Event Management and Monitoring Framework for HPC Environments using ServiceNow and Prometheus

Nitin Sukhija
Department of Computer Science
Slippery Rock University of Pennsylvania
Slippery Rock, PA
nitin.sukhija@sru.edu

Elizabeth Bautista
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
ejbautista@lbl.gov

Owen James
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
o1james@lbl.gov

Daniel Gens
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
dygens@lbl.gov

Siqi Deng
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
siqideng@lbl.gov

Yulok Lam
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
yllam@lbl.gov

Tony Quan
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
twquan@lbl.gov

Basil Lalli
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA
bdlalli@lbl.gov

ABSTRACT

The challenge of monitoring and event response management of a high performance computing facility grows significantly as the facilities employs and orchestrates more complex and heterogeneous systems and infrastructure. As the computational components encompassing the HPC facility system increases, the computational staff experiences rise in alert fatigue due to the false alarms and noise related to the similar events generated by monitoring tools. The National Energy Research Scientific Computing Center (NERSC) at the Lawrence Berkeley National Laboratory (LBNL) has begun to address the issues of duplication of alerts and alert remediation. However, more automation and integration is needed for collecting, aggregating, correlating, analyzing, managing and visualizing the scale of events that will be generated by the emergent hybrid computing infrastructures. In this paper, we present an event management and monitoring framework that addresses the operational needs of the future pre-exascale systems at the Lawrence Berkeley National Laboratory's National Energy Research Scientific Computing Center (NERSC). The framework integrates the Operations Monitoring and Notification Infrastructure (OMNI) at NERSC with the Prometheus, Grafana and ServiceNow platforms to help identify, diagnose, and resolve incidents in real-time, as well as conduct more thorough post-incident reviews enabled by the intuitive dashboards that provides a single pane of glass console for an efficient operations management and real-time proactive monitoring.

MEDES '20, November 2–4, 2020, Virtual Event, United Arab Emirates

ACM ISBN 978-1-4503-8115-4/20/11…$15.00
https://doi.org/10.1145/3415958.3433046

CCS CONCEPTS

• **Information systems → Data centers**; • **General and reference →** *Design*; • **Computing methodologies →** *Machine learning approaches*; • **Hardware → Failure recovery, maintenance and self-repair**;

KEYWORDS

Event Response; Data Monitoring; ServiceNow; Prometheus; Grafana; Visualization

ACM Reference Format:
Nitin Sukhija, Elizabeth Bautista, Owen James, Daniel Gens, Siqi Deng, Yulok Lam, Tony Quan, and Basil Lalli. 2020. Event Management and Monitoring Framework for HPC Environments using ServiceNow and Prometheus. In *12th International Conference on Management of Digital EcoSystems (MEDES '20), November 2–4, 2020, Virtual Event, United Arab Emirates.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3415958.3433046

1 INTRODUCTION

As we progress towards the next milestone of exascale supercomputing, the complexity and heterogeneity of the computational center infrastructure enabling such exaflops peak capacity will also increase substantially. Moreover, with this growth in the infrastructure complexity, the scale of events generated by these systems tend

to grow exponentially, thus increasing the management and security challenges of the computational facility handling these events [21]. This challenge is exacerbated by number of unpredictable and highly dynamic factors, such as, 1) diverse size, data processing, cloud workloads, virtualization requirements of the data center; 2) multiple environmental considerations of a data center ,such as, temperature, air conditioning, power distribution unit status, humidity, air flow etc.; and 3) numerous server and storage system metrics, such as, CPU, RAM, Power consumption and many more factors. To address the above-mentioned challenges, several monitoring solutions have been successfully proposed and deployments by modern computational centers to successfully manage and monitor these complex distributed high-performance computing environments [24].

Given the inexorable drift of mandating the utilization of automation and orchestration systems for anticipating diverse issues with heterogeneous computing system components operating under dynamically changing production environments, the management of these computational facilities is mounting in complexity. The new monitoring solutions currently deployed and proposed for the post peta-scale computational centers consider scalability and maintainability along with low-overhead as the most important design attributes of a successful management solution. Furthermore, an effective monitoring solution requires gathering heterogeneous information from the complex systems and sources encompassing a computational infrastructure, analyzing such extreme-scale data, and compiling a plan of action to respond to incidents in near-real time with minimal response time. Thus, operational efficiency in such rapidly changing high performance environments requires computational center staff to gather, manage and analyze exponential amounts of data [28]. The computing facilities now face thousands or even millions of events occurring across their infrastructure per day, which translates to extreme operations staff fatigue and hardships in effectively prioritizing events and in separating signals from the noise. This translates into the need of integration of an event management component to the modern monitoring solutions in order to decrease the noise related to similar events generated by multiple monitoring tools and to enable correlation of the events to facilitate actionable alerts and incidents via use of some predictive modeling techniques [33] [27].

