IEEE Transactions on Cloud Computing 7.1 (2019): 34 – 47

DOI: https://doi.org/10.1109/TCC.2015.2469666

Cost-aware Multi Data-Center Bulk Transfers in the Cloud From a Customer-Side Perspective

José Luis García-Dorado[1] and Sanjay G. Rao[2]

[1]Department of Electronics and Communications Technology, Universidad Autónoma de Madrid, Spain.
[2]Department of Electrical and Computer Engineering, Purdue University, USA.

NOTE: This is a version of an unedited manuscript that was accepted for publication. Please, cite as:
J.L. García-Dorado and S.G. Rao. Cost-Aware multi data-center bulk transfers in the cloud from a customer-side perspective. IEEE TRANSACTIONS ON CLOUD COMPUTING, 7 (1), 34-47, (2019).
The final publication is available at: `https://doi.org/10.1109/TCC.2015.2469666`

Abstract

Many cloud applications (e.g., data backup and replication, video distribution) require dissemination of large volumes of data from a source data-center to multiple geographically distributed data-centers. Given the high costs of wide-area bandwidth, the overall cost of inter-data-center communication is a major concern in such scenarios. While previous works have focused on optimizing the costs of bulk transfer, most of them use the charging models of Internet service providers, typically based on the 95^{th} percentile of bandwidth consumption. However, public Cloud Service Providers (CSP) follow very different models to charge their customers. First, the cost for transmission is flat and depends on the location of the source and receiver data-centers. Second, CSPs offer discounts once customer transfers exceed certain volume thresholds per data-center. We present a systematic framework, CloudMPcast, that exploits these two aspects of cloud pricing schemes. CloudMPcast constructs overlay distribution trees for bulk-data transfer that both optimizes dollar costs of distribution, and ensures end-to-end data transfer times are not affected. CloudMPCast monitors TCP throughputs between data-centers and only proposes alternative trees that respect original transfer times. After an extensive measurement study, the cost savings range from 10% to 60% for both Azure and EC2 infrastructures, which potentially translates to millions of dollars a year assuming realistic demands.

Index Terms: Cloud Service Providers; Data-Center MultiCast; Volume Discounts; Heterogeneity Discounts.

1 Introduction

The past few years have witnessed an explosion in the popularity of cloud computing services. A key advantage of cloud computing is the ability to geo-distribute data over multiple data-centers, both to increase availability and reliability as well as provide lower user latency —low latencies are critical to business revenues, for example, Amazon estimated every 100 ms of latency costs 1% in sales [1].

There are numerous examples of research efforts and successful commercial applications that distribute content across the globe to achieve good

performance [2, 3, 4, 5]. Netflix is a prominent example of an application that disseminates its contents over a Cloud Service Provider (CSP) infrastructure to offer on-demand media services. In addition to significant reductions in latencies, geo-replication has become important to ensure high availability despite failures —e.g., for disaster recovery purposes. Further, other applications including software distribution, virtual machines cloning, distributed databases, and data warehousing may require geo-replication.

The common denominator of all these applications is that a large amount of that data must be disseminated to multiple data-centers. To get a sense of data volumes involved, consider recent surveys [6] which have shown that more than 77% of data-center operators run both backup and replication applications among three or more sites. Further, more than half of the interviewed operators indicated that they had more than a PB of data in their primary location. Specifically, 70% of the surveyed IT firms have between 1 and 10 Gb/s running between data-centers, nearly half having 5 Gb/s or more —i.e., between 330 TB and 3.3 PB a month. Disseminating these large volumes of data is prohibitively expensive given the high costs of inter data-center bandwidth [7, 8, 9].

There has been much effort in the research community directed at studying cost-effective bulk data transfer over the Internet [10, 11, 12]. Most of these works are in the context of popular Internet Service Provider (ISP) pricing models which bill clients based on the 95^{th} percentile of bandwidth usage. These works then focus on how to schedule bulk data transfer so the 95^{th} percentile usage is not significantly increased. However, the bandwidth pricing models of public CSPs' radically differs from ISPs. Most public CSPs charge clients a flat cost per byte for traffic outbound from cloud data-centers. Inbound traffic and intra-data-center traffic is typically free. Optimization techniques proposed under the 95^{th} percentile usage model therefore do not directly apply.

We make two observations about CSP pricing policies. First, the cost per byte varies widely based on the source data-center, and whether the recipient is an internal EC2 data-center, or an external data-center —e.g., data may be replicated across multiple cloud providers, or the Cloud and an on-premise data-center. For example, within Amazon EC2 infrastructure, the cost per GB ranges from \$0.02 to \$0.16 based on the source data-center for communication within EC2 —prices are more expensive for external receivers. Second, CSPs often offer discounts once the outbound traffic of a customer in a specific data center exceeds some fixed threshold over a month. These two observations open opportunities to define cheaper distribution trees for multi-point transfers than the trivial one of sending data from the data-center source to each of the destinations. First, as pricing costs across data-centers are heterogeneous, intuitively it would be cheaper to transmit once from the source to the lowest cost data-center, and then, retransmit to other destinations. Secondly, as volume discount are applied on a data-center basis, concentrating routes in a subset of data-centers will increase the likelihood of such discounts being applied.

In this paper, we present a systematic framework, named CloudMPcast, that exploits the above-mentioned unique aspects of pricing models of CSPs. CloudMPcast runs in each data-center of a deployment and works as a routing planner, that is, it constructs alternative distribution trees for bulk-data transfer to the trivial solution. CloudMPcast formulates the problem of finding such a cost-efficient distribution structure as an optimization problem. The proposed framework both reduces dollar costs of distribution and ensures that end-to-end data transfer times are not affected.

We conduct a detailed trace-driven simulation study using pricing models and bandwidth data obtained from Amazon EC2 and Microsoft Azure infrastructures, two of the major CSPs [13]. The bandwidth data was obtained through measurements of inter-data-center bandwidth between all EC2 and Azure data-centers. Our results show that CloudMPcast's savings range from 10% to 60% for both Azure and EC2 infrastructures, which potentially translate to savings of million dollars a year. Our extensive sensitivity studies show the benefits hold over a variety of data-center and traffic models. The results also point to the critical importance of exploiting both volume discounts and heterogeneity to work well in a variety of settings.

2 Related work

The Internet community has proposed several mechanisms to cut operational expenditures of online service companies many of which may also apply to CSPs. In contrast to these works, our focus is on reducing costs of customers of CSPs.

Prior work [14] has pointed out that the cost of electricity varies with time of day and across locations, and explored moving computation to data-centers that are cheaper at a given time. Similarly, some studies focus on how ISP providers charge wholesale clients for bandwidth [15]. Specifically, researchers [10] have proposed to carry out data backups and other bulk data transfer tasks during off-peak hours when usage is lower. This approach was further improved [16] by splitting backup activity into chunks and leveraging software defined networking (SDN).

Several works [10, 11, 12] have designed bulk data transfer schemes assuming ISPs charge their clients for bandwidth following a 95^{th} percentile usage model. That is, given a time series that represents the bandwidth used by clients, the bill corresponds to the 95^{th} percentile sample regardless of traffic at lower percentiles. These works [10, 11, 12] propose to transfer data when usage is below the 95^{th} percentile as no additional charges would be incurred. Moreover, the authors in [12] showed that not only bulk transfers, but also multimedia content with different quality requirements can be forwarded over an overlay topology so as to minimize the probability of exceeding pre-estimated percentiles. In contrast, we focus on bandwidth pricing models employed by CSPs, which are very different from ISP pricing models [15]. CSPs neither charge according to the time of day nor follow a percentile charging model, but instead charge based on the location of source and destination data-centers, and provide volume discounts.

Trial-and-better [17] is a mechanism to improve the operation of customers of CSPs. This work found significant heterogeneity in CPU, network, memory, and disk performance across instances of the same VM size, and proposed a mechanism to exploit this fact. The authors propose to identify and retain the best performing VMs, and discard the remaining ones. Thus, customers of a CSP may reduce the total number of VMs used, and consequently costs, while achieving desired performance. In contrast, the focus of CloudMPcast is on distributing data across multiple cloud data-centers in a manner that minimizes costs.

SPANStore [5] is a geo-replicated storage system that seeks to minimize costs for multiple consistency objectives. Among other techniques, SPANStore leverages heterogeneity in pricing of inter-data-center traffic based on source location. Our work is distinguished by the fact we consider volume discounts, which we are the first to explore to our knowledge. In many scenarios where pricing is relatively more homogeneous, use of volume discounts is critical to ensuring better cost savings. Volume discounts also ensures greater cost savings even in scenarios where pricing heterogeneity exists.

Further, in contrast to SPANStore which targets latency-sensitive replication, our focus is bulk transfers, TCP throughputs and transfer times. As will be shown, CloudMPcast's ILP formulations are different and consider minimum bandwidth along a path rather than the sum of latencies. Our evaluations include measurement studies of inter-data-center bandwidth, and the results will show that data-centers with lower network costs also exhibit better network throughput which opens the possibility to exploit heterogeneity discounts without impacting performance. Finally, our detailed trace driven simulations helps to systematically explore the benefits of pricing heterogeneity and volume discounts in a variety of scenarios.

3 Problem statement

We begin by explaining how CSPs charge their customers for data transmissions. We focus on EC2 and Azure, although most CSPs follow similar schemes. Next, we detail the opportunities that users have to reduce their data transfer bills in distributed and multi-cloud deployments.

