Towards Next-Generation Alert Management of Data Centers

Praveen Venkateswaran†, Aruna Malapati
BITS-Pilani Hyderabad,
Hyderabad, India.
Email: {f2011640, arunam}@hyderabad.bits-pilani.ac.in
† Student author

Maitreya Natu, Vaishali Sadaphal
Tata Research Development and Design Centre,
Pune, India.
Email: {maitreya.natu, vaishali.sadaphal}@tcs.com

***Abstract*—The performance of today's enterprise IT systems depends upon the accurate generation of alerts to identify any anomalous behavior. Today's solutions however suffer from several shortcomings. The configurations are manual and do not adapt to any system changes. A large volume of redundant alerts are generated leading to inefficient resolution. Moreover, the generated alerts are reactive in nature and provide less time to take corrective actions. In this paper, we address the issues of setting the correct configurations, aggregating redundancies and generating proactive alerts. We demonstrate the effectiveness of our approach on a real-world case-study.**

I. Introduction

The health of today's enterprise IT systems is continuously monitored to maintain high levels of availability and performance. Various business, application, and infrastructure components are monitored to report any abnormal behavior such as failures, anomalies, violations in service level agreements, and outages. The abnormal behavior is notified in the form of alerts which are analyzed by a team of resolvers to take corrective actions.

Various tools [4] are available to monitor system components and generate alerts. However, the effectiveness of monitoring and alert management relies on accurate configurations. Much of the prior work has developed solutions to analyze the monitoring and alerts data. These works have developed sophisticated algorithms for various objectives such as performance management, capacity planning, strategy planning, and risk analysis [5]. However, these works rely on the correctness and quality of the data generated by monitoring and event management agents which is often unreliable.

Our experience with various real-world case-studies has provided us with the following insights on the limitations of the state of the art:

- Imprecise configurations: Alert configurations today, are manual and intuition-based, leading to either too many false alerts or missing many legitimate problems. Furthermore, the configurations are static and do not reflect changes in the system. We propose to generate dynamic configurations by analysing the behavior of the system components.
- Redundant alerts: A multitude of wasteful alerts are observed because one fault often generates many symptoms. We propose to aggregate these redundancies to allow critical alerts to be identified quickly.
- Reactive alerts: Most alerts trigger upon observing an anomaly and fail to give sufficient time margins to take corrective actions. With reactive alerts, measures can be taken only to find a quick work-around. We propose to generate proactive alerts based on predictions allowing enough time for alert resolution.

In this paper, we address these limitations and articulate a solution approach to create a next-generation alert management system. Though the concepts presented in this paper are generic, we take an example of batch systems to explain our proposed solution.

II. Solution Approach

In this section, we present initial ideas of the proposed solution approach.

A. Alert configuration

Today's alerting mechanisms support flat single valued thresholds and are agnostic to temporally varying behavior. Also, setting the right threshold is left to the experts' tacit knowledge. This is not a scalable solution and leads to inconsistent quality of alert configurations. We propose to overcome these shortcomings by using the historical data of each component in order to accurately model its normal behavior. Our approach involves the following steps. (1) We first identify the most recent steady state of the component using changepoint analysis (Fig. 1(a)). This is because the current behavior of the component would be best represented by its most recent steady state. (2) We then identify temporally varying behavior such as weekdays and weekends, or mornings and evenings. This is challenging as the number of possible temporal dimensions is large. They can range from simple choices such as "day of month" to complex ones such as "friday and last week of month". We identify the different behavior using Classification and Regression Trees (CARTs) (Fig. 1(c)). (3) We then define a normal range of values in order to set a threshold for each of these identified behaviors (Fig. 1(b)). A way to compute this range of values is $median \pm k * MAD$, where MAD refers to Median Absolute Deviation, and k is a configurable factor based on the skewness of the distribution.

