Microsoft

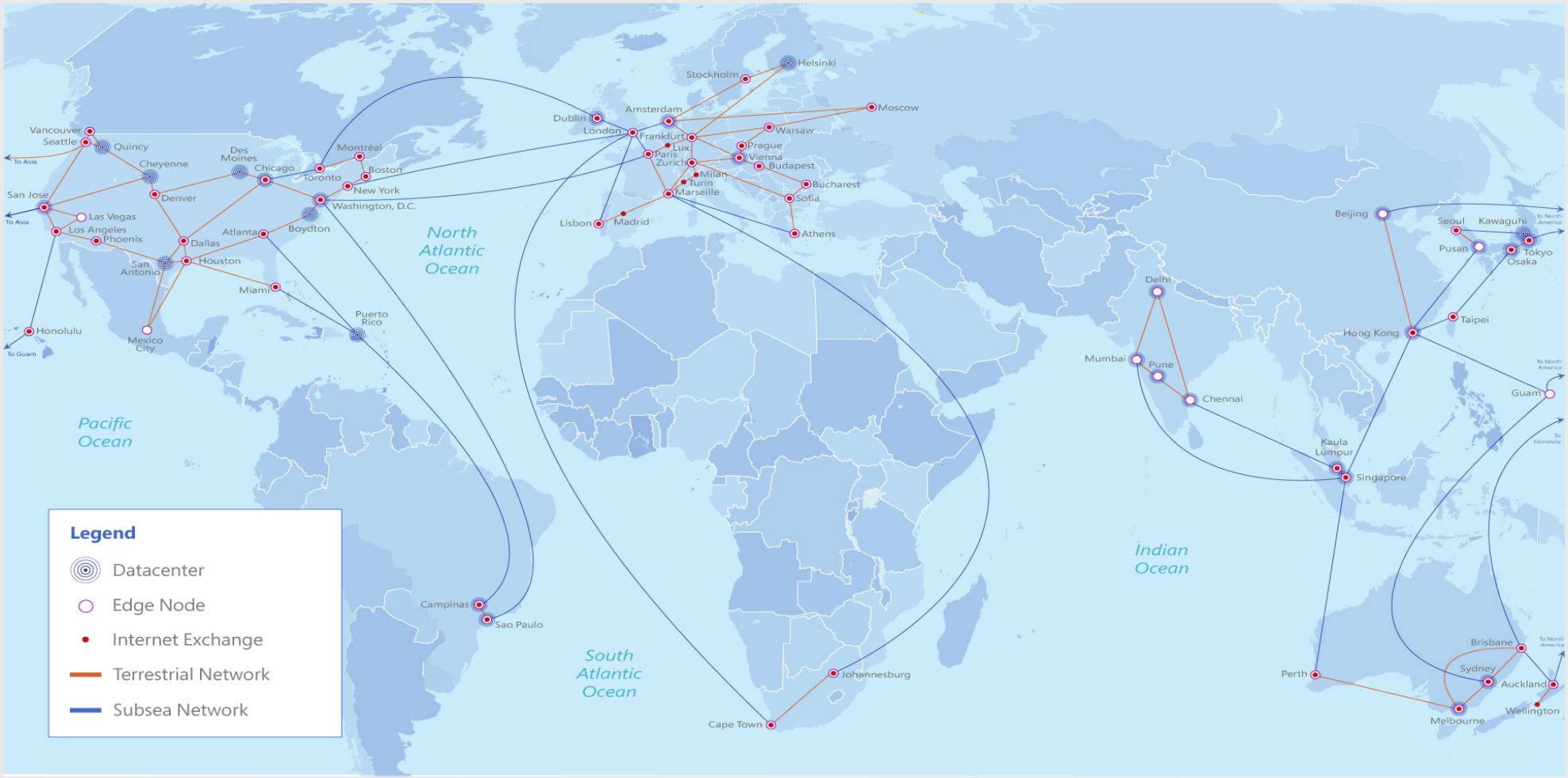
SONiC: Software for Open Networking in the Cloud

