International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-06, Issue-02, May 2020

Way to Industry 4.0: Integrated IT monitoring with auto-identification of root cause for heterogeneous environment

[1]Brajesh Kumar, [2]Shakti Manchanda
[1]Senior Engineer, [2]Deputy Manager, [1,2]Bharat Electronics Limited, India.
[1]brajeshkumar@bel.co.in, [2]shaktimanchanda@bel.co.in

Abstract-**For large enterprises driving the nation through their services or for systems addressing mission critical needs of the country even a bare minimum downtime of 0.1 % may prove fatal not just for the enterprise but also for the entire nation and hence must be strengthened with systems capable of detecting failure of components / sub-systems at an early stage. Industry 4.0 entails automation wherein systems visualize the entire operations and make decision autonomously. However, quick detection of faults followed by even faster isolation of the true cause is a longstanding challenge especially when one considers the count of endpoints employed (ranging from few hundred to several thousands) and the heterogeneity involved (be they physical servers or virtual, software or hardware, COTS / enterprise or embedded, applications or databases or middleware). No single product is available off the shelf which caters to such a wide scope in terms of monitoring. Moreover, implementation of different suites of products for monitoring different resources results in scattered, unrelated data which is meaningless unless huge manual effort is invested in analysis of the data to derive meaning out of it. The paper proposes a robust health monitoring solution for a heterogeneous environment with quick diagnosis of the root cause of issues / faults and paves way for realization of Industry 4.0. The proactive monitoring of health and availability of IT infrastructure, applications, resource utilization and databases provides the better identification of the fault. The proposed solution also provides the proactive support by classifying and organizing collected monitored data in order to detect the preliminary phases of known malfunctions and to provide an active support to speed up the recovery activities. Specific tools have been selected to make the proposed solution fault-tolerant, horizontally scalable and modular.**

Keywords-Health monitoring, end point monitoring, agent less monitoring, trend analysis, Automatic fault detection, Root cause analysis, SNMPv3, etc.

I. INTRODUCTION

With advancement in technology and development of unique tailored solutions to specific problems from the similar underlying hardware, software and middleware the complexity of the solutions and the data centers is increasing at a quick pace. The use of modular enterprise components (COTS hardware, software, middleware) and the usage of various programming languages and their associated libraries not only provides flexibility but also adds overheads in terms of compute required. There is a compulsion to use the entire component even if only some of the offered features are desired. Similarly, it also necessitates the additional security to be provided against misuse / sabotage as the components are readily available and often are having vulnerabilities which may be exploited.

Although large scale applications have encountered resource problems when the load on the overall system becomes very large. Large levels of load can expose bugs (from application to operating system to hardware) that are only caused under these conditions. Also, multiple faulting components can destabilize the entire environment and thus affect the service / operations being tendered by the system. As demand for the service increases, so does the need to increase the capacity of their systems in operational environment.

Organizations today rely on intuitive enterprise endpoint monitoring for optimal performance of their infrastructure. Endpoint monitoring tools [1], [4] provides real time health status and generate alerts in case of health/utilization issues. Monitoring is the process of checking the health status of all hardware resources & application associated with the end points to ascertain everything is as expected. It ensures that the one's endpoints are still capable of hosting/ serving applications by providing sufficient data relating to the performance of operating system and hardware, thereby

DOI : 10.35291/2454-9150.2020.0400

providing complete application and network server monitoring, and gives proper glance into the working of the system. An ideal monitoring tool should not only notify but also provide comprehensive insight into the root cause of the problem and should thus, help to resolve them quickly.

