

# The Management of Software Service Workflow based on Cloud Environment

Shengwu Gao, Guangtai Ding, Lingyu Xu

School of Computer Engineering and Science, Shanghai University, Shanghai, 200072, China

**Keywords:** cloud computing, SSW, marine resource, RESTful Web Service.

**Abstract.** Based on the concept of traditional workflow and the thought of SaaS in cloud computing, puts forward a new concept named software service workflow. An application system based on this concept was built for processing marine resources automatic. This system put forward a visual customization interface to custom a software service and a monitoring interface for monitoring and interaction with the software service workflow that have being submitted. The information about the customized workflow is saved in a XML file. We build a workflow engine based on RESTful Web Services framework to analysis the XML file and drive the workflow execution on cloud. A software service workflow contains some data resource and some applications which we named computing model. In order to build a correct workflow, we need to solve two problems: one is the matching between the data and computing mode, another is the correct structure of workflow which can be driven by the workflow engine.

## Introduction

With the development of cloud computing, it have proved that this technology have its advantage in vast amounts of data computing. This characteristic of cloud computing promote it being used in the infrastructure of the platform and the management of traditional resources, such as meteorological application [1] and the management of digital rain forest [2]. The workflow technology have being used to make work more effective. We put this technology in cloud computing to process marine resource. In order to achieve this goal, we need a system to manage the resource and workflow in cloud. There are some systems like this available, such as: Kepler [2], Tirana [3].

In this paper, we give a detailed description of Software Service Workflow (SSW) and solve three main problems: match between SSW nodes, the checking of SSW and interaction with SSW.

## Software service workflow

Traditional workflow is a computational model of work sequence. It tells computers to do jobs according to priority automatic. In this way, all stages and operations a user would normally carry out manually become automated. With its advantage in automatic processing and the thought of Software-as-a-Service(SaaS) in cloud computing, we put forward a concept named SSW) Using this concept, complex and huge number of executable program in the cloud will be abstracted as calculation model which can be visually used by users.

### The Definition of SSW

A customized SSW is a model of GAG [5] (Directed Acyclic Graph). A SSW we can define like this:  $S = (M, D, \text{ and } R)$ .

$M = \{m_i \mid m_i \text{ is a result set of computing modle in cloud}\};$

$D = \{d_i \mid d_i \text{ is a result set of data and data resource in cloud}\};$

$R = \{(d_i \rightarrow m_i \mid m_i \rightarrow m_j) \mid d_i \text{ or } m_i\text{'s output is the output of } m_j\};$

On these formulas,  $M$  is a result set of computing model in cloud, and  $m_i$  represents one computing model. A computing model is a processing procedure in cloud and its data input is  $d_i$  or output of  $m_i$  and data output is a data which can also be a data input.  $D$  is a result set of marine resource data storage in cloud which need to be process.  $R$  is the relations between the nodes in a SSW. The symbol before is as a data input and another is the processing procedure.

### The Structure of SSW

From the definition of SSW, we can know that a software service workflow is not just the simple addition of the basic computing model. There are relationships of serial, parallel, merging between multiple models. In figure 1, we list two basic structure of SSW which can describe the basic relations between models. Figure I represent the simple serial relationships between models.

The model a's output is the input of b. Figure II is a little sophisticated. Model a process its data and produce an output. This output data is input of model b and c. When model b and c run has completed. Their outputs is inputs of model d. Model d processes the both outputs and get the final result. Between model a and model b is a serial relationship. Model b and model c execute parallel, and their output merge by model d. A complex SSW is formed by lots of computing models and marine data with relationships like figure 1.

