SLA-Guided Data Integration on Cloud Environments

Nadia Bennani
LIRIS, INSA-LYON
Universite de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
nadia.bennani@insa-lyon.fr

Chirine Ghedira-Guegan
MAGELLAN, IAE,
Universit Jean-Moulin Lyon 3, France
chirine.ghedira-guegan@univ-lyon3.fr

Martin A. Musicante
DIMAp, UFRN
Natal, Brazil
mam@dimap.ufrn.br

Genoveva Vargas-Solar
CNRS, LIG-LAFMIA
St. Martin D'Hères, France
genoveva.vargas@imag.fr

***Abstract*—Existing data integration techniques have to be revisited in order to query multiple data on the Cloud. Service Level Agreements implement the contracts between the cloud provider and the users, as well as between the cloud and service providers. Owing to SLA heterogeneity and to data integration scalability problems, we propose an SLA guided data integration for querying data on multiple clouds.**

***Keywords*-SLA; Cloud Computing; Data Integration;**

I. Introduction

The recent emergence of the cloud paradigm opens new challenges to data processing. Indeed, unlimited access to cloud resources and the ”pay as U go” model change the hypothesis for processing big data collections. Nevertheless, integrating and processing heterogeneous data collections, calls for efficient methods for correlating, associating, filtering those data taking into consideration their structural characteristics (due to the different data models) but also their quality, e.g., trust, freshness, provenance, partial or total consistency.

Existing data integration techniques have to be revisited in order to integrate data that are both weakly curated and described through metadata or schemas. This issue is highlighted by the numerous resources deployed by several providers on the cloud, and for which Service Level Agreement Contracts (SLA) are associated. In the current model, users sign a contract with one or many cloud providers. The contract is materialized as an *user SLA*. Each user has his own constraints and quality of service requirements when querying data on the cloud.

Let us illustrate our problem by an example from the domain of energy management. We are interested in queries like: *What is the 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 labelled as green?*. We consider a simplified SLA cloud contract inspired in the lowest contract provided by Azure: *cost of $0,05 cents per call, 8 GB of I/O volume/month, free data transfer cost within the same region, 01 GB of storage.* The user is ready to pay a maximum of *$5 as total query cost*; she requests that only *green* energy providers are listed (provenance), with at least *85*% of precision of provided data, even if they are not fresh; she requires an availability rate of at least 90% and a response time of *0,01 s*.

The question is how can the user obtain efficiently results for her queries that meet her Q&S requirements while respecting her subscribed contracts with the cloud and also without neglecting services contracts, especially for queries to several services deployed eventually on different clouds?

To our knowledge, projects and deployments of SLAs in the context of Cloud Computing aim and rely on two principles: (i) The negotiation of conditions, which are statically agreed between the parts ([**?**], [**?**], [**?**] (ii) The monitoring of these conditions during the use of cloud resources in order to detect SLA contract violation ([**?**], [**?**]).

Our work intends to address data integration on a cloud guided by the SLA exported by different cloud providers and by QoS measures associated to data collections properties: trust, privacy, economic cost. This implies several granularities of SLA: first, at the cloud level; the SLA ensured by providers regarding data; then at the service level, as unit for accessing and processing data, to be sure to fit particular service needs; and finally at the integration level i.e the possibility to process, correlate and integrate big data collections distributed along different cloud storage supports, providing different quality properties to data (trust, privacy, reliability, etc). The goal is to propose an SLA guided data integration system exported as a distributed DaaS by a set of clouds providers that handles the SLA interoperability and collaboration.

In this paper, we present our approach and optimized economic driven and client-oriented data integration (lookup, aggregation, correlation) strategies adapted to the vision of the economic model of the cloud. We aim at (i) accepting partial results delivered on demand or under predefined sub-

