

Self-Compressive Approach for Distributed System Monitoring

Akshada T Bhondave

Dr D.Y Patil COE

Computer Department, Pune University, India

Santoshkumar Biradar

Assistant Prof. Dr D.Y Patil COE,

Computer Department, Pune University, India

ABSTRACT

Large-Scale distributed hosting infrastructures have become the basic platforms for several real-world production systems. But a challenging task is to achieve both scalability and high precision while monitoring a large number of intra-node attributes that contain information relating to each node and inter-node attribute that denote measurements between different nodes. This paper presents a new distributed monitoring framework Based on video coding techniques of named RBOIC (Replica Based Online Information Compression for Scalable Hosting Infrastructure Monitoring) which uses novel image based approach in which system models snapshots of the monitored distributed system images and applies lightweight online reference block search algorithm to compress monitoring traffic from worker to management node to achieve scalable full coverage monitoring. RBOIC explores the design, implementation of adaptive rood pattern search algorithm to find out optimal or near optimal reference values for each attribute. RBOIC effectively achieve failure-resilient monitoring by restoring monitoring data on replica node. The experimental evaluation of compressive monitoring system has been done using real time monitoring data .The experimental results show that RBOIC can achieve much higher compression ratios with less overhead and adaptive rood pattern search (APRS) achieves better performance by preventing unnecessary intermediate search than diamond search pattern.

Keywords

Distributed System Monitoring, Online Data Compression, Adaptive Rood Search Pattern

1. INTRODUCTION

The last few decades have seen computing becomes a part of our day to day life. Most of the applications nowadays use distributed computing paradigm in which the user accesses resources spread across different locations. As these distributed computing infrastructures continue to grow, how to efficiently manage those complex infrastructures has become a challenging problem.

A hosting infrastructure consists of a large number of distributed worker nodes that collect inter-node and intra-node monitoring data and a management node that provide varied monitoring services. A distributed monitoring system deploys monitoring agents on distributed worker nodes to continuously collect various attributes and periodically report sampled attribute values to management nodes. As these computing infrastructures continuously growing, efficiently manage such large scale distributed systems has become a challenging problem.

Distributed monitoring system must be capable of tracking system information such as disk usage, CPU memory and

bandwidth. Monitoring system need to know the various attribute values on a per node basis as well as the link information between various nodes. Two important factors, namely scalability and accuracy need to be addressed while designing the large-scale distributed information management service. To obtain complete and fine-grained information regarding all hosts and network connection within the hosting infrastructure is necessary to achieve efficiency and accuracy. However, it is a challenging task to deploy fine-grained monitoring for large-scale hosting infrastructures due to scalability overhead [12], [13],[14].

Specifically, this paper makes the following contributions:

- Replica Based Online Information Compression (RBOIC) framework, to enable scalable full-coverage, fine-grained monitoring for Scalable Hosting Infrastructure.
- Introduce a fast and efficient reference block search algorithm that uses an Adaptive Rood Pattern search to find the optimal or near optimal reference block with low overhead.
- Present failure-resilient RBOIC, which can tolerate management node failure by restoring monitoring data on replica node which work as management node after management node failure.

The paper continues to exist as follows: Section 2 presents related work. Section 3 deals with an overview of the distributed monitoring model and problem formulation, and Section 4 is about design details of proposed monitoring system, RBOIC .Section 5 present experimental evaluations. Finally the paper concludes in Section 6.

2. RELATED WORK

Distributed monitoring is critical for large scale infrastructure. CoMon [1] have proven extremely useful for their user communities. However, for practical purpose, system is statically configured with long update interval (e.g., five minutes for the CoMon infrastructure). Previous work (e.g. Astrolabe [3], SDIMS [4]) has proposed to leverage hierarchical or decentralized architectures to achieve scalable distributed monitoring. Other research work (e.g., Info-Eye [6], STAR [7]) has proposed to trade off information coverage or information precision for lower monitoring cost. Different from the above work, Info-Track [8] focus on exploring correlation patterns to achieve self-compressing continuous information tracking, which can be used to not only answer various information queries but also extract important system patterns to guide system management decisions. Info-Track explores both spatial and temporal correlations to compress the live monitoring streams in a large scale distributed system .however, temporal correlation schemes has limited compression ratio while the spatial correlation scheme is often

