UNIVERSITY OF TRENTO
DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE

Bachelor degree in Computer Science

Thesis

PROSIT
A TOOL FOR OPTIMAL DESIGN OF SOFT-REAL TIME SYSTEMS

Supervisor: Prof. Luigi Palopoli

Student: Gianluca Bortoli

130

ACADEMIC YEAR 2014-2015

Contents

Abstract

Nowadays data and information are primary assets for companies, organizations and governments. To exploit this huge amount of data new and innovative analytics techniques are proposed and applied in many domains. One of the greatest issues that arise regards the management of the many available datasets and the ability to have readily access to them, alongside the tools needed for their exploitation. Common operations such as data storing, accessing and processing are no-more tasks for just one machine, leading to all the problems related to distributed systems. Some solutions were provided in order to address these issues, but none of them fulfills our needs.

In this work we analyze the existing solutions in the field of data stores, data portals and analytic workflow managers, and we highlighted important problematics coming from the needs of big data managing. To address these issues we designed and implemented DIP, the Dataset Integration Project. Our solution aims at overcoming many challenges in the daily tasks regarding the management of datasets and tools for data analysis. DIP is a system able to mange both datasets and algorithm implementations alike, and that helps the user applying the ones to the others. Through DIP any organization can store, catalog and index datasets and their metadata for easy access and retrieval. The system also allows the creation and execution of jobs and enables workflow re-use, facilitating experiment repetition and the application of the available analytic tools on top of any new datasets. It is designed to plan and execute data analytics pipelines and enabling, on top of that, to generate data provenance information to validate through it the qualities of the data produced in this way. Moreover, in order to

address privacy and copyright issues, our solution provides a flexible resources access control, which is used to support sharing code and data with fine grained permission management. Its functionalities are offered to the users via a web based interface that builds up on a complete set of APIs. Through this APIs, our system is not only of easy integration within any existing environment, but is also a valuable resource for any new data management application. Finally, we present use cases and preliminary evaluations that show how DIP can be a valuable solution to the daunting problem of fully take advantage of the many datasets and algorithm available today.

Chapter 1

Introduction

Nowadays the world is facing an information deluge. Data is produced not only by large organizations, companies and research institutions, but also by every person that holds a mobile device or uses a social network. On one side purchase records, financial transactions and the result of sophisticated experiments may constitute an enormous amount of data, on the other side photos, text messages and location information reach easily comparable dimensions. Single information records (datum) are grouped in datasets whose distribution size is varies a lot: it ranges from only few Megabytes up to hundreds of Terabytes. This huge amount of information, and the fact that there is not a single data format, is ground for a number of issues for data storing, accessing and processing. In order to store these datasets some alternatives are available. There are cloud based solutions, where the storage layer is hidden to users, and they have no direct physical access to their data. In these cases users can manage uploaded datasets usually via a web based interface or, when available, via an API. Other systems address the issues of storing and sharing information at the file system level (e.g., distributed file system solutions), allowing the user to easily partition the data over many machines. They usually provide also some software that tries to fill the gap between the physical and logical storage layout providing simplified higher level functionalities such as directory listing and file search, masking in this way the complexity of a distributed storage. Finally there are systems that provide the possibility to store the data in a more structured way, such as distributed databases (e.g., Apache Cassandra [5]), but these are solutions that are not easily adaptable to all datasets and storage needs.

In order to be able to retrieve the intended information stored in a specific format, some kind of search functionality is needed. This functionality is usually achieved by employing indexes. Indexes are structures (e.g. B-Tree [4]), built upon a subsets of the properties of data that are to be indexed. The storage space and the computational time required to maintain an index is usually payed off from the big speed improvement in retrieving the desired data record.

Nonetheless, being able to store and access raw information is generally not enough by its own. In order to be useful, data has to be cleaned, filtered, processed and analyzed. Researchers and companies continuously develop new procedures and workflows aimed at exploiting the data they collect to obtain new insights about their research or their business [2, 3]. New analysis often builds upon a composition of preexisting results and processing pipelines, in many cases it has to be compared to the outcome of alternative approaches. This means that to successfully conduct valuable analysis, one cannot discard existing results and tools. To this end the reuse of previously developed analytic processes and the integration with third party techniques is essential. Especially in the research field, often experimental results are released only on papers, and analytics techniques are usually described only in theory. This means that many research teams, have not only to implement their intuition, but also the existing techniques they want to use or compare with. This is an issue not only for new research, but also for any existing work that needs to be validated. The scientific method in fact relies on the reproducibility of the experiment. This operation would be highly simplified if source codes and used datasets would be easily accessible. Of course not all the code and the data can be easily shared and made publicly available, either for technical issues or legal restrictions. Thus, the ideal setting where analytic tools and datasets are freely distributed needs to come to terms with many other practical limitations.