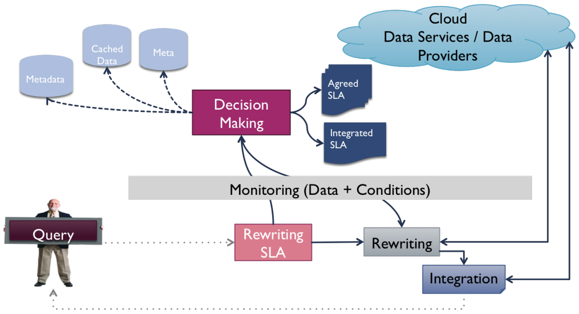

Figure 1. SLAG-DIAAS architecture

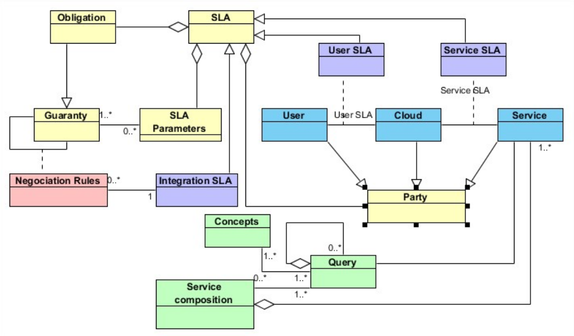

Figure 2. SLA extension

scription models that can affect the quality of the results; (ii) accepting specific data duplication that can respect privacy but ensure data availability; (iii) accepting to launch a task that contributes to an integration on a first cloud whose SLA verifies security requirement rather than a more powerful cloud but with less security guarantees in the SLA.

II. The SLA-guided data integration approach

Our SLA guided data integration approach proposes three steps starting from the processing of the request to the delivery of the result sets. Given an expressed query and a set of Q&S preferences: cost, data provenance, service reputation, execution deadline and so on, the system processes in three steps, as follows: first (i) an SLA derivation computation is performed to filter possible data and services providers. This may be done using a set of matching algorithms based on graph structures and RDF specification. Then, (ii) a query rewriting is done, in term of possible service compositions giving partial or exhaustive results according to SLAs content; and finally, (iii) the results are integrated into an answer. These steps generate intermediate results that are stored as knowledge in order to reduce the overhead of further query evaluation processes. Moreover, an integration SLA is generated to archive the negotiated rules obtained during the integration. For an incoming query, the whole process is monitored to determine whether a integration SLA is being honoured .

Figure 1 shows the SLAG-DIAAS architecture of an SLA guided data integration system that is supported by data services which are data providers deployed in a cloud and that provide agreed SLAs. The architecture keeps a *directory* together with meta-data about the way queries are evaluated for producing results. The system uses this information by query processing and monitoring modules for rewriting queries according to given quality of service (QoS) preferences expressed by a data consumer.

Figure 2 depicts how we extend the SLA structure in order to be able to capitalize on previous SLA negotiation. The yellow part of the schema gives an abstract presentation of SLA content necessary to describe our extension. Compared to standard SLA where two mandatory parties are concerned and a set of optional ones, we propose a set of parties formed by those (clouds, services, user) actors that are concerned in the data integration process. After the filtering process, the service selects the service composition that will cover the total result set. For each selected composition an integration SLA should be derived from the services and the user SLA. For each, obligations concerning the same items will be confronted to produce some new guaranties specified by the negotiation rules. The semantic analysis of the query allows to extract a set of concepts that are associated to the built *Integration SLA*. These concepts are helpful in case of a future query that uses the same concepts and will allow to fetch for compositions that already meet the requirements of the user at first and the SLA of the entities implied thank to the Integration SLA.

III. Conclusion

Current big data settings impose today to consider SLA and different data delivery models. We believe that given the volume and the complexity of query evaluation that includes steps that imply greedy computations, it is important to combine and revisit well-known solutions and adapted to this contexts. We are currently developing the strategies and algorithms sketched here applied to energy consumption applications as the one described in the paper and also to elections and political campaign data integration in order to guide decision making on campaign strategies.

References

[1] V. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level slas - lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments," in *HPCS 2010*, 2010, pp. 48–54.

[2] A. V. Dastjerdi, S. G. H. Tabatabaei, and R. Buyya, "A dependency-aware ontology-based approach for deploying service level agreement monitoring services in cloud," *Softw. Pract. Exper.*, vol. 42, no. 4, pp. 501–518, Apr. 2012.

[3] J. Ortiz, V. T. de Almeida, and M. Balazinska, "A vision for personalized service level agreements in the cloud," in *Proc. 2nd Workshop on Data Analytics in the Cloud*. ACM, 2013, pp. 21–25.

[4] M. Hale and R. Gamble, “Secagreement: Advancing security risk calculations in cloud services,” in *IEEE SERVICES*, 2012, pp. 133–140.

[5] I. Brandic, V. Emeakaroha, M. Maurer, S. Dustdar, S. Acs, A. Kertesz, and G. Kecskemeti, “Laysi: A layered approach for sla-violation propagation in self-manageable cloud infrastructures,” in *IEEE COMPSACW*, 2010, pp. 365–370.