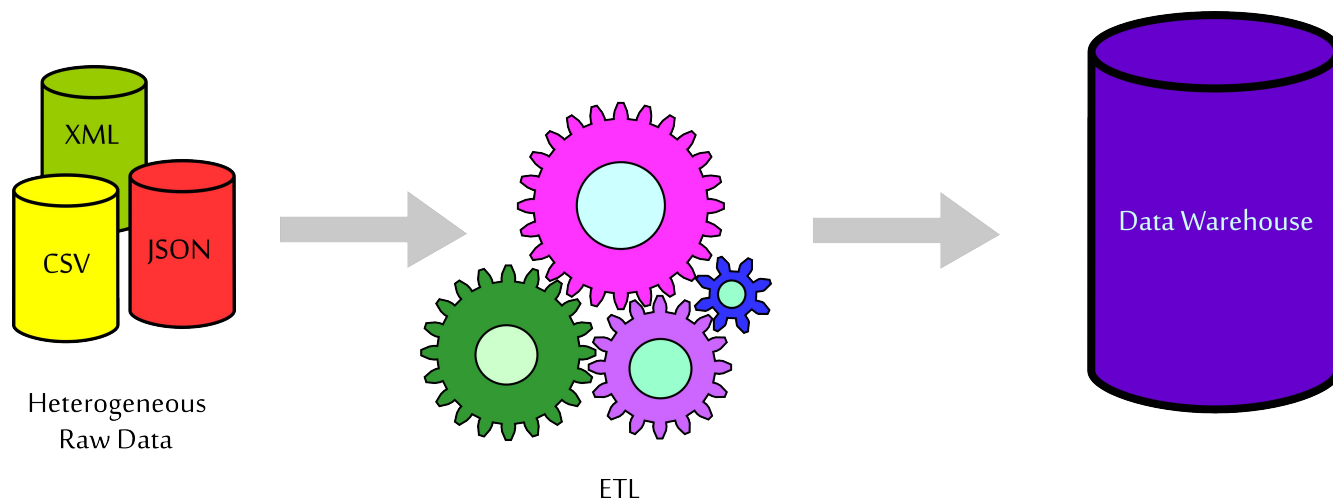


# پیاده‌سازی یک سیستم اجماع و انبار داده ها

پروژه درس پایگاه های داده ی توزیع شده و موبایل

حسن عابدی

۹۲۷۲۳۱۴۷



## شرح مسأله

طراحی و ساخت یک سیستم مدیریت تصفیه و انبار اطلاعات با فرمت های مختلف ورودی به صورت متجانس.

به زبان ساده ذخیره داده های جمع آوری شده از منابع و پایگاه های گوناگون که از لحاظ قالب و اسکلت با هم تفاوت دارند.. این گونه داده ها باید به صورت متداول سه مرحله را پشت سرگذارند تا به صورتی یکپارچه برای استفاده های بعدی اعم از آنالیز و یا تولید دانش به کار برده شوند. سه کار اصلی در این مرحله عبارتند از:

۱. Extraction (بیرون کشیدن و جمع آوری داده)
۲. Transforming (تبدیل)
۳. Load (انباشت)

در زیر شرح عمل یک پیاده سازی نرم افزاری از یک نمونه راه حل برای چنین مسأله ای شرح داده شده است.

می خواهیم دو فایل با مشخصات زیر را در یک پایگاه داده و یا چیزی شبیه به آن ذخیره کنیم.

