Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Adaptive, scalable and reliable monitoring of big data on clouds

Mauro Andreolini *,[1], Michele Colajanni [2], Marcello Pietri [2], Stefania Tosi [2]

University of Modena and Reggio Emilia, Italy

HIGHLIGHTS

- Real time monitoring of cloud resources is crucial for system management.
- We propose an adaptive algorithm for scalable and reliable cloud monitoring.
- Our algorithm dynamically balances amount and quality of monitored time series.
- We reduce monitoring costs significantly without penalizing data quality.

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form
8 May 2014
Accepted 18 August 2014
Available online xxxx

Keywords:
Adaptivity
Monitoring
Cloud computing
Big data
Scalability

ABSTRACT

Real-time monitoring of cloud resources is crucial for a variety of tasks such as performance analysis, workload management, capacity planning and fault detection. Applications producing big data make the monitoring task very difficult at high sampling frequencies because of high computational and communication overheads in collecting, storing, and managing information. We present an adaptive algorithm for monitoring big data applications that adapts the intervals of sampling and frequency of updates to data characteristics and administrator needs. Adaptivity allows us to limit computational and communication costs and to guarantee high reliability in capturing relevant load changes. Experimental evaluations performed on a large testbed show the ability of the proposed adaptive algorithm to reduce resource utilization and communication overhead of big data monitoring without penalizing the quality of data, and demonstrate our improvements to the state of the art.

1. Introduction

An increasing number of applications deployed over the cloud operates on big data that we consider a collection of data sets so large and complex that it becomes difficult to gather, store, analyze and visualize through traditional approaches [19,23]. To effectively manage large-scale data centers and cloud systems, operators must understand the behavior of systems and applications behavior producing big data. This requires continuous real-time monitoring integrated with on-line analyses that can be related to performance, prediction, anomaly detection and SLA satisfaction. In similar contexts, the monitoring system presents all the features that are typical of an application producing and working on big data: volume, variety, velocity, veracity (the so called "4 Vs" of IBM scientists). Hence, a key challenge of a scalable monitoring infrastructure is to balance the monitoring and analysis costs incurred with the associated delays, against the benefits attained from identifying and reacting timely to undesirable or non-performing system states such as load spikes.

Previous attempts at reducing the overhead of a real-time monitoring infrastructure present several drawbacks that make them inapplicable to a context of cloud-based applications handling big data. Some proposals for reducing the data set dimension operate on the whole time series (e.g., [8,32]). Others use fixed sampling intervals and do not consider that in highly heterogeneous systems the statistical characteristics of the monitored time series change [12,4]. Another class of work focuses on specific classes of performance indexes and it is not generalizable (e.g., [15]). Finally, there are well designed architectures that do not scale to the volume of data requested by big data applications [6].

This paper introduces a novel real-time adaptive solution for scalable and reliable monitoring of applications producing big data. It strives to achieve this goal by reducing the amount of monitoring

* Corresponding author.
E-mail addresses: mauro.andreolini@unimore.it (M. Andreolini), michele.colajanni@unimore.it (M. Colajanni), marcello.pietri@unimore.it (M. Pietri), stefania.tosi@unimore.it (S. Tosi).

[1] Department of Physics, Computer Science and Mathematics, Modena 41125, Italy.

[2] Department of Engineering "Enzo Ferrari", Modena 41125, Italy.

http://dx.doi.org/10.1016/j.jpdc.2014.08.007
0743-7315/© 2014 Published by Elsevier Inc.

data produced. By adapting sampling intervals to continuously changing data characteristics, it reduces computational and communication costs without a penalization on the reliability of monitored data. The main idea that drives the algorithm adaptivity is simple. When the system behavior is relatively stable, our solution settles for large sampling intervals so that the quantity of data that is gathered and sent for further analysis is reduced. When significant differences between samples occur, the sampling interval is reduced so to capture relevant changes in system performance. The proposed solution automatically chooses the best settings for monitoring parameters, and it updates such settings so to adapt to data characteristics. Moreover, monitoring settings are adapted to the preference of the administrator.

The proposed algorithm gives system administrators the possibility of choosing the best trade-off between reducing computational and communication overhead and preserving the reliability of monitored data. However, this reduction comes at the cost of penalizing the reliability of monitored data because discarding samples keeps the monitoring overhead low but limits the possibility of capturing load changes promptly. Moreover, monitoring solutions should adapt to the frequent changes in the statistical characteristics of monitored datasets. An effective monitoring solution has to support dynamic data acquisition from heterogeneous sources, and to be adaptive to data characteristics and operational needs [27].

