Hyperscale networking in a smallish datacentre

Freek Dijkstra (SURFsara) and Sander Boele (SURFsara)

keywords: datacentre, automation, provisioning, tooling, hpc

Summary

Academic datacentres are nowhere near as large as Google, Amazon or Facebook. Yet a datacentre of 100 racks can deploy many of the same techniques as a datacentre of 10,000 racks, and reap the same benefits. In this talk, we'll explain how we took the lessons from our big friends, and applied them to our datacentre. In a few years, SURFsara (the Dutch national supercomputing centre with 170 racks) migrated from a network consisting of traditional vendor brands like Juniper, Cisco and Arista, with mostly manual NOC operations, to a datacentre with white label network devices, automated tooling, a scalable Clos network with overlay network, and telemetry-based monitoring and metrics collection. In this talk, we describe the steps we took so far at SURFsara, what steps are still ahead of us, and what we think the future holds for networking in medium- or small-scale datacentres. This paper describes our results. While we try to generalise our findings to other networks, your mileage may vary.

Automation

The operations of many network engineers have rapidly changed over the last few years due to automation. Logging in to individual network devices has largely been replaced by configuration management tools that automatically push the configuration to the devices. Examples of these tools are Puppet, Chef, SaltStack, CfEngine and Ansible. The SURFsara network team decided to use Ansible. The benefits of these tools are clear: reduced OPEX, and fewer human errors. This requires network engineers to be trained into learning Ansible. They become more like system administrators, and one of the advantages in our organisation is that the shared tooling means more interaction between the network team and the sysadmin teams. Further integration of the teams is not likely. Not only are network skills still required, we also don't expect to reach the 100% automation figure anytime soon. SURFsara has currently automated 70% of its network devices, and we expect to reach over 90%, but there are still devices where it is not beneficial to automate them, because automation only pays off at scale, and there are only a few of these devices. In our network, this is the case for our two core routers and the Wi-Fi controllers, but we see similar situations in other networks, with e.g. DWDM equipment, load balancers or firewalls.

The uptake of configuration management tools has been accelerated by open network equipment, such as white label switches running one of the many ONIE-compatible network operating systems. At SURFsara, we use Cumulus Linux. However, automation it is not limited to these devices, and most traditional network operating systems have nice libraries and example scripts to work with their devices. As an example, last fall we created Ansible playbooks to bring our Juniper QFX switches under automated management. We also added L3 functionality to our Ansible playbooks, where it was previously limited to L1 and L2. By using this step-by-step approach, we ensure a relatively smooth transition.

Hyperscale network

Automation only pays off at scale, where you do many changes on a few devices or when you have many similar devices to manage. One of the limitations specific to SURFsara is that the SURFsara network used to be rather heterogeneous, with each compute and data service running on their own cluster, with their own tailor-made network. While this ensured the best high performance compute (HPC) environment, it did hurt scalability and limited our ability to deploy automation.

However, now that compute hardware is becoming even more generic, the need for a unified computing model prevails, where the services can be run on more-or-less any hardware in our datacentre. This also eliminates the need for dedicated network solutions, and we could introduce a standard product portfolio for the network.

In an ideal unified compute architecture, a container, virtual machine or bare metal service could migrate live from any one server to any other server in the datacentre. This requires two features from the network: it should be possible to dynamically migrate the IP (or VLAN) from any one server to any other server in the datacentre, and the bandwidth and latency should be similar for any two servers far apart in the datacentre as for two servers in the same rack.

One solution to these two problems is to build a Clos network with overlay. The advantage of a Clos network is that it can easily scale vertically in bandwidth (by adding network devices in the distribution layers) as well as horizontally in the number of connected servers (by adding network devices in the access layer). The Clos network we decided upon adds an extra layer compared to our existing spine-leaf design. We decided not to build a fat tree network yet, meaning there will be an overbooking factor (the edges have a higher combined capacity than the core). We expect that in the near future, most services are still coupled to neighbouring racks and because of statistical multiplexing we will not see any congestion issues. If this situation changes later, it is straightforward to add more networking devices and reduce the overbooking. The overlay network ensures the decoupling of logical name (the IP address of the VM) and the physical location (the IP address of the rack of hypervisor).

