Data Science Workflows Using R and Spark

E. James (Jim) Harner

Department of Statistics
Department of Management Information Systems
West Virginia University

August 29, 2018

Outline

The Components of this Tutorial

This tutorial covers data science workflows using R as both an analysis and graphics engine and as an interface to databases, Hadoop, Spark, etc.

In following are the required components of this tutorial:

- the DSAA tutorial slides (these slides)
- the tutorial notebook content
- the `rspark` computational environment
- the `rspark-docker` images

These components are discussed in the next few slides.

The R Computational Environment

The primary language used in this tutorial is R, but a working knowledge of bash scripts and SQL is useful. You should download and install R if you have not already done so. Go to:
`https://www.r-project.org`

Likewise, you should install the RStudio IDE, which is found here:
`https://www.rstudio.com`

The DSAA Tutorial Beamer Slides

The tutorial slides provide an overview of the tutorial content. The GitHub repo for the slides can be found here: https://github.com/jharner/DSAA2018rspark-tutorial

Go to the site and click on the green button to clone the repo to your computer. Alternatively, clone this repo by issuing the following command from a terminal:
`git clone https://github.com/jharner/DSAA2018rspark-tutorial.git`

These slides are not the main source of content for this tutorial The R markdown notebook documents in the next slide provide the detailed, execuatable content.

The Tutorial Notebook Content

The interactive, executable content of this tutorial is available in a GitHub repo called rspark-tutorial found at:
`https://github.com/jharner/rspark-tutorial`

As before, go to the site and click on the green button to clone the repo to your computer. You can also clone this repo with the following command:
`git clone`
`https://github.com/jharner/rspark-tutorial.git`

The `rspark-tutorial` consists of executable R markdown documents organized into modules containing sections. These tutorial documents are executed within `rspark`.

The rspark Computational Environment

This `rspark-tutorial` local repo can then be imported into the computional environment used in this tutorial, which is called rspark.

The `rspark` computational environment is available in several GitHub repos depending on whether you want to built the environment from scratch or you you want to download pre-built images. Assuming the latter, go to the rspark-docker repo:
`https://github.com/jharner/rspark-docker`

Go to the site and click on the green button to clone the repo to your computer. Alternatively, clone this repo by issuing the following command from a terminal:
`git clone`
`https://github.com/jharner/rspark-docker.git`.

Running rspark

In order to execute the content in the `rspark-tutorial`, the `rspark` computational environment must be run as a web applicaton. Follow the directions in the `rspark-docker` to launch `rspark`.

Once `rspark` is running, import the `rspark-tutorial` by clicking on the `Files` tab in RStudio server. The `rspark-tutorial` directory must be zipped before being imported. Click on the `Upload` option under the `Files` tab and navigate to the `rspark-tutorial.zip` file and upload it.

You can now execute the R markdown documents, i.e., those with the `.Rmd` suffix. These files can be executed interatively as notebooks or knitted to html, pdf, or Word documents.

