

# Big Data Technologies

Editor: Gregor von Laszewski

March 19, 2017

## 0.1 Contributors

Name	HID	Title	Pages
Avadhoot Agasti	SL-IO-3000	TBD	1
Matthew Lawson	S17-IO-3010	TBD	1

# Apache Ranger

AVADHOOT AGASTI<sup>1,\*</sup>, +

<sup>1</sup>School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

\* Corresponding authors: aagasti@indiana.edu

+ HID - SL-IO-3000

paper2, March 10, 2017

Apache Hadoop provides various data storage, data access and data processing services. Apache Ranger is part of the Hadoop eco-system. Apache Ranger provides capability to perform security administration tasks for storage, access and processing of data in Hadoop. Using Ranger, Hadoop administrator can perform security administration tasks using a central UI or Restful web services. He can define policies which enable users/user-groups to perform specific action using Hadoop components and tools. Ranger provides role based access control for datasets on Hadoop at column and row level. Ranger also provides centralized auditing of user access and security related administrative actions.

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

**Keywords:** Apache Ranger, LDAP, Active Directory, Apache Knox, Apache Atlas, Apache Hive, Apache Hadoop, Yarn, Apache HBase, Apache Storm, Apache Kafka, Data Lake, Apache Sentry

<https://github.com/avadhoot-agasti/sp17-i524/tree/master/paper2/S17-IO-3000/report.pdf>

## 1. INTRODUCTION

Apache Ranger is open source software project designed to provide centralized security services to various components of Apache Hadoop. Apache Hadoop provides various mechanism to store, process and access the data. Each Apache tool has its own security mechanism. This increases administrative overhead and is also error prone. Apache Ranger fills this gap to provide a central security and auditing mechanism for various Hadoop components. Using Ranger, Hadoop administrator can perform security administration tasks using a central UI or Restful web services. He can define policies which enable users/user-groups to perform specific action using Hadoop components and tools. Ranger provides role based access control for datasets on Hadoop at column and row level. Ranger also provides centralized auditing of user access and security related administrative actions.

## 2. ARCHITECTURE OVERVIEW

[1] describes the important components of Ranger as explained below:

### 2.1. Ranger Admin Portal

Ranger Admin Portal is the main interaction point for the user. Using Admin Portal, user can define policies. The policies are stored in Policy Database. The Policies are polled by various plugins. The Admin Portal also collects the audit data from plugins and stores in HDFS or in a relational database.

### 2.2. Ranger Plugins

Plugins are Java Programs, which are invoked as part of the cluster component. For example, the ranger-hive plugin is embedded as part of Hive Server2. The plugins cache the policies, and intercept the user request and evaluates it against the policies. Plugins also collect the audit data for that specific component and send to Admin Portal.

### 2.3. User group sync

While Ranger provides authorization or access control mechanism, it needs to know the users and the groups. Ranger integrates with Unix users management or LDAP or Active Directory to fetch the users and groups. The User group sync component is responsible for this integration.

## 3. HADOOP COMPONENTS SUPPORTED BY RANGER

Ranger supports auditing and authorization for following Hadoop components [2].

### 3.1. Apache Hadoop and HDFS

Apache Ranger provides plugin for Hadoop, which helps in enforcing data access policies. The HDFS plugin works with Name Node to check if the user's access request to a file on HDFS is valid or not.

### 3.2. Apache Hive

Apache Hive provides SQL interface on top of the data stored in HDFS. Apache Hive supports two types of authorization -

Storage based authorization and SQL standard authorization. Ranger provides centralized authorization interface for Hive which provides granular access control at table and column level. Ranger implements a plugin which is part of Hive Server2.

### 3.3. Apache HBase

Apache HBase is NoSQL database implemented on top of Hadoop and HDFS. Ranger provides coprocessor plugin for HBase, which performs authorization checks and audit log collections.

### 3.4. Apache Storm

Ranger provides plugin to Nimbus server which helps in performing the security authorization on Apache Storm.

### 3.5. Apache Knox

Apache Knox provides service level authorization for users and groups. Ranger provides plugin for Knox using which, administration of policies can be supported. The audit over Knox data enables user to perform detailed analysis of who and when accessed Knox.

### 3.6. Apache Solr

Solr provides free text search capabilities on top of Hadoop. Ranger is useful to protect Solr collections from unauthorized usage.

