
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2019

Go FaaSter: Cold Start Optimization in a Serverless Platform

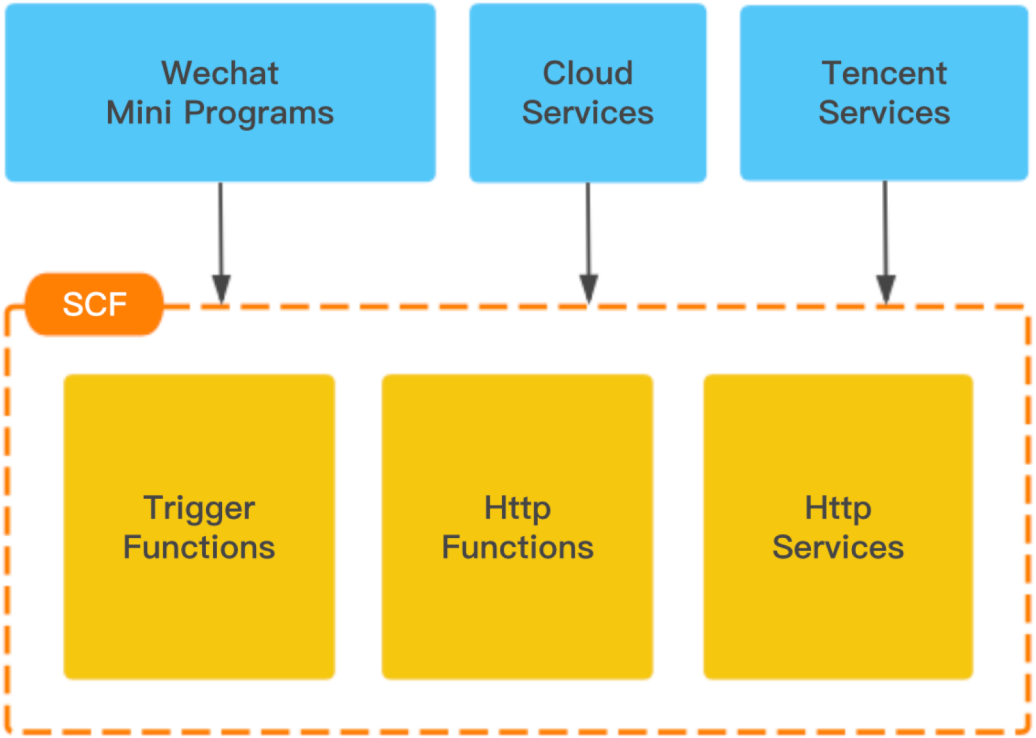
Scott Zhou & Yanbo Li

Agenda

1. Tencent Cloud serverless cloud functions
2. What is function cold start ?
3. Several ways to optimize cold start

1.1 Tencent Cloud SCF

Serverless Cloud Functions



1. Wechat mini programs

   millions of mini programs and developers

   a billion mini program end users
2. Cloud services

   grows rapidly
3. Tencent services

   lots of SNS services migrate to SCF

1.2 Challenges we meet

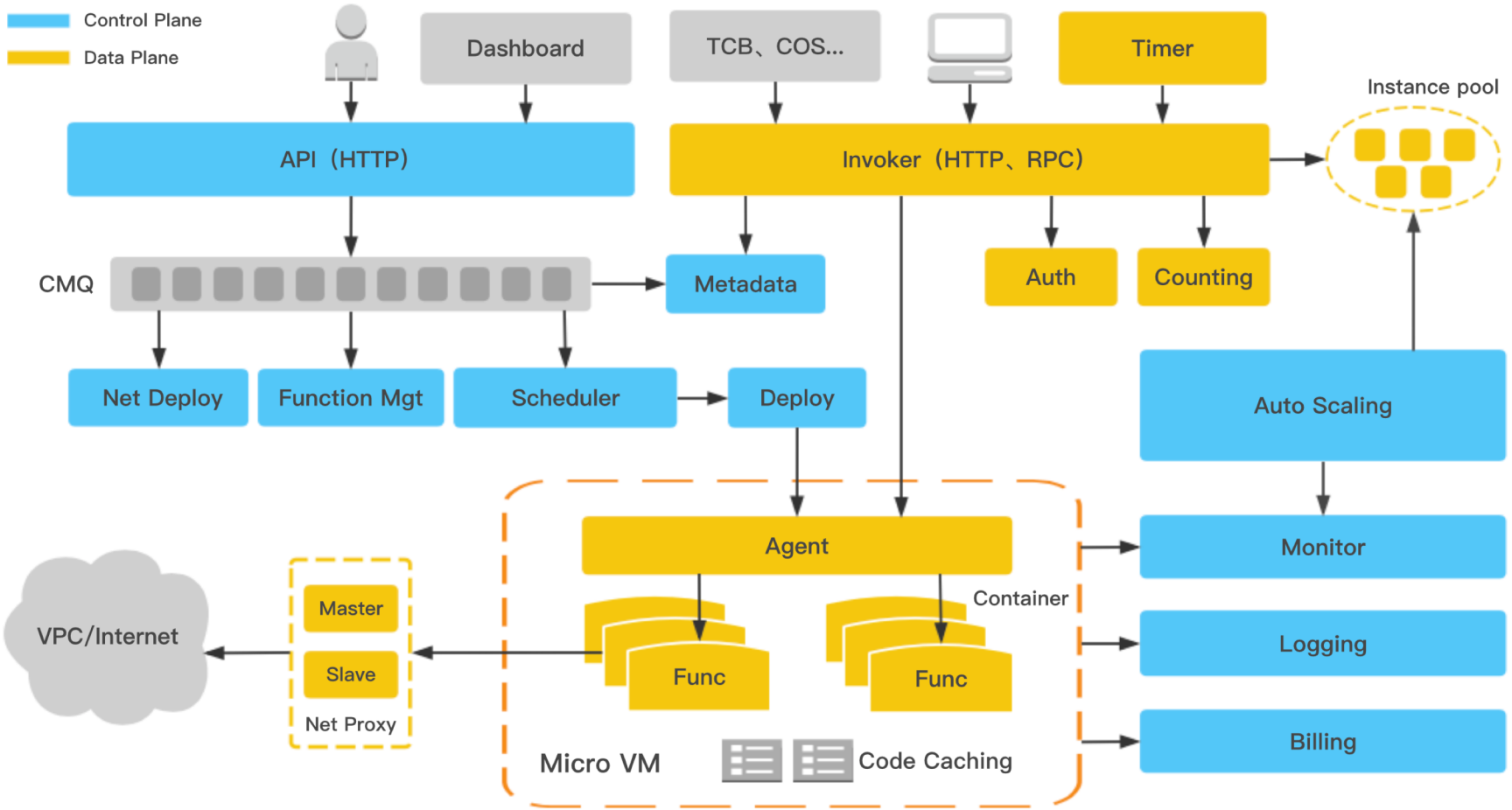
1. Security
   full isolation at VM level