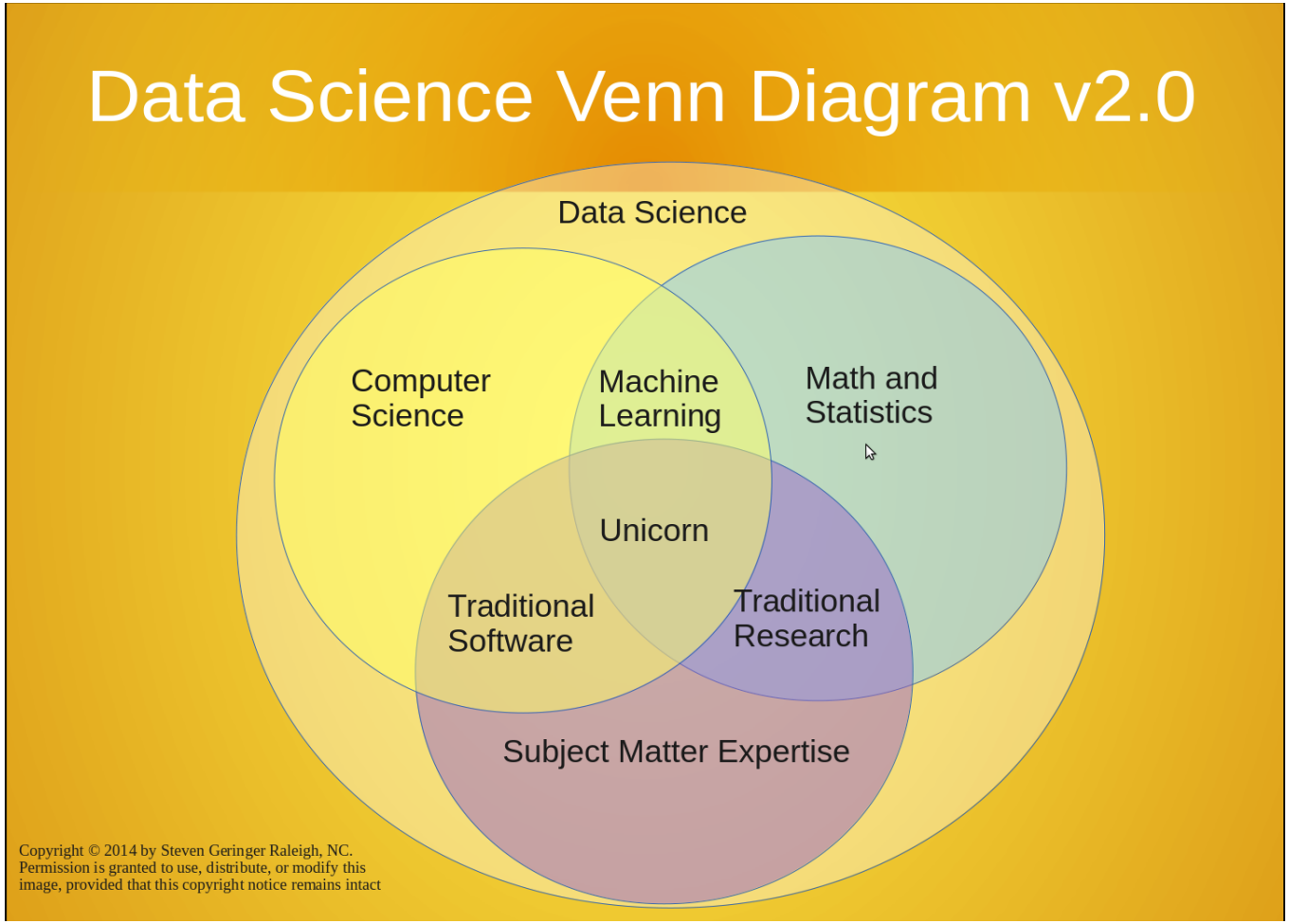
Running the DSAA Tutorial Slides

The slides cannot be executed in `rspark`'s `rstudio` docker container since the full publishing capability of RStudio is not supported. In particular the LaTeXBeamer package (and other components of LaTeX) was not installed to keep the container size reasonable.

As a result, the `DSAA2018rspark-tutorial` repo must be executeds within a local version of RStudio.