3.1 CSP pricing policy for bandwidth

CSPs charge their customers for data transfers both to other data-center and the Internet by counting the number of GBs each customer transmits per data-center during a month and multiplying by its cost. The cost rates are published in the terms and

Table 1: EC2 and Azure pricing ($/GB on monthly basis) as of 2014. Internal refers to connections inside EC2. For the rest, the header shows the volume discount thresholds

CSP	Data-centers	Internal	0/10/50/150/500 TB
EC2	Virginia,California Oregon,Ireland	0.02	0.12/0.09/0.07/0.05
	Tokyo	0.09	0.201/0.158/0.137/0.127
	Singapore	0.09	0.19/0.15/0.13/0.12
	Sydney	0.14	0.19/0.17/0.15/0.14
	SaoPaulo	0.16	0.25/0.23/0.21/0.19
Azure	Amsterdam,Ireland California,Virginia	·	0.12/0.09/0.07/0.05
	HongKong,Singapore	·	0.19/0.15/0.13/0.12

conditions sheets of each CSP. Table 1 shows Azure and EC2 pricing in $/GB on a monthly basis as of 2014.

The rates depend on the location of the data-center and on the total volume a customer transmits in a month. The price per GB decreases as customer volume increases. Transfers within a data-center or inbound traffic to a data-center is typically free. Additionally, EC2 makes a distinction between intra-EC2 traffic and traffic to the Internet. Specifically costs are lower if destination is another EC2 data-center and no volume discounts apply here. Azure does not pay attention to destinations and divides its data-centers into two regions according to bandwidth costs. Region 1 data-centers have lower outbound traffic costs than those located in Region 2.

The volume discounts are triggered after a given threshold has be exceeded. These thresholds are 10, 50 (after other 40 TB), 150, and 500 TB for both providers. We remark that discount applies on a data-center basis, not on the total traffic generated by a given deployment.

3.2 Opportunities for customers

The trivial solution for multi-point transfers consists of transmitting from the data-center source to each of the destinations individually using the CSP's infrastructure. Consequently, charges are the result of summing the transfer cost associated with each source-destination pair. This transfer cost for a pair is itself obtained by computing the product of the total data transmitted and the cost per GB for that pair.

We point out two opportunities to reduce the dollar costs incurred by such a trivial approach: heterogeneity and volume discounts. To illustrate these opportunities, consider Fig. 1, which depicts a four data-center topology of an EC2 customer. The small boxes represent services running in different data-centers that transmit data between them. The largest boxes represent a software module or a VM that runs CloudMPcast which communicates with other processes in the same data-center for free and at high rates [18]. The cost for transmitting is heterogeneous across data-centers as shown previously. In this example, Virginia is cheaper than Singapore, Singapore is cheaper than Tokyo, and Sao Paolo is the most expensive data-center.

Heterogeneity discount: Assume that an application in Tokyo (Node T in Fig. 1) needs to replicate a piece of information of 1 GB to the other three data-centers. The trivial routing consists of transmitting once to each of the destinations which results in a cost equal to three times the pricing for outbound traffic in data-center T. This is shown in the left part of Fig. 1. However, consider an overlay routing solution where T sends data to Virginia (V), and then V forwards to the other two destinations. The cost of this operation is the sum of transmitting once from T to V and twice the cost of transmitting the same amount of data from V, since V in turn transmits to Sao Paulo (B) and Singapore (S). As the Virginia data-center is cheaper than Tokyo, there is a net savings in data transmission costs. Similarly, assume that now the source is Singapore and it is not necessary to replicate the information to Virginia. With the trivial solution, the costs are twice the cost for transmitting from S. However, there is another possibility: S may transmit to V, and then from V to both T and

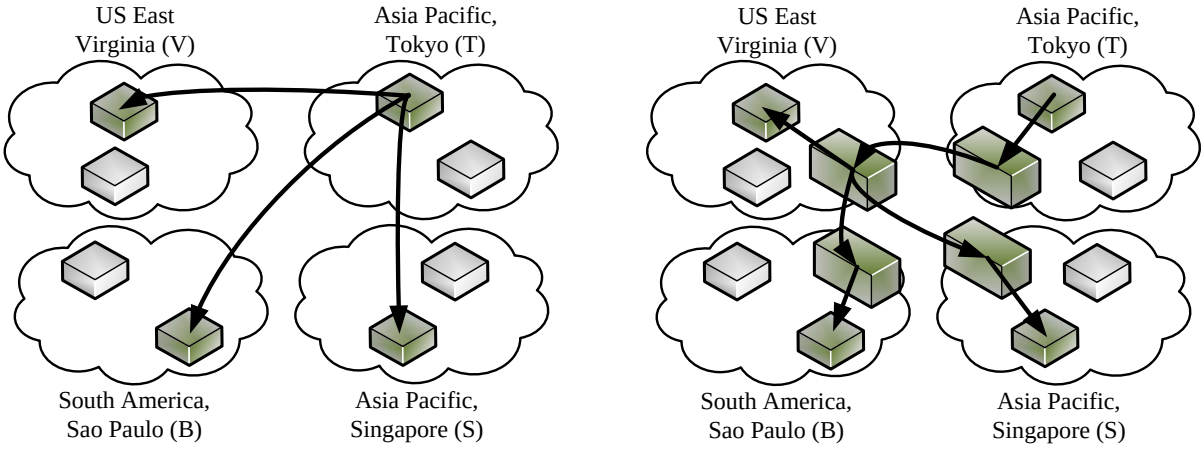

Figure 1: Trivial transfer solution (left) and CloudMPcast's overlay transfer proposal (right) for an example deployment

B. This option would be preferable if cost of two transmissions from Virginia added to the costs of a single transmission from Tokyo is cheaper than the costs of a single transmission from Tokyo. This is indeed the case in EC2. Nodes such as Virginia which are not a destination but useful in reducing the costs are often referred to as Steiner nodes as we discuss in the following section. Note that when the number of nodes and destinations increases the opportunities to find discounts also increase. The alternative solutions are shown in the right part of Fig. 1.

Volume discount: The other opportunity that arises from the pricing policies of Azure and EC2 is related to volume discounts. The idea is to aggregate as much data as possible in a given data-center in order to achieve further volume discounts for future transmissions. Assume a deployment similar to that shown in Fig. 1 but comprising EC2 data-centers in Virginia, California, Oregon and Ireland. These data-centers have the same costs of transmission. Even in this case, the trivial approach can be significantly improved upon. Specifically, we propose to have each source transmit its data to a hub data-center, which in turn transmits data to other receivers. By concentrating data in the hub data-center, volume discounts may be triggered sooner than with the trivial solution.

Minimizing costs without impacting transfer times: In this paper, we seek to exploit these two opportunities to reduce costs, but not at the expense of performance. We consider overlay routing to reduce costs only if the estimated time to complete a transmission is equivalent to the time that the trivial solution requires to transfer to the same destination. In practice, this means that a data-center may act as a retransmitter if and only if the bandwidth between the retransmitter and the following data-center in a given path is comparable to the direct bandwidth between the source and the destination. This ensures that the required time of transmission is equivalent to the trivial solution We refer to this problem as multi-point transfers in the Cloud with reliable transfer times.

4 CloudMPcast

We have developed CloudMPcast, a system that leverages the opportunities for cost reduction described in the previous section. We discuss the different modules of CloudMPcast, and the rationale behind them in the rest of this section.

4.1 System overview

CloudMPcast is an overlay system that executes in each data-center of an application deployment, where each application/tenant instantiates its own deployment of CloudMPCast. The key component is a routing planner, which computes the most cost-

efficient transmission plan (possibly the trivial solution) while observing the transfer time, for any given application request. Once a transmission plan is generated, CloudMPcast contacts the other instances of CloudMPcast in the data-centers involved in the solution to configure the overlay routing topology.

Each CloudMPcast node monitors the TCP throughput between itself and other data-centers in the deployment. CloudMPcast periodically (every 5 minutes) updates its inter data-center bandwidth matrix using well-known techniques [10, 19] to inform computations for future transfers. Our measurements on real cloud deployments (Section 5) indicate that inter data-center bandwidth is relatively stable over time-scales longer than individual bulk transfers. However, if significant throughput changes are detected which indicates that an overlay distribution tree previously computed for an ongoing transfer is no longer performing well from a transfer time perspective, CloudMPcast aborts the rest of the transfer. The destination data-centers directly contact the source data-center to obtain data not yet received. To reduce system complexity, CloudMPcast does not dynamically create alternate overlay distribution plans for an ongoing bulk transfer.

Since CloudMPcast may forward traffic to hub data-centers, a potential concern is whether this may induce network congestion at those data-centers and wide-area network in general, resulting in a reduction in the real throughput. Since CloudMPcast only manages requests of a given application/tenant, the total number of flows managed by CloudMPcast is small compared to the total number of flows competing for bandwidth in the network core. Consequently, a TCP flow introduced by CloudMPCast is unlikely to impact wide-area network bottlenecks [10, 20, 5]. However, in the unlikely case that this occurs, the adaptation of the mechanisms above introduced can help to handle the situation.

It is possible that the VMs corresponding to CloudMPcast in the hub data-center itself may be rate limited, resulting in a bottleneck at that VM. CloudMPCast could deal with this by using larger sized VMs, or allocating more VMs [19, 21]. In the limit, CloudMPCast could use separate VMs per transfer —e.g., separate source VMs in Virginia could be used for transfers to Tokyo and Singapore. Notice that VM costs are small compared to the costs of inter data-center bandwidth, and hence our formulations focus on bandwidth costs.

