Resource Estimation in Distributed Data Stream Processing Systems

Minglu Fan[1, a], Yi Liang[1,b], Fei Liu[1,c], Mangmang Yang[1,d] and Haihua Wang[1,e]

[1]Beijing University of Technology, Beijing, China

{[a]fanminglu, [c]S201507079, [d]ymm, [e]wanghaihua}@emails.bjut.edu.cn, [b]yliang@bjut.edu.cn

Keywords: Big Data, real-time stream processing system, resource estimation.

Abstract. Distributed data stream processing systems(DSPS) are widely used in real-time massive data processing scenarios for its characteristics of real-time and high throughout. In the real-world DSPS, the fluctuating arrival rate of the input data leads the consuming computing resource of DSPS to be time-variable. To guarantee the performance of DSPS, the accurate prediction of DSPS's consuming resources is necessary. In this paper, we proposed approaches to make the online prediction of computing resources that DSPS consumes. We monitor the usage of computing resources such as CPU and memory in a DSPS, and use temporal data streams clustering algorithm and linear regression method to make online prediction of CPU resources and memory resources respectively. Our prediction approaches are proved efficient and quickly enough.

Introduction

In a DSPS, the workload at a time is variable since the amount of input data is variable over time, which causes the demand of computing resources is variable. Fig. 1 shows the variation of user visits of twitter in two weeks [1], from which we can see that the user's access behavior is periodical. Based on this, we did an experiment to monitor the usage of CPU and memory resource with the fluctuation arriving data set as input. We conduct the experiment in JStorm [2] and use SequenceTopology as benchmark.

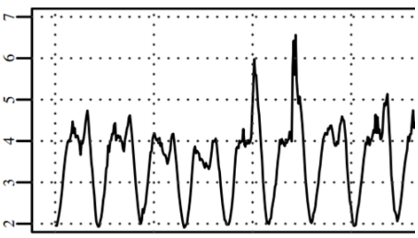

Figure 1: user visits in Twitter

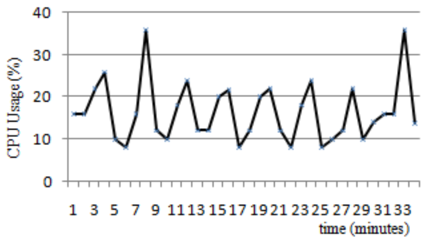

Figure 2: CPU variation curve

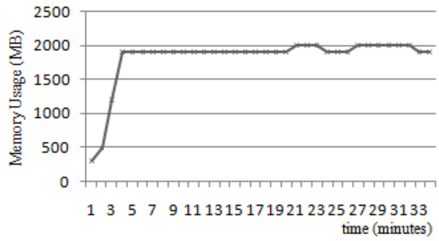

Figure 3: Memory variation curve

Fig. 2 and Fig. 3 shows that different type of resource shows different usage characteristic with dif- ferent data arrival speed. With the increase of input data speed the demand of CPU resource increase while the usage of memory resource varies little with it. However, memory shows strong relationship with running time. To predict the usage of resource, Khoshkbarforoushha [3] made an offline model to predict both CPU and memory, while as we conclude above that the demand of CPU and memory depends on different variate, which make the prediction inaccuracy.

In this paper, we propose two online adaptive algorithms to predict the usage of CPU and memory respectively. Our contributions are as follows:

- We found the different variation characteristics of CPU resource and memory resource with input data speed and time.

- We propose two algorithms to predict the usage CPU and memory resource online.
- We conduct an experiment in real application scenario and make performance evaluation toward these two resource prediction approaches.

CPU Resource Prediction

In our prediction of CPU resource, the current usage amount of CPU should be captured to predict the usage of it in the next window. We define the CPU usage set is $C=\{c_1,c_2,...,c_n\}(n\in(1,t))$, which is a continuously updated real-time stream that a temporal data streams clustering algorithm is necessary. Therefore we propose a clustering algorithm [4] to acquire the usage amount of CPU in the next window by finding the most similar item from historical data set.

