WHEN TWO CHOICES ARE NOT ENOUGH

Balancing at Scale in Distributed Stream Processing

Muhammad Anis Uddin Nasir[1$], Gianmarco de Francisci Morales[3¢], Nicolas Kourtellis[2¥], Marco Serafini[3€]
[1] KTH Royal Institute of Technology, [2] Telefonica Research Barcelona, [3] Qatar Computing Research Institute
anisu@kth.se[$], gdfm@acm.org[¢], nicolas.kourtellis@telefonica.com[¥], mserafini@qf.org.qa[€]

Abstract

- Given a highly skewed input stream, **minimize the load imbalance** across the machines to achieve **better hardware utilization, higher throughput and lower processing latency**
- We propose two novel algorithms : a) **D-Choices** and b) **W-Choices**. Both the algorithms operate by identifying the head of the input stream and placing the head on more than two workers

Problem Statement

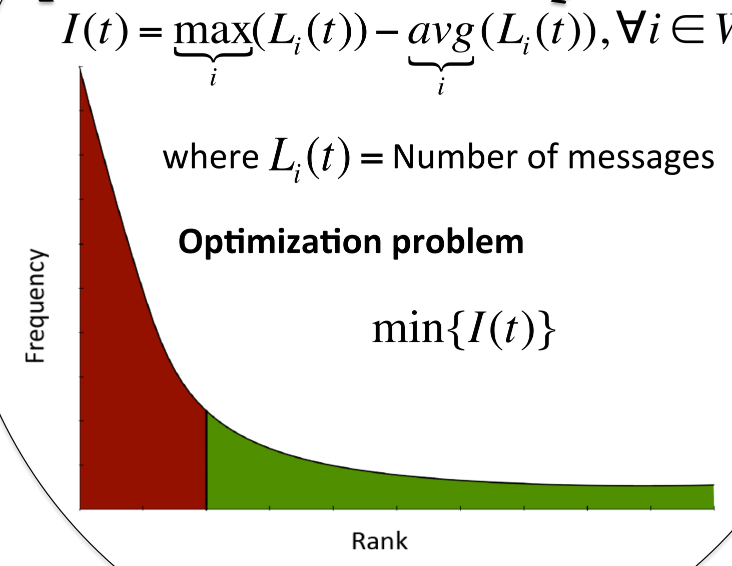
Imbalance — Load — Worker

$$I(t) = \max_i (L_i(t)) - \underset{i}{avg}(L_i(t)), \forall i \in W$$

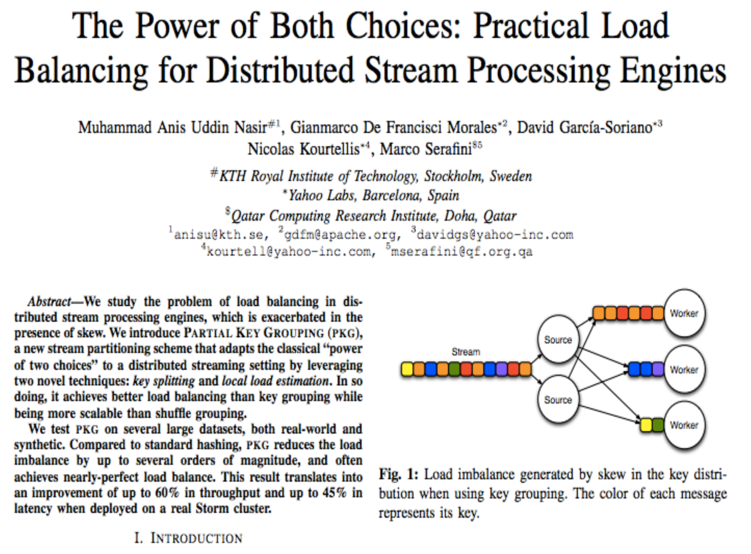
where $L_i(t)$ = Number of messages

Optimization problem

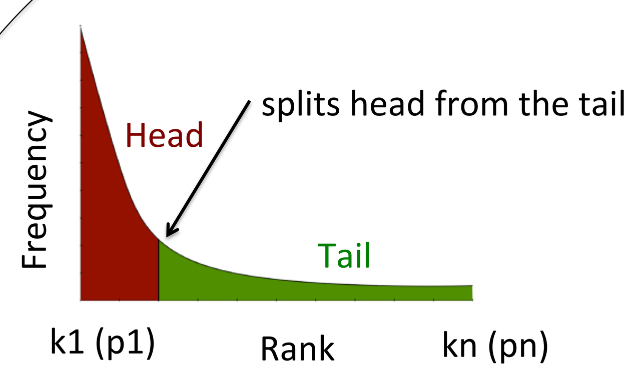
$$\min\{I(t)\}$$

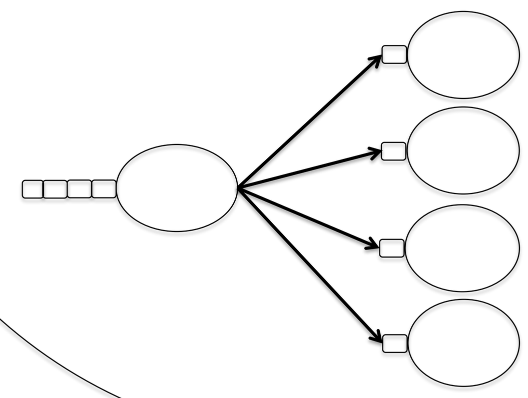