4.2 Abstraction and formulation

The problem faced by CloudMPcast's planner when addressing heterogeneity discounts is related to the traditional Steiner Tree Problem [22]. Let $\mathcal{G}$=($\mathcal{V}$,$\mathcal{E}$,$\mathcal{C}$) be a graph comprising a set of vertices $\mathcal{V}$, edges $\mathcal{E}$, and cost per edge $\mathcal{C}$. Given a subset $\mathcal{D} \in \mathcal{V}$, the Steiner Tree Problem consists of finding a tree $\mathcal{T}$=($\mathcal{V}_\mathcal{T}$,$\mathcal{E}_\mathcal{T}$,$\mathcal{C}_\mathcal{T}$) that spans $\mathcal{D}$ such that $\sum \mathcal{C}_\mathcal{T}$ is minimum. Variations of the Steiner Tree Problem have been studied by the networking community in the context of Internet multicast. These problems typically differ from the Steiner Tree problem in that edges are directed, costs are not symmetric, and one vertex is considered the source. Thus given a source s, and a set of destinations $\mathcal{D}$, the multicast problem is to find a subset of edges $\mathcal{S}$ that provide a path between s and each node in $\mathcal{D}$ such that the sum of the costs of edges in $\mathcal{S}$ is minimum. This problem has been proven to be NP-complete and the existence of an approximation with a constant performance guarantee is as unlikely as $P = NP$ [22].

Problem: The problem that CloudMPcast seeks to solve is a specific case of the multicast problem in which (i) the solution has to guarantee that end-to-end data transfer times are not affected with respect to a given trivial solution; and (ii) edge costs are not fixed as they depend on the volume of data previously transmitted. Further CloudMPcast works on a full mesh network given that there exists an edge from each data-center to the others.

Definitions: Let O_{ij} denote a direct connection between data-centers i and j using the CSP's infrastructure. Let $E2E_{sd}$ denote an end-to-end-path between data-centers s and d, i.e., a sequence of direct connections, O_{ij}, starting at s and ending at d such that $i \neq j$, and there are no repeated connections. Then, a distribution tree, $\mathcal{DT}\{s, \mathcal{D}, \mathcal{V}\}$, is defined as a set of end-to-end paths between data-center s and a set of destinations $\mathcal{D}$, $\{s, \mathcal{D}\} \in \mathcal{V}$, such that for each destination $d \in \mathcal{D}$ there exists an end-to-end-path between s and d.

The trivial solution to our problem is one where the source data-center transfers directly to each of destinations. Thus, we define the solution to

the multi-point transfers in the Cloud with reliable transfer times problem, $\mathcal{MPT\_RT}$, as any distribution tree such that it offers comparable end-to-end transfer times to the trivial solution for each of the destinations. Note that it is possible that the only solution to this problem is the trivial one.

Let W_d denote the measured bandwidth from s to destination using the direct connection O_{sd}. A distribution tree, $\mathcal{DT}\{s, \mathcal{D}, \mathcal{V}\}$, is a solution to the $\mathcal{MPT\_RT}$ problem if for each of the destinations, $\forall d \in \mathcal{D}$, the TCP throughput of each link O_{ij} in the end-to-end path from s to d ($E2E_{sd}$), is comparable to W_d. Specifically, we require that the throughput of O_{ij} is at least $\alpha \cdot W_d$, where $(0 < \alpha \leq 1)$.

Our aim is to find the solution to the $\mathcal{MPT\_RT}$ that minimizes the cost. We first present an ILP [23] for the problem assuming only heterogeneous costs. Volume discounts are considered in the next section.

ILP formulation: Let x_{ijd} and e_{ij} be two binary LP variables. Then, the formulation to $\mathcal{MPT\_RT}$ with minimal cost is as follows:

$$\text{Minimize} \quad \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} e_{ij} \cdot f^c(i,j) \quad \text{with respect to } e, x \tag{1}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{V}} x_{ijd} - \sum_{j\in\mathcal{V}} x_{jid} = 1, \ i = s, \ \forall d \in \mathcal{D} \tag{2}$$

$$\sum_{j\in\mathcal{V}} x_{ijd} - \sum_{j\in\mathcal{V}} x_{jid} = -1, \ i = d, \ \forall d \in \mathcal{D} \tag{3}$$

$$\sum_{j\in\mathcal{V}} x_{ijd} - \sum_{j\in\mathcal{V}} x_{jid} = 0, \ i \in \mathcal{V} \setminus \{s, d\}, \ \forall d \in \mathcal{D} \tag{4}$$

$$x_{ijd} \leq e_{ij}, \ \forall \{i,j\} \in \mathcal{V}, \forall d \in \mathcal{D} \tag{5}$$

$$\alpha W_d \cdot x_{ijd} \leq L_{ij}, \ \forall \{i,j\} \in \mathcal{V}, \forall d \in \mathcal{D} \tag{6}$$

$$x_{ijd} \in \{0,1\}, \quad e_{ij} \in \{0,1\} \tag{7}$$

where x_{ijd} indicates if the direct connection O_{ij} is used as part of the $E2E_{sd}$ path. Intuitively, if a given O_{ij} is part of at least one $E2E_{sd}$ $\forall d \in \mathcal{D}$, e_{ij} would be one indicating that such O_{ij} belongs to the $\mathcal{DT}$ solution. If O_{ij} is not part of any E2E path, O_{ij} does not belong to the solution and e_{ij} is zero. $f^c(i,j)$ represents the function that returns the cost per GB taking into account CSP's pricing policies for transmitting from data-center i to j. As $f^c(i,j)$ is multiplied by e_{ij} in Eq. 1, a cost is added by each O_{ij} included in the solution. Note that the cost is only considered once regardless of the number of $E2E$ paths that include a given O_{ij}. Finally, L_{ij} represents the TCP capacity as measured by CloudMPcast' probe module from data-center i to j.

Constraints: Constraints (2–4) ensure that there exists a $E2E$ path between data-center source, s, and each of destinations d. Constraint (5) ensures that if at least one pair s-d uses a direct connection O_{ij} for an $E2E$ path, this connection is used by the distribution tree, and it will represent a cost. Constraint (6) ensures that delivery transfer times per destination are met. Specifically, it ensures that if O_{ij} is used as part of an $E2E_{sd}$ path, its TCP capacity L_{ij} is at least $\alpha \cdot W_d$.

4.3 Volume discounts

When volume discounts come into play, the costs incurred for a given request depend on both the routing decisions made for prior requests and the current volume to transmit. Specifically, the cost per GB may be represented as $f^c(i, j, B, \mathbf{C}, \mathbf{T}, P_i)$ where B accounts for the traffic volume to transmit, and P_i is the amount of data sent from node i in the past. $\mathbf{T}$ and $\mathbf{C}$ respectively are the volume thresholds for different source and destination pairs, and the cost per GB for different thresholds. More concretely, $T_{ijt} \in \mathbf{T}$ denotes the t^{th} threshold interval in TBs for transmissions from data-center i to j, and C_{ijt} denotes the per-GB costs for the corresponding threshold levels. As an example, assume a deployment that involves the two data-centers located in Virginia, one of EC2 (i) and the other of Azure (j). According to Table 1, T_{ij} is (10 50 150 500 ∞) in TBs and C_{ij} is (0.12 0.09 0.07 0.05) in \$.

An optimal solution to the problem with volume discounts requires knowledge of all future requests, which is not realistic [24]. Therefore, we propose a simple greedy heuristic that adapts our solution for heterogeneity discounts, with a bias towards those data-centers that have served most traffic in the past. The rationale behind this approach is that by concentrating traffic in data-centers that previously sent more traffic, there may be a better chance of incurring bulk discounts at some nodes

in the future.

We propose to add a term to Eq. 1 that represents this fact. This term $g(i)$ is a weight (between 0 and 1) assigned to i based on how much data i has transmitted in the past. In particular, we define $g(i)$ as $\left(1 - \frac{P_i}{\sum_{i \in \mathcal{V}} P_i}\right)$, which ensures nodes that sent more in the past (P_i) are assigned a lower cost. In other words, those data-centers with the lowest previously transmitted volumes receive a penalty inversely proportional to its total contribution to the traffic.

Intuitively, a trade-off exists between choosing an optimal cost tree for the current set of costs, and using nodes that may have a better likelihood of triggering volume discounts in the future. Thus, we add a parameter to control the importance of volume discounts, β ($\beta \geq 0$), resulting in a final term to add to Eq.1 equal to $\beta \cdot g(i)$. An adequate value of β depends directly on the pricing policies from CSPs. For scenarios where volumes discounts are significant, better results would be achieved with higher β values. Section 6 will study β parametrization based on an extensive set of real measurements on both EC2 and Azure.

Thus, the function to minimize (Eq. 1) is replaced by:

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} e_{ij} \cdot \left(\frac{f^c(i, j, B, \mathbf{C}, \mathbf{T}, P_i)}{max(f^c(\cdot))} + \beta \left(1 - \frac{P_i}{\sum_{i \in \mathcal{V}} P_i}\right)\right) \tag{8}$$

where the first term has been also normalized to make it easier comparable to the second term (i.e., it also ranges between 0 and 1), and the function cost has changed as it needs to consider the thresholds and previously transmitted volumes.

