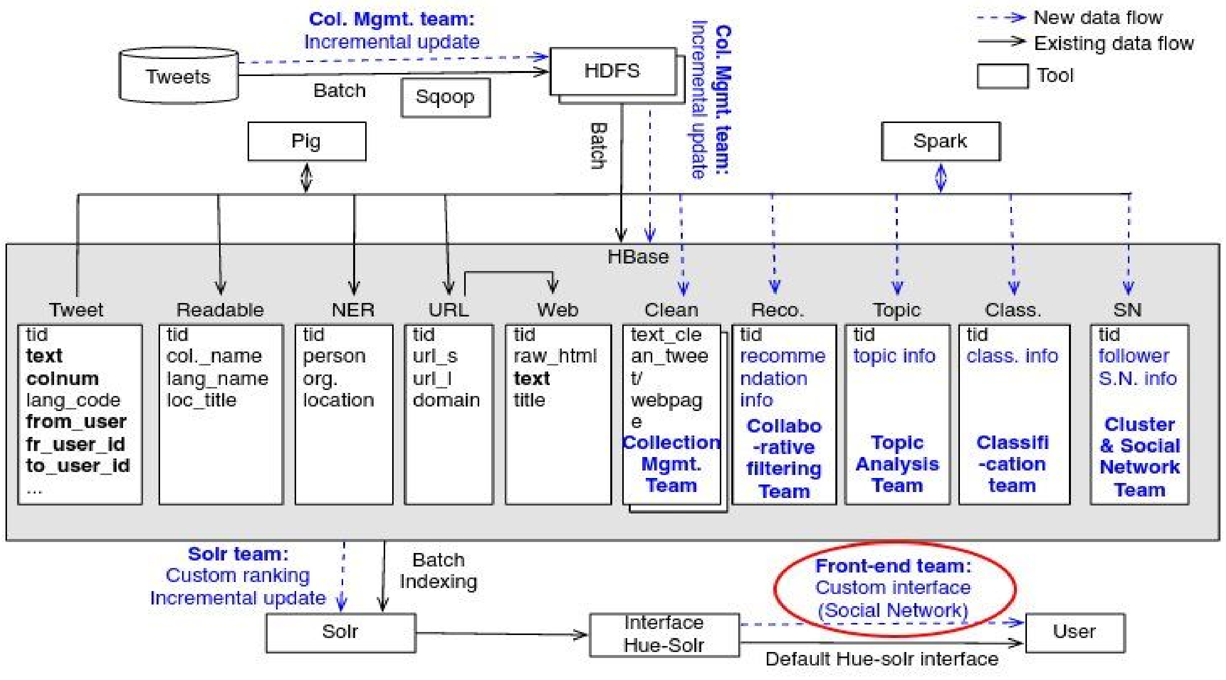


Figure 1. Work Flow Chart

a) Work Integration with Groups

We had to work closely with the SOLR team to ensure a smooth integration into the system, as they are our gateway to everything as seen in Fig 1. We have expectations given to us from each individual group, but we had to clear everything with the SOLR team first to ensure we are not causing a problem within the system because we have data coupling with the contents of the index which was created by them. Hue makes calls exclusively through this index to do all forms of communication to other groups which means the SOLR Team is responsible for any information we expect to display within our interface. No matter what information needs to be shown, or what format it needs to be provided in, HUE utilizes a single index and bases all displayed information off of whatever is found within that single index.

2. Literature Review

We have been using the documentation from the websites to gather background information on each of the 3 components we believe we will be working with in addition to the virtual machine we run to allow us to manipulate the default user interface.

a. SOLR

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites. Solr makes it easy for programmers to develop sophisticated, high-performance search applications with advanced features such as faceting (arranging search results in columns with numerical counts of key terms). Solr builds on another open source search technology called Lucene which is a Java library that provides indexing and search technology, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities. Both Solr and Lucene are managed by the Apache Software Foundation [6]. Lucene/Solr downloads have grown nearly ten times over the past three years, to over 6,000 downloads a day. The Solr search server, which provides application builders a ready-to-use search platform on top of the Lucene search library, is the fastest growing Lucene sub-project. Apache Lucene/Solr offers an attractive alternative to the proprietary licensed search and discovery software vendors. At its heart, Solr is a Web application, but because it is built on open protocols, any type of client application can use Solr. HTTP is the fundamental protocol used between client applications and Solr. The client makes a request and Solr does some work and provides a response. Clients use requests to ask Solr to do things like perform queries or index documents. Client applications can reach Solr by creating HTTP requests and parsing the HTTP responses. Client APIs encapsulate much of the work of sending requests and parsing responses, which makes it much easier to write client applications. Clients use Solr's five fundamental operations to work with Solr. The operations are query, index, delete, commit, and optimize. Queries are executed by creating a URL that contains all the query parameters. Solr examines the request URL, performs the query, and returns the results. The other operations are similar, although in certain cases the HTTP request is a POST operation and contains information beyond whatever is included in the request URL. An index operation, for example, may contain a document in the body of the request. Solr also features an EmbeddedSolrServer that offers a Java API without requiring an HTTP connection.

b. HUE

Hue is a lightweight Web server that lets you use Hadoop directly from your browser. Hue is just a ‘view on top of any Hadoop distribution’ and can be installed on any machine.The next step is then to configure Hue to point to your Hadoop cluster. By default Hue assumes a local cluster (i.e. , there is only one machine) is present. In order to interact with a real cluster, Hue needs to know on which hosts are distributed the Hadoop services. There are a number of applications in the HUE Ecosystem which separate tasks and handle them individually shown in Fig 2.

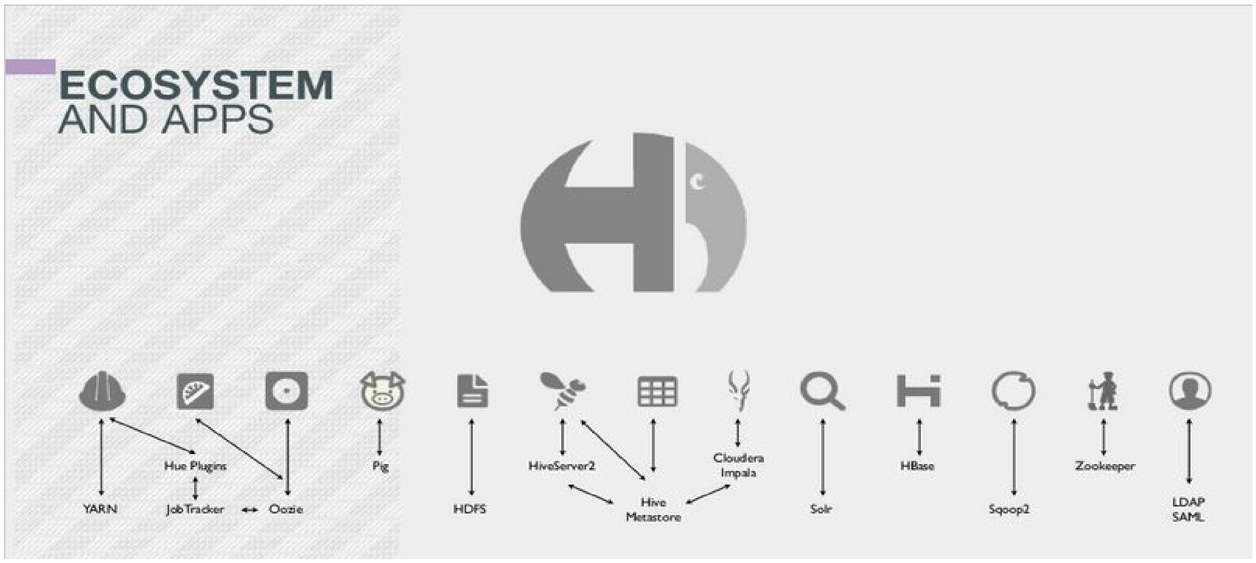


Figure 2. Hue Ecosystem and Apps

HDFS is required for listing or creating files

YARN is the Resource Manager

Hive is what we need running to allow HiveServer2 to send SQL queries.

Impala is needed to specify one of the Impalad address (an Impalad address is a daemon process that runs on each node of the cluster) for interactive SQL in the Impala app.

Solr Search used to specify the address of a Solr Cloud.

Oozie server should be up and running before submitting or monitoring workflows.

The **Pig** Editor requires Oozie to be setup with its sharelib.

The **HBase** app works with a HBase Thrift Server version 1. It lets you browse, query and edit HBase tables.

Sentry privileges determine which Hive / Impala databases and tables a user can see or modify. The Security App let's you create/edit/delete Roles and Privileges directly from your browser

Sqoop Efficiently transfers bulk data between Apache Hadoop and structured datastores

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

c. Pig

Is a scripting platform for processing and analyzing large data sets. With YARN as the architectural center of Apache ^{TM} Hadoop, multiple data access engines such as Apache Pig interact with data stored in the cluster. Pig runs on Apache Hadoop YARN and makes use of MapReduce and the Hadoop Distributed File System (HDFS). The language for the platform is called Pig Latin, which abstracts from the Java MapReduce idiom into a form similar to SQL. While SQL is designed to query the data, Pig Latin allows you to write a data flow that describes how your data will be transformed (such as aggregate, join and sort). Since Pig Latin scripts can be graphs (instead of requiring a single output) it is possible to build complex data flows involving multiple inputs, transforms, and outputs. Users can extend Pig Latin by writing their own functions, using Java, Python, Ruby, or other scripting languages. Pig Latin is sometimes extended using UDFs (User Defined Functions), which the user can write in any of those languages and then call directly from the Pig Latin.

Pig was designed for performing a long series of data operations, making it ideal for three categories of Big Data jobs: Extract-transform-load (ETL) data pipelines, Research on raw data, and Iterative data processing.

3. Requirements

As stated on the syllabus, one of our course requirements is to work in groups on a section of the IDEAL project. We were separated based on each individual foreseen branch of the project and divided work amongst the people within those groups. As time went on those initial requirements evolved and were expanded due to the dynamic workflow of the class. The Front-End team was expected to create a tool which would do the following:

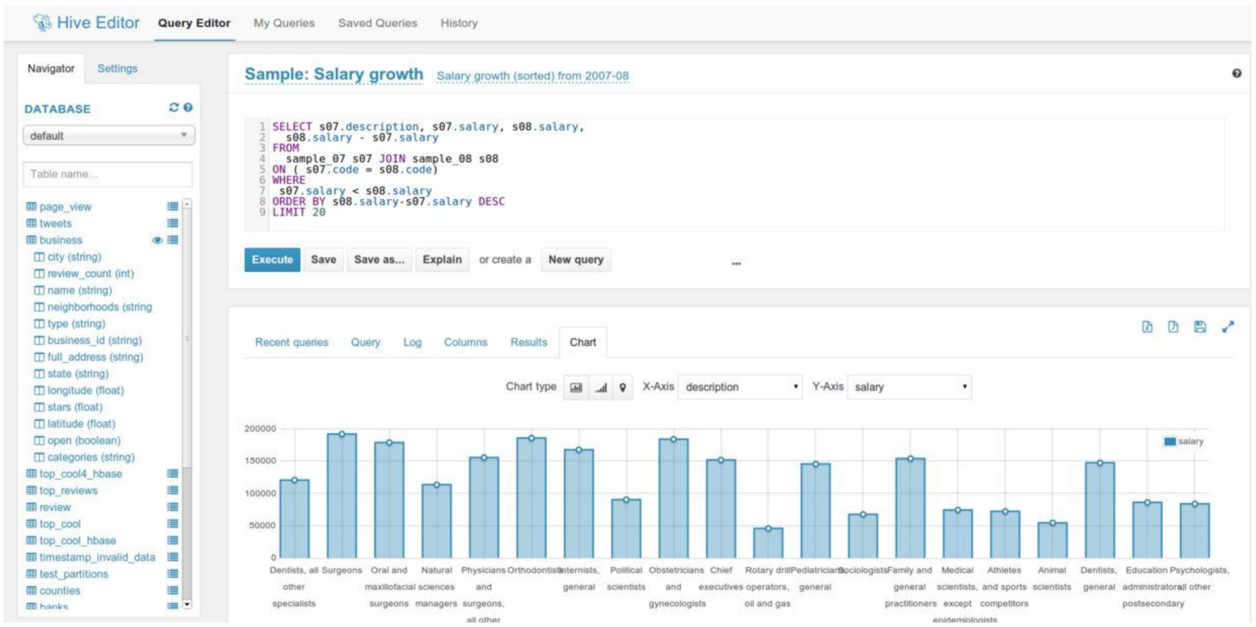
- Faceted searches needed to be allowed and specifically defined with more parameters.
- Needed to be able to perform searches utilizing the provided social network hierarchy.
- Possibility to give users the ability to create dynamic graphs and interactive polygons on command.
- Language should have been extremely straightforward and be put in simple to understand format to prevent new users from being forced to learn a lower level language in order to initiate more advanced functionalities.
- Implemented a way to save and track user authentication data.

While meeting the needs and expectations of the other teams through the following actions;

- Collaborated with various sections of the SOLR management team to discuss how to receive more defined facets to allow more in depth and user specific searches on either tweets, web pages or both.
- Collaborated with Front-End Visualization team to decide if graphing capabilities are feasible within our established UI or if it needs to be completely separate.
- Conferred with Social Network team to plan how to deal with cold case user data.
- Investigated a method of documenting user query data in order to give more unique related document selection.

4. Design

Hue provides us with an interface which is already compatible, however it is lacking in a few areas and is not very easy to navigate. Our main goal was not to reinvent the wheel but more to give it a shiny new look and a new set of tires. HUE is open source and we intend to continue altering the given code to allow us to make adjustments which will give the user the ability to give a more defined search, while simultaneously simplifying the requirement of a user to perform said search. In the past, in order to perform a search a user had to use a format similar to SQL which is not easy for people who are not well versed in a technological field. Our intention is to create an interface which is a much easier for a person to use.



```sql
SELECT s07.description, s07.salary, s08.salary,
  s08.salary - s07.salary
FROM
  sample_07 s07 JOIN sample_08 s08
ON ( s07.code = s08.code)
WHERE
 s07.salary < s08.salary
ORDER BY s08.salary-s07.salary DESC
LIMIT 20
```

Figure 3. Faceted Search Example

Figure 3. illustrates a general idea of what a faceted search should look like with its expected actions. We are starting with a couple of facets (Topic, Class, domain, dates) and will expand from there. We have each installed Cloudera on our personal machines and are actually using a mockup of the interface provided by HUE. We also found a way to give an option to capture user data by having a login feature to help identify more relevant queries based on search patterns. We were able to change the logging level to include username and query information and are now in discussions with the Collaborative Filtering Team to ensure we are all getting them all of the information they need. We discovered the log information can be found in '/var/log/hue/runcpserver.log' and are able to now supply executed query to allow better related search data. The general consensus amongst all of the groups is that everyone will give queries using Pig, and luckily, we can optimize our UI to work better given Pig queries. Hue has its limitations so we had to have a discussion with all of the groups in the classroom about how indexing would be handled and passed along to the UI. We had to establish that there would either be a single index given since HUE only takes one index for searching in a dashboard, or there would have to be two separate HUE query sites which we would then have to find a way to seamlessly allow a user to search both tweets and webpages simultaneously. Adding the index to our data table was our main priority so we could get the UI functional and let us define the queries for the facets. When we were finally given access to the physical cluster, we were able to instantiate test dashboards to utilize the index and dataset on the real machine within minutes. The only hurdle now is to be sure that the given dataset has columns formatted to allow us to utilize the different aspects of the UI. For example there is a mapping option which can only be used if there is a column for latitude and longitude given as a float value and will not work any other way. HUE is a great option for standing up a quick instance, but we have come to find out that it is not the best choice if you need to do many adjustments as HUE is very particular as to how it receives and sends data between its components.

5. Implementation

a. Timeline

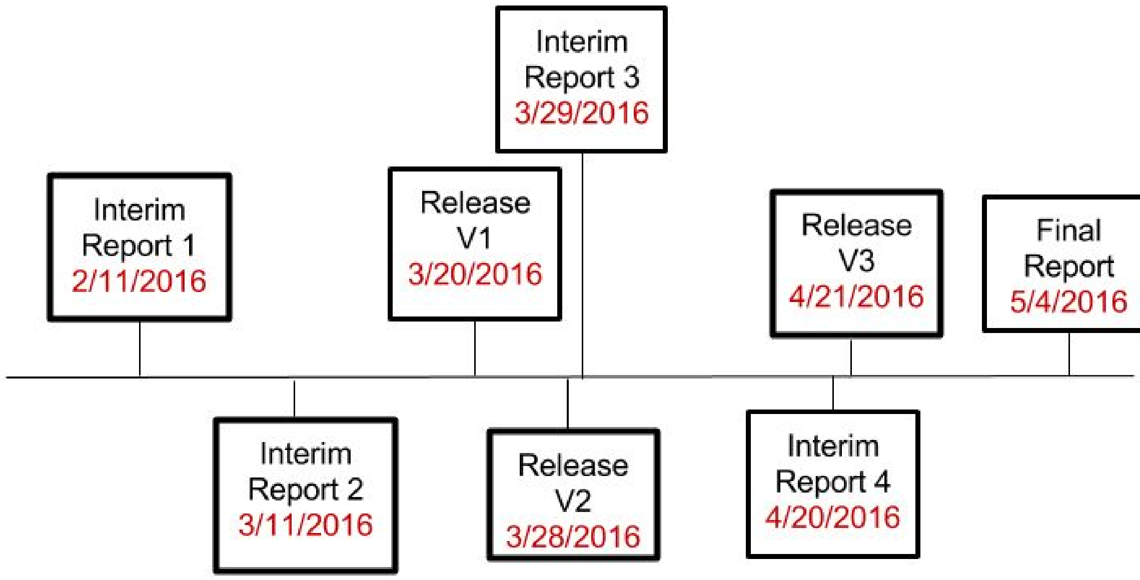

Interim Report 1 2/11/2016
Interim Report 2 3/11/2016
Release V1 3/20/2016
Release V2 3/28/2016
Interim Report 3 3/29/2016
Interim Report 4 4/20/2016
Release V3 4/21/2016
Final Report 5/4/2016

Fig 4. Timeline

b. Milestones

See the list of milestones below and for the dates associated with these milestones, see the schedule in Table 1 below:

- Complete Hue tutorial for Faceted search
- Import data into Solr
- Complete working version of Hue
- *Search* implementation running on VM/localhost on small dataset (V1)
- *Search* implementation on Hadoop cluster on small dataset (V2)
- Complete fully functional implementation of *Search* on full dataset (V3)

c. Deliverables

See the list of deliverables below and for more details, see Table 1 below:

- Stand-alone implementation of a D3 social network graph on VM/localhost
- Stand-alone implementation of Hue interface on VM/localhost
- *Search* implementation running on VM/localhost on small dataset (V1)
- *Search* implementation on Hadoop cluster on small dataset (V2)
- *Search* implementation on Hadoop cluster on full dataset (V3)

Schedule	
Milestone	**Dates**
Install Hue	2/8/2016
Interim Report 1	**2/11/2016**
Complete Hue tutorial for Faceted search	2/18/2016
Complete working version of Hue	3/6/2016
Interim Report 2	**3/11/2016**
Search implementation running on VM/localhost on small dataset	3/20/2016
Search implementation on Hadoop cluster on small dataset	3/28/2016
Interim Report 3	**3/29/2016**
Interim Report 4	**4/20/2016**
Search implementation on Hadoop cluster on full dataset	4/21/2016
Complete fully functional implementation of *Search* on full dataset	4/21/2016
Final Report	**5/4/2016**

Table 1. Schedule

d. Discussion

Initially, we explored four possible options for implementing this project. These are:

1. Create a new interface that implements faceted search and big data visualization
2. Expand upon the Hue interface and add visualization capabilities like rendering a network graph that shows relationships between tweeters, system search users, search result contents, etc. By creating a new widget that displays big data visualizations within the Hue interface. This requirement has since been modified and the deliverable is now out of scope for this project.
3. Modify the existing Hue HTML widget to make it display big data visualizations
4. Expand on the IDEAL Archives Interface Demonstration created by Mohamad Magdy Farag

After exploring the options above and learning more about Hue, we decided to proceed with the implementation of a faceted search interface that includes result suggestions based on historical and current search results. In order to do this, we needed to have a way of identifying users and logging their search history. Two potential ways to do this include the use of session cookies and the use of logging user activities based on the user ID. We decided to pursue the latter. Due to the underlying Hue infrastructure limitations, we cannot provide search capabilities for both registered and unregistered users. Either user logins are turned on or off. Our proposed solution involves running two concurrent instances of our search interface. One for users with logins and the other for those without login credentials. Currently, we have succeeded in creating user logins and a search user group that limits group members to search activities only. This prevents search users from accessing other parts of the Hue system and also prohibits them from elevated user functionality that could compromise the security and availability of the system.

