Distributed Architecture Applied for Large Trace Analysis and Visualization

Yonni CHEN KUANG PIAO

May 10, 2018

École Polytechnique de Montréal

Distributed Open Reliable Systems Analysis Lab

Presentation plan

1 Why a distributed system?

2 What we want

2 The proposed architecture

3 The Trace Analysis Server Protocol

4 Implementation

5 Evaluation

Why a distributed system?

Why would we use a distributed system for trace analysis?

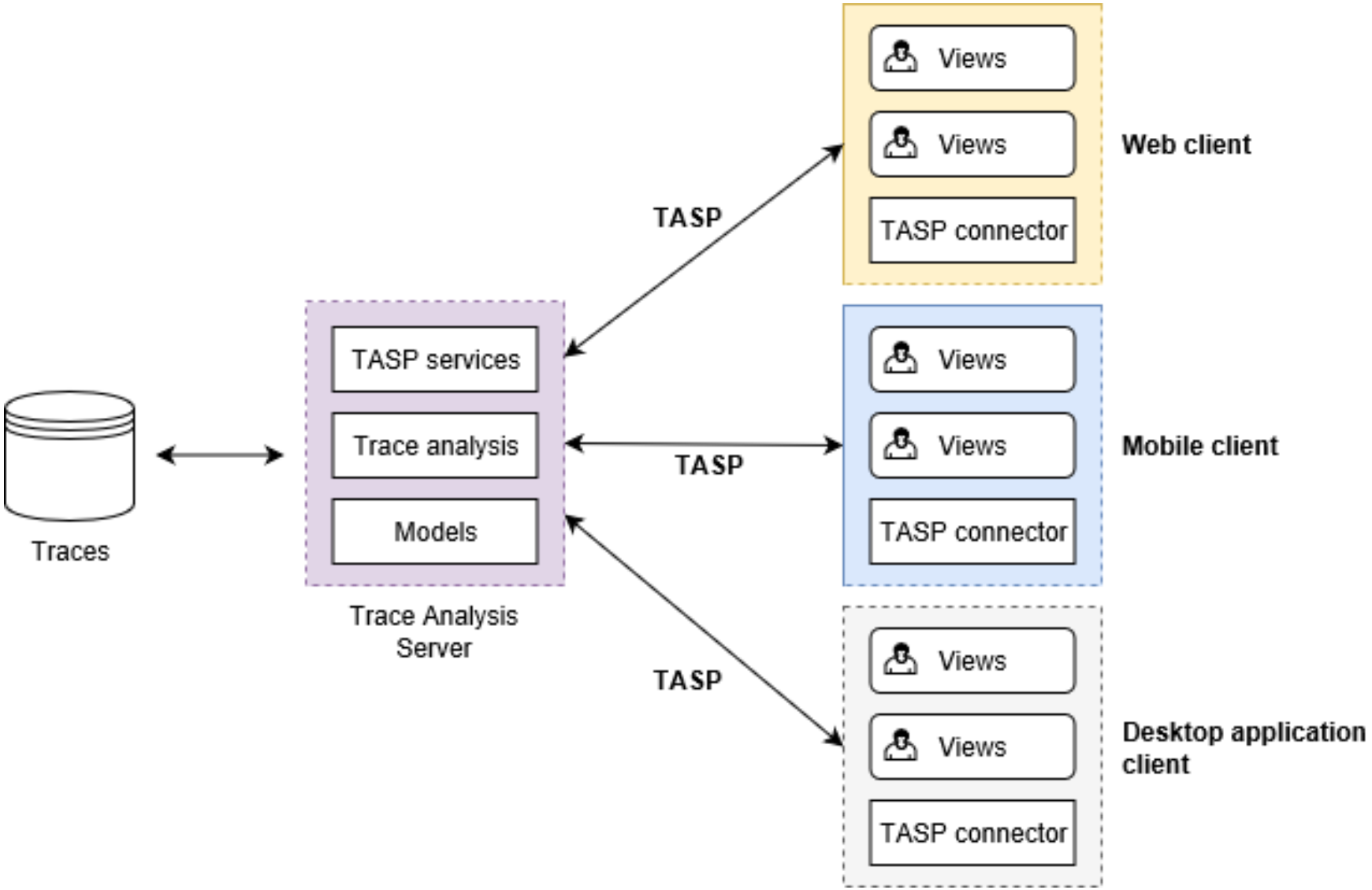
- Support multiple simultaneous clients
- More computing power
- Available everywhere
- Available anytime
- No special installation except a modern browser