Rita Hui

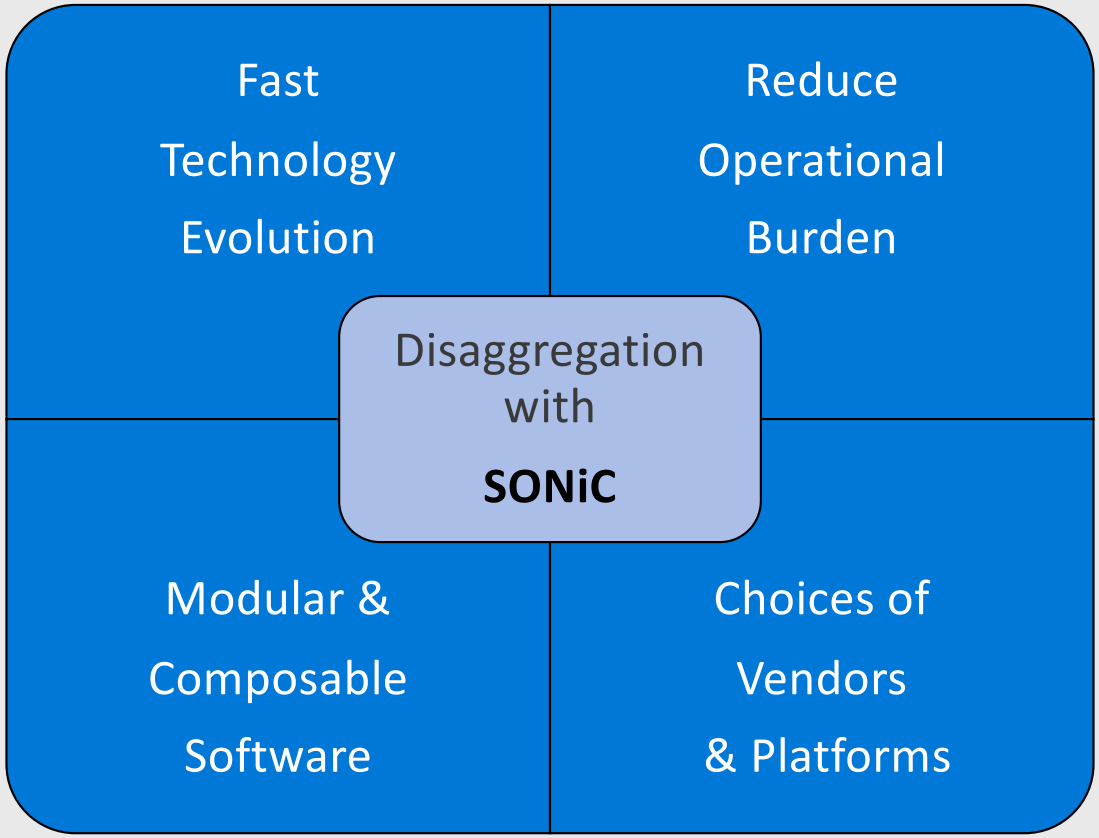
Principal Software Manager

Feb 10, 2020

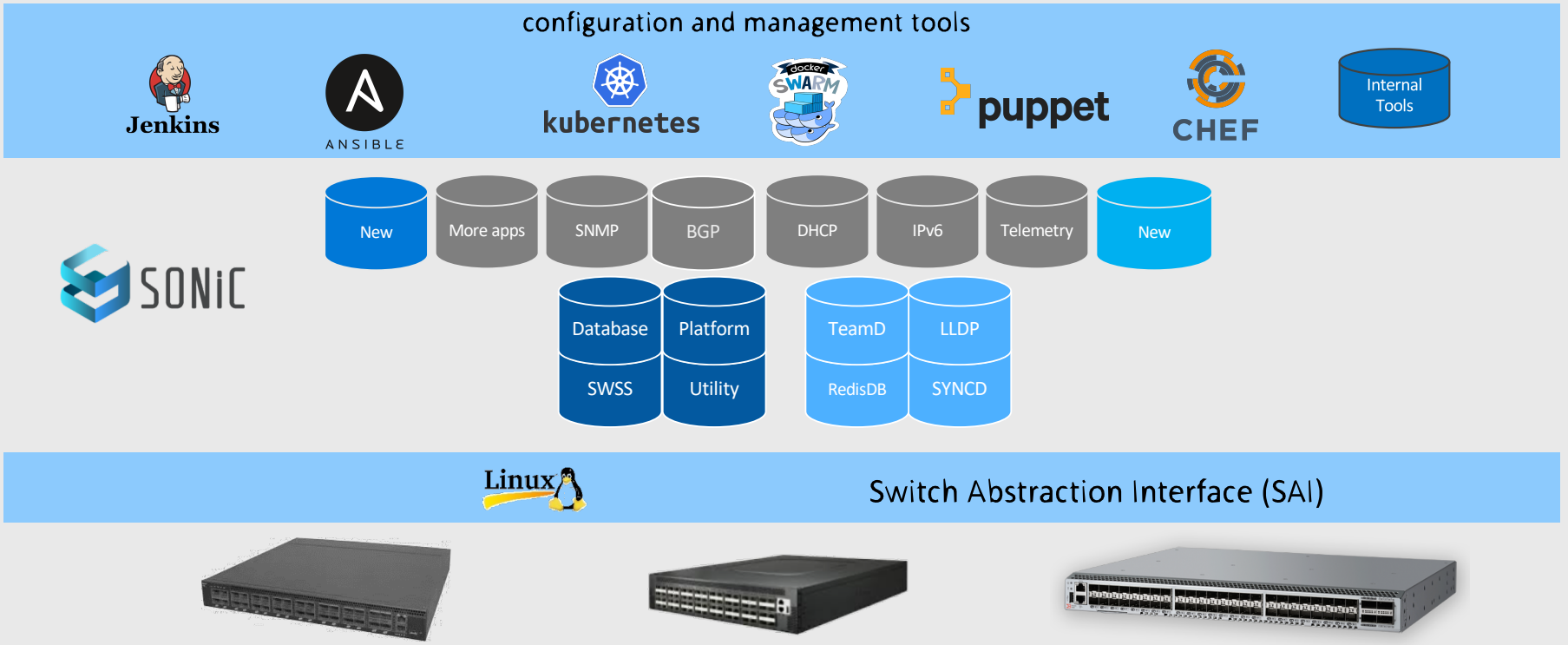
Microsoft Cloud Network – A Multi-Billion Dollar Bet



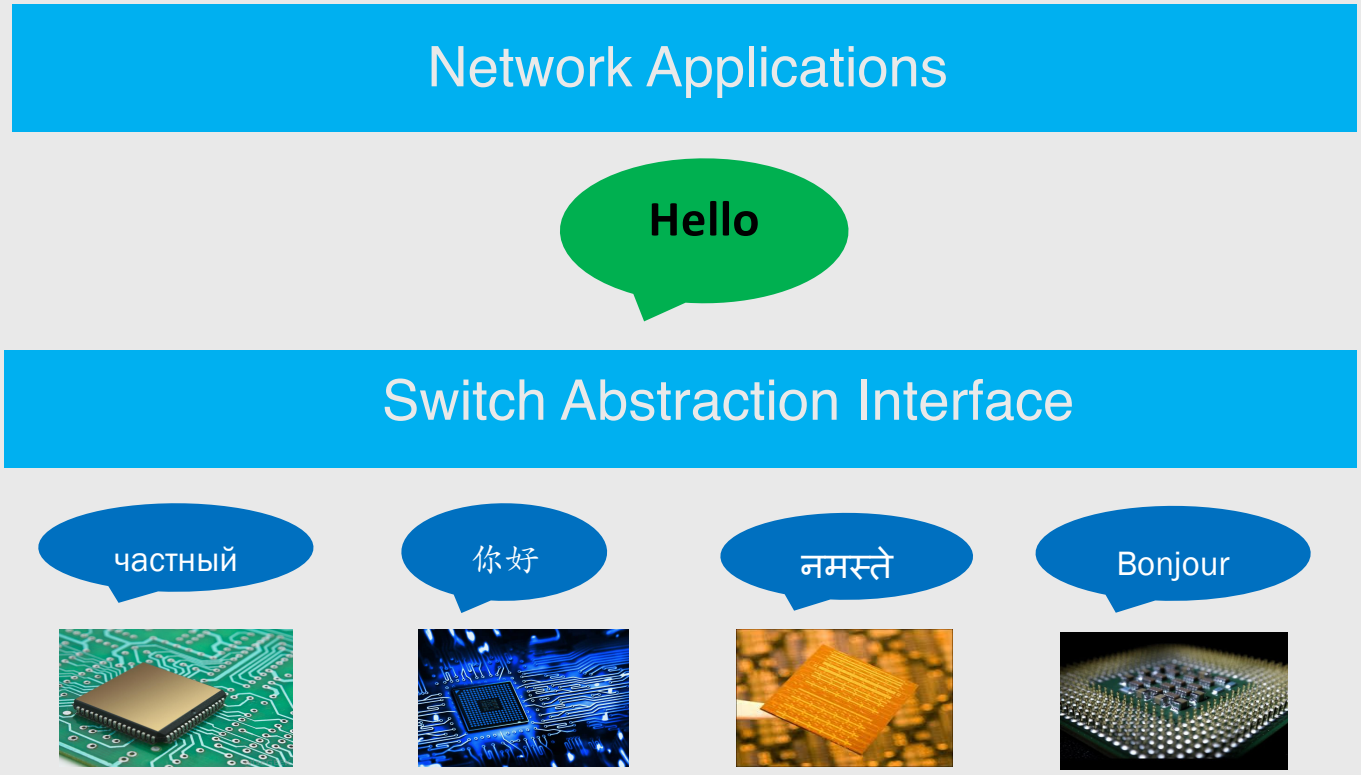
Goals of SONiC



SONiC Software for Open Networking in the Cloud



Switch Abstraction Interface (SAI)



