Mokslas – Lietuvos ateitis / Science – Future of Lithuania
ISSN 2029-2341 / eISSN 2029-2252
2020 Volume 12, Article ID: mla.2020.11454, 1–5
https://doi.org/10.3846/mla.2020.11454

Electronics and electrical engineering
Elektronika ir elektros inžinerija

OVERVIEW OF KUBERNETES CNI PLUGINS PERFORMANCE

Narūnas KAPOČIUS *

Vilnius Gediminas Technical University, Vilnius, Lithuania

Received 02 Novemebr 2019; accepted 07 November 2019

Abstract. Containerisation and microservices architecture are getting momentum in nowadays ICT field. Containers are deployed in both public and private cloud environments and usually for flexibility purposes are deployed in VM (Virtual Machines) environment. Microservices have a demand on a high number of containers which requires orchestration and Kubernetes is one of the most popular choice. However, Kubernetes does not offer networking solution and it is provided by CNI (Container Networking Interface) and its' plugins. In order to choose best plugin their performance needs to be evaluated. In this paper nine most popular CNI plugins TCP and HTTP protocols performance is evaluated in virtualised VMware ESXi and physical data centre environment. The results help to choose which CNI plugins to use either in virtualised or physical data centre environment.

Keywords: Kubernetes, orchestration, plugin, performance, Container Network Interface (CNI), VMware, cloud.

Introduction

To adapt to high demand of computing power virtualization technology is used. It provides flexibility and helps to save costs. In the recent years microservices paradigm and containerization emerged. Containerization is a new type of virtualization which offers less computing overhead compared to traditional hypervisors. However, both type of virtualization technologies adds noticeable amount of computing overhead (Li et al., 2017).

Microservices architecture requires high number of containers which need to be orchestrated with such tools as Kubernetes, Docker Swarm, Apache Mesos. Containers orchestration tools such as Kubernetes and Apache Mesos use Container Network Interface (CNI) for networking implementation between containers. CNI offers variety of plugins which solve same networking problem in a different way. Applications based on microservices architecture can be deployed on physical servers or for flexibility/cost purposes can be deployed in virtualised environment – VMs. In this paper most popular nine Kubernetes CNI plugins performance is compared. The performance was measured in data centre environment. CNI plugins performance was first measured in virtualised environment – VMs and then compared to CNI plugins performance on physical environment. Lastly both results were compared to baseline bare metal server performance results. From protocols perspective TCP and HTTP protocols were chosen because these are one of the popular protocols used by todays' modern microservices applications.

The reminder of this paper is organized as follows. Related work section summarizes recent publications on Kubernetes CNI plugins performance evaluation, Test environment setup section describes test environment in the data centre, in Results analysis section test results are presented, in the last section final conclusions and future work is presented.

1. Related work

Performance of Docker Swarm and Kubernetes containers orchestrators was evaluated by Großmann and Klug (2017). Großmann and Klug developed and used new monitoring framework PyMon to compare orchestrators CPU and RAM usage. PyMon which has small footprint on monitored system was used to measure orchestrators performance of four Raspberry Pi 3 single boards computers (SBCs) on which orchestrators clusters were created. The results show that Kubernetes master's node CPU utilization in idle state is around 30 percent and around 10 percent on worker nodes compared to Docker swarm master's node ~0.5 percent and ~1 percent on worker nodes. However, RAM utilization of all Kubernetes and Docker swarm nodes are all around 10 percent, except

*Corresponding author. E-mail: *narunas.kapocius@stud.vgtu.lt*

Kubernetes master node consumed around 40 percent of memory compared to Docker swarm master's node 10 percent memory usage. The authors emphasize that this comparison is not quite fair because Kubernetes has more features and capabilities compared to Docker Swarm and there is no ability to turn of those additional features to make comparison fairer. Also, it is worth noticing that the research was orientated to SBCs performance and orchestrators performance was evaluated only in the idle mode and the results in the data centre environment when network load is applied could be different.

