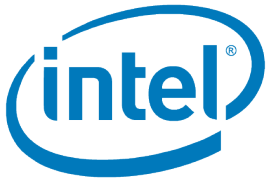
CLOUD TELEMETRY

Marcin Spoczynski
Researcher @ Intel Labs

LinuxFest
06-05-2017



Legal Disclaimer

Co-funded by the Horizon 2020 Framework Programme of the European Union

Who Am I

- Researcher on H2020 Mikelangelo project / Cloud Engineer
- Work/Research: telemetry, tools, cloud software management, visualizations
- now Researcher Intel Labs previously Python/C Developer and DevOps

Agenda

- Systems Before Cloud (BC) and Now
- Telemetry
- Let's complicate it – Cloud
- Metric Aggregation
- Anomaly Detection
- Closing the Loop
- Q&A

Systems BC

- Single or multithreaded application
- Application deployed on one or multiple servers without clear dependencies
- Big applications very often deployed only on the one server
- Single instance apps
- High maintenance costs

intel

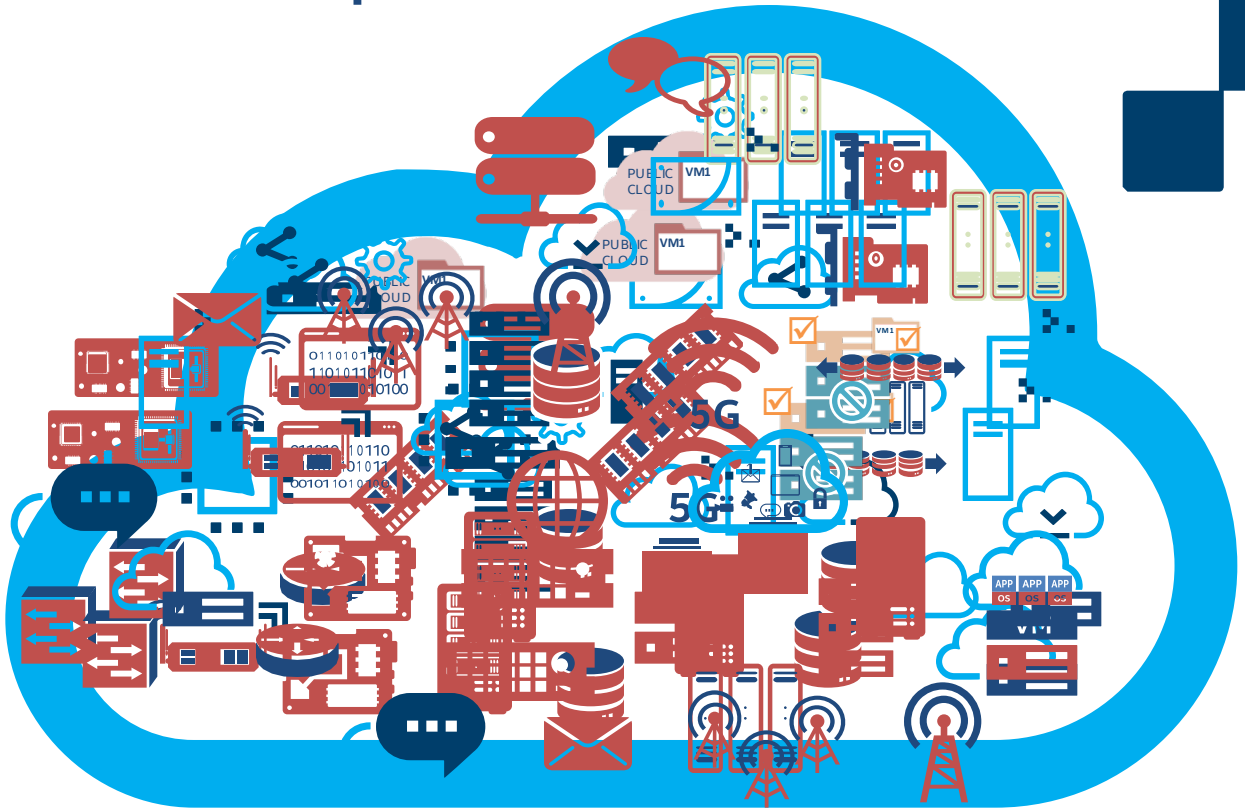
MIKELANGELO

Cloud Systems

- Multithreaded, concurrency applications
- Workload dependencies
- Micro services
- Multi-instance application deployed across servers
- Hardware appliances implemented as workloads on the Cloud (switches, routers, firewall, VPN)
- Low maintenance costs, stacks moved to the public cloud