The methodology used for the implementation of *Search* is a combination of iterative and modular software development. These concepts are both borrowed from agile and modular software development methodologies, respectively. An initial faceted search module based on Hue along with the user login constraint will be implemented. Through iterations, these modules will be developed further to expand search functionalities until all requirements are met.

The challenge lies in learning the intricacies of Hue including how to modify the results and present them in a way that is intuitive for end users, while providing the required core functionality. An additional challenge is discovering the best way to keep track of current and past user search history.

6. User Manual

Click on the links below to view detailed instructions for each topic:

a. Accessing Search Page

To access the search interface, go to the Hue web page using the URL <host>:8888/search. ***Replace <host> with the IP address or fully qualified domain name of the Hue server*** Enter login credentials when prompted (see Figure 5 below). A search page will load.

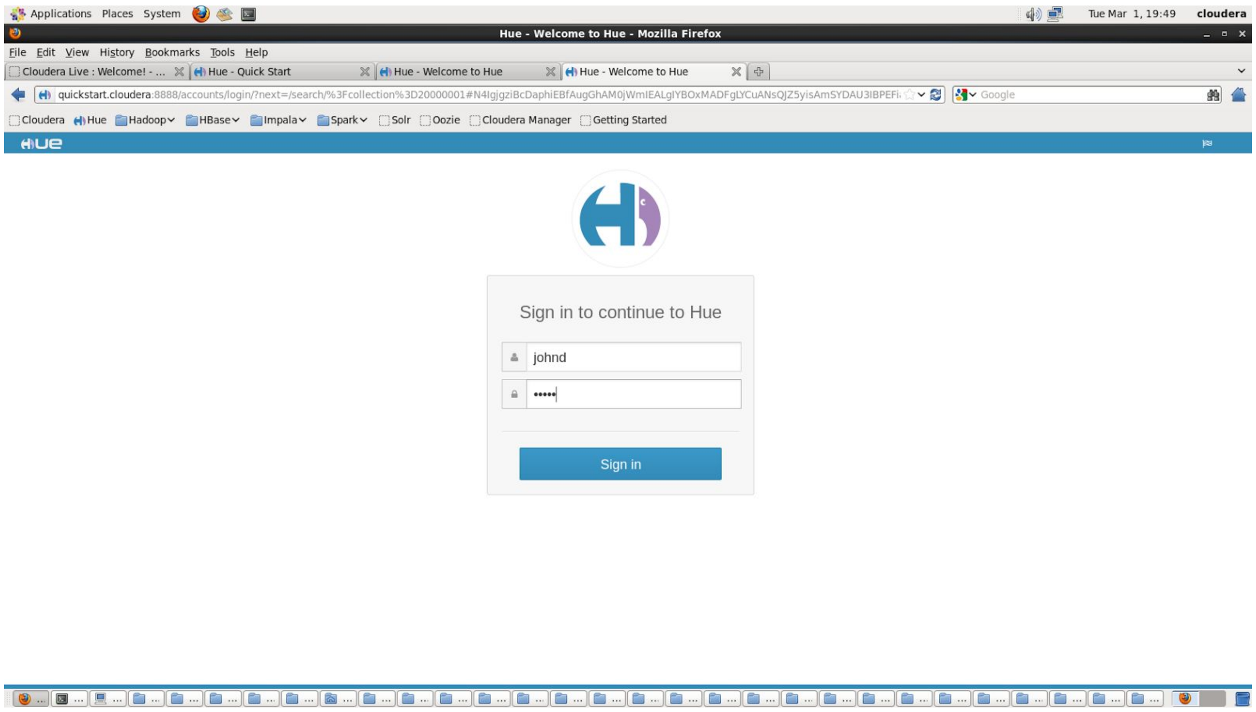
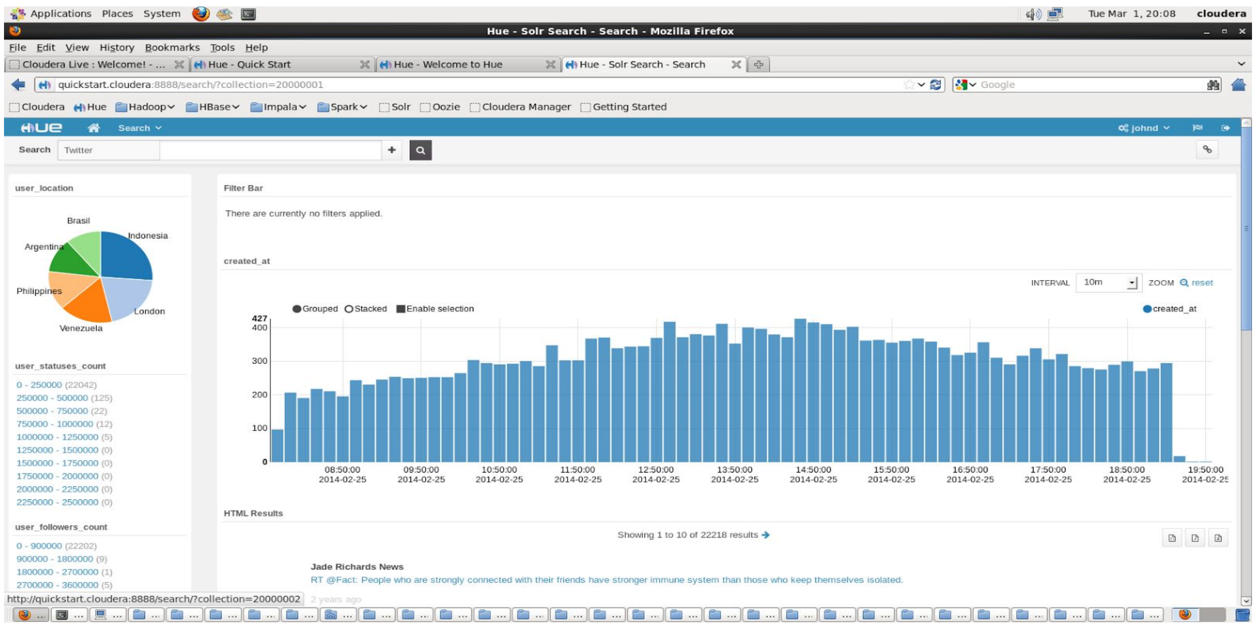

Applications Places System
Tue Mar 1, 19:49 cloudera
Hue - Welcome to Hue - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Cloudera Live : Welcome! - ...
Hue - Quick Start
Hue - Welcome to Hue
Hue - Welcome to Hue
quickstart.cloudera:8888/accounts/login/?next=/search/%3Fcollection%3D20000001#N4IgjgziBcDaphiEBfAugGhAM0jWmIEALgIYBOxMADFgLYCuANsQJZ5yisAmSYDAU3IBPEFi
Google
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
Sign in to continue to Hue
johnd
Sign in

Fig 5. Search page login


Applications Places System
Tue Mar 1, 20:08 cloudera
Hue - Solr Search - Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Cloudera Live : Welcome! - ...
Hue - Quick Start
Hue - Welcome to Hue
Hue - Solr Search - Search
quickstart.cloudera:8888/search/?collection=20000001
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
HUE Search johnd
Search Twitter
user_location
Brasil
Indonesia
Argentina
Philippines
London
Venezuela
user_statuses_count
user_followers_count
Filter Bar
created_at
INTERVAL 10m ZOOM reset
Grouped Stacked Enable selection
created_at
HTML Results

user_statuses_count

0 - 250000 (22042)
250000 - 500000 (125)
500000 - 750000 (22)
750000 - 1000000 (12)
1000000 - 1250000 (5)
1250000 - 1500000 (0)
1500000 - 1750000 (0)
1750000 - 2000000 (0)
2000000 - 2250000 (0)
2250000 - 2500000 (0)

user_followers_count

0 - 900000 (22202)
900000 - 1800000 (9)
1800000 - 2700000 (1)
2700000 - 3600000 (5)

There are currently no filters applied.

Showing 1 to 10 of 22218 results

Jade Richards News
RT @Fact: People who are strongly connected with their friends have stronger immune system than those who keep themselves isolated.

http://quickstart.cloudera:8888/search/?collection=20000002

Figure 6. Search page

b. Faceted Search

On the search page (see search help for instructions for loading search), facets are listed on the left hand side. Click on relevant Facets to modify search results (see Fig. 7 below).

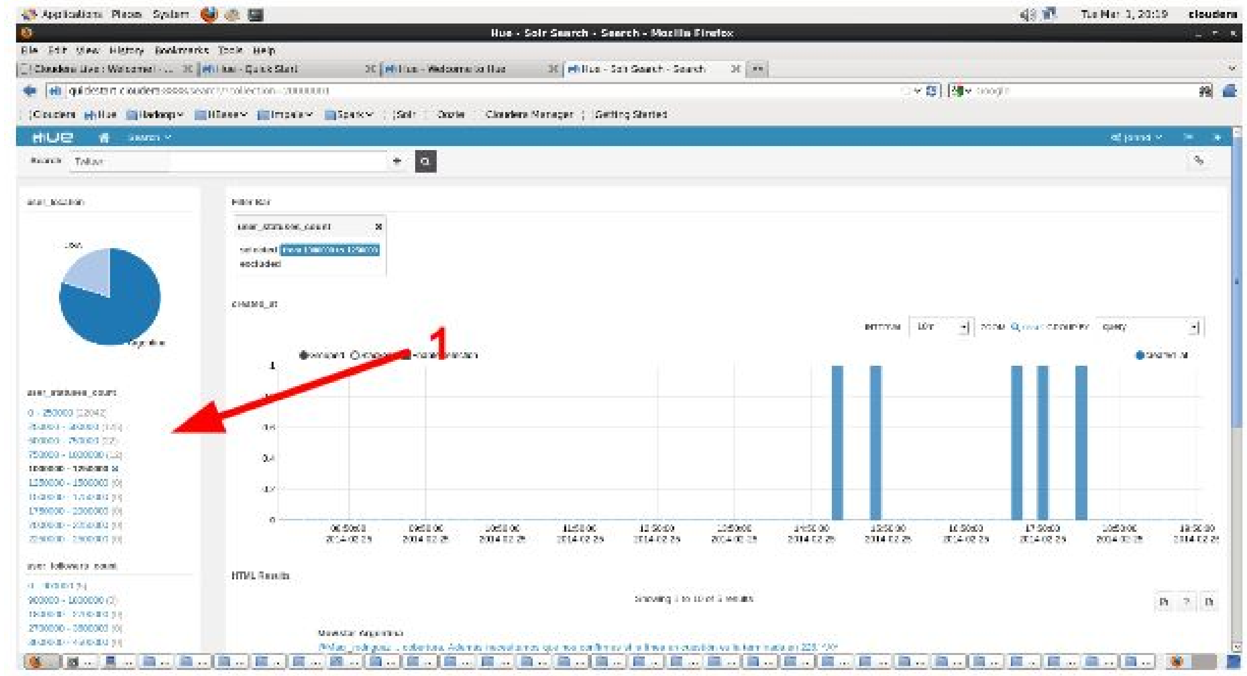


Fig 7. Facets

Clicking on a facet value will modify the current search results and an ‘x’ will appear near to the facet that has been clicked on. To cancel the current faceted search, click on the ‘x’ (highlighted by red arrow and numeral 1) besides the facet that had been previously clicked on.

c. Free Text Search

1. On the search page (see search help for instructions for loading search), enter search terms in the search box (see Fig. 8 below) and hit Enter key
2. The search results will be displayed in the middle of the page.

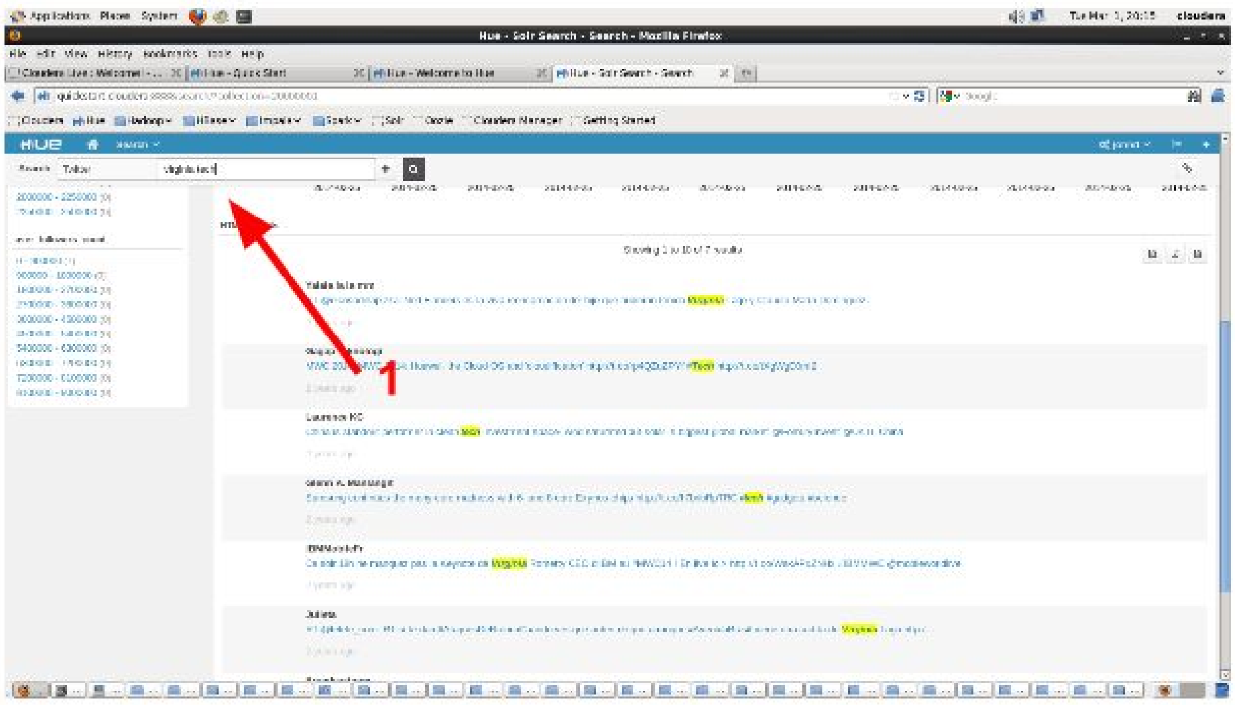


Fig 8. Free text search

7. Developer Manual

a. Software Development Process

The software development process has been iterative with the following involved in each iteration:

- Conducting team meetings
- Meeting with GRAs
- Reviewing relevant material and documentation
- Implementing what has been learnt on individual virtual machines
- Collaborating with partner teams

Team collaboration has been either in person or via email. Collaboration with GRAs has also been both in person and via email, while collaboration with other teams has been mainly via email, with some meetings being held in person during classroom discussions. Progress thus far can be tracked by reviewing the schedule in Section 5 of this paper. We are pleased to note that we are on schedule.

We initially intended to use GitHub for our software version control, but due to the fact that our Hue implementation is contained within virtual machines, the idea of using GitHub was discarded. We continue to work on individual virtual machines with the intention of using the knowledge gained in these initial stages to collaboratively build the final version of the interface on the cluster, which will be available to the whole class. As shown in the schedule, we are ready to start working on the cluster. However, we are awaiting the ability to login to the cluster so we can start building.

b. Architecture

Search is built using a client - server architecture. The client consists of a web based user interface that is accessed using a browser that resides on the client. In addition to the front-end that provides search functionality as discussed in this paper, a visualization module that is being provided by the Graph Viz team will also provide additional search capabilities. Search processing occurs on the server, which consists of a Hadoop cluster in a remote location running Solr, Spark and other Hadoop services.

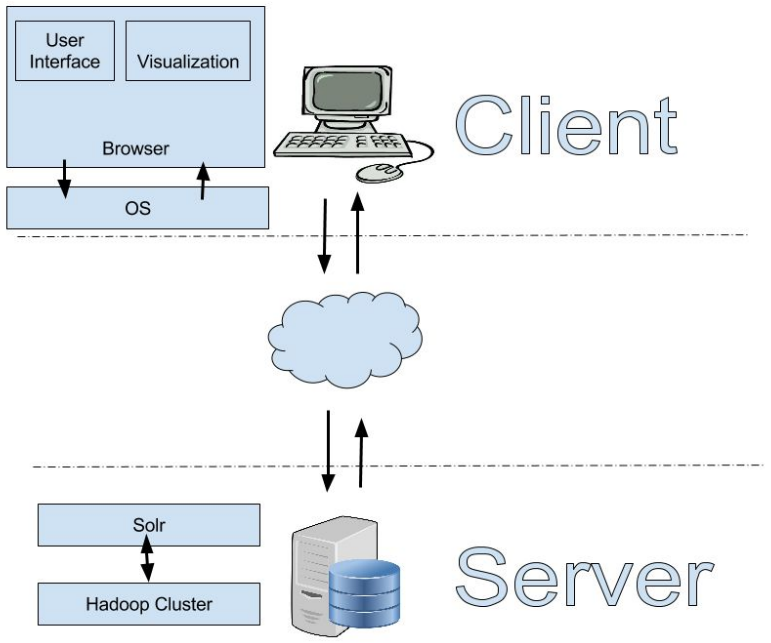


Fig 9. Client server architecture

c. Data Flow Overview

Data flows are shown in Fig. 10 (below). User requests are entered into the Hue interface (Step 1) and Hue queries the database (Step 2). In addition to Search, the graph viz will also be implementing visualization capabilities as shown in Figs. 9, 10 and 11.

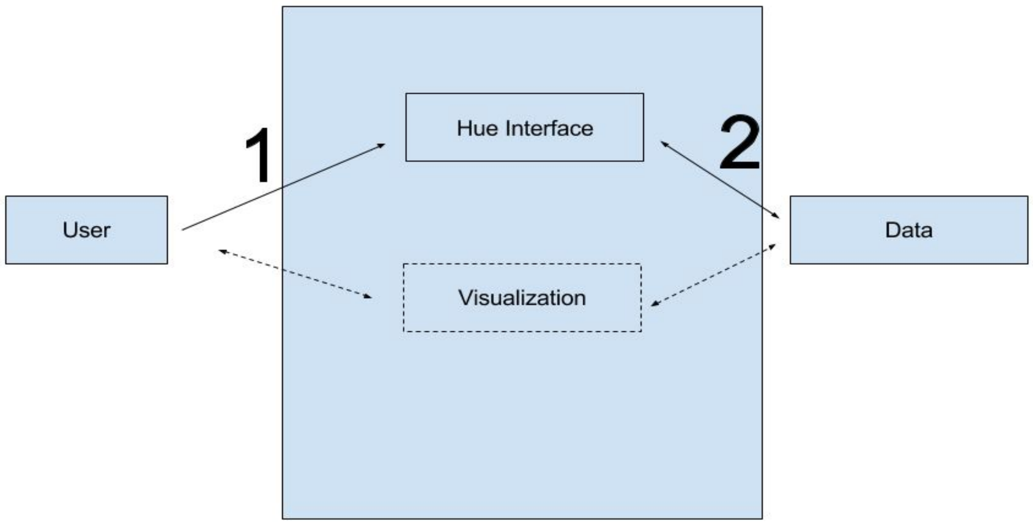


Fig 10. Data flow

d. Module Overview

Our interface consists of 3 modules as shown in figure 11 below: 1) Hue interface module which accepts search queries, 2) the Visualization module, which visualizes search results when invoked, and the data module that includes all services running on the server side to retrieve relevant search results.