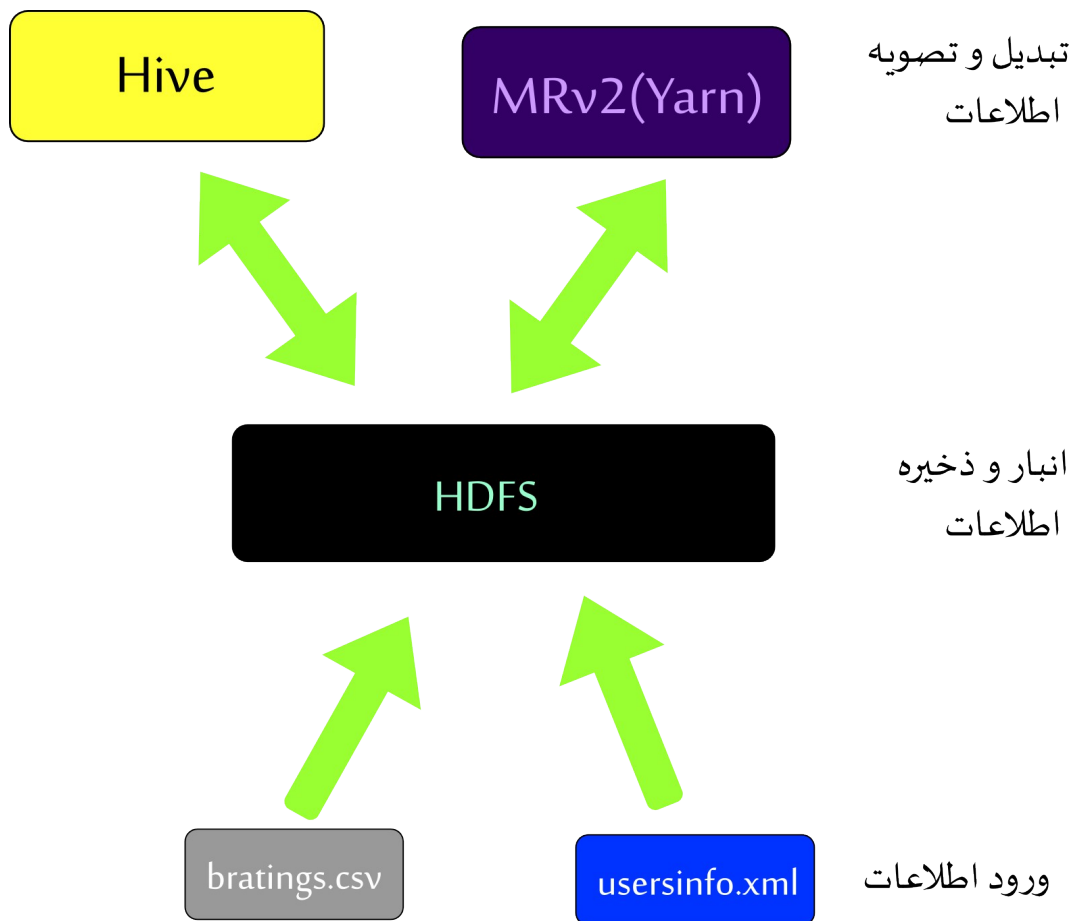
نام فایل	نمونه از محتویات	قالب محتویات															
bratings.csv	<table> <tr> <td>276725</td><td>034545104X</td><td>0</td></tr> <tr> <td>276726</td><td>0155061224</td><td>5</td></tr> <tr> <td>276727</td><td>0446520802</td><td>0</td></tr> <tr> <td>276729</td><td>052165615X</td><td>3</td></tr> </table>	276725	034545104X	0	276726	0155061224	5	276727	0446520802	0	276729	052165615X	3	<table> <tr> <td>نام کاربری</td><td>Isbn کتاب</td><td>نمره ای که کاربر داده</td></tr> </table>	نام کاربری	Isbn کتاب	نمره ای که کاربر داده
276725	034545104X	0															
276726	0155061224	5															
276727	0446520802	0															
276729	052165615X	3															
نام کاربری	Isbn کتاب	نمره ای که کاربر داده															
usersinfo.xml	<pre> &lt;user&gt; &lt;uid&gt;13&lt;/uid&gt; &lt;location&gt; "barcelona; barcelona; spain" &lt;/location&gt; &lt;age&gt;26&lt;/age&gt; &lt;/user&gt;  &lt;user&gt; &lt;uid&gt;14&lt;/uid&gt; &lt;location&gt; "mediapolis; iowa; usa" &lt;/location&gt; &lt;age&gt;0&lt;/age&gt; &lt;/user&gt;  &lt;user&gt; &lt;uid&gt;15&lt;/uid&gt; &lt;location&gt; "calgary; alberta; canada" &lt;/location&gt; &lt;age&gt;0&lt;/age&gt; &lt;/user&gt; </pre>	<pre> &lt;user_info&gt; &lt;user&gt;    &lt;uid&gt;     X:INT   &lt;/uid&gt;   &lt;location&gt;     Y:STRING   &lt;/location&gt;   &lt;age&gt;     Z:INT   &lt;/age&gt;  &lt;/user&gt;  ... &lt;/user_info&gt; </pre>															

همان‌طور که می‌بینید قالب دو فایل متفاوت است. اولی یک فایل CSV و دیگری یک فایل XML می‌باشد ولی اطلاعات درون این دو فایل به هم مرتبط می‌باشند بدین صورت که فایل اول حاوی شماره‌ی مجموعه کتاب‌ها در یک سایت مثل آمازون می‌باشد به همراه شماره‌ای که هر کاربر آن سایت بدان کتاب به عنوان نمره منتصب کرده است به همراه شماره‌ی کاربری. فایل دوم اطلاعات مکان و سن افراد عضو سایت ما را به همراه شماره‌ی کاربری هر کاربر نشان می‌دهد.

