Exploring the cloud from passive measurements: The Amazon AWS case

Exploring the Cloud from Passive Measurements: the Amazon AWS Case

Ignacio Bermudez, Stefano Traverso, Marco Mellia, Maurizio Munafò
DET, Politecnico di Torino, Italy – {lastname}@tlc.polito.it

***Abstract*—This paper presents a characterization of Amazon's Web Services (AWS), the most prominent cloud provider that offers computing, storage, and content delivery platforms. Leveraging passive measurements, we explore the EC2, S3 and CloudFront AWS services to unveil their infrastructure, the pervasiveness of content they host, and their traffic allocation policies.**

Measurements reveal that most of the content residing on EC2 and S3 is served by one Amazon datacenter, located in Virginia, which appears to be the worst performing one for Italian users. This causes traffic to take long and expensive paths in the network. Since no automatic migration and load-balancing policies are offered by AWS among different locations, content is exposed to the risks of outages.

The CloudFront CDN, on the contrary, shows much better performance thanks to the effective cache selection policy that serves 98% of the traffic from the nearest available cache. CloudFront exhibits also dynamic load-balancing policies, in contrast to the static allocation of instances on EC2 and S3.

Information presented in this paper will be useful for developers aiming at entrusting AWS to deploy their contents, and for researchers willing to improve cloud design.

I. Introduction

Last years witnessed the growth of cloud-based services that provide computing, storage and offloading capabilities on remote datacenters, offering the opportunity to customers to reduce costs by virtualizing hardware management. The leading position in this panorama is taken by Amazon, which offers a large gamma of cloud-based services, named Amazon Web Services (AWS). The most well-know Amazon cloud services are "Elastic Compute Cloud" (EC2), and "Simple Storage Service" (S3), with "CloudFront", the Content Delivery Network (CDN).

Following the definitions provided in [1], AWS represents an *Infrastructure Provider*, and EC2 and S3 correspond to *Infrastructure as Service* products. In other words, through virtualization, a large set of computing resources, such as storing and processing capacities can be split, assigned, and dynamically sized to satisfy customers' demand. *Customers* are represented by companies aiming at offering their *services* without carrying on costs and risks of building and managing their own hardware and infrastructure. Many successful companies like Dropbox, Zynga and Netflix to name a few, successfully rely on AWS.

AWS has gained a large interest within the research community too. In particular, many works investigate the possibility of exploiting AWS EC2 for research purposes [2], [3]. Others instead focus on evaluating the performance of AWS computing and networking virtual resources [4], [5]. However, to the best of our knowledge, all the previous works focus on the benchmarking of AWS services and infrastructure, and they all rely on "active" probing. What is missing is the characterization of Amazon Web Services as perceived by the end-users, i.e. an evaluation of actual AWS workload and performance by means of "passive" observation of traffic.

This work was supported by the European Commission under the FP7 IP Project "An Intelligent Measurement Plane for Future Network and Application Management" (mPlane).

The goal of this paper is to provide an extensive study of AWS through passive network analysis of traffic collected from our University campus and from three large Points of Presence (PoP) of an Italian national-wide Internet Service Provider (ISP). Our datasets span more than 60 days, and collect the traffic generated by more than 50,000 end-users.

In this work, we dig into a one week long portion of our dataset with a twofold goal: first, we shed light on the AWS infrastructure itself, using a simple yet accurate methodology to reveal the number of datacenters, their locations, and resulting traffic allocation policies. Second, we evaluate which are the services that run on AWS, and how they are accessed by end-users. Notice that providing such characterization is challenging due to the nature of cloud services, where encryption schemes and proprietary solutions are very common.

