



How Distributed Data Mining Tasks can Thrive as Services on Grids

Domenico Talia and Paolo Trunfio

Università della Calabria, Italy

talia@deis.unical.it



NSF NGDM'07 – Baltimore - USA – 10-12 October, 2007

Outline

- Introduction
- The Grid for Data Mining
- Data Mining Tasks as Services
- Weka4WS
- Knowledge Grid
- Mobile Data Mining Services
- Final Remarks



Distributed data mining on the Grid

- **Knowledge discovery (KDD)** and **data mining (DM)** are:
 - compute- and data-intensive processes/tasks
 - Often based on distribution of data, algorithms, and users
- The Grid integrates both distributed computing and parallel computing, thus it can be **a key infrastructure** for high-performance distributed knowledge discovery.
- It also offers
 - security, information service, data access and management, communication, scheduling, fault detection, ...



Distributed data mining on the Grid

- The Grid extends the distributed and parallel computing paradigms allowing resource negotiation, dynamical allocation, heterogeneity, open protocols and services.
- As the Grid became a well accepted computing infrastructure it is necessary to **provide data mining services, algorithms, and applications.**
- Those may help users to **leverage Grid capability** in supporting high-performance distributed computing for **solving their data mining problems in a distributed way.**



Grid services for distributed data mining

- Exploiting the SOA model and the Web Services Resource Framework (**WSRF**) it is possible to define **basic services for supporting distributed data mining tasks** in Grids
- Those services can address all the aspects that must be considered in data mining and in knowledge discovery processes
 - data selection and transport services,
 - data analysis services,
 - knowledge models representation services, and
 - visualization services.



Grid services for distributed data mining

- It is possible to define services corresponding to

Single Steps

that compose a KDD process such as preprocessing, filtering, and visualization.

Single Data Mining Tasks

such as classification, clustering, and association rules discovery.

Distributed Data Mining Patterns

such as collective learning, parallel classification and meta-learning models.

Data Mining Applications or KDD processes

including all or some of the previous tasks expressed through a multi-step workflow.



Data mining Grid services

- This collection of data mining services can constitute an

Open Service Framework for Grid-based Data Mining

- Allowing developers to design distributed KDD processes as a composition of single services available over a Grid.
- Those services should exploit other basic Grid services for data transfer and management such as
 - Reliable File Transfer (RFT),
 - Replica Location Service (RLS),
 - Data Access and Integration (OGSA-DAI) and
 - Distributed Query processing (OGSA-DQP).



Data mining Grid services

- By exploiting the Grid services features it is possible to develop data mining services accessible every time and everywhere.
- This approach may result in
 - Service-based distributed data mining applications
 - Data mining services for virtual organizations.
 - A sort of knowledge discovery eco-system formed of a large numbers of decentralized data analysis services.



Grid services for distributed data mining

- Service-based systems we developed
 - Weka4WS
 - Knowledge Grid
 - Mobile Data Mining Grid Services



Knowledge Grid



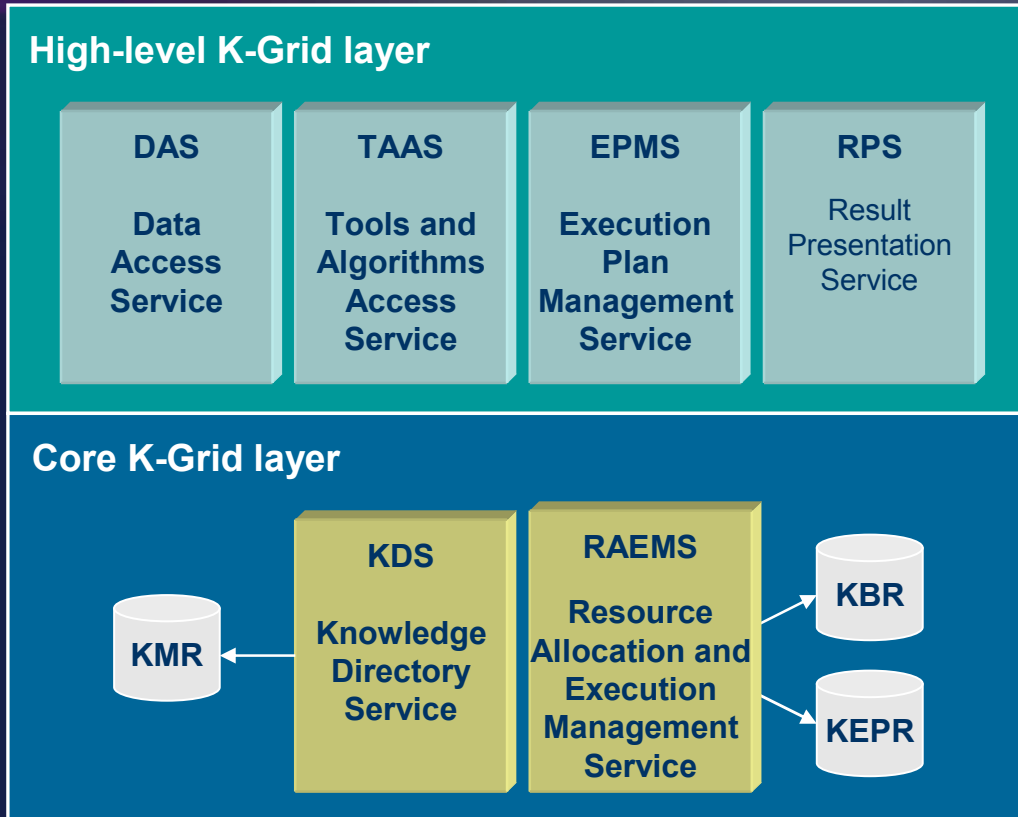
The Knowledge Grid

- **Knowledge Grid**: a distributed knowledge discovery architecture that can be configured on top of generic Grid middleware
- A first prototype has been implemented on GT2 based on a high-level user interface for application composition (**VEGA**)
- The Knowledge Grid services are currently being re-implemented as **WSRF-compliant Web Services**.

M. Cannataro, D. Talia, The Knowledge Grid, *Communications of the ACM*, vol. 46, no. 1, pp. 89-93, 2003.

Knowledge Grid architecture

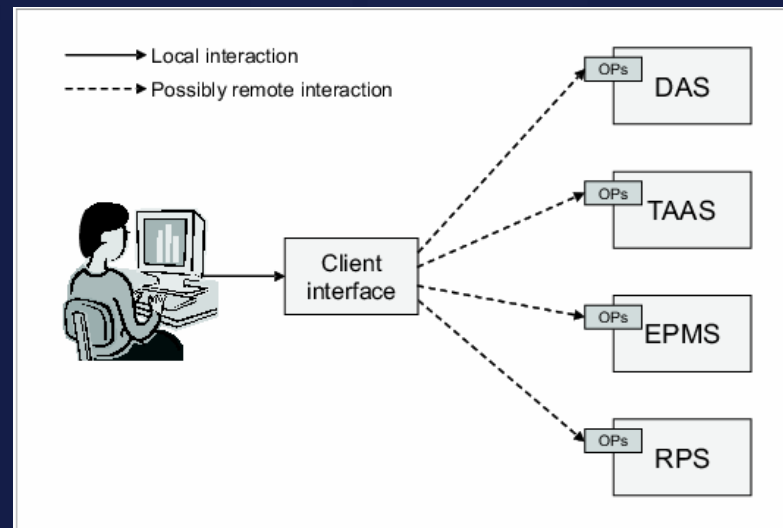
Knowledge Grid Layers



Basic Grid services

The Knowledge Grid and WSRF

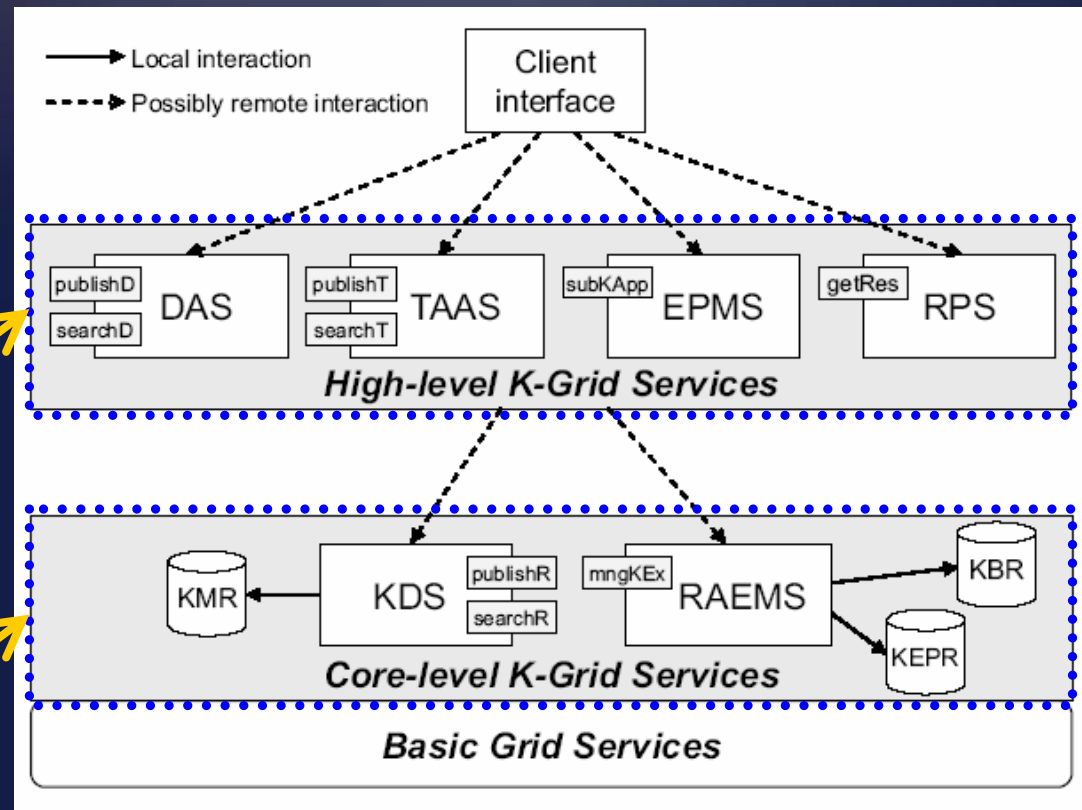
- The Knowledge Grid services are currently being re-implemented as **WSRF-compliant Web Services**.
- They can be invoked by client interfaces, programs, and other services



A. Congiusta, D. Talia, P. Trunfio, Distributed Data Mining Services Leveraging WSRF, *Future Generation Computer Systems*, vol. 23, no. 1, pp. 34-41, 2007.

The Knowledge Grid and WSRF

- Each K-Grid service is exposed as a Grid Service that exports one or more operations using WSRF
- The operations exported by the High-level K-Grid services are invoked by user-level applications
- The operations provided by the Core K-Grid services are invoked both by High-level and Core K-Grid services



Knowledge Grid: Service operations

Service	Operation	Description
DAS	publishData	This operation is invoked by a client for publishing a newly available dataset. The publishing requires a set of information that will be stored as metadata in the local KMR.
	searchData	Data to be used in a KDD computation is located during the application design by invoking this operation. The searching is performed on the basis of appropriate parameters.
TAAS	publishTools	This operation is used to publish metadata about a data mining tool in the local KMR. As a result of the publishing, a new DM service is made available for utilization in KDD computations.
	searchTools	It is similar to the searchData operation except that it is targeted to data mining tools.

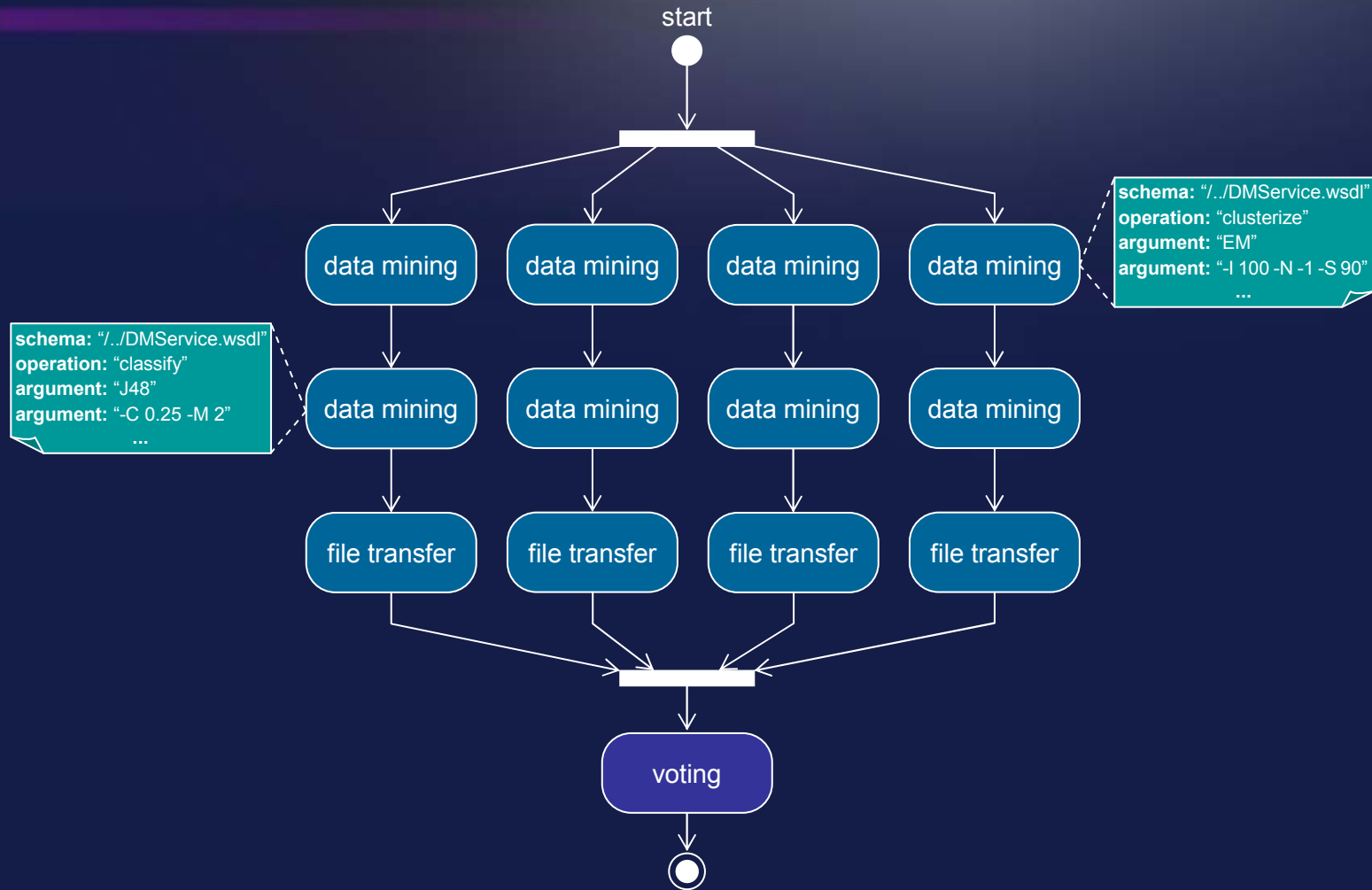


Knowledge Grid: Service operations

Service	Operation	Description
EPMS	submitKApplication	This operation receives a conceptual model of the application to be executed. The EPMS generates a corresponding abstract execution plan and submits it to the RAEMS for its execution.
RPS	getResults	Retrieves results of a performed KDD computation and presents them to the user.
KDS	publishResource	This is the basic, core-level operation for publishing data or tools. It is thus invoked by the DAS or TAAS services for performing their own specific operations.
	searchResource	The core-level operation for searching data or tools.
RAEMS	manageKExecution	This operation receives an abstract execution plan of the application. The RAEMS generates an instantiated execution plan and manages its execution.



