The Christian Doppler Laboratory for Client-Centric Cloud Computing

Application-Oriented Fundamental Research

Klaus-Dieter Schewe[1,2], Károly Bósa[2], Harald Lampesberger[2]
Ji Ma[2], Boris Vleju[2]

[1] Software Competence Centre Hagenberg, Hagenberg, Austria
kd.schewe@scch.at
[2] Johannes-Kepler-University Linz, Christian-Doppler Laboratory for
Client-Centric Cloud Computing, Linz, Austria
[kd.schewe|k.bosa|h.lampesberger|j.ma|b.vleju]@cdcc.faw.jku.at

Abstract. Though cloud computing is considered mature for practical application, we outline the need for more research, and sketch the research agenda of the newly established Austrian Christian Doppler Laboratory for Client-Centric Cloud Computing.

1 The Case for Research in Cloud Computing

Currently, "cloud computing" is the most often used buzzword in computing, and many providers (Amazon, Google, Microsoft, etc.) of cloud services (IaaS, SaaS, PaaS, DaaS, . . .) emphasize the many benefits of outsourcing application into a (private or public) cloud. In other words, it is suggested that cloud computing represents a mature technology that is ready to be massively used. The recently established Christian Doppler Laboratory for Client-Centric Cloud Computing (CDCC) is a research centre dedicated to application-oriented fundamental research. As such it holds up the natural claim that cloud computing still requires lots of fundamental research.

In particular, most of the offerings in cloud computing are provider-centric. For instance, a client (or tenant) may rent a certain piece of infrastructure, load and execute a piece of software on it, pay for the use, and leave the cloud without leaving permanent traces. Certainly, there are many computing-intensive applications, e.g. web crawling, image processing, machine learning, etc. that fit well into such a scenario. However, if we think of a multi-user database application, its usefulness decreases significantly.

First, we have to cope with interaction, which implies massive transfer of data from and to a cloud leading to a performance problem. Second,

in such applications it is usually required that the use of the database is transparent to the database owner, i.e. credentials of user must be maintained by the client and not the cloud provider. Third, if several such applications are combined, the problem of cloud interoperability arises, for which the state of the art in cloud computing is not yet well prepared.

2   The Forgotten Tenant

Among many other problems in cloud computing the CDCC identified the lack of client-orientation as a serious problem that needs to be addressed in research. This subsumes the problems of identity of tenants, access rights, adaptivity to the needs of clients, and more. For instance, in the database application scenario above it would be indispensible to keep knowledge of users and their rights in the authority of the client instead of in the cloud. The immediate consequence of such an approach is that cloud applications become hybrid and distributed, as parts of data and software will reside on promise, while others reside in the cloud.

As adaptivity describes a property of a system to adapt itself to different contexts, the CDCC aims at adaptivity to preferences of clients, restrictions arising from channels, and specific requirements for end-devices, in particular mobile devices. Preferences of clients refer to the way they interact with cloud servers. This already applies to the selection of services, for which we need customer-specific preference rules giving weights to providers and quality of service attributes. Our aim is to investigate the full range of possible selection preferences and to come up with languages for the expression of selection preferences for clients.

Analogously, client preferences may impact on the way parts of a cloud are handled that are owned by them or for which a client has exclusive access rights. In this case preferences can affect the location and method of storage as well as access protection; clients may even have preferences for particular protocols that are to be applied in these cases. Here, we also want to investigate, which options should be made available for clients to express preferences.

Furthermore, client preferences affect the content, functionality and the presentation of selected services. We aim at extending the rewriting-based approaches to automatically adapt choices in plots [5], which at the moment are restricted to high-level specifications. In particular, this is expected to lead to adaptation algorithms that take a specification of a (composed) service and a set of preference rules as input and produce a modified service that respects the client preferences.

3   Contracting and Legal Aspects

The creation of applications, in which larger portions of software are services that are offered somewhere by some service provider, is not only a scientific and technological challenges, it also implies the need to handle the relationship between service providers and service users. In a service-centric system the use of a services replaces the purchase or lease of data or a software product. The use of other's hardware as a platform replaces the purchase and maintenance of hardware by clients. The use of software services also brings with it the benefit of not being bothered by new releases, bug fixes, etc., which become part of the service.

However, what can be expected by a service user requires contracts that state explicitly, what a service offers, when it will be available, which performance can be expected, which maintenance is guaranteed, which security mechanisms are in place, how privacy is guaranteed, and how much the service costs. It has to contain regulations that apply in case one of these assertions of service is violated. Such service contracts may replace licensing agreements. As such there must be different classes of contracts depending on whether individuals use the service or groups. It also has to prescribe whether the client or the provider takes care of group-specific regulations. On the other hand, contracts must also contain rules that prescribe health conditions on the side of the client, rules, the level of security and privacy the client has to guarantee when using the service.

