

DirectFaaS: A Clean-Slate Network Architecture for Efficient Serverless Chain Communications

Qingyang Zeng¹, Kaiyu Hou², Xue Leng³, Yan Chen⁴

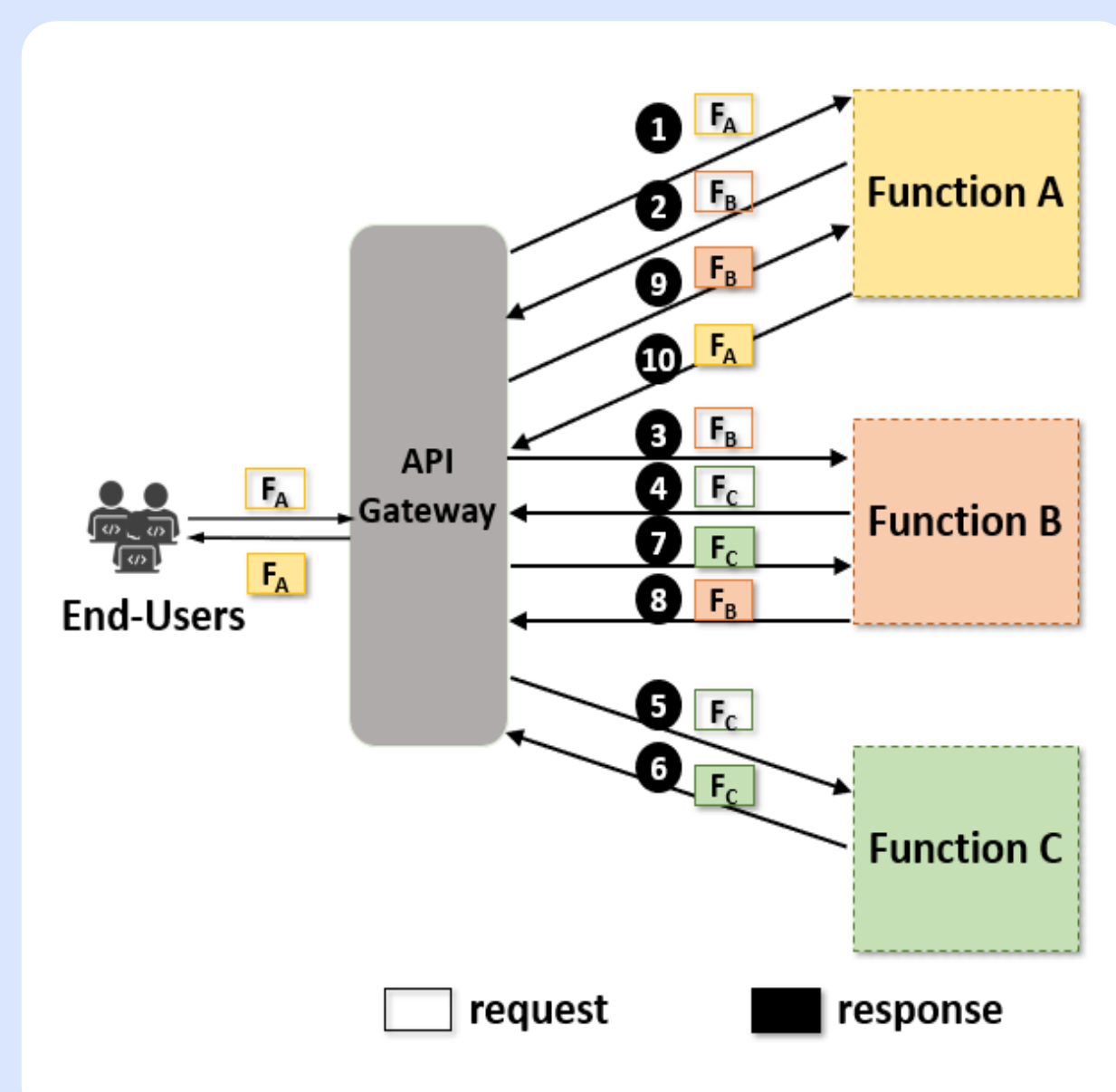
¹Zhejiang University, ²Alibaba Cloud, ³Hangzhou Institute of Technology, Xidian University, ⁴Northwestern University



