Case Study: Implementing a Big Data Analysis System on Cloud from User Interface to Spark Cluster

Christian Alameda, Joshua Gonzalez-Leon, Ray Duenas (Advisor: Dr. Daehee Kim)
Computer Science, California State University Stanislaus

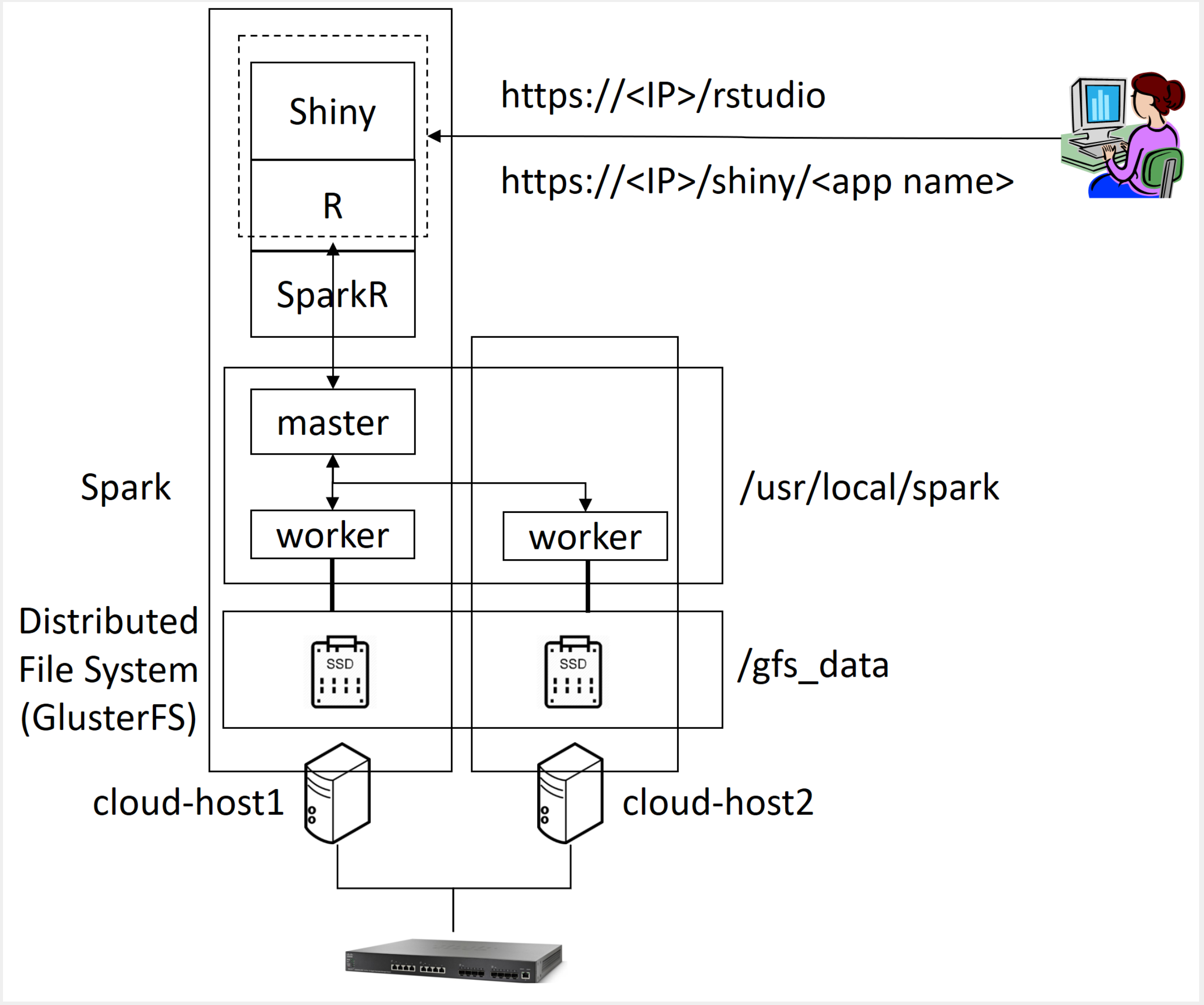
Abstract

Big Data Analysis has enabled the yielding of such great insights on relationships that it has become an essential tool finding implementation in social networking, business, and the many fields of sciences. There is an overwhelming amount of information on how to conduct Big Data analysis, this work aims to provide a concise demonstration on how one may gain new knowledge utilizing Big Data. We introduce the implementation instructional materials to construct a comprehensive big data analytics infrastructure.

