Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

# Exploring cloud-based Web Processing Service: A case study on the implementation of CMAQ as a Service

Chen Zhang, Liping Di*, Ziheng Sun, Li Lin, Eugene G. Yu, Juozas Gaigalas

*Center for Spatial Information Science and Systems, George Mason University, 4400 University Dr, MSN 6E1, Fairfax, VA, 22030, USA*

ARTICLE INFO

*Keywords:*
CMAQ
WPS
Cloud computing
Earth science model
Web service
OGC

ABSTRACT

As an important tool for air quality simulation, the Community Multiscale Air Quality (CMAQ) model is widely used in the environmental modeling community. However, setting up and running the CMAQ model could be challenging for many scientists, especially when they have limited computing resources and little experience in handling large-scale input data. In this study, we explore the cloud-based Web Processing Service (WPS) and present the Cloud WPS framework to support implementing Earth science model as WPS. Specifically, to make CMAQ easier to use for scientists through the latest standard-based Web service technology, CMAQ-WS, a prototype of CMAQ as a Service, is developed and tested. The result of the experiment shows the framework significantly improves not only the performance of but also ease of use of the CMAQ model thus providing great benefits to the environmental modeling community. Meanwhile, the proposed framework provides a general solution to integrate Earth science model, WPS, and cloud infrastructure, which can greatly reduce the workload of Earth scientists.

## 1. Introduction

The volume of Earth Observation (EO) data is growing in the exponential level during the past decade. According to the statistic of NASA's Earth Observing System Data and Information System (EOSDIS), the average daily archive growth of EO data is 15.3 TB/day between Oct 1, 2016 to Sept 30, 2017, and the total archive volume is 23.8 PB, which has been increased over 3 times with 7.4 PB as in 2012 (EOSDIS, 2018). As the result, the computing and storing capabilities of traditional desktop GIS tools are becoming limited and insufficient. Scientists may face a lot of big data issues while processing large volume of EO data, such as how to unify the formats of different remote sensing data, how to obtain useful information in the huge volume of data, how to improve the efficiency of remote sensing data management/analysis, and how to facilitate data security, accessibility, connectivity, and quality (Di, 2016; Di et al., 2016). Fortunately, with the rapid development of information technology, new approaches such as big data management/analysis, cloud computing/service, Web services, the Internet of Things (IoT), machine learning, semantic web, workflow, sensor web, scalable database, data-intensive computing, and advanced analytics are being enabled in the Geographic Information System (GIS) and remote sensing, which drive GIS from the traditional Geographic Information Systems to Geographic Information Services (GIServices) (Yue et al., 2015a).

The dissemination of massive Web-based GIServices significantly lowered the entry barrier for accessing EO data and greatly simplified the workflow for processing EO data (Swain et al., 2015; Tan et al., 2016; Vitolo et al., 2015; Yue et al., 2015b). GIServices have very extensive applications in many societal sectors ranging from government, academia, to industry. For example, GEOSS (Global Earth Observation System of Systems) coordinates a set of independent Earth observation, information and processing resources with contributions from countries around the world and international organization members of the Group on Earth Observations (GEO) to facilitate access, monitoring, sharing of global environmental data and information via GIServices. It makes use of Web-as-a-Platform to implement a System-of-System architecture (Nativi et al., 2015). GeoBrain and its GeOnAS (GeoBrain Online Analysis System) integrate Web service based analysis and OGC standards-based data access to discover, retrieve, analyze, and visualize geospatial data (Di, 2004a). The SEPS (Self-adaptive Earth Predictive System) framework sets the general sensor web approach for coupling Earth system models with EO data via GIServices (Chen et al., 2010; Di, 2007; Yu et al., 2010; Zhang et al., 2012).

Along with the development of GIServices, the concept of Model Web, Model as a Service, and Service-Oriented Architecture (SOA) are proposed to facilitate the interoperability of models across

* Corresponding author.
*E-mail addresses:* czhang11@gmu.edu (C. Zhang), ldi@gmu.edu (L. Di), zsun@gmu.edu (Z. Sun), llin2@gmu.edu (L. Lin), gyu@gmu.edu (E.G. Yu), jgaigala@gmu.edu (J. Gaigalas).

https://doi.org/10.1016/j.envsoft.2018.11.019
Received 16 February 2018; Received in revised form 5 July 2018; Accepted 28 November 2018
Available online 13 December 2018

environmental modeling community (Castronova et al., 2013; Goodall et al., 2011; Granell et al., 2010; Nativi et al., 2013). On the other hand, as a major technology for delivering computing to address big data challenges, cloud computing provides a cost-effective way for Earth scientists to store, access, and model with the large volume of EO data (Buyya et al., 2009; Nativi et al., 2015). Rather than using the traditional desktop GIS software on the local PC, modeling EO data with the cloud-based GIServices system can relieve users, especially non-technical ones, from the time-consuming EO big data processing and requirement on local high-end computing facility due to the high-performance computing capability offered by the powerful server-end infrastructure. A series of studies on utilizing cloud-based cyberinfrastructure to facilitate GIServices have been done (Horsburgh et al., 2016; Jin et al., 2017; Morsy et al., 2018; Sun et al., 2017b; Wang et al., 2013).

It is known that the traditional way of setting up the Earth science model for researchers and scientists who do not have the strong technical background can be difficult. Since most models are open source project that many scientists and institutes have contributed to, the process of installing and configuring the model is not as easy as commercial software. Even after successful installation, users may still face many barriers, including the difficulty of collecting input data from different sources, incompatibility of data formats, unavailability of the data products, and limitation of computing resources. From the perspective of modelers, a series of questions are put forward: Can the installation and configuration processes of the Earth science model be simplified? Can the Earth science model be published as a service? Can the service be accessed and interoperated through the standardized interface? Can the performance of the model be improved by moving the computing from local to the cloud? To address the above questions, in this study, we present a cloud-based Web Processing Service (WPS) framework. Specifically, the Community Multiscale Air Quality (CMAQ) model, an essential tool that is being widely used for air quality and pollution simulation in the environmental modeling community, is implemented as CMAQ-WS, a prototype of CMAQ as a Service, based on the proposed framework.

The following is the structure of this paper. Section 2 introduces the design of the Cloud WPS framework. Section 3 demonstrates system architecture, workflow, and benchmark of CMAQ-WS. Section 4 summarizes the answers to the questions that posed above, addresses limitation of the current implementation, and discusses the potential improvement methods and future works. The conclusion is given in Section 5.

## 2. Design of Cloud WPS framework

### 2.1. OpenGIS Web Processing Service

The OpenGIS® Web Processing Service is an OGC (Open Geospatial Consortium) Web service interface standard for geospatial processing services. Generally, a Web service could be described as the self-contained, self-describing, modular applications that can be published, located, and dynamically invoked across the Web (Di, 2004b). Further, a geospatial Web service is a modular Web application that provides services on geospatial data, information, and knowledge (Di et al., 2005). OGC Web Services (OWS) refers to those services that reflect OGC vision for geospatial data and application interoperability and is one of the most widely used geospatial Web Services. With the proliferation of GIServices, a lot of OGC standards and specifications have been developed. Among them, WPS serves as the standard for the application service that provides the capability to wrap any geospatial processing capability, regardless of the source, as a Web service with a standard interface so that it can be integrated into existing Web service workflows. Besides, OGC also has developed a series of Web service specifications that have been widely disseminated in geospatial domains such as Catalogue Services for the Web (CSW), Sensor Observation Service (SOS), Web Coverage Service (WCS), Web Feature Service (WFS), and Web Map Service (WMS).

OGC WPS standard defines (1) rules for standardizing how inputs and outputs (requests and responses) for geospatial processing services, such as polygon overlay; (2) how a client can request the execution of a process; (3) how the output from the process is handled; and (4) an interface that facilitates the publishing of geospatial processes and clients' discovery of and binding to those processes (OGC, 2018a). Due to the popularity of WPS in the geospatial field, several open source GIServices packages have implemented WPS, such as 52° North WPS, PyWPS, and GeoServer WPS. Many WPS-based products have been developed to facilitate the geoprocessing over the Web (Dubois et al., 2013; Feng et al., 2011; Rosatti et al., 2018; Yue et al., 2010).

Currently, OpenGIS Web Processing Service 1.0.0 (OGC, 2007) is the most widely used WPS interface standard. Core operations defined in WPS 1.0 interface standard include GetCapabilities, DescribeProcess, and Execute, which are similar with other OGC Web Services such as WCS, WFS, and WMS. As the continuation of WPS 1.0 standard interface, WPS 2.0.2 (OGC, 2018b), the latest version of OGC WPS interface standard, offers two new core operations: GetStatus and GetResult, supporting both immediate processing for quick computational tasks and asynchronous processing for more complex and time-consuming tasks. However, due to the retro-compatibility issue, WPS 2.0 does not interoperate very well with WPS 1.0. Table 1 summarizes the description of core operations offered by the WPS standard interface.

### 2.2. Cloud computing technology

Cloud computing technology plays an important role in GIServices. By shifting the computing resources to the cloud, the computing capability available for users could be far more powerful than what a single local device (such as a workstation, desktop, laptop, smartphone or tablet) can ever offer. According to the description by National Institute of Standards and Technology (NIST), the cloud could be described with: (1) five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, measured Service; (2) three service models: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS); and (3) four deployment models: private cloud, community cloud, public cloud, and hybrid cloud. The architecture of spatial cloud computing covering IaaS, PaaS, SaaS, and DaaS has been discussed in (Yang et al., 2011). The feasibility of integrating geospatial Web services and cloud infrastructure for EO data processing has been explored in many studies (Chen et al., 2012; Shao et al., 2012; Zhang et al., 2017).