Knowledge Grid: High-level application design



Weka4WS



The Weka4WS framework

- **Weka** is one of the most used open source suite for data mining.
- In Weka, the overall data mining process takes place on a single machine; the algorithms can be only locally executed.
- **Weka4WS** extends Weka to support **distributed execution of the Weka data mining algorithms**
 - All data mining algorithms provided by the Weka library are **exposed as WSRF-compliant Web Services**
 - Globus Toolkit 4 is used for basic Grid functionalities such as security and data transfer.

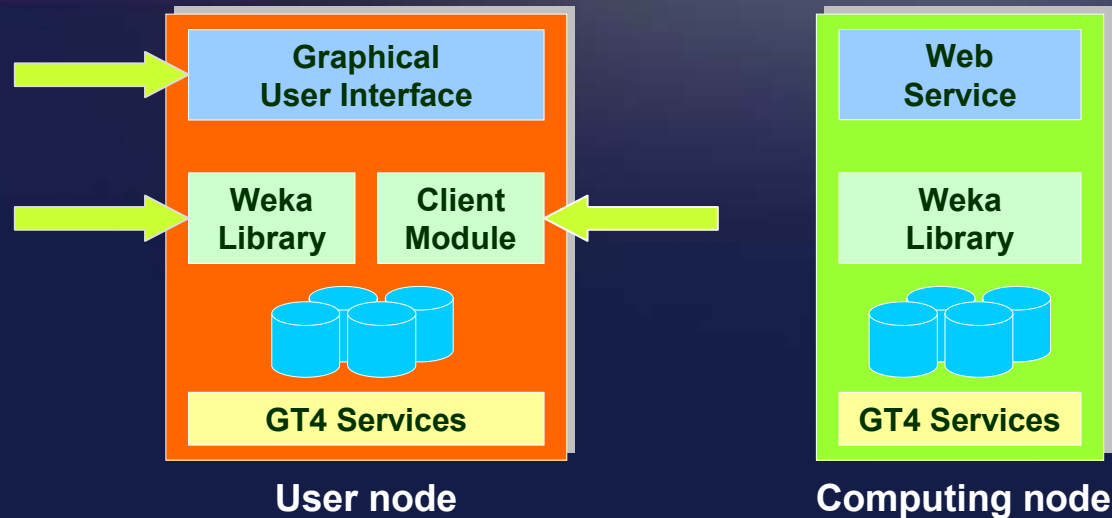
Talia D. , Trunfio P. , Verta O., Weka4WS: a WSRF-enabled Weka Toolkit for Distributed Data Mining on Grids. *Proc. PKDD 2005*, LNCS, pp. 309-320, 2005.

Weka4WS architecture

- We distinguish Weka4WS nodes in two categories:
 - **user nodes**, which are the local machines of the users providing the Weka4WS client software
 - **computing nodes**, which provide the Weka4WS Web Services allowing the execution of remote data mining tasks
- Data can be located on *computing nodes*, *user nodes*, or third-party nodes
- If the dataset to be mined is not available on a computing node, it can be copied or replicated by means of the GT4 data management services.

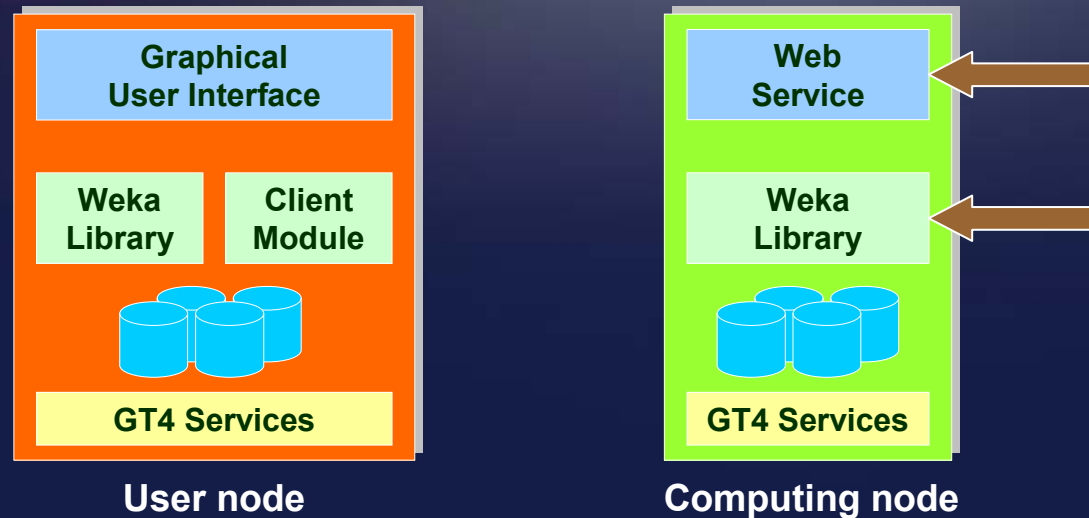


Software components



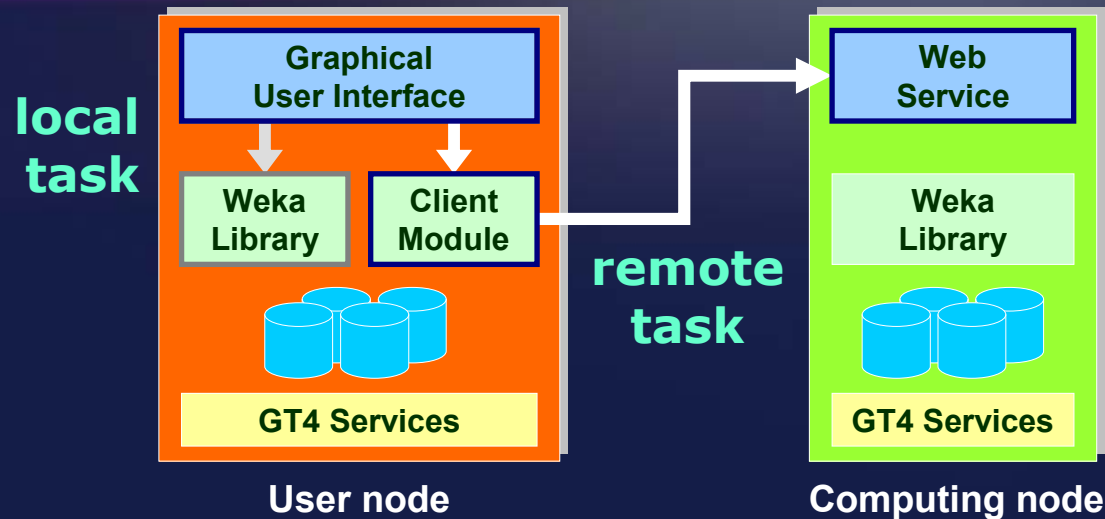
- User nodes include three software components:
 - Graphical User Interface (GUI)
 - Client Module (CM)
 - Weka Library (WL)

Software components



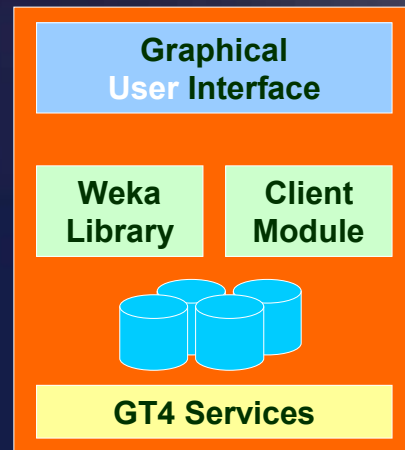
- Computing nodes include two software components:
 - Web Service (WS)
 - Weka Library (WL)

Software components

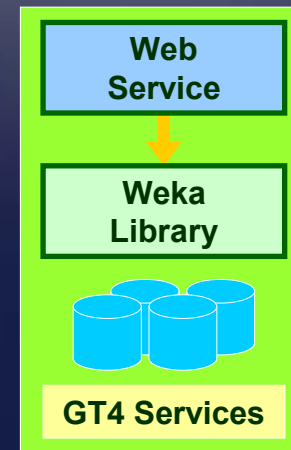


- The GUI extends the Weka Explorer environment to allow the execution of both *local* and *remote* data mining tasks:
 - *local tasks* are executed by directly invoking the local WL
 - *remote tasks* are executed through the CM, which operates as an intermediary between the GUI and Web Services on remote computing nodes

Software components



User node



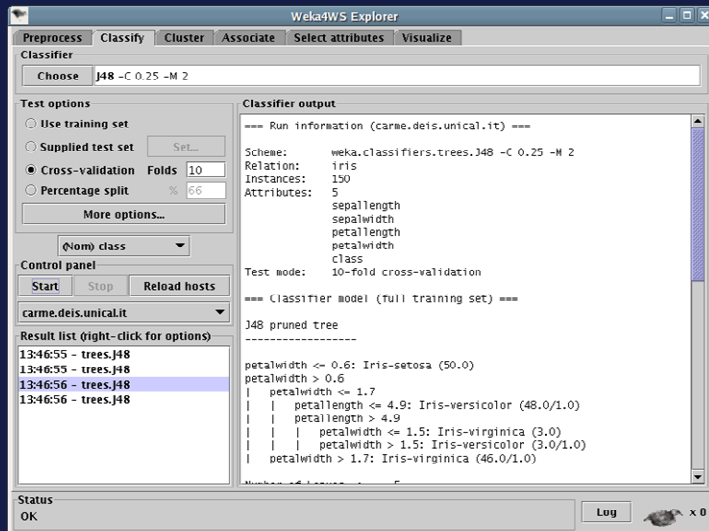
Computing node

Algorithm invocation

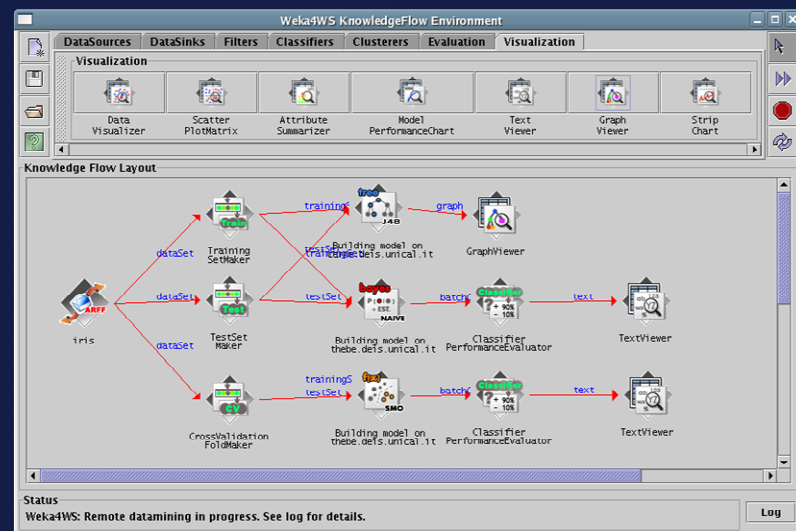
- The WS is a WSRF-compliant Web Service that exposes the data mining algorithms provided by the underlying Weka Library
- Requests to the WS are executed by invoking the corresponding WL algorithms

Weka4WS Graphical User Interfaces

- Weka4WS extends the GUIs of Weka:
 - **Explorer**
 - available with **Weka4ws 1.0** (grid.deis.unical.it/weka4ws)
 - **KnowledgeFlow**