Introduction

We deployed components of Big Data Analysis systems by subscribing Google cloud resources [1], by learning data visualization with Shiny application [2] and integrated development environment with RStudio server [3], by deploying a distributed file system with Gluster File System [4] and big data analysis engine with Spark cluster [5], by utilizing NoSQL Mongo database [6] for the performance increase, and by connecting front user interface with back-end server with SparkR [9] that allows search with SQL.

System Architecture



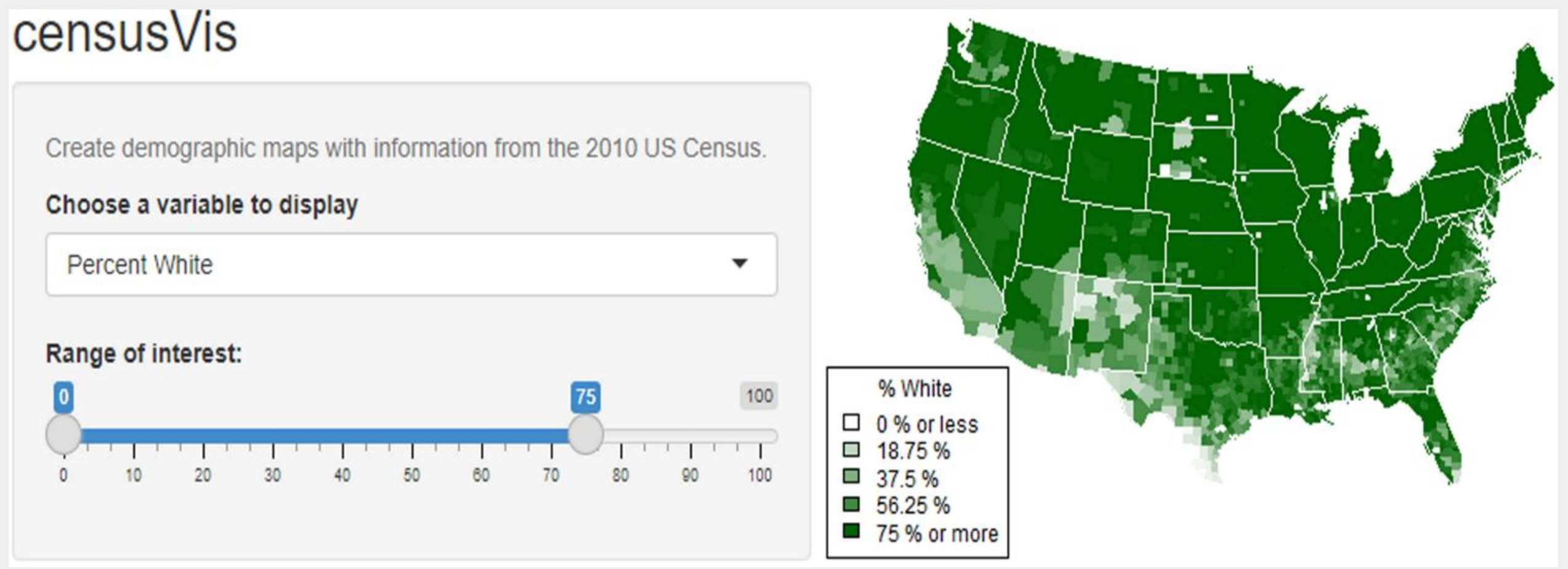
Ubuntu 20.04 LTS Linux Virtual Machine
x86/64 version, amd64, 1 vCPU, 1.7 GB memory, 10 GB disk size, Allow HTTP/HTTPS traffic