What we want

We want a generic, flexible, performant and reusable system

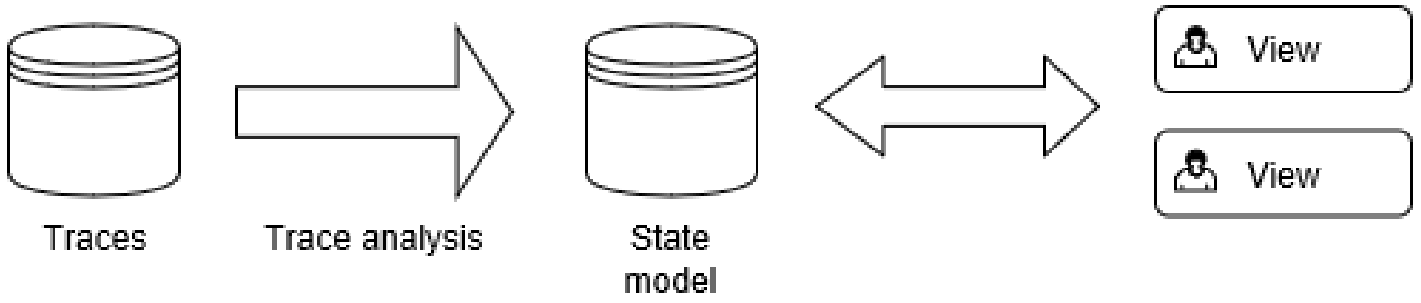
- Support multiple trace formats;
- Support for user-defined analysis;
- With the smallest overhead
  - In data transferred
  - In execution time
- Could be integrated in IDEs and monitoring systems;

The Architecture

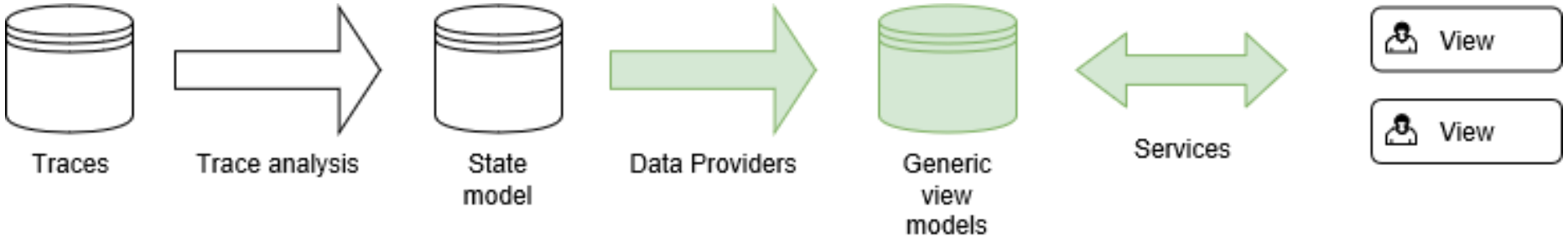


The Architecture - Server

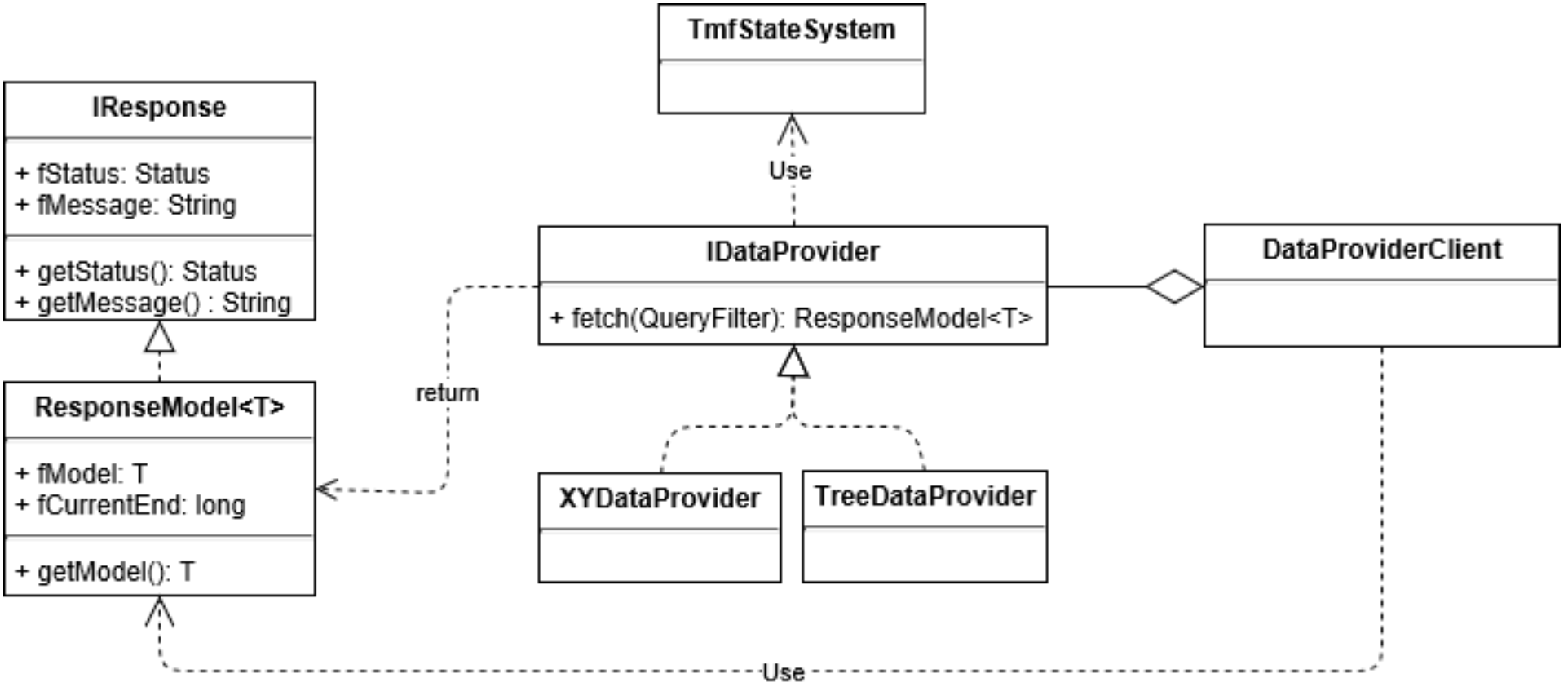
Common Trace Analysis Approach :



