Publishing Platform for Geospatial Operations

Kate Keahey[1], Jonathan Pastor[2], Maverick Chardet[1]

[1]Argonne National Laboratory, [2]University of Chicago

Keywords: Geospatial platforms, Reproducibility, Cloud Computing.

Introduction

The last several years have seen an unprecedented growth in data availability with dynamic data streams from sources ranging from social networks to small, inexpensive sensing devices. The latter in particular is increasingly creating new sources of information: the proliferation of energy efficient, cheap and robust sensors, sometimes referred to as "second Moore's law", is now creating new opportunities for measuring various physical, chemical and biological characteristics of the environment. As small, specialized sensor devices, capable of both reporting on environmental factors and interacting with the environment, become more ubiquitous, reliable, and cheap, increasingly more domain sciences are creating instruments, composed of dynamic, groups of sensors whose outputs are capable of being aggregated and correlated to answer new questions. This new data availability creates an opportunity – and nowhere is it greater than in geospatial data science where this new, dynamic, data allows unprecedented insight into phenomena ranging from environmental to social sciences.

Much work has focused on creating venues for publishing and accessing data. However access to data in itself is not sufficient -- to turn data into information the data needs to be filtered, correlated, and otherwise analyzed using methods that are dynamically developed and constantly improved by a distributed community of experts. Further, the methods are often used to generate results with specific qualities of service, e.g., providing results by a certain deadline or ensuring a certain accuracy of the results. Delivering such qualities of service requires generic but often sophisticated tools managing the execution of such methods and ensuring their correctness.

We propose to develop an operations management platform that combines access to data and computation within one geospatial portal. In particular, the proposed platform supports the following capabilities:

- It allows users to *publish* new operations, captured as appliance, i.e., full representation of an environment as well as operation configuration and execution options. Using this representation, users are able to reproduce exactly the conditions of a method execution across multiple runs on the same hardware domain. Further, published methods can be identified and referenced by a Data Object Identifier (DOI) providing traceability of published results.
- Supports *automated deployment and execution* of the published operations within the parameters requested by the user, pertaining to the correct version of executed operation, and ultimately managing qualities of service such as time of delivery of the results.
- Operates on *multiple NSF-supported as well as commercial infrastructures*. The extent of platforms includes (but is not limited to) local platforms (such as e.g., the Roger system at NCSA), academic clouds in the public domain such as Jetstream (Stewart, C. A., Cockerill, T. M., Foster, I., Hancock, D., Merchant, N., Skidmore, E., ... & Vaughn, M. ,

July 2015.), and Chameleon (Chameleon Cloud Project, 2014), or commercial platforms such as Amazon.

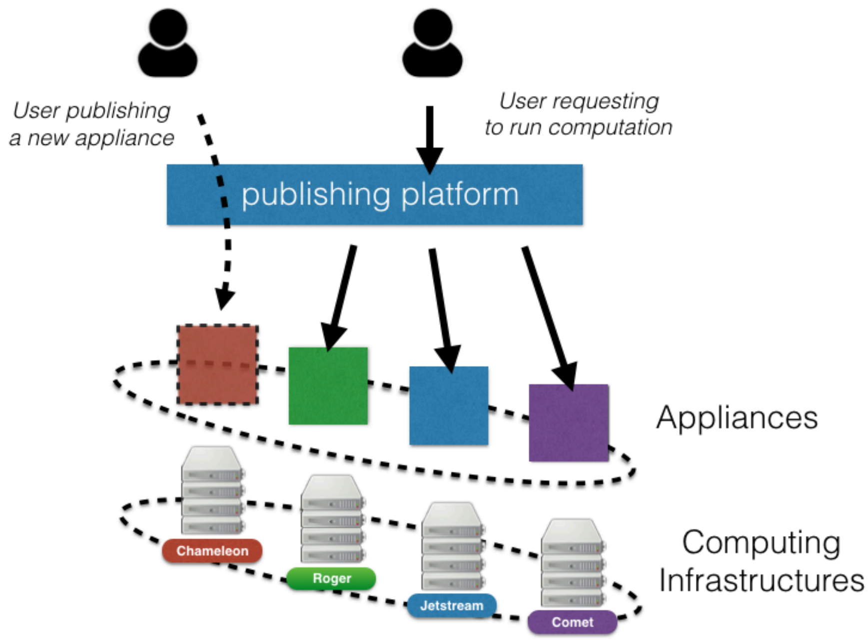

Figure 1: Illustration of the functioning of the operation publishing platform.

This paper is organized as follows. In Section 2 we introduce the concept of appliance. Section 3 describes the architecture of the operation publishing platform. We summarize in Section 4.

Appliances

The basis of our approach lies in the separation of an environment from hardware that became popularized by Infrastructure-as-a-Service (IaaS) cloud computing. Previously, hardware and software configuration were always combined; since operations implemented as programs rely closely on the environment in which they execute and this environment was static and administered by resource provider, the designer of the operation could in practice exercise little control over the configuration and its suitability as container for the developed methods. With the increased popularity of virtual machines (VMs) and systems supporting them, such control became possible so that the user can now manage different versions of an environment independently of the provider. Further, the representation of the environment can be saved, and automatically restored, and moved from provider to provider -- containers that allow this separation are called appliances.

More formally, an appliance is a representation of an environment that can be automatically and accurately recreated on a set of resources supplied by a set of providers. Such representation can be implemented for example in terms of Docker containers, KVM virtual machines, or bare metal images as used in the Chameleon system. Each of those implementations represents different trade-offs in terms of deployment time, performance, and accuracy. A single appliance, representing e.g., a ubuntu 14.04 distribution can thus be implemented in multiple formats not all of which will be suitable for all providers. In addition, automatic deployment implies the presence of tools that allow multiple deployed environments share the same context as needed e.g., to deploy virtual clusters where e.g., Torque headnode needs to know about worker nodes and vice versa.

