

Large scale observation and analysis of Amazon AWS traffic

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

Abstract—Cloud Providers are nowadays the most popular way to quickly deploy new services on the Internet. Understanding mechanisms currently adopted in cloud design is fundamental to identify possible bottlenecks, to optimize performance, and to design more efficient platforms.

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 collected from several vantage points in Italy for several months, 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 single Amazon datacenter located in Virginia despite it 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 outages, as we were able to observe in our data.

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). Active since July 2002, they offer computing and storage cloud solutions. The most well-know Amazon cloud services are “Elastic Compute Cloud” (EC2), and “Simple Storage Service” (S3), with “CloudFront” (CF), the Content Delivery Network (CDN), that has augmented Amazon’s portfolio in late November 2008.

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 *content* without carrying on costs and risks of building and managing their own hardware and infrastructure. Given the great scala-

bility and extremely low costs of pay-as-you-go cloud services, many successful companies like Dropbox, Zynga and Netflix to name a few, have been attracted by AWS and successfully relies on them. Amazon indeed represents today one of the largest source of traffic in nowadays Internet, accounting for about 3.1% of the total HTTP/HTTPS traffic flows, i.e., one fifth of Google’s [2].

Given its success, AWS has gained a large interest within the research community. In particular, many works specifically focus on the possibility of exploiting AWS EC2 for research purposes [3], [4]. Other works instead focus on evaluating the performance of AWS computing and networking virtual resources [5], [6], [7]. 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, where the actual workload and performances can be evaluated by means of “passive” observation of traffic. Only by the passive characterization of services it is possible to discover eventual performance degradation and, most of all, to gauge their impact on end-users.

The goal of this paper is thus to provide an extensive study of AWS through the passive analysis of network 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 during which the traffic generated by more than 50.000 end-users has been observed.

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, proposing 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 characterizations is a challenging task due to the nature of cloud services, where encryption schemes and proprietary solutions are very common, and virtualization allows to share resources and dynamically move content and services over time.

This paper represents the first attempt of observing the Amazon cloud from passive measures. We follow the direction suggested by other works that provide a characterization of popular services such as social networks [8], [9] or YouTube [10]. Our main findings are:

- Among the seven EC2 and S3 datacenters, the one placed

in Virginia, close to Washington DC, is the most used one, hosting more than 6.000 EC2 virtual machines 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 contents from EC2 and S3 systems tend to rely upon one datacenter only. This makes the network to pay the large cost of carrying information 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. This is confirmed by the results we provide about the outage happened on the 30th of June 2012, in Virginia.
- Performance of datacenters in terms of response time (for EC2) and goodput (for S3) show that the most popular datacenter is also the worst performing one. Evidence shows that some contents suffer because of under-provisioned instances, 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-user. The remaining 2% of traffic comes from worldwide caches. This happens to be due to CloudFront directing traffic to other caches for load-balancing purpose. A minimal fraction can be due to incorrect DNS configuration of end-user Clients [11].

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

The rest of this paper is organized as follows: Sec. II describes products offered by AWS, and Sec. III overviews the data collection procedure and the datasets we study. In Sec. IV and Sec. V we present the techniques and the metrics, respectively, we employed for the analysis of our datasets. We then show the results of our study, starting with a spatial characterization of AWS infrastructure (Sec. VI), moving on the analysis of properties of hosted contents (Sec. VII) and ending with the performance evaluation of AWS platform (Sec. VIII). Finally, Sec. IX concludes the paper.

II. AMAZON WEB SERVICES PRIMER

Amazon Web Services offers on-demand cloud computing services to web companies. Often AWS is accessed through HTTP, REST and SOAP protocols. In addition TLS/SSL encryption is optional to guarantee privacy on network communications. Throughout this paper we focus on the three most popular AWS services which are used by Amazon’s customers to reach the end-users:

- **EC2** is an on-demand virtual computing environment supported by Xen virtualization [12]; EC2 lets customers rent free-to-use virtual machines, called *instances*, to run any applications. EC2 instances can be allocated on seven datacenters (called also *Availability Zones* by AWS terminology¹) world-widely distributed. Customers can choose any Availability Zone as support for their instances. Inside the Availability Zone, AWS provides an optional DNS-based load-balancing service, called Elastic Load Balancing, whose task is to uniform the workload over rented instances. To the best of our knowledge, no automatic tool for migration of instances among different Availability Zones has been implemented yet. Examples of web applications relying on EC2 are social-gaming applications like Zynga and Playfish, or social networks like FourSquare.

- **S3** offers storage services through standard interfaces (REST, SOAP, and BitTorrent). Data are stored by means of “objects” whose size can span from 1B to 5TB each. S3 is reported to store more than a trillion objects as of June 2012². EC2 and S3 are co-located in each Availability Zone.

