Big Data Processing Architecture for Legacy Sports Video Information Extraction in the Cloud

Rafael Pereira
WebMedia
Globo.com
Rio de Janeiro, Brazil
rafael.pereira@corp.globo.com

Hélio Lopes, Karin Breitman
Dep. de Informática
PUC-Rio
Rio de Janeiro, Brazil
{lopes, karin}@inf.puc-rio.br

Abstract — The online availability of digital content is growing exponentially, creating a demand for more sophisticated search and retrieval mechanisms. In this scenario, the ability to semantically annotate content is extremely important, since it provides relevant metadata in order to feed search engines and improve query precision. Multimedia annotation, and of video in particular, is done semi-automatically at best. In this paper, we propose an OCR-based solution for automatic metadata extraction and further annotation of sports videos. The proposed solution is combined with the Split&Merge architecture to allow for deployment in the cloud, which dramatically reduces time and overall costs associated with the process.

Keywords: cloud computing, metadata extraction, video annotation, OCR.

I. Introduction

The increasing popularity of videos on the Internet, allied with the advances in network technology seen in the last decade, is drastically changing TV as we know it. In this light, the ability to automatically extract information about events in sports videos, e.g. the moment when one team scores a goal, faults, player exchange, as well as additional information such as team, player, and stadium names, is extremely interesting in situations where no additional data is associated with the video. This information can be employed to make useful annotations that will help index, store, retrieve, and correlate video files from huge media archives, since now, with online and on-demand media, they are an important asset for monetization.

Usually this process of video data extraction is done manually, where each video is watched and annotated according to the data seen by a person. There are some initiatives to make this extraction task automatic or semi-automatic [1]. However, data extraction and annotation of a large amount of video make this process extremely slow and costly, which, in many cases, is not desirable. Therefore, developing algorithms that could reproduce the human capacity to extract and classify information is extremely important.

Sports videos are very good candidates for automatic information extraction because there are often several bits of important information on the video itself [2]. Figure 1 exemplifies a video frame that contains information that can be identified and extracted for content annotation, using, for example, an Optical Character Recognition (OCR) technique [3]. However, to obtain satisfactory results, it is necessary to apply, in each video frame, several image-processing techniques for character recognition, which makes this task a very complex one.

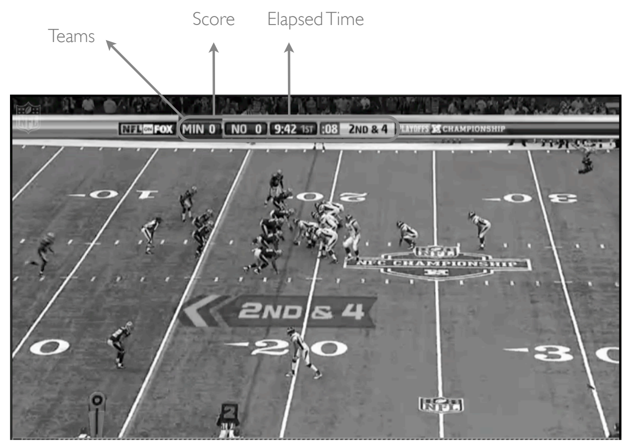

Figure 1. Information present inside a sport video.

We would like to call attention to the fact that video and image processing are computationally very expensive, i.e., they require a great amount of processing power and require large storage resources. Solutions to reduce processing times and improve the rational use of available resources are thus very important. In this scenario, the cloud computing paradigm [4,5,6,7] is emerging as a viable strategy for dynamic resource provisioning.

Contributions. Based on this scenario, we propose a solution that uses distributed video processing for an OCR-based automatic metadata extraction of large volumes of videos, which makes use of the flexibility provided by public cloud services, thus adjusting resource usage according to demand fluctuations. With this approach, detailed below, it is possible to extract events from sports videos automatically, in a timely and efficient process, just by using dynamic resource provisioning provided by cloud services. This represents a huge competitive advantage because, unlike the traditional process, it is possible to process within minutes (in a fixed number of minutes, for that matter) very large video archives that would typically take several hours or even days to process.

Paper Outline. In the next section we introduce some background information, discussing related works in video data extraction and cloud Computing. In Section III we introduce the Split&Merge (S&M) approach for high performance video processing, an architecture that explores on-demand cloud computing infrastructures to tackle the time efficiency issue. Sections IV and V detail the supporting tools used and the proposed architecture itself. Finally, in Sections VI and VII we present some results and conclusions, and discuss further work.

