

Transfer as a Service: Towards a Cost-Effective Model for Multi-Site Cloud Data Management

Radu Tudoran, Alexandru Costan, Gabriel Antoniu

▶ **To cite this version:**

Radu Tudoran, Alexandru Costan, Gabriel Antoniu. Transfer as a Service: Towards a Cost-Effective Model for Multi-Site Cloud Data Management. Proceedings of the 33rd IEEE Symposium on Reliable Distributed Systems (SRDS 2014), IEEE, Oct 2014, Nara, Japan. hal-01023282

HAL Id: hal-01023282

https://inria.hal.science/hal-01023282

Submitted on 11 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Transfer as a Service: Towards a Cost-Effective Model for Multi-Site Cloud Data Management

Radu Tudoran*, Alexandru Costan†, Gabriel Antoniu*

*INRIA Rennes - Bretagne Atlantique, France
{radu.tudoran, gabriel.antoniu}@inria.fr

†IRISA / INSA Rennes, France
alexandru.costan@irisa.fr

***Abstract*—The global deployment of cloud datacenters is enabling large web services to deliver fast response to users worldwide. This unprecedented geographical distribution of the computation also brings new challenges related to the efficient data management across sites. High throughput, low latencies, cost- or energy-related trade-offs are just a few concerns for both cloud providers and users when it comes to handling data across datacenters. Existing cloud data management solutions are limited to cloud-provided storage, which offers low performance based on rigid cost schemas. Users are therefore forced to design and deploy custom solutions, achieving performance at the cost of complex system configurations, maintenance overheads, reduced reliability and reusability. In this paper, we are proposing a dedicated cloud data transfer service that supports large-scale data dissemination across geographically distributed sites, advocating for a Transfer as a Service (TaaS) paradigm. The system aggregates the available bandwidth by enabling multi-route transfers across cloud sites. We argue that the adoption of such a TaaS approach brings several benefits for both users and the cloud providers who propose it. For users of multi-site or federated clouds, our proposal is able to decrease the variability of transfers and increase the throughput up to three times compared to baseline user options, while benefiting from the well-known high availability of cloud-provided services. For cloud providers, such a service can decrease the energy consumption within a datacenter down to half compared to user-based transfers. Finally, we propose a dynamic cost model schema for the service usage, which enables the cloud providers to regulate and encourage data exchanges via a *data transfer market*.**

I. Introduction

With their globally distributed datacenters, cloud infrastructures enable the rapid development of large scale applications. Examples of such applications running as cloud services across sites range from office collaborative tools (Microsoft Office 365, Google Drive), search engines (Bing, Google), global stock market financial analysis tools to entertainment services (e.g., sport events broadcasting, massively parallel games, news mining) and scientific applications [1]. Most of the web-based applications are deployed on multiple sites to leverage proximity to users through content delivery networks. Besides serving the local client requests, these services need to maintain a global coherence for mining queries, maintenance or monitoring operations, that require large data movements. Studies show that the inter-datacenter traffic is expected to triple in the following years [2], [3]. This geographical distribution of computation becomes increasingly important for scientific discovery as well. Processing the large amounts of data (e.g., 40 PB per year) generated by the CERN LHC overpasses single site or single institution capacity, as it was the case for the Higgs boson discovery, where the processing was extended to the Google cloud infrastructure [4]. Accelerating the process of understanding data by partitioning the computation across sites has proven effective also in solving bio-informatics problems [5]. However, the major bottlenecks of these geographically distributed computations are the data transfers, which incur high costs and significant latencies [6].

Currently, the cloud providers' support for data management is limited to the cloud storage (e.g., Azure Blobs, Amazon S3). These storage services, accessed through basic REST APIs, are highly optimized for availability, enforcing strong consistency and replication [7]. Clearly, they are not well suited for end-to-end transfers, as this was not their intended goal: users need to upload data into the remote persistent storage, from where it becomes then available for download to the other party. In case of inter-site data movements, the throughput is drastically reduced by the high latency of the cloud storage and the low interconnecting bandwidth. Recent developments led to alternative transfer tools such as Globus Online [8] or StorkCloud [2]. Although such tools are more efficient than the cloud storage, they act as third party middleware, requiring users to setup and configure complex systems, with the overhead of dedicating some of the resources (initially leased for computation) to the data management. Our goal is to understand to what extent and under which incentives the inter-datacenter transfers can be externalized from users and be provided as a service by the cloud vendors.

In our previous work [9] we have proposed a user managed transfer tool that was monitoring the cloud environment for insights on the underlying infrastructure, used to choose the best combination of protocol and transfer parameters. In this paper, we investigate how such a tool can be "democratized" and offered transparently by the cloud provider, using a *Transfer as a Service (TaaS)* paradigm. This shift of perspective comes naturally: instead of letting users optimize their transfers by making (possible false) assumptions about the underlying network topology and performance through intrusive monitoring, we delegate this task to the cloud provider. Indeed, the cloud owner has extensive knowledge about its network resources, which it can exploit to optimize (e.g., by grouping) user transfers, as long as it provides a service to enable them. Our working hypothesis is that such a service will offer slightly lower performances than a highly-optimized dedicated user-based setup (e.g., based on multi-routing through extensive use of network parallelism) but substantial higher performance than todays' state-of-the-art transfer solutions (e.g., using the cloud storage or GridFTP). In turn, this approach has the advantage of freeing users from the burden of configuring and maintaining complex data management systems, while

providing the same availability guarantees as for any cloud managed service.

We argue that by adopting TaaS, cloud providers achieve a key milestone towards the new generation datacenters, expected to provide mixed service models for accommodating the business needs to exchange data [10]. In [11], the authors emphasize that the network and the system innovation are the key dimensions to reduce costs. Cloud providers rent the interconnecting bandwidth between datacenters from Tier 1 Internet Service Providers and get discounts based on the committed transfer levels [12]. Coupled with the flexible pricing schema that we propose, TaaS can regulate the demand and increase the number of users which move data. Enabling fast transfers through simple interfaces, as advocated by TaaS, cloud providers can therefore grow their outbound traffic and increase the associated revenues.

Our contributions can be summarized as follows:

- We introduce two user managed options for data transfers in the cloud (Section II)
- We propose an architecture for a dedicated cloud TaaS, targeting high performance inter site transfers, and we discuss its declinations (Section III)
- We perform a thorough comparison between the user- and the cloud- managed strategies in different scenarios, considering several factors that can impact the throughput (concurrency, data size, CPU load etc.) (Section IV)
- We propose a flexible pricing schema for the service usage, that enables a "data transfer market" (Sections V-A,V-B)
- We analyze the energy efficiency of user- versus cloud-managed inter site transfers (Section V-C)
- We provide an overview of the cloud data management solutions and their issues (Section VI)

II. Context of data management in the cloud

We first introduce the terminology used throughout this paper and present the existing user-managed options for data transfers.