As microservices architecture is becoming a standard for new applications in public and private cloud CNI solutions performance needs to be evaluated in order to improve network performance of cloud systems (Park & Yang, 2018). Park and Yang evaluated cloud network architecture based on OpenStack cloud platform and Kubernetes. Virtual network infrastructure was based on OpenStack Kuryr-OVS (Open Virtual Switch) and CNI Flannel plugin. Cluster consisted of 3 nodes which were working on Xeon® E5-2697, 64G RAM and 10G SFP+ physical server where OVS were used to connect Kubernetes pods to common network. Network performance was evaluated using iperf3 network bandwidth measuring tool where data was routed from pod working on Node1 to pod working on Node3 through pod working on Node2. It is worth mentioning that this network architecture was also designed to test Flannel CNI plugin. In this case OVS was also used to connect pods to a common network, but OVS switch was connected to Docker bridge network which was connected to Flannel bridge network where data was transmitted using Flannel UDP or VxLAN daemon. Analysis results show that in all investigated MTU sizes (64, 512, 8192, 16384 B) OVS transmission performance in terms of throughput is better compared to Flannel CNI transmission mode. Also, there is noticeable drop in throughput, especially for Flannel plugin, when MTU size decreases. Park and Yang explain that Flannel CNI plugin performance is lower compared to Kuryr-OVS due to frequent Flannel CNI OS kernel calls which causes significant computing overhead. It is important to emphasize that this research only covers network architecture where Kubernetes nodes are deployed at the same physical hosts and from the results it is not clear how long each performance measurement took and how much data was sent through the network.

Performance of three most popular containers orchestrators network solutions Flannel, Calico and Swarm Overlay was evaluated by Zeng et al. (2017). In the paper description experiment network architecture is not given, yet from the results it is clear that network performance was measured using few physical servers connected to 10 Gbit network. Flannel and Calico CNIs and Swarm Overlay network solution were compared by average delay, TCP and UDP throughput. Results show that Calico, Calico ipip and Flannel VxLAN delay is much lower compared to Flannel UDP CNI plugin. Also, all results were compared to physical infrastructure and it is evident that virtualization adds noticeable computing overhead to the system. In TCP case the best throughput was achieved by Calico CNI plugin which is a bit less compared to physical infrastructure throughput results (919 Mbit/s vs 919.9 Mbit/s). Second best throughput result was achieved by using Swarm Overlay solution (887.6 Mbit/s) and Flannel performed worst (815.9 Mbit/s). It is worth noticing that UDP throughput of all three network solutions were comparably worse compared to TCP throughput, however Calico performance was again the best of all three compared solutions – physical infrastructure ~185 Mbit/s, Flannel UDP ~50 Mbit/s, Swarm Overlay ~60 Mbit/s and Calico ~110 Mbit/s. Authors linked network solutions performance to solutions' configuration complexity. It is stated that Calico CNI plugin, which has the best performance of all three compared solutions, is the one which is the most complex to configure. However, Flannel CNI plugin which performance was worse is the easiest to configure and start using. This research is interesting in that way that virtualised network solutions performance was compared to bare metal infrastructure performance and that helps to set a baseline for results comparison.

The most comprehensive study of CNI plugins performance was published in 2018 (Ducastel, 2018) and updated in 2019 (Ducastel, 2019). CNI plugins performance was tested using three Supermicro physical servers connected to 10 Gbit Supermicro switch using Directly Attached Cable DAC. All three servers were configured to work on the same VLAN which supports 9000B Jumbo frames and were running Ubuntu 18.04 LTS OS, Kubernetes 1.12.2 and Docker 17.12 versions. 8 CNI plugins (Calico, Canal, Cillium, Flannel, Kube-router, Romana, WeaveNet and WeaveNet-encrypted) performance were studied. Contrail and Tungsten Fabric plugins were not added to the study because these plugins need lower, 3.10 Linux kernel version. All performance measurements were taken at least 3 times, but from the publication it is not clear how long time each measurement have taken. Tools used to measure CNI plugins performance are depicted in Table 1. Ducastel emphasize that to evaluate SSH protocol performance random 10 GB file was transmitted. However, for other protocols amount of transmitted data is not specified. Results show that TCP and UDP throughput for all CNI plugins and bare metal configuration are almost identical. In TCP case Calico plugin achieved the highest throughput compared to other plugins and the result is very close to bare metal test case (9844 Mbit/s VS 9903 Mbit/s). In UDP test scenario best throughput was achieved by Romana plugin which was even better than bare metal test scenario (9907 Mbit/s vs 9873 Mbit/s). It is interesting that UDP performance results published by Zeng et al. (2017) are so different from results published by Ducastel (2018, 2019) and it is also not clear why results are so different even for bare metal configuration. One explanation is that there is no robust way to measure UDP protocol throughput performance. It is worth noticing that in both Ducastel's publications it is not said which Flannel configuration (VxLAN or UDP) is used, but by results comparison it can be guessed that VxLAN configuration was used.