Past Work: ICDE 2015



Solution Overview





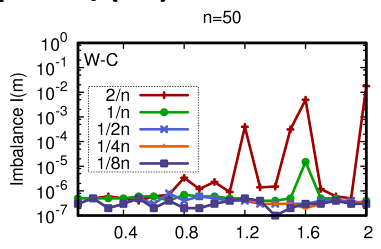
Solution Overview

Observation

PKG guarantees nearly perfect load balance for **p1 ≤ 1/(5n)**

Solution

Stream Summary to handle the Head



D-Choices:
- adapts to the frequencies
- allows subset of workers

W-Choices:
- Independent of frequencies
- allows all the workers

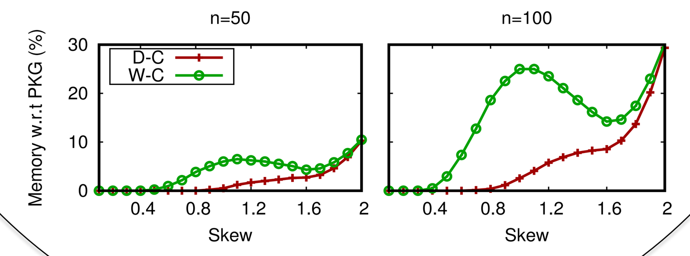
Algorithm:
- Assign head to the set of d/w workers
- Handle tail using power of both choices

Solution Overview

Pseudocode

```
Algorithm 1: Stream partitioning algorithm.
upon message m = ⟨k, v⟩
    H ← UPDATESPACESAVING(k)
    d ← 2 // Default as in PKG
    if k ∈ H then
        if D-CHOICES then
            d ← FINDOPTIMALCHOICES()
        else if W-CHOICES then
            d ← n
    w ← MINLOAD(F_1(k), ..., F_d(k))
    send(w, m)
```

Memory Overhead



Experimental Setup

Dataset

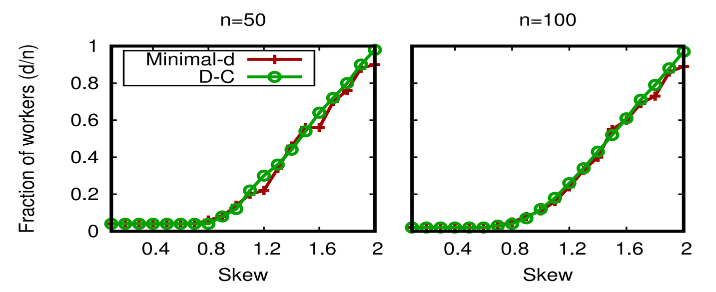
Dataset	Symbol	Messages	Keys	$p_1(\%)$
Wikipedia	WP	22M	2.9M	9.32
Twitter	TW	1.2G	31M	2.67
Cashtags	CT	690k	2.9k	3.29
Zipf	ZF	10^7	10^4,10^5,10^6	$\propto \frac{1}{\sum x^{-z}}$