intel MIKELANGELO

Cloud is complicated



TELEMETRY

Telemetry

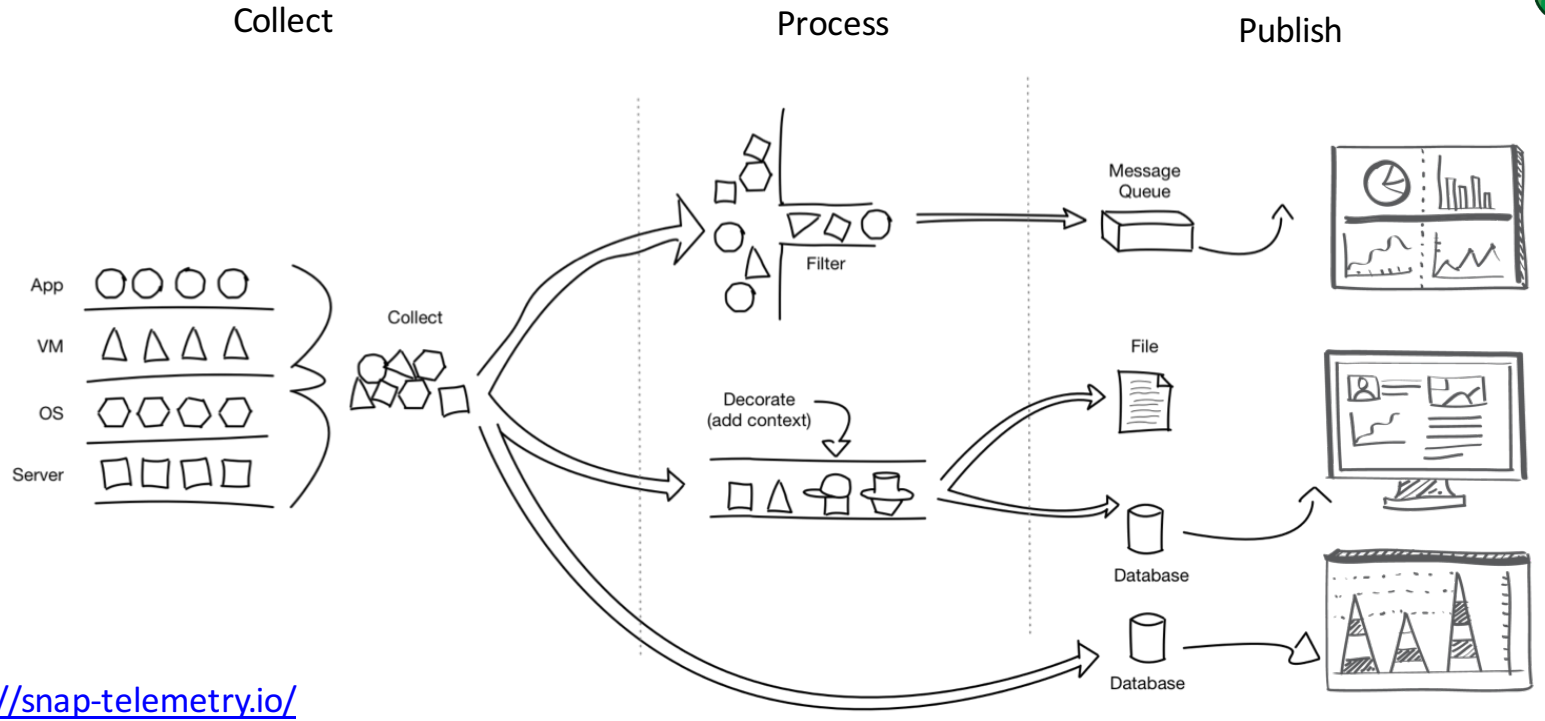
Telemetry is an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring. The word is derived from Greek roots: *tele* = remote, and *metron* = measure.