With the proliferation in the hybrid computing models and orchestration of many complex services encompassing modern computational center operations, the collection of computing infrastructures health data and metrics along with the archival of the gathered data in real-time is of paramount importance to reduce response time or downtime due to the physical and the digital threats. Currently, there exist many data center management and monitoring tools integrated with an array of complexities and equipped with advanced alerting capabilities. However, there exist very few tools that significantly stress on a holistic view of the computation center monitoring by gathering diverse data in different formats from computing environments and by proactively responding to the threats and intrusions in real-time by providing to up-to-minute information via dashboard views of the whole center operations. Therefore, with the continuing growth in the scale and complexities of the computational center hybrid architecture and policies, the existing monitoring approaches involving Nagios [25], Spiceworks [14], Icinga [5], or Zabbix [16] will soon grow to be obsolescent. The comprehensive monitoring solutions facilitates a low-overhead, scalable and integrated operational data collection and analytics infrastructure for ingesting diverse data and metrics, correlating events, and providing real-time incidence reporting and management, thus aids in achieving operational excellence in the computational centers. Recently, the Lawrence Berkeley National Laboratory's (hereafter referred to Berkeley Lab) National Energy Research Scientific Computing Center (NERSC) have developed and deployed an Operations Monitoring and Notification Infrastructure, OMNI to gather near real-time data and metrics pertaining to the health of computing systems as well as to archive the extreme scale data to facilitate monitoring of such highly complex orchestration and automation platforms. The OMNI infrastructure built on open-source technologies, such as the Elastic Stack [3][20] enables storing of the real-time streaming time-series monitoring data from multiple sources, including computing systems at NERSC and its supporting computational infrastructure, environmental sensors, mechanical systems and more. In this paper, we propose an automated event response management framework that integrates the OMNI infrastructure with Prometheus [12], Grafana [4] and ServiceNow [13] platforms for enabling proactive alert monitoring, predictive intelligence to reduce events noise, root cause analysis, data visualizations and single pane of glass for performance management. The proposed infrastructure will aid in reducing Mean Time to Repair (MTTR) by understanding the root cause of the operational issues in real-time via actionable alerts and in improving the service availability of the computational center by proactively eliminating outages and other critical issues at NERSC. The design of the proposed comprehensive monitoring and event response framework is motivated by the emergent trends of efficient monitoring and scalable management of the current and future heterogeneous computing systems, such as, Perlmutter at NERSC [11].

The rest of the paper is organized as follows. A review of related work and extreme-scale computational center management challenges is presented in Section 2. The design and organization of the comprehensive event response and monitoring framework is described in Section 3. The event Management and monitoring workflows are summarized in Section 4 followed by the conclusions and possible future directions in Section 5.

2 BACKGROUND AND RELATED WORK

With the growth in the efficiency of the data centers, the management and monitoring these highly complex hybrid environments is becoming increasingly difficult and burdensome. As the margin for error in the current and future computational facilities is becoming dangerously thin, the traditional monitoring solutions are limited by their ability to scale and to provide real-time, precise, and fine grain monitoring of the computing infrastructure [17]. Moreover, as the cloud computing and hyperscale innovations and complexities are displacing the traditional computational systems, an efficient monitoring solution for the emergent modern data centers requires integration of the right mix of tools and technologies that facilitate not only monitoring but also identifying and prioritizing the infrastructure service and security events to proactively respond

or altogether avoid potential incidents [31]. Thus, superior management of the computational centers necessitates employing a monitoring and event management platforms that incorporate the key design considerations, such as:

Real-time Event Collection: It is imperative to collect various events data and metrics pertaining to not only the health of the computing systems and the underlying infrastructure but also pertaining to the environmental factors encompassing a computational center. These diverse event datasets and metrics pose the volume, velocity, variety, and veracity challenges for a computational center to provide an efficient real-time monitoring solution. To address the above-mentioned issues, there is a need of scalable mechanisms for collecting high volume events stemming from various sources, such as system logs, job and network statistics, environment data and more. Thus, monitoring and event management platforms should be capable of collecting extreme-scale events each day which translates to exceptionally high event ingestion rates per second which poses significant investment and validation challenges.

Real-time Event Correlation and Analysis: With the increase in the complexity of the modern computational centers, the production of event data has also reached an exponential scale. In addition to event gathering and collection, the interpretation of this highly disparate event data is equally important to perform alerting, notification, and correlation in order to prioritize and respond to the potential critical faults or service level exceptions. The event notifications can grow at a spamming rate spawning multiple alerts for the same event. Thus, grouping and inhibition mechanisms needs to be built in the monitoring and event management platform to reduce the noise generated by multiple events and to correlate the events for acting on gathered event data to respond in real-time with corrective actions based on the severity of the alerts.

Real-time Reporting: One of the key goals of a monitoring and event management platform is to facilitate and speed up the decision-making process of the computational facility or organization by reporting accurate, real-time information in a useful and meaningful way. However, providing a consolidated view of the summarized analysis results based on aggregation and correlation of the top alerts in real-time for different organizational levels can be extremely challenging. Therefore, the monitoring and event management platform should facilitate a single pane of glass dashboards that will aid the computational staff to transform infrastructure events into actionable alerts and associated incidents for performing root cause analysis in near real-time leading to informed decisions, thus avoiding errors, inefficiency, downtime and wasted capacity of the computational facility.

Scalability: Today's computational centers are comprised of heterogeneous nodes, network, application, server, and virtualized infrastructure elements. With the number of the events generated by the diverse components comprising the computational center growing exponentially, the need for greater efficiency, thus scale-out technologies are becoming more popular than the scale- up approaches. The scale-out approaches currently deliver greater application volumes and larger storage capabilities at a reduced expense, increased availability, and lowered management tasks in comparison to the scale-up approaches involving a monolithic machine to monitor and manage computational center.