Compared with the conventional client-server architecture, the cloud-based architecture provides a more flexible environment for

**Table 1**
Core operations of Web Processing Service.

| Operation | Description | Supported WPS Interface |
|---|---|---|
| GetCapabilities | request information about the server's capabilities and processes offered | WPS 1.0/2.0 |
| DescribeProcess | request detailed metadata on selected processes offered by a server | WPS 1.0/2.0 |
| Execute | execute a process comprised of a process identifier, the desired data inputs and the desired output formats | WPS 1.0/2.0 |
| GetStatus | query status information of a processing job | WPS 2.0 |
| GetResult | query the results of a processing job | WPS 2.0 |

Earth scientists to publish geospatial services and applications. As two most common cloud deployment models, public cloud and private cloud have been widely used to support GIServices. Public cloud is the cloud service that is open to any user. According to the difference in computing resources and services, consumers would be charged differently from the vendors. Public cloud providers, such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud, IBM Cloud, and Oracle Cloud, offer similar services on computing, storage, databases, developer/management tool, and application services. Comparing with the public cloud, a private cloud provides more flexible operations and capabilities but only available to a specific organization or a group of permitted users. Characterized by its privacy, a private cloud is managed by an organization or person the user belongs to or knows, rather than some provider whom the user does not know. Thus it is accessed only by a specific group of users and is not open to the public in most cases. A private cloud can be quickly set up with the open-source IaaS cloud management frameworks such as Eucalyptus, OpenStack, CloudStack, and OpenNebula. Both public and private clouds have been used to facilitate environmental modeling (Ercan et al., 2014; Essawy et al., 2018; Kurtz et al., 2017).

### 2.3. High-level architecture of the Cloud WPS framework

This study has developed the Cloud WPS framework to facilitate publishing the Earth science model as the open standard-based Web service over the cloud. Fig. 1 shows the high-level architecture of the Cloud WPS framework. The architecture consists of four building blocks: hardware, platform, service, and client, and each block has a number of subcomponents. This framework is designed based on the existing cloud computing service model (Ghazouani and Slimani, 2017; Yang et al., 2011). The hardware block is the foundational block of the cloud-based WPS framework which consists of the physical cloud infrastructure. Generally, the physical cloud infrastructure is built with the server cluster and provides the essential raw computing resources for the cloud. The computing capability supported by the cloud depends on the computing resources (e.g. CPU, memory, disk, and bandwidth) the server cluster offers. The Cloud WPS framework supports both public cloud and the private cloud, which means it could be implemented inside in both public clouds (e.g. AWS, Google Cloud, Azure), or private clouds powered by the open source IaaS cloud management frameworks (e.g. Eucalyptus, OpenStack, CloudStack, and OpenNebula). The platform block handles what PaaS cloud service model does and basically follows the architecture of PaaS, which consists of networking, storage, servers, virtualization, operating system, middleware, and runtime. It is responsible for: (1) provision of the managed application development and deployment platform; (2) virtualization and virtual machine (VM) support above the physical servers; (3) optimization of networking capabilities and management of IPs; and (4) automatic scale and load-balance of computing instances.

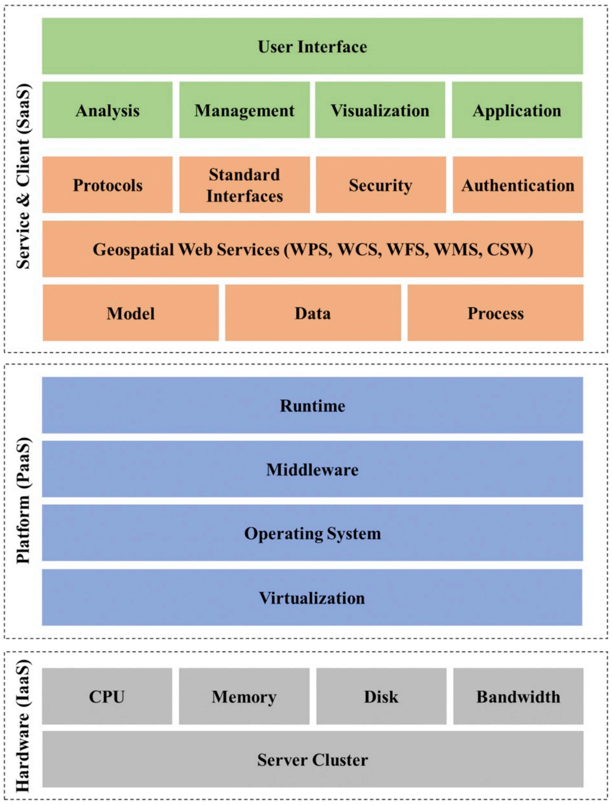


**Fig. 1.** High-level architecture of the Cloud WPS framework.

To integrate WPS with the existing cloud service model, the service & client block are added on the top of the framework as the components of SaaS layer. It is the core part of the framework, which provides protocols, standard interfaces, security/authentication to wrap all Earth science models, EO data, and geospatial processes into geospatial Web services. Models, data, and processes are hosted inside the VMs that managed by the cloud platform. In each VM, the service is published through the standardized geospatial Web service such as WPS, WCS, WMS, and WFS. Therefore, in this framework, all services are independently deployed but interoperable with each other. When users send a request to execute a model, the model-ready VM would be called and the entire computation would be executed inside the cloud infrastructure. Moreover, the service & client block provides the user interface with client interaction modules such as visualization, analysis, management, and other applications. The Web-based client can be accessed by any Web browser on a desktop, laptop, smartphone, and tablet. As a gateway of the framework, the client allows users to send a request and receive the result from the server without worry about the details of the development and deployment of the model, processes, and applications.

### 2.4. Interoperability of the Cloud WPS framework

Fig. 2 illustrates the relationship among the components of the Cloud WPS enabled system and the interoperability with other Earth science information systems. The users of the Cloud WPS framework enabled system could be divided into four groups: Earth scientists, developers, administrators, and other users. Each group of users plays its specific roles and holds specific permission and privilege. Earth scientists are the top-level users who have the permission to directly manipulate model, processes, and data through geospatial Web services. Developers have the permission of accessing cloud instances and are responsible for developing and publishing Earth science models as geospatial Web services. Administrators have the root permission of accessing the cloud management module. Other users share the computing resources with Earth scientists but only have the permission of accessing their own instances or Web services.

As shown in Fig. 2, in a Cloud WPS framework enabled system, all models, processes, data, and Web services are deployed inside the VMs. All VMs are isolated but can be interoperable with each other. Like all private cloud platforms, the Cloud WPS framework enabled system is powered by the cloud management module, which plays the role of deploying and managing large networks of both physical server zones and virtual servers. Besides the service and data deployed in the local cloud infrastructure, the framework is interoperable with other data/service providers such as NASA and NOAA. Moreover, it can be built as a part of the hybrid cloud by integrating with public clouds and private clouds. All VMs in the Cloud WPS framework enabled can be disseminated through general VM formats (e.g., qcow2) compatible with most of the cloud platforms. The model pre-installed VM could be encapsulated as the model template. With the model template, users could

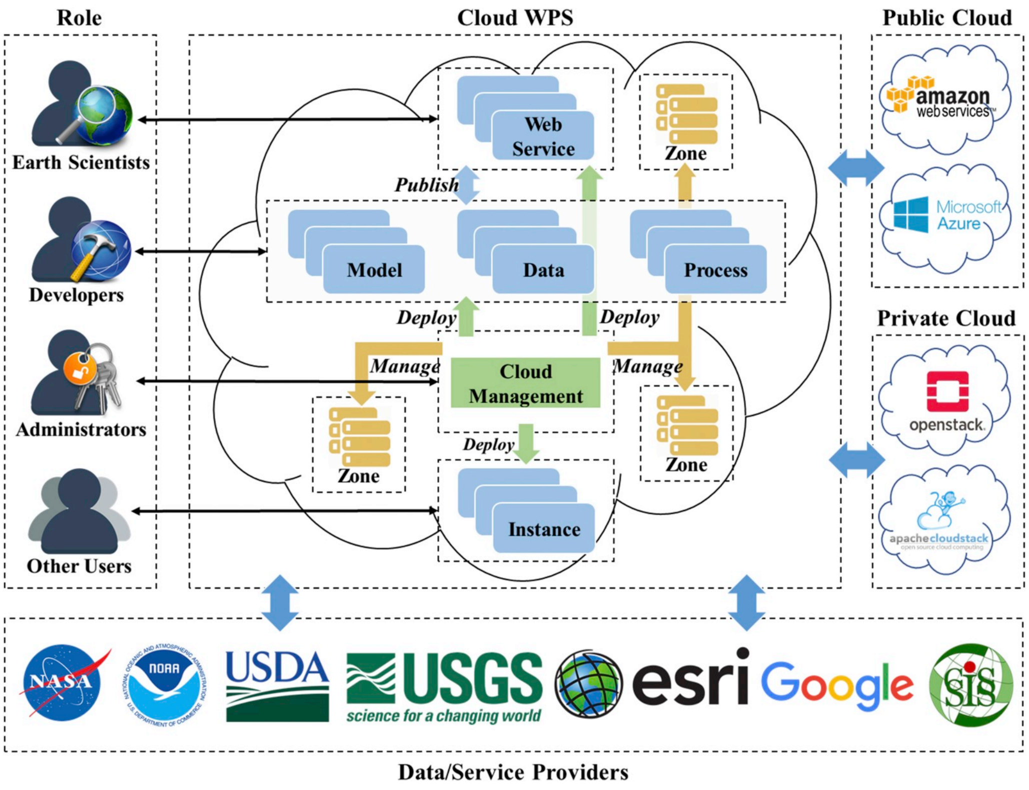


**Fig. 2.** Interoperability among the Cloud WPS framework enabled system, public/private cloud, and data/service providers.

deploy and reproduce one or more model-ready VMs inside any cloud.

## 3. Prototype of CMAQ as a Service

### 3.1. CMAQ model

Global air pollution and air quality issues have made a huge impact on climate, environment, and especially human health (Akimoto, 2003). U.S. Environmental Protection Agency (USEPA) set six common air pollutants as the "criteria air pollutants": ground-level ozone, particulate matter, carbon monoxide, lead, sulfur dioxide, and nitrogen dioxide (USEPA, 2017). Each pollutant has the specific physical and chemical property which could affect human health and natural environment at different levels and in different ways (Kampa and Castanas, 2008). As a well-vetted peer-reviewed scientific model for modeling and simulating air quality, Community Multiscale Air Quality (CMAQ) model provides the capability of estimating the distribution of ozone, particulates, toxics, and acid deposition in the atmosphere (CMAS, 2018). Before implementing CMAQ as Web Processing Service, it is essential to understand how CMAQ model works.