- **CloudFront** is the Amazon’s Content Delivery Network (CDN) for distributing content from locations near to end-users, thus guaranteeing low latency access and high data transfer speeds. CloudFront can deliver dynamic, static as well as streaming content, using a unknown number of caches placed at Internet Exchange Points around the globe. An example of application supported by CloudFront is Instagram, which exploits CloudFront network to distribute user-generated content among end-users.

III. DATASET

We rely on passive measurements to characterize AWS services in operational networks. We employ Tstat, the open-source traffic monitoring tool developed at Politecnico di Torino, to analyze packets exchanged by actual end-users inside monitored vantage points [13].

Tstat was installed in four different vantage points where it has been collecting measures from April to June 2012, observing more than 50.000 end-users normally accessing the Internet. The resulting dataset is large enough to reveal significant information about AWS infrastructure as well as about end-users’ habits and performance of the content served by the AWS cloud.

For the sake of brevity, being the goal of this paper the understanding of AWS, we restrict our analysis on measurements obtained from *Home 1* PoP, an ISP PoP in a large city in Italy, where about 15.000 ADSL lines are active. The entire week starting from April 1st, 2012 is considered, during which 6M TCP connections were directed to AWS servers, for a total of more than 340GB of data exchanged overall. We have repeated the analysis considering traffic from other two ISP PoPs and from our Campus network, and at different periods of time. We found that findings from *Home 1* are general and not biased,

¹<http://aws.amazon.com/about-aws/globalinfrastructure/>

²<http://aws.typepad.com/aws/2012/06/amazon-s3-the-first-trillion-objects.html>

being the results collected from the other datasets practically identical. We acknowledge that some of the results presented in this paper are biased by observing AWS traffic from a single country. Naturally, we expect some of the results to change based on different cultural influence and geographical location of the vantage point.

IV. ANALYSIS METHODOLOGY

Several challenges have to be faced when passively monitoring cloud services: the separation between *content* and *server* makes it hard to identify which content is actually accessed by the end-users, and in which datacenter the server delivering the content is placed. In cloud services indeed, virtualization, load-balancing and migration techniques do not allow to simply rely on information collected from the Network Layer to identify the content, i.e., the server IP address and its official owner (retrieved from *whois*) give no information on the content being used.

In addition, the increasing adoption of TLS/SSL encryption by large content providers (including most of those running in the AWS platforms [2]) makes classic Deep Packet Inspection (DPI) useless.

For example, consider the same AWS server hosting *www.acmegame.com* and *www.acmeshop.com* content over HTTPS. The IP addresses of instances hosting them allow only to identify that both are handled by AWS (being the IPs registered to Amazon). Hence, no indication about handled contents can be achieved looking at information from Network Layer. Furthermore, identifying which datacenter is being used, and where it is located is not trivial.

A. Content Discovery

To identify the content retrieved by end-users from a AWS server, Tstat has been augmented to implement DN-Hunter [14], in such a way that the improved Tstat snoops DNS queries performed by clients to resolve URLs, and uses this information to tag network flows extracting the right Fully Qualified Domain Name (FQDN). In the example above, end-user’s client has first to resolve the *www.acmegame.com* hostname into a list of IP addresses by contacting the DNS server. Obtained the list of IP addresses, the client contacts one of them to fetch the actual content. DN-Hunter caches all DNS responses, and associates the original hostname to the actual server IP addresses being contacted, associating *www.acmegame.com* to the observed TCP flows. This key feature allows Tstat to recover the original content name requested by the client and being served by an AWS server. Details on DN-Hunter are out of scope of this paper, and can be found in [14].

B. Datacenter and Service Discovery

To identify the AWS traffic, and which platform is serving it, we adopt a the following methodology. First, we proceed isolating all known Amazon’s IP addresses as listed by the MaxMind³ organization database, or equivalently returned by

the *whois* database. Once Amazon IP addresses are identified, we differentiate the traffic by EC2, S3 or CloudFront services. Also in this case, we rely on the information provided by the DNS. AWS indeed follow a strict naming rule for EC2: the instance IP address *a.b.c.d* is registered with a Type-A DNS record as **ec2-a-b-c-d.XXXXXX.amazonaws.com**, where XXXXXX is a variable string. A simple DNS reverse lookup from the IP address allows to discover that *a.b.c.d* hosts an EC2 instance.

Unfortunately, the Type-A record for an IP address of S3 and CloudFront does not always reveal which service that server is providing. Hence, we employ a technique we called *IP-knocking*: a HTTP HEAD request is sent to inspect the `Server` field returned in the HTTP response; either `AmazonS3` or `CloudFront` are always returned in the HTTP response, allowing easy classification of the service hosted by that IP address.

We have found that the address space used by CloudFront and S3 servers is separated. Furthermore, we have verified that the allocation of IP addresses to these services is “static”, allowing IP-knocking to be performed once for each IP address.

Notice that IP-knocking cannot be employed for EC2 since instances are managed by Amazon’s customers who are free to run different Operating Systems as well as different HTTP servers.