The proposed solution is an integrated monitoring & root cause detection / identification solution for heterogeneous endpoints. It serves as an endpoints monitoring solution that offers proactive server performance [6] monitoring across physical, virtual and cloud environments and monitors critical performance parameters of endpoints, providing absolute monitoring service and help to detect and co-relate the issues before they become severe threats. The solution strives to detect and diagnose complex application performance problems to maintain an expected level of service. Integrated performance monitoring is the translation of IT metrics into meaningful business parlance. The Root-cause analysis component has been implemented using custom stream based & topology based algorithms. Finally, results are forwarded to event console for fault resolution. We also propose to monitors track critical metrics, the health, performance and availability of heterogeneous endpoints over a specific period [10]. On the basis of trend analysis on application availability, CPU utilization, memory utilization, packet loss, response time, and disk utilization, it assists and simplifies decision-making processes on capacity additions, upgrades, or maintenance schedules.

II. BACKGROUND

Because of the distributed nature [11] of the applications, failures can occur at various points, some of which are outside of administrator’s control. Being able to quickly ascertain which components are functioning and which have failed can dramatically reduce the time it takes to diagnose the problem. Once the applications are back up and functioning, hard data (related to fault generation, detection & cause analysis) can be used to inform users and management of the causes of a given failure. Monitoring can prevent problems by revealing patterns of resource usage and performance that might otherwise go undetected. For example, full disks typically cause a host of problems for applications and operating systems. Simply knowing that a disk is nearing capacity may save administrator hours of time fixing the problems caused by a full disk. Monitoring capabilities in integrated health & performance tool keep a vigilant eye on the abnormalities occurring within the server infrastructure. The server and application monitor allows one to set threshold limits for vital parameters that are important to maintain server uptime and to get instant notifications if the thresholds are violated. In addition, one can automate remedial actions such as starting or stopping a server or drill down to the root cause of the issue to understand and co-relate them thoroughly.

Often, the applications are expected to be available at 24*7, and even short periods of unavailability are often considered unacceptable. Proper monitoring can identify (and sometimes correct) the fault and potential problems as effectively as possible, even without having someone to monitor it actively.

A "monitor" is normally a diagnostic command or emulation of actual use of the resource. Its results are logged so that they can be stored and/or acted upon. Monitoring systems or processes check services and components on a periodic basis, frequently enough to catch failures in a timely manner, but not so often as to significantly impact system resources. Monitoring systems [1] consist of a set of monitor, features for alerting administrators if failures occur, and a historical log of data collected by the monitors. This monitoring system check/process these three types of information:

Events/Alerts - are those states that indicate a problem or issue needs attention (application availability, Threshold breach: CPU, Memory, Disk, network availability, system availability);

Trends - provide statistics on how usage and activity are changing with time, and are most often used to plan the timing and extent of upgrades and expansions (processor, memory, and i/o utilization, bandwidth utilization, disk utilization and capacity, activity counts);

Historical data - is used to track and report on outages, failures, and activity levels for such things as service level reporting, problem tracking, and resource or activity charging.

The emphasis of various monitor types covers a broad spectrum, from “deep” monitors that simulate user actions and that test many components of application simultaneously, to “shallow” monitors that measure a single aspect of a single component. Tests performed by deep monitors involve a large number of components, and often simulate a typical user transaction; results are measured in “user units,” such as the number of seconds to complete a common task.

Monitoring of IP based device can be done either through agent based or through agent less technology.

Agent based monitoring requires an agent to be installed into the endpoints. It provides broader & deeper monitoring. In this monitoring the data is collected and filtered by the agent in the local endpoint itself and the processed results are forwarded to the centralized console / manager. Installed agent push the data to central monitoring instead of direct remote collection and agent can also temporary store / cache monitored logs when network connection is lost. Normally agent based monitoring is used for monitor servers, workstation, laptop, touchscreen etc.

Agent-less monitoring is carried out in case of endpoints which do not permit installation of agents. The monitoring is done through a remote monitoring server. This monitoring mechanism is easier to deploy since the software installation is required only on the remote data collector. This is less intrusive, easy & fast to deploy. At the same time, the agent-

DOI : 10.35291/2454-9150.2020.0400

less monitoring has lower maintenance cost since no agent version update / upgrade etc. is needed. Additionally, overheads of having an additional agent as called for in case of agent based monitoring are eliminated in this mechanism. However, in comparison to agent based solution the feature-set offered is less in case of agent-less monitoring solutions as it relies primarily on the device's potential. In general, agent-less monitoring is recommended for network devices and storage devices, UPS, UTM & different embedded electronic equipment.

