Report: 1. Aug 2020 - Apr 30. 2021, NIST Award # 60NANB20D162

Generalized Data Analytics Services

Geoffrey C. Fox and Gregor von Laszewski

I. Addresses:

Institution:	PI:	co-PI:
Trustees of Indiana University Office of Research Administration 509 E. 3rd Street Bloomington, IN 47408-3654	Geoffrey C. Fox Digital Science Center Indiana University 2401 N Milo B Sampson Ln Bloomington, IN 47408	Gregor von Laszewski Digital Science Center Indiana University 2401 N Milo B Sampson Ln Bloomington, IN 47408

II. The specific component MSE research grant program to which the application is being submitted:

Information Technology Laboratory (ITL)

III. Statement of Relevance and Benefit to the General Public:

Today's tasks require a plethora of analytics tasks to be conducted to tackle state-of-the-art computational challenges posed in society impacting many areas including health care, automotive, banking, natural language processing, image detection many more data analytics related tasks.

To avoid vendor lockin, we require a generalized composable analytics service framework that allows users to integrate their own services and those offered in clouds, not only one but by many cloud compute and service providers.

The proposed work will provide (a) a technology overview (b) a service integration framework to design and compose generalized analytics frameworks, and (c) demonstrations to use such generalized analytics services created by our framework.

Generalized Data Analytics Services

Geoffrey C. Fox and Gregor von Laszewski

1 Summary

We report the progress made in our work to lay out the approach to enable analytics as services and define composable analytics mechanisms that leverage the NIST Big Data Interoperability Framework (NBDIF). With this work, we address the needs posed explicitly by tasks in support of data analytics. The focus was placed on the investigation of how we can formulate and implement such analytics services, the definition of a suitable workflow for simplified service generation, the integration of OpenAPI from the start, the preparation for multi-cloud providers and the exploration of simple examples that highlight the use of our framework. Through this new interplay of cloud providers, services, and the use of concepts introduced by the NBDIF, we are able to integrate and leverage these technologies in a new way platform-independent way.

2 Achievements

We present next the summary of achievements conducted as part of this work:

Task 1 – Data Analytics Technology Assessment and adaptation to changes: While we previously have focussed on virtual machine placement we have this year expanded our investigation into container technologies with a focus on microservices. The focus on this investigation was first placed on the reuse of docker, docker swarm , and Kubernetes as part of the DevOps process to deploy them. Due to restrictions in the university environment and to avoid costly exploits in the cloud, we have created a locally independent testbed in which we can deploy such environments. We focussed mainly on creating our own environment with the help of Raspberry PIs as they allowed us to work outside the restricted network infrastructure provided by the university. It also allowed us to create a "Physically Distributed" Lab environment in which students participated as part of this activity. Unfortunately, Chameleon Cloud still did not pass our quality control as they changed their deployments without assuring that enough tests were conducted and tutorials that worked were distributed to us. As we have spent 4 months over the last year including this January to debug their infrastructure, we decided no longer to use it for this project. Instead, we used our Physically Distributed PI Clusters as well as Amazon, Azure, and Google cloud providers.
While evaluating docker swarm vs Kubernetes we observe that docker swarm is easier to install and use, but Kubernetes is by now the community-dominated technology.
Some of the major results from this work includes that (some of which are reverified from the last report period):

1. The use of OpenStack by community offered services (e.g CHamelen Cloud) can not replace services offered by companies focusing on production systems.

2. The dominance of container framework requires us to expand development towards microservices and the reuse within Kubernetes.

3. We have showcased that our frameworks can easily be integrated with jupyter notebooks. We have written a special jupyter notebook command that can start notebook services on any node within our Distributed Clusters.

The work conducted has been the basis of an article that is going to be published in COMPSAC2021 [1] (see artifacts section). Due to significant software changes on the operating side and the quality control issues within Chameleon cloud (outside of our control) the work on using Kubernetes has started but is not yet completed.

Task 2 – Generalized Formulation Framework for Analytics Services: We improved the previous work that we presented in the NIST BDRA WG while formulating Analytics REST Service. We used the work as a teaching framework and have produced easy-to-follow tutorials. We especially emphasized on the easy deployment of the framework As already previously reported the approach that is often used to develop REST services and is depicted in Figure **??** is unnecessarily complicated.

Figure 1: Component flow to specify an analytics service.

We provided a vendor-independent high-level multi-survive analytics framework. This framework leverages our multi-cloud provider that needed to be updated, as reported in Task 1. The complexity-reduced framework steps are showcased in Figure **??**.

