Astronomical Data Analysis Software and Systems XXI
ASP Conference Series, Vol. 461
Pascal Ballester, Daniel Egret, and Nuria P. F. Lorente, eds.
©2012 Astronomical Society of the Pacific

IceCore: A Web Portal for Workflow Execution

Sami Maisala, Tero Oittinen, Tuure Takala, Otto Solin, and Juhani Huovelin

Department of Physics, Division of Geophysics and Astronomy, University of Helsinki, Finland

Abstract. The large amount of data, complexity of data analysis software and increasing need of computing power are challenges for traditional desktop oriented data analysis. IceCore is a prototype web portal for executing scientific workflows in a workflow engine and for controlling administrative tasks of workflow execution and resources. IceCore is a scalable system that can be used as a common interface for workflows distributed in multiple workflow engine servers and for managing distributed databases.

1. Introduction

IceCore is a workflow engine manager with a portal as the user interface. IceCore can be used to select and run workflows stored in databases in a selected workflow engine, monitor the workflow executions and view the results via a collection of portlets (Figure 1). The logic behind the Three-Tier architectural design is that each tier is independent from the other tiers and can be implemented separately. Also the transfer of data should occur only between the adjacent tiers and not between the ends directly.

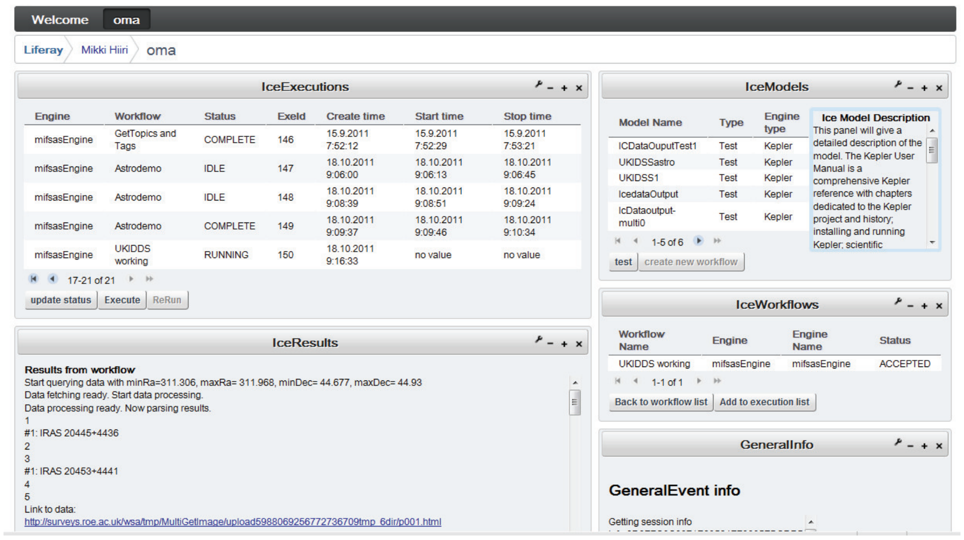
The system architecture is designed to be distributed and supports parallel computing of any number of attached workflows. Engines handle the data processing and computations and the system offers communication between engines, databases and GUI with Knowledge Bus. This combination of various engines with a middleware management system is powerful, highly scalable and extendable, and can be accessed from any remote location.

1.1. Scientific Workflows and Engines

Workflows are constructed from various modular components or individual business processes bound together to form a data flow. The flow takes place through the different type of links formed between the workflow subcomponents. Typically the workflow also has a number of inputs and outputs. The outputs correspond to end products or data inputs to another successive workflow. Workflow engines are the enacting or executing elements that run workflow pipelines. Examples of scientific workflow engines are Taverna[1] and Kepler[2]. A search for stellar clusters presented in a poster at this conference (Solin et al. 2012) has been implemented also as a Kepler workflow.