Technology

For the dataplane of the overlay network we choose vxLAN, which does encapsulation of layer 2 traffic in UDP and thus allows it to be routed across an IP network. The end-points of the overlay network, the vTAPs, will reside in the top-of-rack network devices, although it is possible to terminate the vTAPs in the hypervisors as well with routing-to-the host solutions like OpenStack networking-bagpipe, OpenContrail and Cumulus Host Pack. This requires integrating the system administration tools with the network administration tools and is something we are looking into.

One of the advantages of vxLAN is that it is also possible to tunnel Layer 2 traffic between different datacentres over the regular Layer 3 internet. This functionality is now provided by VPN's or lightpaths. There are currently no plans to replace this, since the VPN's and lightpaths we use offer the added advantage of a secure connection, while a vxLAN tunnel is not encrypted.

The reason to choose a layer 2-over-layer 3 solution instead of a layer 3-over-layer 3 solution is that layer 2 is commonly used by our systems, not only for DHCP-based bootstrapping and management of servers, but also as a grouping mechanism to deploy security policies with ACL's. In general, there is a trend to replace layer 2 Ethernet by layer 3-only networks, but we don't feel we're ready for that step yet.

While vxLAN defines the data frames on the wire. In addition, we use BGP-EVPN as the control plane routing protocol to distribute information about the location of MAC addresses behind vTAPs.

Since a Clos network provides multiple routes to reach the same destination, a traffic engineering solution needs to be in place to decide on the best route for a traffic flow. This is a hot topic in large datacentres, with Microsoft's CONGA a popular choice, although there are many alternatives (Hedera, HULA, MPTCP, Presto, LetFlow, etc.). However, none of these protocols are yet supported by merchant silicon ASIC's and Mellanox Spectrum with Cumulus Linux, so we simply opt for ECMP using a flow-based round-robin scheme till these protocols are supported. The industry is picking up on it though, as Broadcom's Trident III will have more dynamic hashing of ECMP traffic.

Monitoring

The added complexity of an overlay network warrants an overhaul of our monitoring system. Our current monitoring system is still largely based on SNMP polling and round-robin databases, while modern monitoring systems are based on streaming telemetry data by an agent locally installed on the devices via a message bus of some kind or directly into a time series database. While we did already deploy InfluxDB for some of our time-series data, and Graphana for the dashboard, the message bus has not been decided upon. Likely we will choose between a system based on Telegraph with no message bus, some DIY python based agent with Apache Kafka, or deploy a Sensu implementation that uses RabbitMQ for messaging.

Continuous Integration

We are looking into continuous integration and continuous deployment of our network. While it is feasible to virtualise a full network with test hosts, we have set up a testbed and plan to first deploy our Ansible playbooks in this testbed, and do automated connectivity and performance checks, before we roll out the playbook in our production environment. This way we can also test the different ASIC's we have in use, while in a virtualized environment we can only test the software. While this will not cover all possible deployment issues, we expect that this will be the first step towards further automation and hope to integrate these findings with the monitoring in the future. That way, we ensure the health of our network before and after deployment of changes in our production environment.

We plan to present further progress on our monitoring and continuous integration setup at TNC18.

Biography

(This work is co-authored by Freek Dijkstra and Sander Boele. The presenter is Freek Dijkstra)

Freek Dijkstra leads the networking and datacentre group at SURFsara. After migrating compute facilities to a brand-new datacentre in 2016, the focus of his team is now to migrate the network to a fully automated datacentre network.
Before working at SURFsara, Dijkstra earned his PhD at the university of Amsterdam while working on multi-domain, multi-layer network descriptions. As co-chair of the OGF NML working group (2009-2013), he turned these ideas into an open standard, called Network Markup Language (NML).
Freek Dijkstra is married, has two children, and plays board games in his spare time.

Sander Boele is network expert at SURFsara. He loves getting his hands dirty on tools and equipment to bring the datacentre network to the next level. Ideally, he likes to automate himself away, and let his users be in control.
Boele earned his MSc in cognitive psychology at the Vrije Universiteit Amsterdam before figuring out that computer network where just as cool as neural networks.
In his spare time, Sander Boele enjoys making things. He has designed custom-made PCBs and has two 3D printers at home.