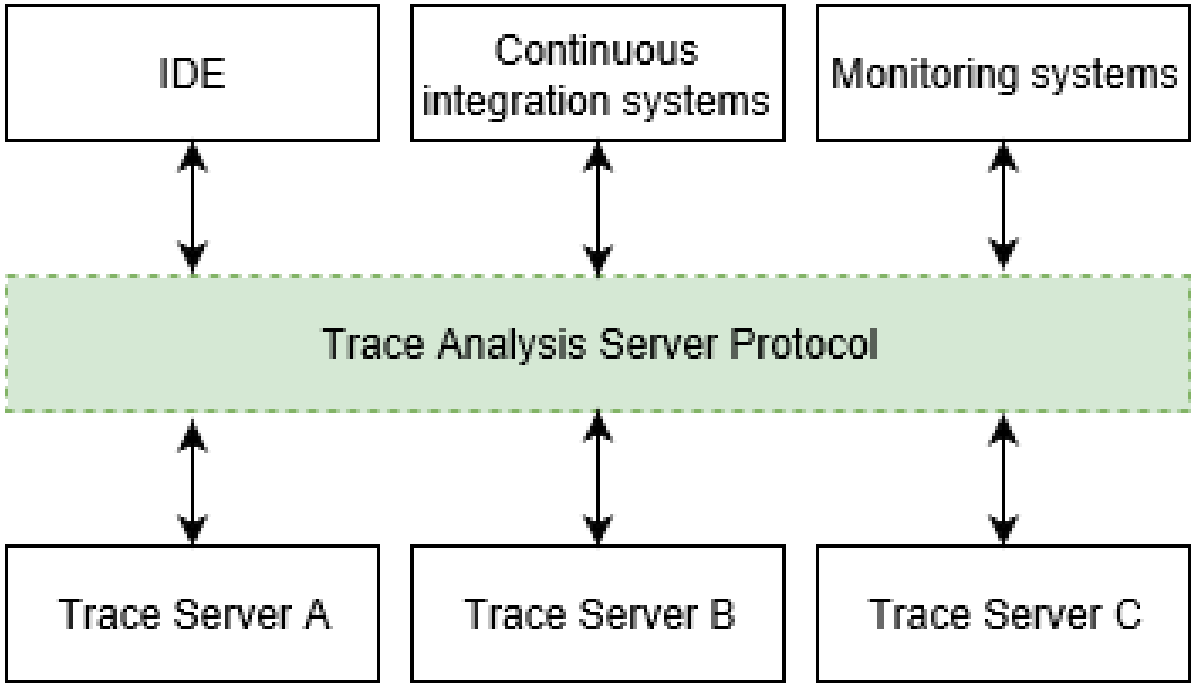
What we propose :