Simple, consistent, and stable network application stack

Helps consume the underlying complex, heterogeneous hardware easily and faster

Switch Abstraction Interface

CRUD operations over extensible Entity/Attribute/Value data model

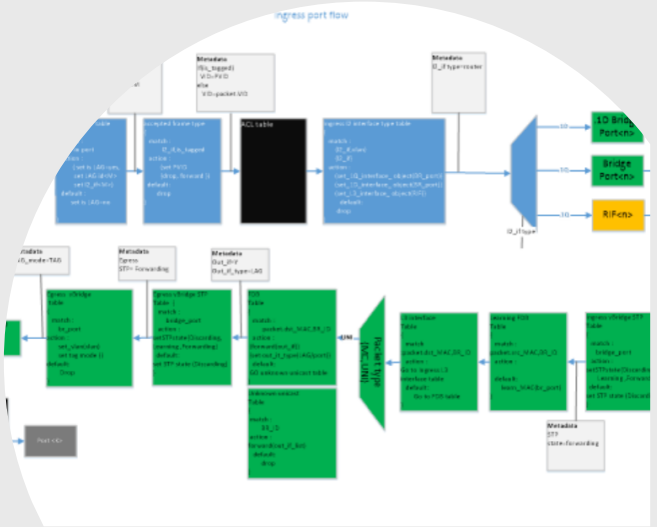
Reference data-plane behavior model supports many devices

Significant feature/partner growth since announcement in 2015

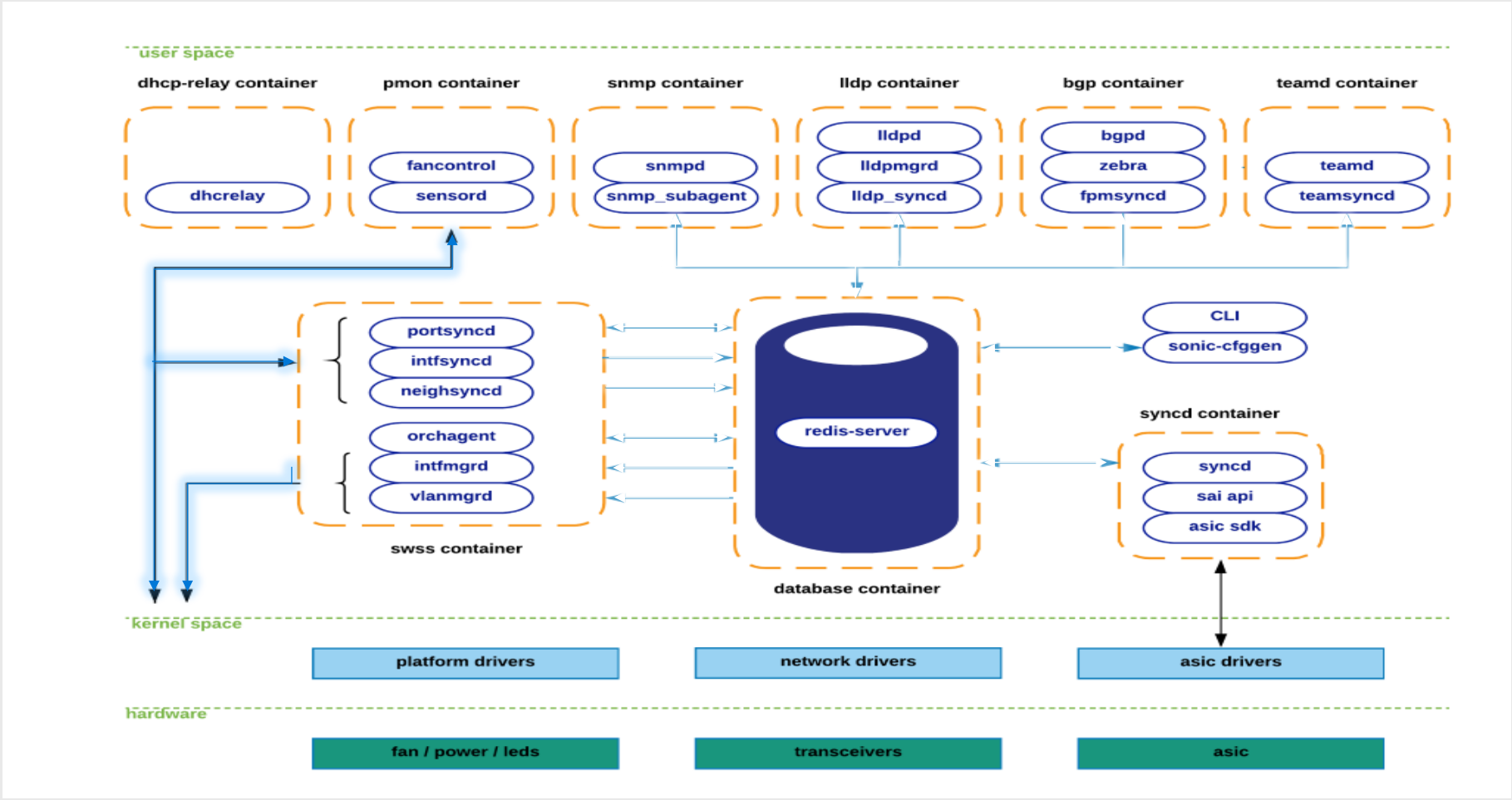
https://github.com/opencomputeproject/SAI

```
* @param[out]  bridge_port_id
* @param[in]   attr_count number of attributes
* @param[in]   attr_list array of attributes
*
* @return  SAI_STATUS_SUCCESS on success
*          Failure status code on error
*/
typedef sai_status_t (*sai_create_bridge_fn)(
    _Out_ sai_object_id_t* bridge_id,
    _In_ uint32_t attr_count,
    _In_ const sai_attribute_t *attr_list
    );

emove bridge
```