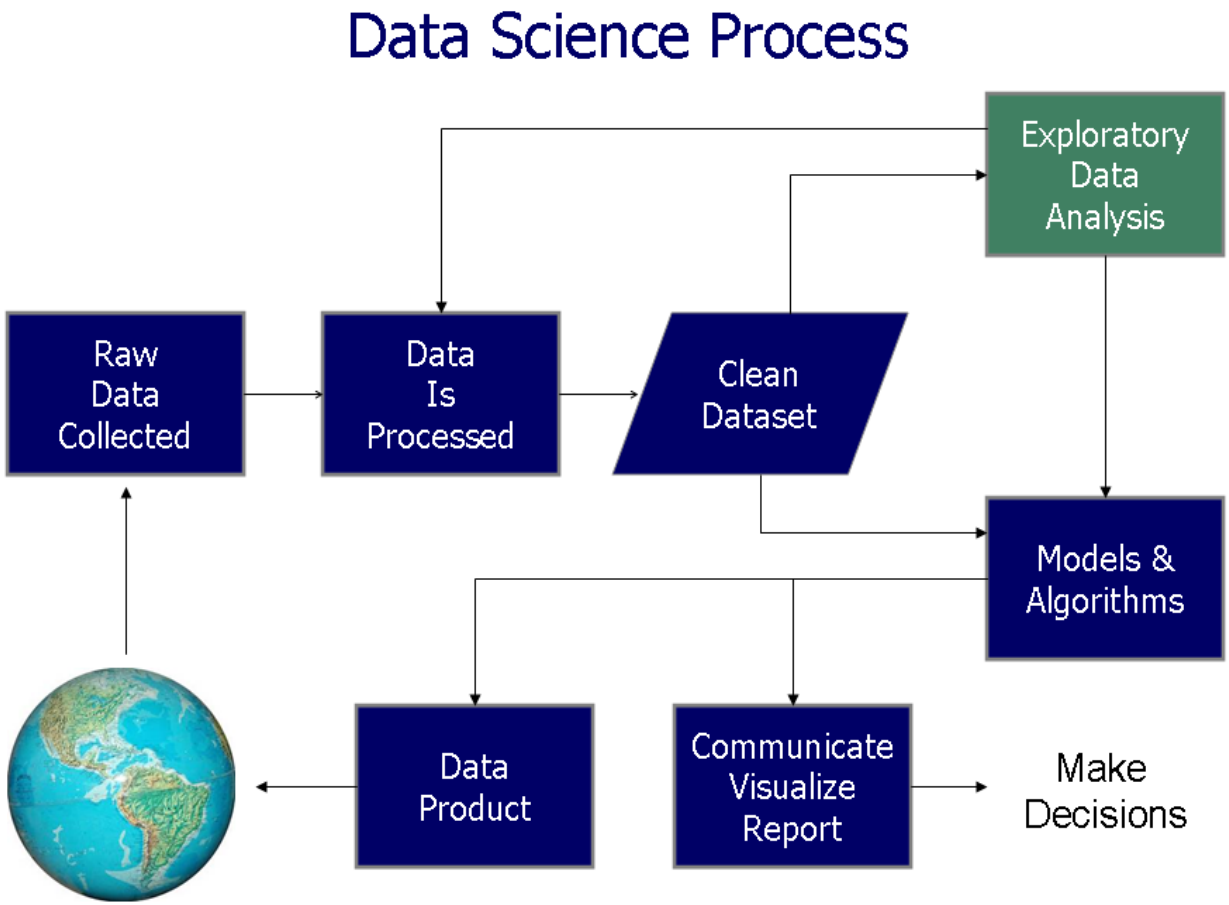
What is Data Science?

Data science combines elements of statistics and computer science to develop methodologies to analyze large, complex data and streaming data within various subject-matter areas.



Data Science Process

The data science process is a workflow from data extraction to data products:



What is a Data Scientist?

Josh Wills:

Data Scientist (n.) Person who is better at statistics than any software engineer and better at software engineering than any statistician.

Programming Languages

R A language and environment for statistical computing and graphics.

Python An interpreted, high-level programming language widely used in data science.

JavaScript A high-level, dynamic programming language for creating interactive web pages and graphics.

bash A processor that runs UNIX (Linux) commands interactively or from shell scrips.

SAS A propriety programming language for data management, statistical analyses, and advanced analytics.

Why I like R!

Hadley Wickham's Advanced R provides a lot of reasons.

My favoriate features:

Functional Programming: Assign functions to variables, store them in lists, pass them as arguments to other functions, create them inside functions, and even return them as the result of a function. (See Newton's method example.)

Multiple Object Oriented (OO) Systems: S3 (generic function OO); S4 (formal class definitions), etc.

Rich Data Structures and Workflows Vectors, data frames, etc.

