Report for IN4392 Large Lab

Authors: Lipu Fei and Hans van den Bogert
Computer Science, EEMCS, Delft University of Technology
Emails: {l.fei, j.c.vandenbogert}@student.tudelft.nl
Course Instrutors: Alexandru Iosup and Dick Epema
Parallel and Distributed Systems Group, EEMCS, Delft University of Technology
Emails: {A.Iosup, D.H.J.Epema}@tudelft.nl

***Abstract*—In this report, we introduce a cloud management platform that provides an online video-transcoding application for users. The platform, which we call "WantCloudFrame", utilizes elastic cloud resource provisioning and allocation to make it scalable in performance.**

I. Introduction

Current situation: For WantCloud providing transcoding[1] facilities have been a great source of income. Due to the popularity of the current system, it is overloaded because it does not scale well during peak usage. In the existing solution there is only one machine which handles incoming jobs. The overloaded system therefor has a relatively high number of outstanding jobs causing it to not meet the deadlines which are guaranteed by WantCloud's Terms-of-Use. To circumvent the shortcomings of using only 1 physical machine we will look into using a IaaS as our platform.

Related Work: The pre-existing cloud environment used for the proposed system is the [**?**] cluster with the [**?**]-stack on top of DAS4. OpenNebula provides a low-level interface for spawning *Virtual Machines* [VM] on which our workload can then be placed.

For the actual conversion of the media files the [**?**] program is being used. For the sake of implementation feasability only the conversion from [**?**] to DVD-Video is considered. This software is freely available for everyone to use under the GPL-license.

As a method for inter-machine communication in the cluster, SSH is being used. SSH is available in every spawned machine by default and provides us the means for secure communication and file-transfer.

Proposed solution: To be able to cope with demand, a new system setup has been made which can tap into a pre-existing cloud environment, to scale during peak usage and thereby load-balance the workload over multiple machines. For the experiment we've looked at multiple methods for allocation of the machine resources. By keeping statistics in our implementation we track the total time it takes for a submitted media-file to be transcoded and sent back to the submitter. This metric will hereafter be called the *makespan* of a submitted job. Another metric is the cost for a job. Leasing a VM in the cloud costs money – we investigate the tradeoffs between leasing more VMs and the effect on the makespan.

@TODO more metrics

To experiment with the proposed solution - a benchmark has been created which uses a predefined sample from a exponentional distribution to simulate arrival times for jobs.

TABLE I
Time Sheet Table for the Whole Lab Exercise

Parts	Time
total-time	
think-time	
dev-time	
xp-time	
write-time	
wasted-time	

Overview: In the next section we will elobarate more on the application and provide more background information. In section III the system's design will be handled so the experiment in section IV can be understood. After the experiment we will discuss the findings in section IV and conclude in section VI.

II. Background on Application

For the experiment

III. System Design

This is system design section.

IV. Experimental Results

This is experimental results section.

V. Discussion

This is discussion section.

VI. Conclusion

This is conclusion section.

Appendix
Time Sheets

[1] Transcoding: The process of converting a media file from one format to another