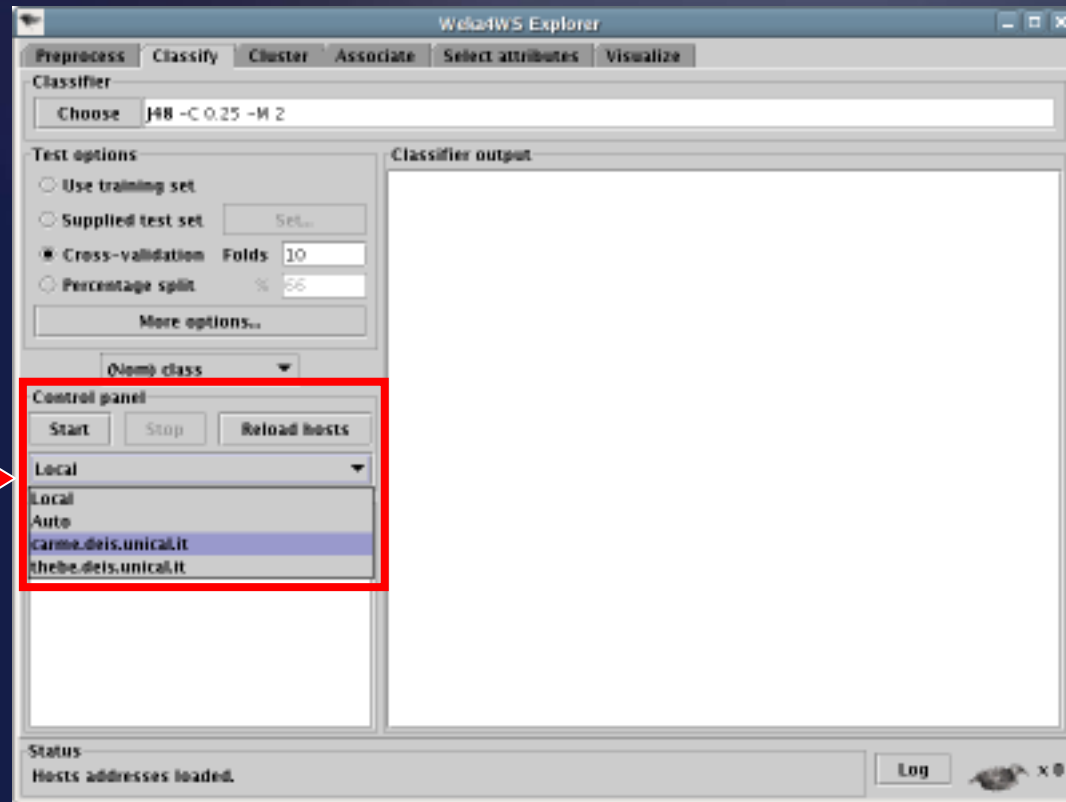


Explorer



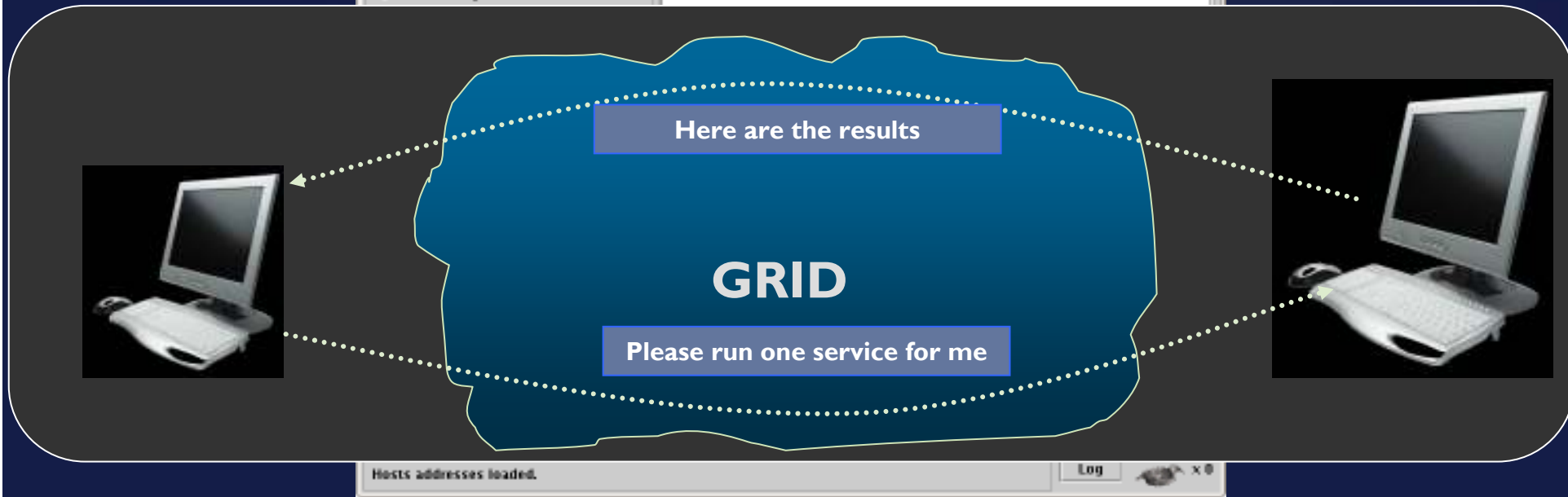
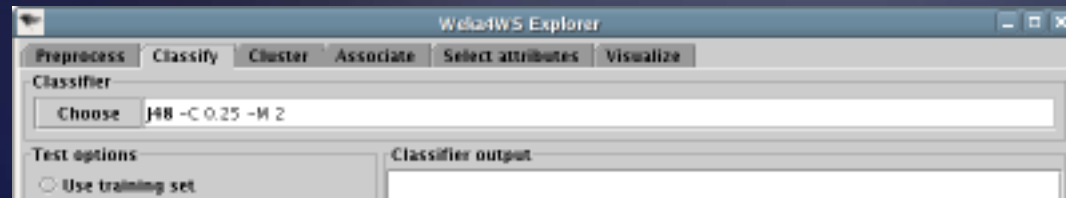
KnowledgeFlow

Weka4WS Explorer



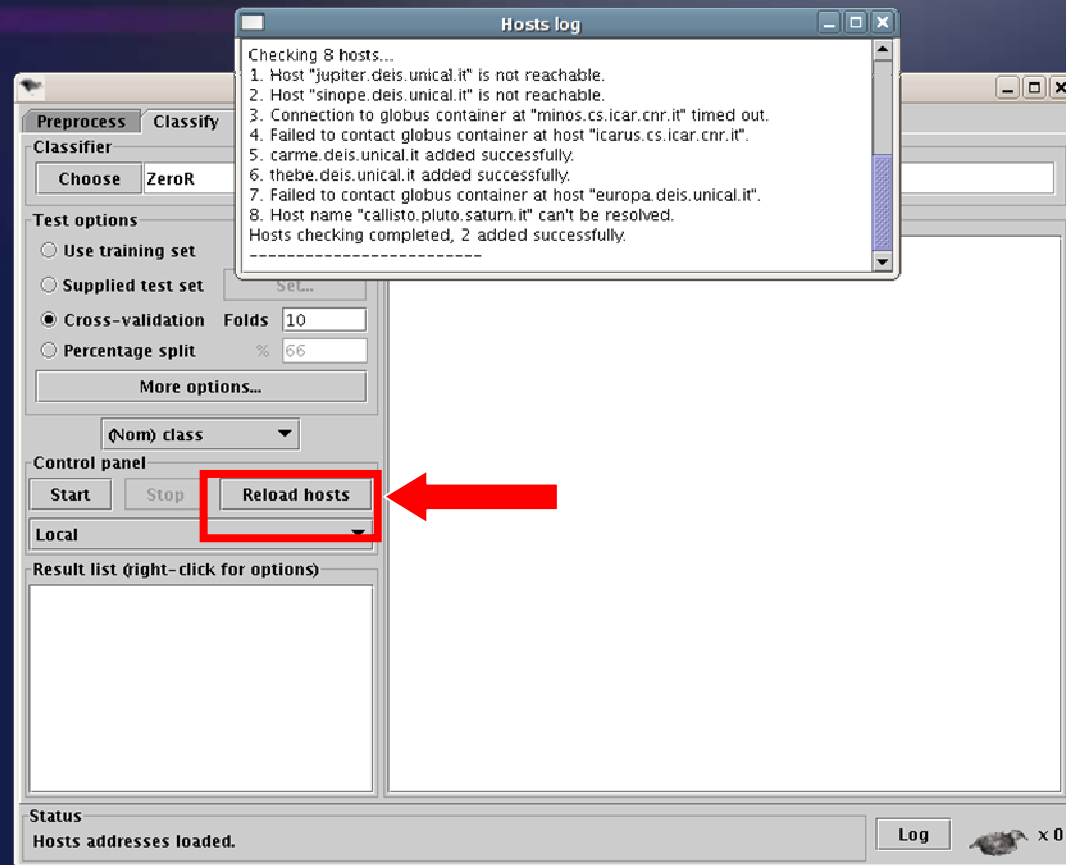
- A “Control panel” allows users to submit both local and remote tasks has been added to the original Weka Explorer environment

Weka4WS Explorer



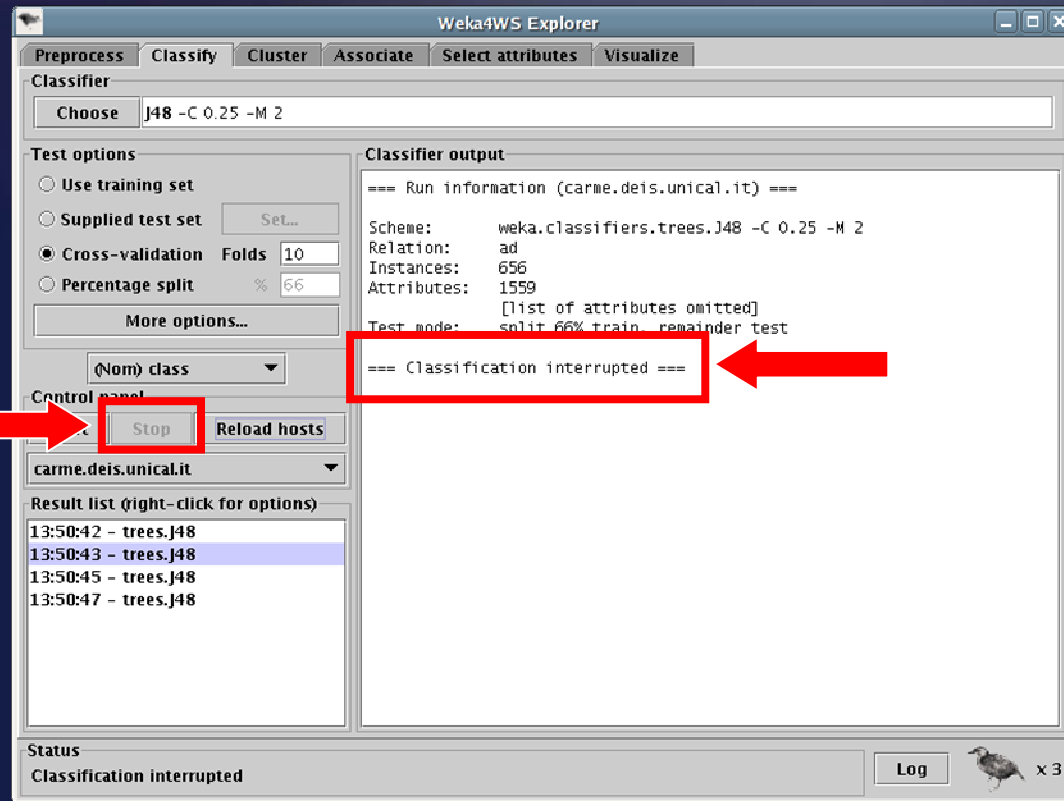
- A drop down menu allows to choose where to run the current data mining task (“Local”, “Auto”, or a specific host)

Weka4WS Explorer



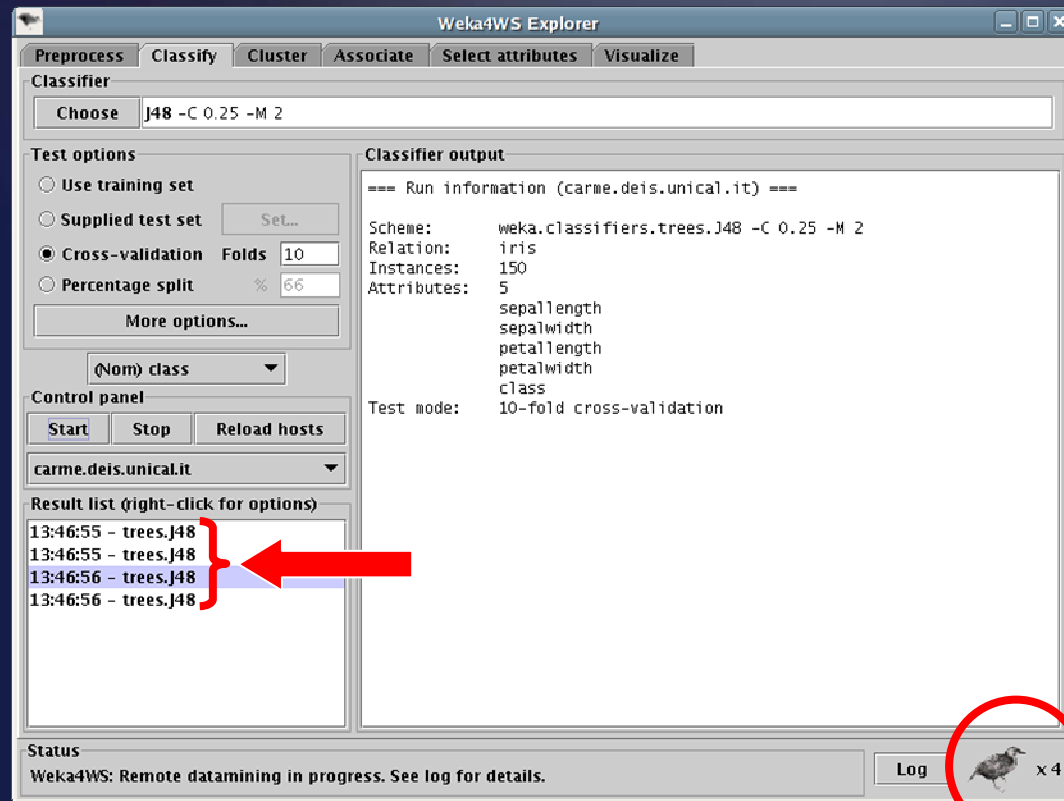
- A button allows to reload the list of hosts and check for the availability of the Globus container on each host

Weka4WS Explorer



- A button allows to stop, if needed, both the local and the remote computation of the data mining tasks

Weka4WS Explorer



- Each task in the GUI is managed by an independent thread. A user can start multiple data mining tasks at the same time on different remote hosts

Weka4WS Explorer

The screenshot shows the Weka4WS Explorer interface. At the top, a 'Log' window displays a detailed timeline of operations, including starting on Thursday, 7 June 2007, and various steps like 'Setting up transfer...', 'Resource creation...', and 'Remote data mining...'. Below the log, there are buttons for 'Start', 'Stop', and 'Reload hosts'. A dropdown menu shows the host 'thebe.deis.unical.it'. A 'Result list' on the left shows two entries: '10:13:24 - rules.ZeroR' and '10:13:27 - rules.ZeroR'. The main area displays a 'Detailed Accuracy By Class' table and a 'Confusion Matrix'.