2. High concurrency to deploy function instances
   10,000 instances in one second

3. Large-scale instances
   hundreds of thousands

4. Low latency of cold start
   less than 100ms
   0% cold start for active functions

1.3 SCF Architecture

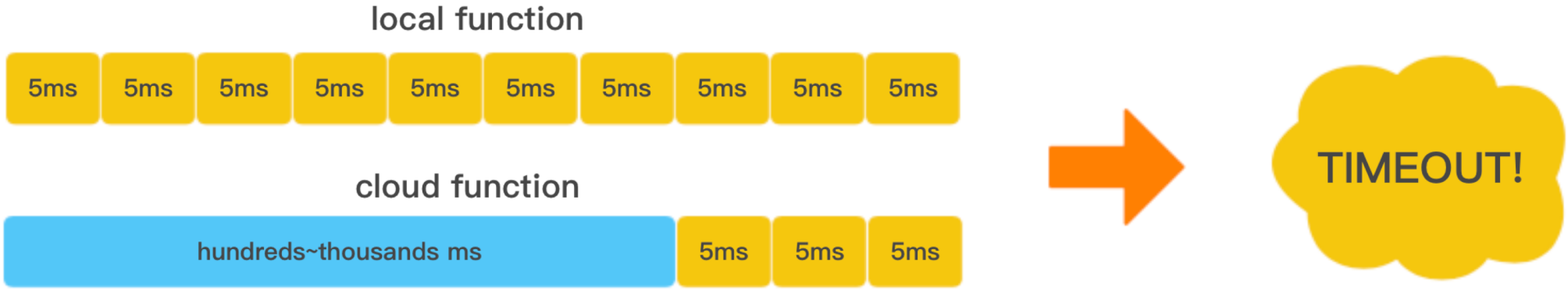


2.1 What is function cold start ?

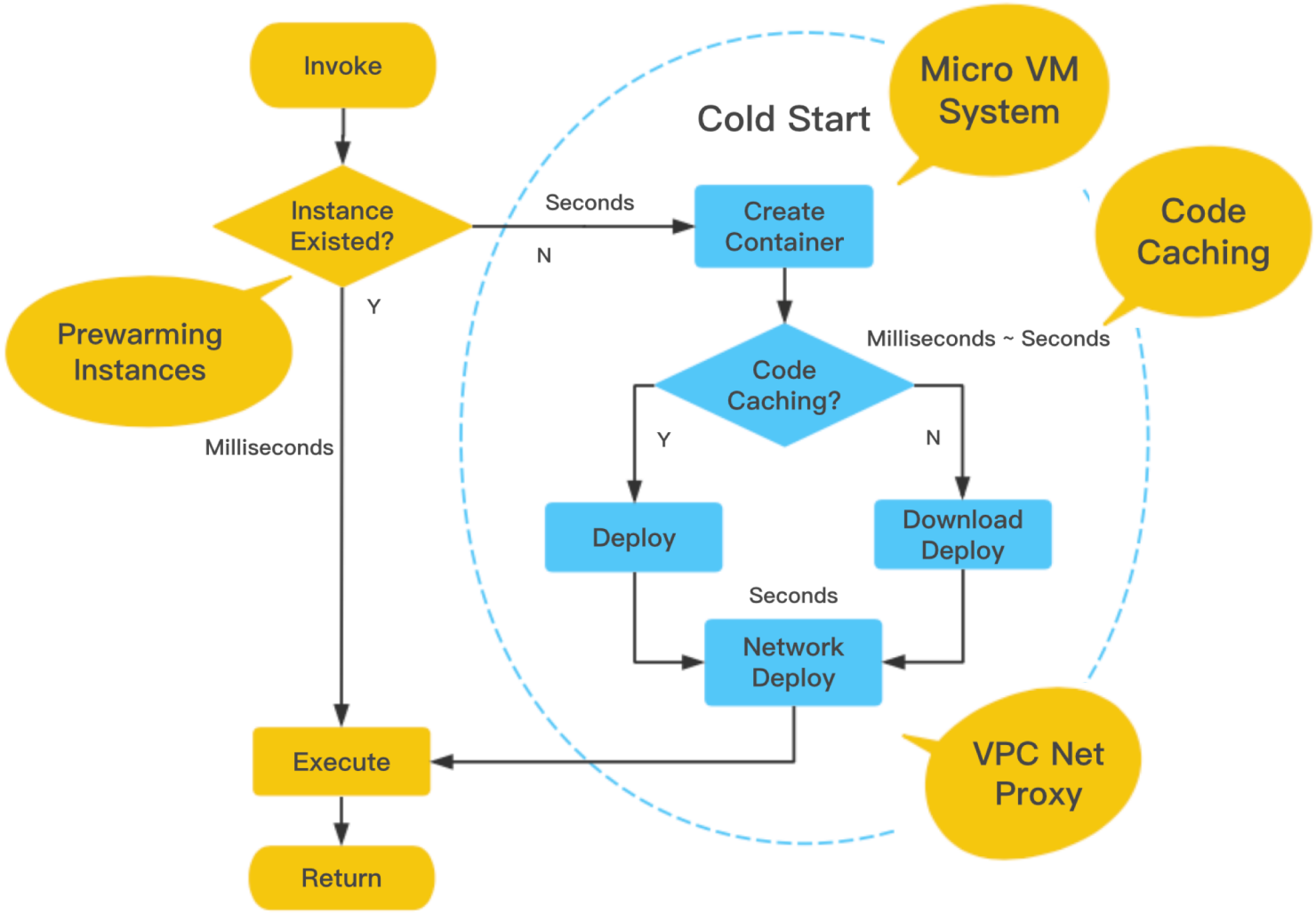
The first time when deploying the function instance !

