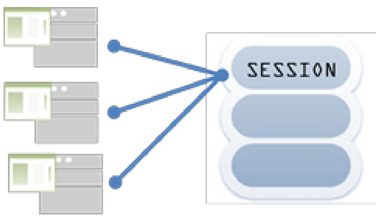
Midokura - The SDN with a Hive Mind

Lori MacVittie, 2012-17-10

#SDN #cloud #openstack Centralized control, decentralized execution comes to life with Midokura's MidoNet

Whether bees or Martians, science or science-fiction, the notion of a hive mind is one that pops up frequently within the realm of psychology, philosophy, theology, science and, last but not least, technology. A hive mind is one that has a collective memory, sharing information from the past and present with every other member of the hive.



This capability (if it really exists) enables incredible resiliency on the population as a whole, because every member of the population has the information necessary to replace another at any moment. This concept has been applied to scaling applications since scaling applications because a necessity. If applications share session state information – usually by sharing a session data base – then any instance can immediately take over for another without disrupting a user session. Like bees, there is no need for on-the-job-training, it just "knows" – as though it tapped into a shared database full of not only standard hive knowledge but of the current state of the hive.

This concept is partially included in many SDN implementations, with varying degrees of success. In the most common, centralized-controller model of SDN a singular entity (the controller) maintains this vault of knowledge but disseminates only partial views of that state to relevant pieces of the infrastructure. Thus it is not a fully participative hive mind, but a partial one. This leads to over-reliance on the controller, which is responsible not just for management of the shared knowledge but of dissemination. Like the queen bee, loss of the controller is devastating to the ability of the controller-focused SDN to function.

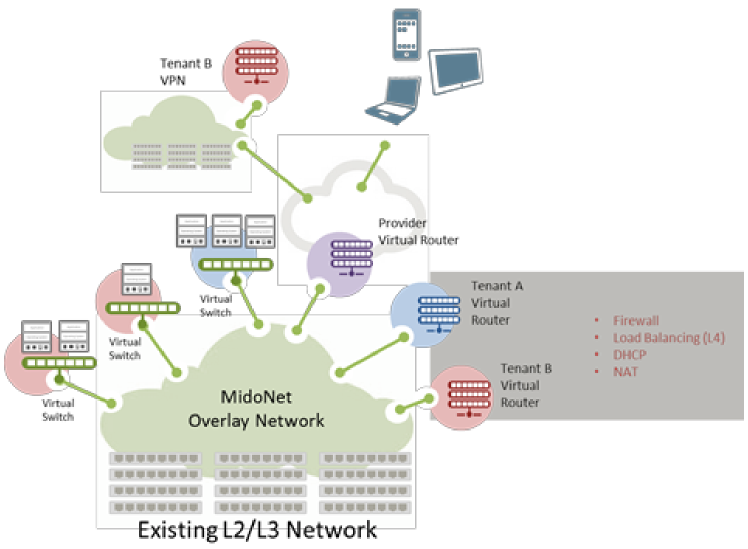
Midokura, offers a new model with a more complete collective "hive mind" that inherently supports resilient software-defined networks and alleviates the potential risk of relying on a singular entity through which to disseminate state of the network.

MidoNet

Midokura is a global startup focused on network virtualization. It officially entered the US market in mid-October 2012 with the introduction of its primary solution: MidoNet.

> MidoNet virtualizes the network stack for popular cloud platforms such as OpenStack®. Midokura's approach not only adds automation that significantly reduces the human cost (OPEX) of managing the network, but also impacts the overall economics of cloud computing (CAPEX) by simplifying network requirements.
>
> MidoNet is a distributed, de-centralized, multi-layer software defined virtual network solution for IaaS. By taking an overlay-based approach to network virtualization, MidoNet sits on top of any IP-connected network, and pushes the network intelligence to the edge of the network, in software.
>
> -- Midokura Press Release

Cutting through the marketing speak, MidoNet is a fabric of distributed, software-defined networking services. It requires no specialized hardware infrastructure, but rather turns any Linux-based host running the Open vSwitch kernel module and the MidoNet agent into a node on a fully-meshed, L2-4 virtual network fabric.The network executes on a role-based networking principle, with each node able to execute on a broad set of L2-4 policies based on its assigned role in the flow. By taking an overlay-based approach to network virtualization, MidoNet can be deployed atop any existing network, using traditional L2/L3 connectivity as the means to create and utilize its peer-to-peer virtualized tunnels.



MidoNet applies faithfully the idea of centralized management coupled with de-centralized execution.Traditional edge services are applied at the perimeter of the network using virtual policy execution, and then packets are routed via a tunnel to the designated end-point. Policies are not so much deployed as they are simply applied at the appropriate ingress node. Each node may play multiple roles, guided by the process governing specific flows.

Failure, then, is inherently managed by the ability of any edge node to apply the appropriate policies based on the role being executed. There is no reliance on a controller - commonly associated with SDN implementations – because local agents manage the application of appropriate policies on ingress and egress traffic. It's a "shared session" approach to networking, in which the entire state of the network is stored in scalable database systems and distributed throughout the network. Just as is the case with "shared session" applications, failure in any given node simply means flows are directed through a different node – which has complete knowledge of all the information previously known to the failed node by virtue of sharing the network state database.

Like a hive mind, every node knows what every other node knows – and has known – and it is only the roles assigned to any given node that indicates a difference in how that node executes on traffic.

The difference between MidoNet's architecture and the centralized architecture of a controller-based SDN is in the execution. While both models "share" state and configuration, ostensibly, a controller-based SDN relies on centralized execution. MidoNet does not, leveraging shared state and configuration as a means to enable resiliency.

MidoNet does not come without questions. Any agent-based system brings with it overhead, and MidoNet is no exception. The question becomes how ***much*** overhead and does it significantly impact performance of the host system. Similarly, how many roles can a single node assume before it becomes overwhelmed? How well does MidoNet react to failures in the underlying L2/L3 physical network?

And while MidoNet offers a mix of stateless and stateful services, the higher up the stack one traverses, the less robust such services become. Layer 4 load balancing as currently offered by MidoNet is acceptable for simple load balancing, but depending on the application and demand may result in uneven distribution that can make capacity planning and elasticity less efficient and more difficult to perform.

Also problematic with any simple L4 load balancing service are issues with application dependencies on persistence and topological architecture and the resulting impact on load balancing algorithms. Midokura does not refute the unique challenges associated with moving up the stack – nor with the rudimentary nature of its existing L4 services – but believes these challenges can eventually be addressed.

All in all, MidoNet is an impressive adaption of SDN principles into a more resilient, flexible model. The application of a shared session architecture combined with role-based networking is a fascinating twist on the more common centralized control and command model put forth by competing SDN players.

- MidoKura Launches MidoNet Network Virtualization Platform
- SDN Watch: Midokura Enters U.S. Market
- Mind blowing L2-L4 Network Virtualization by Midokura MidoNet
- Reactive, Proactive, Predictive: SDN Models
- SDN is Network Control. ADN is Application Control
- Integration Topologies and SDN
- Applying ‘Centralized Control, Decentralized Execution’ to Network Architecture

---