

The Power of Both Choices: Practical Load Balancing for Distributed Stream Processing Engines

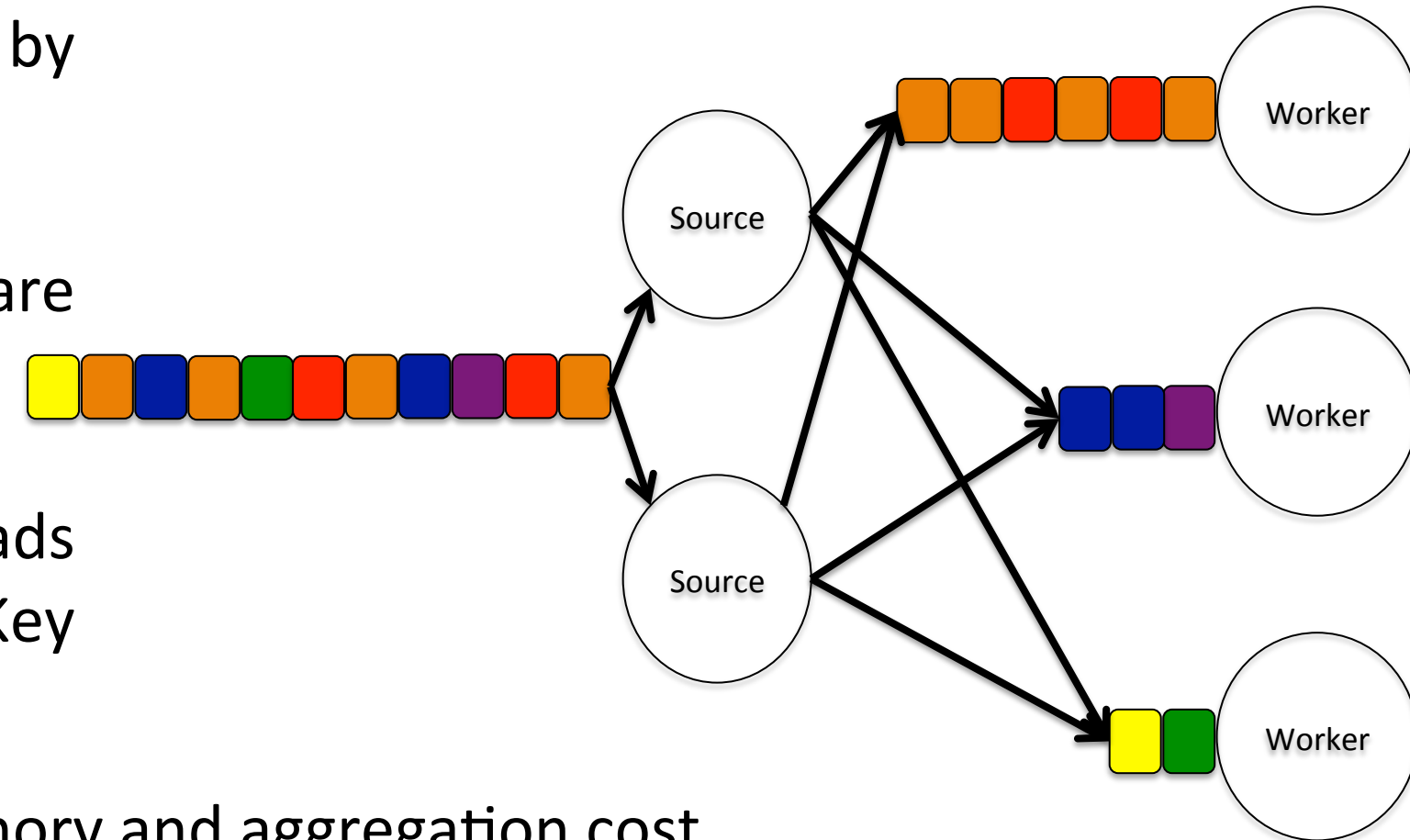
Muhammad Anis Uddin Nasir^{1§}, Gianmarco de Francisci Morales^{2c}, David Garcia Soriano^{3f}, Nicolas Kourtellis^{3¥}, Marco Serafini^{4€}
¹KTH Royal Institute of Technology, ²Aalto University, ³Yahoo Labs Barcelona, ⁴Qatar Computing Research Institute
 anisu@kth.se[§], gdfm@acm.org^c, {davidgs^f, kourtel[¥]}@yahoo-inc.com, mserafini@qf.org.qa[€]