II. Related Works

Next we present some previous works in video data extraction and background information about cloud computing.

A. *Video Data Extraction*

It is acknowledged that video event detection and recognition have greatly evolved over the past years, with different approaches for robust detection and representation of spatio-temporal interest points and motion features, modeling of events and approaches to represent domain knowledge and contextual information of activities and actions.

There are several techniques for video information extraction: from established and developed approaches, such as OCR, to more modern solutions for video understanding that use object detection, facial recognition, motion modeling, etc., which usually require much more processing resources.

For example, [2] proposes an approach to automatically compose a highlight video by reference to a soccer match video that detects activities involved in a highlight by the local motion appearance. Other approaches such as object detection [8,9] and human poses and actions [10] are also successfully used to extract information from videos. A number of approaches that use Hidden Markov Model (HMM) have also been proposed to analyze sports videos, since the events that are typical for this domain are very well suited for this approach [1].

Several works in the sports video domain apply heuristics or rule-based approaches to automatically recognize simple events. An example is given by Xu et al. [11], in which recognition of play/break events of soccer videos is performed using classification of simple and mutually exclusive events (obtained by using a simple rule-based approach).

More complex events can be recognized using Finite State Machines (FSMs). This approach was initially proposed by Assfalg et al. in [12] to detect the principal soccer highlights, such as shot on goal, placed kick, forward launch, and turnover, from a few visual cues such as playground position, speed, and camera direction.

However, information gained through Video OCR is often unique and unobtainable from another video understanding technique [3], making accurate Video OCR a vital technology for searching video archives. Besides providing unique information about the content of video, accurate Video OCR poses challenging technical problems, as described by [3]. Low-resolution data and complex backgrounds are two problems that contribute to text extraction degradation.

In summary, since the aim of this paper is to extract information from large sports video archives to allow for the use of search and retrieve techniques in such datasets, the data extraction is performed using an OCR approach, which is usually faster. However, the extraction technique implemented by the proposed architecture is a hotspot and can be easily replaced by a different detection and extraction method.

B. *Cloud Computing*

Cloud computing is a paradigm shift following the move from mainframe to client–server in the early 1980s. Details are abstracted from the users, who no longer need to have expertise in, or control over, the technology infrastructure "in the cloud" that supports them [13,14]. Cloud computing describes a new supplement, consumption, and delivery model for IT services based on the Internet. It typically involves over-the-Internet provision of dynamically scalable and often virtualized resources [15,16].

In cases where there is a seasonal computation demand, the use of public clouds for information processing and storage is emerging as an interesting alternative. The Infrastructure as a Service (IaaS) [17,18] paradigm relieves the burden of making huge investments in infrastructure and, at the same time, supports on-the-fly resizing of resources and adaptation to current needs.

With a public cloud, one can quickly make provision for the resources required to perform a particular task and pay only for the computational resources effectively used. This is a good solution, not only because it deploys faster, as opposed to having to order and install physical hardware, but it also optimizes overall costs, as resources can be released immediately after the task is completed.

One of the largest IaaS providers in the public cloud is the Amazon Web Services platform (AWS), with its Elastic Cloud Computing (EC2) [19] and Simple Storage Service (S3) [20] services. Amazon EC2 is a web service interface that provides resizable computing capacity in the cloud, allowing a complete control of computing resources and reducing the time required to obtain and boot new server instances. This feature is of particular interest because it allows applications to quickly scale up and down their processing and storage resources as computing requirements change. Amazon S3 provides a simple web services interface that can be used to store and retrieve data on the web, and provides a scalable data storage infrastructure.

The AWS Platform also provides several additional services, as previously mentioned, which are not the focus of this research. One of them is the Elastic Map-Reduce, an implementation of Map-Reduce [21] algorithm built on top of the basic AWS infrastructure blocks (EC2 and S3). This feature is particularly interesting because EC2 and S3 alone are not sufficient to provide efficient and scalable high performance processing architecture. To achieve these goals, one has to build applications that are able to take advantage of the characteristics of IaaS infrastructures, for example, the ability to automatically start and stop machines according to processing demands, or the ability to use several machines to simultaneously process parts of content. An architecture that deals with these issues is the Split&Merge, detailed in the next section.

