Professorship for Computer Science
Communication Services, Telecommunication Systems and Computer Networks

Cloudless Resource Monitoring in a Fog Computing System Enabled by an SDN/NFV Infrastructure

July 11, 2022

Duy Thanh Le, Marcel Großmann, Udo R. Krieger

Content

Introduction

Major evolutions in computer science

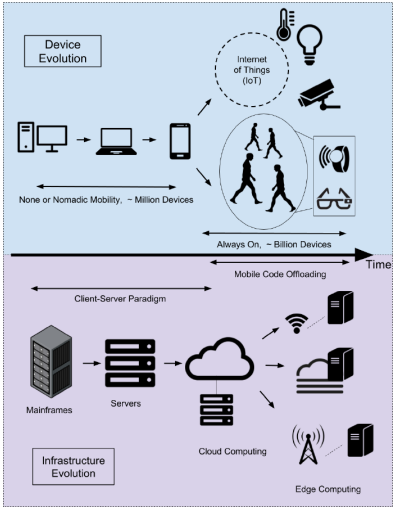


Figure 1: Device & infrastructure evolution [cf. 1]

Characteristics:

- Ubiquitous computing vision a reality now.
- Enormous audio-visual data.
- Dispersed Cloud and resources geographically.

Performance issues [2]:

- High latency.
- Long processing/queueing delay.
- Availability, QoS, resource management.

Introduction

Major evolutions in computer science

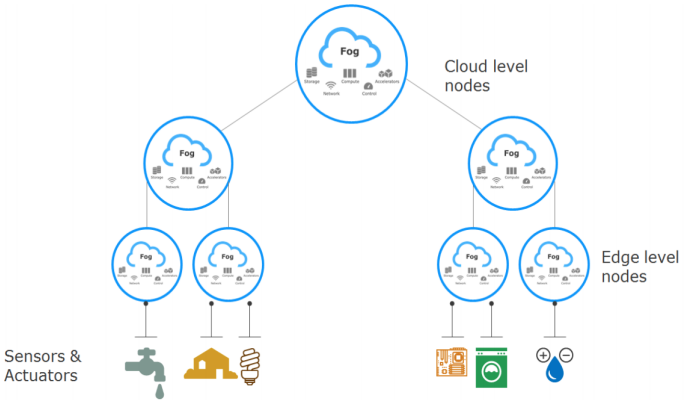


Figure 2: The reference model of multi-tier nodes deployment [cf. 3, p. 4]

Fog computing - a distributed computing paradigm - extending the Cloud computing to support the vision of IoT without restriction[4].
Benefits:

- Low latency
- Mobility support
- Context-awareness
- Real-time analytics
- Efficiency and cost

Traditional network architects increasingly incompetent due to the "vertical" lock between data and control plane inside networking devices[5]. Two approaches to solve that inherent problems:

- Clean-slate: GENI[6], Named Data Networking (NDN) [7, 8]
- **Software-Defined Network (SDN)[5, 9]**

SDN, "In essence, SDN gives network designers freedom to 'refactor' the network control plane..."[9], offers following benefits:

- Programmable network
- Scalability
- Efficiency and cost-effective
- Vendor-agnostic
- Logically centralized
- Proactive and analytical forwarding decisions.

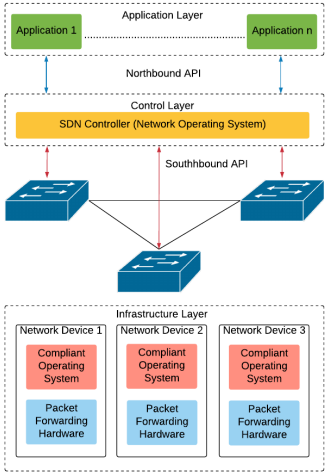


Figure 3: SDN architecture proposal ([cf. 10])

Recent technology advancements also open new possibilities to develop an intelligent & context-aware resource management system:

- Containerization
- Autonomic computing
- Modular monitoring frameworks (Pymon [11], TICK [12], Prometheus [13])