The core CMAQ programs and its workflow is described in Fig. 3. Major processors and the chemical-transport models of CMAQ modeling system includes Sparse Matrix Operator Kernel Emissions (SMOKE) model, Meteorology Model (MM5 or WRF), Meteorology-chemistry interface processor (MCIP), Photolysis rate processor (JPROC), Initial conditions processor (ICON), Boundary conditions processor (BCON), and CMAQ chemical-transport model (CCTM). Among them, Emission Model and Meteorology Model are third-party programs, which are not part of CMAQ program but supply the necessary input data for air quality simulation. MCIP, ICON, BCON, and JPROC are standard CMAQ preprocessors providing the mandatory inputs for CCTM. The output generated by CCTM then feeds back to ICON and BCON for nested simulations. The function of each preprocessor is described as follow.

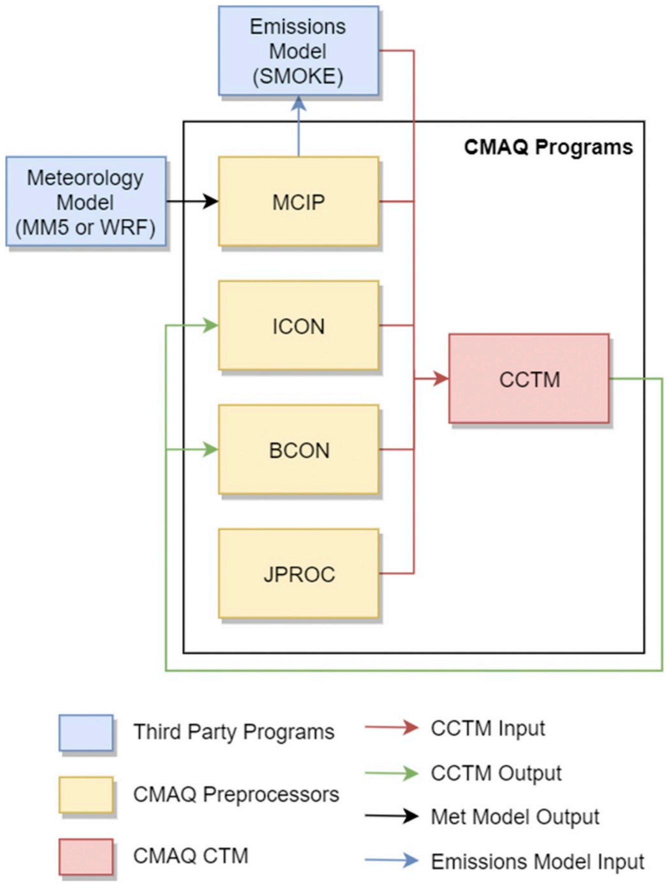


**Fig. 3.** CMAQ core programs (adopted from (CMAS, 2012)).

**ICON**: Initial Conditions Processor (ICON) preprocessor outputs a gridded binary NetCDF file of the chemical conditions of the chemical

**Table 2**
CCTM output files.

| Output file | File type | Description |
|---|---|---|
| CONC | CCTM hourly instantaneous concentration file | The 3-D CCTM hourly concentration file (CONC) contains gas-phase species mixing ratios (ppmV) and aerosol species concentrations ($\mu g\ m^{-3}$); CONC files include instantaneous model species concentrations at the end of each model hour. |
| CGRID | CCTM restart file | The 3-D CCTM ending concentration file (CGRID) contains gas-phase species mixing ratios (ppmV) and aerosol species concentrations ($\mu g\ m^{-3}$); the CGRID file includes model species concentrations at the end of each simulation period. |
| ACONC | CCTM hourly average concentration file | The 3-D CCTM integral average concentration file (ACONC) contains average model species concentrations for each model hour, as opposed to instantaneous concentrations at the end of each output time step. |
| DRYDEP | CCTM hourly cumulative dry deposition file | The 2-D CCTM dry deposition file (DRYDEP) includes cumulative hourly dry deposition fluxes ($kg\ hectare^{-1}$) for selected model species. |
| WETDEP | CCTM hourly cumulative wet deposition file | The 2-D CCTM wet deposition file (WETDEP) includes cumulative hourly wet deposition fluxes ($kg\ hectare^{-1}$) for selected model species |
| AEROVIS | CCTM hourly instantaneous visibility metrics | The 2-D CCTM visibility file (AEROVIS) contains hourly Mie and reconstructed visual range coefficients ($km^{-1}$) and normalized extinction coefficients (deciviews). |

condition for the first hour of simulation in the modeling domain.

**BCON**: Boundary Conditions Processor (BCON) preprocessor outputs a gridded binary NetCDF file of the chemical conditions along the horizontal boundaries of the modeling domain.

**MCIP**: Meteorology-Chemistry Interface Processor (MCIP) preprocessor outputs the NetCDF formatted meteorology data required by SMOKE and CCTM.

**JPROC**: Photolysis Rate Processor (JPROC) preprocessor outputs the chemical-mechanism-specific clear-sky photolysis rates at fixed altitudes, solar hour angles, and latitude bands from tabulated absorption cross-section and quantum yield (CSQY) data.

**CCTM**: CMAQ Chemistry-Transport Model (CCTM) is the core processor of CMAQ, which simulates continuous atmospheric chemical conditions by integrating the outputs from the above preprocessors and SMOKE. CCTM outputs binary NetCDF files of gridded and temporally resolved air pollutant information including 3-D CCTM hourly concentration file (CONC), 3-D CCTM ending concentration file (CGRID), 3-D CCTM integral average concentration file (ACONC), 2-D CCTM dry deposition file (DRYDEP), 2-D CCTM wet deposition file (WETDEP), 2-D CCTM visibility file (AEROVIS). Table 2 describes the details of the CCTM output files.

## 3.2. System architecture

CMAQ-WS, a prototype of CMAQ as a Service, is developed based on the Cloud WPS framework. A suite of advanced techniques, software, and services have been applied to the implementation of this prototype. As shown in Fig. 4. The prototype consists of four major components: GeoBrain Cloud, CMAQ-WS, CyberConnector, and the Internet. This section describes the major components of the CMAQ-WS prototype.

### 3.2.1. GeoBrain Cloud

All VM instances of the CMAQ-WS are powered by the GeoBrain Cloud (http://cloud.csiss.gmu.edu), a private cloud platform running by Center for Spatial Information Science and Systems, George Mason University. GeoBrain Cloud mainly serves for Geospatial Web services and provides computing source for EO data processing. Besides, a large volume of EO data, such as Landsat data and MODIS data, are archived inside the cloud. It is built with IaaS architecture, using Apache CloudStack as the management framework to manage the network, storage, computing resources, and VMs. By taking advantage of the powerful open framework, the cloud management module offers an elastic and flexible mechanism to assign the load to particular services and dynamically assign the computing resources depending on the input data volume. Users and developers may access GeoBrain Cloud in different ways, including Web user interface, command line tools, and a full-featured RESTful API. By integrating the EC2 compatibility interface for Apache CloudStack, GeoBrain Cloud offers the API that is compatible with API of AWS EC2 and S3.

The deployment of GeoBrain Cloud is based on the small-scale deployment architecture. As shown is Fig. 5, the cloud platform physically contains one management server, one layer-2 switch, Network File System (NFS) server, and computing node. GeoBrain Cloud currently manages over 300 CPU cores, 500 GB RAM, and 600 TB storage in total. Plus, a cluster of NVIDIA Tesla K80 GPUs are equipped in the GeoBrain Cloud's computing node which is mainly used for offering computing capability for advanced machine learning tasks.

### 3.2.2. CyberConnector

CyberConnector is an EarthCube building block supporting the automatic preparation and feeding of on-demand EO data into Earth science models. It adopts open geospatial standards/specifications to process the EO and geoprocessing workflow. In the implementation of CMAQ-WS, CyberConnector (1) automates the handshaking between VMs that distributed in application layer, interface layer, service layer; (2) plays as the front-end client between users and the service; and (3) connects the CMAQ-WS with the Internet and other clouds. Additionally, CyberConnector offers the capability to automatic tailor multi-source EO data to model-ready input files and feed Earth Science Models, which could save a tremendous amount of time and effort for Earth science modelers (Sun et al., 2017a).

The procedure of using CyberConnector to execute CMAQ-WS is simple. It utilizes Virtual Data Products (VDP) to build geoprocessing workflow. CyberConnector provides a Web-based user interface allowing users to search and order VDP. Fig. 6 shows the screenshot of using CyberConnector to search and order VDP. In this case, CMAQ-WS is implemented as a VDP which can be searched and ordered through the CyberConnector Web client. After the order of CMAQ-WS is placed, CMAQ required input data would be automatically tailored from different sources through CyberConnector and delivered to the CMAQ VMs. Once the process is done, the result will be sent to users via emails.

### 3.2.3. CMAQ-WS

CMAQ-WS is the core component of this prototype. It is composed of a bunch of VM instances. According to the difference in the functionality, the instances deployed inside the CMAQ-WS can be divided into three layers: application layer, interface layer, and service layer. The followings are a description of the implementation and functionality in each layer.

The application layer is the gateway of CMAQ-WS where the Web-based clients are deployed. As the front-end client to access CMAQ-WS, the Web-based client of CyberConnector is deployed inside this layer. Besides, we implemented a lightweight Web application to test the prototype. It is deployed inside the application layer as well. The user interface of the CMAQ-WS test client is shown in Fig. 7. Two required inputs in the CMAQ processes are CMAQ input configuration file and cloud configuration file. CMAQ input configuration file contains the

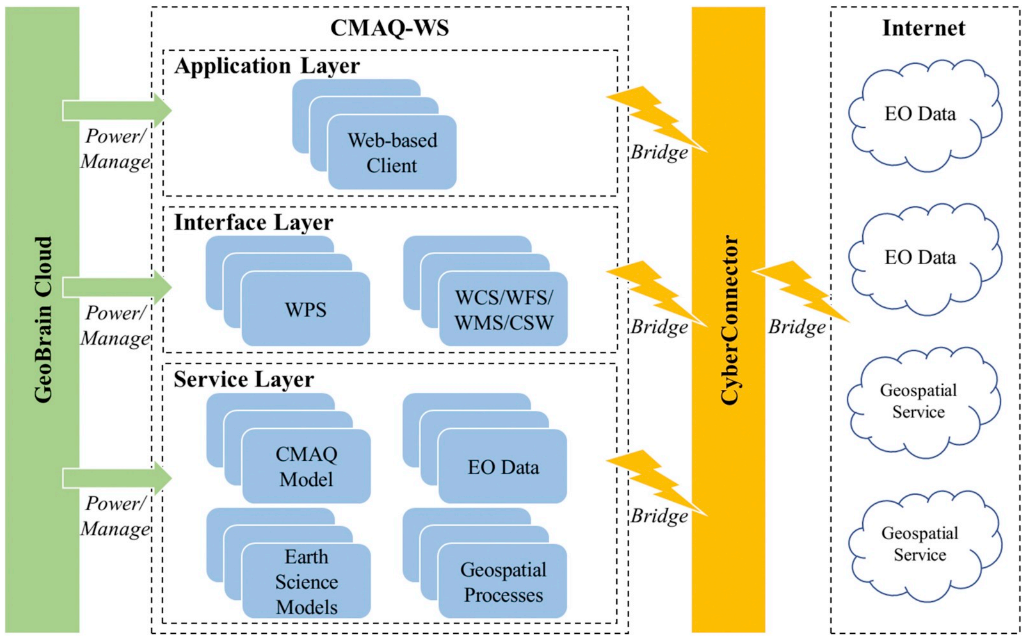


**Fig. 4.** Components of the CMAQ-WS prototype. The prototype is composed by a group of VM instances which are managed/powered by the GeoBrain Cloud and bridged by the CyberConnector.

