DZone

RESEARCH

DZONE RESEARCH PRESENTS

# 2014 GUIDE TO

# BIG DATA

BROUGHT TO YOU IN PARTNERSHIP WITH

MAPR

Typesafe

GOGRID

New Relic.

# WELCOME

Dear Reader,

Welcome to our fifth DZone Research Guide and welcome to The Age of Big Data. It's fitting that this guide follows our Internet of Things guide, as by all accounts, IoT is driving the creation of more data than ever before. In the blink of an eye we can fill more storage, in a smaller form factor, than the largest hard drives of 15 years ago. Through the never-ending march of technology and the broad availability of cloud, available storage and computing power is now effectively limitless. The combination of these technologies gives developers and analysts such as ourselves a wealth of new possibilities to draw conclusions from our data and make better business decisions.

Just as our fourth guide was focused around the platforms and devices that are driving this amazing creation of new data, this guide is focused around the tools that developers and architects will use to gather and analyze data more effectively. We've covered a wide spectrum of the tools: from NoSQL databases like MongoDB and Hadoop to business intelligence (BI) tools like Actuate BIRT, down to traditional relational databases like Oracle, MySQL, and PostgreSQL. Gathering the data is easy, it's what you do with it after the fact that makes it interesting.

As you'll find while you read through our findings from nearly 1,000 developers, architects, and executives, Big Data is no longer a passing fad or something that people are just beginning to explore. Nearly 89% of all respondents told us that they are either exploring a Big Data implementation or have already rolled out at least one project. This is amazing growth for an industry that barely existed even 5 years ago. So, welcome to the DZone Big Data Guide, and we hope you enjoy the data and the resources that we've collected.


**MATT SCHMIDT**
CTO, PRESIDENT
research@dzone.com


## TABLE OF CONTENTS



## CREDITS


### DZONE RESEARCH

**JAYASHREE GOPALAKRISHNAN**
DIRECTOR OF RESEARCH

**MITCH PRONSCHINSKE**
SENIOR RESEARCH ANALYST

**BENJAMIN BALL**
RESEARCH ANALYST, AUTHOR

**MATT WERNER**
MARKET RESEARCHER

**JOHN ESPOSITO**
REFCARDZ EDITOR/COORDINATOR

**ALEC NOLLER**
SENIOR CONTENT CURATOR

**MIKE LENTO**
GRAPHIC DESIGNER

Special thanks to our topic experts David Rosenthal, Oren Eini, Otis Gospodnetic, Daniel Bryant, Marvin Froeder, Isaac Sacolick, Arnon Rotem-Gal-Oz, Kate Borger, and our trusted DZone Most Valuable Bloggers for all their help and feedback in making this report a great success.

### DZONE MARKETING AND SALES

**ALEX CRAFTS**
SENIOR ACCOUNT MANAGER

**KELLET ATKINSON**
DIRECTOR OF MARKETING

**ASHLEY SLATE**
DIRECTOR OF DESIGN

**CHRIS SMITH**
PRODUCTION ADVISOR

### DZONE CORPORATE

**RICK ROSS**
CEO

**MATT SCHMIDT**
CTO, PRESIDENT

**BRANDON NOKES**
VP OF OPERATIONS

**HERNÂNI CERQUEIRA**
LEAD SOFTWARE ENGINEER

# THE EVOLUTION OF MAPREDUCE AND HADOOP

*by Srinath Perera & Adam Diaz*

*With its Google pedigree, MapReduce has had a far-ranging impact on the computing industry [1]. It is built on the simple concept of mapping (i.e. filtering and sorting) and then reducing data (i.e. running a formula for summarization), but the true value of MapReduce lies with its ability to run these processes in parallel on commodity servers while balancing disk, CPU, and I/O evenly across each node in a computing cluster. When used alongside a distributed storage architecture, this horizontally scalable system is cheap enough for a fledgling startup. It is also a cost-effective alternative for large organizations that were previously forced to use expensive high-performance computing methods and complicated tools such as MPI (the Message Passing Interface library). With MapReduce, companies no longer need to delete old logs that are ripe with insights—or dump them onto unmanageable tape storage—before they've had a chance to analyze them.*

## HADOOP TAKES OVER

Today, the Apache Hadoop project is the most widely used implementation of MapReduce. It handles all the details required to scale MapReduce operations. The industry support and community contributions have been so strong over the years that Hadoop has become a fully-featured, extensible data-processing platform. There are scores of other open source projects designed specifically to work with Hadoop. Apache Pig and Cascading, for instance, provide high-level languages and abstractions for data manipulation. Apache Hive provides a data warehouse on top of Hadoop.

As the Hadoop ecosystem left the competition behind, companies like Microsoft, who were trying to build their own MapReduce platform, eventually gave up and decided to support Hadoop under the pressure of customer demand [2]. Other tech powerhouses like Netflix, LinkedIn, Facebook, and Yahoo (where the project originated) have been using Hadoop for years. A new Hadoop user in the industry, TRUECar, recently reported having a cost of $0.23 per GB with Hadoop. Before Hadoop, they were spending $19 per GB [3]. Smaller shops looking to keep costs even lower have tried to run virtual Hadoop instances. However, virtualizing Hadoop is the subject of some controversy amongst Hadoop vendors and architects. The cost and performance of virtualized Hadoop is fiercely debated.

