PrivApprox
Privacy-Preserving Stream Analytics
https://privapprox.github.io

[USENIX ATC’17]

Do Le Quoc, Martin Beck, Pramod Bhatotia, Ruichuan Chen, Christof Fetzer, Thorsten Strufe

July 2017
Motivation

Clients

Private data

Recommendation, Ads

Analysts

Strong privacy guarantees

High utility analytics in real-time

How to preserve users’ privacy while supporting high-utility data analytics for low-latency stream processing?
PrivApprox

Personal data should be stored locally under the clients’ control

Execution

- Latency/throughput guarantees
- Desired computing resources for query processing

PrivApprox

Analyst

Low latency
Zero-knowledge Privacy

Clients

Budget

Latency/throughput guarantees
#1: Approximate computing

**State-of-the-art-systems**

1. Compute
2. Add noise
3. (Privacy-preserving) approximate output

**Idea:** To achieve low latency, compute over a sub-set of data items instead of the entire data-set.

**Approximate computing**

1. Take a sample
2. Compute
3. Approximate output ± Error bound
#2: Randomized response

**Idea:** To preserve privacy, clients may not provide *truthful answers* all the time.

Provides *plausible deniability* for clients responding to sensitive queries; achieves *differential privacy* (RAPPOR [CCS’14])

---

Client

- "Head" → Truthful Answer
- "Tail" → "Yes"
- "Head" → "No"
Summary

**PrivApprox**: a privacy-preserving stream analytics system over distributed datasets

<table>
<thead>
<tr>
<th>Privacy</th>
<th>Zero-knowledge privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practical</td>
<td>Adaptive execution based on query budget</td>
</tr>
<tr>
<td>Efficient</td>
<td>Randomized response &amp; sampling techniques</td>
</tr>
</tbody>
</table>

Thank you!

https://privapprox.github.io