III. SPLIT&MERGE ARCHITECTURE FOR VIDEO PROCESSING

The Split&Merge architecture allows for the distributed and parallel processing of large volumes of data, at the same time that it provides scalability, fault tolerance and the flexibility to deploy in either private clusters or in public cloud platforms [22,23]. These requirements are increasingly desirable to ensure the flexibility and robustness needed to develop computer systems that deal with big data efficiently.

In this paper we are interested in extract events from legacy sports videos automatically, in a timely and efficient process. Once video is characterized for its high volume of data, it becomes interesting to use parallel and distributed processing paradigms, in which the main goal is to reduce the time required for large dataset processing. The fact that video processing applications are not fully addressed by existing techniques for high performance and parallel processing provides an additional motivation for the S&M architecture.

The Map-Reduce paradigm [21], for example, is a framework for processing huge datasets of certain kinds of distributable problems using a large number of computers (nodes), collectively referred to as a cluster. A popular Map-Reduce implementation is Apache's Hadoop [24], which consists of one Job Tracker, to which client applications submit Map-Reduce jobs. The Job Tracker pushes work out to available Task Tracker nodes in the cluster, which execute the map and reduce tasks.

If we analyze the Map-Reduce paradigm in its essence, we note that the key point for processing optimization is the way in which tasks are distributed and how they make use of available computing resources. It is precisely this characteristic, i.e., breaking down an input and processing its parts in parallel, which must be preserved to achieve significant reductions in the overall processing time.

The S&M architecture is similar to the Map-Reduce paradigm, yet more simple and general [22]. In a nutshell, it is a split, process, and merge architecture, in which a video input is fragmented, its fragments are processed in a distributed environment, and partial results are merged, as illustrated in Figure 2. Similarly to Map-Reduce, this architecture makes very efficient use of available computing resources and, furthermore, allows the use of more complex inputs and the choice among more specific video processing techniques, such as different OCR mechanisms.

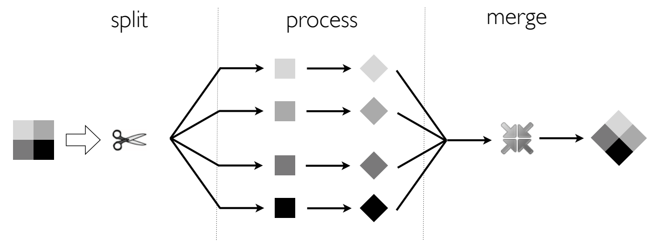

Figure 2. The split, process, and merge concept

This architecture differs from the most popular approach, where the entire video is processed on a single server. The traditional process proves to be effective only when there is a great deal of videos to be processed. However, if there is only one video, the total processing time cannot be optimized, nor are resources used rationally, as all the work is concentrated on a single machine. With the S&M architecture this does not happen. Resources are used rationally, independently of the size of the demand.

The S&M architecture was conceived in modules, to ensure flexibility, i.e., allow the algorithms used in the split, process, and merge steps to be replaced and/or customized as needed. In the case of video processing, for example, the type of entry, codecs in use, containers, and the amount of audio streams, among others, greatly impact on and may result in significant changes to the techniques used in each individual step. In such cases, the idea is to allow for the mix and match of different implementations, in a modular and componentized way [23].

IV. UNDERLYING IMPLEMENTATION TECHNIQUES AND USED TECHNOLOGIES

To implement this project we made use of different supporting technologies that had to be combined and extended to obtain the desired results. Because development and improvement of OCR algorithms is not the focus of this work, we used a combination of existing techniques for character recognition related tasks. We chose a well-known, free optical character recognition engine, Tesseract [25], which provides character accuracy greater than 97% [26], and is currently developed by Google. We combined it with ImageMagick [27], a tool that pre-processes video frames, cropping out irrelevant information and transforming video frames to monochrome with white background to increase the OCR efficiency.

In addition, to allow for parallel and distributed processing of video frames across multiple nodes, we used CloudCrowd [28], a Ruby framework that implements a scheduler that delegates tasks to different nodes and monitors their status. This tool was used to implement a queue controller responsible for scheduling tasks in our distributed environment.