Automation and Integration: With an exploding volume of unstructured event "big data" generated from all underlying infrastructure, network and applications components of the today's modern computational centers, there is a tremendous pressure on the center's operations staff to collect, normalize, store, and correlate events to keep the computation center agile and to ensure health of the infrastructure optimized via superior decision making metrics and alerting rules. Thus, automation of many routine operations management activities is required to reduce load on the operations staff and to achieve more effective monitoring and event management. The demand for agility and deployment at scale requires the monitoring and event management platforms to employ application program interfaces (API) for integrating facilities and IT operations, thus facilitating effective communication and collaboration leading to optimized performance, capacity, and availability.

High Availability and Operational Intelligence: The purpose of the monitoring and event management platform is to aid in achieving operational intelligence by employing various operational metrics and machine learning approaches to identify and prioritize computational center events and to proactively aid in understanding performance issues and in capacity planning. Moreover, in addition to resolving critical infrastructure performance issues, the monitoring and event management platform also aids in achieving high availability of the computational center services by detecting the root cause incidents and by triggering automated remediation solutions.

Over years various monitoring solutions have been proposed and deployed in the computational environments. Moreover, many studies involving event management solutions have been researched and reported. However, only a few studies published illustrate the integration of the real-time monitoring and event management to provide a comprehensive framework based on the above considerations. Over 15 years, Nagios tool [8] has been utilized by many organizations to monitor the status of the network devices and their services and to aid in resolving critical issues related to them. Even though Nagios is the leading solution for monitoring, visualizing, and alerting critical problems, the static version of Nagios is limited by the scalability and its management becomes extremely difficult in dynamic environments. The Spiceworks Network Monitoring tool [14] is like Nagios in terms of limited scalability in spite of featuring extraordinary dynamic dashboards. Moreover, Spiceworks tool also lacks the support for Simple Network Management Protocol (SNMP) 3.0 [22]. Some other top monitoring tools include Paessler PRTG Network Monitor [10], Zabbix [16] and Icinga [5], which comes equipped with many customization's, such as good Web User Interface and API support, but are still limited in performance and efficiency when employed in large scale deployments comprising of thousands of devices and sensors.

There are only few recent projects which enlists integration of event management and monitoring infrastructure. The Service Management Project [18] is one of the projects which replaced Remedy with the ServiceNow infrastructure for the event management and monitoring of the servers available and services offered at the CERN IT Computer Centre. The researchers in [29] proposed an event monitoring system, named Trumpet that utilizes the CPU resources and end-host programmability to monitor network events,

report and identify root cause of congestion in network at millisecond timescales. However, the authors provides only the design of an event manager with its applicability in real data center environments. Furthermore, a new multilayer node event processing (MNEP) approach for enhancing data center energy efficiency along with the event management monitoring of the physical infrastructure was presented in [28]. Along with above mentioned studies and projects, several projects, such as, Trinity Monitoring infrastructure [23], OpenLorenz [9] and others [30] have also been deployed for facilitating scalable monitoring solutions. In our previous work [34], we proposed a comprehensive scalable monitoring solution. To best of our knowledge, in comparison to the above mentioned related solutions, the proposed monitoring and event management framework gathers data from a Prometheus data source, collects the data in OMNI and creates a tracking mechanism in ServiceNOW. This delivers a complete scalable real-time monitoring and event management solution enabling operational intelligence by collecting diverse events data, normalizing alerts, storing health and operational metrics, correlating events and utilizing machine learning approaches for discovering the root causes of the computational center critical issues and for proactive decision making. Our proposed monitoring and event management framework will facilitate real-time operational intelligence , thus will aid in optimizing performance, availability and capacity planning of NERSC's current Petascale computing systems infrastructure and will also aid in planning our future systems deployments.

3 DESIGN AND ARCHITECTURE OF EVENT MANAGEMENT AND MONITORING FRAMEWORK

In this section we detail the design and architecture of the scalable infrastructure that will be employed to provide monitoring and event management of the current and future heterogeneous computing system, network, and applications at NERSC computational facility. The comprehensive event management and monitoring framework proposed in this paper (as shown in Figure 1) integrates various open source platforms, such as Prometheus [12], Grafana [4], Kafka [1], and ServiceNow [13] with the existing OMNI infrastructure for enabling automatic event collection, event correlation, alert prediction, alert noise reduction, root cause analysis, operational metrics reporting, and single pane view dashboards for enhancing operational intelligence. The proposed framework aids in reducing Mean Time to Repair (MTTR) by providing alert insight analysis and resolution via machine learning techniques, thus facilitates high service availability of the computational center by prioritizing alerts and proactively resolving critical issues at NERSC.

