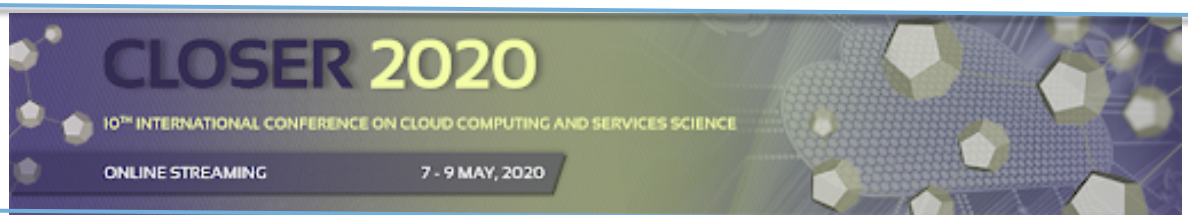
Cloud Computing for Enabling Big Data Analysis Services

Domenico Talia
DIMES
Università della Calabria & DtoK Lab

talia@dimes.unical.it

SCALab
Scalable Computing & Cloud Laboratory

DtoK LAB



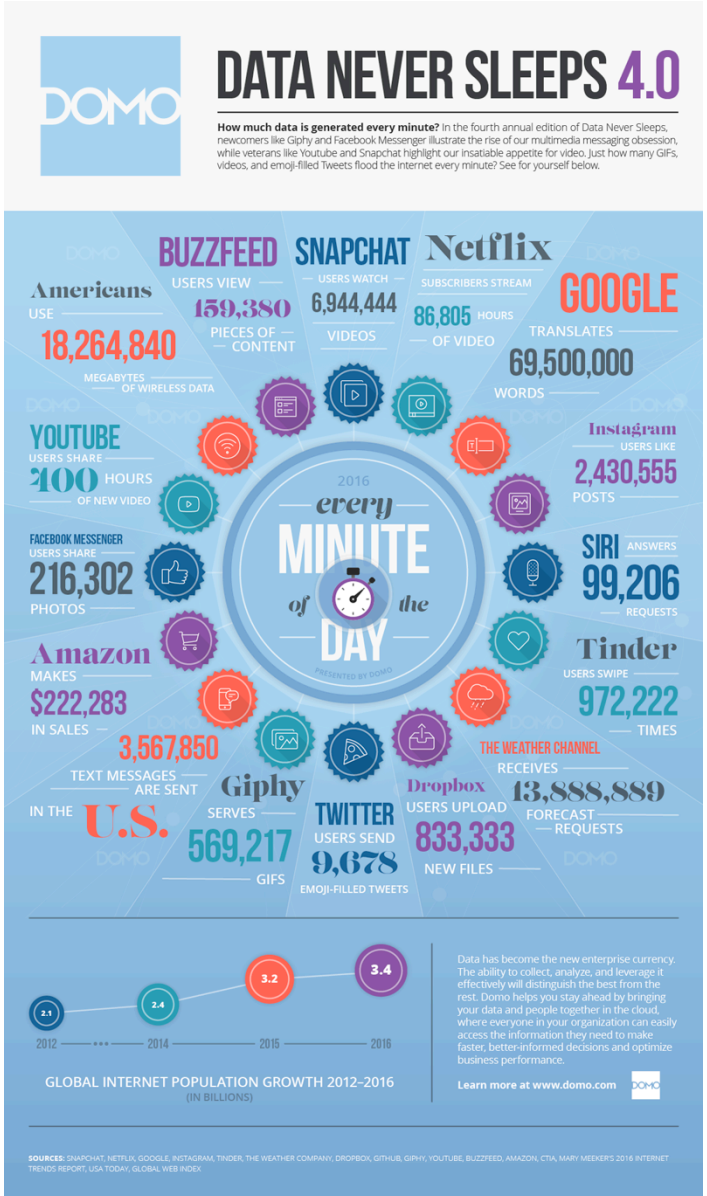
Goals of this talk

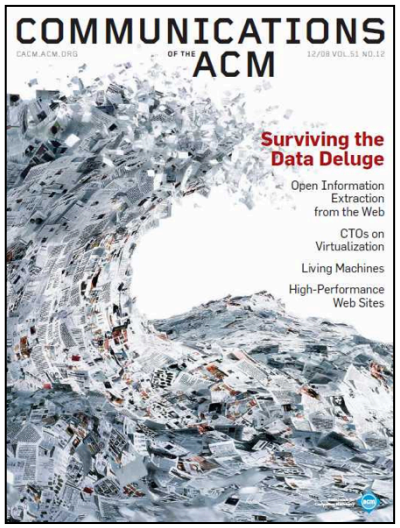
- Discuss **how to design** Cloud **services** for scalable execution of **data analysis workflows**.
  - Present a programming environment for data analysis: **Data Mining Cloud Framework (DMCF).**
  - Introduce a visual programming interface **VL4Cloud** and the script-based **JS4Cloud** language for implementing service-oriented workflows.
  - Introduce **Nubytics**: a high-level big data analytics framework.
- Outline some **open research topics**.

Outline

- Big problems and Big data
- Using Clouds for data mining and machine learning
- A collection of services for scalable data analysis
- Data mining workflows
- *Data Mining Cloud Framework (DMCF)*
- *JS4Cloud* for programming service-oriented workflows
- *Nubytics*: a high-level tool for data analytics
- Open Research Topics
- Final remarks

Big Data is huge and never sleeps





Big is quite a moving target? Some example

- Some data challenges examples we face today
  - Business data produced at a rate of hundreds of gigabits-per-second that must be stored, filtered and analyzed.
  - Millions of images per day that must be mined (analyzed) in parallel.
  - One billion of tweets/posts queried in real-time on an in-memory database.

Big Data needs scalable analysis

- Volume is only one dir
- Th
  is
- Sc
  ke
  Va

Big Data needs scalable analysis

Combination of

- **Big data analysis** and **machine learning** techniques with
- **scalable computing systems**

for

- an **effective strategy** to obtain new insights in a shorter period of time.

- **Cloud computing** helps!

... the potential interoperability and scaling convergence of HPC computing and data analysis is crucial to the future.

D.A. Reed & J. Dongarra, CACM 2015

Data Analysis as a Service

Data analysis as a service

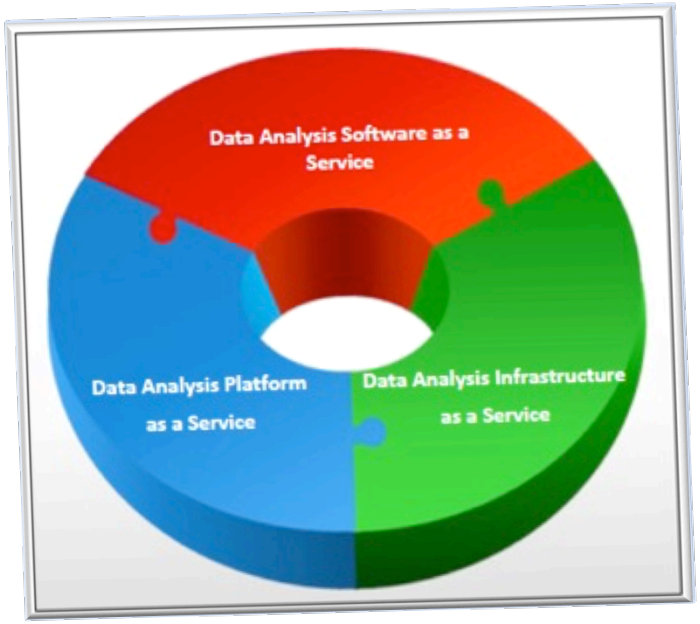
- For Big Data analysis on Clouds:
  - PaaS (*Platform as a Service*) can be ***an appropriate model to build frameworks*** for designing and executing scalable data mining and machine learning applications.
  - SaaS (*Software as a Service*) can be an ***appropriate model to*** allow end users ***to implement scalable data analysis applications***.
- Those two cloud service models can be effectively specialized for **delivering data analysis tools and applications as services**.

Services for distributed data mining (1)

- Data mining tasks and applications can be offered as high-level services.

- A new way to delivery data analysis software is

Data Analysis as a Service (*DAaaS*)



Services for distributed data mining (2)

- We can design services corresponding to

Data Mining Applications and KDD processes

This level includes the previous tasks and patterns composed in **multi-step workflows**.

Distributed Data Mining Patterns

This level implements, as services, patterns such as **collective learning**, **parallel classification** and **meta-learning** models.

Single Data Mining Tasks

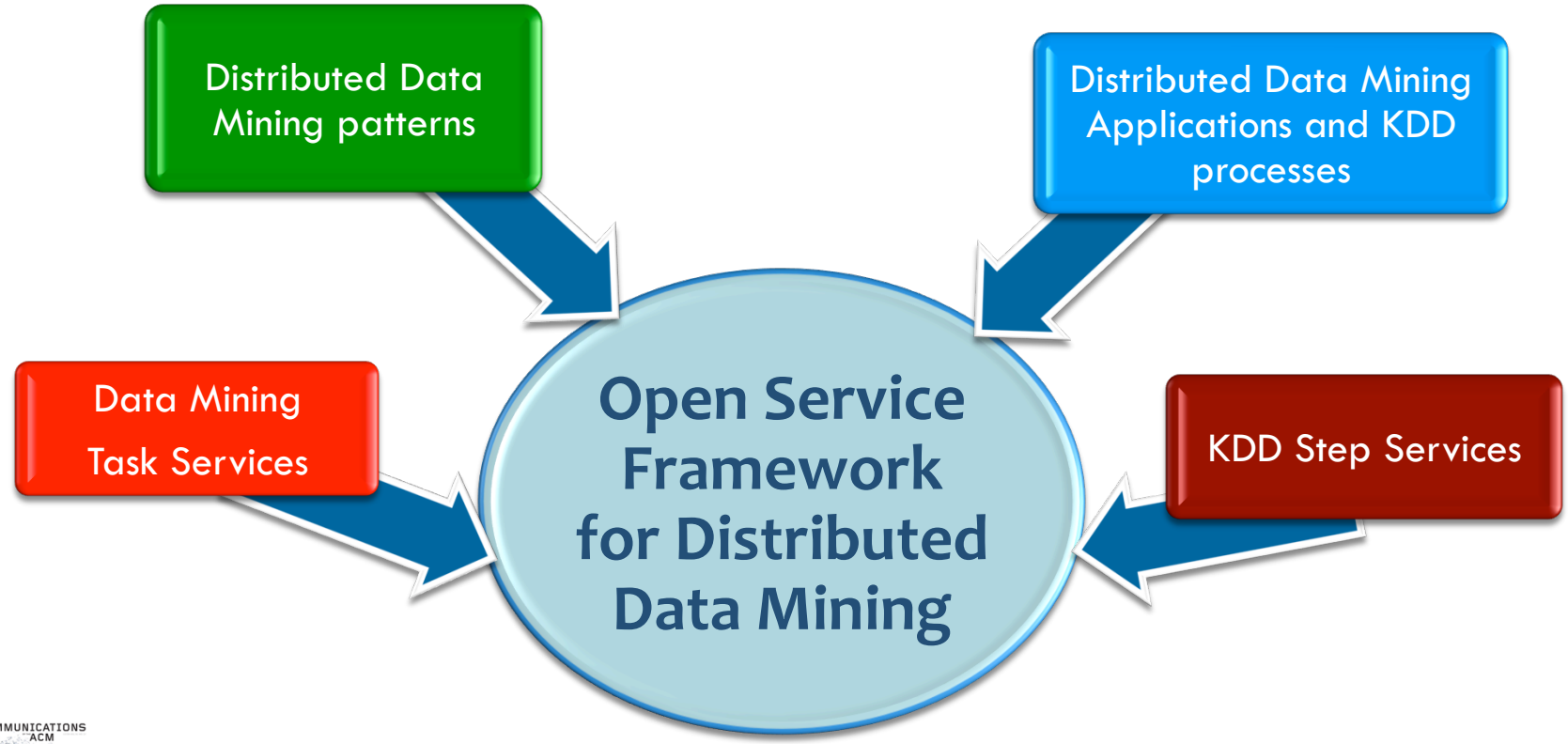
Here are included tasks such as **classification**, **clustering**, and **association rules discovery**.