In order to perform video stream manipulation and frame extraction and sampling, MEncoder [29] and FFMpeg [30] video processing tools were used, since they support the decoding of a wide range of different video formats.

Finally, to allow for the implementation of an elastic architecture, capable of scaling up according to fluctuations in the demand, as well as easy to deploy in a public cloud service, we used the AWS platform[19,20] as the cloud infrastructure provider. It is important to note that, in addition to providing processing resources on demand, the AWS platform also provided distributed and fault-tolerant storage, and relational database services.

V. An Architecture For OCR-Based Video Information Extraction

As described in Section III, there are several techniques whose goal is to perform parallel and distributed processing of large volumes of information, such as Map-Reduce. However, despite being considered a large dataset, the video presents some characteristics that hinder and even prohibit the use of key techniques of distributed processing, without an adjustment in how they deal with information. To address these situations the Split&Merge architecture was proposed.

In order to extract events from sports videos automatically and to allow for the scalability of this process, we propose an implementation of S&M using a public cloud, that provides an infrastructure to deal with video processing in a simple split-distribute-process-merge process. That is, for every video received, it is fragmented, its fragments are processed in a distributed environment using several server instances in the cloud, and, finally, the result of processing is merged. With this approach, it is possible to scale the computational resources up and down according to the video that is processed and the time required to obtain the desired information can be greatly reduced.

A. The Split Step

We begin by detailing the sampling technique used for reducing processing times, implemented using the Split&Merge architecture, as illustrated in Figure 3.

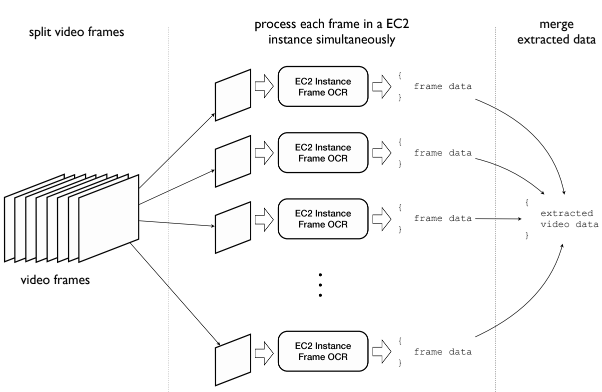

Figure 3. The Split&Merge deployed in the cloud for OCR data extraction.

The fragmentation of media files and the distribution of processing tasks in a cluster consist of an advanced solution for increasing performance and an evolution of the simple distribution of single complete video processing tasks in a cluster or cloud. The idea is to break the media files into smaller files so that their multiple parts can be processed simultaneously on different machines, thereby reducing the total time required to obtain the desired output.

In order to eliminate redundant information and significantly reduce the amount of information to be processed, the split step consists of breaking the video into its frames and performing a frame sampling. Sampling is only possible in situations where the difference in information between subsequent frames is minimal. In the case of sports videos, at 30 frames per second, we can extract only one frame per second, or even less, which proved enough to identify the desired information.

B. The Process Step

Once the video is fragmented, the extracted frames should be distributed among the nodes to be processed. In this step, each frame is individually processed in a node (e.g. Amazon Elastic Compute Cloud—EC2—instance in the cloud), as proposed by the S&M architecture. This distribution can be implemented through some simple HTTP REST [31] calls, where a master node, responsible for the split step, starts one server instance in the cloud for each frame obtained in the first step and delegates the frame processing to this instance, as Algorithm 1 describes.

ALGORITHM 1: Running at Master Node

*Remove a sampled **frame** from queue*
*Unless there is an idle server **instance***
    *Start a server **instance***
End
*When **instance** is available*
    *Send the **frame** processing job to **instance***
End

After the distribution, each frame is individually processed, which means that a sequence of image-processing algorithms is used in order to retrieve the existing information inside the frame. For example, as shown in Figure 4, for soccer videos the processing step searches and extracts the score inside the frame, then prepares the extracted score to be recognized, applying some filters such as monochrome transformation and noise reduction, and finally extracts the information using the OCR.

As a result of the processing step, for each sampled frame there is an object—as presented in Figure 5—that describes the information extracted from that frame. Typical information includes identification of the teams, total scored goals (for each team), and elapsed time in minutes and seconds. Figure 4 illustrates one such frame.