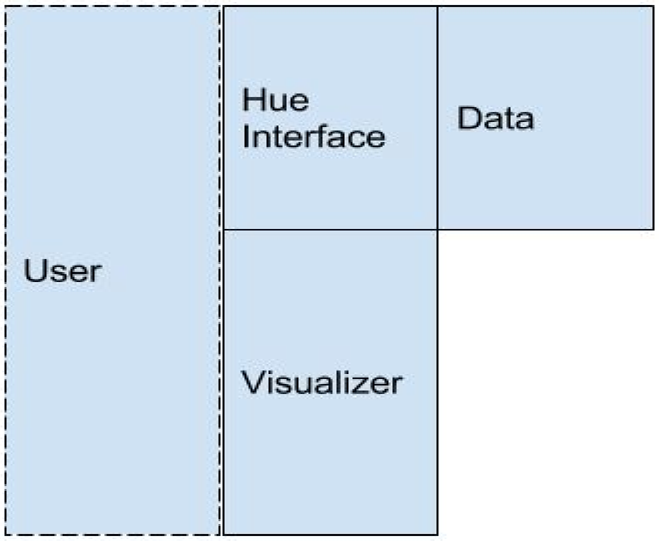


Fig 11. Module overview

e. Project Data:

The data used for this project falls into 3 categories:

i. Complete Corpus Dataset

This is the entire corpus of data that all project teams will eventually interact with. It consists of multiple collections of data from Twitter. In addition to tweet data, there is also web page data that *Search* will query.

ii. Big Collection Dataset

Each project team is assigned a big dataset of tweets in addition to a small collection dataset (see Small collection dataset). The big collection that relates to this paper is the Community dataset. This dataset consists of tweets that have been collected using the ‘#veterans’ hashtag.

iii. Small Collection Dataset

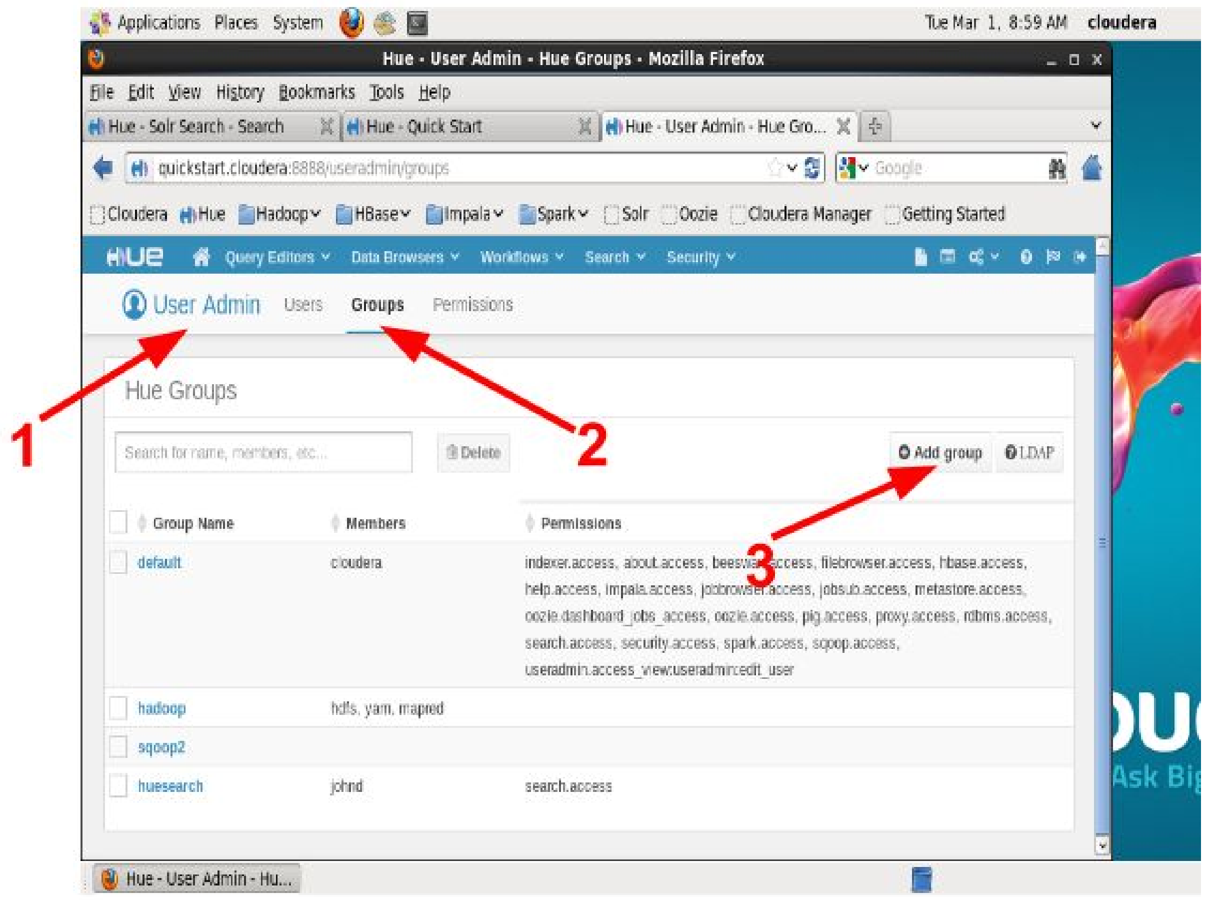
Each project team is also assigned a small collection dataset of tweets in addition to a big collection (see Big collection dataset). This collection consists of 105K of Twitter data collected using the ‘#4thofJuly’ hashtag.

f. Installation, Operation and Maintenance:

i. Creating a Search Group:

Managing users is challenging and as the number of users grows, keeping track of the users and modifying their permissions becomes more challenging. One method that helps administrators keep track of users and simultaneously manage a similar group of users is the creation of groups. In this project, we create a group of users whose only capability is creating search queries using Search. This group of users is also referred to as a search group. In order to create a search group in Hue, take the following steps. (Please note that the steps below are highlighted by the corresponding highlighted red arrows and numerals in Figs. 12 - 14)

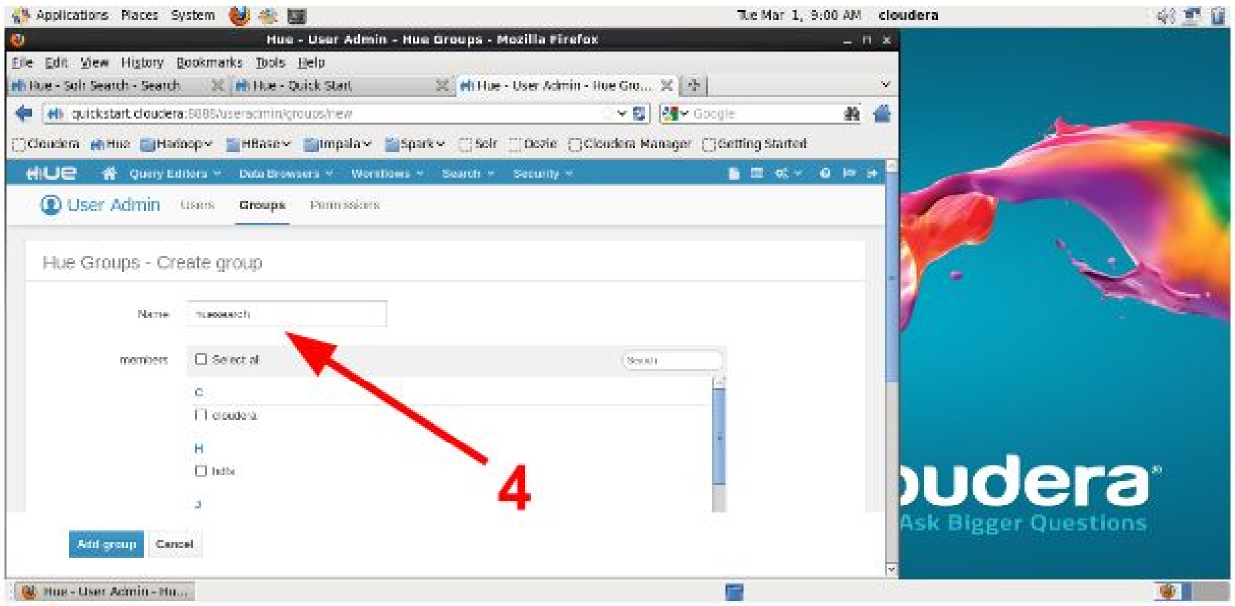
1. Navigate to the User Admin section of the Hue page
2. Click on Groups
3. Click on Add group
4. Enter a group name
5. Select permissions for the group


Hue - User Admin - Hue Groups - Mozilla Firefox
quickstart.cloudera:8888/useradmin/groups
User Admin
Users
Groups
Permissions
Add group
LDAP
Delete
1
2
3

Hue Groups

Group Name	Members	Permissions
default	cloudera	indexer.access, about.access, beeswax.access, filebrowser.access, hbase.access, help.access, impala.access, jobbrowser.access, jobsub.access, metastore.access, oozie.dashboard_jobs_access, oozie.access, pig.access, proxy.access, rdbms.access, search.access, security.access, spark.access, sqoop.access, useradmin.access_view:useradmin:edit_user
hadoop	hdfs, yarn, mapred	
sqoop2		
huesearch	johnd	search.access

Fig 12. Creating groups 1 of 3


Hue - User Admin - Hue Groups - Mozilla Firefox
quickstart.cloudera:8888/useradmin/groups/new
User Admin
Users
Groups
Permissions
Select all
Add group
Cancel
4

Hue Groups - Create group

Name: huesearch

members: cloudera, hdfs

Fig 13. Creating groups 2 of 3

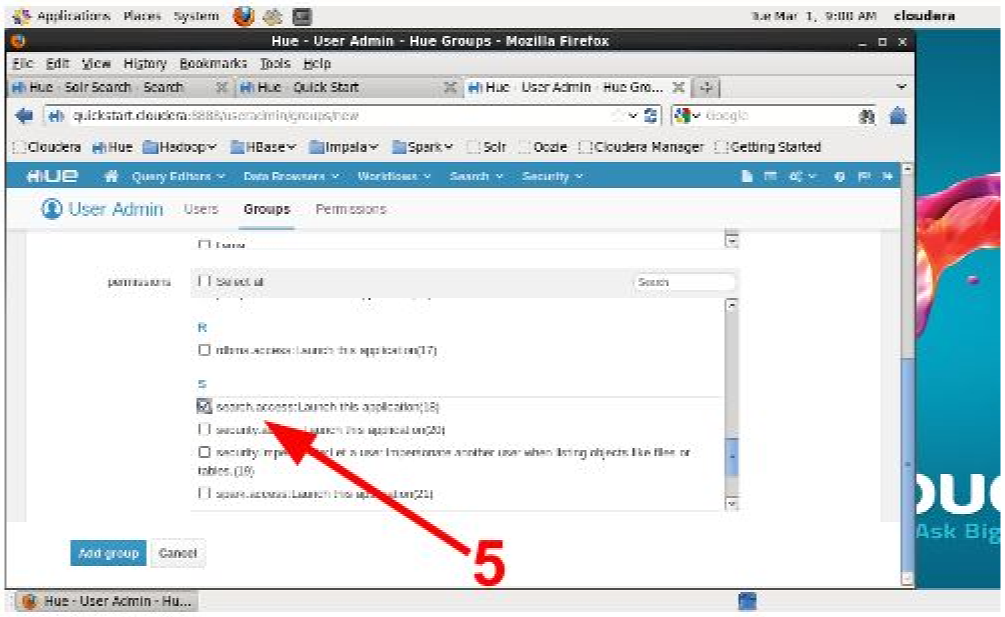
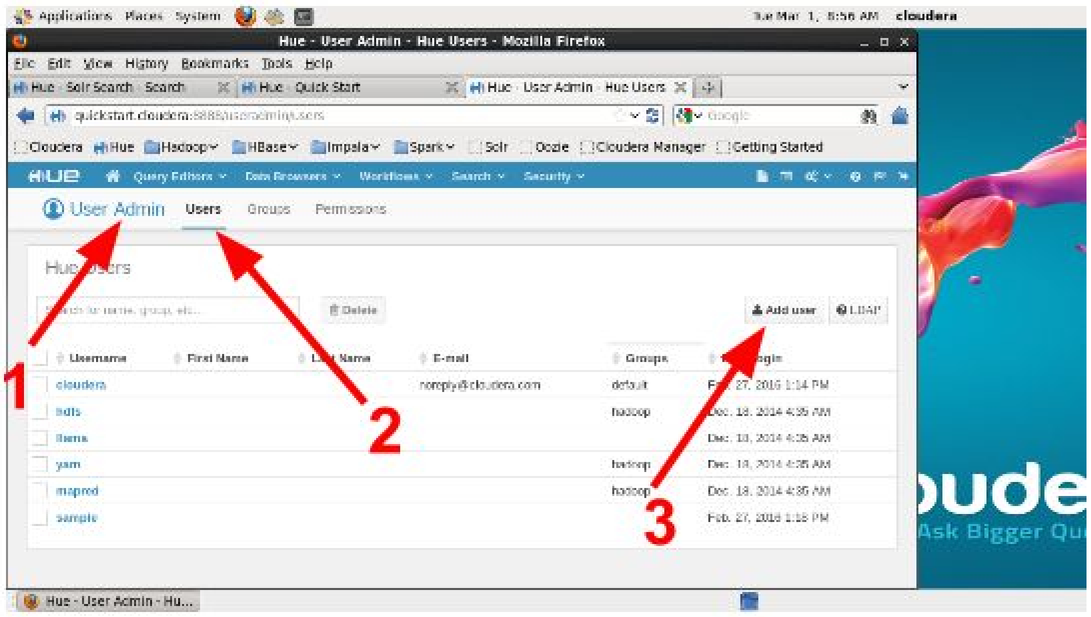

Hue - User Admin - Hue Groups - Mozilla Firefox
User Admin
Users
Groups
Permissions
permissions
Select all
search.access:Launch this application(18)
Add group
Cancel
5

Fig 14. Creating groups 3 of 3

ii. Creating a User and Assigning Them to a Group:

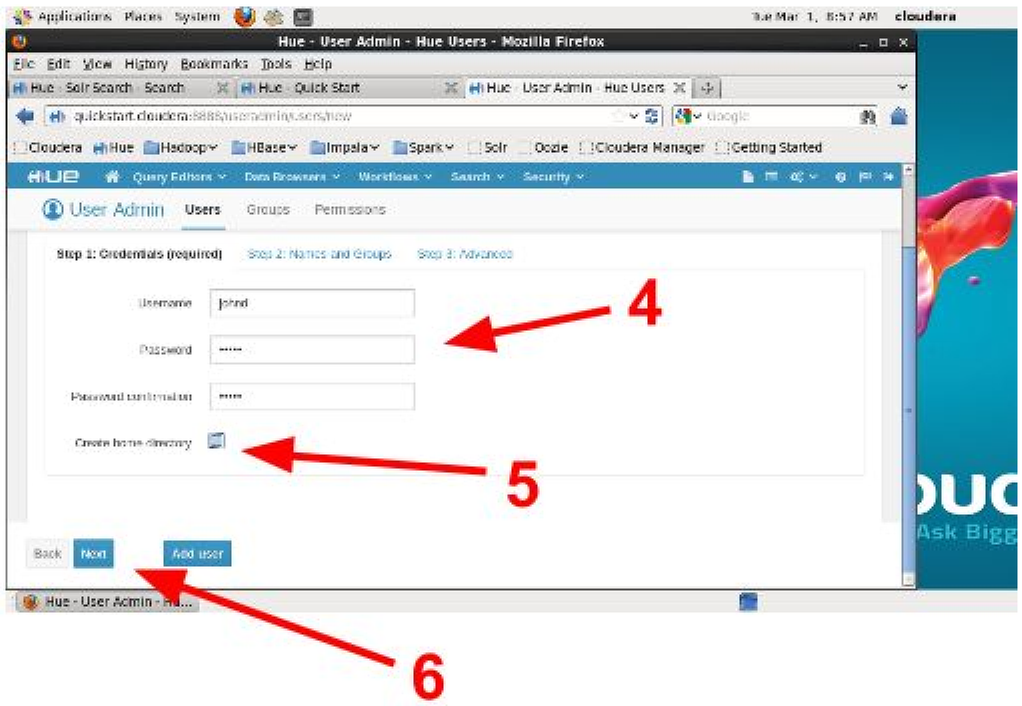
To create a user and assign them to an existing group, follow the steps below. If the group has not been created, see the ‘Creating a search group’, instructions above.
Navigate to the User Admin section of the Hue page. (Please note that the steps below are highlighted by the corresponding highlighted red arrows and numerals in Figs. 15-18)

1. Click on Users
2. Click on Add user
3. Enter the new user’s credentials
4. Uncheck “Create home directory’ option
5. Click Next to select the group that the user will belong to
6. Click Next
7. Select the group(s) that the user should belong to
8. Click Next
9. Select the Active option
10. Click Add user



	Username	First Name	Last Name	E-mail	Groups	Last Login
	cloudera			noreply@cloudera.com	default	Feb. 27, 2016 1:14 PM
	hdfs				hadoop	Dec. 18, 2014 4:35 AM
	llama					Dec. 18, 2014 4:35 AM
	yarn				hadoop	Dec. 18, 2014 4:35 AM
	mapred				hadoop	Dec. 18, 2014 4:35 AM
	sample					Feb. 27, 2016 1:15 PM

Fig 15. Creating users 1 of 4



Username	johnd
Password	•••••
Password confirmation	•••••
Create home directory	✓

Fig 16. Creating users 2 of 4

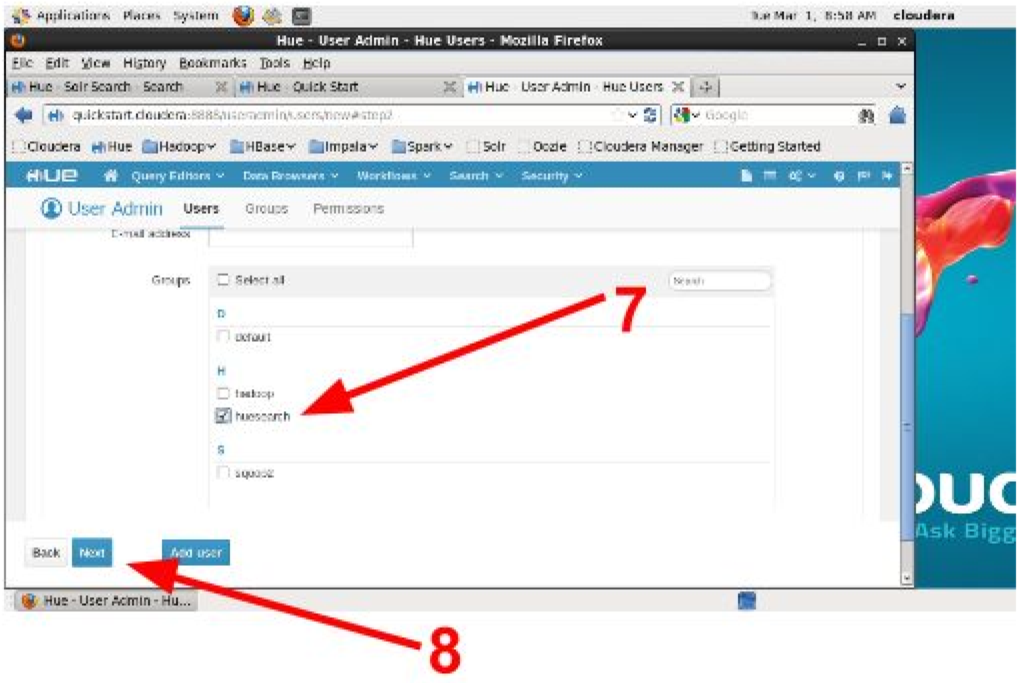


Fig 17. Creating users 3 of 4

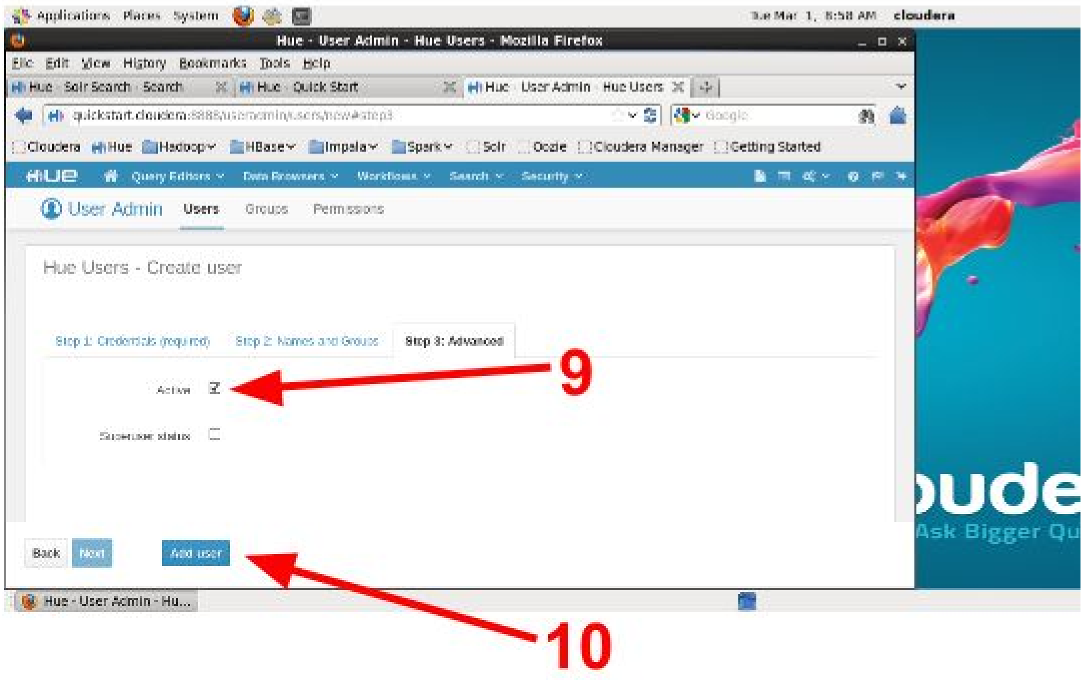
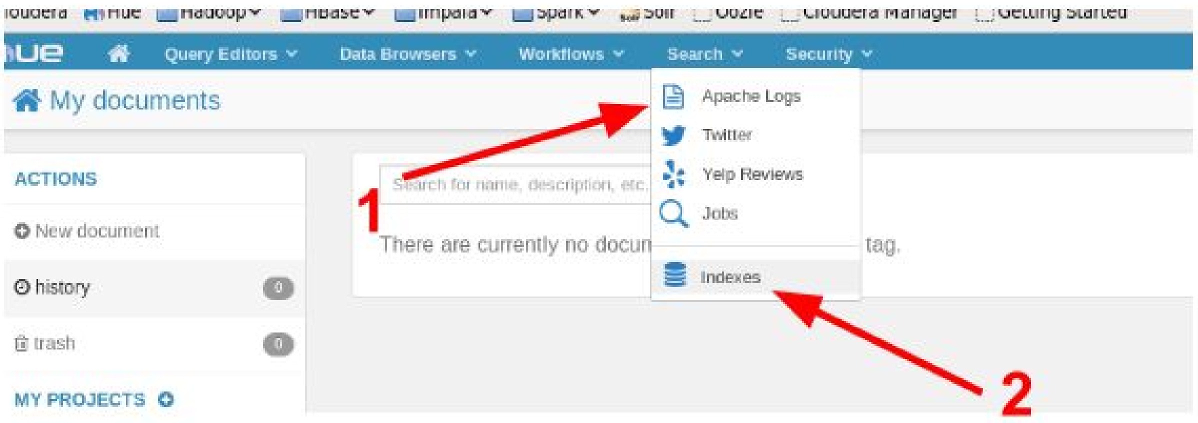