source: wikipedia

Cloud Telemetry vs Native Telemetry

- More data to collect
- More visible correlations between applications and management software
- Data needs to be highly aggregated: hundreds or thousands of data points from many stack layers per second
- Data collected from one system can be very large (~1 GB per hour: 200 metrics / 10 servers / 50 VMs)
- Multiple layers of dependent management systems (Openstack, OpenVswitch, K8S, Mesos, Linux, Cobbler, Puppet, Ansible etc.)

Introducing Snap: Open-source Telemetry



http://snap-telemetry.io/

```
$ go get github.com/intelsdi-x/snap
```

COLLECTING TELEMETRY

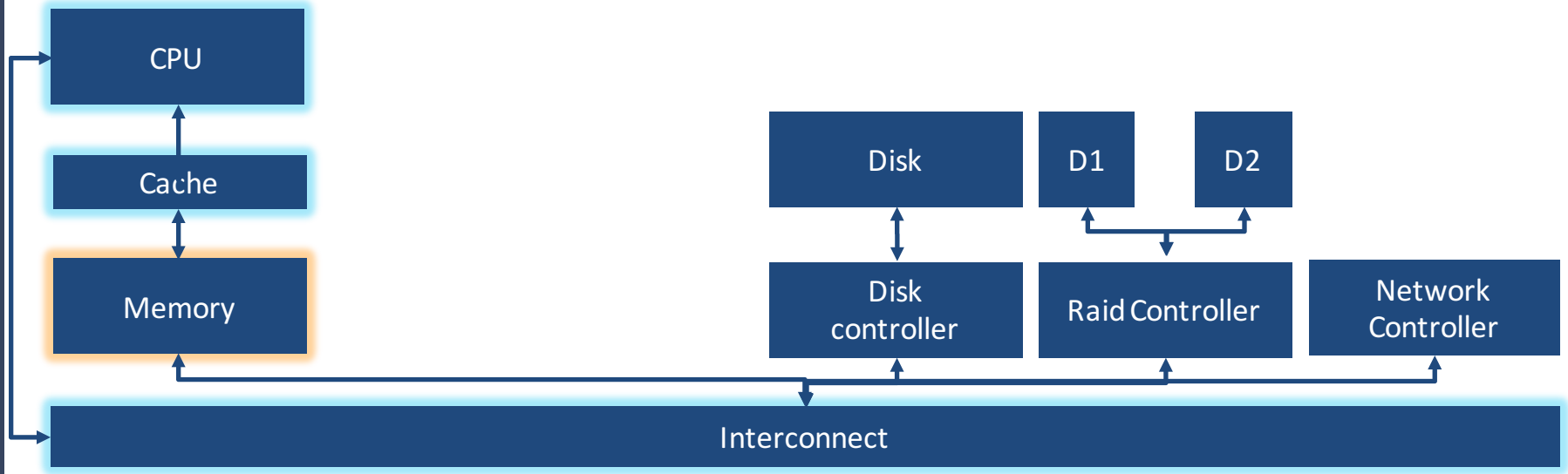
CPU – Key metrics

- L2, L3 Cache usage
- Amount of time that spent performing various kinds of work (normal and nice for user)
- Amount of time waiting for i/o to complete
- Amount of time servicing interrupts and softirqs
- Load Avg
- CPU Power Consumption, Temperature

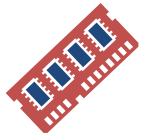
https://github.com/intelsdi-x/snap-plugin-collector-cpu

https://github.com/intelsdi-x/snap-plugin-collector-pcm

Interconnections - CPU



Memory – Key metrics



- Memory Active
- Memory Swap
- Memory Total
- Swap In, Swap Out
- Huge Pages
- Memory Hardware Corrupted (only ECC memory)
- Memory Power Consumption, Temperature