IP-knocking and name reversing have been performed over any detected Amazon IP address in our datasets, letting us identify every IP address associated to EC2, S3 and CloudFront.

C. Server Geolocation

Last problem to solve is to geolocate the servers. Geolocation of IP addresses represents a well-known problem. Common public databases, such as RIPE⁴, ARIN⁵ or MaxMind, do not represent reliable sources when seeking for information about the physical location of a machine associated to a given IP address [15].

A simple, yet accurate, approach is to exploit again information provided by the Type-A DNS record assigned to servers by Amazon administrators. Indeed, the Type-A record returned when performing the reverse lookup of AWS IP addresses often unveils information about server placement. In particular, the standard International Air Transport Association (IATA) airport location identifiers are used by Amazon in the form *server-a-b-c-d.AIR.r.cloudfront.net*, where AIR is the IATA airport code of the nearest large airport. Unfortunately, not all Type-A entry of AWS servers include the IATA code, e.g., *s3-1.amazonaws.com*.

To geolocate IP addresses for which no IATA code is found, we rely on active `traceroute`. Intuitively, when running `traceroute` towards a given target IP address, we look for the DNS names of routers returned along the path. Many of

³<http://www.maxmind.com/app/ip-location>

⁴<http://www.ripe.net/db/index.html>

⁵<http://www.arin.net>

these names reveal routers' location. We call the last router on the path for which location is revealed a *pivot* router. It represents the closest router to the targeted AWS IP address whose geographical position is known. The difference of RTT from the pivot to the targeted IP address is a measure of the geographical distance among the two nodes, so that the smaller delay, the smaller their distance. This allows to position IP addresses close to well-known locations. The approximation obtained by running this simple algorithm have been proven to be almost perfect on those AWS servers whose position has been cross-checked via the IATA code (RTT errors below 1ms on average). Due to lack of space, we do not provide performance evaluation of the algorithm in this paper.

In the remainder of this paper, we use IATA codes to identify datacenters instead of conventional names of AWS Availability Zones.

V. MEASUREMENTS DEFINITION

A. Per-flow Metrics

In our study we consider each TCP flow that Tstat passively tracked. When running on the vantage point, Tstat observes packets, rebuilds each TCP flow, tracks it, and, at the flow end, logs more than 100 metrics [13]. Among the different measurements, we consider the server IP address, its FQDN 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 straightforward to monitor and we direct the reader to [13] and references therein for more details.

More complicated metrics can be extracted from the log files. In particular, we define:

1) *Response Time*: it is the time a server employs to start sending 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 and 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 to transmit the response to the first HTTP request⁶(e.g. an HTTP response). For HTTPS flows, ΔR represents the time taken by the server during the SSL handshake to send the first SSL message.

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 sent by the server and carrying data. Let D be the size of effective 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 Internet to transport data generated by AWS servers to the monitored end-users. To this extent, we define the Network Cost as the weighted average of the distance traveled by information units. Formally, given a flow, let $b(c, s)$ be the number of Application Layer bytes 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

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

Observe that distance $d(c, s)$ can be defined in different ways, e.g., considering i) the average RTT, ii) the number of traversed AS on the path⁷ or iii) the geodetic physical distance, leading respectively to $d^{RTT}(c, s)$, $d^{AS}(c, s)$, $d^{km}(c, s)$. Given these different definitions, we can obtain different network cost metrics β^{RTT} , β^{AS} , β^{km} , respectively.

VI. SPATIAL CHARACTERIZATION

We start by providing some aggregate information about the spatial distribution of datacenters, the traffic they generated toward monitored end-users, and its cost for the network.

Table I provides the breakdown of the AWS traffic distinguishing the identified datacenters.

A. EC2 and S3

the top part of the table reports the list of locations where both EC2, S3 services were detected. Those located in Virginia (IAD), Ireland (DUB) and California (SJC) appear to be the largest datacenters hosting AWS from the point of view of our vantage points placed in Italy.

Several observations hold. Focusing on the size of datacenters, observe how the number of detected IP addresses associated to EC2 service is much larger than to any other service. This is due to the nature of EC2 service itself, that, thanks to virtualization, is capable of allocating, re-sizing and switching on/off independent EC2 instances, in general each provided and reachable by a different public IP address. S3 and CloudFront, instead, offer storage services that are

⁶The 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.

⁷The number of traversed AS is obtained running a traceroute and checking the AS of returned routers.