SONiC High-Level Architecture



SONiC Containerization

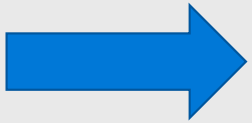
Strengths of Containers

Clean isolation

Easy deployment

Transactional

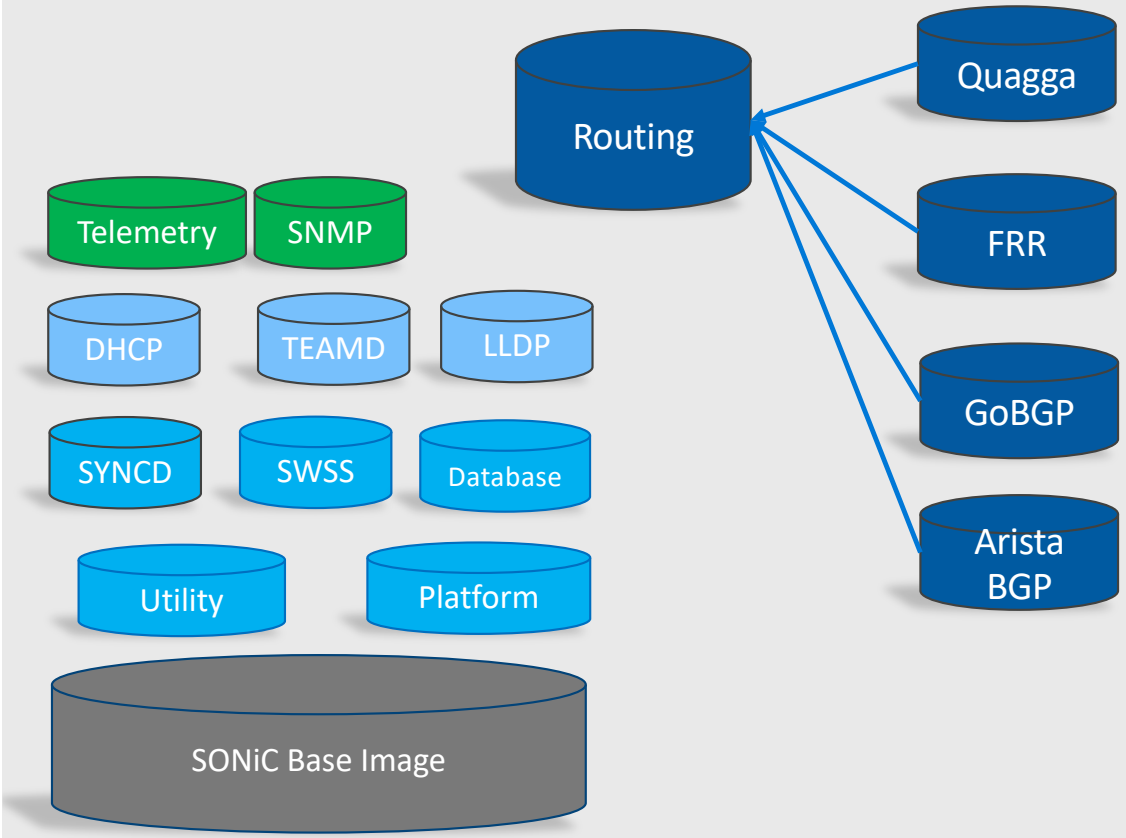
Run universally



SONiC Benefits

- Serviceability
- Extensibility
- Development agility
- Cross-platform

SONiC Containerization



Components developed in different environments

Source code may not be available

Enables choices on a per-component basis

SONiC for Network Engineers and Software Engineers

CLI-style interaction enabled by scripts written in Python

Contributed largely by Network Engineers who are using SONiC

Linux bash prompt enables direct access to containers and Redis

```
admin@str-acs-8:~$ show ip bgp summary
BGP router identifier 10.1.0.32, local AS number 65100
RIB entries 13011, using 1423 KiB of memory
Peers 48, using 218 KiB of memory

Neighbor        V    AS   MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.1        4 65200      3218      51        0    0    0 00:13:16        6402
10.0.0.5        4 65200      3217      50        0    0    0 00:12:54        6402
10.0.0.9        4 65200      3217      50        0    0    0 00:12:41        6402
10.0.0.13       4 65200      3218      51        0    0    0 00:13:12        6402
10.0.0.17       4 65200      3218      52        0    0    0 00:13:18        6402
10.0.0.21       4 65200      3218      52        0    0    0 00:13:17        6402
10.0.0.25       4 65200      3218      52        0    0    0 00:13:18        6402
10.0.0.29       4 65200      3218      52        0    0    0 00:13:18        6402
```

```
admin@str-acs-8:~$ docker ps --format "table {{.Names}}\t{{.Image}}"
NAMES               IMAGE
dhcp_relay          docker-dhcp-relay:latest
syncd               docker-syncd-brcm:latest
snmp                docker-snmp-sv2:latest
radv                docker-router-advertiser:latest
teamd               docker-teamd:latest
swss                docker-orchagent-brcm:latest
bgp                 docker-fpm-quagga:latest
pmon                docker-platform-monitor:latest
lldp                docker-lldp-sv2:latest
database            docker-database:latest
```

```
admin@str-acs-8:~$ docker exec -it database bash
root@sonic:/# redis-cli
127.0.0.1:6379> keys ROU*
    1) "ROUTE_TABLE:20c0:a86a:0:b0::/64"
    2) "ROUTE_TABLE:20c0:a898:0:20::/64"
    3) "ROUTE_TABLE:20c0:a819:0:c1::/64"
    4) "ROUTE_TABLE:20c0:a817:0:f0::/64"
    5) "ROUTE_TABLE:20c0:a829:0:70::/64"
    6) "ROUTE_TABLE:20c0:a851:0:71::/64"
```

Build a SONiC Image

Preparation:

