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 Travaille

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 [10]. 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 [8]. Levy *et al.* proposed the *bucket algorithm* [11]. Duschka and Genesereth introduced the *inverse-rules algorithm* [7]. Pottinger and Halevy presented the *MiniCon algorithm* in [13]. [13] extended some ideas of [7], and developed an implementation more efficient than the others. 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 requires a lot of computing resources and time to process the rewriting and integration.

Current data integration implies consuming data from different data services and integrating the results while meeting users' quality requirements. Such requirements include the data that is retrieved and integrated, but also the properties of the data, its producers and the conditions in which such data is produced and processed. For example, whether the user accepts to pay for data, its provenance, veracity and freshness and how much is the user ready to pay for the resources necessary for integrating her expected result. Data services provide data according to specific APIs that specify parameters describing the data to be retrieved and the type of results they can produce. Moreover data provision can be done by services according to different data quality measures. Such measures describe the conditions in which a service can provide or process data and they can be summarized within a service level agreement (SLA). An SLA states, what the user can expect from a service or system. For example, whether it implements an authentication process, if it respects data consumers' privacy and the quality of the data the service can deliver, for example freshness, veracity, reputation and other non-functional conditions like the business model that controls data delivery.

Data provision and data processing services may need to require a considerable amount of storage, memory and computing capacity that can be provided by cloud architectures. Furthermore, users could need to integrate the data provisioned by services in a homogeneous and general result. This integration process can also require important computing resources that can be obtained from the cloud.Data provision and processing services can be deployed in the cloud. Their SLA will include then also the measures about the cloud services that they require to execute requests. The cloud exports a general SLA that specifies the conditions in which users can access the services (infrastructure, platform and software) deployed in it. A user willing to use the cloud services establishes a contract with the cloud guided by an economic model that defines the services she can access, the conditions in which they can be accessed (duplication, geographical location) and their associated cost. Different cloud providers have different possible contracts to establish with users. Thus, for a given requirement, a user could decide which cloud services (from one or several cloud providers) to use according to the type of contracts she can establish with them.

Data integration can be seen in the cloud computing as a service composition problem. Selecting services and producing service compositions is computationally costly. Data integration solutions on the service-oriented domain deal with query rewriting problems. In our previous work [5], we have identified that QoS aspects has started to considered while integrating data, and the cloud has become a popular environment to perform data integration. Moreover, data integration combined with SLA is an open issue in the cloud. 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 order 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 requires an important amount of resources to process the rewriting and integration.

In consequence, data consumption is determined by quality constraints specified by the user (data consumer) and different contracts (i.e., SLA's) of the clouds providing the required services. User's constraints define the storage and computing capacity of the device that consumes the data, the data transmission bandwidth and cost, whether the data consumption is critical (time constraints) and energy consumption.

The SLA of the cloud determines the type of quality a user can expect from its services, according to the contract (subscription) signed between her and the cloud provider. The user profile, her quality requirements and her execution context determine the conditions in which she is expecting to consume data by using such cloud services.

Thus, the first challenge is to compute what we call an integrated SLA that matches the user's quality constraints and requirements with the SLA provided by the cloud services given a specific user cloud subscription. The second challenge is to guide data integration taking into consideration the integrated SLA. Here, the data integration process includes looking for services that can be used as data providers, and for services required in order (i) to retrieve the data; (ii) to build an integrated result; and (iii) to deliver it to the user considering the user quality requirements and resources consumption.

The problem described in previous paragraph suggests to take into consideration the following challenges: (i) The user may have general preferences depending on the context she wants to integrate her data such as economic cost, bandwidth limit, free services, and storage and processing limits; (ii) The SLA is a contract between a service provider and a service consumer. Different entities could take place as a provider or as a consumer building, in this sense, a hierarchy of SLA such as contracts between user and the data service, data service and cloud provider, data service and data service, and cloud provider and cloud provider. In this context, matching the user integration preferences with the services that can contribute to produce a result for her can lead to search and identify in the chain of SLAs the one which contains the desired information to be matched with the user preferences, and probably it is possible to find an incompatibility between the preferences and a SLA in the chain; (iii) In order to fulfill requirements and satisfy user expectations, it is possible to have a collaboration between different clouds. This collaboration implies the agreement through SLAs between services deployed in different cloud providers. In such way, matching user preferences can deal with a heterogeneity of SLA. This mean they probably do not have the same structural schema, and also the same semantic to SLA measures.

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 [14]. Mavrogeorgi *et al.* introduced a SLA management framework that allows the creation and enforcement of customized SLAs [12]. Leitner *et al.* presented an approach to monitor and predict SLA violations before they impacted the provider's SLA [9]. 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.

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. 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.

