Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Constant-Time Approximate Sliding Window Framework with Error Control

Álvaro Villalba

Former Research Engineer

05/08/2019

ISORC 2019 - València

A bit about me

- PhD Student at UPC - BarcelonaTECH
  - Computer Architecture Department
- Data-Stream Processing Lead at NearbyComputing
- Research Engineer at BSC (2012 – 2018)
  - Data-Centric Computing Group
  - IoT and Stream Processing

Overview

- Motivation
  - Stream processing + Edge Computing
- Constant-Time Scalable Sliding Window Framework – AMTA
  - Scalability and Complexity
- Approximate Aggregation with Error Control – A^2MTA
  - Sum-like Aggregations
  - Max-like Aggregations

Motivation

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

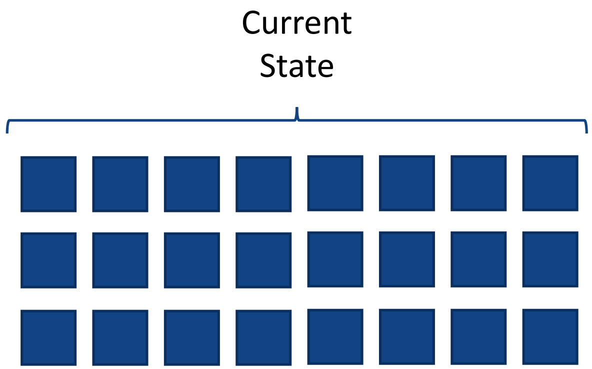
IoT and Big Data Convergence

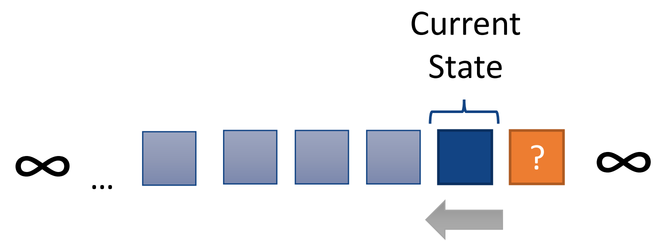
- Internet of Things has become ubiquitous
  - Gartner predicted that IoT will have nearly 21 billion connected devices by 2020
  - Cisco and Ericsson expects the number of connected IoT devices to be 50 billion by 2020
  - Largest spending technology category in 2018 with $800 billion
- Large amounts of data are being generated
  - Cisco predicts 14.1ZB per year by 2020

Edge Computing

- Cloud computing enables computing resources and storage with virtualized resources accessible to many users over the internet
  - Standard for Big Data
  - 14.1ZB per year by 2020 of data streams over the internet
  - Latency reaching data warehouses
- Edge computing brings the computation near the data sources
  - Freeing bandwidth from the internet
  - Reducing latencies between telemetry and actuation

Data Processing: Batches and Streams

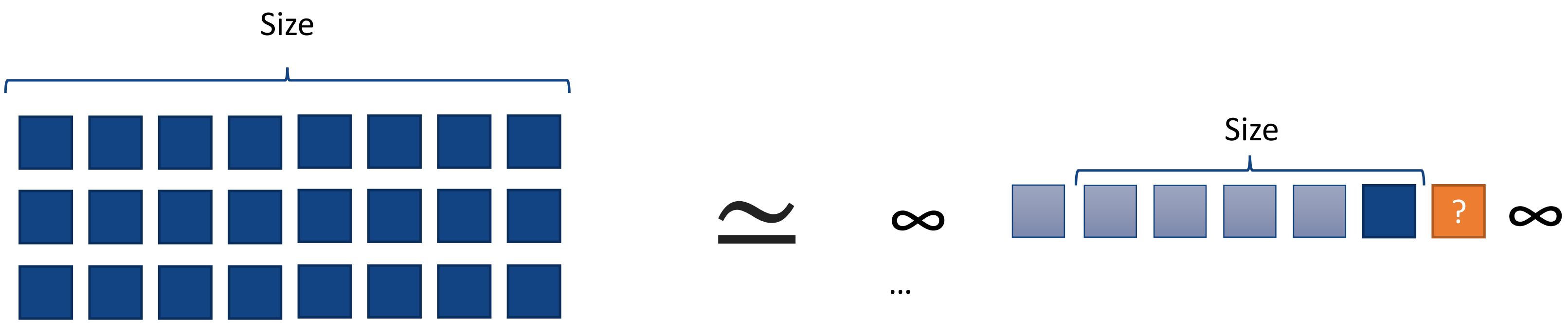




- **High throughput** but high latency
  - Throughput in ~100K+ TPS
- Big size of aggregation functions