حال هدف ادغام این دو فایل نامتجانس و استفاده از اطلاعات درون آن‌ها به صورتی یکپارچه و متجانس می‌باشد.

## راه حل پیشنهادی

استفاده از زیرساخت‌های آپاچی هدوپ<sup>۱</sup> به همراه نرم‌افزارهای تصویه و انبار داده مثل پیگ<sup>۲</sup> و هایو<sup>۳</sup> برای تصویه و تبدیل و انباشت داده‌ها.



---

1 Apache Hadoop  
2 Apache Pig  
3 Apache Hive

## مراحل انجام کار

۱. ساختن و تنظیم یک کلاستر آپاچی هدوپ به همراه نصب هایو روی آن.

در زیر مشخصات کلی کلاستر هدوپ مورد استفاده برای حل مشکل توصیفی قسمت قبل را می بینید.

تعداد گره های داده ای و نام	نسخه ی هایو	فضای ذخیره سازی گره داده	نسخه ی هدوپ	نسخه ی جاوای مورد استفاده	رم سیستمی که کلاستر روی آن اجرا می شود
1	0.13.1	80 گیگابایت	2.2.0 (stable)	1.7.0_55	8 گیگابایت

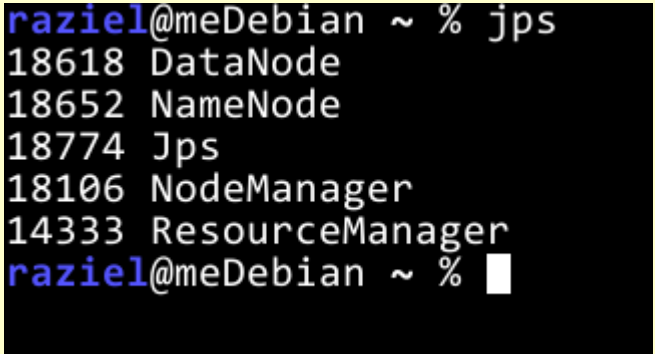
توجه: برای فهمیدن چگونگی انجام این عمل یعنی نصب و راه اندازی کلاستر هدوپ به همراه هایو می توانید به سایت خود آپاچی مراجعه کنید.

باید متذکر شد که حتماً متغیرهای محیطی جدول زیر را روی سیستم پیش از نصب و اجرای هدوپ و هایو ثبت و مقدار دهی کنید.

نام متغیرهای محیطی مهم	نمونه مقدار دهی
HADOOP_HOME	<pre>\$# hadoop home environment variables \$export HADOOP_HOME=~/.Big-Data/hadoop-2.2.0 \$export PATH=\$HADOOP_HOME/bin:\$PATH</pre>
HIVE_HOME	<pre>\$#hive home environment variables \$export HIVE_HOME=~/.Big-Data/apache-hive-0.13.1-bin \$export PATH=\$HIVE_HOME/bin:\$PATH</pre>
JAVA_HOME	<pre>\$#me custom java home \$export JAVA_HOME=\$BIGDATA_DIR/jdk</pre>

## ۲. راه اندازی کلاستر و انتقال دو فایل مورد نظر روی HDFS<sup>۵</sup>.

بعد از راه اندازی هدوپ باید به طور کلی با فرض تنظیمات مناسب سیستم دستورالعمل های زیر را به ترتیب در خط فرمان وارد کنید.

لینک ها و دستورالعمل ها	شرح
<code>\$yarn resourcemanager</code>	یکی از دو دیمن اصلی هدوپ که باید روی هر گره رئیس از کلاستر همیشه در حال اجرا باشند.
<code>\$yarn nodemanager</code>	یکی از دو دیمن اصلی هدوپ که باید روی هر گره کارگر از کلاستر همیشه در حال اجرا باشند.
<code>\$hdfs namenode -format</code>	اگر کلاستر را برای بار اول اجرا می کنید این دستور را حتماً وارد کنید. (مواظب باشید این دستور اطلاعات فعلی رور فایل سیستم هدوپ را از بین می برد.)
<code>\$hdfs namenode</code>	سرویس مسئول راه اندازی مکاندار اطلاعات ذخیره شده روی فایل سیستم هدوپ.
<code>\$hdfs datanode</code>	سرویس مسئول بلاک های فایلی روی فایل سیستم هدوپ.
<code>\$jps</code>	اگر همه چیز درست انجام شده باشد خروجی این دستور چیزی شبیه به تصویر پایین می باشد.  

