Journal of Network and Computer Applications 87 (2017) 100–115

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

Real-time scheduling based on optimized topology and communication traffic in distributed real-time computation platform of storm

Chunlin Li[a,b,*], Jing Zhang[a], Youlong Luo[c]

[a] Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P.R. China
[b] Jiangsu Key Laboratory of Big Data Analysis Technology, Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing, China
[c] School of Management, Wuhan University of Technology, Wuhan 430063, China

ARTICLE INFO

Keywords:
Storm
Topology optimization
Executor scheduling
Load balancing

ABSTRACT

In recent years, Storm, an open source distributed real-time computation system, has gained significant amount of popularity in cloud computing industry due to its high reliability and good processing mode. The key in tuning Storm performance lie in the strategy deployed a topology on Storm cluster and the scheduling method used in Storm scheduler. A Storm topology refers to a graph of real-time computation, which provides the logic view of the data process. Currently, Storm adopts a static topology deployment strategy and a simplistic scheduling method, which not only limits flexibility in topology tuning, but also leads to low efficiency in load balancing among its worker nodes. To this end, a Storm topology dynamic optimization algorithm based on the theory of constraints (STDO-TOC) is proposed to dynamically eliminate the performance bottleneck of the topology. In addition, a real-time scheduling algorithm based on topology and traffic (TS-Storm) is proposed to effectively solve the problem of inter-node load imbalance. Extensive experiment results show that, our newly proposed topology deployment strategy and scheduling method can largely improve performance of Storm in term of better system throughput, shorter average delay and latency, and less inter-node traffic.

1. Introduction

As real-time big data processing (Apache Storm; Jones, 2013) is critical to the performance of many newly emerged IT technologies including cloud computing, Internet of Things, mobile Internet, social media, etc., it has drawn enormous concerns from both industry and academia. Among all the variant real time big data processing technologies, Storm (Pham et al., 2016; Toshniwal et al., 2014; Karunaratne et al., 2016), an open source distributed real-time computation system, has gained substantial concerns among many top tier IT companies such as Twitter, Groupon, Yahoo, Alibaba (Apache Storm), due to its explicit performance advantage of low latency, high throughput, distributed processing, good scalability, and real time fault-tolerance.

Currently, the key aspects in tuning Storm performance lie in the strategy used to deploy a topology on Storm cluster and the scheduling policy used in scheduler. However, most of Storm topology deployment methods cannot adjust topologies adaptively. And as well as the other stream processing systems, Storm which lacks an intelligent scheduling mechanism cannot effectively solve the problem of load balancing among worker nodes.

Some research work has been done to optimize Storm topologies for better execution efficiency and performance. Most of them focused on the alteration of topology structure and the modification of topology parameters. However, in their work storm topology cannot be adjusted adaptively due to lack of automatic topology structure checking mechanisms, which may lead to unsatisfactory optimization result. Meanwhile, the scheduler responsible for the cluster resources allocation is another critical component making great impact on performance of Storm (Petrini et al., 2013). As Storm adopts pseudo-random round robin scheduling algorithm as its default scheduling algorithm, it fails to consider the correlation between topology structure and overall performance. Some researches on Storm scheduler have been done to ameliorate the default scheduling algorithm scheduling algorithms with respect to topology structure, inter-node traffic, resource utility, and QoS. While most existing proposal may be able to improve one performance metric in terms of traffic, resource availability or QoS, they are not able to solve global problem such as load balancing.

This paper aims at settling above problems, and the main contributions are summarized as follows.

(1) In order to eliminate Storm topology performance bottleneck,

* Corresponding author at: School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China.
E-mail address: chunlin74@aliyun.com (C. Li).

http://dx.doi.org/10.1016/j.jnca.2017.03.007
Received 27 October 2016; Received in revised form 28 February 2017; Accepted 7 March 2017
Available online 09 March 2017

reduce messages processing latency, and improve system throughput, the Storm topology dynamic optimization algorithm based on the theory of constraints is proposed. In addition, for the sake of realizing load balancing among cluster worker nodes and improving Storm performance, the real-time scheduling algorithm based on topology and traffic is put forward.

(2) Our proposed Storm topology dynamic optimization algorithm adopts the idea of using the constraints theory to deal with the assembly line work. In order to eliminate the performance bottlenecks of the topology, the blot's capacity is calculated and the message processing queue congestion degree is analyzed. Large amounts of experiments verify that the performance of the system in average latency and throughput which reflect the QoS level of Storm has great improvement by STDO-TOC algorithm.

(3) In order to solve the problem of inter-node load imbalance, which is another QoS requirement, our proposed real-time scheduling algorithm jointly considers the topology structure, inter-node traffic and worker nodes load balancing to improve Storm scheduler in two steps. The proposed real-time scheduling algorithm optimizes the executors' assignment and slots' allocation in Storm scheduler. Comparison experiment results show the performance of our proposed scheduling algorithm is better than the other algorithms in system throughput and the average latency.

