Jupyter Notebook vs Apache Zeppelin - A comparative study

SRIRAM SITHARAMAN[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: srirsith@iu.edu*

April 12, 2017

With the development of new technologies for the purpose of exploring, analysing and visualizing big datasets, there exists a growing demand of a collaborative platform that combines the process of data analysis and visualization.The idea of computer notebooks has been around for a long time with the launch of Matlab and Mathematica. Two such platforms: Apache Zeppelin and Jupyter notebook are taken in to consideration for comparison. Both of them were ranked against characteristics such as their ability to support multiple programming techniques, multi-user support and integrated visualization.

Keywords: Jupyter, Zeppelin, Notebook, Comparison

https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf

CONTENTS

1. INTRODUCTION - JUPYTER NOTEBOOK

Jupyter notebook, part of project Jupyter is the third version of IPython notebook. It is a web based interactive development environment and supports multiple programming languages (Python, Julia, R etc.) [1]. It started supporting Julia, Python and R in its initial release, hence the term Jupyter. The web based interative environment provided by Jupyter is facilitated through a notebook interface which contains the programming code written by the user as well mark down elements and outputs that are generated as result of the written code[2]. Hence, Jupyter Notebook documents allows a seemless view of the code written and the corresponding output (graphs,tables,etc.). The Jupyter Notebook app is a client-server application as shown in the figure 1 that can be run on a local machine for personal use and can be accessed without the internet, or can be installed in a remote server which can be accessed via an internet connection. The final component of the Jupyter Notebook is the kernel which performs the execution of code written by user. Jupyter comes with a native Ipython kernel (supporting Python), and also supports 80+ programming languages' kernels [3].

2. INTRODUCTION - APACHE ZEPPELIN

Apache Zeppelin [5], similar to Jupyter notebook, is also a web based interactive notebook but aimed towards supporting big data and data analytics. It supports multiple programming languages. It is useful in exploration of data, creation of visualizations and sharing insights, as web pages, with various stakeholders involved in a project. It supports a range of programming languages like Scala, Spark SQL, Shell, etc with the default being scala. Zeppelin has an architecture similar to Jupyter notebook with an exception that notebook in Zeppelin supports the integration of multiple programming languages in a single notebook. Zeppelin's notebook is shipped with some basic charts which can be used to visualize output generated by any supported programming language. Apart from that, it has the option of pivot charts and Dynamic forms that can be generated inside the notebook interface.

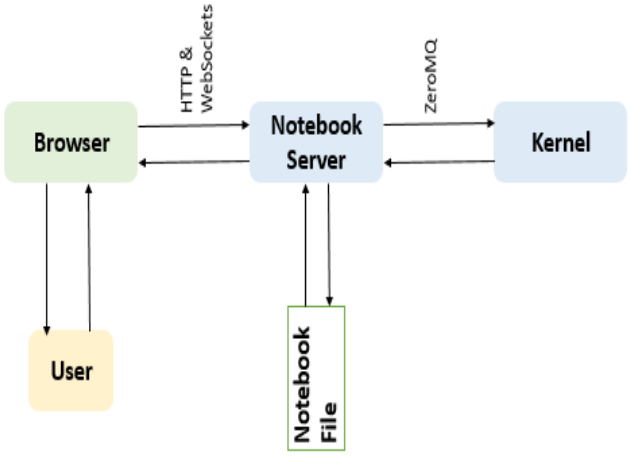

Fig. 1. Jupyter Architecture. The main components of the architecture comprises of the browser (user interface) which contacts with the Notebook server. The Notebook server can connect with multiple kernels.[4]

3. COMPARISON

This section would compare the following aspects of Apache Zeppelin and Jupyter notebook:

1. Interpreter configuration
2. Interface
3. Supported Languages
4. Visualization
5. Multi-user capability
6. Community Support