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	1	0.333	1	0.5	Iris-setosa
0	0	0	0	0	Iris-versicolor
0	0	0	0	0	Iris-virginica

Below the table is a 'Confusion Matrix' section with the following text:

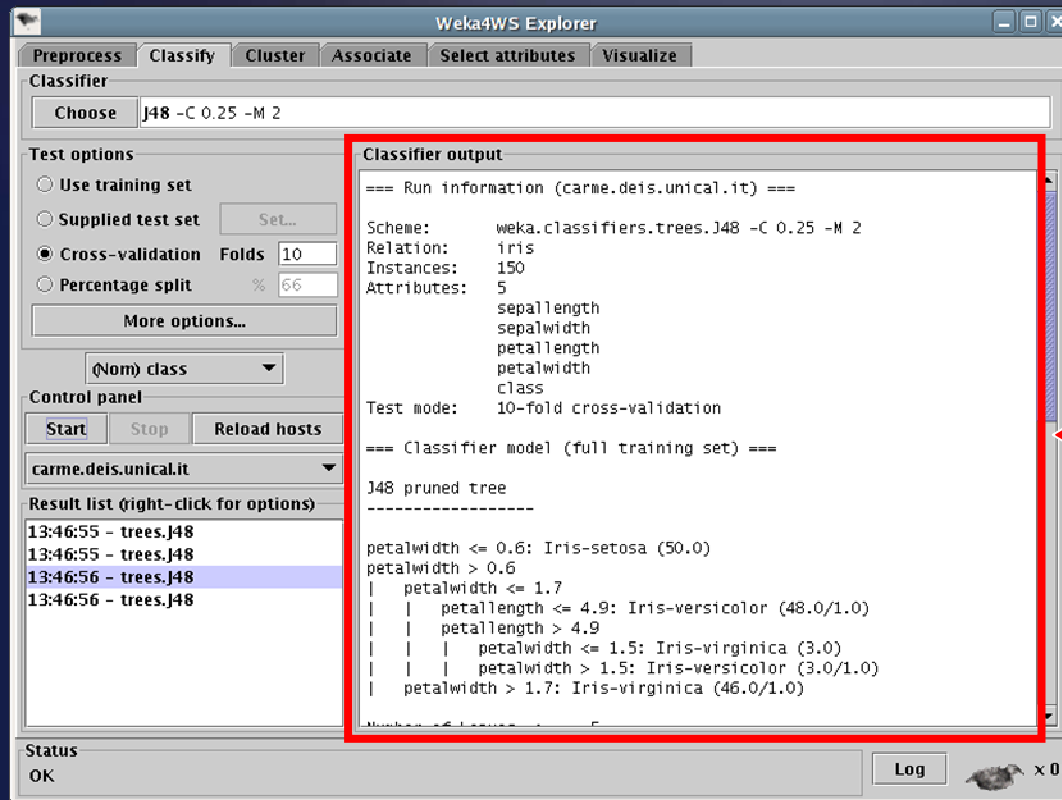
```

=== Confusion Matrix ===
 a  b  c  <-- classified as
50  0  0 | a = Iris-setosa
50  0  0 | b = Iris-versicolor
50  0  0 | c = Iris-virginica
    
```

At the bottom of the interface, there is a 'Status' bar showing 'OK'. A red arrow points to a 'Log' button in the bottom right corner, which is highlighted with a red box.

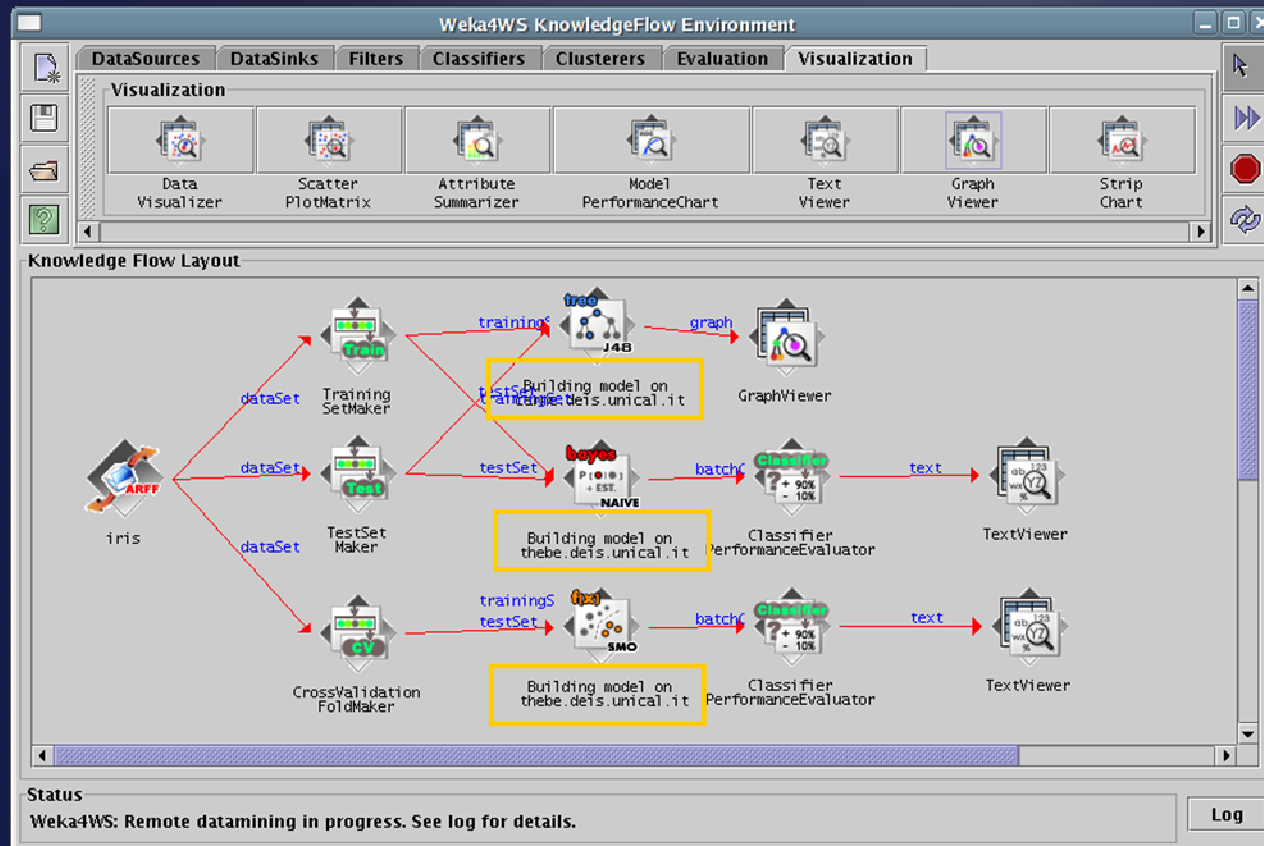
- The detailed log allows to follow the remote computations step by step

Weka4WS Explorer



- Whenever the output of a data mining task has been received from a remote computing node, it is visualized in the standard Output panel

Weka4WS KnowledgeFlow



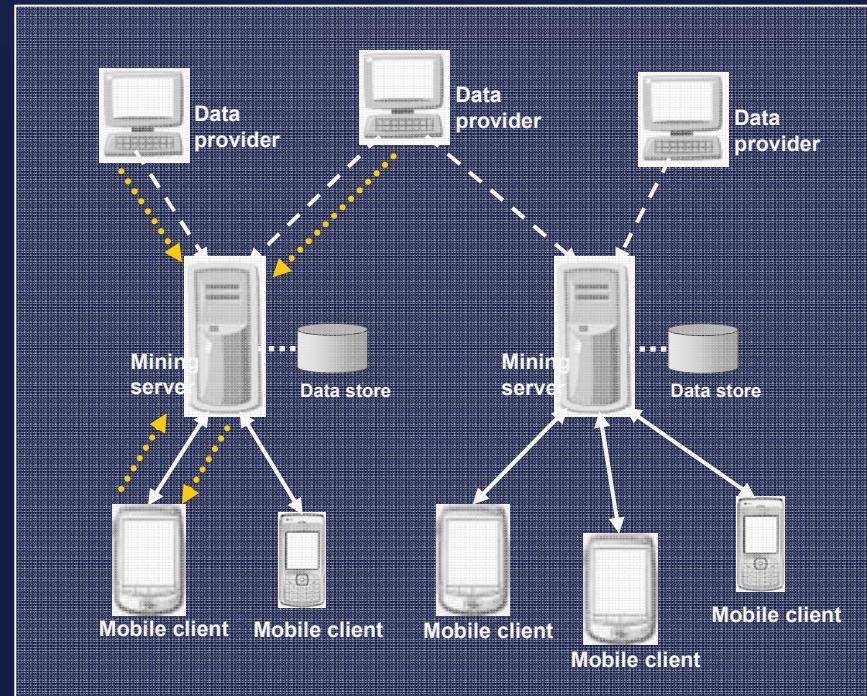
- A data mining workflow can be composed and run on several Grid nodes.

Mobile Data Mining Grid Services



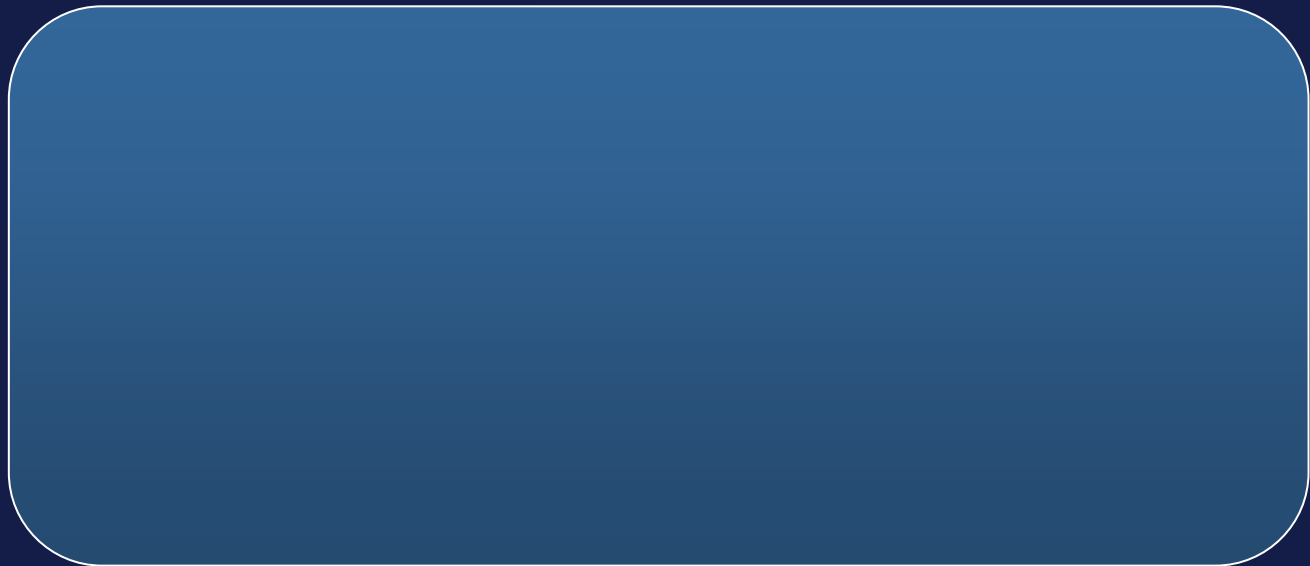
Grid Services for Mobile Data Mining

- The main research goal is to support a user to access **data mining services on mobile devices**.
- The system includes three components:
 - **Data providers.**
 - **Mining servers.**
 - **Mobile clients.**



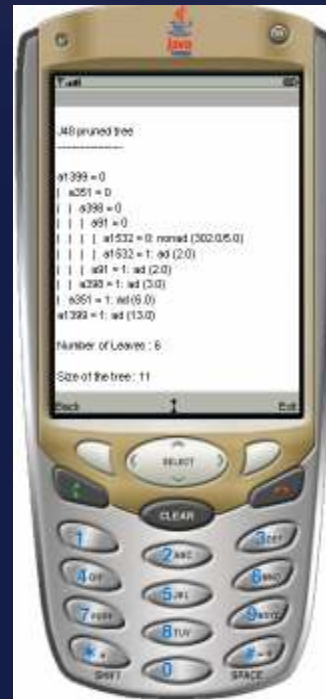
The Mining server

- A Mining server implements two Grid Services:
 - **Data Collection Service (DCS)**: invoked by a data provider to store data in the *data store*.
 - **Data Mining Service (DMS)**: invoked by a mobile client to ask for the execution of a data mining task.



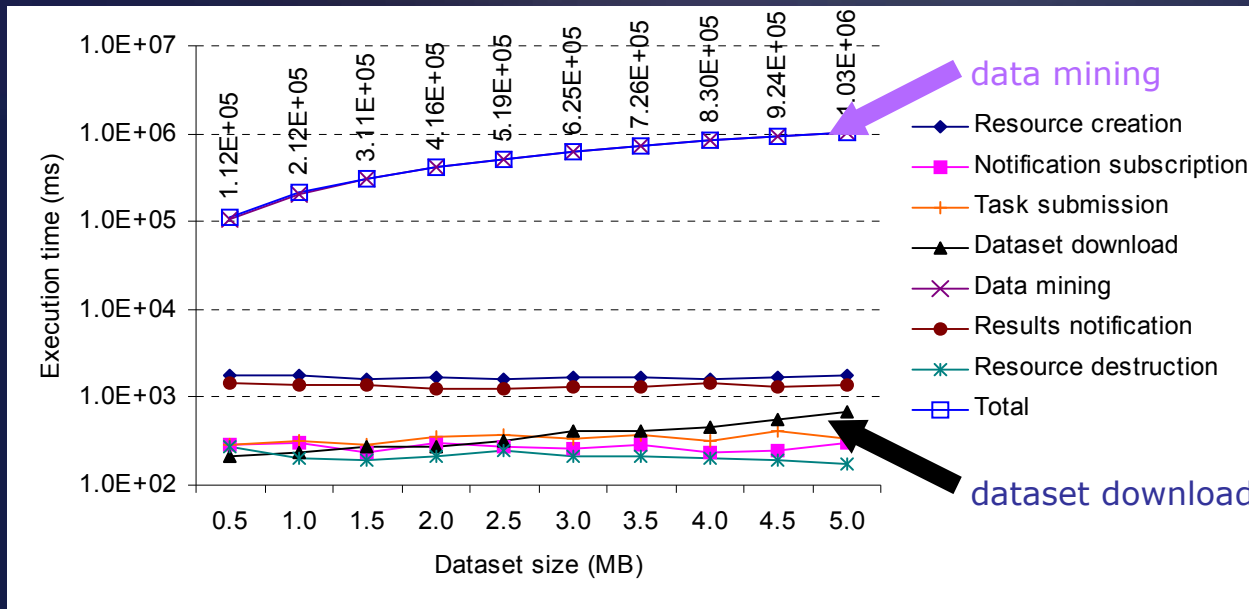
Grid Services for Mobile Data Mining

- A user can select which part of a result (data mining model) he wants to visualize.



