Real-Time Integration of Service Instances From Distributed Data Streams

Feng-Lin Li
ICT School, University of Trento, Italy
maillifenglin@gmail.com

Outline

- Motivation and Background
- Problems and Approaches
- Experiments Validation
- Conclusion and Future Work

Motivation and Background

Customer Needs

- Good service quality
- Service intelligence

Problem

- Traditional off-line analysis is not enough
- Composite structure of service

Approach

- Data stream based approach for integrating real-time data

Motivation and Background

Service

- Service is a set of capabilities as well as their functional context. The capabilities suitable for invocation are expressed via a published service contract (API)

Service Intelligence

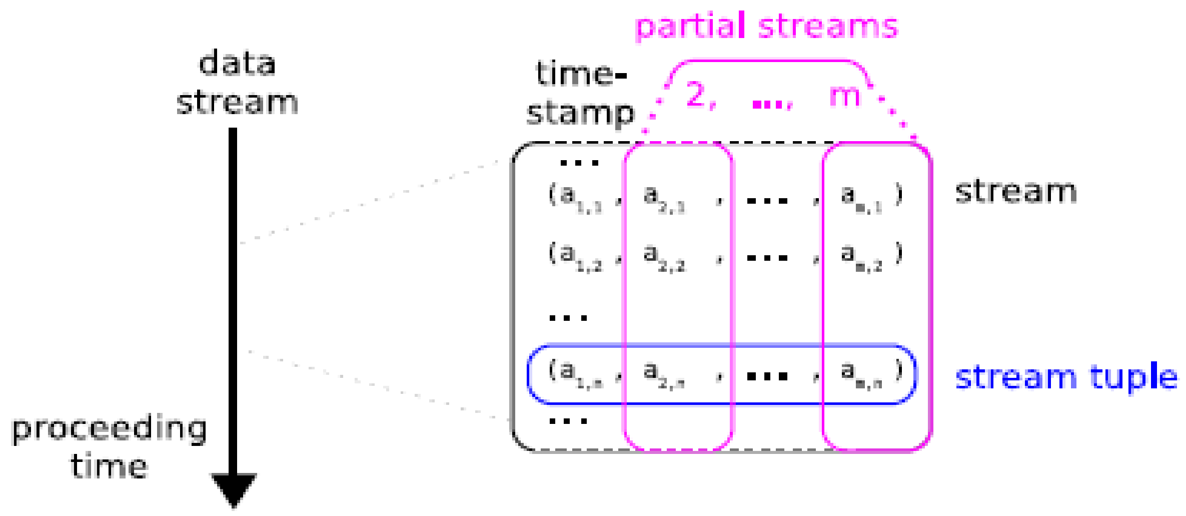
- Service Recommendation
- Service Trust
- Service Requirement
- Service Management

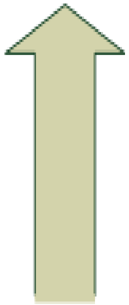
Motivation and Background

- **Service Level Agreement**
  - A set of Quality of Service (QoS) a provider guarantees
    - **Response time**
    - Through output
    - Delay
    - Usage
    - Pricing
- **Compliance**
  - The conformity degree between **runtime measurements** and **guaranteed quality** on those indicators in the SLA

