

SCALA: A Framework for Graphical Operations for iRODS

Wataru Takase**KEK**E-mail: wataru.takase@kek.jp***Adil Hasan***University of Liverpool**E-mail: adilhasan2@gmail.com***Yoshimi Iida***KEK**E-mail: yoshimi.iida@kek.jp***Francesca Di Lodovico***Queen Mary, University of London**E-mail: f.di.lodovico@qmul.ac.uk***Yoshiyuki Watase***KEK**E-mail: yoshiyuki.watase@kek.jp***Takashi Sasaki***KEK**E-mail: takashi.sasaki@kek.jp*

Statistical Charts And Log Analyzer (SCALA) is a framework for graphically displaying operational data for a distributed data management system. The framework has been applied to the integrated Rule Oriented Data Management System (iRODS), but can be applied to any data management system. The framework allows operational information such as disk usage, number of users etc to be displayed. In addition SCALA also allows remote debugging through the discovery and display of iRODS error messages from log files. This makes the detection and debugging of errors in a running iRODS system simpler. The system has also been extended to display iRODS rules and their execution. This SCALA system is currently being used in the KEK production iRODS system. This paper describes the SCALA framework and its application to the iRODS.

*The International Symposium on Grids and Clouds (ISGC) 2013,
March 17-22, 2013
Academia Sinica, Taipei, Taiwan*

*Speaker.

1. Introduction

A lot of information related to a service can be extracted from the service's log files, or by executing command line functions etc. Sometimes the volume information is overwhelming. A visualization system makes it possible to grasp overview of the system easily. There are many tools for visualizing services. At KEK, Ganglia [1] and MRTG (Multi Router Traffic Grapher) [2] have been installed to visualize cluster system status. Ganglia displays charts with regard to system resource usage such as CPU and memory. MRTG is for monitoring network data traffic and computer resources. These tools allow KEK cluster system administrators to easily determine the status of the service.

These tools are very helpful for a variety of service information, but for more detailed information on a service custom tools need to be written. In this paper we describe the development of the Statistical Charts And Log Analyzer (SCALA) framework for the integrated Rule Oriented Data System (iRODS) [3]. The framework allows monitoring and, to some degree, debugging capabilities for iRODS. The framework is based on experience of existing monitoring tools as described above and also on experience in operating and managing the iRODS at KEK. Using SCALA, administrators can easily grasp iRODS service status and the framework is applicable to different types of data management systems. This paper describes the SCALA framework and shows how SCALA can be applied to iRODS.

This paper starts with a description of iRODS in Section 2. Section 3 describes the SCALA design and the application of SCALA to iRODS is described in Section 4. Related work is shown in Section 5 and finally in Section 6 we provide a summary of the work done.

2. iRODS Overview

The integrated Rule Oriented Data System (iRODS) is a data grid software system developed by the Data Intensive Cyber Environments research group as the successor to Storage Resource Broker (SRB) [3]. It provides a logical file system connecting geographically distributed physical devices (which are named *resources* in iRODS). A mapping between the logical namespace and physical location is maintained in a database that serves as the iRODS metadata catalog. Understanding how storage resources are used can lead to a better management of the service.

A particular feature of iRODS is the *Rule Engine*. The Rule Engine allows policies to be enforced on the data [4]. The policies are encoded as rules which are interpreted by the Rule Engine. Rules can be written for many different types of policies (such as data replication, checksumming, authorization etc) and can be invoked automatically in response to certain conditions or triggers and can also be invoked from the command line. It is important to monitor the iRODS service to ensure the rules are functioning correctly and enforcing the policies.

3. SCALA Design

SCALA is a framework to visualize operational data for iRODS. SCALA provides two functions. One is displaying *Statistical Charts* that allows administrators to grasp system statistics

information graphically. The information includes storage resource usage per user, number of accesses and number of errors. The other is the *Log Analyzer* that helps administrators to debug the system by displaying number of errors with error messages.

