

ALLOCATIONS DOCTORALES DE RECHERCHE ARC
Rapport d'activité 2016

ARC No.: 6

Année d'obtention de l'ADR: 2014

Etablissement gestionnaire de la subvention (celui qui a établi le contrat de travail): Université Jean Moulin Lyon 3

Etablissement d'inscription en thèse: Université Jean Moulin Lyon 3

Titre du projet de thèse: Trusted SLA-Guided Data Integration on Multi-Cloud Environment

Nom & prénom du bénéficiaire de l'ADR: AGUIAR DA SILVA CARVALHO Daniel

e-mail: danielboni@gmail.com

Nom du responsable de thèse: GHEDIRA-GUEGAN Chirine

e-mail: chirine.ghedira-guegan@univ-lyon3.fr

Intitulé et coordonnées complètes du laboratoire:

Centre de Recherche Magellan (EA 3713, Lyon3) – Université Jean Moulin Lyon 3

6, Cours Albert Thomas

BP 8242

69355 Lyon cedex 08

Tél.: 04 78 78 71 58

centremagellan@univ-lyon3.fr

Nom du Directeur: Denis Travail

1. Objectifs et l'Originalité du sujet de thèse (2 pages maxi)

Data integration is a widely studied issue in the database domain. It consists in merging data from different databases and providing a unified view of this data to the user [9]. Commonly, data integration is referred in the literature as a problem of answering queries using views. Many authors have reported their algorithms for this purpose [7]. Levy *et al.* proposed the *bucket algorithm*. It builds a *bucket* for each subgoal in the query. A *bucket* includes all views that can be mapped to a given subgoal. Then, the algorithm generates combinations of the different buckets produced, and checks whether each one is a rewriting of the query. Duschka and Genesereth introduced the *inverse-rules algorithm* [6]. It produces a set of inverse rules (one for each subgoal in the view) from local views to the global view. Rewritings are obtained by unfolding the query in terms of the inverse rules produced. Pottinger and Halevy presented the *MiniCon algorithm* in [11]. It identifies a set of views that contains query subgoals. Once a view is selected, the algorithm creates mapping from the query subgoal to the view subgoal if the mapping of variables are possible. These mappings are named MiniCon description (MCD). Finally, a rewriting is a combination of MCDs covering all query subgoals and satisfying the relations between predicates. In general, algorithms in this domain share the same performance problem while combining views to produce a rewriting. Depending on the size of the query and the amount of available views, these algorithms require a lot of computing resources and time to process the rewriting and integration. New architectures like the cloud open challenges to data integration.

Data integration can be seen in the cloud computing as a service composition problem. Selecting services and producing service compositions is computationally costly. Moreover, executing compositions can lead to retrieve and process data collections that can require an important amount of memory, storage and computing resources. Data integration solutions on the service-oriented domain deal with query rewriting problems. In our previous work [5], we have identified trends and open issues regarding the use of SLA in data integration solutions on multi-cloud environments. Barhamgi *et al.* proposed a query rewriting approach which processes queries on data provider services [3]. The query and data services are modeled as RDF views. A rewriting answer is a service composition in which the set of data service graphs fully satisfy the query graph. Benouaret *et al.* introduced a service composition framework to answer preference queries [4]. In that approach, two algorithms based on [3] are presented to rank the best rewritings based on previously computed scores. Ba *et al.* presented an algorithm based on *MiniCon* that produces and orders rewritings according to user preferences [2]. The user preference concept is a score used to rank the order in which services are selected. As in the database domain, these approaches require an important amount of resources to process the rewriting and integration. It is important to review the existing data integration studies to be adapted to the cloud model.

The cloud computing paradigm provides computing resources (as services) in an on-demand and scalable manner to cloud consumers. This on-demand provision and the pricing model imposed by the cloud changes the way data integration solutions should be tackled. Instead of designing processes and algorithms taking into account the limits on resources availability, the cloud sets focus on the economic cost allowing resource consumption and producing results while delivering data under subscription oriented cost models.