With the growth of the number of platforms and components encompassing modern petascale and post-petascale computational centers, the scale of events generated and reported across their infrastructure per day will rise exponentially. This translates to growth in incidents and software management issues leading to operations staff fatigue and hardships in effectively prioritizing events, in filtering alerts from the noise and other alert remediation challenges, thus degrading response to critical computational facility issues. Therefore, to address the above challenges, there is a need for system integration and automation infrastructure which enables automation of event monitoring, reduction of noise related to similar events, correlation of the generated events for facilitating automatic prediction and prevention of operational performance degradation issues. For this reason we investigated automated solutions like Prometheus, Grafana, Kafka, and ServiceNow platforms to facilitate orchestration of the services and system deployments, automation and correlation analysis of streaming data, and adaptation and resolution of the alerts for identification of the core operational inefficiency issues.

Given the dynamicity of the computational environments, NERSC is investigating platforms which will be best suited for monitoring the computing systems and infrastructure in rapidly changing ecosystems. One such tool is Prometheus which is open source and highly adaptive to monitor such extreme scale and dynamic computational centers. The tool supports SNMP, advanced alerting mechanisms better than Nagois and specifically designed to monitor and health metrics of the hybrid computing infrastructure built upon containers and services with no expense of any computing resources.

A. Prometheus:

Prometheus developed in 2012, is an open source monitoring solution that was released in 2016 by SoundCloud. Prometheus facilitates a multi-dimensional data model, collects all data from various data sources as stream of timestamped values and stores the time series data which can be identified by metric name and key/value pairs. Counter, Gauge, Histogram and Summary are four type of metrics that are offered by the Prometheus client libraries. Moreover, in comparison to most of the monitoring tools such as OpenTSDB , Nagios, Sensu, the Prometheus platform also provides a powerful and flexible querying and real-time alerting by employing PromQL, a functional query language that utilizes pull model over HTTP and enables user to select and aggregate real-time metrics from the time series database. Moreover, the queried data can be visualized as graphs or tables over browser using the HTTP API. Moreover, the alertmanager component of the Prometheus platform comes handy with many more features than just silencing of alerts as performed by Nagios tool. Furthermore, Prometheus enables inspection of the innards of the applications (whitebox monitoring) in comparison to tools such as Nagios, Sensu which are only allows inspecting the state of the host or service not aids in understanding how the host/service reached that current state (blackbox monitoring). The Prometheus ecosystem comprises of the following main components: 1) Prometheus server that enables the scraping of the metrics from the jobs and collecting the multi-dimensional data as numeric time series; 2) Client Libraries that implements the four Prometheus metric types and enables the exposure of internal metrics via HTTP endpoint for the matching application instance; 3) Alertmanager that facilitates handling of alerts sent by Prometheus server and aids in grouping (single notification for similar alerts), Inhibition (suppressing alert notification which is unrelated to real issue), Silencing (muting alerts for specified time period) , and deduplication(reduce the number of same alerts fired over course of a single incident and prevent alert fatigue) ; 4) Exporters enable the dissemination of existing time series metrics from the third-party systems such as, HAProxy, StatsD, Graphite as Prometheus metrics; and 5) Grafana facilitates an analytics platform by pulling the

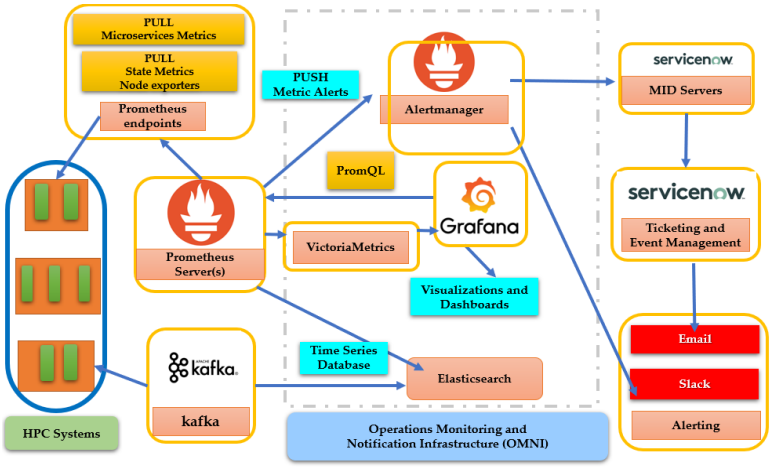

Figure 1: Design and Architecture of the Event Management and Monitoring Framework.

metrics by querying Prometheus and enabling visualization of data and alerts via intuitive Dashboards. The monitoring time series data pertaining to the NERSC computational infrastructure events gathered and aggregated by Prometheus will be stored in OMNI's Elasticsearch platform, and will be used to provide historical trends and metrics for capacity planning, and to display dashboards comprising graphs and visualizations for gaining operational insights and enhancing decision making.