Algorithms

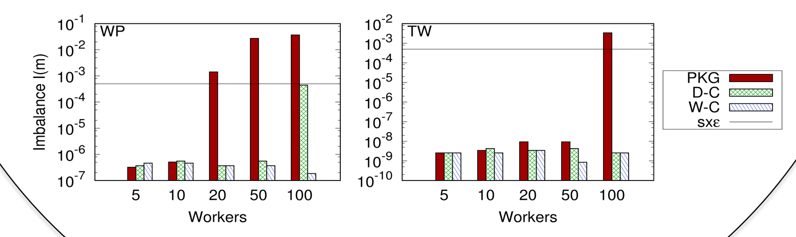
Symbol	Algorithm	Head vs. Tail
D-C	D-Choices	Specialized on head
W-C	W-Choices	
RR	Round-Robin	
PKG	Partial Key Grouping	Treats all keys equally
SG	Shuffle Grouping	

Experimental Results

Estimation

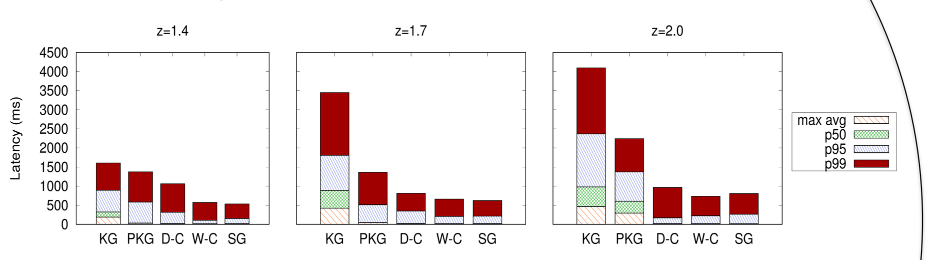


Load Imbalance

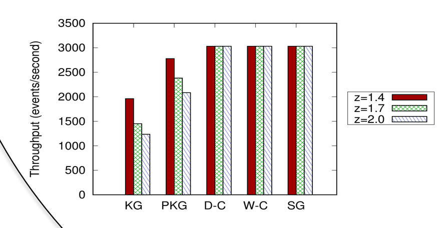


Experimental Results

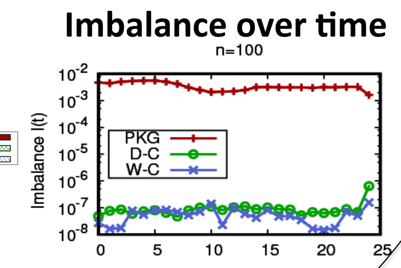
Latency



Throughput



Imbalance over time



Analysis

$$\underset{d}{\text{minimize}} \quad f(d;\mathcal{D},\theta) = d \times |\mathcal{H}_{\mathcal{D},\theta}|$$
$$\text{subject to} \quad \underset{d}{\mathbb{E}}[I(m)] \leq \epsilon .$$

Expanding the constraint we get:

$$\sum_{i \leq h} p_i + \left(\frac{b}{n}\right)^d \sum_{h<i\leq|H|} p_i + \left(\frac{b}{n}\right)^2 \sum_{i>|H|} p_i \leq \left(\frac{b}{n}\right) + \varepsilon$$

where

$$b = n - n\left(\frac{n-1}{n}\right)^{h \times d}$$

Conclusion

- We propose two algorithms to achieve load balance at scale for DSPEs
- Use heavy hitters to separate the head of the distribution and process on larger set of workers
- Improvement translates into 150% gain in throughput and 60% gain in latency over PKG

Future Work

- Dynamic Load Balancing for stateful operators
  - Key Migration
  - Partition Migration
  - Queuing Theory
- Load Balancing in Heterogeneous Cluster
  - Load Prediction
  - Worker Communication

32nd IEEE International Conference on Data Engineering May 16-20, 2016 · Helsinki, Finland