VideoConversionSim.py

Simulation of System That Converts Uploaded Video Files

Göran Gustafsson
Computer Science Student
University West
Trollhättan, Sweden
gogu0001@student.hv.se

***Abstract*— Estimating the amount of resources a system needs to be sustainable under certain scenarios is an important but hard question to answer. In this paper we use a custom built simulation program using Python and SimPy to find out how a video conversion system could act under a specific scenario while using different amounts of resources. We use simulation to find out what is needed for the system to work sufficiently well, how many resources are needed to avoid congestion in our system. Two simulations are performed 1000 times each and the results are compared to illustrate how well our video conversion system works with X amounts of resources and how badly it performs with Y amounts of resources.**

Keywords*— *Simulation; Video conversion; SimPy

I. Introduction

Simulation can be a valuable tool for estimating the unknown. It can amongst other things be used to get a clear picture of how something works before it even exists, and it can also be used to predict the amount of resources that is needed for a system to be sustainable under certain scenarios [1]. This is how simulation has been utilised during our project, simulation of something unknown and estimation of the amount of resources needed for that particular system.

II. Problem Formulation

Estimating the amount of resources needed for a sustainable system can be a difficult task because there might be a lot of different variables that affect how well a system works. This is especially the case when there are several random events, for example users interacting with a system at random times. Assumptions of a system must be correct for a simulation to give valuable results [2]. If the usage of a system is hard to predict it will also be hard to estimate the correct amount of resources needed (hardware, network capacity etc) by yourself and therefor it makes sense to try and simulate the system usage before investing money into a real system.

The system that we have decided to simulate is a uploaded video conversion system a la YouTube. Basically a system which receives video files of random length at somewhat random intervals and each video file is converted to other formats under different time durations. The system can convert a limited amount of video files simultaneously. This is a simulation with the purpose of finding out the potential congestion that might arise (queuing of conversion jobs) when using different amount of resources, if system X can handle scenario Y.

III. Purpose and Question

This paper covers the scenario and results of two different simulations that has been performed. Both simulations has one common scenario (the same assumptions of the system/real-world) but with two different amounts of resources so a comparison between the two can be performed.

The purpose and our question can be summarised the following way:

- Simulate a video conversion system to see how it could act in real life.
- Find out how well the simulated system works with different amounts of resources.

The scenario used was chosen because we feel that it might be close to that of a real-world system and that the system itself makes sense to simulate. The end results are however only relevant to the specific example scenario used.

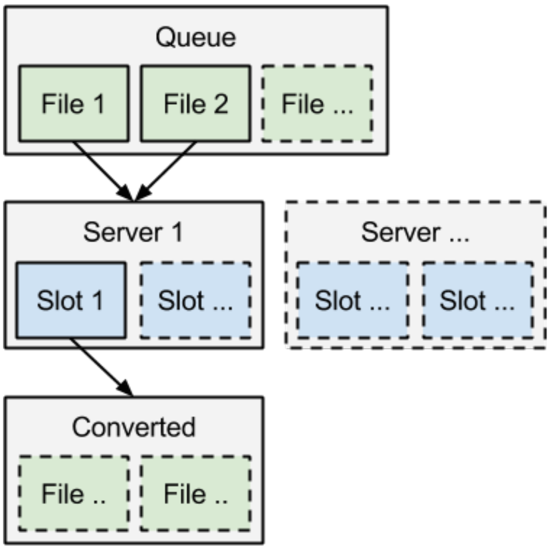
IV. Method

Here we will go into detail about the simulation model and scenario (the system that has been simulated), the simulation program (overview of what it does) and how we collected data from our simulation runs.