	ID	#IPs		Exchanged Data (%)		Avg. RTT [ms]	β^{RTT} [ms]		β^{AS}	β^{km} [km]
		EC2	S3	EC2	S3		EC2	S3		
Datacenters	IAD	6429	121	85.31%	64.22%	132.13	113.97	116.18	3	6709
	DUB	1167	24	12.65%	35.14%	45.10	48.73	43.77	3	1365
	SJC	632	12	1.71%	–	203.06	182.14	174.81	4	9556
	NAR	18	0	–	–	298.67	–	–	4	9843
	SIN	71	0	0.03%	–	235.60	228.10	–	3	10390
	SEA	0	32	–	0.02%	196.04	–	214.79	4	8617
					97.26GB	37.13GB				
Caches	IAD	2	–	–	–	132.13	102.75	–	3	6709
	DUB	222	–	0.05%	–	45.10	49.76	–	3	1365
	SJC	–	–	–	–	–	–	–	4	9556
	NAR	115	–	–	–	298.67	–	–	4	9843
	SIN	51	–	–	–	235.60	–	–	3	10390
	SEA	64	–	–	–	196.04	–	–	4	8617
	SFO	253	–	0.83%	–	172.80	175.21	–	4	9537
	CDG	246	–	0.13%	–	32.09	38.43	–	3	584
	FRA	245	–	0.17%	–	19.56	21.87	–	2	566
	MXP	232	–	98.03%	–	18.34	21.26	–	3	124
	EWR	208	–	–	–	105.28	109.53	–	3	6380
	AMS	205	–	0.04%	–	23.94	29.88	–	3	837
	LHR	182	–	0.17%	–	31.84	31.60	–	3	920
	ANR	151	–	0.56%	–	41.02	41.48	–	3	1734
					104.19GB					

Table I
SUMMARY OF AMAZON’S DATACENTERS HOSTING EC2, S3 SERVICES (TOP) AND TOP 14 CLOUDFRONT CACHES WE LOCATED (BOTTOM).

implemented at application level, and data are made accessible through URI pathnames. The pool of IP addresses needed to keep the service alive is much smaller then, as confirmed by values in Table I.

The large unbalance in the number of instances (number of IP addresses in EC2 column) suggests that the Availability Zone 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. This suggests that IAD datacenter is much larger than all the others⁸. The column reporting the fractions of data generated by EC2 services further confirms this, being the IAD datacenter responsible for generating more than 85% of the total amount of data produced by EC2. It is 7 times higher than the volume handled by the DUB (Dublin) datacenter, the second largest in our ranking.

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.

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 to be DUB 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, are more oriented to deploy their services on one Availability Zone. IAD may represent the first choice for AWS customers because of its lower price⁹.

⁸Confirmed by <http://aws.amazon.com/about-aws/globalinfrastructure/>

⁹<http://aws.amazon.com/ec2/spot-instances/>

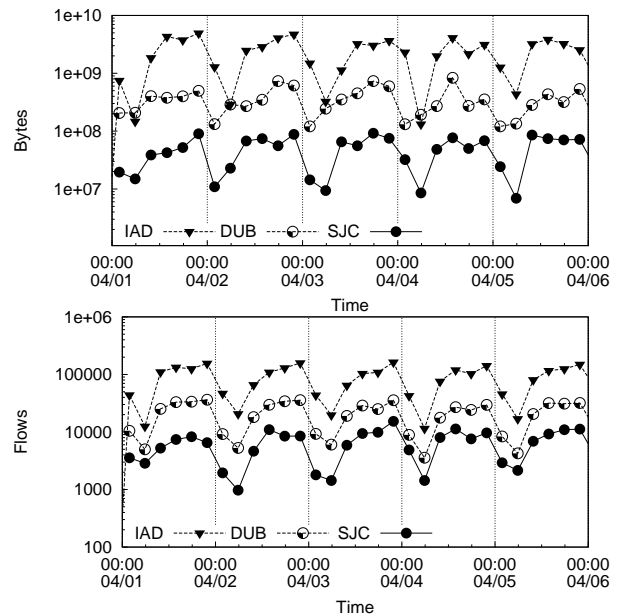


Figure 1. Evolution over time of data traffic volume (top) and traffic flows (bottom) for EC2 service.

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 Availability Zones that are 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 Availability Zones, suggesting that Amazon (and the ISP) have good peering agreements with many providers.

We can enrich the picture about EC2 focusing on plots in

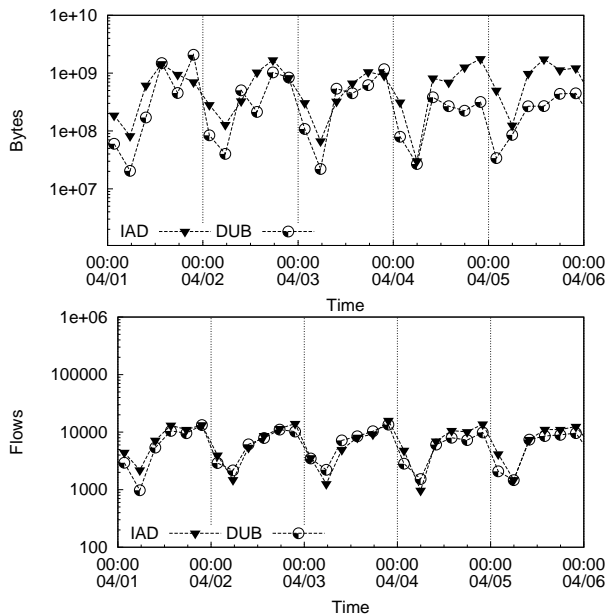


Figure 2. Evolution over time of data traffic volume (top) and traffic flows (bottom) for S3 service.

