ISSN:0975-9646

A Survey On: Distributed Monitoring for Scalable Hosting Infrastructures

Akshada Bhondave[1], Santoshkumar Biradar [2]

[1,2]Department of computer Engineering, University of Pune,
Dr D.Y.Patil Collage of Engineering
Pune, Maharashtra, India

***Abstract*-Large scale distributed system has become the fundamental platforms for many real-world production systems. However, Automatic and continuous management of such distributed computing infrastructures is a challenging task since solution has to achieve both scalability and high precision while monitoring a large number of intra-node and inter-node attributes (e.g., CPU usage, free memory, free disk, inter-node network delay). In this paper we focus on specific domain that is fine-grained, scalable, failure resilient distributed system monitoring framework based on video coding techniques. This approach models the distributed system metrics as an image and uses both intra-image and inter-image encoding to compress monitoring traffic.**

In this article, we highlight the importance of fine grain monitoring and scalability challenge of fine-grained distributed system monitoring .We identify the state-of-the-art of scalability issues related to fine grained monitoring and discuss the various solutions proposed that are Centralized, Decentralize Monitoring, and Compression and correlation Based Approach.

***Keywords:* Distributed System Monitoring, Online Data Compression, Video Compression, Block Matching Algorithm.**

I. 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. Distributed computing Infrastructures like Planet Lab and world community Grid are serving as large-scale resource pools for scientific research. The world community grid depends upon individuals collectively contributing their unused computer cycles. Efficiently managing such large scale distributed system has become a challenging problem.

Distributed monitoring system must be capable of tracking system information such as disk usage, CPU memory and bandwidth dynamically. Monitoring system need to know not only 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. On one hand, quality-of-service (QoS) sensitive applications require accurate up-to-date distributed system information. On the other hand, a large-scale distributed system can include tens of thousands of geographically distributed nodes. The actual monitoring traffic for such a system would be in the range of 6.1Mb/sec at a sampling rate of ten seconds and 400 metrics collected per node. However, it is a challenging task to deploy fine-grained monitoring for large-scale hosting infrastructures due to scalability overhead.

Obtain complete and fine-grained information regarding all hosts and network connections within the hosting infrastructure is necessary to achieve efficiency and accuracy. A production hosting infrastructure often comprises Thousands of physical hosts and many more virtual machines (VMs), each of which can be associated with hundreds of dynamic metrics Thus, without reducing the monitoring traffic to the management node; it is impractical to apply fine-grained monitoring to large scale hosting infrastructures [1].

Thus in this article we study how the Resilient, self-Compressive Monitoring system (RCM) for large-scale hosting infrastructures monitoring perform better than exiting approaches in terms of scalable resilient monitoring. The System takes a novel image-based approach to alleviate the bottleneck on the management node by reducing monitoring traffic towards it.

II. Solutions To Scalable Monitoring

A. Centralized Monitoring

These systems perform data monitoring by aggregating information from a variety of sources and presenting it to system operators through some graphical user interfaces [4].

B .Decentralize Monitoring

Decentralize architecture such as hierarchical aggregation or peer-to-peer structure is employed to distribute monitoring workload.

Instead of expose all information to all nodes, hierarchical aggregation allows a node to access detailed views of nearby information and summary views of global information [6,7,8].

C. Correlation Based Approach

Monitoring data typically have some redundancies that can be exploited to leverage correlation patterns to reduce monitoring cost. Temporal correlation is explored within one node (Self-similarity) and spatial correlation (group-similarity) among distributed nodes to suppress unnecessary remote information update [11].

D. Offline /Online Compression Based Approach

Offline compression schemes that can only be applied after the monitoring data have been reported to the management node whereas online compression schemes that can apply over live monitoring data streams during monitoring runtime [3].

TABLE I-COMPARISON OF SYSTEM MONITORING TOOLS BASED ON SCALABILITY, SECURITY, AND ROBUSTNESS.