Data Stream

- Schema
- A sequence of infinite data tuples
- Continuous query

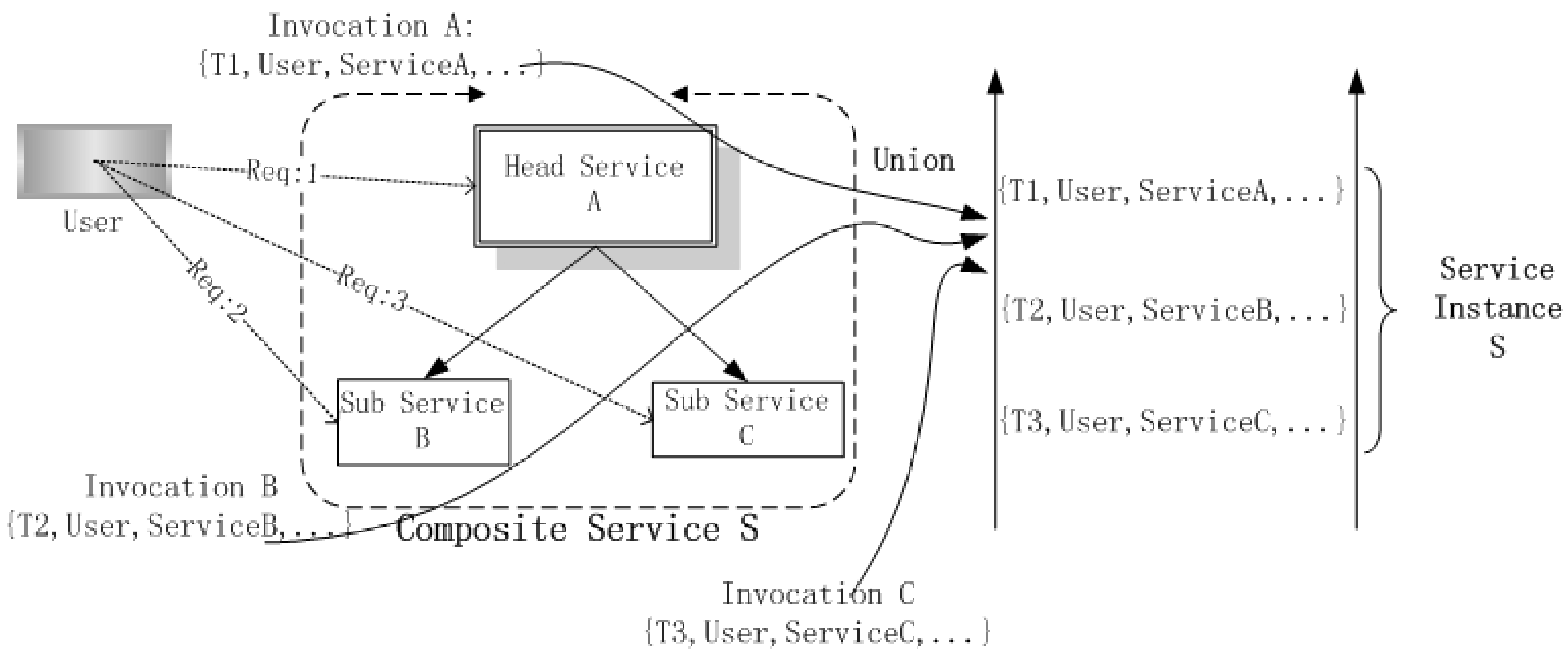




Data Stream

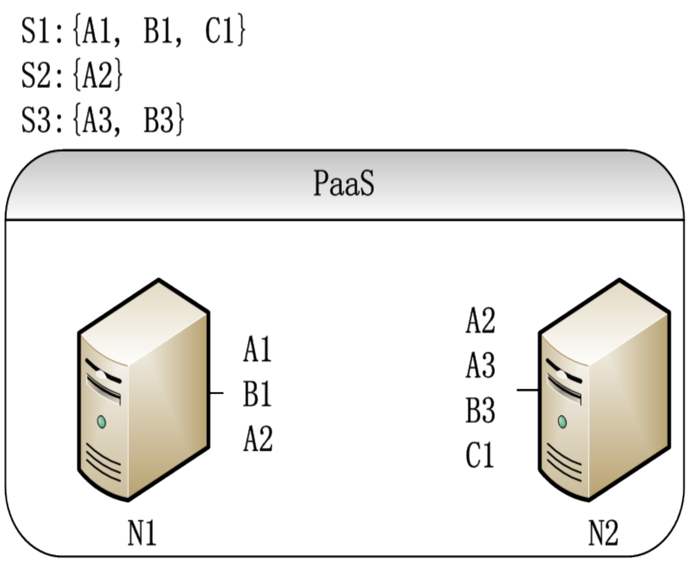
Composite Structure of Service

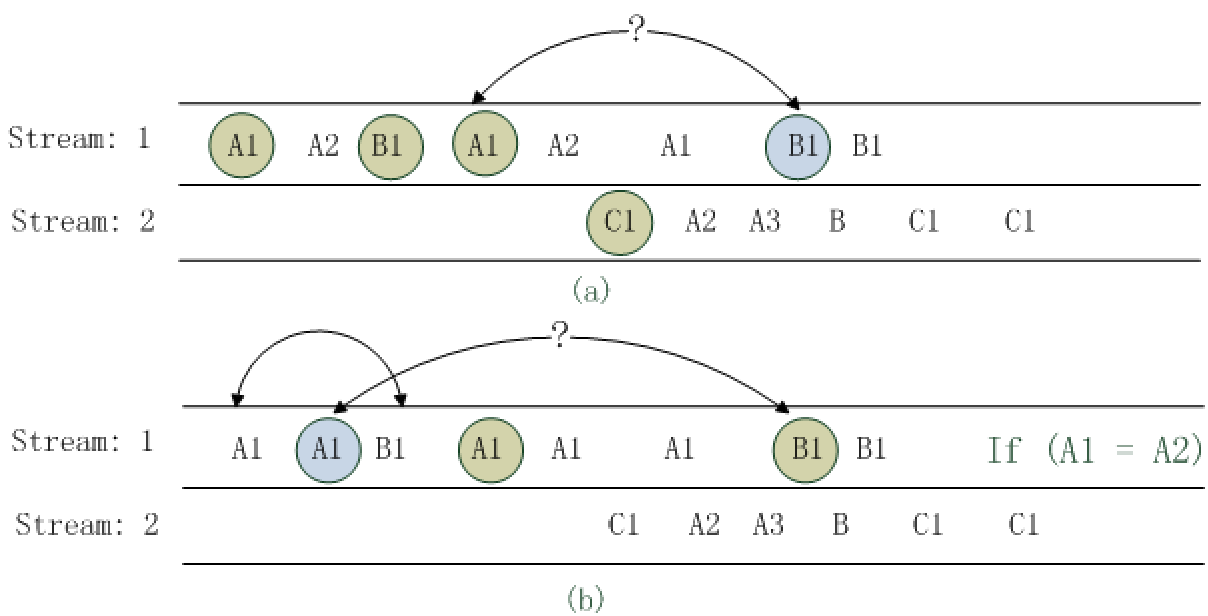
- Atomic and Composite Service
- Invocation Tuples and Service Instance
- Service Degree