We transfer set C into resource variation set $C'=\{\Delta c_2,\Delta c_3,...,\Delta c_n\}$, among which Δc_n is calculated by Equation $\Delta c_n = c_n - c_{n-1}$. The current usage variation of CPU is Δc_n, we cut off C' according to the length l and form data stream sub-sequence $R=\{R_2,R_3,...,R_{n-l}\}$ as the point in clustering space. Each element in sub-sequence is the dimension of point in the clustering, as Eq. 1 shows. l is a variation re- presenting the size of sliding window. The procedure of clustering is defined as Eq. 2:

$$\begin{aligned} &R_1=\{\Delta c_2,\Delta c_3,...,\Delta c_{l+2}\}\\ &R_2=\{\Delta c_3,\Delta c_4,...,\Delta c_{l+3}\}\\ &...\\ &R_{n-l}=\{\Delta c_{n-l},\Delta c_{n\text{-}l+1},...,\Delta c_n\}\end{aligned} \quad . \tag{1}$$

$$d_m=\sqrt{(\Delta c_{n-1}-\Delta c_m)^2+(\Delta c_{n-l+1}-\Delta c_{m+1})^2+...+(\Delta c_n-\Delta c_{l+m})^2}\ . \tag{2}$$

m represent *m-th* element in R. We find the shortest distance set $d_{mim}=\{d_2,d_3,...,d_{n\text{-}l\text{-}1}\}$ by clustering each point in R with the R_{n-1} as central point, then use $R_k=\{\Delta c_k,\Delta c_{k+1},...,\Delta c_{l+k}\}$ to represent the stream of time sequence that the most similar to current time t. The prediction for time t+1 is calculated by Eq.3:

$$c_{t+1}=c_t+\Delta c_{l+k+1}\,. \tag{3}$$

With t increasing, the effect of data located away from time t decrease. Hence to reduce the computation complexity, we adapt slide window to delete some data far away from current time t. Since the central point is ensured during clustering, the time complexity of CPU prediction algorithm is O(n) to find the shortest distance set to central points.

Memory resource prediction

Regression is widely used to make prediction for it's expression of relationship between variations directly. In this paper, we adapt liner regression to predict the usage amount of memory resource. By monitoring the usage amount of memory in every calculate container period, we get the sequence of the usage of memory $M=\{m_1,m_2,...,m_n\}$. m_n is the usage amount of memory for the n-th time window. According to M, the usage amount of memory of $(n+1)$-th time window is represented as Eq. 4:

$$y_{n+1}=a_n+b_nx_n+\varepsilon\ . \tag{4}$$

ε is random error, a_n is the usage amount of nth time window, x_n is n-th time window, b_n is the prediction step length. The time window is updated continuously. Every time we update current usage amount of memory, we make a model of memory. we set x=1, ε=0. Therefore we can get the predicti- on of usage amount of memory as Eq. 5:

$$m_{n+1} = m_n + |m_n - m_{n-1}| . \tag{5}$$

Our prediction of memory does not need the whole sequence of usage amount of memory M, but only current and the latest time window's usage amount. We monitor and make the online prediction. Every time window n we modify current usage amount of memory m_n to get the accurate prediction for next window. Since we record and calculate the usage amount subtraction between current n-th window and (n-1)-th window, there is no loop operation in our algorithm and our algorithm's time complexity is O(1).

Experiments

To prove the accuracy and efficiency of our proposed two resource prediction algorithms of CPU and memory in DSPS, we conduct experiments in our implemented system. Our system is based on JStorm 2.0.4(Java version of Storm [5]), and we adapt SequenceTopology as our Benchmark. There are 5 nodes in our cluster, with one master node for Nimbus and Zookeeper and 4 computational node. We install CentOS 6 in every node, each node is equipped with 4 cores and 8GB RAM, and the inter- net bandwidth is G-bits in Ethernet. We simulate the data arrival rate according to the curve of Poisso- n probability density, and the average data arrival rate is 2100tsp/s.