When a service is billed to a consumer, the provider and customer agree on price, quality guarantees and penalties associated to its violation in service level agreement (SLA) contracts. Several researches have reported their studies on SLA in different domains [1]. In the cloud context, Rak *et al.* proposed an approach to specify security requirement and to associate them to cloud services [12]. Mavrogeorgi *et al.* introduced a SLA management framework that allows the creation and enforcement of customized SLAs [10]. Leitner *et al.* presented a approach to monitor and predict SLA violations before they impacted the provider's SLA [8]. In general, proposals regarding SLAs in the deployment of services focus on two aspects: (i) approaches focusing on the life cycle of the SLA mainly interested in the contract negotiation phase between the cloud and the service consumer; and (ii) works monitoring contracts and cloud resources in order to avoid SLA violations, and consequently penalties due to its violation. In this sense, to the best of our knowledge, we have not identified any other approach that proposes the use of SLA associated to a data integration solution in a multi-cloud environment.

With growing needs and requirements, applications use different cloud providers for externalizing different data processing and management resources. This fact adds more challenges to data integration considering (i) the large amount and diversity of data; (ii) the user quality and security requirements of the

integration; and (iii) the cloud heterogeneity in expressing and enforcing the corresponding clauses. Indeed, we believe that given the volume and the complexity of query evaluation, it is important to combine and revisit well-known data integration solutions and adapt them to this context. We strongly believe that this can be done according to quality of service requirements expressed by the consumers and service level agreement contracts exported by the cloud providers that host data collections and deliver resources for executing the associated management processes.

Considering the aforementioned, this thesis project intends to address data integration in a multi-cloud hybrid context. The originality of our approach consists in guiding the entire data integration solution taking into account (i) user preferences statements; (ii) SLA contracts exported by different cloud providers; and (iii) several QoS measures associated to data collections properties (for instance, trust, privacy, economic cost). The objective is to propose data integration strategies adapted to the vision of the economic model of the cloud such as (1) accepting partial results delivered on demand or under predefined subscription models that can affect the quality of the results; (2) accepting specific data duplication that can respect user preferences but ensure data availability; and (3) accepting to launch a task that contributes to an integration on a first cloud whose SLA verifies a given requirement rather than on a more powerful cloud but with less quality guarantees in the SLA.

In our work we consider an example from the domain of energy management. My directors are working on two national projects in this domain. So for instance, we assume we are interested in queries like: Give a list of energy providers that can provision 1000 KW-h, in the next 10 seconds, that are close to my city, with a cost of 0,50 Euro/KW-h and that are labeled as green? The question is how can the user efficiently obtain results for her queries such that they meet her QoS requirements, they respect her subscribed contracts with the involved cloud provider(s) and such that they do not neglect services contracts? Particularly, for queries that call several services deployed on different clouds. This work is part of an international collaboration with the DiMAp, Federal University of Rio Grande do Norte.

The project development is occurring naturally and well. In fact, this happens due to several aspects: (i) the close relationship and the easy access to my directors; (ii) the good frequency of meetings. Normally, we have at least one meeting for a week (that can be web conferences) in order to evaluate the status of the work; (iii) the research center infrastructure and team; and (iv) the monthly group meetings where we can discuss about the development of the projects with other colleagues. In addition, there are meetings in another laboratory, LIRIS. These moments are important because we can have an external view and comments about our research.

2. Synthèse du travail effectué dans l'année (2 pages maxi)

During the second year, we have been working on the development of our data integration approach. Given a query and a set of user preferences associated to it, the query execution process is divided in three phases. The figure 1 illustrates our data integration approach.