The SCALA framework has functions that utilize the iRODS tools to extract information from the service, filter it and present it to the administrator in a digestible manner. Figure 1 shows the SCALA workflow. SCALA consists of three processes which are *parse*, *summarize* and *display*. System output is processed by these and finally visualized as charts. These processes are described as follows:

1. Parse

This process parses data management system output. SCALA collects system output and displays charts. However, the output also include unnecessary information for SCALA. This process extracts necessary information from the output periodically.

2. Summarize

This process summarizes the information extracted by the parse process. It sums up extracted information grouped by each date, each user and each storage resource. The summary information is used by the following the display process to display the data. This process reduces time to draw charts.

3. Display

This process functions as GUI (Graphical User Interface). When the user requests to this process, it handles the requests and updates the view. It displays operational data transition in time series as charts using the summarize process output. The operational data can be filtered by user, date, storage resource. Moreover, this process provides an error debugging tool displaying error chart with error messages. The tool enables to display log information related to each error message. It helps administrators to grasp detailed errors.

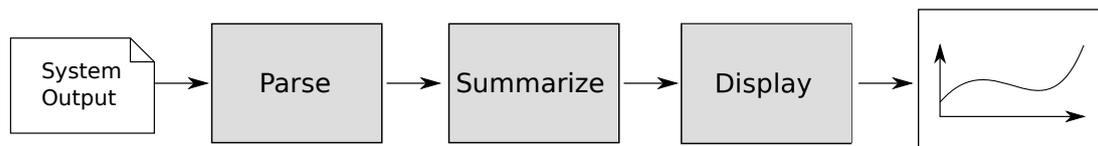


Figure 1: SCALA workflow

4. Application to iRODS

The SCALA framework has been implemented in the KEK iRODS system which enabled proactive management of the service by frequent monitoring of important system parameters. The tabbed Web pages acted as the administrator interface. The first tab contained the Statistical Charts that displays usage information, data volume, number of files, number of accesses and number of errors for an iRODS server on a daily basis. The second tab provides the Log Analyzer that displays daily error messages and helps administrators to debug errors easily by mouse and keyboard operations. Figure 2 shows SCALA workflow in iRODS. iRODS dumps resource usage and system

log information daily. The parse process extracts necessary information from the output and stores it in database. The summarize process queries the database to summarize stored information. The display process displays charts using the database information when a user accesses the Web page. The next three sub-sections give a detailed description of each of the processes.

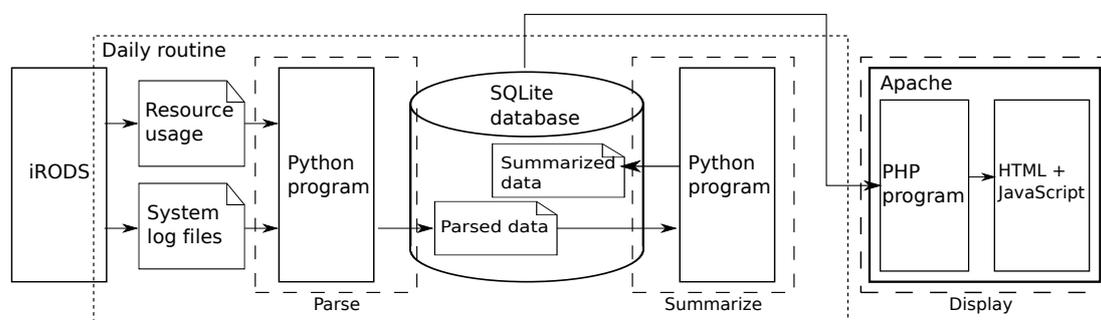


Figure 2: SCALA workflow in iRODS

4.1 Parse

iRODS provides a command line function named *iqrest* which can be used to query the iRODS metadata catalog to obtain metadata on stored files. The command enables users to query iRODS database using an SQL-like statement. To extract metadata, user executes *iqrest* commands with *SELECT* statement which also supports aggregate functions such as *COUNT*, *SUM*. Using the *iqrest* commands the parse process obtains current data volume and number of files each user and each resource. The *iqrest* command is executed daily by a Linux cron daemon. A Python program parses the *iqrest* output and then inserts the data with the date into the resource usage table in an SQLite database. Table 1 is an example of the table. In the table, resource usage transitions are stored for each user and each resource daily. The record ID 1 means *user1* had 10 files in *ResourceA* and its data volume is 10240 as of *November 22*. The record ID 4 means *user1* added 2 files in *ResourceA* and its data volume increase 4960 on *November 23*. The table is used for data volume and number of files charts on the Statistical Charts page.