Author /Website	System Monitoring Tools.	Monitoring Architecture/ Approach	Features	Results		
http://www 01.ibm.com/software/Tivoli/2012	IBM Tivoli	Centralize	**Security**: Through Tivoli Access Manage, Privacy & Risk Manager, IBM Directory Server & Integrator **Availability:** Through Tivoli Enterprise Console , Storage Mgr Family, Tivoli System Automation Optimization,	Less Scalable		
http://comon.cs.princeton.edu/.2012	CoMon	Centralize	Monitoring system for Planet Lab			
Renesse, Birman and Vogels (2003)	Astrolabe	Peer to peer	**Scalability:** Through Zone hierarchy, hierarchical attribute aggregation **Flexibility:** Through mobile code, Dynamically installed aggregation functions **Robustness:** via a gossip-based peer-to-peer protocol, Self-management and recovery **Security:** through Certificates ,Integrity and write access control	Astrolabe could scale to thousands and perhaps millions of nodes, with information propagation Delays in the tens of seconds.		
Yalagandula and Dahlin (2004)	SDIMS	Hierarchical	**Scalability:** with respect to both nodes and attributes through a new aggregation abstraction that helps leverage DHT's internal trees for aggregation. **Flexibility**: Through a simple API that lets applications control propagation of reads and writes. **Administrative Isolation :** Through simple augmentations of current DHT algorithms **Robustness:** to node and network reconfigurations through lazy re-aggregation, on-demand re-aggregation, and tuneable spatial replication.	Node Stress i.e. Amt. of incoming and outgoing information is less than Astrolabe		
Massie, Chun and Culler (2004)	Ganglia	Hierarchical	Monitoring compute clusters **Scalability:** both as a function of cluster size and the number of clusters being federated. Tree of point-to point connection to federate cluster and aggregate their state. Multicast-based Listen/Announce protocol to monitor state within clusters.	Scales on clusters of up to 2000 nodes and federations of up to 42 sites.		
Liang, Gu and Nahrsteadt (2007)	InfoEye	Automatic Self Configuration	**Scalability:** Through answering information queries With minimum monitoring overhead by utilizing the statistical patterns of application needs and system conditions to intelligently configure the information management system.	Much lower management overhead than static solutions.		
Zhao, Tan and Xu (2009)	InfoTrack	Spatial & Temporal Correlation	**Scalability:** Through lightweight temporal and spatial correlation discovery methods to minimize continuous monitoring cost. Applied to any centralized or decentralized monitoring architecture.	**Attribute**	**CR %**	**Error bound**
				Stable	95%	0.01
				Dynamic	50%	0.01
					90%	0.05
Tan, Gu ,and Venkatesh (2011)	OLIC	Image Based Approach	**Scalability:** Through novel image based approach to achieve scalable fine-grained hosting infrastructure monitoring based on video coding approach.	**Attribute**	**CR %**	**Error bound**
				Stable	70%	0
					88%	0.01
				Dynamic	35%	0.01
Tan, Gu ,and Venkatesh (2013)	RCM	Image Based Approach	RCM extends OLIC by adding failure resilience support to achieve robust monitoring under host failures.	**Failure Rate**	**q**	**CR %**
				10%	3	20-45
				30%	3	21-47
J. Ziv and A. Lempel (1978)	Gzip	Offline Compression Based Approach	Gzip is a lossless compression scheme which perform compression after the monitoring data have been sent to the management node	RCM can achieve Similar compression ratios as Gzip for the Planet Lab inter-node delay data sets		

III. SYSTEM MONITORING TOOLS CLASSIFICATION

There are various system monitoring tools are available to solve scalability challenge of fine-grained monitoring that are Centralized and Decentralize Monitoring, Correlation and Online/Offline Compression Based Approach . As shown in Table1 named "Comparison of system monitoring tools based on scalability, security, and Robustness " compare the performance of system monitoring tools on the basis of different features.

Following Result is observed from Table 1

- Centralized systems do not scale to the required number of flows, while pure peer-to-peer architectures cannot provide a global view of the system state.
- Decentralized, per-data-center, hierarchical monitors are limited to computing average measures spanning over several nodes.
- Hierarchical monitors overcome some of the limitations of centralized solutions at the cost of limited system manageability but the root node in the system may present a single point failure similar to the centralized model.
- Astrolabe is highly scalable and resilient. Its manageability is a complex task since it generates a lot of network traffic.
- InfoEye achieve minimum monitoring overhead at the cost of losing some information coverage.
- Temporal correlation has limited compression power while discovering spatial correlation is often costly due to the expensive clustering operation.

IV. COMPRESSION ALGORITHMS

A. Temporal Correlation Algorithm

Temporal correlation algorithm that suppresses the monitoring updates if the last attribute value can be used to predict the current attribute value within the error bound.

B. Spatial Correlation Algorithm

Spatial Correlation Algorithm that uses the k- medoids clustering algorithm to group all monitored nodes into different groups. One node in the group (i.e., the cluster head, usually the medoid of each cluster) is elected as the representative node. Other cluster members do not need to send their updates if the difference between their values and the cluster head is within the error bound.