Hadoop's strengths are more clearly visible in use cases such as clickstream and server log analytics. Analytics like financial risk scores, sensor-based mechanical failure predictions, and vehicle fleet route analysis are just some of the areas where Hadoop is making an impact. With some of these industries having 60 to 90 day time limits on data retention, Hadoop is unlocking insights that were once extremely difficult to obtain in time. If an organization is allowed to store data longer, the Hadoop File System (HDFS) can save data in its raw, unstructured form while it waits to be processed, just like the NoSQL databases that have broadened our options for managing massive data.

> *We don't really use MapReduce anymore.*
>
> *- Urs Hölzle, Google*

## WHERE MAPREDUCE FALLS SHORT

- It usually doesn't make sense to use Hadoop and MapReduce if you're not dealing with large datasets like high-traffic web logs or clickstreams.
- Joining two large datasets with complex conditions—a problem that has baffled database people for decades—is also difficult for MapReduce.
- Machine learning algorithms such as KMeans and Support Vector Machines (SVM) are often too complex for MapReduce.
- When the map phase generates too many keys (e.g. taking the cross product of two datasets), then the mapping phase will take a very long time.
- If processing is highly stateful (e.g. evaluating a state machine), MapReduce won't be as efficient.

As the software industry starts to encounter these harder use cases, MapReduce will not be the right tool for the job, but Hadoop might be.

## HADOOP ADAPTS

Long before Google's dropping of MapReduce, software vendors and communities were building new technologies to handle some of the technologies described above. The Hadoop project

*With YARN, developers can run a variety of jobs in a YARN container. Instead of scheduling the jobs, the whole YARN container is scheduled.*

made significant changes just last year and now has a cluster resource management platform called YARN that allows developers to use many other non-MapReduce technologies on top of it. The Hadoop project contributors were already thinking about a resource manager for Hadoop back in early 2008 [4].

With YARN, developers can run a variety of jobs in a YARN container. Instead of scheduling the jobs, the whole YARN container is scheduled. The code inside that container can be any normal programming construct, so MapReduce is just one of many application types that Hadoop can harness. Even the MPI library from the pre-MapReduce days can run on Hadoop. The number of products and projects that the YARN ecosystem enables is too large to list here, but this table will give you an idea of the wide ranging capabilities YARN can support:

| CATEGORY | PROJECT |
|---|---|
| Search | Solr, Elasticsearch |
| NoSQL | HBase, Accumulo |
| Streaming | Storm, Spark Streaming |
| In-Memory | Impala, Spark |
| Proprietary Apps and Vendors | Microsoft, SAS, SAP, Informatica, HP etc. |

## THREE WAYS TO START USING YARN

Below are three basic options for using YARN (but not the only options). The complexity decreases as you go down the list but the granular control over the project also decreases:

1. Directly code a YARN application master to create a YARN application. This will give you more control over the behavior of the application, but it will be the most challenging to program.
2. Use Apache Tez, which has a number of features including more complex directed acyclic graphs than MapReduce, Tez sessions, and the ability to express data processing flows through a simple Java API.
3. Use Apache Slider, which provides a client to submit JAR files for launching work on YARN-based clusters. Slider provides the least programmatic control out of these three options, but it also has the lowest cost of entry for trying out new code on YARN because it provides a ready to use application master.

For organizations migrating from Hadoop 1.x (pre-YARN) to Hadoop 2, the migration shouldn't be too difficult since the APIs are fully compatible between the two versions. Most legacy code should just work, but in certain very specific cases custom source code may need to simply be recompiled against newer Hadoop 2 JARs.
As you saw in the table, there are plenty of technologies that take full advantage of the YARN model to expand Hadoop's analysis capabilities far beyond the limits of the original Hadoop. Apache Tez greatly improves Hive query times. Cloudera's Impala project is a massively parallel processing (MPP) SQL query engine. And then there's Apache Spark, which is close to doubling its contributors in less than a year [5].

## APACHE SPARK STARTS A FIRE

Spark is built specifically for YARN. In addition to supporting MapReduce, Spark lets you point to a large dataset and define a virtual variable to represent the large dataset. Then you can apply functions to each element in the dataset and create a new dataset. So you can pick the right functions for the right kinds of data manipulation. But that's not even the best part.

*Apache Spark is close to doubling its contributors in less than a year.*

The real power of Spark comes from performing operations on top of virtual variables. Virtual variables enable data flow optimization across one execution step to the other, and they should optimize common data processing challenges (e.g. cascading tasks and iterations). Spark streaming uses a technology called "micro-batching" while Storm uses an event driven system to analyze data.

## JUST ONE TOOL IN THE TOOLBOX

MapReduce's main strength is simplicity. When it first emerged in the software industry, it was widely adopted and soon became synonymous with Big Data, along with Hadoop. Hadoop is still the toolbox most commonly associated with Big Data, but now more organizations are realizing that MapReduce is not always the best tool in the box.


[1] http://research.google.com/archive/mapreduce.html
[2] http://www.zdnet.com/blog/microsoft/microsoft-drops-dryad-puts-its-big-data-bets-on-hadoop/11226
[3] http://blogs.wsj.com/cio/2014/06/04/hadoop-hits-the-big-time/
[4] https://issues.apache.org/jira/browse/MAPREDUCE-279
[5] http://inside-bigdata.com/2014/07/15/theres-spark-theres-fire-state-apache-spark-2014/



WRITTEN BY
**Adam Diaz**

Adam Diaz is a Hadoop Architect at Teradata. He has previously worked at big data powerhouses like IBM, SAS, and HortonWorks.

WRITTEN BY
**Srinath Perera**

Srinath Perera is a Research Director and architect at WSO2. He is a member of the Apache Software foundation, a PMC member of Apache Web Service project, a committer on Apache Axis, Axis2, and Geronimo, and a co-founder of Apache Axis2.