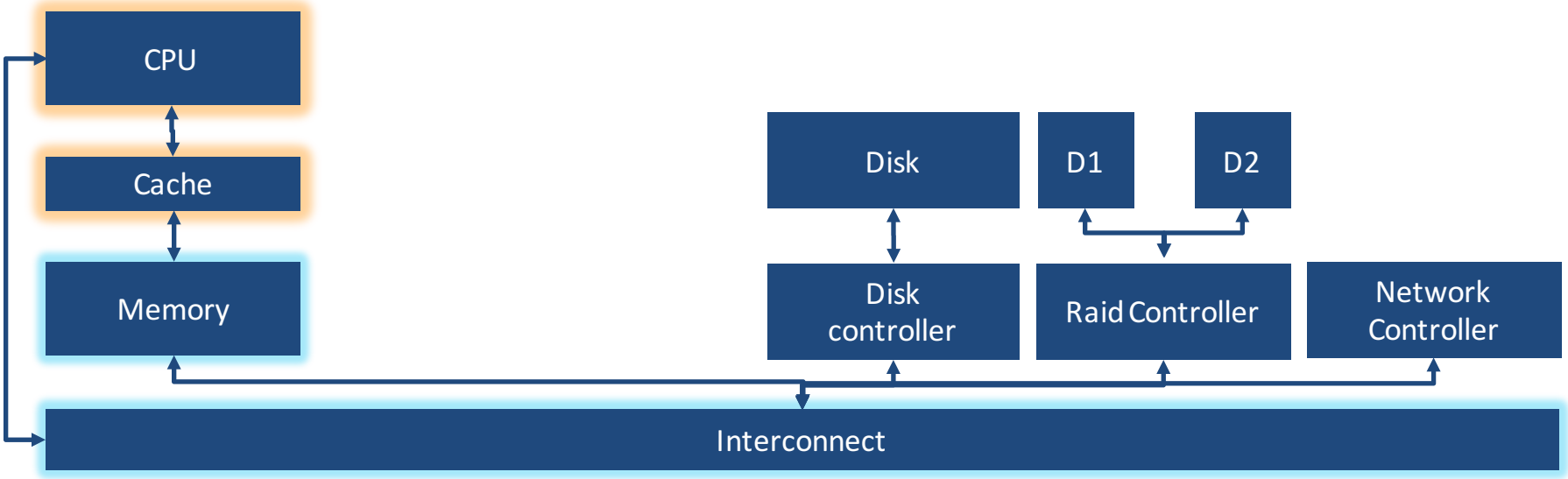
https://github.com/intelsdi-x/snap-plugin-collector-meminfo

https://github.com/intelsdi-x/snap-plugin-collector-psutil

intel

MIKELANGELO

Interconnections - Memory



Disk – Key metrics

- Merged Read, Merged Write
- Time Read, Time Write
- I/O Time
- Weighted I/O Time
- Pending OPS
- Low level SMART statistics