A. The cloud ecosystem

Our proposal relies on the following concepts:

The datacenter (the site) is the largest building block of the cloud. Public clouds typically have tens of datacenters distributed in different geographical areas, each datacenter holding both storage and computation nodes. The compute infrastructure is partitioned in multiple fault domains, delimited by rack switches. The physical resources of a node are shared among several VMs, generally belonging to different users, unless the largest VM type is running, which is fully mapped to a physical node. Cloud providers do not own the backbone that interconnects datacenters; instead, they pay for the inter-site traffic to Tier 1 ISPs. Multiple network links interconnect the physical nodes within a site with the ISP infrastructure [11], for higher performance and availability.

The deployment is the virtual space that aggregates several VMs, in which a user application is executed. The VMs are placed on different compute nodes in separate fault domains. A load balancer distributes all external requests among these VMs. A deployment runs within a single data-center and the number of cores that can be leased within a deployment is often limited (e.g., in Azure that is 300 cores / deployment). This means that large scale applications, using several thousand cores, need to be distributed across multiple deployments, on multiple sites.

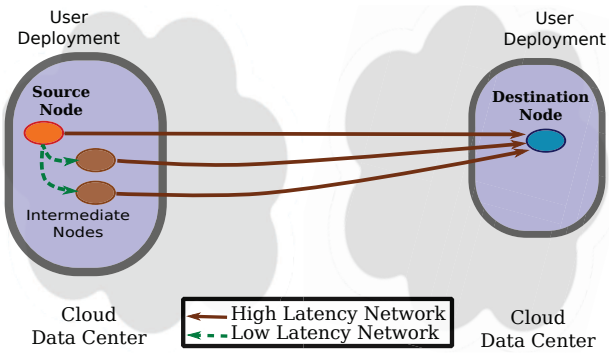


Fig. 1. Multi-route user transfers

The storage can be used as a high-latency intermediate for data transfers through some basic store (PUT) or retrieve (GET) operations. For inter-site transfers, choosing the best temporary storage location is not trivial. Putting data at the destination side, at the sender side or in-between is ambiguous and depends on the access pattern. Adding to the high latencies and low throughput this increased sensitivity to each particular transfer, the cloud storage is clearly an inefficient option for common transfers.

B. User-managed inter-site transfer scenarios

Users can set up their own tools to move data between deployments, through direct communication, without intermediaries, at higher transfer rates. They can adhere to two transfer strategies, depending on their cost and performance requirements:

Endpoint to endpoint solutions leverage the basic transfers from source to destination, regardless the technology used (e.g., GridFTP, scp, etc.). This baseline option is relatively simple to set in place, using the public endpoint provided for each deployment. The major drawback in this scenario is the low bandwidth between sites, which limits drastically the throughput that can be achieved.

Multi-route transfers. Building on the observations that a user application typically runs on multiple VMs and that communication between datacenters follows different physical routes, we have proposed in [9] a multi-route transfer strategy illustrated in Figure 1. Such a schema exploits the intra-site low-latency bandwidth to copy data to intermediate nodes within the deployment. Next, this data is forwarded towards the destination across multiple routes, aggregating additional bandwidth between sites. This approach is better suited for managing Big Data, but comes at an increased costs: the performance speedup is sub-linear with respect to the leased resources (i.e., N times more intermediate nodes do not provide N times faster throughput). The factors which limit the speedup are the (small) overhead of inner-deployment transfers and the congestions in ISP infrastructures.

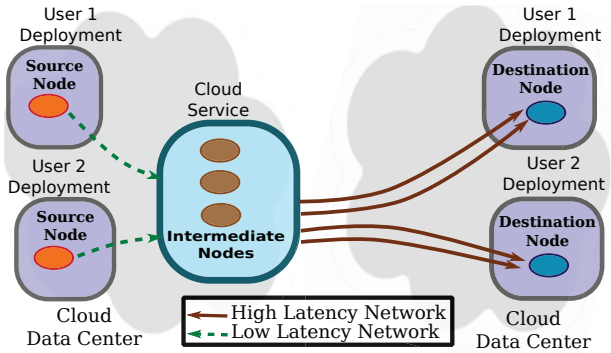


Fig. 2. An asymmetric Transfer as a Service approach

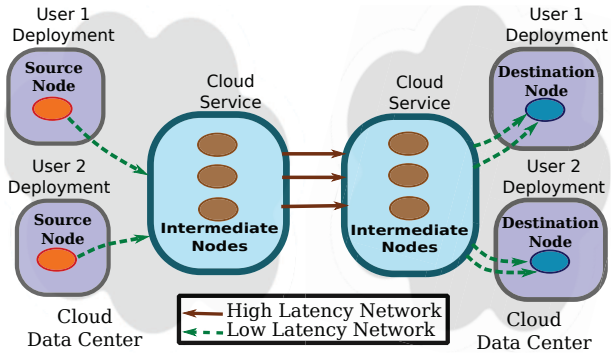


Fig. 3. A symmetric Transfer as a Service approach

The main issue with these user-managed solutions is that they are not available out-of-the-box. For instance, prohibiting factors to deploy the multi-route strategy range from the lack of user networking and cloud expertise to budget constraints. Applications might not tolerate even low intrusiveness levels linked to handling data in the intermediate nodes. Finally, scaling up VMs for short time periods to handle the transfer is currently strongly penalized by the VM startup times.

From the cloud provider perspective, having multiple users that deploy multi-route systems can lead to an uncontrolled boost of expensive Layer 1 ports towards the ISP [11]. Bandwidth saturation or congestion at the outer datacenter switches are likely to appear. The bandwidth capacity towards the Tier 1 ISP backbones, with a ratio of 1:40 or 1:100 compared to the bandwidth between nodes and Tier 2 switches, can rapidly be overwhelmed by the number of users VMs staging-out data. Moreover, activating many rack switches for such communications increases the energy consumption as demonstrated in Section V-C. Our goal is to find the right trade-off between the (typically contradicting) cloud providers economic constraints and users needs.

III. Zoom on the Transfer as a Service

We argue that a cloud-managed transfer service could substitute the user-based mechanisms without significant performance degradations. At the core of such a service lies a set of dedicated nodes within each datacenter, used by the cloud provider to distribute the transferred data and to further forward it towards the destination. As opposed to our previous approach, the dedicated nodes are owned and managed by the cloud provider, they no longer consume resources from the users deployments. Building on elasticity, the service can accommodate fluctuating user demands. Multiple parallel paths are then used for all chunks of data, leveraging the fact that the cloud routes packages through different switches, racks and network links. This approach increases the aggregated inter-datacenter throughput and is based on the empirical observation that intra-site transfers are at least 10x faster than the wide-area transfers.

The proposed architecture makes the service locally available to all applications within a datacenter, as depicted in Figure 2. The usage scenario consists in: 1) applications transferring data through the intra-site low-latency links to this service; and 2) the service forwarding the data across multiple routes towards the destination. The transfer process becomes transparent to users, as the configuration, operation and management are all handed to the cloud provider (*cloud-ified*), making it resilient to administrative errors.