- **Low latency** but low throughput
  - Latency in milliseconds or less
- Reduced size of aggregation functions

Stream Aggregation: Challenge



Stream Processing and Edge Computing

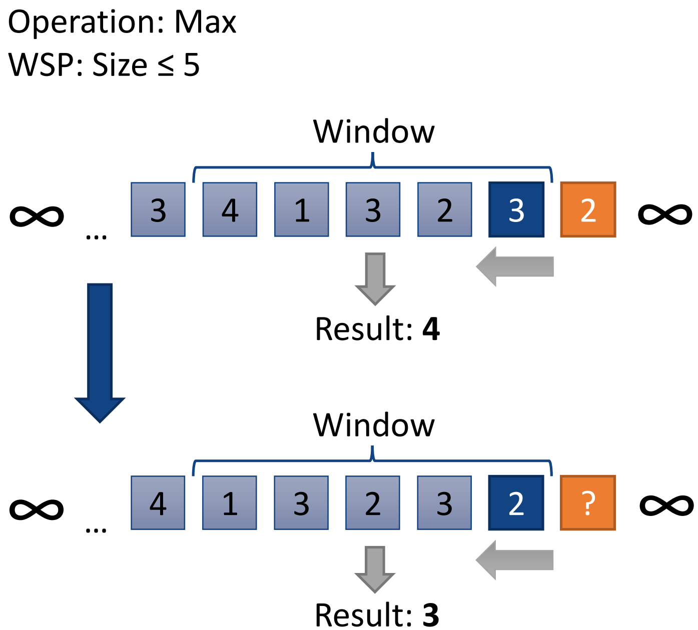
- Both paradigms prioritize low latency computation
  - Immediately after data is generated
  - Close to the data source
- Edge computing environment can be adverse
  - Limited and shared resources
  - Unreliable network
  - Slow maintenance

Constant-Time Scalable Sliding Window Framework

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Background: Sliding Window

- Projection from a stream that includes its newest element
  - FIFO structure
- Operation
- Window Slide Policy (WSP)
  - Usually only defines the size of the window



Background: Monoid

- Algebraic structure with the following properties:
  - Associativity
    - $\forall a, b, c \in S: \ (a \cdot b) \cdot c = a \cdot (b \cdot c)$
  - Neutral element
    - $\forall e \in S: \forall a \in S: e \cdot a = a \cdot e = a$
  - Closure
    - $\forall a, b \in S: \ a \cdot b \in S$

- Monoids can be an aggregation Reduce phase:
  - Associativity enables partial aggregation
  - Neutral element replaces values that are not aggregated anymore
  - Closure is obeyed by surrounding the Reduce with Maps, i.e.:

Mean aggregation:

Map: $\boldsymbol{f(x) = \{x, 1\}}$

Reduce: $\boldsymbol{f(x, y) = \{x_1 + y_1, x_2 + y_2\}}$

Map: $\boldsymbol{f(x) = \frac{x_1}{x_2}}$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Amortized Monoid Tree Aggregator (AMTA)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Amortized Monoid Tree Aggregator

- General sliding window framework
  - User provided monoid operation and slide policy
  - Operation invertibility agnostic
    - i.e. Sum (invertible) and Max (non-invertible)

