Cookery: A Framework for creating data processing pipeline using online services

Branowski, M.; Belloum, A.

DOI
10.1109/eScience.2018.00102

Publication date
2018

Document Version
Final published version

Published in
IEEE 14th International Conference on eScience

Cookery: A Framework for creating data processing pipeline using online services

Mikolaj Branowski
University of Amsterdam
Amsterdam, The Netherlands
mikolajb@gmail.com

Adam Belloum
University of Amsterdam
Amsterdam, The Netherlands
Amsterdam, The Netherlands
a.s.z.belloum@uva.nl

***Abstract*— With the increasing amount of data the importance of data analysis has grown. A large amount of this data has shifted to cloud-based storage. The cloud offers storage and computation power. The Cookery framework is a tool developed to build application in the cloud for scientists without a complete understanding of programming. In this paper with present the cookery systems and how it can be used to authenticate and use standard online 3rd party services to easily create data analytics pipeline. Cookery framework is not limited to work with standard web services, it can also integrate and work with the emerging AWS Lambda. The combination of AWS Lambda and Cookery, which makes it possible for people, who do not have any program experience, to create data processing pipeline using cloud services in short time. (*Abstract*)**

Keywords*—*Function-as-a-Service (FaaS), AWS Lambda, OAuth, IFTTT, Cloud

I. Introduction

In order to make programming in using cloud services easier, Cookery has been developed in the context of PhD research work at the UvA [1]. Cookery framework allows users to develop applications that can connect features of multiple cloud service providers together. The Cookery approach is focused on the combination of multiple systems, this approach is similar in terms of functionality to IFTTT (If This Than That) [2].

Cookery enables developers to combine multiple cloud applications in an easy way. On top of that, Cookery uses its own Domain Specific Language (DSL) to make it more accessible for people without any programming experience. The cookery system comes with a couple of handy features to make using cloud application seamless and easy to use, among interesting: (1) Cookery is integrated with Jupyter notebook (2) it implements the OAuth 2.0 protocol for convenient authorization [3], and (3) it supports the use of AWS Lambda functions [4].

II. Cookery

Cookery makes it possible to combine cloud services using a high-level language. This high-level language, or Cookery language, has the same syntax as English, which makes it possible for users without any programming experience to easily create applications pipeline. A closer look at Cookery shows that it is composed of three layers.

- The first layer (Layer 1) is used by a user to create Cookery applications using the Cookery DSL language. In this layer data processing pipeline can be defined.
- The second layer (Layer 2) is for developers and instead of the previous layer; this layer makes use of the cookery Domain Specific Language.
- The third and last layer (Layer 3) is the Cookery backend and is also intended for developers. At this level, developers can implement protocols, which are for the activities and data, and communication with execution environments.

III. Application: Finding life trends using data analysis in Cookery

To demonstrate how to use the Cookery framework, we create a data pipeline in Cookery [5]. The use case is about analyzing Gmail in order to visualize life trends. Cookery framework offers a separation between the creating of general function by developers (layer 2) and the creation of user-specific programs by the end-user (Layer 1). The layers reduce the amount of code that the end-user has to write, but still allows flexibility. Layer 1 where the end-user will be interacting. The user will combine already defined activities to create a Cookery program.

The use case aims to create in Cookery a data processing pipeline similar to Stephen Wolfram 'The Personal Analytics of My Life' [6]. In our case the personal analytics is limited to analyzing the emails one of the authors. The data analytics pipeline combines a number of online services providing the initial data Gmail service, and Google analytics service. A visual representation of the data flow for the project is shown in Figure 1. This means that Cookery and the data pipeline within need to handle data from multiple sources, some of which need to be stored to a file or be kept in the program memory, and give visual results back to the user.

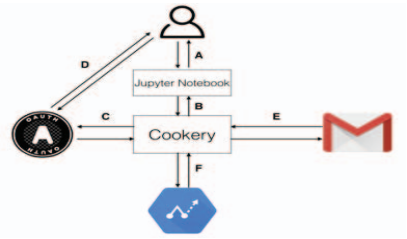

Figure 1: The data processing pipeline composed of two third party services (Gmail, Google analytics): A. Front-end communication between user and Cookery. B. Jupyter accessing cookery kernel. C. Cookery requesting OAuth for access. D. Login process of user and granting access. E. Retrieving emails from Gmail F. Analyzing language and semantics wit Google's Prediction API

The data pipeline can be broken in three steps

1. The authentication step using OAuth protocol (Figure 1 - C): The first step is to register the application at Google in order to obtain the credentials. After the registration, it is possible to download the 'client secret',

DOI 10.1109/eScience.2018.00102

which is stored on client side. In this case the application is limited to read emails and access the prediction API. The credentials are stored, so this verification only needs to be done only once. Unless the scopes get changed, the credentials expire or the user retracts the permission.

