A Split&Merge Architecture for Distributed Video Processing in The Cloud

Rafael S. Pereira[1], Karin K. Breitman (advisor)[1]

[1]Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro – RJ – Brazil

{rpereira,karin}@inf.puc-rio.br

Abstract. *The Map Reduce approach, proposed by Dean & Ghemawat [10], is an efficient way for processing very large datasets using a computer cluster and, more recently, cloud infrastructures. Traditional Map Reduce implementations, however, provide neither the necessary flexibility (to choose among different encoding techniques in the mapping stage) nor control (to specify how to organize results in the reducing stage), required to process video files. The Split & Merge tool, proposed in this thesis, generalizes the Map Reduce paradigm, and provides an efficient solution that contemplates relevant aspects of intense processing video applications.*

1. Introduction

As computer systems evolve, the volume of data to be processed increases significantly, either as a consequence of the expanding amount of available information, or due to the possibility of performing highly complex operations that were not feasible in the past. Nevertheless, tasks that depend on the manipulation of large amounts of information are still performed at large computational cost, i.e., either the processing time will be large, or they will require intensive use of computer resources.

In this scenario, the efficient use of available computational resources is paramount, and creates a demand for systems that can optimize the use of resources in relation to the amount of data to be processed. This problem becomes increasingly critical when the volume of information to be processed is variable, i.e., there is a seasonal variation of demand. From an economical perspective seasonal demands do not justify a massive investment in infrastructure, just to provide enough computing power for peak situations. In this light, the ability to build adaptive systems, capable of using on demand resources provided by Cloud Computing [16][17][18], is very interesting.

One of the areas where the volume of information to be processed is increasing at a very fast pace is Video Production and Distribution [38]. The increasing popularity of videos on the Internet, allied to the advances in network technology seen in the last decade, is drastically changing TV as we know it. What we see, in fact, is that every day the evolution of technology is challenging established business models, being driven by consumers that require a more complete, flexible and interactive experience. From a technical point of view, there are also big challenges, which include the ability to process, index, store and distribute very large amounts of data. In summary, the challenge for video production and distribution companies is to process unprecedented volumes of high definition video, and distribute it across several different devices and several different media platforms, which means a significant increasing in the encoding process, and, as consequence, in the time and cost required to perform this task.

Based on this scenario, the scope of this work is focused to the range of software applications that have the following characteristics:

- there is a need for processing large volumes of data, i.e., applications where the amount of information to be processed to obtain the desired result is so large that the time required to complete this task using a traditional approach is not acceptable;
- there is a seasonal demand for processing, i.e., there are moments where the amount of information to be processed exceeds the capacity of available resources, while at other times the same resources may be idle;
- extraordinary and *ad hoc* situations, where the infrastructure will need to scale up or down repeated times, e.g., transcoding all legacy content from one compression technology to a different one;
- there is a gain in reducing the maximum possible time necessary to obtain the desired result, e.g., immediacy;

Main Contributions. In this thesis we present an architecture for processing large volumes of video. Our proposal takes advantage of the elasticity provided by Cloud Computing infrastructures. As main contributions of this research, we can highlight:

- The proposal a general architecture for large scale distributed video processing system, which is capable to address these problems;
- Development of an algorithm that allows parallel and distributed task processing, which takes advantage of the elasticity provided by Cloud platforms, adapting computational resources according to current demands;
- The implementation different prototypes to demonstrate the feasibility of the proposed approach;

2. Background and Related Work

It is universally acknowledged that video compression processes have greatly evolved over the past decades, especially in what concerns the efficiency of the process in relation to the amount of bits encoded [15,22]. However, as overall compression efficiency increases, also does its computational cost, often translated into the increase of production times [24]. This behavior becomes obvious when we compare compression efficiency of different codec's, e.g., MPEG-2, and the more modern H.264. We observe that H.264 compression is much more efficient than that obtained by MPEG-2, especially when dealing with very low bitrates (fundamental for Internet distribution). However H.264 encoding requires much more time to process.

We observe that a great part of the research effort in the area has been directed to achieving better compression efficiency, in respect to the volume of encoded data [22,23,24]. Our experience with the major TV broadcaster in Latin America, TV Globo, has demonstrated that, for those who make practical and intensive use of video compression processes, more attention must be given to the compression time efficiency issue [10].

Today there are several computing paradigms whose focus is to reduce the processing time of large volumes of data. The Map-Reduce paradigm [2], 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.

However, despite being a very appealing and efficient technique for processing large volumes of data, to a Map-Reduce architecture truly effective, one needs several machines acting as nodes, which often requires a large upfront investment in infrastructure. This point is extremely critical in situations where the processing demand is seasonal. In such cases, the use of public Clouds, for information processing and storage, is emerging as an interesting alternative. The Hardware as a Service (HaaS) [5] paradigm, relieves the burden of making huge investments in infra-structure, and at the same time supports on-the-fly resizing of resources, and adaptation to current needs. In addition, fault tolerance issues and the need of a shared file system make the deployment of a Map-Reduce architecture complex and costly.

