SciSpark: Highly Interactive & Scalable Model Evaluation and Climate Metrics

Brian Wilson, Chris Mattmann, Duane Waliser, Jinwon Kim, Paul Loikith, Huikyo Lee, Lewis John McGibney, Maziyar Boustani, Michael Stark, Kim Whitehall
Jet Propulsion Laboratory / California Institute of Technology

**Summary**

Under a NASA AIST grant, we are developing a lightning fast Big Data technology called SciSpark based on Apache Spark. Spark implements the map-reduce paradigm for parallel computing on a cluster, but emphasizes in-memory computing, “spilling” to disk only as needed, and so outperforms the disk-based Apache Hadoop by 100x in memory and by 10x on disk, and makes iterative algorithms feasible. This 2nd generation capability for NASA’s Regional Climate Model Evaluation System (RCMES) will compute simple climate metrics at interactive speeds, and extend to quite sophisticated iterative algorithms such as machine-learning (ML) based clustering of temperature PDFs, and even graph-based algorithms for searching for Mesoscale Convective Complexes (MCC’s). The goals of SciSpark are to: (a) Decrease the time to compute comparison statistics and plots from minutes to seconds; (b) Allow for interactive exploration of time-series properties over seasons and years; (c) Decrease the time for satellite data ingestion into RCMES to hours; (d) Allow for Level-2 comparisons with higher-order statistics and/or full PDF’s in minutes to hours; and (e) Move RCMES into a near real time decision-making platform.

The capabilities of the SciSpark compute cluster will include:

1. **On-demand data discovery and ingest for satellite (A-Train) observations and model variables (from CORDEX and CMIP5) by using OpenDAP and webification (2010n) to subset arrays out of remote or local HDFS and netCDF files;**
2. **Use of HDFS, Cassandra, and SparkSQL as a distributed database to cache variables/arrays for later reuse with fast, parallel I/O back into cluster memory;**
3. **Parallel computation in memory of model diagnostics and decade-scale comparison statistics by partitioning work across the SciSpark cluster by time period, spatial region, and variable;**
4. **An integrated browser UI that provides a “live” code window (python & scala) to interact with the cluster, interactive visualizations using D3 and webGL, and search forms to discover & ingest new variables.**

The research described here was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

**SciSpark Contributions**

- Parallel Ingest of Science Data from HDF & netCDF
- Using OpenDAP and Webification to access arrays
- Scientific RDD’s for large arrays (SRDD’s)
  - Bundles of 2,5,4-dimensional arrays keyed by name
  - Partitioned by time and/or space
- More Operators
  - ArraySplit by time and space, custom statistics, etc.
- Sophisticated Statistics and Machine Learning
  - Higher-Order Statistics (skewness, kurtosis)
  - Multivariate PDF’s and histograms
- Cluster, graph algorithms
- Persisted in Cache
  - Store named arrays in distributed Cassandra db or HDF5

**Visualizations**

- Interactive Statistics and Plots
  - “Live” code window submits jobs to SciSpark Cluster
  - Incremental statistics and plots “stream” into the browser UI

**The SciSpark Cluster**

Apache Spark is an in-memory map-reduce platform. http://spark.apache.org/

- Spark features include:
  - Distributed processing
  - SQL Query Syntax
  - Persistence of data
  - Graph processing
  - RDD (Resilient Directed Dynamic Datasets)
- Spark SQL (Spark SQL is a SQL engine built on top of Hadoop MapReduce)
- Spark ML (Spark ML is a library for machine learning algorithms)

SciSpark’s is a research platform to develop scientific Spark processing workflows.

The SciSpark test cluster provides:

- Multiple nodes for parallel computation (4 nodes, 32 cores)
- Spark processing environment (Python, Scala, Java)
- Distributed file system (HDFS, Cassandra)
- Apache Mesos cluster manager

**Parallel Clustering & PDF Generation**

**Apache Spark**

- Cancer Genomics, Energy Debugging, Smart Buildings
- S3, HDFS, YARN

**Apache Mesos**

- Ingest global station data in CSV format, exercise SparkSQL, data integration code for merging data from netCDF, HDF, and DAP
- Architecture & Design
  - Designing data structures for scientific RDD’s
- Challenges
  - Interoperating between Python/ruby/arrays and Java/Scala

**Software Prototypes**

- Ingest global station data in CSV format, exercise SparkSQL, data integration code for merging data from netCDF, HDF, and DAP
- Architecture & Design
  - Designing data structures for scientific RDD’s
- Challenges
  - Interoperating between Python/ruby/arrays and Java/Scala
  - Prototyping Cassandra as key/value store for named arrays

**Next Steps**

- Reproduce prior RCMES model diagnostic runs in SciSpark paradigm
- Quantify speedup
- Implement custom statistics algorithms and “scale up” the cluster
- Develop & Integrate the browser UI: Live code, interactive visualization