The resulting $f^c(\cdot)$ can be easily and analytically calculated by assuming requests fall entirely between two thresholds —i.e., all GBs of a given request cost the same. This is by far the most common case as the volume of individual requests (order of GBs) is typically much smaller than the thresholds at which volume discounts are triggered (order of tens to hundreds of TBs). The cost per GB may then be expressed formally as:

$$f^c(i, j, B, \mathbf{C}, \mathbf{T}, P_i) = C_{ij0} - \left(\sum_{t=1}^{|T_{ij}|} (C_{ij[t-1]} - C_{ijt}) \cdot H(B + P_i - T_{ijt}) \right) \tag{9}$$

where $H(\cdot)$ is the Heaviside step function [25] ($H(x) = 1$ if $x > 0$, $H(x) = 0$ if $x < 0$), which helps to decide if a threshold was exceeded (and a discount applies), or not. If requests are comparable in size to thresholds, the Heaviside step function can be also used to relate the fraction of a request that falls into each interval and its cost.

Consider again the example involving an EC2 data-center located in Virginia (i) and an Azure data-center (j). For the first transmission (P_i is zero), the cost per GB is C_{ij0} ($0.12) because every term inside the summation of Eq. 9 is zero. When $P_i + B$ exceeds the first threshold (10 TB), the summation evaluates to $C_{ij0} - C_{ij1}$, which implies a discount of $0.03 ($0.12 - $0.09) on the initial cost. The same applies when the second threshold is exceeded. Here, the discounts are ($0.12 − $0.09) and ($0.09 − $0.07) which implies a total discount of $0.05 on the initial cost. A similar process is used when other thresholds are exceeded. Finally, $max(f^c(\cdot))$ is equal to the highest rate per transmitted GB as fixed by the CSP's pricing policy, in this example, $0.25.

Note that this heuristic assumes that the relative cost of data-centers stays the same across different volume thresholds. That is, if $C_{ijt} < C_{pqt}$ then $C_{ij(t+1)} < C_{pq(t+1)}$ $\forall$ $i, j, p, q \in \mathcal{V}$ and $0 \leq t < k$. We believe this assumption is reasonable. In fact, all current CSPs that apply volume discounts including EC2 and Azure (see Table 1) meet this premise (to the best of our knowledge), as discounts tend to be a fraction of the original pricing.

4.4 Implementation issues

The implementation of CloudMPCast comprises of two modules: retransmitter and routing planner. The former was implemented using raw sockets for port forwarding. The latter codes the optimization problem in Java (Eqs. 1-8) following a matrix approach and solves the problem using the IBM CPLEX optimizer. An important consideration is

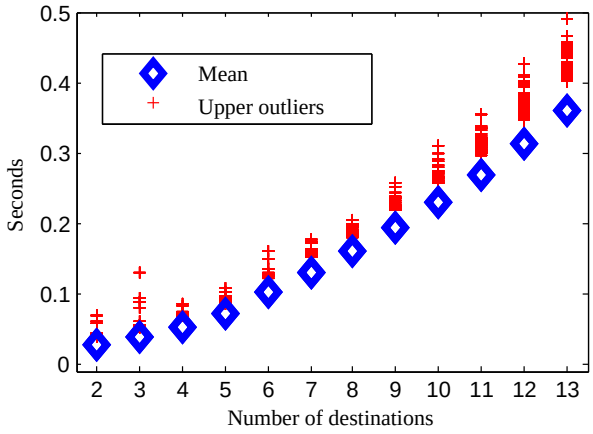

Figure 2: Computation time of routing planner

the execution time associated with the planner, since it must respond to requests for alternative routes in an on-line fashion. Fig. 2 shows the time required to solve a request according to the number of destinations. The measurements were conducted on a host with a 2.0 GHz Xeon processor, similar in capabilities to EC2 and Azure medium instance VMs. The figure shows the mean along with the upper outliers. Upper outliers are defined as samples larger than the third quartile plus 1.5 times the difference between the third and first quartile. The results show that even for largest requests involving 14 nodes, the execution time is below 500 ms, which demonstrates the run time efficiency of CloudMPcast. The computation time with the planner was not sensitive to the choice of α and β parameters.

5 Evaluation Methodology

In this section, the methodology used to evaluate CloudMPcast is described. We begin by describing our data-center and traffic models, and next describe how we collected real inter data-center throughput and latencies.

5.1 Data-center and traffic models

Modeling cloud data-center locations: We modeled our data-center locations by considering all available data-center locations with EC2 and Azure. As of 2014, EC2 and Azure offer eight and six locations respectively spread over the world. Fig. 3 shows the location of these data-centers. We consider a variety of deployments:

- *EC2-only and Azure-only:* These respectively consider data-centers corresponding only to EC2, or only to Azure.
- *Azure-subset:* This deployment includes the data-centers of Azure located in Europe and North America, a common scenario in practice. Note that in this deployment, pricing is homogeneous across all data-centers —hence, this scenario helps us to evaluate the potential opportunities in exploiting volume discounts even in the absence of pricing heterogeneity.
- *Multi-cloud:* This is a deployment that combines data-centers corresponding to both providers together. Multi-cloud deployments could be of interest for customers who seek to increase availability through use of multiple providers, and have been suggested as a useful mechanism to enhance quality of service in some areas of the South America and Asia which are not well covered by a given CSP in isolation [3].

Modeling source/destination demands: A key factor that impacts our evaluations is the probability that different nodes serve as sources, and the destinations to which data are sent, for any individual bulk transfer request. We use a random number of destination data-centers between 2 and $D-1$, where D is the total number of available data-centers of the deployment, and pick the specific set of source and destinations following these models:

- *Homog:* Both source data-center and data-centers that serve as destinations are picked at random.
- *Bias-Src:* The probability that a data-center is chosen as the source node for a given request is proportional to the popularity of the data-center region. We base our popularity model on a report [26] that reviewed several cloud traffic trends by geographical region. In 2012, North America led the generation of cloud traffic with 469 EB, Asia Pacific held the second place with 319 EB, Western Europe represented 225 EB and Latin America about 77 EB. For the *Bias-Src* model, we pick sources with probability proportional to the measurements above —e.g., nodes in North America are more likely to be picked as sources.
- *Spread-Rcv:* In many disaster recovery deploy-

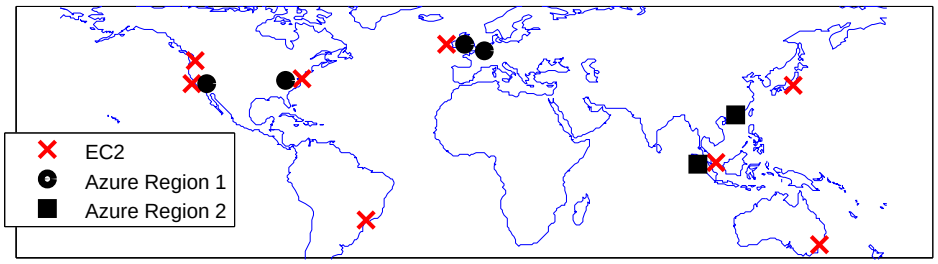

Figure 3: Multi-cloud testbed deployment

ments, it is desirable to replicate data to data-centers that are as geographically spread out as possible to minimize the likelihood of correlated failures. Consequently, in the *Spread-Rcv* model, once the source data-center is fixed, the set of destinations is chosen proportional to the latency between the source and the possible destination data-centers —higher the latency, higher the likelihood of being a destination.

• *Hetero:* This deployment simply combines the *Bias-Src* and *Spread-Rcv* models.

5.2 Measurements of inter-data-center bandwidth and latency

To drive the evaluation of CloudMPcast, realistic data that captures typical inter-data-center bandwidth and latency is needed. Such data is not readily available today. To remedy this, we have conducted extensive measurements on EC2 and Azure (Az).

To measure TCP bandwidth between data-centers, we started a medium instance VM in each of eight available locations that EC2 offered. We chose medium instances as smaller ones showed erratic behavior in terms of inter-network measurements [18]. These instances have compute capacity equivalent to a 2.0 GHz Intel Xeon processor. Similarly, we launched six medium VMs in each of the locations offered by Azure —each having 2 virtual cores with a capacity of 1.6 GHz. We actively measured the TCP throughput between each pair of data-centers among this total set of 14 data-centers during one day using Iperf [27]. Measurements were conducted every hour with each measurement lasting two minutes. Fig. 4 depicts the mean throughput in Mb/s for each data-center pair during the measurement campaign. We also characterized the variability in TCP throughput over time. Overall, low variations on the measured throughputs were found —the coefficient of variation was less than 0.2 for 80% of the paths, and less than 0.43 for all the paths over the period of measurement. Generally, those links with high mean bandwidth exhibited the lowest variability.

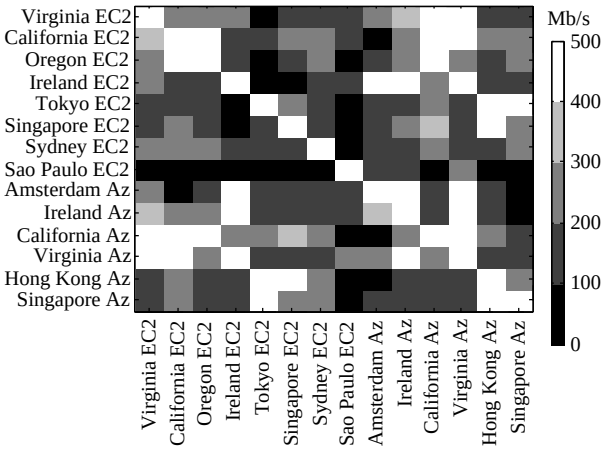

Figure 4: Measured TCP throughput for each pair of data-centers (in Mb/s)

Alternating with the bandwidth measurements, we also measured RTTs by conducting pings between all pairs of data-centers —needed for the *Spread-Rcv* model. Given that Azure blocks ICMP traffic, we circumvented this by developing a simple UDP ping server. Fig. 5 shows these measurements for completeness and comparison purposes.