Associating Distributed Data Tuples

- Supposing that a set of services : {S1, S2, S3}, S1 includes A1, B1 and C1; S2 includes only A2 ; S3 includes A3 and B3
- Each Service (S1, S2, S3) could be requested multiple times
- A component service (A1, B1 … B3) could belong to multi-service





Problems and Approaches

Association Strategies

- Service Structure + Client Information (IP)
- Service Structure + Runtime Status Information (Timestamp)
- Service Structure + Client Information (IP) + Runtime Status Information (Timestamp)

Requested Service	Head Service	Invocation Timestamp	Client Information	Other Information
B	A	2010-8-9 17:11:29	192.168.10.28	…
B	A	2010-8-9 17:11:37	192.168.10.111	…
B	A	2010-8-9 17:11:29	192.168.10.111	…

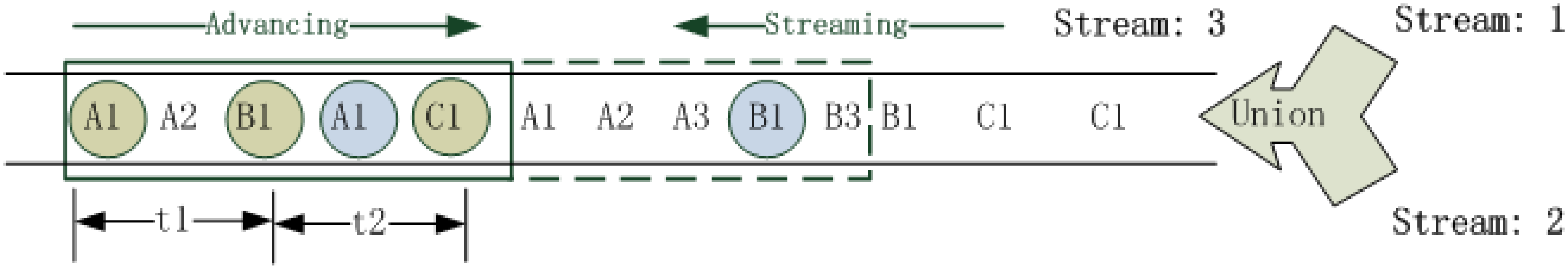
Quantitative Analysis

Strategies	Head + IP +Time	Head	Head + Time	Head + IP
Recall	100%	91.3053%	99.3141%	99.9857%
Accuracy	100%	85.1683%	98.6283%	99.9786%
Error rate	0%	6.1370%	0.6859%	7.14e-3%