Centralized health monitoring & root cause analysis tool offers numerous capabilities to measure and track resource utilization and their performance trends of infrastructure. While Trend Analysis reports [6], [10] allows admin to understand the performance trends of different parameters in endpoints. The health and application monitoring enables to plan load distribution and resource allocation adequately by providing capacity planning reports [5] that facilitates in identifying servers that are over utilized or underutilized. Additionally, it provides the facility to create custom reports and schedule automatic report generation based on set intervals.

In the recent research on monitoring, server-performance monitoring [2] is an increasingly important field of research. Our review focuses primarily on the seamless monitoring of various type of endpoints and to find the root cause of generated fault. Existing monitoring solutions usually perform monitoring by analyzing the log files, which hold information about each single request and the consequent response and errors according to the Common Log File Format, where the fields are filled with data coming from the request and the response. Given the high workload for popular servers, log file sizes can quickly grow to hundreds of megabytes. Log analyzers must present aggregated data so that they effectively represent a snapshot of the server's behavior and functioning over a given time interval. Several popular monitoring tools [2] exist, including Stephen Turner's Analog, Boutell. Com's Wusage and Maher Consulting's Access Watch provide a variety of parameters and statistics that are visualized using graphical interfaces. They aggregate monitoring data to inform Webmasters, for example, which pages in the Web site are the most popular, which countries (domains) are accessing the site, which pages outside the site contain broken links to pages in the Web site, and so on. The need for integrating several views of the log file motivated researchers to add a third dimension to graphical representation.

Mission critical system require different type [9] of electronics equipment like radar, sensors, data collection servers, data processing servers, Databases, web servers etc. for the operations. Several IP based networking and security devices are also employed for seamless secure and protected communication. Currently various type of end point monitoring tool [3], [4] are used for performance & health monitoring. As explained, there is a variety of agent based and agent less solution in the market but to meet the monitoring [7] requirements of a large scale mission critical system which entails heterogeneous environment of COTS software, hardware working in close conjunction with other legacy and state-of-art embedded systems, no single solution is available in the market. Different monitoring tools cater to specific set of endpoints and have their own fault detection and resolution techniques which are completely different from the others and so is the management of each of these solutions unique in its own self. The proposed integrated monitoring [6] solution provides the capability to carry out monitoring of different type of endpoints from a single console and at the same time assists in root cause analysis through event correlation [8], event consolidation, and suppression using stream & topology based policies.

III. SOLUTION APPROACH

In this paper we propose the integrated health monitoring of heterogeneous endpoints with root cause analysis of detected faults. In this approach, we are monitoring the IT infrastructure [5], the endpoints resource utilization, the end application processes &the databases. The data generated based on the solution architecture, the monitors applied, and the rules established provides real-time health statistics, the performance statistics and the trend analysis. The functional view of implemented solution is shown in figure 1.Our solution employs a mix of agent based, agent-less and in-built diagnostic and monitoring solutions for monitoring of heterogeneous endpoints. Different type of endpoints are monitored through different methods, the data is collected through agents and agent-less probes like SNMPv3 walk and the KPI generated by the inbuilt monitoring tool all are consolidated at central data store which further processes this monitoring & performance data. On the basis of consolidated data, the following services are derived in monitoring paradigm.

