Gringotts: Fast and Accurate Internal Denial-of-Wallet Detection for Serverless Computing

Junxian Shen🎙 , Han Zhang, Yantao Geng

Jiawei Li, Jilong Wang, Mingwei Xu

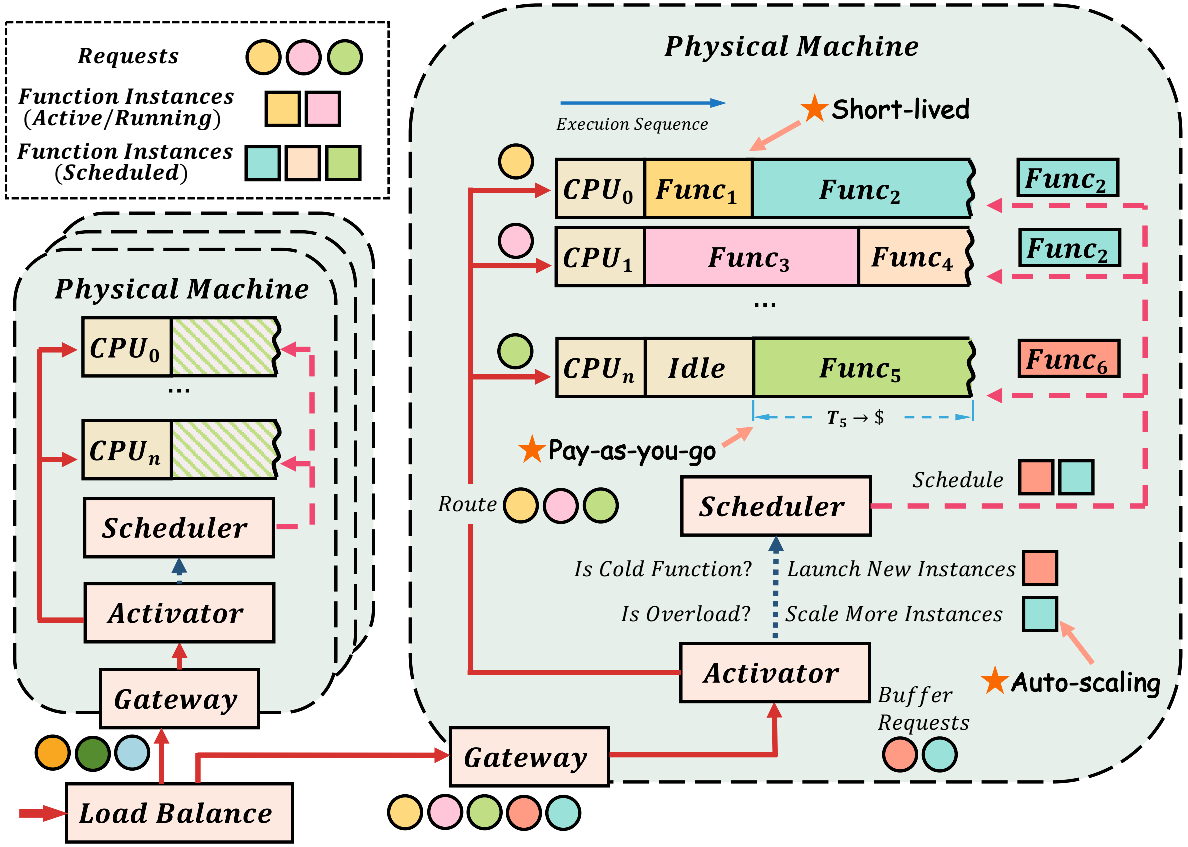
清華大學
Tsinghua University

- **Background and Motivation**
- Denial-of-Wallet Attack
- Design
- Evaluation
- Conclusion

