# 转载自：soul orbit: <https://r12f.com/posts/sonic-1-intro/>

# SONiC学习笔记（一）：概述与安装

这几天都在看和SONiC相关的资料，所以按照习惯，这里也简单的对学习的结果进行一个总结，希望也能对其他对SONiC感兴趣的小伙伴有所帮助。

## **1. 为什么要做SONiC**

我们知道交换机内部都有一套可大可小的操作系统，用于配置和查看交换机的状态。但是，从1986年第一台交换机面世开始，虽然各个厂商都在进行着相关的开发，到现在为止种类也相当的多，但是依然存在一些问题，比如：

* 生态封闭，不开源，主要是为了支持自家的硬件，无法很好的兼容其他厂商的设备
* 支持的场景很有限，难以使用同一套系统去支撑大规模的数据中心中复杂多变的场景
* 升级可能会导致网络中断，难以实现无缝升级，这对于云提供商来说有时候是致命的
* 设备功能升级缓慢，难以很好的支持快速的产品迭代

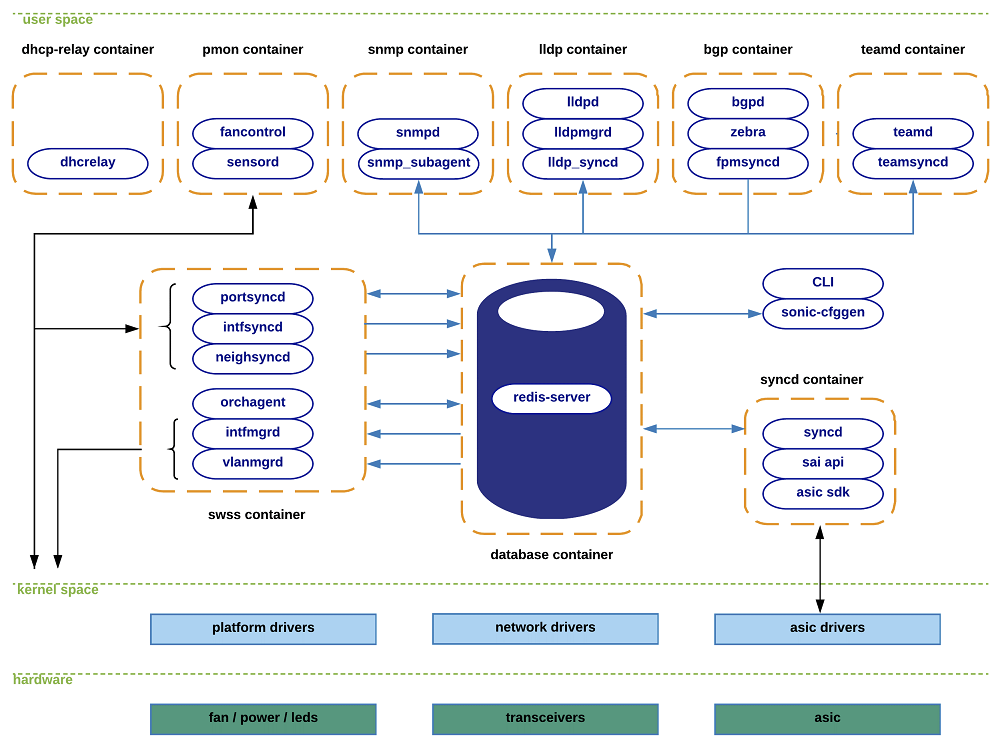
所以，微软在2016年开源了SONiC，希望能够通过开源的方式，让SONiC能够成为一个通用的网络操作系统，从而解决上面的问题。而且，由于Azure的背书，也能保证SONiC确实能够承受大规模的生产环境的考验，这也是SONiC的一个优势。

## **2. 主体架构**

SONiC是微软开发的基于debian的开源的网络操作系统，它的设计核心思想有三个：

* ****硬件和软件解耦****：通过SAI（Switch Abstraction Interface）将硬件的操作抽象出来，从而使得SONiC能够支持多种硬件平台。这一层抽象层由SONiC定义，由各个厂商来实现。
* ****使用docker容器将软件微服务化****：SONiC上的主要功能都被拆分成了一个个的docker容器，和传统的网络操作系统不同，升级系统可以只对其中的某个容器进行升级，而不需要整体升级和重启，这样就可以很方便的进行升级和维护，支持快速的开发和迭代。
* ****使用redis作为中心数据库对服务进行解耦****：绝大部分服务的配置和状态最后都被存储到中心的redis数据库中，这样不仅使得所有的服务可以很轻松的进行协作（数据存储和pubsub），也可以让我们很方便的在上面开发工具，使用统一的方法对各个服务进行操作和查询，而不用担心状态丢失和协议兼容问题，最后还可以很方便的进行状态的备份和恢复。

