Proactive RCA with Vitrage, Kubernetes, Zabbix and Prometheus

Anna Reznikov (Nokia CloudBand)

Dr. Liat Pele (Nokia CloudBand)

VITRAGE

an OpenStack Community Project

We're going to talk about…

- Reactive vs. Proactive monitoring & RCA
- Why Vitrage?
- Newly added data sources
- Demo
- Other ways of using Vitrage for Proactive RCA
- Future plans
    - Diagnostic actions
    - Bell Labs change detection algorithm

VITRAGE
an OpenStack Community Project

Reactive vs. Proactive Monitoring and RCA

Reactive:

- Application execution is stopped
- Errors are not prevented
- Warnings are not correlated

Proactive:

- The end user is not affected
- Application execution is not interrupted
- Errors are prevented (manually or automatically)
- Warnings are correlated
- Diagnostics and preventive actions are triggered

Introduction to Vitrage – Root Cause Analysis Service

Project background

- Started 2.5 years ago at Nokia, during the Mitaka cycle
- Became an official project 6 months later
- First official version – Newton
- ~10 active contributors in the Queens release

Introduction to Vitrage

Vitrage is the OpenStack Root Cause Analysis project:

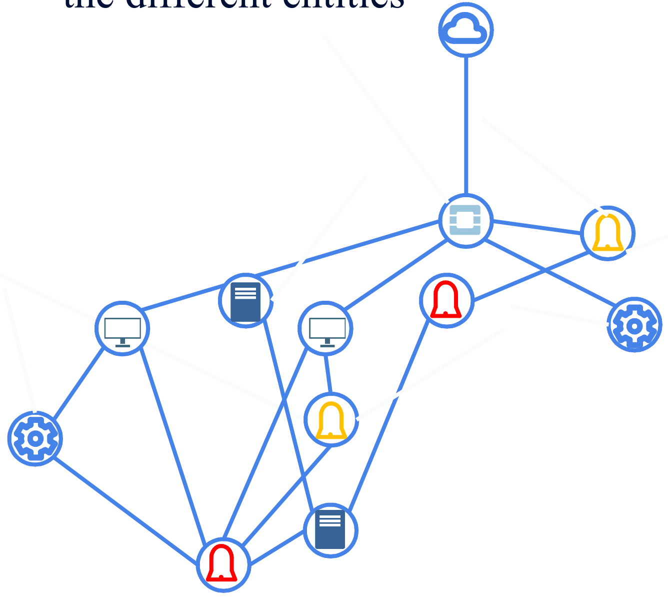
- Holistic & complete view of the system structure
- Organize OpenStack alarms & events
- Deduce alarms and states
- Root Cause Analysis
- Passing information through Vitrage notifiers

https://docs.openstack.org/vitrage/latest/

What does Vitrage Include?

