Geophysical Research Abstracts
Vol. 18, EGU2016-10658, 2016
EGU General Assembly 2016
© Author(s) 2016. CC Attribution 3.0 License.

SciSpark: Highly Interactive and Scalable Model Evaluation and Climate Metrics

Brian Wilson (1), Chris Mattmann (1), Rahul Palamuttam (2), Kim Whitehall (1), Renato Marroquin Mogrovejo (3), Maziyar Boustani (1), and Rishi Verma (1)

(1) Jet Propulsion Laboratory, Pasadena, United States (Brian.Wilson@jpl.nasa.gov), (2) University of California San Diego, United States, (3) ETH Zurich

Remote sensing data and climate model output are multi-dimensional arrays of massive sizes locked away in heterogeneous file formats (HDF5/4, NetCDF 3/4) and metadata models (HDF-EOS, CF) making it difficult to perform multi-stage, iterative science processing since each stage requires writing and reading data to and from disk. We are developing a lightning fast Big Data technology called SciSpark based on ApacheTM Spark under a NASA AIST grant (PI Mattmann). Spark implements the map-reduce paradigm for parallel computing on a cluster, but emphasizes in-memory computation, “spilling” to disk only as needed, and so outperforms the disk-based ApacheTM Hadoop by 100x in memory and by 10x on disk.

SciSpark will enable scalable model evaluation by executing large-scale comparisons of A-Train satellite observations to model grids on a cluster of 10 to 1000 compute nodes. 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 based clustering of temperature PDFs, and even graph-based algorithms for searching for Mesocale Convective Complexes.

We have implemented a parallel data ingest capability in which the user specifies desired variables (arrays) as several time-sorted lists of URL’s (i.e. using OPeNDAP model.nc?varname, or local files). The specified variables are partitioned by time/space and then each Spark node pulls its bundle of arrays into memory to begin a computation pipeline. We also investigated the performance of several N-dim. array libraries (scala Breeze, java jblas & netlib-java, and ND4J). We are currently developing science codes using ND4J and studying memory behavior on the JVM. On the pyspark side, many of the science codes to be integrated already use the numpy and SciPy ecosystems.

The talk will cover: the architecture of SciSpark, the design of the scientific RDD (sRDD) data structure, our efforts to integrate climate science algorithms in Python and Scala, parallel ingest and partitioning of A-Train satellite observations from HDF files and model grids from netCDF files, first parallel runs to compute comparison statistics and PDF’s, and first metrics quantifying parallel speedups and memory & disk usage.