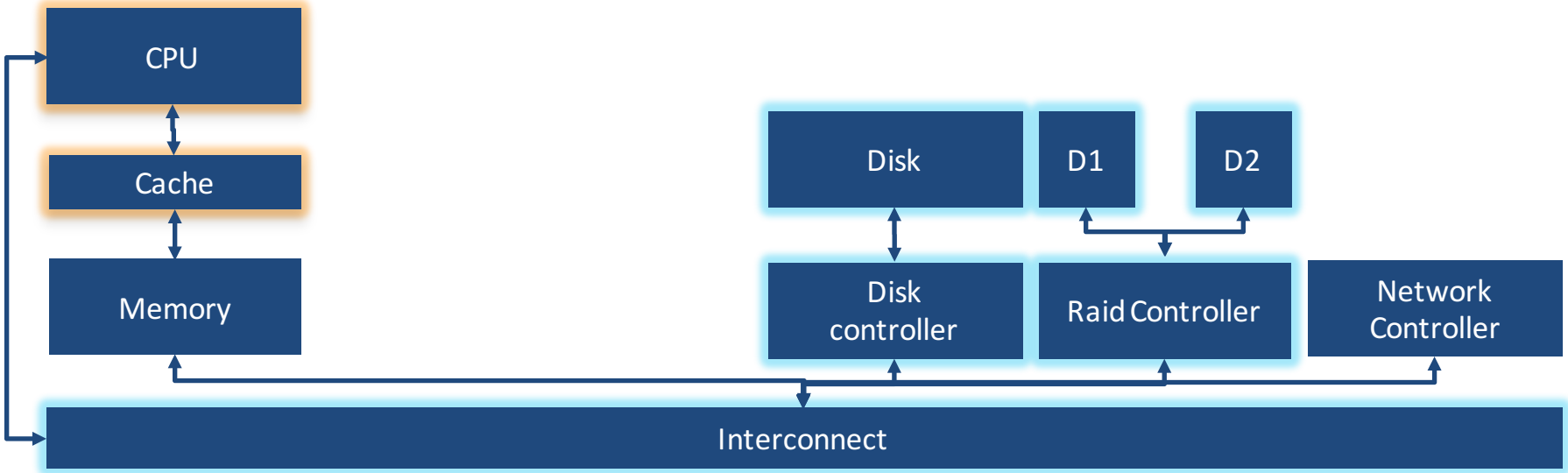
https://github.com/intelsdi-x/snap-plugin-collector-disk

https://github.com/intelsdi-x/snap-plugin-collector-iostat

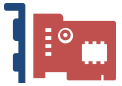
intel

MIKELANGELO

Interconnections - Disk



Network Interface – Key metrics



- Bytes In, Bytes Out
- Packets In, Packets Out
- Errors In, Errors Out
- Drop In, Drop Out
- Link errors

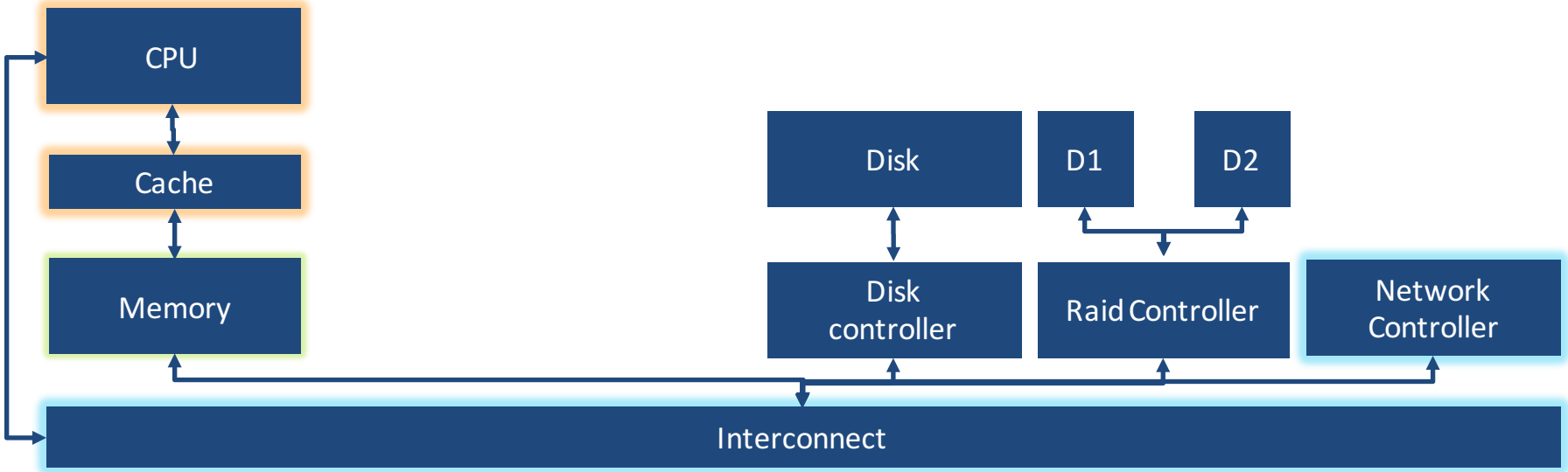
https://github.com/intelsdi-x/snap-plugin-collector-interface

https://github.com/intelsdi-x/snap-plugin-collector-ethtool

intel

MIKELANGELO

Interconnections – Network Interfaces



COLLECTING FROM OS (LINUX)