Single KDD Steps

All steps that compose a KDD process such as **preprocessing**, **filtering**, and **visualization** are expressed as services.

Services for distributed data mining (3)

- This collection of data mining services implements:

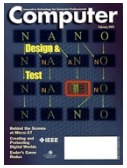


D. Talia, P. Trunfio, How Distributed Data Mining Tasks can Thrive as Knowledge Services. Communications of the ACM, vol. 53, n. 7, July 2010.

Services for distributed data mining (4)

- This approach supports not only service-based distributed data mining applications, but also
  - Data mining services for communities.
  - Distributed data analysis services on demand.
  - A sort of knowledge discovery eco-system made by a large numbers of decentralized data analysis services.

- Data analysis services on Cloud make **Big Data mining services accessible every time and everywhere, also** remotely and from small devices (microservices).



D. Talia, **Clouds for Scalable Big Data Analytics**, IEEE Computer, vol. 46, n. 5, May 2013.

Data analysis on Clouds: Systems

SYSTEMS:

- Spark, Mahout, HPC-ABDS, Sphere/sector, CloudFlows, Swift/T… & commercial systems.

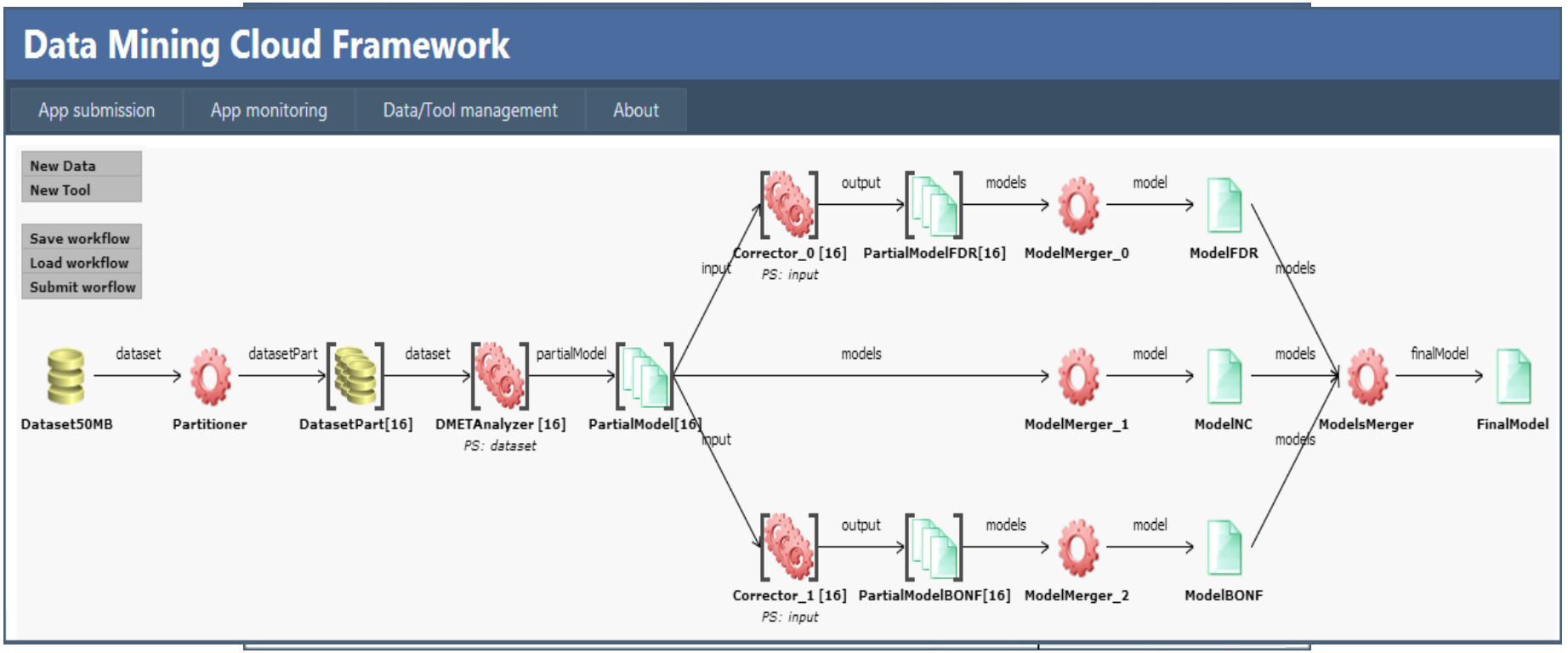
DMCF – the Data Mining Cloud Framework supporting Cloud-based data analysis apps as visual and script-based workflows.

Nubytics – an SaaS system for data analysis and machine learning on the Cloud.

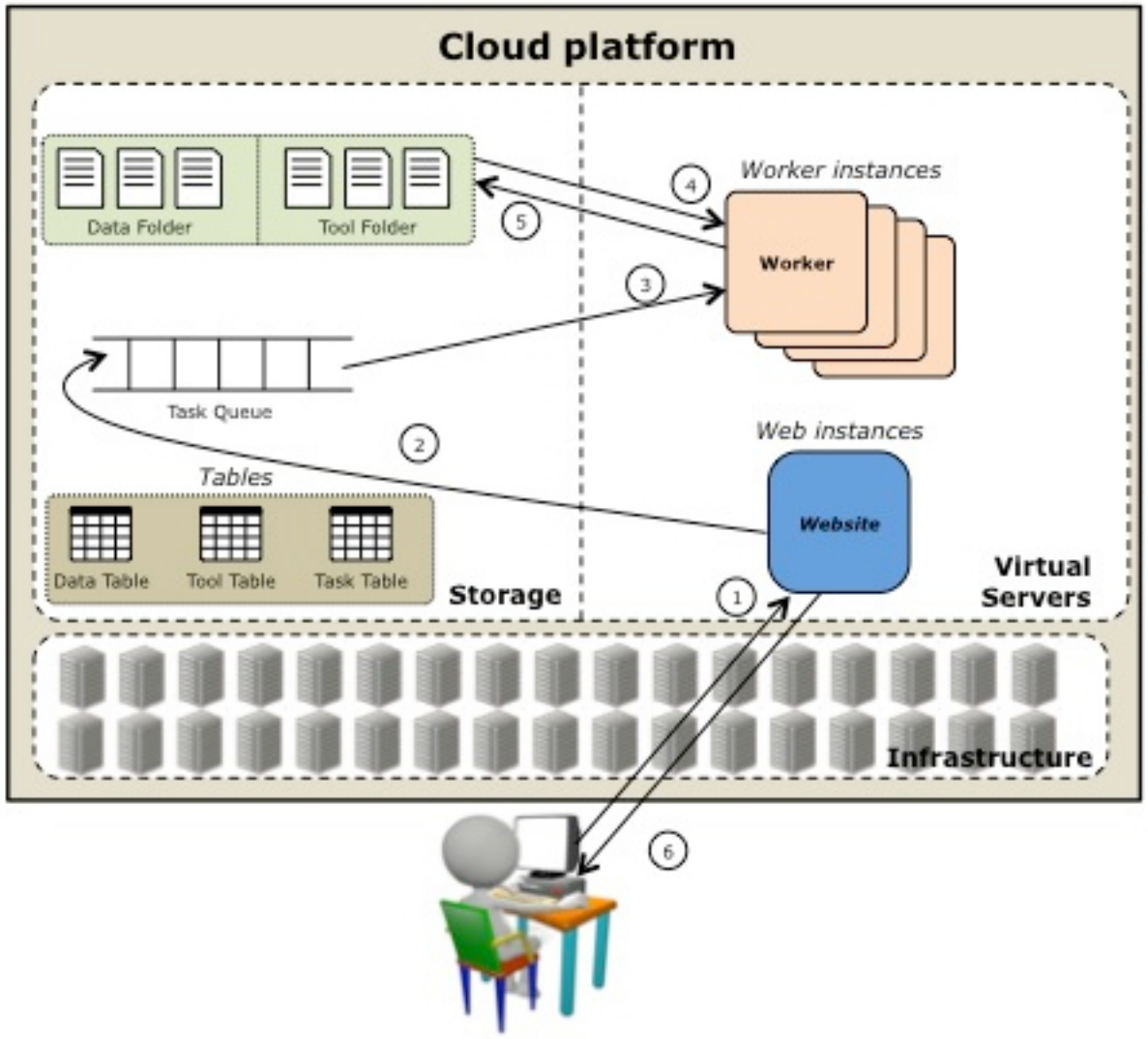
The Data Mining Cloud Framework

The Data Mining Cloud Framework

- The **Data Mining Cloud Framework** supports *workflow-based KDD applications*, expressed (visually and by a script language) as a graphs that link together data sources, data mining algorithms, and visualization tools.

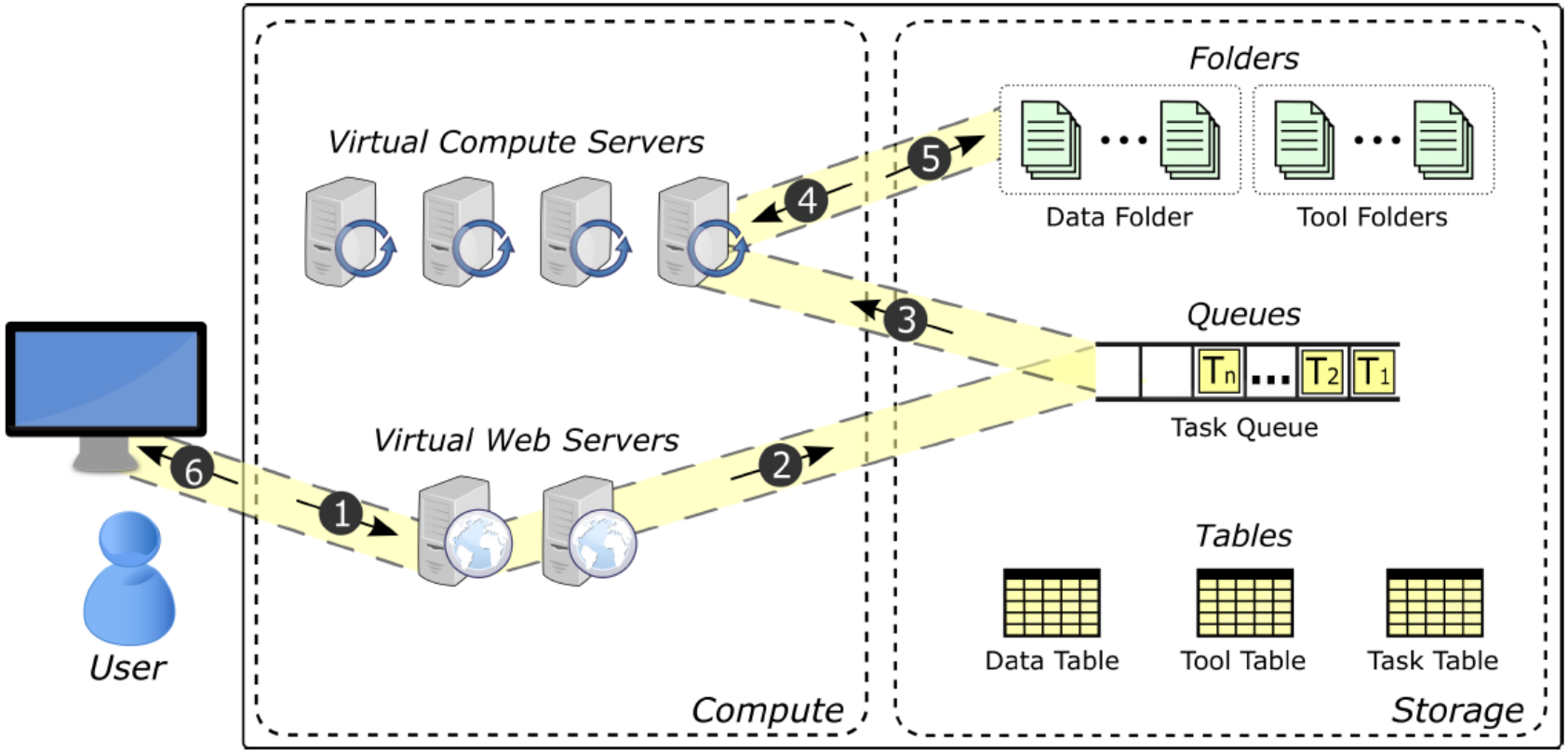


The Data Mining Cloud Framework: Execution



The Data Mining Cloud Framework: Architecture

- A **parallel computing** approach distributes the analysis on multiple virtual machines for **scalability**.