iRODS writes system information to log files when a user accesses, or a rule is executed or an error occurs. The parse program checks for updates to log files daily. If there are updates, SCALA extracts information added newly from the log files. For the number of accesses chart the program extracts access date and the name of access user from the log files and inserts them in the access log table. Table 2 is an example of the table. The example shows *user1* accessed twice and *user2* accessed once on *November 22*. For the number of errors chart, the program extracts error occurred date, error message, error status and log file name containing the error and inserts them into the error log table. Table 3 is an example of the table. The record ID 1 means that an error occurred with status -1000 on *November 22* and the error message was written to *rodsLog1*.

4.2 Summarize

A Python program submits SQL queries to the SCALA database after the parse process has completed. For the data volume and number of files charts, the data from the resource usage table

Table 1: Example of Resource Usage Table

| Record ID | Date | User | Stored Resource | Data Volume | Number of Files |
|-----------|---------------|-------|-----------------|-------------|-----------------|
| 1 | Nov. 22, 2012 | user1 | ResourceA | 10240 | 10 |
| 2 | Nov. 22, 2012 | user1 | ResourceB | 768 | 2 |
| 3 | Nov. 22, 2012 | user2 | ResourceA | 5120 | 5 |
| 4 | Nov. 23, 2012 | user1 | ResourceA | 15200 | 12 |
| 5 | Nov. 23, 2012 | user1 | ResourceB | 150 | 1 |
| 6 | Nov. 23, 2012 | user2 | ResourceA | 6000 | 15 |

Table 2: Example of Access Log Table

| Record ID | Date | User |
|-----------|---------------|-------|
| 1 | Nov. 22, 2012 | user1 |
| 2 | Nov. 22, 2012 | user2 |
| 3 | Nov. 22, 2012 | user1 |

Table 3: Example of Error Log Table

| Record ID | Date | Message | Status | Log File |
|-----------|---------------|----------------|---------|----------|
| 1 | Nov. 22, 2012 | error message1 | -1000 | rodsLog1 |
| 2 | Nov. 24, 2012 | error message2 | -803000 | rodsLog2 |
| 3 | Nov. 24, 2012 | error message3 | -9000 | rodsLog2 |

Table 4: Example of Access Log Summarized Table

| Record ID | Date | User | Number of Accesses |
|-----------|---------------|-------|--------------------|
| 1 | Nov. 22, 2012 | user1 | 2 |
| 2 | Nov. 22, 2012 | user2 | 1 |

have already been summarized for each date, each user and each resource. However, the parse process just extracts each access message and error message from log files and inserts its as record such as Table 2 and Table 3. For the number of accesses chart, the program sums up records from the access log table grouped by user and date. For the number of errors chart, the program sums up records from the error log table grouped by date. The summarized information are inserted to summarized tables. Table 4 is an example of the access log summarized table of Table 2.

Once the summarized tables have been updated, they only need to be updated the next time the parse application runs. Creating summarized tables reduces the time to sum up records and leads to a speed up when displaying charts on the Web page.

4.3 Display

Accesses to the SCALA Web site causes a PHP program to read data from the database and processes the data and display charts through the Apache HTTP server. The charts are implemented in JavaScript allowing interaction with the data.