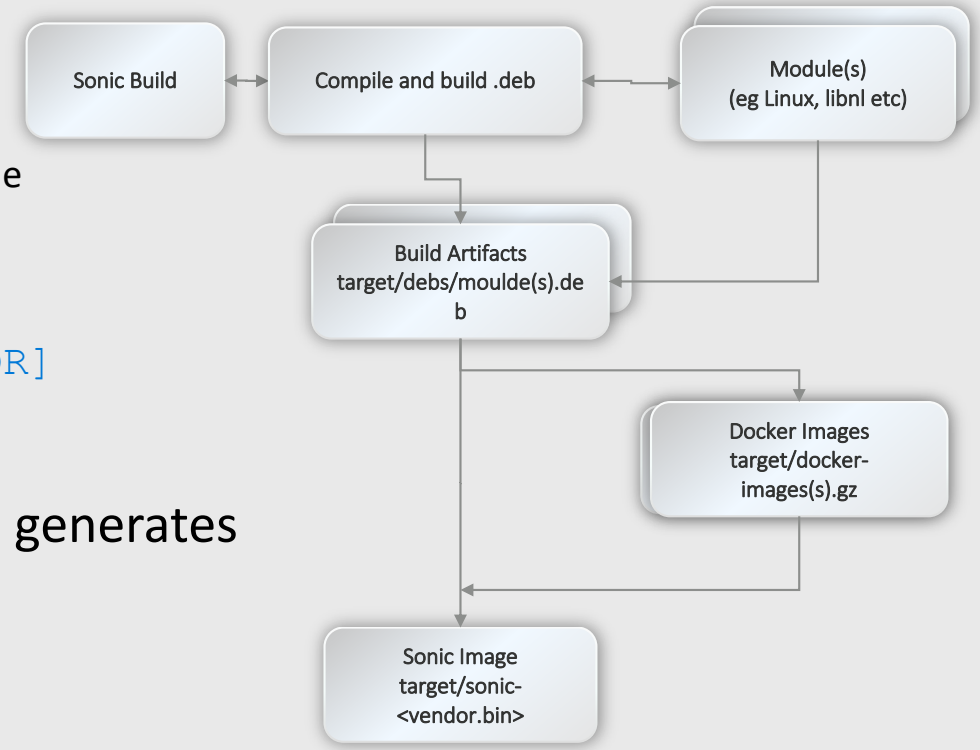
any server > 1T hard disk, ubuntu linux 16.04

Prerequisite: install PIP and JINJIA on the host build machine

Clone code repos with all Git submodules

Build image with a specific ASIC choice

```
make configure PLATFORM=[ASIC_VENDOR]
make all
```