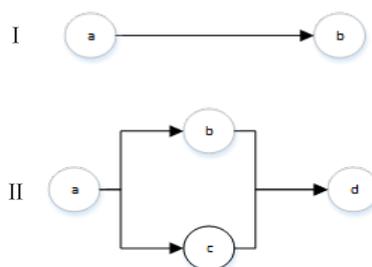


Figure 1 Basic structure of a SSW The Management of Software Service Workflow

With the thought of SSW, we build an application system to process marine data resource. The system's framework shows in figure 2. In this system, we provide the visual interface for user to upload data and model, customize workflow and monitor the status of the submitted workflow. The resource management centre manage the resource in the cloud, include the data and models user uploaded and the hardware resource that support the cloud to ensure it works. The workflow engine is a middleware between the visual platform and the Hadoop platform. It analyses the workflow submitted by user and drive the workflow running on the Hadoop platform. In order to make the system work effective, we need to solve three problem. First, we need to ensure that the data node and the model node can match well. Second, we have to check the structure of the workflow which will be submitted correct. Last, the system need a set of interactive mechanism while the workflow running.

### The Match between Nodes

In a SSW, each computing model has a data input interface and a data output interface and each data also have its data format. The interface between data (or the output interface of model) and the input interface of model should match. If not, the model cannot process the data and the SSW cannot run on the Hadoop platform. To solve this problem, we add a data format interface in the uploading procedure. This information will store in database with the other information of the resource.

While user customize SSW in the visual custom interface, the resource icons represent the real resources can free drag to the custom panel. The line between the icons represent the relation between nodes. While connecting two icons, the system reads the resource information from database and check the two nodes' data interface check whether the two node data interface match. If match, the two icons can connect. Otherwise, the two icons cannot connect and the system give the error message to remind user to check the math between the two nodes.

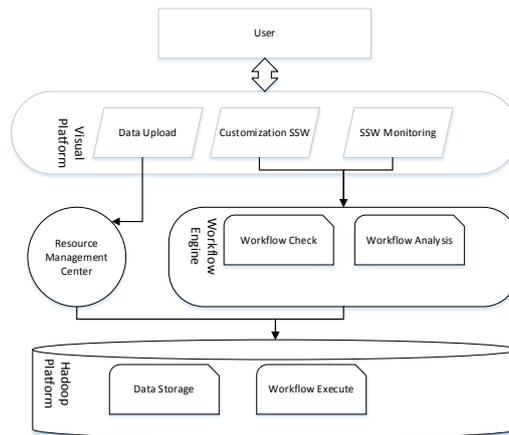


Figure 2 Software Service Workflow System Framework

### Check the Structure of SSW

The information of successfully customized workflow is saved in an XML file which is stored in the database with other information about the SSW. The format of the XML file is described in figure 3.

```

<WorkFlow id="flow-id " name="flow-name">
  <Interfaces>
    <route id="route-id" from="resource-id1" to="resource-id2"/>
    .....
  </interfaces/>
  <ComputeNodes>
    <computeNode id="resource-id" name="resource-name" />
    .....
  </ComputeNodes/>
  <DataNodes>
    <datanode id="resource-id" name="resource-name"/>
    .....
  </DataNodes/>
</WorkFlow/>

```

Figure3 the Format of SSW XML File

The workflow engine use the XML file to analysis the data and computing model in the SSW and the structure of the SSW. In figure2, the root element Workflow's property id is the decisive representation in the cloud. The child elements Compute Nodes and Data Nodes represent the data resource and computing models used in this SSW. Their property id is resource or model unique symbol in the database. This property can help the workflow engine to find the storage position in cloud. The child element Interfaces is describe the relations between model and data (or model and model).

Before the SSW submitted, we need to ensure the structure of the SSW correct. To analysis the XML file we can know the structure. From the definition of the SSW we know that the structure of the SSW is a DAG. So, a correct workflow structure should be that has one end, no cycles, no existing sub graph. We can analysis the child element Interfaces to get the SSW structure. Then, check the structure using graph algorithm to find whether there error conditions in the three conditions happy in the structure. If there one or more happy, the system will give an error message. Otherwise, the SSW can successfully submit.

