ITSM Based Server Monitoring System

Swapnil H Sonawane [1], Swati M Khandare[1], Shruti Patil [1]

[1]*Dept of Electronics and Telecommunication, S. L. R. Tiwari College of Engineering, Thane, Mumbai University, India.*

Corresponding Author:Capt.swapnil.sonawae@gmail.com

Abstract: - Resource availability is one of the most important requirements in the Business-Critical application. Techniques are required to increase the availability of the network to deal with new services and the consequent new traffic profiles and characteristics. This project proposes a more flexible scheme using ITSM lead to an efficient solution for Server Monitoring. The IT Infrastructure Library (ITIL) provides guidelines for IT service providers how to design, manage and support IT services. ITIL is the most widely used IT service management (ITSM) framework. It consists of best practices that can be used in implementing, for example service support processes, such as incident management and problem management. Although ITIL includes a wide list of process metrics. Considering servers being the core position in network, this Project introduces how to monitor servers through simple WMI. We expand WMI resources by defining WMI objects to monitor the resources of sever and use multi-threading technology to collect data and process them, which can improve the collection efficiency. The Project results prove that it is a successful way of integrated to ITSM and monitor, control for servers.

Key Words: — ***Traffic profile, ITSM, Server Monitoring, incident management, WMI.***

I. INTRODUCTION

Techniques for increasing the efficiency of Server Monitoring in Enterprises due multiple servers to deliver business critical services for their end users. Some of them include database servers, core app servers, caching servers, web servers, and more. Performance of each of these servers are critical because even if one of the servers fail, then it impacts the delivery of business-critical services. Therefore, it is imperative to know any performance issues proactively so that they are identified at the early stage and fixed before they turn big and pose a threat to business. Server monitoring tools help in monitoring servers as well as the entire infrastructure. They also provide intensive reports on capacity planning to maintain the network without any hassle.

Server Monitoring is the process of monitoring a server's system resources like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Process etc. Server Monitoring also helps in capacity planning by understanding the server's system resource usage. A Server Monitor software helps in automating the process of server monitoring. Monitoring Server performance also helps in identifying other performance related issues like resource utilization, app downtime and response time.

Why is it important to Monitor Server performance?

- To monitor server availability and data loss.
- To monitor the responsiveness of the server.
- To know the server capacity, user load and speed of the server.
- To detect and prevent any issues that might affect the server proactively.

Information technology service management (ITSM) are the activities that are performed by an organization to design, plan, deliver, operate and control information technology (IT) services offered to customers. Differing from more technology-oriented IT management approaches like network management and IT systems management, IT service management is characterized by adopting a process approach towards management, focusing on customer needs and IT services for customers rather than IT systems, and stressing continual improvement.

Why is it important to IT Service Management?

- ITSM Provides a Clear Line of Sight Between Individual Contributions and Business Results.
- ITSM Creates Structure within the Business.
- ITSM Facilitates Consumer Self-Service and Self-Help.
- ITSM Provides the Basis for Automation of Routine Operational Activities.
- ITSM Helps IT Identify and Implement Justifiable Improvements

Core part of IT Service Management Service Desk

A Service Desk is a primary IT function within the discipline of IT service management (ITSM) as defined by ITIL. It is intended to provide a Single Point of Contact ("SPOC") to meet the communication needs of both users and IT staff, and also to satisfy both Customer and IT Provider objectives. "User" refers to the actual user of the service, while "Customer" refers to the entity that is paying for service.

Manuscript received March 18, 2021; revised March 20, 2021; accepted March 21, 2021. Date of publication March 22, 2021.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898

The ITIL approach considers the service desk to be the central point of contact between service providers and users/customers on a day-to-day basis. It is also a focal point for reporting incidents (disruptions or potential disruptions in service availability or quality) and for users making service requests (routine requests for services).