Everything has a cold start ?

And why does is matter ?



2.2 Function cold start stages



Stage 1 : Create VM & Container

creating traditional VM takes minutes

creating container takes seconds

Stage 2 : Download Function Code

depending on package size

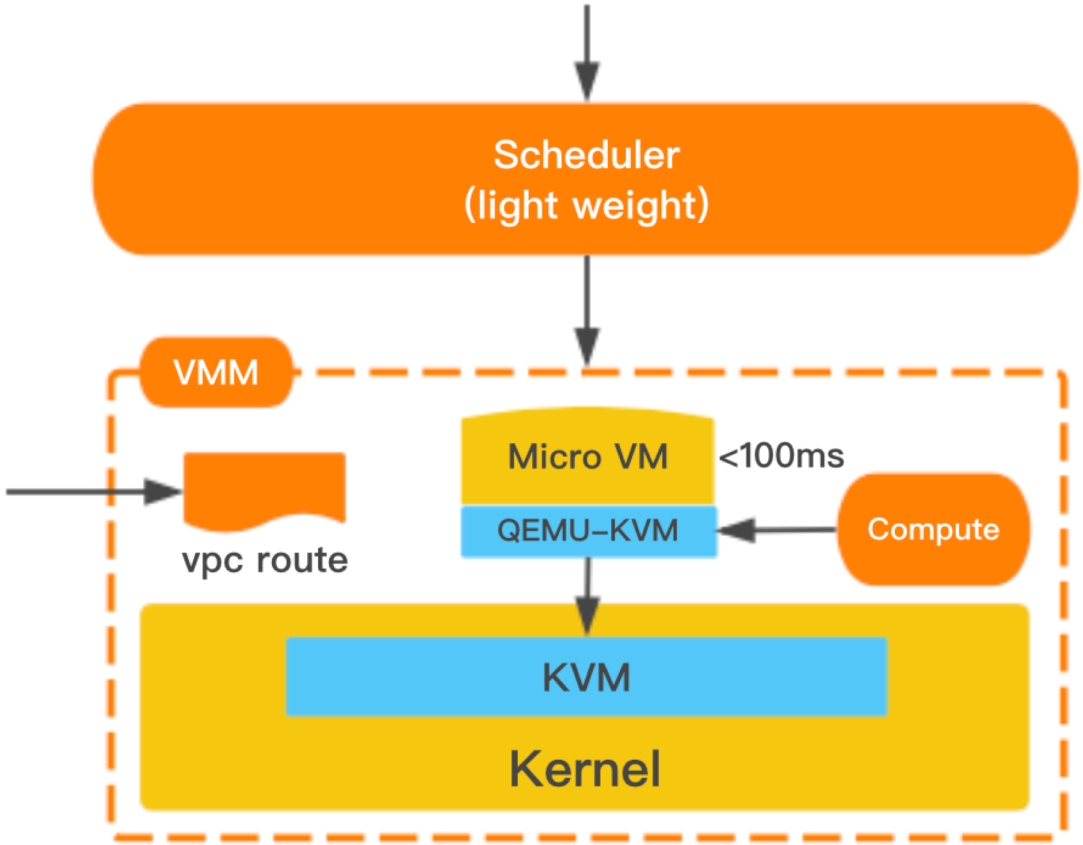
Stage 3 : Deploy VPC Network

deploying ENI takes seconds

3. Several ways to optimize cold start

1. Micro VM system
2. Code caching
3. Prewarming instances
4. VPC net proxy

3.1 Micro VM System



1. Scheduler：rebuild scheduler module

   light weight, from seconds to milliseconds

   concurrency from 1,000 to 10,000s

2. Networking：

   VPC route data prebuilding，taking 0s

3. Virtualizing：

   light weight VM, taking <100ms to boot

3.1.1 How does it work ?

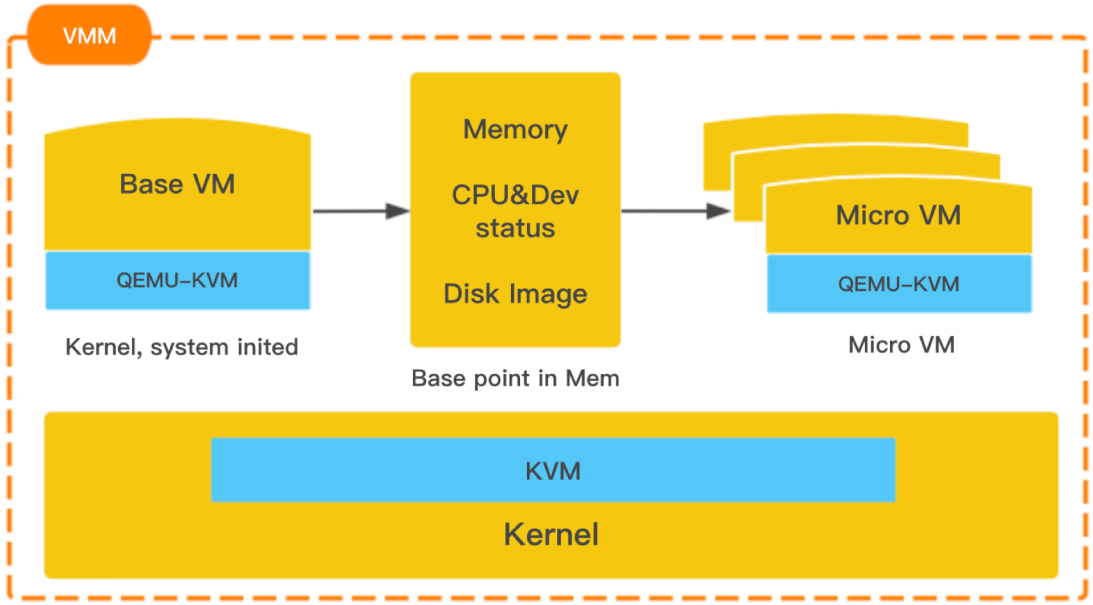
Heavy weight scheduler

- all kinds of VM flavors : CPU, mem, disk…
- affinity and anti-affinity
- deployment group
- resource utilization
- …