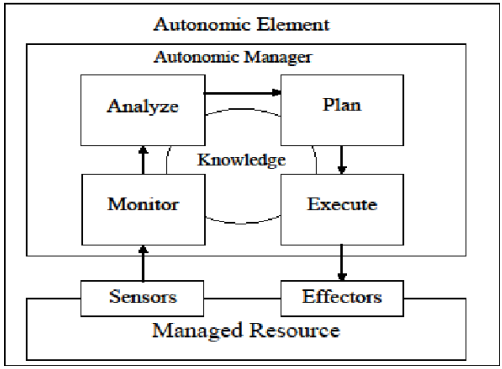


Figure 4: Generic MAPE-K architecture [cf. 14]

Cloudless Resource Monitoring (CRM)

General architecture of the CRM system

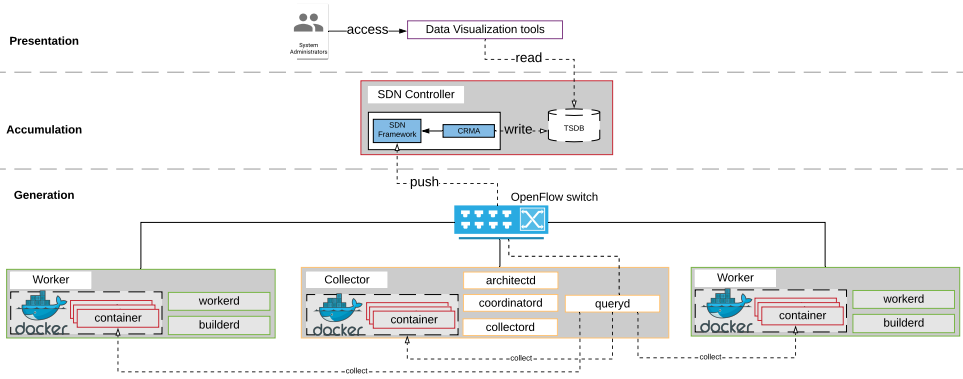
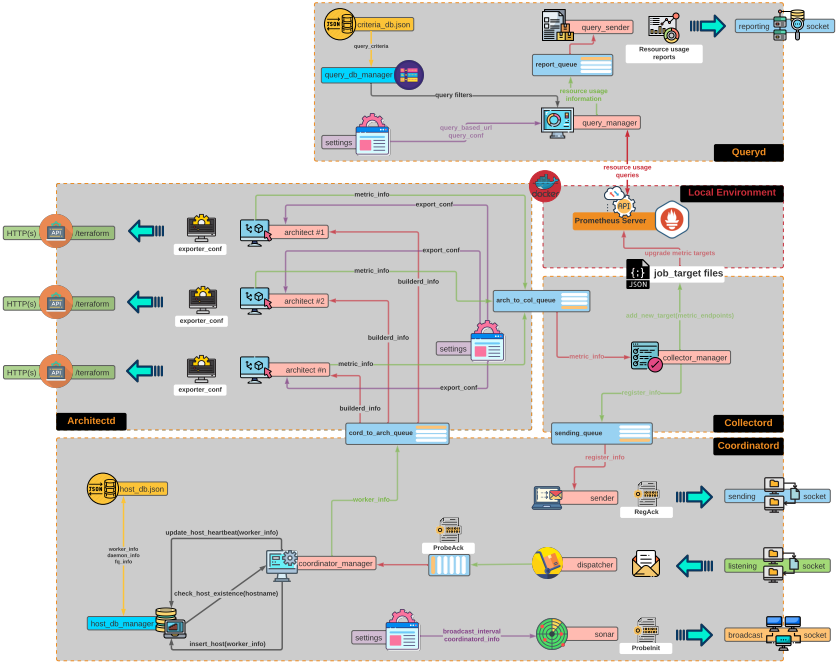