1. Abstract

- **Load Balancing** in Distributed Stream Processing Engines directly affects the hardware utilization, latency, and throughput of the system
- We introduce **Partial Key Grouping** that leverages the classical “power of two choices”
- It achieves **better load balancing** than Key Grouping while being **more scalable** than Shuffle Grouping

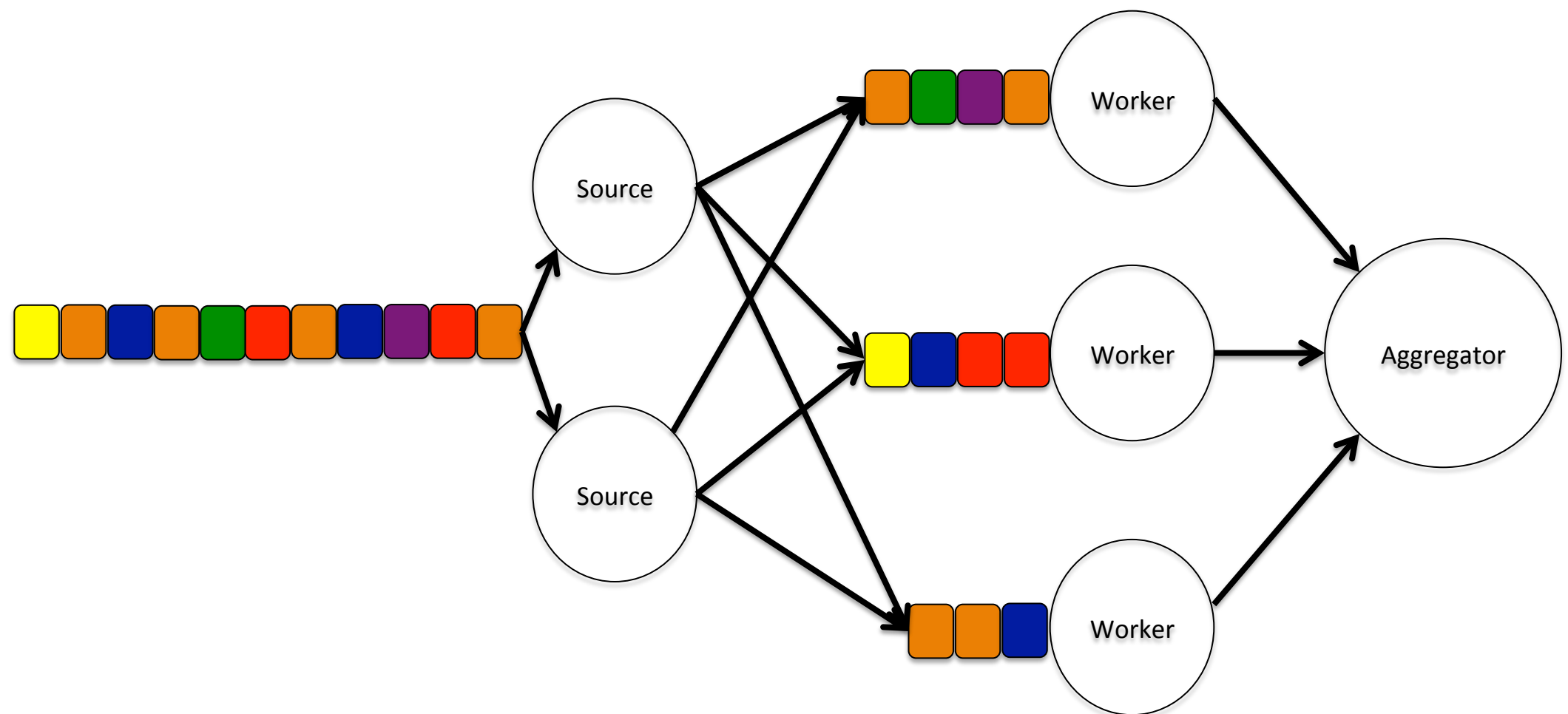
2. Problem

- Streaming applications are represented by directed acyclic graphs (DAG)
- Key Grouping and Shuffle Grouping are two common partitioning strategies
- Highly skewed distribution of workloads creates imbalance among workers for Key Grouping
- Shuffle Grouping incurs additional memory and aggregation cost
- A streaming and distributed solution is required for the problem