B. Operations Monitoring and Notification Infrastructure (OMNI)

OMNI is a flexible big data solution, a warehouse to collect, manage and analyze data related to monitoring of extreme scale computing systems [19]. This infrastructure facilitates a single location for storing the heterogeneous datasets and is backed by a scalable and parallel time-series database, Elasticsearch [3]. Examples of operational data include time series data from the environment (e.g., temperature, power, humidity levels, and particle levels), monitoring data (e.g., network speeds, latency, packet loss, utilization or those that monitor the filesystem for disk write speeds, I/O, CRC errors), and event data (e.g., system logs, console logs, hardware failure events, power events essentially anything that has a start and end time). Getting data from the various systems and sensors into Elasticsearch [3] occurs via RabbitMQ [35], a messaging broker that supports multiple messaging protocols and queuing. Data from multiple sources may be queued directly into a RabbitMQ queuing system with a json format. Alternatively, it may be first collected from a system via Collectd then parsed by Logstash [7] before being queued into RabbitMQ. We have implemented multiple RabbitMQ streams to differentiate different datasets, each type of dataset having its own queue. Using RabbitMQ, Logstash, and Elasticsearch, OMNI is able to ingest over 25,000 messages per second from heterogeneous and distributed sources in and out of the data center. Once ingested, Elasticsearch indexes the data for near real-time retrieval and querying. Data may be directly queried from Elasticsearch using the native RESTful APIs or using visualization and data discovery tools, such as Kibana [6] or Grafana. Moreover, this solution has high availability with minimal maintenance operations disrupting the functioning of the data collection infrastructure. As systems become larger, more heterogeneous, and more complex, existing monitoring methods will become unmanageable. Perlmutter, the 2020 system will have over 3 times the compute power of the current system, Cori, and contain a mixture of CPU-only and GPU-accelerated nodes. The facility now has more than 20,000 sensors with hundreds of sources of environmental, computation and event data from water flows and humidity readings, temperatures in different areas of the racks and center, sensors at the substation, the different levels of PDUâĂŹs available in the building, UPS/generator setup, compute nodes, filesystem servers, login nodes, gateway nodes, support nodes, Lustre and GPFS filesystems [32], syslogs and other logs.

C. ServiceNow

ServiceNow founded in 2003, provides a unified cloud computing platform for automating and transforming IT processes, such as, security operations, IT service, asset, business operations and event management and more. ServiceNow is a Platform-as-a-service provider (PaaS) [26] with core focus on fostering management of the incident, problem and change events pertaining to the IT infrastructure and services. ServiceNow implements a configuration management database (CMDB), which aids in providing an accurate and up-to-date records of organization's IT assets to effectively assist in asset, compliance and configuration management. The CMBD database assist in service impact analysis as it stores current and accurate information regarding all technical services in a Configuration Item (CI) corresponding to those services. Furthermore, the service mapping is tightly coupled with CMDB database to make it service-aware, where service maps employ discovery and infrastructure information in CMDB to create an accurate and complete tag based map of all applications, virtual systems, underlying network, databases, servers and other IT components that supports the service. Furthermore, the automated service mapping enables not only a user interface showing accurate service-level relationships but also updates the service maps in real-time, thus aids in avoiding irrelevant infrastructure data and in gaining faster insights and history of service topology. The event management

platform enabled by ServiceNow allows an easy integration with multiple monitoring tools, such as, Prometheus to receive events data for analysis and response. The operational metrics captured by the ServiceNow is used to reduce the Mean Time to Resolution (MTTR) by employing machine learning for root cause analysis and defining alert rules, thresholds, and remediation actions. Moreover, ServiceNow dashboards provide visualization of each service that aids in prioritizing service issues based on the impact and criticality.

NERSC computational facility is now transitioning to integrate the event management and monitoring platform in production environment to prepare for the upcoming next-generation supercomputing system. In preparation for the upcoming system, a new data collection network infrastructure facilitating VXLAN connectivity is being developed. Moreover, NERSC has also investigated many versions of the Kubernetes, for enabling orchestration and management of containers and for investigating the best management fit into the OMNI data collection model. The goal is to test and further the knowledge about NERSC-9's (N9) implementation before moving one step closer to the next generation supercomputer. Furthermore, the Prometheus collectors and exporters have already being deployed into the NERSC Global Filesystems (NGF) development system, Cori's ES login nodes and Cori's GPU cabinets [2]. More collaboration efforts are currently being focused on to extend this into the rest of Cori's external nodes, the new HPSS systems, the new community filesystem system, and other internal support systems. Along with Prometheus platform, the Grafana and ServiceNow platform are also being deployed and integrated into the OMNI infrastructure. Several event management and monitoring workflows utilizing the above-mentioned comprehensive platform have been implemented and tested for automating the alert management and remediation process's as well as for providing intuitive dashboards for enhancing operational intelligence and efficiency.

4 EVENT MANAGEMENT AND MONITORING WORKFLOWS

As the number of components comprising the computing infrastructure and their corresponding incidents and management issues evolve, system integration and automation is becoming more important. There is an increasing need to implement and streamline workflows utilizing the proposed event management and monitoring platform to simplify management of various interdependent processes, reduce operational costs, and increase staff efficiency. Due to advances in automation technology, these efforts are focused not only on triage and troubleshooting efficiency, but also on the remediation workflows. Alert remediation is becoming an important part of NERSC-9 operational monitoring and event management infrastructure. Both Event Management and Alert Remediation will be implemented using ServiceNow - an ITSM and ITOM solution with a MySQL database backend. At NERSC, ServiceNow is used as an Incident Management platform for users and staff. Given its versatility and custom API functionality, ServiceNow platform is a core component of NERSC-9 monitoring layout.