Introduction

Background

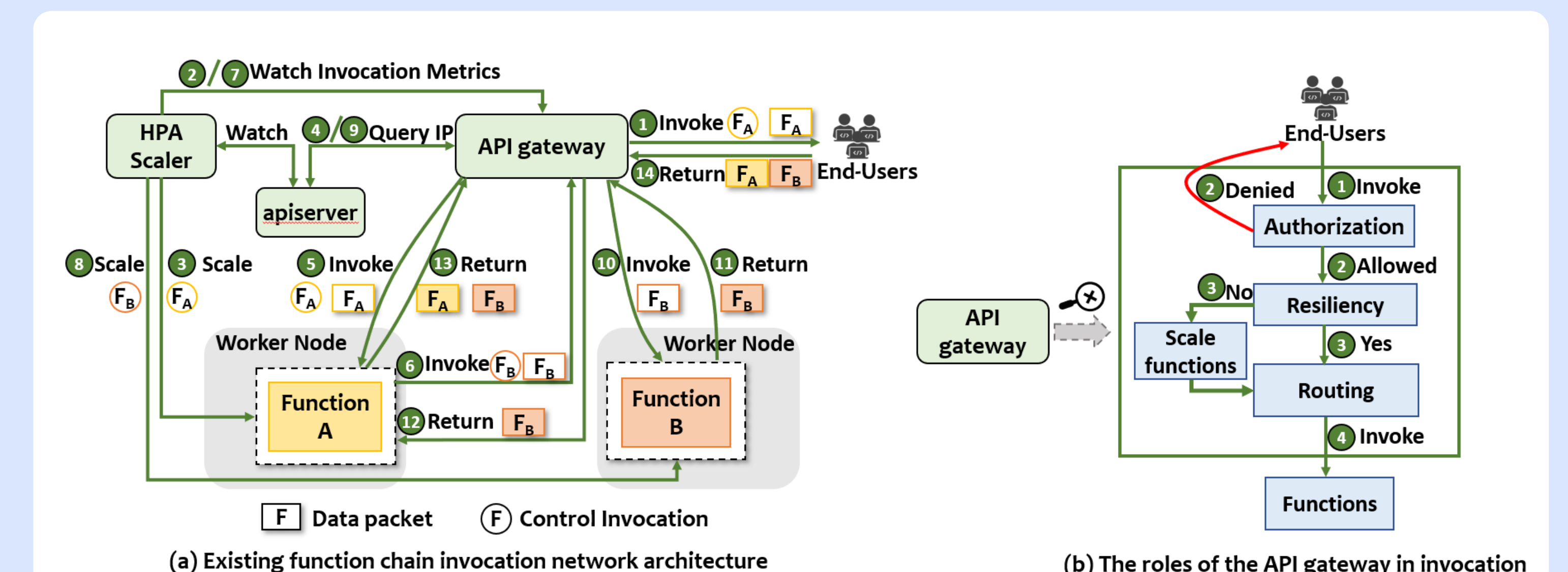
- Serverless computing has gained popularity for deploying web applications. Running web applications without the need to configure the runtime environment is attractive for cloud tenants, especially when their management and provisioning of computing resources are complex. In serverless computing, web applications run as function chains. However, function chain communications in the current serverless architecture are not efficient. Serverless function chains are triggered by the API gateway through external events or HTTP requests.



- The centralized API gateway assists in forwarding requests between functions, performing 5 internal requests.
- Compared to direct function-to-function invocations which only need 3 internal invocations, the current architecture introduces 2 additional network round trips.
- The resulting additional network round trips add more execution time for cloud applications, potentially compromising service level objectives (SLOs).

