SpaceSaving Algorithm

STORM IMPLEMENTATION

Overview

- Frequent Elements in Data Streams
- *SpaceSaving* approach
- Storm Topology
- Experimental results

The Frequent Elements Problem

- In a stream S of N size, find all frequent elements
- Frequency must be greater than $\lceil \varphi N \rceil$, where φ in [0,1]
- φ is a proportion , e.g elements that occur more than **50%** times in N stream
- Need: Solve the problem with one pass over the data

SpaceSaving

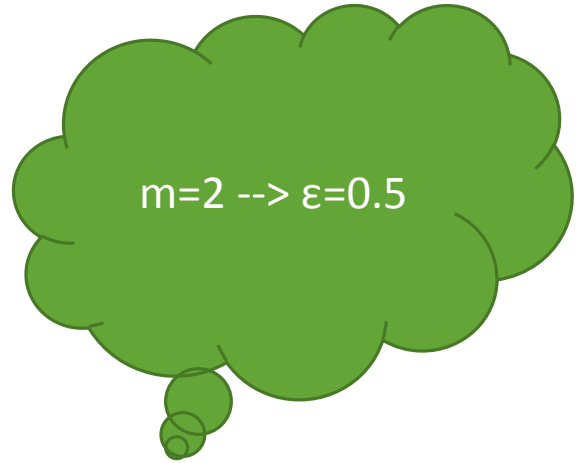
- ***Counter-Based*** Technique
- Use m counters to monitor m elements
- If element has already a counter, increment its counter by 1
- If not, replace the element with minimum counter (min)
- Set the frequency of the new element to min+1 , remember overestimation **min**

SpaceSaving

- Given an error rate ε, use at least $m = \frac{1}{\varepsilon}$ counters
- Then all elements with $f > \lceil \varphi N \rceil$, with $\varphi \geq \varepsilon$, are guaranteed to be reported
- We set φ=ε and $m = \frac{1}{\varepsilon}$, and report as frequent elements those with $\boldsymbol{f} > \lceil \varepsilon N \rceil$
- **Approximate** frequency $\boldsymbol{f}$ of a stored element derives of "count-overestimation"