Figure 5: Overview of CRMS system

CRM


criteria_db.json
query_criteria
query_db_manager
query_filters
query_based_url
query_conf
settings
query_manager
resource usage information
report_queue
query_sender
Resource usage reports
reporting
socket
Queryd
resource usage queries
Local Environment
api
Prometheus Server
upgrade metric targets
job_target files
JSON
HTTP(s)
API
/terraform
exporter_conf
architect #1
architect #2
architect #n
metric_info
export_conf
builderd_info
Architectd
arch_to_col_queue
add_new_target(metric_endpoints)
collector_manager
register_info
Collectord
cord_to_arch_queue
sending_queue
Coordinatord
host_db.json
worker_info
update_host_heartbeat(worker_info)
worker_info
daemon_info
hj_info
coordinator_manager
check_host_existence(hostname)
insert_host(worker_info)
host_db_manager
ProbeAck
sender
RegAck
sending
socket
dispatcher
listening
broadcast_interval
coordinatord_info
sonar
ProbeInit
broadcast

Cloudless Resource Monitoring (CRM)

The `Worker` system

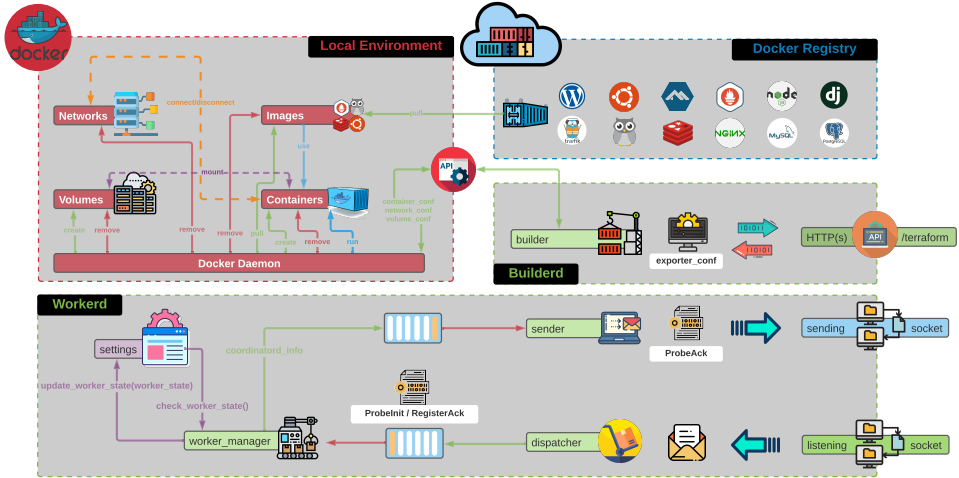


Figure 6: The main components of a Worker system

Cloudless Resource Monitoring (CRM)

Primary phases in the CRM system

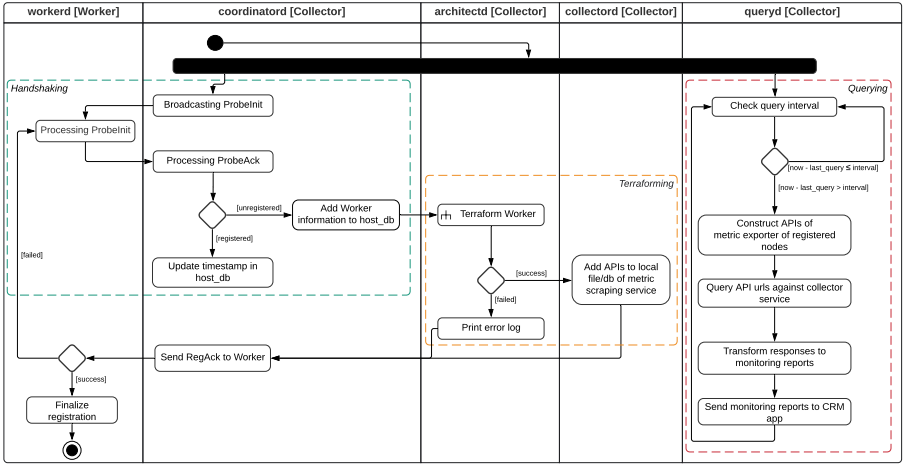


Figure 7: Activities in Generation layer

Cloudless Resource Monitoring (CRM)