Light weight scheduler

- fixed standard flavors
- off-line computing

3.1.2 How does it work ?



1. Create Base point VM template

   run a base VM, qemu-agent detects status

   save VM memory, CPU & Dev status in mem

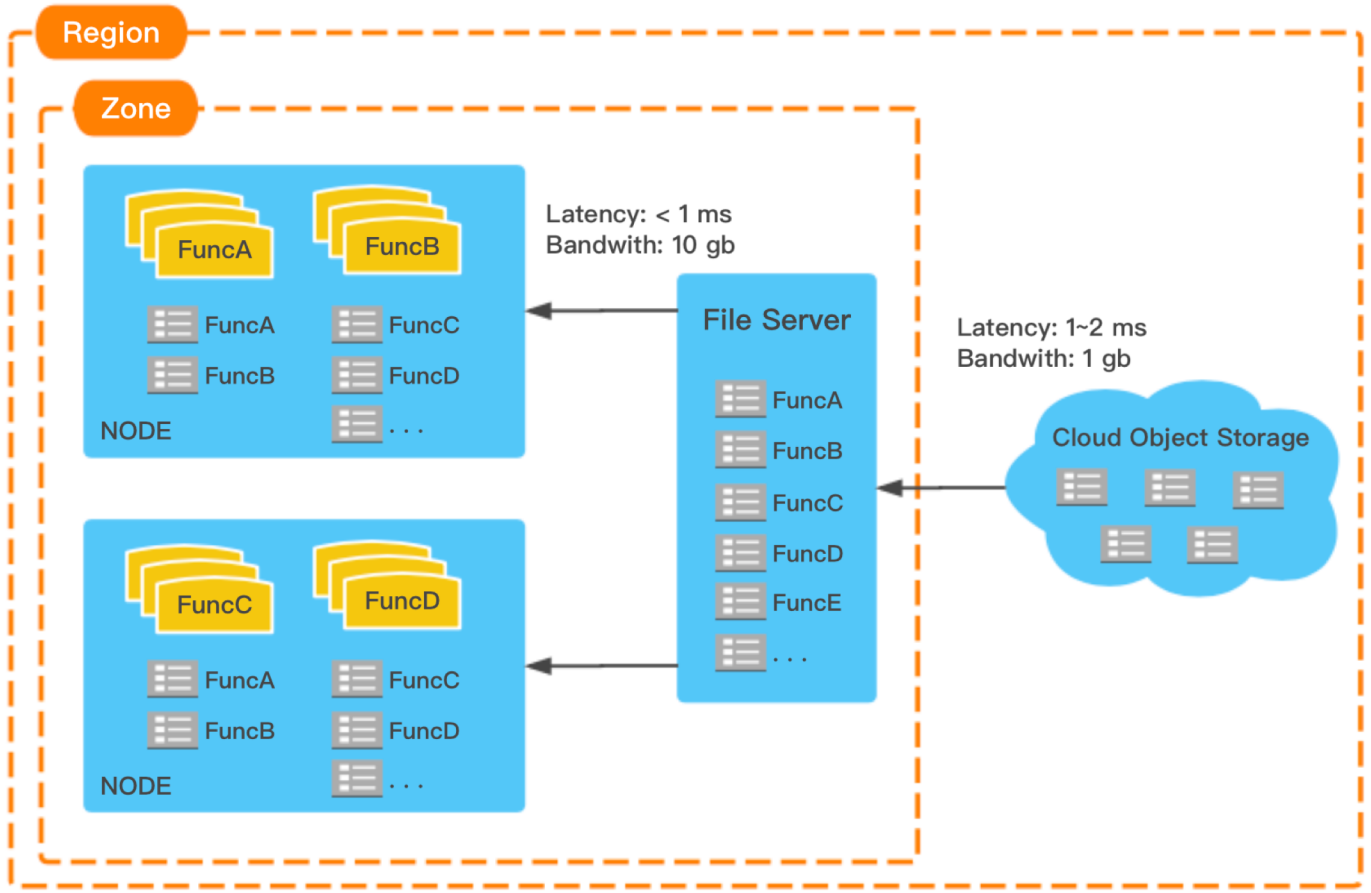
2. Run Micro VM from VM template

   -incoming exec:cat /dev/shm/vm_template

   disk: qcow2 snapshot, 3 layers

   modify Mac, IP, hostname...

3.2 Code Caching



L1 Node local cache :

cache all functions of the same developer

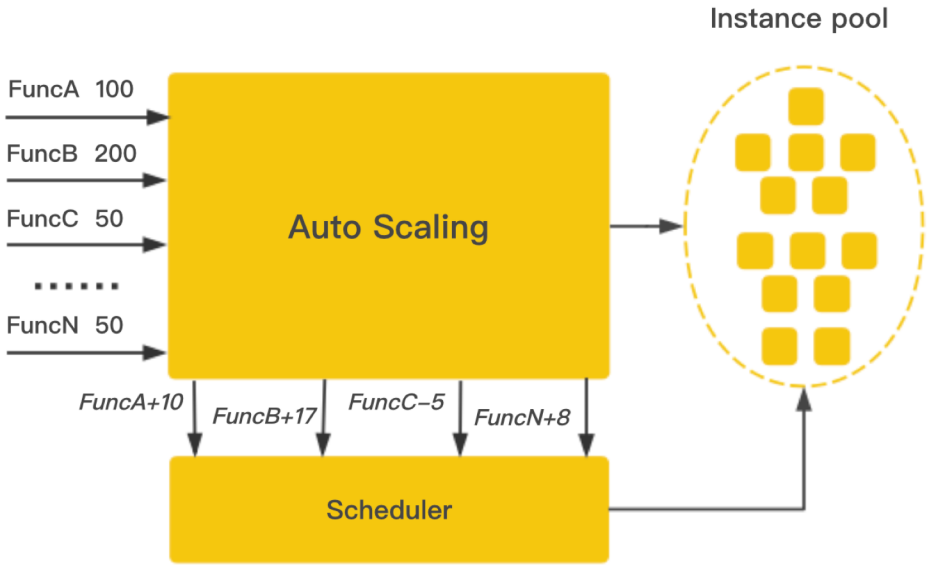
cache hit 100%

L2 Zone local cache :

