JMeterAndAmazon

JMeter and Amazon

Elastic Load Balancer (ELB) Issues

- The ELB is a name, not IP, and can suffer from DNS caching. Make sure you use "-Dsun.net.inetaddr.ttl=0" when starting JMeter
- For a description of how the ELB works, see http://www.shlomoswidler.com/2009/07/elastic-in-elastic-load-balancing-elb.html, but if the link is down or you just need a high level overview:
    - Because the ELB is a DNS name, Amazon can (and is) load balancing the load balancers. Example DNS lookup: www.mydomain.com -> loadbalancer123.amazon.com (this is controlled by you, and can be a long TTL), loadbalancer123.amazon.com -> 1.2.3.4 (this is controlled by amazon, and is a short-lived TTL, currently 60 seconds)
    - Thus, each ELB is backed by a pool of load balancer IPs (which amazon can scale up or down based on load)
    - The ELB can be associated with one to many availability zones, but each load balancer IP is only associated with a single zone
    - Each load balancer IP evenly distributes load among instances in its availability zone
    - Thus for normal web traffic, load will be distributed fairly evenly. But, if the traffic originates from a small number of clients (like during load testing), you can easily get unbalanced loads on a per availability zone basis. There are two solutions: make sure there are enough instances to handle 100% of the load in each availability zone, or only use one availability zone.
- The motivation for this page was I thought I had bad load balancer behavior given this scenario:
    - I had two availability zones (for redundancy) with auto-scaling for 1 -> N in each zone.
    - I started a test that generated a small amount of load forever
    - I checked all backend instances, and all the load was on one box
    - On the JMeter box, I ran "dig mydomain.com" and watched the TTL count down from 60 to 0
    - When the ELB IP changed, all load moved to a different backend instance (and if the ELB IP stayed the same, the load stayed in the same place)
- But, if I changed the setup to have one availability zone with auto-scaling for 2 -> N, then each instance had ~50% of the load.