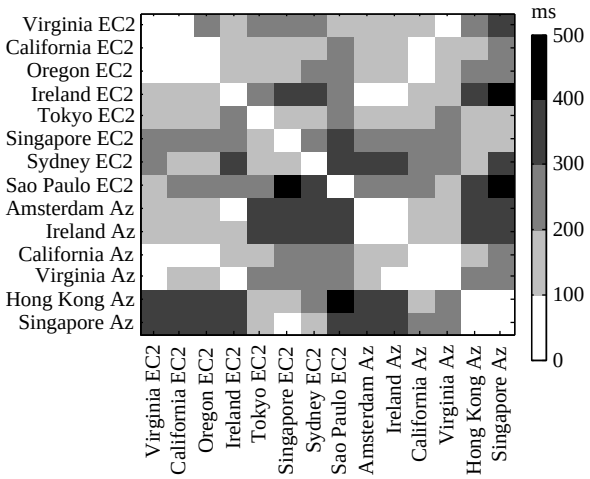

Figure 5: Measured RTTs for each pair of data-centers (in milliseconds)

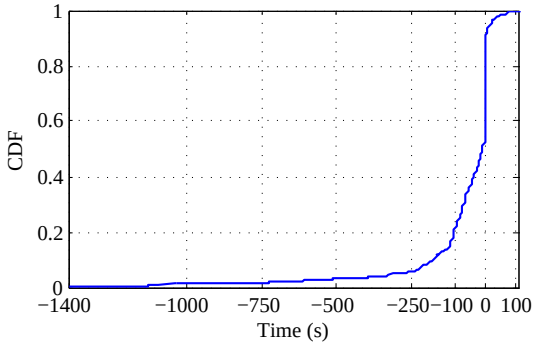

Figure 6: Impact of CloudMPcast on transfer times

6 Results

This section evaluates the impact of CloudMPcast on both transfer times and cost savings. First, results are presented using the *Multi-cloud* deployment that involves 14 data-centers in all EC2 and Azure locations, and assuming the *Homog* traffic demand model. Then, sensitivity results to both data-center location and traffic demand models are presented. The bandwidth pricing summarized in Table 1 is used, and the inter-data-center bandwidth and latency are as described previously.

6.1 Impact on transfer times

The use of CloudMPcast alters the dissemination paths but thanks to its design, it only chooses paths that can provide equivalent transfer times. To validate this claim, we conducted experiments using a real *Multi-cloud* deployment. Specifically, we started VMs in each of the data-centers under evaluation and measure the bandwidth between them periodically as explained in Section 5. 1000 requests were generated following the *Homog* model. Then, for each request, the times required to replicate a 5 GB file using the trivial (direct transmission from the source to each destination) and CloudMPcast solutions (generated assuming an α value of 1, which implied transfer times must match the trivial solution) are measured for each destination iteratively.

Fig. 6 shows the increase in transfer time with CloudMPcast. Each point corresponds to the increase in transfer time with CloudMPcast compared to the trivial solution for each destination in every request. A negative value indicates that CloudMPcast reduced transfer times compared to the trivial solution. As shown, the time required for CloudMPcast is lower in more than 50% of the transfers. This is because bandwidth using an overlay approach can sometimes outperform bandwidth of direct transfers [28]. While CloudMPcast proposes alternative paths which are at least as fast as the trivial solution, such alternative paths are often not only fairly equivalent but also much faster. In about 40% of the cases, CloudMPcast has similar performance to the trivial solution – these are cases in which it is not possible to reduce cost while maintaining transfer times. Finally, in a small fraction of cases (roughly 5-10%) transfer time increased — these cases corresponded to ongoing bulk transfers, where throughput on some of the links changed unpredictably during the transfer. However, note that this latter group not only represents a small fraction of the total, but also their impact in absolute terms is modest compared to the gain that more than half of the transfers obtained. Overall, these results show the effectiveness of CloudMPcast in ensuring comparable and often better transfers than the trivial solution.

6.2 Cost savings with CloudMPcast

The cost savings with CloudMPcast is sensitive to the amount of data exchanged between each pair of data-centers, given that volume discounts are triggered at different thresholds. We study the cost savings in three scenarios: (i) *low-volume* scenarios where no volume discounts are involved since data-centers generate less traffic than the first volume discount threshold; (ii) *middle-volume* scenarios where each data-center pair exchanges up to 10 TB worth of data. Note that volume discounts start kicking in these regimes since the total traffic out-bound from a source data-center (cumulative over all destination data-centers) may exceed volume thresholds, especially with CloudMPcast; and (iii) *large-volume* scenarios, which extend up to a regime where all volume discounts have been exceeded. For each scenario, we generated 100,000 bulk-transfer requests to evaluate savings per scenario. We begin by presenting results with the *Homog* and *Multi-cloud* models. The next section will extend the study to other models.

Low-volume scenario: Fig. 7 shows the distribution of the savings ratio for low-volume scenarios. Specifically, if TS_r^c and $CloudMPcast_r^c$ are the costs of the trivial solution and solution with CloudMPcast for a given request r, we measure the savings ratio for each request as follows:

$$\frac{TS_r^c - CloudMPcast_r^c}{TS_r^c}$$

The figure is a box-plot with boxes representing 25th and 75th percentile of cost savings over different runs. The line in the center is the median. Additionally, the mean is also included. Given that volume discounts do not apply, note that the amount of data in absolute terms per request is irrelevant. We consider different values for α (constraint 6), the performance tolerance. With $\alpha = 1$, transfer times are required to be the same as (or better than) the trivial solution. Lower α values indicate that lower transfer times are acceptable. The results show that the savings are over 20% even with $\alpha = 1$, and grow even higher with smaller α values. In the rest of the simulations, we use $\alpha = 1$ unless otherwise mentioned.

Middle-volume/Large-volume scenarios: We turn our attention to the performance of CloudMPcast as volumes increase and eventually exceed volume discount thresholds. To evaluate these scenarios, we use bulk-transfer requests involving 5 GB of data each, which is representative of scenarios that include the distribution of MPEG-2 movies or the cloning of small VMs [29]. We stopped the simulation when the average data exchanged between each pair of data-centers reached the desired target for each scenario.

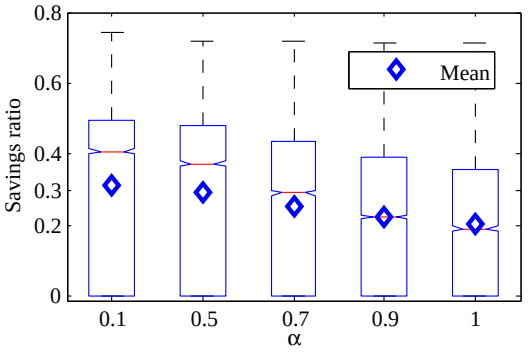

Figure 7: Impact of α on savings ratios in low-volume scenarios

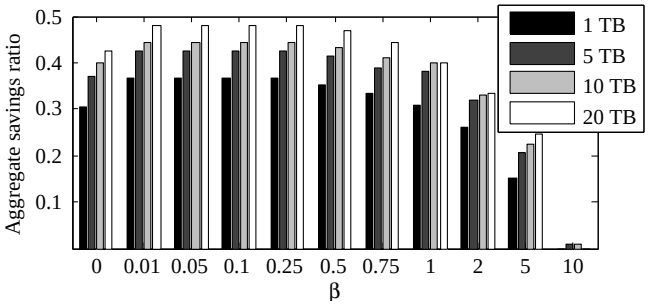

Figure 8: Impact of β for different values of average traffic exchanged between pairs of data-centers

Impact of β: We focus on the impact of the parameter β on the performance of CloudMPcast. Larger β values tend to bias delivery trees towards using data-centers that have transmitted more data in the past, so volume discounts could be triggered sooner. Fig. 8 shows the aggregate savings ratio for different values of mean data exchanged between data-center pairs (1, 5, 10 and 20 TB), for α fixed to 1. It is worth remarking that 20 TB a month is equivalent to a constant throughput of about 60 Mb/s between data-centers, not an unreasonable

