Granular Computing and Network Intensive Applications: Friends or Foes?

Arjun Singhvi, Sujata Banerjee, Yotam Harchol, Aditya Akella, Mark Peek, Pontus Rydin
Evolution of Computing Services

Decrease user involvement in management tasks

Bare Metal

Virtual Machines

Containers

Granular Serverless Computing

Allow more user focus on application logic
Programming Paradigm Shift

Traditional

Monolithic Application

→

Serverless

Collection of Functions
Serverless Computing Control Flow

Upload your function to the serverless computing platform and register for event-based triggers.

Execution is triggered when an event occurs.

Platform runs your function ONLY when triggered.

Pay just for the computation time.
Serverless Computing Building Blocks

- Function Scheduler
- Key Value Store
- Blob Store

Serverless Functions are Stateless!
Serverless Computing Platforms

- Amazon
- Google
- Microsoft
- IBM
- Iron.io
- nuclio
- OPENFAAS
- BINARIS
- UW-Madison
- VMware
- CLOUDFLARE
Research on Serverless Computing

• Video Processing Applications
  – ExCamera (NSDI ’17)

• Distributed Computing Applications
  – PyWren (SoCC ’17)

• Network Intensive Applications -> Network Functions
  – Our Work (HotNets ‘17)
Network Functions

• Examine and perform *stateful* actions on packets/flows

• Ensure security, improve performance and provide other network-related functionality

• Lie in the critical path between source and destination

• Should be capable of handling packet bursts and failures
Network Functions State Taxonomy

State created or updated by an NF applies to either a **single flow** or a **collection of flows**

Per-flow state
- Connection
  - TcpAnalyzer
  - HttpAnalyzer

Cross-flow state
- Connection
  - TcpAnalyzer
  - HttpAnalyzer
  - ConnCount
Network Functions Virtualization

• Leverage cloud computing and SDN in Telco infrastructures
  – Utilize commodity hardware
  – Reduce costs
  – Dynamically allocate NF instances
  – Introduce new services fast
Why Network Functions?

• Network Functions are a demanding class of applications
• Trend of pushing Network Functions to the cloud
  – APLOMB (SIGCOMM ‘12), EMBARK (NSDI ‘16), AT&T Domain 2.0
• In the future, if serverless computing becomes the de-facto cloud standard
Why is it challenging to run network functions atop serverless computing platforms?
Foes?

**Serverless Computing**
- Short-lived
- Stateless
- No support for efficient chaining
- No QoS guarantees

**Network Functions**
- Long-lived
- Stateful
- Needs efficient chaining
- Need QoS guarantees
What opportunities do serverless computing platforms provide to run network functions?
Friends?

- Network Functions Vision – Micro-service based architecture that scales dynamically
  - Better resource utilization
  - Scale components independently
  - Reuse across chains
  - Reduce the costs

- Serverless Computing provides the building blocks
  - Has the capability to scale based on load
  - Encourages developers to design their applications in terms of micro-services
  - Reduces the costs
Hourly Cost

Firewall

40 Gbps

EC2 VMS

AWS LAMBDA

$8

60¢
Serverless Packet Processing Design

• What is the granularity at which we want to launch lambdas? Per-packet

• Not viable
  - Cost
  - Performance
Serverless Packet Processing Design

- What is the granularity at which we want to launch lambdas? Per-flow
Per-flow Vs. Per-packet

• Pros –
  - Cost effective
  - State management easier

• Cons –
  - Needs additional infrastructure to coordinate launching of new lambdas
  - Cannot natively leverage what serverless platforms provides
AWS Lambda Benchmarking Study

• Within an AWS region
• EC2 VMs act as TCP traffic sources and sinks
• Three Click-based NFs –
  – Packet Counter
  – Firewall
  – Intrusion Detection System
AWS Lambda – Network Provisioning

Bidirectional Bandwidth of around 500 Mbps
• Bandwidth does not scale linearly
• No per lambda request network guarantees
AWS Lambda – Chaining Overhead

High chaining overhead observed
Modest results; but give us a hope as to what future serverless platforms may enable
Conclusion

• Serverless Computing Platforms are not perfect yet – are continuously evolving!
• Packet processing applications are not a natural fit; need –
  – Efficient in-network triggers
  – Efficient chaining support
  – Bandwidth guarantees
Open Questions

• Scalable remote storage service that meets our requirements?
• Better fault tolerance support?
• Faster control plane decisions?
• And many more ...

How to effectively support demanding stateful applications atop serverless platforms?