Domain Specific Languages: A powerful toolkit for creating embedded domain specific languages (DSLs), e.g., HTML, LaTeX, R formulas, SQL, etc.

R Code

R code

embedded within Latex Beamer.

```
> newton_search <- function(f, df, guess, conv=0.001) {
+ # Note: If f does not have a root, we could be in an infi
+   improve <- function(guess, f, df) {
+     guess - f(guess) / df(guess)
+   }
+   while(abs(f(guess)) > conv) {
+     guess <- improve(guess, f, df)
+   }
+   guess
+ }
> newton_search(f = sin, df = cos, guess = 3)
```

```
[1] 3.142547
```

Data Wrangling

Garrett Grolemund's and Hadley Wickham's R for Data Science is being developed to look at data:

Import Import data from text files, the Web (curl), SQL databases, NoSQL databases, etc.

Tidying Match the semantics of a dataset with how it is stored.

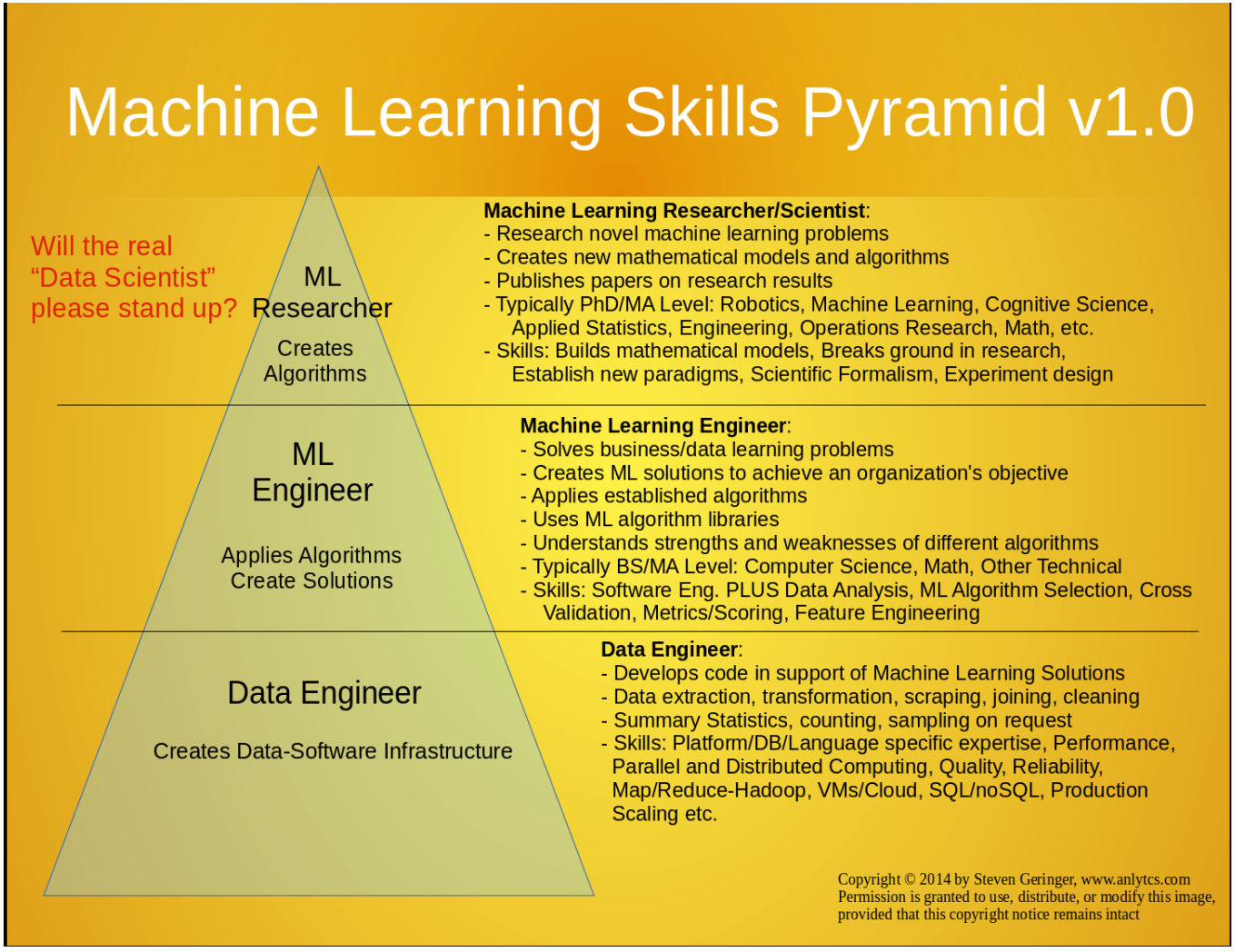
Transformation Subset data, create new variables, summarize data, etc.

