Global Knowledge®

Expert Reference Series of White Papers

Learning How to Learn Hadoop

Learning How to Learn Hadoop

Rich Morrow, IT Consultant, Developer, System Administrator, Trainer, Mentor, and Team Builder

Introduction

Hadoop's Value Proposition

Learning how to program and develop for the Hadoop platform can lead to lucrative new career opportunities in Big Data. But like the problems it solves, the Hadoop framework can be quite complex and challenging. Join Global Knowledge instructor and Technology Consultant Rich Morrow as he leads you through some of the hurdles and pitfalls students encounter on the Hadoop learning path. Building a strong foundation, leveraging online resources, and focusing on the basics with professional training can help neophytes across the Hadoop finish line.

If I've learned one thing in two decades of IT, it's that the learning never ends.

As the leader of an independent consulting firm, one of the most important decisions I make throughout the year is choosing what technologies I and other consultants need to learn. If we can identify and quickly ramp up on technologies that really move the needle for our clients, then everyone wins. In the following pages, I'd like to walk you through the path that I took in identifying Hadoop as a "must have" skill that our clients will need, and how I quickly got ramped up on the technology.

Hadoop is a paradigm-shifting technology that lets you do things you could not do before – namely compile and analyze vast stores of data that your business has collected. "What would you want to analyze?" you may ask. How about customer click and/or buying patterns? How about buying recommendations? How about personalized ad targeting, or more efficient use of marketing dollars?

From a business perspective, Hadoop is often used to build deeper relationships with external customers, providing them with valuable features like recommendations, fraud detection, and social graph analysis. In-house, Hadoop is used for log analysis, data mining, image processing, extract-transform-load (ETL), network monitoring – anywhere you'd want to process gigabytes, terabytes, or petabytes of data.

Hadoop allows businesses to find answers to questions they didn't even know how to ask, providing insights into daily operations, driving new product ideas, or putting compelling recommendations and/or advertisements in front of consumers who are likely to buy.

The fact that Hadoop can do all the above is not the compelling argument for it's use. Other technologies have been around for a long, long while which can and do address everything we've listed so far. What makes Hadoop shine, however, is that it performs these tasks in minutes or hours, for little or no cost versus the days or weeks and substantial costs (licensing, product, specialized hardware) of previous solutions.

Hadoop does this by abstracting out all of the difficult work in analyzing large data sets, performing its work on commodity hardware, and scaling linearly. -- Add twice as many worker nodes, and your processing will generally complete 2 times faster. With datasets growing larger and larger, Hadoop has become the solitary solution businesses turn to when they need fast, reliable processing of large, growing data sets for little cost.

Because it needs only commodity hardware to operate, Hadoop also works incredibly well with public cloud infrastructure. Spin up a large cluster only when you need to, then turn everything off once the analysis is done. Some big success stories here are The New York Times using Hadoop to convert about 4 million entities to PDF in just under 36 hours, or the infamous story of Pete Warden using it to analyze 220 million Facebook profiles, in just under 11 hours for a total cost of $100. In the hands of a business-savvy technologist, Hadoop makes the impossible look trivial.

The Pillars of Hadoop - HDFS and MapReduce

Architecturally, Hadoop is just the combination of two technologies: the Hadoop Distributed File System (HDFS) that provides storage, and the MapReduce programming model, which provides processing[1] [2].

HDFS exists to split, distribute, and manage chunks of the overall data set, which could be a single file or a directory full of files. These chunks of data are pre-loaded onto the worker nodes, which later process them in the MapReduce phase. By having the data local at process time, HDFS saves all of the headache and inefficiency of shuffling data back and forth across the network.

In the MapReduce phase, each worker node spins up one or more tasks (which can either be Map or Reduce). Map tasks are assigned based on data locality, if at all possible. A Map task will be assigned to the worker node where the data resides. Reduce tasks (which are optional) then typically aggregate the output of all of the dozens, hundreds, or thousands of map tasks, and produce final output.

The Map and Reduce programs are where your specific logic lies, and seasoned programmers will immediately recognize **Map** as a common built-in function or data type in many languages, for example, **map(function,iterable)** in Python, or **array_map(callback, array)** in PHP. All map does is run a user-defined function (your logic) on every element of a given array. For example, we could define a function **squareMe,** which does nothing but return the square of a number. We could then pass an array of numbers to a map call, telling it to run **squareMe** on each. So an input array of (2,3,4,5) would return (4,9,16,25), and our call would look like (in Python) **map("squareMe",array('i',[2,3,4,5])**.

Hadoop will parse the data in HDFS into user-defined keys and values, and each key and value will then be passed to your Mapper code. In the case of image processing, each value may be the binary contents of your image file, and your Mapper may simply run a user-defined **convertToPdf** function against each file. In this case, you wouldn't even need a Reducer, as the Mappers would simply write out the PDF file to some datastore (like HDFS or S3).This is what the New York Times did when converting their archives.