Motivation

- The API gateway greatly facilitates function invocations for end-users. However, the effectiveness of serverless networking is compromised for such convenience.
 - P1: Multiple extra connections increase invocation latency:* A function chain needs to call the API gateway multiple times to forward requests.
 - P2: The centralized architecture is prone to traffic bottlenecks:* The API gateway performs various roles during the function invocation. The heavy tasks consume numerous resources, making the API gateway become the traffic bottleneck.
 - P3: Lack of data flow isolation compromises security:* The data flows from multi-tenants coexist within the API gateway. The internal attack surface is expanded in the current serverless architecture.



Design

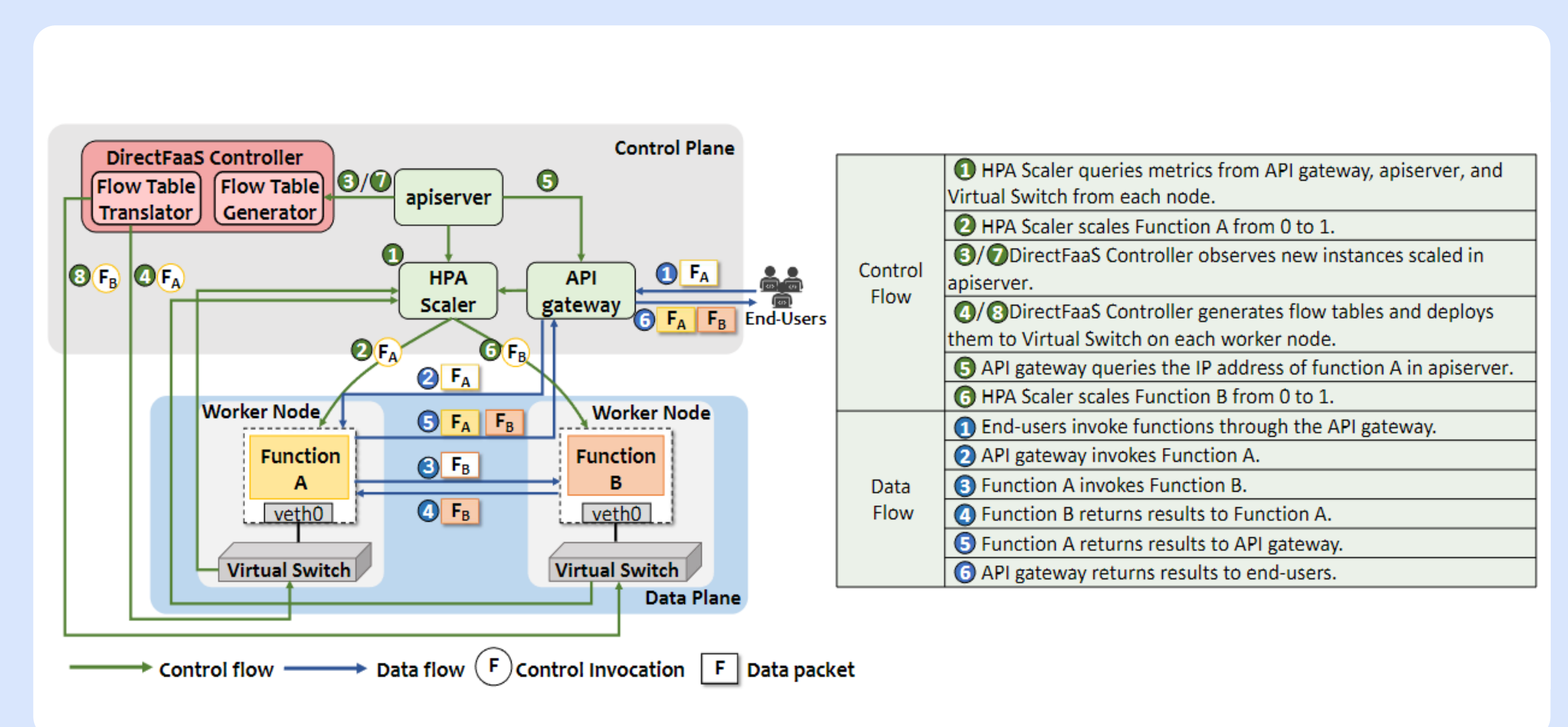
Challenges

- We believe that direct communication between functions is the key to addressing these networking problems. However, due to the various roles that the API gateway performs during the function invocation, releasing the API gateway from data forwarding is not easy.
 - C1: Handling the authorization of the internal function invocation.*
 - C2: Routing functions without knowing the IP address.*
 - C3: Scaling functions when functions are scaled to zero.*