3.1. Interpreter configuration

Zeppelin interpreter is a language/data-processing-backend that can be plugged into Zeppelin notebook. Currently, Zeppelin supports over 30 interpreters such as Scala (with Apache Spark), Python (with Apache Spark), Spark SQL, JDBC, Markdown, Shell, etc [6]. Zeppelin has a separate Interpreter configuration page that you can be accessed for multiple language parameters. For instance, Spark home directory as well as spark master string can be modified to our preference. It would create the Spark Context automatically so you don't need to deal with it in each notebook.For example, to use Scala code in Zeppelin, "%spark" has to be included to load the interpreter. Apache Zeppelin provides several interpreters as community managed interpreters [7] which can be installed at once if **netinst** binary package has been installed. It also supports installation of 3rd party interpreters.

For Jupyter, IPython is the default kernel defined as Kernel zero, which can be obtained through the kernel **ipykernel**. Jupyter supports the use of 80+ kernels and [3] shows how each of them can be installed, required dependencies and the corresponding programming language version needed.

3.2. Interface

Zeppelin leverages a common UI framework with Bootstrap and Angular.js for the notebook interface. It is a easy to use interface with the support for creating dynamic forms within Zeppelin's notebook for which input can be obtained from a programming languages' output. Zeppelin provides an interface that is similar to RShiny's interactive web interface which allows user to manipulate in the front end rendered using Javascript with R running in the background [8].

Jupyter, on the other hand has a simple interface that is not user interactive as Zeppelin. When jupyter notebook is launched, the first page that is encountered is the Notebook Dashboard which shows the notebook files that has been created already and residing in the server. A new notebook can be created that is associated to a specific programming language's kernel (Python 2/Python 3/ R etc.) [9]. It's interface simple interface having a cell where the code can be written. Once the code is executed, the area below the cell would display the corresponding output.

3.3. Supported Languages

Zeppelin has the support for the 16 interpreters mentioned in Table 1. Since Apache Zeppelin has the default interpreter as spark which is a one of the major technology used for solving big data problems, the list includes interpreters like hbase, cassandra, elastic search etc. that works along with spark in solving big data problems.To overcome the limitation of these list of interpreters, Zeppelin provides the support for writing our own interpreter as described in [10].

Table 1. Programming languages supported by Apache Zeppelin

alluxio	file	jdbc	md
angular	flink	kylin	postgresql
cassandra	hbase	lens	python
elasticsearch	ignite	livy	shell

Jupyter, on the other hand has a huge list of about 80+ kernels being supported currently [3]. Though Jupyter has an upper-hand over Zeppelin over the number of supported programming languages, it lags behind Zeppelin in the support of using different programming language in the same notebook.

3.4. Visualization

Zeppelin provides the freedom to play around with various types of chart as shown in figure 2. It provides charting options by default for the output generated as a result of execution of code. The Apache Zeppelin community has been working on Project Helium [11], which aims to seed growth in all kinds of visualizations. This follows the model created by pluggable interpreters. Helium aims to make adding a new visualization simple, intuitive and can be accessed through a packaged code.

Jupyter, on the other hand has no charting options by default. Hence, it relies on existing charting libraries from the programming language that the notebook uses.

3.5. Multi-user capability

Zeppelin is still working towards providing multi user capability. Jupyter Hub [12] provides multiple user support for Jupyter. It

