



## REDUNDANCY CHECKS ON ARCHITECTURE FOR CLIENT SYSTEMS

Dr. A. P. Nirmala

Senior Assistant Professor, Department of Master of  
Computer Application New Horizon College of Engineering  
Bangalore, India

Sudhanshu Kumar

Cerner Health care Pvt. Ltd  
Bangalore, India

**Abstract:** The availability (uptime) of UNIX servers which are hosting critical patient care applications is of utmost importance. This availability is ensured by maintaining redundant network; storage connections to a physical server (Ethernet cables; Optical Fibres). The health of these redundant connections, are monitored daily, so that issues can be fixed proactively, which ensures the availability of the servers. If issues are identified with these connections, the concerned team is involved in troubleshooting the issue. Due to the critical nature of the project, perfect ITIL (Information Technology Infrastructure Library) procedures should be followed in identifying, recording and fixing the issues. Unfortunately, this requires a lot of manual work. To help reduce this manual labour and save time, a Web Application was proposed to be developed. This web application will be useful to track, monitor, and analyze the work progress and to visualize the data for easy understanding. It reduces errors by reducing human interaction in the process, thereby saving valuable labour time and resources for the organization. Thus, the generated report becomes more accurate.

**Keywords:** Path check, UNIX servers, HP-UX, AIX, CSV, Analytics, ITIL

### I. INTRODUCTION

Maintaining redundancy for UNIX servers which are hosting critical patient care applications is one of the top priority metrics provided to the customer. Redundancy for both SAN disks provided to server and the Network by which the applications are accessed by the end user are present in the architecture of data centre in which the physical servers are housed and stored. These architectural specifications are provided to all servers which contain client data. Daily background analysis is done for these nodes and the report of the failed checks are sent for resolution. Currently these reports are being sent over mail, which must be extracted to Excel and manual assignments should be done. This requires a lot of effort on a regular basis.

To overcome these hurdles, a Web Application will be developed to automate the manual process and remove the arduous tasks which require human intervention that is time consuming. Its purpose will be to automate the process of maintaining the failed checks which the team receive daily. Functionality will include to assign these requests and to provide a visual aspect of the data such as number of pending tasks, completed tasks, work load on each employee, work progress of each employee, etc. Daily reports based on these data will be generated and will be sent to the team. This web application will be developed on a LAMP (Linux, Apache, MariaDB and PHP) Server. The other technologies used are HTML 5, CSS3 (Cascading Style Sheets), JavaScript (JS), Java Script Object Notation (JSON). The UI design and behavior will be responsive and is based on Bootstrap3.

This web application will be useful to track, monitor, and analyze the work progress and to visualize the data for easy understanding. It reduces errors by reducing human interaction in the process, thereby saving valuable labour time and resources for the organization

### II. RELATED WORK

The authors in [1] presented a measurement-based dependability analysis of a UNIX server. The event logs of a

UNIX server, are collected to form the dependability data basis. The event log data are classified and categorized to calculate parameters such as MTBF (Mean Time Between Failures) and availability. The authors performed component analysis in order to identify modules that are prone to errors in the system. They analyzed system error activity preceding each system failure to find error patterns that might be precursors of the observed failure events. They presented the study on failure results from the measurement in view of the fault/error assumptions made from fault injection analysis.

The possibility of enhancing an existing performance monitoring system for UNIX servers, by adding the capability of predicting upcoming failures, using generic UNIX operating system performance metrics like used server memory, CPU utilization, I/O traffic etc. as input data for machine learning and pattern recognition is highlighted in [2]. In this paper, the authors mentioned possible research methods based on input data they process, and propose a new approach for symptom based failure predicting. In order to make a generic solution that can be used on any UNIX computer, they have only used open source software. They implemented the classifiers Naive Bayes and Logistic Regression with input data in both standard and vectorized format. The authors used search algorithm Forward stepwise selection to find an optimal generic set of variables (features) that improves the quality of the classification. The implementation of empirical testing showed that this approach is capable of predicting symptoms with high overall accuracy, but the uncertain quality of the monitored performance data used as input makes it difficult to ascertain if the symptoms are actually failures.

Assurance and auditing are the obligatory activities to secure the information of any organization. Auditing has to be considered as a continuous and ongoing process, no matter what system or provider is being used. The audit and assurance program needs to examine the system configuration and the status of information security on a periodic basis to avoid cyber attack. It is vital to assure that it is configured properly to make sure the security of business information. The authors in [3] highlighted that a comprehensive, all-encompassing auditing

solution that has to be implemented at operating system level. Otherwise, organizations put critical information at risk.

The authors in [4] highlighted the study of server log data and the detection and potential prediction of anomalies related to the monitored servers. The issue is relevant in many mission-critical systems consisting of multiple servers. There it is favorable to be able detect and even foresee problems to be able to react promptly and apply required corrections to the system. In their study, the authors made off-line analyses based on pre-recorded data. In reality, if the objective is to come up with solutions for detecting anomalies in real-time, additional requirements and constraints would be imposed on the algorithms to be used.

