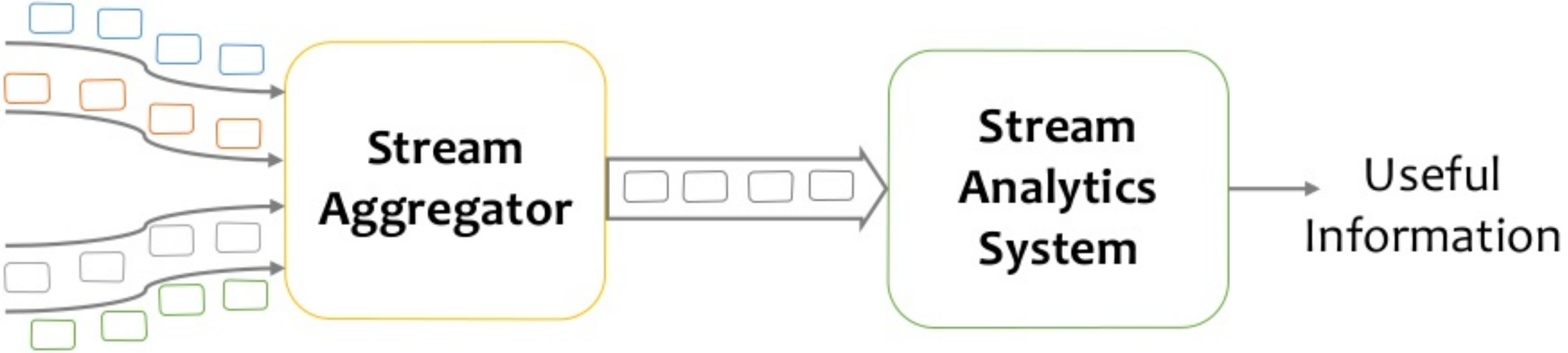
StreamApprox

Approximate Stream Analytics in Apache Flink

Do Le Quoc, Pramod Bhatotia,
Ruichuan Chen, Christof Fetzer, Volker Hilt, Thorsten Strufe

Sep 2017

Modern online services



Modern online services

Approximate computing

Tension

Low latency

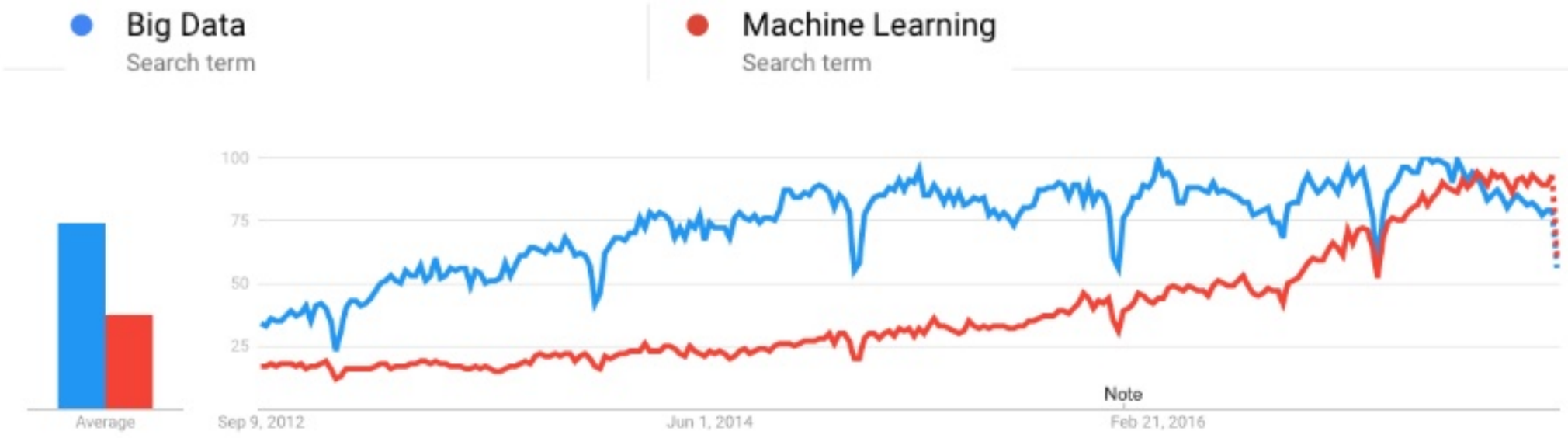
Efficient resource utilization

Approximate Computing

Many applications:
Approximate output is good enough!