When the TaaS approach is available at only one endpoint of the transfer, it can be viewed as an *asymmetric service*. This is often the case within federated clouds, where some providers may not propose TaaS. Users can still benefit from the service when migrating their data to computation instances located in different infrastructures. Such an option is particularly interesting for scientific applications which rely on hybrid clouds (e.g., scaling up the local infrastructure to public clouds). The main advantage with this architecture is the minimal number of hops added between the source deployment and the destination, which translates into smaller overheads and lower latencies. However, situations can arise when the network bandwidth between datacenters might still not be used at its maximum capacity. For instance, applications which exchange data in real-time can have temporary lower rates of transferred packages. Taking also into account that the connection to the user destination is direct, multiplexing data from several users is not possible. In fact, as only one end of the transmission over the expensive inter-site link is controlled by the cloud vendor, communication optimizations are not feasible. To enable them, the cloud provider should manage both ends of the inter-site connection.

We therefore advocate the use of the *symmetric solution*, in which TaaS is available at both transfer ends. This approach makes better use of the inter-datacenter bandwidth, and is particularly suited for transfers between sites of the same cloud provider. With this architecture, the TaaS is deployed on every datacenter and when an inter-site transfer is performed, the local service forwards the data to the destination service, which further delivers it to the destination node, as depicted in Figure 3. This approach enables many optimizations which only require some simple pairwise encode/decode operations: multiplexing data from different users, compression, deduplication, etc. Such optimizations, which were not possible with the asymmetric solution, can decrease the outbound traffic, to the benefit of both users and cloud providers. Moreover, the topology of the datacenter can now be taken into account by the cloud provider when partitioning the nodes of the service, such that load is balanced across the Tier 2 switches. Enabling this informed resource allocation has been shown to provide significant performance gains [13]. Despite the potential lower performance compared to the symmetric solution, due to the additional dissemination step at destination, this approach has the potential of bringing several operational benefits to the cloud provider.

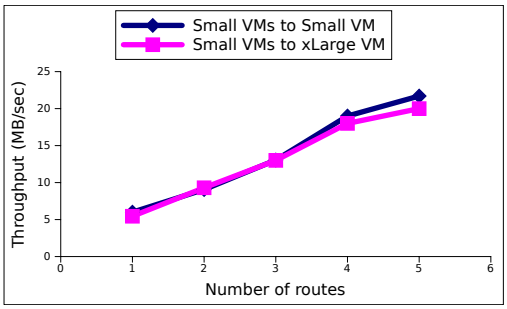


Fig. 4. Aggregated throughput from multiple routes towards different types of destination VMs

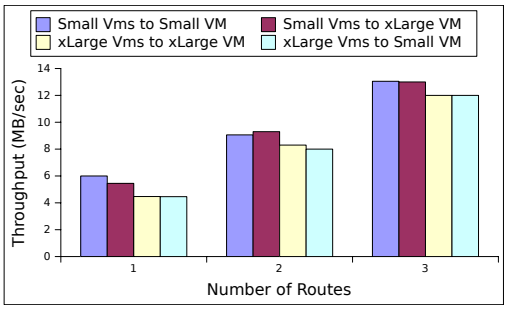


Fig. 5. The throughput of multiple routes with respect to different combinations of VM types

The service is accessed through a simple API, that currently implements *send* and *receive* functions. Users only need to provide a pointer to their data and the destination node to perform a high performance, resilient data movement. The API can be further enhanced to allow experienced users to configure several transfer parameters (e.g., chunk size, number of routes).

IV. Evaluation

In this section we analyze the performance of our proposal and compare it to user managed schemas through experiments focusing on realistic usage scenarios. The working hypothesis is that user based transfers are slightly more efficient but a cloud service can deliver comparable performance with less administrative overhead, lower costs and more reliability guarantees.

A. Experimental setup

The experiments were performed on the Microsoft Azure cloud, using two datacenters: North Central US, located in Chicago, and North Europe, located in Dublin, with data being transferred from US towards EU. These distant sites were selected in order to ensure a wide geographical setup across continents, with high-latency interconnecting links crossing the Atlantic ocean and communication paths across the infrastructures belonging to multiple ISPs. Considering the time zone differences between sites, the experiments are relevant both for typical user transfers and for cloud maintenance operations (e.g., bulk backups, inter-site replication). The latter are regularly performed by cloud providers and allow the TaaS approach to be further tuned in order to take into account the hourly loads of datacenters, as discussed in [3].

The cloud is used at the Platform as a Service (PaaS) level with Azure Web Roles running Small and xLarge VMs. The Small VM type is the elementary resource unit in Azure, offering 1 virtual CPU, mapped to a physical CPU, 1.75 GB memory, 225 GB local ephemeral storage. The xLarge VM type spans over an entire physical node, offering 8 virtual CPUs, 14 GB memory and 2 TB ephemeral local disk. From the network point of view, a physical node in Azure is connected through a 1 Gbps Ethernet card, meaning that an xLarge VM will benefit entirely from it, while a Small VM might get only one eighth of the network when other user VMs are deployed on the same physical node.

The experiments are performed by repeatedly transferring data chunks of 64 MB each from the memory. The intermediate nodes handle the data entirely in memory, both for user and cloud transfer configurations. For all experiments scaling up the number of resources, the amount of transferred data is increased proportionally, always handling a constant amount of data per intermediate node. The throughput is computed at the receiver side by measuring the time to transfer a fixed amount of data. Each sample is the average of at least 100 independent measurements.

B. User-manged multi-route transfers

We first discuss the throughput gains which can be achieved by users with multi-route transfer strategies. The performance shift is represented in Figure 4 based on the overall cumulative throughput of Small VMs when increasing the number of intermediate nodes. We notice that the gain obtained when scaling up to more nodes is asymptotically bounded as first discussed in [9]. However, our previous results do no conclusively show whether the performance bound is caused by a bottleneck at the destination side. To answer this question, we devise a new experiment in which the same sender setup is kept and the destination node is replaced by a xLarge VM, which has eight times more resources than the previously used Small instance. The results show a similar throughput, despite the extra resources, which means that the performance bound is not due to a bottleneck at the destination. This observation prevents wasting resources and increasing costs by using larger VMs when trying to increase the performance.

This finding raises a new question: is then the performance bounded due to the sender setup, the bandwidth between the datacenters or both? To answer it we continue by changing the sender VM type too and evaluate all resulting combinations: Small to Small, Small to xLarge, xLarge to Small and xLarge to xLarge. The results are presented in Figure 5. The number of intermediate nodes is 3, which is the upper limit of our resource subscription (i.e., 4 xLarge VMs = 32 cores). Nevertheless, at this point there is no need to go beyond that limit as we have already determined the performance trend in the experiment shown in Figure 4. Contrary to the expectations, using xLarge VMs at the sender does not improve the aggregated throughput. This shows that the interconnecting bandwidth within the sender datacenter has low impact on the overall transfer. However, the topology of the virtual network between the sender and intermediate nodes, scattered based on their size across different physical nodes, fault domains or racks, can increase the overhead for intra-site communication. Using Small VMs is therefore sufficient to aggregate the bandwidth between datacenters. We can conclude that the transfer performance is mainly determined by the number of distinct physical paths through which the packages are routed across the ISP infrastructures connecting the datacenters.