4.3.1 Statistical Charts

Figure 3 is the Statistical Charts page. This page displays four charts data volume, number of files, number of accesses and number of errors. The charts are drawn on an HTML 5 Canvas [5] using JavaScript. Each chart has pull-down menus: a) the user menu can filter which users displayed, b) the resource menu can filter which resources displayed and c) the data span menu which can filter date range. If an item is selected from the pull-down menu, the program re-reads the tables, filters the data using selected item and redraws chart dynamically. This page helps to grasp iRODS operational status overview.

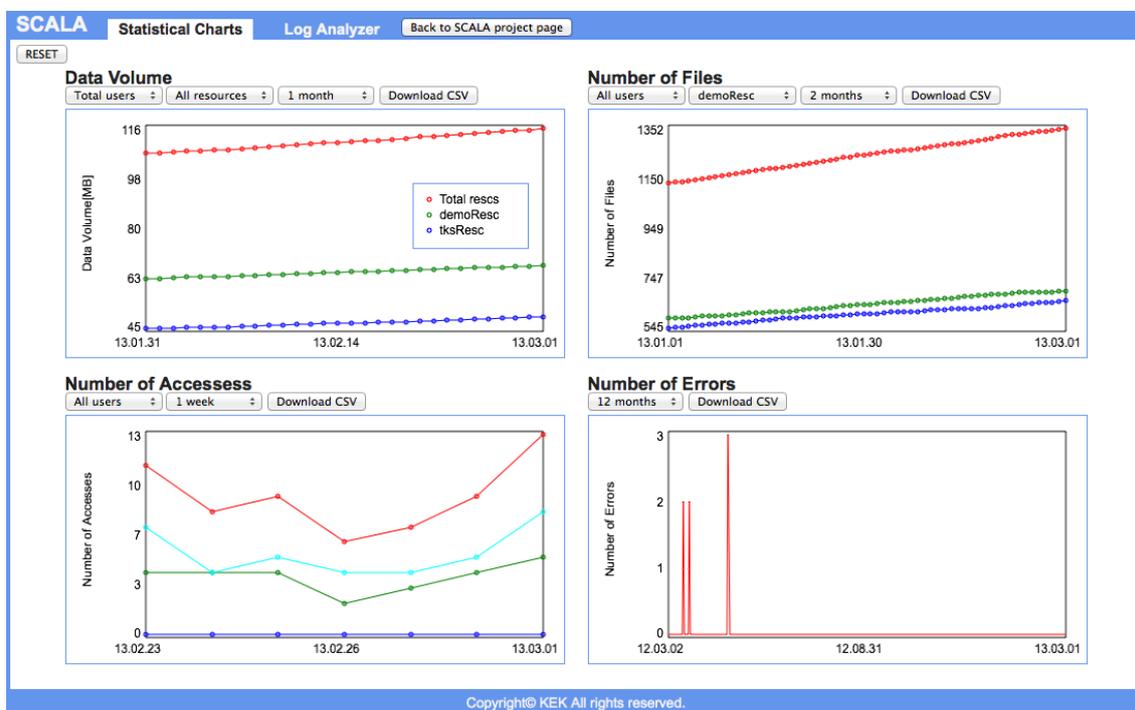


Figure 3: Statistical Charts page

4.3.2 Log Analyzer

Figure 4 is the Log Analyzer page. The chart is drawn using the JavaScript library Flot [6]. The page displays the number of errors as bar chart with error messages each day. This page allows the user to zoom in and out, pan, filter status and check detailed errors. When the user clicks each bar, error messages and number of errors are displayed on the bottom right window. Figure 4 shows there were eight errors on February 20th and shows also these eight error messages. Moreover when user clicks each error message, log file messages containing the error are displayed in the

POS (ISGC 2013) 021

bottom left window. The interface allows a summary of error messages to be displayed and all messages for a given period to be listed. This page helps the administrators to debug iRODS errors easily.