Our main findings are:

- Among the seven EC2 and S3 datacenters, the one placed in Virginia is the most used, with more than 6,000 EC2 IP addresses and 120 S3 nodes regularly accessed by end-users. It handles alone 85% of total traffic generated by EC2 and more than 64% for S3 – serving daily more than 15TB of data to the ISP end-users in Italy. Surprisingly, the datacenter in Ireland is not the preferred one, and it serves only about 20% of AWS traffic to Italian end-users.
- Web companies that offer their services from AWS systems tend to rely upon one datacenter only. This makes the network pay for large cost of carrying data to far away end-users. Moreover, it represents a large risk in case of failures, since no automatic load-balancing and migration are offered by AWS.
- Performance of datacenters in terms of response time (for EC2) and goodput (for S3) shows that the most popular datacenter is also the worst performing one. Evidence shows that some services suffer because of under-provisioned instances or poor design, but we cannot exclude that the whole infrastructure may be overloaded.
- Considering CloudFront, 24 out of 33 different world-wide caches that build the CDN infrastructure have been spotted in

our traces. However, the cache selection policies adopted by CloudFront wisely serve 98% of traffic from the cache placed in Milan, the closest to Italian end-users. The remaining 2% of traffic comes from worldwide caches, possibly because of load-balancing policies, or by incorrect DNS configuration of end-user clients [6].

We believe this paper provides useful insights about AWS infrastructure, helping in understanding the properties of services relying on cloud-based platforms EC2, S3 and CloudFront. Provided information may result worthwhile for developers aiming at entrust AWS to deploy their contents.

II. Dataset

We rely on passive measurements to characterize AWS services in operational networks. We employ Tstat[1], the open-source traffic monitoring tool developed at Politecnico di Torino, to analyse packets exchanged by actual end-users inside monitored vantage points [7]. Tstat was installed in four different ISP vantage points where it has been collecting traffic from April to June 2012, observing more than 50,000 end-users normally accessing the Internet.

We restrict our analysis on traffic collected during an entire week (starting from April 1st, 2012) from an ISP PoP which aggregates 15,000 ADSL lines. During this week 6M TCP connections were established with AWS servers, exchanging about 340GB of data. We considered traffic monitored on two ISP PoPs and on our Campus network, showing same characteristics. Hence, findings for considered PoP are general and not biased. However, we acknowledge that some of the results in this paper are biased by observing AWS traffic from a single country. Naturally, we expect that some of these results may change if we analyse ISP traffic monitored in another geographical region.

III. Analysis Methodology

We start our analysis isolating all known Amazons IP addresses as listed by the MaxMind[2] organization database, or equivalently returned by the *whois* database. Then, relying on the information provided by the DNS, we identify flows addressed to the cloud computing (EC2) AWS services. AWS indeed follows a strict naming rule for EC2: an instance IP address *a.b.c.d* is registered with a Type-A DNS record as **ec2**-*a-b-c-d.XXXXX.amazonaws.com*, where XXXXX is a variable string. A simple DNS reverse lookup from the IP address allows to discover that *a.b.c.d* corresponds to an EC2 instance.

Unfortunately, same procedure cannot be employed to isolate S3 and CloudFront severs, since the Type-A records obtained from their IP address reverse lookup do not always reveal which AWS service it is. To overcome this, we adopt a technique named *HTTP-knocking*, whose detailed description can be found in [8].

To unveil contents or services delivered by each connection, we relied on DN-Hunter [9], which let us recover the original server hostname requested by the clients and being served by an AWS server. Notice that content is unveiled even for encrypted traffic. Geographical locations of datacenters (*Availability Zones*[3] in AWS terminology) are inferred using traceroute and latency together (details available in [8]). In the rest of this paper, we use IATA codes to identify datacenters instead of conventional names of AWS Availability Zones.