The aim of the incident management process is quickly resolving incidents that affect the normal running of an organization's IT services. An incident is an intimation of some error or failure of some component in IT systems. Figure 1 shows the incident management workflow which can be used for resolving an incident. In a typical service desk, incident is either reported by the customer or automatically generated by system monitoring/event generation system. Customer report incidents by describing the system condition using natural language text whereas automatically generated incidents only have structured data specifying system and event-class [17]. In this paper we are considering customer reported incidents only. For such incidents, Chat boot does a quick "keyword-based search" from a database of historic incidents. If any matching incident is found, its solution may be used to resolve the incoming incident. If the Chat boot cannot provide any resolution, an incident record is created. This incident record is classified for various purposes such as assigning priority based on urgency and impact, selecting the appropriate SME, etc. Then failing component is identified by manually associating hardware or software components (configuration items) responsible for the incident. Information about these configuration items (CIs) is maintained in a Configuration management database (CMDB) which is also used by other ITSM processes as an underlying data storage framework.

L1 person uses keyword search along with human intelligence to guess the possibly responsible CIs. Then the incident ticket is forwarded to L2 support to diagnose the problem in the selected CI. For diagnosis the CI is monitored and various probes [14] may be used. If the identified CI is wrong ticket is bounced back and forth between L1 and L2 support. If any code change is required external support (L3) is contacted. After resolving the problem customer is informed and incident ticket is closed. In this paper we propose techniques to automate and improve various stages of the incident management workflow.

Next, we describe our approach with the help of an example and outline our contributions.

II. INCIDENT MANAGEMENT PROCESS

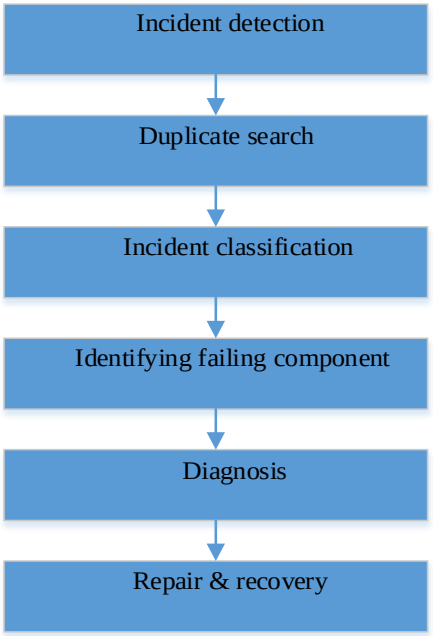

Figure.1. Incident Management Workflow

A. Contributions

We explain our contributions with the help of an example. Let us assume that on of customer critical server has failed and describes the problem as:

Example1: "I tried launching inventory application on server avalanche.server.net from 10.10.10.70. I get an error saying that database transaction failed."

First, we automatically identify relevant keywords which can be used to create incident and categories and identify possible failing. When customer describes the system problem, (s)he describes only her or his perception of the problem. Thus, problem description may partially or not at all mention the possible failing component explicitly. We use the incident description data to get the context of the search and identify the failing component. For identifying the incident context, incident classification plays a key role.

The rest of the paper is organized as follows. In Section 3, we give details of data model of the server monitoring system. We use ITSM model and relationship between objects for performing server monitoring. Steps to automatically identify server status and crate incident automatically with description are presented in Section 4. Function design of system. in Section 5. Implementation details and performance results showing effectiveness of show that accuracy of problem resolution can be improved by more than 70% using our approach. By our proposed automations, we help the service desk by: describe towards the end of the paper. Using performance results over the actual customer data.

III. SERVER MONITORING

With the increasing expansion of the computer network and communication size, network has become essential to communicate for people's daily life. At the same time, the network monitoring has also developed rapidly. At present, the majorities of network management software are focused on the link and network equipment, but server should also be given adequate attention for being the carrier loading network services. In this paper, we introduce a server monitoring system, which not only monitors hardware and software of the system, but also monitors the security of the server's information.

At present, there are two types of network management protocol in computer network management field, which occupy the dominated position. One is common management information protocol and service (CMIP/CMIS), which is proposed by OSI organization. And the other is WMI. The two protocols are corresponding to two different management programs. WMI has become the most popular windows management protocol for its simplicity and scalability. Almost all of the windows server support WMI.

