AUTOMATED SERVICE BUILDER FOR SEMANTIC SERVICE ORIENTED ARCHITECTURES – ASB

Bernard Valentin, Matthieu Melcot, Leslie Gale

Space Applications Services

ABSTRACT

We propose a novel approach for the execution of EO Products Processing Chains. The processors are deployed into a Cloud Environment that brings scalability and flexibility. In addition, each processor is interfaced through OGC Web Processing Services (WPS), leading to a modular multi-mission processing platform.

Index Terms— IPF, Processing, Cluster, Container, Cloud, OGC WPS, Semantics

1. INTRODUCTION

The framework used for the delivery of EO Services has evolved from Server-based to Grid-based processing environments (e.g. G-POD[1]), and onward to Cloud Computing. ESA at ESRIN within the context of several projects has been investigating and applying OGC standards, and more specifically the Web Processing Service (WPS[2]) Interface Standard that provides rules for standardizing inputs and output for geospatial processing services. These standards are adopted in various domains such as Industry, Government and Academic, but still underused in EO Services.

ASB will establish a generic concept and apply it to Instrument Processing Facilities (IPF's). IPF's are responsible for transforming instrument raw data into higher-level products by means of processors forming end-to-end Processing Chains. Each processor applies algorithms to correct, transform, merge data products or extract specific features from them. General opinion is that IPF's are very mission dependent and not suited to an automated build process. ASB will demonstrate current capabilities and make recommendations for further automation.

In the last five years, the enthusiasm of the IT sector for Big Data fostered the development of Cluster-based computing platforms such as Hadoop and Spark, and Cluster Management frameworks such as Mesos. Additionally, Cloud technology led major EO actors such as NASA JPL[3] to experiment the distribution of image processing into the Cloud.

However, there are still many operational IPFs and similar implementations of Processing Chains used at prototyping level that were designed and deployed prior to the Cloud era that do not take advantage of the distribution of the processing power on a Cluster. They are typically deployed into static architectures[4], where updates are tedious, and scalability is hardly possible. ASB[5] adopts the latest technologies promoting the work of ESRIN to overcome some of the persistent problems with existing IPFs by proposing a scalable and dynamically re-configurable platform for the execution of IPF's, or of any resources demanding processing facilities. Also, and for the intended users of ASB of great importance, ASB will focus on the (automated) building of workflows from user specifications making the deployment of the processors in the cloud transparent to users, but under their control.

2. OBJECTIVES

The main goal of the platform is to provide a dynamic, scalable multi-mission (automated) processing environment. To that end, the following objectives are addressed:

Scalability – EO products are huge and processing them requires great amounts of computing power, network and storage. The infrastructure shall be able to scale and adapt to the users needs in term of products generation and get rid of the classical infrastructure limitations.

Automated Generation of Workflows – Although endeavors have been made to standardize the interfaces among the processors and the processes — see the Generic IPF Interface Specifications[6] — disparities remain between processor interfaces of different missions. To automate the generation of workflows — that is, the organization of processes with defined inputs and outputs, implementing a processor — a generic interface for all the processors is needed to provide modularity and make it possible to share processes between different processing chains and missions.

Generic Orchestration – The orchestration of the processes is closely related to their related missions. A generic mechanism, adaptable to the specific orchestration baselines, is essential to run workflows from different missions into the same platform.

IPF Evolution – Traditional IPF's are static and upgrading the workflow definitions or the processes are not easy tasks. A simplified means to deploy new (versions of) processes and edit the workflows accordingly is therefore essential.

Monitoring and Control – Today's IPF's do not provide fine-grained control over the workflows execution. The solution shall provide advanced monitoring and control capabilities to overcome this limitation.

3. CONCEPTS

Cloud Environments – A scalable infrastructure where new Virtual Machines (VM's) can be deployed, started and stopped, automatically or on-demand to cope with changing computing needs. A Cloud environment also provides an elastic storage capacity which allows adapting the volumes seamlessly during the operations.

To benefit fully from parallel programming models such as MapReduce with Hadoop, the processors must be designed taking this paradigm into account. This is obviously not the case for the vast majority of the processors used today, including the ones running in the IPFs. Moreover, it is not necessary to adopt this way of developing and operating processors when input parameters are already fragmented in units of manageable size. This is in particular the case of EO products that are traditionally packed in tiles with limited surface coverage. To cover a bigger area, the contained and the intersecting tiles can be processed separately and the individual results merged to generate the final product.