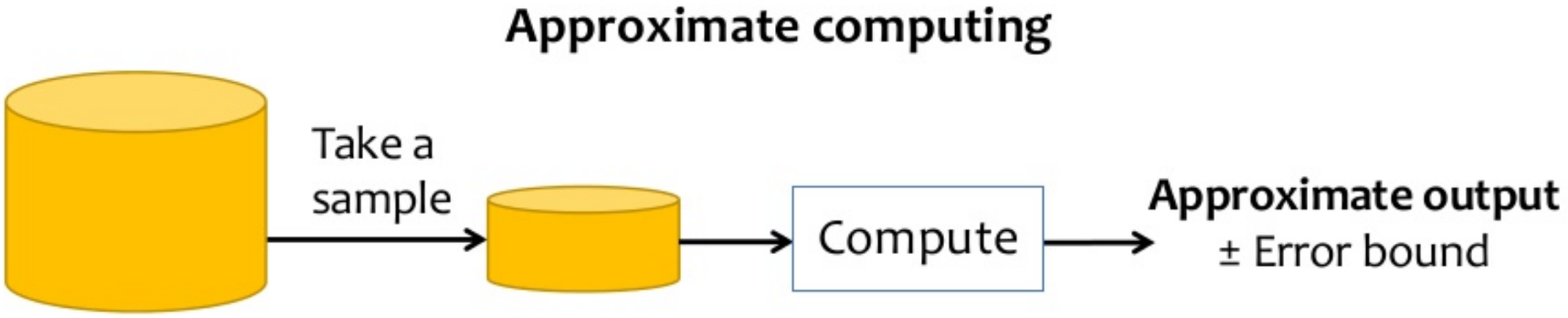
The trend of data is more important than the precise numbers

E.g. : Google Trends -- Big Data vs Machine Learning (Sep/2012 – Sep/2017)



Approximate Computing

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



State-of-the-art systems

BlinkDB [EuroSyS'13]

Using pre-existing samples

Not designed for stream analytics

ApproxHadoop [ASPLOS'15]

Using multi-stage sampling

Quickr [SIGMOD'16]

Injecting samplers into query plan

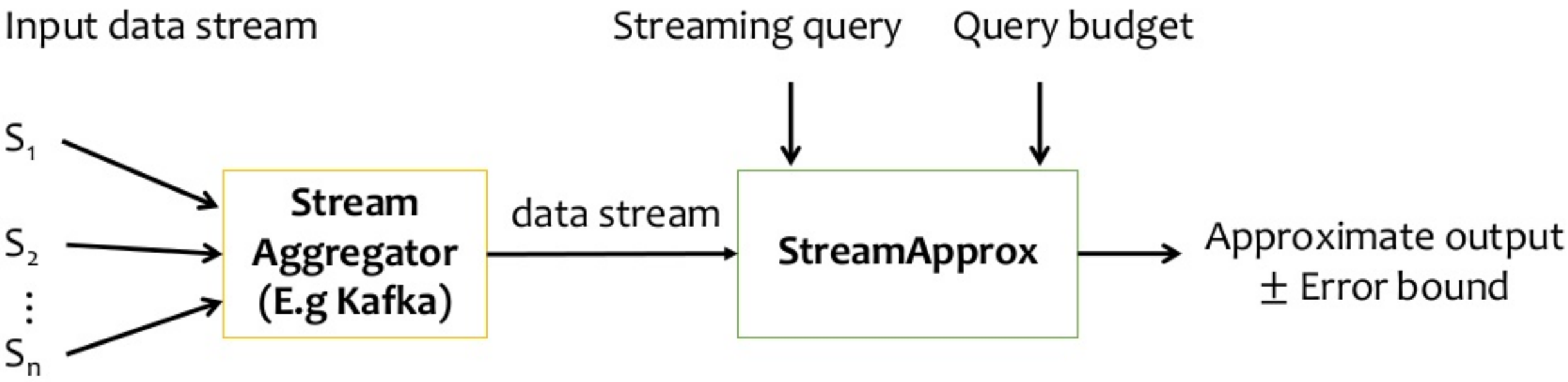
StreamApprox: Design goals

Transparent	Targets existing applications w/ minor code changes
Practical	Supports adaptive execution based on query budget
Efficient	Employs online sampling techniques

Outline

- ~~Motivation~~
- Design
- Evaluation

StreamApprox: Overview

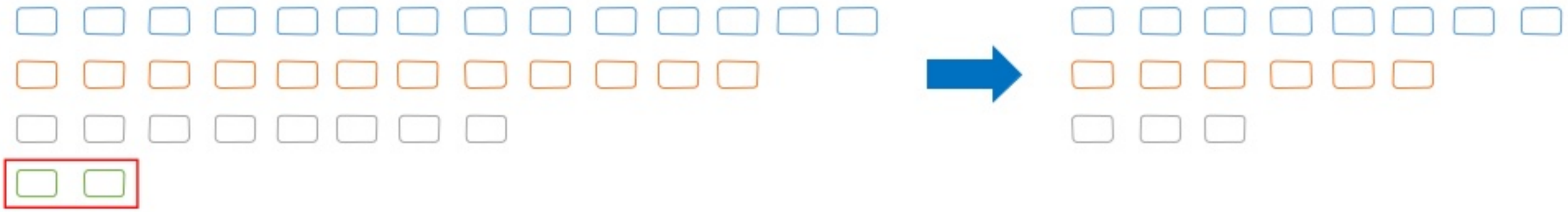


Query budget:

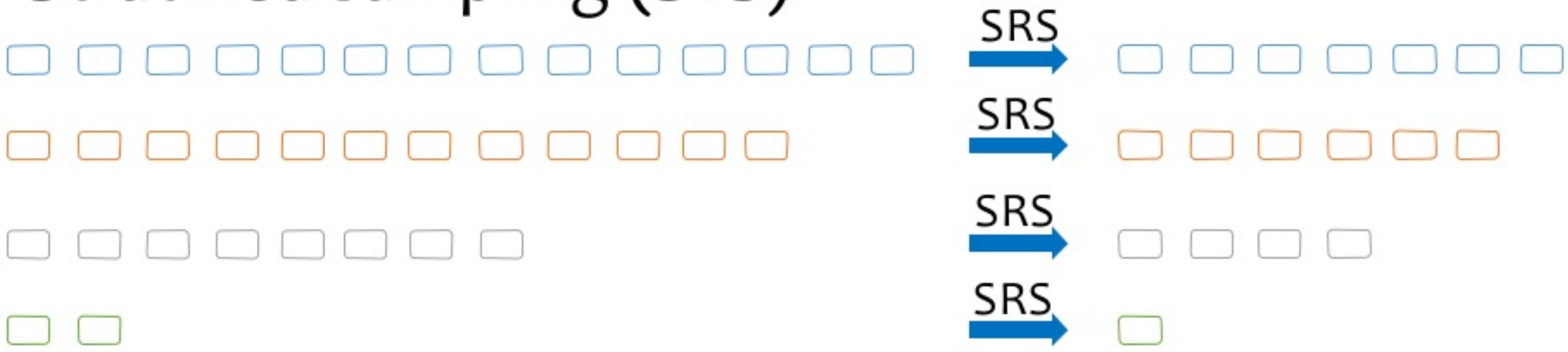
- **Latency/throughput** guarantees
- Desired **computing resources** for query processing
- Desired accuracy

Key idea: Sampling

Simple random sampling (SRS):

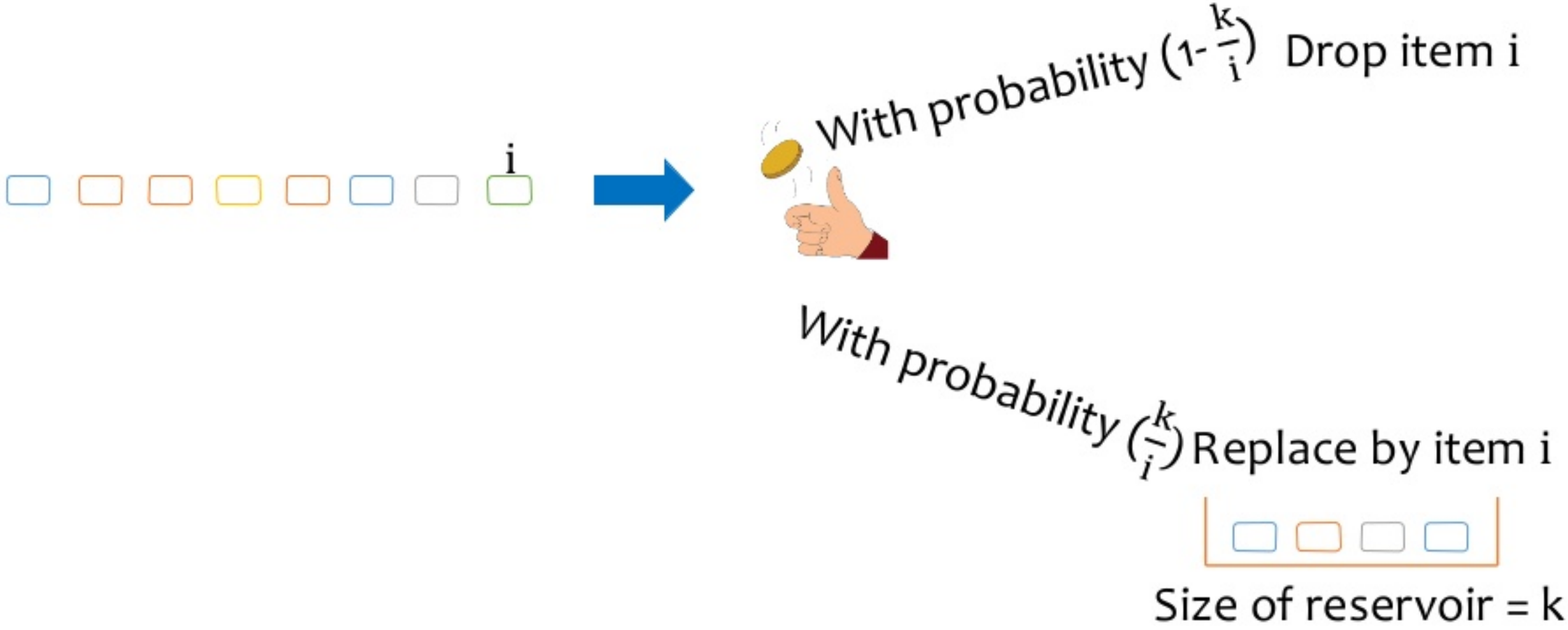


Stratified sampling (STS):