3. Partial Key Grouping

- PKG makes “power of two choices” applicable to our problem by using two novel techniques: **Key Splitting** and **Local Load Estimation**
- **Key Splitting** relaxes the atomicity constraint of key grouping and assigns each key to exactly two servers
- **Local Load Estimation** solves the problem of gauging the load of downstream servers without any communication overhead



4. Theoretical Analysis

- Suppose we have m messages (balls), k keys (colors), n workers (bins). Each key is placed in d bins (choices). Assume a key distribution D with maximum probability $p_1 \leq d/n$. Then, imbalance at time t is given by:

$$I(t) = \max_i(L_i(t)) - \text{avg}(L_i(t)), \text{ for } i \in \mathcal{W}$$

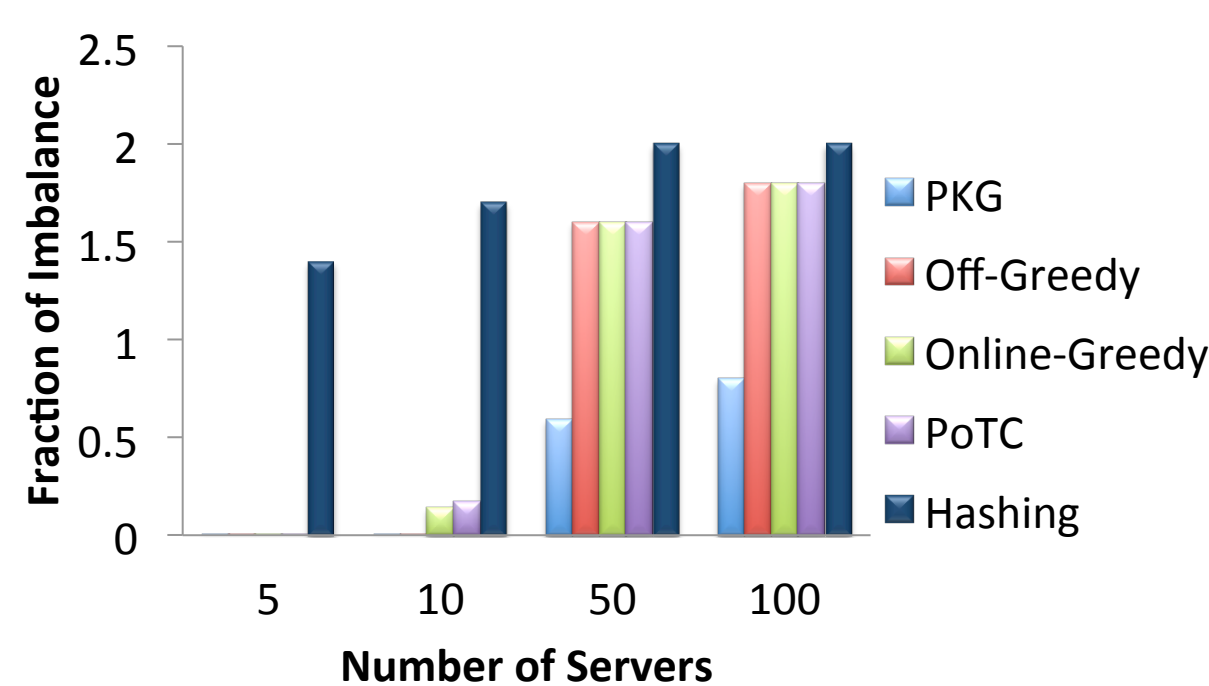
- Total imbalance is given by:

$$I(m) = \begin{cases} O\left(\frac{m}{n} \cdot \frac{\ln n}{\ln \ln n}\right), & \text{if } d = 1 \\ O\left(\frac{m}{n}\right), & \text{if } d \geq 2 \end{cases}$$

5. Experimental Setup

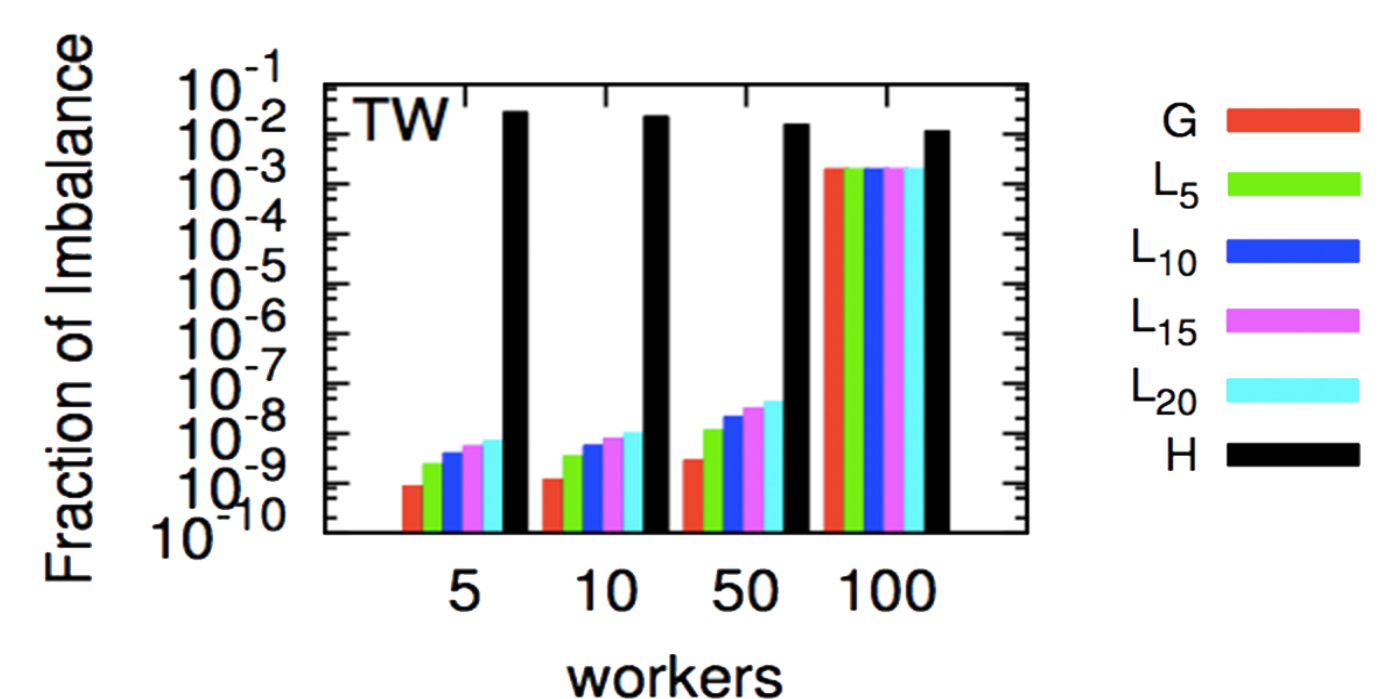
Questions

- Effect of Key Splitting
- Local vs Global Estimation
- Robustness
- Real Implementation