[1]`www.taverna.org.uk/`

[2]`https://kepler-project.org/`



IceExecutions

Engine	Workflow	Status	ExeId	Create time	Start time	Stop time
mifsasEngine	GetTopics and Tags	COMPLETE	146	15.9.2011 7:52:12	15.9.2011 7:52:29	15.9.2011 7:53:21
mifsasEngine	Astrodemo	IDLE	147	18.10.2011 9:06:00	18.10.2011 9:06:13	18.10.2011 9:06:45
mifsasEngine	Astrodemo	IDLE	148	18.10.2011 9:08:39	18.10.2011 9:08:51	18.10.2011 9:09:24
mifsasEngine	Astrodemo	COMPLETE	149	18.10.2011 9:09:37	18.10.2011 9:09:46	18.10.2011 9:10:34
mifsasEngine	UKIDDS working	RUNNING	150	18.10.2011 9:16:33	no value	no value

IceResults

```
Results from workflow
Start querying data with minRa=311.306, maxRa= 311.968, minDec= 44.677, maxDec= 44.93
Data fetching ready. Start data processing.
Data processing ready. Now parsing results.
1
#1: IRAS 20445+4436
2
3
#1: IRAS 20453+4441
4
5
Link to data:
http://surveys.roe.ac.uk/wsa/tmp/MultiGetImage/upload5988069256772736709tmp_6dir/p001.html
```

IceModels

Model Name	Type	Engine type
ICDataOuputTest1	Test	Kepler
UKIDSSastro	Test	Kepler
UKIDSS1	Test	Kepler
IcedataOutput	Test	Kepler
IcDataoutput-multi0	Test	Kepler

Ice Model Description
This panel will give a detailed description of the model. The Kepler User Manual is a comprehensive Kepler reference with chapters dedicated to the Kepler project and history; installing and running Kepler; scientific

IceWorkflows

Workflow Name	Engine	Engine Name	Status
UKIDDS working	mifsasEngine	mifsasEngine	ACCEPTED

GeneralInfo

GeneralEvent info

Getting session info

Figure 1. View from Liferay portal with IceCore portlets.

2. High Level Architecture

The IceCore system is following the Three-Tier architectural scheme in which the data, application and presentation layers are separated and developed independently. IceCore consists of four basic functional components that interact with each other: IceCore User Interface (ICUI), IceCore Controller (ICC), IceCore Engine (ICE) and IceCore Data Manager (ICDM). The fifth element Knowledge Bus can be described as an intelligent mediator which ties up interoperability with messaging between individual IceCore components like engines, databases and user interfaces.

The system supports service-oriented architecture (SOA) on implementing synchronous and asynchronous request-response messages to connect remote processes in a highly distributed system. Communication and data transfer between server, client and dynamic web pages are implemented using several techniques; internal IceCore component communication is using synchronous REST (Representational State Transfer) services, and communication between individual portlets are handled by loose coupling using EventBus object pattern with Remote Procedure Calls (RPC) based communication which is AJAX (Asynchronous JavaScript and XML) and GWT (Google Web Toolkit[3]). The overall architectural diagram is illustrated in Figure 2.

2.1. Architectural Design

- **User Tier** connects the user to the system through the portal UI (ICUI), where the execution of various workflows can be monitored and controlled, and results displayed to the GUI.
- **Business Tier** communicates with other components via the IceCore Controller (ICC). IceCore Engine (ICE) is a web service extension to the underlying execution engine.