Fig 18. Creating users 4 of 4

iii. Creating a Search Interface/Page

A search interface is the actual search page that users interact with when creating search queries. All of the work conducted in this project is manifested through the search interface. In other words, the search interface is the major deliverable of this project. To a regular end user, the search interface is the search engine. To create a search interface, follow the steps below. Login to Hue as a user with permissions to create a new interface. If you do not see any of the options listed below, verify that you are using an account with sufficient permissions. Please note that the steps below are highlighted by the corresponding highlighted red arrows and numerals in Figs. 19-25.

1. Navigate to Search at the top of the window
2. Click on Indexes
3. Click on Dashboards
4. Click on Create
5. Click on the edit pencil on the top right corner
6. Select an index in the ‘Select a search index’ drop down in the top-left corner
7. Select a layout
8. Drag and drop search components into the search grid



There are currently no docum... tag.

Fig 19. Creating a search interface 1 of 7

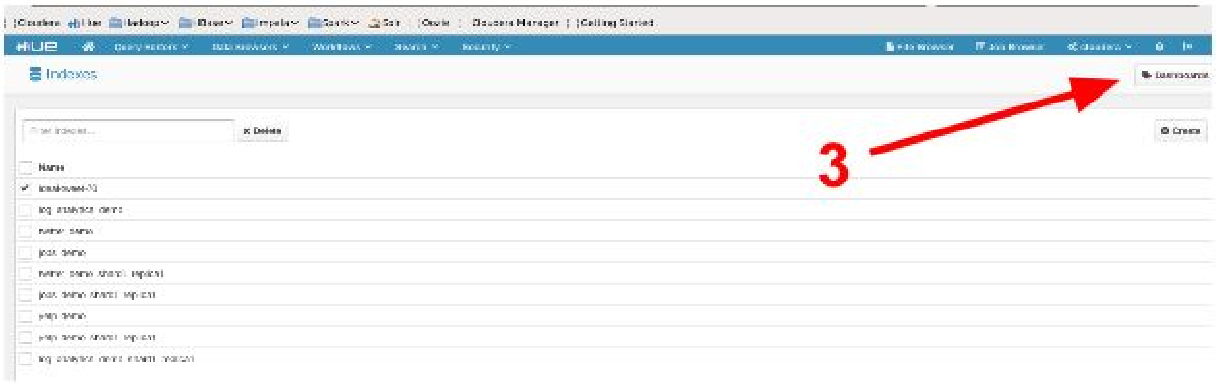
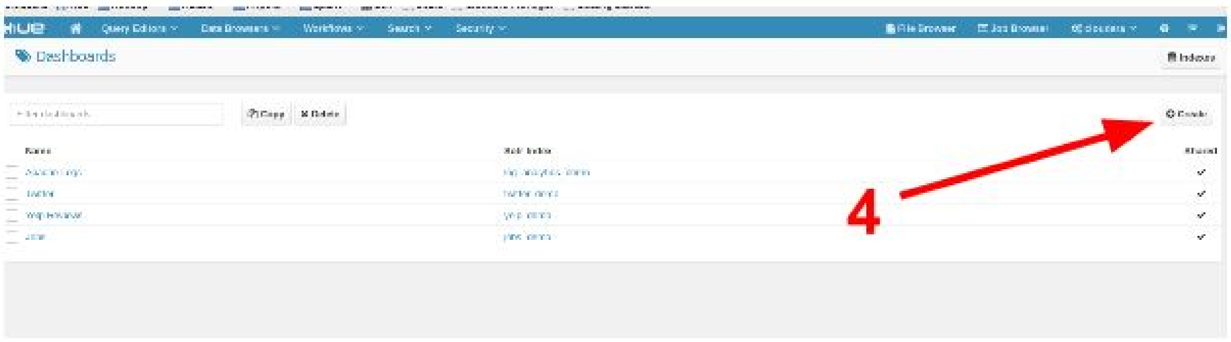
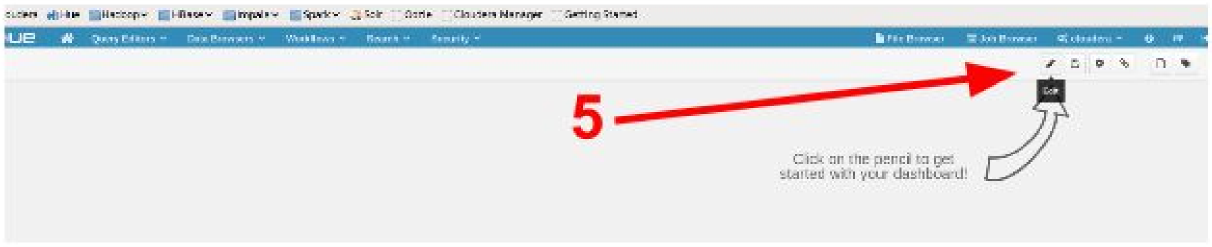


Fig 20. Creating a search interface 2 of 7



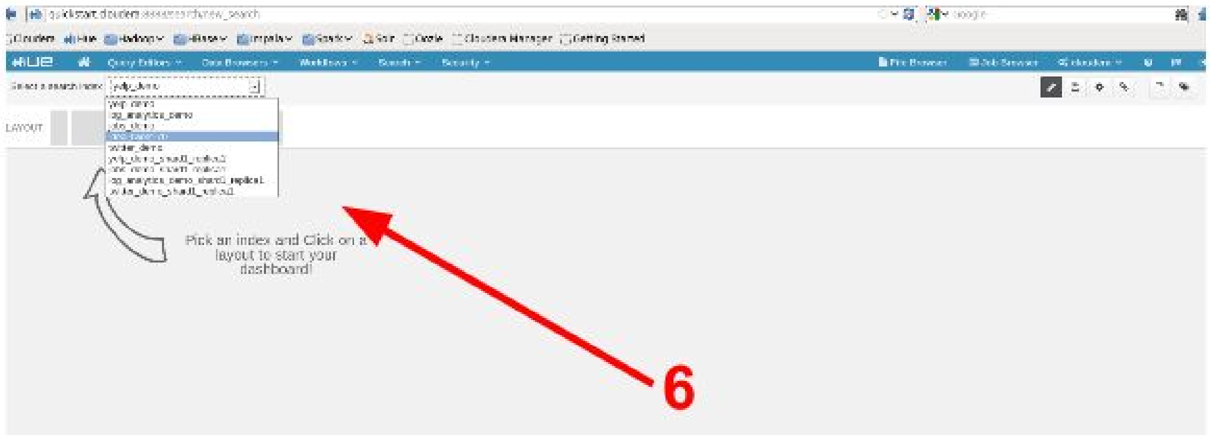
Name	Solr Index	Shared
Apache Logs	log_analytics_demo	✓
Twitter	twitter_demo	✓
Yelp Reviews	yelp_demo	✓
Jobs	jobs_demo	✓

Fig 21. Creating a search interface 3 of 7



Click on the pencil to get started with your dashboard!

Fig 22. Creating a search interface 4 of 7



Select a search index: yelp_demo

Pick an index and Click on a layout to start your dashboard!

Fig 23. Creating a search interface 5 of 7

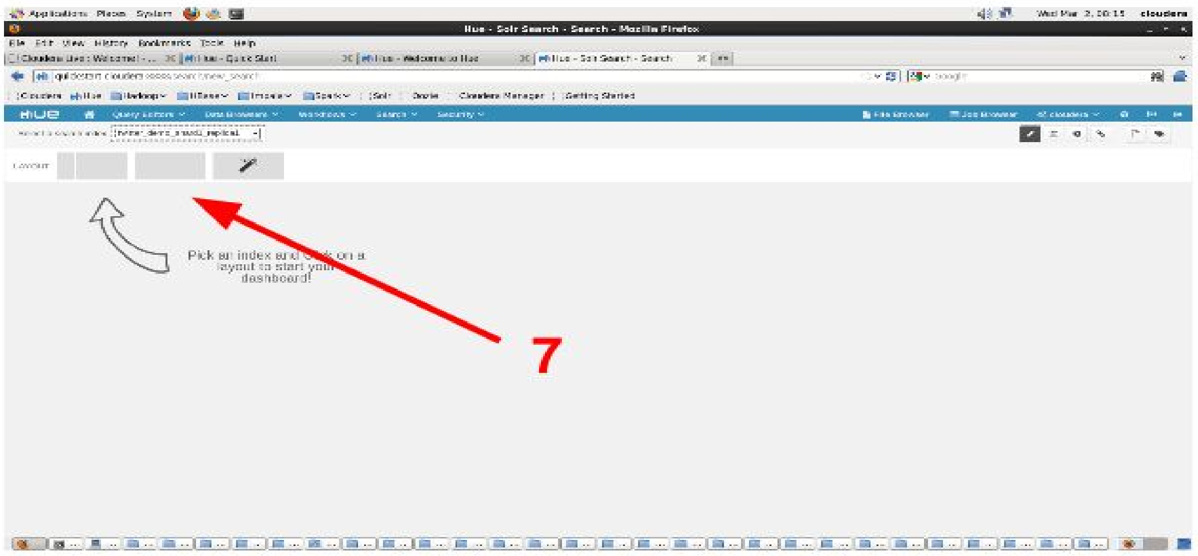

Hue - Solr Search - Search - Mozilla Firefox
LAYOUT
Pick an index and Click on a layout to start your dashboard!
7

Fig 24. Creating a search interface 6 of 7

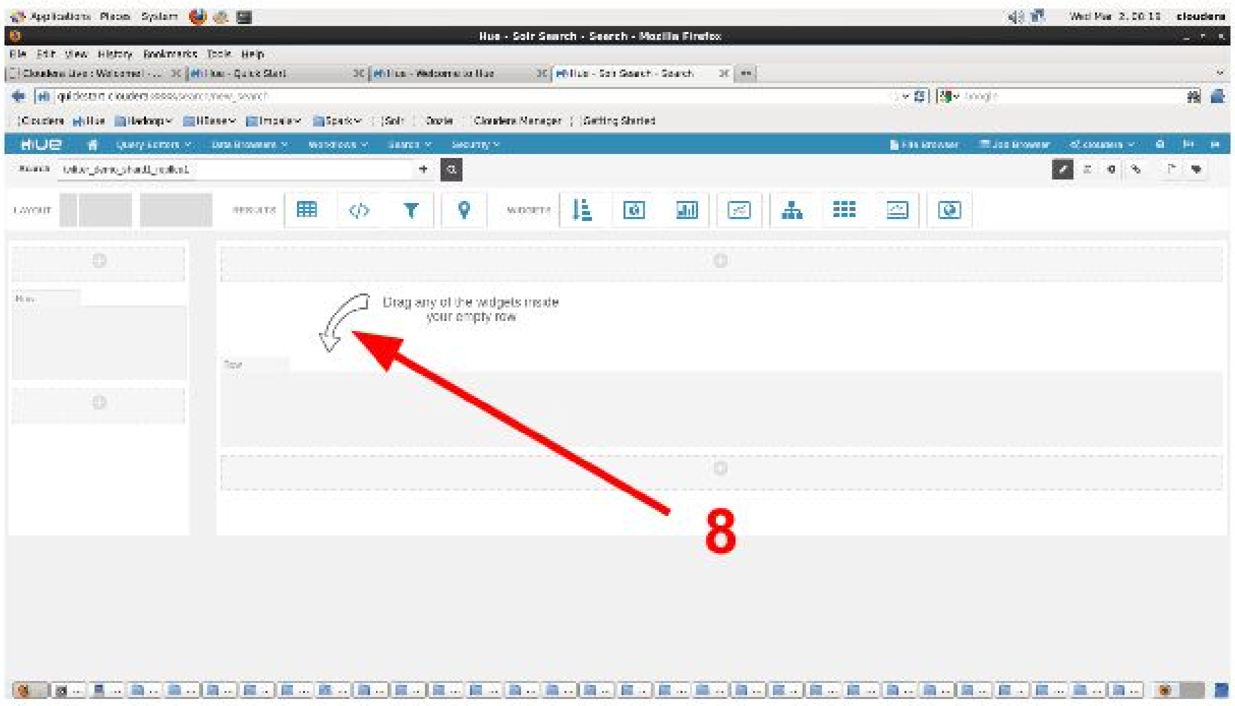

Hue - Solr Search - Search - Mozilla Firefox
LAYOUT
RESULTS
WIDGETS
Drag any of the widgets inside your empty row
8

Fig 25. Creating a search interface 7 of 7

iv. Sharing the Search Interface/Page

After creating a search interface, it is important to deliver it to the end users. In other words, it is important to provide the search engine to end users so they can use it. The process of providing the search engine to users is known as sharing. In order to share the newly created search interface, login to Hue as a user with permissions to edit a search interface. if you do not see any of the options listed below, verify that you are using an account with sufficient permissions. To share a search interface, follow the steps below. Please note that the steps below are highlighted by the corresponding highlighted red arrows and numerals in Figs. 26-27.

1. Click on Share
2. Copy the URL that appears in the 'Share this dashboard' window that pops-up

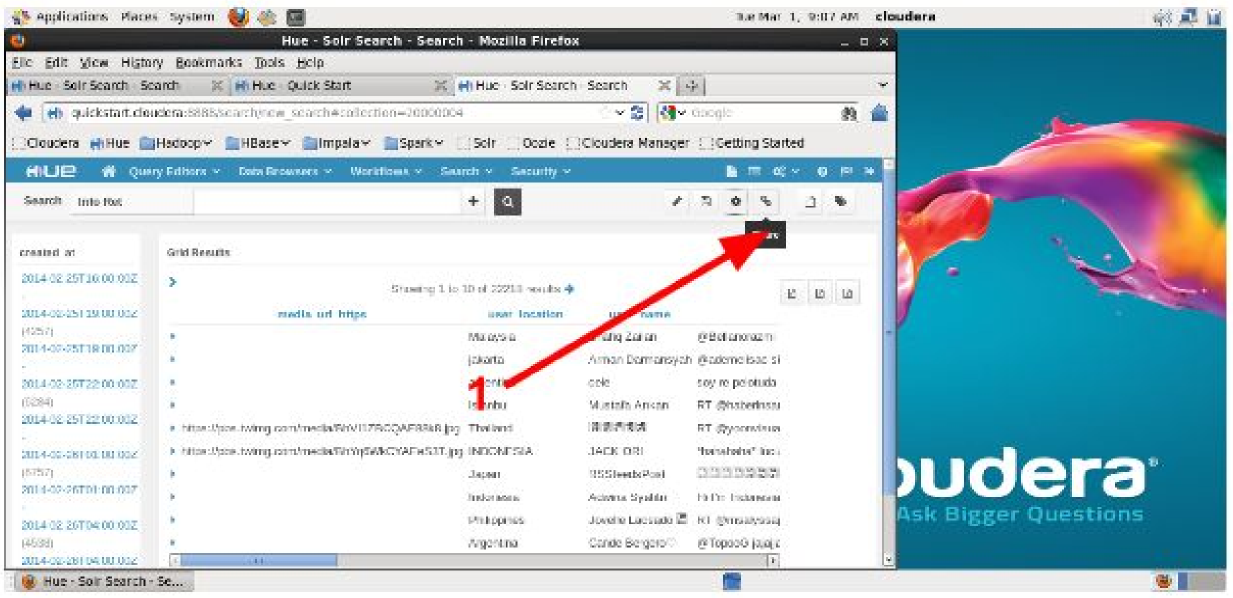


Fig 26. Sharing a search interface 1 of 2

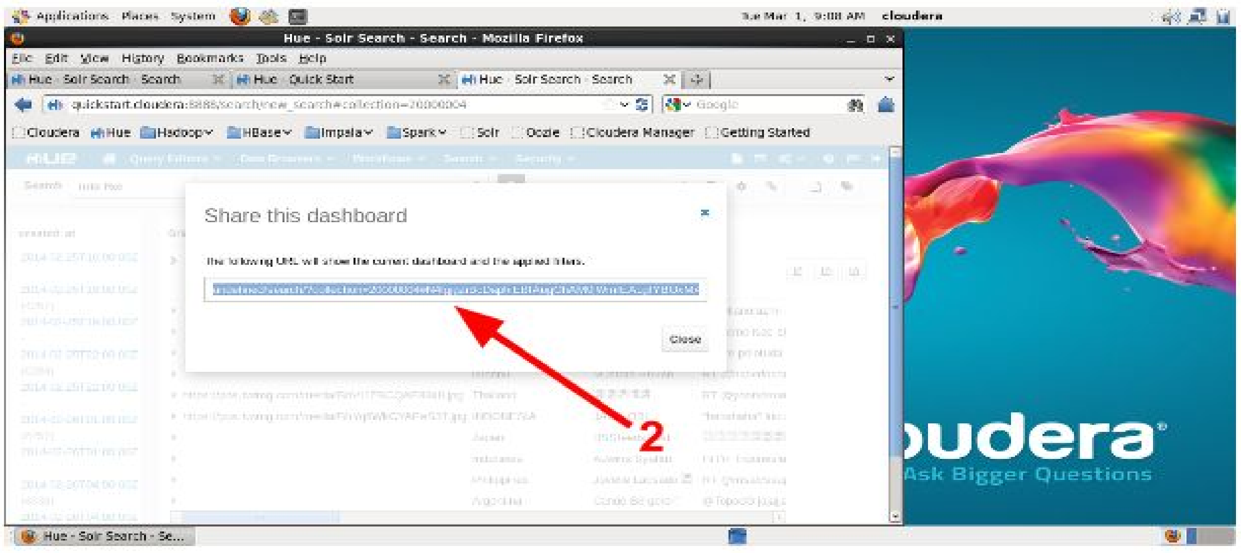

Share this dashboard
The following URL will show the current dashboard and the applied filters.
Close
2
cloudera
Ask Bigger Questions

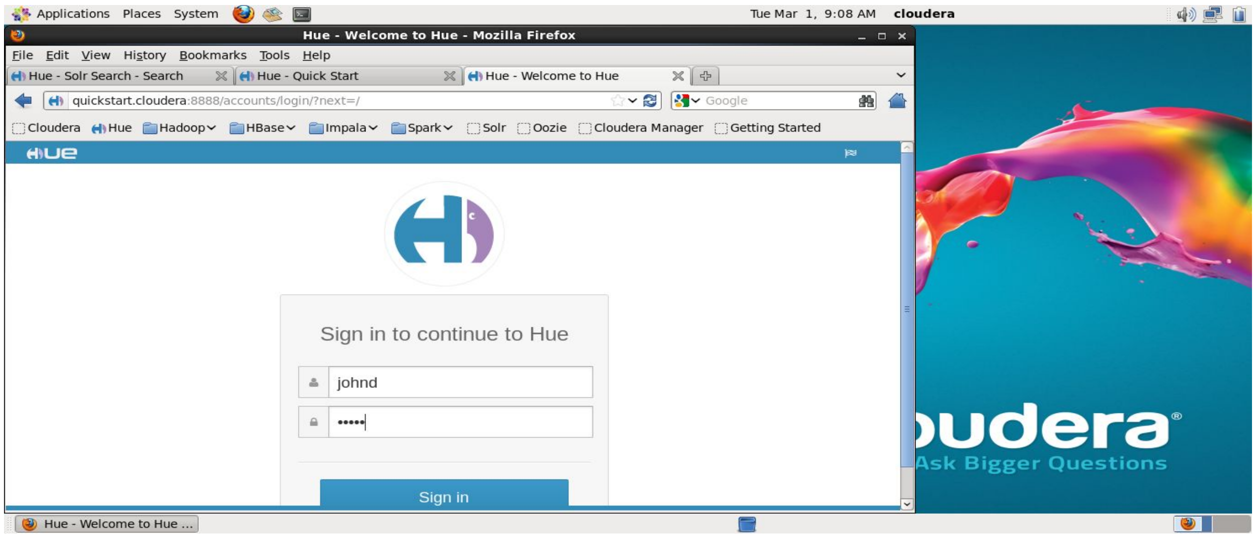
Fig 27. Sharing a search interface 2 of 2

v. Testing User Access and Search

Use the ‘Creating a user and assigning them to a group’ steps described above to create a test user for testing purposes. To test a user interface, follow the steps below.