Emerging Serverless Computing

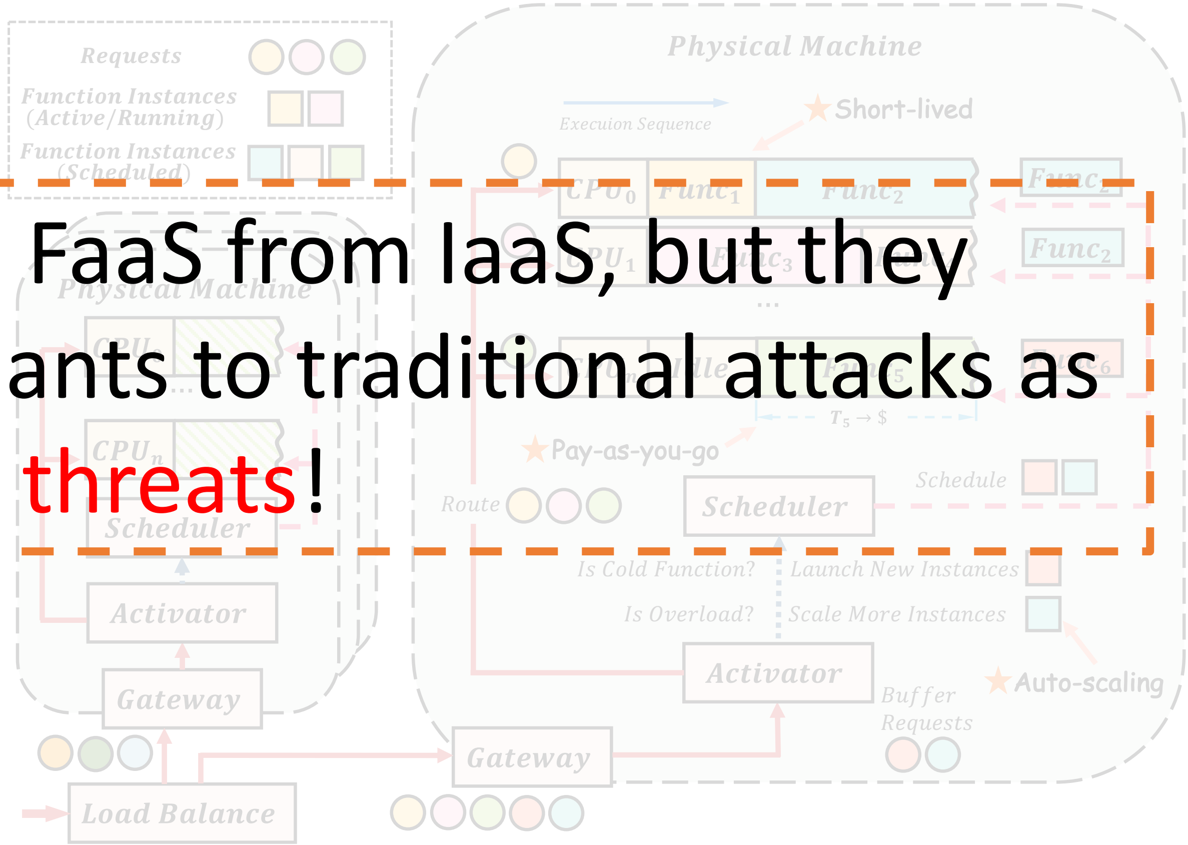
- Additional responsibilities
  - task deployment
  - environment configuration
  - auto-scaling

- Key features
  - pay-as-you-go
  - auto-scaling
  - short-lived



Emerging Serverless Computing

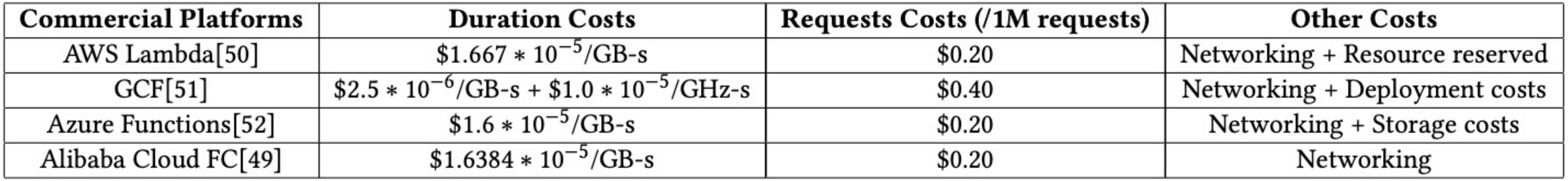
- Key features
  - pay-as-you-go
  - auto-scaling
  - short-lived



These features distinguish FaaS from IaaS, but they also expose serverless tenants to traditional attacks as well as additional security threats!

Serverless Billing Model

- Billing model = Duration + Requests + Other
- Functions can either actively (e.g., sleep) or passively (e.g., wait for I/O operations) yield their allocated CPUs during the execution
- Victims will still be charged until their functions **return the response**
    - Resource contention (Prolonged execution time) + Billing model = **Financial Exhaustion**!



Commercial Platforms	Duration Costs	Requests Costs (/1M requests)	Other Costs
AWS Lambda[50]	$\$1.667 * 10^{-5}$/GB-s	\$0.20	Networking + Resource reserved
GCF[51]	$\$2.5 * 10^{-6}$/GB-s + $\$1.0 * 10^{-5}$/GHz-s	\$0.40	Networking + Deployment costs
Azure Functions[52]	$\$1.6 * 10^{-5}$/GB-s	\$0.20	Networking + Storage costs
Alibaba Cloud FC[49]	$\$1.6384 * 10^{-5}$/GB-s	\$0.20	Networking

Table 1: Current billing models of major commercial serverless platforms.

- Background and Motivation
- **Denial-of-Wallet Attack**
- Design
- Evaluation
- Conclusion

Denial-of-Wallet Attack

- DoW attack
  - a variant of the Denial-of-Service (DoS) attack specifically conducted on serverless platforms

- Three key differences
  - Different focus
  - Different results
  - Different financial consequences

- External DoW attack
  - repeatedly invoking the APIs that the victims unwittingly expose
  - can be defended by existing DoS detection mechanisms
    - ingress filtering, traceback, source validation
- Internal DoW attack
  - triggering resource contentions on shared hardware

Memory Bus Locking

- Memory bus locking
  - atomic memory operation crossing the cache line boundary will triggers the briefly halt of other memory operations

- Requires no additional privileges
- Simple to implement
- Feasible

- Not fundamental - contentions on shared resources
  - PCIe I/O switches
  - caches
  - power management

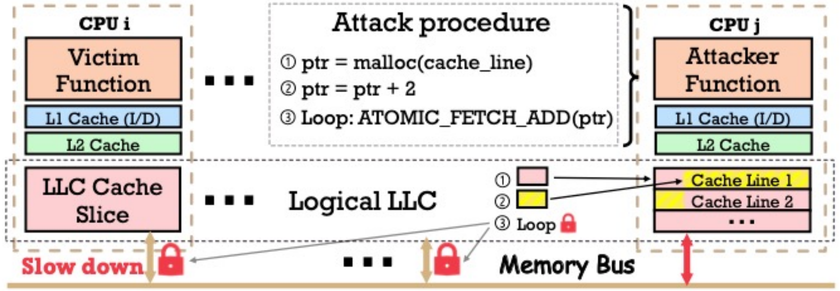


Figure 2: Overview of memory bus locking.

