Big Data Analysis Services in the Cloud

Domenico Talia
DIMES, Università della Calabria & DtoK Lab
Italy
talia@dimes.unical.it
Talk outline

- Big problems and Big data
- Using Clouds for data mining
- A collection of services for scalable data analysis
- Data mining workflows
- *Data Mining Cloud Framework (DMCF)*
- *JS4Cloud* for programming service-oriented workflows.
- Final remarks
Knowledge Discovery methods and data mining techniques are used in every application domain to extract useful knowledge from Big Data.

- Single-task applications
- Parameter-sweeping applications/regular parallel applications
- Complex applications (Workflow-based, distributed, parallel)

Cloud Computing can be used to provide developers and end-users with computing and storage services and scalable execution mechanisms needed to efficiently run all these classes of applications.

Goals (1)
Goals (2)

- Discuss 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.
- Evaluate the performance of data mining workflows on DMCF.
- Outline some open research issues.
Big Data never sleeps
Big Data needs scalable analysis

- **Volume** is only one dimension of the problem.

- The most important issue is **Value**.

- **Scalable data analysis** is a key technology to **obtain** Value from Big Data.
Big Data needs scalable analysis

Combination of

- **Big data analysis and knowledge-discovery techniques with**
- **scalable computing systems** for
- an **effective strategy** to obtain new insights in a shorter period of time.

- **Cloud computing** helps!
Data analysis as a service
Knowledge discovery (KDD) and data mining (DM) apps:
- Need to run **compute- and data-intensive processes/tasks**
- Are often based on **distribution of data, algorithms, and users.**

**PaaS (Platform as a Service)** can be an appropriate model to build frameworks that allow users to design and execute scalable data mining applications.

**SaaS (Software as a Service)** can be an appropriate model to implement scalable data mining applications.

Those two cloud service models can be effectively exploited for delivering data analysis tools and applications as services.
By exploiting the SOA model it is possible to define **basic services for supporting distributed data mining tasks/applications.**

Those services can address all the aspects of data mining and in knowledge discovery processes:

- data selection and transport services,
- data analysis services,
- knowledge models representation services, and
- knowledge visualization services.
Services for distributed data mining

- 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)**
Data mining as a service

- It is possible to design services corresponding to

<table>
<thead>
<tr>
<th>Data Mining Applications and KDD processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>This level includes the previous tasks and patterns composed in multi-step workflows.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distributed Data Mining Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>This level implements, as services, patterns such as collective learning, parallel classification and meta-learning models.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Single Data Mining Tasks</th>
<th>Single KDD Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Here are included tasks such as classification, clustering, and association rules discovery.</td>
<td>All steps that compose a KDD process such as preprocessing, filtering, and visualization are expressed as services.</td>
</tr>
</tbody>
</table>
Services for distributed data mining

- This collection of data mining services implements an Open Service Framework for Distributed Data Mining.

Diagram:
- Distributed Data Mining patterns
- Distributed Data Mining Applications and KDD processes
- Data Mining Task Services
- KDD Step Services
Services for distributed data mining

- It allows developers to program distributed KDD processes as a composition of single and/or aggregated services available over a service-oriented infrastructure.

- Those services should exploit other basic Cloud services for data transfer, replica management, data integration and querying.
Services for distributed data mining

- By exploiting the Cloud services features it is possible to develop **data mining services accessible every time and everywhere** (remotely and from small devices).

- This approach can produce not only service-based distributed data mining applications, but also
  - **Data mining services for communities/virtual organizations.**
  - 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 on Clouds: Systems

- **Spark** - an open-source framework for in-memory data analysis and machine learning developed at UC Berkeley.

- **Sphere/Sector** - a compute service built on top of Sector to provide a set of programming interfaces for distributed data analysis applications.

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

- **Swift/T** – a workflow-based system using functional data-driven task parallelism for scaling data-intensive apps.

- Others: **Mahout, HPC-ABDS, CloudFlows**, ... & commercial systems.
The Data Mining Cloud Framework
The Data Mining Cloud Framework supports workflow-based KDD applications, expressed (visually and by a script language) as graphs that link together data sources, data mining algorithms, and visualization tools.
Architecture components

- **Compute** is the computational environment to execute Cloud applications:
  - *Web role*: Web-based applications.
  - *Worker role*: batch applications.
  - *VM role*: virtual machine images.

- **Storage** provides scalable storage elements:
  - *Blobs*: storing binary and text data.
  - *Tables*: non-relational databases.
  - *Queues*: communication between components.