IV. Measurement Definitions

A. Per-flow Metrics

Among the different measurements provided by Tstat for each flow, we consider the server IP address, its original hostname as retrieved by DN-Hunter, the flow RTT, the amount of bytes exchanged at the Application Layer, and the presence of TLS/SSL at the Presentation Layer. These metrics are straightforwardly monitored. More details can be found on [7]. We then consider also the following additional metrics. In particular, we define:

1) Response Time: it is the time the server employs to send the reply after receiving the first request from a client. Let T_{Ack} be the timestamp of the first TCP ACK message sent by server with relative ACK number greater than 1, i.e., acknowledging the reception of some data sent by the client. Let T_{Reply} be the timestamp of the first TCP segment sent by the server carrying application data. The response time is defined as

$$\Delta R = T_{Reply} - T_{Ack}. \quad (1)$$

For HTTP flows, it represents an estimation of the time the server takes to elaborate and transmit the response for the first HTTP request[4] (e.g. an HTTP response).

2) Flow Goodput: it is defined as the rate at which information generated at Application Layer by the server is delivered to the client. Let T_{First} and T_{Last} be the timestamps of the first and the last packet data sent by the server and, let D be the size of the application level data sent by the server. The server goodput is thus defined as

$$G = \frac{D}{T_{Last} - T_{First}}. \quad (2)$$

To avoid the bias of short-lived flows and of Persistent-HTTP requests, the server goodput is evaluated only on flows in which the client sent exactly one data packet, and for which $D > 500kB$. Notice that HTTPS flows are automatically filtered out (requiring more than 1 data packet on the client side to complete the SSL handshake).

B. Network Cost

We aim at evaluating the cost sustained by the network to transport data generated by AWS servers to the end-users. To this extent, we define the Network Cost as the weighted average of the distance travelled by information units. Formally, given a flow, let $b(c, s)$ be the amount of Application Layer

[1] http://www.tstat.polito.it

[2] http://www.maxmind.com/app/ip-location

[3] We will interchangeably use terms datacenter and Availability Zone hereafter.

[4] The response time estimation can be affected by client requests that are longer than 1 TCP segment. We assume these cases are independent from the server, thus they do not bias the comparison.

	ID	#IPs		Exchanged Data (%)		β^{RTT} [ms]		β^{AS}
		EC2	S3	EC2	S3	EC2	S3	
Datacenters	IAD	6429	121	85.31%	64.22%	113.97	116.18	3
	DUB	1167	24	12.65%	35.14%	48.73	43.77	3
	SJC	632	12	1.71%	–	182.14	174.81	4
	NAR	18	0	–	–	–	–	4
	SIN	71	0	0.03%	–	228.10	–	3
	SEA	0	32	–	0.02%	–	214.79	4
				97.26GB	37.13GB			

	ID	#IPs	Exchanged Data (%)	β^{RTT} [ms]	β^{AS}
Caches	MXP	232	98.03%	21.26	3
	EU	208.5	1.14%	43.42	2.83
	NA	230.5	0.83%	142	3.5
	ASIA	76.6	-	-	3
			104.19GB		

Table I
SUMMARY OF MEASUREMENTS ON AMAZON'S DATACENTERS HOSTING EC2, S3 SERVICES (TOP) AND CLOUDFRONT CACHES (BOTTOM).

data a client c exchanges with a server s, and let $d(c, s)$ be the distance between client c and server s. The resulting network cost $\beta(s)$ for a given server s is computed as

$$\beta(s) = \frac{\sum_c d(c,s)b(c,s)}{\sum_c b(c,s)}. \quad (3)$$

The average network cost of servers in a datacenter $\mathcal{S}$ results

$$\beta(\mathcal{S}) = E[\beta(s)|s \in \mathcal{S}]. \quad (4)$$

We consider different definitions of distance, $d(c, s)$, in the following: i) the TCP connection average RTT, ii) the number of traversed AS on the path[5] or iii) the geodetic physical distance, leading respectively to $d^{RTT}(c, s)$, $d^{AS}(c, s)$, $d^{km}(c, s)$. Thus, we obtain different network cost metrics β^{RTT}, β^{AS}, β^{km}, respectively.

V. SPATIAL CHARACTERIZATION

We start by providing some aggregated information in Table I about the spatial distribution of AWS datacenters and caches, the traffic they generate toward monitored end-users, and its cost for the network.

