Illuminating Web Service Back-ends in Public Infrastructure-as-a-Service Clouds - Final Report

CREU Final Reports 2015

Student Researchers: Maimuna Lubega

Advisors: Aditya Akella, Associate Professor
Aaron Gember-Jacobson, PhD Candidate

Institution: University of Wisconsin-Madison

Webpage: http://sdn.cs.wisc.edu/creu/

Goals of the Project

Many popular web services (e.g., Dropbox, Pinterest, Reddit, and Netflix) are hosted in public infrastructure-as-a-service clouds such as Amazon EC2 and Windows Azure. However, we have only recently begun to understand how these services use the cloud infrastructure and whether they are taking advantage of the features offered by cloud computing - e.g., using multiple geographically distributed cloud data centers to improve fault tolerance and move services closer to users. The goal of this project was to further our understanding of cloud usage patterns by discovering how cloud infrastructure is used to host web-service backends. In particular, we focused on the problem of counting the number of virtual machines sitting behind a load balancer.

Process

We reviewed prior network measurement literature and Amazon EC2 documentation to identify possible techniques for counting the number of virtual machines (VMs) behind a load balancer. From our reading, we identified two techniques used to count hosts behind network address translators (NATs) that we thought might extend to load balancers. We also developed two heuristics based on the specifics of EC2’s design. We then evaluated the effectiveness of the four techniques by: (1) using controlled virtual machine and load balancer setups we created in EC2 and (2) sending probe traffic to actual services hosted in EC2.

The specific approaches we evaluated were:
1) IPid Sequences -- A prior measurement study of networks with NATs showed that each host behind a NAT assigns IPids sequentially based on its own internal counter. Thus, patterns in IPids from traffic emitted by the NAT can be used to estimate the number of hosts behind the NAT. Since load balancers multiplex traffic in a similar way to NATs, we hypothesized this technique could also be used to count the number of VMs behind a load balancer.

2) TCP Timestamps -- Hosts typically include a timestamp in TCP headers which is set based on their own internal clock. Since each host has minor deviations in hardware that cause a unique clock skew, prior work on counting hosts behind a NAT suggested that this skew could be used to discover individual hosts. Again, since the
multiplexing performed by load balancers is similar to NATs, we hypothesized we could use this technique estimate the number of VMs behind a load balancer.

3) Identifiers Embedded in Web Pages -- Some web services hosted in EC2 embed VM instance identifiers in the web pages they serve to clients. For example, the bottom of the Netflix landing page includes the EC2 instance identifier for the VM that served the page. Similarly, hovering over the pi symbol at the bottom right of a Reddit page reveals an instance name and process ID. By fetching a web page multiple times and counting the number of unique identifiers, we can get a lower bound on the number of VMs behind a load balancer.

4) Cookies -- When EC2’s elastic load balancer is configured to use sticky sessions (i.e., serve all requests from the same client from the same VM), the load balancer includes a special cookie in HTTP replies. This cookie is included in future HTTP requests to inform the load balancer which VM should handle the request. By examining the contents of the cookie, we can get the VM identifiers and count the number of unique identifiers to estimate the number of VMs behind a load balancer.

Conclusions and Results

Our measurements with controlled setups and actual web services revealed the following:

1) In our controlled experiments, the IPid sequences of traffic were randomly distributed. Further research revealed that Linux kernel versions 2.4 and later randomly generate a starting IPid for each new connection, whereas earlier versions use the same IPid counter for all connections. Hence, looking for IPid sequences is not a viable approach for modern web service deployments.

2) In the same controlled setup, there were also no differences in skew between TCP timestamps. This stems from a fundamental difference between NATs and the default load balancer configuration in EC2: NATs simply forward TCP packets to/from hosts, while EC2 load balancers terminate TCP connections from clients and establish separate TCP connections with servers. Hence, all timestamps on packets received by the client come from the load balancer, not from the backend VMs. Using TCP timestamps is only viable for a very specific load balancer configuration, which we do not expect to be a common choice.

3) After issuing requests to over 4,000 different web sites that are on Alexa’s top 1 million list and use load balancers in EC2, we found that only 10% of the sites had different content in the page each time. Furthermore, only in a handful of cases did the difference include some VM identifier, such as that included on Netflix pages. Thus, this technique only works in some cases.

4) With the same probing, we found that less than 4% of websites included a sticky session cookie in their reply. Moreover, the contents of the cookies were encoded and therefore we could not extract VM identifiers.

In summary, counting the number of VMs behind a load balancer is quite challenging and we could only obtain estimates for a small number of web services. As future work, we plan to explore whether the load balancers in other clouds (e.g., Azure, Rackspace, and Google) are more amenable to the aforementioned probing techniques.

Presentations and Publications

We submitted a poster abstract to Grace Hopper 2015.