2. The retrieval of individual emails from the server (Figure 1 - E): In this phase, it is possible to specify characteristics of the request to read the emails an example could be the label 'INBOX' or 'SENT'. In the list of arguments of the request are the ids of individual messages that were sent, which can be retrieved by their id. For this example, we are interested in the body of the email, the time and the date and the list of recipients.
3. Implement the machine learning API (Figure 1 - F): This was done in a similar way as with the Gmail API. We focused on the pre-trained models of the API. These models are used to analyze and predict the language of the emails. The possible outcomes of the semantics analysis are positive, negative or neutral. The text of each email is categorized as either English, Spanish or French. The complete code implementing the data analysis workflow for the email analysis case is available in GitHub[1]. For the incoming emails we are only interested in the date and time and the number of emails. For the outgoing emails we are interested in the date and time, the participants, number of emails, semantics and languages.

After Cookery finished analyzing emails, we visualize the dictionary of analyzed emails. The plots are based on the last six month of authors personal emails which contain 470 outgoing and 1200 incoming messages. the results of the sentiment analysis are shown in Figure 2.

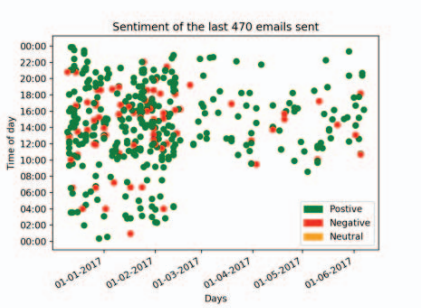

Figure 2: Results based on the number of emails and the time stamp of the email.

IV. CONCLUSION

The goal of the cookery framework[2] is to enable persons with the little programming background to define a data analysis pipeline and execute it online services from multiple cloud providers. To achieve this goal cookery had to address the authentication and authorization issues, in this paper we described extension to the cookery systems that allow for connections with cloud services using the OAuth 2.0 protocol.

We used the "life trends" example to demonstrate how to create a data processing pipeline in Cookery using online services namely the Gmail service, and Google analytics. When developing the "life trends" example we faced a couple of limitations of the free version of the Google analytics, the machine learning API has a 'User rate limit'. This limits the number of requests to the API to 650. Also, it was not possible to upload batch request. Every message has to be individually sent and analyzed, which might have an impact on the performance of the time critical data processing pipeline. In the "life trends" example, we only made use of service from a single service provider namely Google. Our next goal is to build data processing pipeline using service form different providers like AWS Amazon and Microsoft Azure. A first step toward this goal is described in the paper; we have developed an extension to use AWS lambda services[3]. In the future Cookery will be extended with more services from different cloud providers, to create a broader framework and to enable more developers to create applications. For example, an interesting extension would be with AWS DynamoDB or AWS RDS (Relational Database Service) to make it easier to create and manage databases using Cookery. When we combine databases with the functions of AWS Lambda, we can create more complicated applications and deploy them using Cookery. This also means that the toolkit of Cookery can be extended with more services and functionalities in future research.

ACKNOWLEDGMENT

The authors thanks Michael van Mill, Timo Dobber, and Dennis Kruidenberg Students at university of Amsterdam who implemented the extensions of the Cookery framework. This work is supported by projects EU H2020-777533 PROCESS.

REFERENCES

[1] Mikolaj Baranowski, Adam Belloum and Marian Bubak. Cookery: A framework for developing cloud applications". In proceedings of IEEE International Conference on High Performance Computing & Simulation (HPCS), 2015, pp. 635-638.

[2] B. Ur, M. Pak Yong Ho, S. Brawner, J. Lee, S. Mennicken, N. Picard, D. Schulze, and M. L. Littman. Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pages 3227-3231. ACM, 2016.

[3] Michael van Mill, A Cookery extension to simplify cloud service integrations, Bachelor Thesis, University of Amsterdam June 9, 2017, https://esc.fnwi.uva.nl/thesis/centraal/files/f704994072.pdf

[4] Timo Dobber, Cookery in AWS Lambda, Bachelor Thesis, University of Amsterdam June 9, 2017 https://esc.fnwi.uva.nl/thesis/centraal/files/f274795790.pdf

[5] Dennis Kruidenberg Finding life trends using data analysis in Cookery, Bachelor Thesis, University of Amsterdam June 9, 2017, https://esc.fnwi.uva.nl/thesis/centraal/files/f628826658.pdf

[6] Stehpen Wolfram. The Personal Analytics of My Life. url: http://blog.stephenwolfram.com/2012/03/the-personal-analytics-of-my-life/

[1] https://github.com/mikolajb/cookery-data/tree/master/

[2] https://github.com/mikolajb/cookery

[3] https://github.com/mikolajb/cookery-lambda