Cloud

We prepared the Google Cloud servers [1] to run analysis on the Internet. The servers are accessed through Secure Hypertext Transfer Protocol (HTTPS). We started with Linux Virtual Machine cloud instances with minimum specifications. People can build up their own system with a little cost, and the system can be scaled out with high performance instances if needed.

Data Visualization

We utilized the Shiny (R package) [2] that allows for the creation of front-end web-based graphic user interfaces. Through the use of Shiny input/output widgets, a user may change settings and input arguments which gets processed and displayed to the user.


censusVis

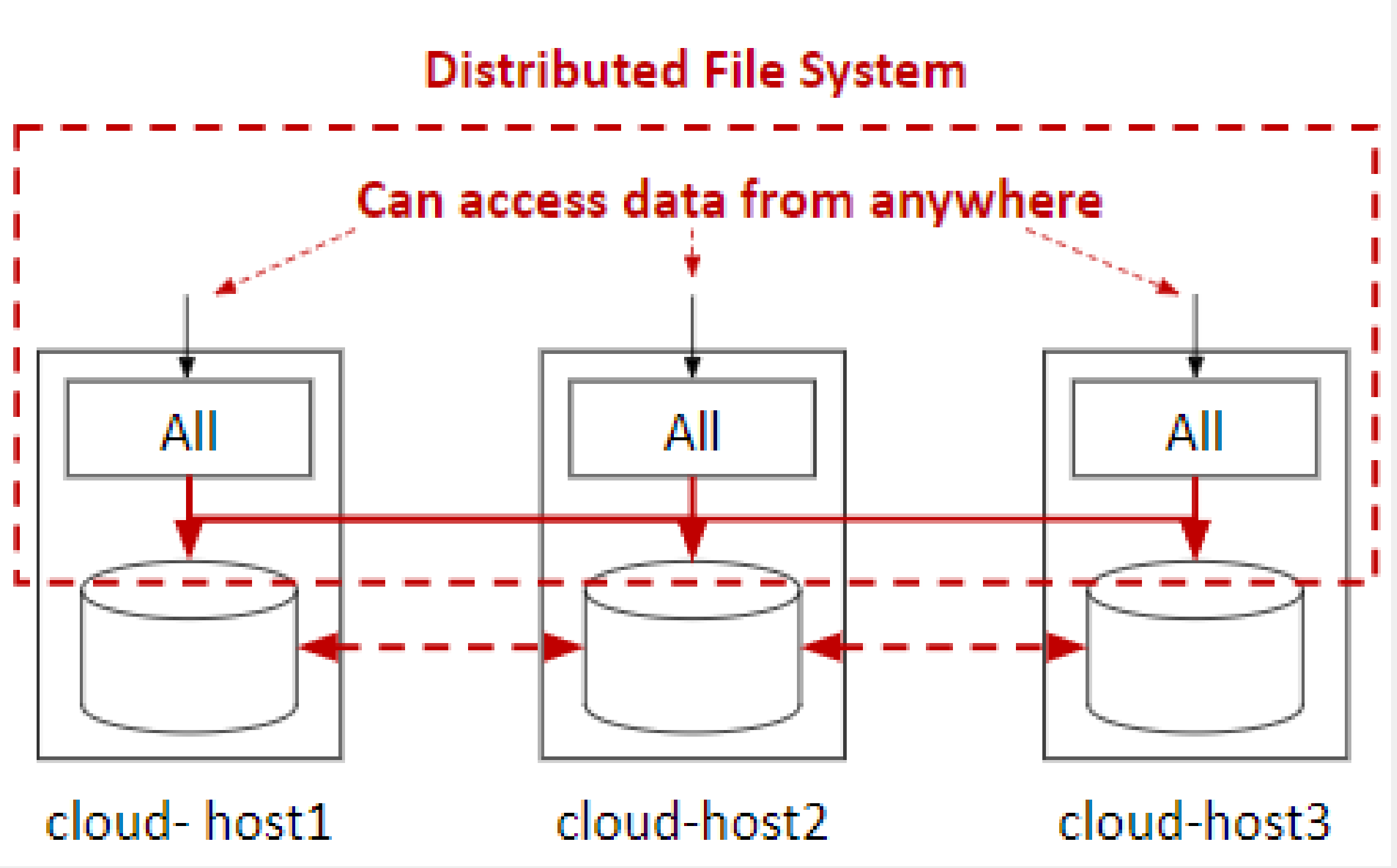
Create demographic maps with information from the 2010 US Census.