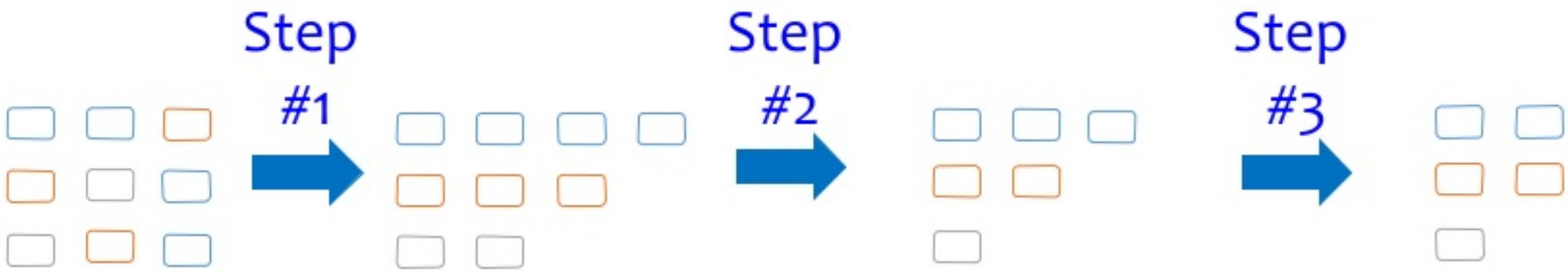
Key idea: Sampling

Reservoir sampling (RS):



Spark-based Sampling

Spark-based Stratified Sampling (Spark-based STS)