DOI : 10.35291/2454-9150.2020.0400

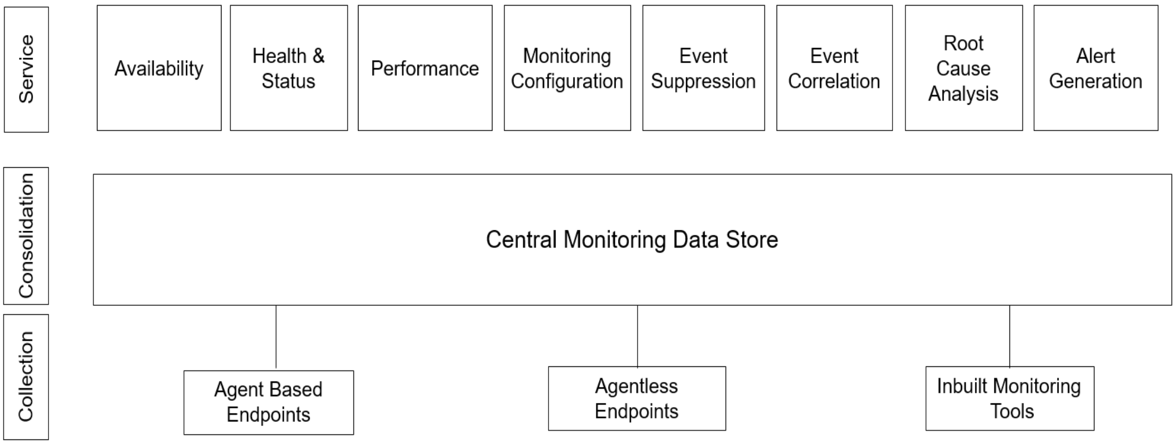


Figure 1 : Functional view of implemented solution

Application availability shows the reachability of endpoints & availability of the applications installed on the endpoints. Health & Status shows the current usage of the resource in endpoints which is used for the decision making in upgrading the resource in endpoints. Performance shows the trends of CPU utilization, memory utilization, packet loss, availability, response time, and disk utilization. Monitoring configuration provides the facility to configure different policy with threshold configuration on the endpoints which assists in deciding the health and status of endpoint for generation of Event. Event suppression facility provides the way to suppress the unwanted events thereby enabling what all events are to worked upon and which may be skipped for the time being. Event correlation is the main feature of the solution which correlates the events on the basis of different criteria like source of events, severity of events, dependency on other policy, related event etc. This serves as a critical factor in determining the root cause of the fault detected. Root cause analysis shows the probable reason of event with the co-related events for of cause & symptom for alert generation and fault reporting in event console for rectification through manual or automatic process.

Figure 2 shows the architecture of the proposed solution which has majorly three aspects: Event console, Event processor &the monitoring of endpoints. Event processor is the main component of the solution which collects the fault generated from all the end points (Agent Based and Agent less). It forwards the events to event console. After resolution of the fault the event processor update the status to the end points. Event processor also provides the user interface to user for the access the end point status. It has different sub components like Agent based monitoring server, Agent less monitoring server, Custom event adapter, monitoring & performance data store, monitoring configuration module, event correlation module, root cause detection & alert generation module. In this monitoring solution heterogeneous type of endpoints are seamlessly monitored and generate the alerts in case a breach of threshold is encountered.

Agent based monitoring server collects the health data & performance matrix through agents installed on the end points. Server also monitors the web services, application process/service, and databases. It also performs pattern match for error codes in log files for generation of the fault. For monitoring the services/process admin have to create rules and policies to define the permissible thresholds, criteria for fault generation, the patterns for error code detection, the severity & priority of the fault. The performance matrix contains the resource utilization of CPU, Memory, Disk, Network etc. The threshold breach of any policy on these device will be forwarded to the monitoring & performance data store further processing.

Agent less monitoring server collects health data through SNMPv3 based monitoring. It collects the data as per the provided management information base (MiB). The monitoring parameter configured with the threshold breach condition, fault generation criteria and polling interval on server adds more value to the monitors. The health data as per configured parameter is collected through the SNMP walk and the fault generated on these device is forwarded to the monitoring & performance data store. The monitoring may be carried out using WMI as well as SSH.