- Distributed binary tree data structure

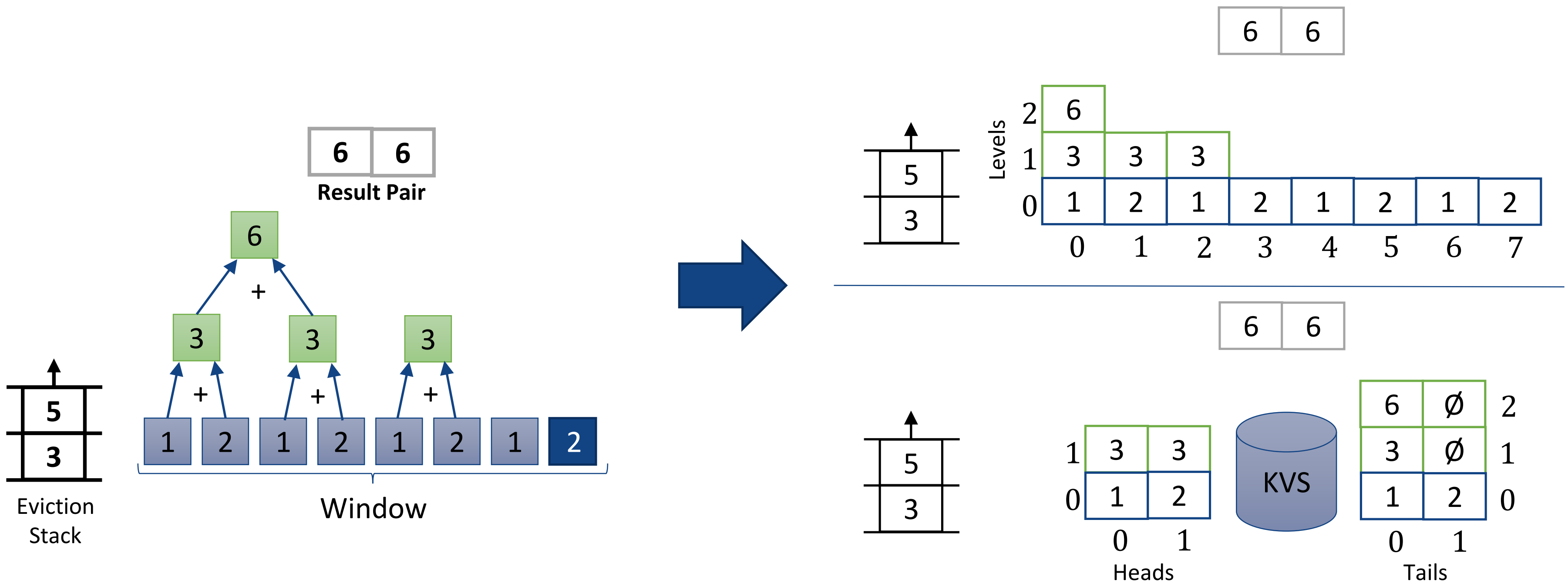
- Bulk eviction operation is atomic

- Amortized constant $O(1)$ time operations

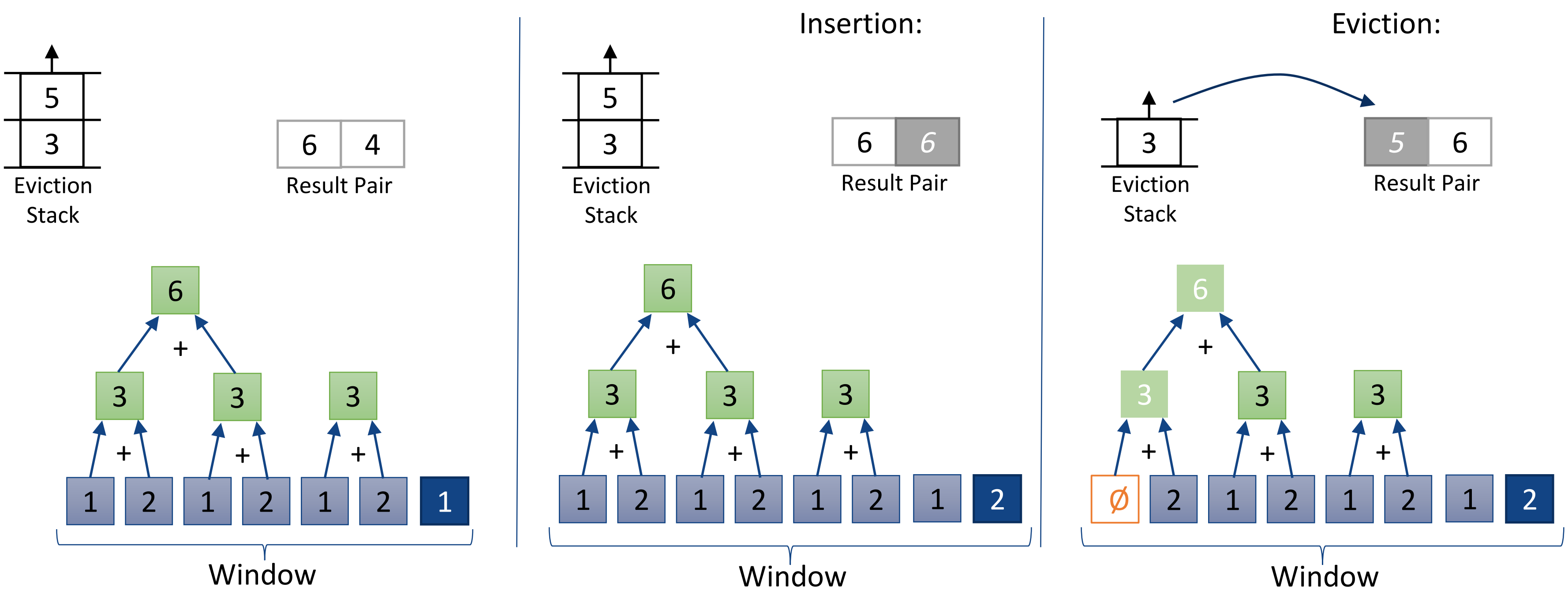
AMTA: Window Slide Policy (WSP)