Fig. 1, where we report the evolution over time of the volume of data traffic (top) and of the number of flows (bottom) seen from the top three EC2 Availability Zones. 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 handling a much larger amount of traffic with respect to DUB and SJC. This is consistent with values presented in the top part of Table I, and it confirms the static allocation of EC2 instances to Availability Zones.

Same observation holds for S3 service, as reported in Fig. 2. In this case, DUB exchanges an amount of data slightly lower than IAD (notice the log scale that flattens differences).

Comparing the number of flows end-users exchange with EC2 and S3 (bottom plot of Fig. 1 and Fig. 2, respectively), it is possible to notice that S3 traffic is made of flows that carry more data than EC2 flows, i.e., more elephants than mice in S3, and vice-versa for EC2. This is confirmed by observing the flow size Cumulative Density Function (CDF), not reported here due to lack of space. Interestingly, comparing the conditional CDF of different Availability Zones, marginal differences are seen.

B. CloudFront

let us focus on CloudFront traffic breakdown, reported in the second part of Table I. Interestingly, observe how biased is the preference toward the MXP cache, located in Milan, which results to be the best cache considering any definition of network costs. As for most CDN, this confirms that CloudFront relies on DNS load-balancing to direct the user to the closer cache. Simply, whenever a client queries the local DNS server for a CloudFront IP address, the local DNS server forwards

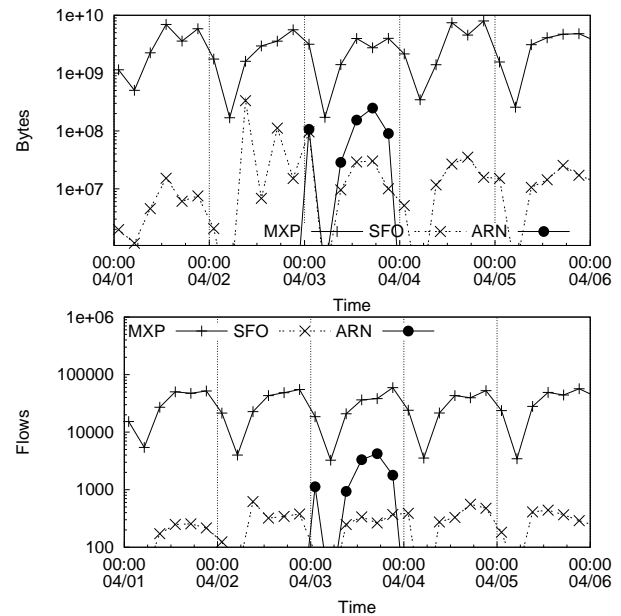


Figure 3. Evolution over time of data traffic volume (top) and traffic flows (bottom) for CloudFront service.

the query to the authoritative CloudFront’s DNS server, whose reply will direct the client to the geographically closer cache. This happens regardless the type of retrieved content.

This has been confirmed by running an active experiment in which we resolved 100 different FQDN hosted by CloudFront considering more than 2000 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 reports the top CloudFront caches whose servers were detected in our passive measurements. Focusing on the column reporting the location of detected caches, it can be seen that EC2/S3 Availability Zones host also CloudFront caches, even if ISP end-users are seldom directed to any of these.

Overall, we can conclude that the CDN policy selection of CloudFront is very effective in directing ISP end-users to the closest MXP cache. Less than 2% of traffic was delivered from caches far away from end-users’ position. This can be explained by the fact that some end-users could rely on alternative DNS servers different from those provided by their ISP; the Amazon Authoritative DNS would reply directing traffic to caches located close to the used DNS server, but eventually far from the ISP. For instance, both OpenDNS and Google’s DNS servers causes requests from the ISP end-users to be directed to FRA. This is consistent with findings in [11].

Fig. 3 reports the evolution over time of the volume of data traffic (top) and of the number of flows (bottom) for the top three caches, i.e. MXP, SFO and ANR. Regular patterns are present for caches placed in Milan and San Francisco. However this does not hold for ANR, in Stockholm, where it 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 not due to some unusual DNS end-users’ setting, but to an intentional change in the Amazon DNS policies. Indeed, many end-users and contents