### 3.7. Apache kafka

Ranger can manage access control on Kafka topics. Policies can be implemented to control which users can write to a Kafka topic and which users can read from a Kafka topic.

### 3.8. Yarn

Yarn is resource management layer for Hadoop. Administrators can setup queues in Yarn and then allocate users and resources per queue basis. Policies can be defined in Ranger to define who can write to various Yarn queues.

## 4. IMPORTANT FEATURES OF RANGER

The blog article [3] explains the 2 important features of Apache Ranger.

### 4.1. Dynamic Column Masking

Dynamic data masking at column level is an important feature of Apache Ranger. Using this feature, the administrator can setup data masking policy. The data masking makes sure that only authorized users can see the actual data while other users will see the masked data. Since the masked data is format preserving, they can continue their work without getting access to the actual sensitive data. For example, the application developers can use masked data to develop the application whereas when the application is actually deployed, it will show actual data to the authorized user. Similarly, a security administrator may choose to mask credit card number when it is displayed to a service agent.

### 4.2. Row Level Filtering

The data authorization is typically required at column level as well as at row level. For example, in an organization which is geographically distributed in many locations, the security administrator may want to give access of a data from a specific location to the specific user. In other example, a hospital data

security administrator may want to allow doctors to see only his or her patients. Using Ranger, such row level access control can be specified and implemented.

## 5. HADOOP DISTRIBUTION SUPPORT

Ranger can be deployed on top of Apache Hadoop. [4] provides detailed steps of building and deploying Ranger on top of Apache Hadoop.

Hortonwork Distribution of Hadoop (HDP) supports Ranger deployment using Ambari. [5] provides installation, deployment and configuration steps for Ranger as part of HDP deployment.

Cloudera Hadoop Distribution (CDH) does not support Ranger. According to [6], Ranger is not recommended on CDH and instead Apache Sentry should be used as central security and audit tool on top of CDH.

## 6. USE CASES

Apache Ranger provides centralized security framework which can be useful in many use cases as explained below.

### 6.1. Data Lake

[7] explains that storing many types of data in the same repository is one of the most important feature of Data Lake. With multiple datasets, the ownership, security and access control of the data becomes primary concern. Using Apache Ranger, the security administrator can define fine grain control on the data access.

### 6.2. Multi-tenant Deployment of Hadoop

Hadoop provides ability to store and process data from multiple tenants. The security framework provided by Apache Ranger can be utilized to protect the data and resources from un-authorized access.

## 7. APACHE RANGER AND APACHE SENTRY

According to [8], Apache Sentry and Apache Ranger have many features in common. Apache Sentry ([9]) provides role based authorization to data and metadata stored in Hadoop.

## 8. EDUCATIONAL MATERIAL

[10] provides tutorial on topics like A)Security resources B)Auditing C)Securing HDFS, Hive and HBase with Knox and Ranger D) Using Apache Atlas' Tag based policies with Ranger. [11] provides step by step guidance on getting latest code base of Apache Ranger, building and deploying it.

## 9. LICENSING

Apache Ranger is available under Apache 2.0 License.

## 10. CONCLUSION

Apache Ranger is useful to Hadoop Security Administrators since it enables the granular authorization and access control. It also provides central security framework to different data storage and access mechanism like Hive, HBase and Storm. Apache Ranger also provides audit mechanism. With Apache Ranger, the security can be enhanced for complex Hadoop use cases like Data Lake.

## ACKNOWLEDGEMENTS

The authors thank Prof. Gregor von Laszewski for his technical guidance.