As soon as server fail system will automatically generate new incident by associating CI and intelligently selected keywords with historic incidents we improve recall of the "duplicate search" step.

By automatically associating responsible CIs, we help in reducing the number of incident tickets being forwarded (rightly or wrongly) to SMEs. As SMEs are costly to get and maintain, this will help in reducing operational costs.

We implemented a rule-based request router which uses various attributes of incident ticket, including associated CI, to automatically route the ticket to most appropriate SME.

Using the importance (or role) of the selected CIs, the service personnel can assign priority to the incident. For example, if the selected CI is a machine on which billing application is installed, and then higher priority can be given compared to the case where selected CI is the backup software.

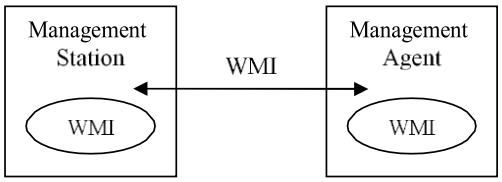

Fig.2. The SNMP-based management model

The management station serves as the interface for the human network manager into the network management system. The management station will have, at minimum: a set of management applications for data analysis, fault recovery, and so on; an interface by which the network manager may monitor and control the network; a protocol by which the management station and the managed entities exchange control and management information; a database of information extracted from the management databases of all the managed entities in the network. Only the last two elements are the subject of WMI standardization.

The management agent responds to requests for information from a management station, responds to requests for actions from the management station, and may asynchronously provide the management station with important but unsolicited information.

In order to manage the resources in a network, these resources are represented as objects. Each object is, essentially, a data variable that represents one aspect of the managed system.

The collection of objects is referred to as Windows management instrumentation bases (WMI), which are written in a language called .net. net is a data-oriented language.

The WMI functions as a collection of access points at the agent for the management station; the agent software maintains the WMI. A management station performs the monitoring function by retrieving the value of WMI objects.

The management station and agents are linked by a network management protocol, which includes WMI messages: GetRequest, GetNextRequest, SetRequest, GetResponse and trap. All the first three messages enables the management station to retrieve or set the values of objects at the agent, are acknowledged by the agent in the form of GetResponse message. In addition, an agent may issue a heart bit message in response to an event that affects the WMI and the underlying managed resources.

A. Analysis of Monitoring Context

Server monitoring context includes the following four areas: static information (hardware description, software description, administrator, physical location, etc.), dynamic information (interface traffic, usage of CPU, memory and disk, etc.) network services (HTTP, FTP, DNS, SMTP, POP3, SQL Server database, etc.) and network performance.

MIB and host resources WMI can monitor the state of hardware and software, and the running state of the system. We also want to achieve a comprehensive server performance monitoring, but the standard WMI cannot meet the need. So we can expand the agent by adding WMI database files which are defined by ourselves according to the standard, in order to expand the WMI.

IV. FUNCTION DESIGN OF SYSTEM

The monitoring system is based on WMI manager- agent model [3, 4]. We use layered structure method to design the system according to the different functions of the system. The

system should include the following modules, which are shown in Fig. 2.

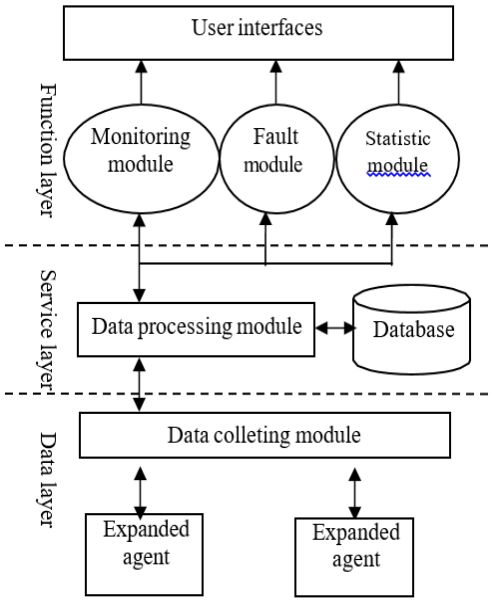

Fig.3. Frame work of the system.