F. Marozzo, D. Talia, P. Trunfio, "Using Clouds for Scalable Knowledge Discovery Applications". Euro-Par Workshops, Lecture Notes in Computer Science, vol. 7640, pp. 220-227, August 2012.

Script-based workflows: JS4Cloud

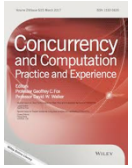
Script-based workflows

- We extended the visual interface **VL4Cloud** adding **JS4Cloud**, a ***script-based data analysis programming model*** as a more flexible programming interface.
- Script-based workflows are an effective alternative to graphical programming.
- A script language allows programmers to code complex applications more rapidly, in a **more concise** way and with **higher flexibility**.
- The idea is to offer a script-based data analysis language as an **additional and more flexible programming interface** to skilled users.

The JS4Cloud script language

- **JS4Cloud** (*JavaScript for Cloud*) is a language for programming data analysis workflows.

- Main benefits of JS4Cloud:
  - it is based on Javascript, a well known scripting language, so users **do not have to learn a new language** from scratch;
  - it implements a **data-driven task parallelism** that automatically spawns ready-to-run tasks to the available Cloud resources;
  - it exploits **implicit parallelism** so application workflows can be programmed in a totally sequential way (***no user duties for work partitioning, synchronization and communication***).



F. Marozzo, D. Talia, P. Trunfio, JS4Cloud: Script-based Workflow Programming for Scalable Data Analysis on Cloud Platforms. Concurrency and Computation: Practice and Experience, vol. 27, n. 17, pp. 5214--5237, Wiley InterScience , December 2015.

JS4Cloud functions

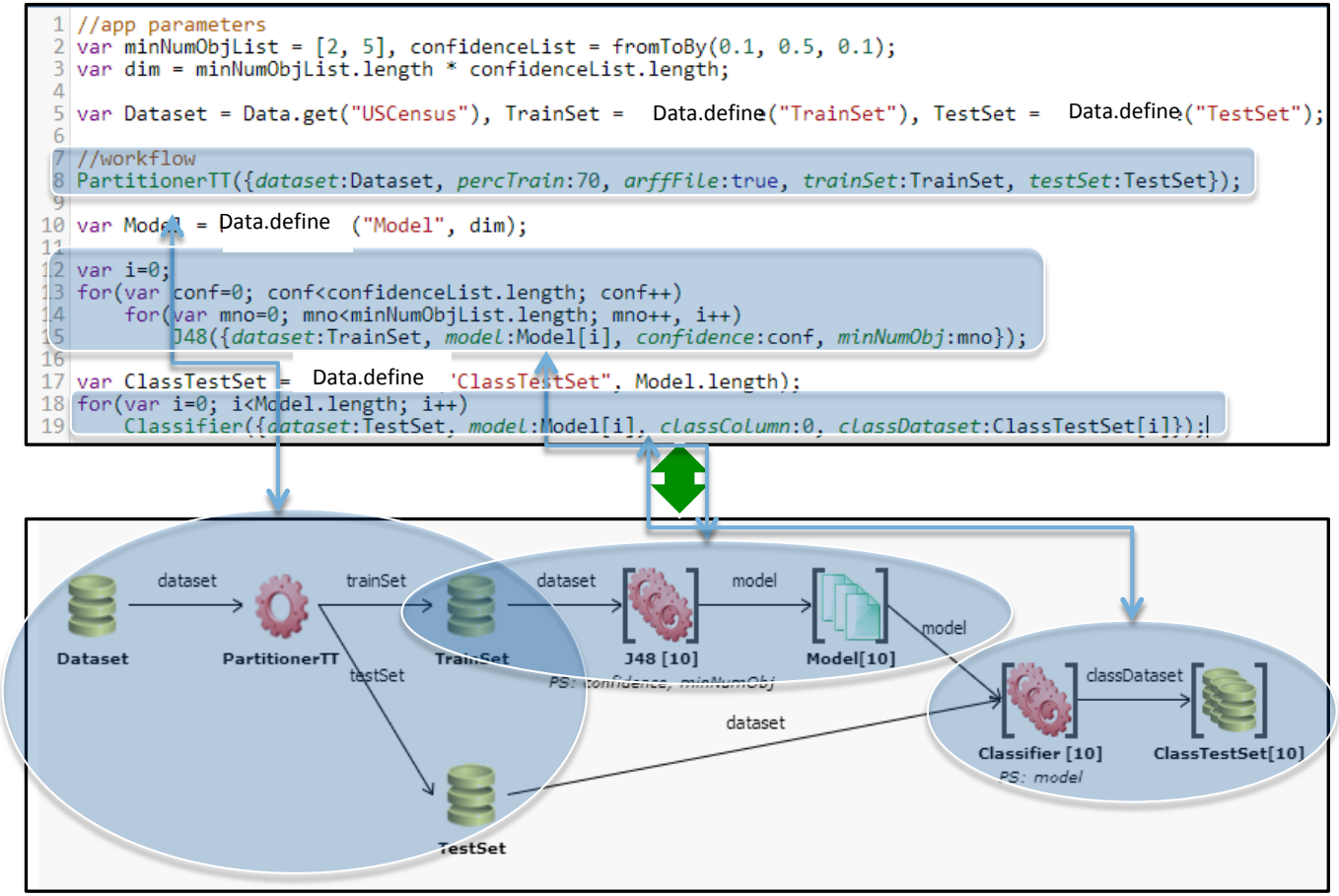
JS4Cloud provides three mechanisms, implemented by the set of functions:

- **`Data.get`**, for accessing one or a collection of datasets stored in the Cloud;
- **`Data.define`**, for defining new data elements that will be created at runtime as a result of a tool execution;
- **`Tool`**, to invoke the execution of a software tool available in the Cloud as a service.

Functionality	*Function*	*Description*
Data Access	Data.get(<*dataName*>);	Returns a reference to the data element with the provided name.
	Data.get(new RegExp(<*regular expression*>));	Returns an array of references to the data elements whose name match the regular expression.
Data Definition	Data.define(<*dataName*>);	Defines a new data element that will be created at runtime.
	Data.define(<*arrayName*>,<*dim*>);	Define an array of data elements.
	Data.define(<*arrayName*>,[<dim_1>,...,<dim_n>]);	Define a multi-dimensional array of data elements.
Tool Execution	<*toolName*>(<par_1>:<val_1>,...,<par_n>:<val_n>);	Invokes an existing tool with associated parameter values.

Script-based applications

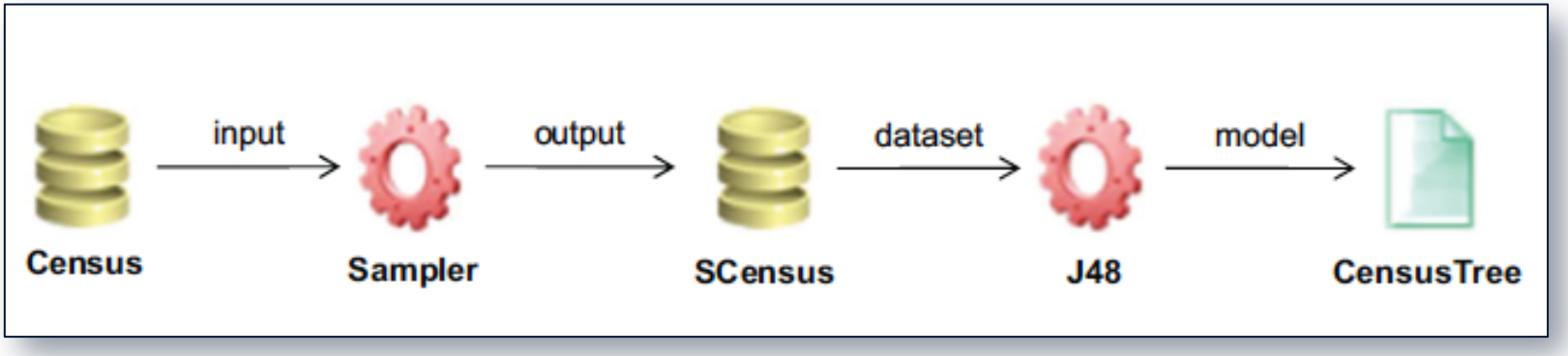
- Code-defined workflows are fully equivalent to graphically-defined ones:



JS4Cloud patterns

Pipeline

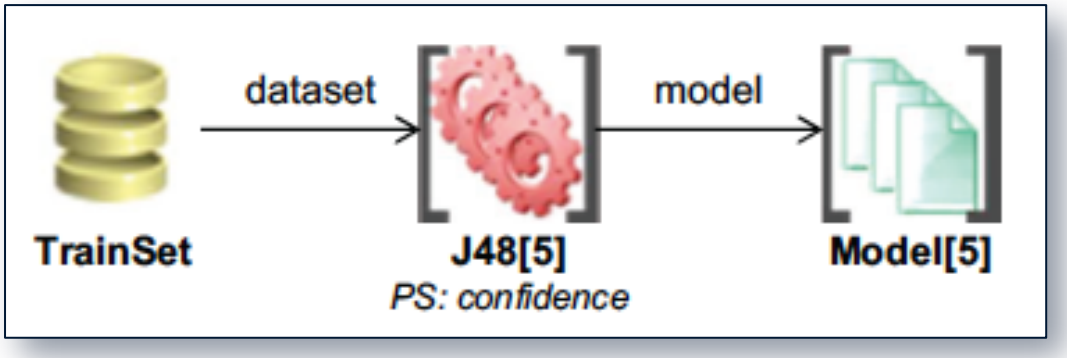
```
var DRef  = Data.get("Census");
var SDRef = Data.define("SCensus");
Sampler({input:DRef, percent:0.25, output:SDRef});
var MRef  = Data.define("CensusTree");
J48({dataset:SDRef, confidence:0.1, model:MRef});
```