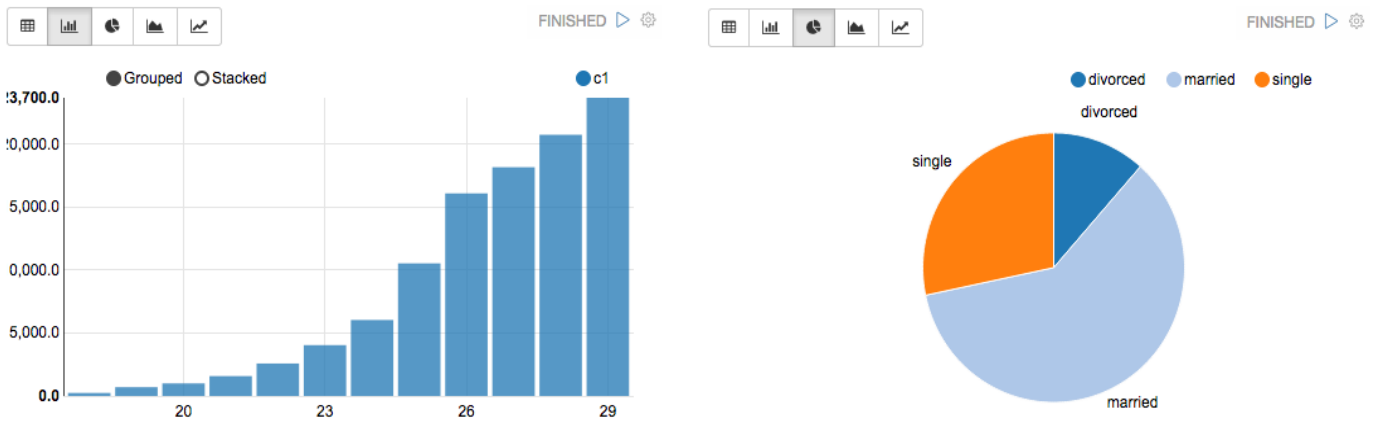

Fig. 2. Snapshot of Integrated graph - Apache Zeppelin. Different types of plots that can be readily viewed for the generated outputs in Zeppelin Notebook [5]

simple to setup because it leverages the Linux users and groups to provide authentication.Three subsystems make up JupyterHub as shown in Figure 3. Each user cannot touch or see any other users because it's restricted at the OS layer. However, Jupyter Hub has to be maintained in a single server which would provide access to multiple users.

1. a multi-user Hub (tornado process)
2. a configurable http proxy (node-http-proxy)
3. multiple single-user Jupyter notebook servers (Python/IPython/tornado)

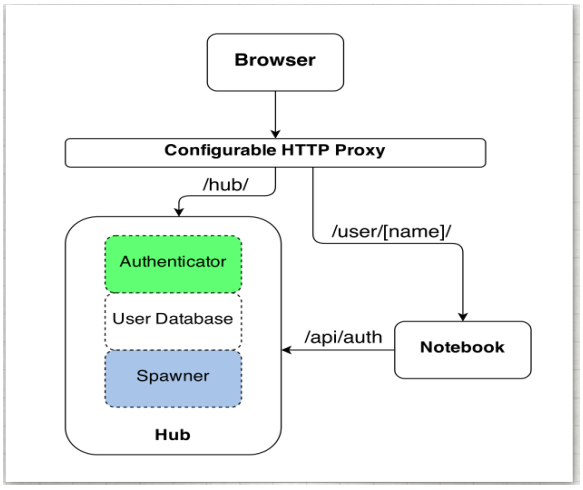

Fig. 3. Jupyter HUB Architecture.It allows for multiple users to interact with the Notebook server to access their respective Notebooks by means of user side authentication.[12]

3.6. Community support

Zeppelin is still in the incubation stage of Apache. It is progressing very slowly relatively to the status of Jupiter today. However, since Jupyter has a long history as IPython, there is lot of support for Jupyter in the online communtiy. A simple google search for the term **IPython** gives around 1.3 million results whereas for **Apache Zeppelin**, it is around 0.45 million. Apache Zeppelin is working with NFLabs, Twitter, Hortonworks, MapR, Pivotal, and IBM among many others delivering new features and fix issues in its platform [13].

4. ALTERNATIVES TO JUPYTER AND APACHE ZEPPELIN

4.1. Beaker Notebook

The Beaker Notebook is built on top of IPython kernel and was designed from the start to be a fully polyglot notebook [14]. It currently supports Python, Python3, R, Julia, JavaScript, SQL, Java, Clojure, HTML5, Node.js, C++, LaTeX, Ruby, Scala, Groovy, Kdb. It follows a cell type interface similar to Jupyter and Zeppelin and allows the user to use multiple programming languages across different cells in the same notebook. (i.e.) For example, the output of the code written in R in cell 1 can be accessed by a block of code written in Python in cell 2 for further manipulations.