Problems and Approaches

Integration Completeness

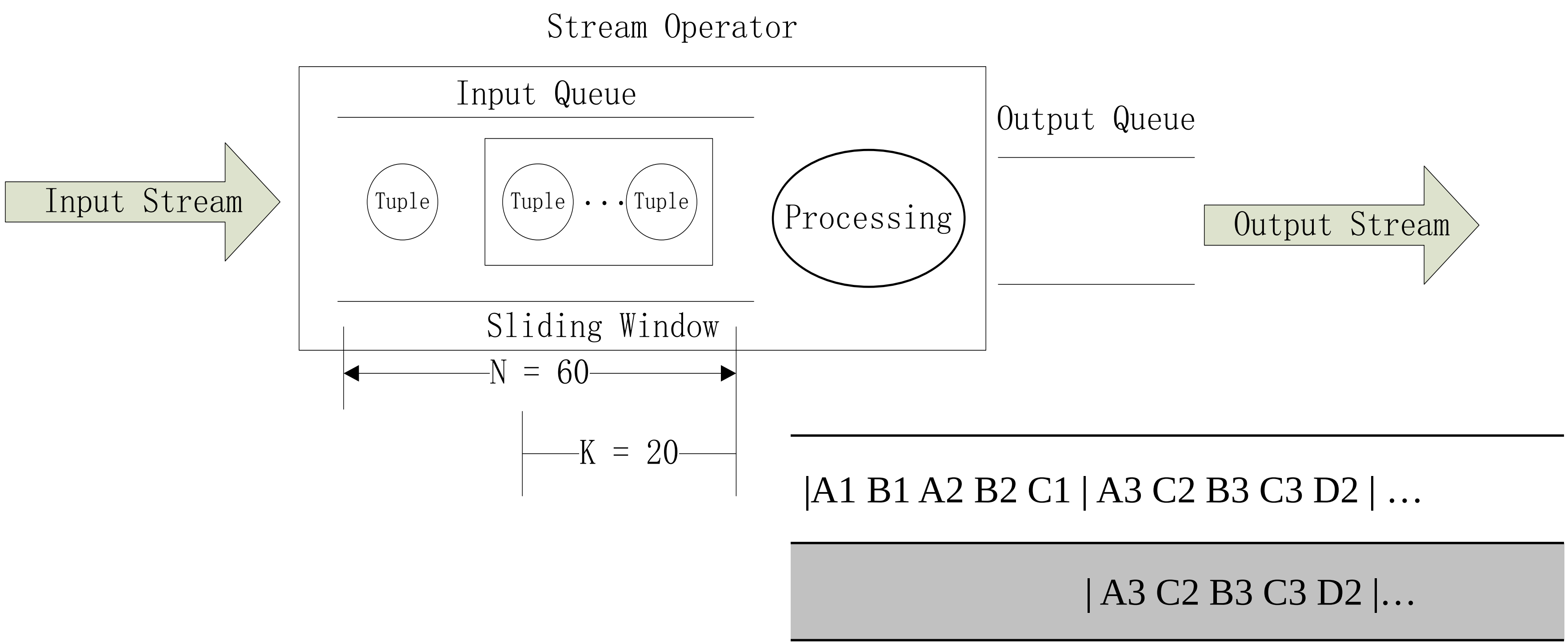
- To integrate service instance of S1, we need to collect A1, B1, C1
- For counting window, if the window size is 5, A1 will get lost; if the size if 6 or other number, it will lose other tuples



Problems and Approaches

Traditional Sliding Window

- Partition on data streams and store in the buffer
- If the buffer full, stream operator will perform grouping according to their key and compute respectively



Problems and Approaches

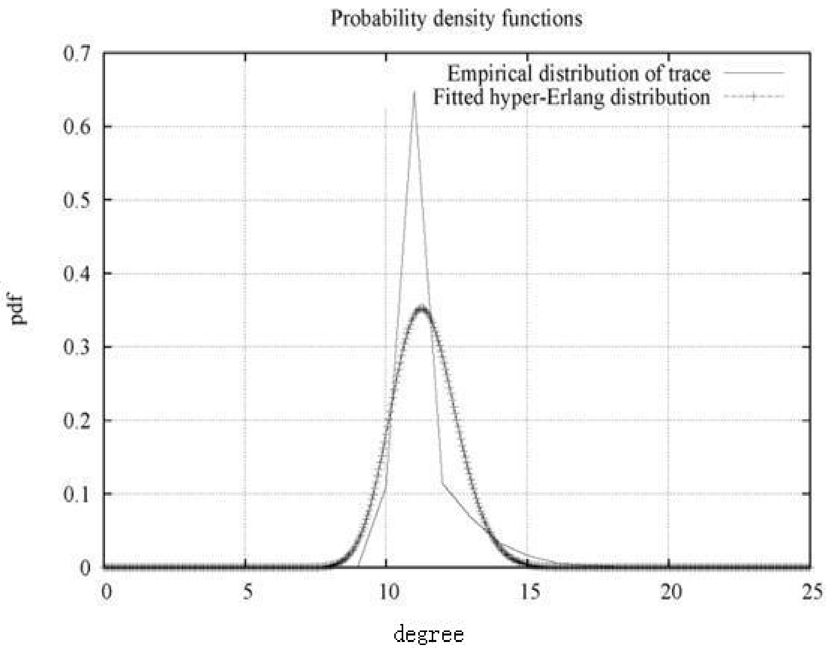
Small Window Array

- A mapping: MAP <Key, Small-Window>
- A tuple for one invocation and one small window for a service instance
- For a key in the tuple, if there is no such key in the system, it will open a new window for the key
- If the key exists in the system, the tuple will be inserted into the window
- A small window will close when it is full or timeout happens