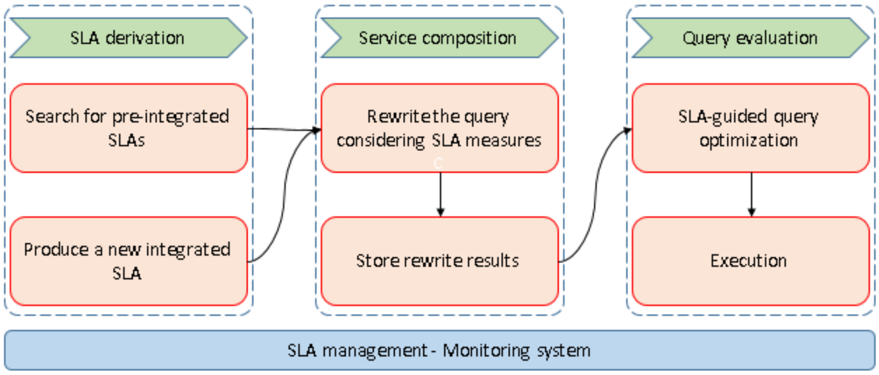

Figure 1: SLA-guided data integration approach

The first phase is the SLA derivation in which a SLA for the user request is created. 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 SLA to the integration (called integrated SLA) is produced. The query is expressed as a service composition with associated user preferences. 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.

[2] have proposed an algorithm for refining services composition. Their goal was to use user preferences to select and rank services in order to avoid the exponential problem while combining and producing compositions. Composition are incrementally produced until to reach a (predefined) desired number. In collaboration with our colleagues in Brazil (authors in [2]), we have worked on an adapted version of [2] to our data integration solution extending their MCD definition, and adding the concepts of user preferences to the query and quality measures to the services. However, while performing the integration we have identified some design issues that made it useless to our approach: (i) their concept of user preferences are scores associated to services previous defined by the user while, for us, user preferences are quality requirements expected by the user concerning the whole integration; (ii) their algorithm accepts rewriting including calls to services that are not interest for the composition. Assuming that on the cloud each service has a price associated to its request, these composition that calls useless services produces an extra cost to the user. This adaptation process helped us to identify important issues to be applied to our own implementation such as developing an better approach to produce the combinations of services.

Based on the related work, we have developed and formalized the *Rhone* service-based query rewriting algorithm guided by service level agreements (SLA). Our work address this issue and proposes the algorithm with two original aspects: (i) the user can express her quality preferences and associated them to his query; and (ii) while trying to fulfill the user quality prefereces, service's quality aspects defined in SLAs guide the service selection and the whole rewriting process (Services and rewritings should meet the user requirements). Given a set of abstract services, a set of concrete services and a user query (both defined in terms of abstract services), and a set of user quality preferences, the *Rhone* derives 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) *Selecting concrete services*. Similar to [11, 13] our algorithm selects services based on the abstract services that exists in the query, but it includes two differences: first, a concrete service cannot be select if it contains an abstract service that is not present in the query; and second, the service' quality aspects (extracted from its SLA) must be in accordance with the user quality preferences; (ii) *creating mappings from concrete services to the query (called concrete service description (CSD))* inspired in [13] including also the information concerning the services' SLA; (iii) *combining CSDs*; and (iv) *producing rewritings* until to fulfill the user requirements according to the services' SLA. Each phase of the algorithm and each concept (query, concrete services, mapping rules, for instance) were formally specified and described.

In order to evaluate our approach and the *Rhone* algorithm, we have worked on configuration of a multi-cloud environment. We have searched for open source solutions instead of privates once they are (i) quite expensive; and (ii) do not allow to extend and access directly the different level of SLAs. The OpenStack was selected as our technology. We have installed and configured the different modules necessaries to the OpenStack. However, we have some issues: (i) the configuration and deployment of cloud infrastructure require important technical skills while configuring the network resources; (ii) it requires a powerful machine. Due to these reasons we have configured a simulation of cloud run our experiments.

The first version of the *Rhone* was implemented using Java according to its formal definition. The algorithm was tested in a cloud simulation containing 35 services in its service registry. We have tested different types of query varying on the size and on the number of user preferences. Although our algorithm shares the same time performance problem as the previous approaches while combining compositions, the preliminary experiments have shown that the Rhone can enhance the quality in data integration by considering the user preferences and service's quality aspects extracted from service level agreements. Our approach avoid selecting and using services to produce compositions that are not interest to the user once they do not fulfill his quality requirements. In addition, as result we have submitted short paper to EDBT 2016.

We developed an improved version of the algorithm that manages better the manner in which lists of objects are managed. We have applied this version to a new set of experiments running 100 concrete services. With the results obtained from this experiment and the final version of our formalization, we are working on a new paper that included an extensive description of the algorithm and its evaluation to be submitted to ADBIS 2016 (deadline 27th March).