Choose a variable to display: Percent White

Range of interest: 0 – 75

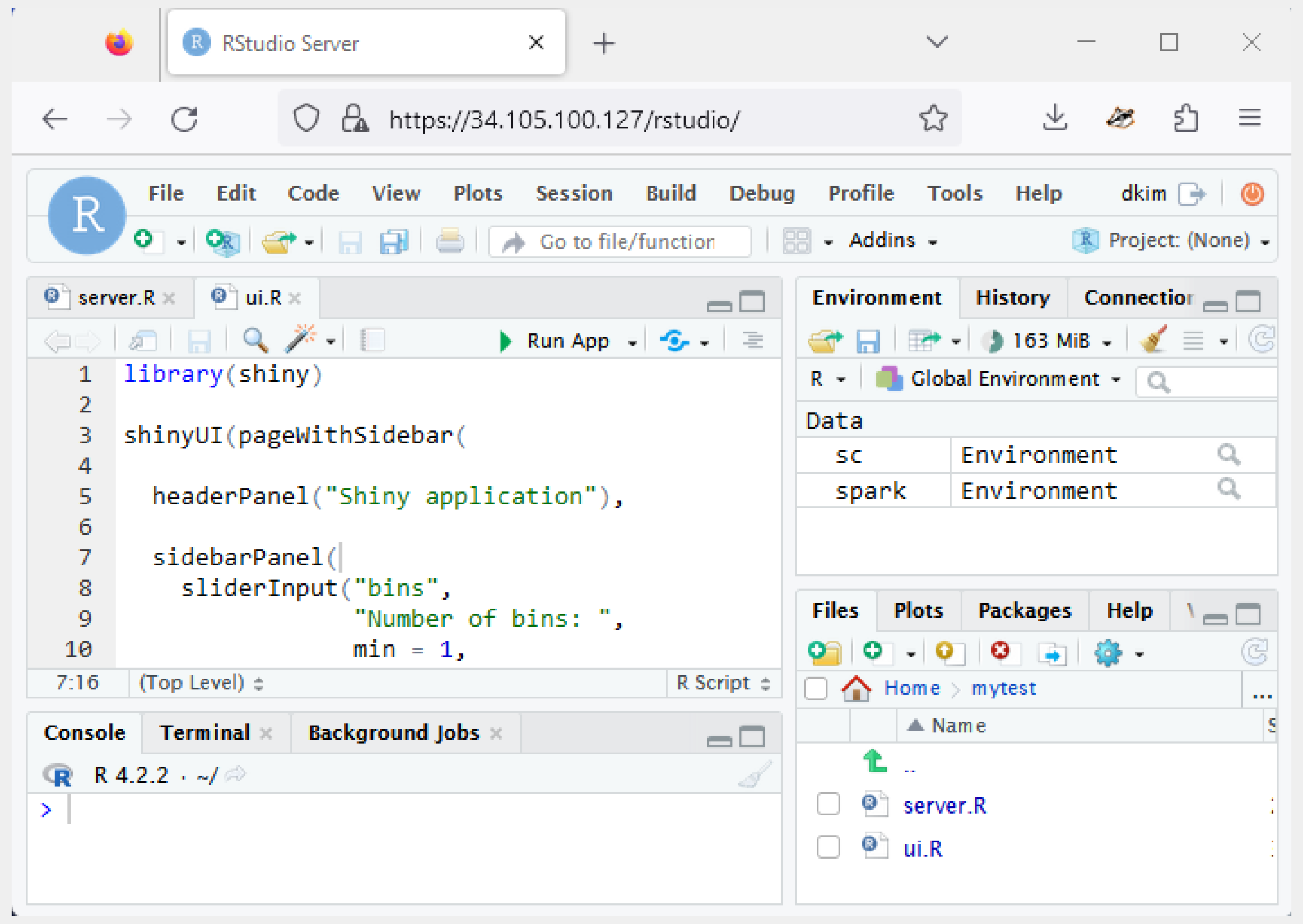
Distributed File System

GlusterFS [4] hides back-end storages from the user (admin) point of view and shows a logical one storage to the user (admin). The data is distributed and maintained through physical back-end storages.



Development Tool

Given that we are employing the R programming language [8] along with the R Shiny library, it is imperative to utilize the RStudio integrated development environment (IDE). RStudio proves to be an ideal tool for efficient file organization and data visualization through its excellent plotting capabilities [3]. It allows managing multiple projects, integrating version control systems, and has a strong debugging system.



```
library(shiny)

shinyUI(pageWithSidebar(

  headerPanel("Shiny application"),

  sidebarPanel(
    sliderInput("bins",
                "Number of bins: ",
                min = 1,
```