- **Fabric controller** links the physical machines of a single data center:
  - *Compute* and *Storage* services are built on top of this component.
The Data Mining Cloud Framework: Architecture
The Data Mining Cloud Framework: Execution
The Data Mining Cloud Framework: Mapping
Script-based workflows: JS4Cloud
Script-based workflows

- We extended DMCF adding 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 experts to program 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 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**).
**JS4Cloud functions**

*JS4Cloud* implements three additional functionalities, 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.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Access</td>
<td>Data.get(&lt;dataName&gt;);</td>
<td>Returns a reference to the data element with the provided name.</td>
</tr>
<tr>
<td></td>
<td>Data.get(new RegExp(&lt;regular expression&gt;));</td>
<td>Returns an array of references to the data elements whose name match the regular expression.</td>
</tr>
<tr>
<td>Data Definition</td>
<td>Data.define(&lt;dataName&gt;);</td>
<td>Defines a new data element that will be created at runtime.</td>
</tr>
<tr>
<td></td>
<td>Data.define(&lt;arrayName&gt;,&lt;dim&gt;);</td>
<td>Define an array of data elements.</td>
</tr>
<tr>
<td></td>
<td>Data.define(&lt;arrayName&gt;,[&lt;dim1&gt;,...,&lt;dimn&gt;]);</td>
<td>Define a multi-dimensional array of data elements.</td>
</tr>
<tr>
<td>Tool Execution</td>
<td>&lt;toolName&gt;(&lt;par1&gt;:&lt;val1&gt;,...,&lt;parn&gt;:&lt;valn&gt;);</td>
<td>Invokes an existing tool with associated parameter values.</td>
</tr>
</tbody>
</table>
Script-based applications

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

```javascript
// app parameters
var minNumObjList = [2, 5], confidenceList = fromToBy(0.1, 0.5, 0.1);
var dim = minNumObjList.length * confidenceList.length;

var Dataset = Data.get("USCensus"); TrainSet = Data.define("TrainSet"); TestSet = Data.define("TestSet");

// workflow
PartitionerT({dataset: Dataset, percTrain: 70, arffFile: true, trainSet: TrainSet, testSet: TestSet});

var Model = Data.define("Model", dim);

for (var i = 0; i < dim; i++)
    for (var mno = 0; mno < minNumObjList.length; mno++)
        for (var conf = 0; conf < confidenceList.length; conf++)
            TrainSet[Model[i][1], confidence: conf, minNumObj:mno];

var ClassTestSet = Data.define("ClassTestSet", Model.length);
for (var i = 0; i < Model.length; i++)
    Classifier({dataset: TestSet, model: Model[i][1], classColumn: 0, classDataset: ClassTestSet[i]});
```
JS4Cloud patterns

**Single task**

```javascript
var DRef = Data.get("Customers");
var nc = 5;
var MRef = Data.define("ClustModel");
K-Means({dataset:DRef, numClusters:nc, model:MRef});
```
JS4Cloud patterns

**Pipeline**

```javascript
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

Data partitioning

```javascript
var DRef = Data.get("CovType");
var TrRef = Data.define("CovTypeTrain");
var TeRef = Data.define("CovTypeTest");
PartitionerTT({dataset:DRef, percTrain:0.70,
    trainSet:TrRef, testSet:TeRef});
```
var DRef = Data.get("NetLog");
var PRef = Data.define("NetLogParts", 16);
Partitioner({dataset:DRef, datasetParts:PRef});
JS4Cloud patterns

**Data aggregation**

```javascript
var M1Ref = Data.get("Model1");
var M2Ref = Data.get("Model2");
var M3Ref = Data.get("Model3");
var BMRef = Data.define("BestModel");
ModelChooser({model1:M1Ref, model2:M2Ref, model3:M3Ref, bestModel:BMRef});
```
JS4Cloud patterns

Data aggregation

var MsRef = Data.get(new RegExp("^Model"));
var BMRef = Data.define("BestModel");
ModelChooser({models:MsRef, bestModel:BMRef});
JS4Cloud patterns

Parameter sweeping

```javascript
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))});
```
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

```javascript
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});
```

Data Mining Cloud Framework
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)

```javascript
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

- Turnaround time: 107 hours (4.5 days)
- Speedup: 7.6
Efficiency
Another application example

- **Ensemble learning workflow**
  (gene analysis for classifying cancer types)
- Turnaround time: **162** minutes on 1 server, **11** minutes on 19 servers.
- Speedup: **14.8**
Trajectory pattern detection

- Analyze trajectories of mobile users to discover movement patterns and rules.
- A workflow that integrates frequent regions detection, trajectory data synthetization and trajectory pattern extraction.
Application main steps

Frequent Regions Detection
- Detect areas more densely passed through
- Density-based clustering algorithm (DB-Scan)
- Further analysis: movement through areas

Trajectory Data Synthesis
- Each point is substituted by the dense region it belongs to.
- Trajectory representations is changed from movements between points into movements between frequent regions

Trajectory Pattern Extraction
- Discovery of patterns from structured trajectories
- T-Apriori algorithm, i.e. ad-hoc modified version of Apriori
Application workflow
**Workflow implementation**

- DMCF visual workflow implementing the trajectory pattern detection algorithm
  - Each node represents either a data source or a data mining tool
  - Each edge represents an execution dependency among nodes
  - 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.)

![Data Mining Cloud Framework Diagram]

128 parallel tasks!
Discovered dense regions on the Beijing map
Experimental evaluation

Turnaround time

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

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

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

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

Scalability indicators

- **speed-up**
  - notable trend, up to the case of 16 nodes
  - good trend for higher number of nodes (influence of the sequential steps)

- **scale-up**
  - comparable times when data size and #servers increase proportionally
  - DBSCAN step (parallel) takes most of the total time
  - other steps (sequential) increases with larger datasets
Final Remarks
Final remarks

- Data mining and knowledge discovery tools are needed to support finding what is interesting and valuable in Big Data.

