Brokering SLAs for end-to-end QoS in Cloud Computing

Tommaso Cucinotta, Diego Lugones, Davide Cherubini, Karsten Oberle
Bell Laboratories, Alcatel-Lucent, Dublin, Ireland
{*firstname.lastname*}*@alcatel-lucent.com*

Keywords: Quality of Service, SLAs, Cloud Computing, Cloud Broker

Abstract: In this paper, we present a brokering logic for providing precise end-to-end QoS levels to cloud applications distributed across a number of different business actors, such as network service providers (NSP) and cloud providers (CSP). The broker composes a number of available offerings from each provider, in a way that respects the QoS application constraints while minimizing costs incurred by cloud consumers.

1 INTRODUCTION

Cloud Computing introduces a novel model of computing that brings several technological and business advantages. Customers can rent cloud services on a pay-per-use model, without the need for big investments for resources that have to be designed for peak workloads, whilst being at risk of remaining under-utilized for most of the time. Providers may offer cloud services for rental, hosting them on big multi/many-core machines, where the infrastructure investments may be amortized over thousands of customers.

However, the requirements of cloud customers are evolving quickly, as cloud technology is being massively used worldwide. Many enterprise applications that might take advantage from the cloud model cannot be hosted on current infrastructures due to their stringent performance aspects requiring more and more from the best effort Internet. Think of virtual desktop, Network Function Virtualization (NFV), professional on-line multimedia editing and collaborative tools, and on-line gaming, just to mention a few.

Even though recent standards and research efforts deal with predictable QoS levels in Cloud Computing (Oberle et al., 2013), spanning across different business and operations domains remains a great challenge. For example, a single user request may have to traverse access, metro, core and data center networks, and the top-down provisioning chain across the various cloud layers (SaaS, IaaS, etc.), and still require tight interactivity. Figure 1 depicts the situation. Understanding how to combine all business actors in agreements to deliver end-to-end QoS may become overly difficult.

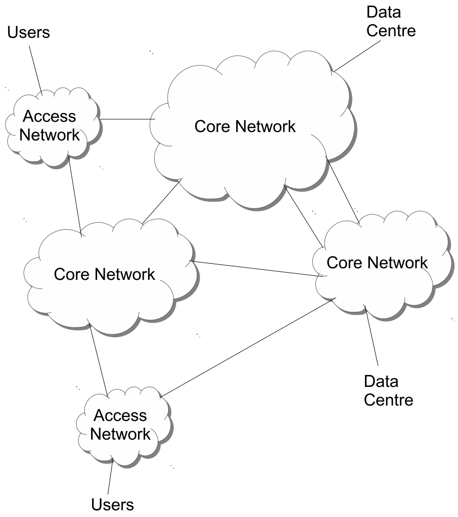

Figure 1: Users crossing multiple network service providers for accessing a cloud data center.

For these reasons, it is becoming increasingly important to have available intermediation services (a.k.a., brokerage) allowing cloud consumers to interface themselves to the multitude of service providers, including network carriers and cloud providers involved in the end-to-end service delivery chain. Brokering of cloud services (Plummer, 2012) allows for dealing with the various aspects of SLAs possibly involved in an end-to-end chain, including legal, economical and technological aspects. However, in this work we deal mainly with the latter aspects. That is, we consider cloud consumers who care about the price of a service and the expected and received end-to-end quality.

In this paper, we propose a brokering approach for deploying distributed cloud applications with guaranteed end-to-end QoS which is achieved as an end-to-end composition of SLAs to be established along the multiple players participating in the chain. We focus on the mathematical formalization of the brokering logic as a mixed-integer geometric programming program. Note that, albeit a prerequisite to the brokerage of cloud services is a certain level of interoperability and compatibility among the offerings, we do not address these specific issues in this paper. Namely, we assume our brokering logic is managing only services and offerings that can interchangeably be used and composed with each other, for deploying end-to-end applications. For an overview of the issues behind cloud interoperability, we refer the reader to (Petcu, 2011).