Consider, however, if you wished to count the occurrences of a list of "good/bad" keywords in all customer chat sessions, twitter feeds, public Facebook posts, and/or e-mails in order to gauge customer satisfaction. Your good list may look like happy, appreciate, "great job", awesome, etc., while your bad list may look like unhappy, angry, mad, horrible, etc., and your total data set of all chat sessions and emails may be hundreds of GB. In this case, each Mapper would work only on a **subset** of that overall data, and the Reducer would be used to compile the final count, summing up outputs of all the Map tasks.

At its core, Hadoop is really that simple. It takes care of all the underlying complexity, making sure that each record is processed, that the overall job runs quickly, and that failure of any individual task (or hardware/network failure) is handled gracefully. You simply bring your Map (and optionally Reduce) logic, and Hadoop processes every record in your dataset with that logic.

Solid Linux and Java Will Speed Your Success

That simplicity can be deceiving though, and if you're assigned a project involving Hadoop you may find yourself jumping a lot of hurdles that you didn't expect or foresee.

Hadoop is written in Java and is optimized to run Map and Reduce tasks that were written in Java as well. If your Java is rusty, you may want to spend a few hours with your *Java for Dummies* book before you even begin looking at Hadoop. Although Java familiarity also implies good coding practices (especially Object-Oriented Design (OOD) coding practices), you may want to additionally brush up on your Object Oriented skills and have a clear understanding of concepts like Interfaces, Abstract Objects, Static Methods, and Variables, etc. Weekend afternoons at Barnes and Noble, or brownbag sessions with other developers at work are the quickest ways I know to come up to speed on topics like this.

Although it does offer a Streaming API, which allows you to write your basic Map and Reduce code in any language of your choice, you'll find that most code examples and supporting packages are Java-based, that deeper development (such as writing Partitioners) still requires Java, and that your Streaming API code will run up to 25% slower than a Java implementation.

Although Hadoop can run on Windows, it was built initially on Linux, and the preferred method for both installing and managing Hadoop. The Cloudera Distribution of Hadoop (CDH), is only officially supported on Linux derivatives like Ubuntu ® and RedHat ®. Having a solid understanding of getting around in a Linux shell will also help you tremendously in digesting Hadoop, especially with regards to many of the HDFS command line parameters. Again, a Linux for Dummies book will probably be all you need.

Once you're comfortable in Java, Linux, and good OOD coding practices, the next logical step would be getting your hands dirty by either installing a CDH Virtual Machine, or a CDH distribution. When you install a Cloudera-provided CDH distribution of Hadoop, you're getting assurance that some of the best minds in the Hadoop community have carefully reviewed and chosen security, functionality, and supporting patches; tested them together; and provided a working, easy-to-install package.

Although you can install Hadoop from scratch, it is both a daunting and unnecessary task that could burn up several weeks of your time. Instead, download either a local virtual machine (VM), which you can run on your workstation, or install the CDH package (CDH4 is the latest) for your platform – either of which will only take 10 minutes. If you're running in a local VM, you can run the full Hadoop stack in pseudo-distributed mode, which basically mimics the operation of a real production cluster right on your workstation. This is fantastic for jumping right in and exploring.

Using Hadoop Like a Boss

Once you're doing real development, you'll want to get into the habit of using smaller, test datasets on your local machine, and running your code iteratively in **Local Jobrunner Mode** (which lets you locally test and debug your Map and Reduce code); then **Pseudo-Distributed Mode** (which more closely mimics the production environment); then finally Fully-Distributed Mode (your real production cluster). By doing this iterative development, you'll be able to get bugs worked out on smaller subsets of the data so that when you run on your full dataset with real production resources, you'll have all the kinks worked out, and your job won't crash three-quarters of the way in.

Remember that in Hadoop, your Map (and possibly Reduce) code will be running on dozens, hundreds, or thousands of nodes. Any bugs or inefficiencies will get amplified in the production environment. In addition to performing iterative **Local,Pseudo,Full** development with increasingly larger subsets of test data, you'll also want to code defensively, making heavy use of try/catch blocks, and gracefully handling malformed or missing data (which you're sure to).

Chances are also very high that once you or others in your company come across Pig or Hive, that you'll never write another line of Java again. Pig and Hive represent two different approaches to the same issue: that writing good Java code to run on Map Reduce is hard and unfamiliar to many. What these two supporting products provide are simplified interfaces into the MapReduce paradigm, making the power of Hadoop accessible to non-developers.

In the case of Hive, a SQL-like language called HiveQL provides this interface. Users simply submit HiveQL queries like **SELECT * FROM SALES WHERE amount > 100 AND region = 'US'**, and Hive will translate that query into one or more MapReduce jobs, submit those jobs to your Hadoop cluster, and return results. Hive was heavily influenced by MySQL, and those familiar with that database will be quite at ease with HiveQL.

Pig takes a very similar approach, using a high-level programming language called PigLatin, which contains familiar constructs such as **FOREACH**, as well as arithmetic, comparison, and boolean comparators, and SQL-like **MIN, MAX, JOIN** operations. When users run a PigLatin program, Pig converts the code into one or more MapReduce jobs and submits it to the Hadoop cluster, the same as Hive.

What these two interfaces have in common is that they are incredibly easy to use, and they both create highly optimized MapReduce jobs, often running even faster than similar code developed in a non-Java language via the Streaming API.