We design the system into three layers: data layer, service layer and function layer.

The data layer at the bottom is responsible for the communication between manager and agent, require and set the information in the WMI. Here we use two ways to collect data. One is real-time collection, which collets the real-time information that need to be displayed and sending them to the upper layer in time. Another way is timing polling, which collects information regularly during an interval and sends them to the upper layer.

The service layer in the middle layer is responsible for dealing with the collected information. If the collected information needs to display timely, it will be directly send to the upper layer to display, or it will be stored into the data- base for querying the history information.

The function layer in the upper layer is the interface for the administrator operating the system. It can show the monitored server's configuration information, performance information, and fault information in a visual and graphical interface.

V. Implementation Of The System

The development environment is Windows 2000 server, which encapsulates the implementation of the WMI protocol, and provides a set of interfaces for developing network management programs based on WMI, called API. We use the SQL Server 2012 as our database and choose .net as the development tool.

A. Agent Expansion

We choose WMI agent to expand the WMI command output. We can define the WMI objects according to SMI standards and add them into WMI agent. The detail steps are as follows.

First of all, we design our own WMI files according to SMI standards. Secondly, we compile the files, and use mib2 tool to generate the procedure framework, and then improve the framework program. At last, we recompile the files, install the new agent and run it.

B. Data Collecting Module

The collecting program runs on a single manager station (a computer or workstation). The colleting process creates four threads. They are sending thread, receiving thread, preprocessing thread and storing thread. The four threads are controlled by a main controlling thread. Sending thread is responsible for sending WMI request message. Receiving thread is responsible for receiving WMI response message. Preprocessing thread is responsible for filtering and integrating the collected flow information.

Fig.4. Memory utilization test figure.

Storing thread is responsible for storing data into database, which are preprocessed by preprocessing thread of the Memory utilization at every acquiring moment and its changing rate.

VI. Conclusions

Considering the advantages of WMI, such as simple, flexible, small network load, and its strong expansion, we develop the system which can monitor and control servers under the condition of not affecting the server's load and its service performance. It also can make alarm and create incident to the administrator immediately when any abnormality occurs. The system runs well so far. It is very convenient for administrator to see the monitoring results because of the graphics display. So it is very meaningful to develop the server monitoring system.

REFERENCES

[1]. Rajeev Gupta,K Hima Prasad, Mukesh Mohania, "Automating ITSM Incident Management Process," IEEE Computer Society©2008.

[2]. Sven Graupner, Sujoy Basu, Sharad Singhal, "Collaboration Environment for ITIL," 2009. IFIP/IEEE Intl. Symposium on Integrated Network Management—Workshops.

[3]. Manuel Moraa,∗, Jorge Marx-Gómezb, FenWangc and Ovsei Gelmand, "IT Service Management and Engineering: An Intelligent Decision-Making Support Systems Approach," IEEE Transactions c 2014.

[4]. Zhong YAO1, Xin WANG2, "An ITIL Based ITSM Practice: A Case Study of Steel Manufacturing Enterprise," IEEE Transactions ©2010.

[5]. Neha Atul Godse, Shaunak Deodhar and Shubhangi Raut, Pranjali Jagdale"Implementation of Chatbot for ITSM Application using IBM Watson," 2018 IEEE, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

[6]. Antti Lahtela and Marko J¨antti, Jukka Kaukola, "Implementing an ITIL-based IT Service Management Measurement System," IEEE Computer Society © 2010 IEEE DOI 10.1109/ICDS.2010.48.

[7]. Wenxian Zeng and Yue Wang, "Design and Implementation of Server Monitoring System Based on SNMP," IEEE Computer Society © 2009 IEEE DOI 10.1109/JCAI.2009.34.