Visualization

Modeling

Communication

Machine Learning



Machine Learning Algorithms

Machine learning algorithms are largely used to predict, classify, or cluster.

- Prediction and classification are examples of **supervised learning**, whereas
- clustering is an example of **unsupervised learning**.

Put another way, supervised learning is concerned with problems that have a measurable (or labeled) response variable and unsupervised learning is concerned with problems without a response variable.

- **Statistical modeling** is done (primarily) to infer the underlying generative process.
- **Machine learning algorithms** are used to predict or classify with the most accuracy. They form the basis of data products.

Reproducible Research

Reproducible research is the idea that data analyses and, more generally, scientific claims, are published with their data and software code so that others may verify the findings and build upon them. (Johns Hopkins University/ Coursera)

Reproducible research can be done using R markdown documents or R notebooks. R markdown allows LaTeX equations and R code to be embedded in markdown documents. In addition, other languages are also supported, e.g., Python and bash engines.

Open Source Big Data Architectures

What data architecture is needed for big data analytics?

HDFS/Hadoop A software framework for distributed storage (HDFS) and distributed processing (MapReduce).

Alluxio/Spark A cluster computing environment using in-memory primitives rather than Hadoop's two-stage, disk-based MapReduce approach.

Flink A streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams.

How do we access these big data processing architectures? From an R perspective: RStudio.