If you're not a developer, or you don't want to write your own Java code, mastery of Pig and Hive is probably where you want to spend your time and training budgets. Because of the value they provide, it's believed that the vast majority of Hadoop jobs are actually Pig or Hive jobs, even in such technology-savvy companies as Facebook.

It's next to impossible, in just a few pages, to both give a good introduction to Hadoop as well as a good path to successfully learning how to use it. I hope I've done justice to the latter, if not the former. As you dig deeper into the Hadoop ecosystem, you'll quickly trip across some other supporting products like Flume, Sqoop, Oozie, and ZooKeeper, which we didn't have time to mention here. To help in your Hadoop journey, we've included several reference resources, probably the most important of which is *Hadoop, the Definitive Guide, 3rd edition*, by Tom White. This is an excellent resource to flesh out all of the topics we've introduced here, and a must-have book if you expect to deploy Hadoop in production.

Applying the Knowledge at Work

At this point, your education could take one of many routes, and if at all possible consider an official Cloudera training class. With a powerful technology such as Hadoop, you want to make sure you get the essentials and basics down as soon as possible, and taking a course will quickly pay for itself by helping you avoid costly mistakes as well as introducing you to concepts and ideas that would've taken you months to learn on your own. There is simply no substitute for having dedicated time with a domain expert from whom you can ask questions and get clarifications.

If training is not available to you, probably the next best way to learn is by giving yourself a real-world task, or even better, by getting company approval to use Hadoop at work. Some potential tasks you may want to look at are: counting and ranking the number of interactions (e-mails, chat sessions, etc.) per customer agent; crawling weblogs looking for errors, or common "drop off" pages; building search indices for large document stores; or monitoring social media channels for brand sentiment. The only requirement of an initial project is that it should be relatively simple and low-risk. You'll want to take baby steps before tackling harder tasks.

As you grow to understand and appreciate the power of Hadoop, you'll be uniquely positioned to identify opportunities for its use inside of your business. You may find it useful to initiate meetings with gatekeepers or executives inside your business in order to help them understand and leverage Hadoop on data that may be just sitting around unused. Many businesses see a 3 percent to 5 percent bump in sales after implementing a recommendation engine alone.

Whatever tasks you decide to tackle in Hadoop, you'll also find that there are abundant, easy-to-find code walkthroughs online. A great example is the Marcello DeSales walkthrough of a TF-IDF implementation. You'll find that Hadoop is a very Google-friendly term, and has almost no usage outside of the technology realm (contrast that with "Chef" or "Puppet" where you'll quickly get lost in the noise). If you live in or near a major metropolitan area, there's also a good chance that there are one or more Big Data Meetups in your area where others meet and share their expertise in Hadoop and other technologies. These presentations often contain bits of gold in the way of problem avoidance and optimization strategies.

If you're stuck for ideas, or still confused about some concepts in Hadoop, you'll find that Cloudera produces some excellent (and free) online web courses at Cloudera University.

Conclusion – Step Back To Move Forward

With a product as deep and wide as Hadoop, time spent making sure you understand the foundation will more than pay for itself when you get to higher level concepts and supporting packages. Although it may be frustrating and/or humbling to go back and re-read a Linux or Java "Dummies" book, you'll be well rewarded once you inevitably encounter some bizarre behavior even in a Pig or Hive query, and you need to look under the hood to debug and resolve the issue.

Whether you choose formal training, on the job training, or just slogging through code examples you find on the Web, make sure you have a firm foundation in what Hadoop does and how it does it.

Appendix

[1] HDFS and MapReduce were heavily influenced by two papers that Google published: *Google File System (GFS)* in 2003, and *MapReduce* in 2004.

[2] Hadoop can run with another file system, but most deployments use HDFS.

Further Reading

White, Tom. Hadoop, The Definitive Guide, 3rd ed.

Lam, Chuck. Hadoop in Action.

Gates Alan. Programing Pig.

Capriolo, Wampler, Rutherglen. Programming Hive.

Online Resources

Global Knowledge Cloudera Certified Training

Cloudera University

Cloudera VM & Linux Package CDH downloads

Apache Hive Project

Apache Pig Project

Interesting Hadoop/BigData Use Cases

10 Ways Companies Are Using Hadoop
Crunching 215 Million Facebook Profiles in 11 Hours for $100

New York Times Converts 4 Million Documents in 36 Hours

Target Uses Big Data to Tell When a Woman Is Pregnant & Mail her Coupons

Excellent Overview of BigData & Hadoop Ecosystem

Learn More

To learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge, Global Knowledge suggests the following courses:

Cloudera Developer Training for Apache Hadoop

Cloudera Administrator Training for Apache Hadoop

Cloudera Introduction to Data Science: Building Recommender Systems

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Rich Morrow brings two decades of experience in IT as a Developer, System Administrator, Trainer, Mentor, and Team Builder. As a consultant, Rich is focused on development and maintenance of large-scale mission-critical custom Web applications, and particularly those leveraging LAMP. Coming from a startup-centric, Development Operations background, Rich has a special appreciation and love for enabling technologies like Cloud and Big Data.