In this paper, we study focuses on the path failures of UNIX nodes which includes RHEL (Red Hat Enterprise Linux), AIX (Advanced Interactive executive) and HP-UX (Hewlett Packard -Unix). These proactive checks are for SAN paths and Network paths; the redundancy for them ensures maximum uptime for the client environments as the data stored in a node can be very critical at times. The product necessity will help us to plot and talk about every one of the prerequisites in detail with the goal that it can be utilized as a kind of perspective at the time of improvement. The current approach by which the path checks are managed is a complete manual method which requires constant human intervention and is prone to errors.

This method requires a lot of effort and time which in turn reduces the productivity of the associates in fixing the actual path failures which occur in UNIX Systems. All the incidents raised for such checks should be written to an Access Database form line by line.

There is no room to gauge any historical data from the current database. Employees cannot visualize any data which is present in this form. Weekly mails should be sent manually by the method of copy and paste. There is no way to detect and track who has edited any of the entries in the database. The Current system which is made to run on the access DB platform have lot of issues starting with handling the user interaction at every stage which results in manual or human errors to be specific. The updating of all the nodes in the environment is a very tedious work as it is to be done by an associate and there are many chances of missing out a single node which may be very important, as the organization is handling patient's information it needs to be the very careful while handling such information. There is no track of the work done by the associates to keep up the server all the time.

The basis of the path check is to maintain and keep the client operating environment as smooth as possible. In terms of

the architecture of the infra provided to the Organization customers, this is done in a manner which requires all possible points of failures have a redundant hardware or software in place in case of any catastrophic event.

In such a case the client would not face any immediate outage as the services would failover to the redundant infrastructure providing time for us to fix the root issue. However, the success factor of this implementation depends upon the fact that the redundancies should also be in a working condition. To have this, our team does proactive path checks to determine and fix any existing issues and to prevent any future ones.

### III.DASHBOARD FOR THE CHECKS OF SERVERS

The proposed system would help the team to evaluate and visualize the progress of path check fixes and failures. This would eventually bring about a streamlining of the entire process for the checks and improve the timing, efficiency and productivity of the team as it will be free from manual intervention which may cause some human errors at times. The web portal would include tools such as automated mailing services to send out alerts to the team and interactive graphs and charts for managers and selective directors for better overview. The portal also includes a privilege system where the view of the web page would differ for associates with different roles. For tracking the web application utilizes a visitor logging system which shows who have logged in and have made any changes to the data. The portal also provides the facilities like sending out an alert to the concerned Engineer depending upon the region

### IV. RESULTS AND DISCUSSION

The Application is developed using various languages such as JavaScript, Ajax, PHP, HTML, CSS. JavaScript is used to provide a realistic functionality by which the functions gets triggered and give the desired output which is required by the Employees, Ajax technology is used for saving the live data, PHP and CSS for the frontend design.

Bootstrap 3 is used to give the frontend design which has a few templates to improve the GUI part for the Dashboard.

The homepage has the data for the employees where they can see the failed or faulty paths for which they should work upon. Figure 1 shows the progress in proactive paths in different locations like KC and LS.