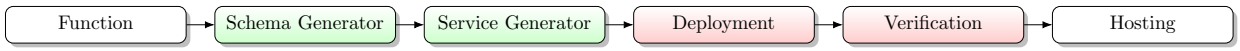

Figure 2: Component flow to specify an analytics service from a function specification.

We have worked on the development of more examples that have been uploaded to the GitHub repository as well as the development of tutorials.

Task 3 – Automated Analytics Service Specification Generator: We have improved the reliability of the generator as well as improved the management to shut down the services generated and run from the command line. This is achieved with the integration of a simplified service registry. To reduce the deployment effort we also replaced the dependency of the registry on a MongoDB deployment with a simpler file-based registry. This significantly simplified the documentation as well as the deployment as it can be completely automated without the special setup of a third-party product such as MongoDB. As MongoDB is also no longer supported in 32-bit the switch allowed us also to reuse the registry on 32 bit OSes.

Task 4 – Analytics Service Command Line Manager: To automate the Analytics service generation we developed a command-line tool that allows the users to easily create not only the OpenAPI from the command line but also the REST service, as well as the instantiation of the REST service.
The work we included this period is to

1. add a command line tool that allows us to burn a preconfigured cluster with all network configurations, and keys needed to stand up a development cluster for Raspberry PIs. This tool is unique and has been welcomed by the community.

2. add a command line tool to shut down the services

3. a command line tool that can deploy Kubernetes

Although we have added the Kubernetes deployment we still work on its verification, will add a dashboard to the deployment and verify the reuse of our generated services within the Kubernetes deployment.

Task 5 and 6 – Analytics Composability and Analytics Notebook Integration: Due to the ability to save outputs to files, our generated services can be reused in a composable fashion. As we have showcased reusability in notebooks, we anticipate that Jupyter notebooks are playing a major role in creating composable services.
Future work will include a documented example for the inclusion in python notebooks. We will also demonstrate the composability of the services.

Task 9 – Dissemination: We continued having great success by integrating the effort in multiple classes taught at IU. This is a nontrivial effort and allows us to address users from different communities due to IU attracting students with different backgrounds.
In addition to the class activities we have

1. attened all Working group meeitings orgianized by the NIST Big Data WG.

2. written a paper for COMSAC21 [1] (see artifacts section) reporting on this effort.

3. held a successful tutorial for showcasing the work on Apr. 23, 2021 with participants of the NIST Big Data WG as well as students from IU.

4. worked on the distribution of information on hackaday.io

5. are in preparation for distributing information on medium.com

All software has been uploaded to GitHub, and the documentation is available in GitHub as markdown files. The paper can not yet be publically distributed due to conference rules.

Artifacts

This project produced the following artifacts:

1. Using GAS for Speedy Generation of Hybrid Multi-Cloud Auto Generated AI Services, accepted at IEEE COMPSAC21, Jul. 12-16 `https://ieeecompsac.computer.org/2021/`

2. Tutorial: Autogenerating Analytics Rest Services `https://cybertraining-dsc.github.io/modules/cloudmesh-rest/`

3. Tutorial: Autogenerating Analytics Rest Services `https://cloudmesh.github.io/pi/tutorial/analytics-services/`
4. Preconfigured SDCards for Raspberry Pi Clusters `https://hackaday.io/project/177874-preconfigured-sdcards-for-raspberry-pi-clusters`
5. Command cms `https://cloudmesh.github.io/pi/tutorial/cms-commands/`
6. Headless Raspberry Pi Cluster from Mac's `HeadlessRaspberryPiClusterfromMac's`
7. Burning a set of pre-configured SD cards with a GUI for Raspberry Pis with Wifi Access `https://cloudmesh.github.io/pi/tutorial/gui-burn/`
8. Software: `https://github.com/cloudmesh/cloudmesh-openapi`
9. General Documentation: `https://github.com/cloudmesh/cloudmesh-openapi/blob/main/README.md`
10. Skikit Learn Example: `https://github.com/cloudmesh/cloudmesh-openapi/blob/main/README-Scikitlearn.md`
11. A tutorial is available from our cybertraining Web Site at: `https://cybertraining-dsc.github.io/modules/cloudmesh-rest/`

The artifacts are following the principles as outlined in our data management plan:

General Approach: All of the material are made available under Apache License in an open-source repository on GitHub.

Standards for making material available: All documentation and code is available in markdown on GitHub.

Availability: All material, software, documentation, and access to project-produced source code is available via GitHub.

Policy for redistribution: All the project-produced software and documents is freely re-used and re-distributed. The license is Apache. The paper published in COMPSAC21 will be made available through IEEE.