Primary phases in the CRM system

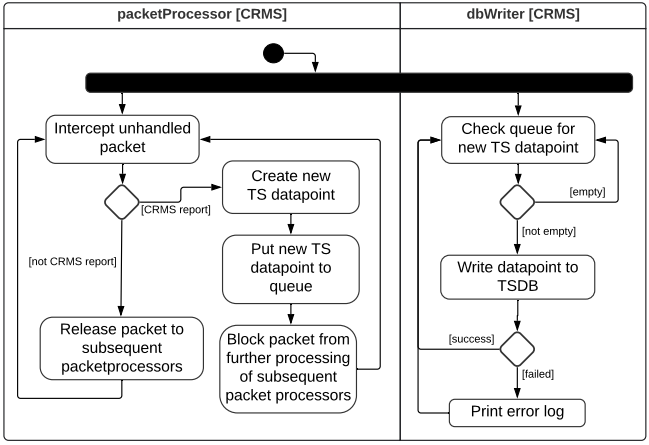


Figure 8: Activities in Accumulation layer

Topology Information	
Name	**mesh**
Device	**4** OVS, **1** Collector; **8** Workers

Hardware specs	
Collector	CPU: 2 cores; RAM: 2GB; Storage Size: 20GB;
Worker	CPU: 2 cores; RAM: 2GB; Storage Size: 20GB; I/O rate: 50MB/s

Name	Version	Architecture
NodeExporter	unibaktr/nodeexporter:0.18.1	linux: amd64;arm64;arm/v6-v7; ...
cAdvisor	unibaktr/cadvisor:v0.36.0	linux: amd64;arm64;/arm/v7; ...
Prometheus	unibaktr/prometheus:2.17.1	linux: amd64;arm64;arm/v6-v7; ...
Influxdb	influxdb:1.8.4	linux: amd64;arm64/v8;arm/v7
Grafana	grafana:latest	linux: amd64;arm64;arm
onos-multiarch	thanhledev/onos-multiarch:0.0.1	linux: amd64;arm64