Quick Example

Item	Count	Overestimation



Quick Example

X

Item	Count	Overestimation
X	1	0

Quick Example

X
Y

Item	Count	Overestimation
X	1	0
Y	1	0

Quick Example

X,Y
Y

Item	Count	Overestimation
X	1	0
Y	2	0

Quick Example

X,Y,Y
Z

Item	Count	Overestimation
Z	2	1
Y	2	0

Quick Example

X,Y,Y,Z
Y

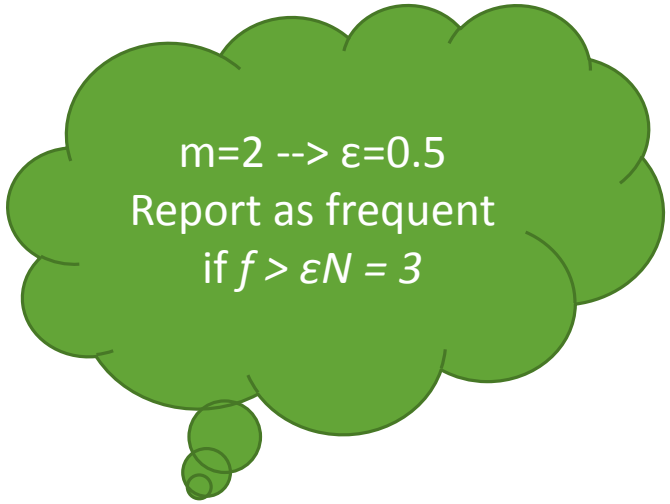
Item	Count	Overestimation
Z	2	1
Y	3	0

Quick Example

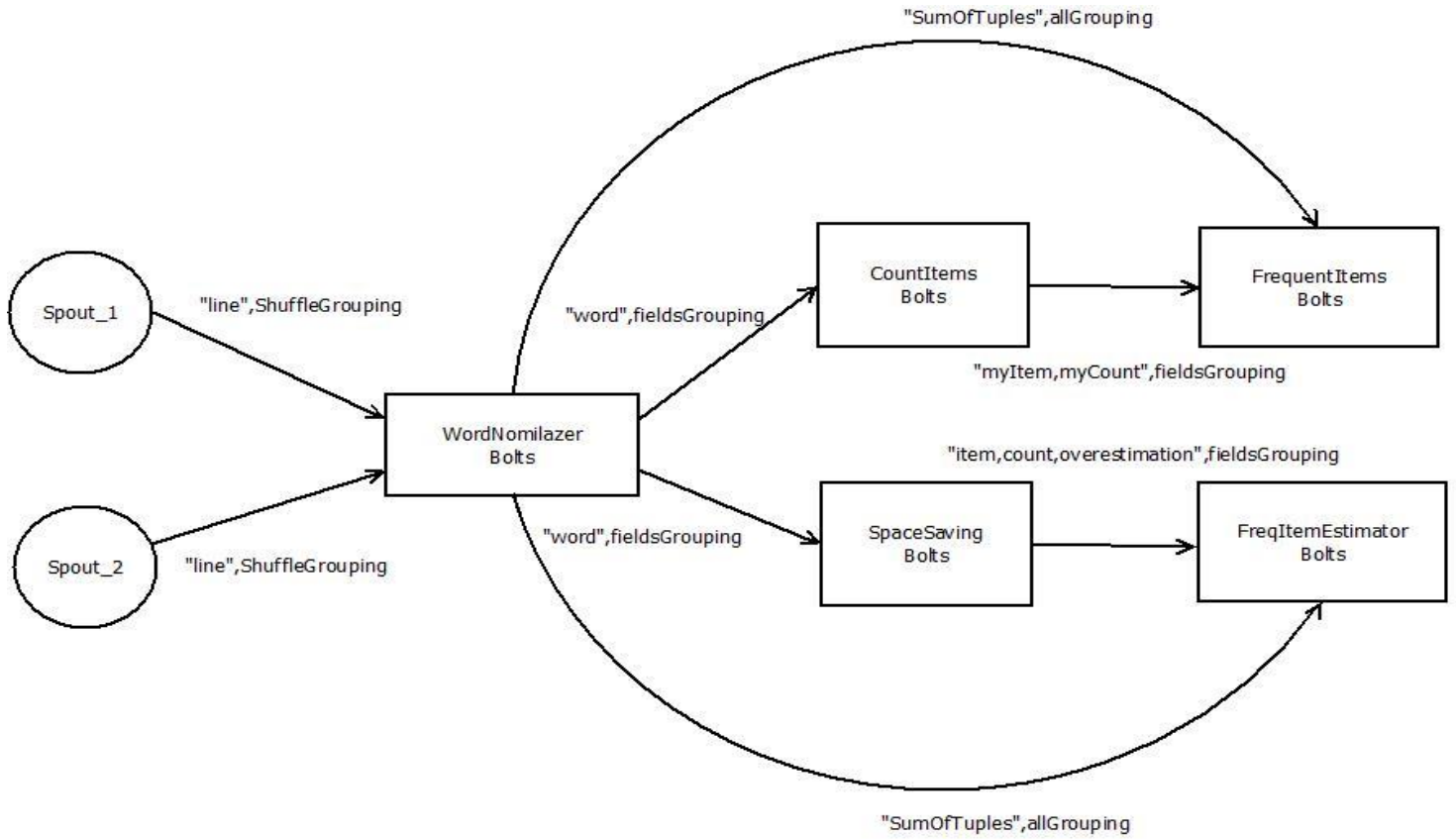
X,Y,Y,Z,Y

Y

Item	Count	Overestimation
Z	2	1
Y	4	0



Storm Topology



Bolt Parallelism: 4

Experiments

- Used Zipfian Synthetic DataSet with parameter $\alpha=\{0.5,1.5,2.5\}$
- Length Of Alphabet: $|A|=10^5$
- Number Of Items: $N=5 * 10^5$
- Parallelism in Bolts : 4
- Metrics Of "Success": Precision, Recall

Experiments

- Metrics Used

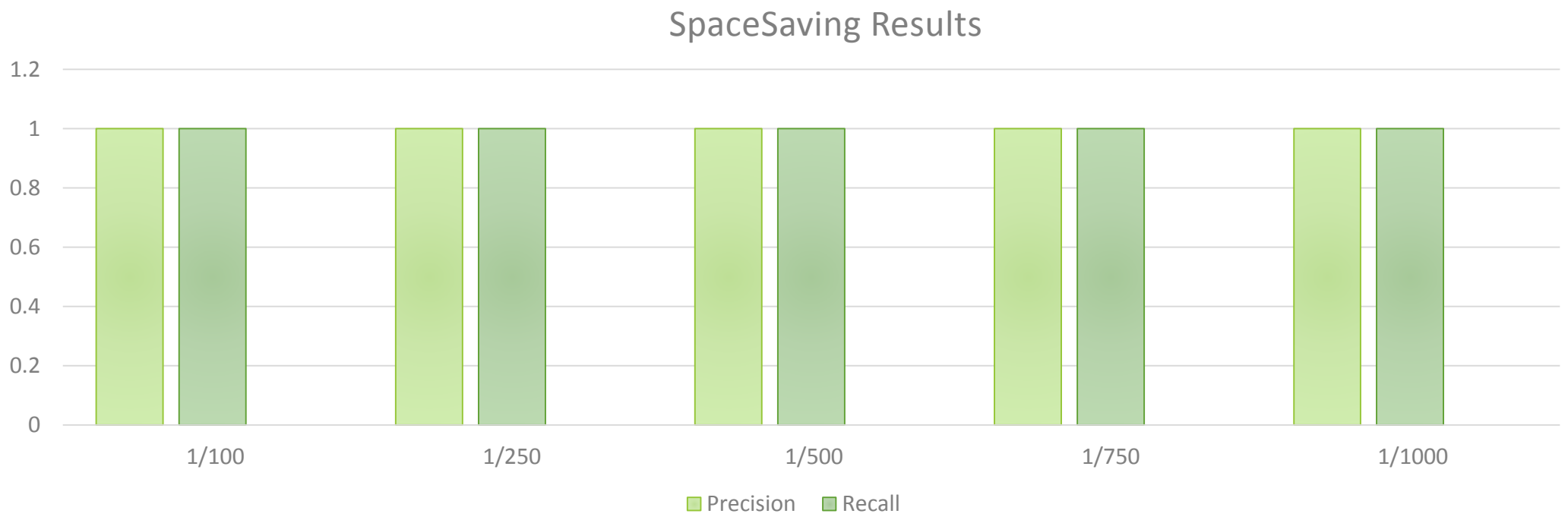
$$precision = \frac{number\ of\ retrieved\ frequent\ elements}{number\ of\ retrieved\ elements}$$