Impact of the WSRF overhead

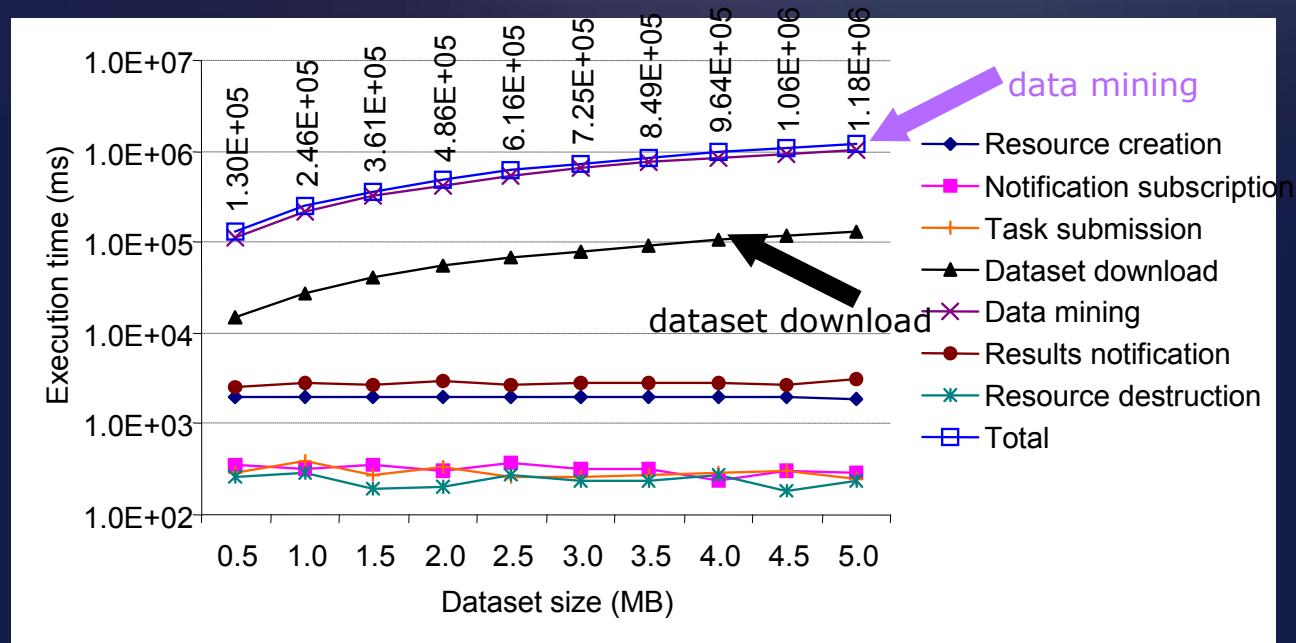
- Execution times



- It can be observed that the data mining phase takes approximately from **95% to 99%** of the total execution time
- Thus the overhead due to the **WSRF invocation mechanisms is negligible** for typical data mining tasks on large datasets

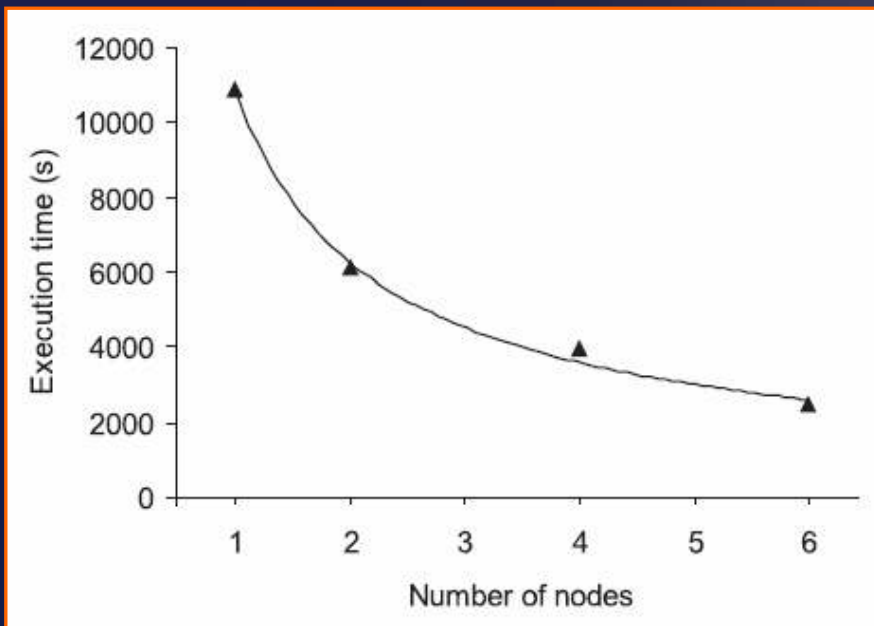
Impact of the WSRF overhead

- Execution times

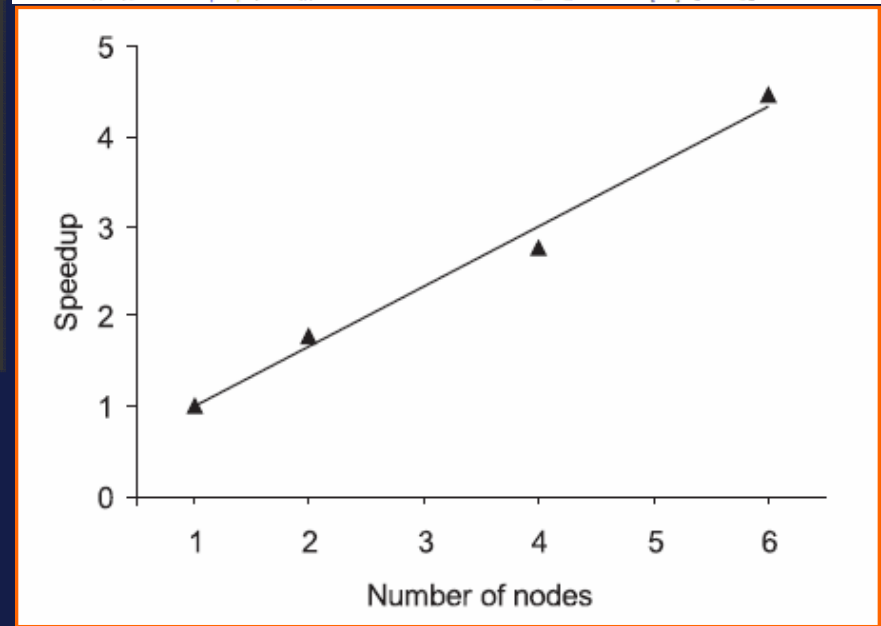


- In a larger scenario the data mining step represents from 85% to 88% of the total execution time, the dataset download takes about 11%, while the other steps range from 4% to 0.5%

Weka4WS: application speedup on a Grid



This document is a technical report from the CoreGRID project. It describes the performance of the Weka4WS application on a Grid. The application is designed to process large datasets in a distributed environment. The results show that the application can scale to a large number of nodes, achieving a speedup of approximately 4.5x when using 6 nodes. The application is based on the Weka machine learning library and is implemented as a distributed application using the Grid4J framework. The application is designed to be scalable and can be used to process large datasets in a distributed environment. The results show that the application can scale to a large number of nodes, achieving a speedup of approximately 4.5x when using 6 nodes. The application is based on the Weka machine learning library and is implemented as a distributed application using the Grid4J framework.



Final remarks

- Single data mining tasks can be delivered as Grid services, knowledge discovery processes can be implemented as complex Grid services.
- Scientific and Business VOs can benefit from their integration and availability
- Systems like the KNOWLEDGE GRID and Weka4WS show the effectiveness of the approach.
- In a long-term vision, pervasive collections of data mining services and applications will be accessed and used as public utilities.



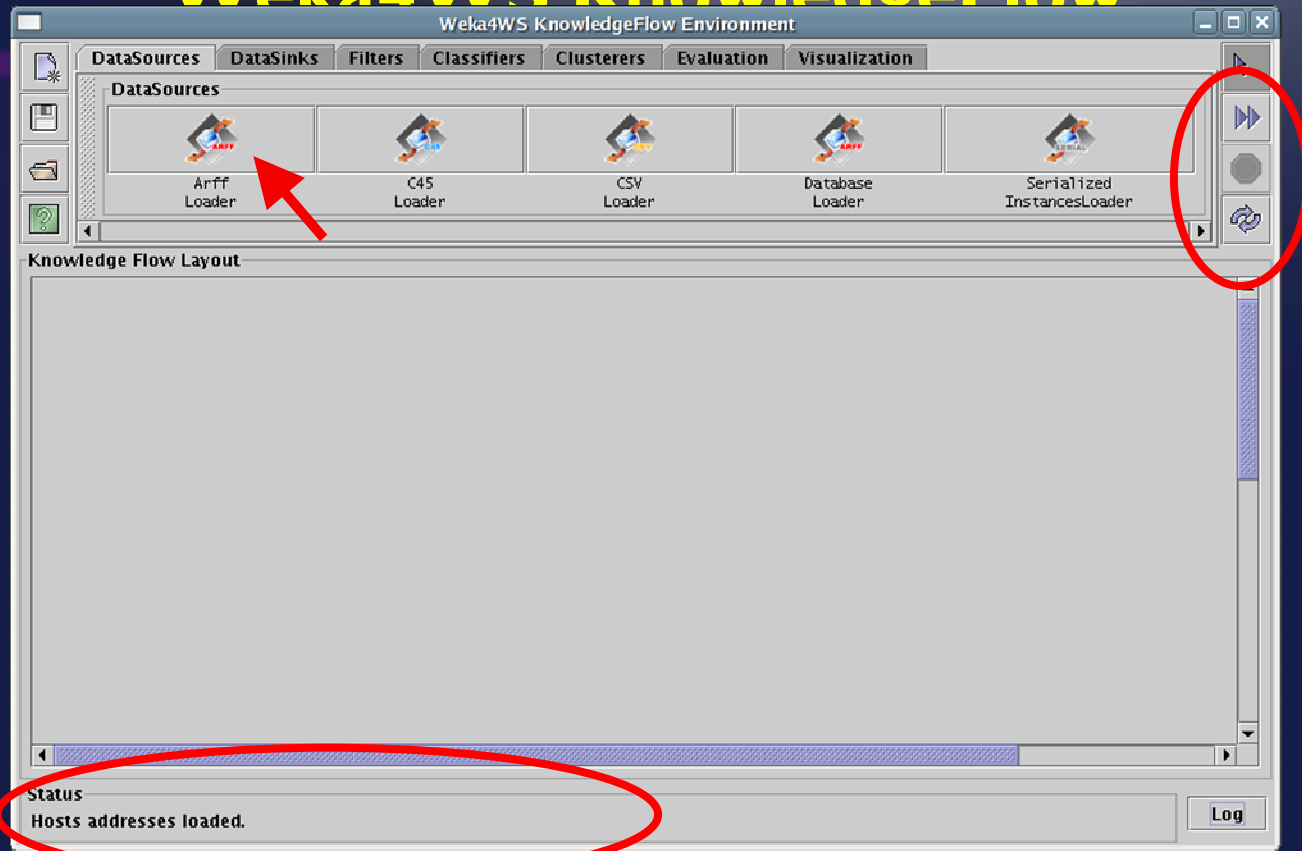
Thank you!

Credits:

Mario Cannataro
Eugenio Cesario
Antonio Congiusta
Marco Lackovic
Andrea Pugliese
Oreste Verta

