**Web Portal**

**1. INSTALLATION MODES:**

Hadoop portal provides user with a way to set up their cluster first before they can do administration over it. It provides three installation modes:

**a)** **Typical Installation**: Typical mode as its name sounds, provides the user with automatic install feature which can be compared with one click cluster setup with the optimal settings which the automation scripts will do it automatically for you.

**b) Customised Installation:** Customised mode feature is for advanced users which provides them with the feature of tweaking and tuning their cluster according to their needs.

**c) Minimal Installation:** Minimal installation is the feature for those who wants to learn hadoop (Beginners). It setup the cluster on the standalone machine viz. also called as pseudo cluster.

**2. JOBS:**

It provides the feature of running the job through the web portal, instead of firing the job from terminal. User is asked their Mapper file and Reducer file and asked the directory address of file in which Mapper and Reducer will work and also output directory where the output from reducer will be kept. If there are some standard that an organisation uses frequently , there is a feature of Example Job , some jobs can be pre compiled and stored in before hand and all we need to do is provide the directory address of input and provide the output directory.  
   
**3. SERVICES**: It provides the default hadoop portal for monitoring cluster and jobs, but we have improved the look and feel by changing the hadoop's default portal's CSS . Now It’s a better looking portal with same power as original portal.But now users can tweak their portal to their liking and we can also provide templates for hadoop portal.

**4.USER CONSOLE:** It gives tools for admin to maintain the user directory. Also a user can be given such a portal to upload and maintain its directory.

**LOOKING UNDER DIFFERENT INSTALLATION MODES:  
TYPICAL INSTALLATION:**

**WelcomeTypical.html** : On this page user will be asked IP of NIS server and NFS Server. Since this is just a beta version we are using NIS server for authentication purpose, in upcoming versions we can also use LDAP servers. On clicking Start user will be taken back to next page.

**FinishedTypical.html**: This page is the confirmation page which will give the status of installation process. It does not do much as in typical installation user is not asked about anything about the cluster. On clicking Finalize button. User will be taken to login page from where user can administer the cluster.  
  
**What happened in the background?**  
   
After clicking the Start button on Welcome Typical.html , provided IP's of NIS and NFS server is given to the setupTypical.py script through the CGI call and NIS and NFS servers will be automatically set up in along with the cluster, and then these servers can be used during user authentication and user quota setup in the background.

**CUSTOMISED INSTALLATION:**  
**WelcomeCustomised.html** : On this page user will be asked IP of NIS server and NFS Server. Since this is just a beta version we are using NIS server for authentication purpose, in upcoming versions we can also use LDAP servers.On clicking Start user will be taken back to next page.

**Scan.html** : On this page user will find a button Start Scan on clicking this button a script from library *scan\_cluster.py* runs in the background which generates a list named nodeInfo.txt of available ip's. This file is read and the user is presented with a list of ip's on the next page viz. *HostCustom.html*: This page makes a java synchronous call and dynamically print a table from which user can select which ip they want to choose for which purpose. After selecting IPs user will click on Save Changes button and move to cluster tuning phase. **CustomTuning.html**: On this page user is asked various settings like:  
 a) Blocksize  
b) Replication Factor

c)Heartbeat Interval  
d)Default Scheduler  
e) Checkpointing Interval  
f) Minimum Heap Size  
g) Maximum Heap Size  
  
**Finished.html**: This page is the confirmation page which will give the status of installation process. It does not do much as in typical installation user is not asked about anything about the cluster. On clicking Finalize button. User will be taken to login page from where user can administer the cluster.  
   
**What happend in the background ?**   
After clicking the Start button on Welcome Typical.html , provided IP's of NIS and NFS server is given to the setupTypical.py script through the CGI call and NIS and NFS servers will be automatically set up in along with the cluster, and then these servers can be used during user authentication and user quota setup in the background. But in Customised Installation all the scripts that will be called with argument passed by the user.  
   
**ADMIN PORTAL**  
   
The admin portal of IVORY is a entire new and customized feature, which provides the easy to use and manage tools for cluster administration. It has tools from metering services and to framework support and some new features like uploading files directly from any platform directly into the HDFS, not only this but we can also process on that data all through the portal.  
  
Admin portal has been divided into five different Tabs:  
a) Dashboard  
b) Jobs  
c)Services  
 i) HDFS  
 ii) Mapreduce  
 iii) Framework  
d) User Console  
e) File Manager  
  
**DASHBOARD:**  
Dashboard is one of the strongest feature of the admin portal. It solves the efforts of writing admin commands to check for the health and status of cluster.  
Dashboard provides with visual tools which makes AJAX calls and brings the status of cluster to the admin. Right now we have provided with following features:

i) Tasktrackers Connected  
 ii) Jobs Completed  
iii) Connected Datanodes  
iv) Number of users  
v) Bandwidth in use  
vi) Memory in use  
vii) Disk Space Usage  
viii)Slots used  
ix) Heap Size Usage  
x) CPU Usage  
xi) Demographics : In this feature the cluster user stats of a particular user can be monitored  
xii) Server Load: Once completed this provides the instantaneous network usage by cluster, this can help admin monitor if some segment of network is getting choked.  
   