Leveraging the ServiceNow Platform for data management using Event Management and CMDB (Configuration Management Database), our automated workflow facilitates: 1) Auto Discovery - in conjunction with Netbox; 2) Intuitive Visual Reports - from incidents (tickets); and 3) Integration of the Prometheus with Promxy - a Prometheus proxy, a single API endpoint facing the user. The automated event response management and monitoring workflow (as shown in Figure 2) implements the following stages:

- High volume, high data rate storage: High Availability (HA) pair provides redundancy, performs deduplication from multiple sources. Moreover, VictoriaMetrics [15] facilitates an

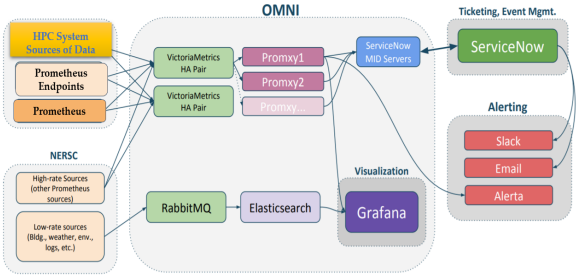

Figure 2: Automated event response and monitoring workflow: Multiple sources provide data from Prometheus endpoints or Kafka, where the data is transformed and can either go to Elasticsearch where it can be queried by Kibana or can go to vmagent that handles sending the information to various area for alerting through the ServiceNow MID server. The data then either automatically goes through an event management module in ServiceNow to handle opening a ticket or notifying various on calls of the alert. Further, through alertmanager, the data can go to slack for processing.

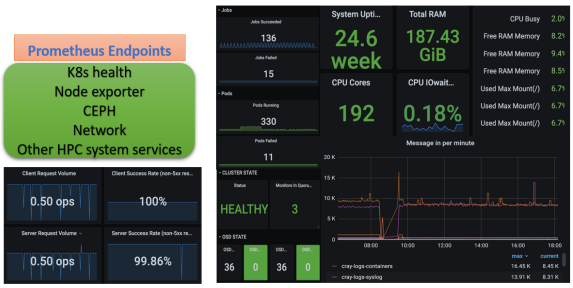

Figure 3: Sample data sources visualized in Kibana or Grafana.

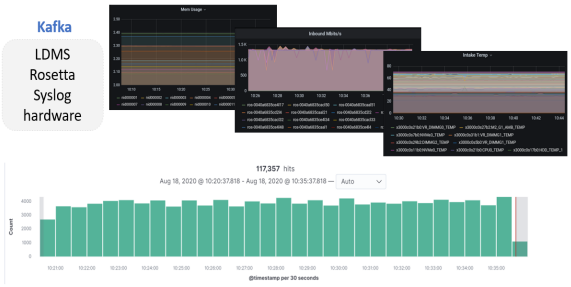

Figure 4: Additional data sources from Apache Kafka visualized.

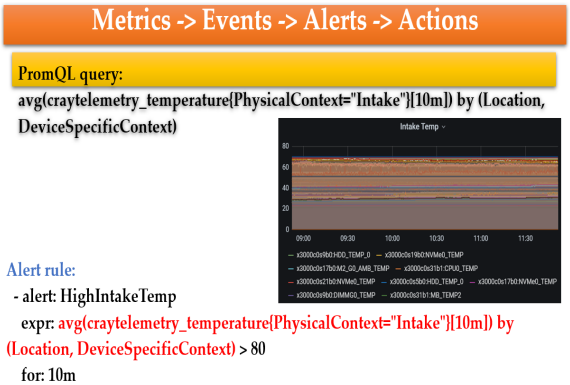

Figure 5: The system metrics are queried and are transformed into events that generate alerts when threshholds are met and actions are performed, such as, open a ticket, notify the on call, or perform an action to fix the problem.

open source time series database to collect, store and observe metrics.

- Promxy, a prometheus proxy, receives aggregated metrics: Alerting rules define alertable conditions and the Alertmanager sends to receivers.
- Promxy alerts are sent to ServiceNow via the Management, Instrumentation and Discovery (MID) server, and metrics are passed to Grafana platform for dashboarding and visualization (as shown in Figure 3 and Figure 4) as well as to Alerta.
- Alerts are transformed into ServiceNow (SN) "Events", which are correlated and grouped into SN "Alerts", (as shown in Figure 5 to Figure 9) which then trigger automated response actions (incidents, notifications, etc.)

In the development and deployment stages of NERSC-9 event management and monitoring infrastructure, we are planning to employ the above described workflow to automate many operational features such as the following:

1. Alert processing for IBM Elastic Storage Servers (ESS) that provide NERSC Community File System (CFS):
There are currently 14 ESS in 7 pairs, each pair collectively manages half of 8 large disk arrays, and acts as each other's high availability partner in case of failure, and a management server. There is also a development management server and a number of development nodes. Each host is monitored with a basic PING check as well as a custom general health check which reports disk, enclosure, recovery group, and other alarms. If the alert is for a down production node, the workflow will proceed to open a ServiceNow Incident with a high priority for 24x7, down dev nodes will result in a low priority ServiceNow Incident for 8x5 support. If the alert is a health check alert for a failed disk, there will be some extra steps after opening a medium priority ServiceNow Incident. Failed disks will need a vendor case, so the workflow will first collect some data through the management server, namely the disk's serial number, the output from a diagnostic command, and a smart data file. The file will be attached to the ServiceNow Incident, and the diagnostic command output and the serial number will be noted in the Incident comments. The vendor case will need to be created manually