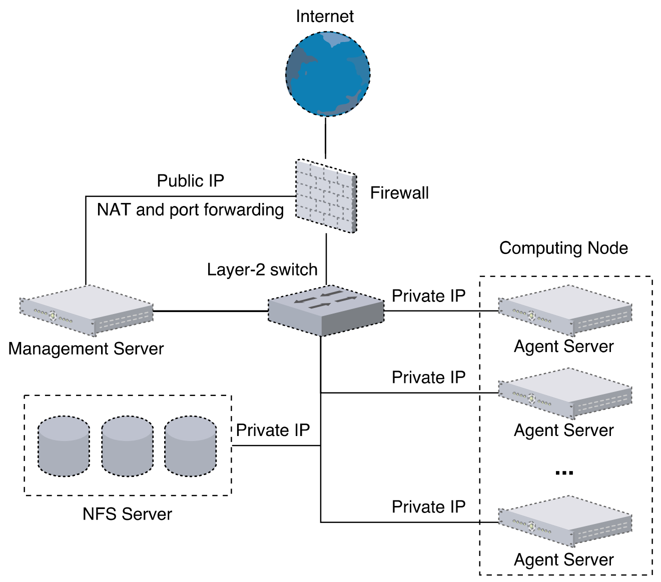


**Fig. 5.** Deployment architecture of GeoBrain Cloud. The management server is the gateway and router of the cloud platform which manages all hosts, storage, and networking. Layer-2 switch deals with switching and redirecting data packets. Computing node contains a cluster of physical server that offering computing resources (e.g. CPU, GPU, RAM). NFS server provides the storage pool for the cloud.

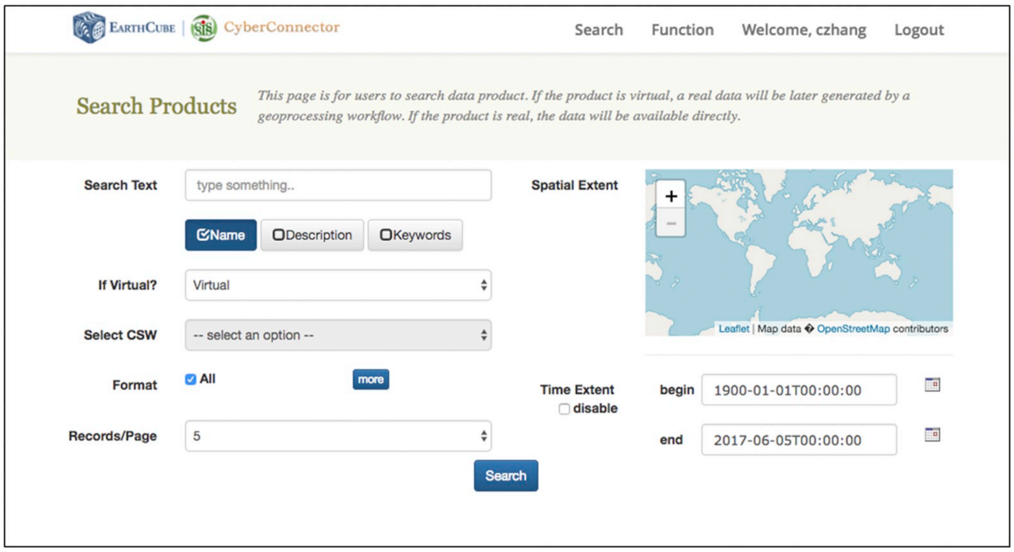
information of all input data that required by the CMAQ model. Cloud configuration file contains the preference of the cloud instance, such as CPU cores and memory size, of the task.

The interface layer provides the standard interface to bridge the application layer and the service layer. Standard geospatial services, including WPS, WMS, WFS, and WCS, are implemented in this layer. The prototype adopts WPS as the standard interface to access CMAQ model. When a user calls the CMAQ service, a WPS request is sent from the client VM that deployed inside the application layer. After the process is done, the CMAQ output data would be returned through WPS standard interface when the process is finished. In this study, CMAQ-WS was implemented and tested with both WPS 1.0 and WPS 2.0 standard interface. We used the PyWPS as the framework of WPS 1.0 standard interface and manually implemented core components of WPS 2.0 standard interface to support asynchronous WPS processes. All basic WPS operations such as GetCapabilities, DescribeProcess, Execute are supported with WPS 1.0/2.0, GetStatus and GetResult are supported with WPS 2.0.

The service layer hosts a number of model instances, geoprocessing instances, and EO data instances. CMAQ model is implemented as the model instance where CMAQ and its dependent libraries/software packages are installed. CMAQ program, an open-source CMAQ implementation developed and released by USEPA Computational Exposure Division, is the main software package of the CMAQ model instance. This prototype used CMAQ program v5.1 as the implementation of the CMAQ model. In addition, required libraries and dependencies such as NetCDF, I/O API, and MPICH need to be pre-installed and configured inside the instance before running CMAQ model. NetCDF (Network Common Data Form) is the software library that developed and maintained by Unidata (http://www.unidata.ucar.edu). To improve interoperability and re-usability, NetCDF became an OGC standards in 2011 and Climate and Forecast (CF) extension to NetCDF Core data model standard was approved in 2013 (Domenico and Nativi, 2013). As the essential component to run CMAQ model, NetCDF-C library and NetCDF-Fortran library need to be installed. Models-3/EDSS Input/Output Applications Programming Interface (I/O API) version 5.1 is another required library for CMAQ version 5.1. It provides model developers with the programming library for data storage and access and is available in C and Fortran. MPICH is the library enabling multiple processors for running CMAQ.

### 3.3. Workflow

The workflow of CMAQ-WS is demonstrated as the sequence diagram in Fig. 8. Before initiating a CMAQ processing task, users could get the list of available processes by sending GetCapabilities request and check the description of specific processes by sending DescribeProcess request with CMAQ-WS Web client, CyberConnector Web client, or any other WPS client. Then user executes the CMAQ process with required WPS inputs (URL of input data and URL of configuration


EarthCube | CyberConnector
Search
Function
Welcome, czhang
Logout
Leaflet | Map data © OpenStreetMap contributors
more
Search

Search Products

*This page is for users to search data product. If the product is virtual, a real data will be later generated by a geoprocessing workflow. If the product is real, the data will be available directly.*

Search Text: type something..

[✓ Name] Description Keywords

If Virtual?: Virtual

Select CSW: -- select an option --

Format: [✓ All]

Records/Page: 5

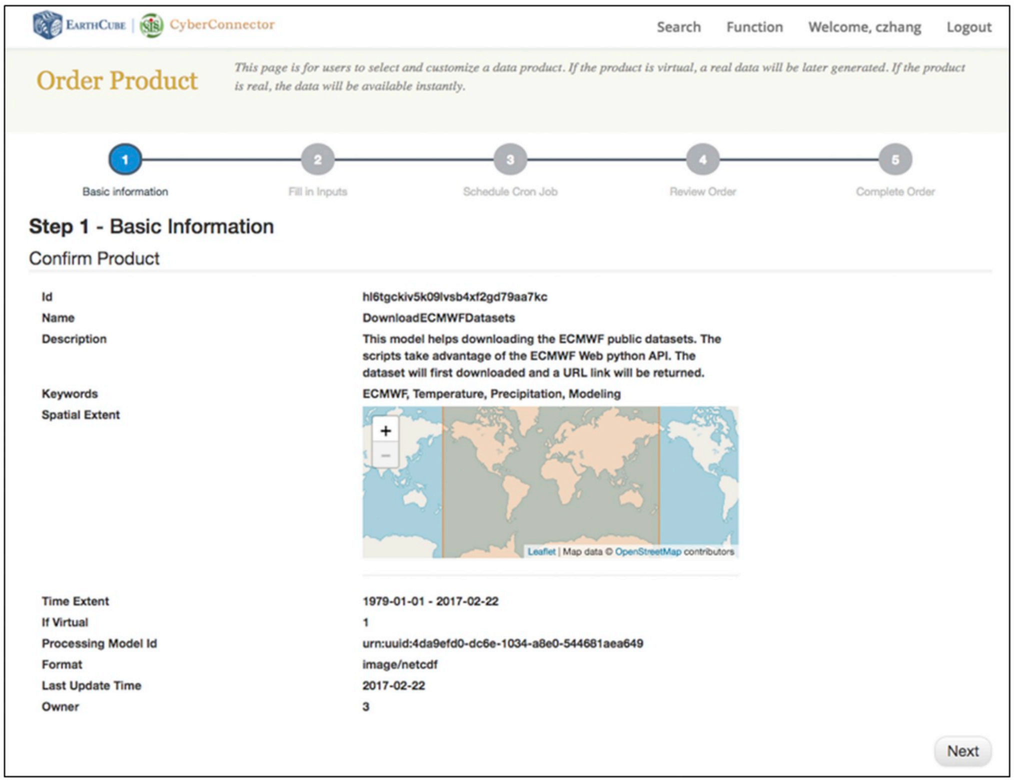
Spatial Extent

Time Extent ☐ disable

begin: 1900-01-01T00:00:00

end: 2017-06-05T00:00:00

(a) VDP searcher. The VDP can be searched and filtered with keyword, spatial information, and temporal information.