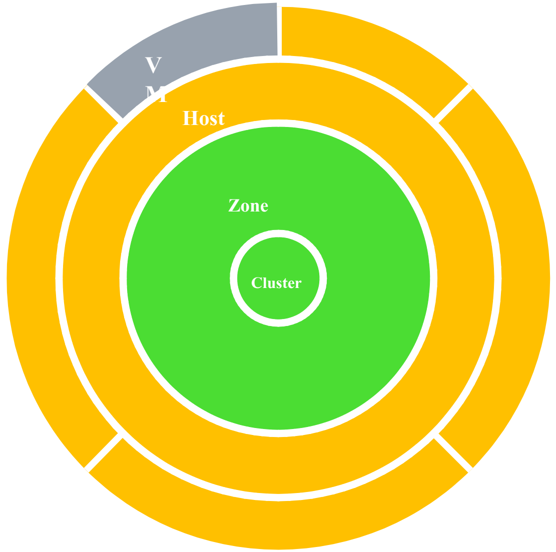
Entity Graph

Represents the relationships between the different entities



Topology Graph

Represents system health, allowing focusing on failing resources



Visualized RCA

Root cause analysis between alarms in the graph

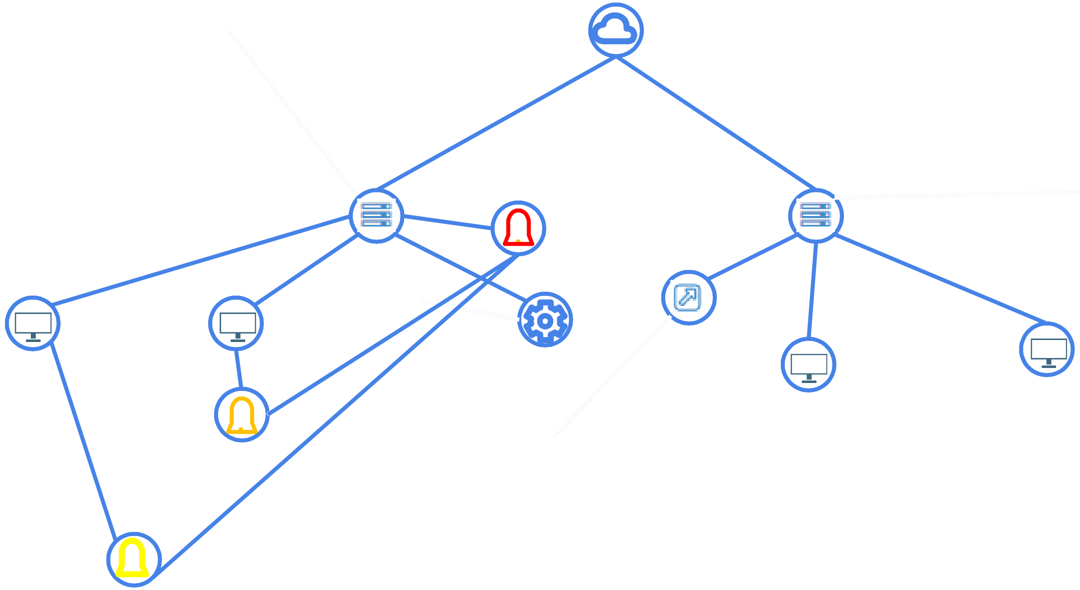
Using Vitrage for Proactive Root Cause Analysis

- Decisions based on information from several data sources
  - Monitors on both physical , virtual and applications layers
- Decisions based on RCA

  - Changing states
  - Deducing alarms
  - Cause actions

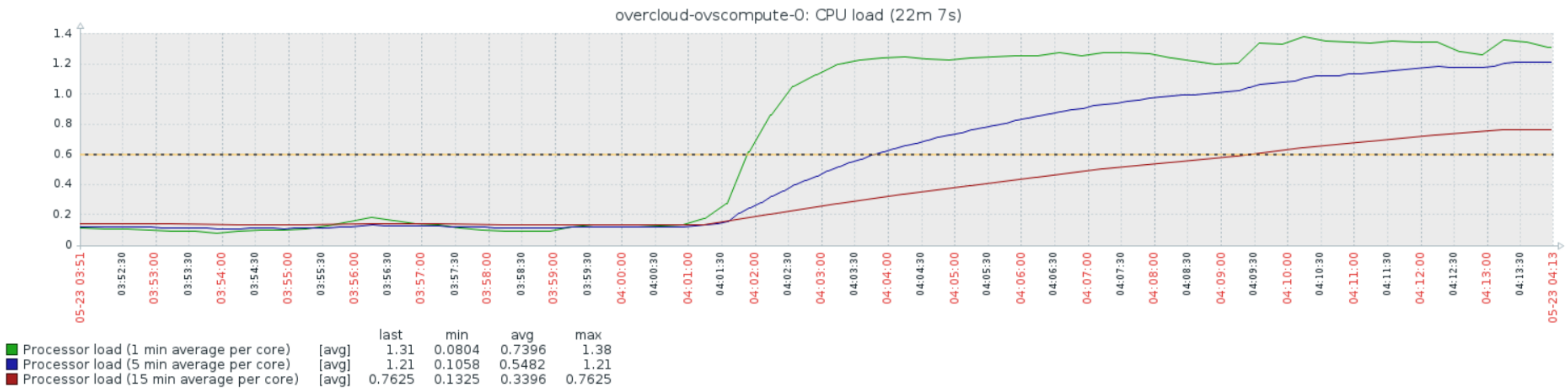
Using Vitrage for Proactive Root Cause Analysis

