Accelerate and Secure Serverless Networks with QUIC

Kaiyu Hou†, Sen Lin†, Yan Chen†, Vinod Yegneswaran‡
†Northwestern University, ‡SRI International

Serverless Computing

Serverless computing has greatly simplified cloud programming. In serverless computing, cloud providers manage responsibility for all server-related tasks, including both hardware resource allocation and software runtime preparation. Cloud tenants are thus free to simply focus on designing discrete stateless functions and orchestrate them together for their high-level business logic.

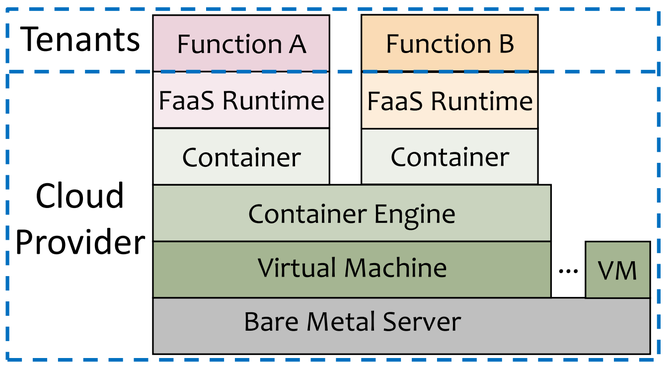


Figure 1. Serverless computing.

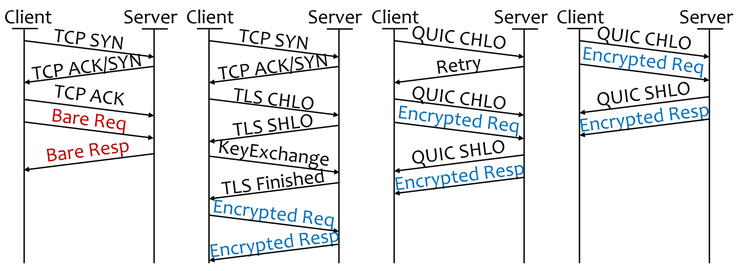
Advantages of Serverless

Agile Auto-scaling: Cloud providers can quickly launch new function instances in response to end-users' burst requests.

Usage-based Billing: Auto-scaled instances can be quickly destroyed by cloud providers. Tenants only pay for the actual function execution time, do not need to reserve resources for burst requests.

Because of both *efficiency* and *economic* advantages, serverless computing garners extensive attention from industry and is expected to become the dominant cloud computing paradigm.

QUIC Protocol



(a) TCP Handshake | (b) TCP+TLS 1.2 Handshake | (C) QUIC 1-RTT Handshake | (d) QUIC 0-RTT Handshake

Scheme	TCP	TCP + TLS 1.2	QUIC 1-RTT	QUIC 0-RTT
New Session	1	3	1	-
Recover Session	1	2	1	0

Figure 2. Extra round-trips in different protocols.

QUIC combines the advantages of TLS 1.3 and UDP to provide a *secure* and *reliable* transport layer under 0-RTT cost: the first encrypted packet can be sent before any handshake round-trips occur.

Function Chain Library

Serverless applications can directly benefit from QFaaS without any code modification. We further provide a QFaaS function chain library and integrate the QUIC server into Gateway to enable QUIC at ❸ | ❻ with slight tenant code modification (2 lines). Thus, all function chains invoked by the library will benefit from QUIC. This design has a side benefit. End-users now can also initiate requests by QUIC and further accelerate ❶ | ❽.

Network Challenges in Serverless Computing

Traditional Cloud Computing. Encrypting all internal connections is now the best practice. Though initiating reliable transportation and encryption introduces extra delays, it is not the dominant bottleneck:

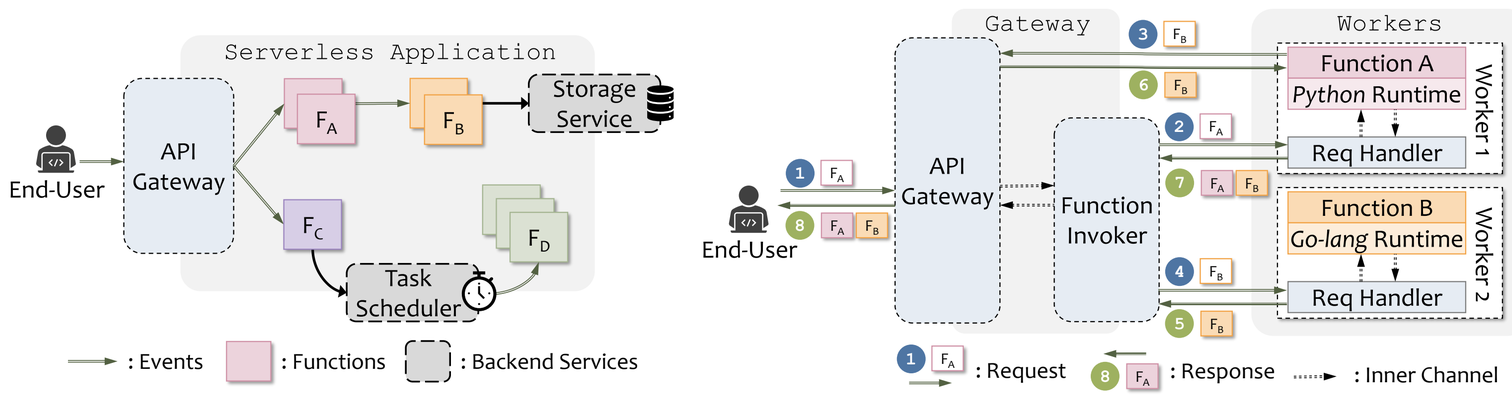
1. Transmission delay within the data center is negligible compared to execution times.
2. Connection setup latency of TCP and TLS can be simply mitigated by using persistent connections.