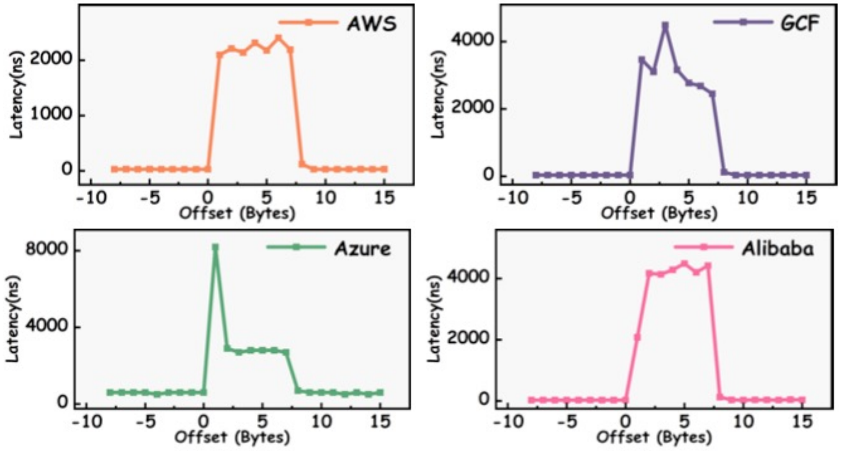
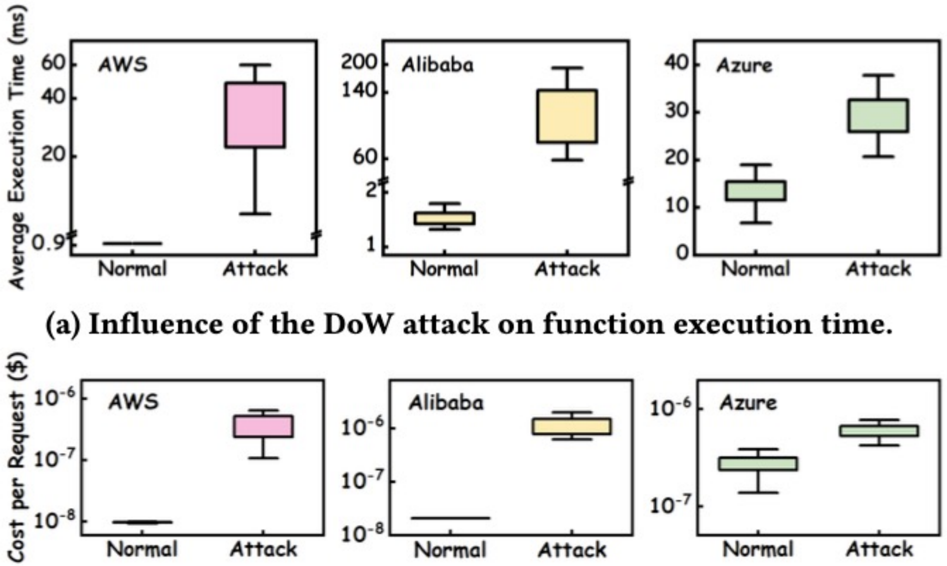


Figure 3: Latency spikes caused by atomic operations.

Denial-of-Wallet Attack

- Step 1: Malicious function placement
- Step 2: Create resource contentions
- Step 3: Direct financial exhaustion

- A hypothetical victim scenario
  - a mobile backend application
    - 1536 MB memory
    - 3 million requests per month (120ms)
  - $2.73 ➜ **$326.33**



(a) Influence of the DoW attack on function execution time.

(b) Influence of the DoW attack on per-request cost.

Accurate Detection is Challenging

- Errors introduced by improper sampling

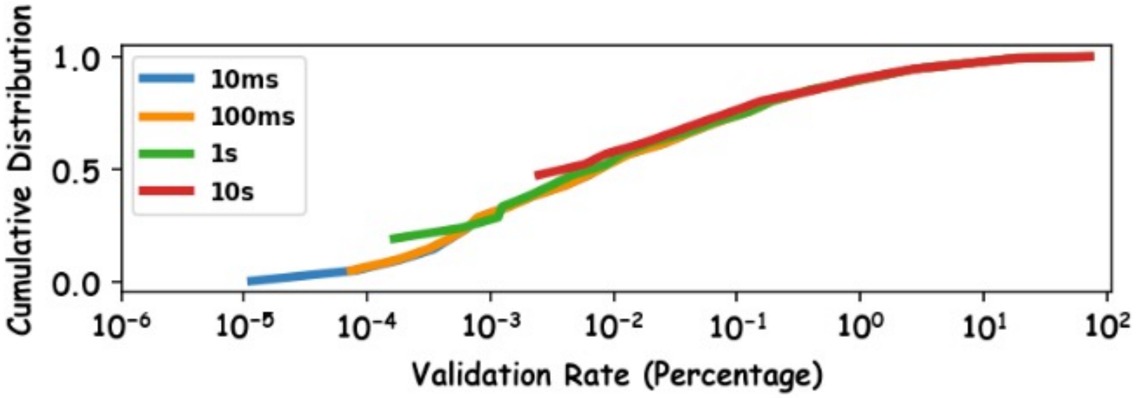


Figure 6: Sample validation rate on Azure traces [80].

