Apache Mahout 0.13.0 Release Notes

The Apache Mahout PMC is pleased to announce the release of Mahout 0.13.0. Mahout's goal is to create an environment for quickly creating machine-learning applications that scale and run on the highest-performance parallel computation engines available. Mahout comprises an interactive environment and library that support generalized scalable linear algebra and include many modern machine-learning algorithms. This release ships some major changes from 0.12.2 including computation on GPUs and a simplified framework for building new algorithms.

To get started with Apache Mahout 0.13.0, download the release artifacts and signatures from http://www.apache.org/dist/mahout/0.13.0/.

Many thanks to the contributors and committers who were part of this release.

RELEASE HIGHLIGHTS

Mahout-Samsara has implementations for these generalized concepts:

- In-core matrices backed by ViennaCL [3] providing in some cases speedups of an order of magnitude.
- A JavaCPP bridge to native/GPU operations in ViennaCL
- Distributed GPU Matrix-Matrix and Matrix-Vector multiplication on Spark
- Distributed OpenMP Matrix-Matrix and Matrix-Vector multiplication on Spark
- Sparse and dense matrix GPU-backed support.
- Fault tolerance by falling back to Mahout JVM counterpart of new solvers in the case of failure on GPU or OpenMP
- A new scikit-learn-like framework for algorithms with the goal for creating a consistent API for various machine-learning algorithms and an orderly package structure for grouping regression, classification, clustering, and pre-processing algorithms together
- New DRM wrappers in Spark Bindings making it more convenient to create DRMs from MLLib RDDs and DataFrames
- MahoutConversions adds Scala-like compatibility to Vectors introducing methods such as toArray() and toMap()

Mahout has historically focused on highly scalable algorithms, and since moving on from MapReduce-based jobs, Mahout now includes some Mahout-Samsara based implementations:

- Distributed and in-core Stochastic Singular Value Decomposition (SSVD)
- Distributed Principal Component Analysis (PCA)
- Distributed and in-core QR Reduction (QR)
- Distributed Alternating Least Squares (ALS)
- Collaborative Filtering: Item and Row Similarity based on cooccurrence and supporting multimodal user actions
- Distributed Naive Bayes Training and Classification

RELATION TO MACHINE LEARNING LIBRARIES

Library	Similar with Mahout	Difference with Mahout
R	Mathematically expressive language for implementing statistical and machine learning algorithms	Mahout runs on distributed engines- R is designed for in memory on single machine.
Scikit-learn (sklearn)	Fit/Model framework, consistent API, and robust functionality for many popular ML methods **(Mahout seeks to add this capability by 0.14.0)**	Python / sklearn library also are single machine- Mahout is designed for distributed engines
Spark MLLib	Statistics/Machine Learning on distributed engines	Mahout makes it extremely easy for end users to compose new algorithms. MLLib has very limited functionality, and is not easily extensible.
TensorFlow	Linear Algebra focused math in a GPU accelerated distributed environment	TensorFlow will work with Spark, but prefers to manage its own cluster. Mahout is designed to modularly work on any distributed engine if a user writes bindings (which includes defining a Distributed Row Matrix structure and implementing a handful of BLAS operations on that structure in a way native to said engine).

STATS

A total of 62 separate JIRA issues are addressed in this release [1].

GETTING STARTED

Download the release artifacts and signatures at https://mahout.apache.org/general/downloads.html The examples directory contains several working examples of the core functionality available in Mahout. These can be run via scripts in the examples/bin directory. Most examples do not need a Hadoop cluster in order to run.

FUTURE PLANS

0.13.1

As the project moves towards a 0.13.1 release, we are working on the following:

- Further Native Integration for increased speedups
- JCuda backing for In-core Matrices and CUDA solvers
- Enumeration across multiple GPUs per JVM instance on a given instance
- GPU/OpenMP Acceleration for linear solvers
- Further integration with other libraries such as MLLib and SparkML
- Scala 2.11 Support
- Spark 2.x Support
- Incorporate more statistical operations
- Runtime probing and optimization of available hardware for caching of correct/most optimal solver

Post-0.13.1

We already see the need for work in these areas:

- Support for native iterative solvers
- A more robust algorithm library
- Smarter probing and optimization of multiplications

ACKNOWLEDGMENTS

Many thanks to Karl Rupp of the ViennaCL [3] project for his help pushing the bindings through with his many email exchanges; we greatly appreciate his involvement. Many thanks as well to Samuel Audet of the JavaCPP [4] project for his help with the team's usage of JavaCPP to produce the JNI layer to produce the native bindings for GPU and OpenMP, which was also key to this major release.

CONTRIBUTING

If you are interested in contributing, please see our How to Contribute [2] page or contact us via email at dev@mahout.apache.org.

CREDITS

As with every release, we wish to thank all of the users and contributors to Mahout. Please see the and JIRA Release Notes [1] for individual credits, as there are too many to list here.

KNOWN ISSUES:

1. The classify-wikipedia.sh example has an outdated link to the data files. A workaround is to change the download section of the script to: `curl https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles10.xml-p002336425p003046511.bz2 -o ${WORK_DIR}/wikixml/enwiki-latest-pages-articles.xml.bz2`
2. Currently GPU acceleration for supported operations is limited to a single JVM instance
3. Occasional segfault with certain GPU models and computations
4. On older GPUs some tests fail when building ViennaCL due to card limitations
5. Currently automatic probing of a system's hardware happens at each supported operation, adding some overhead

[1]https://issues.apache.org/jira/issues/?jql=project%20%3D%20MAHOUT%20AND%20issuetype%20in%20(standardIssueTypes()%2C%20subTaskIssueTypes())%20AND%20status%20%3D%20Resolved%20AND%20fixVersion%20in%20(0.13.0%2C%200.13.1%2C%201.0.0)
[2]http://mahout.apache.org/developers/how-to-contribute.html
[3]http://viennacl.sourceforge.net/
[4]https://github.com/bytedeco/javacpp