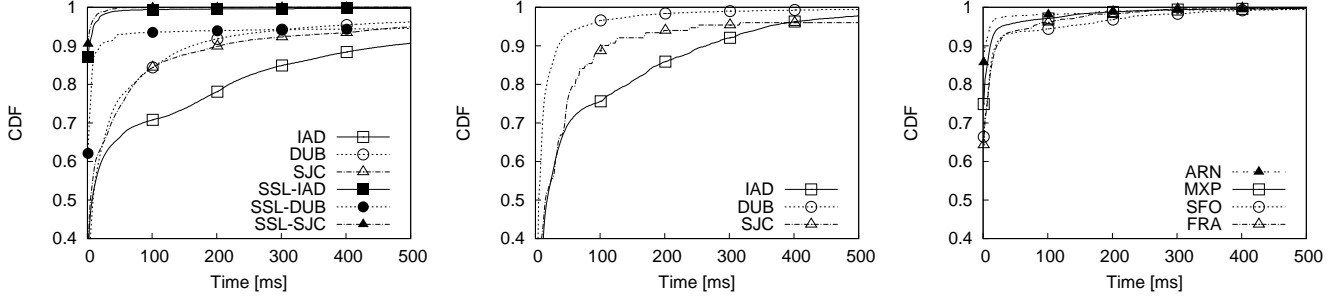


Figure 4. Distribution of response time ΔR for EC2 (left), S3 (center) and CF (right) services.

that were typically available from MXP had been redirected to ARN during that period. This is similar to what has been observed for YouTube CDN [10].

While it is impossible to know why this happens, it is observed that CloudFront policies are dynamic, in contrast with the static allocation of the EC2/S3 content.

VII. CONTENT ANALYSIS

Around 50% of contents observed on CloudFront presents a size smaller than 10kB, mostly probably being CSS or JavaScript files employed for the rendering of web pages. Other 20% of contents, which present a size larger than 100kB, is binary data, e.g., Flash objects or images. Differences between sizes of contents served by different caches are negligible. Files hosted on S3 show in general the same size distribution of files hosted by CloudFront. The average size of contents distributed by CloudFront and S3 is 78kB for both. Flows directed to EC2 carry smaller data in general, being the 60% of them smaller than 1kB. These could be small XML files or messages directed to APIs. For these files, the TCP three-way-handshake and tear-down procedures last longer than the data transfer. Even for EC2, no significant difference among different locations is evident. The average size of contents served by EC2 flows is 26kB.

IAD			DUB		
Service	%Flows	%Vol.	Service	%Flows	%Vol.
zynga	14	5	wooga	24	4
farmville	13	6	invitemedia	18	3
playfish	9	3	cdn.com	6	0.2
widdit	6	1	360yield	6	0.1
chartbeat	3	>0.1	mydlink	2	>0.1
dropbox	0.4	59	wetransfer	>0.1	16

Table II

THE TOP CONTENTS HOSTED BY EC2 BY NUMBER OF FLOWS OR VOLUME.

In Table II we report the most popular contents served by EC2, i.e., those contents which are generating the largest number of flows, or volume. The type of content/service is quite heterogeneous, with the larger portion of flows generated by social games, such as Zynga, Farmville, Wooga and DigitalChocolate, and by advertising companies, e.g., InviteMedia, 360yield. Notice how none of these services is replicated among different Availability Zones, their traffic coming exclusively from a single datacenter. The only notable service we have found which is present on both IAD and SJC

datacenters is DigitalChocolate; it relies on the former for the distribution and execution of games, while on the latter for the management of system logins and subscriptions.

Considering the top services by volume, we can notice the important presence of storage services like Dropbox and WeTransfer. The former is responsible for 59% of volume of data handled by EC2 alone, and it is only available from IAD!

Contents hosted by S3 presents similar properties of EC2. S3 contents are in general services for storage, advertisement and social games, but even in this case none of them results distributed over different Availability Zones.

VIII. PERFORMANCE EVALUATION OF AWS

In this section we evaluate the performance of AWS on a Availability Zone and on a content basis.

A. Availability Zones and Caches Performance Evaluation

Fig. 4 depicts the distribution of the estimated response time ΔR for EC2, S3 and CloudFront on left, center and right plot, respectively. Top popular datacenters are shown. Data refer to a single day of April 2012.

Focusing on the performance of different locations, EC2 in IAD shows values of ΔR larger than 100ms in 30% of the cases, resulting the worst performing datacenter. These plots combined with information about the large number of IPs allocated on IAD might suggest that IAD is overloaded because of the large number of hosted EC2 instances. However, the average bad performance of IAD could be caused by popular and poorly performing contents running on congested instances. We will return on this in Sec. VIII-B.

DUB appears to be the best choice among Availability Zones for S3, while it competes with SJC in the case of EC2.

Since a sizable part of content hosted on EC2 is encrypted using TLS/SSL (14% of flows), we report the response time for HTTPS flows in left Fig. 4. Recall that ΔR is a measure of the server reactivity in the SSL handshake in this case. Both IAD and SJC look very reactive, with 93% of the cases responding to SSL initial negotiation almost with no delay. This conflicts with the hypothesis of IAD being congested. DUB instead shows large values of ΔR for 10% of the flows, that we have found out to be due to a poor design of *proxy.eu.mydlink.com* content. Flows directed to this content suffer more than 10s of response time.