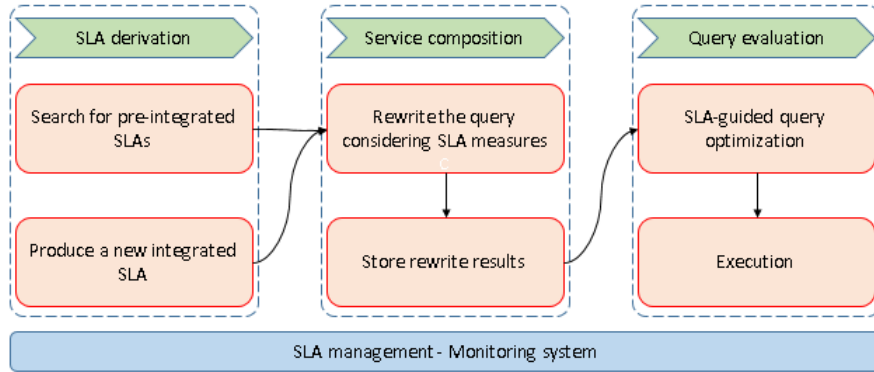


Figure 1: SLA-guided data integration approach

The first phase is the SLA derivation. It consists in looking for a (stored, integrated) SLA derived for a similar request. If a similar SLA is found, the request is forwarded to the query evaluation phase. Otherwise, a new integrated SLA is produced. The query is expressed as a service composition with associated user preferences (specified as SLA measures in the derived SLA). In the second phase, service composition, the query is rewritten in terms of different services considering the user preferences and the SLAs of each service involved in the composition. The rewriting result is stored for further uses. Finally, in the query evaluation phase, the query is optimized in terms of user preferences and SLAs concerning the consumed resources and the economic cost of the query. Once optimized, the query processed in the execution engine. In addition, we are assuming a SLA management module and monitoring system responsible to verify if the SLA contracts are being respected. We have worked on the phases two and three, firstly. The following activities were developed to achieve our objectives:

a) Adapting a previous query rewriting algorithm to our approach. This work was performed in collaboration with our colleagues in Brazil. The idea was to adapt a previous work from their lab to our data integration solution. While integrating we have identified some design problems that made it useless to our approach. However, the work in group help us to identify important issues to be applied to our own implementation.

b) Query rewriting algorithms. We have produced a state of the art about query rewriting algorithm. The approaches are divided into two different domains: database and service-oriented architecture. In the database domain, the different works concerning query rewriting using views have been studied in the literature. In general, the approaches deal with an exponential problem depending on the size of the query and the quantity of views. In the service-oriented domain, the approaches share the same problem. Producing services composition is extremely costly depending on the size of the composition (query) and the amount of services to be combined. Some approaches have tackled this issue. They have tried to minimize the effort to produce rewritings by considering user preferences or by limiting the desired number of rewritings. These works have been used as base and reference to produce our own algorithm.

c) Formalization of the Rhone service-based query rewriting algorithm. We have worked on developing our query rewriting algorithm guided by SLAs. Our work address this issue and proposed the algorithm with two original aspects: (i) the user can express his quality preferences and associated them to his query; and (ii) service's quality aspects defined in SLAs guide the service selection and the whole rewriting process. Given a set of abstract services, a set of concrete services, a user query and a set of user quality preferences, derive a set of service compositions that answer the query and that fulfill the quality preferences regarding the context of data service deployment. The algorithm consists in four steps: (i) select concrete services; (ii) create mappings from concrete services to the query (called concrete service description (CSD)); (iii) combine CSDs; and (iv) produce rewritings. Each phase of the algorithm and each concept (query, concrete services, mapping rules, for instance) are formally specified and described.

d) Configuration of a multi-cloud environment. We have worked on the configuration of a multi-cloud environment. However, we found some problems at this point: (i) the configuration and deployment of cloud infrastructure using open source technologies is not easy and requires important technical skills while configuring the network resources; (ii) it requires a powerful machine. Due to that we have configured a simulation of cloud; and (iii) we have searched for private cloud providers, but they allow few access and power permissions to manage resources. In addition, they are quite expensive. We are still working on best way to manage this issue.