We highlight the need for a system capable of storing any amount of datasets of any size. Such system should provide common data management functionalities such as Create, Read, Update, Delete operations and searching, and to work on any kind of data source, from local *CSV* files to remote distributed databases. We aim for a solution that allows the re-use of algorithm implementations, simplifying in this way validation of theoretical results and experiment repetition. Allowing to share also source code of third party applications when needed. The system we propose has to provide a centralized point of control, avoiding for the user to interact directly to many heterogeneous subsystems, and it should be accessible from everywhere in the world, using any kind of devices, being them smart-phones or workstations. Moreover the system has to generate and guarantee data provenance informations. The solution we present should also allow to plan job execution and support complex workflows and pipelines for data processing and analytics. We highlight that, a solution combining a data store with a workflow manager, would address many existing data management challenges and processing issues, and that precious information such as data provenance could be generated and handled almost for free.

Currently, there exist solutions that try to address only partially these issues, these are systems like *CKAN* [7] (a data store), *Oozie* [6] (a workflow manager), and *Hue* [1] (a composition of a data store and a workflow manager). We found these solutions to be not suitable for our needs because they address only a subsets of the aforementioned requirements or enforce constraint that are not acceptable for our use cases. For examples CKAN is designed to be only a data portal and does not feature any capability of process execution. Oozie, on the other end, does not provide datasets storage functionalities. Finally Hue, as many other solutions, only supports the HDFS storage system and can execute only hadoop jobs.

In this work we present the design and the implementation of the Dataset Integration Project, in short, DIP. DIP is a system that stores, catalogs and indexes metadata in order to enable data provenance information generation and validation. It exposes an user-friendly web based graphical user interface (GUI) that allows users to register datasets and algorithms. Moreover it allows the creation and execution of jobs and provides a simple and secure way to share resources and results. It allows experiment repetition, and simplifies code and dataset sharing. Moreover, DIP offers the possibility to catalog also the algorithm implementations and to apply them on the archived datasets, with a simple click. This important feature saves time in many recurrent situations, such as testing algorithms on different datasets under different conditions. We highlight that by sharing existing implementations of data analytics algorithm and by offering the possibilities to apply mining algorithms on public datasets we would empower non-expert users too. DIP coherently integrates existing technologies in one framework as it naturally relies on third party softwares for both storage and job execution, making it easy for it to be deployed on top of any existing environment, exploiting in full the functionalities of these softwares and giving the possibility to implement the missing ones. Concerning the privacy and copyright issues, we offer a group-based access control system that applies to both algorithms and datasets, granting fine control over the sharing process.

DIP has been designed to be inserted in any existing environment, and to adapt quickly to any non currently managed *corner situation*. We are aware that there could be non common uses cases or settings which we may have not considered. In order to allow DIP to be used even in these cases it was designed to be easily adaptable through configuration and completely extensible adding modules.

We describe the existing solutions for data storage and workflow management in Chapter **??** highlighting the limitations we intend to overcome. Then in Chapter **??**, we describe in details the requirements that a complete solution should match in our view and that guided the design of our system. The details of DIP and its implementation are described in Chapter **??** while in Chapter **??** we design methods to evaluate it effectiveness. In Chapter **??** we conclude presenting our result and outlining the next steps that will bring DIP to be a complete solutions ready to be adopted in real word scenarios.

Bibliography

[1] Hue: Analysing data with hadoop. `http://gethue.com/`. Accessed: 2015-03-09.

[2] Manish Kumar Anand, Shawn Bowers, and Bertram Ludäscher. Techniques for efficiently querying scientific workflow provenance graphs. In *EDBT*, volume 10, pages 287–298, 2010.

[3] Sarah Cohen-Boulakia and Ulf Leser. Search, adapt, and reuse: the future of scientific workflows. *ACM SIGMOD Record*, 40(2):6–16, 2011.

[4] Douglas Comer. Ubiquitous b-tree. *ACM Computing Surveys (CSUR)*, 11(2):121–137, 1979.

[5] Elif Dede, Bedri Sendir, Pinar Kuzlu, Jessica Hartog, and Madhusudhan Govindaraju. An evaluation of cassandra for hadoop. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 494–501. IEEE, 2013.

[6] Mohammad Islam, Angelo K Huang, Mohamed Battisha, Michelle Chiang, Santhosh Srinivasan, Craig Peters, Andreas Neumann, and Alejandro Abdelnur. Oozie: towards a scalable workflow management system for hadoop. In *Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*, page 4. ACM, 2012.

[7] Joss Winn et al. Open data and the academy: An evaluation of ckan for research data management. 2013.