cache all functions of the same zone

cache hit 100%

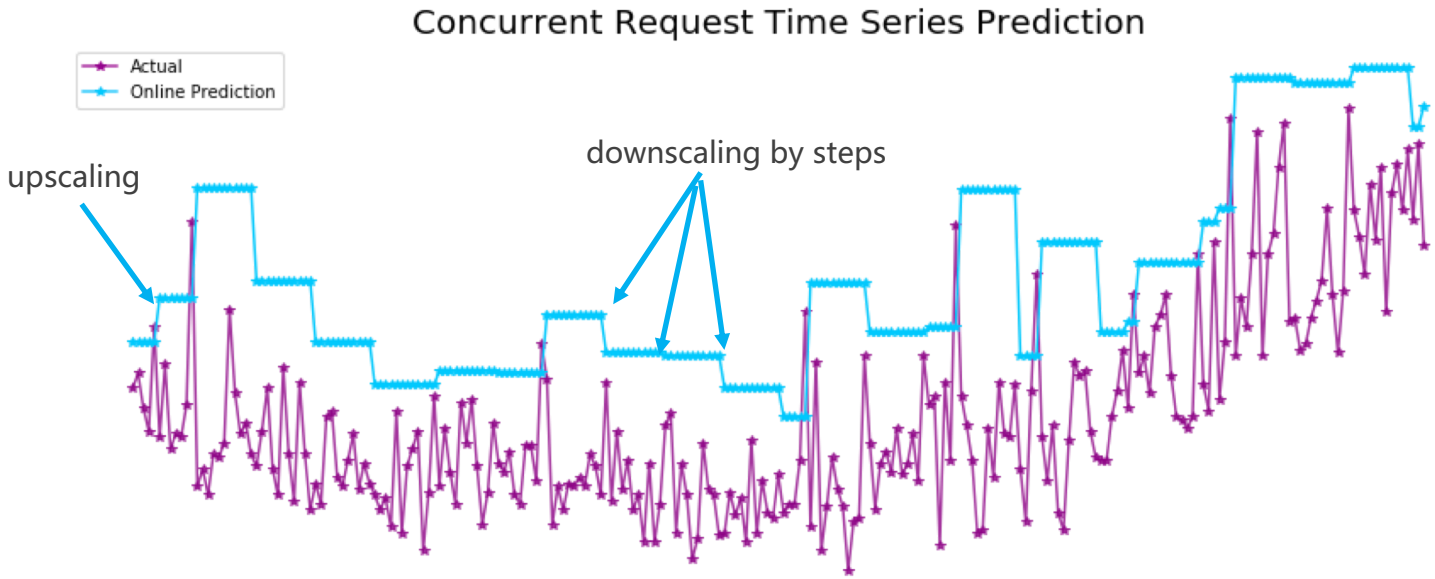
3.3.1 Auto Scaling



Real Time Feedback

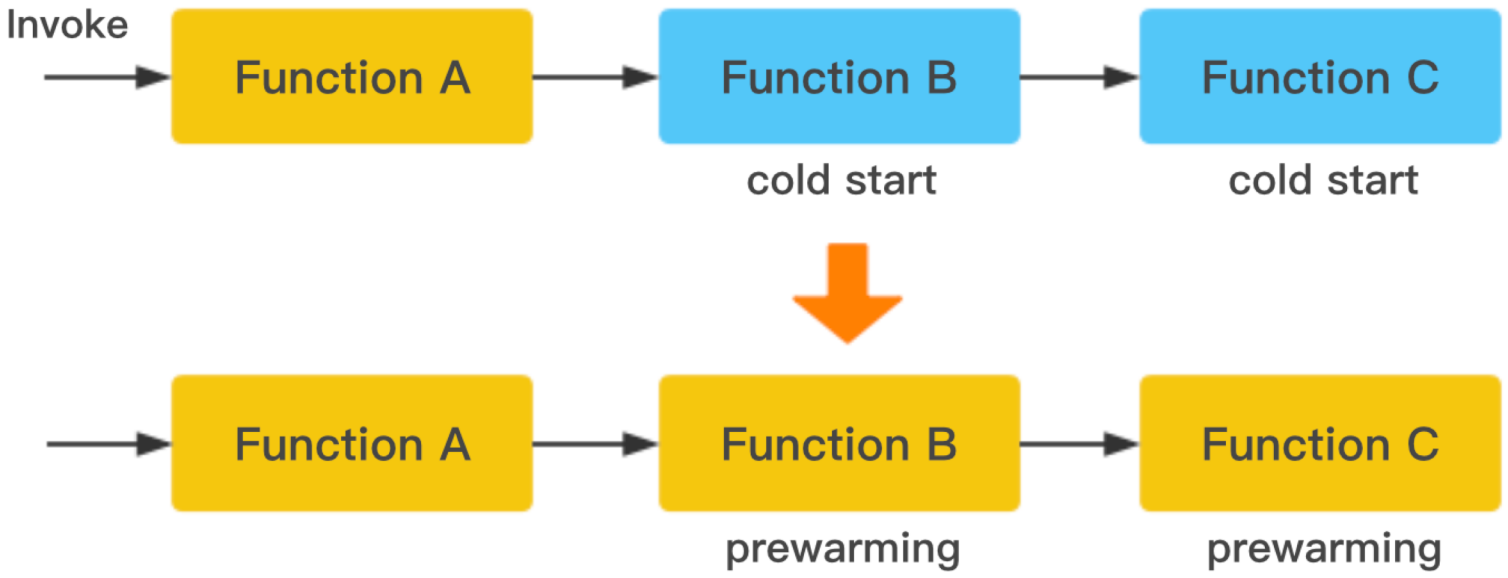
Instant Upscaling

Cooling Buffer

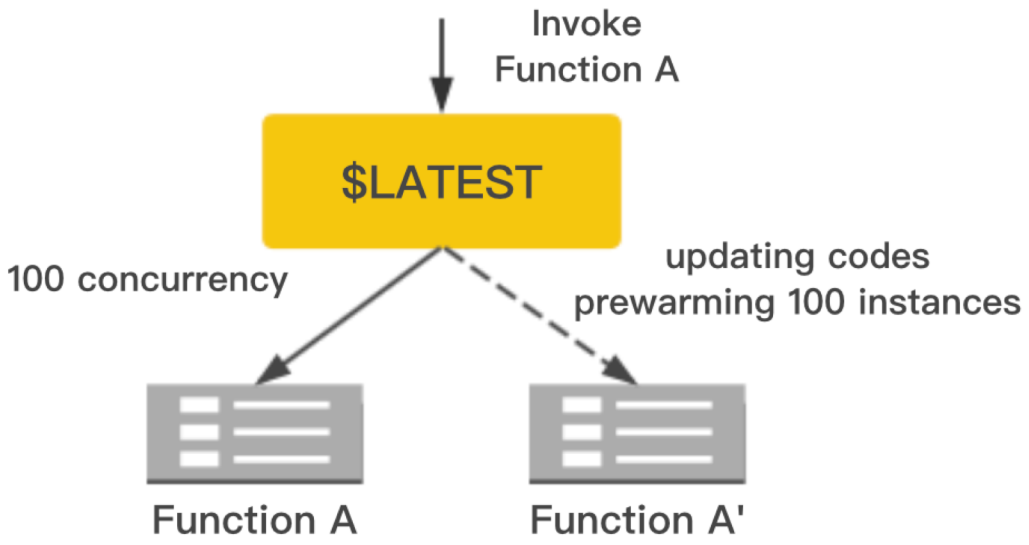


3.3.2 Predictable Prewarming

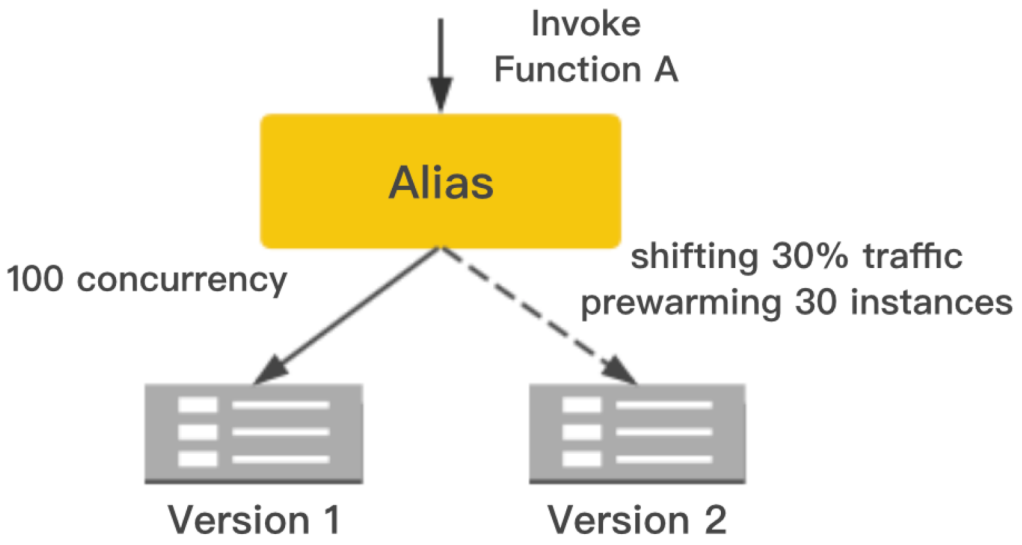
Function C&B are invoked by Function A



3.3.3 Predictable Prewarming