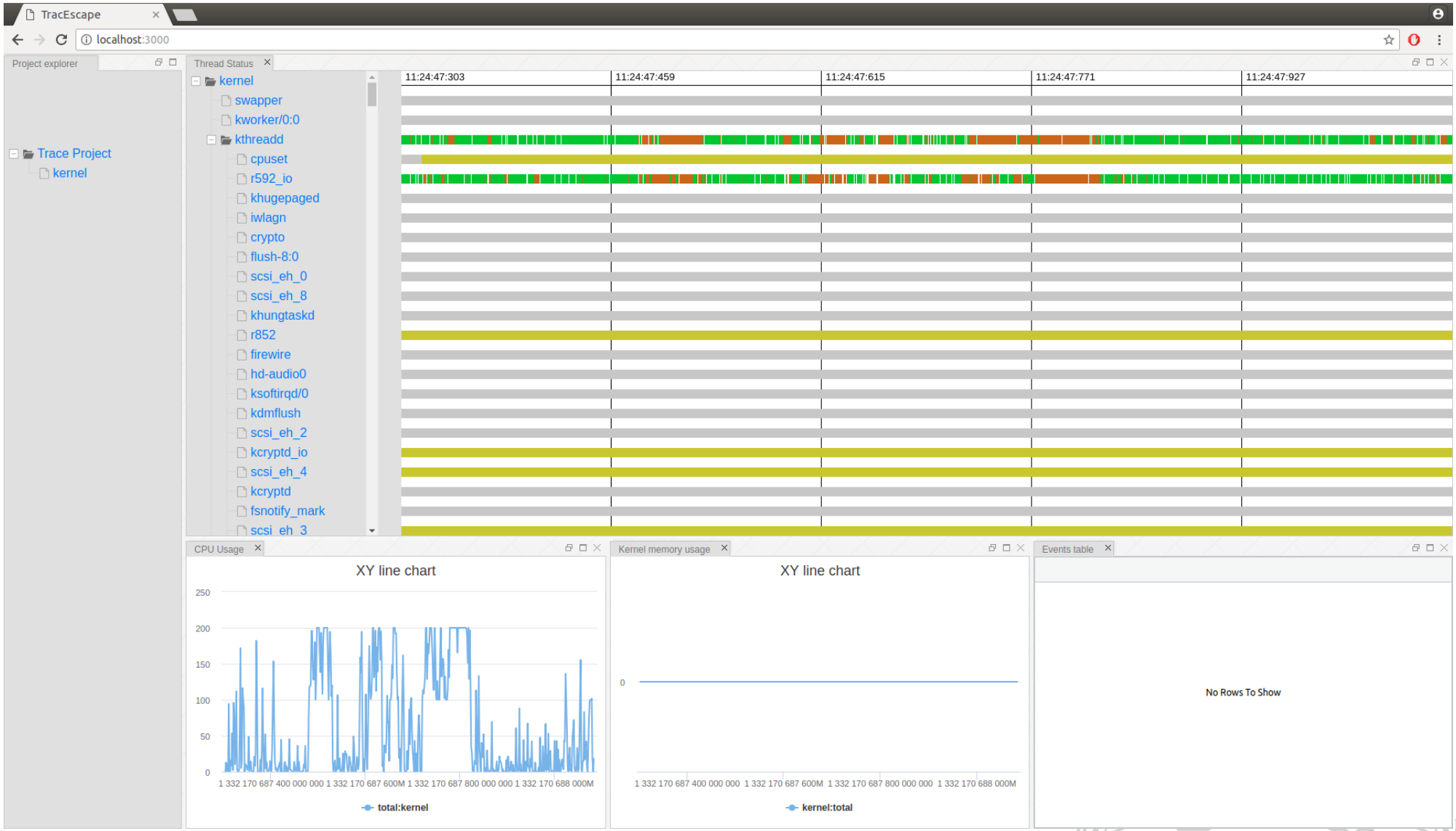
The Architecture - Server



The Trace Analysis Server Protocol



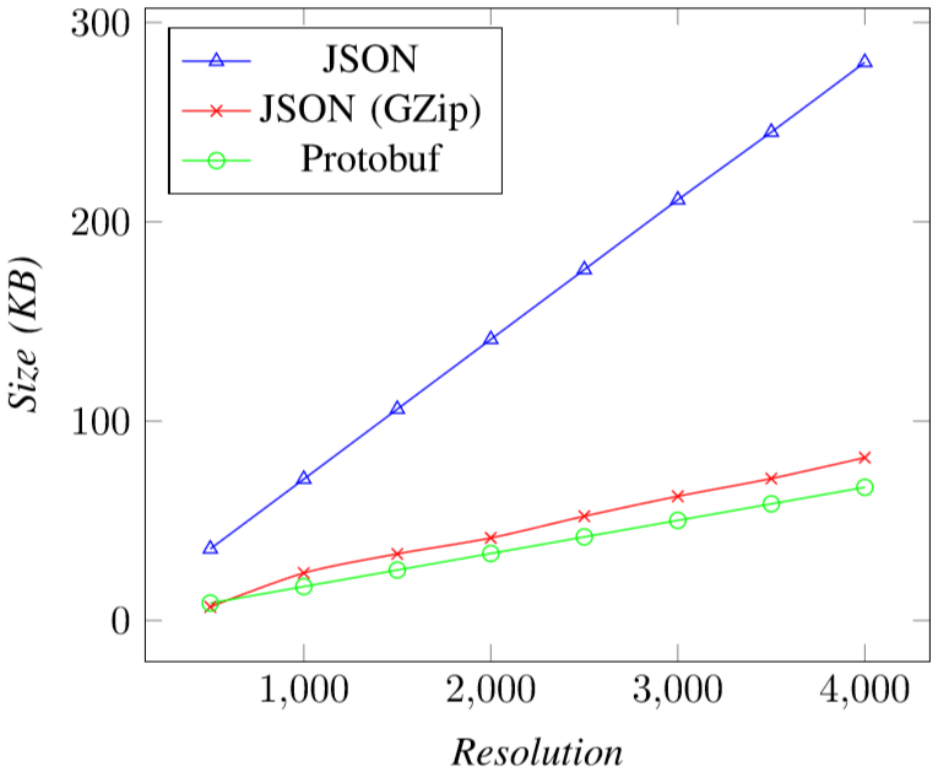
Implementation - TraceScape

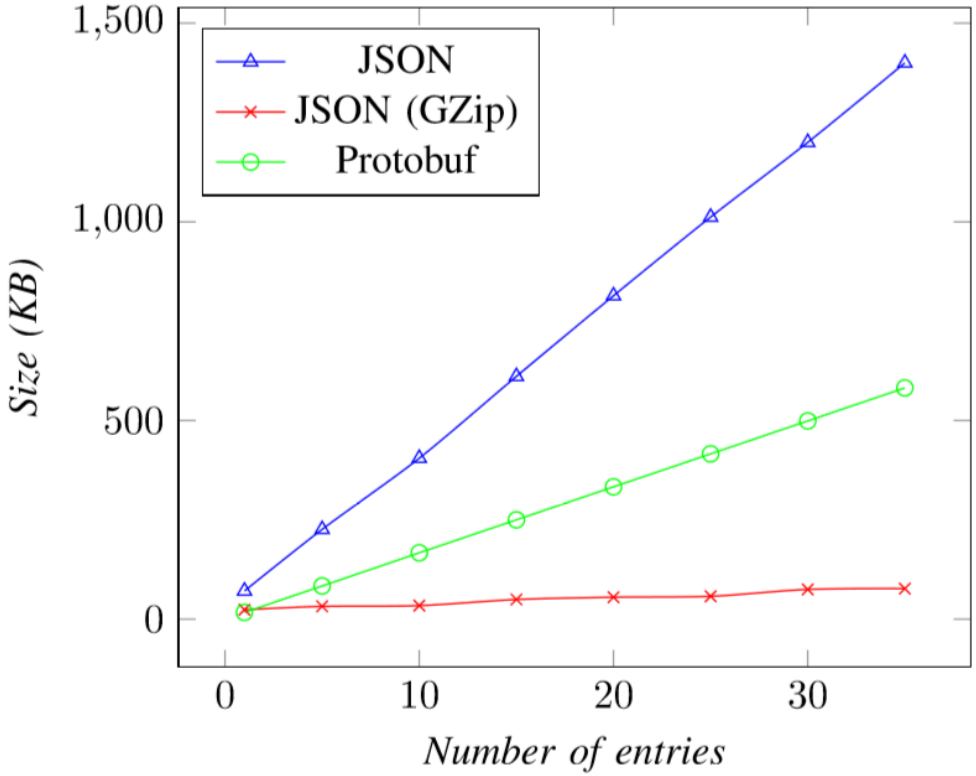


Evaluation

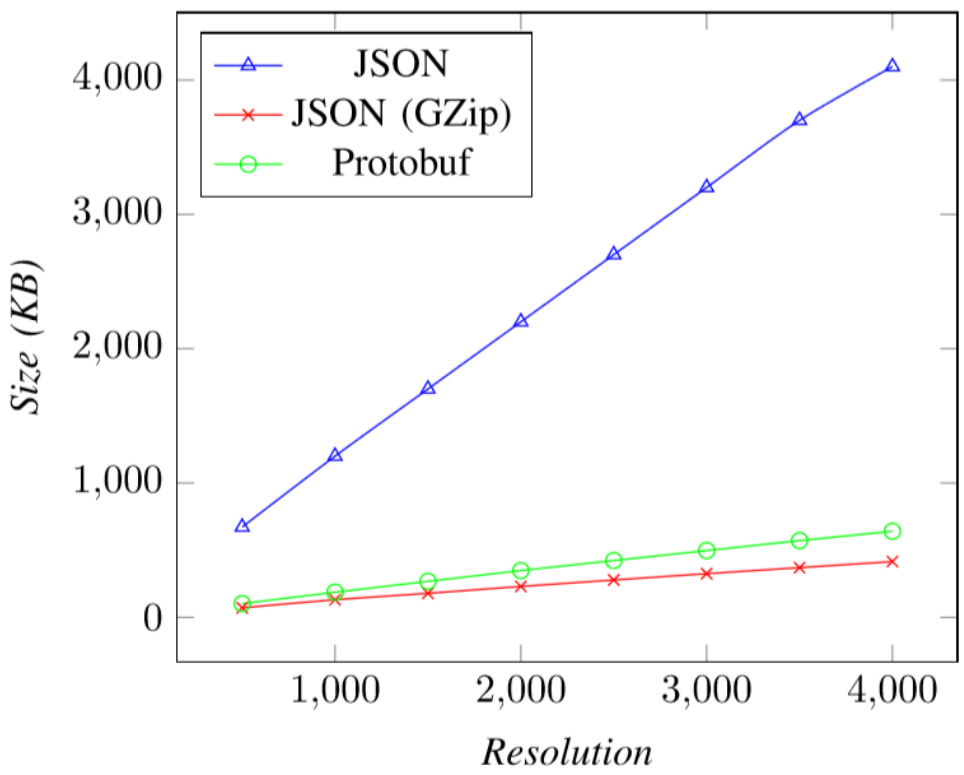
- We use a 2.5 GB trace for the evaluation
  - State model on disk : ~2.1 GB
- We request from the start to the end of the trace
- Data transferred per request
  - XY and time graph models
- Execution time overhead per request
  - XY and time graph models