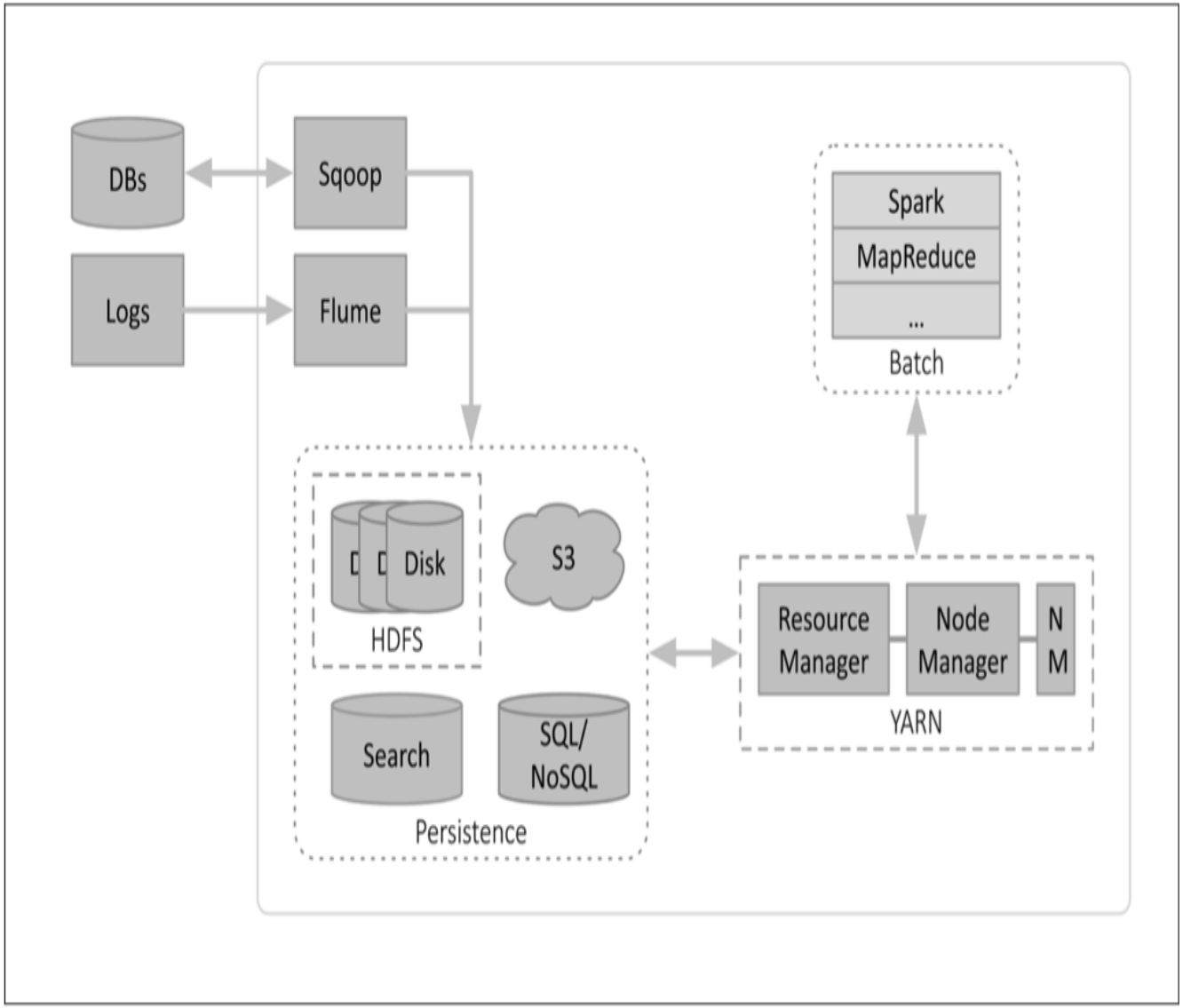
HDFS/ Hadoop

This demo illustrates MapReduce run in a dockerized container environment. Two containers are created here—one for RStudio Server and one for Hadoop.