Solutions

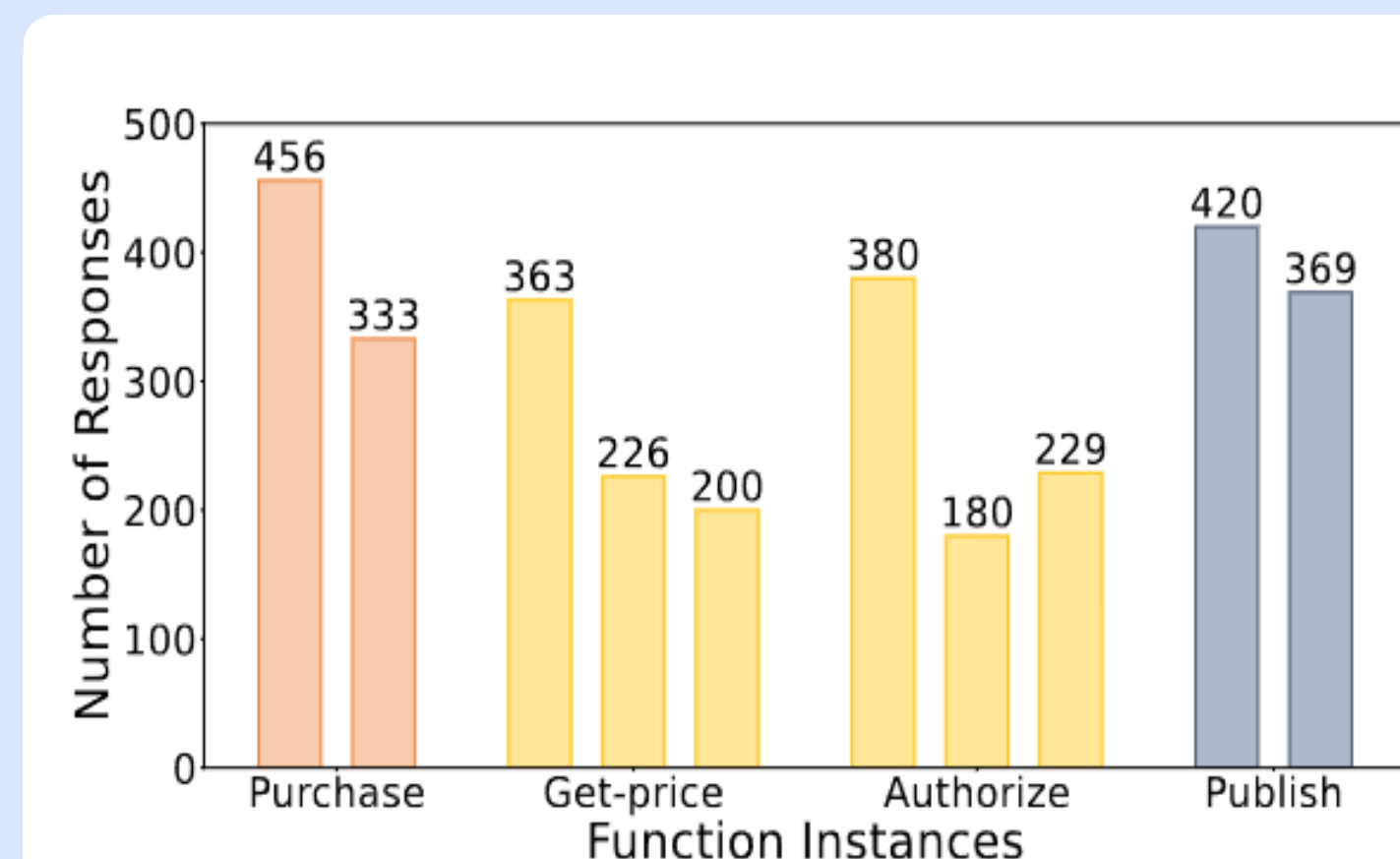
- DirectFaaS* architecture separates the control plane and the data plane in serverless chain communications. Components of the control plane are responsible for generating control flows. Following these control flows, internal functions communicate directly in the data plane.
 - S1: Authorizing internal functions through the network policy.*
 - S2: Using the static virtual IP (vIP) to route internal function communications.*
 - S3: Monitoring internal function invocations to scale function instances.*