Order Product

*This page is for users to select and customize a data product. If the product is virtual, a real data will be later generated. If the product is real, the data will be available instantly.*

Step 1 - Basic Information

Confirm Product

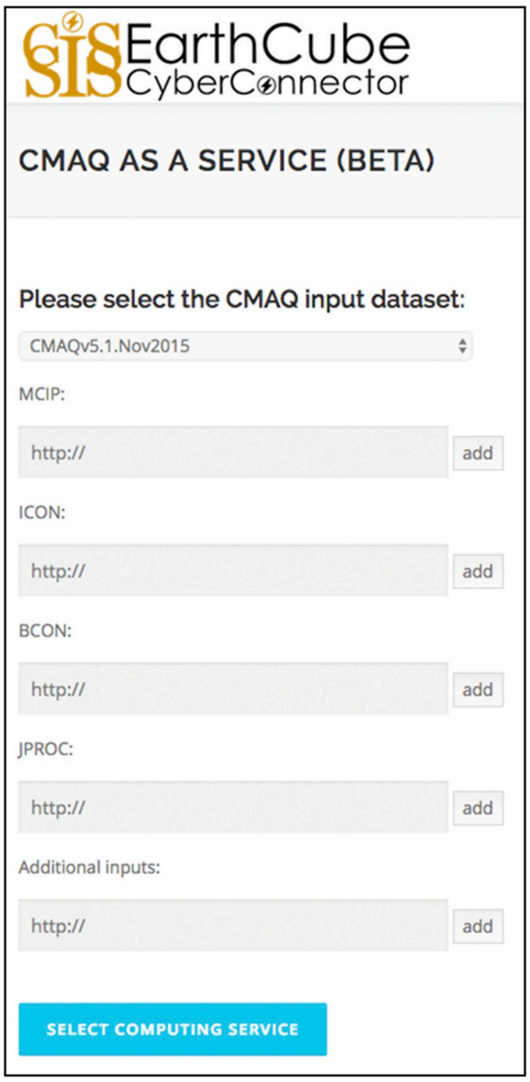
| | |
|---|---|
| Id | hl6tgckiv5k09lvsb4xf2gd79aa7kc |
| Name | DownloadECMWFDatasets |
| Description | This model helps downloading the ECMWF public datasets. The scripts take advantage of the ECMWF Web python API. The dataset will first downloaded and a URL link will be returned. |
| Keywords | ECMWF, Temperature, Precipitation, Modeling |
| Spatial Extent | |
| Time Extent | 1979-01-01 - 2017-02-22 |
| If Virtual | 1 |
| Processing Model Id | urn:uuid:4da9efd0-dc6e-1034-a8e0-544681aea649 |
| Format | image/netcdf |
| Last Update Time | 2017-02-22 |
| Owner | 3 |

(b) Step-by-step VDP ordering. The procedure of ordering VDP: Basic Information, Fill in Inputs, Schedule Cron Job, Review Order, Complete Order

**Fig. 6.** Graphical user interface of CyberConnector.

file) to initiate a CMAQ processing task.

After the request is sent, CMAQ model VMs are automatically instantiated and the task will be assigned to a particular instance based on the cloud configuration file. By taking advantage of the elasticity of the cloud, the CMAQ model VMs are dynamically allocated by WPS requests. Then all input data will be fetched and fed into the CMAQ instance. Three sources of the CMAQ input data are accepted in this prototype: local data servers/VMs where the required input data are stored, geoprocessing/model (e.g. WRF model) services where the required input data are tailored and generated, and other data providers from the Internet. After fetching required input data, the CMAQ model starts running inside the CMAQ model VMs. Once the process is done, the output data are copied from the CMAQ model VMs to the data server then the CMAQ model VMs are automatically destroyed. The URL of the output data is returned as the output of the process and CyberConnector will notify users via email. The running state of the



CMAQ AS A SERVICE (BETA)

Please select the CMAQ input dataset:

CMAQv5.1.Nov2015

MCIP: http:// add

ICON: http:// add

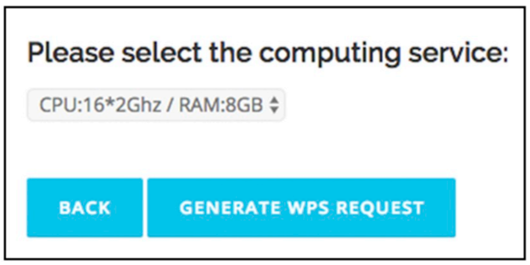
BCON: http:// add

JPROC: http:// add

Additional inputs: http:// add

SELECT COMPUTING SERVICE

(a) CMAQ input dataset (benchmark dataset is selected by default)

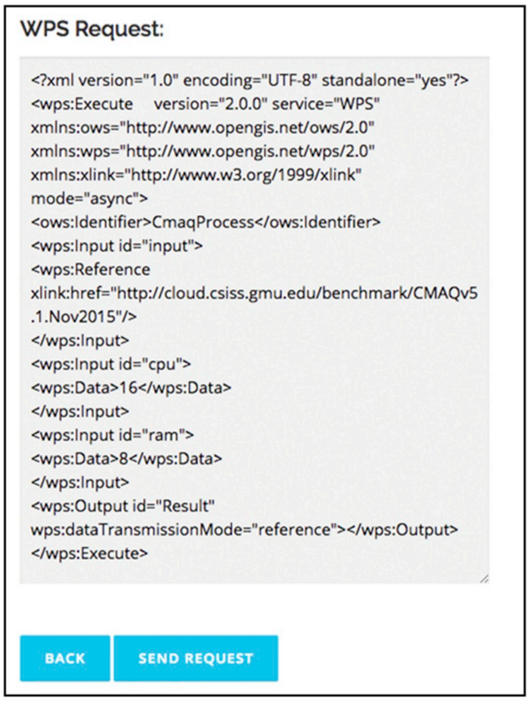


Please select the computing service:

CPU:16*2Ghz / RAM:8GB

BACK GENERATE WPS REQUEST

(b) Computing service offered by the cloud platform



WPS Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute    version="2.0.0" service="WPS"
xmlns:ows="http://www.opengis.net/ows/2.0"
xmlns:wps="http://www.opengis.net/wps/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
mode="async">
<ows:Identifier>CmaqProcess</ows:Identifier>
<wps:Input id="input">
<wps:Reference
xlink:href="http://cloud.csiss.gmu.edu/benchmark/CMAQv5
.1.Nov2015"/>
</wps:Input>
<wps:Input id="cpu">
<wps:Data>16</wps:Data>
</wps:Input>
<wps:Input id="ram">
<wps:Data>8</wps:Data>
</wps:Input>
<wps:Output id="Result"
wps:dataTransmissionMode="reference"></wps:Output>
</wps:Execute>
```

BACK SEND REQUEST

(c) WPS request generated based on user's inputs

**Fig. 7.** Graphical user interface of CMAQ-WS test client.

CMAQ processes could be checked through the GetStatus operation, either the status of "running" or "done" is returned depending on the actual processing status. And the final product could be retrieved through GetResult operation. The user can download the CMAQ output data from the data server with the output URL.

The prototype can be integrated as a part of the geoprocessing workflow. Fig. 9 illustrates the sequence diagram of building a geoprocessing/model workflow with the prototype. In this workflow, CMAQ-WS is not directly requested by users. Instead, as a part of the requested geoprocessing/model workflow service, CMAQ-WS is called when the required CMAQ input data is ready during the process of the service. The output of CMAQ-WS then would be used for the rest steps of the workflow. The final output of the workflow is copied/stored in the local data server. As the intermediate data of the workflow, all output data from the CMAQ-WS would be removed with the destruction of geoprocessing/model VM.

The prototype provides an easy-to-use dissemination solution. All components of the prototype, including client, WPS interface, CMAQ, geoprocessing/model service, data server, are implemented as VM instance. Each component can be disseminated through VM template in qcow2 format which is a general VM format compatible with major cloud platforms. With the VM templates image, users can upload the template image to the private or public cloud platforms such as OpenStack, CloudStack, and AWS.

### 3.4. Benchmark and validation

A series of experiments on the implementation using the CMAQv5.1 benchmark data as the input dataset has been conducted in this study. The CMAQv5.1 benchmark data is the open access dataset, which can be downloaded at https://www.cmascenter.org/download/software/cmaq/tracker/cmaq5-1.cfm. To generate the CCTM data, all input data need to be fed into the prototype and CMAQ processes need to be executed. Three basic processes, which CMAQ model executes in the prototype, are (1) MCIP to prepare the CMAQ-ready meteorological data from the WRF output; (2) ICON and BCON to create initial and boundary conditions input data; and (3) CCTM to estimate air quality fields.

Assuming the WRF meteorology and SMOKE emissions data are ready to use, WPS is expected to do is to generate ICON profile, BCON profile, and CCTM output data. In this experiment, it takes only a few seconds to generate ICON profile and BCON profile. However,

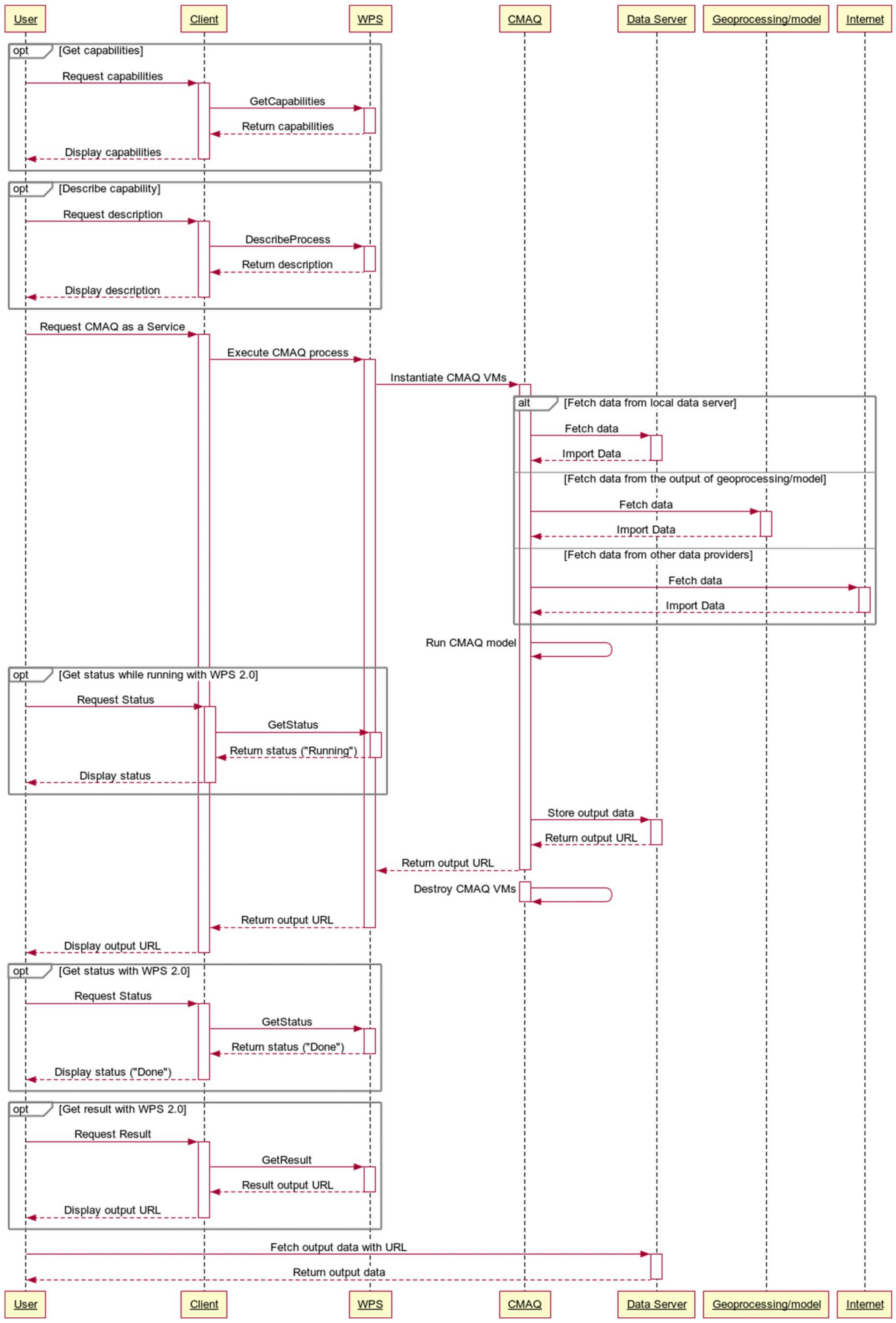


**Fig. 8.** Sequence diagram of the workflow of CMAQ-WS prototype.

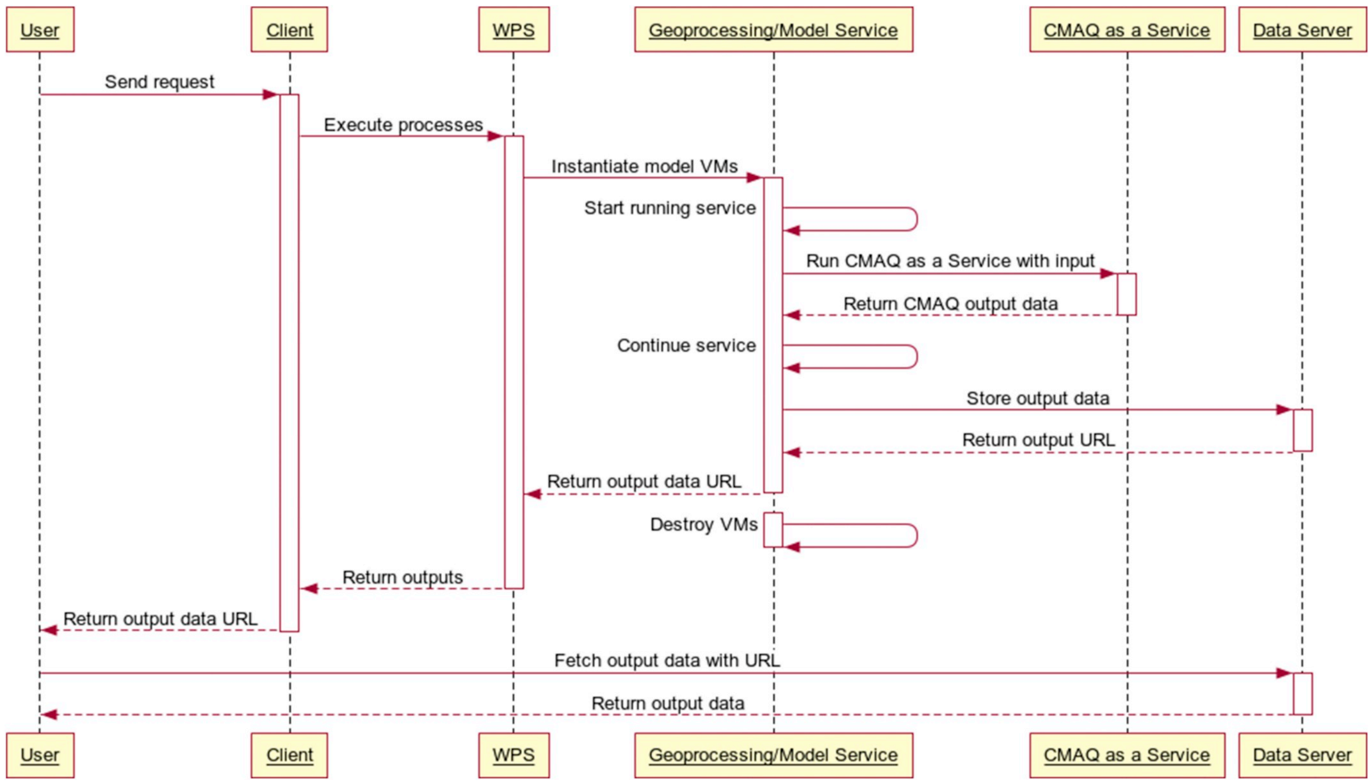


**Fig. 9.** Sequence diagram of building geoprocessing/model workflow with CMAQ-WS.

generating CCTM output data could take a couple of hours. The prototype aims to publish CMAQ model as the Web Processing Service. It could be a tough job for users to run CMAQ model through WPS 1.0 due to the longtime process of CCTM and unavailability of asynchronous processing capability in WPS 1.0. To handle this issue, the Cloud WPS framework adopts the WPS 2.0 standard interface, which provides the asynchronous processing capability.

A group of tests with different computing resource offerings have been conducted for testing the performance of CMAQ-WS. The instance types and the specification that GeoBrain Cloud offers are listed in Table 3. The CMAQ program supports serial mode and Message Passing Interface (MPI) mode. Serial mode only allows CMAQ model running on a single CPU core, no matter how much the instance offers. MPI, on the other hand, is a standard interface for parallel computing (Pacheco, 1997). Running the CMAQ model in MPI mode could make the most of the power of multi-core CPU and substantially reducing the computation time. This test compares the performance of CMAQ-WS in both serial mode and MPI mode. Fig. 10 displays the test result of running the CMAQ-WS with multiple instance type. In the serial mode, increasing the number of CPU cores does not observably improve the performance of CMAQ-WS. In the MPI mode, the performance is significantly improved with the increasing of the number of CPU cores. There is no distinct difference for the performance of the service while running on the Small instance with single CPU core.

**Table 3**
Instance type of GeoBrain Cloud.

| GeoBrain Cloud Instance Type | CPU (core) | Memory (GB) | Equivalent Amazon EC2 Instance Types |
|---|---|---|---|
| Small | 1 | 2 | t2.small |
| Medium | 2 | 4 | t2.medium |
| Xlarge | 4 | 16 | t2.xlarge |
| 2xlarge | 8 | 32 | m5.2xlarge |
| 3xlarge | 12 | 48 | N/A |
| 4xlarge | 16 | 64 | m5.4xlarge |

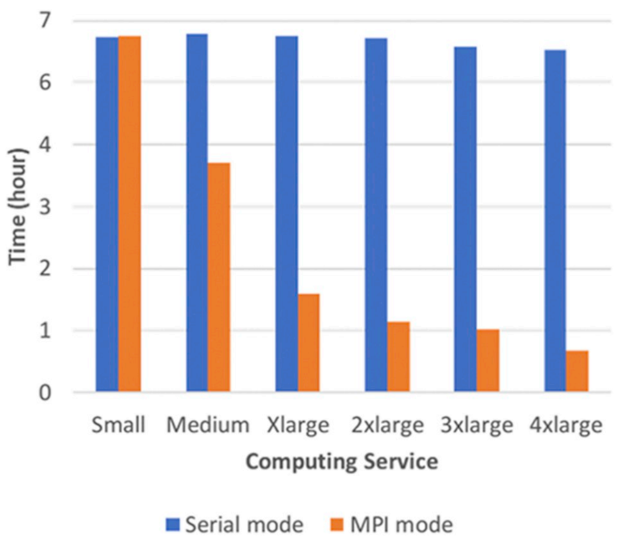


**Fig. 10.** Performance of running CMAQ-WS with multiple instance type.

To validate the output data, the CCTM outputs have been compared with CMAQ reference benchmark data which could be accessed at https://www.cmascenter.org/cmaq/. The result shows the content and size of CCTM data generated from the prototype are the same as the reference benchmark data. Fig. 11 displays the selected pollutants of CMAQ hourly average concentration (ACONC) data generated by the CMAQ-WS. Fig. 12 compares the output ACONC data of CMAQ-WS and CMAQ reference benchmark data.

## 4. Discussion

This section is organized to (1) summarize the answers to the

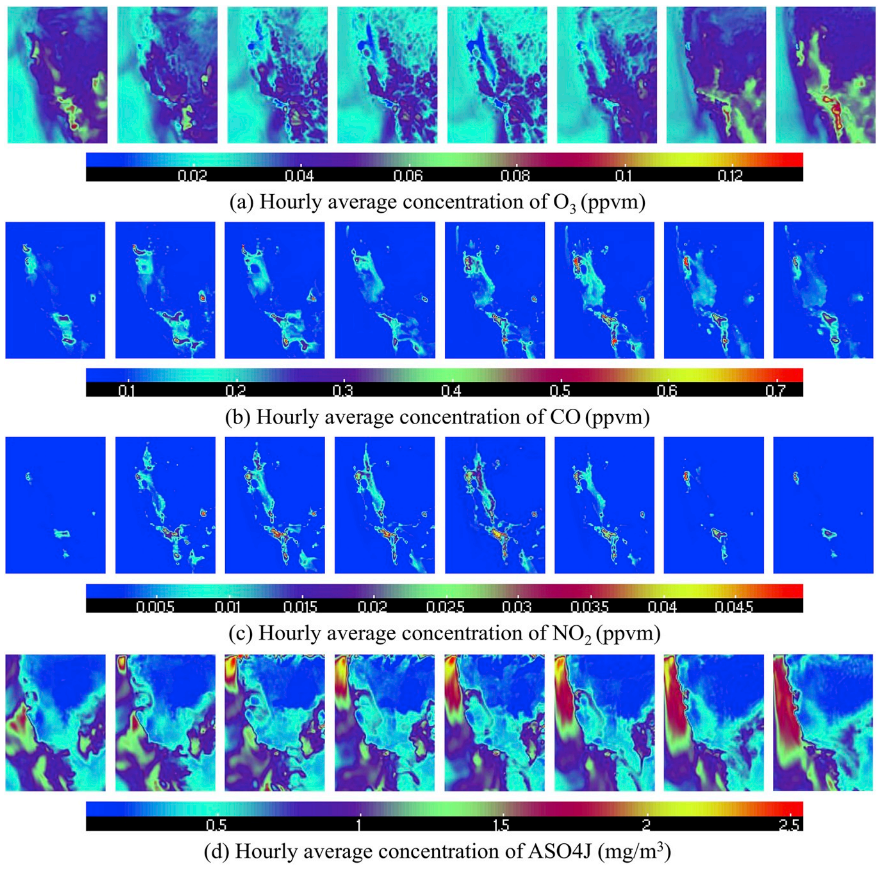

0.02 0.04 0.06 0.08 0.1 0.12
(a) Hourly average concentration of $O_3$ (ppvm)
0.1 0.2 0.3 0.4 0.5 0.6 0.7
(b) Hourly average concentration of CO (ppvm)
0.005 0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045
(c) Hourly average concentration of $NO_2$ (ppvm)
0.5 1 1.5 2 2.5
(d) Hourly average concentration of ASO4J (mg/m$^3$)

**Fig. 11.** Visualization of hourly average pollutant concentration output data.

questions that are posed in Section 1; (2) address some limitations on the current implementation; and (3) discuss the potential improvement methods and the future work.

The first question is how the Cloud WPS framework simplifies the installation and configuration process of the Earth science model. In the case of the implementation of CMAQ as a Service, since the CMAQ-WS has been pre-installed and configured inside the VM, which is instantiated from the model-ready VM template image, users do not have to spend time on the installation and configuration of CMAQ model. Technically, developers still have to set up the original VM template image, and this process is required and not simplified. But once the model VM template image is ready, the model VM would be directly

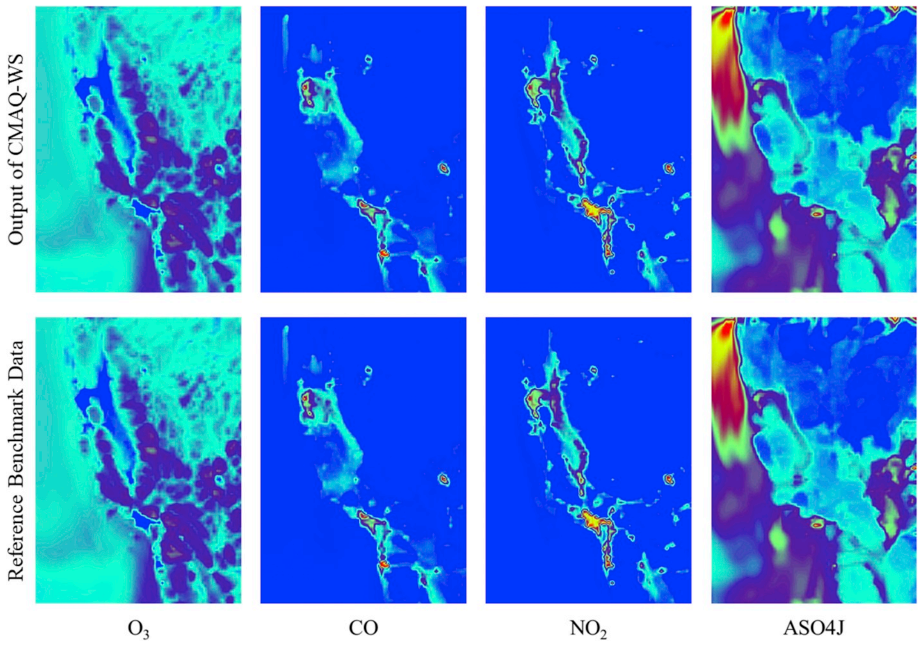


**Fig. 12.** Comparison between CMAQ-WS outputs and reference benchmark data.

instantiated without installation and configuration for each use of any user.