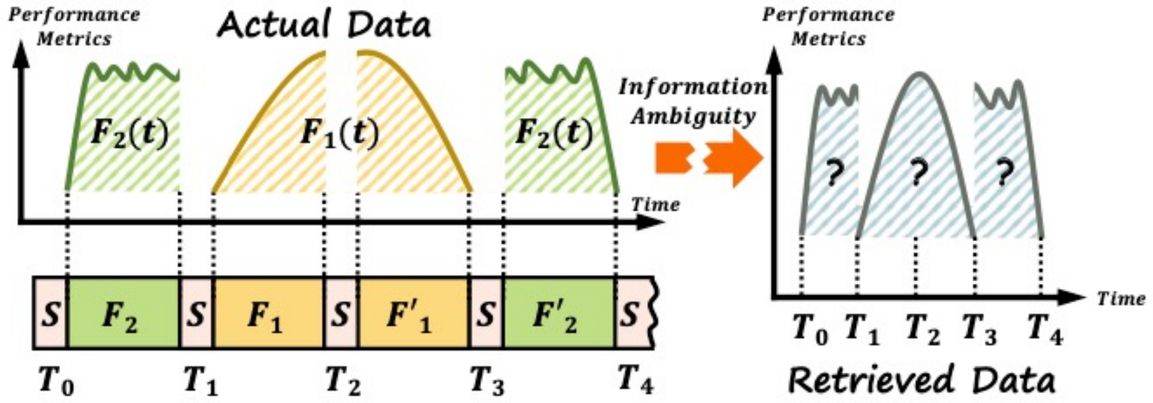
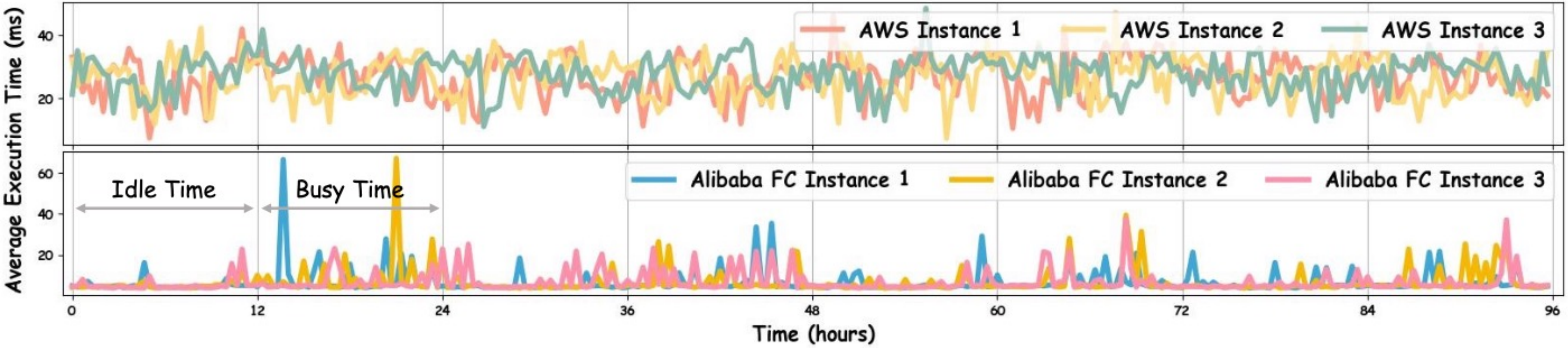


Figure 7: Sampling with a fixed-length window.

Accurate Detection is Challenging

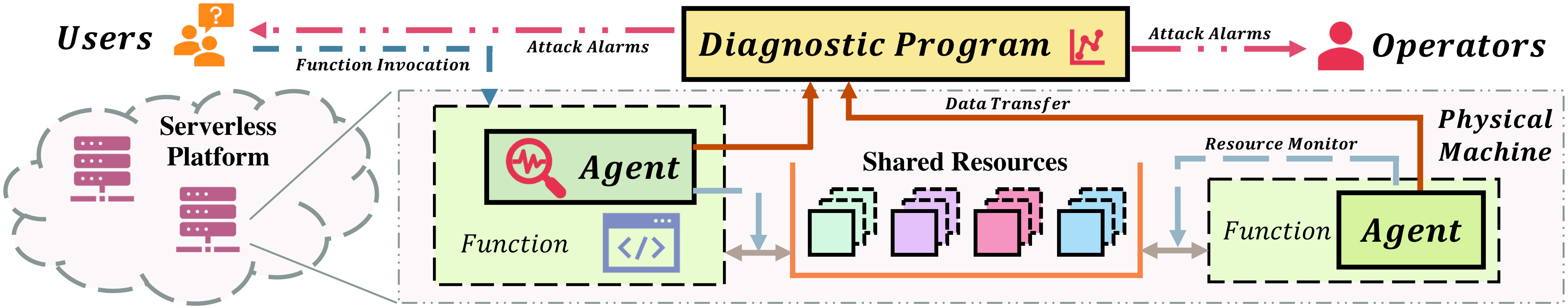
- Errors introduced by improper sampling
- Errors introduced by the noisy environment