Network Paradigm Shift in Serverless Clouds. Nevertheless, leading commercial serverless providers still use unencrypted TCP connections between internal serverless functions, sacrificing security for performance. This is due to new constraints and demands imposed by serverless networking:

1. A function instance can be initialized in milliseconds and only processes a small sliver of the computational task. The latency introduced by TCP and TLS handshakes can no longer be ignored.
2. With the scale-zero-to-infinity feature, function instances are quickly scaled up and down by cloud providers. It is thus tough to maintain persistent connections between ephemeral functions.
3. As serverless functions are commonly chained together to form task-specific workflows, cumulative handshakes aggravate the end-to-end latency.

Our Solution–QFaaS: Simultaneously improves performance and retrofits security of existing serverless platforms without requiring any tenant code modification and by leveraging the QUIC protocol.

Network Models for Serverless Architecture



(a) Serverless Architecture in Logic View | (b) Serverless Architecture in Network-centric View

Figure 3. Serverless architecture: (a) widely used Logic Model vs. (b) Network-centric Model.

Connections from Gateway to workers (❷ | ❼, ❹ | ❺), which are fully controlled by providers, expose opportunities to optimize serverless networks without tenant code modification.

QFaaS System Design

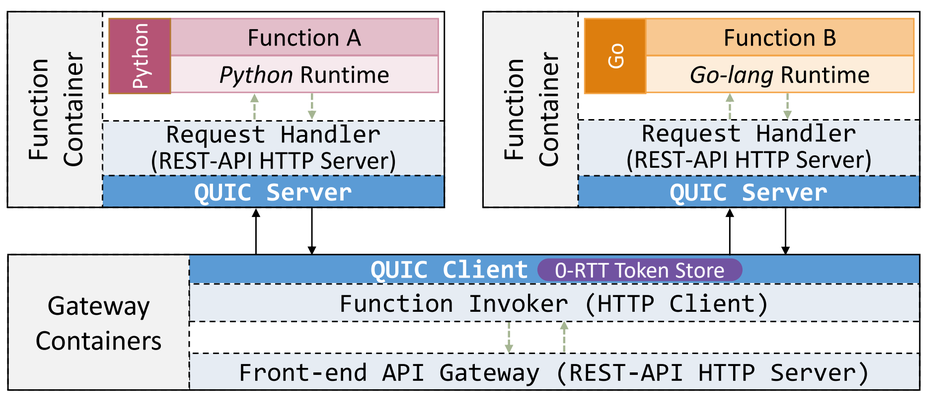


Figure 4. QFaaS system design.

We integrate the QUIC client in Gateway and QUIC servers in Request Handlers (to replace the TCP and TLS client and servers, respectively). All function requests that go through Gateway to Workers would now benefit from the efficiency and security of QUIC. Modifications on the serverless platform are totally transparent to tenant applications. We implement the QFaaS prototype into OpenFaaS. The entire system code will be made publicly available.

Experiment: Single Function

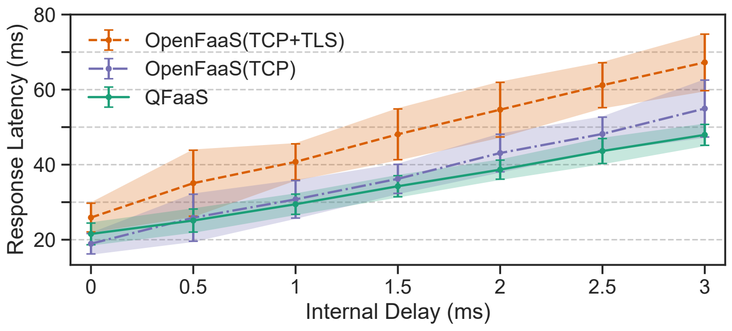


Figure 5. Latency under variant internal delays.

QFaaS can reduce the latency by up to 28% compared with OpenFaaS (TLS+TCP). It even outperforms insecure OpenFaaS (TCP).

Experiment: Function Chains

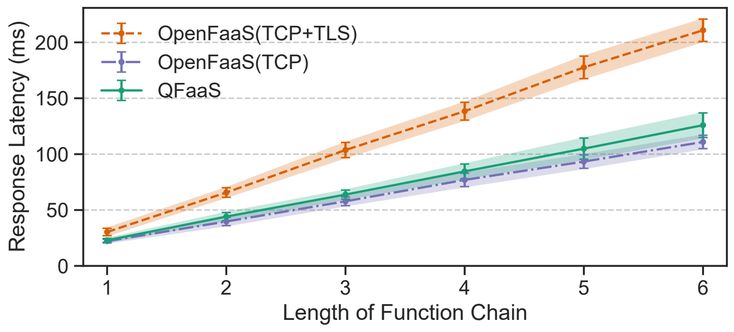


Figure 6. QFaaS with function chain library.

The latency difference between QFaaS and OpenFaaS (TCP+TLS) increases as the chain length increases and reaches 40% when the length is 6.