JS4Cloud patterns

Parameter sweeping

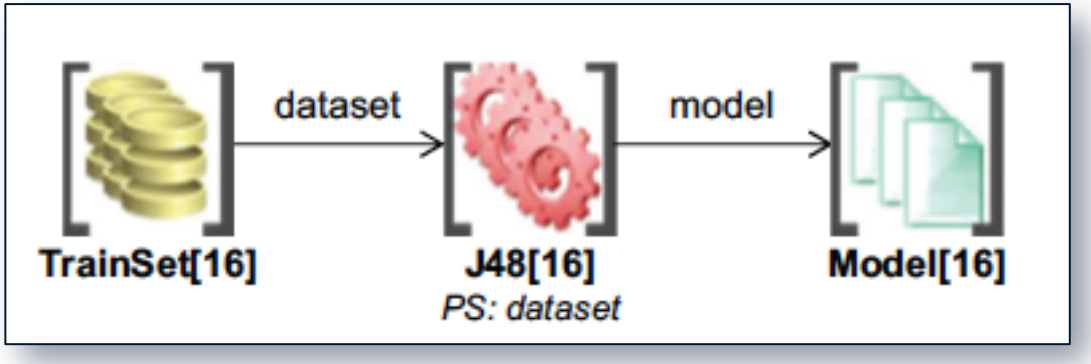
```
var TRef = Data.get("TrainSet");
var nMod = 5;
var MRef = Data.define("Model", nMod);
var min = 0.1;
var max = 0.5;
for(var i=0; i<nMod; i++)
   J48({dataset:TRef, model:MRef[i],
        confidence:(min+i*(max-min)/(nMod-1))});
```



JS4Cloud patterns

Input sweeping

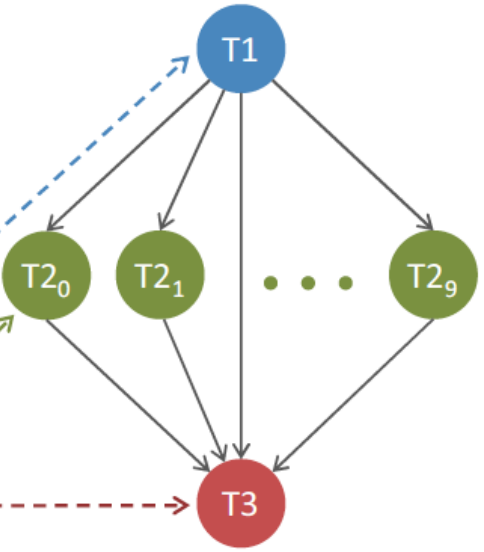
```
var nMod = 16;
var MRef = Data.define("Model", nMod);
for(var i=0; i<nMod; i++)
   J48({dataset:TsRef[i], model:MRef[i],
        confidence:0.1});
```

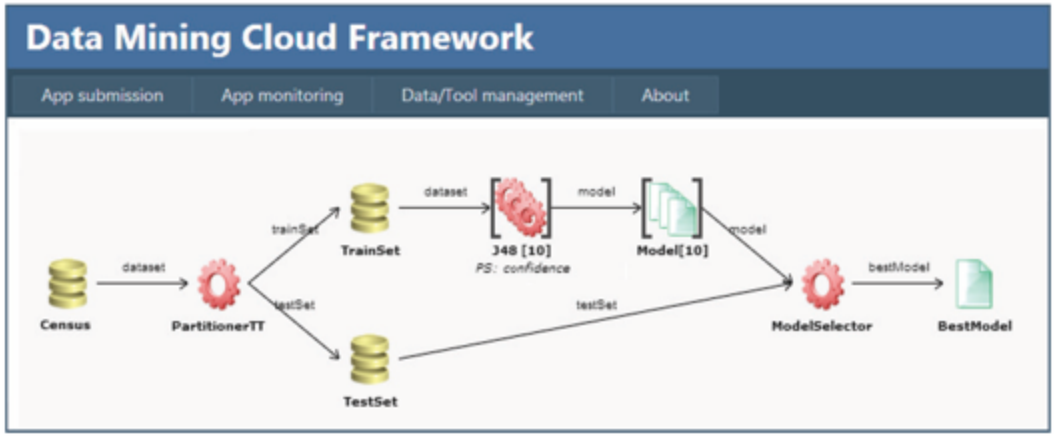


Parallelism exploitation

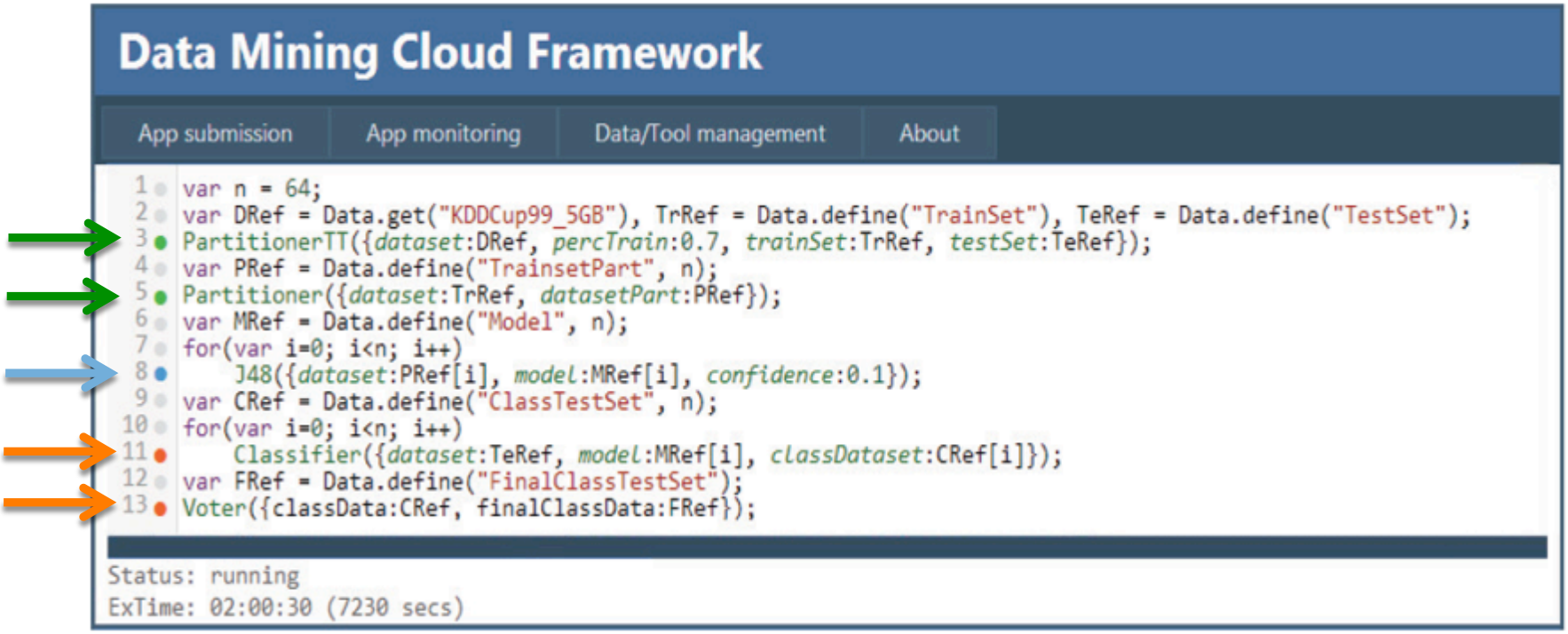
```
var DRef = Data.get("Census");
var TrRef = Data.define("TrainSet");
var TeRef = Data.define("TestSet");
var min = 0.1, max = 0.5; nMod = 10;
var MRef = Data.define("Model", nMod);
var BRef = Data.define("BestModel");

PartitionerTT({dataset:DRef, percTrain:0.70, trainSet:TrRef, testSet:TeRef});
for(int i=0; i<nMod; i++)
   J48({dataset:TrRef, model:Model[i], confidence:(min+i*(max-min)/(nMod-1))});
ModelSelector({testSet:TeRef, model:Model, bestModel:BRef});
```





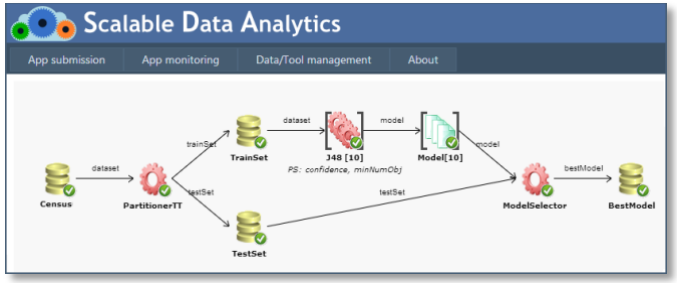
Monitoring interface

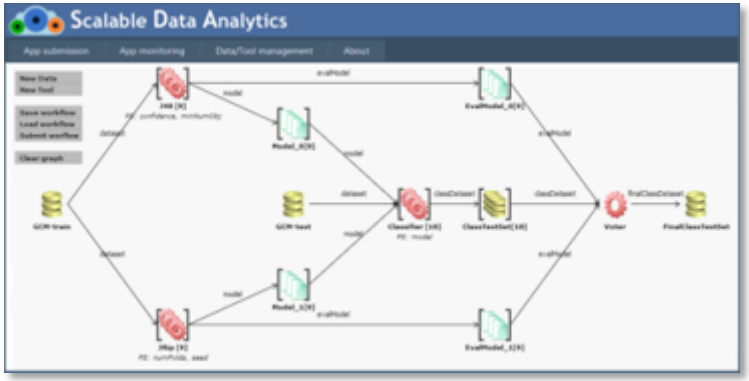


- A snapshot of the application during its execution monitored through the programming interface.

Example applications (1)

Finance: Prediction of personal income based on census data



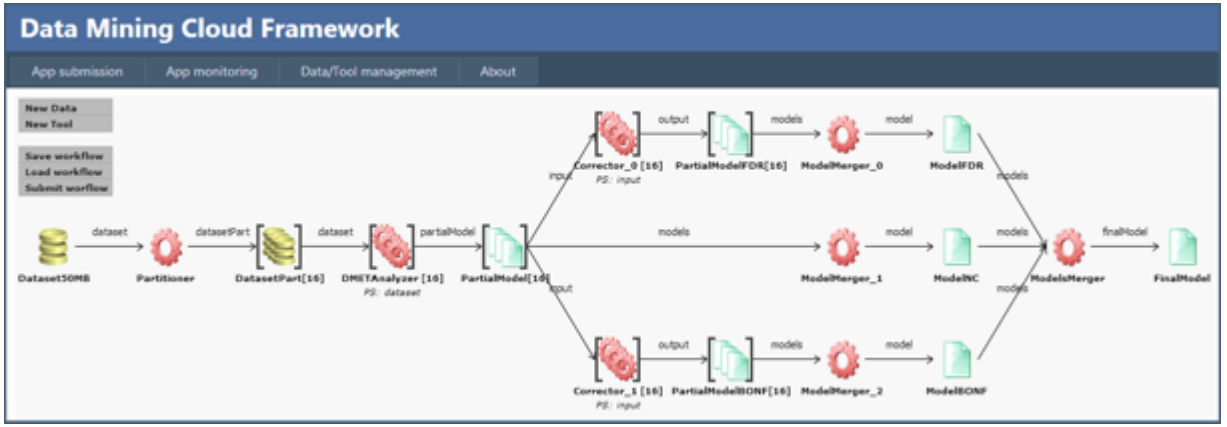


E-Health: Disease classification based on gene analysis

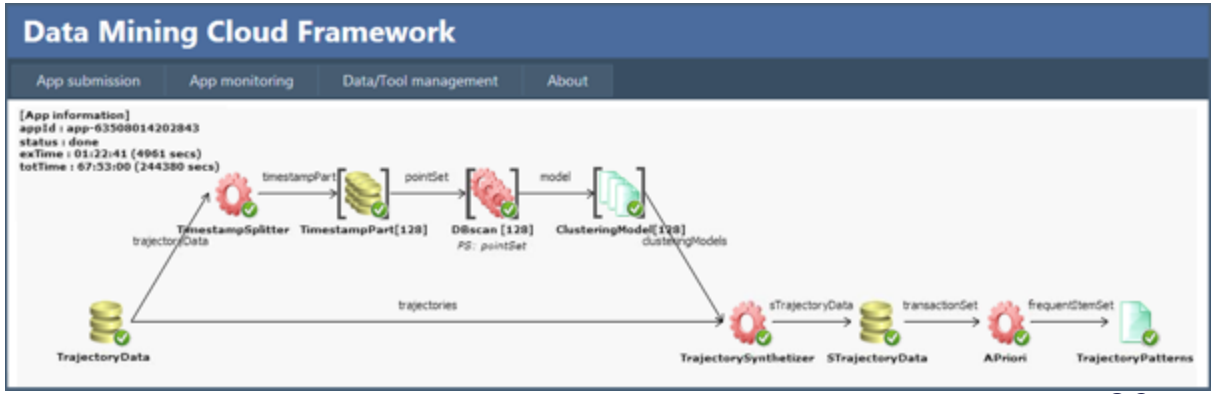
Networks: Discovery of network attacks from log analysis.