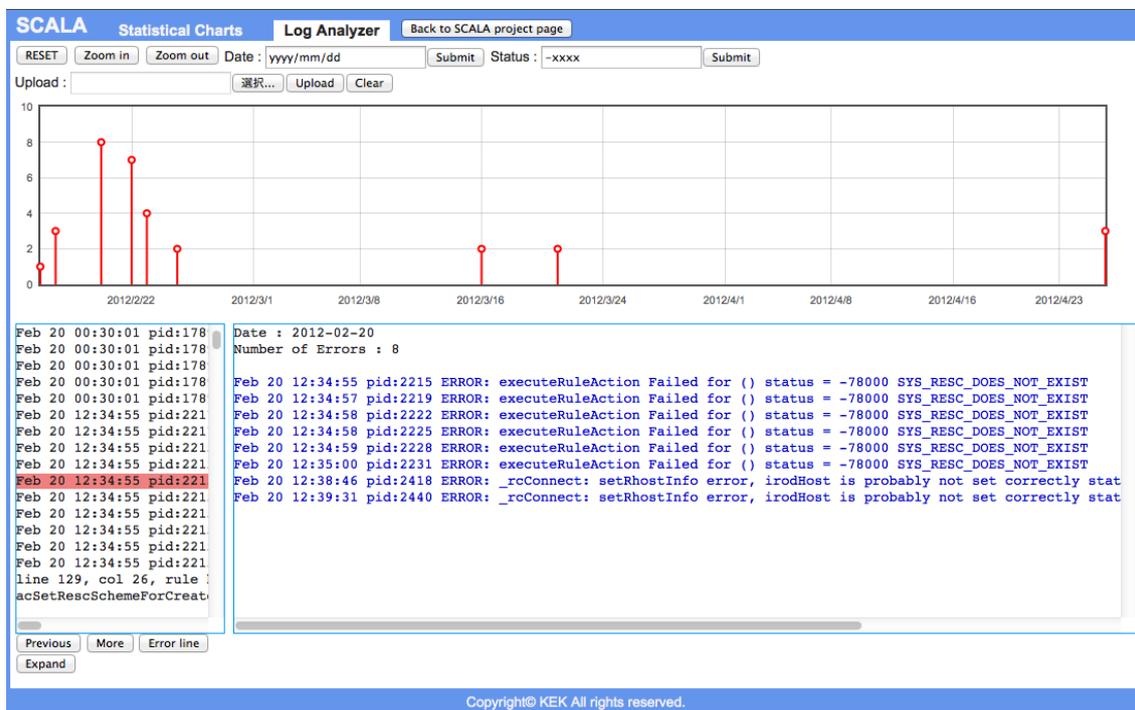


Figure 4: Log Analyzer page

4.4 SCALA in KEK

In KEK, several iRODS have been running and the Tokai to Kamioka (T2K) Neutrino experimental group [7] is the main user of the KEK iRODS [8]. In the T2K iRODS, data quality data relating to the experiment are stored every day. The data volume is several terabytes and there are hundreds of thousands of files in the iRODS. Moreover complex rules to handle the stored data run periodically. Application of SCALA to iRODS in KEK allows the administrators to check operational data simpler and detect and debug errors easier.

Furthermore, SCALA shows that the summarize work does reduce the time to draw charts in T2K iRODS. Without summarized tables, drawing charts takes about two minutes in T2K iRODS whereas using the summarized tables takes less than a second. This is due to the large size of the SQLite database containing the original data.

4.5 SCALA Extensions

iRODS is a rule based data management system. In iRODS, many kind of rules are executed automatically or manually every day. With regard to rules, it would be useful to know: a) what kind of rules are executed, b) how long a rule takes to execute, c) when is a rule executed and d) whether a rule is executed as administrators intend. To manage rules well, we have added a *Rule Management* page to SCALA. iRODS writes rule execution information to log files. For

Pos (ISGC 2013) 021

Table 5: The Classification Result of SCALA

| Dimension | Statistical Charts | Log Analyzer |
|--------------------------------------|-----------------------|--|
| Data to be visualized | One-dimensional data | One-dimensional data |
| Visualization technique | Standard 2D display | Standard 2D display |
| Interaction and distortion technique | Interactive filtering | Interactive filtering Interactive zooming Interactive distortion |

Rule Management page, parse, summarize and display program have been added handling the information related rules. Moreover the rule log table have been added to the database to store the rule log information. The iRODS Rule Engine interprets the rules written in rule files. Using the rule ID from the log files and the rule descriptions in the rule files, the administrators can determine which rule ran through the Web interface.