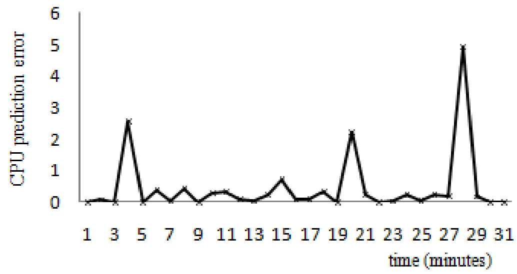

Figure 4: CPU prediction error

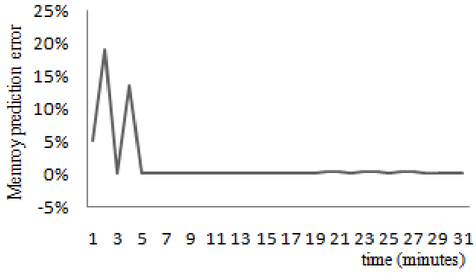

Figure 5: Memory prediction error

The accuracy of our prediction is measured by Mean-Square Error(MSE), as Eq. 6:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{p_i - r_i}{r_i}\right)^2 . \tag{6}$$

n is the size of time window(default as one minute), p_i is the prediction amount of resource needed in i-th time window, r_i is the practical amount of resource used in i-th time window. The result is shown as Fig. 4 and Fig. 5. The MSE for CPU prediction is 38.5%, and more than 74% result's error is within 30%. Besides, the MSE for memory prediction is 0.9%, and 93.5% result's error is within 5%. Therefore, our prediction of CPU and memory resource is proved to be efficient in real-time data stream processing system.

Literature References

In real-time prediction for batch processing application, B. Sharma [6] proposed an approach to get the remaining completion time of a task by calculating current system processing speed and then predict the resource allocation. While in DSPS, the data is arrive endlessly and the speed is unpredictable, hence we cannot get the remaining completion to predict. It means the prediction approach in batch processing system is not suitable for DSPS. In real-time prediction for stream processing system, making offline prediction model, such as machine learning algorithm to learn the parameter as the input of prediction model [3,7]. In their model, they predict both CPU and memory resource in a same model, which make the result of prediction inaccurate. Besides, a large amount of training data is necessary in this model, while it is hard to know the exact usage amount of resource in a DSPS. Other researches [8,9] determine the parallelism degree according to the stay time of data in each component. In contrast, we form stream of time sequence for CPU and memory resource respectively in the form of time window. We achieve the online prediction to estimate CPU usage and memory usage in the next window according to history windows.

Summary

In this paper, we proposed two different prediction models for CPU and memory resource respectively with the consideration of their relationship with data arrival rate. Our models are proved efficient and accurate enough in the real load scenarios. The future work is achieving dynamic resource scheduling according to our prediction of resources.

Acknowledgements

This research is partially sponsored by Natural Science Foundation of China (No. 91546111 and No.61202075), the Research Fund of Beijing Municipal Commission of Education (PXM2015_014204_500221) and the BNSF project (Grant No.4133081).

References

[1] Lohrmann B, Janacik P, Kao O: Elastic Stream Processing with Latency Guarantees, ICDCS(2015).

[2] JStorm on https://github.com/alibaba/jstorm

[3] Khoshkbarforoushha A, Ranjan R, Gaire R, et al. Resource Usage Estimation of Data Stream Processing Workloads in Datacenter Clouds, arXiv preprint arXiv:1501.07020, 2015.

[4] Han J, Kamber M, Pei J. Data mining: concepts and techniques, chapter, 11, Elsevier (2011).

[5] Toshniwal, A., Taneja, S., Shukla, et al: Storm@ twitter, In Proceedings of the 2014 ACM SIGMOD international conference on Management of data.

[6] B. Sharma, R. Prabhakar, S. Lim, et al: Mrorchestrator: A fine-grained resource orchestration framework for mapreduce clusters, International Conference on Cloud Computing, CLOUD(2012) .

[7] Akdere M, Cetintemel U, et al. Learning-based query performance modeling and prediction， International Conference on Data Engineering, ICDE (2012).

[8] Fu, Tom ZJ, et al: DRS: dynamic resource scheduling for real-time analytics over fast streams, International Conference on Distributed Computing Systems, ICDCS (2015).

[9] Li T, Tang J, Xu J: A predictive scheduling framework for fast and distributed stream data processing, IEEE International Conference on Big Data, Big Data(2015).