Lumberyard Demo Setup Instructions

Author: Josh Patterson
jpatterson@floe.tv

Instructions

- Download VMWare
- Download a VM image
    - http://www.thoughtpolice.co.uk/wmware
    - Centos 5.5 is a great choice, 32bit
    - You may want to crank up memory to 2GB for the instance.
- Start your VM in VMWare
- Install Hadoop 0.20 on your VM in pseudo-distributed mode.
    - To make it easy, we'll use CDH3
    - https://wiki.cloudera.com/display/DOC/Hadoop+%28CDH3%29+Quick+Start+Guide
    - Easy way to do this:
        - yum -y install hadoop-0.20-conf-pseudo
    - You will need to upgrade Java (as talked about in the instructions)
    - Will also need to add Cloudera's repository
- Install HBase 0.89
    - To make it easy, we'll use Cloudera's repository
        - https://wiki.cloudera.com/display/DOC/HBase+Installation
        - Follow instructions closely
- Build jmotif from trunk at
    - http://code.google.com/p/jmotif/
    - A build of this is already included in the lib directory
    - We need this for the latest iSAX indexing commits
        - provides the bulk of the mathematics of the indexing scheme.
    - If you want to build jmotif yourself, you'll need at least ant 1.7 and you'll need to set the following env vars
        - export JUNIT_HOME=<junit_home>
        - export FINDBUGS_HOME=<findbugs_home>
        - export WEKA_HOME=<weka_home>
            - you'll need a later weka, preferably 3.7.2
    - To build the source:
        - ant
    - To build the jmotif jar (already one pre-built in the lib dir)
        - ant -f jar.build.xml
- Use Git to pull down Lumberyard from github

    - git clone git://github.com/jpatanooga/Lumberyard
    - Build
        - use ant to build the Lumberyard jar
        - ant
- Once we have both jmotif.lib.jar and Lumberyard.jar built, copy both jars into the VM onto the hbase path somewhere.
- To execute the Lumberyard shell, type:
    - **hbase tv.floe.lumberyard.hbase.isax.index.shell.Shell** <command>
        - To view the help
            - **hbase tv.floe.lumberyard.hbase.isax.index.shell.Shell**
        - To create an index
            - **hbase tv.floe.lumberyard.hbase.isax.index.shell.Shell** CreateIndex “my_index”
                - -base_card 4
                - -dim_split 1
                - -base_word_len 8
                - -base_ts_sample_size 16
                - -split_threshold 100
        - Now let’s index one of the two DNA files in the data subdir:
            - in the data subdirectory, we have the mitochondrial DNA of both the polar bear and the Hippo. These samples were downloaded from http://www.cs.ucr.edu/~eamonn/iSAX/iSAX.html
        - For a technical demo, let’s search our index:
            - **hbase tv.floe.lumberyard.hbase.isax.index.shell.Shell** SearchGenomeIndex "polar_bear" "TGGACTAAATGACTAA"
                - which should find one result
        - If we search for a slightly different pattern, we’ll see its not found:
            - **hbase tv.floe.lumberyard.hbase.isax.index.shell.Shell** SearchGenomeIndex "polar_bear" "TGGACTAAATGACTAG"
- The current demo
    - is built to demonstrate iSAX indexing and iSAX indexing persisted in HBase.
    - is meant to build indexes from a single thread
        - parallel index construction from Map Reduce is coming, requires some distributed lock coordination with something like Zookeeper
    - should not be counted on for more than prototyping purposes
    - is a proof of concept for scalable index construction and search
- Potential Use Cases
    - Genomic pattern matching
    - General timeseries search engine
    - Use variations with Map Reduce
    - Audio search
    - Image pattern search

References

- Based on the original paper by Shieh and Keogh:
    - http://www.cs.ucr.edu/~eamonn/iSAX.pdf
- iSAX homepage:
    - http://www.cs.ucr.edu/~eamonn/iSAX/iSAX.html