Acme Dancer – Item 4

PERFORMANCE REPORT

Gonzalo Delgado Chaves

Jesús Javier Escobar Aguilera

Juan Carlos López Muñoz

Laura Padial González

Adrián Sánchez Reyes

Juan José Valle Zarza

Table of contents

1. Introduction
2. Setup
3. Performance analysis
4. Bottleneck diagnosis

1. Introduction

In this report, we offer an analysis of the causes and effects of running a performance test suit on the deployed web information system Acme-Dancer, using the application JMeter to organize regulated tests on the system’s cases of use.

First, we will analyze the results obtained on the tests, determining possible causes of error and the viability of the system based on error margins. In second place, we will determine possible bottleneck components on the user’s computer based on the performance of the system under stress of many concurrent petitions.

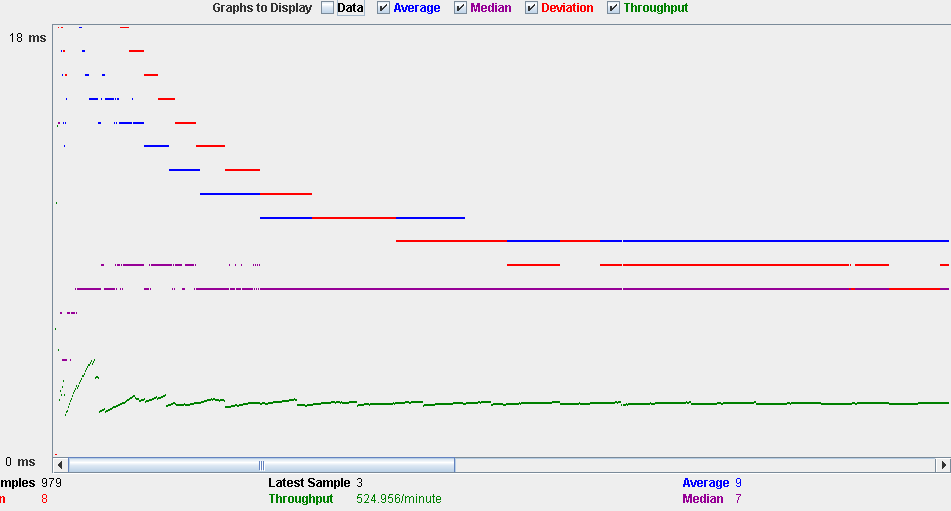
2. Setup

The web information system has been deployed in a pre-production environment, prepared to deploy and execute the system from the server’s administration console. A WAR artifact has been used for the system, and a MySQL script to initialize the necessary data on the database (in this case, we employ MySQL Server 5.5).

Once deployed, we utilize JMeter 2.9 in its last revision as tool to design and carry out performance testing, and the Performance Monitor tool included on Windows XP (accessible as perfmon.exe) to analyze the impact on the running computer.

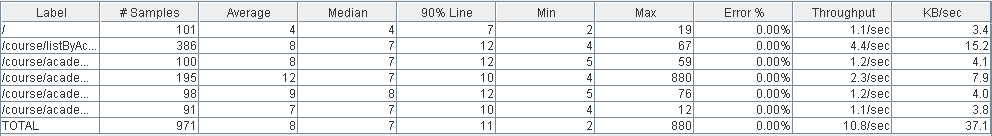
3. Performance analysis

On simple tests, such as profile registering and displaying simple lists of items, the system performs at averages between **10** and **12** milliseconds per instance, and average throughputs of between **400.000** and **540.000** operations per minute, however going as high as 900.000 in the most simple tests which include no more than one interaction between user and system. Overall, a very optimal speed and configuration.



HTTP errors display a stable **0%**. The average 90% line rests at between **17** and **24** milliseconds. These numbers are quite optimal, and assure that the system performs simple operations under stress without high possibility of error or inappropriate consequences.

For complex tests with multiple operations and checks to perform, such a managing entities (listing, creating, editing and deleting them), the system performs at averages between **10** and **15** milliseconds per instance, and average throughputs of between **600.000** and **650.000** operations per minute. HTTP errors lie at a stable **0%.** The average 90% line rests at between **58** and **63** milliseconds.



The narrow margins could represent the similarity between the most complex use cases, whereas the numbers don’t indicate there’s much difference with smaller tests. It is to be noted that the 90% line returns higher values, since these tests include more operations, and throughput doesn’t reach any significant spikes.

In general, the system performs admirably under all circumstances, and does not drop in performance under stress even in the most complex use cases.

4. Bottleneck diagnosis

The specifications of the system in which the tests were performed are as follows:

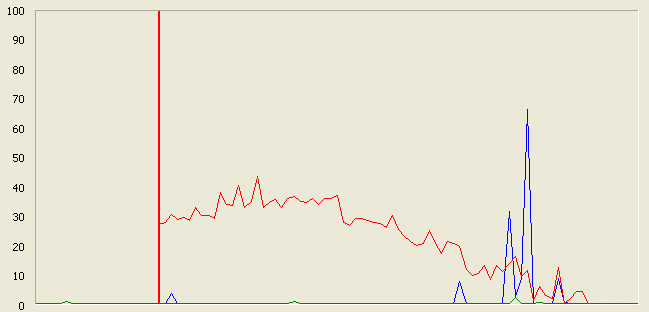
Processor: Intel i7-6700k at 4.00 GHz.

Disk: SATA hard drive disk at 6700 RPM.

Memory: 16 GB DD4 RAM at 3000 MHz.

Network Interface: 1 GBit Ethernet interface.

Given the specs, the system performs well under stress, and only reaches highs in very specific occasions. During the most complex performance tests, processor activity and memory activity ramp up, with memory in particular reaching spikes when the system is testing access to multiple pages.



As processor activity remains under 50%, our diagnosis is that the computer’s memory is the most susceptible component to failure, and that in heavier stress tests it will cause a bottleneck in the system.