

# The Importance of Elasticity in a Computing Cluster

Marcel Stolin  
marcelstolin@gmail.com

## Abstract

Since the beginning of Big Data, batch processing was the most popular choice for processing large amounts of generated data. These existing processing technologies are not suitable to process the large amount of data we face today. Research works developed a variety of technologies that focus on stream processing. Stream processing technologies bring significant performance improvements and new opportunities to handle Big Data. In this paper, we discuss the differences of batch and stream processing and we explore existing batch and stream processing technologies. We also explain the new possibilities that stream processing make possible.

## 1 Introduction

## 2 Related Work

## 3 Elasticity

<code>map(String key, String value):</code>	<code>reduce(String key, Iterator values):</code>
// key: document name	// key: a word
// value: document contents	// values: a list of counts
for each word w in value:	int result = 0;
EmitIntermediate(w, "1");	for each v in values:
	result += ParseInt(v);
	Emit(AsString(result));

Figure 1: Pseudo code of counting the number of occurrences of each word in a large selection of documents.

## 4 Design

## 5 Conclusion

This paper explains the two data analysis concepts batch processing and stream processing. Since realtime analysis is needed to face the issues of today's demands, batch processing is still being used for legacy implementations and data analysis where no efficient algorithms are known. Stream processing offers new opportunities to handle big data and response with an immediate result to the user.