$$recall = \frac{number\ of\ retrieved\ frequent\ elements}{total\ number\ of\ frequent\ elements\ in\ the\ dataset}$$

- The method has found all frequent items with no false alarms if both are equal to‘1.0’

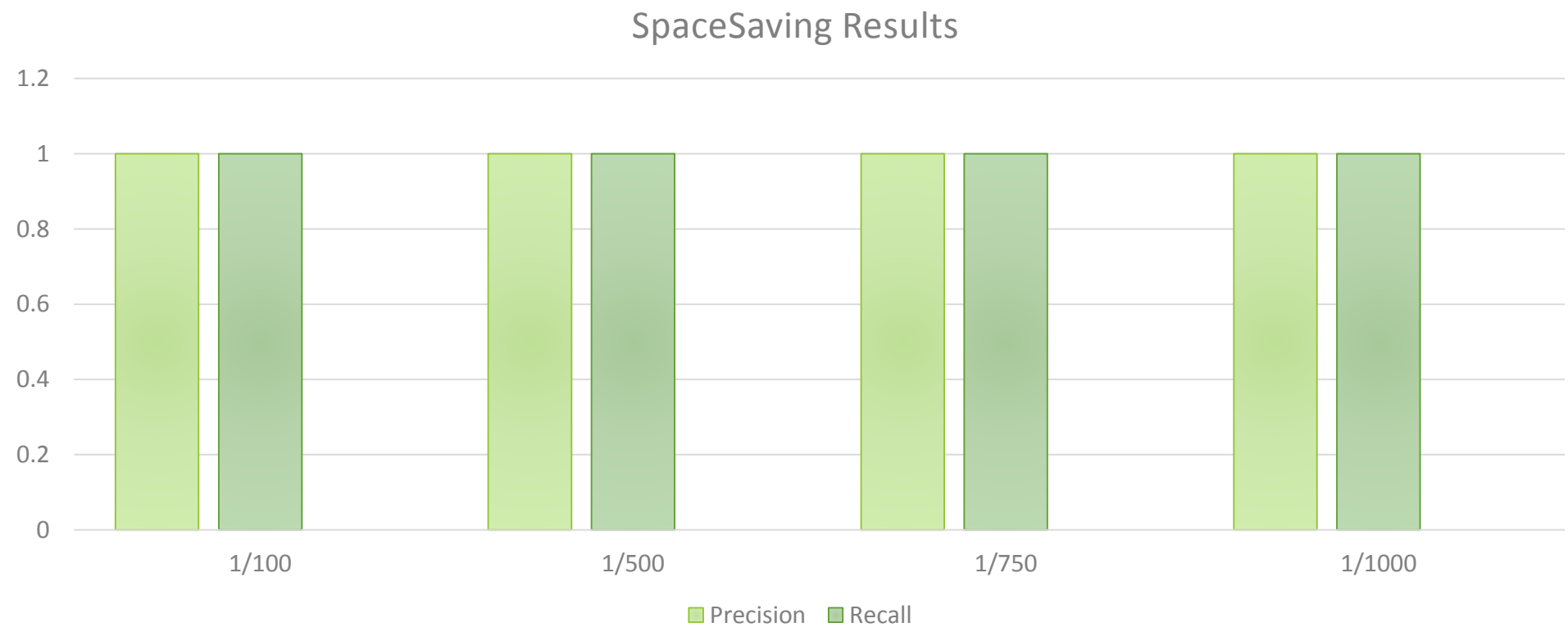
Experimental Results

- Zipfian Data with α=1.5 and N=$5 * 10^5$



Experimental Results

- Zipfian Data with α=2.5 and N=$5 * 10^5$




Thank You!