1. Update $LATEST version



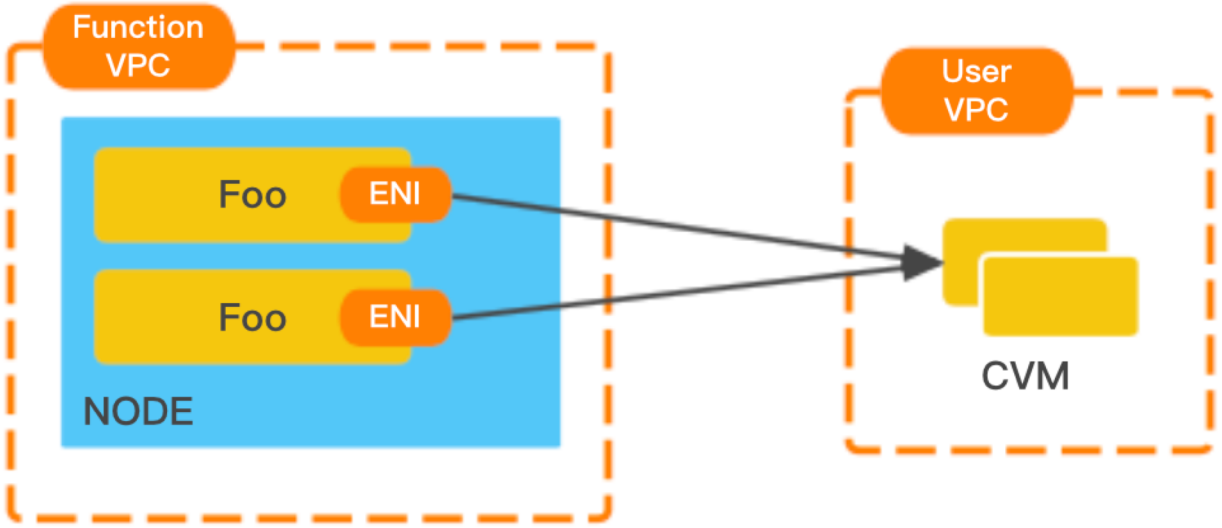
2. Shift traffic between two versions

3.4 Function Network Model

Functions access resources in VPC, e.g. CVM, CDB…

Functions access resources in the internet, with optional fixed EIP

3.4.1 Legacy Network – access VPC



Architecture (1)

Access the VPC resources through pod ENI

ENI is attached in function pod when deploying function instances.