Example applications (2)

Biosciences: drug metabolism associations in pharmacogenomics.



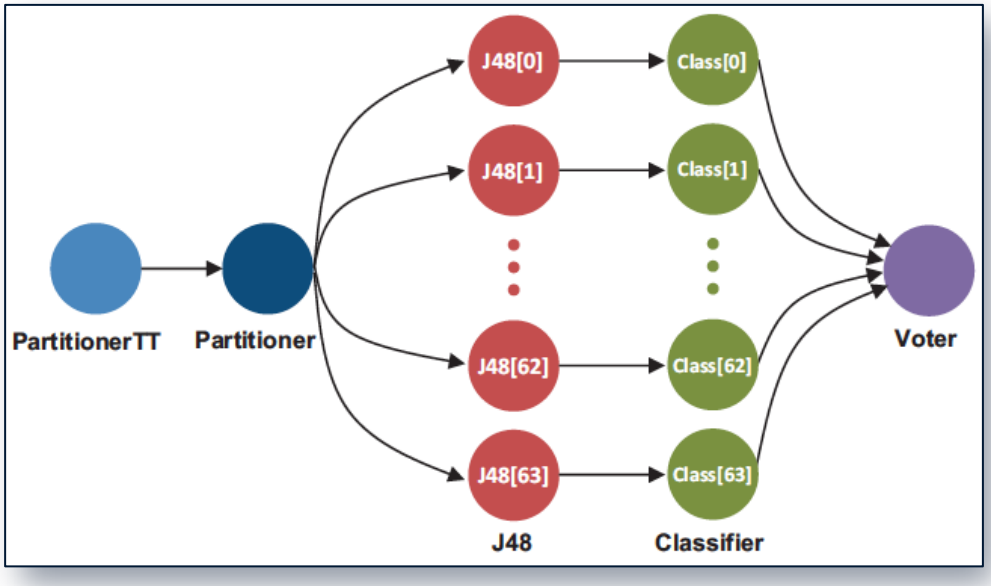
Smart City: Car trajectory pattern detection applications.



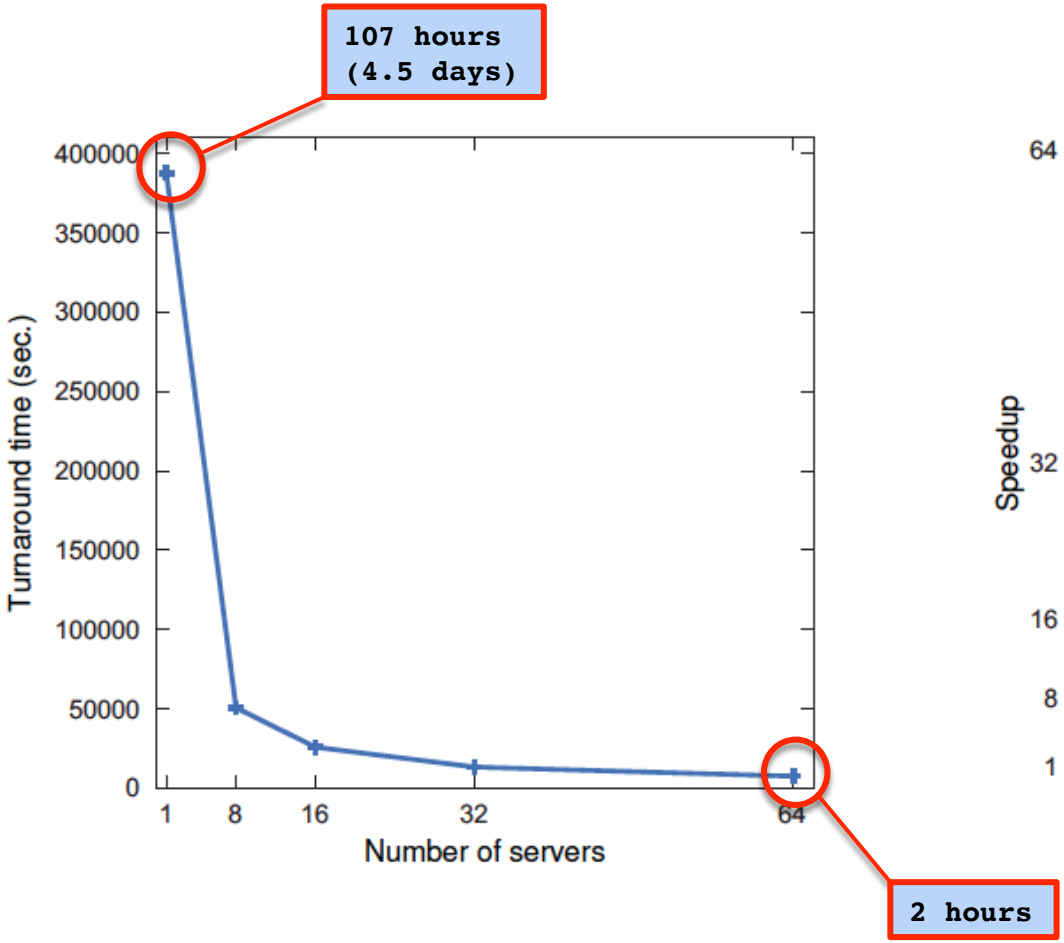
KDDCup99 example

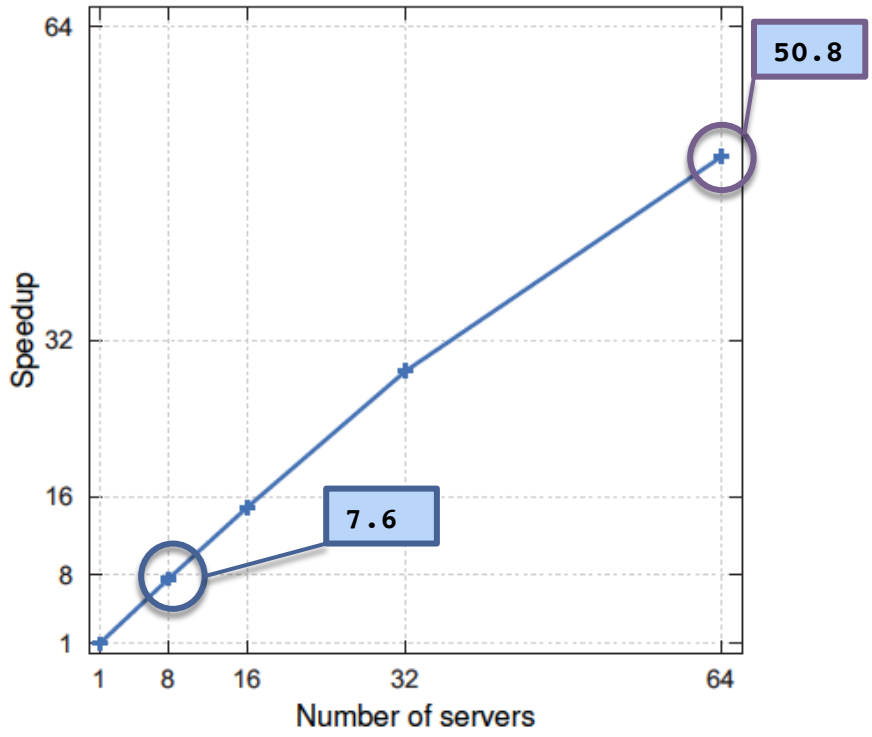
- Input dataset: **46 million tuples**
- Used Cloud: **up to 64 virtual servers** (single-core 1.66 GHz CPU, 1.75 GB of memory, and 225 GB of disk)

```
 1: var n = 64;
 2: var DRef = Data.get("KDDCup99_5GB"),
        TrRef = Data.define("TrainSet"),
        TeRef = Data.define("TestSet");
 3: PartitionerTT({dataset:DRef, percTrain:0.7,
        trainSet:TrRef, testSet:TeRef});
 4: var PRef = Data.define("TrainsetPart", n);
 5: Partitioner({dataset:TrRef, datasetPart:PRef});
 6: var MRef = Data.define("Model", n);
 7: for(var i=0; i<n; i++)
 8:     J48({dataset:PRef[i], model:MRef[i],
           confidence:0.1});
 9: var CRef = Data.define("ClassTestSet", n);
10: for(var i=0; i<n; i++)
11:     Classifier({dataset:TeRef, model:MRef[i],
           classDataset:CRef[i]});
12: var FRef = Data.define("FinalClassTestSet");
13: Voter({classData:CRef, finalClassData:FRef});
```