- Background and Motivation
- Denial-of-Wallet Attack
- **Design**
- Evaluation
- Conclusion

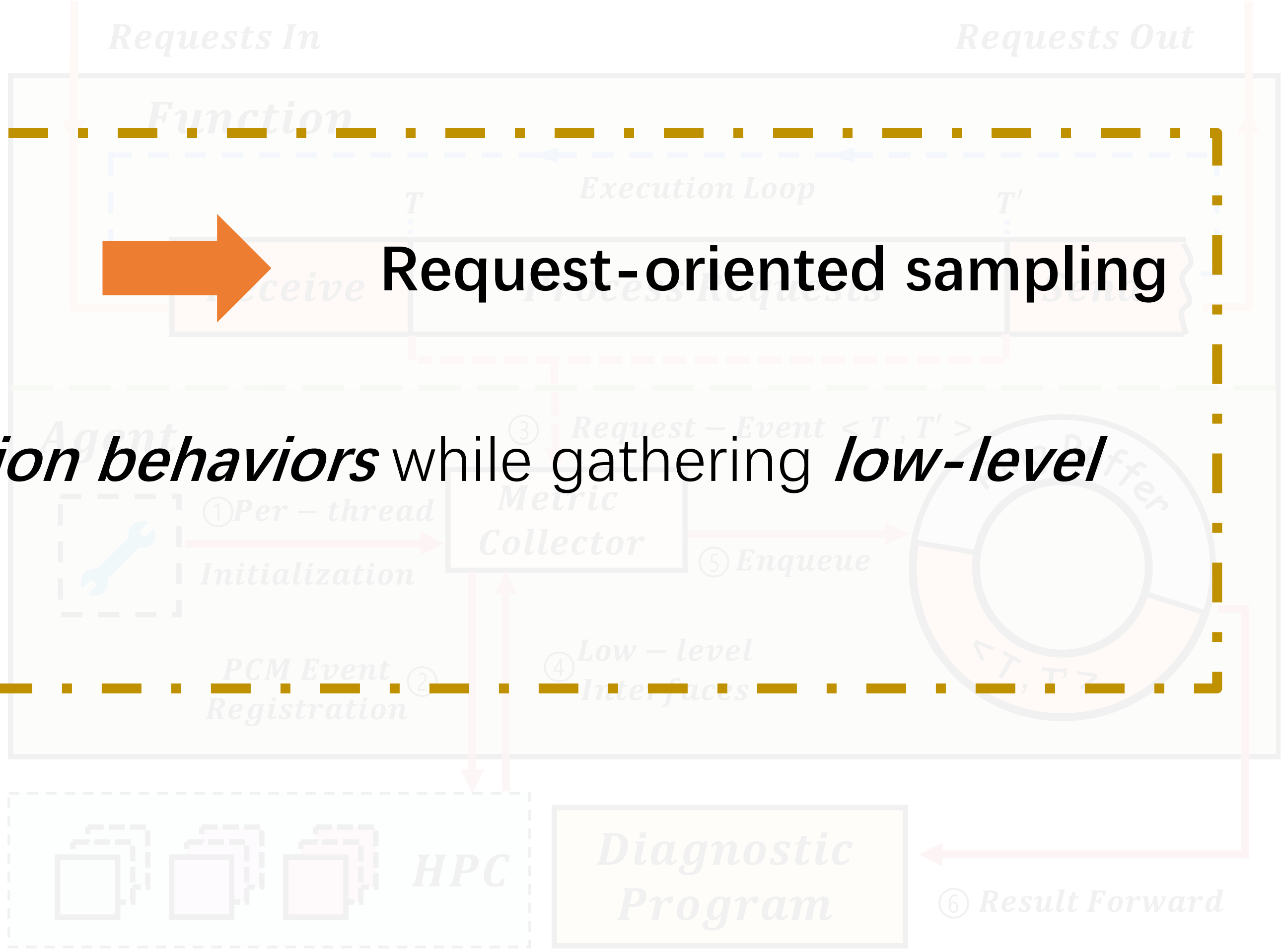
Overview of Gringotts

- Agent
  - monitoring the resource utilization of the target functions
- Diagnostic Program
  - performs the actual detection atop physical machines



