AUTOSCALING WEBSERVICES ON AMAZON EC2

M. A. AZEEZ

This dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfillment of the requirements for the Degree of M Sc in Computer Science specializing in Software Architecture

Department of Computer Science & Engineering
University of Moratuwa,
Sri Lanka
February 2010

ABSTRACT

Fault tolerance, high availability, & scalability are essential prerequisites for any Enterprise application deployment. One of the major concerns of enterprise Application architects is avoiding single points of failure. There is a high cost associated with achieving high availability & scalability. We will look at an economical approach towards automatically scaling Web service applications whilemaintainingtheavailability&scalabilityguaranteesatanoptimumeconomicalcostThisapproach,involving the Amazon EC2 cloud computing infrastructure, makes it unnecessary to invest in safety- net capacity & unnecessary redundancy. The Web service application developer should only need to write the application once, and simply deploy it on the cloud. The scalability & availability guarantees should be provided automatically by the underlying infrastructure. Auto scaling refers to the behavior where the system scales up when the load increases & scales down when the load decreases. Auto-healing refers to an approach where a specified minimum deployment configuration is maintained even in the event of failures. Such an approach is essential for cloud deployments such as Amazon EC2 where the charge is based on the actual computing power consumed. Ideally, from the clients' point of view, in an auto scaling system, the response time should be constant and the overall throughput of the system should increase. We will describe in detail an economical approach towards building auto-scaling Apache Axis2 Web services on Amazon EC2. In the course of this article, we will introduce well-known address (WKA) based membership discovery for clustering deployments where multicast-based membership discovery is an impossibility. We will also introduce an approach towards dynamic load balancing, where the load balancer itself uses group communication & group membership mechanisms to discover the domains across which the load is distributed. In a traditional setup, a single load balancer fronts a group of application nodes. In such a scenario, the load balancer can be a single point of failure. Traditionally, techniques such as Linux HA have been used to overcome this. However, such traditional schemes have quite a bit of overhead and also require the backup system to be in close proximity to the primary system. In case of catastrophic situations, this approach can result in complete failure of the system. We will introduce an auto healing scheme in case of load balancer failure using Amazon Elastic fP addresses & a load balancer group, which can overcome these shortcomings.