A. EC2 and S3

The top part of the table reports information about both EC2 and S3. Each row in the table represents traffic associated to a different datacenter. Those located in Virginia (IAD), Ireland (DUB) and California (SJC) appear to be the most used datacenters from the perspective of an ISP placed in Italy.

Several observations hold. First, the number of detected IP addresses associated to EC2 service is much larger than any other service. This is due to the nature of EC2 service itself, that thanks to virtualization, it is capable of allocating, re-sizing and switching on/off independent EC2 instances. In general, each one could be reached by means of a different public IP address. For S3 instead, allocating too many IP addresses is needless since each particular content could coexist in same servers and can been accessed using different URIs. The pool of IP addresses needed for serving content is thus much smaller, as confirmed by values in Table I[6].

Second, the large unbalance in the number of instances (number of IP addresses in EC2 column) suggests that the datacenter located in IAD is the most popular among the ISP end-users, i.e., the most employed by AWS customers to run their EC2 instances. Furthermore, the column reporting the fractions of data generated by EC2 services shows that the IAD datacenter in the east coast of US is responsible for generating more than 85% of the total amount of traffic associated to EC2, i.e., 7 times larger than the volume handled by the DUB datacenter, the second popular among Italian customers. This suggests that IAD datacenter is much larger than all the others[7].

Interestingly, IAD EC2 (S3) generates more than 80GB (23GB) of data traffic in one day. Considering the user population of the monitored PoP, we can extrapolate that the IAD datacenter serves about 15TB of data per day to the all ISP end-users, i.e. 1.38Gb/s on average.

Surprisingly, such large amounts of data are exchanged with such a distant location. Given that Ireland is much closer to Italy than US, indeed, one may expect DUB to be the best candidate to host EC2/S3 instances for serving Italian (and European) end-users. All but β^{AS} network cost metrics, indeed, look sizeable for IAD, from 233% to 491% more expensive than the DUB datacenter. This may suggest that AWS customers, for the sake of a simple management and/or economical reasons, are more oriented to deploy their services on only one datacenter, and IAD may represent the first choice for AWS customers because of its lower price[8].

AWS offers load-balancing-based forwarders for incoming traffic to enhance performance of instances, but no location-aware policy is offered. Furthermore, recall that EC2 and S3 services are statically allocated to datacenters chosen by customers, and no automatic migration policy for instances/objects among datacenter is provided. This at the expenses of network cost, and, possibly, user experience. Observe how β^{AS} looks comparable for all datacenters, suggesting that Amazon (and the ISP) have good peering agreements with many providers.

At last, Fig. 1 (left plot) reports the evolution over time of the volume of data traffic seen from the top three datacenters for EC2. One point refers to a 4h long time interval; the first five days of the dataset, starting from Sunday, April 1st, 2012, are reported. Other datasets and periods of time show very similar trends: a very periodic pattern that follows busy period of end-users. IAD datacenter is consistently responsible for providing much larger amount of traffic with respect to DUB and SJC, confirming values presented in the top part of Table I. Same observation holds for S3 service (center plot in Fig. 1). In this case, DUB exchanges an amount of data slightly lower than IAD (notice the log scale that flattens differences).

B. CloudFront

Let us focus on CloudFront results reported in the second part of Table I. We report statistics about MXP (Milan)

[5]The number of traversed AS is obtained running a traceroute from the vantage point and checking the AS of returned routers.

[6]Same observations hold for CloudFront.

