SLA-Guided Data Integration on Cloud Environments

Nadia Bennani
Univ. Lyon, CNRS INSA-Lyon, LIRIS, UMR5205, France
nadia.bennani@insa-lyon.fr

Chirine Ghedira-Guegan
MAGELLAN, IAE, Univ. J-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 to query big data collections on the Cloud. Service Level Agreements implement the contracts between the cloud provider and the users, and between the cloud and service providers. Given SLA heterogeneity and 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 for 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 to integrate big data collections that are both weakly curated and sometimes 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 her own constraints and quality of service (QoS) 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: *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 labelled as green?* We consider a simplified SLA cloud contract inspired in the cheapest 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, 1 GB of storage.* The user is ready to pay a maximum of *$5 as total query cost*; she requests that only *green* energy providers should be 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 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.

To our knowledge, projects and deployments of SLAs in the context of Cloud Computing aim and rely on two principles: (i) The negotiation of use conditions, which are statically agreed between the parts ([1]–[3] (ii) The monitoring of these conditions as cloud resources are used, to detect SLA contracts violation ([4], [5]).

Our work addresses data integration on a cloud, guided by SLAs 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 data providers. Then, at the service level, where a service is a data accessing and processing unit, to be sure that the SLA fits particular needs (e.g., response time, availability). At the integration level i.e the possibility to process, correlate and integrate big data collections distributed across 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 data as a services (DaaS) by a set of cloud providers, that handles SLA interoperability and collaboration.

In this paper, we present our SLA data integration guided approach based on strategies (lookup, aggregation, correlation) adapted to the vision of the economic model of the cloud. We aim at (i) accepting partial results delivered on demand or under predefined subscription models that can affect the quality and cost 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 requirements 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 query processing to the delivery of result sets. Given a query and a set of QoS preferences (cost, data

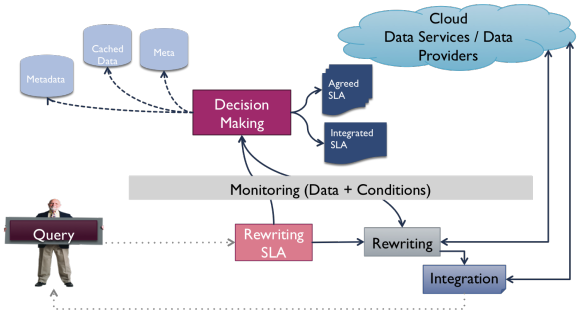

Figure 1. SLAG-DIAAS architecture

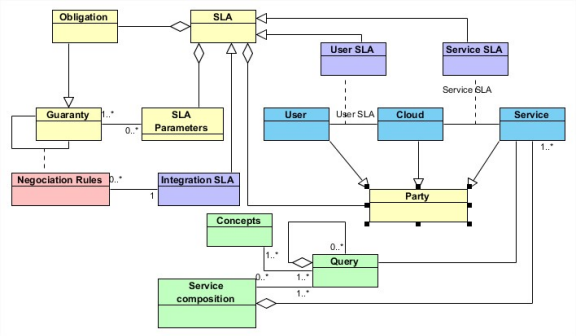

Figure 2. SLA extension

provenance, service reputation, execution deadline and so on), the system processes it in three steps: (i) *SLA derivation*, performed to filter possible data and services providers using a set of matching algorithms based on graph structures and RDF specifications; (ii) *query rewriting* for computing possible service compositions giving partial or exhaustive results according to defined SLAs; (iii) *results integration* into an answer. These steps generate intermediate results that are stored as knowledge to reduce the overhead of further query evaluation processes. Moreover, an integrated SLA is generated to archive negotiated rules obtained during the integration. For an incoming query, the whole process is monitored to determine whether the integration SLA is being honoured.

Figure 1 shows the SLA Guided - Data Integration As A Service (SLAG-DIAAS) architecture supported by data services which are data providers deployed in a cloud and that provide agreed SLAs. The SLAG-DIAAS keeps a *directory* together with meta-data about the way queries are evaluated for producing results. Query processing and monitoring modules use this information for rewriting queries according to given quality of service (QoS) preferences expressed by a data consumer.

Figure 2 depicts our extension of the SLA model that capitalizes on previous SLA negotiation. The yellow part of the schema gives an abstract representation of standard 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 actors that are concerned in the data integration process, namely, clouds, services, and user. Once the filtering process has been completed, the SLA-GIDAAS selects the service composition that will produce the total result set. For each selected composition, an integrated SLA should be derived from the services and the user SLA. Obligations concerning the same items will be confronted to produce some new guarantees specified by negotiation rules. The semantic analysis of the query allows to extract a set of concepts that are associated to the built *Integrated SLA* (see Figure 2). The integrated SLA concepts can be reused for evaluating other queries using the same concepts. They allow to use previous compositions that met the requirements of the user and an SLA. To briefly give a very simplified idea of our integrated SLA, let us consider the example of query expressed above, and considering the time response facet, let us assume that two cloud services G1 and G2 are able to deliver information on green energy providers and one service E is able to give provided quantity of energy of the corresponding providers. Assume that composing the two kind of service that will meet the response time requirement will be G1 and E, the integrated SLA will mention that the suitable composition is $< S1, E >$.

III. Conclusion

Current big data settings impose 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 adapted to these 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.