Accepted Manuscript

Efficient finer-grained incremental processing with MapReduce for big data

Liang Zhang, Yuanyuan Feng, Peiyi Shen, Guangming Zhu, Wei Wei, Juan Song, Syed Afaq Ali Shah, Mohammed Bennamoun

PII: S0167-739X(17)30313-8
DOI: https://doi.org/10.1016/j.future.2017.09.079
Reference: FUTURE 3735

To appear in: *Future Generation Computer Systems*

Received date : 25 February 2017
Revised date : 31 May 2017
Accepted date : 30 September 2017

Please cite this article as: L. Zhang, Y. Feng, P. Shen, G. Zhu, W. Wei, J. Song, S.A. Ali Shah, M. Bennamoun, Efficient finer-grained incremental processing with MapReduce for big data, *Future Generation Computer Systems* (2017), https://doi.org/10.1016/j.future.2017.09.079

Efficient Finer-grained Incremental Processing with MapReduce for Big Data

Liang Zhang[a], Yuanyuan Feng[a], Peiyi Shen[a*], Guangming Zhu[a], Wei Wei[c], Juan Song[a], Syed Afaq Ali Shah[b], Mohammed Bennamoun[b]

[a] School of Software Engineering, The University of XiDian
[b] School of Computer Science and Software Engineering,The University of Western Australia
[c] School of Computer Science and Engineering，Xi'an University of Technology, Xi'an 710048, China

HIGHLIGHTS

- Illustrate the shortcoming of coarse grained result reusing for incremental processing.
- An algorithm to divide input datasets stably and quickly.
- Optimize the procedure of finding delta data and deliver an efficient and stable implementation

ARTICLE INFO

Article history:

Keywords:
Big data
Incremental processing
Finer grained reusing
Yarn

ABSTRACT

With the continuous development of the Internet and information technology, more and more mobile terminals, wear equipment etc. contribute to the tremendous data. Thanks to the distributed computing, we can analyze the big data with quite high speed. However, many kinds of big data have an obvious common character that the datasets grow incrementally overtime, which means the distributed computing should focus on incremental processing. A number of systems for incremental data processing are available, such as Google's Percolator and Yahoo's CBP. However, in order to utilize these mature framework, one needs to make a troublesome change for their program to adapt to the environment requirement.

In this paper, we introduce a MapReduce framework, named *HadInc,* for efficient incremental computations. HadInc is designed for offline scenes, in which real-time is needless and in-memory cluster computing is invalid. HadInc takes the advantages of finer-grained computing and Content-defined Chunking(*CDC*) to make sure that the system can still reuse the results which we have computed before, even if the split data has been changed seriously. Instead of re-computing the changed data entirely, *HadInc* can quickly find out the difference between the new split and the old one, and then merge the delta and old results into the latest result of the new datasets. Meanwhile, the dividing stability of the datasets is a key factor for reusing the results. In order to guarantee the stability of the dataset's division, we propose a series of novel algorithms based on *CDC*.

We implemented *HadInc* by extending the Hadoop framework, and evaluated it with many experiments including three specific cases and a practical case. From the comparing results it can be seen that the proposed *HadInc* is very efficient.

1. INTRODUCTION

The thriving of big data is to benefit from the Internet and mobile communication technology. In our daily life, different kinds of devices collect information round the clock. The servers receive and analyze such information by using some intelligent methods like machine learning, and return the results to users[1]. To this end, we have to face some specialties of the big data, such as large scale[2], incremental or dynamic data changing, immediately user response, and so on.

As times passes, the datasets accumulate and continuously grow in size. If the fresh results are required periodically, in view of the enormous size of datasets, it's unadvisable to re-compute all the datasets at each time[3]. Re-computation will not only cost much time, which will decay timelines of the results, but also affect the commercial value of the datasets. For example, when people surf websites, the most action they do is click[4], which is unique and immutable in the timeline. If we recalculate all the datasets at each time for the browsing information, the reduplicative and useless calculation is beyond 85%. Besides, because of the modification on old data which can make the computation more complex, the schema for processing incremental datasets should be enough flexible and must guarantee a stable efficiency in different incremental cases.

In this paper, we propose an incremental processing system named HadInc, which simultaneously takes the characters of big data and incremental processing into consideration. HadInc performs efficiently both in appending and modifying cases of the big datasets, which means HadInc can be applied to varying incremental processing scenes with impressive stability. We will introduce the HadInc around following properties:

Stability: Some of the existing incremental processing frameworks always divide the datasets into splits with CDC[6] just in Preprocessing, which means they can only reuse the pre-results in a coarse grain. The other frameworks like *HadUP*[8] divide the datasets by a fixed length chunking. *HadUP* can overcome the coarse-grained reusing problem, but it's not stable enough because of the dividing schematics. When the datasets changes heavily at some specific parts, the dividing result can be totally different from last division, which will increase the calculation load sharply. In contrast, *HadInc* differs from those traditional frameworks in that it takes advantage of *CDC* both