[7]Confirmed by http://aws.amazon.com/about-aws/globalinfrastructure/

[8]http://aws.amazon.com/ec2/spot-instances/

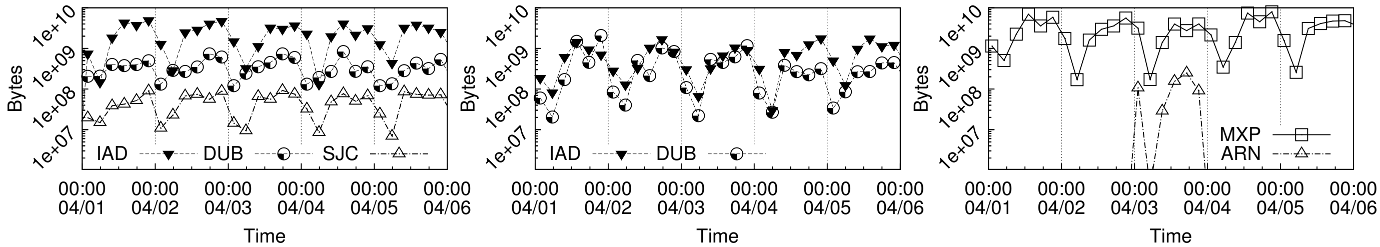

Figure 1. Evolution over time of data traffic volume for EC2 (left), S3 (center) and CF (right) services.

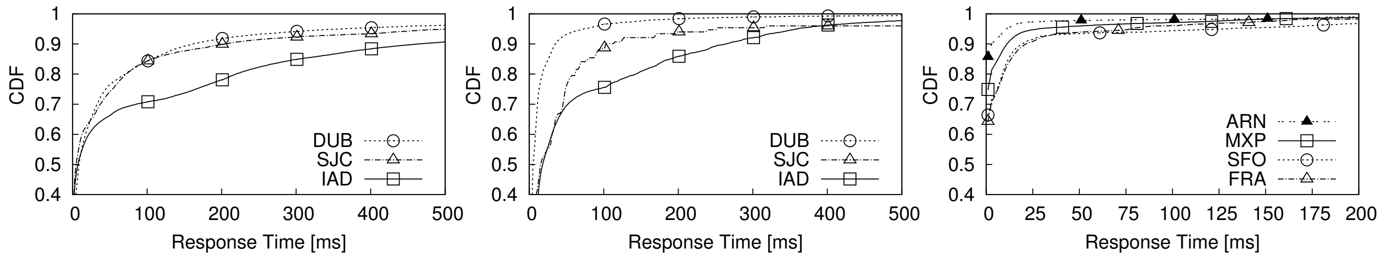

Figure 2. Distribution of response time ΔR for EC2 (left), S3 (center) and CF (right) services.

cache, and statistics for other caches averaged on a continent basis. Observe how biased is the preference towards the MXP (Milan) cache, which results the best cache considering any definition of network costs.

This has been validated by running an active experiment in which we resolved 100 different services hosted by CloudFront considering more than 2,000 DNS servers scattered worldwide. As a side discovery of this process, we identified 33 different CloudFront caches, each hosting a /24 subnet. The bottom part of Table I refers to the CloudFront caches whose servers were detected in our passive measurements too.

Overall, we can conclude that the CDN policy selection of CloudFront is effective in directing ISP end-users to the closest cache (MXP in Italy), as expected for a CDN. However still less than 2% of traffic is delivered from caches far away from end-users' position. This may be because of some end-users employing alternative DNS servers, different from those provided by their ISP. For instance, both OpenDNS and Google DNS servers cause requests from the ISP end-users to be directed to FRA (Frankfurt). This is consistent with findings in [6].

Fig. 1 (right plot) reports the evolution over time of the volume of data traffic for the top two European caches, i.e. MXP and ARN. The pattern is regular for cache placed in Milan. However, this does not hold for ARN, in Stockholm, which presents an unusual peak on the third day of measurements, precisely from 10pm of April 2 to 6pm of April 3. Investigating further, we verified that this was due to an intentional change in the Amazon DNS policies. Indeed, many end-users that were typically served by MXP had been redirected to ARN during that period. While it is impossible to know why this happened, it allows to conclude that CloudFront policies are dynamic, in contrast with the static allocation of the EC2/S3 services.