Approach

The architecture of the system is shown in Figure 2 and supports two roles in the system: operation publisher and operation consumer.

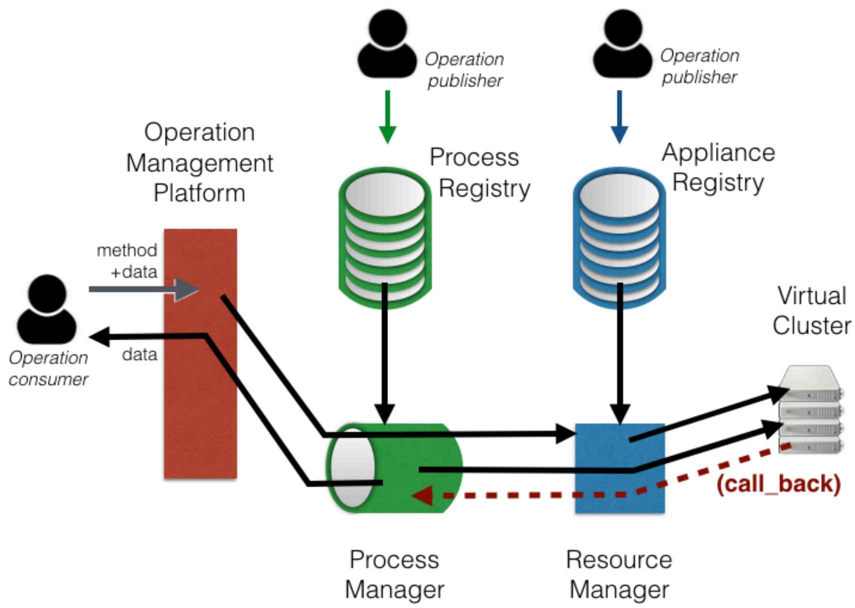

Figure 2: Architecture of the operation publishing platform.

The architecture consists of the following components.

Operation Management Platform serves as the end-user endpoint; it is in charge of translating user's interactions into computing requests that are send to the *Process Manager.* The platform interacts directly with user-facing infrastructure, such as a portal.

The *Process Registry* allows users to define entries linking operations to actions that need to be taken to implement them. A complete entry links the operation to information such as the appliance in which the program implementing the operation resides, links it to data sources (and/pr filters required by data sources), different configuration options, etc.

The Process Manager creates an execution of a process using information stored in the *Process Registry* and manages the workflow of a process execution, including running it and ensuring that results are reported as desired, as well as reporting errors of execution. Before an execution takes place the Process Manager needs to ensure that conditions for an execution are met; for example that the required appliances are deployed.

The *Appliance Registry* allows users to create entries for appliances. The entries include a set of pointers to disk images and contextualization scripts that can tie the deployed images into shared constructs, such as virtual clusters. To ensure reproducibility, changes made on an appliance are versioned; logical information is kept in the registry, while the versioning of appliance representations is delegated to the image service used by the different providers.

The *Resource Manager* deploys contextualized appliances based on appliances committed to *Appliance Registry*. An IaaS provider can only deploy appliances that it supports so that the reach of an appliance is limited by the supported implementations. The Resource Manager decides which infrastructure will be used to deploy the appliance. The decision can be influenced by users if they provided hints in the process metadata to optimize deployment and reflect their preferences.

The *operation publishing workflow* takes place as follows. An operation publisher develops and tests a new geospatial operation. The user then uses a dedicated tool to generate appliances representing the environment required by the operation (e.g., a Hadoop cluster). The generated appliances will span all the platforms the user wants to cover and are uploaded to appliance repositories corresponding to those platforms (e.g., Roger or Chameleon). Meta-data describing the existing appliance implementations is then uploaded to the Appliance Registry. Next the operation publisher describes a set of actions relating to the job execution as well as pre processing and post processing action, which can range from setting of environment variables to copying data to preserving results in specific repositories in the Process Registry. In conjunction, those two descriptions will allow the system to automatically execute the operation on demand.

The *operation executing workflow* takes place as follows. The operation consumer accesses the operation platform and requests the execution of a specific operation version. The platform contacts the process manager and requests the execution of the operation. The Process Manager looks up the operation in the Process Registry, finds the program and its meta-data that the operation corresponds to, and examines its dependencies. The description includes meta-data on which environment is required for execution. If the environment is already present in the system, the job is executed in the existing environment. If however, the environment has not yet been deployed, or the deployment is experiencing heavy load, the Process Manager requests the Resource Manager to deploy the environment on resources sufficient to provide the requested quality of service. Upon compliance, the Process Manager executes the program. In each case, the Process Manager ensures that all the pre- and post- processing actions are carried out correctly.

Summary

In this paper we describe an operation publishing platform that provides a way of formalizing, sharing, and executing new operations easily. Operations are in effect turned into data that can be described, referenced, and versioned. Our publishing platform leverages the concept of appliance, which represents an environment that the operation depends on. The ability to deploy this environment as needed is critical to providing automated execution of operations. Ultimately, our goal is to not only deploy operations but also ensure their execution with qualities of service specified by the user.

References:

Chameleon Cloud Project (2014). *A configurable experimental environment for large-scale cloud research*. Retrieved June 13, 2016, from https://www.chameleoncloud.org/

Stewart, C. A., Cockerill, T. M., Foster, I., Hancock, D., Merchant, N., Skidmore, E., ... & Vaughn, M. (2015, July). Jetstream: a self-provisioned, scalable science and engineering cloud environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure* (p. 29). ACM.