A Cloud Computing Platform for Data Analysis based on R Cluster

Yiming Tong, Zeyu Zheng, Dianzheng Fu, Yang Fu, Shuai Li
Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang 110016, China
Key Laboratory of Network Control System, Chinese Academy of Sciences
Shenyang 110016, China
tongyiming@sia.cn

Abstract—Translation of Data analysis algorithms from data analysis language to high-level programming language is hard work when we construct a cloud computing platform for data analysis. It adds implementation difficulty and maintenance cost of constructing the platform. This paper suggests a new method to use the popular data analysis language R directly on constructing a cloud computing platform. By using resource virtualization techniques, computing resource environment is virtualized into R cluster based on given configurations. By allocating different R machines from R cluster to specified data analysis service, we solve the problem that data analysis algorithms implemented in R should be run in single-user mode only and cannot be customized after being configured. After verification, this method simplifies the work of translating data analysis algorithms and speeds up the process of constructing a cloud computing platform for data analysis.

Keywords—cloud computing platform; data analysis; resource virtualization; R

I. INTRODUCTION

Cloud computing platform for data analysis is a platform to provide data analysis service from the Internet. By getting service from this platform, enterprise managers can find the value buried in their business data conveniently [1]. Nowadays, the construction of cloud computing platform for data analysis mostly adopts the pattern that data analysts design the data analysis algorithms with data analysis languages, platform developers translate the algorithms into high-level languages [2]. The translation of data analysis algorithms adds the difficulty to construct the platform. R is a popular data analysis language among data analysts. If we use R as the basic data analysis environment in the cloud computing platform directly, we can get rid of the procedure of translating the data analysis algorithms. Therefore, to find a way to utilize R as the basic data analysis environment on the cloud computing platform is very important. However, R is a data analysis language that should be run in single-user mode only and cannot be customized after being configured. Furthermore, R doesn’t provide a good interface to interact with the high-level programming language [3].

In this paper, we propose a method for constructing a cloud computing platform based on R cluster. This platform can use algorithms implemented in R to provide data analysis service to multiple users directly. The platform implemented by the proposed method mainly consists of five modules, including external system interface, data exchange bus, R cluster, resource controller and algorithm controller. External system interface receives requests and response the requests from the users. R cluster uses resource virtualization techniques to allocate and destroy R resource environment. Resource controller manages and controls the R resource environments allocated from the R cluster. Algorithm controller decides the execution of the data analysis algorithms in each R runtime environment. With the above construction, cloud computing platform based on R cluster can give multiple users data analysis service concurrently without the translation of data analysis algorithms from R to high-level programming language.

The remainder of this paper is organized as follows. Section 2 describes related works. In Section 3, the proposed system architecture is described. In Section 4, the detail of the main modules of the platform is introduced. In Section 5 and Section 6, we compare this method with other methods to construct the platform and test the performance. Finally, conclusion and future work is introduced in Section 7.

II. RELATED WORK

A. R

R is a nice data analysis language and environment. It is a GNU project which is free to use, free of charge and open source. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical test, time series analysis, classification, clustering) and graphical techniques, and is highly extensible [3]. R simplifies the procedure of data analysis, and becomes the most popular data analysis language today.

R can only be used by single user at one time and doesn’t provide an interface with other language directly [3]. Open source package ‘Rserve’ provides a way to help high-level programming language like C, Java and Python to invoke some simple R commands.

B. Cloud Computing Platform for Data Analysis

Cloud computing platform is a platform to provide shared, dynamic and extendable computing resources from the Internet on demand. The resources from the platform include networks, servers, storage, applications, and services, which can be rapidly provisioned and released with minimal management effort [4]. By the main kind of the platform’s
services, cloud computing platform can be classified as platform for storage and platform for data processing. Cloud computing platform for data analysis can be regarded as a platform for data processing. Its main task is to provide data analysis and data mining services to multiple users.

III. SYSTEM DESCRIPTION

The goal of constructing a cloud computing platform for data analysis based on R cluster is to help the data analysis analysts put their algorithms to use in the cloud computing platform directly without doing any extra work like translating the algorithms from data analysis language to high-level programming language. Each user of the platform can get the data analysis service wherever they are can share the newest algorithms. To accomplish this goal, the architecture of the platform can be divided into five main modules. They are external system interface, data exchange bus, R cluster, resource controller and algorithm controller. These five main modules interact with each other to make the data analysis service run in the platform. The system architecture of the platform is showed in Fig. 1.