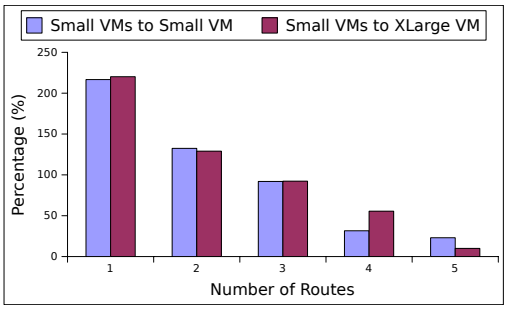


Fig. 6. The coefficient of variation for an increasing number of routes

Next, we focus on the variability with respect to multi-route transfers. Figure 6 shows the coefficient of variation (i.e., standard deviation/average%) for the throughput measurements in Figure 4. Surprisingly, using multiple paths decreases the otherwise high data transfer variability. This result is explained by the fact that with multiple routes the drops in performance on some links are compensated by bursts on others. The overall cumulative throughput, perceived by an application in this case, tends to be more stable. This observation is particularly important for scientific applications which build on predictability and stability of performance.

C. Evaluating the inter-site transfer options

We present in Figure 7 the comparison between the average throughput of the cloud transfer service and the user-based multi-route strategies. The experimental setup consists of 5 nodes per transfer service dedicated for data handling.

The *asymmetric solution* delivers slightly lower performance (∼16%) than a user-based multi-route schema. The first factor causing this performance degradation is the overhead introduced by the load balancer that distributes the incoming requests (i.e., from the application to the cloud service) between the nodes. The second factor is the placement of the VMs in the datacenter. For user-based transfers, the sender node and the intermediate nodes are closer rack-wise, some of them being even in the same fault domain. This translates into less congestion in the switches in the first phase of the transfer when data is sent to the intermediate nodes. For the cloud-managed transfers, the user source nodes and the cloud dedicated transfer nodes clearly belong to distinct deployments, meaning that they are farther apart with no proximity guarantees.

The *symmetric solution* is able to compensate for the previous performance degradation with the extra nodes at the destination site. The overhead of the additional hop with this approach is therefore neutralized when additional resources are provisioned by the cloud provider. The observation opens the possibility for differentiated cloud-managed transfer services in which different QoS guarantees are proposed and charged differently.

D. Dealing with concurrency

The experiment presented in Figure 8 depicts the throughput of an increasing number of applications using the transfer service in a configuration with 5 intermediate nodes. The goal of this experiment is to assess whether a sustainable quality of service can be provided to the user applications in a highly-concurrent context. Not surprisingly, an increase in the

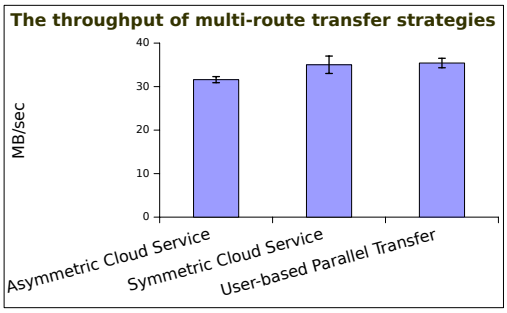


Fig. 7. The average throughput and the standard deviation with different transfer options with 5 intermediate nodes used to multi-route the packets.

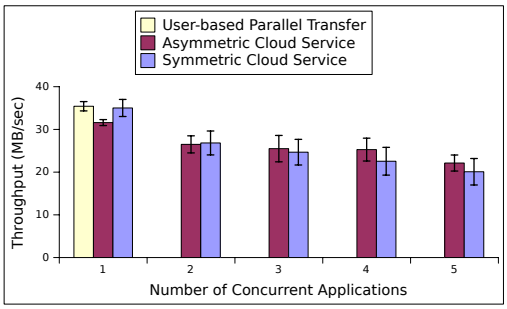


Fig. 8. The average throughput and the corresponding standard deviation for an increasing number of applications using the cloud transfer service concurrently

number of parallel applications decreases the average transfer performance per application with 25%. This is generated by the congestion in the transfers to the cloud service nodes and by the limit in the aggregated inter-site bandwidth that can be aggregated by these nodes. While this might seem a bottleneck for providing TaaS at large scale, it is worth zooming on the insights of the experiment to learn how such a performance degradation can be alleviated. We have scaled the number of clients up to the point where their number matches the number of nodes used for the transfer service. Hypothetically, we can consider having 1 node from the transfer service per client application. At this point the transfer performance delivered by the service per application is reduced, but asymptotically bounded, with less than 25% compared to the situation where only one application was accessing the service and all the 5 nodes where serving it. This shows that by maintaining a number of VMs proportional to the number of applications accessing the service, TaaS can be a viable solution and that it can in fact provide high performance for many applications in parallel.

We further notice that under increased concurrency, the performance of the symmetric solution drops more than in the case of the asymmetric one. This demonstrates that the congestion in handling data packets in the service nodes is the main cause of the performance degradation, since its effects are doubled in the case of the symmetric solution. At the same time, the aggregated throughput achieved by the applications using the transfer service would require 3 dedicated nodes from each of them (i.e., 15 nodes in total) compared to 5 or 10 nodes with the asymmetric or the symmetric solution, respectively. Deploying such services would make the inter-site transfers more energy efficient and the datacenters greener.

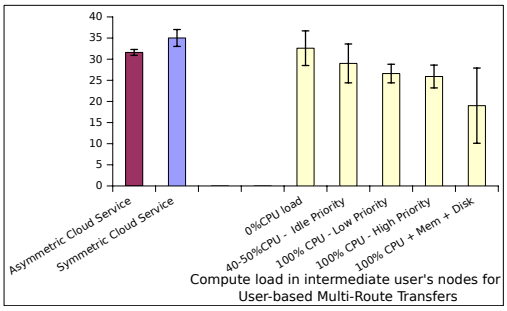


Fig. 9. Comparing the throughput of the cloud service against user-based multi-route transfers, using 4 extra nodes. The measurements depict the performance while the intermediate nodes are handling different CPU loads

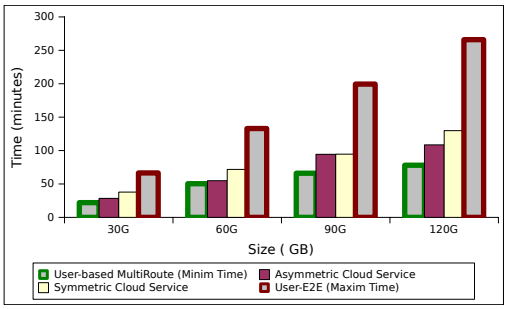


Fig. 10. The time to transfer large data sets using the available options. The user default Endpoint to Endpoint (E2E) option gives the upper bound, while the User-Based Multi Route offers the fastest transfer time. The cloud services, each based on 5 nodes deployments give intermediate performances, closer to the lower bounds

E. Towards a cloud service for inter-site data transfers

Not all applications afford to fully dedicate several nodes just for performing transfers. It is interesting to analyze to what extent, the computation load from the intermediate nodes can impact the performance of user-based transfers. We present in Figure 9 the evolution of the throughput when the computation done in the intermediate nodes has different CPU loads and execution priorities. All 100% CPU loads were induced using the standardized HeavyLoad tool [14], while the 40%-50% load was generated using system background threads which only access the memory.