2 RELATED WORK

The problem of brokerage of multiple CSP offerings for hosting distributed cloud applications with minimum cost has been addressed in (Houidi et al., 2011). It has been modeled as a MILP, similarly to what is done in this paper. However, Houidi's model is too simple as it does not consider QoS requirements nor capacity constraints. Pawluk et al. defined a cloud broker service (Pawluk et al., 2012) (STRATOS) based on comparing different CSP offerings. However, the presented broker logic is simply based on configuration properties matching (e.g., minimum values to be satisfied on certain properties for the offered services). InterCloud (Buyya et al., 2010) and other architectures (Grivas et al., 2010) (Ferrer et al., 2012) for federation of cloud providers include a broker component that may be an excellent candidate for realizing the brokering logic described in this paper.

The IRMOS European Project has addressed how to enhance execution of real-time multimedia applications in virtualized Cloud infrastructures (Cucinotta et al., 2010; Kyriazis et al., 2011). IRMOS ensures resource allocation for individual hosted applications. However, one of its limitations is that it does not address how to possibly guarantee service levels across domain boundaries (i.e., whenever interacting with other providers and infrastructures).

The SLA@SOI EU Project[1] developed a framework (Wieder et al., 2011) for negotiation, provisioning, monitoring and adaptation of SLAs through the entire cloud service life-cycle. It includes both functional and non-functional characteristics of services, such as QoS constraints, which can be formalized through an XML-based syntax.

The ETICS (Economics and Technologies for Inter-Carrier Services) European Project[2] investigated on the critical issues for the creation of a new ecosystem of innovative QoS-enabled interconnection models between Network Service Providers (NSPs) impacting all of the actors involved in the end-to-end service delivery value-chain. ETICS produced a novel architecture (Zwickl and Weisgrab, 2013) for control and management of automated end-to-end delivery of QoS-enabled services across heterogeneous carrier networks.

Furthermore, many studies exist trying to identify critical factors affecting a market of cloud offerings, including considerations related to QoS-aware provisioning of resources and services. An overview of such approaches can be found in (Breskovic et al., 2013).

Concerning standarization, the European Telecommunications Standards Institute (ETSI) is currently involved in several activities related to the issues mentioned above with the ETSI NFV Industry and Specification group. The Technical Committee CLOUD is also addressing interoperability aspects of end-to-end applications. NIST is also working in this area, with a series of reports being of value to the topic of end-to-end cloud service quality[3], in addition to the well-known reports on the definition of cloud computing (Mell and Grance, 2011) and its reference architecture (Liu et al., 2011), where the importance of having cloud brokerage services is emphasized. Additional details on research and standardization efforts in this area can be found in (Oberle et al., 2013).

3 BROKERING OPTIMIZATION LOGIC

Notation and Assumptions. From an abstract viewpoint, a distributed real-time cloud computing application is modeled as a set of components $A = \{1, 2, \dots\}$, among which we include also client/user-side components. These are interconnected in a logical application topology (limited to be a linear workflow in this paper) expressed as the sequence $S \equiv (s_1, \dots, s_n)$ of n elements of A that activate one after another, whenever triggered by users requests, com-

[1] More information is available at: http://sla-at-soi.eu/.

[2] More information at: https://www.ict-etics.eu/.

[3] More information can be found at: http://www.nist.gov/itl/cloud/index.cfm.

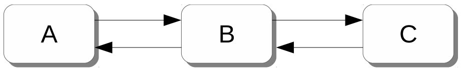

Figure 2: Simple application sequence graph.

ing at a minimum inter-arrival time T. Each activation i of component s_i is associated with its computing requirements C_i, representing the amount of time needed for (sequential) computations within the component for each request received from the user. After computations, the component sends a message of size M_i to the next component s_{i+1} in the sequence S. For example, consider a cloud application composed of 3 components $A = \{a, b, c\}$, where a might represent the client part of the application running at the consumer premises (e.g., a web browser), b might represent the front-end server (e.g., a webserver), and c might represent the back-end component (e.g., a database). Then, a sequence might look like $S = (a, b, c, b, a)$, where user requests traverse the components forward and backwards.

A cloud application has to be deployed over a set of cloud data centers $\mathcal{P}$ belonging to one or more Cloud Service Providers (CSPs), interconnected by means of network providers (NSPs). Each NSP is characterized by a set of Edge Routers (ERs) $\mathcal{R}$, through which it is connected either to other NSPs or to CSPs or directly to users. From a mathematical perspective, we consider in what follows data center locations as edge routers as well, resulting in $\mathcal{P} \subset \mathcal{R}$. The situation can be depicted as a graph where each node represents an ER which has arcs towards all the other ERs to which it is connected.