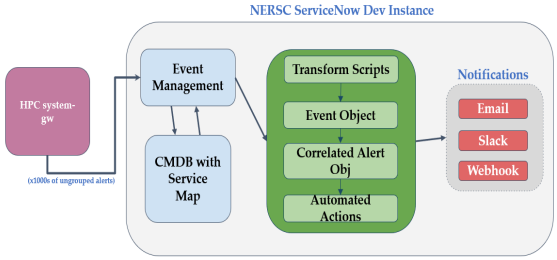

Figure 6: The event management and monitoring workflow map once it hits ServiceNow. The CMDB provides a service map of nodes and other assets and depending on their function and what happened, the event management determines what to do, who to notify via e-mail, notify via slack or activate a webhook to open a ticket in ServiceNow.

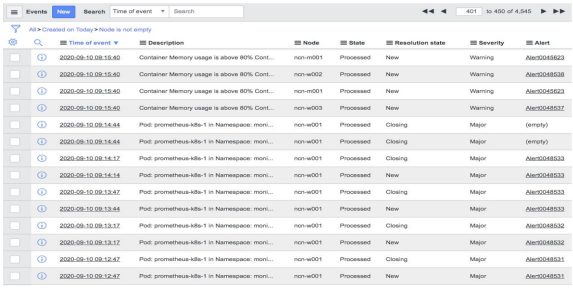

Figure 7: A list of events in ServiceNow obtained from the data originating from Prometheus.

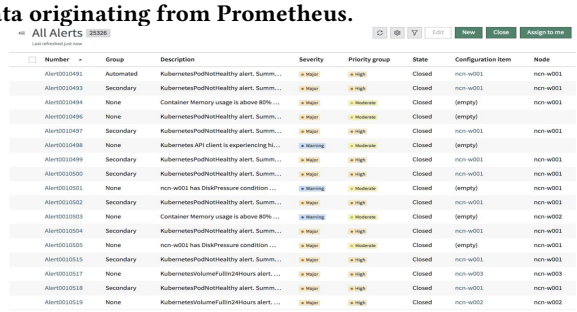

Figure 8: View displaying Alerts in ServiceNow as processed by the Alertmanager.

by an engineer, however we are investigating the possibility of relying IBM support portal API if it's provided. For other health check errors, the workflow will run a script that will evaluate CFS health for possible issues, and will act according to that report.

2. Gathering logs for down or drained HPC compute nodes:
Given a very large number of available compute nodes, alarms for down nodes make up the majority of our routine alarms. There are multiple steps to gather logs and troubleshooting information that will help categorize the node failure for any given compute node. The compute node remediation workflow is expected to be able to gather that information, categorize the failure, and proceed to predetermined scenarios for this category - for most node failure, the action will be to reboot the node, but there are a number of exceptions to that. If the failure doesn't fall in any given category,

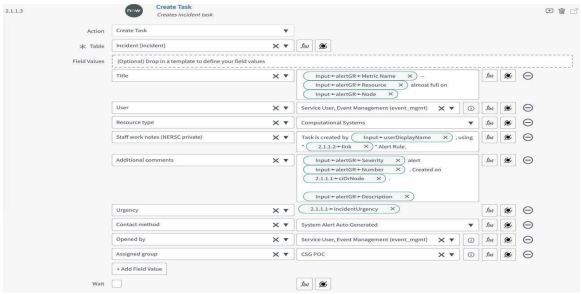

Figure 9: Auto creation of a ticket in ServiceNow once an alert is identified.

the workflow will submit a ServiceNow Incident, assigned to the appropriate group. All collected logs will be attached or noted in the Incident for an engineer's consideration.

3. Remediation for disk space, load and service health checks for various NERSC servers:

A lot of these alerts are repetitive and will have clear-cut guidelines on resolving the issue, for example a host where /var/log is always filling up, or a system service that often needs to be restarted following a crash. These types of issues require very little triage work, so their remediation can certainly be automated in ServiceNow.

5 CONCLUSION

The event management and monitoring framework presented in this paper facilitates the proactive monitoring, event collections and correlation, alerts and metrics visualizations, orchestration and remediation and real-time automated root cause analysis and response management in the highly changing heterogeneous cluster deployments and services using an integrated solution comprising of the OMNI, Prometheus, Grafana and ServiceNow Platforms. Moreover, the alert remediation enabled by the automated workflows enabled by the proposed framework will potentially reduce drastically the number of incidents that require operational staff intervention or need rerouting to the appropriate groups. Furthermore, the integrated framework will considerably speed up the time to resolution for the potential critical computational center issues with the proposed automated remediation workflows deployments, as well as for other issues that requires the data and the logs to be gathered automatically.

6 ACKNOWLEDGMENTS

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DEAC02-05CH11231.