The second question is how can the model be published as a service and interoperated through the standardized interface. The feasibility of implementing CMAQ model as the WPS has been explored in this study. With the support of OGC WPS standard interface, the CMAQ-WS is interoperable with any standardized service. As the originality of this work, the Cloud WPS framework provides a general solution to integrate Earth science model, WPS, and cloud infrastructure. In a Cloud WPS framework enabled system, all models, geoprocessing, are EO data are deployed as VM instance. All instances are isolated but can be interoperable with each other.

The third question is how much performance of the model can be improved by moving the computing from local to the cloud. The implementation of the CMAQ-WS prototype is completely powered by GeoBrain Cloud. It is an IaaS architecture-based cloud platform that offers a lot of advantages including: (1) services and applications that are deployed on the cloud are easy to access, and data stored in the cloud are easy to share; (2) rather than storing in the local storage of individual computing devices, data and software that stored in the cloud are more reliable and ready for use; (3) computing capability offered by the cloud infrastructure is much more powerful and flexible than local computing resources traditionally available to researchers; and (4) users do not have to take responsibility for maintaining the computing resources. These merits could significantly benefit the Earth scientists, especially those who are non-professional IT users but dealing with large-scale EO data processing.

The originality of this work is the Cloud WPS framework, which provides a general solution to integrate Earth science model, WPS, and cloud infrastructure. Compared to the traditional way of Earth science modeling, this framework brings a lot of benefits, such as simplicity, ease of use, and no local computing resource requirement, to Earth scientists. However, the current prototypical implementation of the Cloud WPS framework still has some limitations. The input data of the CMAQ model are usually generated from the WRF model. In this study, we focused on the implementation of the CMAQ model and assumed that the WRF meteorology and SMOKE emissions data are ready to use. To build a full environmental modeling system and implement WRF-CMAQ two-way coupled model upon the current prototype, the WRF as a Service needs to be implemented along with the CMAQ as a Service.

The prototype has been implemented only on the GeoBrain Cloud platform and tested internally. To test its interoperability between the GeoBrain Cloud and other private and public cloud, the prototype needs to be migrated to other private and public cloud. Although the dissemination of VM is not a labor-intensive task, the large size of the model VM template image could be a limitation when disseminating the VM. For each instance of the prototype, it contains not only model related programs but the entire operating system. A potential solution is adding a Docker container layer between the service layer and the platform layer in each VM instance. By integrating the framework with Docker, the model, client, data, or standard interface could be exported as the Docker image and may significantly facilitate the dissemination of the service.

In the next stage, we are going to tackle the above limitations and improve the proposed framework. Additionally, further tests will be performed by applying advanced parallel computing approaches, such as MapReduce, with a larger volume of data from different sources. The CMAQ model will be updated from CMAQv5.1 to CMAQv5.2, which is the latest version of CMAQ software. Moreover, more Earth science models other than CMAQ will be tested with the Cloud WPS framework. To make the prototype more powerful and more flexible, a series of upgrades to the current geospatial Web services and cloud infrastructure will be conducted. More services and datasets will be registered into CyberConnector, and more high-performance computing resources will be added to the GeoBrain Cloud. With the improvement of both software and hardware, more features and functions of cloud-based WPS will be explored in the future.

## 5. Conclusion

This study explored the cloud-based Web Processing Service and presented the Cloud WPS framework. Particularly, a prototype of CMAQ as a Service was implemented as CMAQ-WS. All components of the prototype, including Web-based client, standard interfaces, models, EO data, and geoprocessing, were deployed as VM instances, managed by GeoBrain Cloud, and seamlessly bridged by the CyberConnector. A group of validation experiments were performed using the CMAQv5.1 benchmark dataset to validate the implementation. The result shows the proposed framework provides a general solution for integrated manipulation of models, data, and processes through OGC standard interfaces over the cloud. By implementing CMAQ-WS inside the cloud infrastructure, the performance of the CMAQ model can be significantly improved. Meanwhile, Earth scientists would be relieved from model installation, data collection, and large-scale computation so that they can concentrate on research issues instead of tedious tasks on software installation and maintenance as well as data collection and preprocessing.

## Acknowledgment

This work is supported by grants from NSF EarthCube program (Grants # ICER-1440294 and AGS-1740693, PI: Dr. Liping Di) and OGC Testbed 13. The authors would like to thank Ms. Julia Di of Columbia University for editing and proofreading the manuscript.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2018.11.019.

## References

Akimoto, H., 2003. Global air quality and pollution. Science 302, 1716–1719. https://doi.org/10.1126/science.1092666.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2009. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Generat. Comput. Syst. 25, 599–616. https://doi.org/10.1016/j.future.2008.12.001.

Castronova, A.M., Goodall, J.L., Elag, M.M., 2013. Models as web services using the open geospatial Consortium (OGC) web processing service (WPS) standard. Environ. Model. Software 41, 72–83. https://doi.org/10.1016/j.envsoft.2012.11.010.

Chen, N., Gong, J., Di, L., Yu, G., 2010. Automatic on-demand data feed service for AutoChem based on reusable geo-processing workflow. IEEE J. Select. Topics Appl. Earth Observat. Remote Sens. 3, 418–426. https://doi.org/10.1109/JSTARS.2010.2049094.

Chen, Z., Chen, N., Yang, C., Di, L., 2012. Cloud computing enabled web processing service for earth observation data processing. IEEE J. Select. Topics Appl. Earth Observat. Remote Sens. 5, 1637–1649. https://doi.org/10.1109/JSTARS.2012.2205372.

CMAS, 2018. CMAS: Community Modeling and Analysis System. [WWW Document]. URL. https://www.cmascenter.org/cmaq/, Accessed date: 14 May 2018.

CMAS, 2012. Operational Guidance for the Community Multiscale Air Quality (CMAQ) Modeling System.

Di, L., 2016. Big data and its applications in agro-geoinformatics. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Presented at the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 189–191. https://doi.org/10.1109/IGARSS.2016.7729040.

Di, L., 2007. Geospatial Sensor Web and Self-adaptative Earth Systems (SEPS). pp. 1–4.

Di, L., 2004a. GeoBrain-a web services based geospatial knowledge building system. In: Proceeding of NASA Earth Science Technology Conference. Palo Alto, CA, USA, pp. 8.

Di, L., 2004b. Distributed Geospatial Information Services-architectures, Standards, and Research Issues. pp. 187–193.

Di, L., Sun, Z., Yu, E., Song, J., Tong, D., Huang, H., Wu, X., Domenico, B., 2016. Coupling of Earth science models and earth observations through OGC interoperability specifications. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Presented at the 2016 IEEE International Geoscience and Remote Sensing Symposium. IGARSS), pp. 3602–3605. https://doi.org/10.1109/IGARSS.2016.7729933.

Di, L., Zhao, P., Yang, W., Yu, G., Yue, P., 2005. Intelligent geospatial web services. In: International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1229–1232. 2. https://doi.org/10.1109/IGARSS.2005.1525340.