- Programmatically decide which values need to be removed
- User-implemented interface
  - Inputs:
    - Current window result
    - Eviction candidate
  - Result:
    - Boolean – Eviction candidate satisfies WSP
- Assumptions
  - *Satisfied WSP* → All smaller eviction candidates satisfy the WSP
  - *Unsatisfied WSP* → Only smaller eviction candidates can satisfy the WSP

AMTA: Data Structure



AMTA: Basic operations



Approximate Aggregation with Error Control

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Background: Approximate Computing

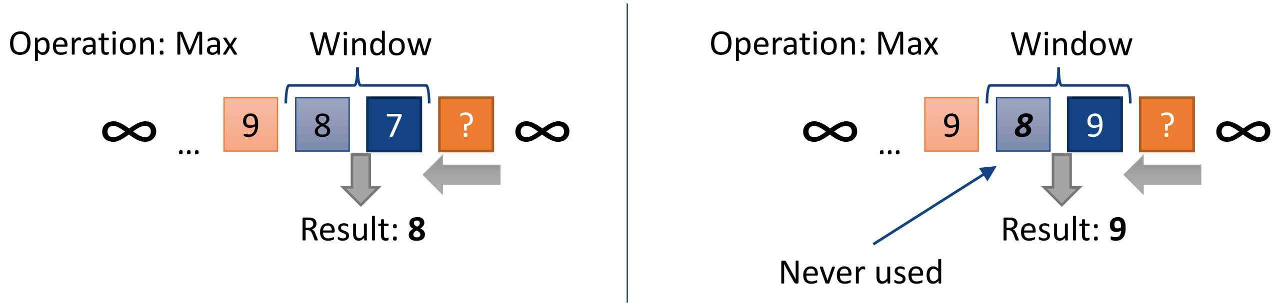
- Aggregation techniques that returns possibly inaccurate results
  - Results may contain some **error** compared to the accurate result
- Aggregation algorithms can benefit by
  - Reducing memory requirements
  - Reducing power consumption
  - Reducing network bandwidth
  - Improving performance
- Usually based on statistical predictions
- For example:
  - HyperLogLog
    - Approximate distinct count

Background: Sum-like aggregations

- Sum-like aggregations have only one effective neutral element
  - Results tend to constantly change
- The more extreme an input value is, the higher impact will have in its result
- Inverse function
- Although they all have an inverse function, it is not necessarily *subtraction*
  - However *subtraction* is used to calculate the error
- *Sum, count, average*

Background: Max-like aggregations

- Multiple values have a neutral effect on the aggregation
  - i.e. $Max(100, 99) = 100, Max(100, 98) = 100$ …
- Some value will never have an effect on the sliding window aggregation