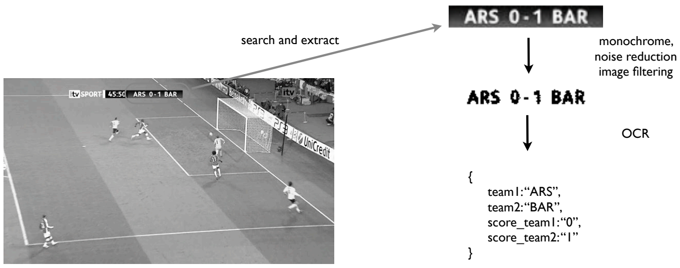

Figure 4. The video frame event extraction process.

Occasionally the OCR engine will not identify some of this information correctly. In such cases the output is discarded in the merge step.

```
{
        team1: "ARS",
        team2: "BAR",
        score_team1: "0",
        score_team2: "1",
        min: "45",
        sec: "32"
}
```

Figure 5. The extracted information of a single frame in a structured object.

C. *The Merge Step*

After processing each frame, which is performed in a parallel and distributed architecture using the Split&Merge, the results obtained are combined. First a frame reordering is executed to restore the logical event sequence, then a list containing all information obtained is created, generating a structured dataset. From this structure, we analyze when there was a change in score, indicating a goal, for example, and generate an output object that has attributes like the teams identified, the final score of the match, and a list of objects that describe the goals, as shown in Figure 6.

```
{
        team1: "ARS",
        team2: "BAR",
        final_score_team1: "0",
        final_score_team2: "1",
        scores: [
              {team: "BAR", min: "45", sec: "23"}
              ]
}
```

Figure 6. The structured object output from video event extraction

The merge step also discards inconsistent information that may be generated during the processing step. Basically, it compares subsequent frame results to evaluate if the returned information is valid or not. For example, for each frame it analyzes which are the most probable teams of previous and next frames to perform a correction in the frame that is being analyzed. This backward and forward analysis increases the efficiency in information identification and extraction. Algorithm 2, below, shows an example of how the merge step could be implemented in order to obtain the structure described in Figure 6.

ALGORITHM 2: Running at Master Node

Order the received frame processed information
*Get the most probable value for **Team 1** in all frames*
*Get the most probable value for **Team 2** in all frames*
*For each **frame***
    *Take previous 10 frames as **P[10]***
    *Take next 10 frames as **N[10]***
    *Obtain the most probable score for **P[10]** as **MPSP***
    *Obtain the most probable score for **N[10]** as **MPSN***
    *if **MPSP** == **MPSN** and **MPSP** != **frame** score*
        Discard frame result
    *if **MPSP** != **MPSN** and **MPSN** == **frame** score*
        Save a goal score for frame time
End

VI. RESULTS

Initially, five different sequences of high-definition 720p soccer videos, with different content and duration were selected and encoded with MJPEG 30Mbps, 29.97fps, and audio PCM/16 Stereo 48kHz. Our initial goal was to identify which were the teams involved in the match, the final score of the game and the moment the goals, if any, were scored. We also wanted to discover which the sample frequency was that offered the best cost-benefit in terms of precision and time required for information identification.
To make the testbed more realistic, the chosen samples were from different matches and were a mixture of short (2 minutes or less) and long (10 minutes or more) duration videos. In addition, we chose samples where the score did not appear throughout the video duration, but rather was displayed intermittently. The OCR engine used was Tesseract 3, without any training, but with a specific team name acronym dictionary.

To evaluate the impact of sampling on processing times and accuracy, each video was processed several times, with different sampling rates: in the first round of tests we sampled one frame every second; in the second round, one frame every two seconds; in the third round, one frame every five seconds; and so on for the intervals of 10, 15, 20, and 30 seconds—the smaller the sampling interval, the greater the number of frames to be analyzed.

Figure 7, as follows, shows how the sampling rate influences extraction efficiency, i.e., the algorithm's capacity to identify and extract a score from a frame where the score is displayed. Frames without scores were not taken into account.