- Cloud computing systems can effectively be used as scalable platforms for service-oriented data mining.

- Design and programming tools are needed for simplicity and scalability of complex data analysis processes.

- The DMCF and its programming interfaces support users in implementing and running scalable data mining.
Big is quite a moving target?

- **Yes**, but not only for the increasing size of data.

- **Big** is a misleading term.

- It must include **complexity** (and difficulty) of handling huge amounts of heterogeneous data.

- In the first report (2001) on Big Data the term Big was not actually used.
Big is quite a moving target? Some example

- Some data challenges examples we face today
  - Scientific data produced at a rate of hundreds of gigabits-per-second that must be stored, filtered and analyzed.
  - Ten 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.
Are the challenges faced today different from the challenges faced 10 years ago?

- **Yes**, because data sources are much more than 10 years ago.

- **Yes**, because we want to solve more complex problems.

- **No**, because in data mining we still work on data produced with different goals.

- **No**, because computers were invented to process data quickly.
Is Size/Volume the most important issue in data analysis?

- **No**, volume is only one dimension of the problem.

- The most important issue is **Value**.

- Size and complexity represent the problem, **Value** is the real benefit.
# TABLE OF CONTENTS

- Foreword ......................................................... 6
- Executive summary .......................................... 9
- The global 2016 Security Jam ............................. 12
- Top ten recommendations and roadmap .................. 14
- Overarching themes ........................................... 63
- Moderators’ conclusions ..................................... 80
- Partners’ conclusions .......................................... 99
- Live chat summaries .......................................... 107
- Statistics ................................................. 127
- VIP Jammers .................................................. 132
- Behind the Jam ................................................ 136
TOP TEN RECOMMENDATIONS AND ROADMAP
THE TOP TEN RECOMMENDATIONS

01 Create a global early-warning mechanism
   Improve international information sharing to provide timely warning of impending problems, whether natural such as diseases, earthquakes and other disasters, or political such as unrest, violent extremism and war. Improve intelligence on potential threats by integrating information from regional players, civil society and the private sector to provide a more accurate picture.

02 Improve big data analysis for early action
   There are massive amounts of data available but we lack the means for analysis and for turning information into recommendations while considering ethical issues on surveillance and privacy. To turn early-warning into early action and faster response we must make better use of open source data and social media.

03 Build trust in the Middle East through cooperation on non-security issues
   Focusing on areas of common interest such as climate change, food security and energy security can help regional actors in the Middle East build trust to overcome conflict. The development of non-governmental channels can promote cooperation and reconciliation.

04 Strengthen women’s role in conflict prevention and resolution
   Enabling women to play a key role at every level adds new perspectives and promotes women’s role as actors of change. Their inclusion in international leadership and in spearheading grassroots community initiatives is key to ensuring lasting peace and stability.
THE MOST POWERFUL NATION IN 2030 WILL BE THE ONE CONTROLLING

- Cyberspace: 52%
- Space: 19%
- Sea: 16%
- Land: 7%
- Air: 6%
02 IMPROVE BIG DATA ANALYSIS FOR EARLY ACTION

Lack of data is often not the problem. What is lacking are tools to analyse the massive amounts of information available and turn it into recommendations that can lead to early preventive action.

Information technology (ICT) and big data need to be incorporated into every conversation, project and initiative related to strategic foresight, conflict prevention and tracking evolving security threats.
Smart algorithms and scalable systems

• HPC systems and Clouds equipped with data analysis tools are becoming the most used (and useful) platforms for Big Data analysis.

• New ways to efficiently compose different distributed data mining models and tools are needed for new platforms.

• MAIN ISSUES:
  Data mining algorithms, tools and applications must be designed and ported on such platforms for developing extreme data discovery solutions.
Scalable Data analysis: Open Research Issues

• *Programming abstracts for big data analytics*. The MapReduce and the workflow models are often used on HPC and clouds, 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*. Cloud service paradigms must be designed to allow worldwide federation and integration of multiple data analytics frameworks and services.
Scalable data analysis : Open Research Issues

- Exascale computing systems represent the next computing step also (in particular) in the Big Data field.

- Those systems raise very important research challenges under investigation with the goal of

- Building scalable hw/sw systems composed of a very large number of multi-core processors expected to deliver a performance of $10^{18}$ operations per second.

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

D.A. Reed & J. Dongarra, CACM 2015
Ongoing & future work

- **DtoK Lab** is a startup that originated from our work in this area.

  ![DtoK Lab](www.dtoklab.com)

  **DtoK Lab**
  Scalable Data Analytics

- **Nubytics** is delivered on public clouds as a high-performance Software-as-a-Service (SaaS) to support innovative data analysis tools and applications.

- Applications in the area of **social data analysis, urban computing, air traffic** and others have been developed by JS4Cloud.
Some publications

Thanks

Credits:
P. Trunfio, E. Cesario, F. Marozzo