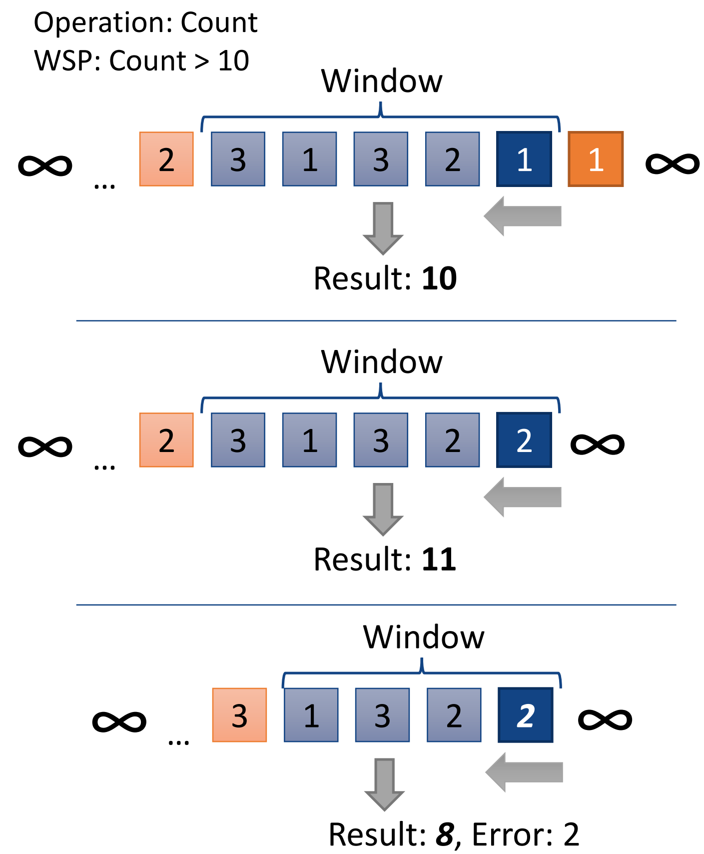


- No inverse function
- *Max, Min, argMax, argMin, maxCount*

Approximate AMTA
(A^2MTA)

Window Bucket

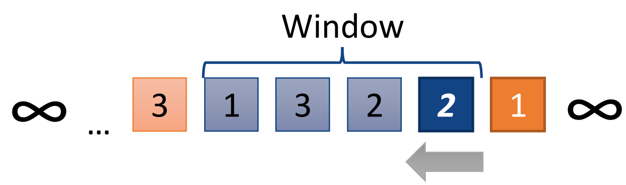
- Buckets are window members that aggregate multiple window input values
  - Reduced footprint
  - Granularity loss
    - Result error prone
- AMTA Trees don't propagate changes from the newest update
  - Performance improvement
- Error control requires a criteria for bucket sizes
  - Different kinds of aggregations require different criteria



Window Bucket: Error

- A bucket generate error in two scenarios
  - False positive eviction
    - The last bucket evicted aggregates values that wouldn't have been evicted outside the bucket

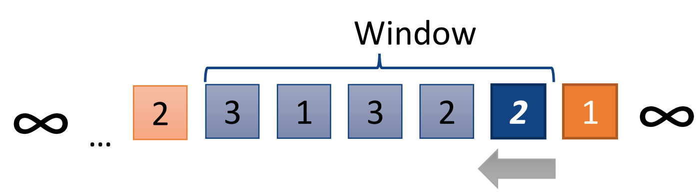
Operation: Count
WSP: result – candidate > 10
result – Ø = result



Result: ***8***
Exact error: 2
Potential error: 2

  - False negative eviction
    - The first bucket to be evicted aggregates values that would have been evicted outside the bucket

Operation: Count
WSP: result – candidate > 10
result – Ø = 10



Result: ***11***
Exact error: 1
Potential error: 2

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Sum-like histogram

- Goal: Keep the error generated by buckets inside user-defined boundaries
  - Decide if a bucket keeps growing considering its error
  - A relative error will depend on the result
  - An absolute error may also depend on the result
    - Not a *sum* aggregation: i.e. multiplicative aggregation
- Result **prediction interval** with a confidence level

$$\left(\bar{x} - t^{*}s\sqrt{1+\frac{1}{n}}, \bar{x} + t^{*}s\sqrt{1+\frac{1}{n}}\right)$$