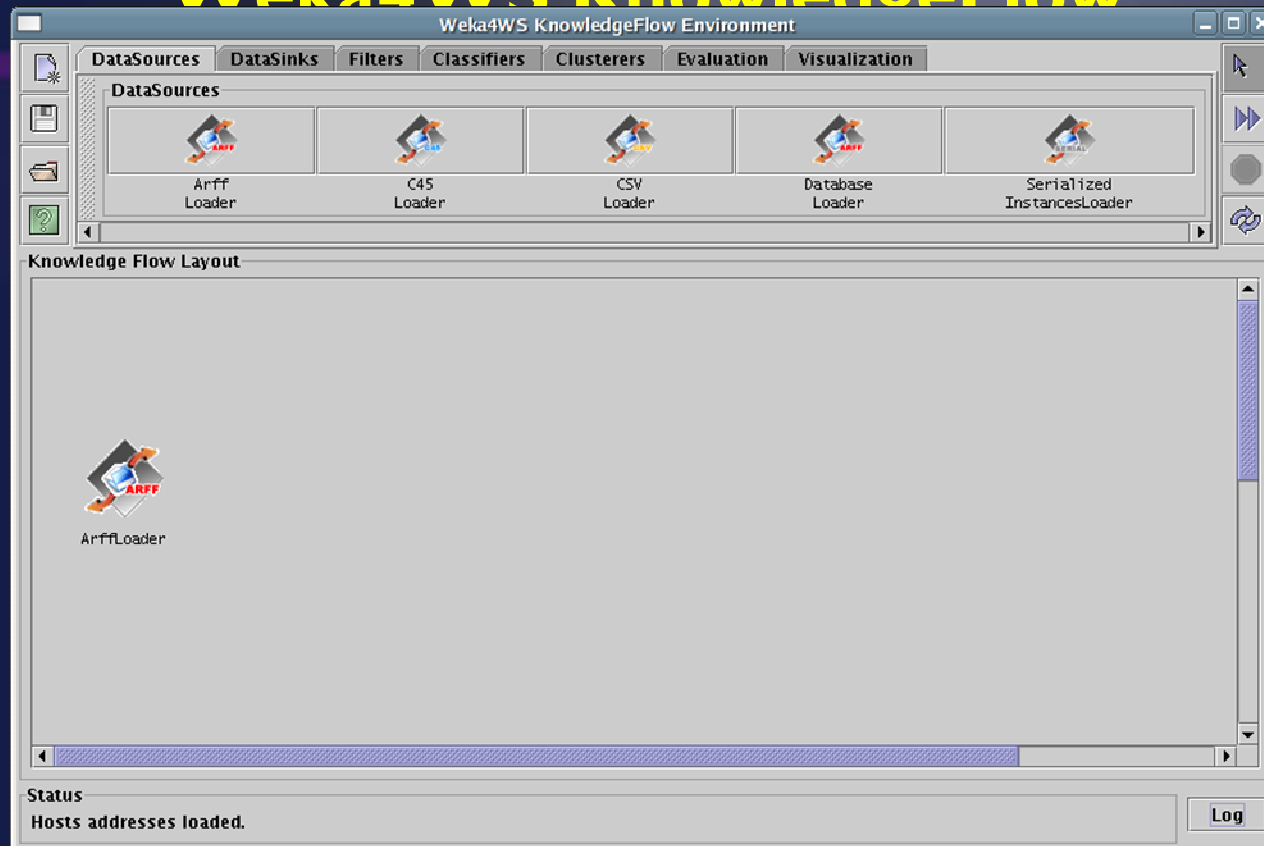


Weka4WS KnowledgeFlow



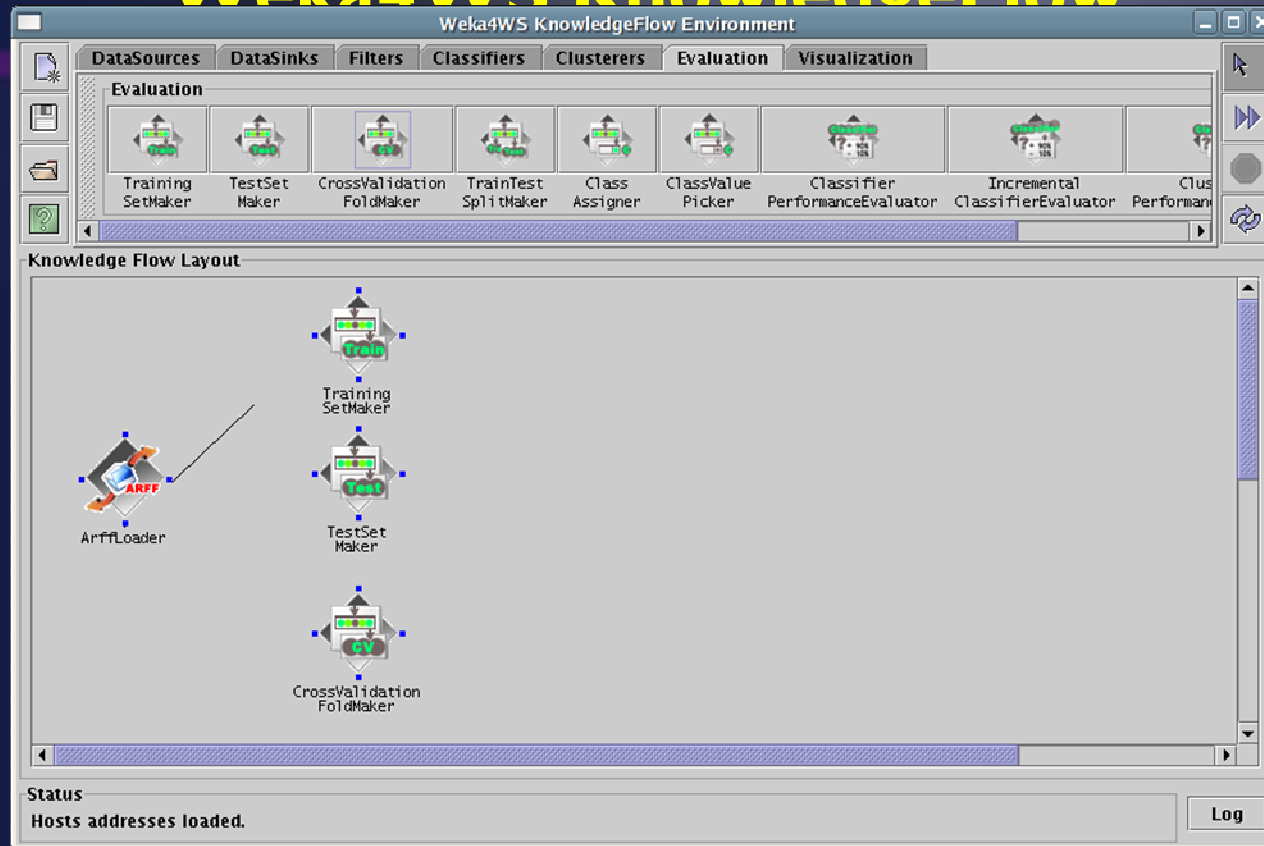
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



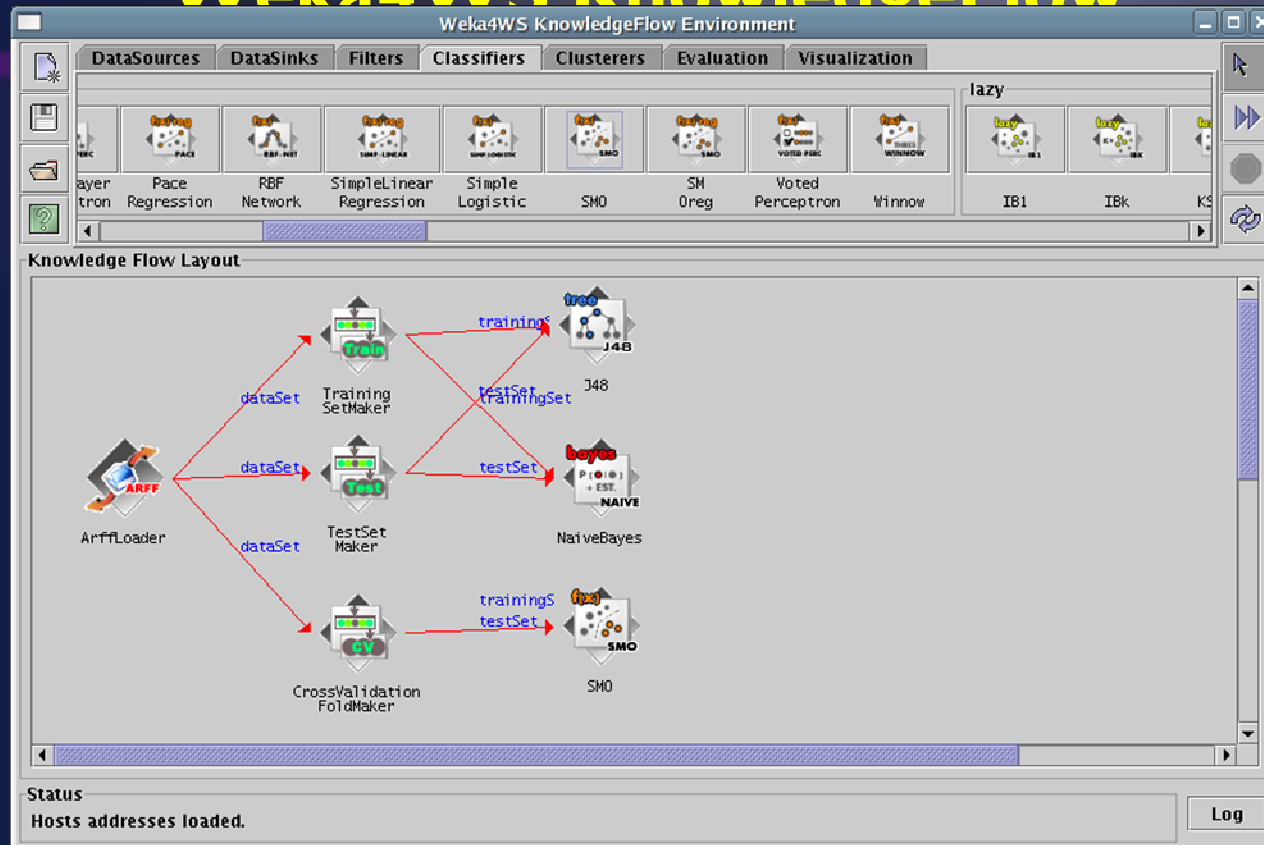
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



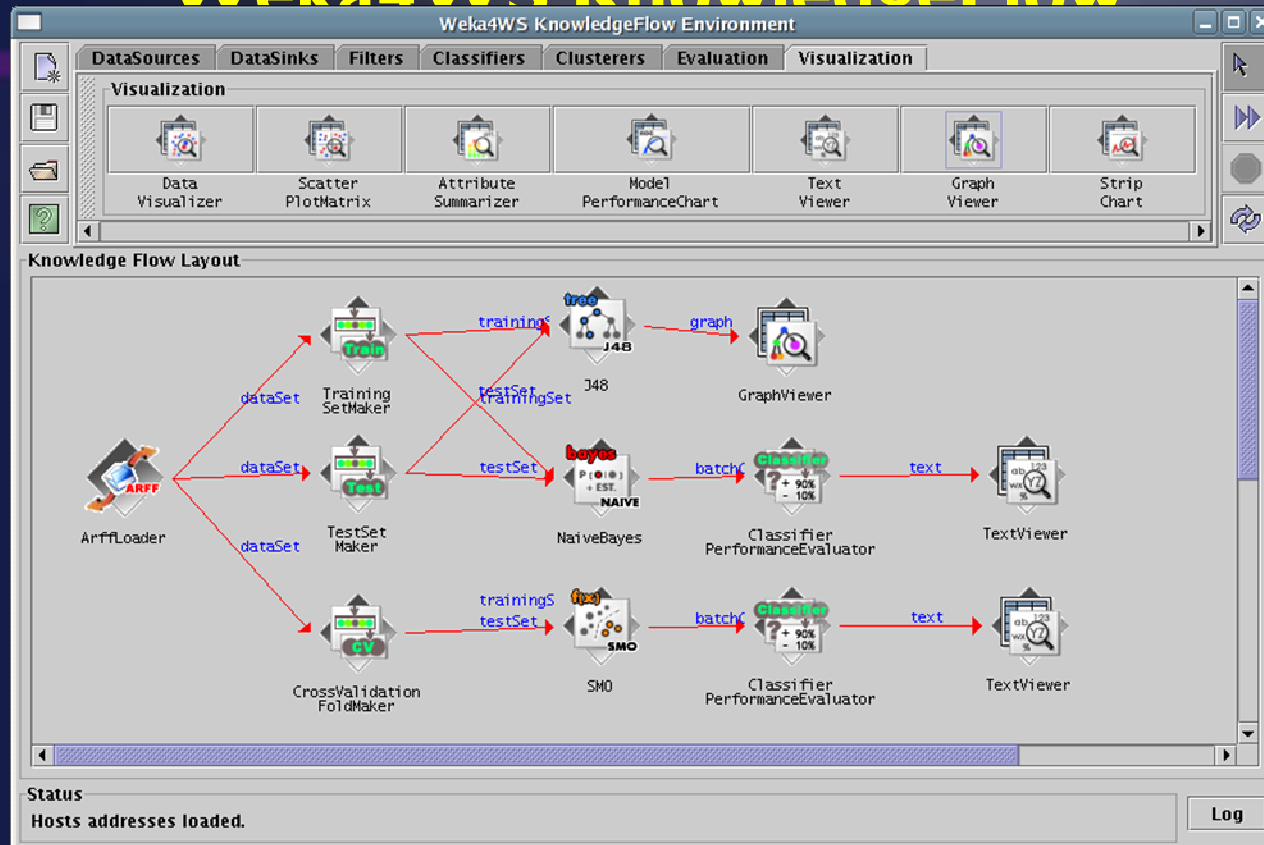
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



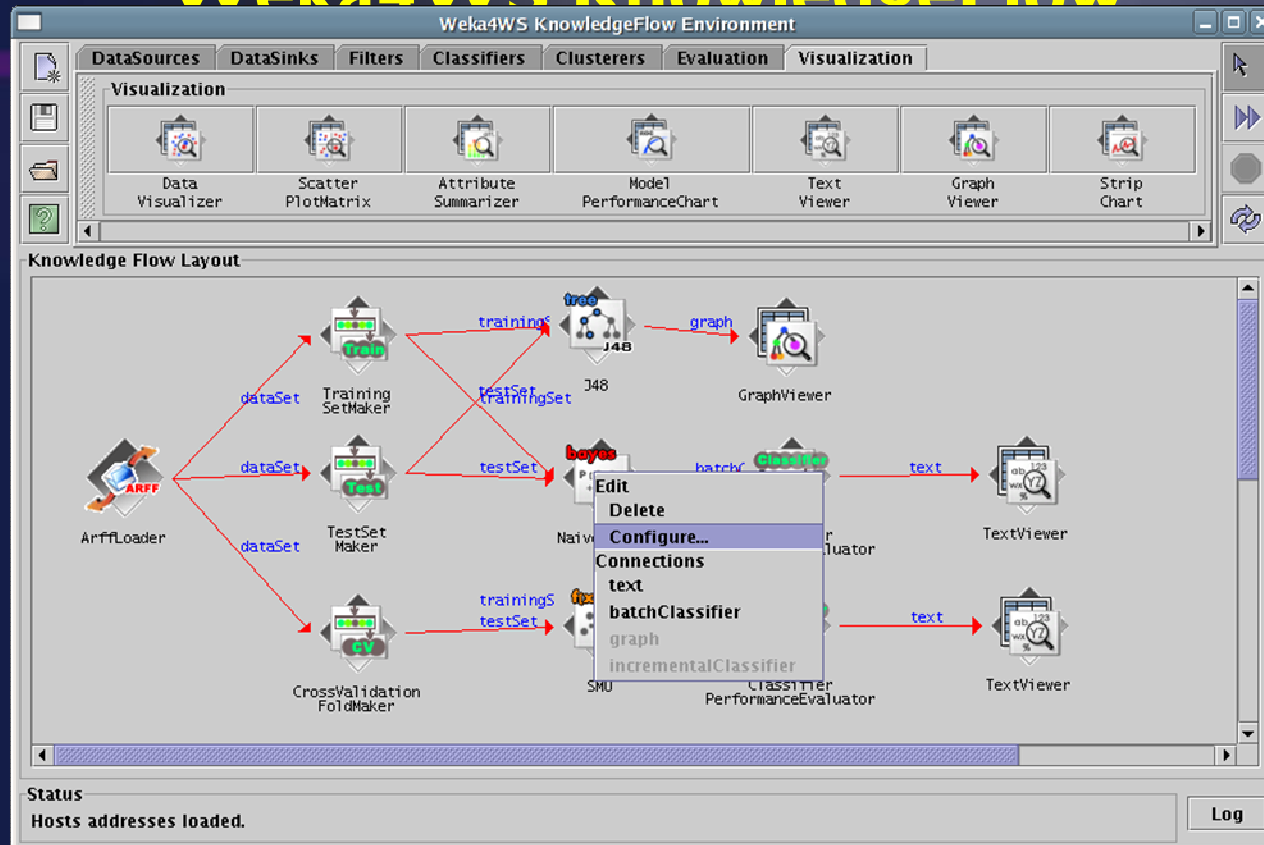
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



