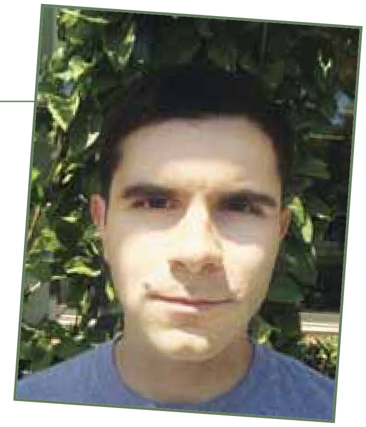
BY REZA BOSAGH ZADEH, PhD



Machine Learning using Big Data: How Apache Spark Can Help

Machine learning is the process of automatically building models from data. In the past two decades, researchers in many fields of study have been generating these models from progressively more data. Because this has led to higher quality learned models, researchers are using even greater quantities of data that require more and more complex distributed computing systems. These systems consist of many hard-drives connected to many machines (CPUs)—often commodity computers to keep costs down. But with many commodity machines come many failures: Hard-drives die; operating systems fail; and someone might trip over a power cord in the data center. The need to problem-solve such single points of failure renders distributed computing quite cumbersome. One solution: Use cleverly designed software to make applications running in clusters more fault-tolerant. Specifically, researchers turn to software known as cluster programming frameworks.

The most successful of these is Apache Spark. Built by the AMPLab at the University of California, Berkeley and now by Databricks, Spark provides users with a distributed array that is fault-tolerant. Many researchers are already accustomed to programming with arrays in their favorite programming language. Spark provides much of the same functionality that arrays provide, with the convenience of the array being seamlessly distributed across a cluster. These arrays are called Resilient Distributed Datasets (RDDs). They can be large and stored on disk, with the portions that are in use swapped in and out of RAM for faster access. Because the generic idea of distributed arrays has nothing to do with any particular programming language, Spark is able to provide clean APIs in Python, Java, Scala, and R.

There are many ways to create RDDs, but the world only lets you create RDDs in ways that can be automatically tracked. The recipe for an RDD is saved along with the RDD, so that in the event of a machine failure, the part for which the machine was responsible can be rebuilt. Called "lineage," this recipe is the primary fault-tolerance mechanism in Spark.

Many researchers are already accustomed to programming with arrays in their favorite programming language. Spark provides much of the same functionality that arrays provide, with the convenience of the array being seamlessly distributed across a cluster.

Given that programming with arrays has been historically successful, it is no surprise that RDDs have also enjoyed fast adoption. Spark provides four libraries out of the box that take advantage of the power of RDDs:

- **ML**: Machine learning algorithms and matrix computations
- **GraphX**: Graph processing library for handling large graphs
- **Streaming**: Handling streams of data (e.g., web logs or stock tickers)
- **Dataframes**: Easy access to tables of heterogeneous data, similar to those found in R and Python

These open-source libraries are developed in a concerted effort across many universities and companies. For example, several Stanford students have worked with me to create the basic building blocks for linear algebra in Spark, such as the singular value decomposition. Only the most widely used and tested algorithms are added to the above libraries. However, there is a vibrant community of people developing Spark packages that can be installed with a single command line. Databricks maintains this package listing at http://spark-packages.org. Together, the Spark ecosystem and its community make Big Data easier to handle. □

DETAILS

Reza Bosagh Zadeh is a Consulting Professor at Stanford University and a Technical Advisor to Databricks. Zadeh received his PhD in Computational Mathematics from Stanford University under the supervision of Gunnar Carlsson. For his PhD work in distributed machine learning, Reza received the Gene Golub Outstanding Thesis Award. During his PhD, Reza built the machine learning algorithms behind Twitter's who-to-follow system, the first product to use machine learning at Twitter. As a Technical Advisor at Databricks, Reza is the initial creator of the linear algebra package in Apache Spark.

DIVERSE DISCIPLINES, ONE COMMUNITY

BiomedicalComputation REVIEW

Published by the Mobilize Center, an NIH Big Data to Knowledge Center of Excellence

Nonprofit Org.
U.S. Postage Paid
Permit No. 28
Palo Alto, CA

Stanford University
318 Campus Drive
Clark Center Room S271
Stanford, CA 94305-5444

seeing science

SeeingScience

BY KATHARINE MILLER

Seeing Inside

The bright light needed to see molecular activity inside a living cell can quickly alter or even halt the very thing scientists want to observe. But a new technique developed by **Eric Betzig, PhD**, Group Leader at the Janelia Research Campus, offers fantastic 3-D resolution of living cells for longer time periods without phototoxicity. Called lattice light-sheet microscopy, the technique uses ultrathin light sheets derived from two-dimensional optical lattices. Rapidly scanned plane-by-plane through the specimen, these light sheets provide excellent illumination with minimal damage to the cell. Betzig, who won the 2014 Nobel Prize for Chemistry for other work, calls lattice light-sheet microscopy "the high-water mark" of his career. □

Lattice light-sheet microscopy allows the imaging of molecules inside living cells. In these screenshots from a film of HeLa cells as they progress through mitosis, chromosomes are shown in orange while the 3-D growth and retraction of microtubules components are shown as points with lines colored according to their velocity. Credit: Betzig Lab, HHMI/Janelia Research Campus; Mimori-Kiyosue Lab, RIKEN Center for Developmental Biology. Reprinted with permission from B-C Chen et al., Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution, Science *346:6208 (2014).*