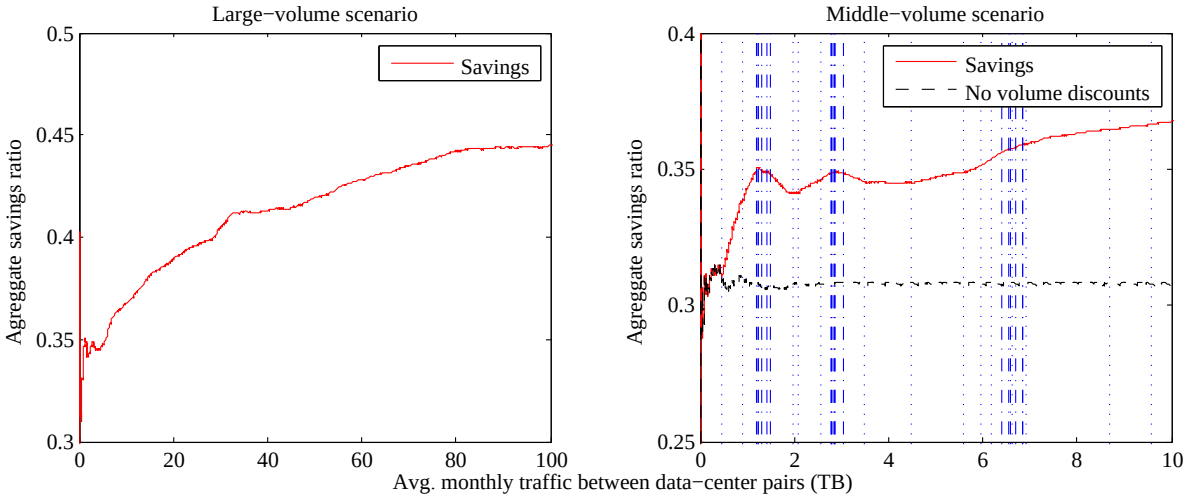

Figure 9: Aggregate savings ratio by averaged traffic volume per source and destination, for large (left) and middle (right) volume scenarios (with $\alpha = 1$ and $\beta = 0.01$)

figure —possibly even a conservative estimate of real traffic volumes [6]. The aggregate savings ratio for each aggregate is calculated as:

$$\frac{\sum_r TS_r^c - \sum_r CloudMPcast_r^c}{\sum_r TS_r^c}$$

Again, TS_r^c and $CloudMPcast_r^c$ are the cost of individual requests with the trivial solution and with CloudMPcast respectively.

Choosing small β values shows better performance than when β is zero. This is because when β is zero, CloudMPcast chooses retransmitter nodes randomly when their costs are the same (e.g., four EC2 data-centers tie for the cheapest cost, and any of them may be chosen with equal likelihood). This tends to distribute traffic uniformly across the retransmitter nodes with equivalent costs (subject to transfer time considerations). Small β values serve to break the tie and make CloudMPcast aggregate data towards only one data-center, resulting in volume discounts being triggered sooner.

For intermediate beta values ($0 < \beta < 0.5$) the savings ratio remains stable, whereas for larger β values, the savings ratio is small. This is because the bias to data-centers that have previously transmitted overwhelms the current transmission costs. A more expensive data-center that sent more of the initial requests at the start of the simulation would result in a bias towards it. In the future, our heuristic could be refined to ensure a minimum volume of traffic has been sent by a given data-center before biasing traffic towards it. Finally, in deployments where transmission costs are homogeneous, larger β values have more benefit (Sec 6.3). In the rest of the simulations, we have used $\beta = 0.01$ unless otherwise mentioned.

Savings Ratio: Fig. 9 (left) shows the aggregate savings ratio as traffic volume increases. Interestingly, the savings grows from 35% to about 45% as total data volume increases. At this latter point, all the volume thresholds have been exceeded, and savings are entirely attributable to pricing heterogeneity. Fig. 9 (right) shows the savings when the volume discounts begin to get triggered. The top curve shows the savings ratio assuming volume discounts. The bottom curve shows the savings ratio assuming no volume discounts. In this case, the mean traffic exchanged between data-center pairs is limited to 10 TB, which translates to an equivalent constant throughput between data-centers of 30 Mb/s.

Several observations can be made. First, while the savings ratio grows overall with aggregate data, the trend is neither linear nor monotonic. The reason is that the trivial solution and CloudMPcast incur volume discounts at different times. In this figure, the vertical dotted lines represent pric-

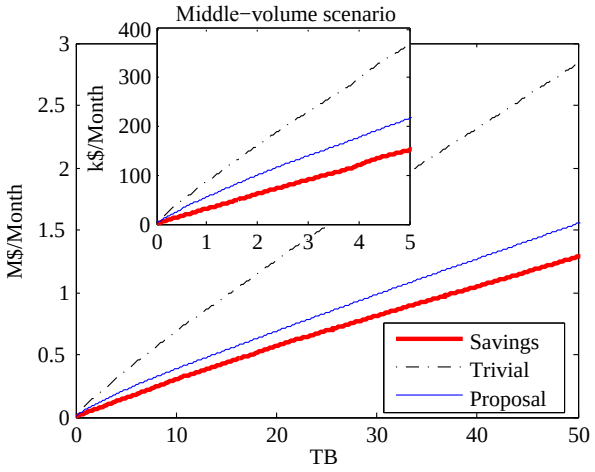

Figure 10: CloudMPcast savings in dollar terms as a function of average monthly traffic between pairs of data-centers (with $\alpha = 1$ and $\beta = 0.01$)

ing reductions for a data-center because of traffic volumes exceeding discount thresholds for the CloudMPcast approach. On the other hand, semi-continuous lines indicate similar pricing reductions for the trivial solution. Observe that when volumes transmitted are low enough, the savings ratio with and without volume discounts are the same. Then, CloudMPcast exceeds the first threshold and incurs discounts, causing the savings ratio to peak when the aggregate traffic is about 1.5 TB. Around that point, the volumes with even the trivial solution are also sufficient for discounts to apply, and hence the savings ratio dips slightly. It increases again when the next volume threshold is reached with CloudMPCast, and so on. Overall, the results show that CloudMPCast is very effective in exploiting discounts.

Savings in dollar terms: Fig. 10 translates the previous savings ratio into dollars in absolute terms. It reflects the savings for middle and large volume scenarios in terms of thousands of dollars (k$) and millions of dollars (M$) respectively. It is worth remarking that none of the curves are straight lines but they have different positive slopes according to the volume discounts that apply at a particular moment, although it is not easily appreciated visually. For a constant rate between data-centers of 10 Mb/s, which roughly equals 3 TB a month, CloudMPcast can achieve a monthly saving of about one hundred thousand dollars, driving up savings to over a million dollars a year. Note that at transmission rates of 100 Mb/s or 1 Gb/s, not unrealistic figures, these savings increase by one to two orders of magnitude which translates into millions of dollars.

Characterizing data dissemination trees: Let us drill into the features of the trees and paths CloudMPcast is proposing. Specifically, we study how CloudMPcast changes the routing matrices (utilization of each direct connection) and the distribution of tree depth (maximum number of hops from source to any of the destinations for a given distribution tree) with respect to the trivial solution. The depth of the distribution trees for the trivial solution is basically one, as only one hop is utilized for transmitting directly from the source to each of the destinations. We illustrate the routing matrices in terms of variation with respect to the trivial solution. Specifically, we use a grey scale such that if a given direct connection has incremented the volume traversing it, the square in the routing matrix that represents such connection (source-destination pair) will turn a lighter grey. Conversely, if the connection has lost importance the color will be darker. The trivial solution uses all the connections homogeneously. Hence the initial color is the darkest grey equivalent to ratio 1 in all the cases. This allows us to visually assess changes in the importance of a data-center.

Fig. 11 (left) shows the routing matrix assuming no volume discounts. The clearest change from the trivial solution is that EC2's data-centers in Virgina, California, Oregon and Ireland as well as California in Azure have gained relevance. The volume traversing the direct connections between these sources and the total set of destinations has increased by a factor between 2 and 5. CloudMPcast produces longer trees, and concentrates traffic through a few data-centers compared to the trivial solution because it is desirable to exploit pricing heterogeneity by routing data through lower cost data-centers

When we consider requests over time, Fig. 11 (right), volume discounts apply, and the concentration of traffic becomes even more noticeable. The California EC2 data-center increases all its direct connections by a factor of 5 or larger, while other previously popular data-centers in EC2 in-

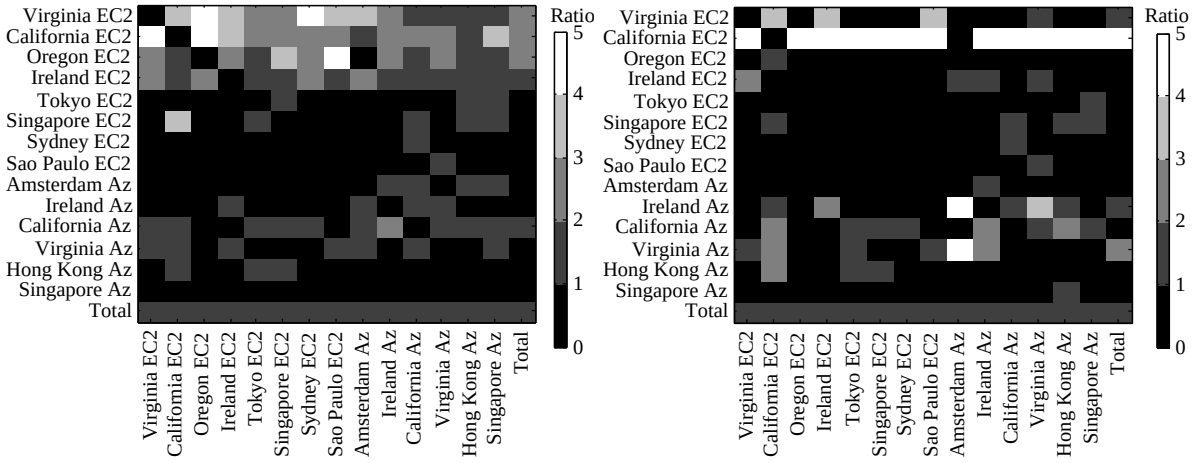

Figure 11: Routing matrices for the low-volume scenario (left), and the large-volume scenario (right), in terms of variation with respect to the trivial solution (with $\alpha = 1$ and $\beta = 0.01$). X axis represents the source data-center, while Y axis is the destination data-center for all possible pairs of direct connections

frastructure decrease their number of connections. Similarly, Virginia in Azure has its relevance increased by a factor of 4. The intuition behind this is that when volume discounts apply, it is cheaper to choose one data-center as hub to make the most of the volume discounts.