Figure 1 Task Details in Network Pathcheck

## Network Bar Chart

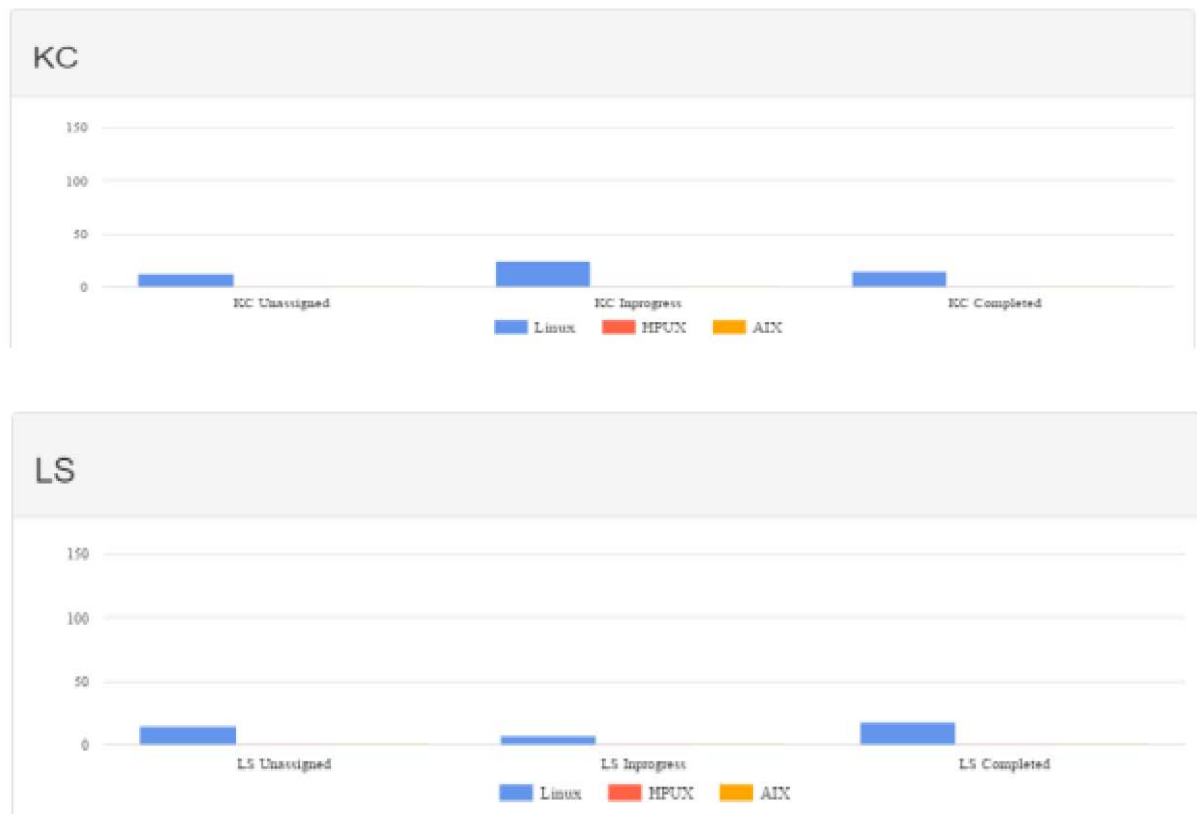


Figure 2. Number of tasks in assigned or unassigned state in two locations



Figure 3 Task Details in SAN Pathcheck

## Associates Logs

Sudhanshu Kumar
sk051189 has logged in at 2017-04-06 12:01:13 in Chrome from Windows 10 and logged out at 0000-00-00 00:00:00
Arindam Lahiri
al045225 has logged in at 2017-04-04 10:16:40 in Chrome from Windows 8.1 and logged out at 2017-04-06 10:19:14
Arindam Lahiri
al045225 has logged in at 2017-04-05 00:26:31 in Chrome from Windows 7 and logged out at 2017-04-06 10:19:14
Pankaj Mishra
pm051182 has logged in at 2017-04-06 09:44:29 in Chrome from Windows 10 and logged out at 0000-00-00 00:00:00

Figure 4 Logs to track user login

The bar graphs in Figure 2, shows the number of tasks in assigned or unassigned state. The pie charts are also available to view the total number of tasks in all the data centres. Global search options are available to view any field from the homepage itself, there is no need to go the tables as well. The task details for SAN Pathcheck also shown in Figure 3.

Figure 4, shows that how to keep tracks of all the users who logs in the dashboard and use them, the time of login and browser as well as logout time is monitored

## V. CONCLUSION AND FUTURE WORK

Since the inception of the “Redundancy checks for the Architecture of Client System” which is being implemented in Organization, various programming and logical instruments were used to fine tune the application and get a better, more accurate representation of the path check data. The client’s requirements were given top priority while designing the Application and all the inputs or data or suggestions were taken into consideration well before delivery of the Application for usage to end users. The system is made in such a way that it meets the requirements of the Associates working upon the proactive path checks and is developed to make future enhancements easier. A modular approach was taken from the start of the Application for easy understanding which facilitates any further development upon this tool. In near future, enhancements can be performed to integrate the web application with BMC Remedy tool to automate the process of

raising an incident for any failed path by providing the information from the dashboard. The portal should also be able to send the mail to the regional engineer depending upon the regions where the nodes will have some issues and if the employee keeps a request for more than 5 days in his queue an auto-generated mail will be triggered to the concerned person as well. A separate reporting module should also be added as it will provide a crystal-clear image of the work done by the associates for the proactive checks for the servers which helps to keep a high uptime for the client environments.

## VI. REFERENCES

- [1] Lal, R. and Choi, G. 1998. Error and failure analysis of a unix server. In IEEE Proceedings of third International High-Assurance Systems Engineering Symposium (HASE). IEEE Computer Society Washington, DC, USA, 232–239
- [2] Svendsen, P. A. Online Failure Prediction in UNIX Systems [Text] / P. A. Svendsen. – Kristiansand: University of Agder, 2011. – 70 p
- [3] Muhammad Mushfiqur Rahman, “Auditing Linux/Unix Server Operating Systems”, ISACA JOURNAL VOLUME 4, 2015
- [4] Nousiainen, S., Kilpi, J., Silvonen, P., and Hiirsalmi, M. (2009). Anomaly detection from server log data. Technical report.