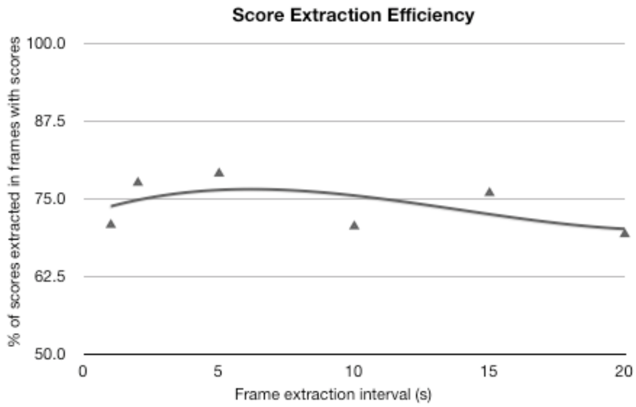

Figure 7. Efficiency in the score extraction process for different sampling rates.

Note that extraction efficiency is around 75%, independent of the sampling rate, which means that three out of four scores are successfully extracted. For example, in a two-minute video, with sampling at one frame per second, there are 120 frames to analyze. If 100 out of the 120 frames display a score, our algorithm can extract 75 scores from these frames, that is to say that in 75 frames, the OCR process returns relevant information. Also note that by increasing the sampling interval, extraction efficiency is slightly reduced, possibly as a consequence of the small number of frames to analyze. In fact, extraction efficiency should not present great variations when using different sampling rates, as frames are processed in isolation.

On the other hand, the algorithm's capacity to extract correct information is directly related to the number of frames analyzed. The greater the number of frames, the greater the volume is of available information upon which to make a decision. With a small volume of information, extraction becomes closer to guessing, as the algorithm chooses the highest probability option.

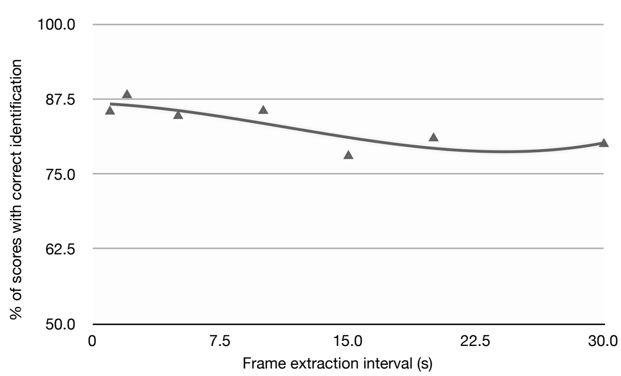

Figure 8. The probability of a correct identification by the OCR engine.

As expected, Figure 8 shows that with lower frame sampling intervals, the probability of correct score identification increases, and it is around 87%. This result means that, in 87% of extracted scores, the output of the OCR process is correct. It correlates directly with the OCR engine efficiency.

One important remark is that all tests, independent of the sampling rate used, returned 100% correct information for the desired attributes, i.e., names of the teams in the match, final score, and times when goals were scored (if any). In fact, the backward and forward analysis done in the merge step, and described in the previous section, allows that even if some error occurs in a frame processing, such as wrong score identification, the final video information will be correct, since the information is composed of several frames and eventual errors are discarded.

To demonstrate the advantages brought forth by the proposed architecture, we compare our cloud-based solution to an implementation using the traditional process (process all frames in a single server). Figure 9 shows the times, measured in seconds, required for processing different numbers of frames, always using one worker per frame (in the cloud environment), the proposed Split&Merge implementation (darker line) and the traditional process—local, single server (gray line).

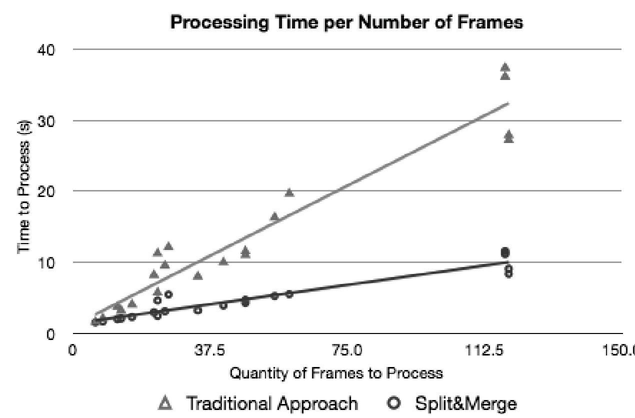

Figure 9. Total processing times for different numbers of workers.