![System Architecture](image)

The description of the five main modules in the cloud computing platform is as the following:

1) **External system interface**: This module mainly provides the interface to receive data analysis request from the users and answer back the data analysis results to the users in the REST [5] (Representational State Transfer) Web Services way.

2) **Data exchange bus**: This module works as the buffer of the input and output data. It saves the data in the CSV files and provides open interfaces to save and get the files as input and output of the data.

3) **R cluster**: This module uses resource virtualization techniques to virtualize all of the resource, allocate each request a R resource environment as required and recycle the redundant R resource environment.

4) **Resource controller**: This module adopts the resource managing method based on resource queues to decide when and how many given configured R resource environments should be allocated and recycled from the R cluster.

5) **Algorithm controller**: This module mainly manages the execution of the data analysis algorithms. By collecting algorithms’ state information from different R environments and analysising the state of the algorithms in each R environment, this module generates and sends the control commands to the corresponding R environment.

IV. FUNCTION AND IMPLEMENTATION

The function and implementation of the platform will be introduced from the five main modules, including the external system interface, data exchange bus, R cluster, resource controller and algorithm controller. The base software and hardware of the platform is showed in Table I.

<table>
<thead>
<tr>
<th>Base Environment</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM X3850</td>
<td>Base server environment</td>
</tr>
<tr>
<td>Red Hat Linux 6.5</td>
<td>Operating system</td>
</tr>
<tr>
<td>OpenStack</td>
<td>Virtual machine server</td>
</tr>
<tr>
<td>Java 7</td>
<td>Platform runtime environment</td>
</tr>
<tr>
<td>R 3.1.3</td>
<td>Data analysis algorithm environment</td>
</tr>
<tr>
<td>Spring 4.2</td>
<td>Data flow frame</td>
</tr>
<tr>
<td>RabbitMQ 3.6</td>
<td>Message queue and Data buffer</td>
</tr>
<tr>
<td>FTP server</td>
<td>File transfer environment</td>
</tr>
<tr>
<td>Apache Kafka 0.9</td>
<td>Multi-source information collector</td>
</tr>
</tbody>
</table>

A. **External System Interface**

External system interface receives request and answers back the request from the users. It is the bridge between the platform and the users. The data transmission format of this module adopts the standard REST Web Services way. The interfaces can be partitioned as authentication interface, request interface and response interface. The two typical REST data transmission format is showed in Fig. 2.

B. **Data Exchange Bus**

One important problem to solve in constructing this platform is how the R environments transfer data to the platform. The input size and output size of the data in the R environment is always large. When multi-users request data analysis service at the same time, it is easy to meet the data transmission bottleneck for the reason of the IO block. We design a data exchange bus to solve the problem of the data exchange between the platform and the R environments.
**Mean Function Request**

```json
{
  "token": "token1",
  "query": {
    "method": "mean"
  },
  "data": {
    "input": [1, 2, 4, 7, 3]
  }
}
```

**Mean Function Response**

```json
{
  "token": "token1",
  "message": "success",
  "status": "200",
  "query": {
    "method": "mean"
  },
  "data": {
    "input": [1, 2, 4, 7, 3],
    "output": 3.4
  }
}
```

Figure 2. External System Interface Data Format

R environment needs to load all the data needed to the memory of servers at one time. The preloading of all the data to the memory is easy to overflow the memory of the servers. Then the memory can be used to do data analysis computing is little. To solve this problem, we should store the intermediate data into a persistent storage. Another problem is that data format of the R language is flexible. The format can be data frame, data matrix, data array and etc. High-level language always doesn’t have so many one-to-one data format to save it. What is more, the flexible of data format always causes the difficulty to save the data in a common relational database. R is born to manipulate CSV format data. High-level language always contains the packages to manipulate CSV data too. For this reason, our cloud computing platform adopts CSV files to be the format to transfer intermediate data. Some tricks are designed to help save and get the intermediate data, such as data buffer techniques and data reliability techniques. Fig. 3 shows the structure of the data exchange bus.