   
**BEHIND THE SCENE:**  
These features are working by making a call to background scripts which uses linux as administration command like sysstat. In sysstat there are commands like iostat, mpstat, pidstat, sadf, sar which can be used to get the system status and used grep command we can cut the required value and print it in a file.Then making AJAX calls to read these files on regular intervals.

**SOME ISSUES WE FACED:**  
What we did right now is not a perfect implementation issues in file read operation because of file read lock can be a problem sometimes. Sometimes the page just freezes.

**JOBS:** It provides the feature of running the job through the web portal, instead of firing the job from terminal. User is asked their Mapper file and Reducer file and asked the directory address of file in which Mapper and Reducer will work and also output directory where the output from reducer will be kept.  
If there are some standard that an organisation uses frequently, there is a feature of Example: Some jobs can be pre compiled and stored in before hand and all we need to do is provide the directory address of input and provide the output directory.

**SERVICES:** It provides the default hadoop portal for monitoring cluster and jobs, but we have improved the look and feel by changing the hadoop's default portal's CSS . Now It’s a better looking portal with same power as original portal.  
 But now users can tweak their portal to their liking and we can also provide templates for hadoop portal.

**USER CONSOLE**: It gives tools for admin to maintain the user directory. Also a user can be given such a portal to upload and maintain its directory.

**Features:**

**User ADD**: Now admin can add and remove a user from the portal. Also File Quota & Space Quota can be set along with it. Restrictions on user: If the user is already present the quotas can be set later on. Both of these calls a user\_quota.py files from the library.

**Job Scheduling**: User can be assigned minimum mappers and reducers and maximum number of jobs quota. Also their priorities can be set before hand.

**Final Manager**: This feature is one of the most important features of dashboard.  
Through this feature we have tried to bring platform independence into the hadoop world. Before this there is no way in which we can upload a file on hadoop setup on linux machine from the windows machine. But we have implemented it by staging the file temporarily on the apache server and from apache server we have uploaded a file inside HDFS. To upload the file to staging area we have used an algorithm that segments the file in smaller blocks that can be uploaded quickly and efficiently.

**Issues:** There are some drawbacks with this method , firstly it is not scalable. Now the file upload limit is an issue , but it’s not an issue of algorithm but it is an issue file size constraint on apache server. As soon as this limit can be managed, we can make this method work by introducing parallelism in the upload process.  
   
   
**FUNCTIONALITIES ONCE THE CLUSTER IS READY AND RUNNING**:  
**FileManager.html** : On this page the basic functionalities of adding and removing a file to and from the cluster are provided. User can list all the files and folders in the base directory and remove them if desired, all by just one click on the provided delete button. User may also upload the directly into the hadoop cluster from any device(not necessarily using a linux distribution).

**What happened in the background?**  
By uploading the file on the web server we use it as a staging ground before the web server, which itself is a client of namenode, uploads the staged file into the hadoop cluster by using a simple CGI script.

**Jobs.html** : On this page user is provided with 2 different forms. In one form user is provided with some precompiled mapreduce jobs as example runs and in second form user may provide a custom job written in any language by asking them for separate mapper and reducer file and the path of input file and output directory in hdfs.

**What happens in background?**

The language independence in hadoop cluster's mapreduce programming is achieved by using hadoop's streaming library for compiling the mapper and reducer. By cleverly using any language for activating shell scripts one may achieve even more flexibility by running some pre installed command line softwares for MR jobs.  
   
 **Resusing and customising Hadoop's original interface:**

**HDFS.html**: For monitoring the cluster instead of rebuilding everything from scratch we embedded the Hadoop's original HDFS interface in our page after changing and customising its original css. As we in simplicity and beauty this helped us in achieving abstraction of port awareness in user for using the original interface at the same time providing him all the required information about his cluster.

**Mapread (2).html:** For managing jobs in the cluster we embedded the jobtracker interface as well in a separate page with a similar mindset. This helped us in achieving high level of abstraction in MR management and administration.  
   
**FUTURE WORK:**   
We are trying to achieve a fully functional Dashboard support with notifications for administrative purposes and will soon be implementing a complete framework support for pig, hive, sqoop and HBase so as to provide a fully ready to use cluster to anyone who would like to give IVORY a chance to catch their eyes.