Fig. 12 shows the distribution trees CloudMPcast proposes without (top) and with (bottom) volume discounts after each data-center has received 50 TB worth of data. In both cases, trees have higher depth than the trivial solution (depth equal to one), with the depth being larger than 5 in some cases. Some of this may be attributed to the ability of CloudMPcast to lower costs through deeper distribution trees. Specifically, we observed that many cases of higher depth trees corresponded to cases where Azure's Singapore and Hong Kong data-centers were the sources. In particular, (i) transmission from these data-centers to other Azure data-centers often involved traversing other EC2 data-centers, given the high transmission costs from these Azure data-centers, and given that internal data transfers within Azure are not charged at a lower rate; and (ii) transmission from these data-centers to other EC2 data-centers typically involved traversing the Singapore EC2 data-center for performance reasons —these Azure data-centers have good bandwidth to the Singapore EC2 data-center which in turn is well connected to other EC2 data-centers as shown in Fig. 4.

When volume discounts do not apply, the higher depth in some cases could be attributed to the optimization framework not explicitly favoring lower depth trees in the case where there are several feasible and equal cost solutions. For instance, in some cases trees had a chain of US EC2 data-centers when an alternative shorter tree may have been feasible with the same cost. This could potentially be addressed in the future by adding a small bias in our cost function to favor lower depth trees when there are multiple equal-cost feasible solutions. Interestingly, note that when CloudMPcast considers volume discounts (even with low values for β), there is a natural preference for shorter trees since there is incentive to concentrate traffic through a small number of hub nodes. That said, paths may also sometimes increase in length since it may be desirable to go through a hub node to reach a receiver data-center which did not previously require intermediate hops.

6.3 Sensitivity to data-center and traffic demand models

While previous results have shown the benefits of CloudMPcast, the evaluation focused only on the *Multi-cloud* data-center location and *Homog* traffic demand models. In this section we conduct sensi-

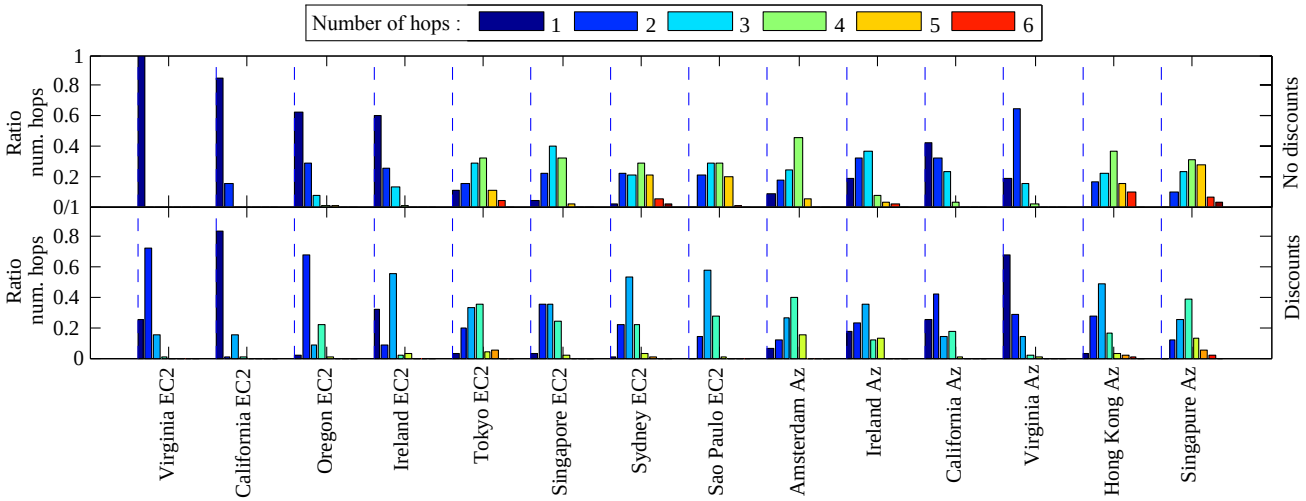

Figure 12: Tree depth distribution in number of hops (identified by colors) per source data-center (assuming $\alpha = 1$ and $\beta = 0.01$), first for the low-volume scenario (no volume discounts) and then for the large-volume one (discounts)

tivity analysis to other models. Similar to the experimental design shown previously, we generated 100,000 new requests per scenario assuming that each request has a size of 5 GB.

Sensitivity to data-center model: Fig. 13 shows the aggregate savings ratio, as well as savings in dollar terms, for *EC2-only*, *Azure-only*, and *Azure-subset*, for both large-volume and middle-volume scenarios. The X-Axis extends to a larger value for *Azure-only* and *Azure-subset* because there are fewer data-centers and it takes a higher traffic volume exchanged between data-center pairs to exceed the largest threshold for volume discounts.

Several observations may be made. First, 10-15% cost savings are achieved in the *Azure-subset* model. Note that this cost savings arises entirely from the ability of CloudMPcast to exploit volume discounts, as in this setting transmission costs are homogeneous across all data-centers.

Second, the savings ratio of *EC2-only* (60%) is higher than with *Multi-cloud*. This is because the heterogeneity in costs between EC2 data-centers is larger than when all data-centers in *Multi-cloud* are considered. In addition, Fig. 4 shows that the cheapest EC2 data-centers are also the most well-connected, which provides greater optimization potential for CloudMPcast. The absolute dollar savings is smaller with *EC2-only* because it has fewer data-centers. However, the dollar savings is still significant.

Finally, the savings ratio with the *Azure-only* model is lower than with the *Multi-cloud* model. This is because pricing in Azure shows less heterogeneity across data-centers. Azure divides its data-centers into two regions with homogeneous pricing within each region. Further, Fig. 4 shows that communication between the two regions is limited in terms of throughput. This makes it even more difficult to find a path that extends across these regions that fulfills the bandwidth requirements. Nevertheless, significant cost savings is still achieved because CloudMPcast is able to leverage volume discounts.

Sensitivity to traffic demand model: Fig. 14 shows the aggregate savings ratio and savings in dollar terms for different traffic demand models. We make several observations. First, the savings ratio with *Bias-Src* decreases by about 5% compared to *Homog*. This is because with *Bias-Src*, a greater fraction of requests originate in US data-centers. These requests do not benefit from pricing heterogeneity discounts since the transmission costs of US data-centers are the lowest. However, savings is still achieved because of volume discounts. Sec-

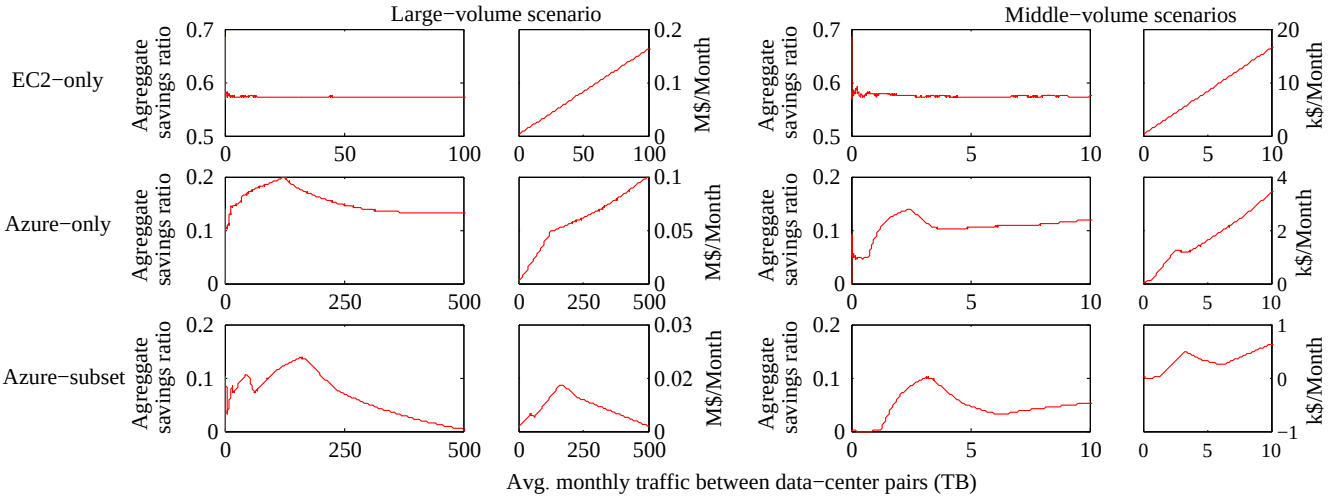

Figure 13: Results for data-center location models: *EC2-only*, *Azure-only* and *Azure-subset* (with $\alpha = 1$ and $\beta = 0.01$)

ond, the savings ratio with *Spread-Rcv* is about 5% higher compared to *Homog*. This is because data-centers which are not as well connected are also the most expensive opening further opportunities for discounts. Finally, the performance with *Hetero* which is a combination of *Bias-Src* and *Spread-Rcv* is similar to *Homog*.