C. Temporal+Spatial Correlation Algorithm

Temporal+Spatial correlation algorithm developed by the InfoTrack system that leverages both temporal and spatial correlations among attribute values to suppress distributed monitoring Traffic [11].

D. Neighbor Search Algorithm

Neighbor search algorithm that performs a similar reference block search as RCM but its search range is limited to eight immediate neighbor blocks (i.e., up left, up, upright, right, downright, down, down left, and left) in the reference image.

E. Diamond Search Algorithm

The diamond search algorithm utilizes two search patterns. One is the large diamond search pattern (LDSP) and other is the small diamond search pattern (SDSP).The LDSP searches nine blocks out of which eight surround the centre block to form diamond .SDSP searches five blocks Out of which four forms a smaller diamond around the centre block [12].

The experimental evaluation of the monitoring system has been done using real system monitoring data. As shown in Table 2 named "Statistics of the Monitoring Traces" shows the characteristics of different traces used to evaluate the Monitoring system.

Data size is the total file size of each monitoring data trace. The coefficient of variation (CV) is defined as the ratio of the standard deviation σ to the mean μ (i.e.,$CV = \sigma/\mu$ .The data set with the smaller CV will have lower variations than other data sets. Length of each compression phase set to 300 system images, training rounds R and reference image both are set to 3 and block size set to be 4[2].

TABLE II-STATISTICS OF THE MONITORING TRACES

Data Trace	Attribute	Description	System Image Dimension	Total Data Size	Coefficient of variation
VCL IP Statistics	Datagrams / Sec	**Intra-node attribute**: 400 nodes, Mean: 38.69, Standard deviation: 82.58, Sampling interval: 5 minutes	20*20	18MB	1.53
Planet Lab Memory (MB)	Free Memory	**Intra-node attribute**: 400 nodes, Mean: 107, Standard deviation: 42, Sampling interval: 10 seconds, Total data size: 744MB	20*20	44MB	0.39
VCL NT Processor Trace	(DPC Queued/sec)	**Intra-node attribute:** 400 nodes, Mean: 45.4, Standard deviation: 58.23, Sampling interval: 5 minutes.	20*20	36MB	1.33
Google Cluster	CPU Usage	**Intra-node attribute:** 1296 nodes, Mean: 0.039, Standard deviation: 0.026, Sampling interval: 5 minutes.	36*36	86MB	0.68
Traffic Matrices (Kbps)	Traffic	**Inter-node attribute:** 23 nodes, Mean: 17040, Standard deviation: 52578, Sampling interval: 15 minutes.	23*23	126 MB	0.41
Planet Lab Delay(ms)	Inter-node Delay	**Inter-node attribute:** 464 nodes, Mean: 241.8, Standard deviation: 58.9, Sampling interval: 10 seconds.	464*464	92GB	0.22

As shown in Table 3 named "Comparison of compression algorithms based on compression ratio under no host failure" [2] compared the performance of RCM, Neighbor search, Temporal, Temporal+Spatial algorithm on the basis of compression ratio.

Following result is observed from Table 3:

- RCM can achieve average compression ratio of 28.2-48.8 percent under a range of tight error bounds (0.01-0.1) for all these data sets.
- RCM can improve the compression ratio by 24, 18, and 46 percent on average compared to the Neighbor search algorithm, the Temproal+Spatial correlation algorithm, and the temporal correlation algorithm, respectively.
- RCM can achieve more than 200 percent higher compression ratio over the temporal correlation scheme under tight error bounds (e.g., 0.01).
- The neighbor search scheme is slightly worse than the Temporal+Spatial correlation scheme because it has a smaller reference block search range than the Temporal+Spatial correlation scheme. However, the neighbor search scheme still has the advantage over the Temporal+Spatial scheme since its computational overhead is much smaller.
- RCM consistently outperforms the other alternative schemes for all the data sets because of its broader Search range of diamond search algorithm than temporal and spatial correlation schemes to search the best reference blocks.

As shown in Table 4 named "Comparison of compression algorithms based on compression ratio under host failure" [2] compared the performance of RCM under certain percentage of host failures on the basis of compression ratio.

Following result is observed from Table 4:

- RCM can achieve better compression performance as the number of backup reference blocks increases.
- RCM without any backup reference block only has 10-20 percent compression ratio loss compared to the non failure case.
- RCM with one backup reference block can achieve similar compression ratio to the non failure case.
- RCM with more than one backup reference block can achieve even higher compression ratio than the non failure case.