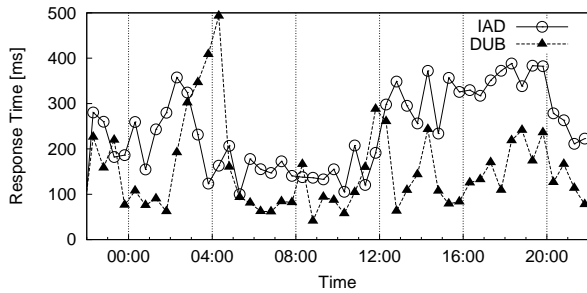


Figure 5. Evolution over time of observed ΔR for EC2 datacenters.

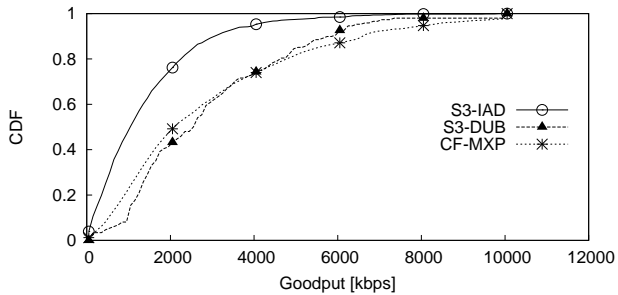


Figure 6. Cumulative distribution function of goodput G for the two most used S3 datacenter, IAD and DUB, and for MXP CloudFront cache.

This finding supports the idea that in general it is not the entire EC2 datacenter to be congested, but rather some instances running on it.

We complement results described above with Fig. 5 which reports the evolution over time of $E[\Delta R]$ for EC2 for a period of one day for IAD and DUB. Measurements confirm previous finding, 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 retrieved at different times), and ii) practically independent on the datacenter load.

Moving to CloudFront, right plot in Fig. 4 shows in general very good performance, being 83% of requests satisfied in less than 20ms in the worst case, i.e., FRA. MXP and ARN caches elaborate 80% of requests in less than 3ms; SJC and FRA serve only 65% and 55% of request in less than 3ms, respectively.

Fig.6 compares the distributions of goodput G . We compare the performance of two main datacenters for S3, IAD and DUB, together with CloudFront MXP cache. The plot shows that more than 50% of flows get a goodput larger than 2Mbit/s for S3 in DUB and CloudFront in MXP. For S3 in IAD, only 21% of flows can achieve a goodput larger than 2Mbit/s. This difference can be due to the larger RTT running from our vantage point to IAD, that affects the TCP congestion control, thus, reducing achievable goodput.

B. Per-content Performance Evaluation

Fig. 7 reports the distribution of the response time ΔR for different social gaming services hosted by different Availability Zones. Notice that all social games hosted by IAD present poor performance with respect to those hosted by DUB and

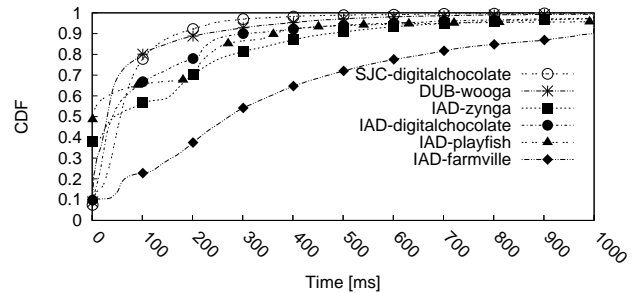


Figure 7. Distribution of response time ΔR for EC2 social gaming services.

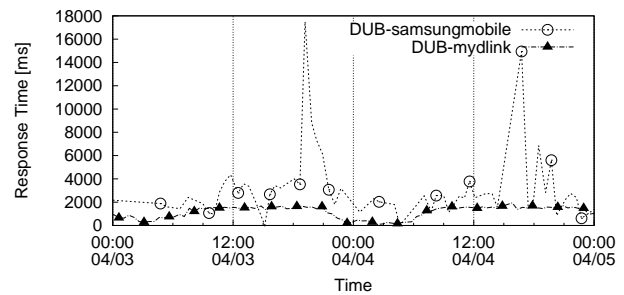


Figure 8. Evolution during time of ΔR for two poorly performing contents hosted on EC2.

SJC. This suggests again that IAD datacenter suffers from congestion due the large number of instances it is hosting. Observe how the instances of Farmville, a popular game, are indeed performing poorly.

Congestion may affect single instances. For example, Fig.8 reports two examples of applications hosted by EC2 in DUB that suffer large average ΔR . These two applications, SamsungMobile and MyDlink, show really impaired performance, with average response time higher than 2s. Recall that DUB is the best performing Availability Zone in our measurements (cfr. Fig. 4). This suggests that such poor performance is due to a bad dimensioning of the instance, and not due to datacenter issues.