Table 1. Tools used to measure CNI plugins performance in ITNEXT (Ducastel 2018, 2019) publications

Protocol	Server tool	Client tool
TCP, UDP	iperf3	iperf3
HTTP	nginx	curl
FTP	vsftpd	
SSH	OpenSSH	OpenSSH

Also, Ducastel's CNI plugins study is interesting because not only TCP and UDP protocols performance is evaluated but it includes most popular internet protocols such as HTTP, FTP and SSH. For all these protocols most of the CNI plugins throughput is almost equal to bare metal results. However, in FTP and HTTP protocols case WeaveNet and Cillium plugins throughput is comparably lower. In SSH protocol case all plugins perform approximately 15 percent worse compared to bare metal scenario. This shows that virtualised solutions perform worse compared to physical solutions in cases where there is a demand for computing resources, such as encryption in SSH protocol. From CPU and RAM resource usage perspective results for all CNI plugins are more scattered compared to throughput results. The least CPU resources were used by Calico plugin and the most CPU resources were needed by Cilium and WeaveNet plugins. Memory usage of all plugins are almost identical except Cilium plugin which uses almost two times more memory compared to other CNI plugins.

2. Test environment setup

Test environment was adapted to measure Kubernetes CNI plugins performance in virtualised and physical infrastructure which results were compared to physical infrastructure baseline performance results. Test environment was setup in three data centres which were interconnected using dedicated fibre channel. Each data centre from each other was separated by approximately 10 km distance. For physical infrastructure "Cisco UCS B200 M5" blade servers and "Cisco UCS UP629 Fabric interconnect" switches were used. 3 test scenarios were taken into account: Kubernetes cluster in physical environment, Kubernetes cluster in virtualised environment and two directly connected bare metal servers for baseline results. General test environment setup is given in Figure 1 and each of the test case environment is described below.

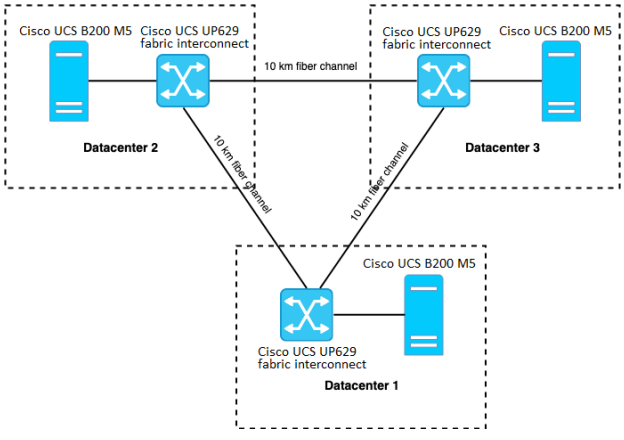

Figure 1. General Kubernetes CNI plugins test environment setup

Table 2. Tested protocols and server/client side tools used for the performance analysis

Protocol	Server	Client	Server tool	Client tool
TCP, UDP	Kubernetes worker node 1	Kubernetes worker node 2	iperf3 server	iperf3 client
HTTP			nginx	curl

In physical Kubernetes cluster test scenario, Kubernetes cluster was running directly on the physical servers OS. In virtual Kubernetes cluster test scenario additional layer of virtualization – VMware ESXi type 1 hypervisor was used where VMware vSphere tool set was used for provisioning virtual machines. Each virtual machine was assigned 24 GB of RAM and 4 cores of CPU and in physical environment test cases physical machines memory was limited to 24 GB to match virtual environment. In all 3 test cases servers were running CentOS 7.6 OS. Kubernetes clusters were created using Docker 18.09.1 and Kubernetes 1.14.1. However, when testing physical infrastructure baseline results on servers only plain OS with tools needed to complete measurements were installed. Kubernetes cluster consisted of 3 nodes – 2 worker nodes and one master node. During each measurement Kubernetes master node was setup in different data centre. Measurements were taken 3 times a day for one week. Tested protocols and tools used for the performance analysis are given in Table 2.

3. Results analysis

As was described in previous section three test scenarios were designed to evaluate CNI plugins performance in virtualised and physical infrastructure. The experiments results are shown below.

Results given in Figure 2 show TCP protocol throughput for different CNI plugins in virtualised and physical Kubernetes cluster. In virtual environment Canal plugin is the best. However, Calico and Romana plugins throughput is almost the same as Canals'. All CNI plugins throughput is significantly less compared to bare metal results in both virtual and physical clusters. In physical cluster Flannel performed the best where Canal and Calico performance is not so far from Flannel.

All CNI plugins TCP throughput decreased in virtual environment (Figure 3). In both 1500B MTU case Canal, Calico and Romana plugins performance decreased less than 1% and similar results were seen in 9000B MTU case. However, for other plugins, in 1500 MTU test case throughput decreased significantly more compared to 9000B MTU test case.

