Server: Red Hat, Intel and SUSE

By *Roy Schestowitz*
Created *10/09/2019 - 9:10pm*
Submitted by Roy Schestowitz on Tuesday 10th of September 2019 09:10:36 PM Filed under Linux [1] Red Hat [2] Server [3] SUSE [4]

- **Introduction to virtio-networking and vhost-net [5]**

  In this post we have scratched the surface of the virtio-networking ecosystem, introducing you to the basic building blocks of virtualization and networking used by virtio-networking. We have briefly covered the virtio spec and the vhost protocol, reviewed the frontend and backend architecture used for implementing the virtio interface and have taken you through the vhost-net/virtio-net architecture of vhost-net (host kernel) communicating with virtio-net (guest kernel).

  A fundamental challenge we had when trying to explain things was the historical overloading of terms. As one example, virtio-net refers both to the virtio networking device implementation in the virtio specification and also to the guest kernel front end described in the vhost-net/virtio-net architecture. We attempted to address this by explaining the context of terms and using virtio-net to only describe the guest kernel frontend.

  As will be explained in later posts, there are other implementations for the virtio spec networking device based on using DPDK and different hardware offloading techniques which are all under the umbrella of the virtio-networking.

  The next two posts are intended to provide a deeper understanding of the vhost-net/virtio-net

architecture. One post will be intended for architects providing a technical deep dive into the vhost-net/virtio-net and explaining how in practice the data plane and control planes are implemented. The other post intended for developers will be a hands on session including Ansible scripts to enable experimenting with the vhost-net/virtio-net architecture.

If you prefer high level overviews we recommend you keep an eye out for the virtio-networking and DPDK introductions, to be published in the upcoming weeks.

- **Intel Issues Second Release Of Its Rust-Written Cloud-Hypervisor For Modern Linux VMs [6]**

  Intel's open-source crew has released version 0.2 of its primarily Rust-developed Cloud Hypervisor and associated firmware also in Rust.

  The Intel Cloud Hypervisor is their experimental VMM running atop KVM designed for modern Linux distributions and VirtIO para-virtualized devices without any legacy device support.

- **Announcing SUSE CaaS Platform 4 [7]**

  SUSE CaaS Platform 4 raises the bar for robust Kubernetes platform operations with enhancements that expand platform scalability options, strengthen application security, and make it easier to keep pace with technology advancements. Integrating the latest releases of Kubernetes and SUSE Linux Enterprise, SUSE CaaS Platform 4 continues to provide industry leading application delivery capabilities as an enterprise-ready solution.

- **A new era in Cloud Native Application Delivery is here [8]**

- **3 Infrastructure Compliance Best Practices for DevOps [9]**

  For most IT organizations, the need for compliance goes without saying. Internal corporate policies and external regulations like HIPAA and Sarbanes Oxley require compliance. Businesses in heavily regulated industries like healthcare, financial services, and public service are among those with the greatest need for strong compliance programs.

Linux Red Hat Server SUSE

Links:
[1] http://www.tuxmachines.org/taxonomy/term/63
[2] http://www.tuxmachines.org/taxonomy/term/142
[3] http://www.tuxmachines.org/taxonomy/term/147
[4] http://www.tuxmachines.org/taxonomy/term/117
[5] https://www.redhat.com/en/blog/introduction-virtio-networking-and-vhost-net
[6] https://www.phoronix.com/scan.php?page=news_item&px=Intel-Cloud-Hypervisor-0.2
[7] https://www.suse.com/c/announcing-suse-caas-platform-4/
[8] https://www.suse.com/c/a-new-era-in-cloud-native-application-delivery-is-here/
[9] https://www.suse.com/c/3-infrastructure-compliance-best-practices-for-devops/