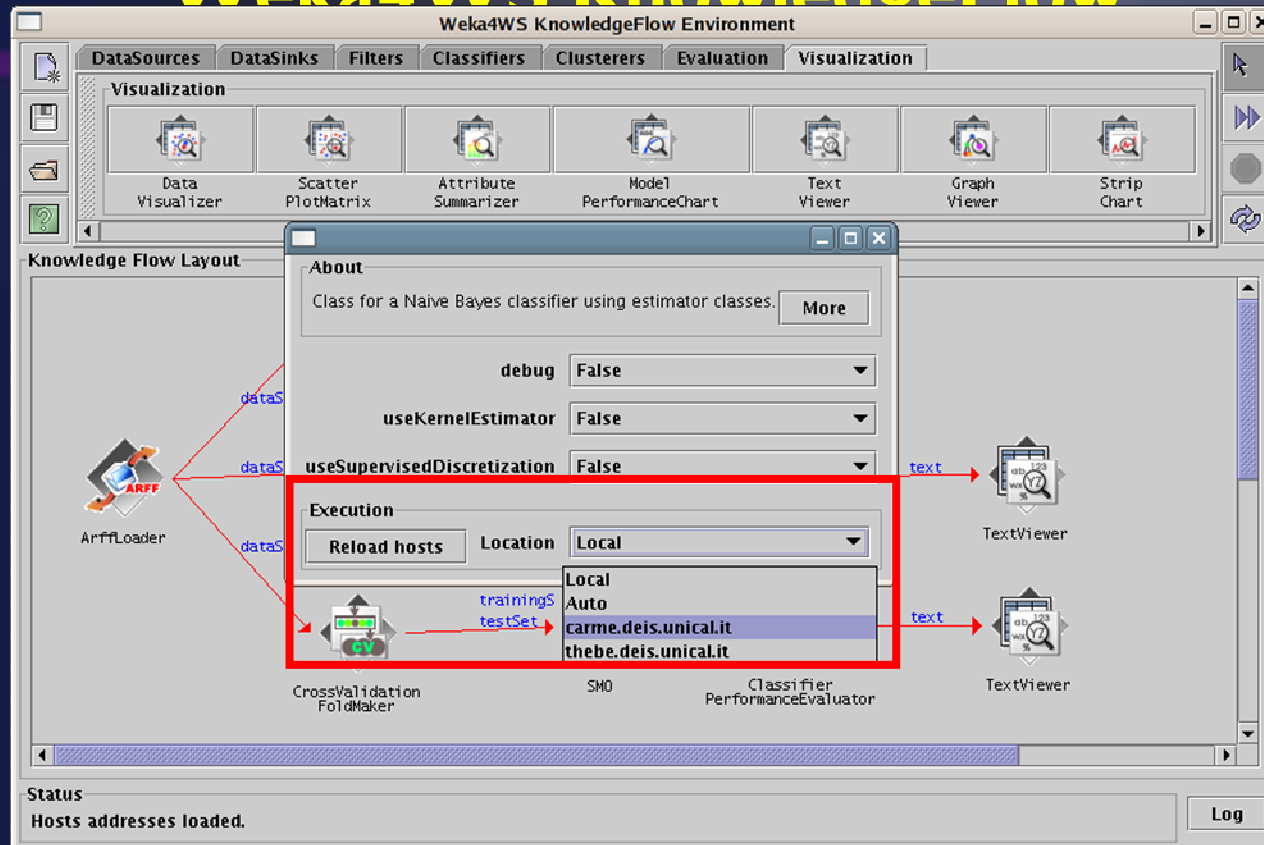
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



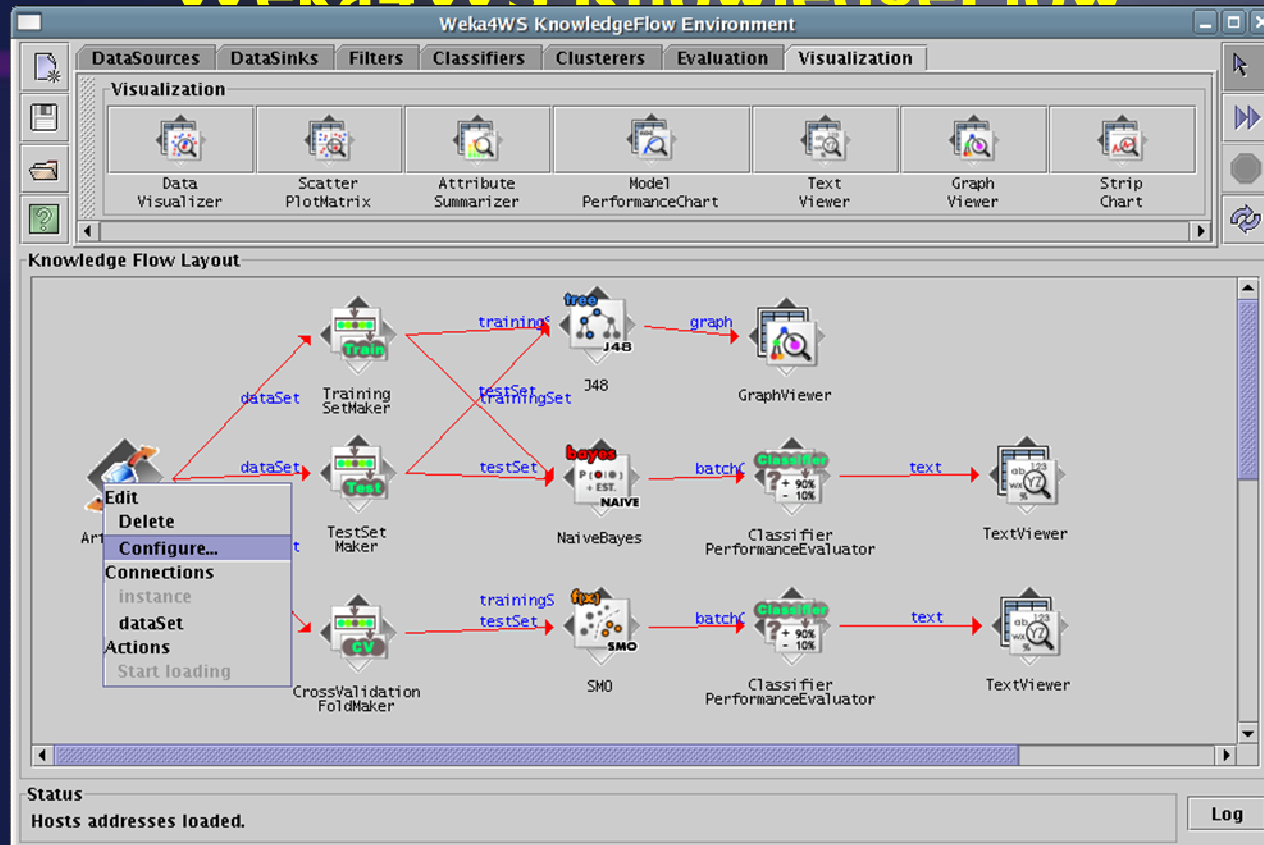
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



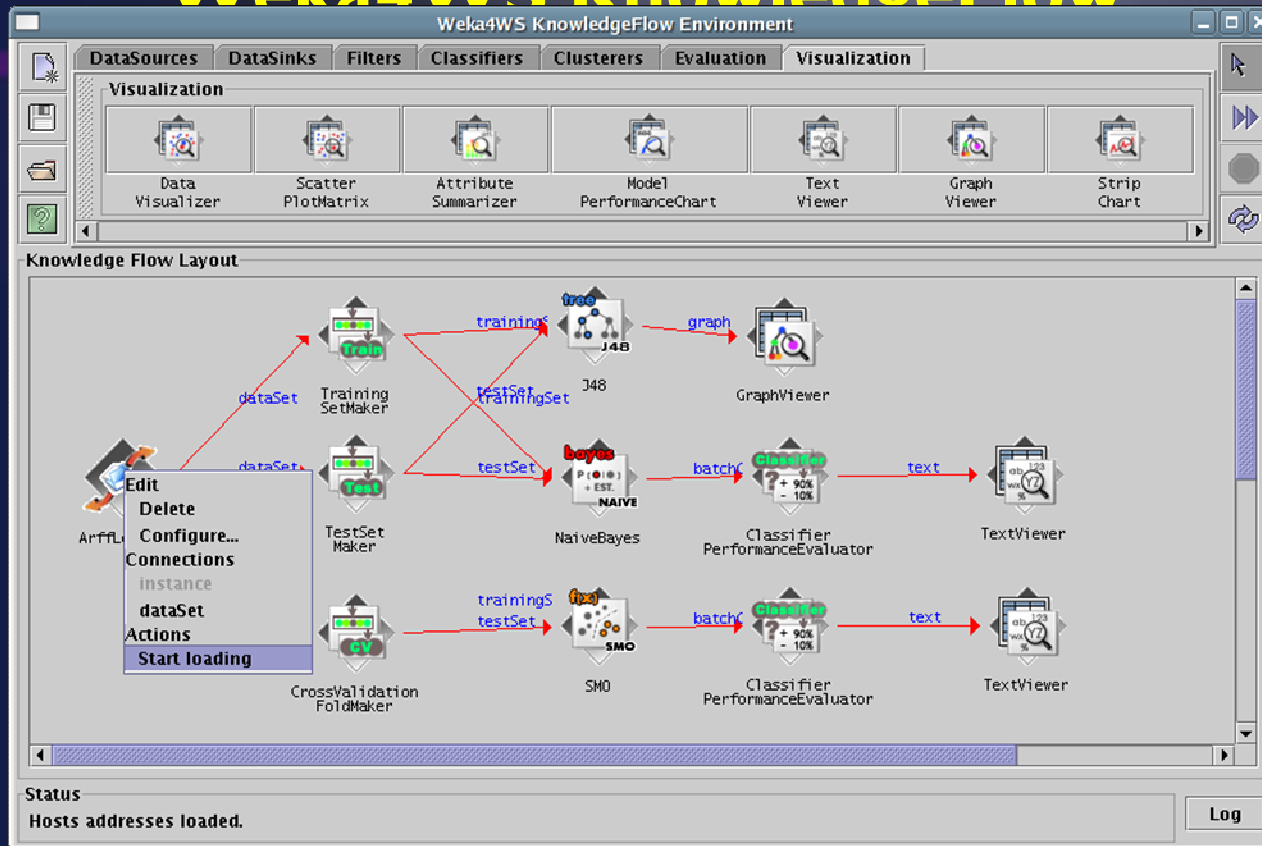
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



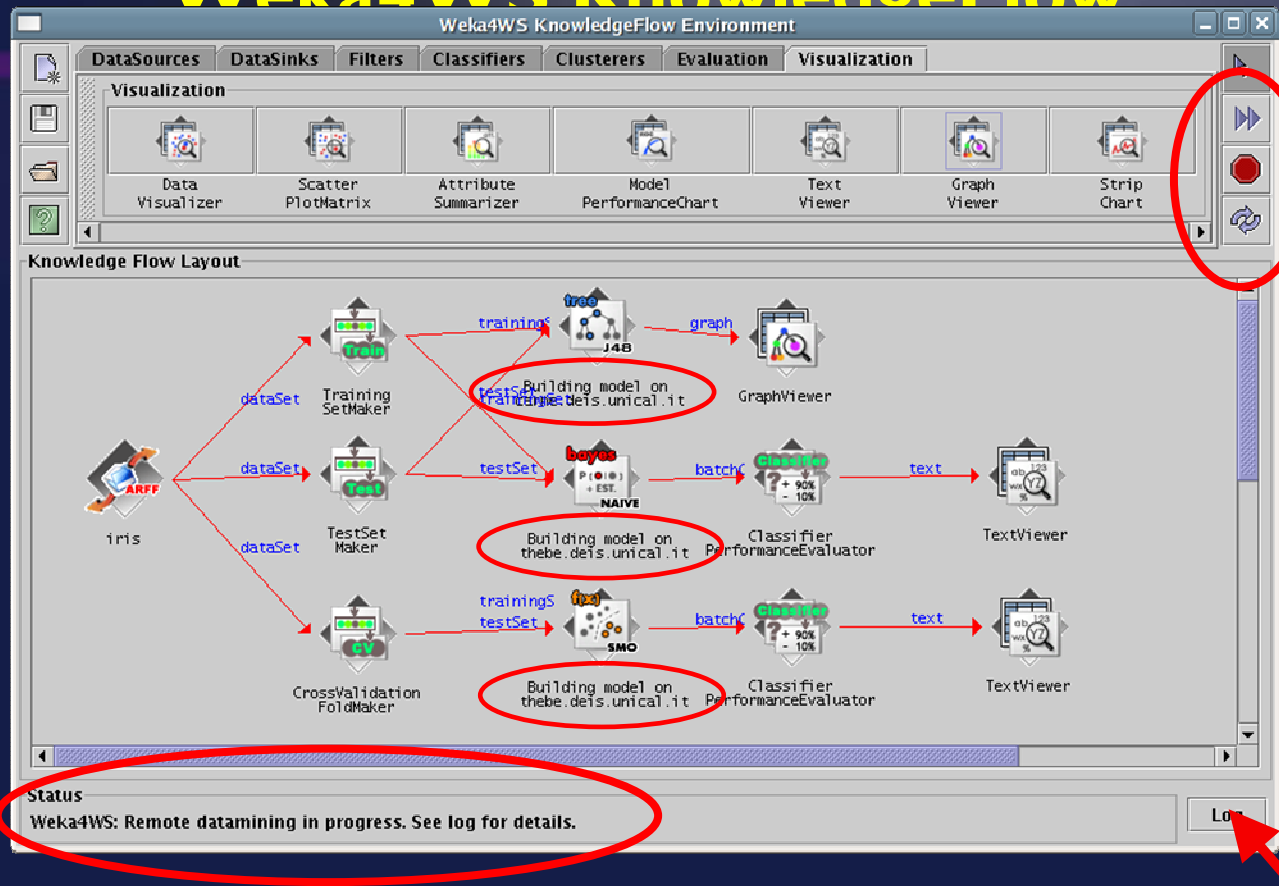
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