## REFERENCES

- [1] Hortonworks, "Apache ranger - overview," Web Page, online; accessed 9-Mar-2017. [Online]. Available: [https://hortonworks.com/apache/ranger/#section\\_2](https://hortonworks.com/apache/ranger/#section_2)
- [2] A. S. Foundation, "Apache ranger - frequently asked questions," Web Page, online; accessed 9-Mar-2017. [Online]. Available: [http://ranger.apache.org/faq.html#How\\_does\\_it\\_work\\_over\\_Hadoop\\_and\\_related\\_components](http://ranger.apache.org/faq.html#How_does_it_work_over_Hadoop_and_related_components)
- [3] S. Mahmood and S. Venkat, "For your eyes only: Dynamic column masking & row-level filtering in hdp2.5," Web Page, Sep. 2016, online; accessed 9-Mar-2017. [Online]. Available: <https://hortonworks.com/blog/eyes-dynamic-column-masking-row-level-filtering-hdp2-5/>
- [4] A. S. Foundation, "Apache ranger 0.5.0 installation," Web Page, online; accessed 9-Mar-2017. [Online]. Available: <https://cwiki.apache.org/confluence/display/RANGER/Apache+Ranger+0.5.0+Installation>
- [5] Horto, "Installing apache rang," Web Page, online; accessed 9-Mar-2017. [Online]. Available: [https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.6/bk\\_installing\\_manually\\_book/content/ch\\_installing\\_ranger\\_chapter.html](https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.6/bk_installing_manually_book/content/ch_installing_ranger_chapter.html)
- [6] Cloudera, "Configuring authorization," Web Page, online; accessed 9-Mar-2017. [Online]. Available: [https://www.cloudera.com/documentation/enterprise/5-6-x/topics/sg\\_authorization.html](https://www.cloudera.com/documentation/enterprise/5-6-x/topics/sg_authorization.html)
- [7] Teradata and Hortonworks, "Putting the data lake to work - a guide to best practices," Web Page, Apr. 2014, online; accessed 9-Mar-2017. [Online]. Available: [https://hortonworks.com/wp-content/uploads/2014/05/TeradataHortonworks\\_Datalake\\_White-Paper\\_20140410.pdf](https://hortonworks.com/wp-content/uploads/2014/05/TeradataHortonworks_Datalake_White-Paper_20140410.pdf)
- [8] S. Neumann, "5 hadoop security projects," Web Page, Nov. 2014, online; accessed 9-Mar-2017. [Online]. Available: <https://www.xplenty.com/blog/2014/11/5-hadoop-security-projects/>
- [9] A. S. Foundation, "Apache sentry," Web Page, online; accessed 9-Mar-2017. [Online]. Available: <https://sentry.apache.org/>
- [10] Hortonworks, "Apache ranger overview," Web, online; accessed 9-Mar-2017. [Online]. Available: <https://hortonworks.com/apache/ranger/#tutorials>
- [11] A. S. Foundation, "Apache ranger - quick start guide," Web Page, online; accessed 9-Mar-2017. [Online]. Available: [http://ranger.apache.org/quick\\_start\\_guide.html](http://ranger.apache.org/quick_start_guide.html)

# Robot Operating System (ROS): A Useful Overview

MATTHEW LAWSON<sup>1</sup>

<sup>1</sup>School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

\* Corresponding authors: laszewski@gmail.com

paper2, March 17, 2017

placeholder text

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

**Keywords:** Cloud, I524, robot, ros, ROS

<https://github.com/eunosm3/classes/blob/master/docs/source/format/report/report.pdf>

## INTRODUCTION

### EXAMPLES OF ARTICLE COMPONENTS

The sections below show examples of different article components.

### FIGURES AND TABLES

It is not necessary to place figures and tables at the back of the manuscript. Figures and tables should be sized as they are to appear in the final article. Do not include a separate list of figure captions and table titles.

Figures and Tables should be labelled and referenced in the standard way using the `\label{}` and `\ref{}` commands.

#### Sample Figure

Figure 1 shows an example figure.

#### Sample Table

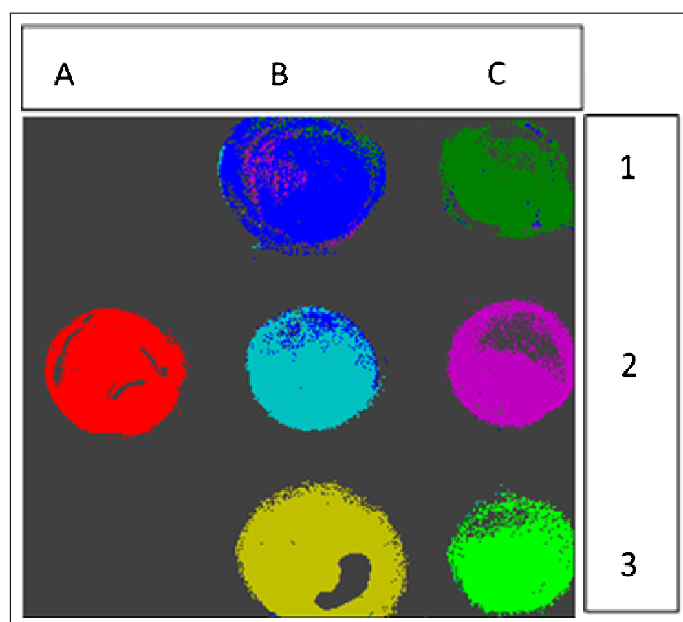
Table 1 shows an example table.