Processes – Key metrics

- Memory usage VM, RSS
- Time spend in user space
- Cache misses
- Disk write and read
- Sum of the zombie, dead and waiting processes

https://github.com/intelsdi-x/snap-plugin-collector-processes

https://github.com/intelsdi-x/snap-plugin-collector-perf

intel

MIKELANGELO

File system – Key metrics

- Space used
- Space free
- Space reserved
- File system Errors from syslog

https://github.com/intelsdi-x/snap-plugin-collector-df

intel

MIKELANGELO

COLLECTING FROM THE CLOUD

Cloud Availability

- Logs from various Cloud Services (Errors, Warnings)
- Number of requests per service, user, tenant
- Cloud Component availability (Nova Compute, Openvswitch, K8S Pods)
- Micro benchmarking to test reliability of the cloud

https://github.com/intelsdi-x/snap-plugin-processor-logs-openstack

https://github.com/intelsdi-x/kubesnap

Network and Cloud Storage Latency

- Latency between interfaces
- Latency between external cloud components
- Latency from client to the SAN storage
- Time spent on Openvswitch process per compute node
- Number of flows (OpenFlow) per interface

https://github.com/raintank/snap-plugin-collector-ping

intel MIKELANGELO

Additional hardware / software

- FPGAs (Temperature, Power, Utilization)
- Accelerators (Temperature, Power, Utilization)
- Sensors (Availability)

PROCESSING TELEMETRY

USE method

The Utilization Saturation and Errors (USE) Method is a methodology for analyzing the performance of any system. It directs the construction of a checklist, which for server analysis can be used to quickly identify resource bottlenecks or errors

source: http://brendangregg.com

https://github.com/intelsdi-x/snap-plugin-collector-use

USE method - metrics

Name	Formula	Threshold
Compute utilization	100 - idle	0 - 100 %
Compute saturation	load1/number of cpus	0 - max
Disk utilization	iostat % util	0 - 100 %
Disk saturation	iostat avg-queue-size	0 - max
Memory utilization	memory – memory used	0 - 100 %
Memory saturation	memory swap in/ memory swap out	0 - max
Network utilization	(tx + rcv bytes)/ bandwidth	0 - 100 %
Network Saturation	(tx + rcv overrun) - # of pkts %	0 - max

Anomaly Detection

In data mining, anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.

source: http://www.wikipedia.com

Anomaly Detection

Typically the anomalous items will translate to some kind of problem such as software misconfiguration, attack, unexpected software behavior, hardware or software errors.

Anomaly Detection – Tukey Method

The intention of this method is to reduce the amount of data that needs to be transmitted without compromising the information that can be gained from potential usages of the data