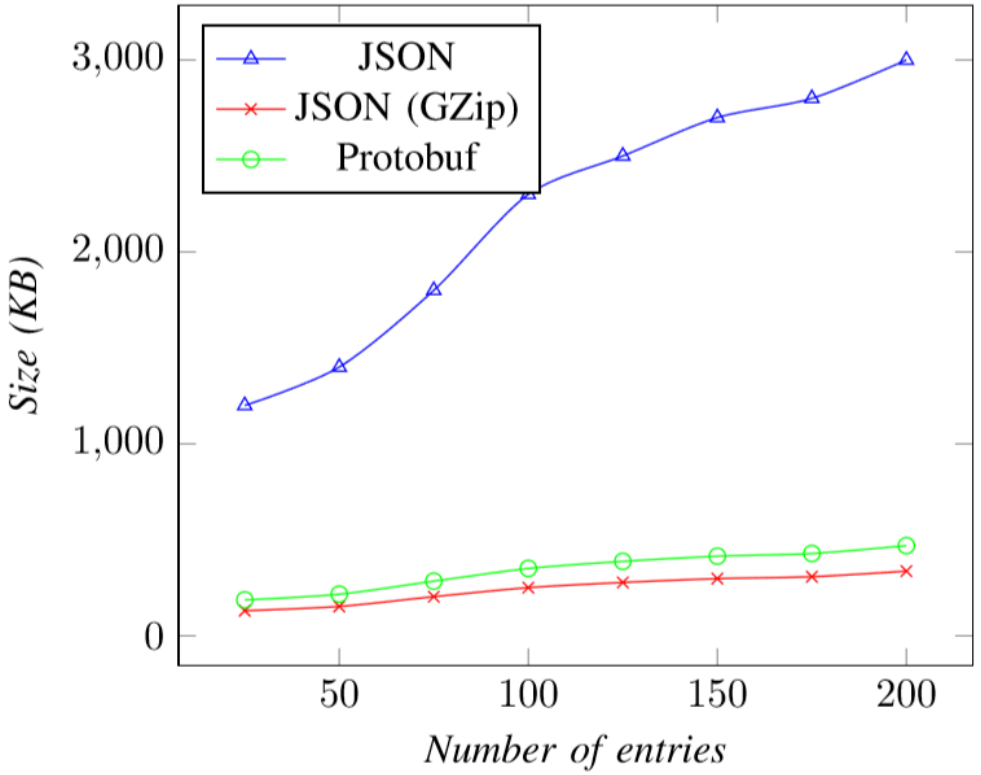
Data transferred - XY models





Data transferred - Time graph models

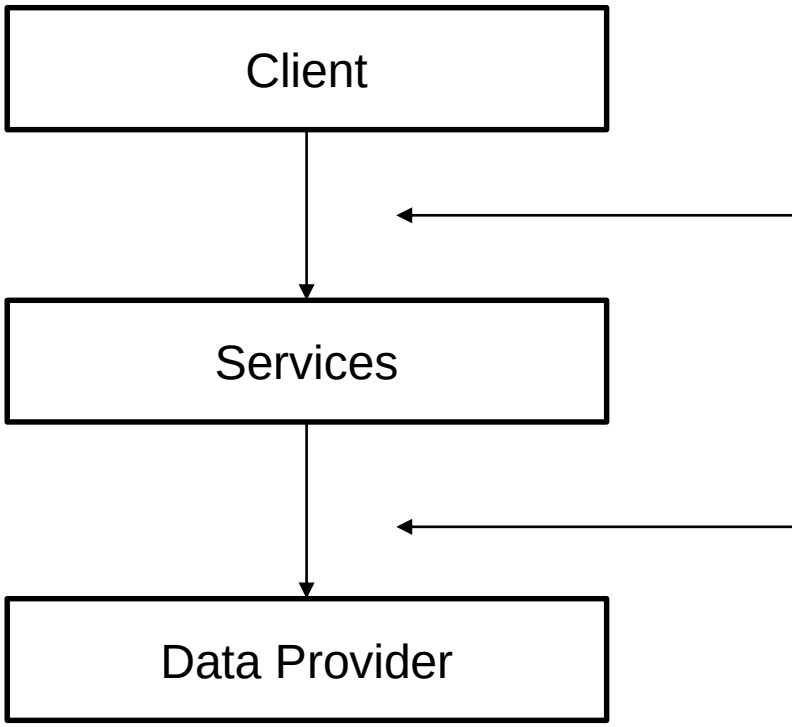




Execution time overhead

$$Overhead(ms) = T_{TCP} + T_{Headers} + T_{Serialization}$$

$$T_{Request} = Overhead(ms) + T_{Query}$$

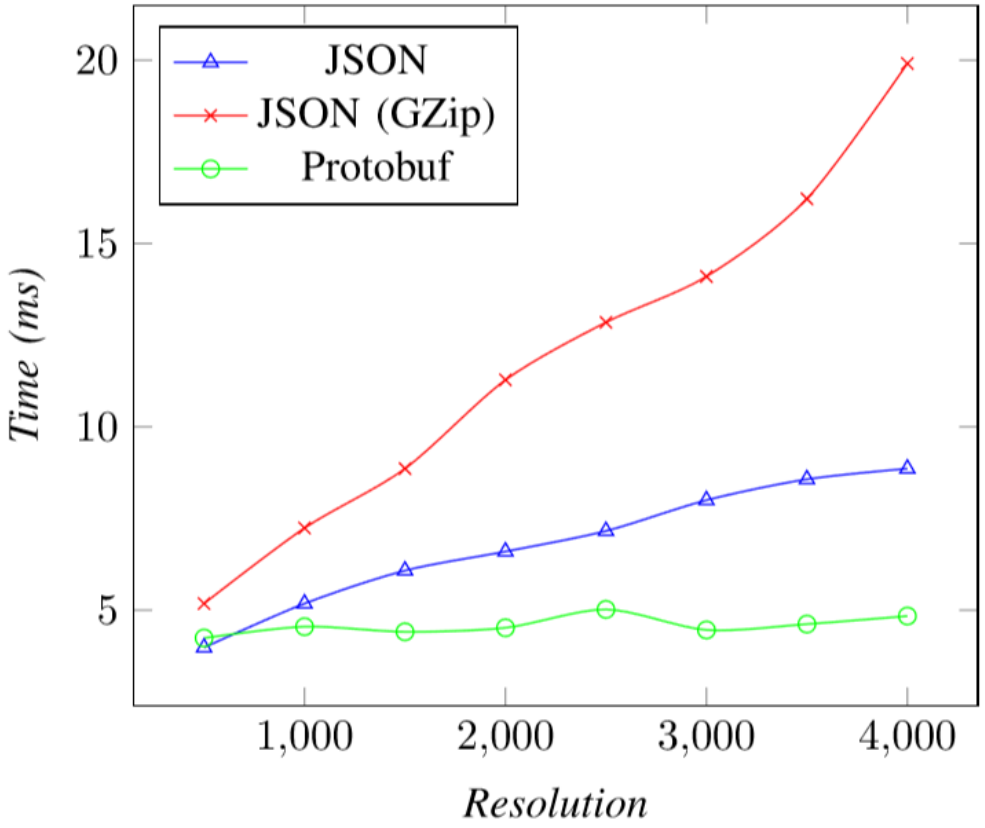


We measure the execution time for completing an HTTP request

We measure the execution time for completing a query