SCALA can also be extended to display an overview of other types of operations such as uploading/downloading files. This can be done by modifying the rules of iRODS to write the operational records to the iRODS log file. The operational log can be processed by the SCALA workflow (it may almost be same as the access log handling described in the sub-sections 4.1 to 4.3).

5. Related Work

There are a lot of tools for visualizing information and SCALA is one of them. Keim [9] propose a classification of information visualization and visual data mining techniques using three dimensions: the data to be visualized, the visualization technique, and the interaction and distortion technique. To make the SCALA's position of the tools clear, we classified SCALA by the Keim's method (Table 5). Statistical Charts displays line charts (standard 2D display) using time series of system status values (one-dimensional data) and each chart can be filtered by date, user and storage resource (interactive filtering). The Log Analyzer displays bar chart (standard 2D display) using time series of number of errors (one-dimensional data). The chart can be filtered by error status (interactive filtering) and zoomed in/zoomed out (interactive zooming). Moreover the page can show detailed error each date (interactive distortion).

In the grid computing field, there are also some visualizing tools and GridMapper [10] is one of them. GridMapper shows statuses related to network communications of grid services mapping geographic locations and focuses real time performance. By contrast, SCALA visualizes detailed information of a service and focuses history of the status and debugging errors.

6. Summary

In this paper we have described the function and operation of the SCALA framework to visualize operational data and to provide a debugging aid for a distributed data management system. Currently SCALA is used in the KEK production iRODS system. This framework can improve usability for system administrators to grasp overview system and debugging log files.

SCALA has been applied to iRODS only because the parse program is currently specific to iRODS output format. Other data management systems have a different output format. In order to extract the necessary information for SCALA from another system output, it is necessary to create a parse program for that format. It may be desirable for administrators to be able to change the parse program for the data management system to be used. This will be considered for future work.

Acknowledgments

We thank the generous help of the iRODS developers: Mike Wan and Wayne Schroeder. This work was also partially funded by the European Research Council, Grant agreement no. 207282-T2KQMUL. A. H. is grateful to the support of the KEK foreign researcher grant that enabled this work to be done.

References

- [1] “Ganglia Monitoring System.” <http://ganglia.sourceforge.net/>.
- [2] “MRTG – Multi Router Traffic Grapher.” <http://oss.oetiker.ch/mrtg/>.
- [3] “iRODS – the Integrated Rule-Oriented Data System.” <http://www.irods.org/>.
- [4] A. Rajasekar, M. Wan, R. Moore, and W. Schroeder, *A Prototype Rule-based Distributed Data Management System*, in *HPDC workshop on Next Generation Distributed Data Management*, 2006.
- [5] “W3C – HTML5.” <http://www.w3.org/TR/html5/>.
- [6] “Flot: Attractive JavaScript plotting for jQuery.” <http://www.flotcharts.org/>.
- [7] Y. Itow et al., *The JHF-Kamioka neutrino project*, in *KEK Report*, 2001.
- [8] F. D. Lodovico, A. Hasan, Y. Iida, and T. Sasaki, *First iRODS Experience in a Neutrino Experiment*, in *iRODS User Group Meeting*, 2011.
- [9] D. A. Keim, *Information Visualization and Visual Data Mining*, *IEEE Transactions on Visualization and Computer Graphics* **8** (2002), no. 1 1–8.
- [10] W. Allcock, J. Bester, J. Bresnahan, I. Foster, J. Gawor, J. Insley, J. Link, and M. Papka, *GridMapper: a tool for visualizing the behavior of large-scale distributed systems*, in *Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing*, pp. 179–187, 2002.