costly due to the expensive clustering operation. OLIC [11] present an initial design and implementation of an image-based online compressive monitoring system to reducing distributed monitoring cost. OLIC uses lightweight, image-based reference block search algorithms to enable a broader search range so that the compression ratio can significantly improved without imposing too much overhead. RCM[12] extends OLIC by adding failure resilience support to achieve robust monitoring under host failures. In comparison, our work addresses an orthogonal problem of reducing the collection cost of known values.

3. SYSTEM OVERVIEW

3.1 System Model

Distributed hosting infrastructure consists of N worker nodes. Monitoring agents on each of the worker node report their inter-node and intra-node attribute values to the management node using a sampling rate of 10 seconds. The intra-node attributes contain information relating to each node and inter-node attributes that denote measurement between different nodes. On each worker node, the monitoring agent periodically samples each intra-node attribute to form a time series $\{a_{i,k}^1, \dots, a_{i,k}^t, \dots, a_{i,k}^{t+m}\}$, where a_i^t denotes the sampled value for the intra-node attribute a_k collected on node v_i at time t . Similarly, the monitoring agent periodically samples each inter-node attribute to form a time series $\{d_{i,j,k}^1, \dots, d_{i,j,k}^t, \dots, d_{i,j,k}^{t+m}\}$ where $d_{i,j,k}^t$ denotes the inter-node attribute d_k between nodes i and j at time t .

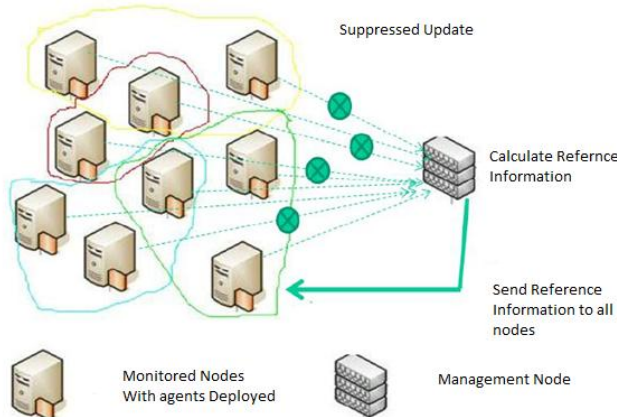


Fig. 1 Distributed Monitoring System

3.2 Problem Formulation

The goal of the RBOIC system is to achieve fine-grained monitoring with low cost by suppressing the update of the attribute values from a worker node to the management node at time t if the management node can infer the worker nodes attribute value using a known reference value. a_i denote real attribute value and a_i' denote its reference value. System allows the user to specify an error bound e_i ($e_i \geq 0$).

If $\frac{|a_i - a_i'|}{a_i} \leq e_i$, RBOIC can suppress the update of this attribute if its reference value within the error bound e_i .

Compression ratio (CR) is defined as follows:

$$CR = \frac{us}{uorig} \quad (0 \leq CR \leq 1), \quad (1)$$

where U_s is the number of attribute values suppressed by system in bytes per second and u_{orig} is the number of original attribute propagated without using any compression in bytes per second. More monitoring cost reduction can be obtained when compression ratio is large [11].

3.3 Mathematical Model

Mathematical model for system is as follow:

▪ **System:**

Let S be a system that describes compressive distributed monitoring system

$$S = \{I, O, A\}$$

Where

I = Input consist of $\{a_{i,k}^t, d_{i,j,k}^t, s, e_i, b\}$ and attribute threshold.