REFERENCES

[1] [n. d.]. Apache Kafka. https://kafka.apache.org/
[2] [n. d.]. Cori: NERSC's newest supercomputer. https://www.nersc.gov/users/computational-systems/cori/
[3] [n. d.]. Elasticsearch: Distributed, RESTful Engine. https://www.elastic.co/products/elasticsearch
[4] [n. d.]. Grafana. https://grafana.com/
[5] [n. d.]. Icinga. https://icinga.com/
[6] [n. d.]. Kibana:Your Window into the Elastic Stack. https://www.elastic.co/products/kibana
[7] [n. d.]. Logstash: Centralize, Transform and Stash Your Data. https://www.elastic.co/products/logstash
[8] [n. d.]. Nagios. https://www.nagios.org/
[9] [n. d.]. OpenLorenz: Web-Based HPC Dashboard and More. https://software.llnl.gov/repo/#/hpc/OpenLorenz
[10] [n. d.]. Paessler PRTG Network Monitor. https://www.paessler.com/prtg
[11] [n. d.]. Perlmutter: NERSC's Next Supercomputer. https://www.nersc.gov/systems/perlmutter/
[12] [n. d.]. Prometheus. https://prometheus.io/
[13] [n. d.]. ServiceNow. https://www.servicenow.com/
[14] [n. d.]. Spiceworks Network Monitoring Management Software. https://www.spiceworks.com/
[15] [n. d.]. VictoriaMetrics. https://victoriametrics.com/
[16] [n. d.]. Zabbix. https://www.zabbix.com/
[17] Erika Abraham, Costas Bekas, Ivona Brandic, Samir Genaim, Einar Broch Johnsen, Ivan Kondov, Sabri Pllana, and Achim Streit. 2015. Preparing HPC applications for exascale: challenges and recommendations. In *2015 18th International Conference on Network-Based Information Systems*. IEEE, 401–406.
[18] R Alvarez Alonso, G Arneodo, O Barring, E Bonfillou, et al. 2014. Migration of the cern it data centre support system to servicenow. In *Journal of Physics: Conference Series*, Vol. 513. 062032.
[19] Elizabeth Bautista, Melissa Romanus, Thomas Davis, Cary Whitney, and Theodore Kubaska. 2019. Collecting, Monitoring, and Analyzing Facility and Systems Data at the National Energy Research Scientific Computing Center. In *2019 International Conference on Parallel Processing*. ACM, in press.
[20] Elizabeth Bautista, Cary Whitney, and Thomas Davis. 2016. Big data behind big data. In *Conquering Big Data with High Performance Computing*. Springer, 163–189.
[21] Keren Bergman, Shekhar Borkar, Dan Campbell, William Carlson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, Jon Hiller, et al. 2008. Exascale computing study: Technology challenges in achieving exascale systems. *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep* 15 (2008).
[22] J Davin, JD Case, M Fedor, and ML Schoffstall. 1989. Simple network management protocol (SNMP). (1989).
[23] Adam DeConinck, A Bonnie, K Kelly, S Sanchez, C Martin, M Mason, James M Brandt, Ann C Gentile, Benjamin A Allan, Anthony Michael Agelastos, et al. 2016. *Design and Implementation of a Scalable Monitoring System for Trinity.* Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
[24] Gabriel Iuhasz and Dana Petcu. 2019. Monitoring of Exascale data processing. In *2019 IEEE International Conference on Advanced Scientific Computing (ICASC)*. IEEE, 1–5.
[25] David Josephsen. 2007. *Building a monitoring infrastructure with Nagios*. Prentice Hall PTR.
[26] Michael J Kavis. 2014. *Architecting the cloud: design decisions for cloud computing service models (SaaS, PaaS, and IaaS)*. John Wiley & Sons.
[27] Antonio Libri, Andrea Bartolini, and Luca Benini. 2018. Dig: Enabling out-of-band scalable high-resolution monitoring for data-center analytics, automation and control. In *2nd International Industry/University Workshop on Data-center Automation, Analytics, and Control (DAAC 2018)*. Data-center Automation, Analytics, and Control (DAAC).
[28] Vojko Matko and Barbara Brezovec. 2018. Improved data center energy efficiency and availability with multilayer node event processing. *Energies* 11, 9 (2018), 2478.
[29] Masoud Moshref, Minlan Yu, Ramesh Govindan, and Amin Vahdat. 2016. Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 129–143.
[30] Dmitry A Nikitenko, Sergey A Zhumatiy, and Pavel A Shvets. 2016. Making large-scale systems observable-another inescapable step towards exascale. *Supercomputing Frontiers and Innovations* 3, 2 (2016), 72–79.
[31] Sam Sanchez, Amanda Bonnie, Graham Van Heule, Conor Robinson, Adam DeConinck, Kathleen Kelly, Quellyn Snead, and J Brandt. 2016. Design and implementation of a scalable hpc monitoring system. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 1721–1725.
[32] Frank B Schmuck and Roger L Haskin. 2002. GPFS: A Shared-Disk File System for Large Computing Clusters.. In *FAST*, Vol. 2.
[33] John Shalf, Sudip Dosanjh, and John Morrison. 2010. Exascale computing technology challenges. In *International Conference on High Performance Computing for Computational Science*. Springer, 1–25.
[34] Nitin Sukhija and Elizabeth Bautista. 2019. Towards a Framework for Monitoring and Analyzing High Performance Computing Environments Using Kubernetes and Prometheus. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 257–262.
[35] Alvaro Videla and Jason JW Williams. 2012. *RabbitMQ in action: distributed messaging for everyone*. Manning.