Performance Metric Collection

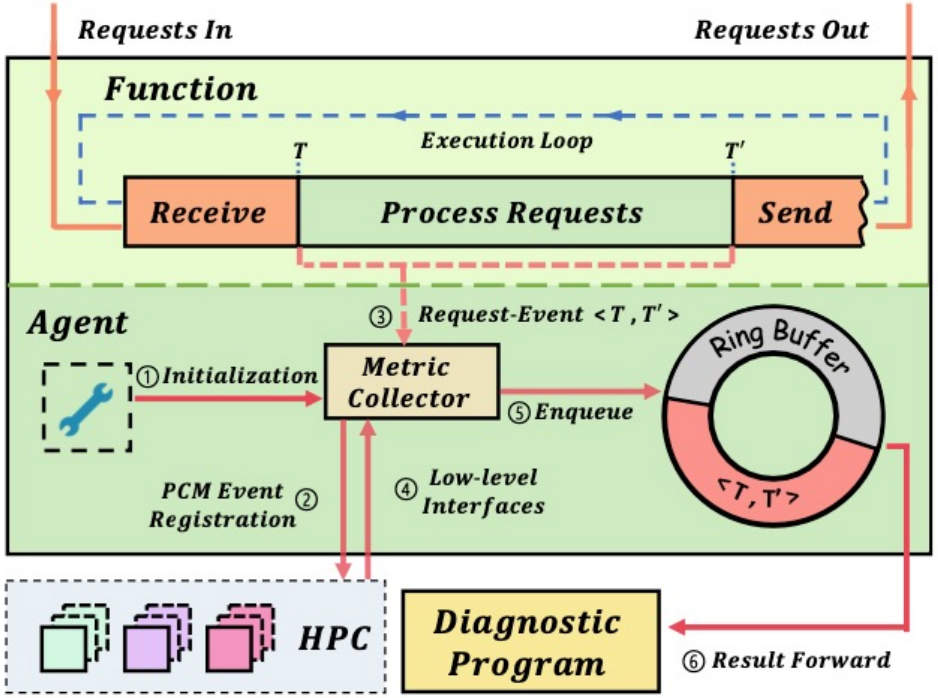
- **Fixed-length sampling method** → **Request-oriented sampling**
- Coordinate ***the high-level function behaviors*** while gathering ***low-level performance metrics***



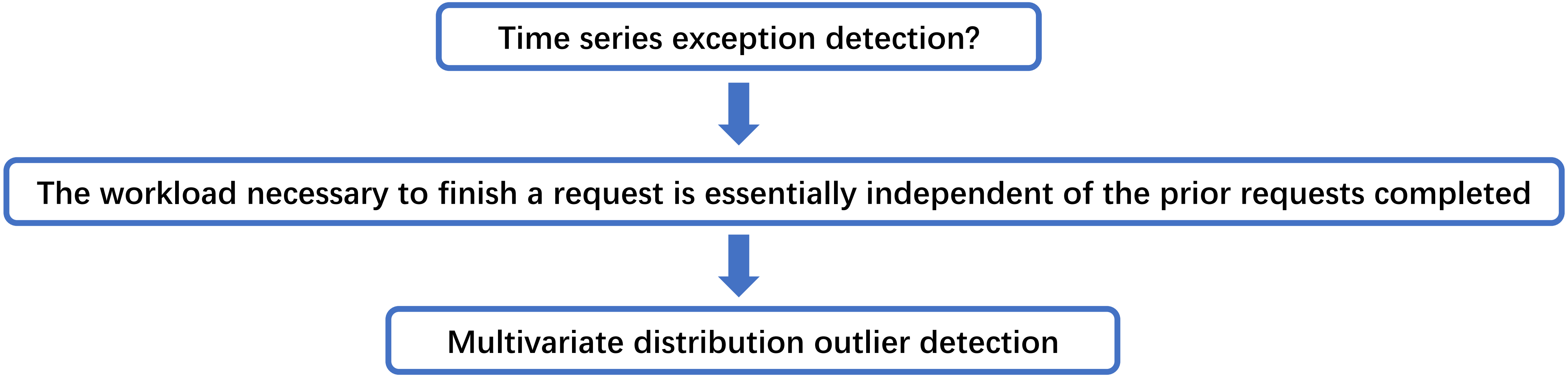
Performance Metric Collection

- P1: multiplexing-related issue
  - metric collection behaviors be bound to the runtime
- P2: runtime destruction
  - dynamic library loading
- P3: changes in function behaviors
  - request-oriented nature of serverless architecture

- Negligible Performance Overhead
- Avoiding event-skid
- Easy-to-use



DoW Detection Model



Time series exception detection?

The workload necessary to finish a request is essentially independent of the prior requests completed

Multivariate distribution outlier detection

- processing time ∝ processing content

$$t_i \propto \sum_{j=1}^{c} W_i(m_i^j), \quad W_i(m_i^j) \propto m_i^j$$

DoW Detection Model

For training / testing sample $[m, t]$, $[\bar{m}, \bar{t}]$

Training

- Multivariate linear regression $\hat{T}(\beta, m') = \beta_0 + \beta_1 * {m'}^1 + \cdots + \beta_c * {m'}^c, \min_{\beta} \| M'\beta - T' \|_2^2$
- New Gaussian distribution $x = (t', {m'}^1, \ldots, {m'}^c, \varepsilon)^T \sim \mathcal{N}(\mu_x, \sum)$

Testing

- Construct prediction vector $\bar{x} = \left(t'', {m''}^1, \ldots, {m''}^c, \bar{\varepsilon}\right)^T$
- Mahalanobis distance $D(\bar{X}, \mu_x) = \sqrt{(\bar{X} - \mu_x)^T \Sigma^{-1} (\bar{X} - \mu_x)} \leq Threshold$

- Background and Motivation
- Denial-of-Wallet Attack
- Design
- **Evaluation**
- Conclusion