![Data Exchange Bus Diagram](image)

Figure 3. Structure of Data Exchange Bus

C. R Cluster

Cloud computing platform sometimes needs to do many data analysis services at the same time. One R environment can execute only one algorithm at one time. To achieve the goal of multi-user used, we adopt the resource virtualization techniques. First, we build a private IaaS (Infrastructure as a Service) cloud computing platform with some resource virtualization tools to make an R cluster. The private IaaS cloud can allocate and destroy R environments on different configurations from the whole resource pool. When the private IaaS cloud receives an R environment allocation command, it clones an R environment virtual machine from the template of the given configuration. When the private IaaS cloud receives an R environment destroying command, it recycles the redundant R environment resource and gives the resource back to the resource pool. Fig. 4 shows the structure of the R cluster.

![R Cluster Diagram](image)

Figure 4. Structure of R Cluster

D. Resource Controller

How to manage the R resource that allocated from the R cluster is the key problem to solve in providing multi-user service from the platform. The function of resource controller is to solve this problem. Resource controller manages the R resource from the R cluster based on multiple resource queues. Resource controller balances the resource by special-designed data buffer. Resource controller first gets the formalized data analysis service from the external system interface, it then sends the input data to the data buffer and generates one required R resource’s information from the resource scheduler. The data buffer then sends the data to the data exchange bus. The resource scheduler then gets required R resource from the R cluster by a scheduling algorithm. The structure of the resource controller is showed in Fig. 5.

The configuration of each R resource environment is pre-defined by the designer, and it cannot be modified by users, so the kind of configurations in the platform is accountable. With this rule, the resource scheduling algorithm based on multiple resource queues is applicable to the platform.

The algorithm first pre-allocates some R resources on each defined configuration waiting to be called. The detail information of each free R resource is put into the queue that is the same as its configuration. When one R resource on some configuration is called, resource controller will get out one R resource’s information from the queue of the called
configuration. The R resource environment then can be used to run the data analysis algorithms implemented in the R language. After the algorithm run completely, the redundant R resource will be put back to the queue of the called configuration. The whole progress will be recognized as putting in and getting out R resource’s information from the resource queues.

**E. Algorithm Controller**

The execution of the data analysis algorithm always needs a long time, there may be any error happening during this process. If there is not a good mechanism to control the algorithm, the servers can be in deadlock, shut down and etc. The task of algorithm controller is to manage the control of the data analysis algorithm processed within one R resource environment. Algorithm controller first collects the states’ information of algorithms from the R resource environments. Then it explains the state’s information. At last, it sends the controlling commands back to the R resource environment according to the state of that R environment. R environment itself doesn’t provide any interface to the other languages. Algorithm controller uses an open source package called Rserv to collect states’ information and send control commands back to the R resource environment. The three main modules of the algorithm controller are log collector, log analyzer and command controller. The structure of the algorithm controller is showed in Fig. 7.

**F. Application Case**

The platform for data analysis based on R cluster has been used to provide data analysis service to some small companies, which includes mobile phone sales companies, industry plants, automated car plants and so on. One advantage of using R as the base computing resource is that R has nearly all popular data analysis algorithms’ packages and functions. A popular algorithm used in data analysis to group data into different families is...
called k-means clustering [15]. When we implement a data analysis algorithm like k-means clustering algorithm in java, we need to write a lot of code, and may make a mistake because the code is so long. R provides all popular packages and functions like k-means clustering to do data analysis at hand. Our platform uses the k-means function in R to provide the k-means clustering service. The result of an example of k-means clustering service from our platform is showed in Fig. 8. We verify the result with coloring the real data in Fig. 9, and get the right result as expected.

```json
K-means Function Request
{
  "token":"token2",
  "query":{
    "method":"kmeans"
  },
  "data":{
    "input":[
      {id: "0", x:187, y:203},
      {id: "1", x:185, y:205},
      ....
    ],
  }
}

K-means Function Response
{
  "token":"token2",
  "message":"success",
  "status":200,
  "query":{
    "method":"kmeans"
  },
  "data":{
    "output":[
      {id: "0", class: "1"},
      {id: "1", class: "1"},
      ....
    ],
  }
}
```

Figure 8. K-means Clustering Example Data Result

Figure 9. K-means Clustering Example Graph Result

Another advantage of using R as the base computing resource is that it provides many utilities to draw graphs to support data analysis. Drawing graphs can help us know the overall status of the data intuitively, especially when the data is very big and has a lot of characters. As we know, decision tree is a very popular algorithm to do data analysis, as it is simple to understand and interpret. Graph is a good tool to explain a decision tree, but there are few mature tools to draw decision tree and it is always hard to draw a good graph in the data analysis field. Luckily, R has the functions of drawing a decision tree. We integrate the graph functions from R into our platform to provide graphs to help user do data analysis. The example of a decision tree graph provided from our platform is showed in Fig. 10.