Execution time overhead - XY models

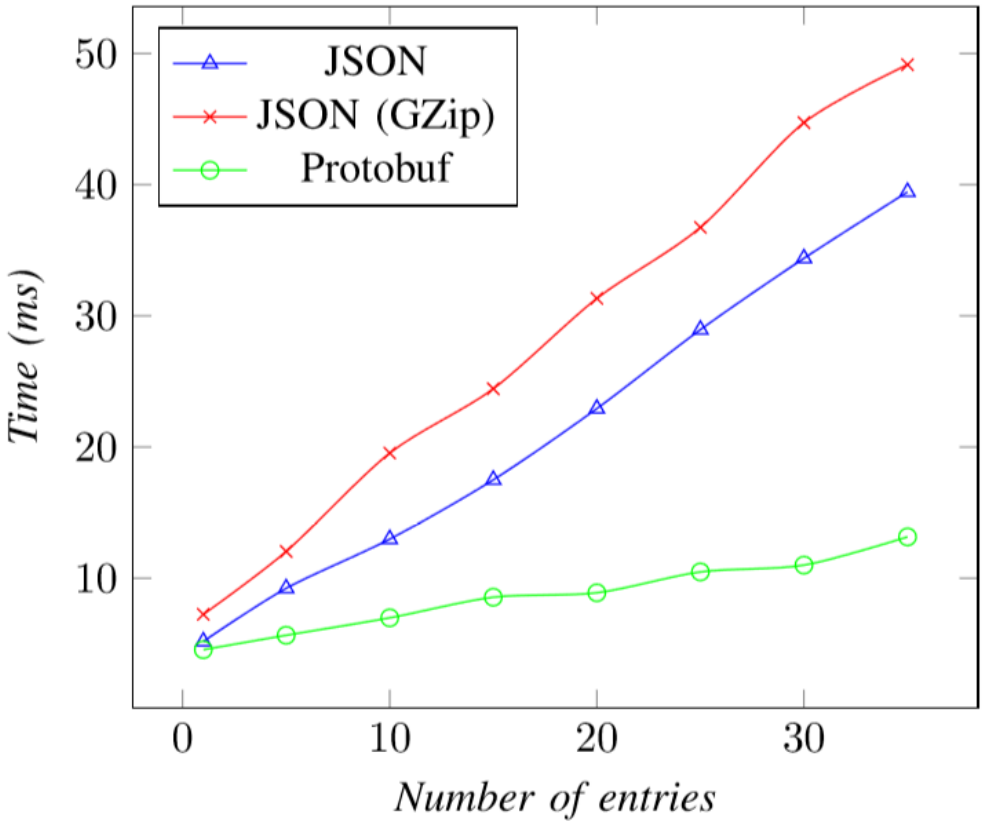
Fixed number of series, changing resolution



	Overhead (%)		
Resolution	JSON	JSON (GZip)	Protobuf
500	7.1	11.4	9.9
1000	5.2	7.8	4.8
1500	4.6	6.7	3.2
2000	3.8	6.6	2.7
2500	3.8	6.5	2.7
3000	3.7	6.2	2.0
3500	3.7	6.5	1.9
4000	3.4	7.3	1.8

Execution time overhead - XY models

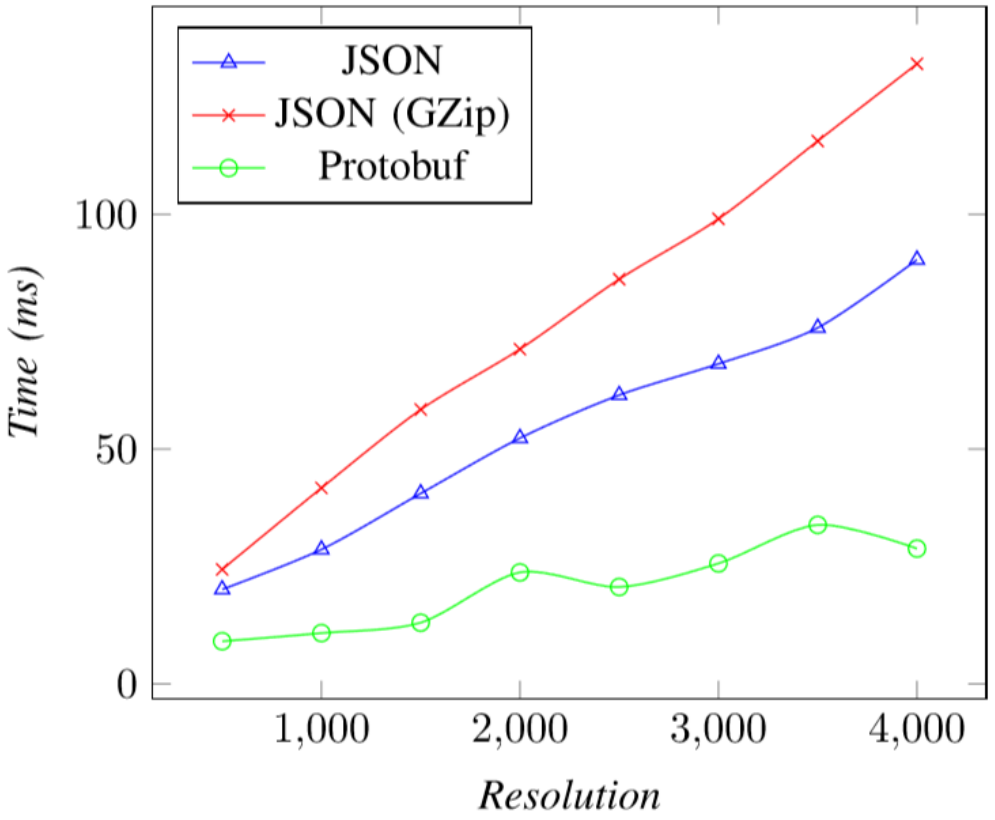
Fixed resolution, changing the number of series