Custom event adapter is event &performance collector module which collects the data from the external monitoring tools for the seamless integration of event &performance. These custom event adapter map the external data to the required data format through groovy script. The custom event adapter read the data from external monitoring tool in fixed polling interval and after resolution of the fault it send back the information to the external monitoring tool. This provides a mechanism to integrate with any third party solution including home grown solutions for easy induction into this proposed solution of integrated health monitoring.

Monitor & Performance data store collects the data from Agent based monitoring, Agent less monitoring server and Custom event adapter. These data contains the status of health information of endpoints, availability of the process/service, utilization of resources, performance metrics etc. This data store also contains the historical data of availability, health & performance data for trend analysis.

DOI : 10.35291/2454-9150.2020.0400

Monitoring Configuration Module provides the facility to define the policy for the threshold breach. It defines the condition of the fault generation, related process/service information, source of the fault, additional information of the fault, define the policy view from topology of endpoints for health status. It also provides the facility to define the aspect for the auto implementation of policy for a type of endpoints.

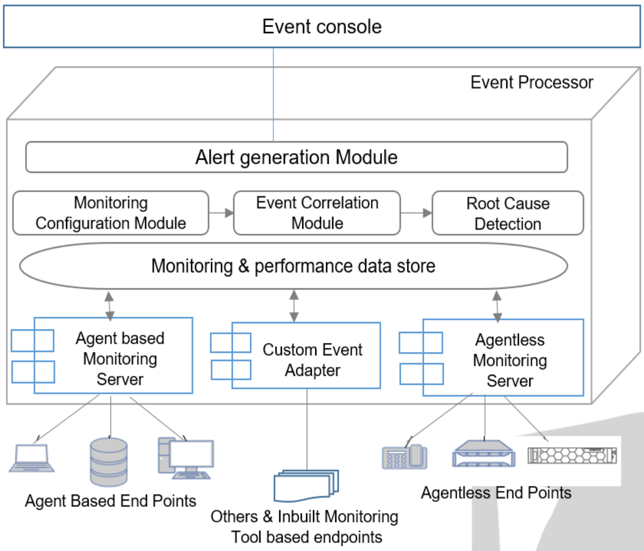

Event console
Event Processor
Alert generation Module
Monitoring Configuration Module
Event Correlation Module
Root Cause Detection
Monitoring & performance data store
Agent based Monitoring Server
Custom Event Adapter
Agentless Monitoring Server
Agent Based End Points
Others & Inbuilt Monitoring Tool based endpoints
Agentless End Points

Figure 2 – Architecture of proposed solution

Event correlation module correlates the events to identify relationships on the basis of data analyzed with different criteria. Event correlation provides the event compression (de-duplication) facility. It reduces multiple occurrences of the same event into a single event. Event correlation module also has facility to suppress the events which are trivial and are repetitive in nature. The rule base correlation is implemented and on the basis of the policy configured, the correlation module correlates the events and assigns the severity& priority to the fault.

Root Cause analysis is the next step of event correlation. It analyze the dependencies between events based on rule based reasoning. In large scale critical networked systems, a single problem at an endpoint may result in multiple symptoms and each symptom detected is reported as an independent event to the management system. The centralized monitoring solution provides better visibility with availability of these events at a single place. However, it is humanely very difficult to analyze each fault and event and treat it individually and independently (in a simplest of environment with around 1000 endpoints dispersed geographically apart with even a single event / fault being reported from say only 10% endpoints around 100 events shall get generated; each of which would require dedicated special attention for resolution) and thus makes it far more difficult to act and to resolve it. Thus, it is important that module correlates all these events and isolates the root cause of the problem in which detection of some events can be explained by the others. Here root cause analysis is done by making rules, on the basis of topology based& stream based event correlation algorithms [8] where symptom and root cause is identified for a problem on endpoints.

On the basis of the root cause analysis the alert is generated through alert generation module and forwarded to the event console for further resolution.