  - Assuming the *central limit theorem*
- Absolute result error prediction

$$|r - M(b, r)|$$

r: predicted result, b: bucket error, M: monoid function

Max-like histogram

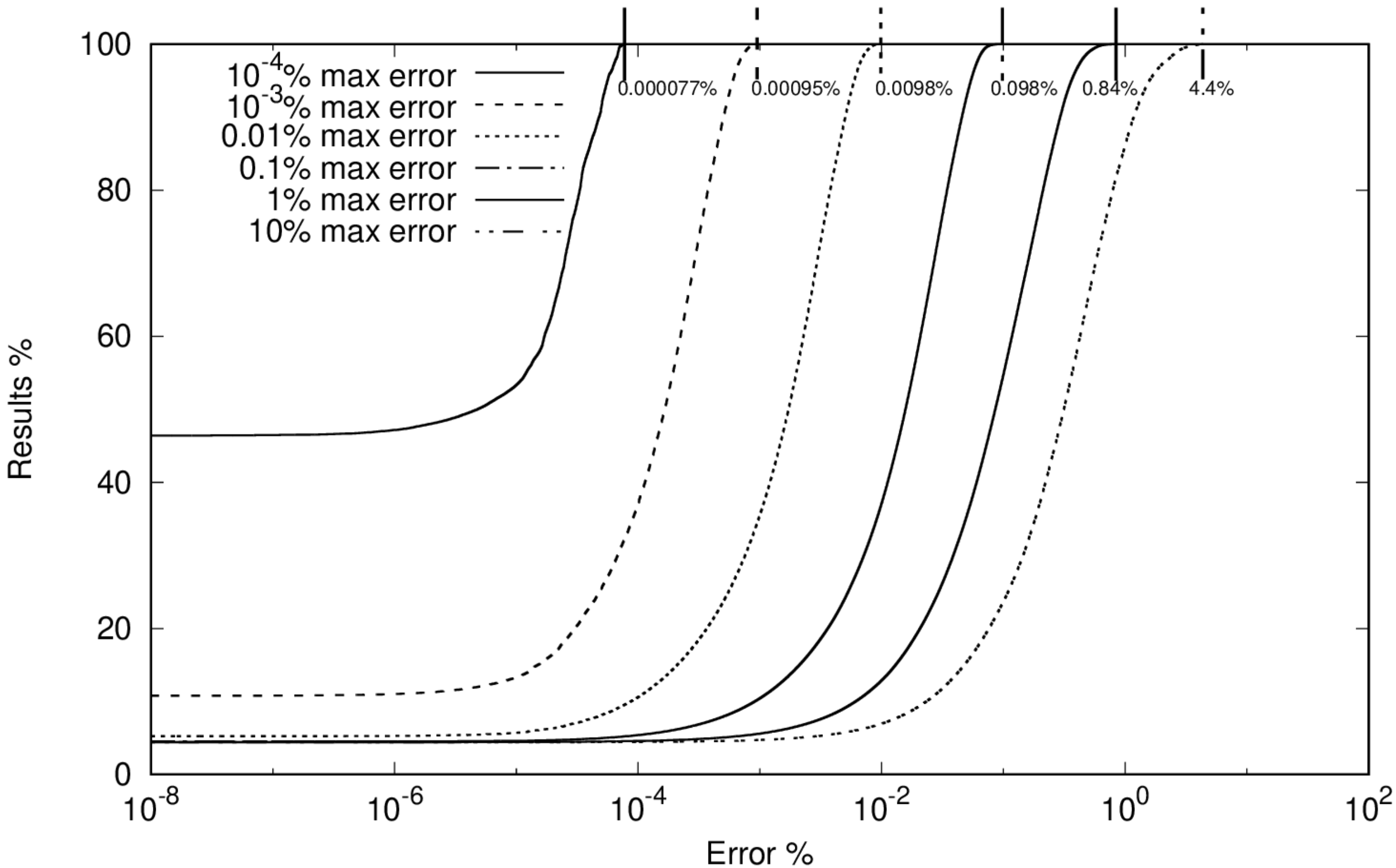
- Goal: Make buckets as big as possible while avoiding to produce any error
  - Aggregate in a bucket all values that are not predicted to become an extreme value
- Extreme value prediction: Fisher-Tippett Theorem
  - **Block Maxima**
  - Obtain *Generalized Extreme Value* distribution moments from the sample
    - Hosking GEV Probability-Weighted Moments (PWM) estimation method
  - Extract upper and lower bounds with a confidence level
- A less extreme input value than the GEV boundaries can be aggregated in the last bucket

Evaluation Methodology

- Data set
  - A year worth of real telemetry data: 1 update/s
- Evaluate **effective error** and **footprint** from methods configuration parameters
  - Sum-like: Parameter → Max error, Operation → Mean
  - Max-like: Parameter → Block size, Operation → Max
  - WSP → Month-worth updates
- Evaluate **latency** comparison:
  - Approximate AMTA (A^2MTA)
  - Amortized MTA (AMTA)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