Each module compiles the source code and generates the .deb package

Main .deb package and zero or more derived packages

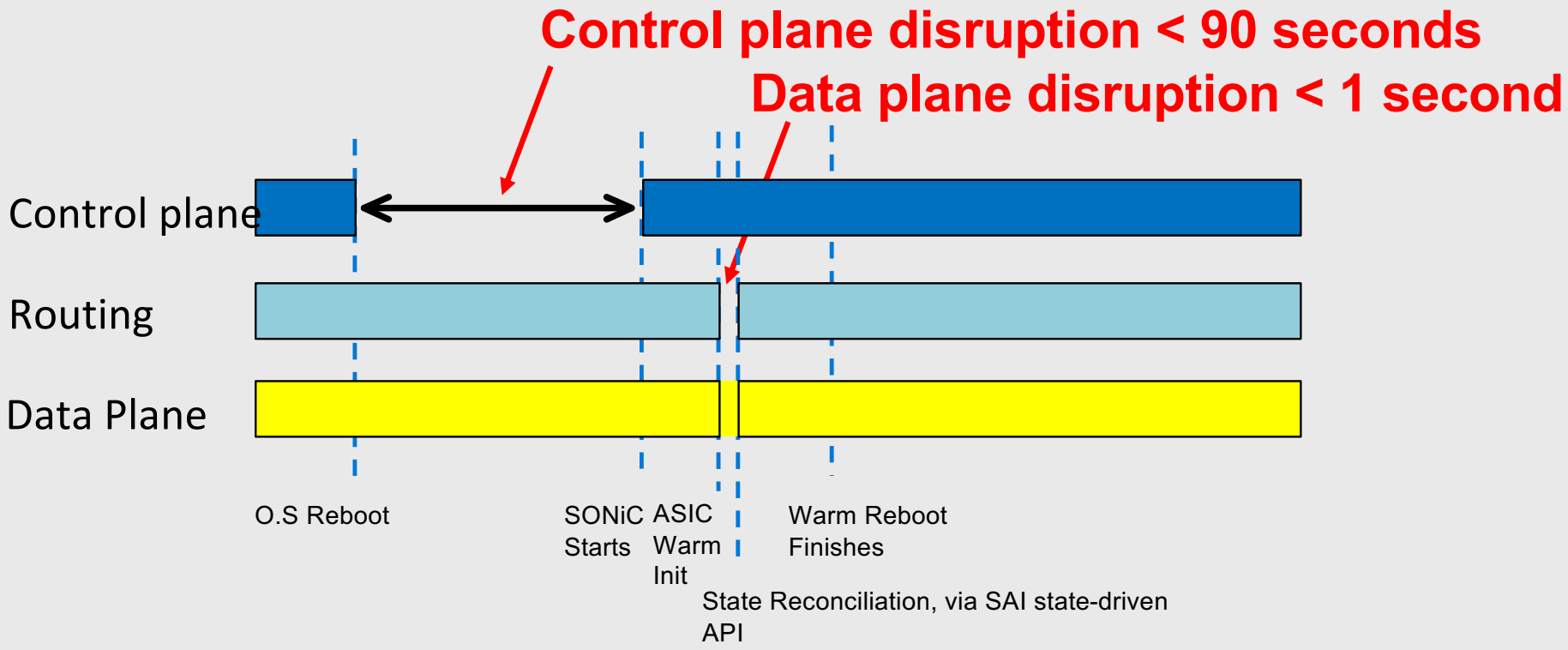
From the .deb packages

docker-<image(s)>.gz

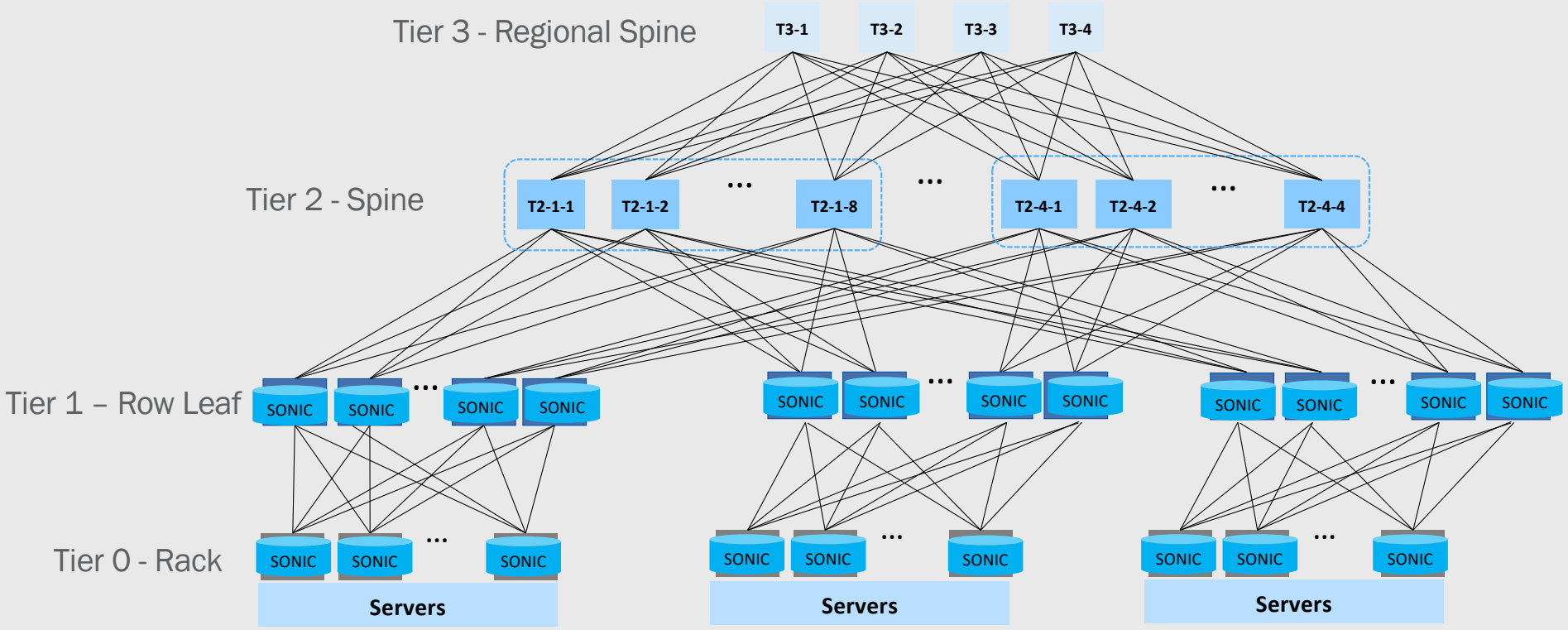
Sonic-<image>.bin

[Building guide on Wiki]

Warm Boot – Subsecond Disruption



SONiC Is Powering Microsoft Cloud At Scale



Debugging Datacenter Network is a Daunting Task

DCNs are large and complex

O(100K) low-cost devices

Complex software stack

Network faults are unavoidable and diverse

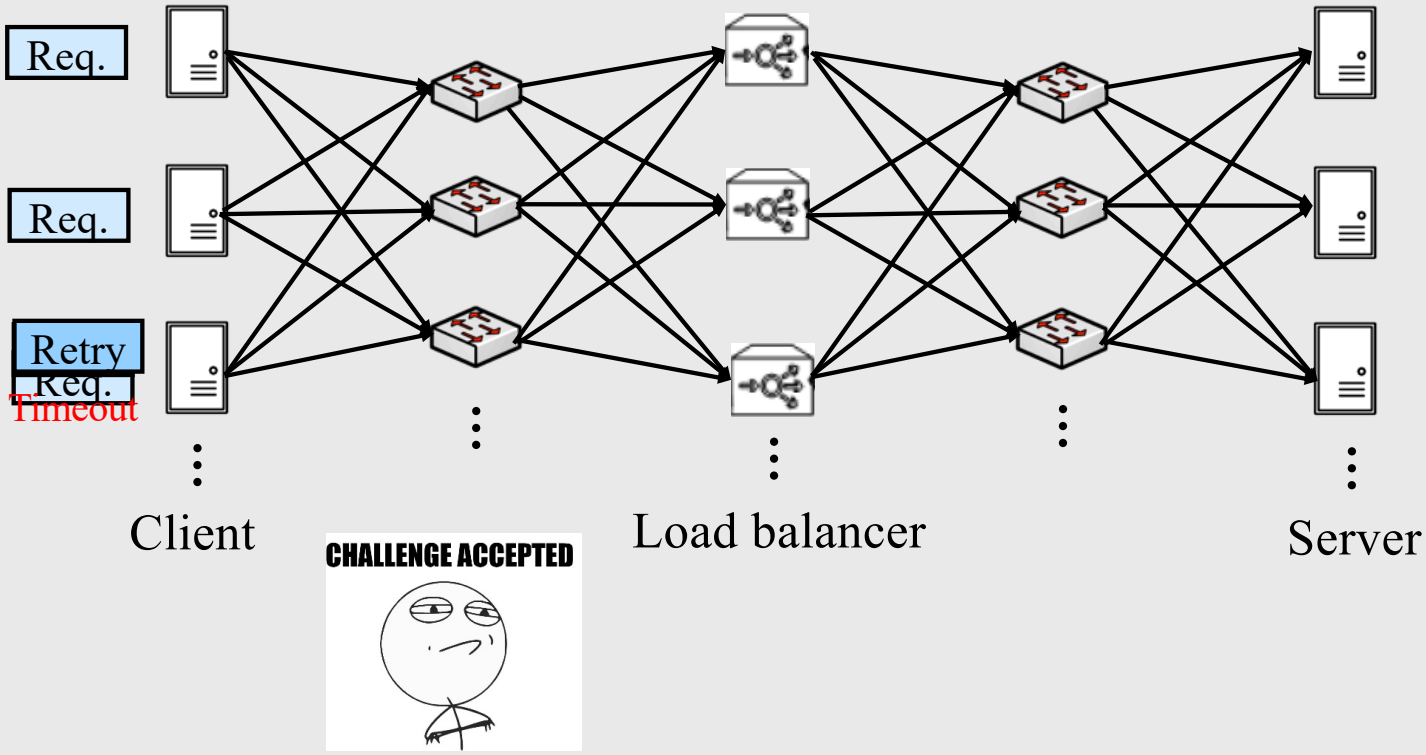
Packet drop, latency spike, low throughput, load imbalance, loop…