[3]`http://code.google.com/webtoolkit/`

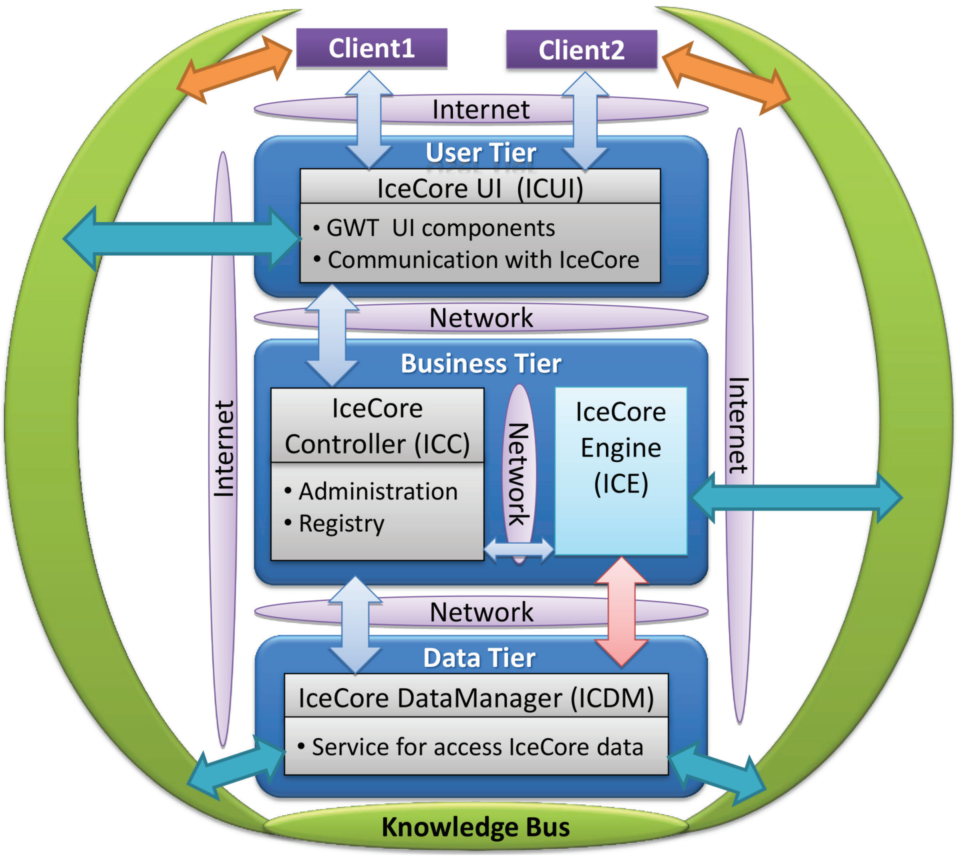


Figure 2. IceCore architectural design.

- **Data Tier** handles the data management for both ICUI and ICE with IceCore DataManager (ICDM).
- **Knowledge Bus** delivers messages between producer and listener applications.

3. Implementation of IceCore

In the top layer (User Tier) ICUI interacts with users via a portal interface (see Figure 1). ICUI is built on top of the open-source Liferay Portal[4] working as a portlet container. Portals represent a matured type of modular GUI, where aggregated views are built from a selection of modular view components called portlets, that are usually developed and deployed independently from each other. This way users can access most of the relevant information from a single web page.

In the middle layer (Business Tier) ICC is a component that acts as a middleware and provides HTTP REST interface to the underlying IceCore system. ICE is the web service extension to the actual workflow engine such as Kepler or Taverna. The

[4]`http://www.liferay.com`

prototype IceCore engine is developed using Kepler-2.0 workflow engine and KFlex Server,[5] which is an extended version of the Kepler release with Web Service interface for workflows and packaged as web application.

In the bottom layer ICDM acts as storage for all the knowledge. This instance, provides executable workflow models to the user, knows the location of the execution engines registered to the system, and keeps track of the executions and their results. The Data Manager is developed using Spring Framework[6] and model mapping to a relational persistence database (MySQL) is done using Hibernate.[7]

Knowledge Bus is a mixture of Web/Message-oriented middleware and Comet type of module in the IceCore system. Comet is an umbrella term for server "push" paradigm which is an inverse to the standard client-server request-response scheme. The server side "push" is implemented using client side long polling query mechanism to the Knowledge Bus. The role of the Knowledge Bus is to provide message interoperability between applications which are working in both server-side and client-side. Knowledge Bus receives messages from different types of applications and forwards messages to applications which are registered to receive this particular type of data.

4. Summary and Future Work

This paper summarizes the principals and architecture of the IceCore system for managing scientific workflows in a portal environment with existing WWW standards and browsers without using third-party extensions like Adobe Flash. The forthcoming release of HTML5 web standard with new features will hopefully ease the development of animated graphics (SVG Canvas) and simplify Comet development with web sockets.

Acknowledgments. We thank the Finnish Funding Agency for Technology and Innovation (TEKES) for making this research possible under the project Mifsas.

References

Solin, O., Ukkonen, E., Haikala, L., & Maisala, S. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASP Conf. Ser., 577

[5] `https://kepler-project.org/developers/interest-groups/webui/kflex`

[6] `http://www.springsource.org`

[7] `http://www.hibernate.org`