CSPs are assumed to be able to host one or more components i by reserving a certain amount of computing resources u_i to it with a minimum VM computing latency of L^C, and we assume an ideal scalability model: if the component is assigned computing capacity $u_i \in (0, U^{max}] \in \mathbb{R}$ with VM wake-up latency of L^C, then the time needed to complete a user request is reasonably approximated or bounded as: $\frac{C_i}{u_i} + L^C$. The VM computing latency may depend on many factors, such as hardware and software configuration of hypervisors and guest OSes, including the CPU schedulers type and their configuration.

Similarly, NSPs are assumed to be able to place the communication flow between each component i and the next one by dedicating a network bandwidth b_i and ensuring a maximum communication latency L^N, in such a way that the time needed to transmit a message among these components (assuming no queuing of messages from the same application/flow is needed) can be reasonably estimated/bounded as: $\frac{M_i}{b_i} + L^N$.

We assume that our brokering logic has available a set of quotations (or equivalently a price list) from all of these providers. More precisely, for each available data center $p \in P$, we have available a few alternate quotations R_p. Each quotation $r \in R_p$ includes the cost for deploying components at the CSP premises that can be made available within conditions of quotation r; this includes $K^C_{p,r}$, the cost of a computing capacity unit up to a maximum computing capacity $U^C_{p,r}$, and $K^N_{p,r}$, the cost of a networking capacity unit up to a maximum networking capacity of $U^N_{p,r}$; the quotation also includes a maximum VM access latency of $L_{p,r}$. For example, the same CSP might provide two different quotations with different latencies, depending on the type of hardware where the VM(s) would be deployed, and/or the hypervisor and guest OS kernel configuration(s) , and/or the level of consolidation of multiple VMs onto the same host pushed by the provider in the context of that quotation; also, the latency might depend on the network configuration within the data center, so the CSP might be willing to offer VMs at particularly chosen locations within the DC, and/or using particular configurations of, e.g., an SDN-enabled data center network, so as to guarantee lower access latencies from the outside, at a conveniently higher unit-cost; different quotations may equally well represent different “sizes” of VMs that can be seen in current CSPs IaaS offerings. Also, for each NSP connecting two data centers or edge routers $p, q \in \mathcal{R}$, we have available a few alternate quotations $R_{p,q}$. Each quotation $r \in R_{p,q}$ is expressed as a cost $K_{p,q,r}$ for deploying, with latency $L_{p,q,r}$, a unit of communication bandwidth between p and q, up to a maximum available bandwidth of $B_{p,q,r}$.

In cases where, among two data centers $p, q \in \mathcal{P}$, there are multiple NSPs connecting them, or there are multiple paths traversing different NSPs for connecting them, we assume that all the corresponding quotations are combined together into a single list $R_{p,q}$ combining together the available multiple quotations to traverse. For example, quotations from two NSPs to be traversed for connecting two locations $p, q \in \mathcal{P}$ can be combined as follows: for each pair of quotations r_1 from the first NSP and r_2 from the second NSP, we consider an equivalent quotation r_{r_1,r_2} in which the bandwidth-unit costs and the latencies of r_1 and r_2 area *added* together, whilst the maximum available bandwidth for the quotation is the *minimum* among the two. After this operation of aggregating quotations from different NSPs, we obtain equivalent end-to-end quotations for all pairs of data center locations $p, q \in \mathcal{P}$, thus in our formalization we can forget about the existence of the intermediate edge routers $\mathcal{R} \backslash \mathcal{P}$ among multiple NSPs carrying the traffic among all CSPs.

