7. Conclusions

The need for systems that can deal efficiently with increasingly large volumes of information, is growing at an astounding pace. New architectures and techniques to optimize the use of available resources are an important part of today's research agenda. The emergence and subsequent evolution of services in the Cloud, provide the necessary flexibility and makes the task of processing of large datasets more accessible and scalable, especially in scenarios with varying demand.

Today there are various solutions geared towards the optimization of computer resource usage, and performance improvement in information processing. Among them, we highlight the Map-Reduce paradigm, firstly proposed as a solution to optimize web searches. We understand, however, that it requires some adaptation if it is to be used efficiently in other domains. In this thesis we proposed a generalization of the Map-Reduce paradigm, adapted and finely tuned to deal with video compression. We proposed a more efficient architecture than the original one proposed by Dean and Ghemawat [10], that can be deployed both in private clusters, as in commercial Cloud infrastructures. We generalized the Map-Reduce paradigm by proposing the Split&Merge approach, a video processing strategy that allows for the use of multiple video compressing techniques (different for audio and video processing), and provides total control during the reduce step.

It is important to note that the techniques used in both the Split and Merge steps are hotspots, i.e., techniques can be exchanged and customized as needed. That ensures flexibility, adaptation, extensibility and generality to the proposed architecture. In the case of video processing, it is paramount to allow a choice among *codecs*, containers, audio streams, and different splitting techniques. To illustrate one such scenario, let's take the cases where the input video has no temporal compression (MJPEG [16] for example). In such cases, the split operation can be performed at any video frame. Conversely, cases when the input is a video with p-frame only temporal compression (H.264 [17] Baseline Profile for example), it is mandatory to identify key-frames before splitting.

The generalization of this idea, lead to an architecture in which it is possible to isolate, and modify the implementations for the split, process and merge steps.

The current implementation encompasses a choice of techniques that range from video compression to image processing, through macroblock fragmentation for simple word counts. This choice can be done at scheduling time, as the current implementation embeds the technique of choice in a web service, at run time.

We also pointed out that the deployment in a private cluster is only justified in situations where there is a constant demand for processing. The alternative, i.e., situations where the demand is seasonal, makes use of a commercial, public cloud platform, such as Amazon, where one pays for the amount of resources used.

We validated the proposed Split&Merge architecture and verified its behavior for processing large volumes of information, both in private clusters and in commercial, public cloud scenarios, by the implementation of the Split and Merge approach for a distributed video compression application. As a result, we were able to dramatically reduce video encoding times. In fact, we demonstrated that if we scale the architecture up to a maximum of one worker per 250 frame chunk, we can guarantee fixed encoding times, independently of the size of the video input. This is extremely interesting for content producers, because it is possible to establish entry independent software level agreements (SLAs) for video encoding services.

We also developed a Split and Merge application for automatic video data extraction using OCR. As result, we obtained a tool capable of extracting basic text information present inside soccer video frames, e.g. the name of the teams, score and elapsed time, which can be used for automatica content annotation. Applications such as this can greatly leverage search and retrieval of legacy video. We also demonstrate that the usage of Split&Merge in this application significantly reduces the time and cost required to perform this process.

7.1 Future Work

The deployment of an application to compress videos in a Cloud, following the proposed architecture, is still something that presents several challenges. The main limiting factor of this approach is, at least in our geographic area, the lack of availability of bandwidth between content producers, and services in the Cloud. For example, using a Cloud service to compress high definition videos for web distribution is prohibitive by today's standards, as the videos to be processed have to be uploaded in their original, high definition, formats. In this particular case, deployment in a private cluster is more efficient in respect to the total production time.

Finally, we believe our architecture could be used in applications other than video processing. Future work includes the experimentation with different datasets, to determine the actual scope of the proposed architecture. Another issue we would like to investigate is the incorporation of autonomic computing mechanisms to help anticipate and identify faults, and the implementation of efficient prevention and recovery mechanisms. That will contribute to making the present solution more robust.