Please note that the Split&Merge implementation takes around 30% of the total time spent using the traditional process, which is extremely interesting for applications where time-to-market is vital. Indeed, because the split and merge steps cannot be parallelized, their time to execute is independent of the number of active workers and results in a slight increase in processing times when using the Split&Merge approach.

To provide an idea of cost savings, we compare the Split&Merge approach when deployed in the public cloud to the costs of having a private infrastructure dedicated to the task. For this, we used the AWS cloud platform. First, we preconfigured instances for master and node servers, generating Amazon Machine Images (AMIs) that are publicly available[1]. Then the architecture was deployed, and

[1] Master Image AMI ID: ami-1148a378 - Node Image AMI ID: ami-245db54d

the processing nodes are dynamically created through the AWS API according to the number of frames to be processed, using the preconfigured images.

Taking into account the results presented in Figure 8, we have an approximate cost of $0.0007 per minute of video, considering a one-second sampling interval, to get three times faster results, using the AWS platform, with the additional advantage that it is possible to process very large videos in a few seconds. The total cost shows that the architecture of Split&Merge deployed in the public cloud is not only efficient in terms of processing time, but also in terms of deploy and operation costs.

Considering an optimal situation where there are unlimited available resources, it is possible to use experimental results to predict total costs and the number of nodes needed to process videos of different durations. Table 1, below, compares the traditional process with the proposed Split&Merge (S&M) approach. In this example, we set our goal of a processing time five or more times faster than the traditional approach. It is important to note that our total costs were computed using a cost-per-minute measure, albeit Amazon's minimum billing unit is the hour. We are taking into consideration scenarios where there are large numbers of videos to be processed, so it would not be the case that machines would shut down after a single process, but rather be in use for a few hours.

TABLE 1. COMPARISON BETWEEN TRADITIONAL PROCESS AND THE SPLIT&MERGE APPROACH, FOR ONE-SECOND SAMPLING INTERVALS.

Input Video Duration	*Traditional Process Duration*	*S&M Process Duration*	*Number of S&M Nodes*	*Normalized S&M Cost Using EC2*
30 sec.	9 sec.	3 sec.	30	$0.0003
5 min.	90 sec.	30 sec.	300	$0.003
30 min.	9 min.	3 min.	1800	$0.018
2 hour	36 min.	12 min.	7200	$0.072

Note that the Split&Merge approach deployed in a public cloud reduces the total processing time for a 2-hour video from 36 minutes to 12 minutes, with a total processing cost of $0.072. This result becomes even more attractive when used to extract information and annotate legacy content, for example, of all 380 matches of the entire last season of Brazilian soccer league. With the traditional process using one single server, the extraction and annotation would take up to nine days to finish. In comparison, using the proposed approach deployed in the cloud, all content could be analyzed and annotated in less than one hour, since all matches can be simultaneously processed in the cloud using multiple server instances, and at a cost of $27.36.

VII. CONCLUSIONS

The increasing demand for efficient processing of large volumes of information promotes the research on architectures and techniques that optimize the use of available resources. Cloud computing provides a viable solution, as it allows computing resources to be used on demand, increasing flexibility, scalability, and reducing costs.

In this paper we introduced a generalization of the Map Reduce paradigm, especially tailored for video processing and annotation. In particular, we proposed an OCR-based technique for video information extraction and further annotation, based on the Split&Merge architecture [22]. We described a case study in which we successfully implemented the proposed solution to extract metadata information from soccer videos, including information on teams involved in the match, final scores, the moment when goals were scored, and which players scored them. OCR data extraction effectiveness was around 87%. Results can, nevertheless, be improved with more training and additional testing aimed at optimizing the parameters of the OCR algorithm in use.

As a next step, we plan to investigate more suitable image processing techniques to increase the efficiency in the information identification process. Moreover, we also plan to perform the detection of additional events, for example, the names of the players exchanged during the match, the names of players in the starting lineup, and other textual information that is present in the video file. Extending the solution to domains other than sports is also in our future plans.