Problem Formalization. Now we proceed to formalize the cloud brokering problem as an optimization program. To this purpose, we introduce Boolean variables $x_{i,p,r}$ representing whether the broker will deploy component i onto data center location $p \in \mathcal{P}$ making use of quotation $r \in R_p$, and the Booleans $y_{i,p,q,r}$ representing whether or not traffic between components s_i and s_{i+1} in sequence S is to be deployed between locations $p, q \in \mathcal{P}$ by using quotation $r \in R_{p,q}$. These variables are tied to each other by the constraints:

$$\begin{aligned} \sum_{p \in \mathcal{P}} \sum_{r \in R_p} x_{i,p,r} &= 1 \, \forall s_i \in S & (1) \\ \sum_{r \in R_{p,q}} y_{i,p,q,r} &= \left(\sum_{r_1 \in R_p} x_{s_i,p,r_1} \right) \left(\sum_{r_2 \in R_q} x_{s_{i+1},q,r_2} \right) \\ & \quad \forall p, q \in \mathcal{P}, \forall s_i \in S \setminus \{s_n\} & (2) \end{aligned}$$

where: the first constraint forces each component to be placed onto a single cloud provider, exploiting one among the available quotations; the second one says that, if component s_i is deployed on p, and component s_{i+1} on q, then only one among the available quotations for communications between p and q has to be chosen; otherwise, $y_{i,p,q,r}$ must be 0 for each $r \in R_{p,q}$. Also, the deployment and quotations choice has to satisfy the overall end-to-end deadline D constraint:

$$\sum_{s_i \in S} \left(\frac{C_i}{u_i} + \sum_{p \in \mathcal{P}} \sum_{r \in R_p} L^C_{p,r} x_{i,p,r} + \frac{M_i}{b_i} + \sum_{p,q \in \mathcal{P}} \sum_{r \in R_{p,q}} L^N_{p,q,r} y_{i,p,q,r} \right) \leq D \quad (3)$$

and the allocation constraints

$$\begin{aligned} \sum_{s_i \in S} u_i x_{i,p,r} &\leq U_{p,r} \, \forall p \in \mathcal{P}, \, \forall r \in R_p & (4) \\ \sum_{s_i \in S} b_i y_{i,p,q,r} &\leq B_{p,q,r} \forall p, q \in \mathcal{P} \forall r \in R_{p,q} & (5) \end{aligned}$$

where u_i and b_i are real problem variables. These are further constrained due to the assumption that requests submitted to the same application workflow at the specified minimum inter-arrival time of T do not queue after each other, namely:

$$\begin{aligned} u_i &\geq \frac{C_i}{T} & (6) \\ b_i &\geq \frac{M_i}{T}. & (7) \end{aligned}$$

On the other hand, the broker will try to minimize the cost for hosting the application:

$$\begin{aligned} \min \sum_{s_i \in S} u_i \sum_{p \in \mathcal{P}} \sum_{r \in R_p} K_{p,r} x_{i,p,r} + & \quad (8) \\ \sum_{s_i \in S \setminus \{s_n\}} b_i \sum_{p,q \in \mathcal{P}} \sum_{r \in R_{p,q}} K_{p,q,r} y_{i,p,q,r}. & \quad (9) \end{aligned}$$

The formalized problem falls in the class of mixed-integer geometric programming optimization programs, for which there are solvers available, such those found in the GAMS suite[4].

4 IMPLEMENTATION CONSIDERATIONS

Network Service Provider. In order to provide end-to-end cloud services with precise QoS levels, NSPs and CSPs must implement selected routing paths and reserve appropriate resources. On the networking side, we distinguish between *intra-domain routing* – if the end-to-end path is entirely within the same Autonomous System (AS), and *inter-domain routing* if the traffic is sent across multiple NSPs.

In *intra-domain routing*, users and DCs where their applications are deployed are all interconnected via the same NSP. In order to compose different networking offers, the NSP can adopt different strategies to build the end-to-end path. For example, a *best effort latency* offer can be realized by adopting traditional IGP protocols (e.g., OSPF or IS-IS) to route traffic between source and destination on the shortest path. Of course, not always shortest path translates into minimum latency. On the other hand, a *premium* offer (or *minimum latency* offer) can be realized using Multi Protocol Label Switching (MPLS) with its Traffic Engineering extensions (MPLS-TE). In particular, a Label Switched Path (LSP) is a one-way tunnel that can transport the traffic from origin to destination and that can satisfy multiple QoS parameters (e.g., bandwidth, delay, jitter, availability, and loss). The LSP routing path can either be configured automatically (e.g., using Constrained Shortest Path First – CSPF), or manually, e.g., as a result of more sophisticated techniques (Cherubini et al., 2011).