O = Output consist of generated optimal reference values and compression ratio.

$$CR = \frac{us}{uorig}$$

A = Adaptive Rood Search Pattern Algorithm, find Optimal reference values for each Attribute.

$$ARP(a_{i,k}^t, d_{i,j,k}^t) \rightarrow a_i'$$

s = Sampling Rate

CR = Compression Ratio

b = Block Size

N = Number of Worker Nodes $\{v_1, v_2, v_3, \dots, v_n\}$

e_i = Error bound for attribute a_i

a_i' = Reference value of attribute a_i

$a_{i,k}^t$ = intra-node attribute a_k collected on node v_i at time t

$d_{i,j,k}^t$ = inter-node attribute d_k between nodes i & j at time t

4. SYSTEM DESIGN

ROBOIC perform online compression over distributed monitoring data as well replicate monitoring data on replica server to achieve failure resilient online compression, which is illustrated by figure 2. RBOIC runs in 2 phases: the training phase and the compressive phase.

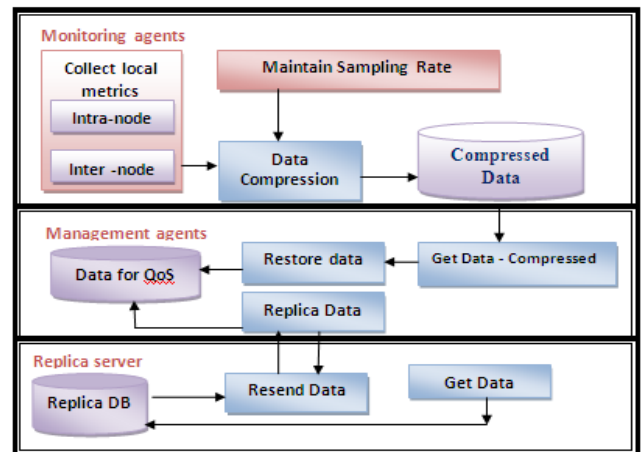


Fig. 2 RBOIC Distributed Monitoring Architecture.

▪ **Training phase**

1. Monitoring agents on all worker nodes starts sending intra-node attribute data to management node.

$$\text{Send} (\{a_{i,k}^1, \dots, a_{i,k}^t, \dots, a_{i,k}^{t+m}\})$$

For monitoring an intra-node attribute a_k the system image at time t is an $M \times K$ matrix ($M.K=N$)

$$\begin{bmatrix} a_{i,1}^t & \dots & a_{i,k}^t \\ a_{i,k+1}^t & \dots & a_{i,2k}^t \\ \vdots & \vdots & \vdots \\ a_{i,N-k}^t & \dots & a_{i,N}^t \end{bmatrix} \begin{bmatrix} a_{i,1}^{t+1} & \dots & a_{i,k}^{t+1} \\ a_{i,k+1}^{t+1} & \dots & a_{i,2k}^{t+1} \\ \vdots & \vdots & \vdots \\ a_{i,N-k}^{t+1} & \dots & a_{i,N}^{t+1} \end{bmatrix}$$

Image at time t Image at time $t+1$

Fig. 3 System image sequence for an inter-node attribute of n nodes.

2. Monitoring agents on all worker nodes starts sending inter-node attribute data to management node.

$$\text{Send} (\{d_{i,j,k}^1, \dots, d_{i,j,k}^t, \dots, d_{i,j,k}^{t+m}\})$$

For monitoring an inter-node attribute d_k , the system image at time t is an $N \times N$ matrix

$$\begin{bmatrix} 0 & \dots & d_{1,j}^t & \dots & d_{1,N}^t \\ \vdots & \backslash & \vdots & \backslash & \vdots \\ d_{i,1}^t & \dots & 0 & \dots & d_{i,N}^t \\ \vdots & \backslash & \vdots & \backslash & \vdots \\ d_{N,1}^t & \dots & d_{N,j}^t & \dots & 0 \end{bmatrix} \begin{bmatrix} 0 & \dots & d_{1,j}^{t+1} & \dots & d_{1,N}^{t+1} \\ \vdots & \backslash & \vdots & \backslash & \vdots \\ d_{i,1}^{t+1} & \dots & 0 & \dots & d_{i,N}^{t+1} \\ \vdots & \backslash & \vdots & \backslash & \vdots \\ d_{N,1}^{t+1} & \dots & d_{N,j}^{t+1} & \dots & 0 \end{bmatrix}$$

Image at time t Image at time $t+1$

Fig.4 System image sequence for an intra-node attribute of n nodes.