1. Login to Hue as the test user you just created
2. Verify that you have access to the search page

If you are unable to see the search page, verify that the ‘Creating a user and assigning them to a group’ and the ‘Sharing the search page’ instructions have been followed correctly.



Hue - Welcome to Hue - Mozilla Firefox

quickstart.cloudera:8888/accounts/login/?next=/

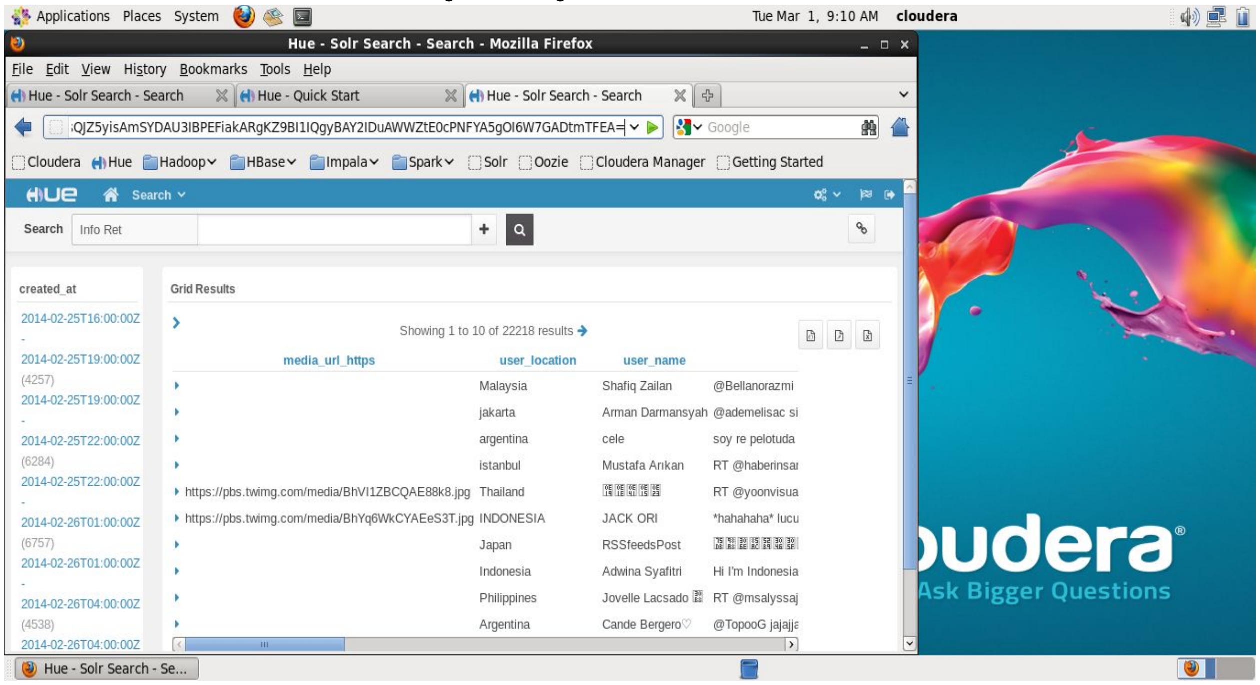
Sign in to continue to Hue

Username: johnd

Password: •••••

Sign in

Fig 28. Testing the search interface 1 of 2



Hue - Solr Search - Search - Mozilla Firefox

Search: Info Ret

created_at
2014-02-25T16:00:00Z - 2014-02-25T19:00:00Z (4257)
2014-02-25T19:00:00Z - 2014-02-25T22:00:00Z (6284)
2014-02-25T22:00:00Z - 2014-02-26T01:00:00Z (6757)
2014-02-26T01:00:00Z - 2014-02-26T04:00:00Z (4538)
2014-02-26T04:00:00Z

Grid Results

Showing 1 to 10 of 22218 results

media_url_https	user_location	user_name	
	Malaysia	Shafiq Zailan	@Bellanorazmi
	jakarta	Arman Darmansyah	@ademelisac si
	argentina	cele	soy re pelotuda
	istanbul	Mustafa Arıkan	RT @haberinsaı
https://pbs.twimg.com/media/BhVI1ZBCQAE88k8.jpg	Thailand	[illegible]	RT @yoonvisua
https://pbs.twimg.com/media/BhYq6WkCYAEeS3T.jpg	INDONESIA	JACK ORI	*hahahaha* lucu
	Japan	RSSfeedsPost	[illegible]
	Indonesia	Adwina Syafitri	Hi I'm Indonesia
	Philippines	Jovelle Lacsado	RT @msalyssaj
	Argentina	Cande Bergero♡	@TopooG jajajja

Fig 29. Testing the search interface 2 of 2

vi. Capturing User Search History

In order to capture user search history, logging can be used. The default log information does not include search query information. The default logging level has to be changed to the Debug level so it can capture this required information. To change the logging level, follow the instructions below. Please note that the steps below are highlighted by the corresponding highlighted red arrows and numerals in Figs. 30 - 32.

1. open a terminal window and type the commands shown in Figure 30 below:
2. type a query in the Hue interface
3. view log file located in '/var/log/hue/runcpserver.log' to verify the logging information level (Fig. 33)

```
[cloudera@quickstart ~]$ sudo su
[root@quickstart cloudera]# gedit '/etc/hue/conf/log4j.properties'
[root@quickstart cloudera]#
```

Figure 30. Adjusting logging level

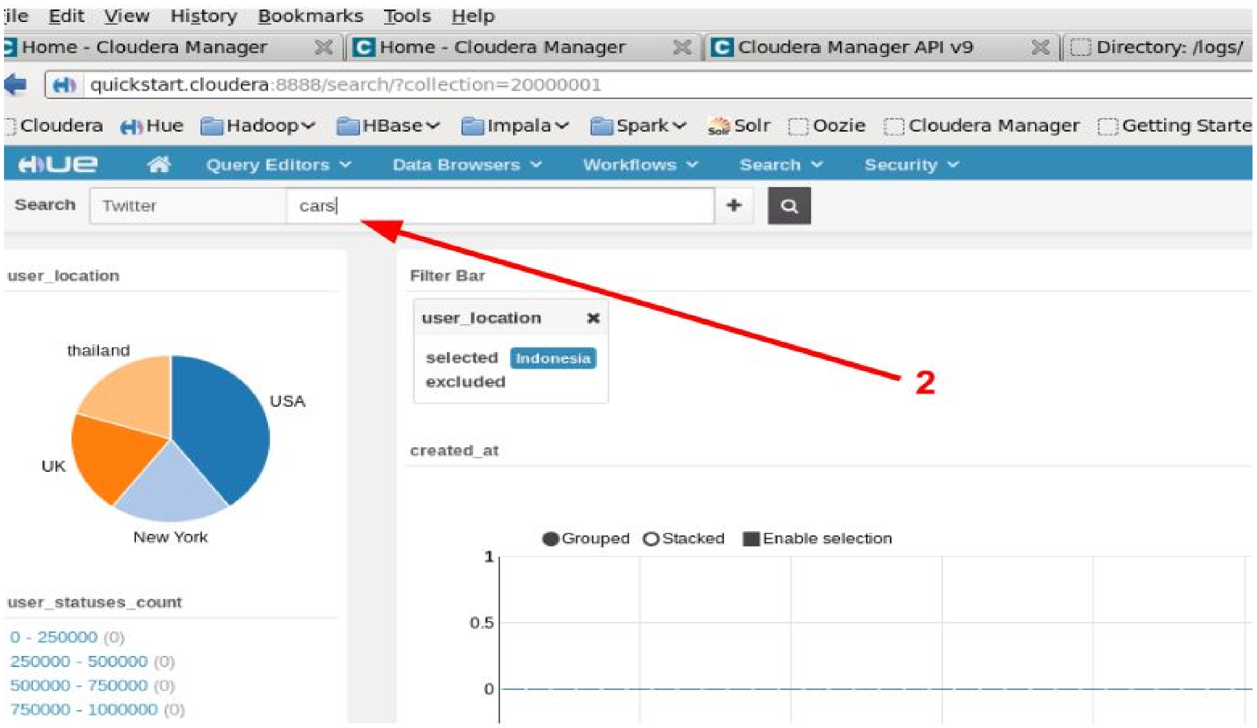


Figure 31. Testing logging information. A screenshot captured from a search page created on one of our virtual machines.

```
"GET
/solr/twitter_demo/select?user.name=hue&doAs=cloudera&q=cars&wt=json&rows=10&start=
0&facet=true&facet.mincount=0&facet.limit=10&facet.range=%7B%21ex%3Dcreated_at%7D
created_at&f.created_at.facet.range.start=2014-02-25T16%3A00%3A00Z&f.created_at.facet.
range.end=2014-02-26T04%3A00%3A00Z&f.created_at.facet.range.gap=%2B10MINUTES&f
.created_at.facet.mincount=0&facet.field=%7B%21ex%3Duser_location%7Duser_location&f.
user_location.facet.limit=6&f.user_location.facet.mincount=0&facet.range=%7B%21ex%3Dus
er_statuses_count%7Duser_statuses_count&f.user_statuses_count.facet.range.start=0&f.use
r_statuses_count.facet.range.end=2500000&f.user_statuses_count.facet.range.gap=250000
&f.user_statuses_count.facet.mincount=0&facet.range=%7B%21ex%3Duser_followers_count
%7Duser_followers_count&f.user_followers_count.facet.range.start=0&f.user_followers_coun
t.facet.range.end=9000000&f.user_followers_count.facet.range.gap=900000&f.user_followers
_count.facet.mincount=0&fq=%7B%21tag%3Duser_location%7D%7B%21field+f%3Duser_lo
cation%7DIndonesia&fl=%2A&hl=true&hl.fl=%2A&hl.snippets=3&hl.fragsize=0 HTTP/1.1" 200
None
[10/Mar/2016 09:45:59 -0800] resource     DEBUG    GET Got response:
{"responseHeader":{"status":0,"Q...
```

Fig 32. Verifying logging information (Username and search query highlighted in yellow)

vii. Creating a database for storing user search history

The ideal method to keep track of user search history involves storing this information in a database. Hue comes with an installed MySQL database. Storing this information involves creating a tables to store this information in, and pushing this information into the database each time a user interacts with the interface. Instructions for creating the database tables are listed below:

1. Connect to the database
2. Locate database login information in the '/etc/hue/conf.empty/hei.ini' file
3. Scroll down to the database section and look for the following text:

```
[[database]]
    engine=mysql
    host=quickstart.cloudera
    port=3306
    user=hue
    password=cloudera
    name=hue
    # Database engine is typically one of:
    # postgresql_psycopg2, mysql, or sqlite3
```

Fig 33. Database connection settings in '/etc/hue/conf.empty/hei.ini'

4. Open a terminal window

5. Type the following commands to connect to the database:

```
'mysql -h quickstart.cloudera --port=3306 -u hue -p'
```

*Replace 'cloudera.quickstart' with your sql server hostname, '3306' with your port number and 'hue' with your username

```
[cloudera@quickstart ~]$ mysql -h quickstart.cloudera --port=3306 -u
hue -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.1.66 Source distribution

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```

Fig 34. Connecting to Hue MySQL database

6. Type 'show databases;' to list the databases.

```
mysql>
mysql>
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hue                |
+--------------------+
2 rows in set (0.00 sec)
```

Fig 35. List Hue databases

7. Connect to a database using the 'connect <database name>' command
8. Create a user table for storing user search history information

```
mysql> create table search_history (id int not null auto_increment,
user VarChar(25),  query varchar(500), time timestamp, primary key
(id));
Query OK, 0 rows affected (0.03 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

Fig 36. Creating the user table in the Hue MySQL database

9. Confirm that your table has been created by running ‘show tables;’ command

```
| oozie_pig                       |
| oozie_shell                     |
| oozie_sqoop                     |
| oozie_ssh                       |
| oozie_start                     |
| oozie_streaming                 |
| oozie_subworkflow               |
| oozie_workflow                  |
| pig_document                    |
| pig_pigscript                   |
| search_collection               |
| search_facet                    |
| search_history                  |
| search_result                   |
| search_sorting                  |
| south_migrationhistory          |
| useradmin_grouppermission       |
| useradmin_huepermission         |
| useradmin_ldapgroup             |
| useradmin_userprofile           |
+---------------------------------+
76 rows in set (0.00 sec)

mysql>
```

Fig 37. Confirming table creation

8. Acknowledgements

Many thanks go to NSF grant IIS - 1319578, III: Small: Integrated Digital Event Archiving and Library (IDEAL) for funding this project, and our professor Dr Edward Fox and the Teaching Assistants Sunshin Lee and Mohamed Magdy Gharib Farag for their guidance and assistance during the creation of this work. We would also like to thank our classmates in the various groups for their great collaborative efforts that included knowledge sharing, processing and providing the data that our front-end manipulates, proofreading the various drafts of this paper, and providing various other forms of feedback. This work would not have been possible without the contributions of everyone involved.

9. References

[1] Zhang Y. 2015. *Discrete Distribution Clustering In Big Data And A Method For Storm Prediction Leveraging Large Historical Archives*. The Pennsylvania State University. The Graduate School. College of Information Sciences and Technology

[2] CDH4 Documentation. (.n.d.)., Date accessed Feb. 9, 2016 http://www.cloudera.com/documentation/archive/cdh/4-x/4-2-0/Hue-2-User-Guide/hue2.html

[3] Apache Solr Reference Guide “Covering Apache Solr 5.5”, Date accessed Feb. 9, 2016

http://apache.arvixe.com/lucene/solr/ref-guide/apache-solr-ref-guide-5.5.pdf

[4] What Pig Does, Date accessed Feb. 14, 2016

http://hortonworks.com/hadoop/pig/

[5] Facets, Date accessed Feb. 14, 2016

https://www.elastic.co/guide/en/elasticsearch/reference/current/search-facets.html

[6] The Apache Software Foundation, Feb. 14, 2016

www.apache.org

10. Appendix

a. Attempts to import an index from another host

While learning Hue, we decided to run it on pre-packaged virtual machines downloaded from http://www.cloudera.com/downloads.html. We realized that the indexes that came with the virtual machines were different from the indexes that we would be using in the final project. This posed a challenge because the fields that we would be using to design the interface were not accessible on the virtual machines. After discussing this with the GRA Sunshin Lee, he provided us with an index from the cluster that we would eventually run our interface on along with instructions for importing this index onto our virtual machines, in the hope that it would have the same fields that we would use in the final implementation. This would allow us to build the interface on our virtual machines and port it over to the cluster when the cluster had been populated with the data provided by all project groups. However, the import was challenging. The instructions we used and the steps we attempted are listed below:

a. We followed the original instructions (below), but they would not work

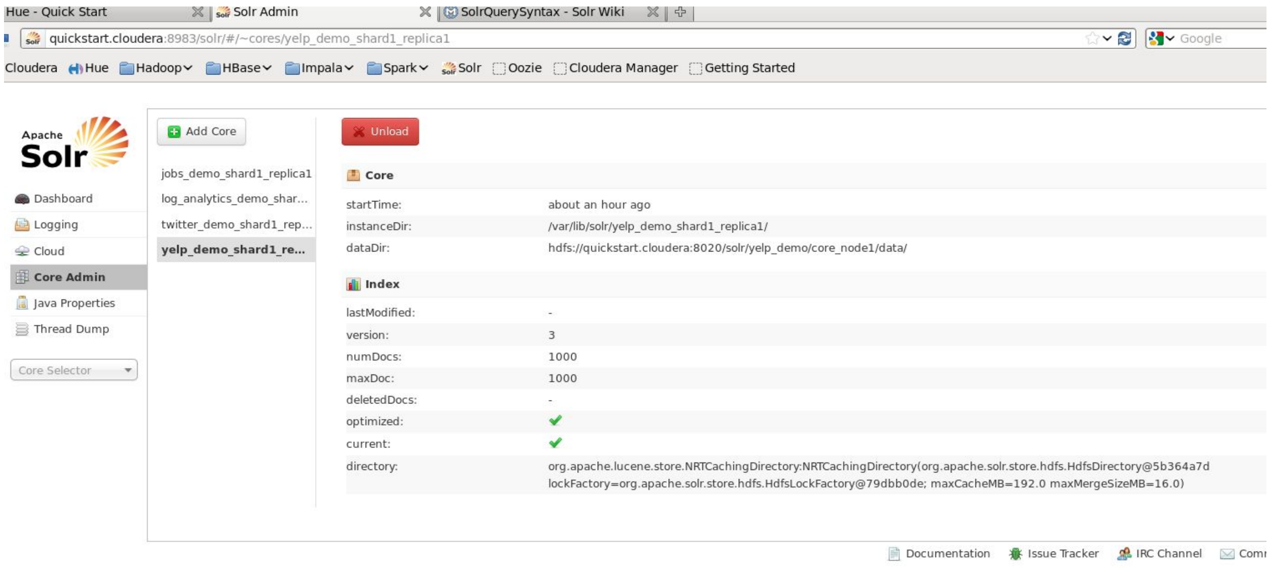
```
// Decide collection name
[sslee777@node1 ~]$ export IDX_SOLR_DIR="index_solr"
[sslee777@node1 ~]$ export COLLECTION="ideal-tweet-70"
[sslee777@node1 ~]$ cd $HOME/$IDX_SOLR_DIR/

// 1. Create Solr instancedir in local
[sslee777@node1 index_solr]$ solrctl instancedir --generate
$HOME/$IDX_SOLR_DIR/$COLLECTION
// verify
[sslee777@node1 index_solr]$ ls -F
ideal_collections_tweets/     ideal-tweet-2/    index.sh*
ideal_collections_tweets.sh*  ideal-tweet-3/    old/
ideal_col_tw/                 ideal-tweet-50/   teamplate_solr-team/
ideal_col_tw.sh*              ideal-tweet-70/  test/

// 2. Copy(replace) schema.xml in $COLLECTION(instancedir) with
pre-defined schema (Attached in mail)
// 2.1. Copy pre-defined schema file: schema.xml
[sslee777@node1 index_solr]$ cp schema.xml
$HOME/$COLLECTION/conf/schema.xml


// 3. Create instancedir in zookeeper
// Copy those configuration files(instancedir) into zookeeper
[sslee777@node1 index_solr]$ solrctl instancedir --create $COLLECTION
$HOME/$IDX_SOLR_DIR/$COLLECTION
```

```
Uploading configs from /home/sslee777/ideal_collections_tweets/conf to
solr2.dlrl:2181,node2.dlrl:2181,node3.dlrl:2181,node1.dlrl:2181,node4.dlrl
:2181/solr. This may take up to a minute.

// Verify
[sslee777@node1 index_solr]$ solrctl instancedir --list
ideal_col_tw
ideal-tweet-3
ideal-tweet-70

// (Optional) Update if you modified schema or anything in instancedir
[sslee777@node1 index_solr]$ solrctl instancedir --update $COLLECTION
$HOME/$IDX_SOLR_DIR/$COLLECTION

// 4. Create a collection under /Solr directory in HDFS and we can see it
in Solr Admin
[sslee777@node1 index_solr]$ solrctl collection --create $COLLECTION

// Verify
[sslee777@node1 index_solr]$ solrctl collection --list
webpages (2)
ideal-tweet-70 (2)

// (Optional) Update
[sslee777@node1 index_solr]$ solrctl collection --reload $COLLECTION

// (Optional) delete (collection -> instancedir -> local directory)
[sslee777@node1 index_solr]$ solrctl collection --delete $COLLECTION
[sslee777@node1 index_solr]$ solrctl instancedir --delete $COLLECTION
[sslee777@node1 index_solr]$ rm -rf $COLLECTION

// 5. Unzip index.tgz(Attached in mail) and Upload Idx into HDFS and
restart Solr
[sslee777@node1 index_solr]$ sudo -u solr hadoop fs -put index
/solr/$COLLECTION/core_node1/data/
```

b. We manually created required folders
   The third instruction '`[sslee777@node1 ~]$ cd $HOME/$IDX_SOLR_DIR/`' under //Cecide collection name failed because the folder '`$IDX_SOLR_DIR`' needed to be created. We ran the instruction below to to create the folder:
   '`[cloudera@quickstart ~]$ mkdir $HOME/$IDX_SOLR_DIR/`'

c. When we tried to verify that all the commands had run successfully, files were missing. Running the '//very' instruction (instruction 3 above) failed because of a missing instance directory in zookeeper

d. We tried to manually create an instance within Solr hoping this would fix the problem



quickstart.cloudera:8983/solr/#/~cores/yelp_demo_shard1_replica1

jobs_demo_shard1_replica1
log_analytics_demo_shar...
twitter_demo_shard1_rep...
yelp_demo_shard1_re...