**Table 1. Shape Functions for Quadratic Line Elements**

local node	$\{N\}_m$	$\{\Phi_i\}_m$ ( $i = x, y, z$ )
$m = 1$	$L_1(2L_1 - 1)$	$\Phi_{i1}$
$m = 2$	$L_2(2L_2 - 1)$	$\Phi_{i2}$
$m = 3$	$L_3 = 4L_1L_2$	$\Phi_{i3}$

### SAMPLE EQUATION

Let  $X_1, X_2, \dots, X_n$  be a sequence of independent and identically distributed random variables with  $E[X_i] = \mu$  and  $\text{Var}[X_i] = \sigma^2 < \infty$ , and let



**Fig. 1.** False-color image, where each pixel is assigned to one of seven reference spectra.

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n} \sum_i^n X_i \quad (1)$$

denote their mean. Then as  $n$  approaches infinity, the random variables  $\sqrt{n}(S_n - \mu)$  converge in distribution to a normal  $\mathcal{N}(0, \sigma^2)$ .

## SAMPLE ALGORITHM

Algorithms can be included using the commands as shown in algorithm 1.

### Algorithm 1. Euclid's algorithm

---

```

1: procedure EUCLID( $a, b$ )           ▷ The g.c.d. of  $a$  and  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do               ▷ We have the answer if  $r$  is 0
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   return  $b$                        ▷ The gcd is  $b$ 

```

---

### Algorithm 2. Python example

---

```

1  for i in range(0,100):
2      print i

```

---

## REFERENCE MANAGEMENT

The best programs to manage your references is jabref or emacs. You can edit the references and verify them with them for format errors. To cite them use the citation key. You can add multiple bib files to the bibliography command separated by comma. Add citations with the cite command. See [?] for an example on how to use multiple clouds. In [?] we list the class content.

Here a test of a citation with an underscore in the url [1].

## ACKNOWLEDGEMENTS

Funding information should be listed in this section. Please evaluate if you like to list your employer that may have funded your activities here. If you receive grants or project numbers, as shown in the example. This work was in part supported by National Science Foundation (NSF) (1234567, 891012345) (These numbers are invented)

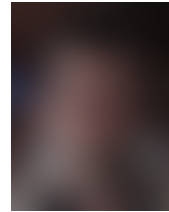
The acknowledgments may also contain any information that is not related to funding:

The authors thank H. Haase, C. Wiede, and J. Gabler for technical support.

## REFERENCES

- [1] Web Page. [Online]. Available: [http://www.google.com/some\\_underscore](http://www.google.com/some_underscore)
- [2] ros.org, "Why Ros?: A distributed, modular design," Web page, mar 2017, accessed 16-mar-2017. [Online]. Available: [www.ros.org/is-ros-for-me/](http://www.ros.org/is-ros-for-me/)
- [3] —, "About ros," Web page, mar 2017, accessed 16-mar-2017. [Online]. Available: <http://www.ros.org/about-ros/>
- [4] Wikipedia, "Bsd licenses — Wikipedia, the free encyclopedia," Web page, mar 2017, accessed 16-mar-2017. [Online]. Available: [https://en.wikipedia.org/wiki/BSD\\_licenses](https://en.wikipedia.org/wiki/BSD_licenses)

## AUTHOR BIOGRAPHIES



**John Smith** received his BSc (Mathematics) in 2000 from The University of Maryland. His research interests include lasers and optics.



**Alice Smith** received her BSc (Mathematics) in 2000 from The University of Maryland. Her research interests also include lasers and optics.



**Bruce Wayne** received his BSc (Aeronautics) in 2000 from Indiana University. His research interests include lasers and optics.