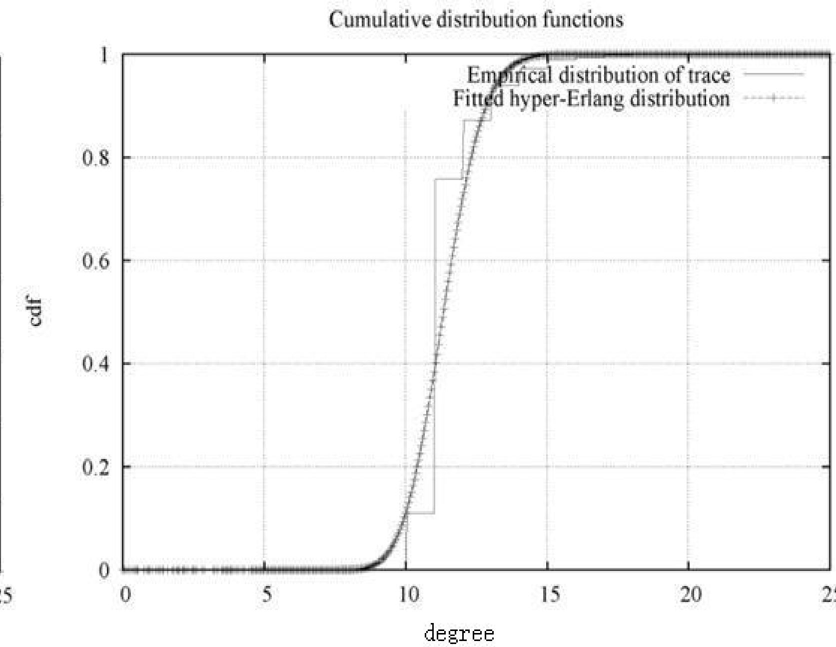
A1 B1 A2 B2 C1 A3 C2 B3 C3 D2 …
|A1 B1 C1 |…
|A2 B2 C2 D2 | …
|A3 B3 C3 | …

The Size of Counting Window

- Service degree's Probability Distribution Function *f*(n) and Cumulative Distribution Function *F*(n)



(a)



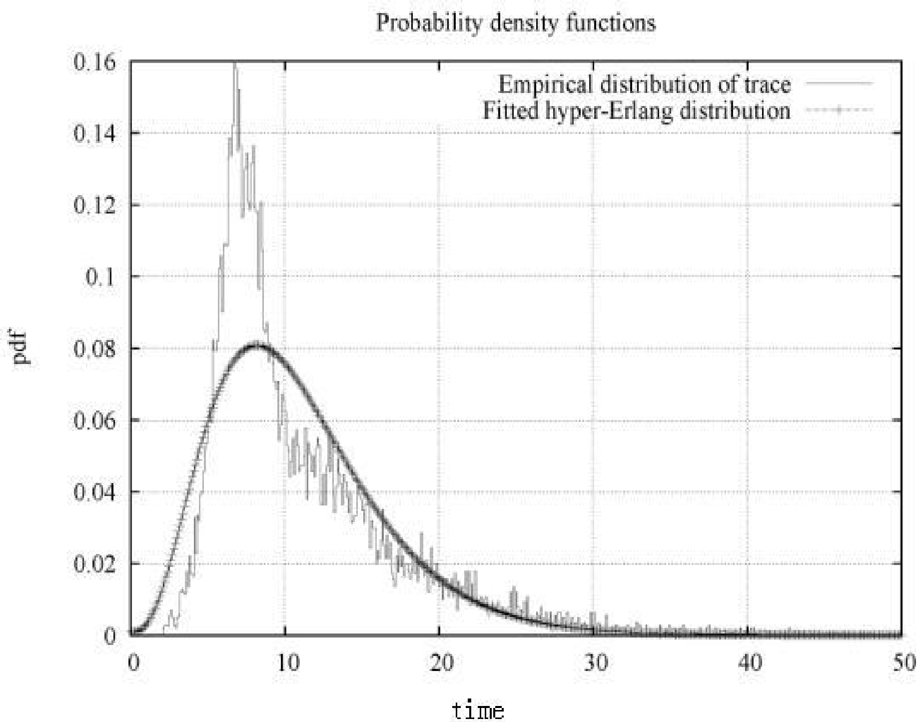
(b)