Core

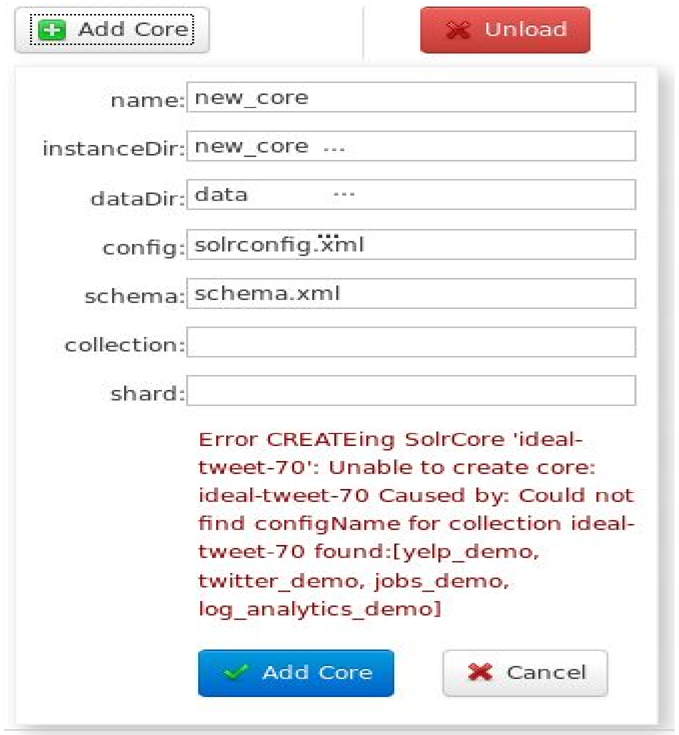
startTime:	about an hour ago
instanceDir:	/var/lib/solr/yelp_demo_shard1_replica1/
dataDir:	hdfs://quickstart.cloudera:8020/solr/yelp_demo/core_node1/data/

Index

lastModified:	-
version:	3
numDocs:	1000
maxDoc:	1000
deletedDocs:	-
optimized:	✓
current:	✓
directory:	org.apache.lucene.store.NRTCachingDirectory:NRTCachingDirectory(org.apache.solr.store.hdfs.HdfsDirectory@5b364a7d lockFactory=org.apache.solr.store.hdfs.HdfsLockFactory@79dbb0de; maxCacheMB=192.0 maxMergeSizeMB=16.0)

Fig. I. Solar error after opening Solr in the virtual machine

e. When we went into Solr and tried to create a new core since all the other collections had cores except for the ‘ideal-tweet-70’ collection whose index we were trying to import. Solr gave the following error: “Error CREATEing SolrCore Unable to create core Could not find configName”

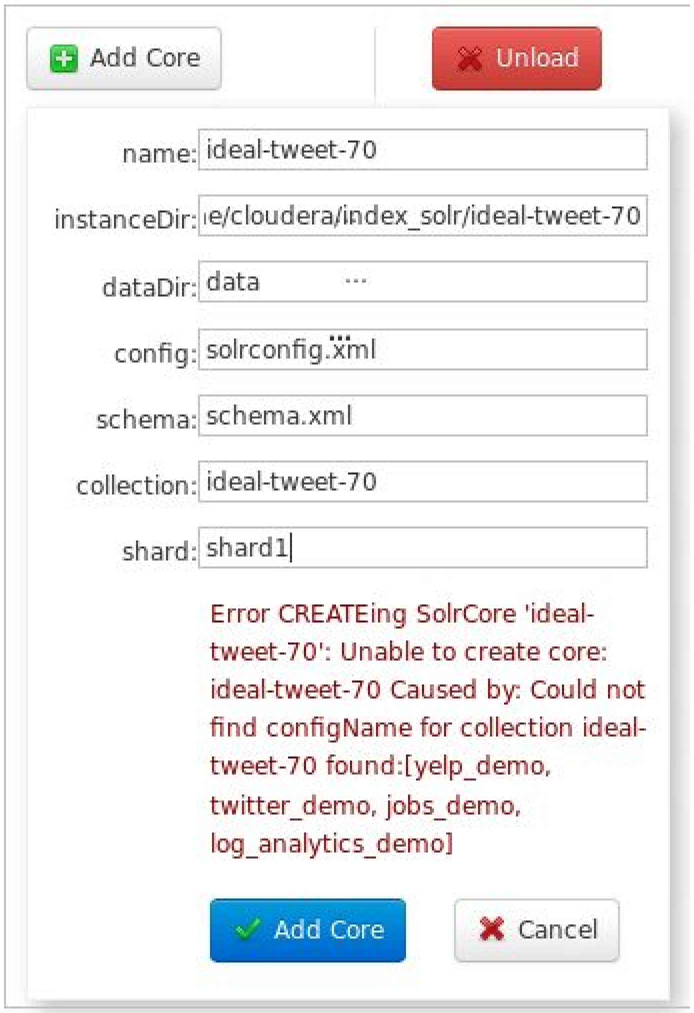


name:	new_core
instanceDir:	new_core
dataDir:	data
config:	solrconfig.xml
schema:	schema.xml
collection:	
shard:	

Error CREATEing SolrCore 'ideal-tweet-70': Unable to create core: ideal-tweet-70 Caused by: Could not find configName for collection ideal-tweet-70 found:[yelp_demo, twitter_demo, jobs_demo, log_analytics_demo]

Fig. II. Solr error when creating a core

f. We used whereis command to find solr files. Found that all collections have core.properties file in cd /var/lib/solr/ folder. We copied an existing folder for a pre-existing collection and used chmod command to change it's permissions. We also modified the file to create a 'core.properties' file for the new index, and went back into solr to create a new instance from Add Core interface

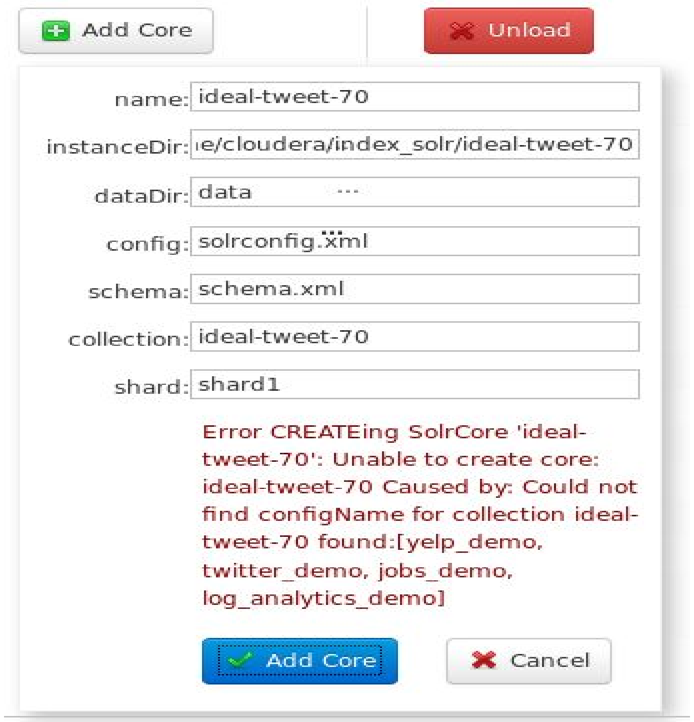


name: ideal-tweet-70
instanceDir: ıe/cloudera/index_solr/ideal-tweet-70
dataDir: data
config: solrconfig.xml
schema: schema.xml
collection: ideal-tweet-70
shard: shard1

```
Error CREATEing SolrCore 'ideal-tweet-70': Unable to create core: ideal-tweet-70 Caused by: Could not find configName for collection ideal-tweet-70 found:[yelp_demo, twitter_demo, jobs_demo, log_analytics_demo]
```

Fig. III. Solr error when creating an instance

g. We got the same error message.

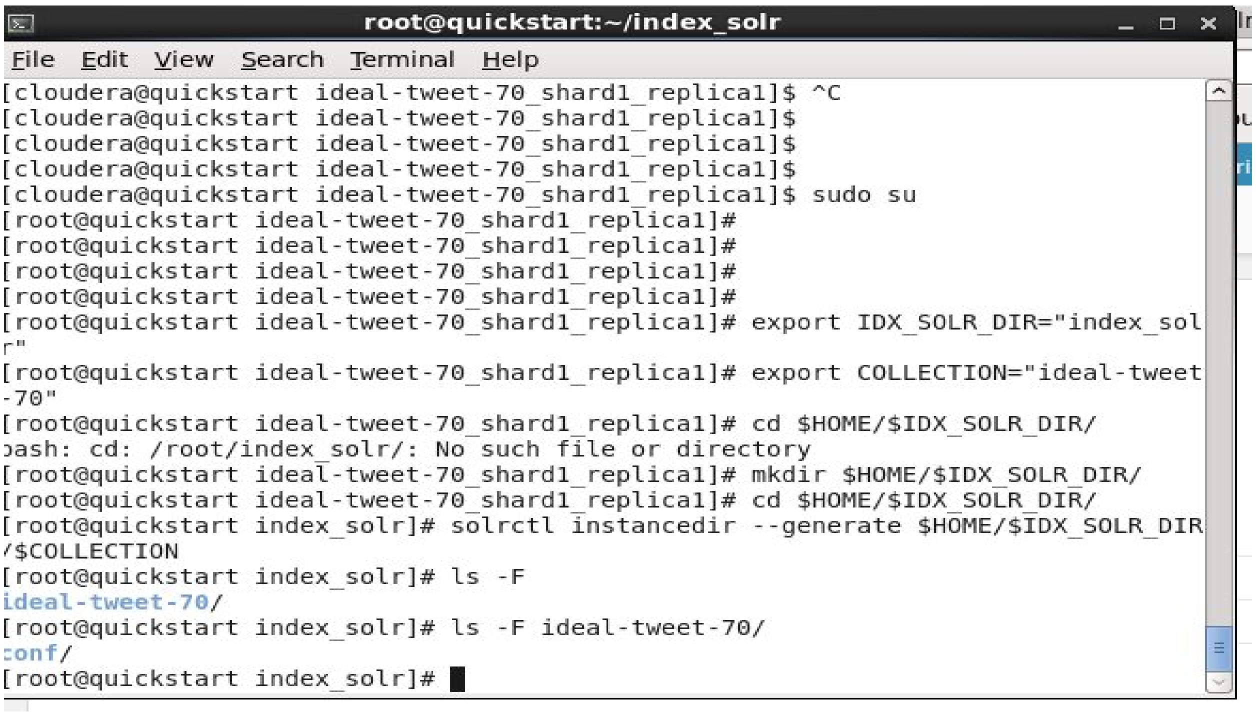


name: ideal-tweet-70
instanceDir: ıe/cloudera/index_solr/ideal-tweet-70
dataDir: data
config: solrconfig.xml
schema: schema.xml
collection: ideal-tweet-70
shard: shard1

Error CREATEing SolrCore 'ideal-tweet-70': Unable to create core: ideal-tweet-70 Caused by: Could not find configName for collection ideal-tweet-70 found:[yelp_demo, twitter_demo, jobs_demo, log_analytics_demo]

Fig. IV. Solr error on second attempt to create core

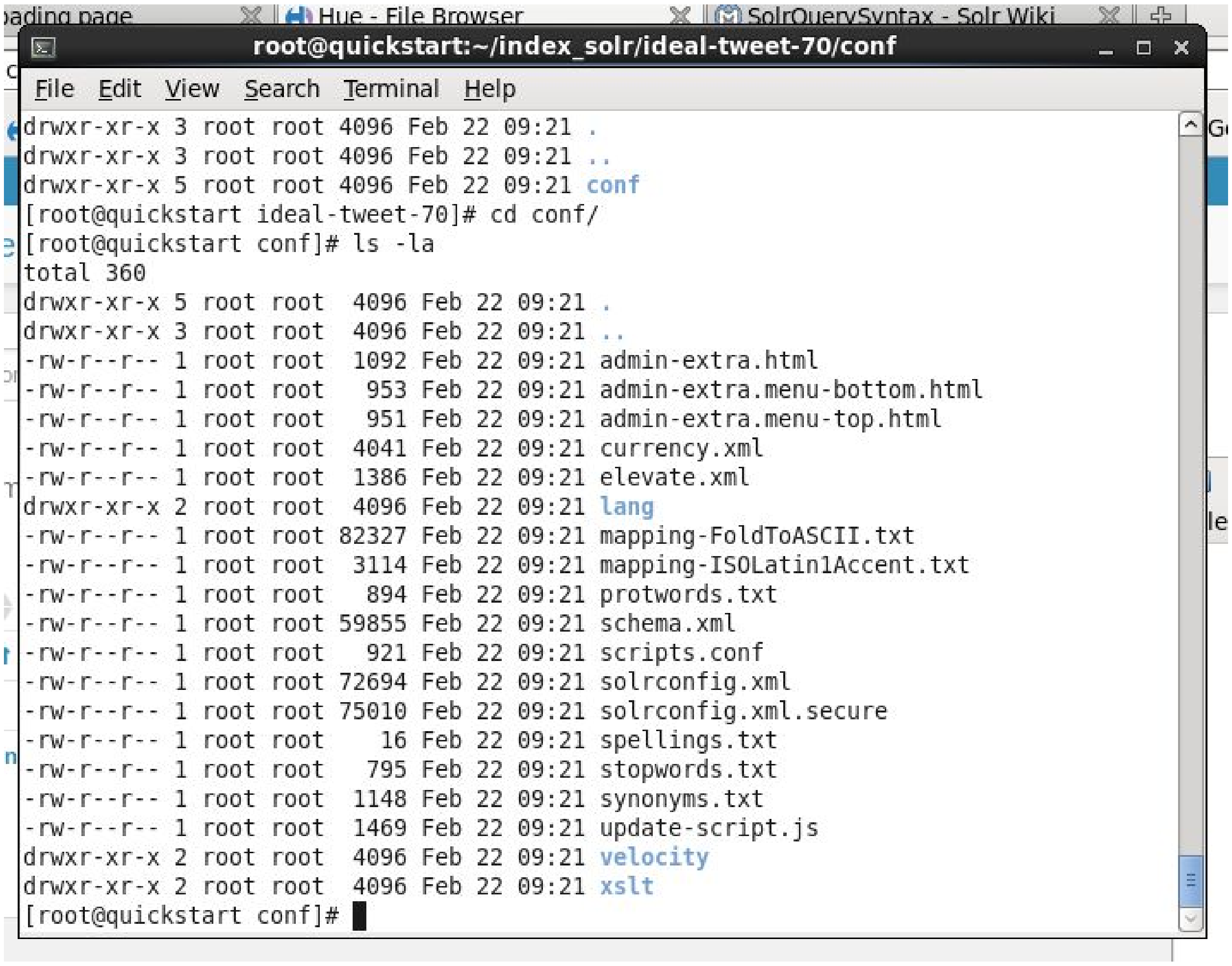
h. We started over and tried to run initial instructions as root, but files we expected to see were missing.


root@quickstart:~/index_solr
File Edit View Search Terminal Help

```
[cloudera@quickstart ideal-tweet-70_shard1_replica1]$ ^C
[cloudera@quickstart ideal-tweet-70_shard1_replica1]$
[cloudera@quickstart ideal-tweet-70_shard1_replica1]$
[cloudera@quickstart ideal-tweet-70_shard1_replica1]$
[cloudera@quickstart ideal-tweet-70_shard1_replica1]$ sudo su
[root@quickstart ideal-tweet-70_shard1_replica1]#
[root@quickstart ideal-tweet-70_shard1_replica1]#
[root@quickstart ideal-tweet-70_shard1_replica1]#
[root@quickstart ideal-tweet-70_shard1_replica1]#
[root@quickstart ideal-tweet-70_shard1_replica1]# export IDX_SOLR_DIR="index_sol
r"
[root@quickstart ideal-tweet-70_shard1_replica1]# export COLLECTION="ideal-tweet
-70"
[root@quickstart ideal-tweet-70_shard1_replica1]# cd $HOME/$IDX_SOLR_DIR/
bash: cd: /root/index_solr/: No such file or directory
[root@quickstart ideal-tweet-70_shard1_replica1]# mkdir $HOME/$IDX_SOLR_DIR/
[root@quickstart ideal-tweet-70_shard1_replica1]# cd $HOME/$IDX_SOLR_DIR/
[root@quickstart index_solr]# solrctl instancedir --generate $HOME/$IDX_SOLR_DIR
/$COLLECTION
[root@quickstart index_solr]# ls -F
ideal-tweet-70/
[root@quickstart index_solr]# ls -F ideal-tweet-70/
conf/
[root@quickstart index_solr]# 
```

Fig. V. Some expected files were not created

i. We also decided to check inside the conf folder, and did not find the files we were looking for.