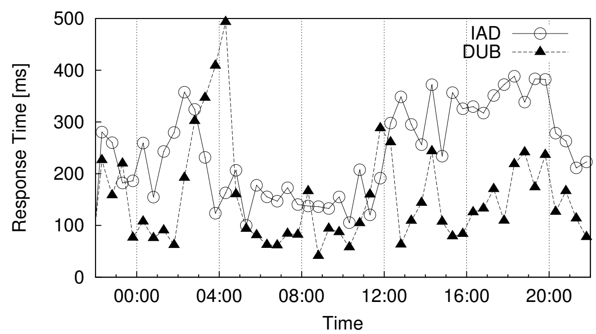

Figure 3. Evolution over time of average ΔR for EC2 datacenters.

VI. Performance Evaluation of AWS

A. Availability Zones and Caches Performance Evaluation

Fig. 2 depicts the distribution of the estimated response time for EC2, S3 and CloudFront on left, center and right plot, respectively. Top popular datacenters/caches are shown. Data refers to a single day of April 2012.

Focusing on the performance of different locations, EC2 in IAD shows response times larger than 100ms in 30% of the cases, resulting the worst performing datacenter. However, the average bad performance of IAD could be caused by popular and poorly performing services running on congested instances. Indeed we found out that some services suffer from extremely poor design. For instance, content *proxy.eu.mydlink.com* served from DUB, shows ΔR larger than 100s during some periods! DUB appears to be the best choice among datacenters for S3, while it competes with SJC in the case of EC2.

We complement above results with Fig. 3, which reports the evolution over time of $E[\Delta R]$ for a period of one day for EC2 in IAD and DUB. Measurements confirm previous findings, with IAD consistently performing worse on average than DUB. Notice that the average is i) a strongly non-stationary measure (being it biased by the different contents

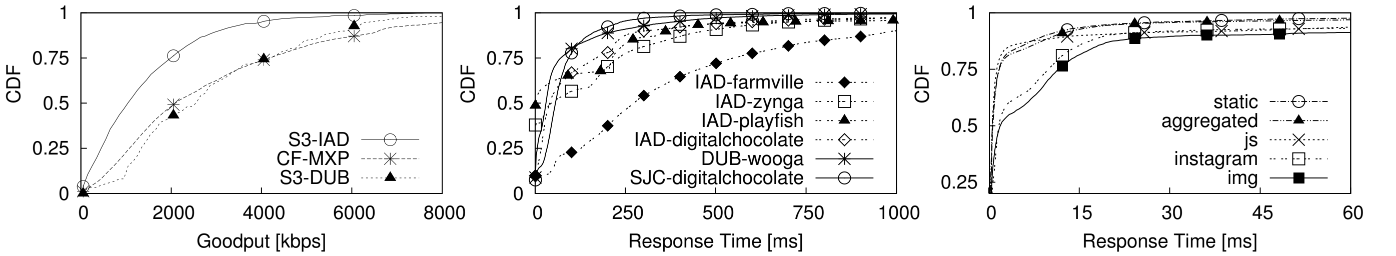

Figure 4. Cumulative distribution function of goodput G for the two most used S3 datacenter, IAD and DUB, and for MXP CloudFront cache 4(a). Distribution of response time ΔR for EC2 social gaming services 4(b). Distribution of response time ΔR for different kinds of contents provided by CloudFront cache located in MXP 4(c).

retrieved at different times), and ii) practically independent on the datacenter load.

Moving to CloudFront, right plot in Fig. 2 shows in general very good performance, being 83% of requests satisfied in less than 20ms in FRA, the worst performing cache. MXP and ARN caches serve 80% of requests in less than 3ms; SFO and FRA serve only 65% of request in less than 3ms, respectively.