In its initial version, ASB will put the focus on the dynamic deployment, the load balancing, and the orchestration of atomic processes. In order to anticipate evolution in the technologies and platform offerings, attention will be paid to integrate abstraction layers, allowing, e.g. switching seamlessly between cloud providers or adopting alternate programming models. These capabilities are made possible by OpenSource tools such as Apache Mesos (resource management and scheduling), Apache Zookeeper (cluster management) and Apache LibCloud (unified API for public and private clouds, including OpenStack).

Flexible Orchestrator – A fully customizable workflow engine provides the necessary flexibility to adapt to the multiple workflow patterns used in the IPFs, such as Parallel Split, Synchronization, and Exclusive Choice. When executed, workflows are transformed from a static to a dynamic representation in order to gain the ability to reconfigure themselves. This allows conditional branching and iterative operations needed in Products such as Calibration Products.

Data Types Semantics – Care must be taken that the processes chained in workflows are compatible and in particular that outputs of a given process may be used as inputs for the subsequent process. In a platform that allows the editing workflow definitions, it is highly recommended to include a mechanism that verifies the consistency of the dataflow between processes. An ontology containing the data types available is used to provide clear inputs and outputs signatures.

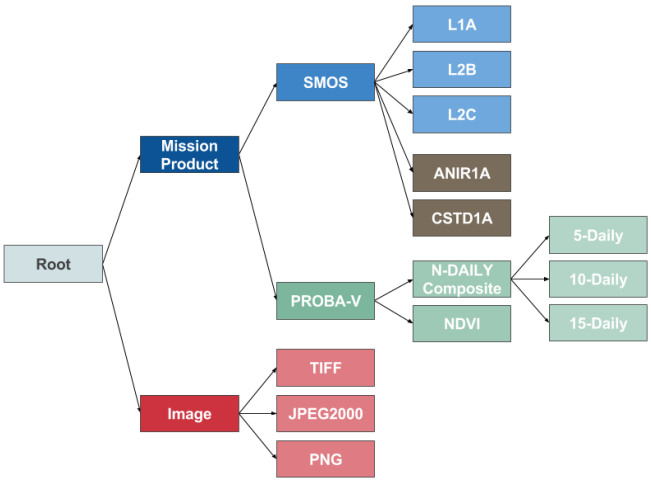

Fig. 1. Data Types Ontology (Fragment)

The data types ontology is used to verify the compatibility of input and output process parameters. This mechanism must have the knowledge of the broader, narrower and equivalent data types. For example, JPG and PNG are both Image types. A JPG output may be provided to a process that accepts an Image as input but not the other way around. The ontology can be enriched at any time with new data types, still paying attention that the modifications do not invalidate existing workflow definitions.

Processes-Data Locality Paradigm – Processes size is usually more than one order of magnitude smaller than the Data size. Furthermore the same Process is often applied on a large list of collocated data. The classical approach of processing remote data by first fetching the data is cumbersome.

A more convenient strategy leads to bringing the processes close to the data to be processed. This method, facilitated by the use of a Cloud Environment, highly reduces the data transfers that are not only time but also money (in the case of public clouds) consuming. Also, placement constraints on the processes deployment enforces the minimization of the data transfers between processes, especially in the case of use of huge intermediate or auxiliary data, such as the 8 Gigabytes SMOS L1B G-Matrix. The dynamic placement of processes is enabled by the Linux Containers technology.

Editable IPF's – A Workflow Editor provides the operators with a graphical means to create and edit workflows from a list of available tasks (the internal representation of an executable process). This is the first step towards an automated build and will greatly simply the effort required by the user in the prototyping and deployment of new IFPs.