e) Implementation and preliminary experiments. We have implemented the first version of our algorithm using Java. The preliminary experiments have shown that our algorithm can enhance the quality in data integration by considering the user preferences and service's quality aspects extracted from service level agreements. For this experiment we have used a service registry of 35 services. We limited to this amount because we have faced a memory management problem in the Java virtual machine. We are working on improvements in our code in order to avoid the use of unnecessary memory.

f) Poster production (short paper). Once we have finished our first experiments and retrieved the results, we have written and submitted short paper to EDBT 2016.

g) Optimization and experiments. We have proceed with some implementation improvements concerning the way lists and objects are managed. The result was a new version in which are able to run 100 services. We are still working on how to improve it and also thinking about changing the programming language.

h) SLA model to data integration. The state of the art on SLA have been analyzed to serve as basis to our SLA model. The works on SLA to cloud computing can be divided in two groups: (i) approaches dealing with the SLA negotiation phase. They focus on methods to stablish good and well-defined agreements between providers and customers; and (ii) works focus on monitoring/allocating resources in order to detect and avoid SLA violations. These works helped while proposing our SLA model and schema. Note that the SLA is the main concept in our proposal. It is responsible to guide the entire approach from the beginning to the end. There are some challenges regarding SLA: (i) once we are inserted into a multi-cloud context, we are dealing with a large heterogeneity of SLAs among different clouds. It is possible to have the same concept defined in a different way depending on the cloud; (ii) there are different levels of SLA: user SLAs, service SLAs and cloud SLAs. It is necessary to have a mapping between SLA measures that can be expressed in a different manner depending on the level; and (iii) it is possible to face a diversity chain of SLA from different levels and to map and identify measures is a hard process.

i) Paper to ADBIS and VLDB PhD consortium. Currently, with our last results, we have been working on two new papers: one concerning the algorithm to be submitted to ADBIS 2016 (deadline 27th March), and another one focusing on the SLA model, schema and data integration approach to be submitted to the VLDB PhD workshop (deadline 4 April).

3 Production scientifique

D. A. S. Carvalho, N. Benani, C. Ghedira-Guegan, P. A. Souza Neto and G. Vargas-Solar. **A Query Rewriting Algorithm for Data Integration Quality (short paper)**. Submitted to 19th International Conference on Extending Database Technology (EDBT 2016).

D. A. S. Carvalho, N. Benani, C. Ghedira-Guegan, P. A. Souza Neto and G. Vargas-Solar. **Rhone: a quality-based query rewriting algorithm for data integration**. To be submitted in 20th East-European Conference on Advances in Databases and Information Systems (ADBIS 2016).

D. A. S. Carvalho, N. Benani, C. Ghedira-Guegan, P. A. Souza Neto and G. Vargas-Solar. **Quality-oriented data integration on multi-cloud environment**. To be submitted in VLDB PhD Workshop (collocated in the 42nd International Conference on Very Large Data Bases).

Poster presentation in the « Journée Scientifique de l'Arc 6 » held in 20th November 2015 at Grenoble, France.

PhD proposal and query rewriting algorithm presentation to the Service-Oriented Computing team meeting from LIRIS, Lyon 1. In 3rd December 2015.

Query rewriting algorithm and ongoing works presentation to the Information System group from the Magellan's Research Center. In 4th February 2016.

4. Perspectives (1 page maxi)

Once we have been working on the phases two and three of our data integration approach, we are going to carry on completing what is missing on these phases. This concerns to build the module that threats the SLA model and extracts measures information to be used in the rewriting algorithm. With this work done, we are going to integrate both parts: the SLA module and the Rhone algorithm. The evaluation of the approach is essential and can be presented to partners in energy domain to validate the feasibility of our approach. Once the modules are connected, a set of experiments in a multi-cloud environment will be performed to evaluate our solution. The results analysis will be the basis to another scientific article. In parallel, we will carry on writing the thesis document. The table below describes our intended calendar.