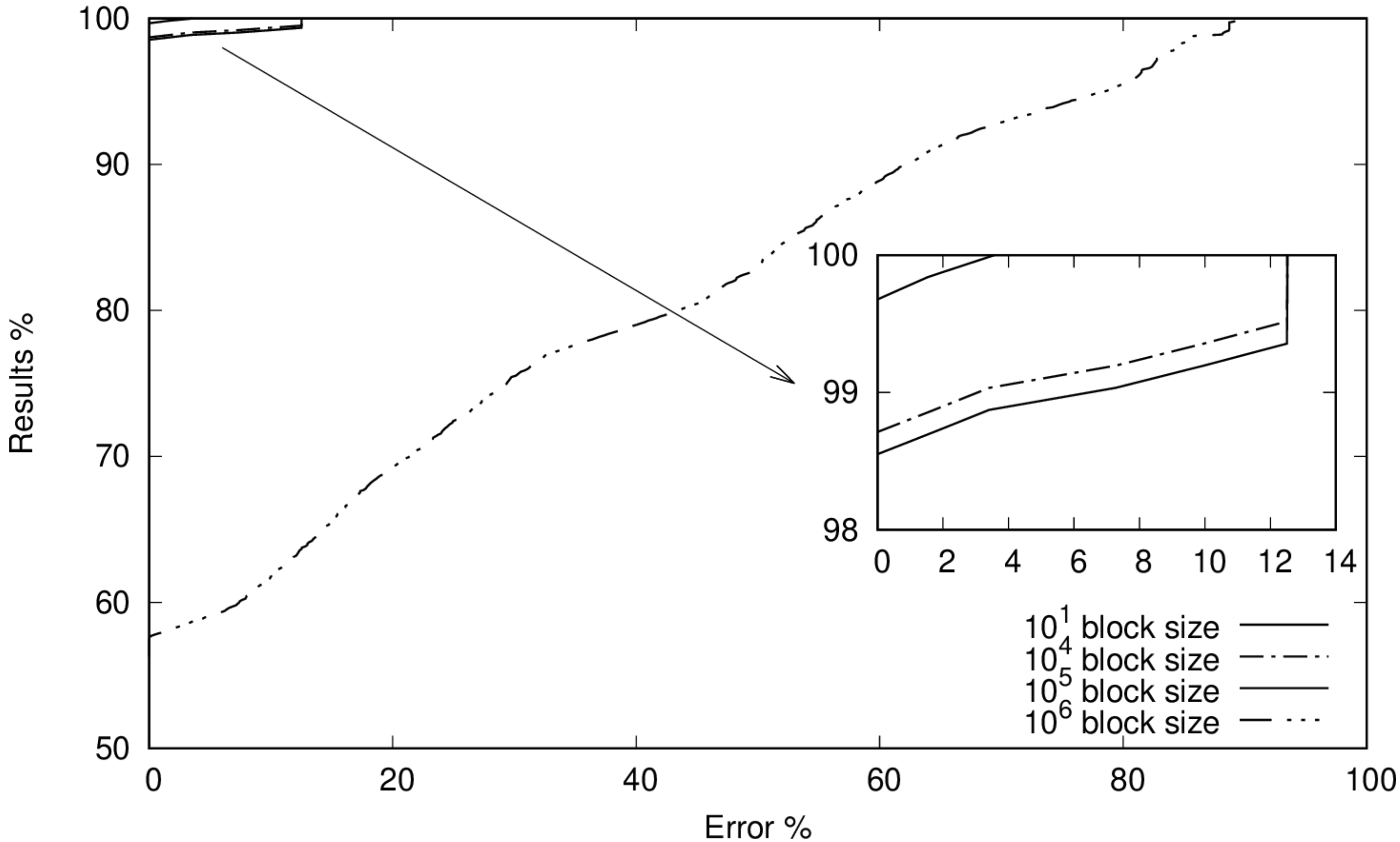
Evaluation: Sum-like Effective Error



Sum-like: Mean

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Evaluation: Max-like Effective Error