TABLE III- COMPARISON OF COMPRESSION ALGORITHMS BASED ON COMPRESSION RATIO UNDER NO HOST FAILURE

Sr. No	Dataset	Error Bound Between 0.01 to 0.10			
		OLIC	**Neighbor Search**	**Temporal**	**Temporal+ Spatial**
1	VCL NT Processor Trace	21-30	8-20	6-12	10-22
2	VCL IP Statistics	18-23	5-14	3-8	7-15
3	Planet Lab Memory (MB)	30-70	20-68	18-65	20-70
4	Traffic Matrices (Kbps)	15-33	5-29	4-25	5-29
5	Planet Lab Delay(Ms)	88-93	88-92	86-91	-
6	Planet Lab CPU Load	21-80	18-80	16-80	18-81
7	Google Cluster CPU Trace	13-18	7-13	5-8	7-13

TABLE IV-COMPARISON OF COMPRESSION ALGORITHMS BASED ON COMPRESSION RATIO UNDER HOST FAILURE

Sr. No	Dataset	Failure Rate	Number of Backup reference blocks (q)	CR %
1	Google Cluster CPU Trace	10%	q=3	14-20
			q=2	13.5-19
			q=1	13-18
			q=0	12-17
			No Failure	13-18
		30%	q=3	15-21
			q=2	14-19
			q=1	12-17
			q=0	10.1-14.2
			No Failure	12-17

V. CONCLUSION

In this survey article, we try to scrutinize the scalability challenge of fine-grained monitoring in large scale hosting infrastructures. Because of a production hosting infrastructure often comprises thousands of physical hosts and many more virtual machines (VMs), each of which can be associated with hundreds of dynamic metrics Thus, without reducing the monitoring traffic to the management node, it is impractical to apply fine-grained monitoring to large scale hosting infrastructures.

We briefly introduce the various solutions to scalability challenge of fine-grained monitoring. From Table 1 we observe that centralize monitoring is less scalable whereas Hierarchical monitoring results in of limited system Manageability. Exploring temporal correlation has limited compression power while discovering spatial correlation is often costly due to the expensive clustering operation In contrast, RCM uses lightweight, image-based reference block search algorithms to enable a broader search range so that the compression ratio can be significantly improved without imposing too much overhead .Table 3 and Table 4 compared the performance of RCM, Neighbor Search, Temporal , Temporal+Spatial algorithm on the basis of compression ratio under no host failure and under host failure. We observe that RCM can achieve average compression ratio of 28.2-48.8 percent under a range of tight error bounds (0.01-0.1) for all these data sets. From table 4 we observed that RCM can achieve better compression performance as the number of backup reference blocks increases.

In future by applying a boarder search range by using the adaptive rood pattern search rather than the existing reference block search algorithms follows dual-diamond search patterns, proposed system can achieve higher compression performance by preventing unnecessary intermediate search. Furthermore, failure of host can be detected by using inter-node monitoring and management node failure can be handled using replica server which act as primary server when management node get fail.

REFERENCES

[1] V.Venkatesh,*"Video Coding Approach to Compressive Distributed System"*, Ph.D.thesis, Department of ComputerScience, North Carolina state University, March 2010.

[2] 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

[3] 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

[4] CoMon," http://comon.cs.princeton.edu/.2012.

[5] IBM Tivoli Monitoring Software http://www01.ibm.com /software/Tivoli/2012.

[6] P.Yalagandula and M.dahlin ,*"A Scalable Distributed Information Management System,"* Proc. ACM SIGCOMM Aug. 2004.

[7] Matthew L. Massie, Brent N. Chun, and David E.Culler, *"The ganglia Distributed monitoring system: design, Implementation and Experience"*, Parallel Computing, 30(7):817 – 840, 2004.

[8] 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.

[9] N. Jain, D. Kit, P. Mahajan, P. Yalagandula, M. Dahlin, and Y. Zhang, *"STAR: Self-Tuning Aggregation for Scalable Monitoring"*,Proc. Int'l ConfVery Large Data Bases (VLDB), 2007.

[10] J. Liang, X. Gu, and K. Nahrstedt, *"Self-Configuring information management for Large-Scale Service Overlays,"* in Proc. IEEEINFOCOM, 2007.

[11] 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.

[12] Zhu and K.K. Ma, *"A New Diamond Search Algorithm for Fast Block- Matching Motion Estimation"* IEEE Transaction On Image Processing, vol. 9, no. 2, pp.287-290, Feb.2000.

[13] 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