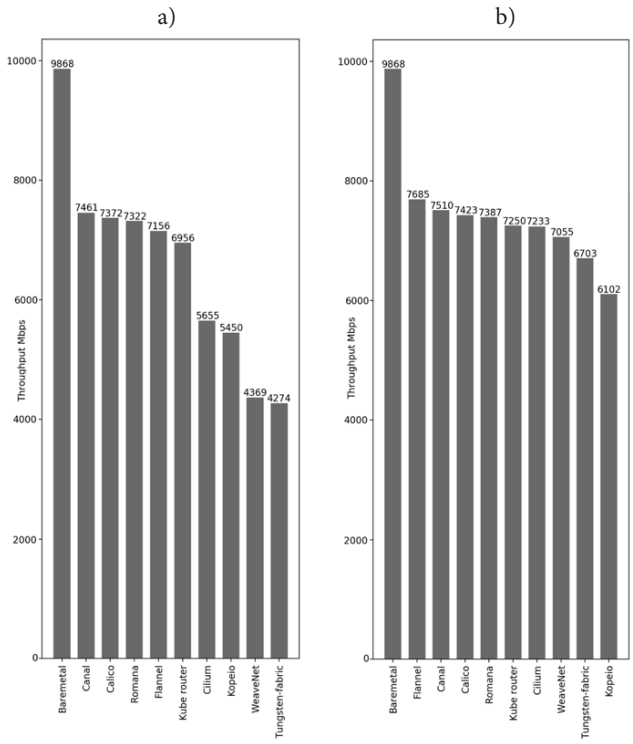

Figure 2. CNI plugins TCP protocol throughput comparison to bare matal results in (a) virtualized environment and (b) physical environment for 1500B MTU

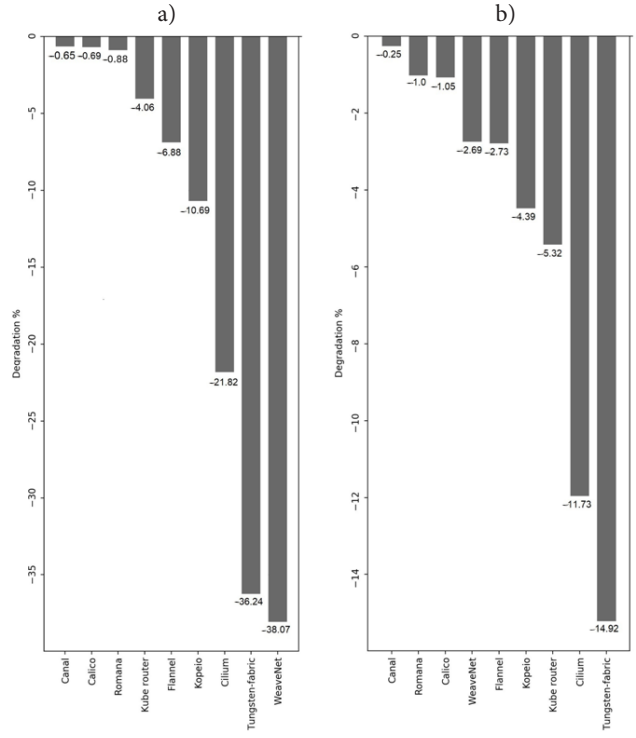

Figure 3. CNI plugins TCP protocol throughput relative performance degradation in virtual environment (a) 1500MTU (b) 9000B MTU

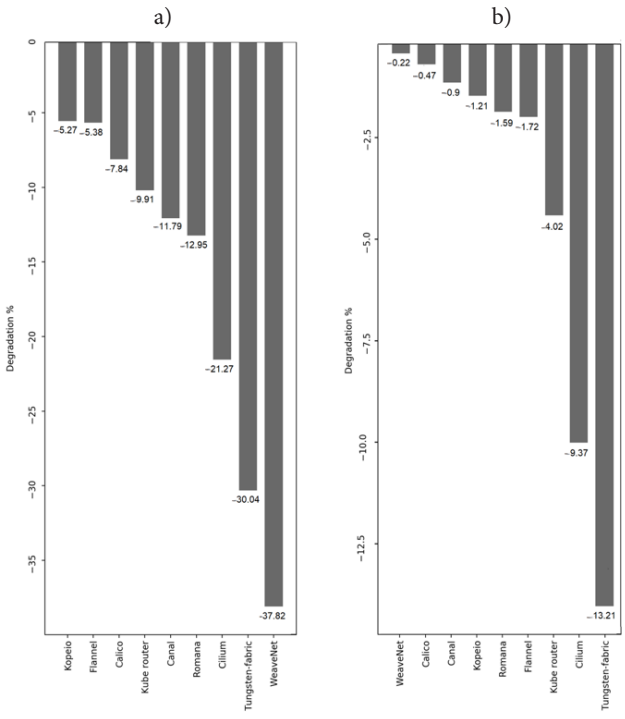

Figure 4. CNI plugins application layer protocol HTTP relative trhoughput degradation in virtual environment (a) 1500MTU (b) 9000B MTU