Create strata using *groupByKey()*

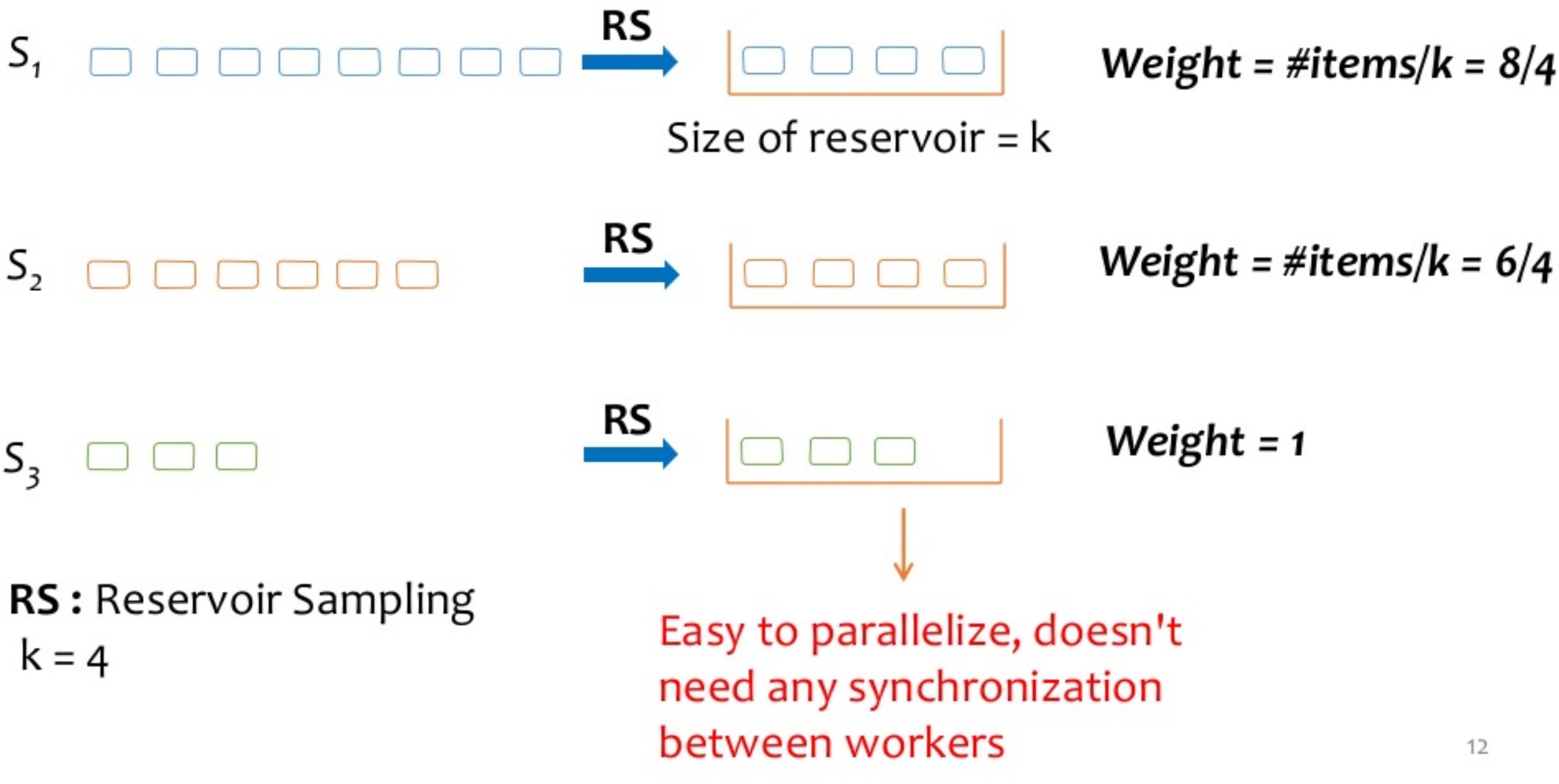
Apply SRS to each stratum S_i

Synchronize between worker nodes to select a sample of size *k*

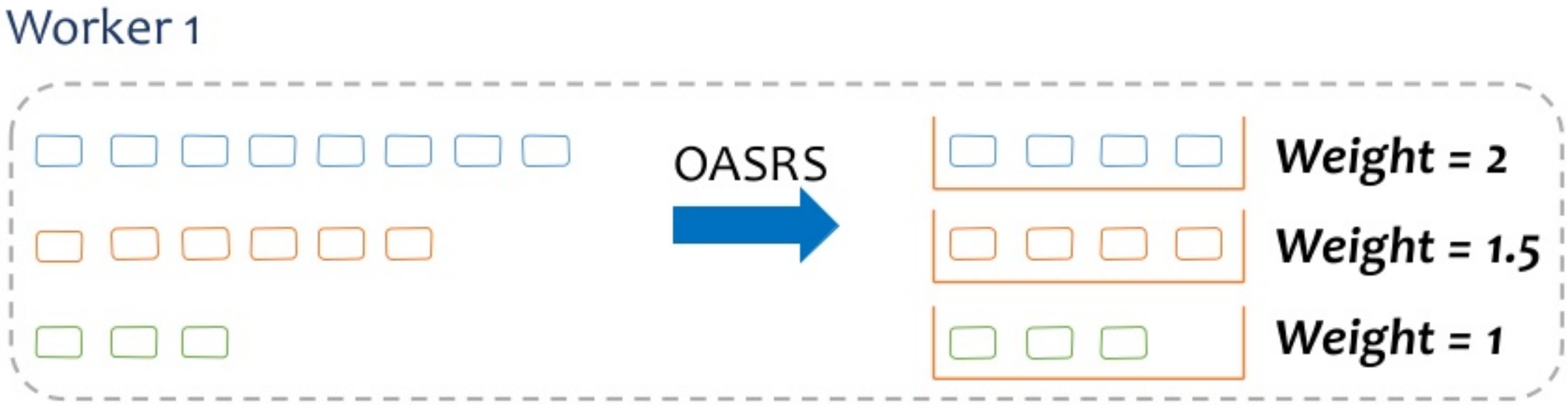
These steps are very expensive

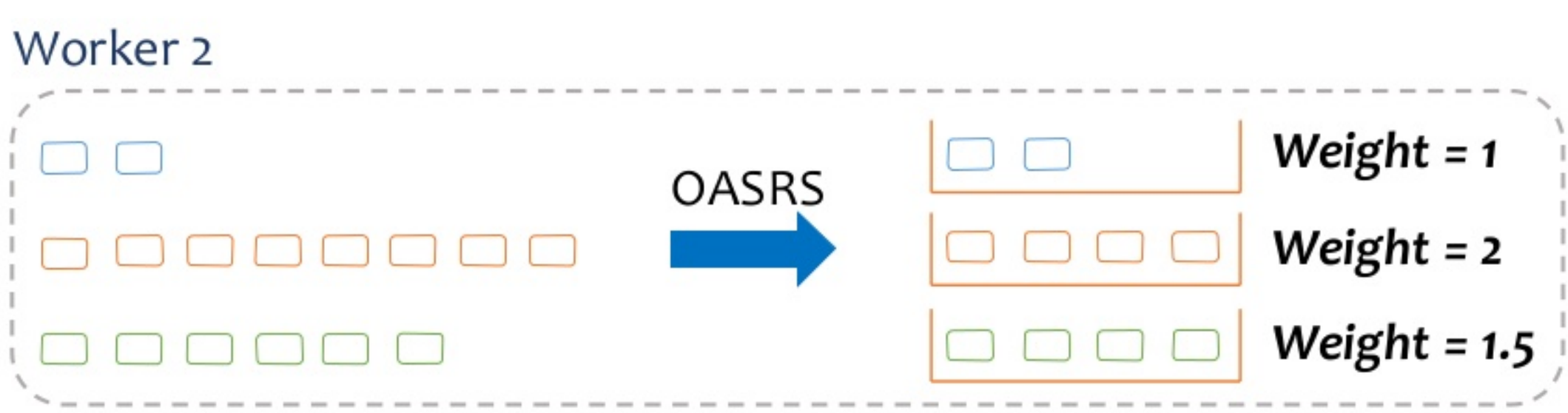
StreamApprox: Core idea

Online Adaptive Stratified Reservoir Sampling (OASRS)