V. Simulation Model and Scenario

The first step in simulation is creating a simplified model of the system that is to be simulated. Take all of the important aspects of the original system, simplify them without losing sight on what is important so you end up with a system model that is easier to study than the real thing. [2]

The video conversion system we set out to simulate can be simplified to great extent. The assumptions drawn are that the system has X amount of servers, each server can run Y amount of conversion jobs simultaneously without sacrificing speed and all incoming video files will vary in size/quality which directly affect the conversion duration time. To simplify it even further we can say that we have X slots and each incoming file will use an empty slot for Y seconds. A visual model of the system can be seen in Fig. 1.



1. Visual model of system

The scenario of the simulations we have performed are best described by looking at the various settings used in the program and our motivation for each value used.

- Video files total, 1440. This value is used because this value in combination with the upload interval setting makes the simulation run for approximately 24 hours.
- Approximate upload interval, 60 seconds. We want a constant high rate of video uploads. Exponential distribution is used to achieve a more realistic behaviour.
- Video length, 30 sec - 30 min. We expect videos to always be between 30 seconds and 30 minutes long. Length is completely random within this range.
- Conversion time, 50% of video length. Conversion time depends on a lot of different factors and therefor testing needs to be performed on the real server hardware to get correct values. It is however somewhat connected to the length of video files and we decided on using 50% for each file.
- Maximum acceptable waiting time in queue, 5 minutes. Each video conversion needs to start within an acceptable time range. Every time we get past this limit it means we got congestion in our system.

Two simulations of the same scenario but with different amounts of resources will be performed. The first simulation (here on referred to as simulation 1) will use 2 servers that has the capability of performing 4 conversion jobs each, a total of 8 simultaneous conversions. The second simulation (referred to as simulation 2) will use 3 servers that also have the capability of performing 4 conversion jobs each, a total of 12 simultaneous conversions.

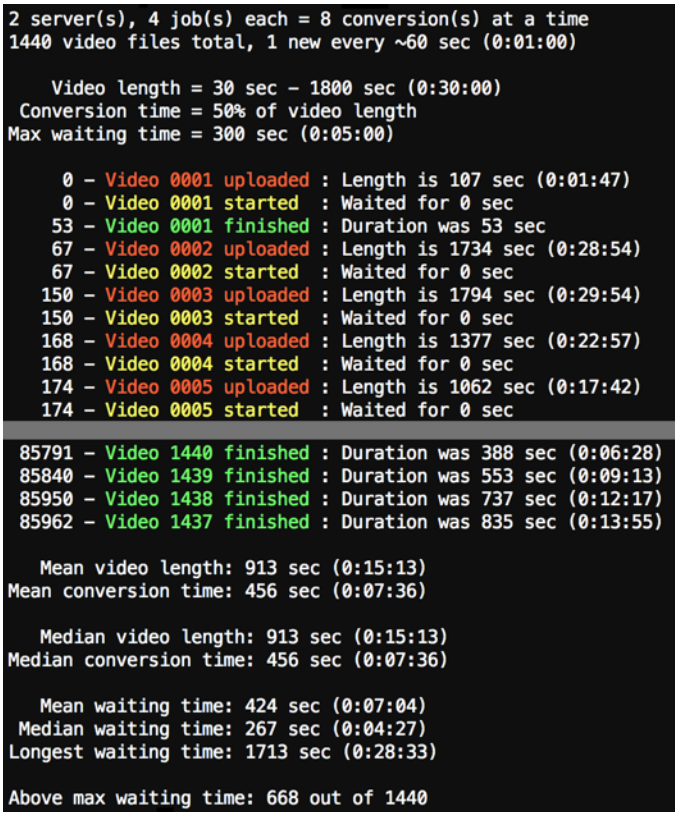
VI. Simulation Program

The simulations are performed with a program that was custom built for our specific system model and simulation scenario. It is built using the Python language (version 3.4.2) and the SimPy simulation framework (version 3.0.5).