7:16 (Top Level) R Script

Data

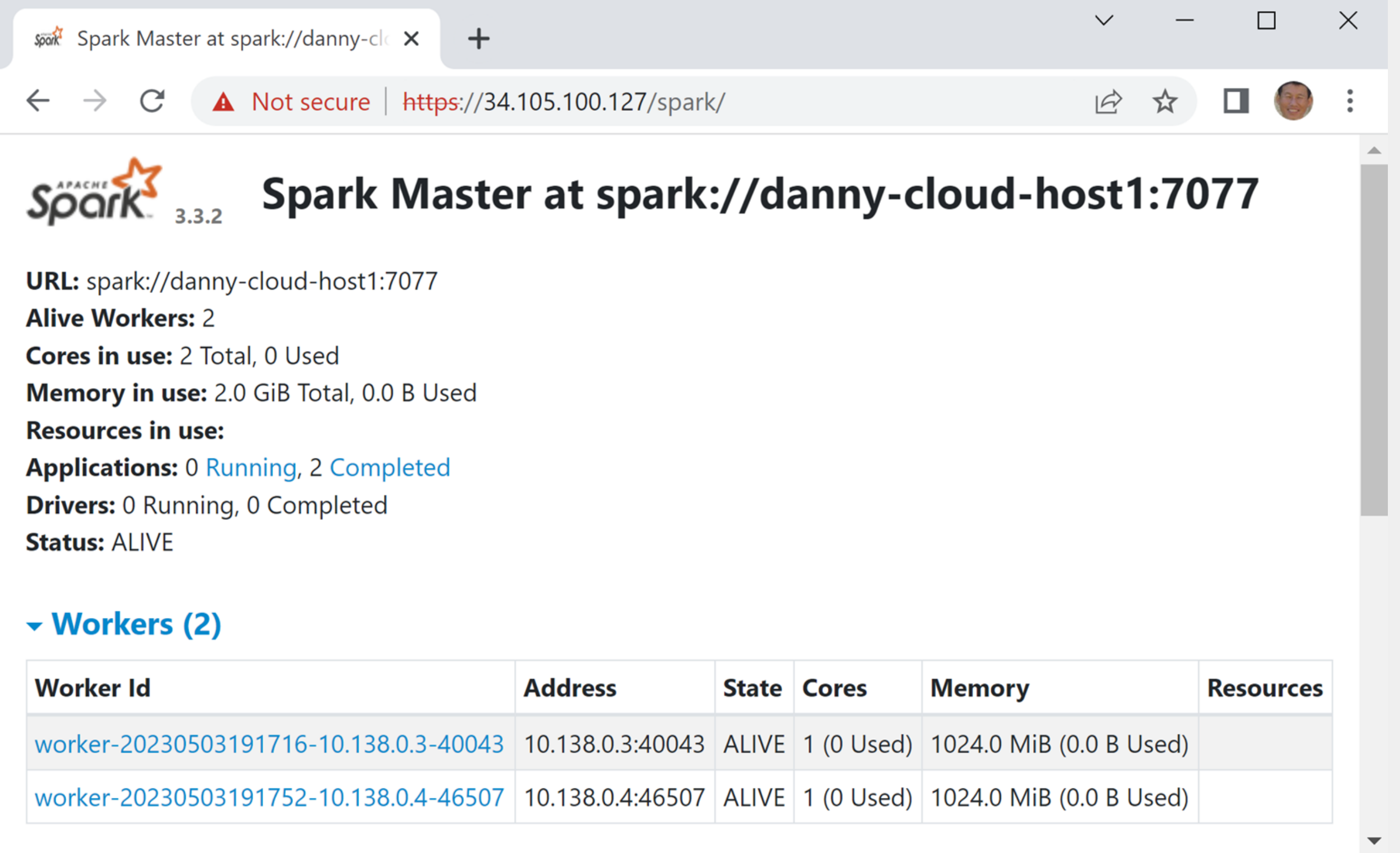
sc	Environment
spark	Environment

Files: server.R, ui.R

```
R 4.2.2 · ~/
>
```

Big Data Analysis Engine

Apache Spark is a unified analytics engine for large-scale data processing [5]. We use the engine to do the analysis using worker nodes. The following figure is the Spark master node Web console showing the information of a master node and two worker nodes.


Spark Master at spark://danny-cl
Not secure
https://34.105.100.127/spark/
Spark 3.3.2

Spark Master at spark://danny-cloud-host1:7077

URL: spark://danny-cloud-host1:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20230503191716-10.138.0.3-40043	10.138.0.3:40043	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20230503191752-10.138.0.4-46507	10.138.0.4:46507	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Connector

SparkR [9] is the connector between front-end (R) and back-end (Spark cluster) [5]. SparkR includes a distributed data frame implementation that performs various operations, such as filtering, selection, and aggregation on large datasets. It also enables distributed machine learning using MLlib.

No SQL Database