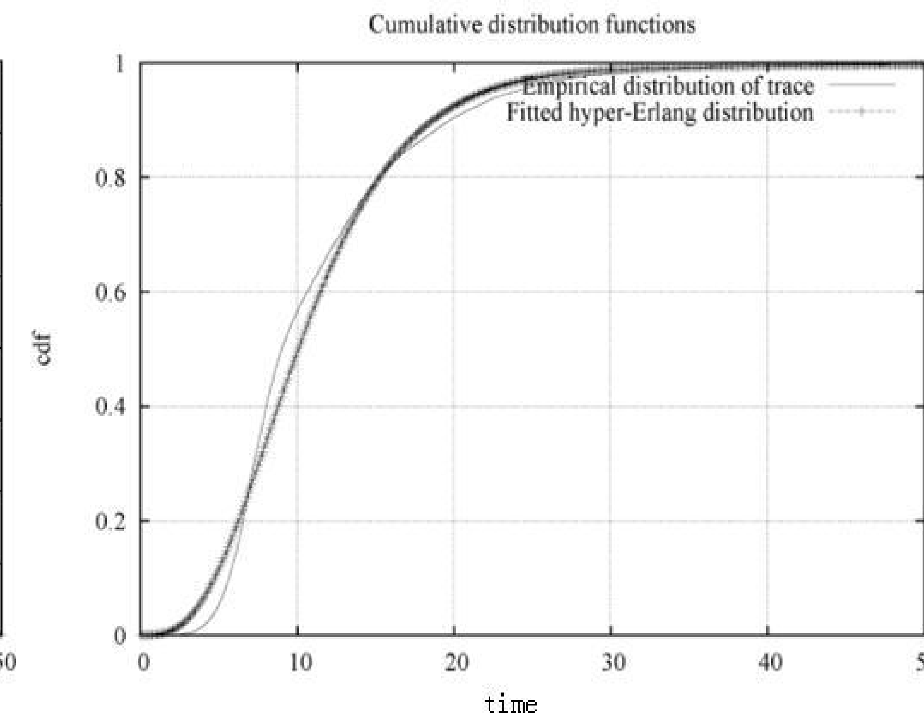
$$f(x;k,\lambda)=\frac{8.7963^{100}x^{99}e^{-8.7963x}}{99!}=(2.8840E-62)x^{99}e^{-8.7963x}$$

$$F(x;k,\lambda)=1-\sum_{n=0}^{99}e^{-8.7963x}(8.7963x)^{n}/n!$$

Timeout Settings

- Service response time distribution PDF *f*(t) and CDF *F*(t)





(a) (b)

$$f_{HErD}\left(x;k,\lambda\right)=0.0009979e^{-0.0404x}+0.0029x^{3}e^{-0.3666x}$$

$$F_{HErD}\left(x;k,\lambda\right)=1-\left(0.0247e^{-0.0404x}+0.9753\sum_{n=0}^{3}e^{-0.3666x}\left(0.3666x\right)^{n}/n!\right)$$