https://github.com/intelsdi-x/snap-plugin-processor-anomalydetection

Anomaly Detection – ESD

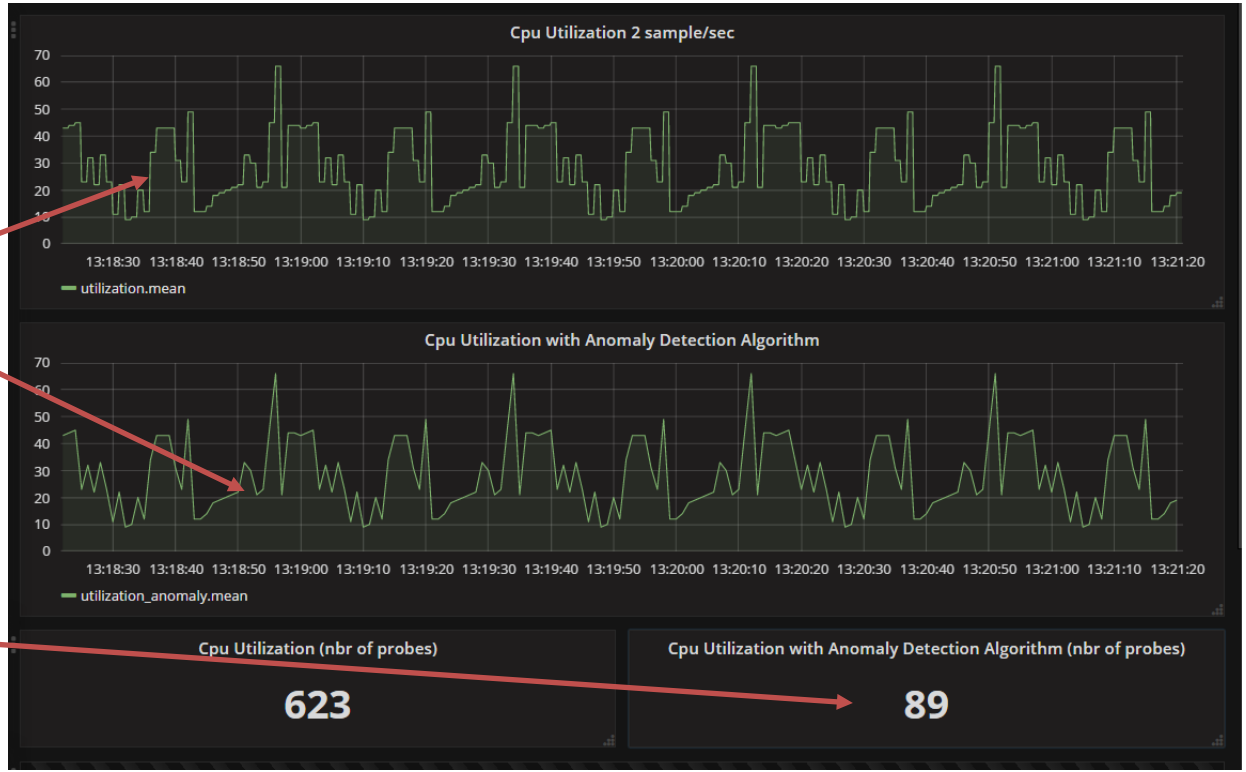
Seasonal Hybrid ESD (S-H-ESD) builds upon the Generalized ESD test for detecting anomalies. Note that S-H-ESD can be used to detect both global as well as local anomalies. This is achieved by employing time series decomposition and using robust statistical metrics, viz., median together with ESD.