3. RBOIC applies online reference block search algorithm that uses an adaptive rood pattern search algorithm to find the optimal or near-optimal reference blocks for each inter-node and intra-node attribute.

$$\text{ARP} (A, D) \rightarrow a'_i$$

4. Then Management node stores optimal reference block information locally and also sends corresponding reference block information to different worker nodes as well as replica server.

After Training phase is done, RBOIC starts to perform compression.

▪ **Compression Phase**

1. Receive reference block information from the Management node.

$$\text{Receive} (a'_i)$$

2. If $\frac{|ai-a'_i|}{ai} \leq e_i$

Worker node suppress update of attribute

Else

$$\text{Send} (a_{i,k}^t, d_{i,j,k}^t)$$

4.1 Online Reference Block Search

One important step in compression phase is to find an optimal or near optimal reference block for each inter-node and intra-node attribute to maximize compression ratio. If current attribute value can be inferred from the corresponding attribute reference value, the monitoring agent on worker node does not need to report the current attribute value to the management node.

RBOIC uses a novel and simple reference block search algorithm uses adaptive rood pattern search (ARPS) pattern, illustrated by figure 5. The first pattern, called adaptive rood pattern (ARP) for the initial search, which consists of four blocks surrounding the center block and predicted block to increase probability of finding optimal reference block. The second pattern, called, unit-size rood pattern (URP) for the redefined local search, consists of four search points surrounding the center block.

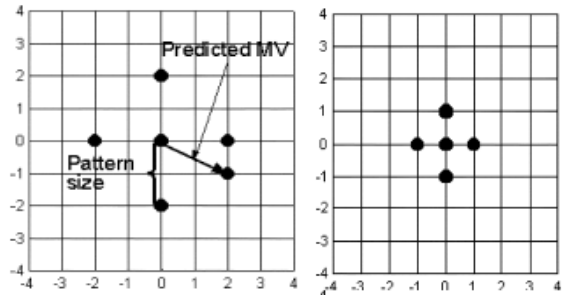


Fig.5 Adaptive Rood Pattern (ARP).

The shape of rood pattern is symmetrical and its size shows the distance between any vertex point and the center point. The size of ARP, Γ , is equal to the length of the predicted block.

$$\Gamma = \text{Max} \{ |MV_{\text{predicted}}(x)|, |MV_{\text{predicted}}(y)| \} \quad (2)$$

For each block, the initial search is performed only once at the beginning in order to find a good starting point for the follow-up refined local search to limit unnecessary intermediate search. For the initial search stage, an adaptive rood pattern (ARP) is used. In the refined local search stage, a unit-size rood pattern (URP) is exploited repeatedly until the final optimal block is found [10].

5. RESULTS AND ANALYSIS

The compressive monitoring system has been deployed on collection of nodes. Monitoring system has an interface showing the real-time monitoring of the system. Compressive monitoring system is divided in three modules, the monitoring agent, the management node and the replica module. The monitoring agent has an interface showing inter-node and intra-node attribute data collected by monitoring agent on each worker node at the sampling rate of 10 second and shows reference attribute values for an each attribute received from management node. Management node has an interface showing system image of an inter-node and intra-node-attribute for current sampling interval, no of attribute received and compression ratio. Replica node has an interface showing connection status with primary server and status of monitoring data restoration. The experimental evaluation of compressive

monitoring system has been done using real system monitoring data.

5.1 System Implementation

The prototype of the proposed system is implemented using java. A monitoring agent is deployed on each of the worker node and the management node program is run on a dedicated server. The server machine has configuration of Intel Core Duo CPU 2.4 GHz with 4GB of RAM. The number of reference images l set to be one. The block size b is set to be four. Each of the monitoring agent collect various intra-node attributes like CPU usage, free memory, total memory and throughput .for the inter-node attribute network delay, each monitoring agent pings other nodes in the system and collects information. The collected attributes are associated with the IP address to identify the information source and the management node aggregates all the information reported by the various agents.