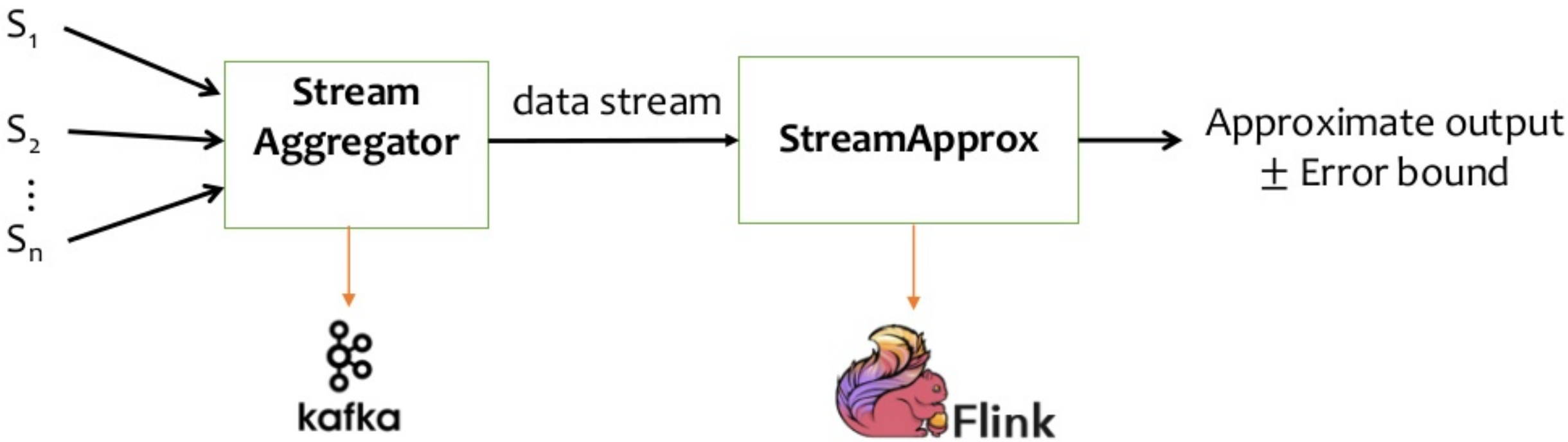
StreamApprox: Core idea



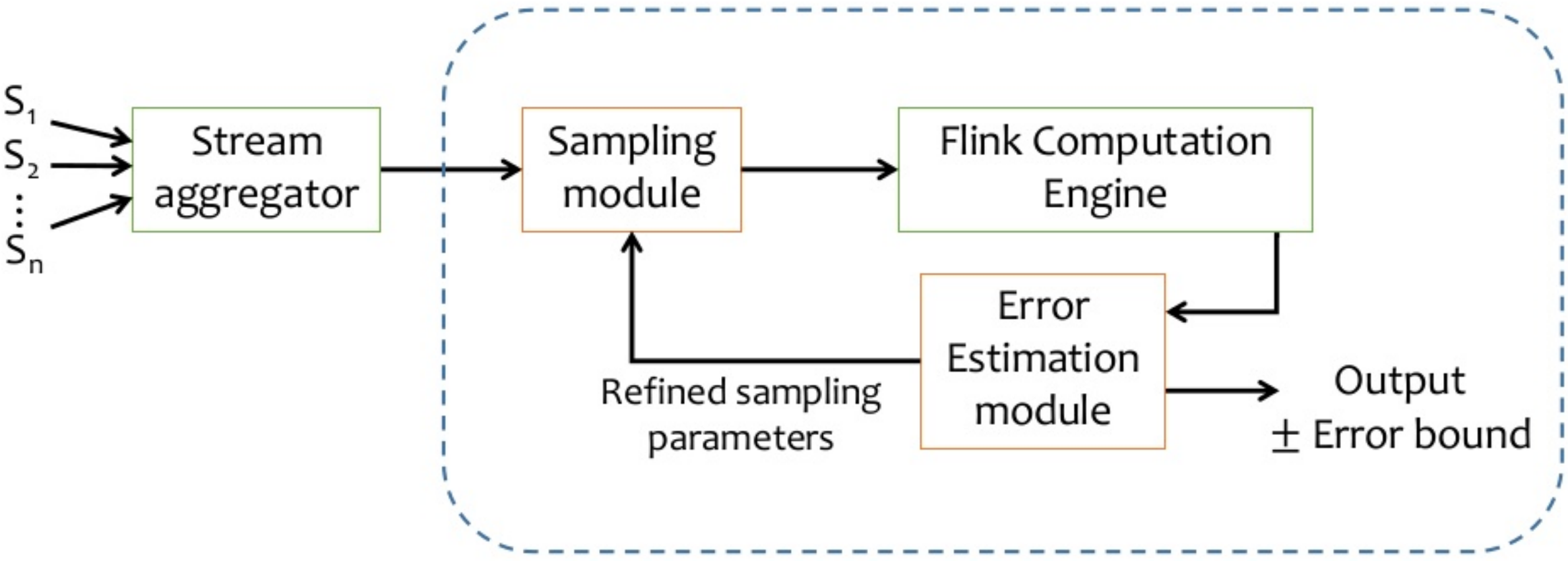


Size of reservoir = 4

Implementation



Implementation



Flink-based StreamApprox

Outline