In *inter-domain routing*, when users can reach CSP's DCs across different NSPs, optimal path calculation is inherently more complex. Each router should have a global view of the network. Unfortunately, NSPs "filter" important information needed for the path establishment (e.g., for scalability or confidentiality reasons). The Path Computation Element pre-

[4] More information is available at: `http://www.gams.com/solvers/`.

sented in (Farrel et al., 2006), represents a possible solution to provide an optimal inter-domain routing path (in the form of MPLS-TE LSP) that meets desired QoS requirements and can scale the end-to-end network up to 100,000 MPLS devices (Leymann et al., 2013).

Cloud Service Provider. The centralized design of today's data centers offers advantages in terms of fabric homogeneity and control, in comparison to the best-effort networks describedabove, which usually include various legacy equipment, specialized boxes, multiple protocols and are potentially operated by independent NSPs.

However, meeting the latency guarantees required by demanding Cloud applications in the data center fabric is still a significant challenge. The shared nature of the network in multi-tenant data centers leads to significant variations in the perceived performance. The lack of predictability increases the tenant costs and causes provider revenue loss. Moreover, the Internet-oriented protocols and flow scheduling mechanisms used in data centers are unaware of flow deadlines and application traffic patterns. Instead, they strive to optimize such low-level metrics as network throughput and fairness, ignoring the actual performance requirements associated to traffic flows.

Another reason that makes it difficult to deliver precise QoS levels in data centers is that typical designs focus on bisection bandwidth by overprovisioning the fabric. However, these are clearly not optimized for ultra-low latency applications with predictable delivery of packets. Moreover, overprovisioning today's data centers is prohibitively expensive.

The most common strategy is to implement QoS mechanisms by segregating traffic into different classes to provide isolation and enable traffic engineering. Typically, these mechanisms are implemented in switches and network cards where traffic is prioritized explicitly by marking packets or implicitly by using port ranges. In addition, the cloud services are typically run at low utilization to meet strict SLA demands, which decreases efficiency. However, advanced research in the field proposes different alternatives such as removing the kernel and network stack from the critical path of communication and load balancing requests across application instances (Kapoor et al., 2012). In (Ballani et al., 2011) authors propose to create virtual network abstractions to allow tenants to expose their network requirements. Proposal in (Alizadeh et al., 2012) run the network with near zero queueing and adaptively respond to congestion marks using DCTCP (Alizadeh et al., 2010).

Also, deadline-based scheduling of packets has been proposed as an alternative to TCP. In (Wilson et al., 2011), a control protocol is proposed that uses deadlines to achieve informed allocation of network bandwidth. In (Andrews, 2000), EDF scheduling is leveraged for providing probabilistic end-to-end guarantees to individual streams.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we addressed optimum brokering of cloud and carrier services accounting for end-to-end QoS requirements of cloud applications, and the availability of multiple offerings by CSPs and NSPs with different and heterogeneous quality levels corresponding to different price conditions. The problem has been mathematically modeled as a mixed-integer geometric programming optimization program that can be solved by available standard solvers. This work constitutes a first step towards the realization of evolved end-to-end cloud services offering to consumers precise, stable and reliable QoS levels across the heterogeneous, multi-provider "supply" chain that is involved in the process.

Our planned and ongoing future work in the area includes development of simplified versions of the introduced problem, as well as fast heuristic solvers for trading off accuracy of the solution versus solving time. Also, an implementation of the described technique is under way and its effectiveness will soon be evaluated by simulation, within our CloudNetSim framework (Cucinotta and Santogidis, 2013), and by real prototyping.

References

Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. (2010). Data center tcp (dctcp). *SIGCOMM Comput. Commun. Rev.*, 41(4):–.

Alizadeh, M., Kabbani, A., Edsall, T., Prabhakar, B., Vahdat, A., and Yasuda, M. (2012). Less is more: trading a little bandwidth for ultra-low latency in the data center. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, NSDI'12, pages 19–19, Berkeley, CA, USA. USENIX Association.

Andrews, M. (2000). Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *INFOCOM*, pages 603–612.

Ballani, H., Costa, P., Karagiannis, T., and Rowstron, A. (2011). Towards predictable datacenter networks. *SIGCOMM Comput. Commun. Rev.*, 41(4):242–253.