Two main observations can be made based on the results shown in Figure 9. First, the throughput is reduced from 20% to 50% when the intermediate nodes are performing other computation in parallel with the transfers. This illustrates that the IO inter-site throughput is highly sensitive to the CPU usage levels. This observation complements the findings related to the IO behavior discussed in [15] for streaming strategies, in [16] for storing data in the context of HPC or in [17] for the TCP throughput with shared CPUs between several VMs. Second, the performance obtained by users under CPU load is similar, or even worse, to the one delivered by transfer service under increased concurrency (see Figure 8). This gives a strong argument for many applications running in the cloud to migrate towards such a TaaS offered by the cloud provider. Doing so, applications are able to perform high performance transfers while discharging their VMs from secondary tasks other than the computation for which they were rented for.

F. Inter-site transfers for Big Data

In the next experiment larger sets of data ranging from 30 GB to 120 GB are transferred between sites, using the cloud- and the user- managed options (grouped in 4 scenarios). The goal of this experiment is to understand the viability of the cloud services in the context of geographically distributed Big Data applications. The results are displayed in Figured 10 and 11. The experiment is relevant both to users and cloud providers since it offers concrete incentives about the costs (e.g., money, time) to perform large data movements in the cloud. To the best of our knowledge, there are no previous performance studies about the data management capabilities of the cloud infrastructures across datacenters.

Figure 10 presents the transfer times for the 4 scenarios. The baseline user endpoint to endpoint transfer gives very poor performance due to the low bandwidth between the datacenters. In fact, the resulting times can be considered as the upper bounds of user-based transfers (i.e., we do not consider here the even slower options like using the cloud storage). On the other hand, the user-based multi-route option is the fastest, and it can be considered as the lower bound for the transfer times. In-between, the cloud transfer service declinations are up to 20% slower than user-based multi-route but two times faster than the user baseline option.

In Figure 11 we depict the corresponding costs of these transfers. The costs can be divided in two components: the *compute cost*, paid for leasing a certain number of VMs for the transfer period and the *outbound cost*, which is charged based on the amount of data exiting the datacenter. Despite taking longer time for the transfer, the compute cost of the user-based endpoint to endpoint is the smallest as it only uses 2 VMs (i.e., sender and destination). On the other hand, user-based multi-route transfers are faster but at higher costs resulted from the extra VMs, as explained in Section II-B and detailed in [9]. The outbound cost only depends on the data volume and the cost plan. As the inter-site infrastructure is not the property of the cloud provider, part of this costs represent the ISP fees, while the difference is accounted by the cloud provider. The real cost (i.e., the one charged by the ISP) is not publicly known and depends on business agreements between the companies. However, we can assume that this is lower than the price charged to the cloud customers, giving thus a range in which the price can potentially be adjusted. Combining the observations about the current pricing margins for transferring data with the performance of the cloud transfer service, we argue that cloud providers should propose TaaS as an efficient transfer mechanisms with flexible prices. Cloud vendors can use this approach to regulate the outbound traffic of datacenters, reduces their operating costs, and minimising the idle bandwidth.

V. Discussion

This section analyses the potential advantages brought by a cloud service for inter-site data transfers. From the users perspective, TaaS can offer a transparent and easy-to-use method to handle large amounts of data. The service can sustain high throughput, close to the one achieved by users when renting and dedicating for the data handling alone at least 4-5 extra VMs. Besides avoiding the burden of configuring and managing extra nodes or complex transfer tools, the performance cost ratio can be significantly increased.

From the cloud providers points of view, such a service

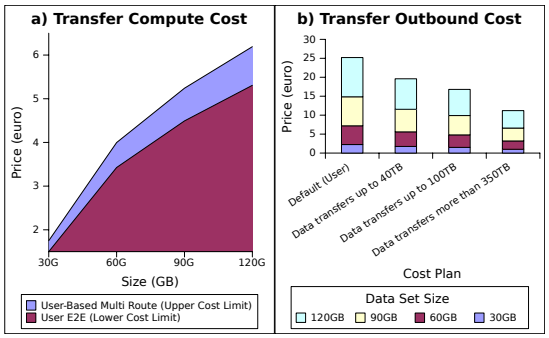


Fig. 11. The cost components corresponding to the transfer of 4 large data sets. a) The cost of the compute resources which perform the transfers, given by their lower and upper bounds. b) The cost for the outbound traffic computed based on the available cost plans offered by the cloud provider.

would give an incent to increase customer demand and brings competitive economical and energy advantages. TaaS extends the rather limited cloud data management ecosystem with a flexibly priced service, that supports a *data transfer market*, as explained in Sections V-A and V-B, and makes the datacenter greener, as shown in Section V-C.

A. Defining the cost margins for TaaS

In our quest for a viable pricing schema, we start by defining the cost structure of the transfer options. The price is based on the outbound traffic and the computation. The *outbound cost* structure is identical for all transfer strategies while the *computational cost* is particular to each option:

Outbound Cost:
$Size * Cost_{outbound}$, where $Size$ is the volume of transferred data and the $Cost_{outbound}$ is the price charged by the cloud provider for the traffic exiting the datacenter.

Computational Cost:

User-managed Endpoint to Endpoint
$time_{E2E} * 2 * Cost_{VM}$, where $time_{E2E}$ is the time to transfer data between the sender and the destination VMs. To obtain the cost, this has to be multiplied with the renting price of a VM: VM_{Cost}.

User-managed Multi-Route
$time_{UMR} * (2 + N_{extraVMs}) * Cost_{VM}$, where $time_{UMR}$ is the time to transfer data from the sender to the destination using $N_{extraVMs}$ extra VMs. As before, the cost is obtained by multiplying with the VM cost.

TaaS
$time_{CTS} * 2 * Cost_{VM} + time_{CTS} * service_{compute_{cost}}$, where $time_{CTS}$ is the transfer time and $service_{compute_{cost}}$ is the price charged by the cloud provider for using the transfer service. Hence, this cost is defined as the price for leasing the sender and destination VMs plus the price for using the service for the period of the transfer.