- ~~Motivation~~
- ~~Design~~
- Evaluation

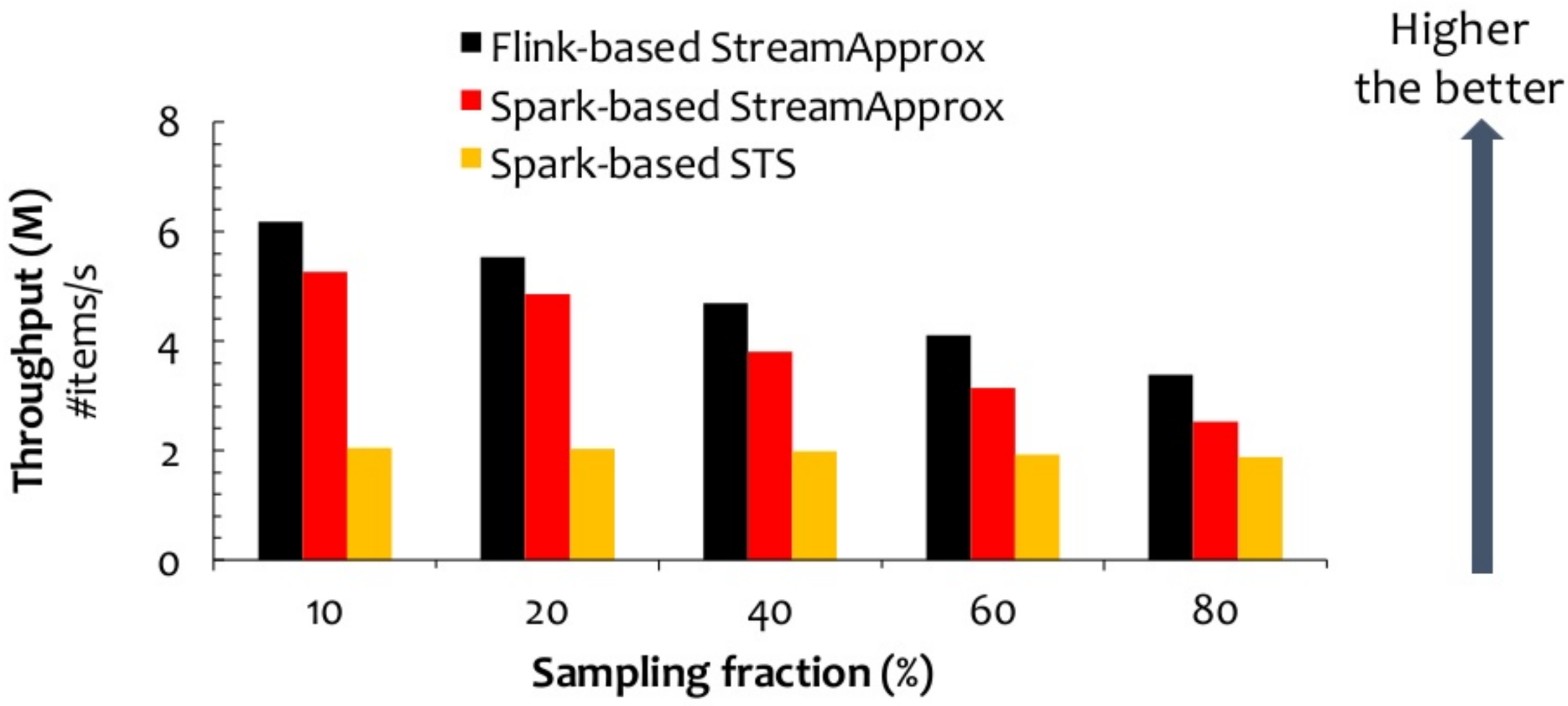
Experimental setup

- Evaluation questions
  - Throughput vs sample size
  - Throughput vs accuracy

See the paper for more results!