这让SONiC拥有了非常开放的生态（[Community](https://sonic-net.github.io/SONiC/index.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)，[Devices](https://sonic-net.github.io/SONiC/Supported-Devices-and-Platforms.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)，[Workgroups](https://sonic-net.github.io/SONiC/workgroups.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)），总体而言，SONiC的架构如下图所示 [[1]](https://github.com/sonic-net/SONiC/wiki/Architecture" \t "https://r12f.com/posts/sonic-1-intro/_blank)：



当然，这样的设计也有一些缺点，比如：对磁盘的占用会变大，不过，现在一点点存储空间并不是什么很大的问题，而且这个问题也都可以通过一些方法来解决。

## **3. 安装**

### **3.1. 硬件交换机**

如果你自己就拥有一台交换机，或者想购买一台交换机，在上面安装SONiC，那么请认真阅读这一小节，否则可以自行跳过。:D

#### **3.1.1. 交换机选择和SONiC安装**

首先，请确认你的交换机是否支持SONiC，SONiC目前支持的交换机型号可以在[这里](https://sonic-net.github.io/SONiC/Supported-Devices-and-Platforms.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)找到，如果你的交换机型号不在列表中，那么就需要联系厂商，看看是否有支持SONiC的计划。有很多交换机是不支持SONiC的，比如：

* 普通针对家用的交换机，这些交换机的硬件配置都比较低（即便支持的带宽很高，比如[MikroTik CRS504-4XQ-IN](https://mikrotik.com/product/crs504_4xq_in" \t "https://r12f.com/posts/sonic-1-intro/_blank)，虽然它支持100GbE网络，但是它只有16MB的Flash存储和64MB的RAM，所以基本只能跑它自己的RouterOS了）。
* 有些虽然是数据中心用的交换机，但是可能由于型号老旧，厂商并没有计划支持SONiC。

对于安装过程，由于每一家厂商的交换机设计不同，其底层接口各有差别，所以，其安装方法也都有所差别，这些差别主要集中在两个地方：

* 每个厂商都会有自己的[SONiC Build](https://sonic-net.github.io/SONiC/Supported-Devices-and-Platforms.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)，还有的厂商会在SONiC的基础之上进行扩展开发，为自己的交换机支持更多的功能，比如：[Dell Enterprise SONiC](https://www.dell.com/en-us/shop/povw/sonic" \t "https://r12f.com/posts/sonic-1-intro/_blank)，[EdgeCore Enterprise SONiC](https://www.edge-core.com/sonic.php" \t "https://r12f.com/posts/sonic-1-intro/_blank)，所以需要根据自己的交换机选择对应的版本。
* 每个厂商的交换机也会支持不同的安装方式，有一些是直接使用USB对ROM进行Flash，有一些是通过ONIE进行安装，这也需要根据自己的交换机来进行配置。

所以，虽然安装方法各有差别，但是总体而言，安装的步骤都是差不多的。请联系自己的厂商，获取对应的安装文档，然后按照文档进行安装即可。

#### **3.1.2. 配置交换机**

安装好之后，我们需要进行一些基础设置，部分设置是通用的，我们在这里简单总结一下。

##### **3.1.2.1. 设置admin密码**

默认SONiC的账号密码是admin:YourPaSsWoRd，使用默认密码显然不安全：

|  |
| --- |
| sudo passwd admin |

##### **3.1.2.2. 设置风扇转速**

数据中心用的交换机风扇声音都特别的大！比如，我用的交换机是Arista 7050QX-32S，上面有4个风扇，最高能到每分钟17000转，放在车库中，高频的啸叫即便是在二楼隔着3面墙还是能听得到，所以如果你是在家使用的话，建议对其进行一些设置，将转速调低。

可惜，[由于SONiC并没有cli对风扇转速的规则进行控制](https://github.com/sonic-net/SONiC/blob/master/thermal-control-design.md" \t "https://r12f.com/posts/sonic-1-intro/_blank)，所以我们需要通过手动修改pmon容器中的配置文件的方式来进行设置。

|  |
| --- |
| *# Enter pmon container* sudo docker exec -it pmon bash *# Use pwmconfig to detect all pwm fans and create configuration file. The configuration file will be created at /etc/fancontrol.* pwmconfig *# Start fancontrol and make sure it works. If it doesn't work, you can run fancontrol directly to see what's wrong.* VERBOSE=1 /etc/init.d/fancontrol start VERBOSE=1 /etc/init.d/fancontrol status *# Exit pmon container* exit  *# Copy the configuration file from the container to the host, so that the configuration will not be lost after reboot.* *# This command needs to know what is the model of your switch, for example, the command I need to run here is as follows. If your switch model is different, please modify it yourself.* sudo docker cp pmon:/etc/fancontrol /usr/share/sonic/device/x86\_64-arista\_7050\_qx32s/fancontrol |