The computation cost paid by users ranges from the cheapest Endpoint to Endpoint option to the more performant, but more expensive, User-managed Multi-Route transfers. These costs can be used as lower and upper margins for defining a flexible pricing schema, to be charged for the time the cloud transfer service is used (i.e., $service_{compute_{cost}}$). Defining the service cost within these limits correlates with the delivered performance, which is between the same limits of the user-based options. To represent the $service_{compute_{cost}}$ as a function within these bounds, we introduce the following gain parameters, that describe the performance proportionality between transfer options: $time_{E2E} = a * time_{UMR} = b * time_{CTS}$ and $time_{CTS} = c * time_{UMR}$. Based on the empirical observations shown in Section IV, we can concretize the parameters with the following values: $a = 3$, $b = 2.5$ and $c = 1.2$. Rewriting the previous computation cost equation and simplifying terms, we obtain in Equation 1 the cost margins for the $service_{compute_{cost}}$.

$$2 * Cost_{VM} * (b-1) \leq service_{compute_{cost}} \leq Cost_{VM} * \frac{2 + N + 2 * c}{c} \quad (1)$$

Equation 1 shows that a flexible cost schema is indeed possible. Varying the cost within these margins, a *data transfer market* for inter-site data movements can be created, giving the cloud provider the mechanisms to regulate the outbound traffic and the demand, as discussed next.

B. Proposal for a data transfer market

Offering diversified services to customers in order to increase usage and revenues are among the primary goals of the cloud providers. We argue that these objectives can be fulfilled by creating a *data transfer market*. This can be implemented based on the proposed cloud transfer service offered at SaaS level with reliability, availability, scalability, on-demand provisioning and pay-as-you-go pricing guarantees. In Equation 1 we have defined the margins within which the service cost can be varied. We illustrate in Figure 12 these flexible prices for the two TaaS declinations (symmetric and asymmetric). The values are computed based on the measurements for transferring the large data sets mentioned in Section IV-F. The cost is normalized and expressed as the price charged when using the service (i.e., the compute cost component) to transfer 1 GB of data. A conversion between the *per hour* and the *per GB* usage is possible due to the stable performance delivered by this approach.

The minimal and maximal values in Figure 12 correspond to the user-managed solutions (i.e., Endpoint to Endpoint and Multi-Route). Between these margins, the cloud transfer service can model the price with a range of discretization values. The two TaaS declinations have different pricing schemas due to their performance gap, with the symmetric one being slightly less performant and having a lower price. As for the outbound cost, the assumption we made is that any outbound cost schema offered today brings profit to the cloud provider. Hence, we propose to extend the flexible usage pricing to integrate this cost component, as shown in Figure 13. The main advantage is that the combined cost gives a wider range in which the price can be adjusted. Additionally, it allows cloud providers to propose a unique cost schema instead of charging users separately for the service usage and for the outbound traffic.

A key advantage of setting up a *data transfer market* for this service is that it enables cloud providers to regulate the traffic. A simple strategy to encourage users to send data is to decrease the price towards the lower bounds shown in Figure 13 in order to reduce the idle bandwidth periods. A price drop would attract users which otherwise would send data by dedicating 4-5 additional VMs, with equivalent performance. Building on such costs and complementing the work described in [3], applications could buffer in VMs the less urgent data

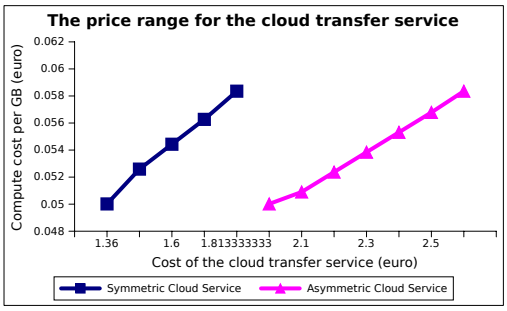


Fig. 12. The range in which the price for the cloud services can be varied.

and send it in bulks only during the discounted periods. On the other hand, when many users send data simultaneously, independently or using TaaS, the overall performance decreases due to switch and network bottlenecks. Moreover, the peak usage of outbound traffic from the cloud towards the ISPs grows, which leads to lower profit margins and penalty fees for overpassing the SLA quotas [12], [18], [19]. It is in the interest of the cloud providers to avoid such situations. With the flexible pricing, they have the means to react to such situations by simply increasing the usage price. With the high prices approaching the ones of user multi-route option, the demand can be temporarily decreased. At this point, it becomes more interesting for users to get their own VMs to handle data.

Adjusting the price strategy on the fly, following the demand, produces a win-win situation for users and cloud providers. Clients have multiple services with different price options, allowing them to pay the desired cost that matches their targeted performance. Cloud providers increase their revenues by outsourcing the inter-site transfers from clients and by controlling the traffic. Finally, TaaS can act as a proxy between ISPs and users, protecting the latter from price fluctuations introduced by the former; after all, cloud providers are less sensitive to price changes than users are, as discussed in [20].

C. The energy efficiency of data transfers

When breaking the operating costs of a cloud datacenter, the authors of [11] find that "over half the power used by network equipment is consumed by the top of rack switches". Such a rack switch connects around 24 nodes and has an hourly energy consumption of about 60W, while a server node consumes about 200W [21]. Our goal is to assess and compare the energy consumed in a datacenter when transferring data using the user-managed multi-route setup (E_{UMR} in Equation 2) and the cloud transfer service (E_{CTS} in Equation 3). We consider N_{App} applications, each using $N_{extraVMs}$ extra nodes to perform user-based multi-route transfers. For simplicity, we use the average transfer time ($time$) of applications, and then the total energy consumed by the application nodes and switches is:

$$E_{UMR} = (\frac{N_{App} * (2 + N_{extraVMs})}{24} * 60W/h + N_{App} * (2 + N_{extraVMs} * 200W/h)) * time \quad (2)$$

where the first part of the equation corresponds to the energy used by the rack switches in which the applications nodes are deployed and the last part gives the power used by the nodes.

$$E_{CTS} = (\frac{N_{App} * 2}{24} * 60W/h + N_{App} * (2 * 200W/h) + \frac{Nodes_{TaaS} * 60W/h}{24} + Nodes_{TaaS} * 200W/h) * time * c \quad (3)$$

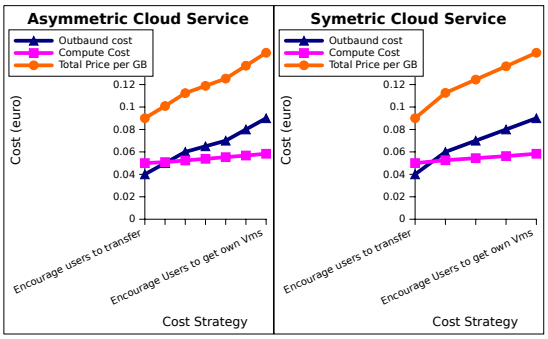


Fig. 13. Aggregating the cost components, outbound traffic cost and computation cost, into a unified cost schema for inter-site traffic

where the total energy is the sum of: 1) the energy used at the application side (i.e., the sender, destination nodes and the rack switches they activate) and 2) the energy consumed at the transfer service side, by the nodes which operate it (i.e., $Nodes_{TaaS}$) and the switches connecting them.