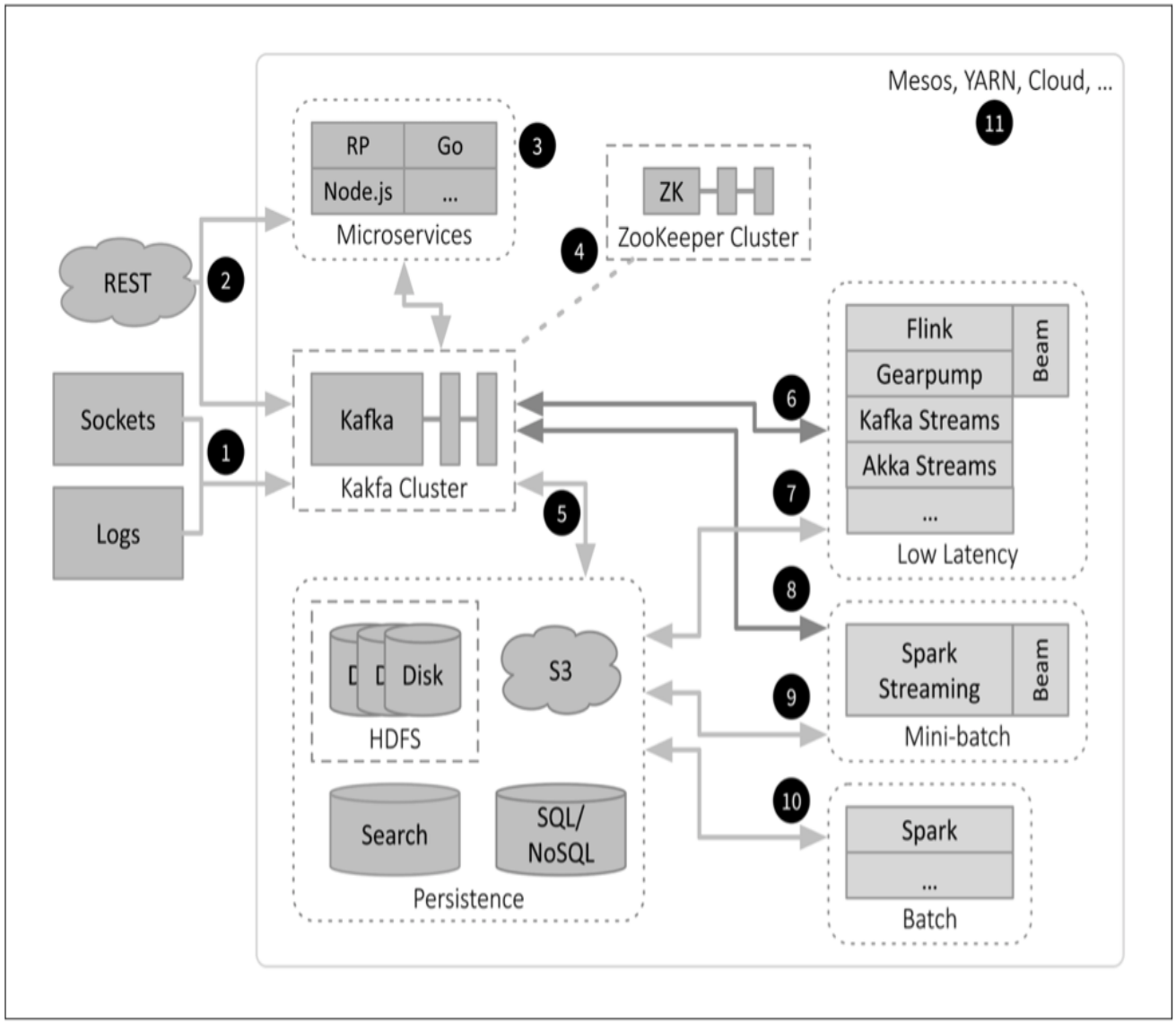
RHadoop Demo

It would be possible to create a virtual cluster (even on your laptop) having a master and worker nodes using Docker containers. These containers could be managed by kubernetes or Mesos.

Batch Architecture



Streaming Architecture



YARN/ Spark

Spark can be run in a general way by downloading the Spark distribution. This allow Spark to be run using Python, R, the command line, etc.

Spark can also be run directly in RStudio using the sparklyr package. This is the main way Spark will be run in this program.

For example, see:
Regression
Regularization
Slump Regressopm

RStudio Server

RStudio is a powerful, open-source IDE for R.

RStudio Server

- provides a productive, web-based user interface for R;
- deploys on Linux platforms;
- supports both Sweave and R Markdown;
- supports interactive web application development using Shiny and Shiny Server;
- allows collaboration among team members (Prp version only).

IDE Features

As an IDE, RStudio:

- supports syntax highlighting, code completion, and smart indentation;
- allows code go be directly executed from the source editor;
- supports integrated R help;
- has a workspace browser;
- has an interactive debugger allowing the developer to find and fix errors quickly;
- has extensive support for developing packages

Projects

RStudio allows the creation of projects.

RStudio projects can be created:

- in a new directory;
- from an existing directory containing R code and data;
- from a version control Git or Subversion directory.

RStudio has support for multiple simultaneous projects.

Version control allows the coordination of team work and benefits individual work.

Projects

RStudio allows the creation of projects.

RStudio projects can be created:

- in a new directory;
- from an existing directory containing R code and data;
- from a version control Git or Subversion directory.

RStudio has support for multiple simultaneous projects.

Version control allows the coordination of team work and benefits individual work.

Git/ GitHub

RStudio internally supports Git and GitHub. Git is a distributed version control system for tracking changes in code or text. It allows multiple creators to coordinate their work on the files in the git repository (repo).

GitHub is a web-based Git repository and Internet hosting service.

I have a repo for building a virtual R/ Hadoop/ Spark cluster environment which mimics the BUDA cluster.