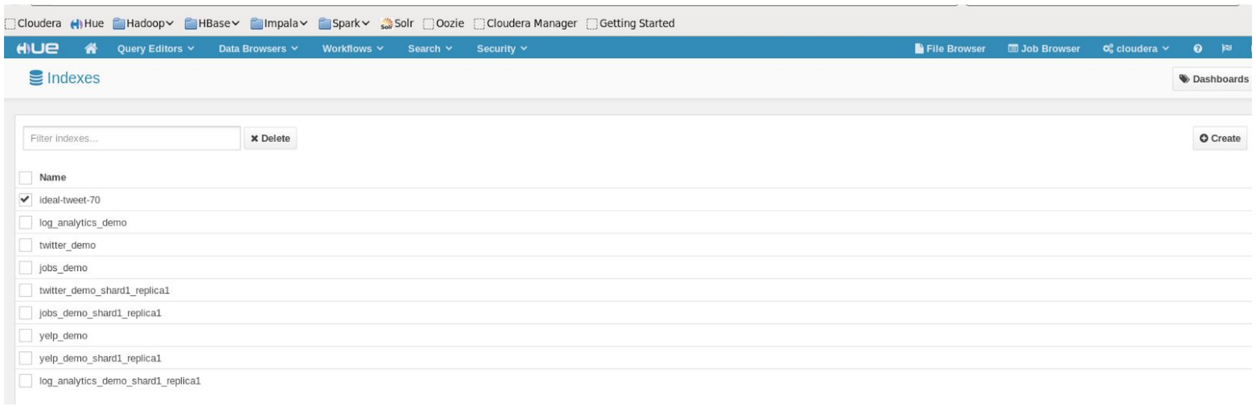


root@quickstart:~/index_solr/ideal-tweet-70/conf
File Edit View Search Terminal Help

```
drwxr-xr-x 3 root root 4096 Feb 22 09:21 .
drwxr-xr-x 3 root root 4096 Feb 22 09:21 ..
drwxr-xr-x 5 root root 4096 Feb 22 09:21 conf
[root@quickstart ideal-tweet-70]# cd conf/
[root@quickstart conf]# ls -la
total 360
drwxr-xr-x 5 root root  4096 Feb 22 09:21 .
drwxr-xr-x 3 root root  4096 Feb 22 09:21 ..
-rw-r--r-- 1 root root  1092 Feb 22 09:21 admin-extra.html
-rw-r--r-- 1 root root   953 Feb 22 09:21 admin-extra.menu-bottom.html
-rw-r--r-- 1 root root   951 Feb 22 09:21 admin-extra.menu-top.html
-rw-r--r-- 1 root root  4041 Feb 22 09:21 currency.xml
-rw-r--r-- 1 root root  1386 Feb 22 09:21 elevate.xml
drwxr-xr-x 2 root root  4096 Feb 22 09:21 lang
-rw-r--r-- 1 root root 82327 Feb 22 09:21 mapping-FoldToASCII.txt
-rw-r--r-- 1 root root  3114 Feb 22 09:21 mapping-ISOLatin1Accent.txt
-rw-r--r-- 1 root root   894 Feb 22 09:21 protwords.txt
-rw-r--r-- 1 root root 59855 Feb 22 09:21 schema.xml
-rw-r--r-- 1 root root   921 Feb 22 09:21 scripts.conf
-rw-r--r-- 1 root root 72694 Feb 22 09:21 solrconfig.xml
-rw-r--r-- 1 root root 75010 Feb 22 09:21 solrconfig.xml.secure
-rw-r--r-- 1 root root    16 Feb 22 09:21 spellings.txt
-rw-r--r-- 1 root root   795 Feb 22 09:21 stopwords.txt
-rw-r--r-- 1 root root  1148 Feb 22 09:21 synonyms.txt
-rw-r--r-- 1 root root  1469 Feb 22 09:21 update-script.js
drwxr-xr-x 2 root root  4096 Feb 22 09:21 velocity
drwxr-xr-x 2 root root  4096 Feb 22 09:21 xslt
[root@quickstart conf]#
```

Fig. VI. Searching for missing files in the Hue ‘conf’ folder

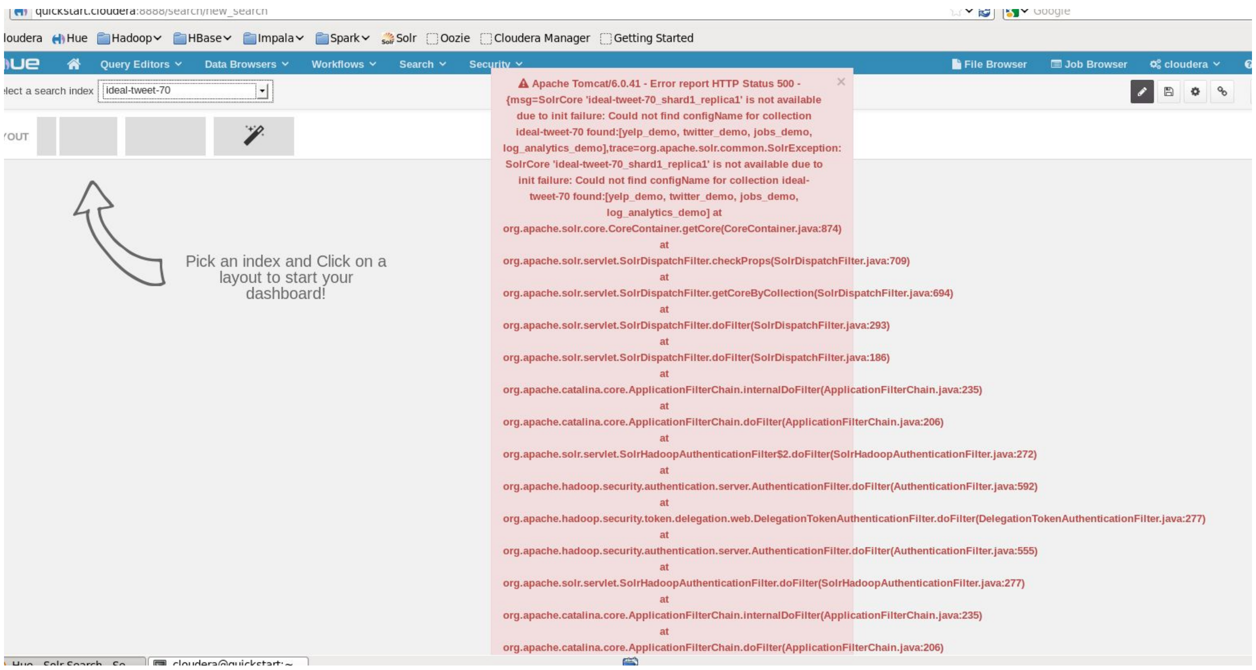
j. We decided to postpone trying to import an index and focused on creating a Hue interface using the example twitter data that came with the VM. We restarted the virtual machine proceeded to add 2 users ‘johnd’ and ‘janed’ in Hue using the steps described in the Developer Manual section above. After creating users we proceeded to create a search interface using the steps described in the Developer Manual section above. When we clicked on search->Indexes, we saw the index listed in Hue (maybe a restart was required).



Name
- [✓] ideal-tweet-70
- log_analytics_demo
- twitter_demo
- jobs_demo
- twitter_demo_shard1_replica1
- jobs_demo_shard1_replica1
- yelp_demo
- yelp_demo_shard1_replica1
- log_analytics_demo_shard1_replica1

Fig. VII. Hue list of indexes

k. We proceeded to create a dashboard using the instructions described in the Developer Manual section above, but when we selected the 'ideal-tweet-70' index in the 'Select a search index' drop down in the top-left corner, we got the following error message:



```
Apache Tomcat/6.0.41 - Error report HTTP Status 500 -
{msg=SolrCore 'ideal-tweet-70_shard1_replica1' is not available
due to init failure: Could not find configName for collection
ideal-tweet-70 found:[yelp_demo, twitter_demo, jobs_demo,
log_analytics_demo],trace=org.apache.solr.common.SolrException:
SolrCore 'ideal-tweet-70_shard1_replica1' is not available due to
init failure: Could not find configName for collection ideal-
tweet-70 found:[yelp_demo, twitter_demo, jobs_demo,
log_analytics_demo] at
org.apache.solr.core.CoreContainer.getCore(CoreContainer.java:874)
at
org.apache.solr.servlet.SolrDispatchFilter.checkProps(SolrDispatchFilter.java:709)
at
org.apache.solr.servlet.SolrDispatchFilter.getCoreByCollection(SolrDispatchFilter.java:694)
at
org.apache.solr.servlet.SolrDispatchFilter.doFilter(SolrDispatchFilter.java:293)
at
org.apache.solr.servlet.SolrDispatchFilter.doFilter(SolrDispatchFilter.java:186)
at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at
org.apache.solr.servlet.SolrHadoopAuthenticationFilter$2.doFilter(SolrHadoopAuthenticationFilter.java:272)
at
org.apache.hadoop.security.authentication.server.AuthenticationFilter.doFilter(AuthenticationFilter.java:592)
at
org.apache.hadoop.security.token.delegation.web.DelegationTokenAuthenticationFilter.doFilter(DelegationTokenAuthenticationFilter.java:277)
at
org.apache.hadoop.security.authentication.server.AuthenticationFilter.doFilter(AuthenticationFilter.java:555)
at
org.apache.solr.servlet.SolrHadoopAuthenticationFilter.doFilter(SolrHadoopAuthenticationFilter.java:277)
at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
```

Fig. VIII. Hue error message when creating dashboard

l. We needed the actual collection in Solr in addition to the imported index. It was time to revert to using one of the demo indexes that came with the demo virtual machine. We decided to use the twitter-demo index

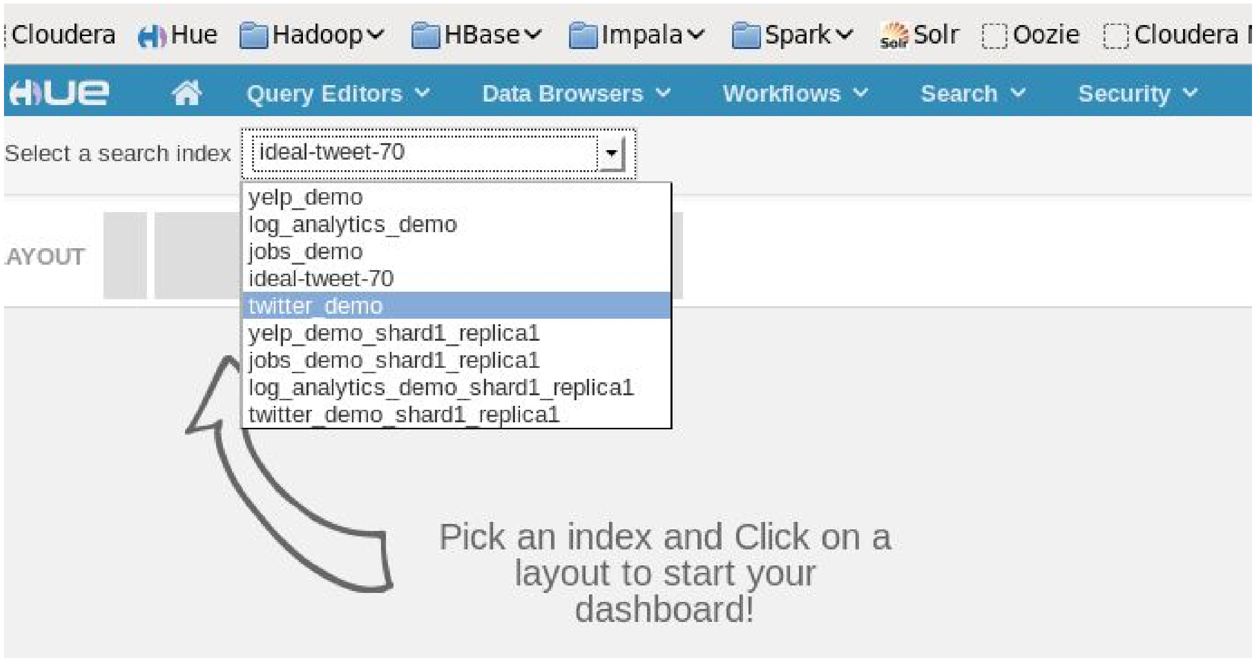


Fig. IX. New index available after restart

b. Attempts to edit Hue web page

The HTML page that Hue runs in is created using Django. Django provides a web server and the ability to create web pages using python. Unlike web servers like Apache, where the web pages reside in 'foo/www' directory, Django pages can live in any folder. The pages also exist in the format of python files as opposed to HTML. Django web pages are known as apps and each app resides in a folder that contains templates amongst other files. Templates contain python scripts that tell the web server what to return when it receives a request like an 'HttpRequest' for example. This means that a HTML file will not be found on the host running the web server, but one or more templates that contain some HTML syntax and these templates have the .mako extension.

To edit edit the Hue search page, follow the instructions below:

1. Navigate to the search app template by opening a terminal window and running the following command:
   'cd /usr/lib/hue/apps/search/src/search/templates/'

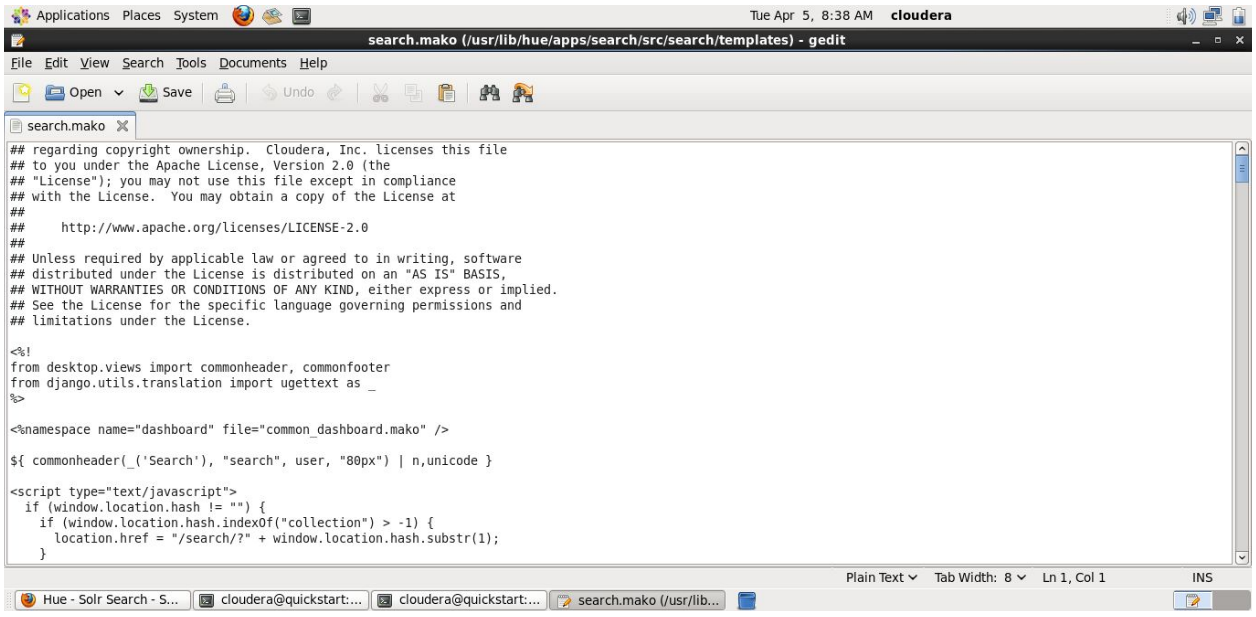
2. Verify that the current folder contains the search.mako template by running the following command:
   'ls'

```
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ cd
/usr/lib/hue/apps/search/src/search/templates/
[cloudera@quickstart templates]$ ls
admin_collections.mako  macros.mako  no_collections.mako
search.mako
[cloudera@quickstart templates]$
```

Fig. X. Locating the search app template.

3. Use a text editor to edit the template. For this example we use the ‘gedit’ editor. Open the file with elevated permissions. In linux, use the sudo command as shown below: ‘sudo gedit search.mako’



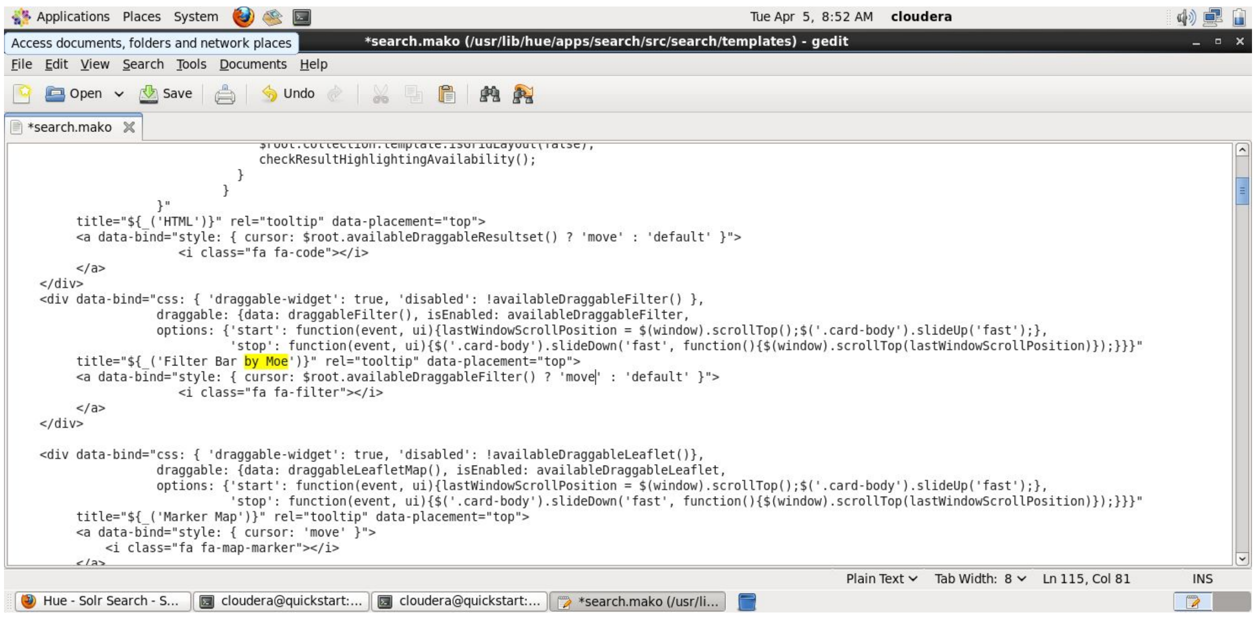
Applications Places System
Tue Apr 5, 8:38 AM cloudera
search.mako (/usr/lib/hue/apps/search/src/search/templates) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
search.mako
Plain Text Tab Width: 8 Ln 1, Col 1 INS
Hue - Solr Search - S... cloudera@quickstart:... cloudera@quickstart:... search.mako (/usr/lib...

```
## regarding copyright ownership.  Cloudera, Inc. licenses this file
## to you under the Apache License, Version 2.0 (the
## "License"); you may not use this file except in compliance
## with the License.  You may obtain a copy of the License at
##
##     http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.

<%!
from desktop.views import commonheader, commonfooter
from django.utils.translation import ugettext as _
%>

<%namespace name="dashboard" file="common_dashboard.mako" />

${ commonheader(_('Search'), "search", user, "80px") | n,unicode }

<script type="text/javascript">
  if (window.location.hash != "") {
    if (window.location.hash.indexOf("collection") > -1) {
      location.href = "/search/?" + window.location.hash.substr(1);
    }
```

Fig XI. Search.mako template open in gedit

4. In this example, we will look for the Tooltip associated with the filter button and append text to it. Use the find command to look for the text “Filter” and then locate the string shown below:
5. title="${_('Filter Bar')}" rel="tooltip" data-placement="top">

6. Change 'Filter Bar' to 'Filter Bar by Moe'
7. Save the changes and reload the search page.
8. Click on the Pencil icon to edit the search page
9. Point your mouse at the Filter icon and verify that the tooltip message has changed


Applications Places System
Tue Apr 5, 8:52 AM cloudera
Access documents, folders and network places
*search.mako (/usr/lib/hue/apps/search/src/search/templates) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
*search.mako
Plain Text Tab Width: 8 Ln 115, Col 81 INS
Hue - Solr Search - S... cloudera@quickstart:... cloudera@quickstart:... *search.mako (/usr/li...

```
                              $root.collection.template.isGridLayout(false);
                              checkResultHighlightingAvailability();
                         }
                       }
                }"
        title="${_('HTML')}" rel="tooltip" data-placement="top">
        <a data-bind="style: { cursor: $root.availableDraggableResultset() ? 'move' : 'default' }">
                      <i class="fa fa-code"></i>
        </a>
   </div>
   <div data-bind="css: { 'draggable-widget': true, 'disabled': !availableDraggableFilter() },
                draggable: {data: draggableFilter(), isEnabled: availableDraggableFilter,
                options: {'start': function(event, ui){lastWindowScrollPosition = $(window).scrollTop();$('.card-body').slideUp('fast');},
                          'stop': function(event, ui){$('.card-body').slideDown('fast', function(){$(window).scrollTop(lastWindowScrollPosition)});}}}"
        title="${_('Filter Bar by Moe')}" rel="tooltip" data-placement="top">
        <a data-bind="style: { cursor: $root.availableDraggableFilter() ? 'move' : 'default' }">
                      <i class="fa fa-filter"></i>
        </a>
   </div>

   <div data-bind="css: { 'draggable-widget': true, 'disabled': !availableDraggableLeaflet()},
                draggable: {data: draggableLeafletMap(), isEnabled: availableDraggableLeaflet,
                options: {'start': function(event, ui){lastWindowScrollPosition = $(window).scrollTop();$('.card-body').slideUp('fast');},
                          'stop': function(event, ui){$('.card-body').slideDown('fast', function(){$(window).scrollTop(lastWindowScrollPosition)});}}}"
        title="${_('Marker Map')}" rel="tooltip" data-placement="top">
        <a data-bind="style: { cursor: 'move' }">
           <i class="fa fa-map-marker"></i>
        </a>
```

Fig XII. Adding text ‘by Moe’ to the ‘search.mako’ template.

