

Programming Assignment 7
Scaling - Performance
{Either MS Azure or AWS}
Due: In Blackboard

Description:

Frequently cloud service based systems need to support thousands or even millions of users (or subscribers) often simultaneously.

To support so many users the cloud services need to automatically scale. Scaling should be done both by scaling up to handle ever increasing demand, and then scaling back down to avoid being charged for unused and unneeded service.

Using assignment 2, 3, or 6 (or any other assignment you wish) as a template, or "driver", you need to create a cloud based system that responds to user queries and that (on the "cloud") performs simple database queries and formats a web interface response to the user. You will test (exercise) that system remotely, from your PC, and gather performance metrics.

You will then slowly increase the demand on that cloud based service, ("Jmeter" is one method or through other drivers creating similar demand.)

1. Create/implement a cloud based web site similar to assignment 2 or 3 or 6, or any that responds to user queries, retrieving information from a relational database.
2. Test your implementation.
3. Obtain Jmeter (or similar), install, test your understanding.
4. Create ever increasing load (demand) on your service.
5. Using AWS or MS Azure facilities configure scaling to handle scaling up and down, by creating additional instances or removing instances, up to 4 instances in total.
6. You may wish to use Jmeter to drive your cloud application. Try several parameters, such as amount of requests per unit time, intervals between requests, and similar variations on request traffic.
7. Try different thresholds on cloud scaling parameters.
8. Get, and interpret, results.

Please, submit in Blackboard. Work must be individualized, but may be done in a group.

You must submit this lab, working (or partially) by the due date.

Your program should be well commented and documented, make sure the first few lines of your program contain your name, this course number, and the lab name and number.

Your comments should reflect your design and issues in your implementation.

Your design and implementation should address error conditions.