With this work performed, we have been designing our SLA model to data integration. As mentioned before, proposals to SLA in cloud computing can be divided in two groups: (i) approaches dealing with the SLA negotiation phase. They focus on methods to establish 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 us while proposing our SLA model and schema. The SLA is the main concept in our proposal. It is responsible to guide the entire approach from the beginning to the end while fulfilling the user requirements. At this point, some challenges arises: (i) In a multi-cloud context, we are dealing with a large heterogeneity of SLAs among different clouds in terms of semantics and structure. A SLA and its measure can be defined in a different way depending on the cloud; (ii) there are different levels of SLA: between users and services, between services and clouds, between clouds and clouds. Consequently, a user requirement defined in a user SLA is computed in terms of different measures on Service and Cloud SLAs. It is necessary to have a mechanism that maps and compute the user requirement and SLA measures; and (iii) it is possible to exists a chain of SLA, and to map and computed measures in this chain can require a hard processing. i) Paper to ADBIS and VLDB PhD consortium. Currently, we have been working on a paper that focus on the description of our SLA model, schema and data integration approach to be submitted to the VLDB PhD workshop (deadline 4 April).

3 Production scientifique

Daniel Carvalho, Nadia Bennani, Genoveva Vargas-Solar, Chirine Ghedira, Placido Souza Neto. **Can Data Integration Quality be Enhanced on Multi-cloud using SLA?**. DEXA 2015, Sep 2015, Valencia, Spain. 2015.

Paper proposal presentation. **Trusted SLA-Guided Data Integration on Multi-cloud Environments**. Seminar presented to the Information System group from the Magellan's Research Center. In March 5th 2015.

Poster presentation. **SLA-Guided Data Integration on Multi-cloud Environments**. Poster presented in the ≪ Journée Scientifique de l'Arc 6 ≫ held in 20th November 2015 at Grenoble, France.

PhD proposal and query rewriting algorithm presentation. **Trusted SLA-Guided Data Integration on Multi-cloud Environments**. Seminar presented to the Service-Oriented Computing team meeting from LIRIS, Lyon 1. In 3rd December 2015.

Query rewriting algorithm and ongoing works presentation. **A Service-based Query Rewriting Algorithm**. 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 second phase of our data integration approach, we are going to carry on completing what is missing on the other phases. This concerns to (i) propose the user, service and cloud SLA schema for data integration; (ii) develop and describe our data integration architecture; and (iii) build the module that threats the SLA and extracts its measures information to be used in the *Rhone* rewriting algorithm. An optimization of our algorithm is also necessary to be able to process a huge amount of services.

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 it 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 start writing the thesis document. The table below describes our intended calendar. In green, you can see activities that we have done (previously described in the section 2). In blue, future activities.

	2nd year												6 months 3rd year					
	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2
1. Adapting [9] to our approach																		
2. State of the art on query rewriting algorithms																		
3. Proposal and Formalization of the Rhone query rewriting algorithm																		
4. Implementation of the Rhone in Java																		
5. Configuration of the cloud environment and preliminary experiments																		
6. Short paper submission to EDBT																		
7. Improving and optimizing the Rhone and new experiments																		
8. Proposal of SLA schema, model and approach to data integration																		
9. Paper ADBIS: describing the Rhone algorithm and its formalization																		
10. Paper VLDB PhD workshop: describing our SLA schema, model and approach																		
11. Refinement of the SLA schema for users, services and clouds																		
12. Refinement and improving of SLA-guided architecture for data integration																		
13. Building the module responsible to threat SLAs																		
14. Integrating different modules of our architecture																		
15. Simulating the multi-cloud and running first experiments																		
16. Producing a scientific paper																		

Avis du Responsable de l'ARC

Avis du Conseiller scientifique de la Région Auvergne-Rhône-Alpes

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] UmbertoS. Costa, MirianHalfeld Ferrari, MartinA. Musicante, and Sophie Robert. Automatic refinement of service compositions. In Florian Daniel, Peter Dolog, and Qing Li, editors, *Web Engineering*, volume 7977 of *Lecture Notes in Computer Science*, pages 400–407. Springer Berlin Heidelberg, 2013.

[7] 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.

[8] Alon Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, December 2001.

[9] 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.

[10] 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.

[11] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 251–262, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[12] N. Mavrogeorgi, V. Alexandrou, S. Gogouvitis, 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.

[13] Rachel Pottinger and Alon Halevy. Minicon: A scalable algorithm for answering queries using views. *The VLDB Journal*, 10(2-3):182–198, September 2001.

[14] 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.