MongoDB [6] is employed as a NoSQL database management system to store and manage data, which can subsequently be retrieved and analyzed for further processing in the background.

Acknowledgement

This research activity is funded all or in part by the Stanislaus State ASPIRE program through a U.S. Department of Education Title III Grant # P031C210159.

Past/Present Research

- Efficient Visualization of Medical Big Data Using R Shiny Application (College of Science poster celebration, 2019)
- Medical Big Data Analysis System to Discover Associations between Genetic Variants and Diseases (IEEE International Conference on Communications (ICC)/College of Science poster celebration, 2021)
- Elastic High Speed Medical Big Data Analysis System to Discover Associations between Genetic Variants and Diseases: 10 Gbps network, 60TB disks, 10 TB medical dataset from MCRI [7]

References

[1] Google Cloud, https://cloud.google.com/
[2] Shiny, https://shiny.rstudio.com/tutorial/
[3] RStudio, https://posit.co/products/open-source/rstudio-server/
[4] GlusterFS, https://www.gluster.org/
[5] Spark, https://spark.apache.org
[6] MongoDB, https://www.mongodb.com/cloud
[7] MCRI (Marshfield Clinic Research Institute), https://www.marshfieldresearch.org/
[8] R, https://www.r-project.org/
[9] SparkR, https://spark.apache.org/docs/latest/sparkr.html