https://github.com/intelsdi-x/snap-plugin-processor-anomalydetection

Anomaly Detection – Use Case

data shape preserved

8 x less samples



PUTTING IT ALL TOGETHER

TSDBs

TSDBs are databases that are optimized for time series data

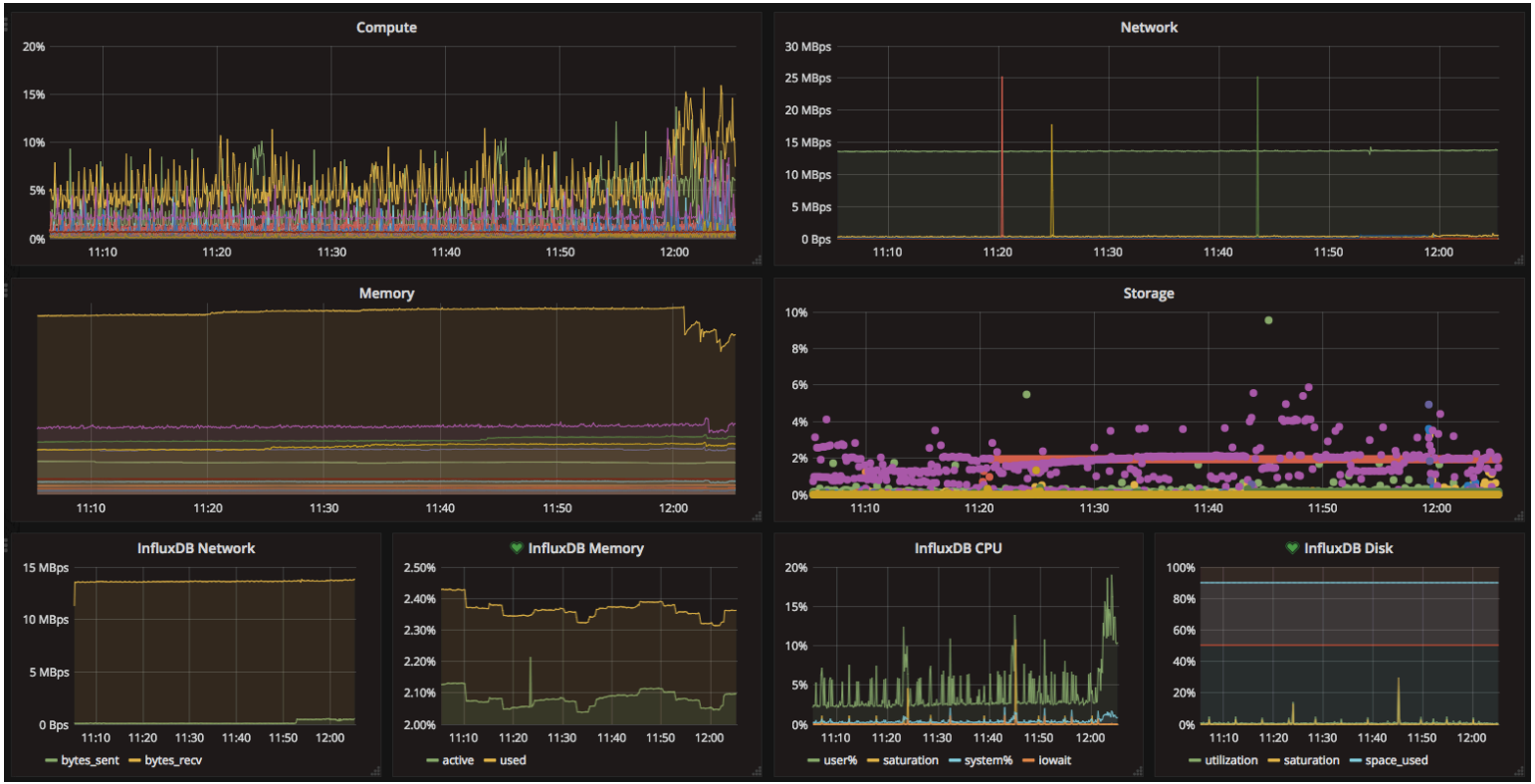
Telemetry Backend - TSDB

- InfluxDB - https://www.influxdata.com (used in Mikelangelo)
- KairosDB - https://kairosdb.github.io
- OpenTSDB - http://opentsdb.net/
- Graphite - https://github.com/graphite-project/graphite-web

Data Visualization

- Grafana – Monitoring visualization - https://grafana.com/
- Kibana – Logs visualization https://www.elastic.co/guide/en/kibana/current/index.html
- Flamegraphs – visualization of profile software https://github.com/brendangregg/FlameGraph
- Heatmap - http://www.gnuplot.info/

Data visualization – Monitoring



CLOSING THE LOOP

Closing the loop

- Enhance Cloud Schedulers – use aggregated metrics like Utilization, Saturation and Errors for Weighter on Openstack and K8S
- Apply Machine Learning algorithms on the datasets to find key metrics for the workloads
- Profile application - then redeploy using Continuous Integration

CLOUD TELEMETRY - DEMO

More Information

- Mikelangelo
- https://www.mikelangelo-project.eu/
- Snap
- http://intelsdi-x.github.io/snap/ – start here!
- https://github.com/intelsdi-x/snap – code, suggest, contribute
- https://medium.com/intel-sdi – technical insights and articles
- https://intelsdi-x.herokuapp.com/ – chat with snap developers
- https://vimeo.com/intelsdi/videos – see it in action

Q&A

marcin.spoczynski@intel.com
@sandlbn
github.com/sandlbn


MIKELANGELO