### The Management of Workflow Interaction

REST (representational state transfer) is an architectural style of a distributed hypermedia system. Doctor Roy Fielding firstly put forward this concept in his doctoral dissertation in 2000. REST refers to a group of frame constraints and principles. We use URI (universal resource identifies) to locate and represent different operations on resources. HTTP protocol is a stateless protocol. This means that every state about resource and operations stored in server. Visiting a website represent an interactive between client and server. And this interaction make the server states change. The tools client use to change state are for verbs in HTTP protocol: GET, POST, PUT and DELETE [6].

The presentation layer module, the service flow engine and the Hadoop platform using different language and architecture. We use RESTful Web Services to manage the interaction with

the SSW while they running in cloud. The main functions of the management of workflow are query, pause, resume, termination and delete of the workflow. The workflow engine release the URIs represents the operations on the workflow along with apt methods of HTTP protocol. The APIs for operations of SSW show in table 1.

Table1 Operation URIs Format

Operation	HTTP Method	URI Format
Submit	POST	http://{IP}/SHU/rest/flow/
Termination	PUT	http://{IP}/SHU/rest/flow/
Pause	PUT	http://{ IP}/SHU/rest/flow/
Resume	PUT	http://{ IP}/SHU/rest/flow/
Monitor	GET	http://{ IP}/SHU/rest/flow/{uni_id}

### Simulation Experiment

The application system is running on dedicated servers in National Marine Information Centre in Tianjin. Now user can upload data, customize the SSW and interactive with the SSW on the visual interface.

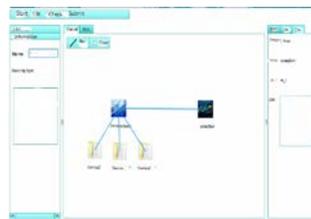


Figure4 SSW Visual Custom Interface

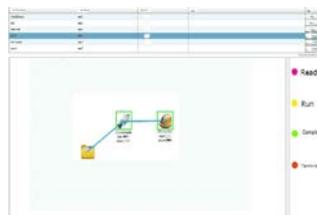


Figure 5 SSW Visual Monitoring and interaction Interface

Figure4 shows the visual custom interface. Users can use this interface to assemble data and computing models available in cloud and customize the SSW they want. After customized, user also can check the SSW in this page before the SSW submitted.

Figure 5 shows the SSW monitoring and interaction interface. From this interface, user can know the SSW submitted in cloud. The information about the SSW show in this interface. While the SSW running, user can know the procedure of SSW and the each model procedure in this SSW. The colour surround model icon show the procedure of the computing model. If the SSW terminated abnormally, this interface will give the error message and show the abnormally termination model.

### Conclusion

This paper describe the SSW concept. With this concept we build the application system. User can use this system to customize SSW to process marine data resource automatic. To solve the match problem between nodes in the uploading interface and check the structure of SSW before the SSW submitted can make the system more effective. By using the RESTful Web Service technology, user can interactive with SSW submitted crossing different platform. With the visual interface, user can get a lot of information about SSW status which can be used to optimize the system.

## **Acknowledgment**

This work is supported by The Ocean Public Welfare Project of the Ministry of Science and Technology (No.201105033) and The National Natural Science Foundation of China. (No.40976108).

## **References**

- [1] J. Zhang. Cloud computing model base on Hadoop and Meteorological Application. Nanjing University of Information Science & Technology. 2009.
- [2] D. S LEE, W. T Sung, S. M Wang. Multi-Cloud Service Fulfilment Architecture Enabling Digital Rain Forest. 2013 IEICE.
- [3] Ludascher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M. Scientific workflow management and the Kelpler system. *Concurrency Computation Practice and Experience*. 2006.
- [4] Majithia S, Shields M, Taylor I. Triana: A Graphical Web Service Composition and Execution Toolkit [C]. *IEEE International Conference on Web Services*.2004: p.514.
- [5] Shadig W, Orłowska M. Analysis process models using graph reduction techniques. *Information Systems*. 2000, 25(2): p.117-134.
- [6] Webber J, Parastatidis S, Robinson I. *REST in practice*. Sebastopol, US: O'Reilly Media, 2011.