Summary: Overall, our results show that CloudMPcast can achieve significant discounts in a rich and varied set of scenarios. A key reason is because CloudMPcast leverages both heterogeneity and volume discounts.

7 Conclusion and future work

We have presented CloudMPcast, a systematic approach to constructing multi data-center distribution trees for bulk transfers. CloudMPcast optimizes dollar costs of distribution taking public cloud charging models into account, while ensuring end-to-end data transfer times are not affected.

Extensive evaluations of CloudMPcast leveraging an extensive set of inter-data-center bandwidth and latency measurements from both Azure and EC2 have shown significant benefits. Cost savings range from 10% to 60% across a wide variety of scenarios, which translates to millions of dollars a year. Further, the results also point to the critical importance of exploiting both volume discounts and pricing heterogeneity across a variety of settings. This ensures savings can be achieved even when only one class of discount is applicable — e.g., Rackspace only offers volume discounts. When both discounts are applicable, considering them together provides even better results.

We have primarily designed CloudMPcast from a cloud customer perspective. However, a CSP may also benefit from its operation. CloudMPcast forwards traffic to lower cost data-centers which may translate to lower revenue for the CSP. However, the bandwidth costs incurred by the CSP in these locations is usually substantially lower as well — as a result, the profit margins for CSPs tend to be higher at US and European data-centers [30, 31]. Exploring this issue is an interesting direction for future work.

While our results are promising, some CloudMPcast's functionalities can be improved. One potential future opportunity for further cost savings is to consider splitting data along multiple paths. Each path could potentially have less bandwidth than the trivial solution, but considering multiple paths could ensure the overall transmission time does not suffer. Another opportunity we are exploring is how CloudMPcast can exploit some partial

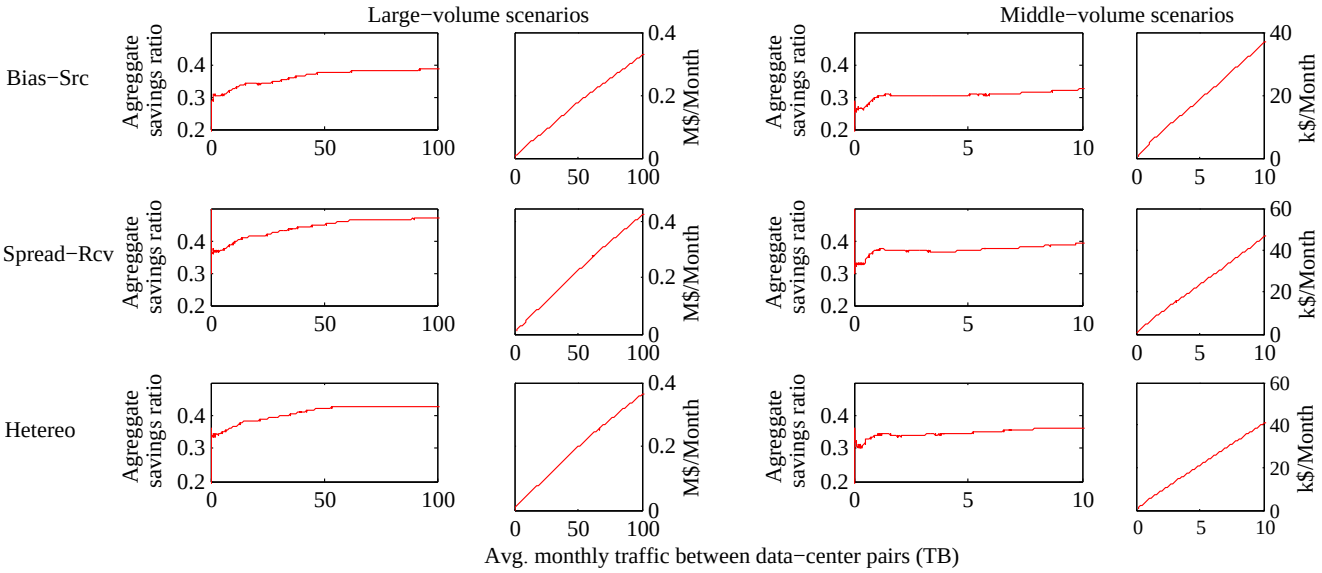

Figure 14: Results for traffic demand models: *Bias-Src*, *Spread-Rcv* and *Hetero demands* (with $\alpha = 1$ and $\beta = 0.01$)

knowledge of future requests in a computationally effective manner, and the impact of this on further savings. As other future extensions, we also plan to consider a wider range of applications beyond bulk transfer, consider pricing schemes of more CSPs, and consider the impact of changes in current cloud pricing schemes.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation (NSF) under Award No.1162333. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

J. L. García-Dorado is thankful for the financial support of the José Castillejo Program (CAS12/00057).

Finally, the authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] High Scalability, "Latency is everywhere and it costs you sales - how to crush it," http://highscalability.com/latency-everywhere-and-it-costs-you-sales-how-crush-it/.

[2] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier Internet services and its applications," in *ACM SIGMETRICS*, 2005.

[3] Z. Wu and H. V. Madhyastha, "Understanding the latency benefits of multi-cloud webservice deployments," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 2, pp. 13–20, 2013.

[4] P. Shankaranarayanan, A. Sivakumar, S. Rao, and M. Tawarmalani, "Performance sensitive replication in geo-distributed cloud datastores," in *IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014.

[5] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha, "SPANStore: Cost-effective geo-replicated storage spanning multiple cloud services," in *ACM SOSP*, 2013.

[6] Forrester Research, "The future of data center wide-area networking," http://www.forrester.com.

[7] J. Hamilton's blog, "Inter-datacenter replication & geo-redundancy," http://perspectives.mvdirona.com.

[8] Y. Chen, S. Jain, V. K. Adhikari, Z. li Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via Yahoo! datasets," in *IEEE INFOCOM*, 2011.

[9] E. Zohar, I. Cidon, and O. Mokryn, "The power of prediction: cloud bandwidth and cost reduction," in *ACM SIGCOMM*, 2011.

[10] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with Netstitcher," in *ACM SIGCOMM*, 2011.

[11] T. Nandagopal and K. P. N. Puttaswamy, "Lowering inter-datacenter bandwidth costs via bulk data scheduling," in *Symposium on Cluster, Cloud and Grid Computing*, 2012.

[12] Y. Feng, B. Li, and B. Li, "Jetway: minimizing costs on inter-datacenter video traffic," in *ACM Multimedia Conference*, 2012.

[13] K. He, A. Fisher, L. Wang, A. Gember, A. Akella, and T. Ristenpart, "Next stop, the cloud: Understanding modern web service deployment in EC2 and Azure," in *ACM Internet measurement conference*, 2013.

[14] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," in *ACM SIGCOMM*, 2009.

[15] L. Gyarmati, R. Stanojevic, M. Sirivianos, and N. Laoutaris, "Sharing the cost of backbone networks: cui bono?" in *ACM Internet measurement conference*, 2012.

[16] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, and F. Lau, "Orchestrating bulk data transfers across geo-distributed datacenters," *IEEE Trans. Cloud Comput, doi:10.1109/TCC.2015.2389842*, 2015.

[17] Z. Ou, H. Zhuang, A. Lukyanenko, J. Nurminen, P. Hui, V. Mazalov, and A. Yla-Jaaski, "Is the same instance type created equal? exploiting heterogeneity of public clouds," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 201–214, 2013.

[18] G. Wang and T. S. Eugene Ng, "The impact of virtualization on network performance of Amazon EC2 data center," in *IEEE INFOCOM*, 2010.

[19] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: a data center network virtualization architecture with bandwidth guarantees," in *ACM CoNEXT*, 2010.

[20] Y. Xu, Z. Musgrave, B. Noble, and M. Bailey, "Bobtail: avoiding long tails in the cloud," in *USENIX NSDI*, 2013.

[21] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: Sharing the network in cloud computing," in *ACM SIGCOMM*, 2012.

[22] L. Sahasrabuddhe and B. Mukherjee, "Multicast routing algorithms and protocols: a tutorial," *IEEE Netw.*, vol. 14, no. 1, pp. 90–102, 2000.

[23] C. Noronha and F. Tobagi, "Optimum routing of multicast streams," in *IEEE INFOCOM*, 1994.

[24] M. Kodialam, T. Lakshman, and S. Sengupta, "Online multicast routing with bandwidth guarantees: a new approach using multicast network flow," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 676–686, 2003.

[25] R. Bracewell, *The Fourier transform and its applications*, 2000.

[26] Cisco, "Global cloud index: Forecast and methodology, 2012–2017," http://www.cisco.com.

[27] A. Tirumala, M. Gates, F. Qin, J. Dugan, and J. Ferguson, "Iperf - the TCP/UDP bandwidth measurement tool," http://sourceforge.net/projects/iperf/.

[28] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *ACM SOSP*, 2001.

[29] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. van der Merwe, "Cloudnet: dynamic pooling of cloud resources by live WAN migration of virtual machines," *SIGPLAN Not.*, vol. 46, no. 7, pp. 121–132, 2011.

[30] Telegeography, "International market trends," in *Pacific Telecommunications Council*, 2013.

[31] ——, "Local access prices vary widely —even among major business centres," http://www.telegeography.com/.