Existing tools insufficient

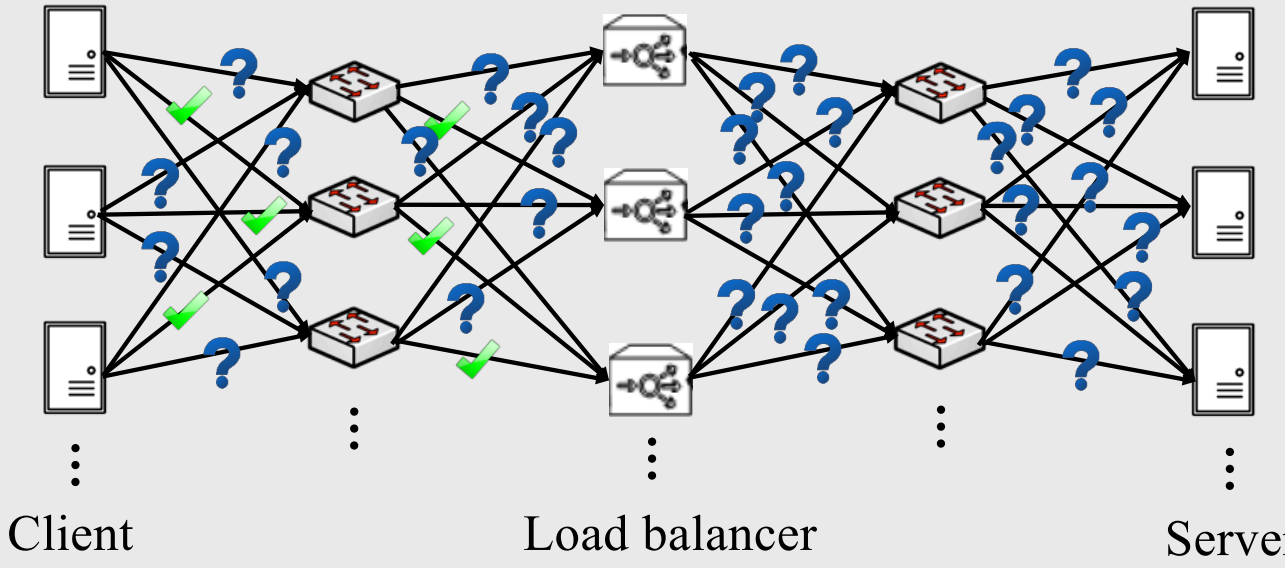
Device counters, ping, traceroute…

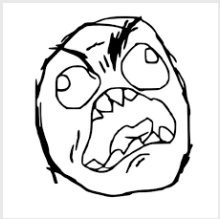
Example #1: Silent Packet Drop

SOS: application requests encounter timeouts!!

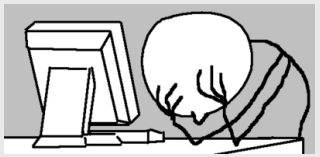


Counters and Traceroute Cannot Help

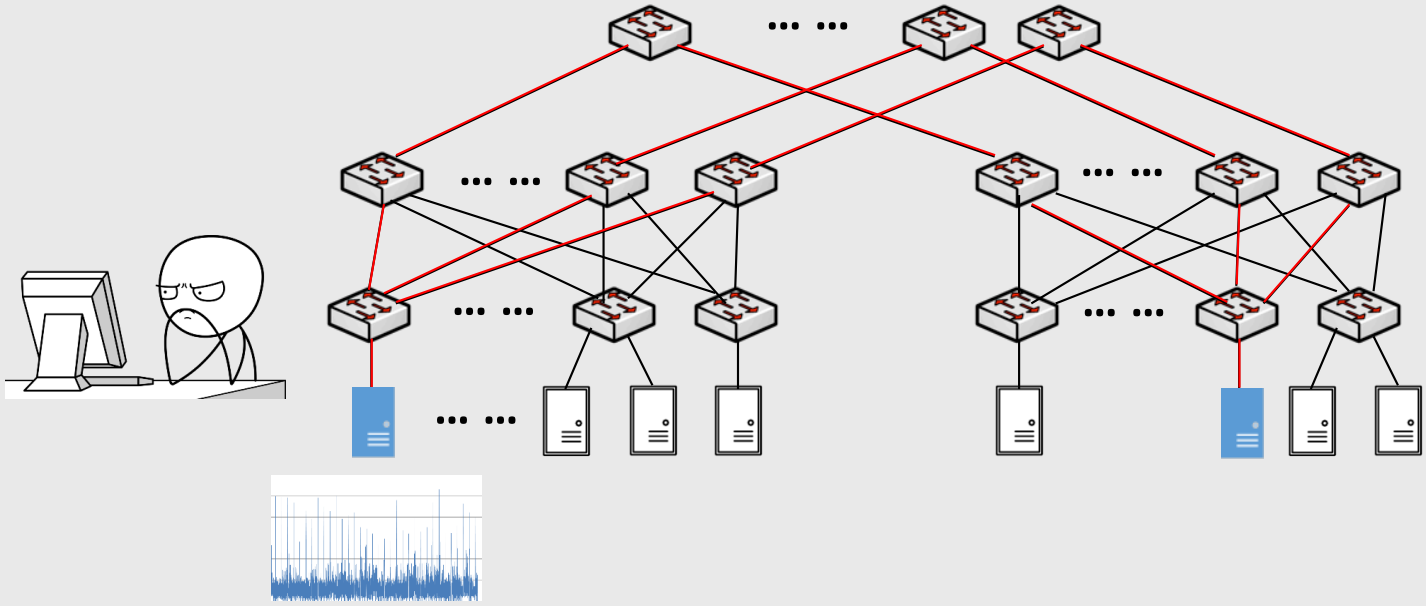




- Counters: no significant drops
- Exhaustive traceroute: prohibitively expensive or even infeasible