The workflow of the simulation program in a simplified form is as follows:

1. Video file is uploaded.
2. Video file enters conversion queue. If it is empty then continue, otherwise wait until it is.
3. Perform conversion of video file. Wait X seconds.
4. Wait approximately Y seconds. Used for uploading the next video at a somewhat random time.
5. Check if there are more video uploads left. If yes then go back to step 1, otherwise continue.
6. End simulation program.

A screenshot of the video conversion simulation program can be seen in Fig. 2.



2. Screenshot of the simulation program

VII. Data Collection

The data used in this paper is collected through a rather simple helper script that executes the video conversion simulation program 1000 times and after each of those simulation runs has been performed the values of mean waiting time, longest waiting time and above max waiting time are written down as a new line in a comma separated values file (CSV file). We can then manually analyse the data with a spreadsheet program, and also automatically create graphs by parsing the data files with a custom built script and visualise that information with the Gnuplot program.

The simulation program is executed 1000 times for each of the two simulations (simulation 1 and simulation 2) to get more reliable results. The outcome of one simulation can vary greatly between two runs because there are so many random variables used in our system, conversion time is the biggest factor here. We end up with 2000 lines of results, three different values on each line.

VIII. Results

Here we will go into detail about the results from simulation 1 and simulation 2. The results are divided up under three sections, one per simulation and one for the results that are related to both simulations.

For each value that was collected (mean waiting time, longest waiting time and above max waiting time) we looked at the highest values we could find and then divided that numerical range (from 0 to highest value) into 10 equally large groups. Afterwards we counted how many values are within the different range groups.

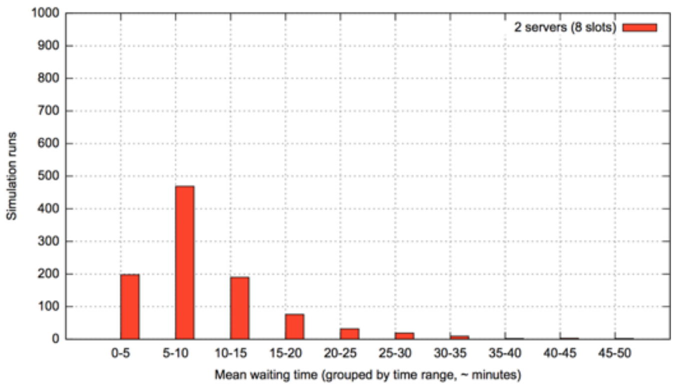
Note that the results of above max waiting time is not shown until section *XI. Both Simulations* because they will always be the same and there is no point in showing the same results twice.

IX. Simulation 1

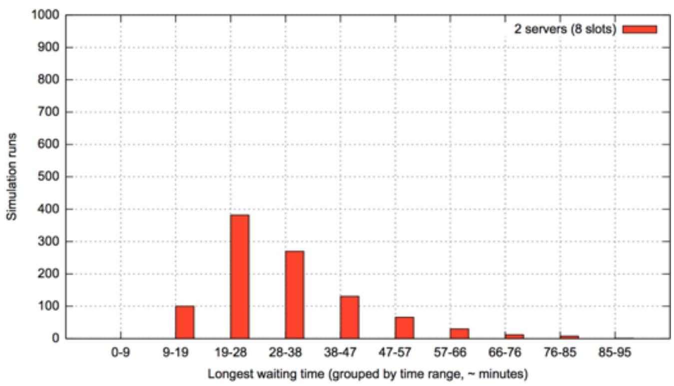
These are the results of the first simulation. The system that used 2 servers that has the capability of performing 4 conversion jobs each, a total of 8 simultaneous conversions.

The most frequent mean waiting time range was ~5-10 minutes, 469 out of 1000 simulation runs. See Fig. 3.

The most frequent longest waiting time range was ~19-28 minutes, 382 out of 1000 simulation runs. See Fig. 4.