- Automatic corrective actions using Mistral
  - Auto evacuation



Using Vitrage for Proactive Root Cause Analysis

- Use monitoring systems’ predictive capabilities
  - Zabbix and Prometheus



		last	min	avg	max
🟩 Processor load (1 min average per core)	[avg]	1.31	0.0804	0.7396	1.38
🟦 Processor load (5 min average per core)	[avg]	1.21	0.1058	0.5482	1.21
🟥 Processor load (15 min average per core)	[avg]	0.7625	0.1325	0.3396	0.7625

VITRAGE
an OpenStack Community Project

Using Vitrage for Proactive Root Cause Analysis

- Execute diagnostic actions
  - Expensive tests – example: memory scan
  - On demand monitoring

  - More details later on

Newly added data sources to Vitrage

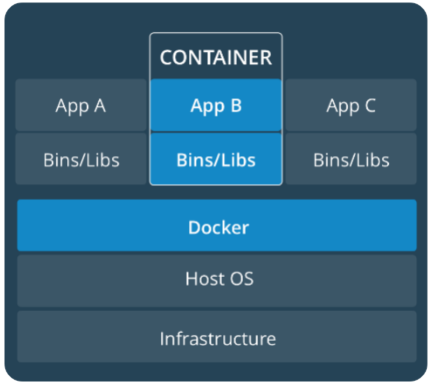
kubernetes

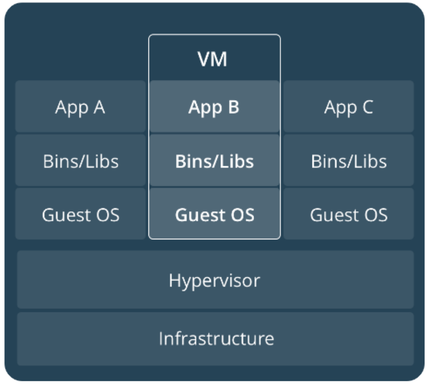
Prometheus

VITRAGE
an OpenStack Community Project

Tech stack of cloud-native VNFs

Docker and Kubernetes



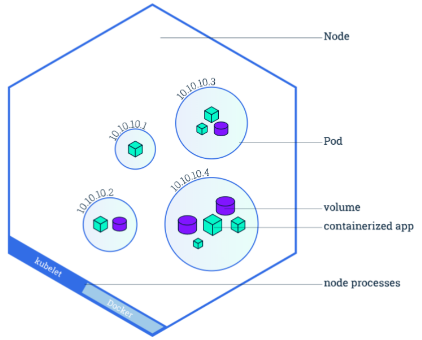


source: docker.com

- "Docker packages applications and their dependencies together into an isolated container making them portable to any infrastructure. Eliminate the "works on my machine" problem once and for all."