This work extends our preliminary findings published in [22] in three directions. We formulate the problem of real-time monitoring in the big data field, where requirements of scalability and reliability are mandatory. We improve the definitions of the proposed adaptive monitoring solution and of its parameters, with detailed descriptions of the algorithm phases. We add an extensive evaluation of the algorithm performance and a comprehensive comparison with respect to state-of-the-art solutions. Experiments show that the proposed adaptive algorithm is able to improve the ability of capturing relevant load changes in up to 55% more than static solutions; in our experiments, the misdetection of load spikes has been also very low (less than 5%). Implementations of adaptive versions for existing solutions do not achieve the performance of our proposal, that benefits an effective tuning of monitoring parameters according to data characteristics and administrator preferences. These results represent a major improvement with respect to the state-of-the-art techniques which either are reliable and resource intensive or tend to be highly scalable by worsening the reliability of sampled data [12,19,13,32].

The remainder of this paper is organized as follows. Section 2 defines the problem of real-time monitoring for big data on clouds. Section 3 presents the proposed adaptive monitoring algorithm. Section 4 introduces the experimental testbed used for the evaluations. Section 5 analyzes experimental results achieved on real scenarios involving big data applications. Section 6 compares our proposal against the state-of-the-art monitoring solutions. Section 7 concludes the paper with some final remarks.

2. Problem definition

In large data centers hosting big data applications the only way to build a scalable monitoring infrastructure is to reduce the amount of monitoring data without sacrificing its statistical properties that are at the basis of any post-gathering analysis. Any monitoring algorithm can be characterized according to this trade-off. To this purpose, we introduce two parameters.

The first parameter G (*Gain*) is defined as one minus the ratio between the number of samples collected by the considered monitoring algorithm and the number of samples collected by the baseline monitoring algorithm that samples data at the highest possible frequency t^0 (e.g., 1 s). Both monitoring algorithms are supposed to operate over the same time interval that must be sufficiently long to be statistically relevant. G assumes values in the [0, 1] interval. Higher values of G denote algorithms aiming to reduce the computational and communication overhead due to monitoring.

$$G = 1 - \frac{N(t)}{N(t^0)}. \tag{1}$$

The second parameter, Q (*Quality*) quantifies the ability of an algorithm to accurately represent load changes in system resources (e.g., load spikes). A comprehensive metrics for estimating Q must take into account two factors: the error introduced by monitors using sampling intervals larger than t^0 (that is, the distance between the original monitored dataset and the reduced one) and the ability to evidence load spikes in the monitored dataset. For this reason, we define Q as a combination of the *NRMSE* (Normalized Root Mean Square Error) [10], and the *Fmeasure* as the weighted average of precision and recall in spike detection [30]:

$$Q = \frac{Fmeasure + (1\text{-}NRMSE)}{2}, \tag{2}$$

where *Fmeasure* and *NRMSE* take values $\in [0, 1]$. Q assumes values in the [0, 1] interval. A further motivation for combining two parameters into Q instead of one is due to the fact that datasets in the considered scenarios are highly variable. As stated in [25], the *NRMSE* measure alone is unable to guarantee an accurate quality measure when the statistic characterization of the dataset is highly variable. For this reason, we integrate *NRMSE* with *Fmeasure*, that measures the ability of the monitoring algorithm to identify significant load spikes.

$$Fmeasure = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{3}$$

As detailed in [7], *recall* is the fraction of spike detections that are successfully identified, while *precision* is the fraction of relevant detections over the total number of spike detections. In this paper, we consider as a "false positive" (FP) a load spike detected by the monitoring algorithm when the original time series does not exhibit one. This can happen because a generic monitoring algorithm modifies the original time series and can accidentally introduce load spikes in the representation of system load. It can happen that precision is lower than 1. By combining the two metrics, *Fmeasure* gives a global estimation of the detection quality. An *Fmeasure* value close to 1 denotes a good detection quality, while it is lower for algorithms with worse capability in capturing load changes.

The trade-off of a monitoring algorithm can be expressed as a weighted mean of G and Q through the E parameter (*Evaluation*):

$$E = w \cdot G + (1 - w) \cdot Q, \tag{4}$$

where $w \in (0, 1)$ is a tuning constant chosen by the administrator. As the amount of saved data impacts on the quality of the representation, we must allow the system administrator to decide how to regulate the trade-off between overhead reduction and information reliability. Values of $w > 0.5$ put more emphasis on G than on Q; the opposite is true for $w < 0.5$; while $w = 0.5$ gives equal importance to both parameters.

Existing monitoring methods (e.g., [12,15]) collect data at fixed sampling intervals and forward new information to analysis modules only if it differs from the previous collected one by some static numeric threshold. Although this approach can achieve high values of G by reducing the amount of gathered and transmitted data, it may lead to highly inaccurate results and very low Q values due to the missing of most of spikes in data. As an example, Fig. 1 reports two scenarios. Fig. 1 offers a detailed