The following activities are:

1. Paper submission ADBIS: this paper aims to present the description and formalization of the Rhone service-based query rewriting algorithm, and the results of the experiments produced in a cloud simulation.
2. Paper submission VLDB PhD workshop: this paper describes our data integration approach focusing on the SLA model and schema.
3. Building the SLA Module. The idea is to implement the module responsible (i) to use the SLA model proposed and (ii) to extract information from a diversity of SLA and produce the inputs required to our rewriting algorithm.
4. Optimizing the Rhone algorithm. The goal is to minimize the loss in performance and show that our algorithm gain in time while avoiding to selecting and composing services that clearly violate the user preferences.
5. Integrating modules. The aim is to integrate the SLA module and the Rhone algorithm.
6. Producing and evaluating experiments. Once the different modules and the three phases of the approach are working integrated, experiments are going to be performed in a simulation of multi-cloud to evaluate the entire solution.
7. Producing a scientific paper. Naturally, the results obtained from the previous experiments is going to be used in another scientific production.
8. Writing the thesis.

	Mar	Apr	May	June	July	Aug	Sep	Oct	Nov	Dec	Jan
1	•										
2		•									
3		•	•	•							
4		•	•								
5			•	•							
6				•	•	•					
7						•	•				
8						•	•	•	•	•	•

Bibliography

- [1] Mohammed Alhamad, Tharam S. Dillon, and Elizabeth Chang. A survey on sla and performance measurement in cloud computing. In *OTM Conferences (2)*, volume 7045 of *Lecture Notes in Computer Science*, pages 469–477. Springer, 2011.
- [2] Cheikh Ba, Umberto Costa, Mirian H. Ferrari, Rémy Ferre, Martin A. Musicante, Veronika Peralta, and Sophie Robert. Preference-driven refinement of service compositions. In *Int. Conf. on Cloud Computing and Services Science, 2014*, Proceedings of CLOSER 2014, 2014.
- [3] M. Barhamgi, D. Benslimane, and B. Medjahed. A query rewriting approach for web service composition. *Services Computing, IEEE Transactions on*, 3(3):206–222, July 2010.
- [4] Karim Benouaret, Djamal Benslimane, Allel Hadjali, and Mahmoud Barhamgi. FuDoCS: A Web Service Composition System Based on Fuzzy Dominance for Preference Query Answering, September 2011. VLDB - 37th International Conference on Very Large Data Bases - Demo Paper.
- [5] Daniel A. S. Carvalho, Plácido A. Souza Neto, Genoveva Vargas-Solar, Nadia Bennani, and Chirine Ghedira. *Database and Expert Systems Applications: 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part II*, chapter Can Data Integration Quality Be Enhanced on Multi-cloud Using SLA?, pages 145–152. Springer International Publishing, 2015.
- [6] Oliver M. Duschka and Michael R. Genesereth. Answering recursive queries using views. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '97, pages 109–116, New York, NY, USA, 1997. ACM.
- [7] Alon Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, December 2001.
- [8] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar. Monitoring, prediction and prevention of sla violations in composite services. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 369–376, July 2010.
- [9] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 233–246, New York, NY, USA, 2002. ACM.
- [10] N. Mavrogeorgi, V. Alexandrou, S. Gogouvis, A. Voulodimos, D. Kiriazis, T. Varvarigou, and E. K. Kolodner. Customized slas in cloud environments. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on*, pages 262–269, Oct 2013.
- [11] Rachel Pottinger and Alon Halevy. Minicon: A scalable algorithm for answering queries using views. *The VLDB Journal*, 10(2-3):182–198, September 2001.
- [12] M. Rak, N. Suri, J. Luna, D. Petcu, V. Casola, and U. Villano. Security as a service using an sla-based approach via specs. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 1–6, 2013.