978-1-4673-9922-6/16/$31.00 ©2016 IEEE

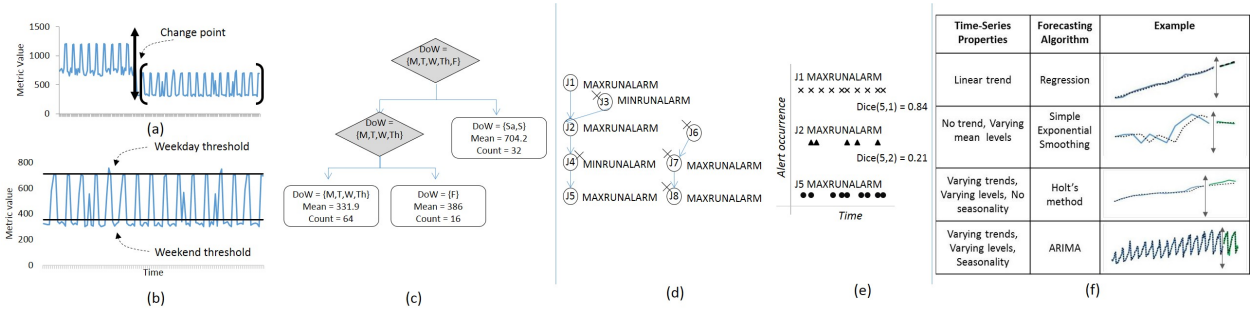

Fig. 1. Proposed solution approach- (a) identifying change points (b) thresholds for different behaviors (c) CART (d) pruning by structure and alert type (e) alert time-series' and similarity (f) prediction based on properties

B. Alert aggregation

There are complex interdependencies between components. Due to this, a fault at one component can manifest itself as alerts at other components, thus causing many redundant alerts. We propose to analyze the past history of alerts to identify these redundancies by recommending rules for alert aggregation. We propose a method based on temporal and structural correlations. In Fig. 1(d), to identify alerts correlated with the MAXRUNALARM alert of job J5, we first prune the jobs not in its dependency tree and having incompatible alerts. We then use Dice's coefficient[2] to derive aggregation rules (Fig. 1(e)). These rules help group related alerts allowing experts to take action on only the causal alerts thus making resolution more efficient.

C. Alert prediction

Today's alerts are reactive in nature and do not provide sufficient time for corrective actions. We propose to use the past history of system components to predict future trends and patterns. We first use various prevalent forecasting algorithms [1], [3] to forecast the metrics for each component. However, for an accurate forecast, it is important to select an appropriate forecasting algorithm depending upon the various properties exhibited by the timeseries data. We use the trend, periodicity, mean levels, and seasonality of a metric to select the right algorithm for forecasting. Fig. 1(f) summarizes the algorithms used for different job properties. We generate proactive alerts upon observing abnormal values in the predicted metrics providing sufficient time for corrective measures.

III. Case Study

We present preliminary experimental results of the proposed ideas on a real-world case study. We analyzed 6-months run history of a slice of a batch system of a bank in the UK having 25 business processes and 404 components.

- We recommended thresholds for 37% of the components by identifying patterns and changes in behavior. We identified 83 jobs with temporally changing behavior and 68 jobs with at least one changepoint. The temporal dimensions were a mix of day of week, day of month, weekday, weekend, etc.
- In order to eliminate redundancies, we generated 72 aggregation rules with group sizes ranging from 2 to 5. The large group sizes consisted of jobs that were dependent on the same data or vendor feed. The aggregation rules were able to aggregate 5315 generated alerts into 2342 alerts resulting in a 56% reduction in alert volume.
- We then utilized the 6-month run history to predict and generate proactive alerts over the next 20 days. We compared our prediction with the actual generated alerts. 759 actual alerts were generated out of which we were able to predict 652 resulting in an accuracy of 86%. The 14% missed alerts arose due to external factors. We are addressing this issue as part of our future research work.

IV. Future Work and Conclusion

There are several issues in the space of alert management that we plan to address in the future. (1) Suppression of duplicate alerts is important to ensure quick identification of legitimate alerts. (2) Identifying the right monitoring frequency is critical to ensure high quality and timely alert generation. (3) Prioritization of alerts is important to ensure that critical issues are resolved quickly. (4) Providing quantitative and qualitative validation using precision and accuracy is required to compare our configurations with the manually set configurations.

Effective monitoring and event management solutions are crucial to ensure healthy enterprise IT systems. In this paper, we present solutions to set correct configurations, aggregate redundant alerts and generate proactive alerts. We articulate different aspects of the solution space and present initial ideas. We present preliminary results of the proposed solutions through a real-world case-study.

References

[1] G. Box and G. Jenkins. Time series analysis: Forecasting and control. *Holden-Day*, 1976.
[2] L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
[3] C. Holt. Forecasting trends and seasonal by exponentially weighted averages. *International Journal of Forecasting*, 20(1):5–10, March. 2004.
[4] http://www-01.ibm.com/software/tivoli/products/netcool omnibus/. Ibm's tivoli netcool.
[5] G. Miao, L. E. Moser, X. Yan, S. Tao, Y. Chen, and N. Anerousis. Generative models for ticket resolution in expert networks. In *16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.