	Overhead (%)		
Number of entries	JSON	JSON (GZip)	Protobuf
1	5.1	7.8	4.8
5	9.2	12.8	5.5
10	13.0	21.2	7.0
15	18.5	25.1	8.6
20	25.3	34.3	9.2
25	29.7	28.4	10.8
30	36.2	47.3	11.7
35	40.9	48.0	13.2

Execution time overhead - time graph models

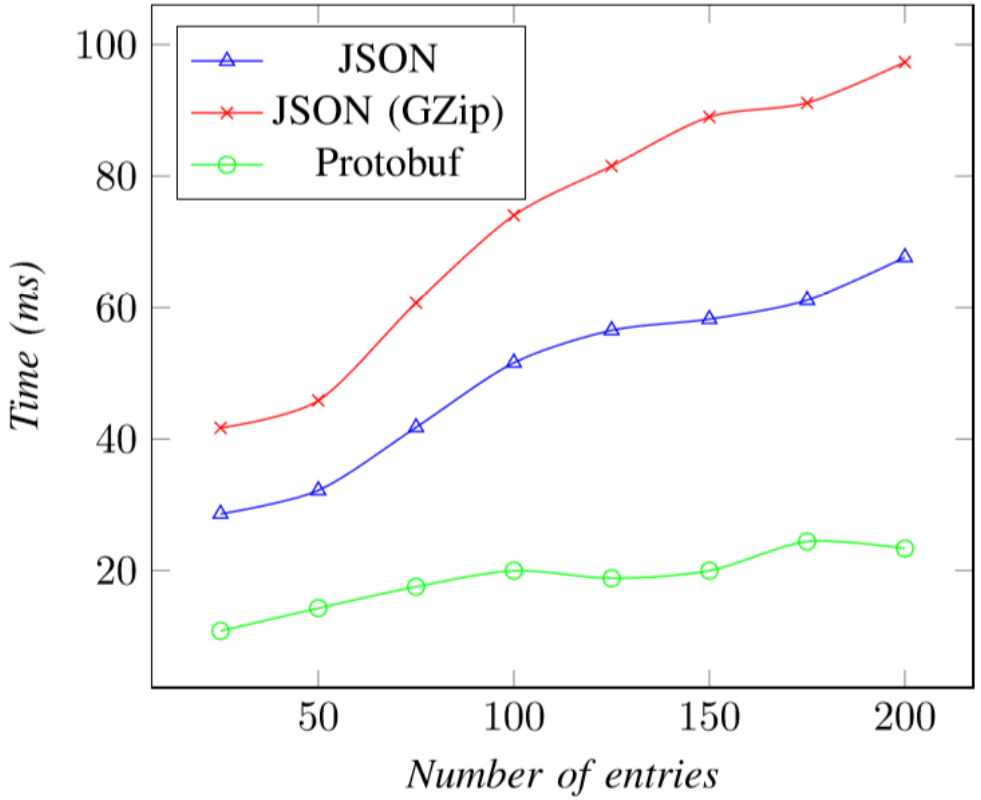
Fixed number of series, changing resolution



	Overhead (%)		
Resolution	JSON	JSON (GZip)	Protobuf
500	11.3	13.4	5.2
1000	9.6	13.3	3.5
1500	9.6	13.5	3.0
2000	9.5	12.5	4.2
2500	8.9	12.1	2.9
3000	8.0	11.4	3.0
3500	7.6	11.3	3.3
4000	7.8	11.2	2.4

Execution time overhead - XY models

Fixed resolution, changing the number of series



	Overhead (%)		
Number of entries	JSON	JSON (GZip)	Protobuf
25	9.6	13.4	3.5
50	10.6	14.7	4.6
75	13.8	19.8	5.6
100	16.7	23.7	6.4
125	18.2	25.6	6.0
150	18.5	27.8	6.2
175	19.3	27.7	7.4
200	21.3	29.4	6.8

Questions ?

yonni.scholars@gmail.com

https://github.com/cheninator/trace-scape