Similar results are observed in HTTP protocol (Figure 4) where in virtual environment all plugins performed worse compared to physical environments except WaveNet in 9000MTU case.

This shows that virtual environment is sensitive to higher computing loads and performance drastically decreases because of computing overhead which is added by virtualization layers.

Conclusions and future work

This paper analysed 9 most popular CNI plugins performance in virtual and physical data centre environment. To accomplish analysis 3 nodes Kubernetes cluster was created in three data centres separated by each other by approx. 10 km distance and connected to each other using dedicated fibre channel. Results confirm that all CNI plugins in TCP and HTTP case in virtualised environment performs worse compared to physical environment due to additional computing overhead added by hypervisor. Using higher MTU value (9000B vs 1500B) increases CNI plugins performance in virtualised environment because of less computing overhead added by protocol data.

Despite Kubernetes CNI plugins performance loss in virtualised environment, throughput loss for most used plugins is not significant and virtualised environment offers higher flexibility. In the future research, we will study and compare Kubernetes CNI plugins performance in other virtualised environments (KVM, Xen, Hyper-V) and see how CNI plugins performance depends on VMs characteristics.

References

Ducastel, A. (2018). *Benchmark results of Kubernetes network plugins (CNI) over 10Gbit/s network*. https://itnext.io/benchmark-results-of-kubernetes-network-plugins-cni-over-10gbit-s-network-updated-april-2019-4a9886efe9c4

Ducastel, A. (2019). *Benchmark results of Kubernetes network plugins (CNI) over 10Gbit/s network*. https://itnext.io/benchmark-results-of-kubernetes-network-plugins-cni-over-10gbit-s-network-36475925a560

Großmann, M., & Klug, C. (2017). Monitoring container services at the network edge. In *29th International Teletraffic Congress* (ITC 29) (pp. 130–133), Taiwan. https://doi.org/10.23919/ITC.2017.8064348

Li, Z., Kihl, M., Lu, Q,. & Andersson, J. A. (2017). Performance overhead comparison between hypervisor and container based virtualization. In *31st International Conference on Advanced Information Networking and Applications* (pp. 955–962), Taiwan. https://doi.org/10.1109/AINA.2017.79

Park, V., & Yang, H. (2018). Performance analysis of CNI (Container Networking Interface). In *International Conference of ICT Convergence* (pp. 248–250), Korea. https://doi.org/10.1109/ICTC.2018.8539382

Zeng, H., Wang, B., Deng, W., & Zhang, W. (2017). Measurement and evaluation for docker container networking. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 105–108), China. https://doi.org/10.1109/CyberC.2017.78

KUBERNETES CNI ĮSKIEPIŲ NAŠUMO APŽVALGA

N. Kapočius

Santrauka

Pastaruoju metu programų konteinerizacija ir mikropaslaugų architektūra tampa vis populiaresnė. Konteinerizuotos programos dėl virtualizacijos teikiamų lankstumo privalumų diegiamos tiek privačiuose, tiek viešuosiuose duomenų centruose naudojant virtualias mašinas. Tačiau mikropaslaugomis grindžiamos programos pasižymi dideliu konteinerių skaičiumi, juos reikia sustyguoti. Vienas populiariausių konteinerių sustygavimo įrankių yra „Kubernetes“. Tačiau šis sprendimas neturi vieno numatyto tinklo įgyvendinimo sprendimo ir remiasi CNI įskiepių modeliu. Norint pasirinkti geriausią CNI įskiepį, jų našumą reikia palyginti. Šiame straipsnyje devynių populiariausių CNI įskiepių TCP ir HTTP protokolų našumas lyginamas fizinėje bei virtualizuotoje „VMware ESXi“ duomenų centro infrastruktūroje. Tyrimo rezultatai padeda pasirinkti, kuriuos CNI įskiepius geriau naudoti fizinėje bei virtualizuotoje infrastruktūroje.

Reikšminiai žodžiai: „Kubernetes“, sustygavimas, įskiepis, našumas, konteinerių tinklų sąsaja (CNI), „VMware“, debesija.