<u>Removing Red links from given XML data for Wikipedia links:</u>

In a distributed environment like Hadoop, each entry is processed parallelly in different machines, there is no way to apply the above algorithm. Thus, a well-known Map-Reduce algorithm would be required; the algorithm is as follows:

1) Parse the XML as blocks of <page></page> tags
2) For each title write a '#' character in the context (since we also need to print pages with no valid outlinks)
3) For each valid outlink encountered in the page, enter a mapping of (link, key)
4) In the reducer, take all the valid mappings for a key and check for a # in the value. If encountered, then print out all the values in reverse order,i.e., value-key (except for key-# - write it as it is). If # not encountered, then it is a red link(only valid page introduces a key-# mapping; no # - no valid page)

5) Write another job to gather the mapping generated from the above job and combine them key-wise(collate all valid outlinks for a particular key) and output.

6) Run the jobs via hadoop jobcontrol in sequence.

Challenges faced:
1) Environment setup on local machines – Hadoop does not have native support for Windows (realized quite late)
2) Map-Reduce concepts:

   a) Difference between reducer and combiner (both implemented as reducer in java)
   b) How a mapper node processes data and how a reducer node processes data (key-wise)
   c) Introduction to Writables and Iterables and their different functioning from similar standard Java structures.
   d) AWS issues (java project running from US-West causes issues etc.)