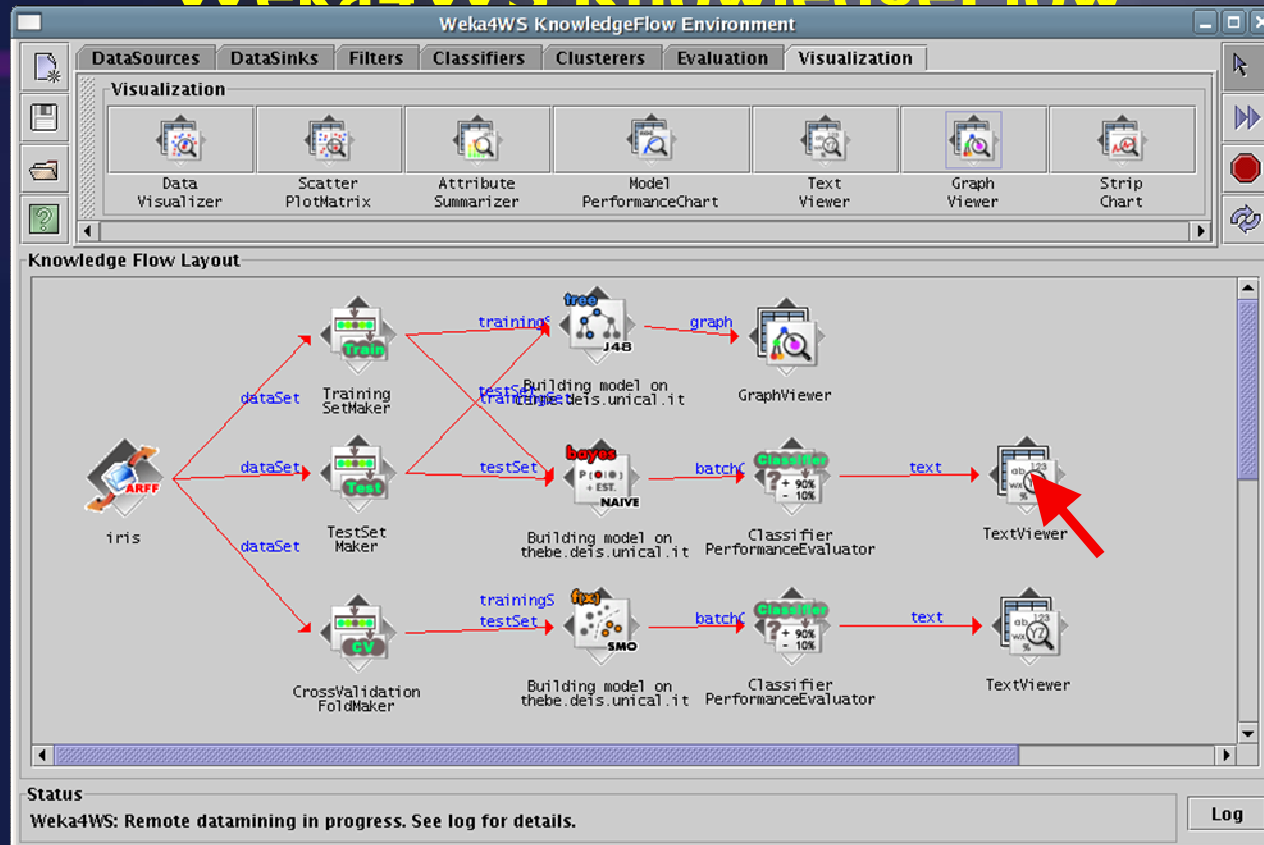
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow

The screenshot displays the Weka4WS KnowledgeFlow Environment. The main window is titled 'Weka4WS KnowledgeFlow Environment' and features a 'Knowledge Flow Layer' on the left with a workflow diagram. A 'Log' window is open in the center, showing a detailed log of the execution process. The log includes timestamps and messages from various components, such as 'Classifier', 'Resource creation', 'Notification subscription', 'Transferring dataset file', 'Remote data mining', and 'Results received'. The status bar at the bottom indicates 'Weka4WS: Remote datamining in progress. See log for details.'

- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow

The screenshot shows the Weka4WS KnowledgeFlow Environment. A 'Text Viewer' window displays the following evaluation results for a NaiveBayes classifier on the 'iris' dataset:

```

=== Evaluation result performed on thebe.deis.unical.it ===

Scheme: NaiveBayes
Relation: iris

=== Summary ===

Correctly Classified Instances      144          96 %
Incorrectly Classified Instances     6            4 %
Kappa statistic                     0.94
Mean absolute error                  0.0324
Root mean squared error              0.1495
Relative absolute error              7.2883 %
Root relative squared error          31.7089 %
Total Number of Instances           150

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
1         0         1           1         1           Iris-setosa
0.96     0.04     0.923      0.96     0.941      Iris-versicolor
0.92     0.02     0.958      0.92     0.939      Iris-virginica

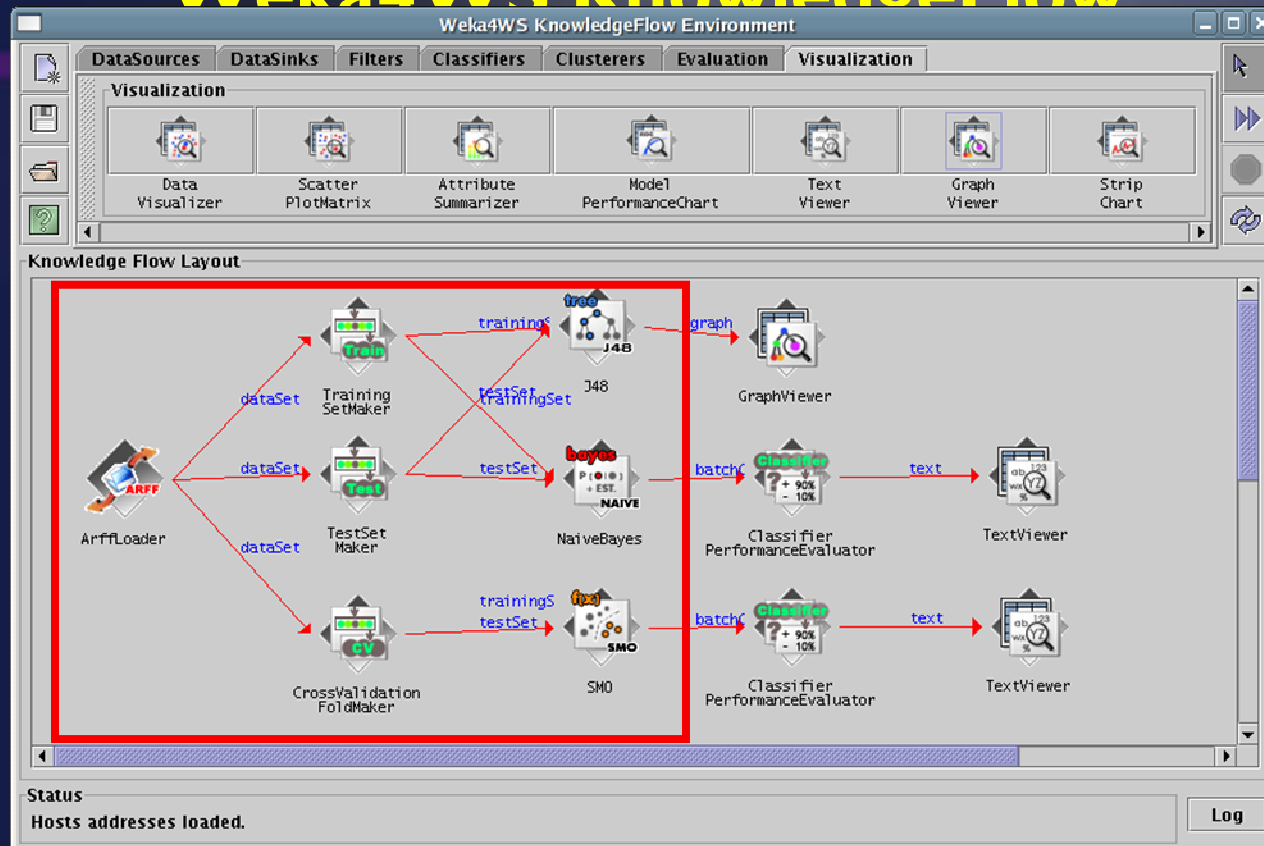
=== Confusion Matrix ===

 a  b  c  <-- classified as
50  0  0 | a = Iris-setosa
 0 48  2 | b = Iris-versicolor
 0  4 46 | c = Iris-virginica
    
```

At the bottom of the window, it says: "Weka4WS: Remote datamining in progress. See log for details." and there is a "Log" button.

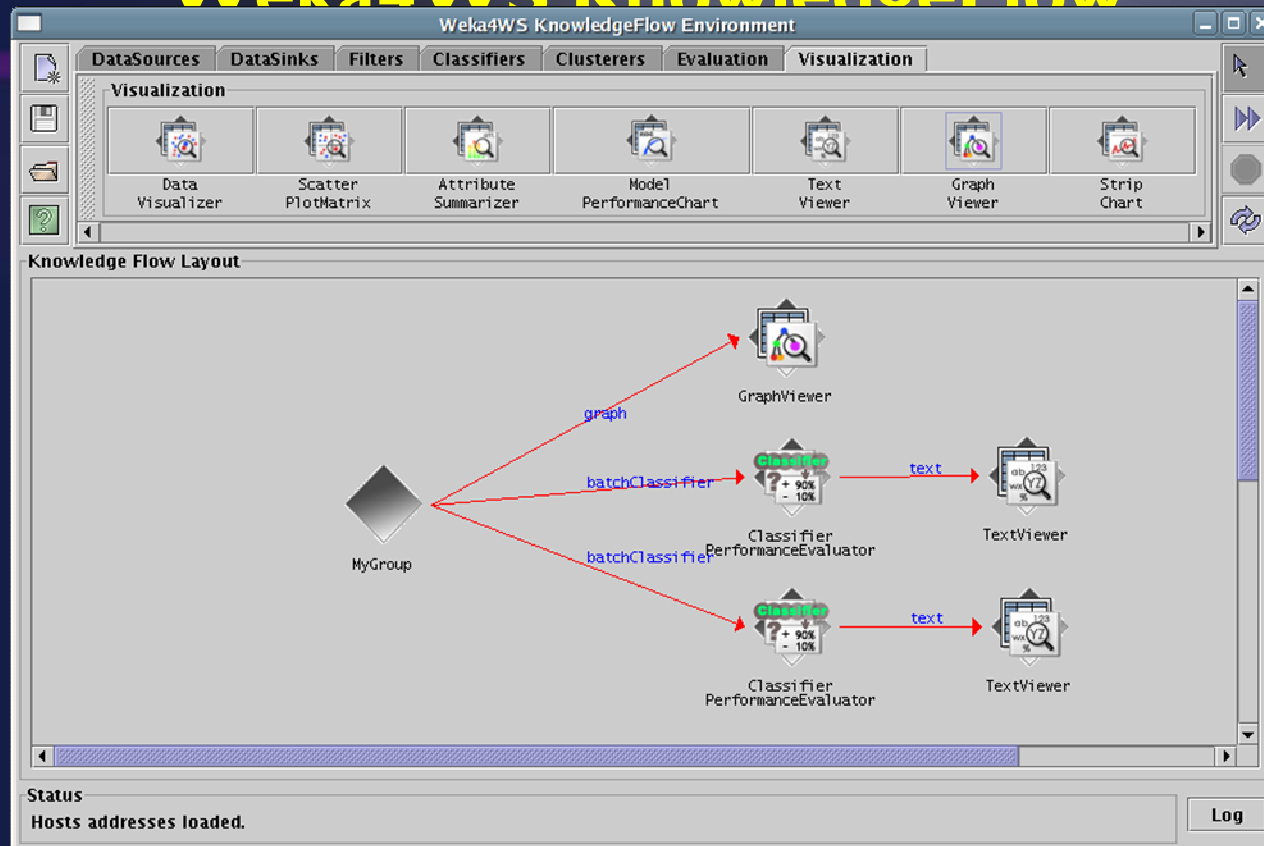
- A data mining workflow can be composed and run on several Grid nodes

Weka4WS KnowledgeFlow



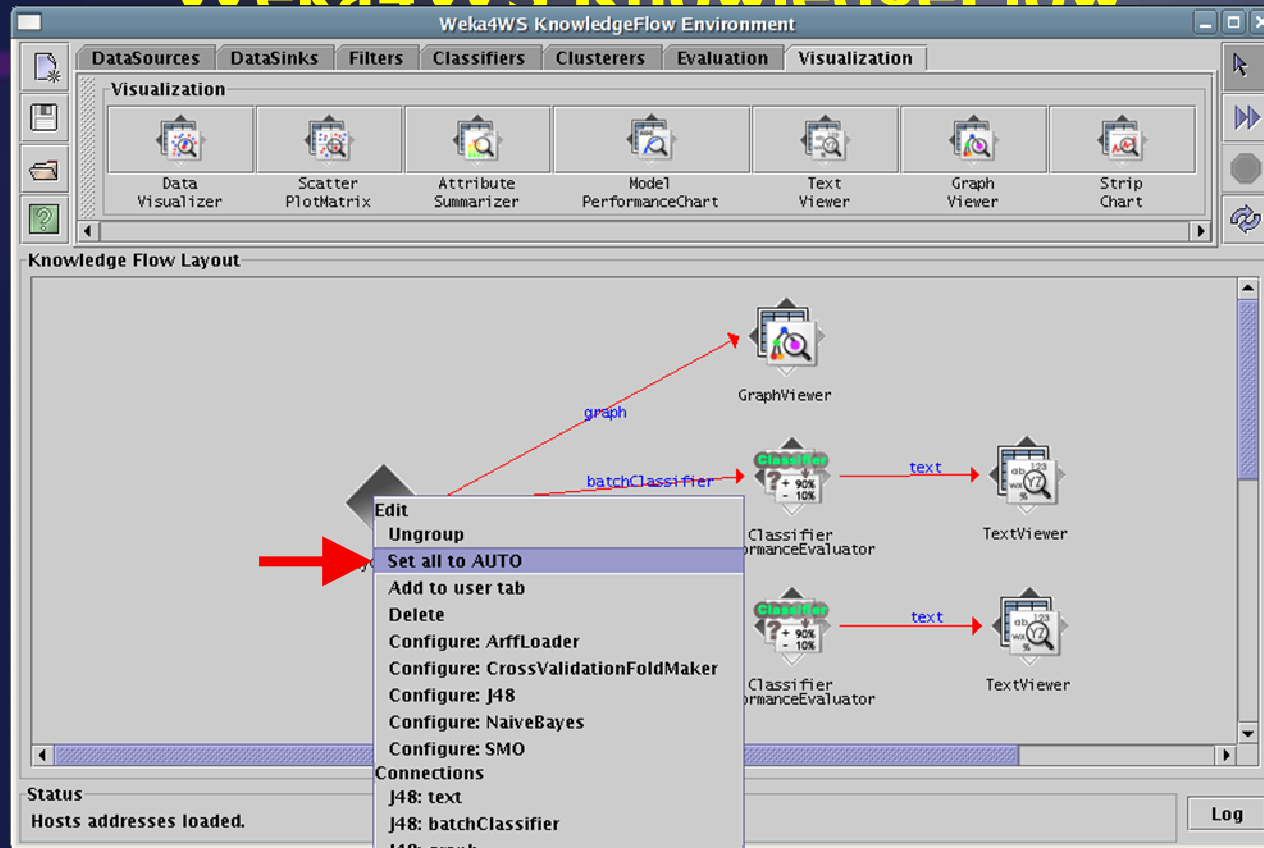
- Nodes in the KnowledgeFlow can be grouped and configured together

Weka4WS KnowledgeFlow



- Nodes in the KnowledgeFlow can be grouped and configured together

Weka4WS KnowledgeFlow



- Nodes in the KnowledgeFlow can be grouped and configured together