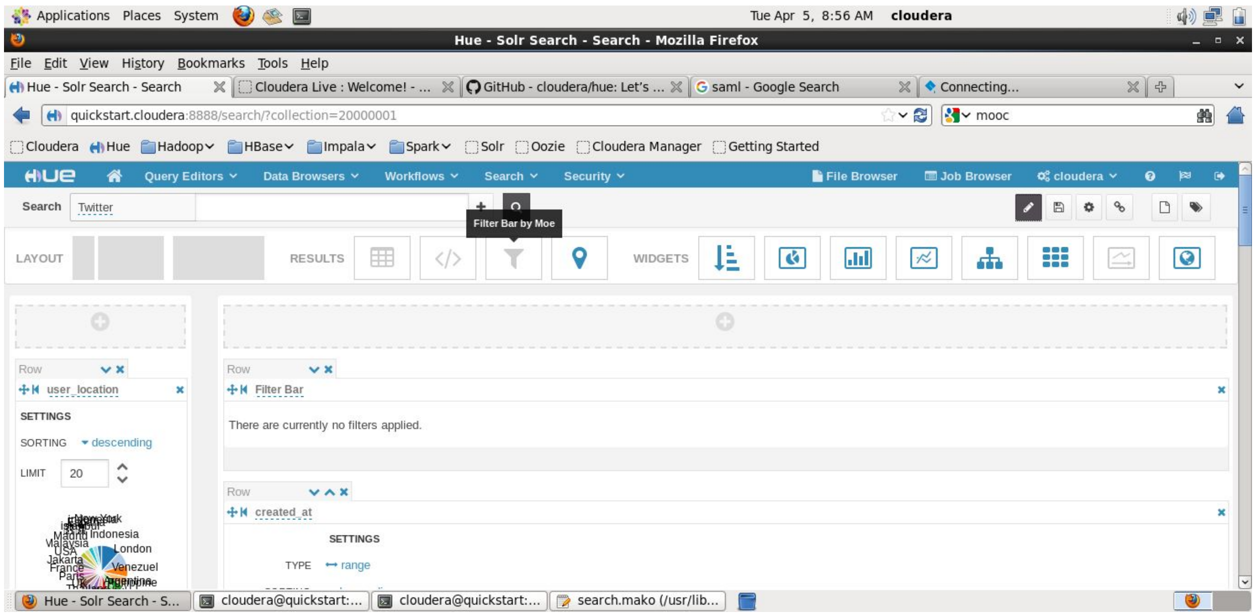

Applications Places System
Tue Apr 5, 8:56 AM cloudera
Hue - Solr Search - Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hue - Solr Search - Search
Cloudera Live : Welcome! - ...
GitHub - cloudera/hue: Let's ...
saml - Google Search
Connecting...
quickstart.cloudera:8888/search/?collection=20000001
mooc
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
HUE Query Editors Data Browsers Workflows Search Security File Browser Job Browser cloudera
Search Twitter
Filter Bar by Moe
LAYOUT RESULTS WIDGETS
Row
user_location
SETTINGS
SORTING descending
LIMIT 20
Row
Filter Bar
There are currently no filters applied.
Row
created_at
SETTINGS
TYPE range
Hue - Solr Search - S... cloudera@quickstart:... cloudera@quickstart:... search.mako (/usr/lib...

Fig XII. Hue search app now showing the text we added in the Filter tooltip

*This method can be used to change any text that shows up on an app, but use caution when making any changes.

c. Attempts to locate Hue search activity logs

We manually searched all log files in the virtual machine trying to find out if Hue logs user activity and if so, where it does this. Hue uses conventional linux logging in the ‘’var/log’ directory. Hue online documentation states the following regarding Hue logging: “In addition to logging INFO level messages to the logs directory, the Hue web server keeps a small buffer of log messages at all levels in memory. You can view these logs by visiting http://myserver:8888/logs. The DEBUG level messages shown can sometimes be helpful in troubleshooting issues”. We created a query logged in as ‘johnd’ and searched all loges for information relating to **johnd**’s activity.

```
[09/Mar/2016 18:42:08 -0800] connectionpool DEBUG   "GET
/solr/twitter_demo/select?user.name=hue&doAs=johnd&q=%2A%3A%2A&wt=json&rows=10&start
=0&facet=true&facet.mincount=0&facet.limit=10&facet.range=%7B%21ex%3Dcreated_at%7Dcreat
ed_at&f.created_at.facet.range.start=2014-02-25T16%3A00%3A00Z&f.created_at.facet.range.end=
2014-02-26T04%3A00%3A00Z&f.created_at.facet.range.gap=%2B10MINUTES&f.created_at.facet.mi
ncount=0&facet.field=%7B%21ex%3Duser_location%7Duser_location&f.user_location.facet.limit=
6&f.user_location.facet.mincount=0&facet.range=%7B%21ex%3Duser_statuses_count%7Duser_st
atuses_count&f.user_statuses_count.facet.range.start=0&f.user_statuses_count.facet.range.end=2
500000&f.user_statuses_count.facet.range.gap=250000&f.user_statuses_count.facet.mincount=0&f
acet.range=%7B%21ex%3Duser_followers_count%7Duser_followers_count&f.user_followers_cou
nt.facet.range.start=0&f.user_followers_count.facet.range.end=9000000&f.user_followers_count.fa
cet.range.gap=900000&f.user_followers_count.facet.mincount=0&fq=%7B%21tag%3Duser_statuse
s_count%7Duser_statuses_count%3A%5B0+TO+250000%7D&fl=%2A&hl=true&hl.fl=%2A&hl.snippe
ts=3&hl.fragsize=0 HTTP/1.1" 200 None
```

Fig.XIV. Snippet of logs from ‘http://myserver:8888/logs’

The ‘/var/log/runcpserver.log’ is where Hue logs all user activity including free text and facet searches. The logging information level can be altered by following instructions described in the ‘Capturing User Search History’ section of the ‘Developer Manual’ section above.

```
[01/Mar/2016 19:46:59 -0800] access       INFO     127.0.0.1 cloudera
- "GET /jobbrowser/ HTTP/1.1"
[01/Mar/2016 19:47:01 -0800] access       INFO     127.0.0.1 cloudera
- "GET /useradmin/users/edit/johnd HTTP/1.1"
[01/Mar/2016 19:47:02 -0800] access       INFO     127.0.0.1 cloudera
- "GET /jobbrowser/ HTTP/1.1"
[01/Mar/2016 19:47:15 -0800] access       INFO     127.0.0.1 cloudera
- "GET /accounts/logout/ HTTP/1.1"
```

```
...
[01/Mar/2016 19:50:51 -0800] access       WARNING  127.0.0.1 johnd -
"POST /accounts/login/ HTTP/1.1" -- "johnd" login ok
[01/Mar/2016 19:50:51 -0800] access       INFO     127.0.0.1 johnd -
"GET /search/ HTTP/1.1"
[01/Mar/2016 19:50:52 -0800] access       INFO     127.0.0.1 johnd -
"POST /search/get_collections HTTP/1.1"
[01/Mar/2016 19:50:52 -0800] connectionpool INFO     Starting new
HTTP connection (1): quickstart.cloudera
[01/Mar/2016 19:50:52 -0800] access       INFO     127.0.0.1 johnd -
"POST /search/get_collection HTTP/1.1"
[01/Mar/2016 19:50:52 -0800] connectionpool INFO     Starting new
HTTP connection (1): quickstart.cloudera
[01/Mar/2016 19:50:52 -0800] connectionpool INFO     Starting new
HTTP connection (1): quickstart.cloudera
[01/Mar/2016 19:50:52 -0800] access       INFO     127.0.0.1 johnd -
"POST /search/search HTTP/1.1"
```

Fig. XV. Snippet of logs from '/var/log/runcpserver.log'

The '/var/log/access.log' is used to log all queries issued against Hue's web server, however it does not log web queries to other web servers that result from clicking on links within returned search results.

```
01/Mar/2016 19:48:06 -0800] DEBUG    127.0.0.1 -anon- - "GET
/static/ext/fonts/fontawesome-webfont.woff HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 -anon- - "POST
/accounts/login/ HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] WARNING  127.0.0.1 johnd - "POST
/accounts/login/ HTTP/1.1" -- "johnd" login ok
[01/Mar/2016 19:50:51 -0800] INFO     127.0.0.1 johnd - "GET /search/
HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/ext/css/font-awesome.min.css HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/css/hue3.css HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/ext/css/bootplus.css HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/ext/css/fileuploader.css HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/js/hue.utils.js HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/ext/js/jquery/jquery-2.1.1.min.js HTTP/1.1"
```

```
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/js/jquery.selector.js HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/js/jquery.rowselector.js HTTP/1.1"
[01/Mar/2016 19:50:51 -0800] DEBUG    127.0.0.1 johnd - "GET
/static/js/jquery.delayedinput.js HTTP/1.1"
```

Fig. XVI. Snippet of logs from '/var/log/access.log'

Rest of the logs examined that had no johnd information in '/var/log/' folder are listed below: *Hive, Hive-catalog, haddop-hdfs-namenode-quicksart.cloudera.log, Impala, llama, ntpstats, Oozie, ppp, prelink, sa, samba, solr, spark, sqoop, sqoop2, zookeeper, boot, mapreduce, sentry, hadoop-httpfs, hadoop-mapreduce, hadoop-yarn, hbase, hbase-solr logging*

We also tried querying the database, but it seemed empty. Hue online documentation states "Hue requires a SQL database to store small amounts of data, including user account information as well as history of job submissions and Hive queries. By default, Hue is configured to use the embedded database SQLite for this purpose, and should require no configuration or management by the administrator. However, MySQL is the recommended database to use; this section contains instructions for configuring Hue to access MySQL and other databases." This does not work. See queries below:

```
[cloudera@quickstart ~]$ sqlite3 /var/lib/hue/desktop.db
SQLite version 3.6.20
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .tables
auth_group                      oozie_dataset
auth_group_permissions          oozie_decision
auth_permission                 oozie_decisionend
auth_user                       oozie_distcp
auth_user_groups                oozie_email
auth_user_user_permissions      oozie_end
beeswax_metainstall             oozie_fork
beeswax_queryhistory            oozie_fs
...
jobsub_jobdesign                pig_document
jobsub_jobhistory               pig_pigscript
jobsub_oozieaction              search_collection
jobsub_ooziedesign              search_facet
jobsub_ooziejavaaction          search_result
jobsub_ooziemapreduceaction     search_sorting
jobsub_ooziestreamingaction     south_migrationhistory
oozie_bundle                    useradmin_grouppermission
```

```
oozie_bundledcoordinator        useradmin_huepermission
oozie_coordinator               useradmin_ldapgroup
oozie_datainput                 useradmin_userprofile
oozie_dataoutput
sqlite> select * from useradmin_userprofile;
sqlite> select * from search_facet;
sqlite> select * from search_result;
sqlite> select * from search_collection;
sqlite> select * from search_sorting;
```

Fig. XVII. Sample queries and output from querying the Hue database

We decided to change logging level and use the '/var/log/runcpserver.log' to capture user search activity. Another more reliable method we would try would be creating a new table in the database and try to store query information in this location in lieu of using log files. This is currently a work in progress.

D. Attempts to Expand Hue Search Page Functionality

In order to modify Hue and give it added functionality e.g. a section for listing suggestions, we need to modify the web page that the Hue interface is displayed on. The goal was to create a second HTML widget that would hold suggested tweets and web pages that system thought would interest the current system user. Hue only allows a single HTML widget to be displayed at a time and that is why we had to try to hack the source code. The steps below show our failed attempt to do this. During our initial attempt to hack the source code, our virtual machine crashed and we lost the documentation of the steps we followed. Below is a high level reenactment of the steps we followed to hack the source code:

1. Navigate to '/usr/lib/hue/apps/search/src/search/templates'
2. Inspect 'search.mako', which is the template for the search interface.
3. Look for the <%dashboard:layout_toolbar> section
4. Locate the <div> containing the "HTML' widget.
5. Create an additional identical widget next to it.
6. Give it a title - 'HTML-new'
7. Change the 'disabled' boolean value to false
8. Change the 'isEnabled' boolean value to true;

```
<div data-bind="css: { 'draggable-widget': true, 'disabled':
false },
                    draggable: {data: draggableHtmlResultset(),
                    isEnabled: true,
                    options: {'start': function(event,
ui){lastWindowScrollPosition =
$(window).scrollTop();$('.card-body').slideUp('fast');},
                              'stop': function(event, ui){

$('.card-body').slideDown('fast',
function(){$(window).scrollTop(lastWindowScrollPosition)});

$root.collection.template.isGridLayout(false);

checkResultHighlightingAvailability();
                               }
                             }
                    }"
         title="${_('HTML-new')}" rel="tooltip"
data-placement="top">
         <a data-bind="style: { cursor:
$root.availableDraggableResultset() ? 'move' : 'default' }">
                        <i class="fa fa-code"></i>
         </a>
    </div>
```

Fig. XVIII. Snippet (1) of modifications to the Hue Search interface template ‘search.mako’

1. Navigate to ‘/usr/lib/hue/apps/search/static/js/’
2. Edit the ‘search.ko.js’ file
3. Find ‘self.draggableHtmlResultset = ko.observable(bareWidgetBuilder("HTML Results", "html-resultset-widget"));’
4. Add the following line below it

```
self.draggableHtmlResultset1 =
ko.observable(bareWidgetBuilder("HTML Results",
"html-resultset-widget1"));

Find the ‘availableDraggableResultset’ and make an additional
one like it that has a different name
‘availableDraggableResultset1’ .

self.availableDraggableResultset1 = ko.computed(function() {
    return getWidgets(function(widget) { return
['resultset-widget',
'html-resultset-widget'].indexOf(widget.widgetType()) != -1;
}).length <= 1;
  });
```

Fig. XVIIII. Snippet of modifications to the Hue Search interface javascript file ‘search.ko.js’

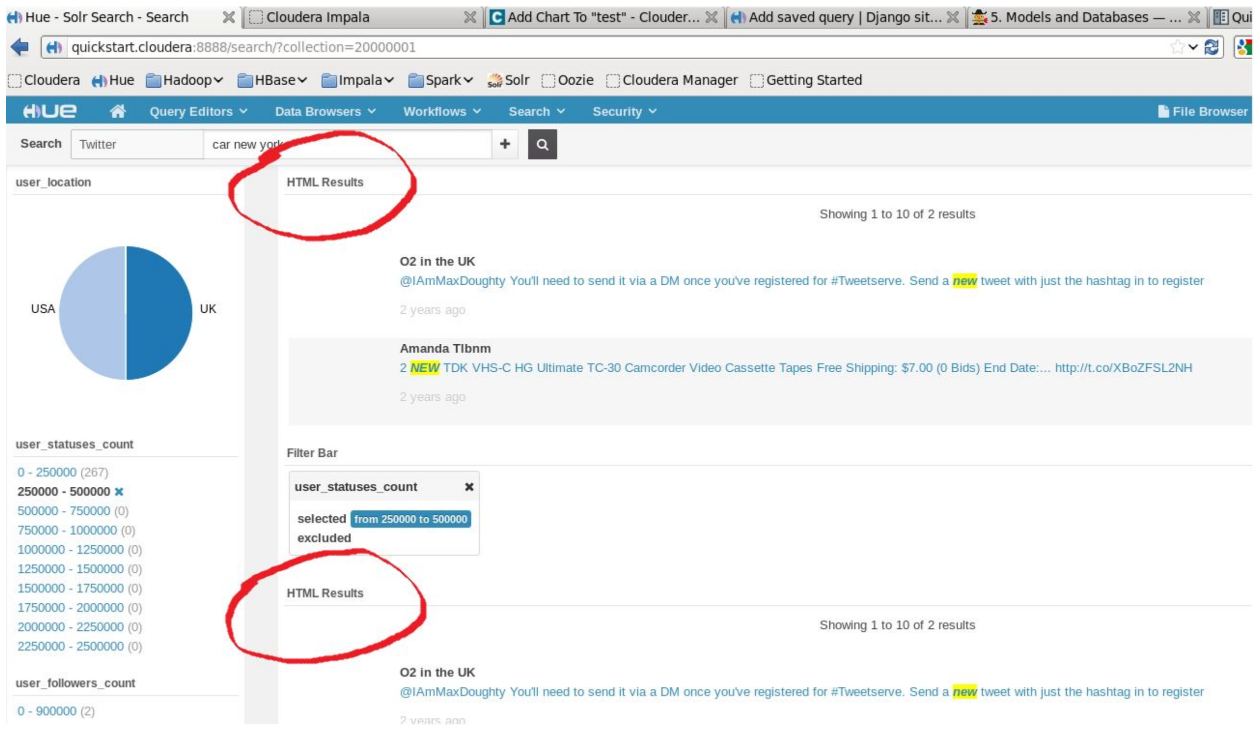
In ‘search.mako’:

1. Find the <script> tag for the ‘html-resultset-widget’
2. Create a similar one below it and name it ‘html-resultset-widget1’

```
<script type="text/html" id="html-resultset-widget1">
  <!-- ko ifnot: $root.collection.template.isGridLayout() -->
    <div data-bind="visible: $root.isEditing"
style="margin-bottom: 20px">
      <ul class="nav nav-pills">
        <li class="active"><a href="javascript: void(0)"
class="widget-editor-pill">${_('Editor')}</a></li>
        <li><a href="javascript: void(0)"
class="widget-html-pill">${_('HTML')}</a></li>
        <li><a href="javascript: void(0)"
class="widget-css-pill">${_('CSS & JS')}</a></li>
        <li><a href="javascript: void(0)"
class="widget-settings-pill">${_('Sorting')}</a></li>
      </ul>
    .
    .
    .
      “Rest of the code was deleted to save space”
    .
    .
    .

      <div class="widget-spinner" data-bind="visible:
$root.isRetrievingResults()">
        <!--[if !IE]> --><i class="fa fa-spinner
fa-spin"></i><!-- <![endif]-->
        <!--[if IE]><img src="/static/art/spinner.gif"
/><![endif]-->
      </div>
    </div>
  <!-- /ko -->
</script>
```

Fig. XX. Snippet (2) of modifications to the Hue Search interface template ‘Search.mako’

After saving and restarting the web server, we were able to get Hue to show two HTML widgets, but these widgets showed duplicate search results.



Hue - Solr Search - Search | Cloudera Impala | Add Chart To "test" - Clouder... | Add saved query | Django sit... | 5. Models and Databases — ...

quickstart.cloudera:8888/search/?collection=20000001

Search: Twitter — car new york

user_location

USA / UK

user_statuses_count

0 - 250000 (267)
250000 - 500000
500000 - 750000 (0)
750000 - 1000000 (0)
1000000 - 1250000 (0)
1250000 - 1500000 (0)
1500000 - 1750000 (0)
1750000 - 2000000 (0)
2000000 - 2250000 (0)
2250000 - 2500000 (0)

user_followers_count

0 - 900000 (2)

HTML Results

Showing 1 to 10 of 2 results

O2 in the UK
@IAmMaxDoughty You'll need to send it via a DM once you've registered for #Tweetserve. Send a *new* tweet with just the hashtag in to register
2 years ago

Amanda Tlbnm
2 *NEW* TDK VHS-C HG Ultimate TC-30 Camcorder Video Cassette Tapes Free Shipping: $7.00 (0 Bids) End Date:... http://t.co/XBoZFSL2NH
2 years ago

Filter Bar

user_statuses_count
selected from 250000 to 500000
excluded

HTML Results

Showing 1 to 10 of 2 results

O2 in the UK
@IAmMaxDoughty You'll need to send it via a DM once you've registered for #Tweetserve. Send a *new* tweet with just the hashtag in to register
2 years ago

Fig. XXI Modified Hue interface showing two HTML widgets (headings circled in red)

We continued to try to change the results in the second HTML widget and the virtual machine became unresponsive and needed to be restored from a previous snapshot.