4.2. SageMath

Sagemath is a is a free open-source mathematics software system [15] built for creating an open source alternative to Magma, Maple, Mathematica, and MATLAB. It is build on top of existing open-source packages: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R etc. It also offers a notebook type interface and the Sage Notebook recently moved to the cloud with SageMathCloud in collaboration with Google's cloud services.

5. CONCLUSION

The very need for experiments, explorations, and collaborations in scientific programming community is addressed by the evolution of these notebooks.Considering these into account, Apache Zeppelin is gaining upper hand over Jupyter in providing a fluid environment to solve big data problems apart from it being in the initial stages of defining multi-user support and relatively small community. With Zeppelin being a part of Apache community, it would be understandable that there would be constant

growth and updates. This can be strengthened from the fact that Apache Zeppelin is in the initial stages of developing Helium which would make visualization easy to use for big data problems.

REFERENCES

[1] Wikipedia, "Jupyter -debian wiki," dec 2016, [Online; accessed 23-March-2017]. [Online]. Available: https://wiki.debian.org/Jupyter

[2] Jupyter, "What is the jupyter notebook?" Web Page, jan 2017, accessed: 2017-03-02. [Online]. Available: http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

[3] ——, "Jupyter kernels," Web Page, feb 2017, accessed: 2017-03-02. [Online]. Available: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

[4] ——, "How ipython and jupyter notebook work," Web Page, feb 2017, accessed: 2017-03-02. [Online]. Available: http://jupyter.readthedocs.io/en/latest/architecture/how_jupyter_ipython_work.html

[5] Apache, "Apache zeppelin," Web Page, nov 2016, accessed: 2017-03-02. [Online]. Available: http://zeppelin.apache.org/

[6] ——, "Interpreters in apache zeppelin," Web Page, nov 2016, accessed: 2017-03-02. [Online]. Available: http://zeppelin.apache.org/docs/latest/manual/interpreters.html

[7] ——, "Interpreters installation," Web Page, nov 2016, accessed: 2017-03-02. [Online]. Available: https://zeppelin.apache.org/docs/0.6.0/manual/interpreterinstallation.html

[8] RStudio, "Shiny," Web Page, dec 2016, accessed: 2017-03-22. [Online]. Available: https://shiny.rstudio.com/

[9] Jupyter, "Ui components - jupyter notebook," Web Page, feb 2017, accessed: 2017-03-02. [Online]. Available: http://jupyter-notebook.readthedocs.io/en/latest/ui_components.html

[10] Apache, "Writing a new interpreter," Web Page, nov 2016, accessed: 2017-03-02. [Online]. Available: http://zeppelin.apache.org/docs/latest/development/writingzeppelininterpreter.html#make-your-own-interpreter

[11] Wikipedia, "Helium proposal," mar 2016, [Online; accessed 21-March-2017]. [Online]. Available: https://cwiki.apache.org/confluence/display/ZEPPELIN/Helium+proposal

[12] Jupyter, "Jupyterhub," Web Page, feb 2017, accessed: 2017-03-22. [Online]. Available: https://jupyterhub.readthedocs.io/en/latest/

[13] hortonworks, "Jupyterhub," Web Page, jun 2016, accessed: 2017-03-23. [Online]. Available: https://hortonworks.com/blog/apache-zeppelin-road-ahead/

[14] T. Sigma, "Beaker notebook," Web Page, dec 2016, accessed: 2017-03-25. [Online]. Available: http://beakernotebook.com/

[15] S. Foundation, "Beaker notebook," Web Page, jan 2017, accessed: 2017-03-25. [Online]. Available: http://www.sagemath.org/