6. Simulations

- Hashing generates large imbalance compared to PKG
- PKG with local load estimation works as well as with global oracle

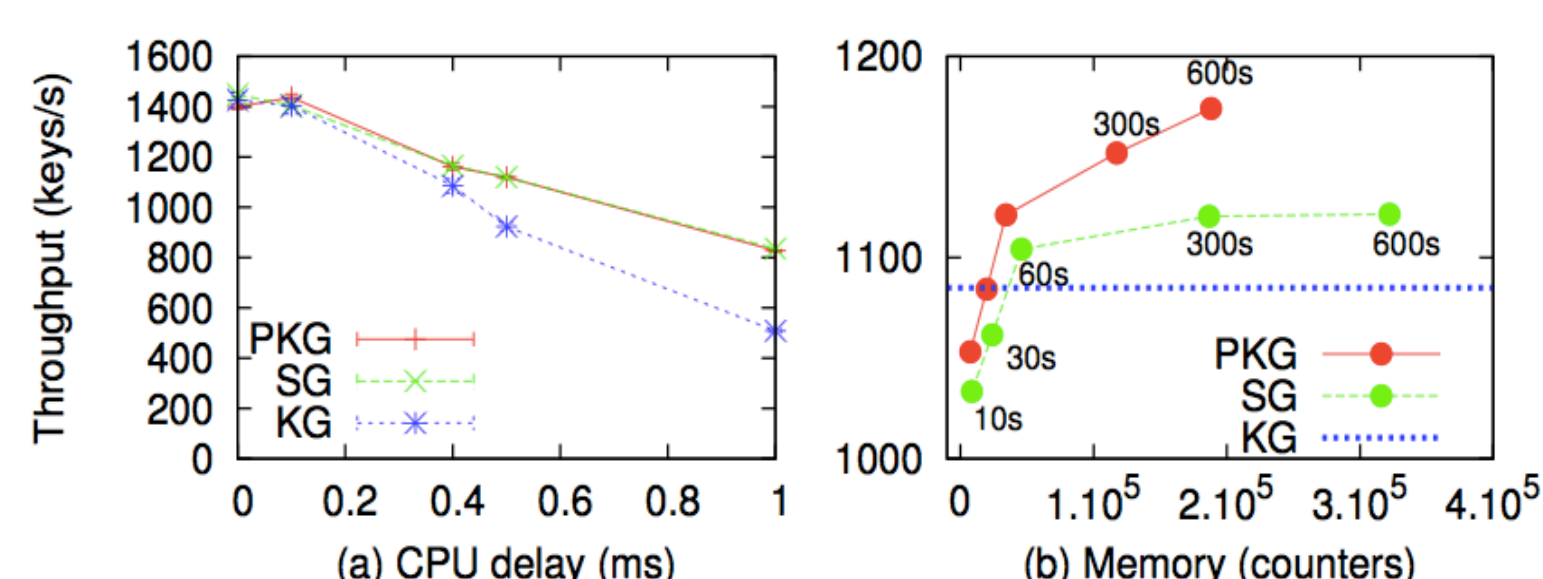


7. Real Implementation

- PKG provides better load balance compared to other strategies, e.g., greedy and power of two choices
- PKG performs robustly to data skew and concept drift
- PKG reduces the load imbalance by up to seven orders of magnitude compared to Key Grouping

Storm Implementation

- 60% improvement in throughput
- 45% improvement in latency



8. Future Work

- Load Balancing for Stateful operators using key migration
- Adaptive Load Balancing for Skewed Data
- Load Balancing for Graph Processing Systems