3. Mean waiting time histogram of results from simulation 1



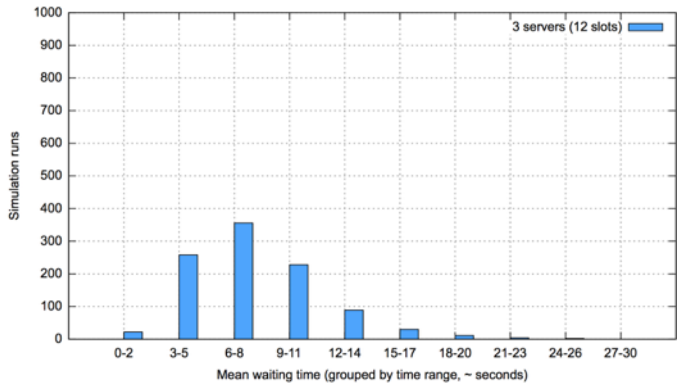
4. Longest waiting time histogram of results from simulation 1

X. Simulation 2

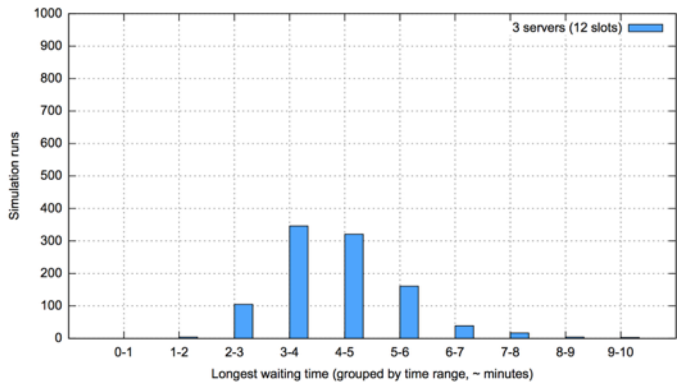
These are the results of the second simulation. The system that used 3 servers that has the capability of performing 4 conversion jobs each, a total of 12 simultaneous conversions.

The most frequent mean waiting time range was ~6-8 seconds, 356 out of 1000 simulation runs. See Fig. 5.

The most frequent longest waiting time range was ~3-4 minutes, 346 out of 1000 simulation runs. See Fig. 6.



5. Mean waiting time histogram of results from simulation 2



6. Longest waiting time histogram of results from simulation 2

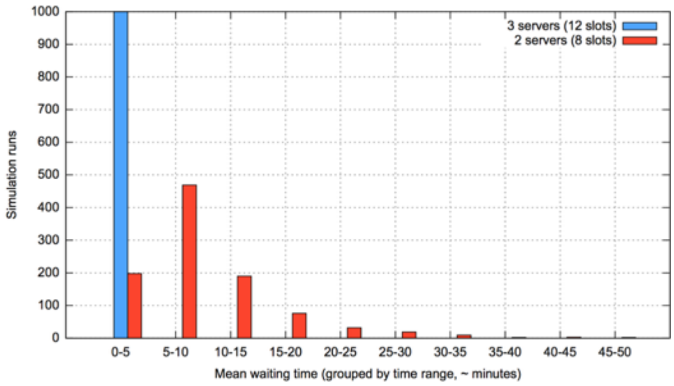
XI. Both Simulations

These are the results of both simulations. We combine the results to see what effect the number of different amount of resources has.

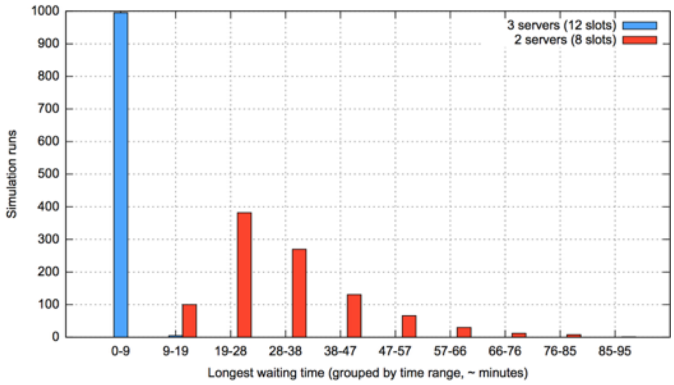
The most frequent mean waiting time range was ~0-5 minutes, 1000 out of 2000 simulation runs. The second most frequent range was ~5-10 minutes, 469 out of 2000 simulation runs. See Fig. 7.