Comparing the two scenarios, we obtain in Equation 4 the generic ratio for the extra energy used when each user is handling his data on its own:

$$\frac{User - based_{energy}}{TaaS_{energy}} = \frac{N_{App} * (2 + N_{extraVMs})}{(2 * N_{App} + Nodes_{TaaS}) * c} \quad (4)$$

When we illustrate this for the configurations used in the evaluation section ($N_{extraVMs} = 5$, $N_{App} = Nodes_{TaaS}$ and $c = 1.2$), we notice that twice more energy is consumed if the transfers are done by users.

D. One more thing: reliability

A cloud managed transfer service has the advantage of being always available, in line with the reliability guarantees of all cloud services. Requests for transfers are carried over network paths that the cloud provider constantly monitors and optimizes for both availability and performance. This allows to quickly satisfy peaks in demand with rapid deployments and increased elasticity. Cloud providers ensure that a TaaS system incorporates service continuity and disaster recovery assurances. This is achieved by leveraging a highly available load-balanced dedicated nodes-farm to minimize downtime and prevent data losses, even in the event of a major unplanned service failure or disaster. Predictable performance can be achieved through strict uptime and SLAs guarantees.

User managed solutions typically involve hard-to-maintain scripts and unreliable manual tasks, that often lead to discontinuity of service and errors (e.g., incompatibility between new versions of some building blocks of the transfer framework). These errors are likely to cause VM failures and, currently, the period while a VM is stopped or is being rebooted is charged to users. With a TaaS approach, both the underlying infrastructure failures and the user errors are isolated from the transfer itself: they are transparent to users and are not charged to them. This allows to automate file transfer processes and provides a predictable operating cost per user over a long period.

VI. Related work

The landscape of cloud data transfers is rather rich in *user managed* solutions, spanning from basic tools for end-to-end communication (e.g., scp, ftp, GridFTP) to complex systems that support large-scale data movements for workflows

and scientific applications (e.g., GlobusOnline, Stork, Frugal). The common denominator of these solutions is their need to be deployed, fully configured and managed by users, with potentially little networking knowledge. Meanwhile, the only viable *cloud provided* alternative is the use of the cloud storage, which incurs large latencies and is subject to additional costs. To the best of our knowledge, our proposal is the first attempt to delegate the intra- and inter- cloud data transfers from users to the cloud providers, following a Transfer as a Service paradigm.

The handiest option for handling data distributed across several datacenters is to rely on the existing *cloud storage services*. This approach allows to transfer data between arbitrary endpoints via the cloud storage and it is adopted by several systems in order to manage data movements over wide-area networks [22], [23]. There is a rich storage ecosystem around public clouds. Cloud providers typically offer their own object storage solutions (e.g., Amazon S3 [24], Azure Blobs [7]), which are quite heterogeneous, with neither a clearly defined set of capabilities nor any single architecture. They offer binary large objects (BLOBs) storage with different interfaces (such as key-value stores, queues or flat linear address spaces) and persistence guarantees, usually alongside with traditional remote access protocols or virtual or physical server hosting. They are optimized for high-availability, under the assumption that data is frequently read and only seldom updated. Most of these services focus on data storage primarily and support other functionalities essentially as a "side effect" Typically, they are not concerned by achieving high throughput, nor by potential optimizations, let alone offer the ability to support different data services (e.g., geographically distributed transfers). Our work aims is to specifically address these issues.

Besides storage, there are few *cloud-provided services* that focus on data handling. Some of them use the geographical distribution of data to reduce latencies of data transfers. Amazon's CloudFront [25], for instance, uses a network of edge locations around the world to cache copy static content close to users. The goal here is different from ours: this approach is meaningful when delivering large popular objects to many end users. It lowers the latency and allows high, sustained transfer rates. However, this comes at the cost and overhead of replication, which is considerable for large datasets, making it inappropriate for simple end-to-end data transfers. Instead, we don't use multiple copies of data, but rather exploit the network parallelism to allow per transfer optimizations.

The alternative to the cloud offerings are the transfer systems that users can choose and deploy on their own, which we will generically call *user-managed solutions*. A number of such systems emerged in the context of the GridFTP [26] transfer tool, initially developed for grids. Among these, the work most comparable to ours is Globus Online [27], which provides high performance file transfers through intuitive web 2.0 interfaces, with support for automatic fault recovery. However, Globus Online only performs file transfers between GridFTP instances, remains unaware of the environment and therefore its transfer optimizations are mostly done statically. Several extensions brought to GridFTP allow users to enhance transfer performance by tuning some key parameters: threading in [28] or overlays in [29]. Still, these works only focus on optimizing some specific constraints and ignore others (e.g., TCP buffer size, number of outbound requests). This leaves the burden of applying the most appropriate settings effectively to users. In contrast, we propose a shift of paradigm and demonstrate the advantages of having an optimized transfer service provided by the cloud provider, through a simple and transparent interface.

Other approaches aim at improving the throughput by exploiting the network and the end-system parallelism or a hybrid approach between them. Building on the *nework parallelism*, the transfer performance can be enhanced by routing data via intermediate nodes chosen to increase aggregate bandwidth. Multi-hop path splitting solutions [29] replace a direct TCP connection between the source and destination by a multi-hop chain through some intermediate nodes. Multi-pathing [30] employs multiple independent routes to simultaneously transfer disjoint chunks of a file to its destination. These solutions come at some costs: under heavy load, per-packet latency may increase due to timeouts while more memory is needed for the receive buffers. On the other hand, *end-system parallelism* can be exploited to improve utilization of a single path. This can be achieved by means of parallel streams [31] or concurrent transfers [32]. Although using parallelism may improve throughput in certain cases, one should also consider system configuration since specific local constraints (e.g., low disk I/O speeds or over-tasked CPUs) may introduce bottlenecks. More recently, a *hybrid* approach was proposed [33] to alleviate from these. It provides the best parameter combination (i.e., parallel stream, disk, and CPU numbers) to achieve the highest end-to-end throughput between two end-systems. One issue with all these techniques is that they cannot be ported to the clouds, since they strongly rely on the underlying network topology, unknown at the user-level (but instead exploitable by the cloud provider).

Finally, one simple alternative for data management involves *dedicated tools* run on the end-systems. Rsync, scp, ftp are used to move data between a client and a remote location. However, they are not optimized for large numbers of transfers and require some networking knowledge for configuring, operating and updating them. BitTorrent based solutions are good at distributing a relatively stable set of large files but do not address scientists' need for many frequently updated files, nor they provide predictable performance.