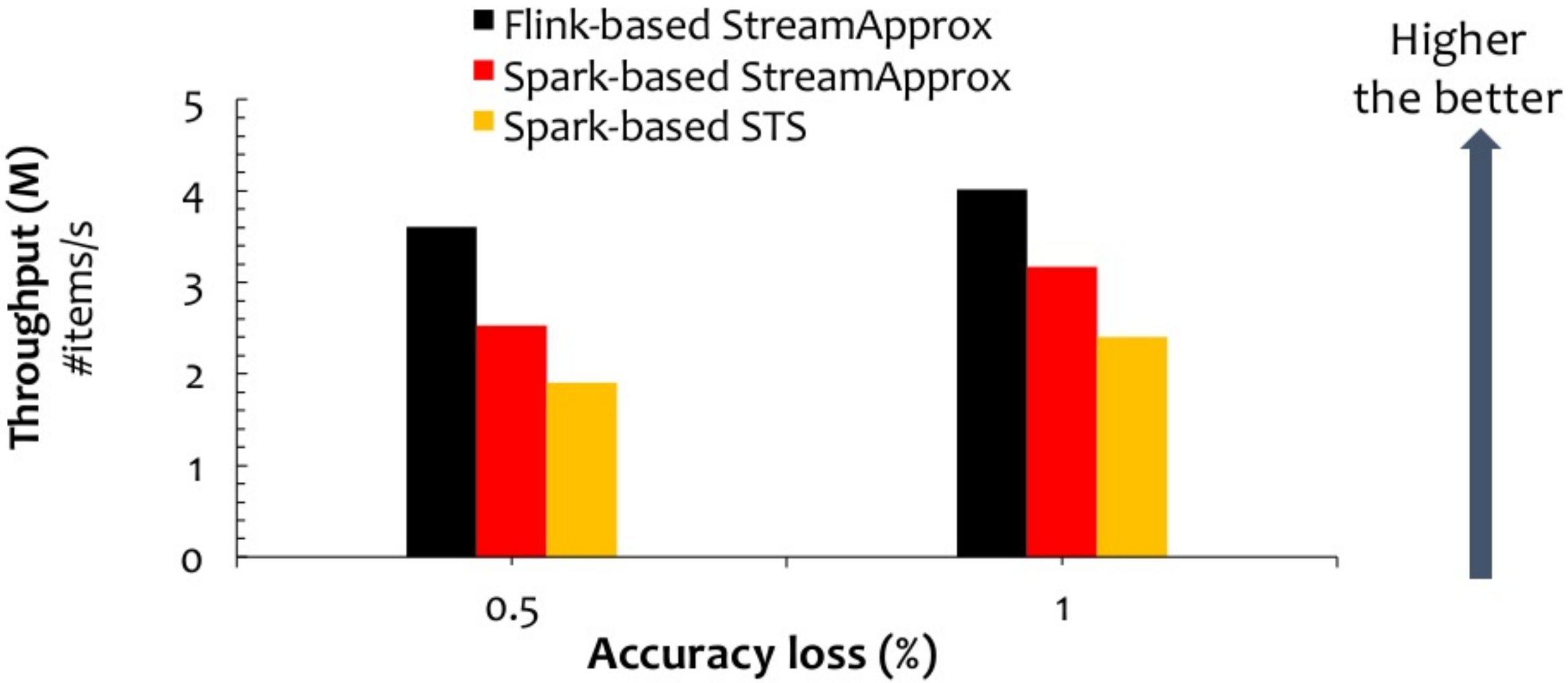
- Testbed
  - Cluster: 17 nodes
  - Datasets:
    - Synthesis: Gaussian distribution, Poisson distribution datasets
    - CAIDA Network traffic traces; NYC Taxi ride records

Throughput



Spark-based StreamApprox: ~2X higher throughput over Spark-based STS
Flink-based StreamApprox: 1.3X higher throughput over Spark-based StreamApprox
With sampling fraction < 60%

Throughput vs Accuracy



Spark-based StreamApprox: ~1.32X higher throughput over Spark-based STS
Flink-based StreamApprox: 1.62X higher throughput over Spark-based StreamApprox
With the same accuracy loss

Conclusion

StreamApprox: Approximate computing for stream analytics

Transparent	Supports applications w/ minor code changes
Practical	Adaptive execution based on query budget
Efficient	Online stratified sampling technique

Thank you!

Details: StreamApprox [Middleware'17]

https://streamapprox.github.io