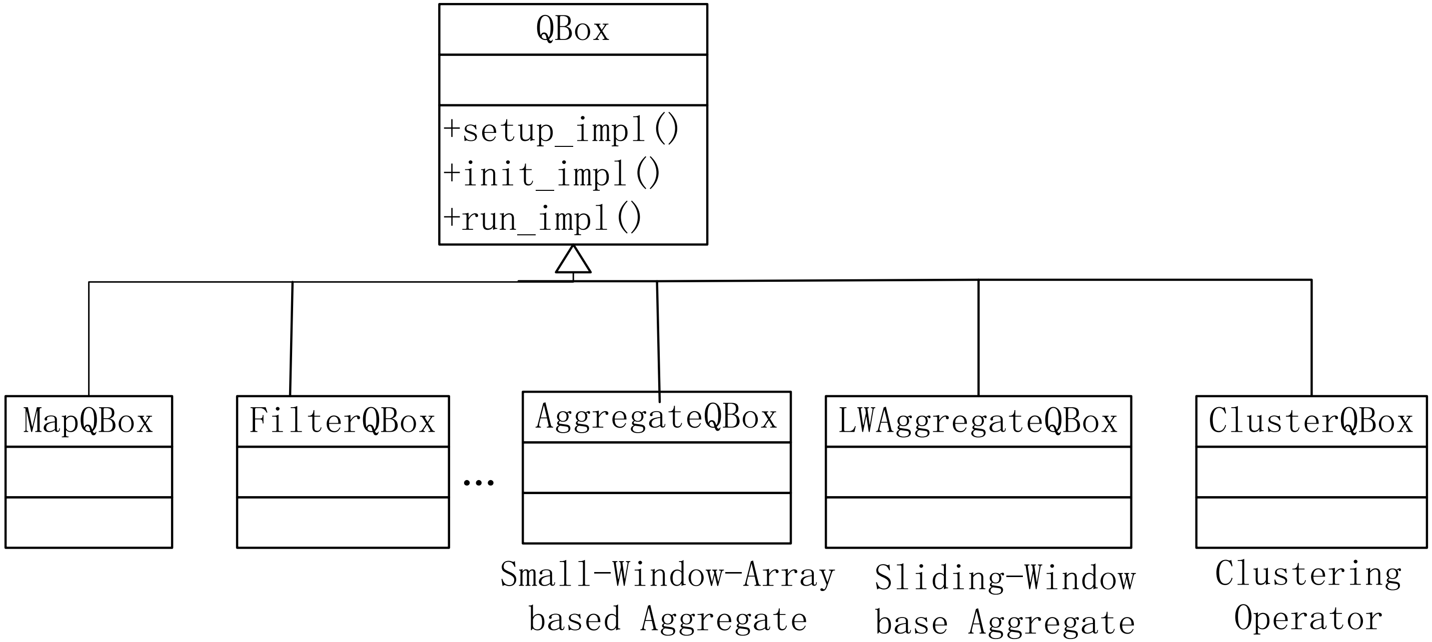
Comparison of Window Mechanism

- Window Size / Advance Step / Timeout

	Small Window Array	Sliding Window		
Parameters	13/13/22	4000/4000	8000/8000	16000/16000
Instance Completeness (=1)	80.4815%	1.66464%	13.1457%	43.4379%
(r = 0.75)	91.6553%	2.8363%	16.9894%	47.6316%
(r = 0.85)	90.2979%	2.2719%	15.1675%	45.8741%
Invocation Completeness	94.0649%	22.0332%	40.1617%	63.1543%

Experiments Validation

- **Theoretical Foundation**
  - Queuing Theory Model
- **Experiment Platform**
  - BOREALIS, a data stream management system by Brandies, Brown and MIT