At last, we show the impaired performance following the outage of IAD datacenter on June 30th, 2012¹⁰. We selected two contents hosted at IAD, and a third one at DUB. The plot shows that i) not all contents were affected by the IAD outage, ii) it had no impact on DUB, iii) affected instances suffer a 100 fold worse performance during the failure, and iv) they kept suffering for performance issues for several hours after the fault.

Focusing on the performance of CloudFront service, we report in Fig. 10 the distribution of ΔR for different 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

¹⁰<http://aws.amazon.com/message/67457/>

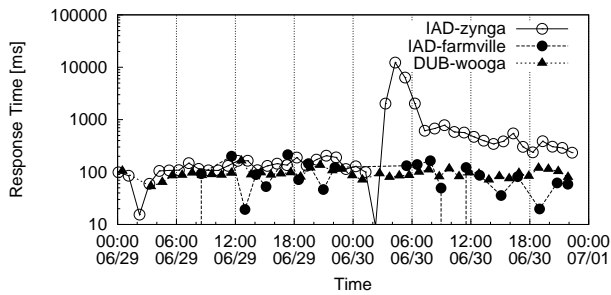


Figure 9. Evolution over time of ΔR during the IAD outage.

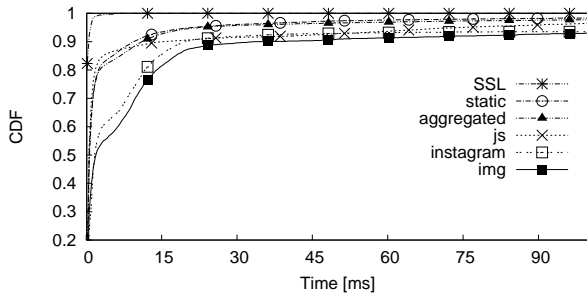


Figure 10. Distribution of response time ΔR for different kinds of contents provided by CloudFront cache located in MXP.

process 50% of requests in less than 2ms, independently on the kind of content. However, ΔR is consistently better on average for static and JavaScript files, 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 catalog, and of the small popularity of each single content [16]. Finally, SSL flows show excellent ΔR , suggesting that the cache, the path and the peering points are not congested.

IX. CONCLUSIONS

We are the first, to best of our knowledge, to study the traffic of *Amazon Web Services* (AWS) on the wild Internet. Our analysis assessed the growth of interest for cloud-based platforms, which, thanks to virtualization technology, represents a scalable and inexpensive solution for many web companies.

Through the study of traces captured from live networks, we confirmed that AWS represents a big player in nowadays Internet, being responsible for the generation of a sizeable portion of traffic.

We presented an extensive characterization of AWS offerings, in particular for EC2, S3 and Amazon's CDN CloudFront. Results presented in this paper 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 and it handles seven times the traffic served by the datacenter in Ireland. We observed that companies which rely on EC2 and S3 tend to concentrate their content on one datacenter, with the drawbacks of i) increasing the cost sustained by the network to carry data to faraway end-users and, ii) increasing risk in case

of failures. We evaluated the performance of contents hosted by EC2 and S3, in terms of servers' reactivity and network goodput, providing comparison among datacenters and among different contents. Our results show that the datacenter in Virginia shows in general poorer performance, but we could not understand if it is due to an actual overloading caused by the large population of EC2 instances, or to congestion of under-dimensioned instances offering particular content.

We also found that CloudFront performs much better, being able to serve 98% of traffic to the best possible cache. However, it presents issues that are typical of CDN systems: i) generic DNS servers returning caches far from end-users, that lower perceived QoS and system's efficiency; 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] A. Finamore, V. Gehlen, M. Mellia, M. Munafò, and S. Nicolini, "The Need for an Intelligent Measurement Plane: The Example of Time-Variant CDN Policies," in *IEEE Networks*, October 2012, pp. –. [Online]. Available: <http://www.retlc.polito.it/finamore/papers/finamore-networks12.ieeexplore.pdf>
- [3] E. Walker, "Benchmarking Amazon EC2 for High-Performance Scientific Computing," *USENIX ;login: Magazine*, October 2008.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] S. Triukose, Z. Wen, and M. Rabinovich, "Measuring a Commercial Content Delivery Network," in *WWW*. ACM, March 2011, pp. 467–476.
- [8] M. Gjoka, M. Sirivianos, A. Markopoulou, and X. Yang, "Poking Facebook: Characterization of OSN Applications," in *ACM WOSN*, Seattle, WA, 2008, pp. 31–36.
- [9] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *ACM IMC*, San Diego, CA, 2007, pp. 29–42.
- [10] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience," in *ACM IMC*, Berlin, DE, 2011, pp. 345–360.
- [11] 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.
- [12] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, Oct. 2003.
- [13] 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.
- [14] 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. [Online]. Available: <http://www.tlc-networks.polito.it/mellia/papers/IMC12-DNHunter.pdf>
- [15] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 53–56, Apr. 2011.
- [16] 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.