Breskovic, I., Brandic, I., and Altmann, J. (2013). Maximizing liquidity in cloud markets through standardization of computational resources. In *Proceedings of the 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, SOSE '13, pages 72–83, Washington, DC, USA. IEEE Computer Society.

Buyya, R., Ranjan, R., and Calheiros, R. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Hsu, C.-H., Yang, L., Park, J., and Yeo, S.-S., editors, *Algorithms and Architectures for Parallel Processing*, volume 6081 of *Lecture Notes in Computer Science*, pages 13–31. Springer Berlin Heidelberg.

Cherubini, D., Fanni, A., Mereu, A., Frangioni, A., Murgia, C., Scutellà, M. G., and Zuddas, P. (2011). Linear Programming Models for Traffic Engineering in 100% Survivable Networks under Combined IS–IS/OSPF and MPLS-TE. *Computers & Operations Research*, 38(12):1805–1815.

Cucinotta, T., Checconi, F., Kousiouris, G., Kyriazis, D., Varvarigou, T., Mazzetti, A., Zlatev, Z., Papay, J., Boniface, M., Berger, S., Lamp, D., Voith, T., and Stein, M. (2010). Virtualised e-learning with real-time guarantees on the irmos platform. In *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2010)*, pages 1–8, Perth, Australia.

Cucinotta, T. and Santogidis, A. (2013). Cloudnetsim - simulation of real-time cloud computing applications. In *Proceedings of the 4th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems*, Paris, France.

Farrel, A., Vasseur, J.-P., and Ash, J. (2006). RFC 4655 – A Path Computation Element (PCE)-Based Architecture.

Ferrer, A. J. et al. (2012). Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, 28(1):66 – 77.

Grivas, S., Kumar, T., and Wache, H. (2010). Cloud broker: Bringing intelligence into the cloud. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 544–545.

Houidi, I., Mechtri, M., Louati, W., and Zeghlache, D. (2011). Cloud service delivery across multiple cloud platforms. In *Services Computing (SCC), 2011 IEEE International Conference on*, pages 741–742.

Kapoor, R., Porter, G., Tewari, M., Voelker, G. M., and Vahdat, A. (2012). Chronos: predictable low latency for data center applications. In *Proceedings of the Third ACM Symposium on Cloud Computing*, SoCC '12, pages 9:1–9:14, New York, NY, USA. ACM.

Kyriazis, D., Menychtas, A., Kousiouris, G., Oberle, K., Voith, T., Boniface, M., Oliveros, E., Cucinotta, T., and Berger, S. (2011). A real-time service oriented infrastructure. *GSTF International Journal on Computing*, 1(2).

Leymann, N. et al. (2013). draft-ietf-mpls-seamless-mpls-04 – Seamless MPLS Architecture.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., and Leaf, D. (2011). Nist special publication 500-292 – nist cloud computing reference architecture.

Mell, P. and Grance, T. (2011). NIST SP800-145: The NIST Definition of Cloud Computing.

Oberle, K., Cherubini, D., and Cucinotta, T. (2013). End-to-end service quality for cloud applications. In *Proceedings of the 10th International Conference on Economics of Grids, Clouds, Systems and Services*, Zaragoza, Spain.

Pawluk, P., Simmons, B., Smit, M., Litoiu, M., and Mankovski, S. (2012). Introducing stratos: A cloud broker service. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 891–898.

Petcu, D. (2011). *Towards a Service-Based Internet*, volume 6994 of *Lecture Notes in Computer Science*, chapter Portability and Interoperability between Clouds: Challenges and Case Study, pages 62–74. Springer Berlin Heidelberg.

Plummer, D. (2012). Gartner, inc. – cloud services brokerage: A must-have for most organizations.

Wieder, P., Butler, J., Theilmann, W., and Yahyapour, R. (2011). *Service Level Agreements for Cloud Computing*, springer edition.

Wilson, C., Ballani, H., Karagiannis, T., and Rowtron, A. (2011). Better never than late: meeting deadlines in datacenter networks. *SIGCOMM Comput. Commun. Rev.*, 41(4):50–61.

Zwickl, P. and Weisgrab, H. (2013). Final ETICS architecture and functional entities high level design.