Section 2 reviews related work and Section 3 describes the Storm topology optimization strategy and real-time scheduling Model. Section 4 presents the implementation of the algorithms. Section 5 evaluates the implementation and validates advantage of our approaches. Finally conclusions and future work are given in Section 6.

2. Related work

In this section, we first review previous research work on Storm topology optimization methods, and then discuss those on scheduling strategies of Storm.

2.1. Storm topology optimization

At present, there are some methods researching on Storm topology optimization, which can mainly be divided into two perspectives. The first is to optimize the topologies from topology structure; the second is to optimize the topologies from performance parameters. In Sven (2012), Sven presented a method of constructing Storm topologies based on DSL (Domain Specific Language), which can implement a higher level of abstraction over GPLs (General-purpose Programming Languages) and provide a better way to model the specialized features of a particular domain. In Santurkar et al. (2014), Santurkar et al. proposed a DSL—Stormgen to create an ad-hoc Storm topologies. They analyzed the features of the Storm topology and its components with GPL, and use DSL to optimize the structure of the topology. The practical application of Stormgen was illustrated by a case study that models the topology for the *WordCount* application. In Zappia et al. (2012), Zappia et al. proposed a parameter optimization strategy based on the tradeoff between system throughput and latency, which can achieve the aim of performance optimization by adjusting the task number reasonably. Besides, Zhu et al. (Zhu et al., 2013) proposed a strategy to efficiently achieve topology optimization by setting proper *maxspoutpengding* value and the emission speed of the data stream.

Among all the approaches mentioned above, none of them implemented topology optimization dynamically, no matter through changing topology structures or modifying topology parameters. Traditional topology optimization methods that a running topology has to be terminated manually before users could adjust its structures or modify its parameters are inflexible when they were implemented. Besides, the traditional optimization methods did not take into account the pipeline design rules of the topology.

2.2. Storm scheduling algorithm

Currently, Storm schedules its tasks and places tasks to physical machines by pseudo-random round robin manner, this default scheduling algorithm is too simplistic to efficiently optimize the inter-node traffic and improve resource utilization (Javad and Sanjay, 2015; Peng et al., 2015), moreover, this simple implementation of scheduler makes bad scheduling decision that leads load imbalance problems (Ali et al., 2011).

Scheduling in Storm is usually divided into two granularities, one is the thread scheduling, and the other is the task scheduling. In this paper, our proposed scheduling algorithm considers the thread scheduling. In Aniello et al. (2013), Aniello et al. proposed an offline scheduler and an online scheduler for Storm. The offline scheduler assigns executors to slots based on topology structure. However, this solution is not as effective as expected. Thus, an on-line scheduler, which is referred as traffic-based scheduling algorithm, was proposed by adding the communication traffic to the topology graph as the edge weight. In this algorithm, the optimal traffic allocation policy was selected for executor assignment, which can reduce the inter-node traffic significantly. In Xu et al. (2014), Xu et al. presented a traffic-aware on-line scheduling algorithm. In order to minimize the inter-node traffic and inter-process traffic and reassign its tasks dynamically, this approach accelerates its data processing time by adding an effective traffic-aware scheduling at runtime. This method can do a fine-grained control during worker nodes allocation, which can make system achieve its best performances, meanwhile reduce the number of the used worker nodes. In Chen et al. (2015), Chen et al. proposed a G-Storm parallel system. In this system, a GPU-based scheduling algorithm was used to process the on-line data stream by the GPU of high throughput and massive parallel computing ability. In Cardellini et al. (2015), Cardellini et al. proposed a distributed QoS-aware scheduling algorithm in Storm, where a *QoSMonitor* module is introduced to evaluate the network delay among nodes and monitor the QoS attributes of each worker node, and a *WorkerMonitor* module is also brought in to monitor the worker processes, in order to enhance the system adaptive ability and schedule the cluster resources automatically. In (Peng et al., 2015), Penget et al. proposed a resource-aware scheduling algorithm making two major improvements: assigning the tasks according to the improved breath-first traversal algorithm of the topology graph and allocating the node ports to obey the soft and hard resource constraints as well as the requirement on minimum network distance. This algorithm can significantly improve the resource utilization while minimizing the network latency.

In summary, among the optimization strategies for improving the default scheduling algorithm, some researches consider the topology structure, some methods take into account the inter-node traffic, other strategies aim at improving the node load balancing. However, most of them just consider one aspect of the scheduler's improvement. Our proposed scheduling algorithm could improve the Storm scheduler in all of above aspects, which jointly considers the topology structure, inter-node traffic and worker nodes load balancing.

3. Storm topology optimization and real-time scheduling model

In this section, we first introduce Storm. Then, describe the model of the proposed Storm topology dynamic optimization algorithm based on the theory of constraints (STDO-TOC) and the real-time scheduling algorithm based on topology and traffic (TS-Storm).

3.1. Background: storm overview

Apache Storm (Apache Storm; Jones, 2013; Sax et al., 2013; Li et al., 2016; Smit et al., 2013) is an open source distributed real-time computation system, which can be used for processing unbounded