Evaluation

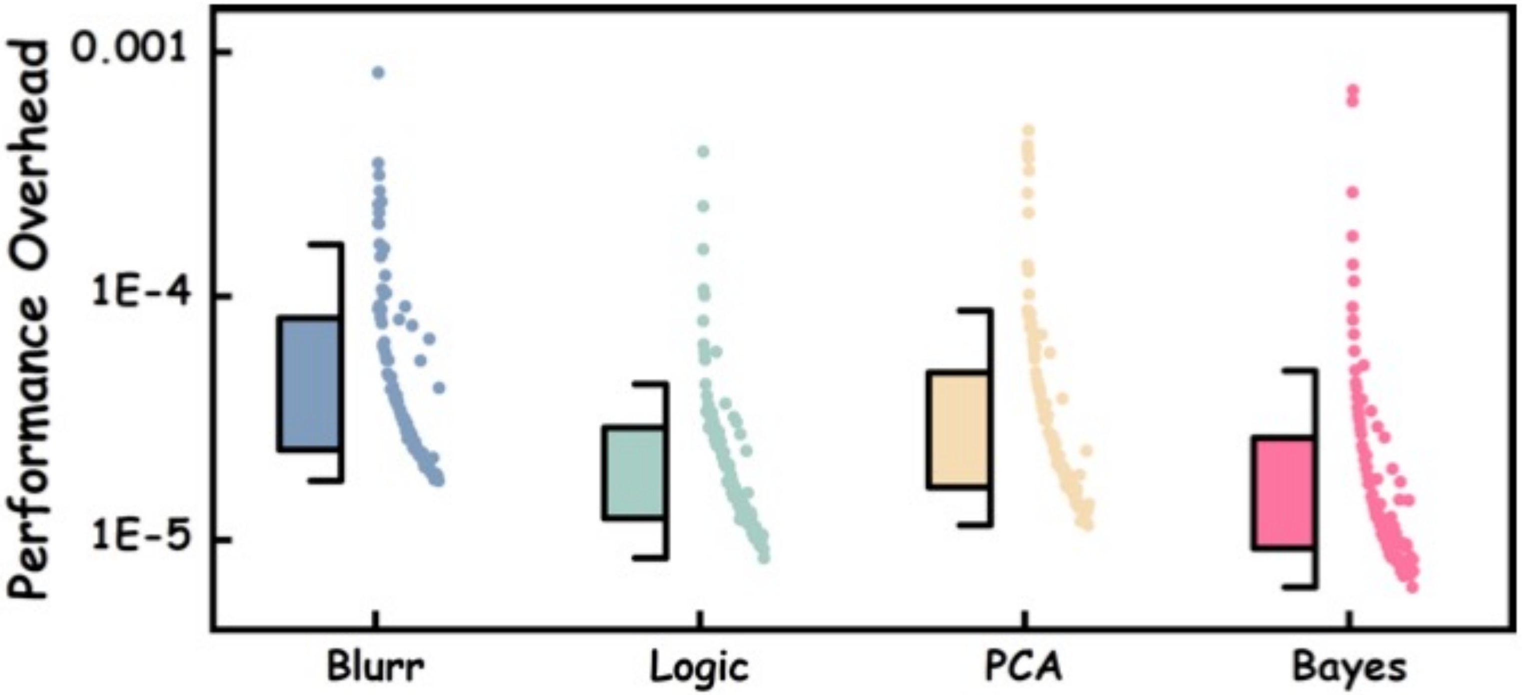
- Is the overhead of Gringotts negligible?

- How do the parameters of the diagnostic program affect the precision of Gringotts?

- What is the overall prediction accuracy of Gringotts’ detection model?