REFERENCES

[1] Lamberto Ballan , Marco Bertini , Alberto Bimbo , Lorenzo Seidenari, Giuseppe Serra, "Event detection and recognition for semantic annotation of video", Multimedia Tools and Applications, v.51 n.1, p.279-302, January 2011

[2] Kai Jiang, Xiaowu Chen, Qinping Zhao, "Automatic Compositing Soccer Video Highlights with Core-Around Event Model", 12th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics), p. 183 – 190, September 2011

[3] Toshio Sato , Takeo Kanade , Ellen K. Hughes , Michael A. Smith, Video OCR for Digital News Archive, Proceedings of the 1998 International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98), p.52, January 03-03, 1998

[4] Q. Zhang, L. Cheng, R. Boutaba. Cloud Computing: State-of-the-Art and Research Challenges. Journal of Internet Services and Applications(JISA). Springer London, Vol. 1(1), pp. 7-18, May 2010.

[5] Cearley, D. – Hype Cycle for Cloud Computing – 2009 – Gartner report number G00 168780, 2009.

[6] Armbrust, M., Fox, M., Griffith, R., et al. (2009) "Above the Clouds: A Berkeley View of Cloud Computing", In: University of California at Berkeley Technical Report no. UCB/EECS-2009-28, pp. 6-7, February 10, 2009

[7] Vouk, M.A., Cloud Computing – Issues, Research and Implementations, 30th International Conference on Information Technology Interfaces, June, 2008, pp. 31-40

[8] P.F. Felzenszwalb, D. McAllester, D. Ramanan, A discrimina- tively trained, multiscale, deformable part model. In CVPR, 1-8, 2008.

[9] A. Vedaldi and A. Zisserman, Structured output regression for detection with partial truncation. In NIPS, 2009.

[10] W. Yang, Y. Wang, and G. Mori, Recognizing human actions from still images with latent poses. In CVPR, 2010.

[11] Xu P, Xie L, Chang SF, Divakaran A, Vetro A, Sun H (2001) Algorithms and system for segmentation and structure analysis in soccer video. In: Proc. of int'l conference on multimedia & expo (ICME)

[12] Assfalg J, Bertini M, Colombo C, Del Bimbo A, Nunziati W (2003) Semantic annotation of soccer videos: automatic highlights identification. Comput Vis Image Underst 92(2–3):285–305

[13] Danielson, Krissi (2008-03-26). "Distinguishing Cloud Computing from Utility Computing". Ebizq.net. Retrieved 2010-08-22.

[14] Michael Miller, Cloud Computing – Web-Based Applications That Change The Way You Work And Collaborate Online, Que Publishing, 2009

[15] Gruman, Galen (2008-04-07). "What cloud computing really means". InfoWorld. Retrieved 2009-06-02;

[16] "Gartner Say's Cloud Computing Will Be As Influential As E-business". Gartner.com. Retrieved 2010-08-22;

[17] Carr, N., Here comes HaaS, 2006 – http://www.roughtype.com/archives/2006/03/here_comes_haas.php

[18] Vogels, W., (2008) "A Head in the Clouds – The Power of Infrastructure as a Service", In: First workshop on Cloud Computing in Applications (CCA'08), October, 2008.

[19] EC2 Amazon Elastic Compute Cloud (EC2) – http://aws.amazon.com/ec2/

[20] S3 Amazon Simple Storage Service (S3) – http://aws.amazon.com/s3/

[21] Dean, J., Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. In OSDI, 2004.

[22] Pereira, R.; Azambuja, M.; Breitman, K.; Endler, M.; , "An Architecture for Distributed High Performance Video Processing in the Cloud," Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on , vol., no., pp.482-489, 5-10 July 2010

[23] Karin Breitman, Markus Endler, Rafael Pereira, Marcello Azambuja, "When TV Dies, Will It Go to the Cloud?," Computer, vol. 43, no. 4, pp. 81-83, Apr. 2010;

[24] Apache Hadoop - http://hadoop.apache.org/mapreduce/

[25] Tesseract – http://code.google.com/p/tesseract-ocr/

[26] Stephen V. Rice, Frank R. Jenkins, and Thomas A. Nartker - The Fourth Annual Test of OCR Accuracy - http://www.isri.unlv.edu/downloads/AT-1995.pdf

[27] ImageMagick – http://www.imagemagick.org

[28] CloudCrowd - http://wiki.github.com/documentcloud/cloud-crowd/getting-started

[29] MEncoder – http://www.mplayerhq.hu

[30] Converting video formats with FFMpeg, Linux Journal archive – Issue 146, June 2006, pp. 10

[31] Zhang, L., RESTful Web Services. Web Services, Architecture Seminar, University of Helsink, Department of Computer Science, 2004.