در این قسمت باید فایل ها را روی فایل سیستم هدوپ قرار دهیم.

```
$hadoop fs -put usersinfo.xml /usersinfo.xml
```

```
$hadoop fs -put bratings.csv /bratings.csv
```

حالا برای مشاهده ی فایل ها که روی فایل سیستم می توانید از یکی از دو روش زیر استفاده کنید.

```
$hadoop fs -ls
```

با مرورگر اینترنتی خود به نشانی زیر بروید.

<http://localhost:50070/dfshealth.jsp>

۳. پردازش و تبدیل دو فایل bratings.csv و usersinfo.xml که روی hdfs قرار دارند به کمک هایو MR.

- برای استفاده از فایل usersinfo.xml باید ابتدا به کمک MR فرمت آن را به صورتی در آوریم که به ساده ترین وجه بتوانیم از آن در هایو برای ساختن جدولی از اطلاعات استفاده کنیم.  
مراحلی که برای این کار باید طی شوند به ترتیب از بالا به پایین در جدول زیر آمده اند.
- توجه: فایل XmlParser11.java همراه این گزارش ضمیمه می باشد.

کمپایل و ساختن فایل جار از برنامه ی پارزر فایل xml

```
## compilation + making the main.jar
$mkdir Main
$javac -classpath 'yarn classpath' -d Main XmlParser11.java
$jar -cvf main.jar -C XmlParser11 .
```

اجرای برنامه پارزر xml توسط MR و ذخیره ی اطلاعات روی HDFS

```
## running main.jar + saving the results as usersinfo.csv on hdfs
$yarn jar main.jar com.org.XmlParser11 /usersinfo.xml /usersinfo.csv
```

در این مرحله دو فایل usersinfo.csv و bratings.csv که روی HDFS قرار دارند را به کمک دستور های زیر درون یک دیتابیس به نام books در هایو ذخیره می کنیم.

کمپایل و ساختن فایل جار از برنامه ی پارزر فایل xml

```
# starting Hive + creating database + loading two files from hdfs
$hive
>CREATE DATABASE BOOK COMMENT 'Holds all book tables';

# creating a table in hive to hold data from '/bratings.csv'
>CREATE TABLE Bratings (ID INT,IBSN STRING, RATING INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ",";
>LOAD DATA INPATH '/bratings.csv' INTO TABLE BRATINGS;

# creating a table in hive to hold data from '/usersinfo.csv'
>CREATE TABLE Users (ID INT,LOCATION STRING, AGE INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ",";
>LOAD DATA INPATH '/usersinfo.csv' INTO TABLE Users;
```

# some query examples + sample results

```
>SELECT Users.ID,Bratings.RATING FROM Users join Bratings  
ON(USERS.ID = BRATINGS.ID);
```

```
99947    49  
99948     0  
99949    27  
9995     20  
99950     0  
99951    49  
99952    16  
99953    28  
99954    39  
99955    14  
99956     0  
99957    22  
99958    16  
99959    35  
9996     29  
99960     0  
99961     0  
99962     0  
99963    40  
99964    35  
99965    30
```

```
>SELECT * FROM Users join Bratings ON(USERS.ID = BRATINGS.ID);
```

```
99989 "lyon; rhone alpes; france" 45 99989 "lyon; rhone alpes; france" 45  
9999 "beaverton; oregon; usa" 0 9999 "beaverton; oregon; usa" 0  
99990 "marl; nordrhein-westfalen; germany" 16 99990 "marl; nordrhein-westfalen; germany" 16  
99991 "boonton township; new jersey; usa" 20 99991 "boonton township; new jersey; usa" 20  
99992 "darwin; northern territory; australia" 55 99992 "darwin; northern territory; australia" 55  
99993 "munich; bayern; germany" 35 99993 "munich; bayern; germany" 35  
99994 "hambergen; lower saxony; germany" 0 99994 "hambergen; lower saxony; germany" 0  
99995 "lancaster; california; usa" 41 99995 "lancaster; california; usa" 41  
99996 "toronto; ontario; canada" 43 99996 "toronto; ontario; canada" 43  
99997 "fredericton; new brunswick; canada" 33 99997 "fredericton; new brunswick; canada" 33  
99998 "cambridge; england; united kingdom" 22 99998 "cambridge; england; united kingdom" 22  
99999 "hamburg; hamburg; germany" 46 99999 "hamburg; hamburg; germany" 46
```

۴. حالا داده ی ما درون دو جدول به نام های **users** و **bratings** ذخیره شده است و آماده ی عمل های بعدی می باشد.