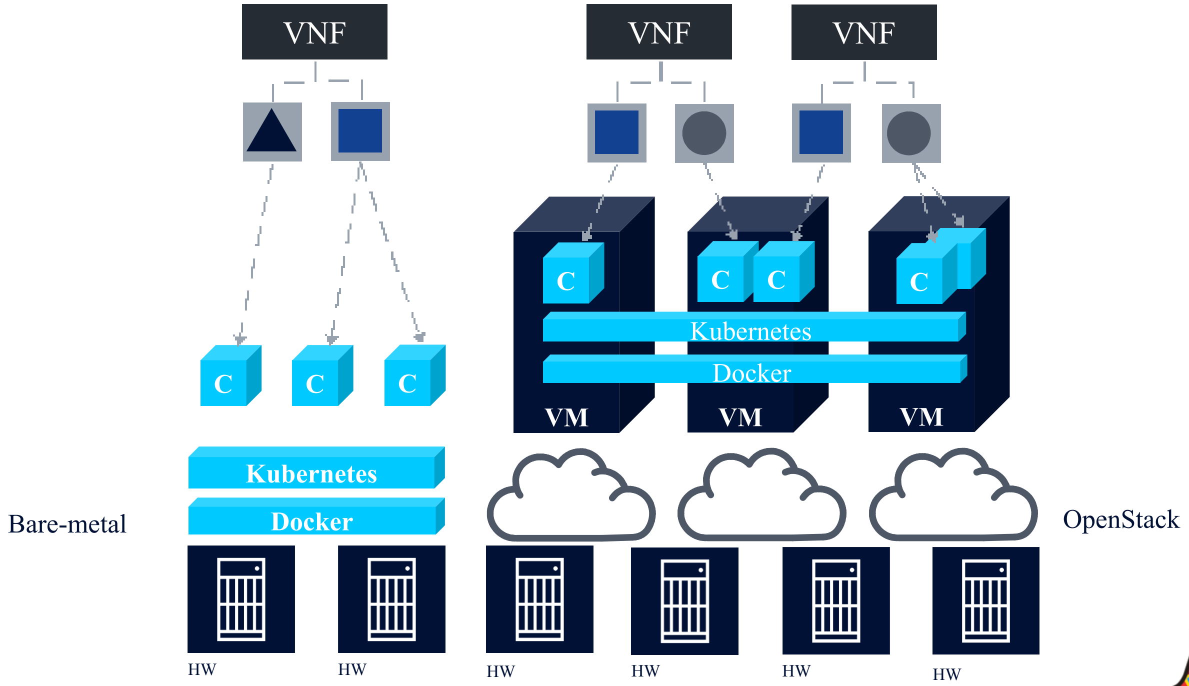
"Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."

Node overview



Deployment methods for container based VNFs

Hybrid environment

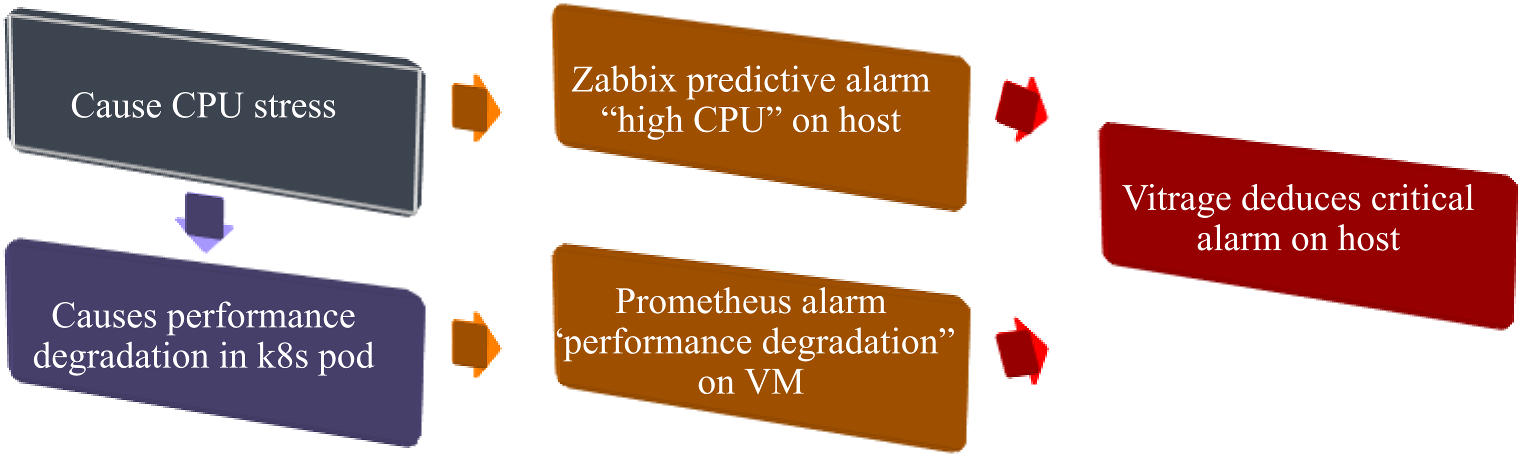


What is Prometheus?