The most frequent longest waiting time range was ~0-9 minutes, 995 out of 2000 simulation runs. The second most frequent range was ~19-28 minutes, 382 out of 2000 simulation runs. See Fig. 8.

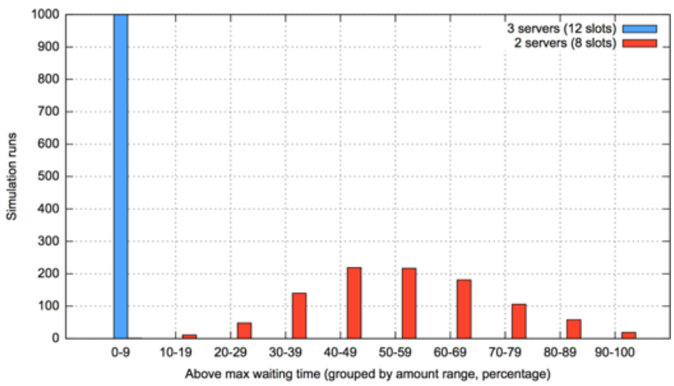
The most frequent above max waiting time range was 0-9 percentage, 1000 out of 2000 simulation runs. The second most frequent range was 40-49 percentage, 219 out of 2000 simulation runs. See Fig. 9.



7. Mean waiting time histogram of results from simulation 1 and 2



8. Longest waiting time histogram of results from simulation 1 and 2



9. Above max waiting time histogram of results from simulation 1 and 2

XII. Discussion

The simulation program made it possible to get a clear picture of how the video conversion system could act under certain scenarios. It made it possible to experiment with the different variables that defines the system and scenario to see what effect they have on the system while it is in use. We could see how our system can handle different loads, we could see what happens with different amount of resources and we could see how the system works under different time lengths amongst other things. These simulations made it possible to do what would otherwise be too hard to estimate.

The biggest drawback to our simulations is that the results returned are of the same quality as the initial assumptions made of the real-world system and usage load. If the variables used under a simulation are set to values that does not correctly reflect reality then the simulation results have zero value. Wrong assumptions taken before simulations could potentially even lead to bad investments afterwards, for example under-commitment or over-commitment when investing in new hardware.

XIII. Conclusions

How our video conversion system would act under our scenario was unknown, we simply could not predict (in a good way) what would happen without using simulation. What we did know however was that it would work sufficiently well with X amounts of resources and badly with Y amounts of resources, we set out to find where that limit was for our scenario. The results from simulation 1 and simulation 2 perfectly illustrates what works well and what works badly, see Table 1.

I. Summarisation of results from simulation 1 and 2

	Simulation 1	Simulation 2
Mean waiting time	~5-10 minutes	~6-8 seconds
Longest waiting time	~19-28 minutes	~3-4 minutes
Above max waiting time	40-49%	0-9%

Note that the values seen in Table 1 is the most frequent results of 2000 simulations. Simulation 1 used 2 servers (8 simultaneous conversion jobs) and simulation 2 used 3 servers (12 simultaneous conversion jobs).

The results from simulation 1 show signs of severe congestion in the system while simulation 2 show no signs of congestion at all. 3 servers can easily handle our specific scenario.

References

1. John S. Carson, “Introduction to modeling and simulation”, WSC '04 Proceedings of the 36th conference on Winter simulation, pp. 9-16, December 2004.
2. Unknown, “JavaSim - User’s Guide. Public Release 0.3”, http://javasim.codehaus.org, accessed 2014-12-19.