Our aim is to investigate the full range of regulations that have to be handled in service contracts. Leaving aside purely legal regulations such as the responsible court in case of disagreement or claims, the question is, to what degree the ingredients of service contracts can become part of the service description, in which case they could become part of a QoS ontology.

4   Security and Privacy – What Clients Need

The first problem area in security and privacy is connected to the management of access rights. This leads to two research problems we intend to investigate. The first one is concerned with the checking of access rights, for which we intend to develop adequate methods. Though this appears to be rather straightforward, the remaining problem is to keep track of dependencies between group rights and rights of group members.

The second problem is dedicated to inferences on access rights. For this we have to take into account that ownership should imply access

rights, that membership in a group should imply that access rights of the group also apply to its members, that the right to execute a service operation should imply the right to execute the underlying view, that the right to use a composed service should imply the right to use the component services, etc.

More generally, access rights can be considered as permissions in a deontic logic. Therefore, we believe it is advisable to investigate not only permissions, but also prohibitions, obligations, and triggered actions. Thus, we aim at embedding access rights into a complex deontic logic and to study inferences in that logic. The goal s to ensure that all implied access rights must be explicitly granted. A particular difficulty arises from rights to grant access and revoke rights.

With respect to privacy tenants have to be protected against malicious users as well as against the cloud provider. With respect to the first of these hazards we pick up on the idea of specification of secrets, i.e. for data stored on a cloud it has to be specified who is permitted to see the data and who is not. Therefore, we first have to investigate data models in more detail in order to be able to identify at which granularity level such secret specifications should be applied. For instance, in case data is organised in relations, secrets may apply to records, clusters of records, or even individual attributes. For tree-based data organisation, e.g. in case XML is used as the data model, secrets might apply to subtrees rooted at particular elements, leaf elements and attributes, or aggregated tree portions that are defined by some query expression or algebra term.

Next we have to be aware that secrets may nonetheless be discovered by means of inferences. Therefore, we investigate, which queries or sequences of queries would be necessary to retrieve a secret that cannot be queried directly. We intend to also distinguish between exact detection of a secret, detection with a tolerable error, and detection with a high probability. These results should give an indication which access rights may need to be restricted to exclude the detection (or almost detection) of secrets by means of inferences. We will also investigate the alternatives of blocking certain queries, if the result could be used to discover secrets and the application of "lying strategies". However, also the rejection of queries and inconsistency of query results can be used as valuable information inferences can be based upon.

The second problem is somehow inverse, as we want to allow a customer to retrieve some data from a cloud – access rights are assumed to be granted – without being able to see the actual query. It is well known that anonymity can only be guaranteed in an efficient way, if data is replicated.

We therefore intend to study replication strategies and develop algorithms for query execution that preserve anonymity.

5 Epistemiological and Formal Foundations

Besides the forgotten tenant there is a serious lack of formal foundations in cloud computing starting from the simple fact that key notions such as service are not defined. Therefore, we address the fundamental research question how a uniform formal model for clouds must look like without any bias to particular languages and technology. We further investigate what a cloud has to offer to enable effective and efficient search for services as well as effective and efficient of multiple access to services by multiple tenants. The basic research component will build upon our previous research on Abstract State Services (AS^{2}s) described above, which has to be extended in various ways.

According to our understanding a cloud is primarily a repository of services, so we first have to specify the notion of service. The model of Abstact State Services (AS^{2}s) [4] starts from the assumption that a service should combine a hidden part, and a public part that is exported to service users. The hidden part is assumed to be a database-based transactional system, for which a universal model of database transformations [6] is adopted. The visual public part is represented by a collection of views, each of which is extended by a set of transaction-oriented service operations that link to the hidden database part. The whole model adopts the theory of Abstract State Machines (ASMs) [2], in particular referring to the so-called ASM thesis [3, 1].

References

1. A. Blass and J. Gurevich. Abstract state machines capture parallel algorithms. *ACM Transactions on Computational Logic*, 4(4):578–651, 2003.
2. E. Börger and R. Stärk. *Abstract State Machines*. Springer-Verlag, Berlin Heidelberg New York, 2003.
3. J. Gurevich. Sequential abstract state machines capture sequential algorithms. *ACM Transactions on Computational Logic*, 1(1):77–111, 2000.
4. H. Ma, K.-D. Schewe, B. Thalheim, and Q. Wang. A theory of data-intensive software services. *Service Oriented Computing and Its Applications*, 3(4):263–283, 2009.
5. K.-D. Schewe, B. Thalheim, and Q. Wang. Customising web information systems according to user preferences. *World Wide Web*, 12(1):27–50, 2009.
6. K.-D. Schewe and Q. Wang. A customised ASM thesis for database transformations, 2009. submitted for publication.