IV. Experimental Result

The solution is configured for more than 5000 agent based and agent less heterogeneous endpoints (physical and virtual Linux RHEL 6.x machines, RHEL 7.x machines, SLES 11 machines, Windows 2012 R2 machines, embedded systems like radios, UPS, UTMs, L2 and L3 switches). The candidates for agent based monitoring are servers, workstations, laptops, tablets etc. with enterprise operating systems and the candidates for agent-less monitoring are embedded systems like radios, UPS, switch, UTM, network devices etc. In built monitoring tools based endpoints was also configured and integrated in centralized monitoring. Finally, the KPI based integration for health status and fault monitoring with root cause analysis was carried out post implementation of correlation rules.

The health status of endpoints shows the utilization trends of CPU utilization and memory as shown in Figure 3

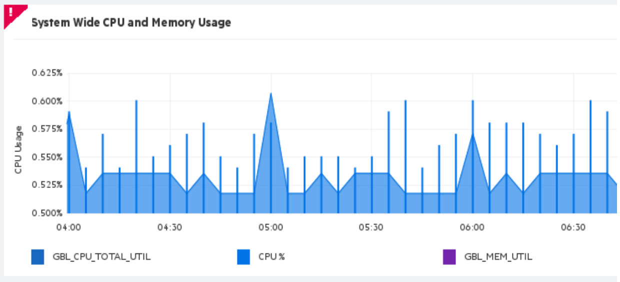
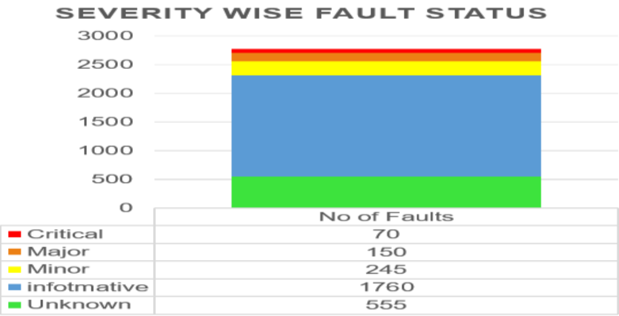


Figure 3 - CPU utilization and memory utilization trends

Figure-4 Shows the severity wise fault status in which shows that number of fault generated in different severity like critical, major, minor, warning, normal & unknown. These fault shows the current status of particular fault, the root cause of the fault with related events due to particular fault with source information and number of time occurrence.



	No of Faults
🟥 Critical	70
🟧 Major	150
🟨 Minor	245
🟦 infotmative	1760
🟩 Unknown	555

Figure 4 - Severity wise fault status

In the experiment for the root cause analysis, multiple policies were configured for database server. These included different infrastructure monitoring policy, file system

DOI : 10.35291/2454-9150.2020.0400

reachability policy, database service monitoring policy, table space monitoring, schema monitoring, connection monitoring policy, query performance monitoring, archival log monitoring, transaction log monitoring etc. More than 25 policy were deployed on the server. All the deployed policy, checked for the health status and performance of the server and reported to the integrated monitoring system. In a scenario wherein, the disk space gets full due to regular increment of archival log, any new request coming from application to database, shall be turned down as it will not be able to connect to database due to above problem. This will result in breach of the several policies configured and deployed and will ultimately lead to generation of multiple independent events. Before implementation of the proposed solution for the above scenario all the faults necessitated identification and resolution of actual issue reported as fault one by one. Thereby, resulting in the wastage of time, manpower and cost. At the same time it results in increased downtime of the system / service as well.