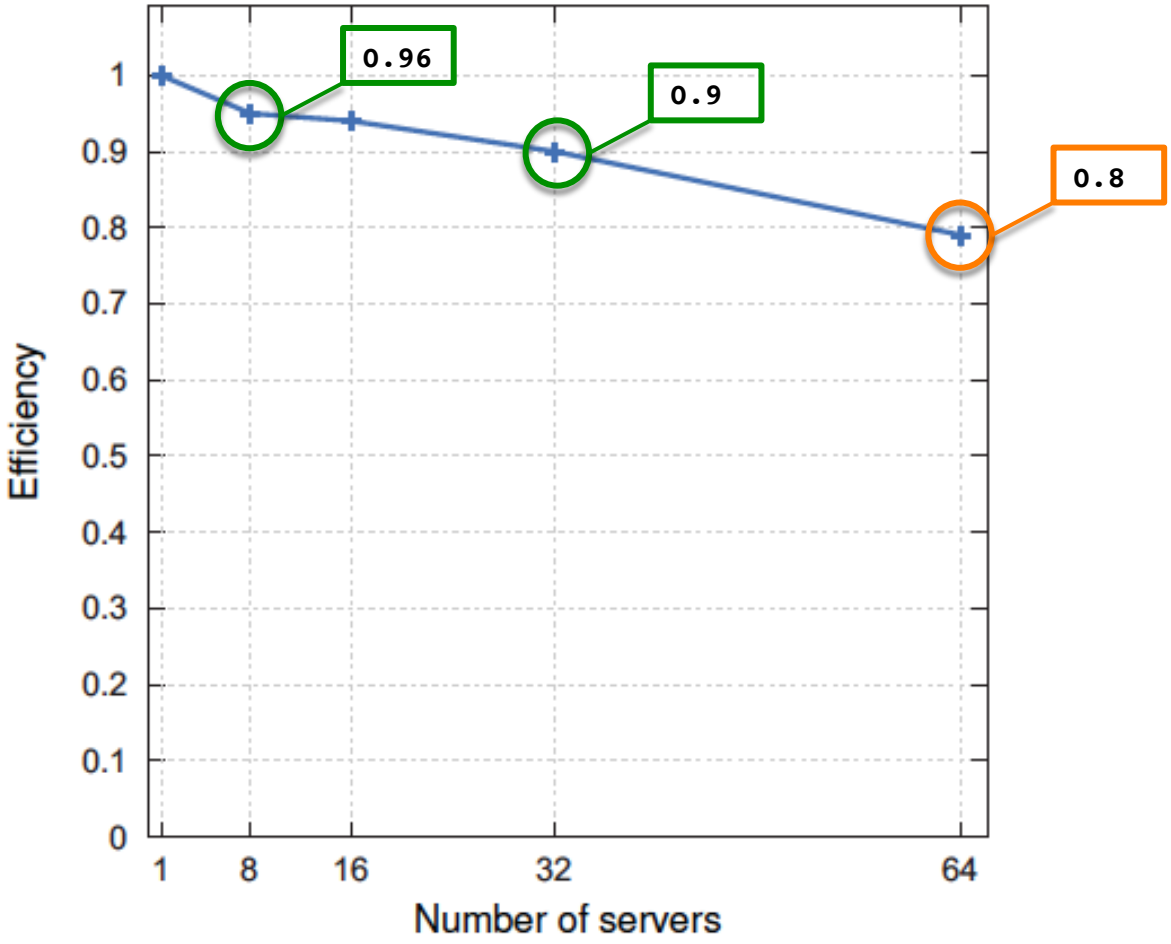


Turnaround and speedup





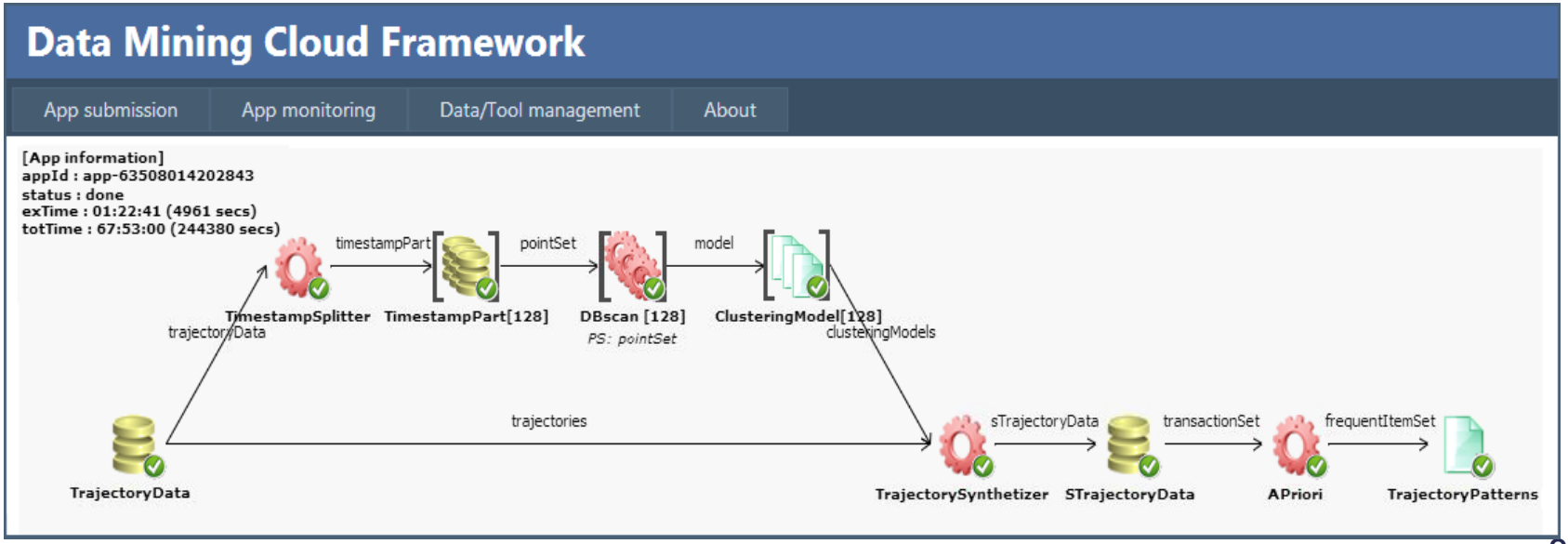
Efficiency


0.96
0.9
0.8
Efficiency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
8
16
32
64
Number of servers

Trajectory pattern detection

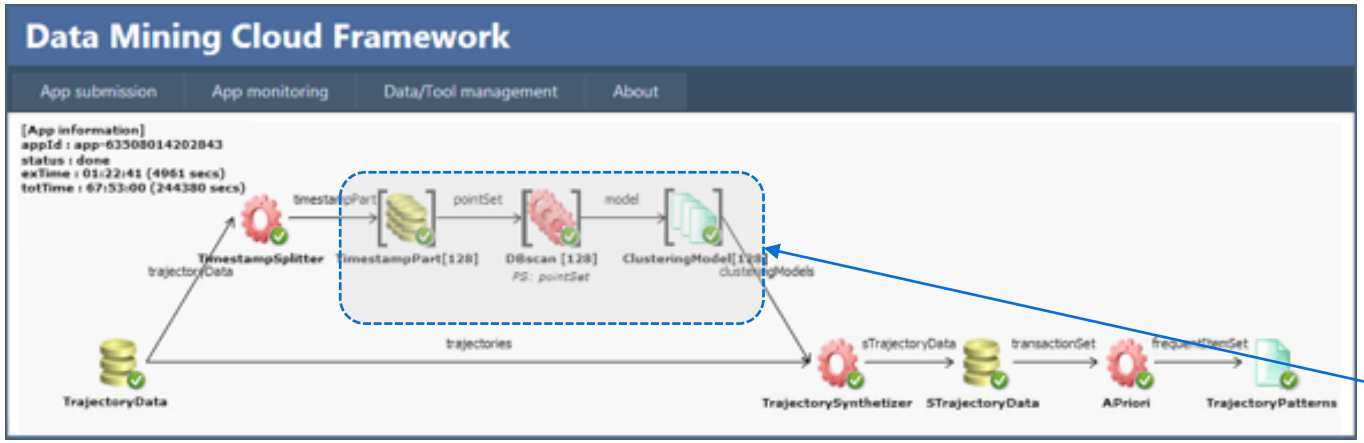
- Analyze trajectories of mobile users to discover movement patterns and rules.

- A workflow that integrates frequent regions detection, trajectory data combination and trajectory pattern extraction.



Workflow implementation

- DMCF visual workflow implementing the trajectory pattern detection algorithm
  - Some nodes are labeled by the array notation
    - Compact way to represent multiple instances of the same dataset or tool
    - Very useful to build complex workflows (data/task parallelism, parameter sweeping, etc.)



128 parallel tasks

A. Altomare, E. Cesario, C. Comito, F. Marozzo, D. Talia, Trajectory Pattern Mining for Urban Computing in the Cloud, IEEE Trans. on Parallel and Distributed Systems, vol. 28, n. 2, 2017.

Discovered dense regions on the Beijing map

(a) time: 6:00 AM

(b) time: 9:00 AM

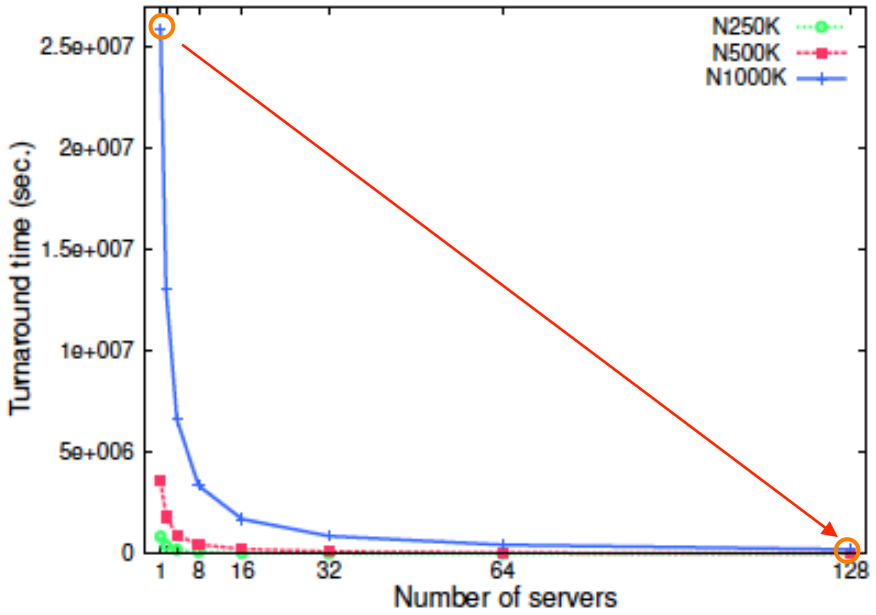
(c) time: 12:00 AM

(d) time: 3:00 PM

Experimental evaluation

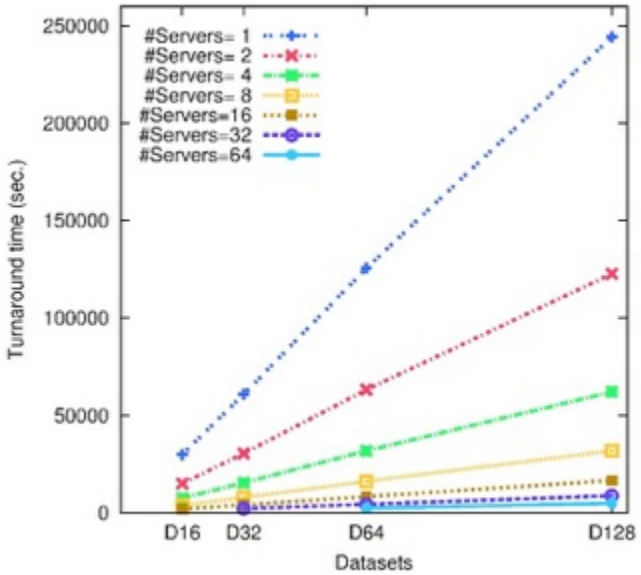
Turnaround time

- vs the number of servers (up to 64), for different data sizes



- comparison parallel/sequential execution
- D_{16} (D_{128}): it reduces **from 8.3 (68) hours to about 0.5 (1.4) hours**

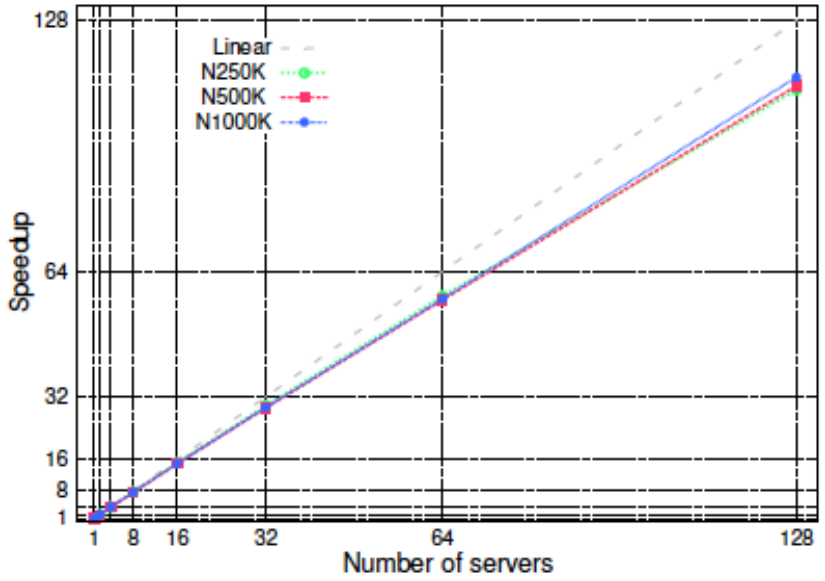
- vs several data sizes (up to 128 timestamps), for different number of servers



- it proportionally increases with the input size
- it proportionally decreases with the increase of computing resources

Experimental evaluation

Speedup



SPEEDUP: 113,5 on 128 servers.