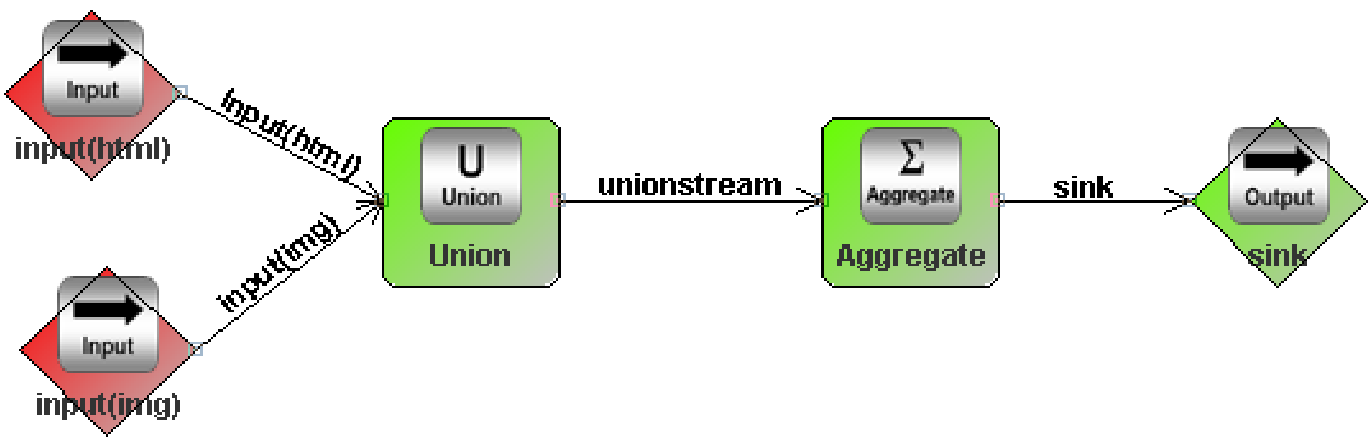
Experiments Validation

Streaming Processing Strategy

- Join ➔ Aggregate (Expensive and Inefficient)
- Union ➔ Aggregate

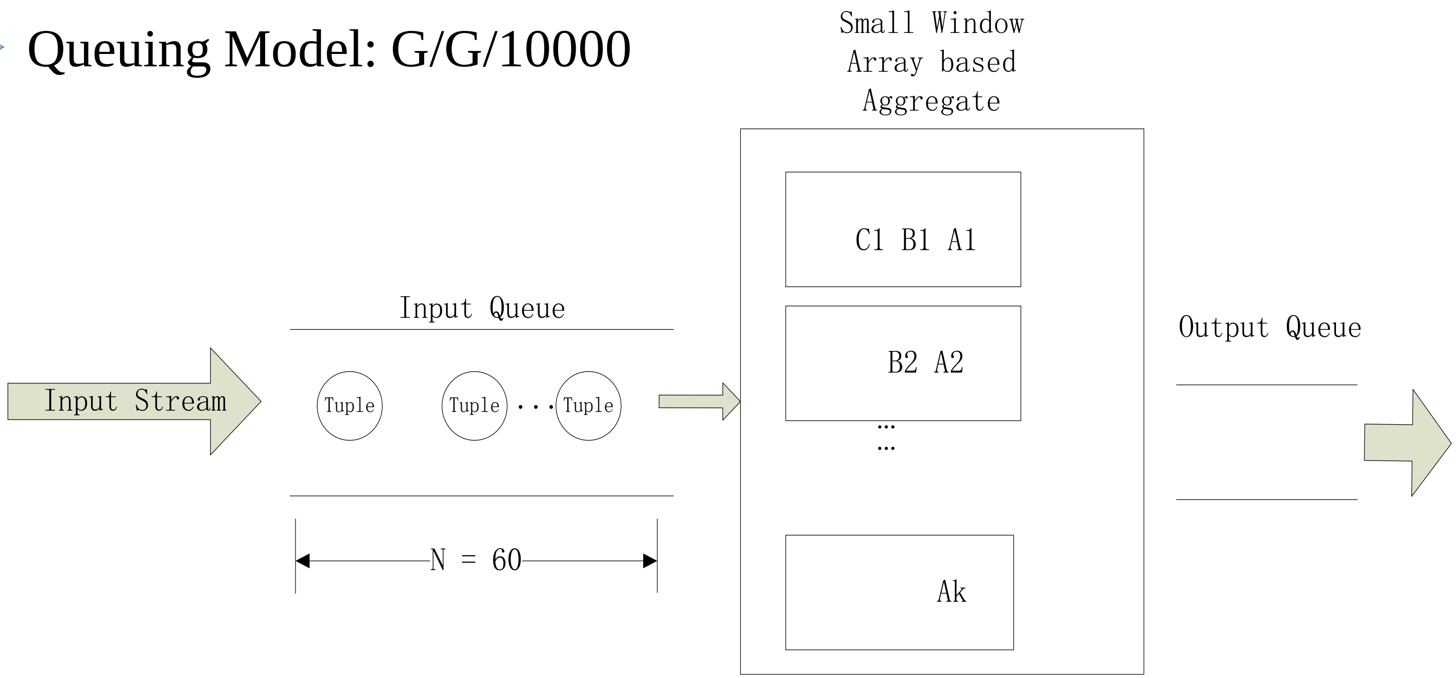
Experiment Data

- A web page is similar with a composite service
- English Wiki / Amazon access log



Small Window Array based Aggregate

- Input: a 2-stage PH distribution with average interval 9.7780 ms
- Service time is a general distribution with average of *avg* = 18.9749s and stand deviation of *delta* = 5.5068
- Queuing Model: G/G/10000



Experiments Validation

Performance Analysis

- The deviation of processing time is 9.04%, that of tuples number is 12.0251%
- Using the input distribution observed from the receiver, the deviation will be 9.04% and 5.6989% respectively

Prediction:			
Total Tuples	2119.2972	Waiting Tuples	3.1560E-53
Residence Time	20.7137	Waiting Time	0
Probability of Wait	7.9272E-05	Probability of Being Dropped	0
Observation:			
Total Tuples	1860.6358	Deviation	12.0251%
Residence Time	19.0129	Deviation	9.0402%

Experiments Validation

Comparison between the two Window Mechanisms

- Window Size / Advance Step / Timeout
- Service Structure + Client Information (IP) + Instance Status (Timestamp)

Indicators	Small Window Array	Sliding Widnow
Window Parameters	13/13/22	32000/32000/-
Completeness = 1	80.4815%	72.1083%
Completeness = 0.85	90.2979%	73.7587%
Completeness = 0.75	91.6553%	74.7302%
0 < Completeness ≤ 1	94.0649%	76.2744%
Average Tuple Numbers	1861(window)	15374(tuple)
Average Storage	3.1148MB	5.2909MB
Average Residence Time	19.0192(S)	16.8402(S)

Conclusion and Future Work

Conclusion

- Accuracy: systematic association strategies
- Completeness: small-window-array, statistical distribution
- Performance Evaluation: queuing theory model

Future Work

- Tradeoff between the completeness and the performance
- Queuing Model for Join operation
- Queuing Network (e.g. Jackson network) on Performance analysis (cost model) for Streaming Operator
- Data Stream Algorithm: clustering

Thanks ! ☺