Experiences with Spark – Building K-means

* Collected benchmark data on…
  + Increasing dataset size
    - 10 million to 2.5 billion 2D points, or ~150 MB to ~38 GB
    - 2 m1.large slaves on EC2
    - Linear size vs. time, as expected
  + Scale-out
  + Scale-up
  + Cache sizes and strategies
* Experiences:
  + As we know, debugging a distributed system can be frustrating
    - Tasks get lost with no explanation up-front
    - Finding the reason requires fast reactions
    - Solution: Present framework logs *persistently* on the Mesos dashboard
    - Tasks can also appear to hang because of insufficient heap size, but when you run top there’s no CPU usage. Mysterious
  + Repeatability is important, so scripting a task and running it on the cloud is a must
    - Is there already an easy way to customize the AMI or hook into the setup script?
    - My workaround: modify the setup script directly, or do the setup over SSH afterwards
  + Writing an application with Spark feels a little kludgy despite using JARs – am I doing it right?
    - I wrote a custom Makefile to build Spark into a JAR and then build my own sources into a JAR with Spark in the classpath
    - I manually added my project’s JAR to the SPARK\_CLASSPATH in spark-env.sh
    - I copied and modified the Spark run script to pass the classpath to Scala
    - This is from trial and error – I would be happy to write up the correct way as a wiki page
* Lessons: