Its always tough to parse XML, especially when it comes to PIG. Here I am explaining two approaches to parse an XML file in PIG.

1. Using Regular Expression  
2. Using XPath

For simplicity I am taking a sample XML as shown below. This file should be in HDFS for processing

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <CATALOG>  <BOOK>  <TITLE>Hadoop Defnitive Guide</TITLE>  <AUTHOR>Tom White</AUTHOR>  <COUNTRY>US</COUNTRY>  <COMPANY>CLOUDERA</COMPANY>  <PRICE>24.90</PRICE>  <YEAR>2012</YEAR>  </BOOK>  <BOOK>  <TITLE>Programming Pig</TITLE>  <AUTHOR>Alan Gates</AUTHOR>  <COUNTRY>USA</COUNTRY>  <COMPANY>Horton Works</COMPANY>  <PRICE>30.90</PRICE>  <YEAR>2013</YEAR>  </BOOK>  </CATALOG> |

**Using Regular Expressions**

I am using the XMLLoader() in piggy bank UDF to load the xml, so ensure that Piggy Bank UDF is registered.  Then I am using regular expression to parse the XML

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | REGISTER piggybank.jar    A =  LOAD 'xmls/hadoop\_books.xml' using org.apache.pig.piggybank.storage.XMLLoader('BOOK') as (x:chararray);    B = foreach A GENERATE FLATTEN(REGEX\_EXTRACT\_ALL(x,'<BOOK>\\s\*<TITLE>(.\*)</TITLE>\\s\*<AUTHOR>(.\*)</AUTHOR>\\s\*<COUNTRY>(.\*)</COUNTRY>\\s\*<COMPANY>(.\*)</COMPANY>\\s\*<PRICE>(.\*)</PRICE>\\s\*<YEAR>(.\*)</YEAR>\\s\*</BOOK>'));    dump B; |

Output

|  |  |
| --- | --- |
| 1  2 | (Hadoop Defnitive Guide,Tom White,US,CLOUDERA,24.90,2012)  (Programming Pig,Alan Gates,USA,Horton Works,30.90,2013) |

**Using XPath**

[XPath](http://en.wikipedia.org/wiki/XPath) is a function that allows text extraction from xml. Starting PIG 0.13 , Piggy bank UDF comes with XPath support. It eases the XML parsing in PIG scripts.  A sample script using XPath is as shown below

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | REGISTER piggybank.jar  DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();    A =  LOAD 'xmls/hadoop\_books.xml' using org.apache.pig.piggybank.storage.XMLLoader('BOOK') as (x:chararray);    B = FOREACH A GENERATE XPath(x, 'BOOK/AUTHOR'), XPath(x, 'BOOK/PRICE');  dump B; |

Output

|  |  |
| --- | --- |
| 1  2 | (Tom White,24.90)  (Alan Gates,30.90) |

Future enhancements:

If you check the XPath UDF [source code](https://github.com/apache/pig/blob/branch-0.14/contrib/piggybank/java/src/main/java/org/apache/pig/piggybank/evaluation/xml/XPath.java)  , you can see that input to Xpath is a Tuple and it always returns a String. But there will be cases in which we might want XPath to return a Tuple instead of String. So there is scope to extend XPath with this feature and contribute to community.

Now start feeding the Pig !