##### **3.1.2.3. 设置交换机Management Port IP**

一般的数据中心用的交换机都提供了Serial Console连接的方式，但是其速度实在是太慢了，所以我们在安装完成之后，都会尽快的把Management Port给设置好，然后通过SSH的方式来进行管理。

一般来说，management port的设备名是eth0，所以我们可以通过SONiC的配置命令来进行设置：

|  |
| --- |
| *# sudo config interface ip add eth0 <ip-cidr> <gateway>* *# IPv4* sudo config interface ip add eth0 192.168.1.2/24 192.168.1.1 *# IPv6* sudo config interface ip add eth0 2001::8/64 2001::1 |

##### **3.1.2.4. 创建网络配置**

新安装完的SONiC交换机会有一个默认的网络配置，这个配置有很多问题，比如对于10.0.0.0的IP的使用，如下：

|  |
| --- |
| admin@sonic:~$ show ip interfaces Interface Master IPv4 address/mask Admin/Oper BGP Neighbor Neighbor IP ----------- -------- ------------------- ------------ -------------- ------------- Ethernet0 10.0.0.0/31 up/up ARISTA01T2 10.0.0.1 Ethernet4 10.0.0.2/31 up/up ARISTA02T2 10.0.0.3 Ethernet8 10.0.0.4/31 up/up ARISTA03T2 10.0.0.5 |

所以我们需要创建一个新的网络配置，然后将我们使用的Port都放入到这个网络配置中。这里简单的方法就是创建一个VLAN，使用VLAN Routing：

|  |
| --- |
| *# Create untagged vlan* sudo config vlan add 2 *# Add IP to vlan* sudo config interface ip add Vlan2 10.2.0.0/24 *# Remove all default IP settings* show ip interfaces | tail -n +3 | grep Ethernet | awk '{print "sudo config interface ip remove", $1, $2}' > oobe.sh; chmod +x oobe.sh; ./oobe.sh *# Add all ports to the new vlan* show interfaces status | tail -n +3 | grep Ethernet | awk '{print "sudo config vlan member add -u 2", $1}' > oobe.sh; chmod +x oobe.sh; ./oobe.sh *# Enable proxy arp, so switch can respond to arp requests from hosts* sudo config vlan proxy\_arp 2 enabled *# Save config, so it will be persistent after reboot* sudo config save -y |

这样就完成了，我们可以通过show vlan brief来查看一下：

|  |
| --- |
| admin@sonic:~$ show vlan brief +-----------+--------------+-------------+----------------+-------------+-----------------------+ | VLAN ID | IP Address | Ports | Port Tagging | Proxy ARP | DHCP Helper Address | +===========+==============+=============+================+=============+=======================+ | 2 | 10.2.0.0/24 | Ethernet0 | untagged | enabled | | ... | | | Ethernet124 | untagged | | | +-----------+--------------+-------------+----------------+-------------+-----------------------+ |

##### **3.1.2.5. 配置主机**

如果你家里只有一台主机使用多网口连接交换机进行测试，那么我们还需要在主机上进行一些配置，以保证流量会通过网卡，流经交换机，否则，请跳过这一步。

这里网上的攻略很多，比如使用iptables中的DNAT和SNAT创建一个虚拟地址，但是过程非常繁琐，经过一些实验，我发现最简单的办法就是将其中一个网口移动到一个新的网络命名空间中，就可以了，即便使用的是同一个网段的IP，也不会有问题。

比如，我家使用的是Netronome Agilio CX 2x40GbE，它会创建两个interface：enp66s0np0和enp66s0np1，我们这里可以将enp66s0np1移动到一个新的网络命名空间中，再配置好ip地址就可以了：

|  |
| --- |
| *# Create a new network namespace* sudo ip netns add toy-ns-1 *# Move the interface to the new namespace* sudo ip link set enp66s0np1 netns toy-ns-1 *# Setting up IP and default routes* sudo ip netns exec toy-ns-1 ip addr add 10.2.0.11/24 dev enp66s0np1 sudo ip netns exec toy-ns-1 ip link set enp66s0np1 up sudo ip netns exec toy-ns-1 ip route add default via 10.2.0.1 |