VII Conclusion

This paper introduces a new paradigm, Transfer as a Service, for handling large scale data movements in federated cloud environments. The idea is to delegate the burden of data transfers from users to the cloud providers, who are able to optimize them through their extensive knowledge on the underlying topologies and infrastructures. We propose a prototype that validates these design principles through the use of a set of dedicated transfer VMs that further aggregate the available bandwidth and enable multi-route transfers across geographically distributed cloud sites. We show that this solution is able to effectively use the high-speed networks connecting the cloud datacenters and bring a transfer speed-up of up to a factor of 3 compared to state-of-the-art user tools. At the same time, it enables a reduction to half of the energy fingerprint for the cloud providers, while it sets the grounds for a data transfer market, allowing them to regulate the data movements.

Thanks to these encouraging results, we plan to further investigate the benefits of TaaS approaches both for users and cloud providers. In particular, we plan to study new cost models that allow users to bid on idle bandwidth and use it when their bid exceeds the current price, which varies in real-time based on supply and demand. We also see a good potential to use our prototype to study the performance of inter-datacenter or inter-cloud transfers. We believe that cloud providers could leverage this tool as a metric to describe the performance of network resources. As a further evolution, they could provide Introspection as a Service to reveal information about the cost of internal cloud operations to relevant applications.

REFERENCES

[1] "Azure Succesful Stories," http://www.windowsazure.com/en-us/case-studies/archive/.

[2] T. Kosar, E. Arslan, B. Ross, and B. Zhang, "Storkcloud: Data transfer scheduling and optimization as a service," in *Proceedings of the 4th ACM Workshop on Scientific Cloud Computing*, ser. Science Cloud '13. New York, NY, USA: ACM, 2013, pp. 29–36.

[3] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 74–85.

[4] "Cloud Computing and High-Energy Particle Physics: How ATLAS Experiment at CERN Uses Google Compute Engine in the Search for New Physics at LHC," https://developers.google.com/events/io/sessions/333315382.

[5] A. Costan, R. Tudoran, G. Antoniu, and G. Brasche, "TomusBlobs: Scalable Data-intensive Processing on Azure Clouds," *Journal of Concurrency and computation: practice and experience*, 2013.

[6] B. Cho and I. Gupta, "Budget-constrained bulk data transfer via internet and shipping networks," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*, ser. ICAC '11. New York, NY, USA: ACM, 2011, pp. 71–80.

[7] B. Calder and et al, "Windows azure storage: a highly available cloud storage service with strong consistency," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, 2011.

[8] I. Foster, R. Kettimuthu, S. Martin, S. Tuecke, D. Milroy, B. Palen, T. Hauser, and J. Braden, "Campus bridging made easy via globus services," in *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond*, ser. XSEDE '12. New York, NY, USA: ACM, 2012, pp. 50:1–50:8.

[9] R. Tudoran, A. Costan, R. Wang, L. Bougé, and G. Antoniu, "Bridging data in the clouds: An environment-aware system for geographically distributed data transfers," in *Proceedings of the 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2014)*, ser. CCGRID '14. IEEE Computer Society, 2014. [Online]. Available: http://hal.inria.fr/hal-00978153

[10] T. Bishop, "Data center 2.0 a roadmap for data center transformation," in *White Paper*. http://www.io.com/white-papers/data-center-2-roadmap-for-data-center-transformation/, 2013.

[11] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[12] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani, "How many tiers?: Pricing in the internet transit market," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 194–205, Aug. 2011.

[13] R. Tudoran, A. Costan, and G. Antoniu, "Datasteward: Using dedicated compute nodes for scalable data management on public clouds," in *Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, ser. TRUSTCOM '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 1057–1064.

[14] "HeavyLoad," http://www.jam-software.com/heavyload/.

[15] R. Tudoran, K. Keahey, P. Riteau, S. Panitkin, and G. Antoniu, "Evaluating streaming strategies for event processing across infrastructure clouds," in *Proceedings of the 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2014)*, ser. CCGRID '14. IEEE Computer Society, 2014.

[16] M. Dorier, G. Antoniu, F. Cappello, M. Snir, and L. Orf, "Damaris: How to Efficiently Leverage Multicore Parallelism to Achieve Scalable, Jitter-free I/O," in *CLUSTER - IEEE International Conference on Cluster Computing*, 2012.

[17] S. Gamage, R. R. Kompella, D. Xu, and A. Kangarlou, "Protocol responsibility offloading to improve tcp throughput in virtualized environments," *ACM Trans. Comput. Syst.*, vol. 31, no. 3, pp. 7:1–7:34, Aug. 2013.

[18] P. Hande, M. Chiang, R. Calderbank, and S. Rangan, "Network pricing and rate allocation with content provider participation," in *INFOCOM 2009, IEEE*, April 2009, pp. 990–998.

[19] S. Shakkottai and R. Srikant, "Economics of network pricing with multiple isps," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1233–1245, Dec. 2006.

[20] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani, "How many tiers?: Pricing in the internet transit market," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 194–205.

[21] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proceedings of the 2008 Conference on Power Aware Computing and Systems*, ser. HotPower'08. Berkeley, CA, USA: USENIX Association.

[22] T. Kosar and M. Livny, "A framework for reliable and efficient data placement in distributed computing systems." *Journal of Parallel and Distributed Computing 65, 10*, p. 11461157, Oct. 2005.

[23] P. Rizk, C. Kiddle, and R. Simmonds, "Catch: a cloud-based adaptive data-transfer service for hpc." in *Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium*, 2011, p. 1242 1253.

[24] "Amazon S3," http://aws.amazon.com/s3/.

[25] "Amazon CloudFront," http://aws.amazon.com/cloudfront/.

[26] W. Allcock, "GridFTP: Protocol Extensions to FTP for the Grid." *Global Grid ForumGFD-RP, 20*, 2003.

[27] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster, "The globus striped gridftp framework and server," in *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, ser. SC '05. Washington, DC, USA: IEEE Computer Society, 2005.

[28] W. Liu, B. Tieman, R. Kettimuthu, and I. Foster, "A data transfer framework for large-scale science experiments," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 717–724.

[29] G. Khanna, U. Catalyurek, T. Kurc, R. Kettimuthu, P. Sadayappan, I. Foster, and J. Saltz, "Using overlays for efficient data transfer over shared wide-area networks," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 47:1–47:12.

[30] C. Raiciu, C. Pluntke, S. Barre, A. Greenhalgh, D. Wischik, and M. Handley, "Data center networking with multipath tcp," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: ACM, 2010, pp. 10:1–10:6.

[31] T. J. Hacker, B. D. Noble, and B. D. Athey, "Adaptive data block scheduling for parallel tcp streams," in *Proceedings of the High Performance Distributed Computing, 2005. HPDC-14. Proceedings. 14th IEEE International Symposium*, ser. HPDC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 265–275.

[32] W. Liu, B. Tieman, R. Kettimuthu, and I. Foster, "A data transfer framework for large-scale science experiments," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 717–724.

[33] E. Yildirim and T. Kosar, "Network-aware end-to-end data throughput optimization," in *Proceedings of the first international workshop on Network-aware data management*, ser. NDM '11. New York, NY, USA: ACM, 2011, pp. 21–30.