- Efficient time series DB
- Flexible query language
- Alerting
- Many exports and integrations
- 63% of Kubernetes clusters

https://prometheus.io/docs/introduction/overview/

Demo


Cause CPU stress
Causes performance degradation in k8s pod
Zabbix predictive alarm “high CPU” on host
Prometheus alarm ‘performance degradation” on VM
Vitrage deduces critical alarm on host

Demo

More proactive possibilities in Vitrage

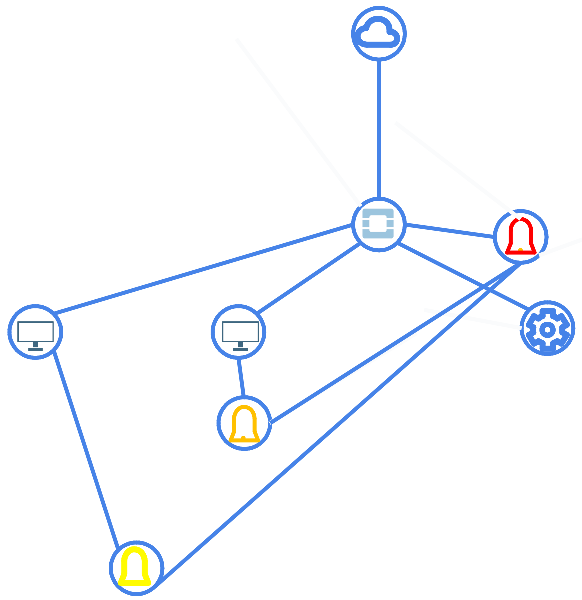
- Instead of deducing alarms, execute actions using Mistral
- Use Prometheus predictive functions
- Pluggable data sources
- Combine data from several data sources across multiple architecture layers

VITRAGE
an OpenStack Community Project

Future plans:

1. Diagnostic actions
2. Bell Labs Change Detection System

Diagnostic actions



Alarm is raised on resource

↓

Vitrage suspects a root cause

↓

Trigger health check – Diagnostic actions

↓

Get response and present it

VITRAGE
an OpenStack Community Project

Bell Labs Change Detection System

Background:

OpenStack systems has many components and each one has a log.

While errors are reflected in the logs, there are too many logs which are difficult to read.

Challenge:

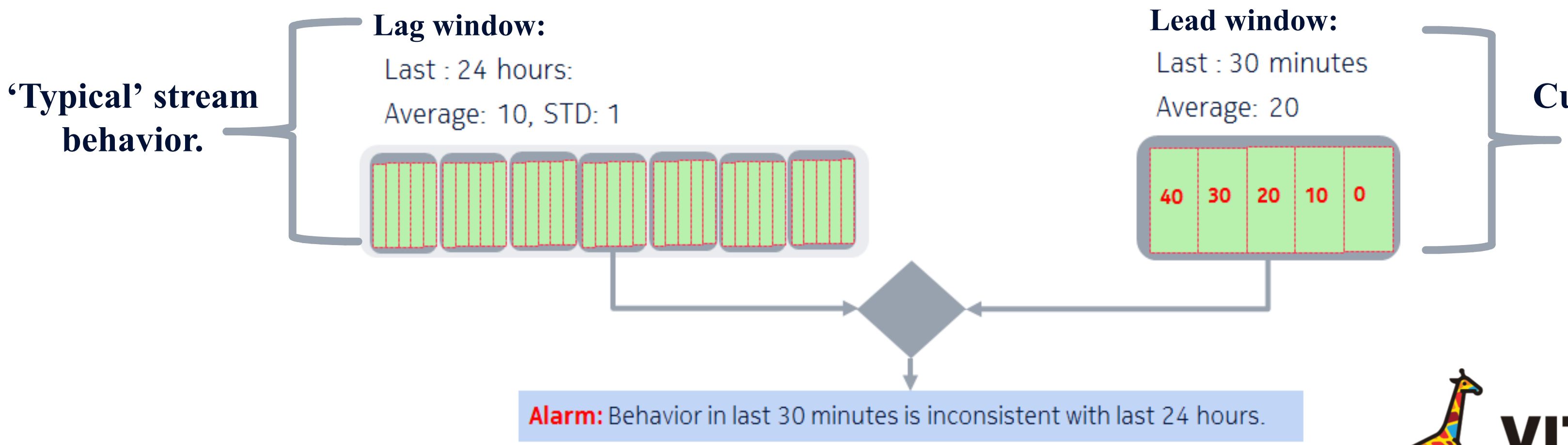
Find the root cause of a problem and proactively notify problems, based on logs changed behavior.