DirectFaaS Network Architecture



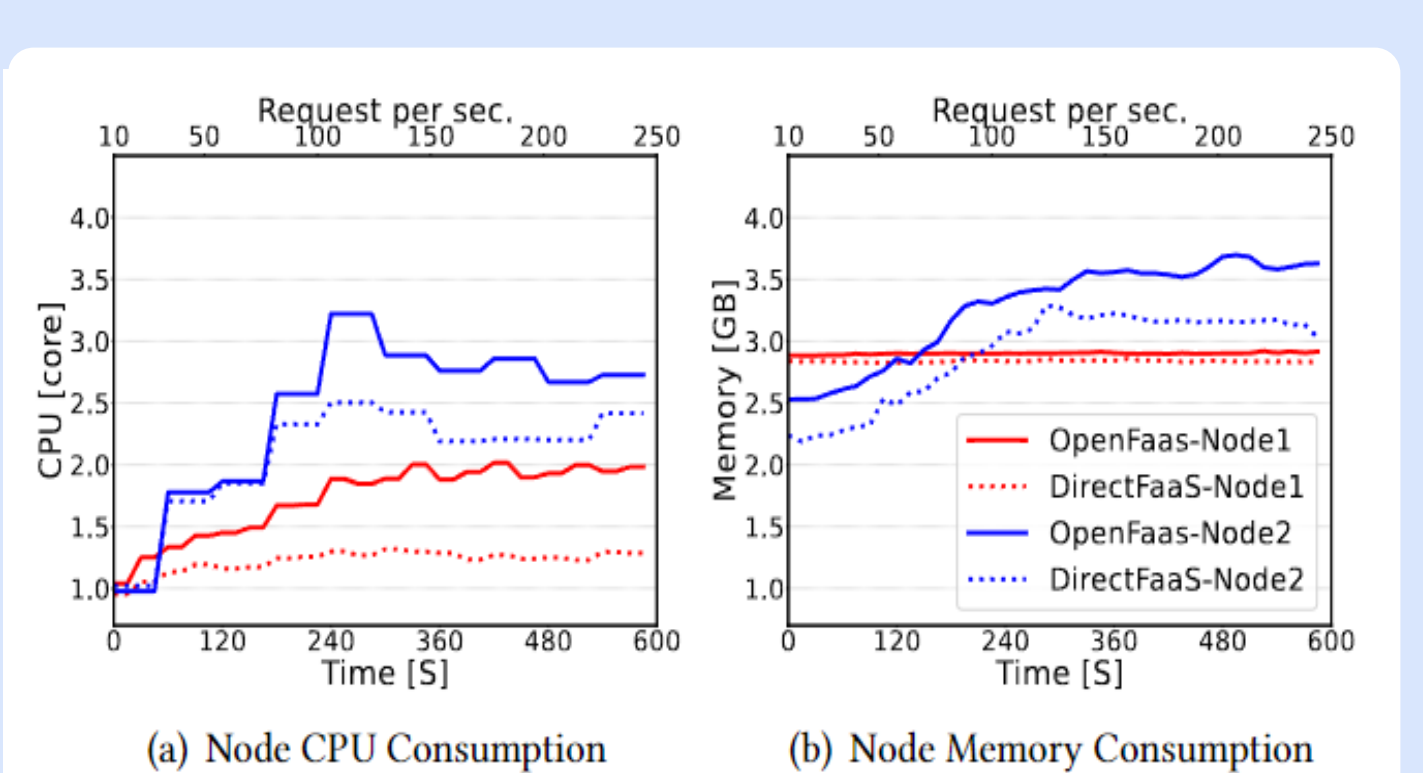
Evaluation

Authorization, Routing, Resiliency

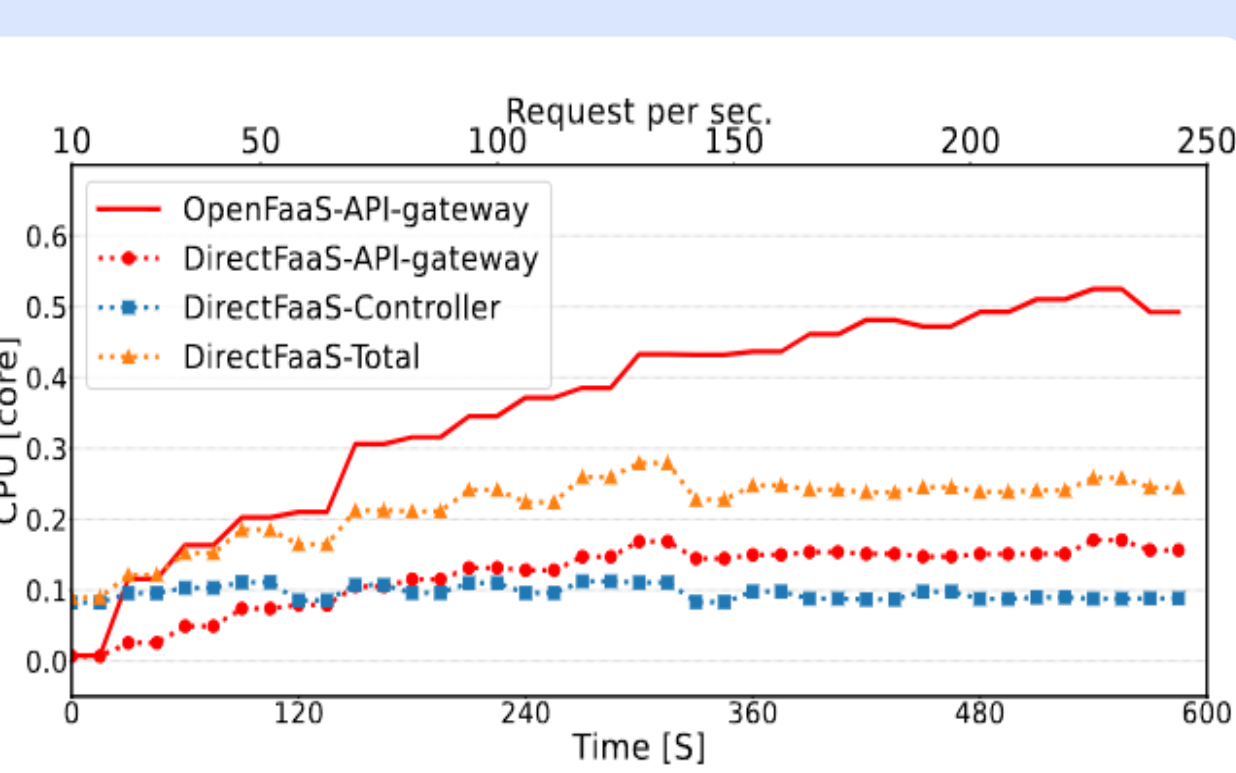


- Requests only successfully be forwarded to functions that have invocation relationships.
- When we invoke the function chain using vIP, each internal function successfully responds to all requests.
- Functions are scaled to different quantities of instances based on the number of invocations.