Table 1: Topology, hardware, container specs

Evaluation of CRM

Test results

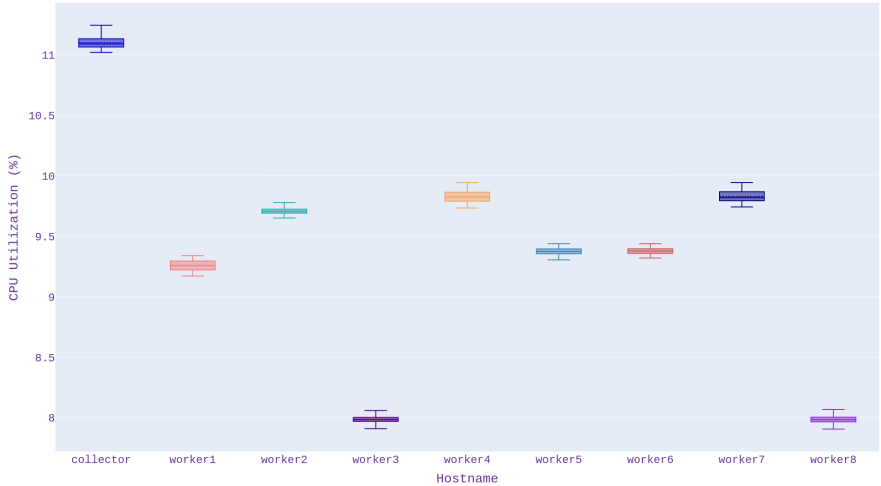


Figure 9: CPU Usage

Evaluation of CRM

Test results

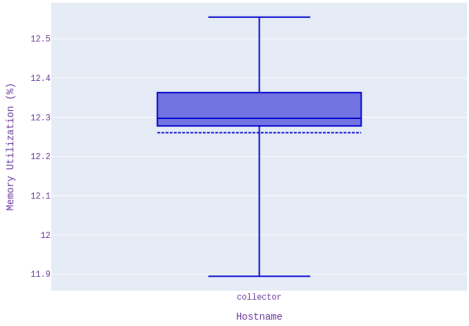


Figure 10: Collector memory usage

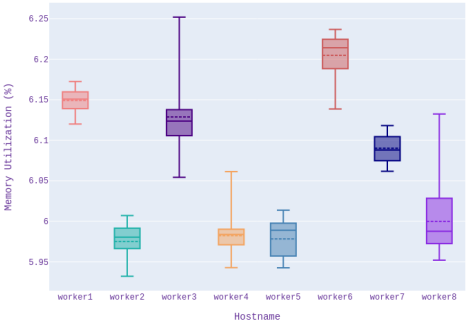


Figure 11: Worker memory usage

Evaluation of CRM

Test results

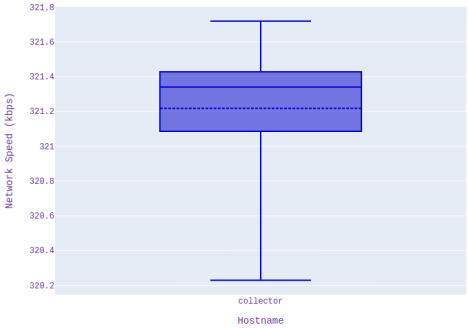


Figure 12: Collector network I/O rate

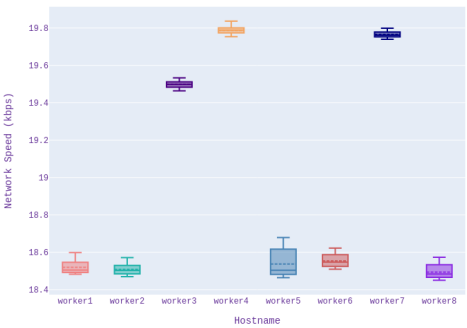


Figure 13: Worker network I/O rates

Conclusions

Achievements of CRM:

- A working prototype of a self-managment container orchestrator
- A working solution of a multi-arch docker image of SND controller ready to be deployed on the egde/fog nodes.

Issues:

- Single point of failure
- Application has not yet supported any GUIs.
- Overall resource utilization is not optimized.

Further developments:

- Implementation of missing elements regarding MAPE-K model.
- Providing utility functions (GUIs, APIs)

References I

[1] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[2] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2017.

[3] O. C. A. W. Group *et al.*, "Openfog reference architecture for fog computing," *OPFRA001*, vol. 20817, p. 162, 2017.

[4] F. Bonomi and R. Milito, "Fog computing and its role in the internet of things," *Proceedings of the MCC workshop on Mobile Cloud Computing*, 08 2012.

[5] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[6] A. Feldmann, "Internet clean-slate design: What and why?" vol. 37, no. 3, p. 59–64, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1273445.1273453

[7] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey," *Computer Science Review*, vol. 19, pp. 15–55, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013715300599

[8] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, "A brief introduction to named data networking," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 1–6.

[9] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized? state distribution trade-offs in software defined networks," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1–6. [Online]. Available: https://doi.org/10.1145/2342441.2342443

References II

[10] T. N. D. and K. Gray, *SDN: Software Defined Networks (1st Edition)*. O'Reilly Media, Inc, 2013.

[11] M. Großmann and C. Klug, "Monitoring container services at the network edge," in *2017 29th International Teletraffic Congress (ITC 29)*, vol. 1, 2017, pp. 130–133.

[12] "InfluxData: Time Series Platform," https://www.influxdata.com/time-series-platform/, accessed: July 11, 2022.

[13] "A monitoring solution for Docker hosts, containers and containerized services," https://stefanprodan.com/2016/a-monitoring-solution-for-docker-hosts-containers-and-containerized-services/, accessed: July 11, 2022.

[14] S. Ouareth, S. Boulehouache, and S. Mazouzi, "A component-based mape-k control loop model for self-adaptation," *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1–7, 2018.

[15] Ivan Pepelnjak, *Management, Control, and Data Planes in Network Devices and Systems*, accessed: July 11, 2022.

Questions ?

Duy Thanh Le
duy-thanh.le@uni-bamberg.de