Domenico, B., Nativi, S. (Eds.), 2013. CF-netCDF3 Data Model Extension Standard.

Dubois, G., Schulz, M., Skøien, J., Bastin, L., Peedell, S., 2013. eHabitat, a multi-purpose Web Processing Service for ecological modeling. Environ. Model. Software 41, 123–133. https://doi.org/10.1016/j.envsoft.2012.11.005.

EOSDIS, 2018. EOSDIS Annual Metrics Reports. [WWW Document]. Earthdata. URL. https://earthdata.nasa.gov/about/system-performance/eosdis-annual-metrics-reports, Accessed date: 14 June 2018.

Ercan, M.B., Goodall, J.L., Castronova, A.M., Humphrey, M., Beekwilder, N., 2014. Calibration of SWAT models using the cloud. Environ. Model. Software 62, 188–196. https://doi.org/10.1016/j.envsoft.2014.09.002.

Essawy, B.T., Goodall, J.L., Zell, W., Voce, D., Morsy, M.M., Sadler, J., Yuan, Z., Malik, T., 2018. Integrating scientific cyberinfrastructures to improve reproducibility in computational hydrology: example for HydroShare and GeoTrust. Environ. Model. Software 105, 217–229. https://doi.org/10.1016/j.envsoft.2018.03.025.

Feng, M., Liu, S., Euliss, N.H., Young, C., Mushet, D.M., 2011. Prototyping an online wetland ecosystem services model using open model sharing standards. Environ. Model. Software 26, 458–468. https://doi.org/10.1016/j.envsoft.2010.10.008.

Ghazouani, S., Slimani, Y., 2017. A survey on cloud service description. J. Netw. Comput. Appl. 91, 61–74. https://doi.org/10.1016/j.jnca.2017.04.013.

Goodall, J.L., Robinson, B.F., Castronova, A.M., 2011. Modeling water resource systems using a service-oriented computing paradigm. Environ. Model. Software 26, 573–582. https://doi.org/10.1016/j.envsoft.2010.11.013.

Granell, C., Díaz, L., Gould, M., 2010. Service-oriented applications for environmental models: reusable geospatial services. Environ. Model. Software 25, 182–198. https://doi.org/10.1016/j.envsoft.2009.08.005.

Horsburgh, J.S., Aufdenkampe, A.K., Mayorga, E., Lehnert, K.A., Hsu, L., Song, L., Jones, A.S., Damiano, S.G., Tarboton, D.G., Valentine, D., Zaslavsky, I., Whitenack, T., 2016. Observations Data Model 2: a community information model for spatially discrete Earth observations. Environ. Model. Software 79, 55–74. https://doi.org/10.1016/j.envsoft.2016.01.010.

Jin, X., Robinson, K., Lee, A., Polhill, J.G., Pritchard, C., Parker, D.C., 2017. A prototype cloud-based reproducible data analysis and visualization platform for outputs of agent-based models. Environ. Model. Software 96, 172–180. https://doi.org/10.1016/j.envsoft.2017.06.010.

Kampa, M., Castanas, E., 2008. Human health effects of air pollution. Environ. Pollut. 151, 362–367. https://doi.org/10.1016/j.envpol.2007.06.012.

Kurtz, W., Lapin, A., Schilling, O.S., Tang, Q., Schiller, E., Braun, T., Hunkeler, D., Vereecken, H., Sudicky, E., Kropf, P., Hendricks Franssen, H.-J., Brunner, P., 2017. Integrating hydrological modelling, data assimilation and cloud computing for real-time management of water resources. Environ. Model. Software 93, 418–435. https://doi.org/10.1016/j.envsoft.2017.03.011.

Morsy, M.M., Goodall, J.L., O'Neil, G.L., Sadler, J.M., Voce, D., Hassan, G., Huxley, C., 2018. A cloud-based flood warning system for forecasting impacts to transportation infrastructure systems. Environ. Model. Software 107, 231–244. https://doi.org/10.1016/j.envsoft.2018.05.007.

Nativi, S., Mazzetti, P., Geller, G.N., 2013. Environmental model access and interoperability: the GEO Model Web initiative. Environ. Model. Software 39, 214–228. https://doi.org/10.1016/j.envsoft.2012.03.007.

Nativi, S., Mazzetti, P., Santoro, M., Papeschi, F., Craglia, M., Ochiai, O., 2015. Big data challenges in building the global earth observation system of systems. Environ. Model. Software 68, 1–26. https://doi.org/10.1016/J.ENVSOFT.2015.01.017.

OGC, 2018a. Web Processing Service. [WWW Document]. URL. http://www.opengeospatial.org/standards/wps, Accessed date: 14 May 2018.

OGC, 2018b. OGC WPS 2.0.2 Interface Standard: Corrigendum 2.

OGC, 2007. OpenGIS Web Processing Service.

Pacheco, P.S., 1997. Parallel Programming with MPI. Morgan Kaufmann.

Rosatti, G., Zorzi, N., Zugliani, D., Piffer, S., Rizzi, A., 2018. A Web Service ecosystem for high-quality, cost-effective debris-flow hazard assessment. Environ. Model. Software 100, 33–47. https://doi.org/10.1016/j.envsoft.2017.11.017.

Shao, Y., Di, L., Bai, Y., Guo, B., Gong, J., 2012. Geoprocessing on the Amazon cloud computing platform AWS. In: International Conference on Agro-geoinformatics, pp. 286–291. https://doi.org/10.1109/Agro-Geoinformatics.2012.6311655.

Sun, Z., Di, L., Hao, H., Wu, X., Tong, D.Q., Zhang, C., Virgei, C., Fang, H., Yu, E.G., Tan, X., Yue, P., Lin, L., 2017a. CyberConnector: a service-oriented system for automatically tailoring multisource Earth observation data to feed Earth science models. Earth Sci. India 1–17. https://doi.org/10.1007/s12145-017-0308-4.

Sun, Z., Di, L., Heo, G., Zhang, C., Fang, H., Yue, P., Jiang, L., Tan, X., Guo, L., Lin, L., 2017b. GeoFairy: towards a one-stop and location based service for geospatial information retrieval. Comput. Environ. Urban Syst. 62, 156–167. https://doi.org/10.1016/j.compenvurbsys.2016.11.007.

Swain, N.R., Latu, K., Christensen, S.D., Jones, N.L., Nelson, E.J., Ames, D.P., Williams, G.P., 2015. A review of open source software solutions for developing water resources web applications. Environ. Model. Software 67, 108–117. https://doi.org/10.1016/j.envsoft.2015.01.014.

Tan, X., Di, L., Deng, M., Huang, F., Ye, X., Sha, Z., Sun, Z., Gong, W., Shao, Y., Huang, C., 2016. Agent-as-a-service-based geospatial service aggregation in the cloud: a case study of flood response. Environ. Model. Software 84, 210–225. https://doi.org/10.1016/j.envsoft.2016.07.001.

USEPA, 2017. Criteria Air Pollutants. [WWW Document]. URL. https://www.epa.gov/criteria-air-pollutants, Accessed date: 14 May 2018.

Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C.J.A., Buytaert, W., 2015. Web technologies for environmental big data. Environ. Model. Software 63, 185–198. https://doi.org/10.1016/j.envsoft.2014.10.007.

Wang, L., Chen, D., Hu, Y., Ma, Y., Wang, J., 2013. Towards enabling cyberinfrastructure as a service in clouds. Comput. Electr. Eng. 39, 3–14. https://doi.org/10.1016/j.compeleceng.2012.05.001.

Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D., 2011. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? Int. J. Digit. Earth 4, 305–329. https://doi.org/10.1080/17538947.2011.587547.

Yu, E.G., Liping, D., Bei, Z., Huilin, W., 2010. Coordination through geospatial web service workflow in the sensor web environment. Selected topics in applied earth observations and remote sensing. IEEE J. 3, 433–441. https://doi.org/10.1109/JSTARS.2010.2049477.

Yue, P., Baumann, P., Bugbee, K., Jiang, L., 2015a. Towards intelligent GIServices. Earth Sci. Informat. 1373–1376. 2015-Novem. https://doi.org/10.1109/IGARSS.2015.7326032.

Yue, P., Gong, J., Di, L., Yuan, J., Sun, L., Sun, Z., Wang, Q., 2010. GeoPW: laying blocks for the geospatial processing web. Trans. GIS 14, 755–772. https://doi.org/10.1111/j.1467-9671.2010.01232.x.

Yue, P., Zhang, M., Tan, Z., 2015b. A geoprocessing workflow system for environmental monitoring and integrated modelling. Environ. Model. Software 69, 128–140. https://doi.org/10.1016/j.envsoft.2015.03.017.

Zhang, B., Di, L., Yu, G., Han, W., Wang, H., 2012. Towards data and sensor planning service for coupling earth science models and earth observations. IEEE J. Selected Topics Appl. Earth Observat. Remote Sens. 5, 1628–1636. https://doi.org/10.1109/JSTARS.2012.2195639.

Zhang, C., Di, L., Sun, Z., Yu, E.G., Hu, L., Lin, L., Tang, J., Rahman, M.S., 2017. Integrating OGC web processing service with cloud computing environment for earth observation data. In: 2017 6th International Conference on Agro-geoinformatics, Agro-geoinformatics 2017, . https://doi.org/10.1109/Agro-Geoinformatics.2017.8047065.