Max-like: Max

Evaluation: Footprint

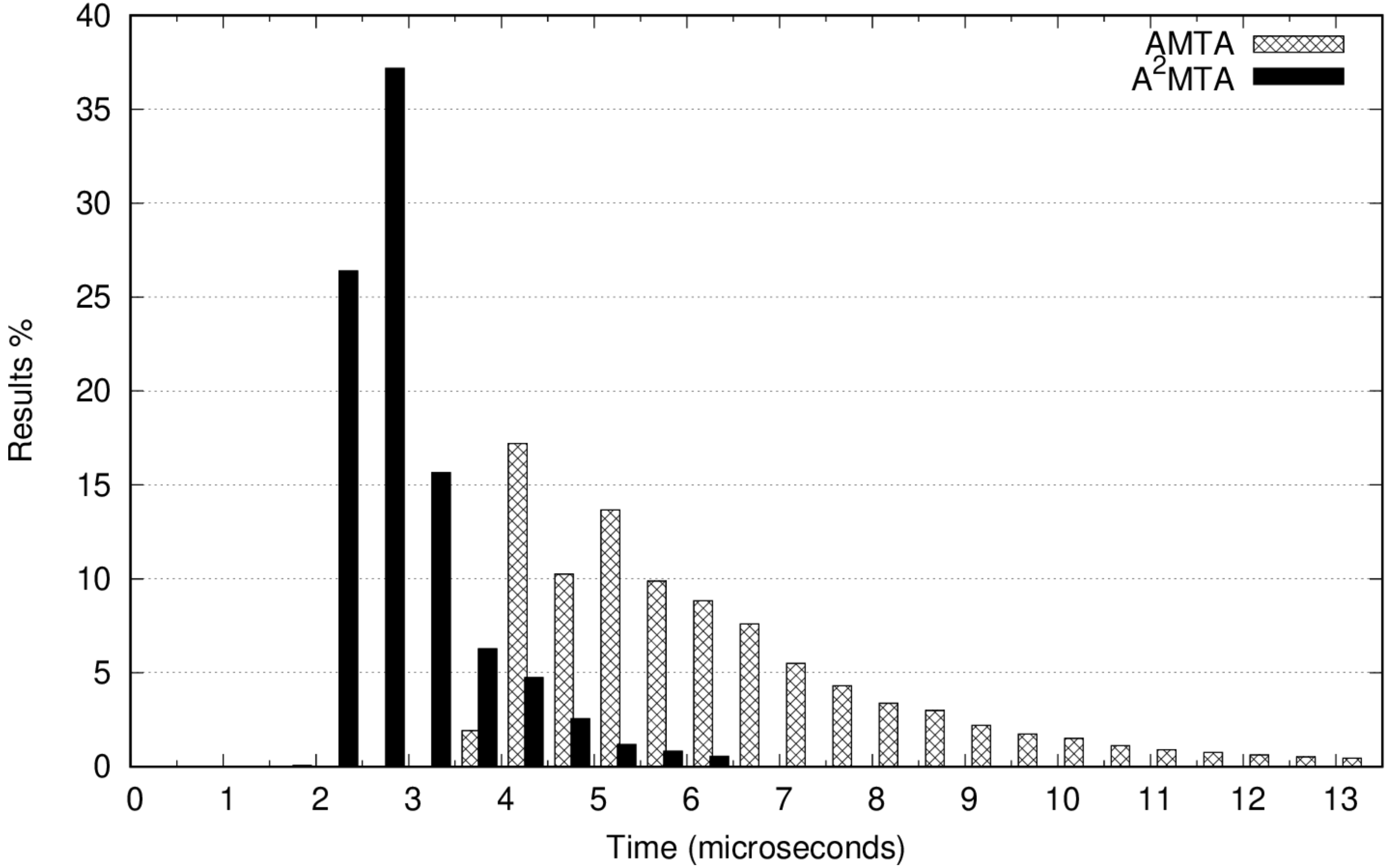
Sum-like histogram

Max error	Footprint
$10^{-4}\%$	$44{,}02\%$
$10^{-3}\%$	$6{,}591\%$
$10^{-2}\%$	$8{,}335 \cdot 10^{-1}\%$
$10^{-1}\%$	$9{,}9 \cdot 10^{-2}\%$
1%	$1{,}022 \cdot 10^{-2}\%$
10%	$9{,}854 \cdot 10^{-4}\%$

Max-like histogram

Block size	Footprint
10	$91{,}33\%$
10^2	$91{,}1\%$
10^3	$95{,}49\%$
10^4	$60{,}97\%$
10^5	$4{,}394\%$
10^6	$19{,}88\%$

Time Performance



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Final Considerations

- A²MTA extends AMTA with approximate computing mechanisms
- The evaluation demonstrated that:
  - General purpose stream processing approximation framework
  - Result error can be controlled with prediction techniques
  - Footprint is greatly reduced
    - Data structure element generation is reduced in the same proportion
    - Less distributed data store network traffic
  - Time performance is better in most cases
- Max-like require a *right* block size

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you

YourEmail@bsc.es