**DAHS LINK FLIGHT OPERATIONS DATA**

Decisions

* How to handle different sample rates. Is the current approach workable?
* Is the data from all aircraft necessary? If no, what aircraft do we need.
* Too much data to process with R on a single computer.
  + We could rent a machine on Amazon with hundreds of GB of RAM, but R can only use a single core. Here are some options…

| **Option** | **Pros** | **Cons** |
| --- | --- | --- |
| Subsample! | * Cheap – no special hardware required * Easy – no special knowledge required * Has not been proven to not work – There is no evidence that will miss an anomaly if the analysis is done on a subsample of the data. * Fast – If the analysis runs too slowly, decrease sample size * Local – analysis can be done on personal computers/laptop | * Might miss some information that indicates an anomaly * Need to design good sampling strategy |
| Elastic Map/Reduce | * Well understood – map/reduce is a course-grained way to distribute work to many nodes. * Big data - handles large files data well * Analysis that takes a single pass through the data (e.g. sum, average, count) works well. | * Expensive – Any AWS solution means creating a cluster and paying by the hour. Impractical to turn machines on and off dynamically because we restarted invokes a 1 hour charge for the machine * Hard – New skills needed to work with HDFS and map/reduce * Limited flexibility – map/reduce paradigm works well when data records or independent of one-another. Graph problems, or problems that require random access to the data do not map/reduce paradigm. * Not local – PCs and laptop only provide SSH access to the platform that does the work. * Time – Hadoop will get the job done, but job time is measured in hours or days. |
| H2O | * Point-and-click easy. * Front-end R interfaces to H2o running in the cloud * Easy to spin up an h2o cluster in EC2 | * Expensive for the same reason as Elastic Map Reduce * Time – H2o is fast, time is still proportional to the size of the data. * Limited flexibility – h2o comes bundled with a few nice modeling techniques, but outside of that you have to write your own in Java. Time consuming to acquire these skills * Lock-in – Any analysis done in h2o stays in h2o. Even more true for any custom Java code |
| Revolution Analytics | * R is the front-end | * Expensive for same reason as EMR * Could be an expensive license because it is a proprietary product. * I do not know anything about the product. * Vendor lock-in potential |