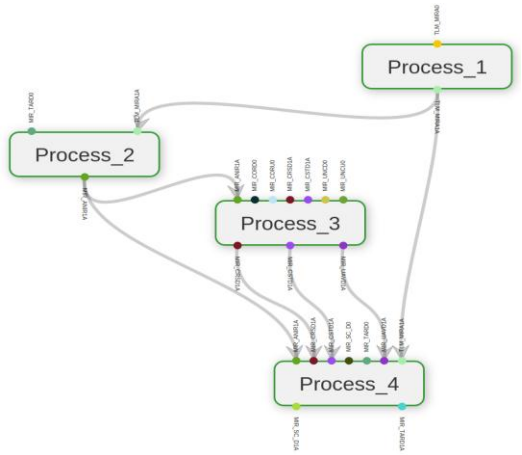

Fig. 2. Workflow Editor

OGC WPS – This gives the ability to interact with a specific process through an ad-hoc interface inside a standard Web Service. Processes are executed through a standard HTTP request, where URL's representing the input data files are provided as inputs. Process outputs may also be fetched through URL's.

4. ARCHITECTURE

4.1. Static Architecture

The architecture of ASB relies heavily on a collaboration between web-services where each one assumes clear responsibilities.

Figure 3 depicts the architecture of ASB. Operators and end-users interact with the ASB Core Components via their user interfaces. These components may be co-located or distributed. Their resource needs are relatively low. The lower part of the figure represents the processing components, meant to be deployed in a Cloud Environment.

The core services communicate with each other through the REST interface they are exposing. This brings the possibility to share the resources between the services and to update specific components individually without impacting the rest of the platform.

Product generation requests are received by the Service Builder core component. The Service Builder obtains workflows, processes and data types definitions from the Knowledge Base and constructs a specific workflow execution order. The order is issued to the Orchestrator component that is in charge of organizing and managing the execution of the processes via the Task Manager.

The Distributed Task Queue ensures the distribution of the process invocations over the Cloud where processes stemming from different workflows are wisely distributed according to the availability of the nodes and the current location of the input data.

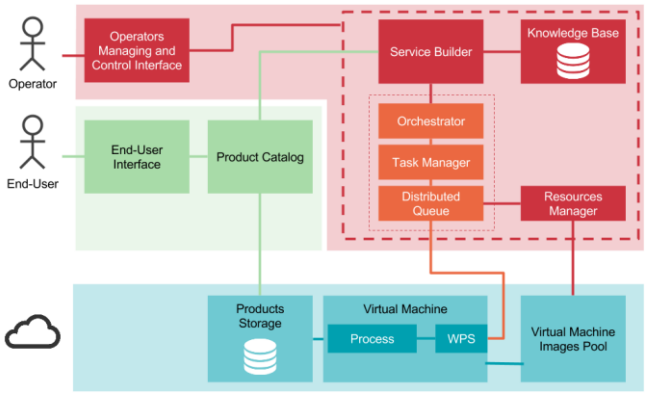

Fig. 3. ASB Architecture

The Resources Manager deploys the required processes from a pool of Virtual Machines (VM's) and container images. The processes sit behind WPS interfaces allowing them to be addressed in a seamless manner.

ASB Core Components are implemented in Python. The Product Catalog stores its data in a PostgreSQL/ PostGIS database. The Service Builder, the Orchestrator, the Task Manager, and the Distributed Queue are asynchronous components that use polling mechanisms. Containers are used to replicate process specific environments in the VM's, with the benefit of sandboxing the resources.

4.2. Dynamic Architecture

The open-source cluster manager Mesos provides the scalability of the platform. Mesos allows abstracting the computing resources and making them available to the applications built on top of it.

This means that the same cluster can be used to execute the processes from the ASB workflow, but also those of an Hadoop (or Spark) environment in cases some ASB processes need it internally. The balancing of the resource allocation between ASB and the Hadoop (or Spark) environment is configurable at the cluster manager level.

5. BEHAVIOUR

There are two key ASB mechanisms. The first one is used to configure processes and workflows platform. The second mechanism takes place at the time a product generation request is issued.

5.1. Workflow Editing

The constitution of a new workflow follows the steps presented in figure 4.

1. The operator provides the URL and name of a new OGC WPS Process. ASB automatically fetches the process description and generates an initial version of the process definition.