Nubytics

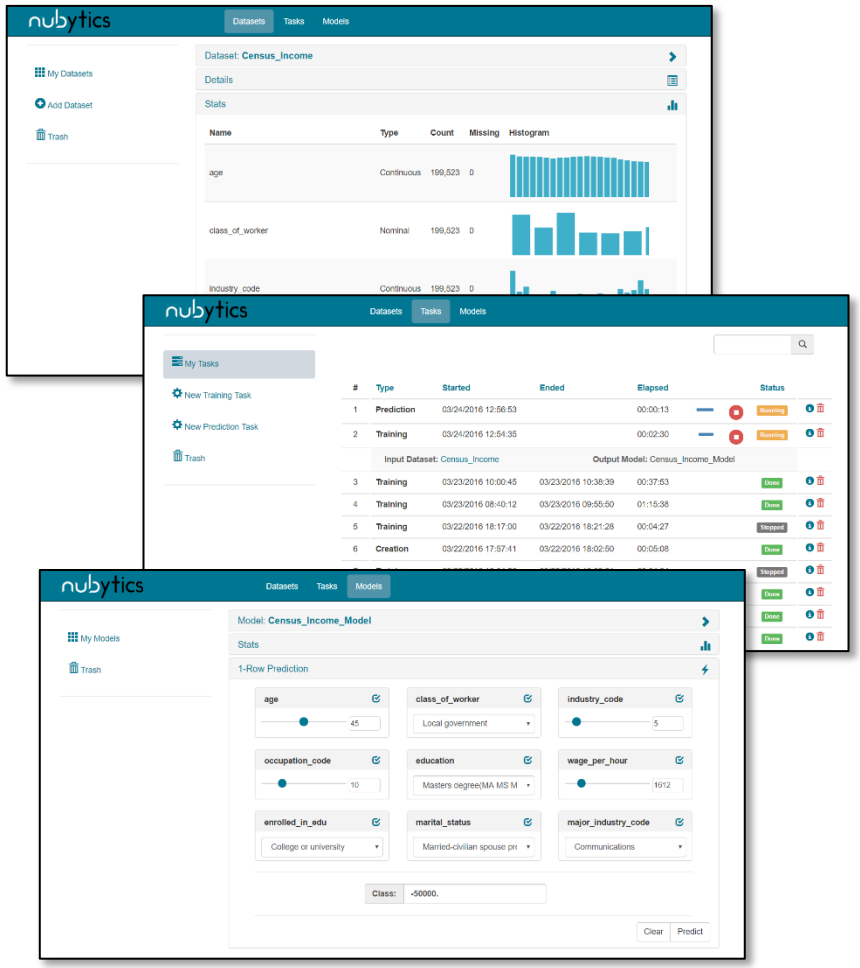
Nubytics

- **SaaS** for data analysis and prediction on the Cloud
- **Nubytics** allows users to **import data** into the Cloud, **extract knowledge models** using high performance **data mining services**, and use the inferred knowledge to **predict new data**
- **Nubytics** provides **data classification** and **regression services** that can be used in a variety of scientific and business applications
- **Scalability** is ensured by a **parallel computing approach** that fully exploits the resources available on the Cloud.
- **Avaliable** at www.nubytics.com

[40]

Nubytics

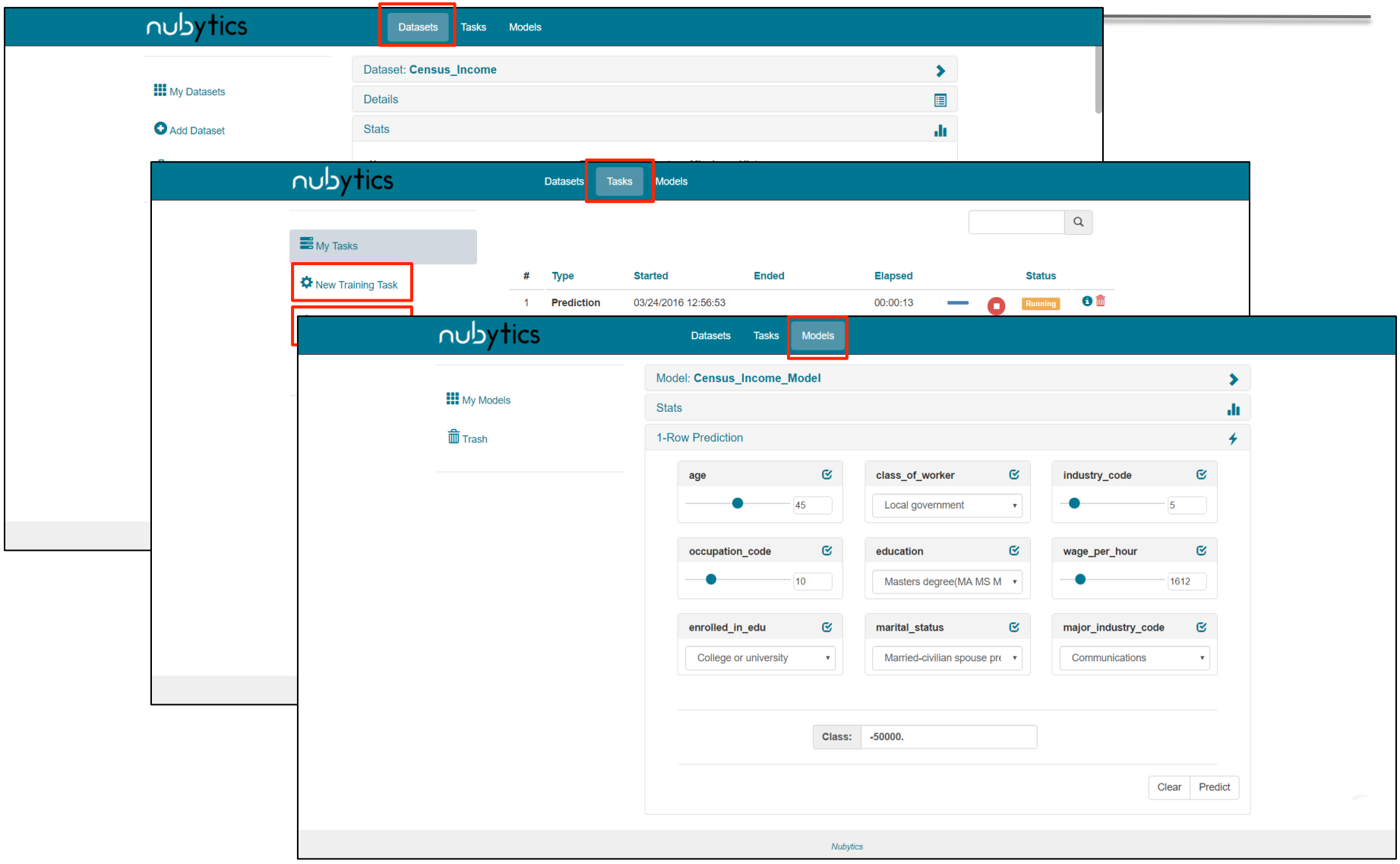
- Main features
  - ✓ Importing and managing *datasets* on the Cloud
  - ✓ Creating *models* from data using high-performance *classification and regression algorithms*
  - ✓ Using the inferred models to *predict* new data



Services (1/2)

- The Nubytics front end is divided into three sections
  - ✓ Datasets
  - ✓ Tasks
  - ✓ Models
- These sections correspond to the three groups of services provided by the system:
  - dataset management,
  - task management and
  - model management.

Services (2/2)


nubytics
Datasets
Tasks
Models
My Datasets
Add Dataset
Dataset: Census_Income
Details
Stats
My Tasks
New Training Task
Type Started Ended Elapsed Status
1 Prediction 03/24/2016 12:56:53 00:00:13 Running
My Models
Trash
Model: Census_Income_Model
Stats
1-Row Prediction
age 45
class_of_worker Local government
industry_code 5
occupation_code 10
education Masters degree(MA MS M
wage_per_hour 1612
enrolled_in_edu College or university
marital_status Married-civilian spouse pr
major_industry_code Communications
Class: -50000.
Clear
Predict
Nubytics

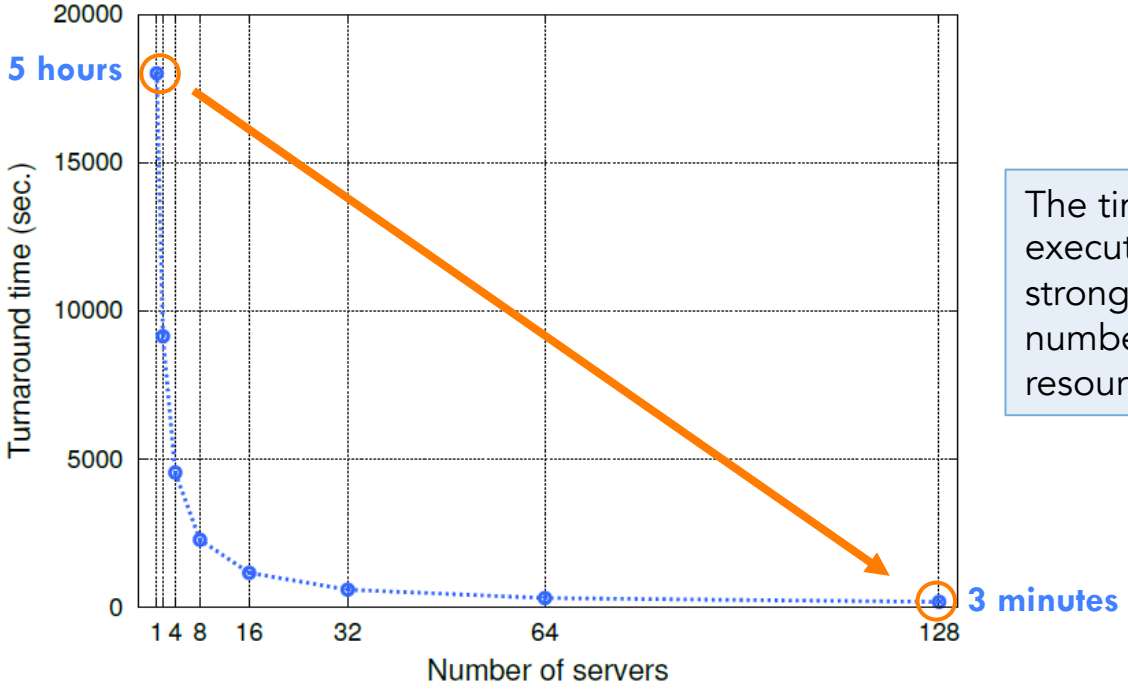
Evaluation (1/3)

- Cloud environment: 128 virtual servers provided by Microsoft Azure, each one equipped with a single-core Intel Xeon E5-2660 2.2GHz CPU, 3.5GB of memory, and 50GB of disk space.