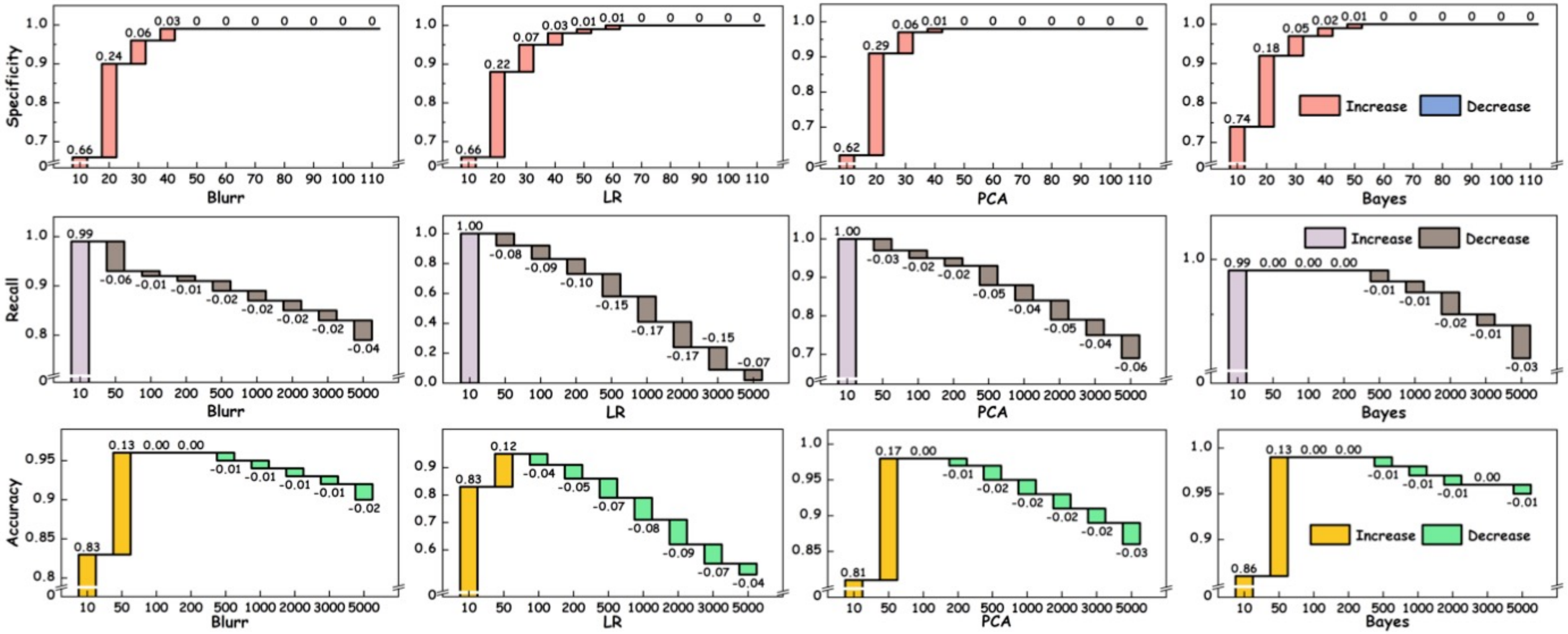
Performance Overhead

- The performance overhead of Gringotts is negligible



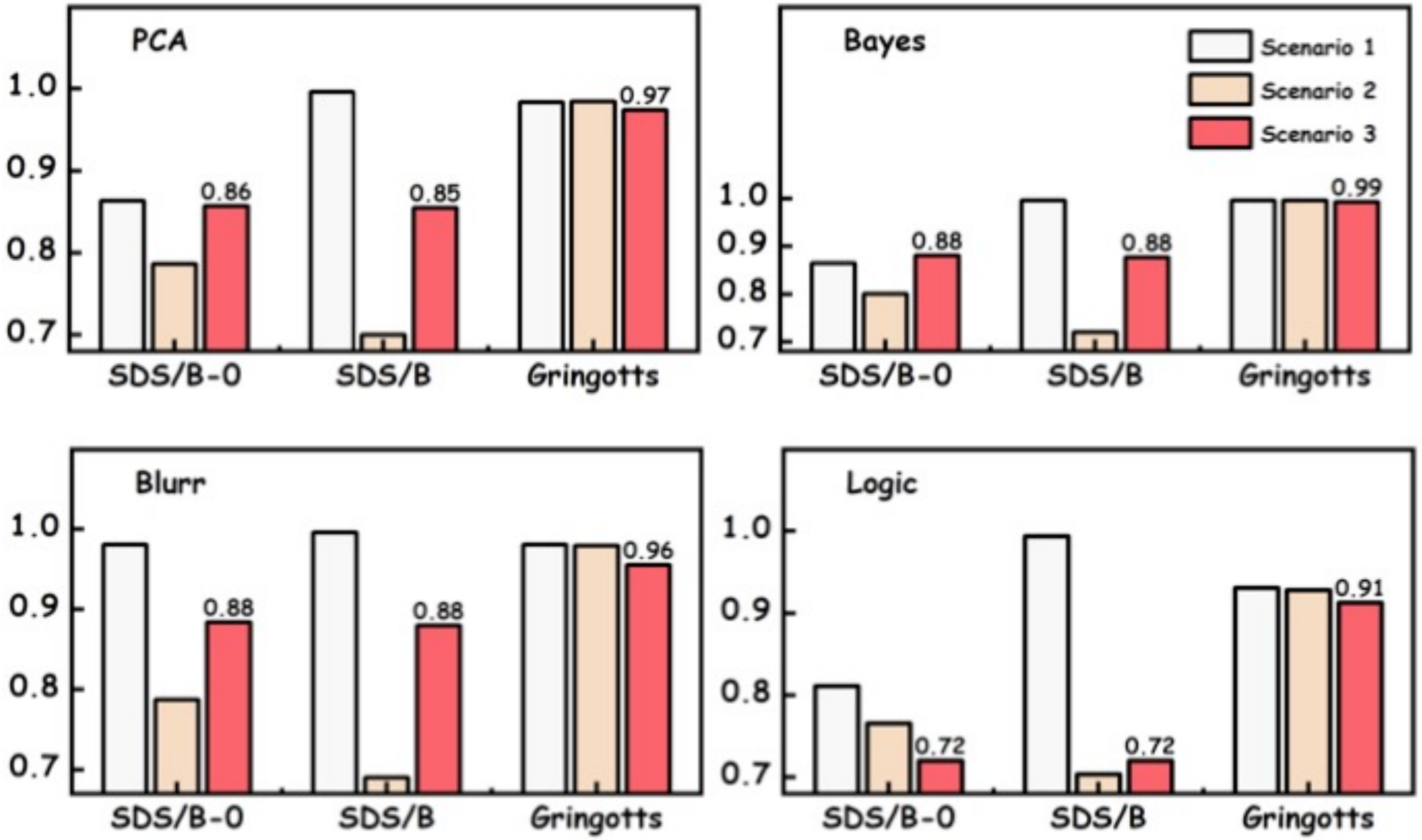
Influences of the Threshold

- Between the thresholds of 50 and 100, accuracy reaches its peak



End-to-end result

- Gringotts retains a high level of accuracy, from 91 percent to 99 percent



- Background and Motivation
- Denial-of-Wallet Attack
- Design
- Evaluation
- **Conclusion**

Conclusion

- Thoroughly analyze the Denial-of-Wallet attack
- Conduct a real-world DoW attack on commercial serverless platforms
- Implement Gringotts as a real system for potential DoW detection on the serverless platform