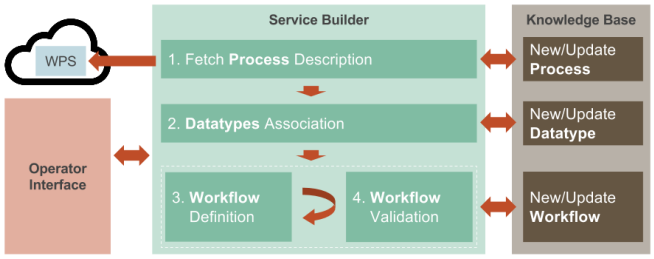

Fig. 4. Workflow Editing Steps

2. The operator augments the initial process definition, e.g. by associating the input and output parameters with data types. New data types are defined if necessary.
3. The operator graphically combines processes to define a workflow (see figure 2).
4. ASB automatically validates the associations by e.g. verifying the compatibility of the connected output and input parameters. At each step, the definitions are persisted in the Knowledge Base.

5.2. Workflow Execution

Building a Product requires the Service Builder to execute the sequence of steps depicted in figure 5.

1. Knowledge Elements, including the workflow definition and the related process definitions, are retrieved from the Knowledge Base.
2. The necessary resources are deployed in the Cloud Environment.
3. A workflow execution order is built and handed over to the Orchestrator.
4. When the requested product is ready, this is registered in the Product Catalog and a notification is issued.

5. CONCLUSIONS AND FUTURE DEVELOPMENTS

Following ESA's perspective to bring user's processors to the huge amount of exploitable EO data, ASB will provide an environment to host custom processors and allow testing and sharing of new workflows and algorithms.

ASB will implement technology and methods for distributed EO Processing Chains offering full support to users for building workflows incorporating processes and (big) distributed databases. This will significantly simplify the creation and management of processors that form an integral part of the IPF. It will increase the reliability of the operational IPF and reduce costs through standardization of the interfacing whilst allowing a transparent interface to the commercially driven processing environments currently led by Hadoop and Cloud processing technologies.

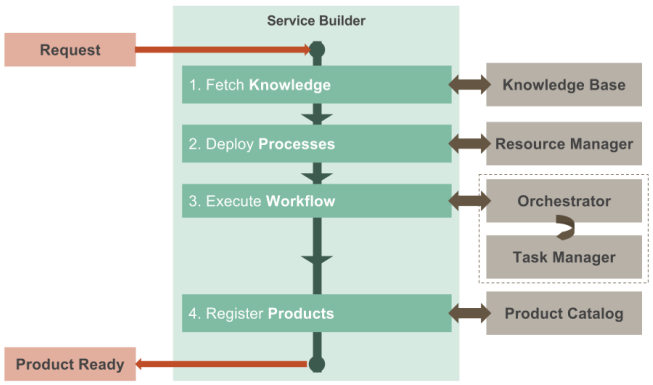

Fig. 5. Workflow Execution Steps

ASB scalability means that it can be used from the prototyping of retrieval algorithms by the scientific community, where execution could be on a local computer, through to the instantiation of a complex operations IPF by the IT community taking advantage of cloud processing capabilities.

The concepts and techniques used in ASB are not limited to the processing of EO data. In addition to enhancing the platform, future developments will also address the applicability in other domains.

The structure of the ASB platform leaves the way open to future extensions where users are closely involved in the creation and sharing of new processes and processors.

A programmatic interface such as iPython Notebook or those provided by Google Earth Engine in addition to the Workflow Graphical Editor would represent a valuable enhancement in order to meet the scientific expectations in terms of customization, sharing and reuse of the algorithms to process data.

The ASB concepts will be validated and compared using existing processors from SMOS and PROBA-V.

6. REFERENCES

[1] ESA Earth Observation G-POD home page, http://gpod.eo.esa.int/

[2] "Web processing service 1.0.0," OGC 05-007r7, OGC, June 2007.

[3] AWS Case Study: NASA/JPL's MER and CARVE Missions, https://aws.amazon.com/fr/solutions/case-studies/nasa-jpl/

[4] "Sentinel-2 Instrument Processing Facility Technical Specification," SRS S2-PDGS-TAS-DI-BPDP-CCTS-IPF, April 2012.

[5] ASB project page on ESA Research & Service Support, https://wiki.services.eoportal.org/tiki-index.php?page=ASB

[6] "Generic IPF Interface Specifications," ICD MMFI-GSEG-EOPG-TN-07-0003 , ESA, August 2009.