## WORK BREAKDOWN

The work on this project was distributed as follows between the authors:

**Matthew Lawson.** Matthew researched and wrote all of the material for this paper.

## REPORT CHECKLIST

- ☐ Have you written the report in word or LaTeX in the specified format?
- ☐ Have you included the report in github/lab?
- ☐ Have you specified the names and e-mails of all team members in your report. E.g. the username in Canvas?
- ☐ Have you included the HID of all team members?
- ☐ Does the report have the project number added to it?
- ☐ Have you included all images in native and PDF format in gitlab in the images folder?
- ☐ Have you added the bibliography file in bibtex format?
- ☐ Have you submitted an additional page that describes who did what in the project or report?
- ☐ Have you spellchecked the paper?
- ☐ Have you made sure you do not plagiarize?
- ☐ Have you made sure that the important directories are all lower case and have no underscore or space in it?
- ☐ Have you made sure that all authors have a README.rst in their HID github/lab repository?
- ☐ Have you made sure that there is a README.rst in the project directory and that it is properly filled out?
- ☐ Have you put a work breakdown in the document if you worked in a group?

## POSSIBLE TECHNOLOGY PAPER OUTLINE

The next sections are just some suggestions, you may want to add sections and subsections as you see fit. Images and references do not count towards the 2 page length. Please use the `\section`, `\subsection`, and `\subsubsection` commands in your paper. Do not introduce hardcoded numbers. Use the `\ref` and `\label` commands to refer to the sections.

**Abstract** 01) ROS provides an OS for robots. It runs exclusively on linux; Ubuntu is suggested «- add citation 02) The main components consist of a) the pubsub communication system, b) something else and c) some final thing. 03) Roboticians and other users can interact w/ ROS via the following methods: a) C++ programs, b) ROS on linux, c) etc. 04) ROS offers a CLI, but no GUI. Gazebo offers a CLI and a GUI, but the GUI only exists as part of the simulation process. 05) The Open Source Robotics Foundation, hereinafter OSRF, distributes ROS under some free license, probably a variant of GPL. 06) Since ROS dominates the robot software market, it has a vast ecosystem of compatible software and hardware. 07) Researchers and robot industry participants use ROS extensively. For instance, a) foo; b) bar; and c) foobar. Big data uses remain limited. 08) Explore ROS more by visiting [www.ros.org](http://www.ros.org).

**1. Introduction** The aptly-named *Robot Operating System*, hereinafter ROS, provides a framework for writing operating systems for robots. ROS offers "a collection of tools, libraries, and conventions [meant to] simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms" [3].

**2. Architecture** If applicable include a description about architectural details. This may include a figure. Make sure that if you copy a figure you put the [?] in the caption also. Otherwise it is plagiarism.

**2.1. API** comment on the API which could include language bindings

**2.2. Shell Access** If applicable comment on how the tool can be used from the command line

**2.3. Graphical Interface** If applicable comment on if the technology has a GUI

**3. Licensing** The Open Source Robotics Foundation, hereinafter OSRF, distributes the core of ROS under the standard, three-clause BSD license, hereinafter BSD-3 license. The BSD-3 license belongs to a broader class of copyright licenses referred to as *permissive licenses* because it imposes zero restrictions on the software's redistribution as long as the redistribution maintains the license's copyright notices and warranty disclaimers [4].

Other names for BSD-3 include: a) BSD-new; b) New BSD; c) revised BSD; d) The BSD License, the official name used by the Open Source Initiative; and, e) Modified BSD License, used by the Free Software Foundation.

ROS has more than 3,000 components available from its distributed community, including proof-of-concept algos to industrial-quality software drivers. [2].

## 4. Use Cases

**4.1. Use Cases for Big Data** Locate and describe major usecases that demonstrate the technology while focussing on big data related use cases. Make sure you do proper references with the [?] command. Do not put URLs in the text.

**4.2. Other Use Cases** Some technologies may not just be used for big data, find other major use cases from other areas if applicable. Make sure you do proper references with the [?] command. Do not put URLs in the text.

**5. Educational material** Put information here how someone would find out more about the technology. Use important material and do not list hundreds of web pages, be selective.

**6. Conclusion** Put in some conclusion based on what you have researched

**Acknowledgement** Put in the information for this class and who may sponsor you. Examples will be given later





**Fig. 2.** The Main Elements of ROS - the Robot Operating System [2]