Considering the usage of a parallel processing architecture for video processing, one important point that must be highlighted is that video files possess characteristics that hinder the direct application of distributed processing techniques, without first adjusting the way they deal with the information contained in the video. Firstly we must remark that a video file is a combination of an audio and a video stream, which are processed independently and using different algorithms, however must be decoded in a synchronized way. This means that the video container must maintain the alignment between the two streams. In addition, a video stream can be decomposed into sets of frames, which are strongly correlated, especially in relation to its subsequent frames. In fact, it is the correlation among the frames that allows the reduction of temporal redundancy in certain codec's.

Therefore, without adaptation, the Map-Reduce approach is of very little use to video compression. Firstly, a classical Map Reduce implementation would divide the tasks using a single mapping strategy. Video encoding requires that we use different strategies to compressing video and audio. Secondly, the traditional Map-Reduce approach does not take into consideration the order, much less the correlation of the pieces processed by individual workers. Video encoding requires that we take into consideration frame correlation, and more importantly, the order of frames.

In the next section we present the proposed Split&Merge approach, an architecture for video compression, which reduces significantly the time required for this process through the parallel and distributed processing based in Cloud Computing platforms.

3. The Split&Merge Architecture

If we analyze the Map-Reduce paradigm [10] in its essence, we note that process optimization is achieved by distributing tasks among available computing resources. It is precisely this characteristic preserved in the proposed architecture. The possibility of breaking an input, and processing its parts in parallel, is the key to reducing overall processing times [31].

With a focus on these issues, the proposed Split&Merge architecture borrows from key Map-Reduce concepts, to produce a simple, and yet general, infrastructure in which to deal with distributed video processing. The generalization of this idea, lead to an architecture in which it is possible to isolate and diversify the implementation for the split, distribute, process and merge steps.

Roughly speaking, the S&M can be summarized in follow steps: ***1) split step:*** split a video input file into several chunks; ***2) process step:*** process all chunks simultaneously in a Cloud infrastructure, creating and destroying virtual machines automatically according to the number of chunks to be processed; ***3) merge step:*** merge all processed fragments into a single continuous compressed video, with audio and video streams perfectly joined and synchronized.

Regarding the Split Step, the greatest challenge is that in video processing, differently from text files, it is not possible to split a video file anywhere. If the input video already provides some form of temporal compression, then it would be necessary to first identify its key-frames, so that the cuts are made at their exact positions. Furthermore, to avoid synchronization problems between audio and video streams, we must firstly separate the two, so that they can be independently compressed. If processed together, chunks containing both audio and video may generate synchronization problems, as audio frames not necessarily have the same temporal size than video frames.

A key point in the fragmentation of the input video is to determine the size of the chunks. This decision is closely related to the output that should be generated, that is, the video codec and compression parameters passed to it in the processing step. Fragmentation in chunks performed indiscriminately, will produce an output video with an excess of key-frames, which reduces the efficiency of compression.

Once video is fragmented, the chunks are distributed among the nodes to be processed. The number of nodes is dynamically managed by the S&M provisioning agent according to the number of chunks, since the elasticity provided by Cloud infrastructures allows for an automatic resource management. In this process step, a compression algorithm is applied to each chunk, resulting in a compressed chop of the original video.

The Merge Step presents a very interesting challenge, which consists of reconstructing the original content from its separate parts, so that the fragmentation process is entirely transparent to the end user. The first phase of the merge step is joining processed video chunks, which can be accomplished easily by ordering the fragments, and rewriting the container. The result is the full compressed video stream, with the same logical sequence of the input. Following, we remix the audio and video streams, synchronizing the contents, and generating the expected output. This remix reconstructs the container realigning the audio stream with the video stream, and, since the duration of both streams do not change during the compression process, the video content is synchronized.

After the split, process and merge steps, implemented using the proposed architecture, we created a fully parallel and distributed video compression process, where different content chunks can be simultaneously processed using resources in the Cloud infrastructures.

4. Results

In order to validate the proposed architecture we experimented using Amazon's AWS services. We deployed S&M for the encoding of different sequences of videos, evaluating the total time required for the encoding process, and comparing it with the total time spent in the traditional process, where the video is encoded without fragmentation, i.e. all content is rendered on a single server. In Figure 1 we depict the comparison between total times, measured in seconds, required for the encoding of different video sequences.