Figure 6 shows number of attribute received by management node at every sampling interval of 10 seconds when number of worker nodes equals to two. CPU and throughput threshold is set to two, free memory and total memory threshold equal to ten and network delay threshold equal to one.

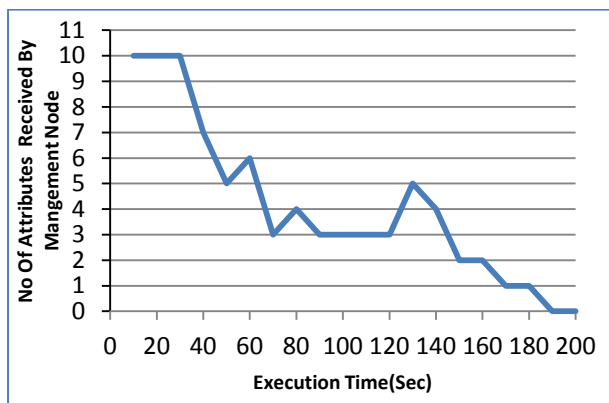


Fig .6 Number of attribute received by management node at every sampling interval of 10 second

Figure 7 show the continuously sampled compression ratio achieved by our monitoring system at every sampling interval of 10 seconds.

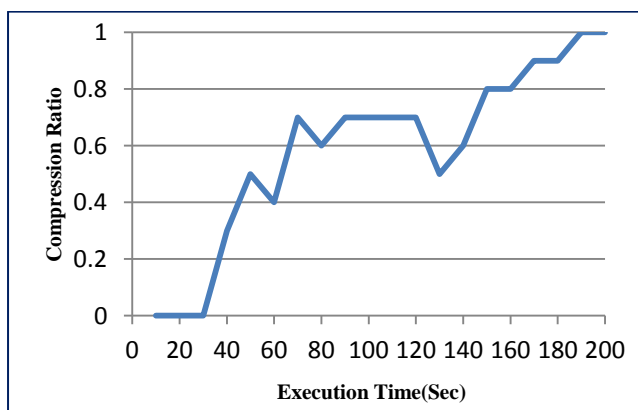


Fig .7 Compression ratio achieved at sampling interval of 10 second.

RBOIC effectively achieve failure-resilient monitoring by restoring monitoring data on replica node. Initially all worker

send monitoring data to management node after management node failure, replica node performs role of management node and compressive monitoring system can still achieve compression under management node failure

Figure 8 shows number of attributed received from worker node to management and replica node under management node failure.

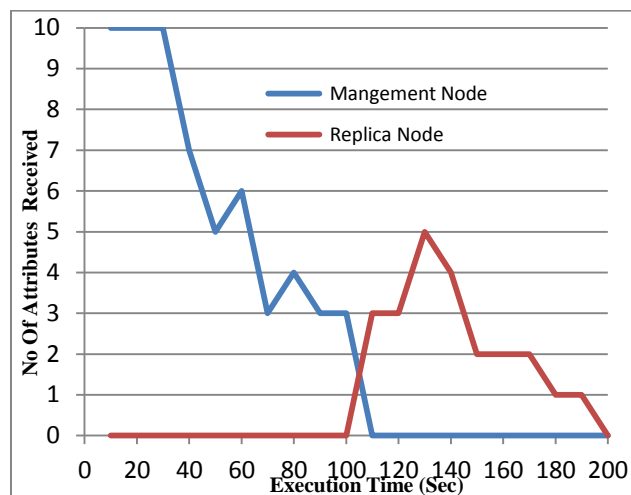


Fig. 8 Number of attributed received from worker node to management and replica node under management node failure

5.2 Compression Comparison with Offline Compression Scheme

In Distributed system source data are distributed on different hosts that can experience transient or persistent failures from time to time. Offline compression schemes that can only be applied after the data have been reported to the management node, RBOIC performs online compression over live monitoring data streams during monitoring runtime. Thus, this image based approach can reduce end-system resource and network bandwidth consumption on both worker nodes and management node.