Figure 10. Decision Tree Analysis Graph Example

V. COMPARISON WITH OTHER METHODS

There are more and more cloud computing platforms for data analysis emerging today. Two typical methods of the platform’s construction are compared with ours in this paragraph. One is using R single server, the other is using high-level language Java. The feature comparison of the three methods to construct the platform is showed in Table II.

After comparison we can conclude that, if we want to construct a cloud computing platform for data analysis, we could choose R cluster or high-level language like Java as the computing base environment. If we want to seek for the high speed and never want to change the algorithms in the platform, we could choose the Java environment. If we want to have a lot of ready-to-use algorithms implemented in R, maybe sometimes add or change some algorithms, and don’t care about the speed of the algorithm’s running, constructing a cloud computing platform for data analysis based on R cluster may be a better choice.

<table>
<thead>
<tr>
<th>Feature</th>
<th>R Cluster</th>
<th>R</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration ability with the platform</td>
<td>hard</td>
<td>hard</td>
<td>easy</td>
</tr>
<tr>
<td>Algorithm running speed</td>
<td>medium</td>
<td>medium</td>
<td>Quick</td>
</tr>
<tr>
<td>Number of ready-to-use algorithms supported</td>
<td>many</td>
<td>many</td>
<td>few</td>
</tr>
<tr>
<td>Algorithms writable</td>
<td>easy</td>
<td>easy</td>
<td>hard</td>
</tr>
<tr>
<td>Multi-user service supported</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Developing cost</td>
<td>low</td>
<td>unknown</td>
<td>high</td>
</tr>
<tr>
<td>Developing time</td>
<td>short</td>
<td>unknown</td>
<td>long</td>
</tr>
<tr>
<td>Maintainability</td>
<td>easy</td>
<td>easy</td>
<td>hard</td>
</tr>
</tbody>
</table>

247
VI. SYSTEM PERFORMANCE TEST

We make a test of the prototyping system based on R cluster and compare this prototyping system with another prototyping system based on R single server only. To be fair compared, we choose that the two systems all run the same random-matrix-multiplication algorithm in only one machine.

Fig. 11. System Performance Comparison

![System Performance Comparison](image)

Fig. 11 shows the change of total executing time with the change of the number tasks processing at one time in our experiment between the two given prototyping systems. It is easily got from the figure that when the tasks are few, executing time difference between the two systems is little, but R cluster cost less executing time when there are a lot of tasks to process at one time. A system like cloud computing platform for data analysis may process a lot of tasks at one time, so R cluster has its advantages over R single server.

VII. CONCLUSION AND FUTURE RESEARCH

The recent rising requirement of value discovery buried in business data leads to the fast development of the cloud computing techniques. Moreover, R has become the most popular language in the data analysis field. To this end, this paper proposed a method to construct a cloud computing platform based on R cluster to provide data analysis service. There are two key features about the proposed platform. First, it doesn’t need the translation of data analysis algorithm from the data analysis language to high-level programming languages. Second, it provides a plan to provide multi-user service using the single-user based data analysis language. With this platform, developers of the platform can use the data analysis algorithms implemented in R directly to provide multi-user data analysis service without any translation work. In this way, it reduces the cost of constructing a platform and speeds up the development of the platform.

The future work will be focus on diminishing the cost of resource management and speeding up the process of data analysis on R cluster. Some work on parallel computation within R cluster is also needed. To diminish the resource allocated by the virtual machine is important in increasing the resource utilization rate. RHadoop and SparkR are two tools that own the ability to process big data with R, they may give us some good idea in processing big data in the future.

ACKNOWLEDGMENT

This work is supported by the Project of Key Laboratory of Network Control System, CAS (WLHKZ2014014).

REFERENCES