Resource Consumption Reduction

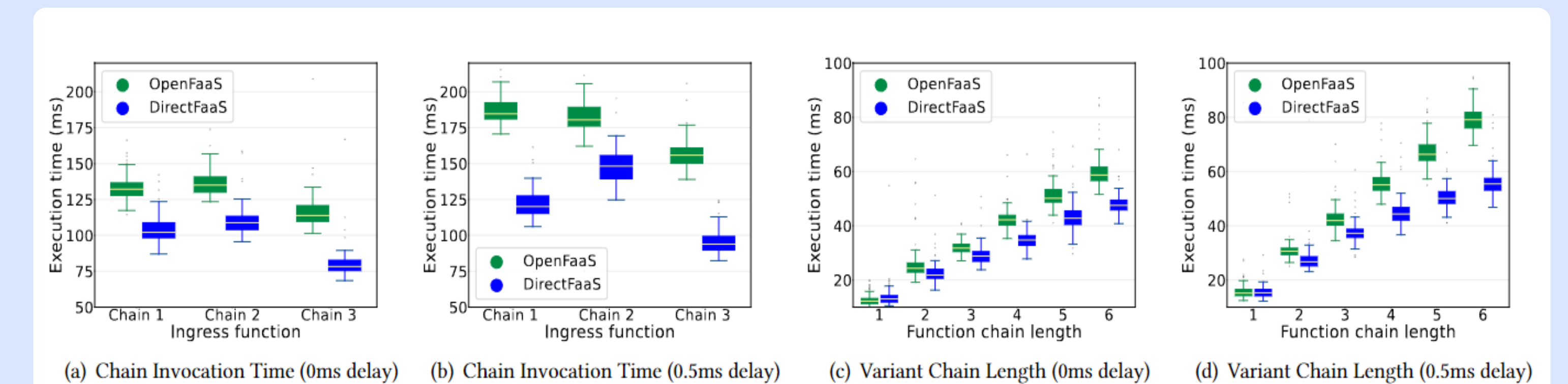


- DirectFaaS* achieved a 30.1% reduction in CPU consumption and a 13.8% reduction in Memory consumption on the control node.



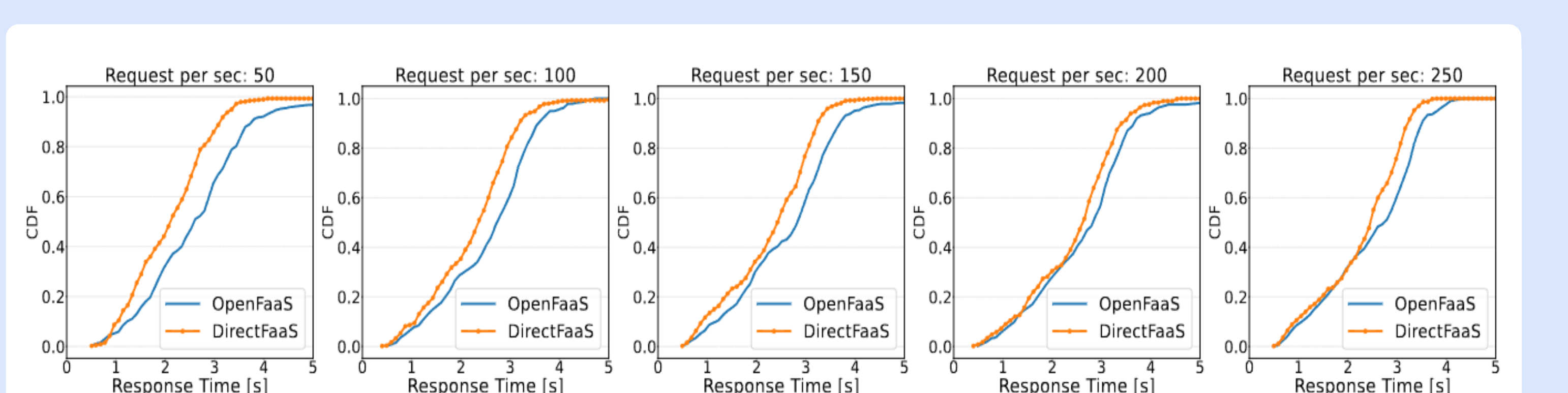
- The OpenFaaS API gateway consumes more CPU than the total control components of *DirectFaaS*.

Execution Time Reduction



- DirectFaaS* reduces 24.4% and 30.9% invocation time with 0ms delay and 0.5ms delay, respectively. It can save up to about 24ms in one end-user request when the chain length is 6 with a 0.5ms delay.

Response Time Reduction



- DirectFaaS* has less response time than OpenFaaS, meaning it can serve more rps. So under the same external rps, *DirectFaaS* only need fewer function instances to meet the requests.