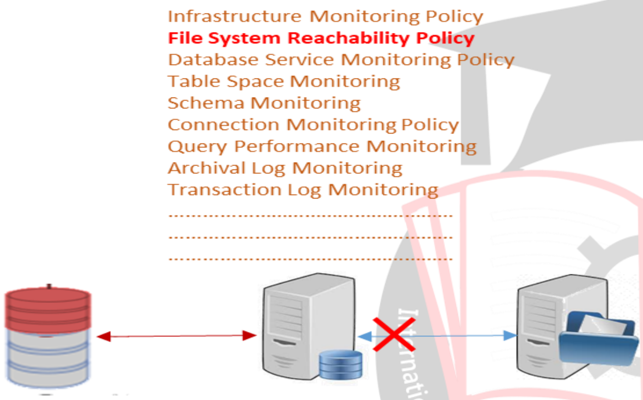

Infrastructure Monitoring Policy
File System Reachability Policy
Database Service Monitoring Policy
Table Space Monitoring
Schema Monitoring
Connection Monitoring Policy
Query Performance Monitoring
Archival Log Monitoring
Transaction Log Monitoring
……………………………………
……………………………………
……………………………………

Figure 5– Root Cause identification of fault

With implementation of the proposed solution, the rule based reasoning for root cause analysis the events due to breach of multiple policies are correlated and the analysis of the symptoms and causes deduce the actual problem. On the basis of the stream based event correlation algorithm multiple events are suppressed like “Database connection id failed”, “listener is not running”, “Table space is not available” and the root cause of the problem “Archival Log is full”. On the basis of these correlation the root cause problem is identified and alert is generated for the breach of file system threshold (Figure 5). All the other related events get tagged to this root cause and are not projected separately to the monitoring admin. After the resolution of this root cause fault all other related events get closed automatically. With implementation of the solution the fault detection / resolution time and thus cost decreases drastically and so does the cost of maintaining the system is reduced.

In this experiment, the analysis were conducted on three month data. The number of fault reported before & after implementation reduced to approx. 95%.

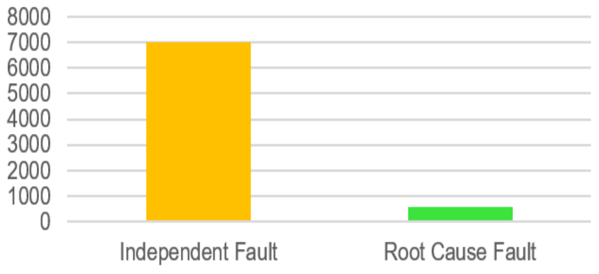


Figure 6– No. of independent fault vs Root cause fault

The number of independent fault vs number of fault after implemented solution is shown in figure 6 for the duration of three month. It describe the independent 6852 fault is the reason of only 370 faults. In time analysis for the fault detection & resolution have also improved approx. 80%. The time to detect &resolve these independent fault verses root cause fault is shown in figure 7.

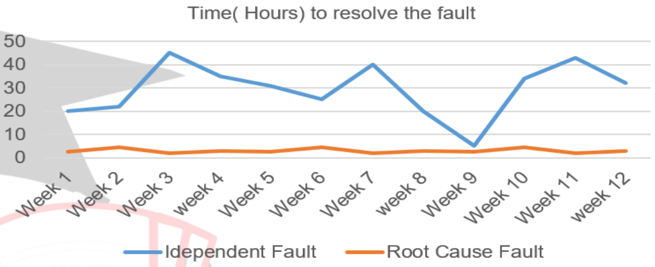


Figure 7– Time to resolve independent fault vs Root cause

It shows the week wise fault resolution time in hour for independent fault & root cause fault.

V. CONCLUSION

This implemented solution is capable to monitor the heterogeneous endpoints, detect the fault after root cause analysis and generate the alerts on the event console for further resolution. This implemented solution provides the integrated monitoring of heterogeneous endpoints which depicts performance metrics, availability of application & end points. The solution also helps to identify, consolidate faults and perform root cause analysis on the basis of rule based correlation. The main finding of solution to improve the performance of the monitoring through reducing the manual intervention and automate the fault detection with actual problem. It also provides the resource utilization trends which assists and simplifies decision-making processes on capacity additions, upgrades, or maintenance schedules. It also provides the historical data which helps to analyze performance and utilization trends of heterogeneous endpoints.