6. CONCLUSION

In this paper, we have presented RBOIC, a novel image-based resilient self-compressive monitoring system for large-scale hosting infrastructure .Goal of compressive monitoring system is to alleviate the bottleneck on the management node by reducing the monitoring traffic distributed worker nodes to the management node. To achieve this goal, RBOIC models snapshots of the monitoring attributes as a sequence of system images and applies lightweight online reference block search algorithm to compress distributed monitoring. RBOIC achieves failure resilient compression by restoring monitoring data on replica node. Compared to previous Diamond search pattern algorithm, adaptive rood pattern search algorithm has a broader reference block search range than the diamond search algorithm with a smaller computational overhead. Compressive monitoring system, RBOIC is implemented and conducted experiments using a real system monitoring data. Prototype implementation shows that RBOIC is efficient, which can achieve higher compression ratio with lower overhead.

In future, our work will explores more options for failure resilient monitoring by increasing number of management

node in monitoring infrastructure to prevent single point of failure and it will reduce bottleneck on single management node. Currently implemented monitoring system which collect monitoring data for five attribute, in future we will try to explore our system for collecting monitoring data for more number of dynamic attributes. For inter-node attribute monitoring, each worker node pings other nodes in monitoring environment, in order to reduce network traffic we can use Piggybacking method through which attribute data can send with ping packets in order to reduce network traffic.

7. REFERENCES

- [1] "CoMon. <http://comon.cs.princeton.edu/>.
- [2] "IBM Tivoli Monitoring Software "http://www.01.ibm.com/software/Tivoli/2012.
- [3] R. Van Renesse, K.P. Birman, and W. Vogels, "Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining," ACM Trans. Computing systems, vol. 21, no. 2, pp. 164-206, 2003.
- [4] P. Yalagandula and M. Dahlin, "A Scalable Distributed Information Management System," Proc. ACM SIGCOMM, Aug. 2004.
- [5] Matthew L. Massie, Brent N. Chun, and David E. Culler, "The Ganga Distributed Monitoring System: Design, Implementation, and Experience", Parallel Computing, 30(7):817 – 840, 2004.
- [6] J. Liang, X. Gu, and K. Nahrstedt, "Self-Configuring Information Management for Large-Scale Service Overlays," in Proc. IEEEINFOCOM, 2007.
- [7] N. Jain, D. Kit, P. Mahajan, P. Yalagandula, M. Dahlin, and Y. Zhang, "STAR: Self-Tuning Aggregation for Scalable Monitoring", Proc. Int'l Conf Very Large Data Bases (VLDB), 2007.
- [8] Y. Zhao, Y. Tan, Z. Gong, X. Gu, and M. Wamboldt, "Self- Correlating Predictive Information Tracking for Large-Scale Production Systems", Proc. Int'l Conf. Autonomic Computing (ICAC), 2009
- [9] S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block- Matching Motion Estimation" IEEE Trans. Image Processing, vol. 9, no. 2, pp.287-290, Feb.2000.
- [10] Yao Nie , Kai-Kung Ma , " Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation" ,IEEE Transaction On Image Processing ,Vol. 11,no.12,December.2002
- [11] Y. Tan, X. Gu, and V. Venkatesh, "OLIC: Online Information Compression for Scalable Hosting Infrastructure Monitoring," Proc. 19th Int'l Workshop Quality of Service (IWQoS), 2011.
- [12] Y. Tan, V. Venkatesh, and X. Gu, "Resilient Self Compressive Monitoring for Large-Scale Hosting Infrastructures," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 3, pp. 576-.586, March 2013, doi:10.1109/TPDS.2012.167
- [13] V.Venkatesh,"Video Coding Approach to Compressive Distributed System, Ph.D. thesis, Department of Computer Science, North Carolina state University, March 2010.
- [14] Y. Tan, "Online Performance Anomaly Prediction and Prevention for Complex dummy Distributed systems. Ph.D. thesis, Department of Computer science, North Carolina state University, 2012.
- [15] T. Sikora. Trends and Perspectives in Image and Video Coding. Proceedings of IEEE,93(1):6–17, January 2005.