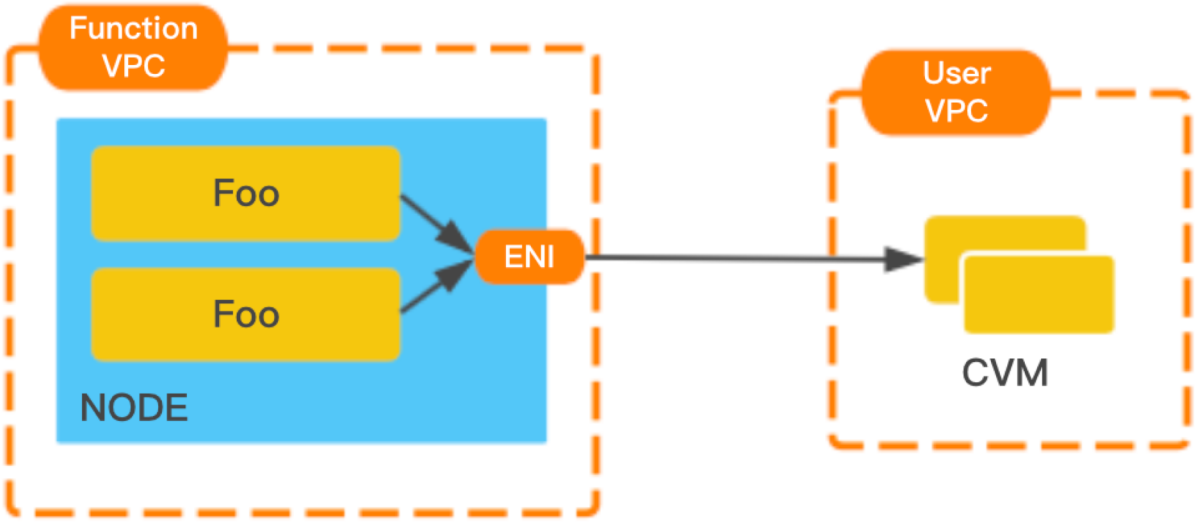
ENI is shared within the same pod.

Problems :

High latency of cold start

Huge consumption of ENI

3.4.1 Legacy Network – access VPC



Architecture (2)

Access the VPC resources through node ENI

ENI is attached in function node when deploying function instances.

ENI is shared within the same node.

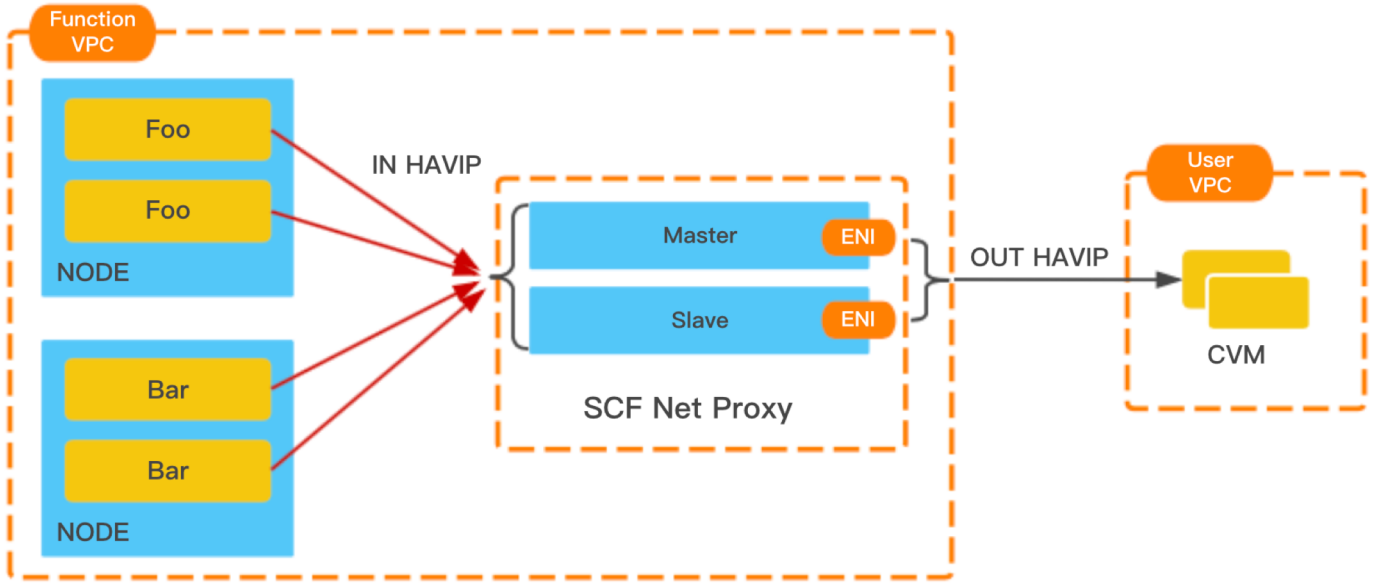
Problems :

A little bit better than architecture (1)

High latency of cold start

Huge consumption of ENI

3.4.2 SCF Solution – access VPC



Advantages:

Cold start time drops from seconds to milliseconds

Just a pair of ENIs consumption

Master-slave mode for high availability

Second level failover

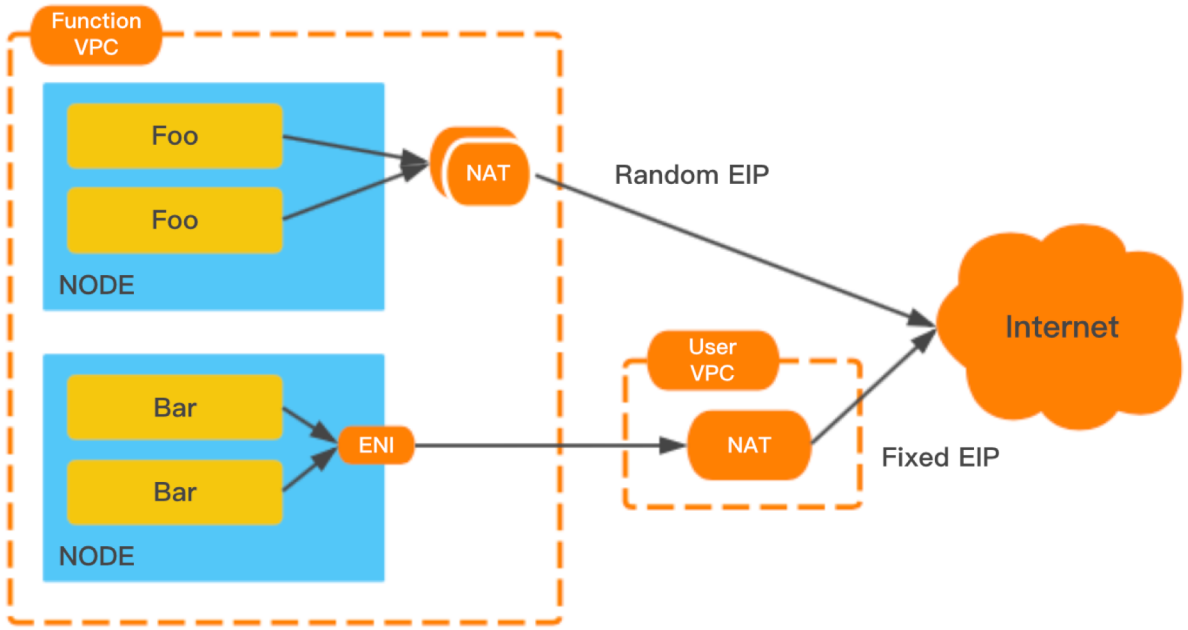
TCP connection retention

Bandwidth auto scaling

Forward traffic through SCF net proxy,

ENI is attached in forwarding node when function is being created

3.4.3 Legacy Network – access Internet



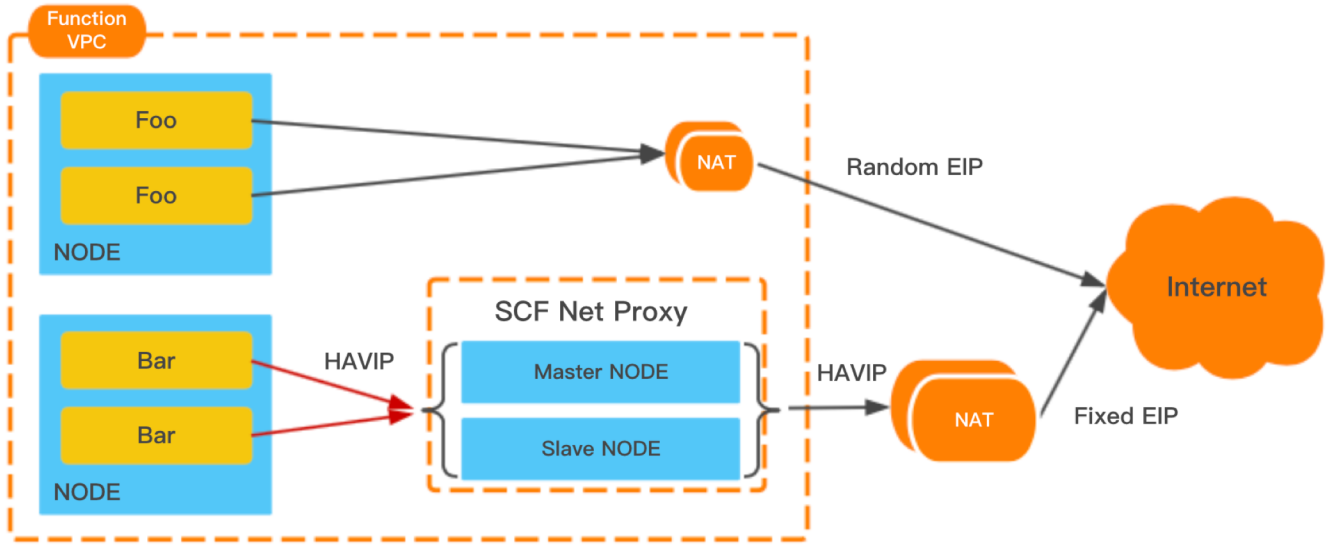
With fixed EIP:

- Deploy NAT in developer's VPC
- Access NAT through ENI

Problems :

- Hard work for long-tail developers
- Fixed expenses for NAT instances
- High latency of cold start
- Huge consumption of ENI

3.4.4 SCF Solution – access Internet



When a fixed EIP is required:

Forward traffic through SCF net proxy,

Forwarding node is deployed when function is being created

Advantages:

Cold start time drops from seconds to milliseconds

Master-slave mode for high availability

Second level failover

TCP connection retention

Bandwidth auto scaling

Conclusion

Optimizing cold start really matters!

Rebuild function architecture for much larger scales.

Develop micro VM system for rapidly deployment.

Build code cache, auto scaling and VPC net proxy for lower cold start latency.

Thanks

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2019

Q&A

云原生公众号

SCF用户交流群

About me

Scott Zhou (Wechat ID: rwxrxrx)

He is an expert engineer leading serverless computing at Tencent Cloud, and is one of the pioneers of Tencent Cloud.

In the past he has worked on DnsPod, OpenAPI platform, VM scheduling and migration for resource utilization, which improved the rate of margin by 50%.

In the spare time, he likes self-driving tour and photography.

About me

Yanbo Li (Wechat ID: HwLshs)

He is a senior engineer at Tencent Cloud Middleware team. He has great interest in containers and networks.

Previously, he worked at Huawei on LTE network protocol stack development.

In the distant past, he built the VPC net proxy to lower the latency of function cold start.