Fig.4(a) compares the distributions of goodput G of S3 at IAD and DUB, together with CloudFront MXP cache. More than 50% of flows get a goodput $G > 2$Mbit/s for S3 in DUB and CloudFront in MXP. For S3 in IAD, only 21% of flows can achieve $G > 2$Mbit/s. This difference may be due to the large RTT running from our vantage point to IAD, that affects the TCP congestion control, thus, reducing achievable goodput.

B. Per-content Performance Evaluation

Fig. 4(b) reports the distribution of the response time ΔR for different social gaming services hosted by different datacenters. Notice that all social games, e.g. Farmville, hosted by IAD present poor performance with respect to those hosted by DUB and SJC.

Focusing on the performance of CloudFront service, we report in Fig. 4(c) the distribution of ΔR for several kinds of contents that end-users downloaded from MXP cache. *Static* refers to static content for web pages (e.g. HTML files), *js* represents JavaScript files, *img* refers to binary data such as images and *Instagram* is referred to contents related to the well-known photo-sharing service. *Aggregate* reports the behavior of all services together. As previously noticed, CloudFront shows really good performance, being able to process 50% of requests in less than 2ms, independently on the kind of content. However, ΔR is consistently smaller on average for static and JavaScript files which are mostly static too, whereas images and Instagram contents show larger response time. This may be due to the nature of the user-generated contents that are the most critical to manage for content delivery services, because of the size of the catalogue, and of the small popularity of each single content [10].

VII. Conclusions

To the best of our knowledge, this is the first work that characterizes *Amazon Web Services* (AWS) traffic from passive measurements.

We presented an extensive characterization of AWS offerings, in particular for EC2, S3 and Amazon's CDN, CloudFront. Results show that there is a big workload unbalance among different datacenters hosting both EC2 and S3 products; in particular, the datacenter in Virginia is responsible for 85% of the total traffic sent to Italian end-users, despite the availability of a datacenter in Ireland. We observed that companies which rely on EC2 and S3 concentrate their content mostly on one datacenter, thus i) increasing the cost sustained by the network to carry data to faraway end-users and, ii) increasing risk in case of failures. Considering end-users performance, our results show that the datacenter in Virginia exhibits in general poorer performance, but we could not pinpoint the actual causes.

We also found that CloudFront shows excellent performance, but presents issues that are typical of other CDN systems: i) generic DNS servers returning caches far from end-users; ii) lower performance when processing unpopular user-generated contents.

References

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008.

[2] E. Walker, "Benchmarking Amazon EC2 for High-Performance Scientic Computing," *USENIX ;login: Magazine*, October 2008.

[3] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The Cost of Doing Science on the Cloud: The Montage Example," in *SC*, Austin, TX, November 2008, pp. 1 –12.

[4] S. L. Garfinkel, "An Evaluation of Amazons Grid Computing Services: EC2, S3, and SQS," Center for Research on Computation and Society, School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, Tech. Rep., 2007.

[5] G. Wang and T. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in *IEEE INFOCOM*, San Diego, CA, March 2010, pp. 1–9.

[6] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Comparing DNS Resolvers in the Wild," in *ACM IMC*, Melbourne, AU, November 2010, pp. 15–21.

[7] A. Finamore, M. Mellia, M. Meo, M. M. Munafò, and D. Rossi, "Experiences of Internet Traffic Monitoring with Tstat," *IEEE Network*, vol. 25, no. 3, pp. 8–14, 2011.

[8] I. Bermudez, S. Traverso, M. Mellia, and M. Munafò, "Large scale observation and analysis of Amazon AWS traffic," Politecnico di Torino, Tech. Rep., 2012. [Online]. Available: http://www.retitlc.polito.it/traverso/papers/amazon-techrep.pdf

[9] I. Bermudez, M. Mellia, M. Munafò, R. Keralapura, and A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web," in *ACM IMC*, Boston, MA, November 2012.

[10] B. Ager, F. Schneider, J. Kim, and A. Feldmann, "Revisiting cacheability in times of user generated content," in *IEEE INFOCOM*, San Diego, CA, March 2010, pp. 1–6.