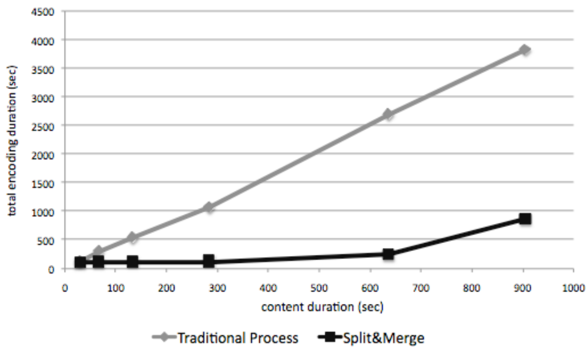

Figure 1. Comparison of encoding times between traditional approach and S&M

Note that, while the total encoding time using the traditional process, grows linearly with increasing duration of the video input, the S&M, average process times remain almost constant for short duration videos. In fact, the total CPU time consumed, which is the sum of the CPU usage in all nodes, will be greater in the S&M approach, however, the distribution of processing among several nodes for parallel execution will reduce the total encoding duration. This result is quite significant when one considers videos of short and average duration. In the case when we have a video about 10 minutes long, the total time for encoding using the technique of S&M is equivalent to less than 10% of the total time spend using the traditional process, which is extremely interesting for applications where time to market is vital.

Another point worth consideration is the monetary cost of the S&M approach when deployed in a public Cloud infrastructure, against the cost of having a private infrastructure dedicated to this task. Taking into account the results of the tests above, and also a production of 500 minutes a month, we will have, at the end of one year, an approximate cost of $25,000 using the platform of Amazon AWS, with the advantage to be possible to produce all content in a few minutes. This value is comparable to the cost of a single server, not considering the depreciation and maintenance, which makes the architecture of S&M deployed in the public Cloud not only efficient in terms of processing time, but also in terms of deployment and operation costs.

5. Conclusions

In this thesis we argued that the combination of mature research in data compression to emergent Cloud Computing technology opens up a great number of research challenges and possibilities. We introduced the Split & Merge architecture, for high performance video processing, a generalization of the MapReduce paradigm, that rationalizes the use of resources by exploring on-demand computing. We also presented experimental results that demonstrate the feasibility of the proposed approach.

The fact that the elastic capacity of a public Cloud such as AWS is very large, i.e., offers access to practically unlimited resources, changes fundamental premises that guided the research in this area so far, and have the potential implications:

- The quality x compression time trade off no longer holds - new algorithms may be developed aiming at maximum compressing quality, as processing times can be reduced by using parallel implementations such as the S&M approach.
- Choosing optimal compression profiles is known to be very difficult. The possibility of generating several options, in parallel and at low cost, may be beginning of a new era of experimentation in data compression.
- Adapting parallel approaches to work with linear algorithms require a great deal of effort. In the case of the S&M approach, we generalized the Map-Reduce paradigm to allow for the use of multiple compressing techniques (different for audio and video processing), as well as to regain control during the reduce step. Although our results are encouraging, we understand a lot remains to be done. In particular, we believe that the development of self tuning algorithms to determine optimal chunk size could bring forth very good results.

Finally we would like to highlight that techniques used in the split and merge steps are hotspots, i.e., they can be exchanged and customized as needed. That ensures flexibility, adaptation, extensibility and generality to the proposed architecture.

5.1 Publications

Books

Pereira, R.; Breitman, K.; "VIDEO PROCESSING IN THE CLOUD", Springer, Springer Briefs in Computer Science, Sep. 2011

Conferences and Periodicals

Breitman K., Pereira R., Azambuja M., Endler M.; "WHEN TV DIES, WILL IT GO TO THE CLOUD?"; IEEE Computer Magazine, vol. 43, no. 4, pp. 81-83, Apr. 2010, ***Classificação CAPES: Qualis A2***

Pereira R., Breitman K.; "A CLOUD BASED ARCHITECTURE FOR IMPROVING VIDEO COMPRESSION TIME EFFICIENCY: THE SPLIT&MERGE APPROACH"; IEEE Data Compression Conference (DCC) '11, 30-mar-11, ***Classificação CAPES: Qualis A2***

Pereira R., Azambuja M., Breitman K., Endler M.; "AN ARCHITECTURE FOR DISTRIBUTED HIGH PERFORMANCE VIDEO PROCESSING IN THE CLOUD"; IEEE Cloud'10, 07-jul-10

Pereira R., Azambuja M., Breitman K., Endler M.; "CLOUD TV"; Microsoft Cloud Futures Workshop, Redmond-WA US, 09-apr-10

Pereira R., Azambuja M., Breitman K., Endler M.; "ARCHITECTURES FOR DISTRIBUTED HIGH PERFORMANCE VIDEO PROCESSING IN THE CLOUD"; CloudSlam'10, 26-mar-10

Pereira R., Breitman K.; "ARQUITETURAS PARA PROCESSAMENTO DE ALTA PERFORMANCE DE VIDEOS NA NUVEM"; Consegi'10, 20-aug-10

Pereira R., Azambuja M., Breitman K., Endler M.; "ARQUITETURAS PARA PROCESSAMENTO DE ALTA PERFORMANCE DE VIDEOS NA NUVEM"; Cloud Computing Brazil, Rio de Janeiro-RJ BR, 27-apr-10

References

All references are available in the full version