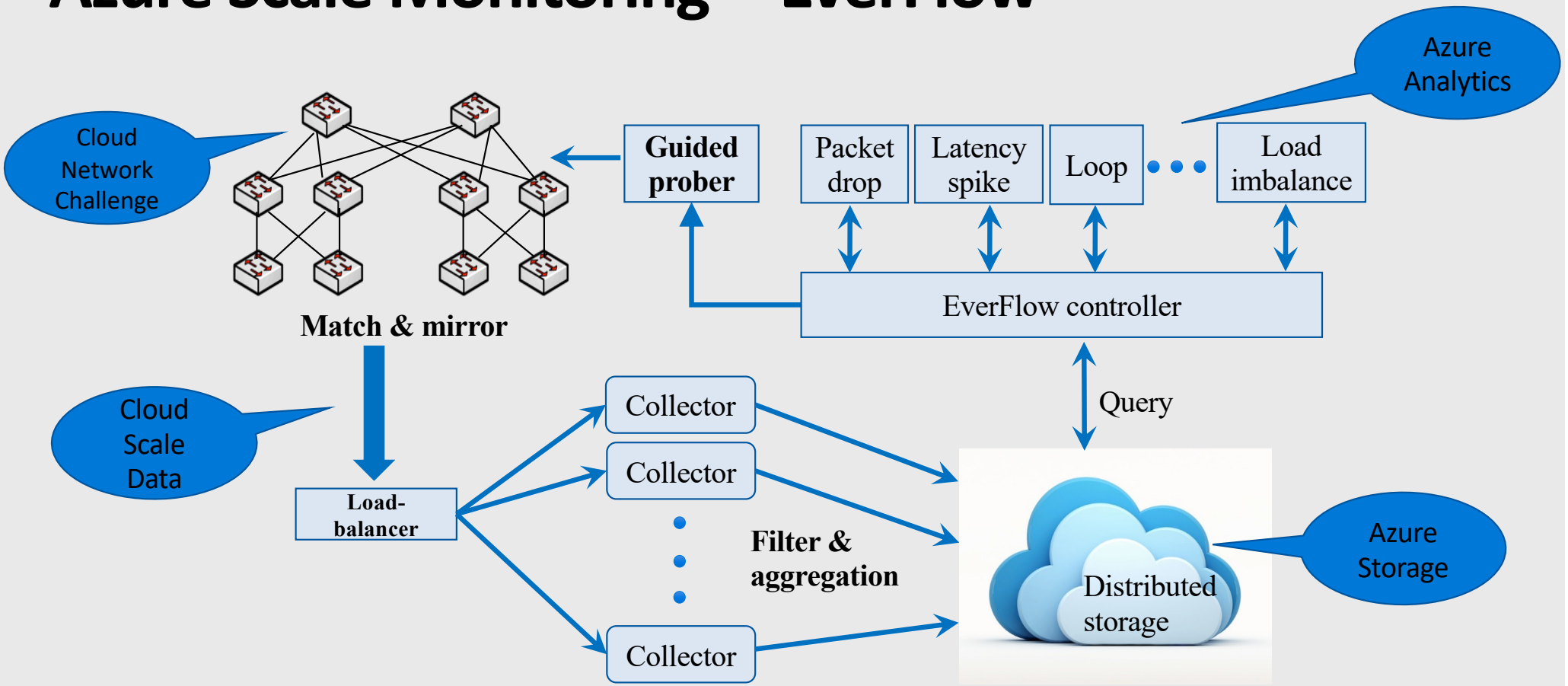
Example #2: Latency Spikes



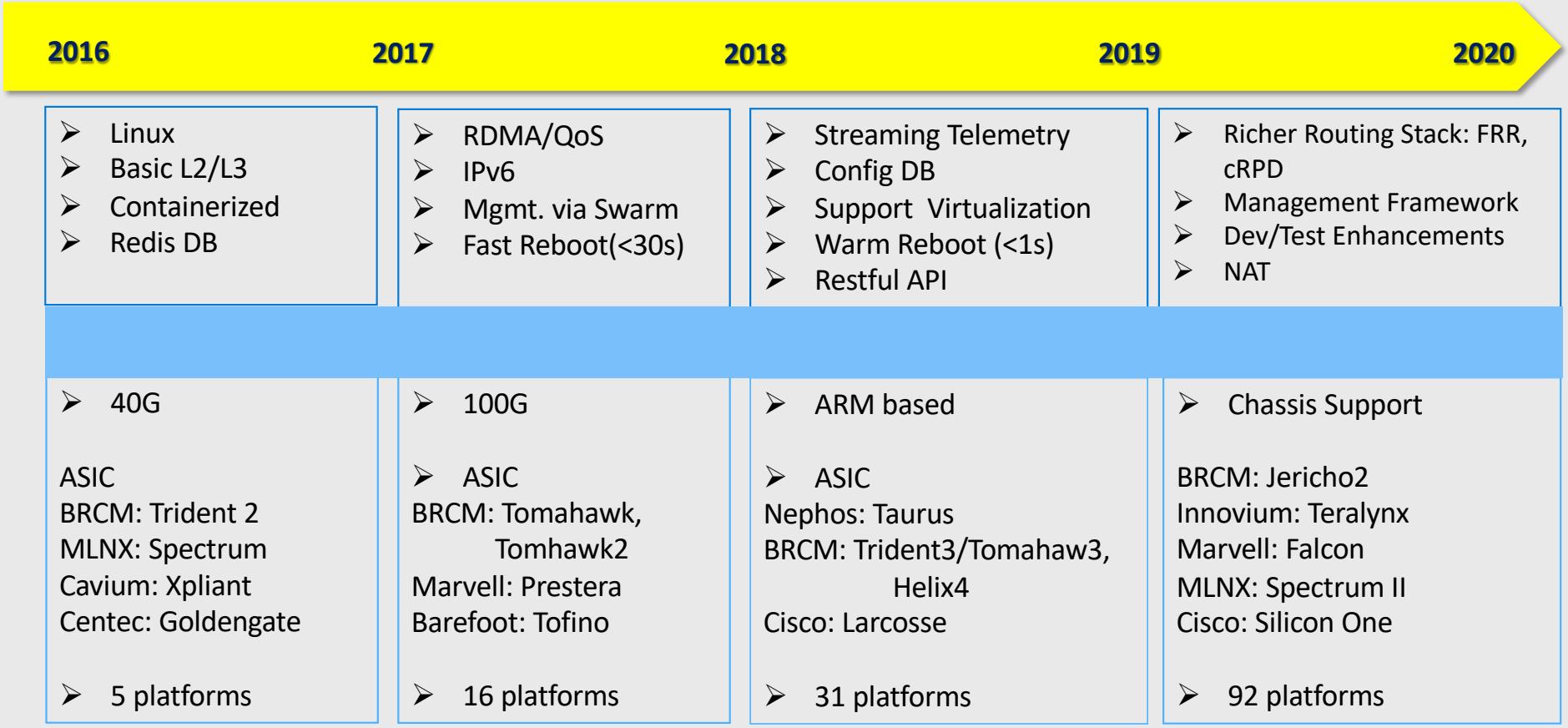
Queue size watermarks: too coarse-grained to correlate w/ affected flows

Ping & traceroute: cannot measure per hop delay

Azure Scale Monitoring -- EverFlow



SONiC Keeps Evolving




Alibaba Group
ALPHA
Alpha Networks Inc.
apstra
ARISTA
Asterfusion
AVIZ
Baidu 百度
BAREFOOT
NETWORKS
BROADCOM
CANONICAL
Celestica
centec
CISCO
DELL EMC
DELTA
DiDi
docker
Edge-corE
恒为科技
EmbedWay
香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY
Innovium
Inventec
京东云
JUNIPER
NETWORKS
LinkedIn
MARVELL
Mellanox
TECHNOLOGIES
Metaswitch
Networks
Microsoft
MiTAC
Nephos
PEGATRON
和碩聯合科技
Quanta Computer
Ruijie
NETWORKS
Tencent 腾讯
ufiSpace
wipro
WNC
Wistron NeWeb Corp.
XCLOUD
NETWORKS

Open Invitation to the SONiC Community

Inviting contributions in all areas

- New ideas on white/open network devices
- SAI proposals
- Hardware platform
- New features, applications and tools
- Download it, Test, Deploy!

Website:	https://azure.github.io/SONiC/
Mailing list:	sonicproject@googlegroups.com
GitHub:	https://github.com/Azure/SONiC
Wiki:	https://github.com/Azure/SONiC/wiki/

Thank You

© Copyright Microsoft Corporation. All rights reserved.