- The input dataset used for the experiments has been generated from the Census-Income Database. From the original dataset, we generated an input dataset containing 4.4 million instances with a total size of about 2.097 GB.

- The goal of the classification task on this dataset is to **train a knowledge model**, that can be used to **predict the income level for a person described by the attributes**.

Evaluation (2/3)



The time required to execute the entire task strongly decreases as the number of computing resources increases

Fig. 7. Turnaround time vs number of servers.

Evaluation (3/3)

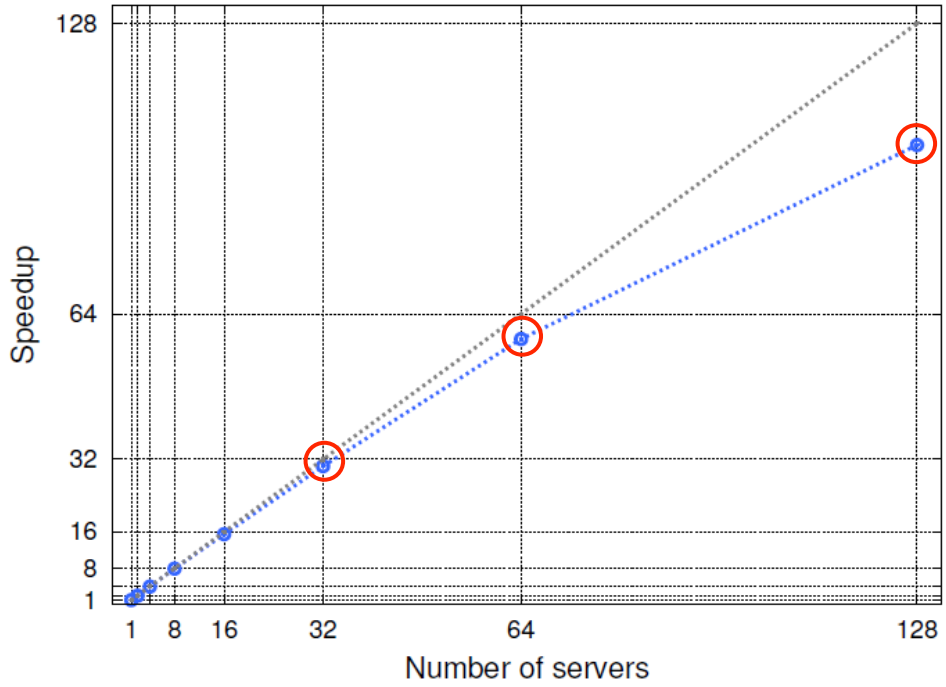


Fig. 8. Speedup vs number of servers.

The speedup is almost linear up to 32 virtual servers and mantains a very good trend for higher number of nodes (about 108 on 128 cores)

Open Issues & Remarks

Scalable Data analysis : Open Research Issues

- *Programming abstracts for big data analytics*.

  MapReduce and the workflow models are often used, but more research work is needed to develop other scalable, **adaptive, general, higher-level** abstract programming structures & tools.

- *Data and tool integration and openness*.

  Code coordination and **data integration** are main issues in large-scale applications that use data and computing resources.

  Standard formats, data exchange models and common APIs are needed.

- *Interoperability of big data analytics frameworks*.

  Large and worldwide federation and integration of multiple data analytics frameworks and services are needed.

Scalable Data analysis : Open Research Issues

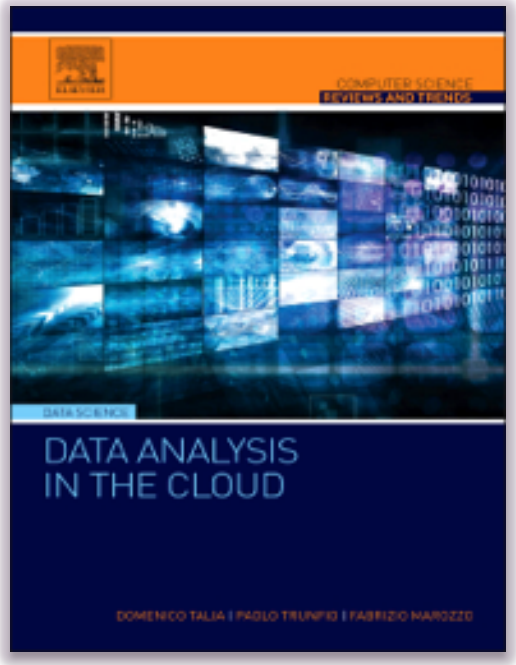
- A significant programming effort of developers will be needed to implement scalable complex mining algorithms and data-driven applications such that used, for example, in big data analysis and distributed data mining.

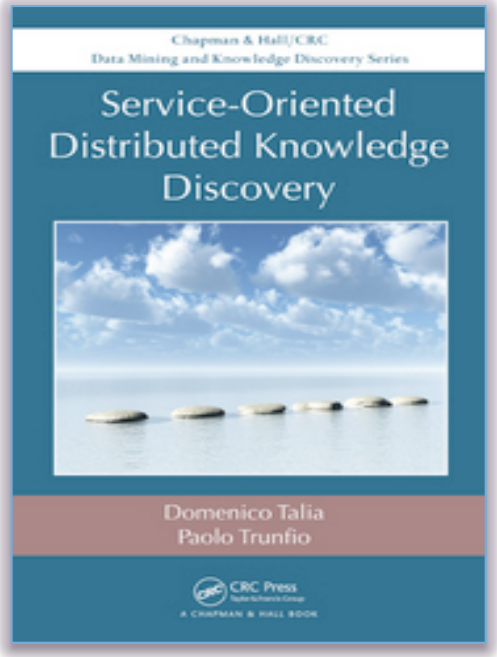
- Parallel and distributed data mining strategies, like
  - collective learning,
  - parallel clustering,
  - meta-learning, and
  - ensemble learning,

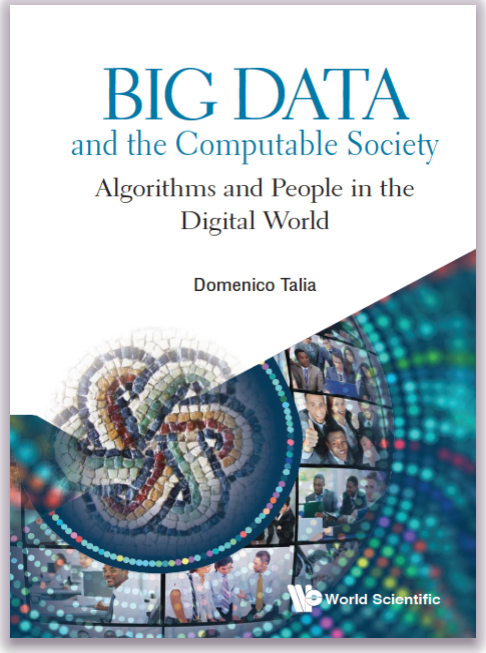
  must be re-designed using parallel and decentralized approaches to be adapted to Cloud and Exascale systems.

Some Books

- D. Talia, P. Trunfio, ***Service-oriented Distributed Knowledge Discovery***, CRC Press, USA, 2012.
- D. Talia, P. Trunfio, F. Marozzo, ***Data Analysis in the Cloud***, Elsevier, USA, 2015.
- D. Talia, ***Big Data and the Computable Society***, World Scientific Press, 2019.

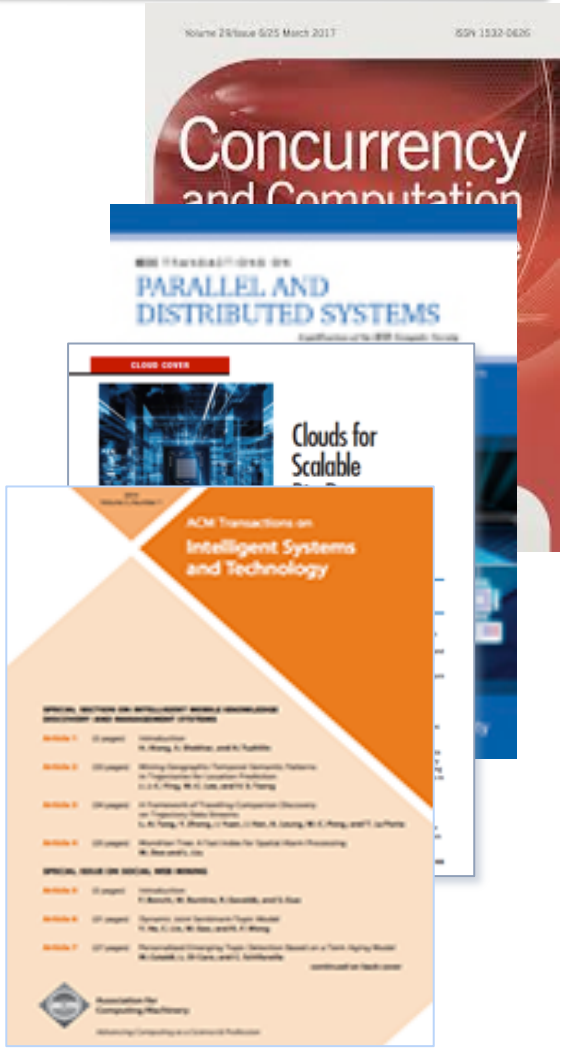






Some papers

- L. Belcastro, F. Marozzo, D. Talia, P. Trunfio, "G-RoI: Automatic Region-of-Interest detection driven by geotagged social media data". *ACM Transactions on Knowledge Discovery from Data*, vol. 12, n. 3, pp. 27:1-27:22, January 2018.
- D. Talia, "Clouds for Scalable Big Data Analytics", *IEEE Computer,* 46(5), pp. 98-101, 2013.
- F. Marozzo, D. Talia, P. Trunfio, "JS4Cloud: Script-based Workflow Programming for Scalable Data Analysis on Cloud Platforms". *Concurrency and Computation: Practice and Experience*, vol. 27, n. 17, pp. 5214-5237, Wiley, December 2015.
- F. Marozzo, D. Talia, P. Trunfio, "A Workflow Management System for Scalable Data Mining on Clouds", *IEEE Transactions on Service Computing*, 2018.
- L. Belcastro, F. Marozzo, D. Talia, P. Trunfio, "Using Scalable Data Mining for Predicting Flight Delays", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8 no. 1, July 2016.
- E. Cesario, A.R. Iannazzo, F. Marozzo, F. Morello, D. Talia, P. Trunfio, Nubytics: Scalable cloud services for data analysis and prediction. IEEE RTSI 2016, 2016.
- A. Altomare, E. Cesario, C. Comito, F. Marozzo, D. Talia, Trajectory Pattern Mining for Urban Computing in the Cloud. *IEEE Trans. Parallel Distrib. Syst.* 28(2): 586-599, 2017.
- C. Comito, D. Falcone, D. Talia, P. Trunfio, "Efficient Allocation of Data Mining Tasks in Mobile Environments". *Concurrent Engineering: Research and Applications*, vol. 21, n. 3, pp. 197-207, Sept. 2013.



Final remarks

- Data mining and machine learning tools are crucial **to support processes finding what is interesting and valuable** in Big Data.
- Cloud cor… as scalable platforms…
- Design an… mplicity and scalability…
- Tools like … plementing and runnin… s.

Questions?