gil.einziger@nokia.com

VITRAGE
an OpenStack Community Project

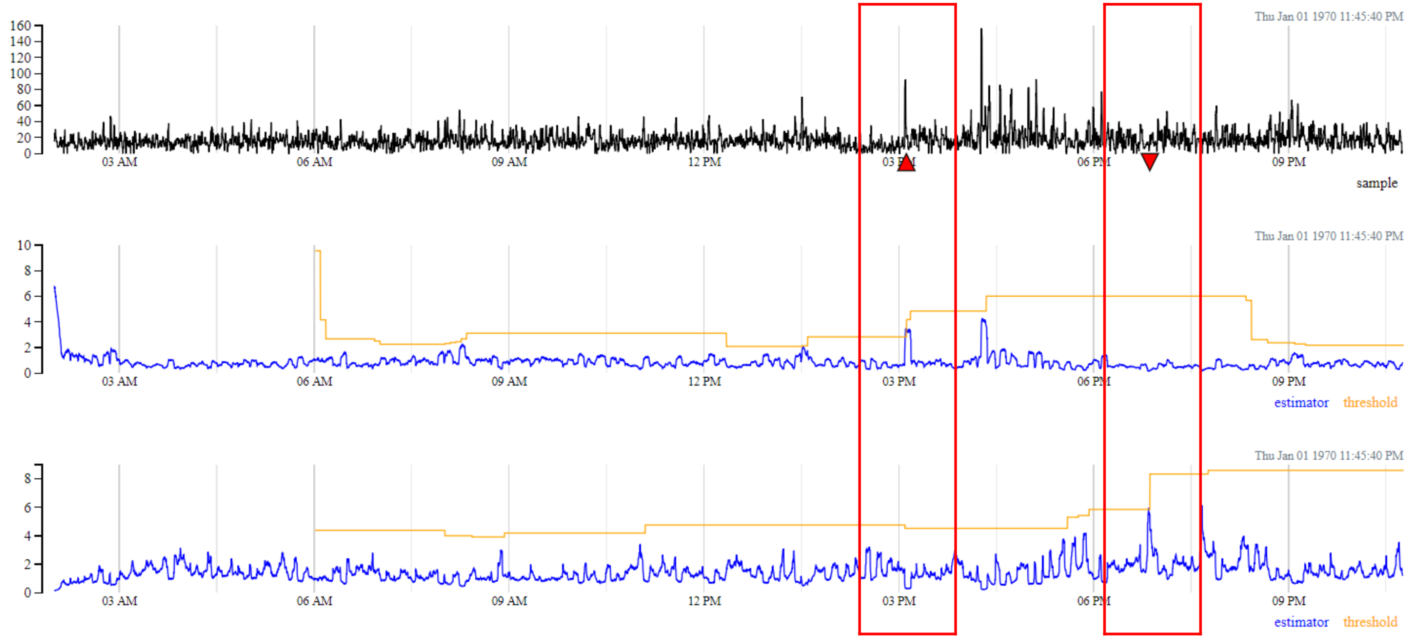
Change Detection System: Under the hood.

- Compare a small ("lead") window to a larger ("lag") window in terms of average & stdev.
- Detection threshold is deduced **dynamically** and **autonomously** from the lag window.
- Space efficiency through approximate windows.



Alarm: Behavior in last 30 minutes is inconsistent with last 24 hours.

Example: keystone logs



VITRAGE
an OpenStack Community Project

VITRAGE

an OpenStack Community Project