With determination of the root cause of an issue observed, the future work may entail auto-resolution of the faults taking a step ahead towards creation of a perfect industry 4.0 eco-system. This would reduce the downtime and outages further. At the same time would involve minimal human intervention.

DOI : 10.35291/2454-9150.2020.0400

VI. Acknowledgment

We would like to thank our Organization Bharat Electronics Ltd, our Unit Head and Executive Director, Mr Joydeep Majumdar and our Divisional Head, Gp Capt RK Dhir for giving us an opportunity to establish and work on the solution for IT infrastructure monitoring and for motivating and inspiring us. We would also like extend our gratitude to Mr. Amit Gupta, Mr. Aditya Kr. Singh and Mr. ManojTyagi for their unwavering support and guidance.We would also like to thank other team members of our department for providing support in implementation of IT infrastructure as and when required.

References

[1] Sorin POPA, WEB SERVER MONITORING, Journal of University of Craiova, vol 1., 2006

[2] D.A Menasce and V.A.F. Almeida, Capacity Planning for Web Performance: Metrics, Models, and Methods, Prentice Hall, Upper Saddle River, N.J., 2003.

[3] GMaria Barra, Tania Cillo, Antonio De Santis, Umberto, Ferraro Petrillo, Alberto Negro, and Vittorio Scarano, “Multimodal Monitoring of Web Servers,” vol. 2 April 2002

[4] A E Aktany, A J Helmicki and V J Hunt Issues in health monitoring for intelligent infrastructure, Smart Mater. Struct. 7 674–692., 2007

[5] Alain Mouttham, Liam Peyton, Ben Eze, Abdulmotaleb El Saddik, Event-Driven Data Integration for Personal Health Monitoring, Journal Of Emerging Technologies In Web Intelligence, Vol. 1, No. 2, 2009.

[6] Chris A. Otto, Emil Jovanov, and AleksandarMilenkovic, A WBAN-based System for Health Monitoring at Home, CSE journal ,2010

[7] Anshulkaushik,”Use Of Open Source Technologies For Enterprise Server Monitoring Using Snmp” International Journal On Computer Science And Engineering, vol. 02, no. 07, 2246-2252, 2010.

[8] Michael Tiffany, A Survey of Event Correlation Techniques and Related Topics, ACM Conference ICCCT. Vol 2 556-558, 2002

[9] Christopher Roblee Vincent Berk George Cybenko, Implementing Large-Scale Autonomic Server Monitoring Using Process Query Systems, International Conference on Autonomic Computing (ICAC’05, 0-7695-2276-9/05 IEEE), 2008

[10] VivekanandMathapati1, Dr.A R Aswatha, Performance Analysis Of System Resources By Server Monitoring, International Journal Of Innovative Research In Science, Engineering And TechnologyVol. 2, Issue 7, 2013

[11] Domenico Elia1 , Gioacchino Vino1,Giacinto Donvito1 and MaricaAntonacci, Developing a monitoring system for Cloud-based distributed data-centers, EPJ Web of Conferences 214, 08012 ,2019

Mr. Brajesh Kumar joined Bharat Electronics Ltd. in 2019 as senior Engineer in Network Centric Systems, Strategic Business Unit. Previously he was working in ISRO from 2014 as Research Scientist. He has vast experience in the field of Network Security, Network Management and data center management. He had completed B.E. and M. Tech in Computer Science in 2009 and 2012. He has published more than 15 technical papers in journals & conferences. He is also reviewer of various international journals.

Mr. Shakti Manchanda joined Bharat Electronics Ltd. in 2009. Currently, he is working as Deputy Manager in Network Centric Systems, Strategic Business Unit. He has vast experience in field of Information Security, Identity & Access Management, Network Management and Endpoint Management. Having completed Bachelor of Engineering in electronics &communication from GB Pant Engineering College, Mr. Shakti Manchanda did Masters of Technology in software systems from BITS Pilani. He has technical papers published in symposium, journal and conferences. He is also a certified PMP.

DOI : 10.35291/2454-9150.2020.0400