这样就可以了，我们可以通过iperf来测试一下，并在交换机上进行确认：

|  |
| --- |
| *# On the host (enp66s0np0 has ip 10.2.0.10 assigned)* $ iperf -s --bind 10.2.0.10 *# Test within the new network namespace* $ sudo ip netns exec toy-ns-1 iperf -c 10.2.0.10 -i 1 -P 16 ------------------------------------------------------------ Client connecting to 10.2.0.10, TCP port 5001 TCP window size: 85.0 KByte (default) ------------------------------------------------------------ ... [SUM] 0.0000-10.0301 sec 30.7 GBytes 26.3 Gbits/sec [ CT] final connect times (min/avg/max/stdev) = 0.288/0.465/0.647/0.095 ms (tot/err) = 16/0 *# Confirm on switch* admin@sonic:~$ show interfaces counters  IFACE STATE RX\_OK RX\_BPS RX\_UTIL RX\_ERR RX\_DRP RX\_OVR TX\_OK TX\_BPS TX\_UTIL TX\_ERR TX\_DRP TX\_OVR ----------- ------- ---------- ------------ --------- -------- -------- -------- ---------- ------------ --------- -------- -------- --------  Ethernet4 U 2,580,140 6190.34 KB/s 0.12% 0 3,783 0 51,263,535 2086.64 MB/s 41.73% 0 0 0  Ethernet12 U 51,261,888 2086.79 MB/s 41.74% 0 1 0 2,580,317 6191.00 KB/s 0.12% 0 0 0 |

### **3.2. 软件交换机**

如果你没有交换机，想自己在家里虚拟一个玩，那么可以参考我的这篇文章：[通过GNS3快速搭建本地SONiC Lab](https://r12f.com/posts/sonic-local-lab-gns3/)。配置过程还是比较简单的，大概半个小时左右就可以搞定（主要是等待下载或者导入镜像之类的）。

## **4. 发展方向**

虽然交换机已经发展很多很多年了，但是随着现在云的发展，对网络的要求也越来越高，不管是直观的需求，比如更大的带宽，更大的容量，还是最新的研究，比如，带内计算，端网融合等等，都对交换机的发展提出了更高的要求和挑战，也促使着各大厂商和研究机构不断的进行创新。SONiC也一样，随着时间的发展，需求一点没有减少。

关于SONiC的发展方向，我们可以在它的[roadmap](https://github.com/sonic-net/SONiC/wiki/Sonic-Roadmap-Planning" \t "https://r12f.com/posts/sonic-1-intro/_blank)中看到。如果大家对最新的动态感兴趣，也可以关注它的Workshop，比如，最近的[OCP Global Summit 2022 - SONiC Workshop](https://www.youtube.com/playlist?list=PLAG-eekRQBSjwK0DpyHJs76gOz1619KqW" \t "https://r12f.com/posts/sonic-1-intro/_blank)。这里就不展开了。

## **5. 总结**

好了，为了避免我的老毛病 —— 文章过长，我们就先到此为止。

这篇文章主要是对SONiC的一个简单介绍，以及如何在家里搭建一个SONiC的实验环境，以便于大家可以自己动手玩一玩。如果你对SONiC感兴趣，可以参考下面的参考资料，了解更多的信息。后面我有时间也会接着进行总结。

## **6. 参考资料**

[SONiC Architecture](https://github.com/sonic-net/SONiC/wiki/Architecture" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC Roadmap Planning](https://github.com/sonic-net/SONiC/wiki/Sonic-Roadmap-Planning" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC Landing Page](https://sonic-net.github.io/SONiC/index.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC Workgroups](https://sonic-net.github.io/SONiC/workgroups.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC Supported Devices and Platforms](https://sonic-net.github.io/SONiC/Supported-Devices-and-Platforms.html" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC User Manual](https://github.com/sonic-net/SONiC/blob/master/doc/SONiC-User-Manual.md" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[OCP Global Summit 2022 - SONiC Workshop](https://www.youtube.com/playlist?list=PLAG-eekRQBSjwK0DpyHJs76gOz1619KqW" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[Dell Enterprise SONiC Distribution](https://www.dell.com/en-us/shop/povw/sonic" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[Edgecore Enterprise SONiC Distribution](https://www.edge-core.com/sonic.php" \t "https://r12f.com/posts/sonic-1-intro/_blank)

[SONiC Thermal Control Design](https://github.com/sonic-net/SONiC/blob/master/thermal-control-design.md" \t "https://r12f.com/posts/sonic-1-intro/_blank)