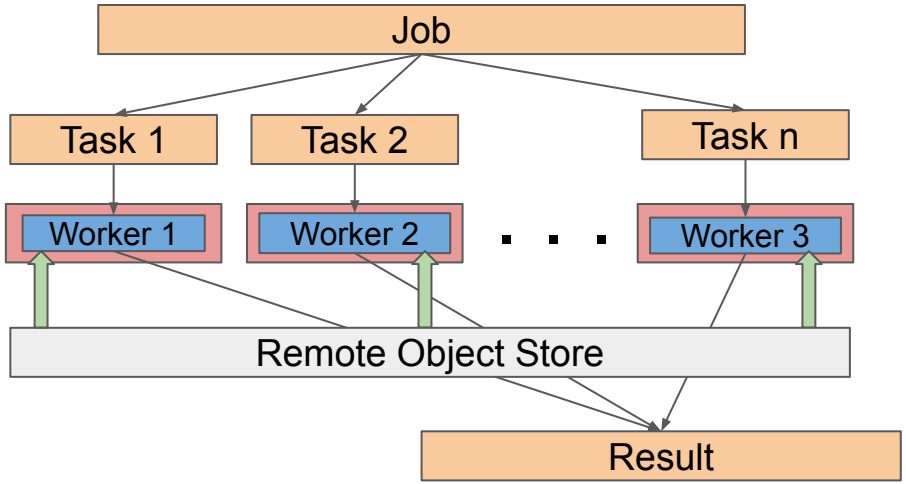
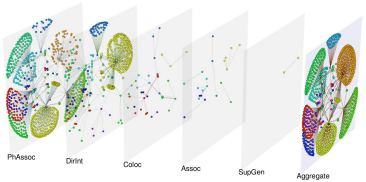
Efficient Estimation of Read Density when Caching for Big Data Processing

Sacheendra Talluri
Alexandru Iosup

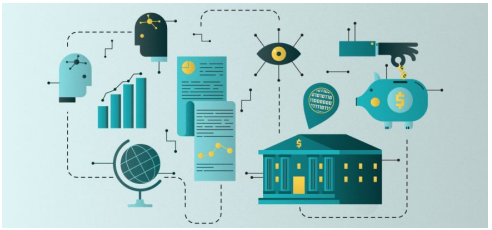


TU Delft VU VRIJE UNIVERSITEIT AMSTERDAM

What is Big Data?



Google



LIGO

Medicine **Search** **Finance** **Science**

- Volume
- Velocity
- Variety
- ...

Processed by multiple machines concurrently.

[1] Human HIV-1 protein interaction network, Wikimedia Commons
[2] Google Inc.
[3] Samsung Insights
[4] LIGO Scientific Collaboration

Motivation

Scarcity of publicly available information about caching for big data processing

Lack of cache policies for big data processing workloads

	Workloads
Huang 2013	Social Media
Berger 2017	Web
Blankstein 2017	Web, Database, Workstation
Einziger 2017	Web, Database, Workstation
Beckmann 2018	Web, Database, Workstation
This Work	**Big Data**

Motivation

Scarcity of publicly available information about caching for big data processing

Lack of cache policies for big data processing workloads

	Workloads
Huang 2013	Social Media
Berger 2017	Web
Blankstein 2017	Web, Database, Workstation
Einziger 2017	Web, Database, Workstation
Beckmann 2018	Web, Database, Workstation
This Work	**Big Data**

Motivation

Scarcity of publicly available information about caching for big data processing

Research Question 1

How do existing cache policies perform for big data processing?

Impact

Enables the selection of better policies by system designers and operators

Motivation

Scarcity of publicly available information about caching for big data processing

Leading to a lack of policies specifically for big data processing systems

Research Question 1

How do existing cache policies perform for big data processing?

Research Question 2

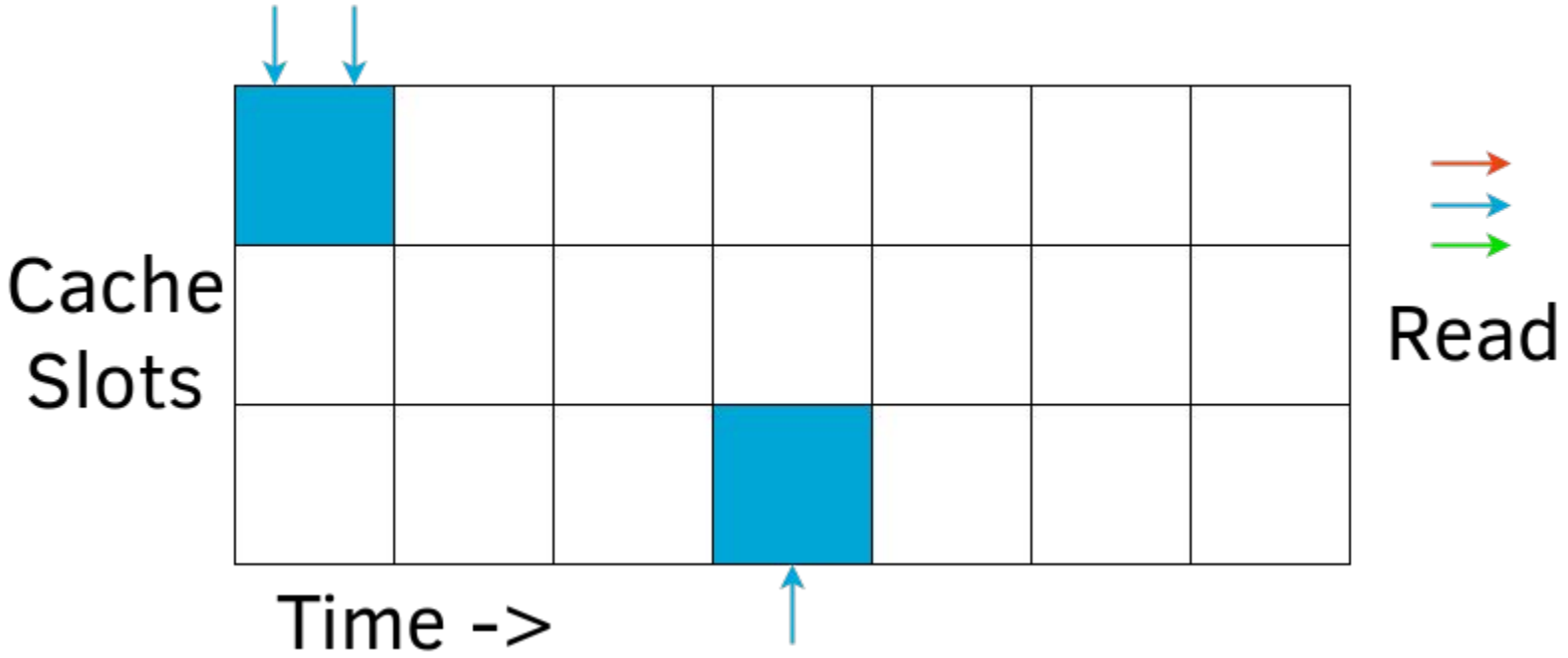
Could a policy be specifically designed with big data processing in mind?

Impact

Improved storage performance for big data processing

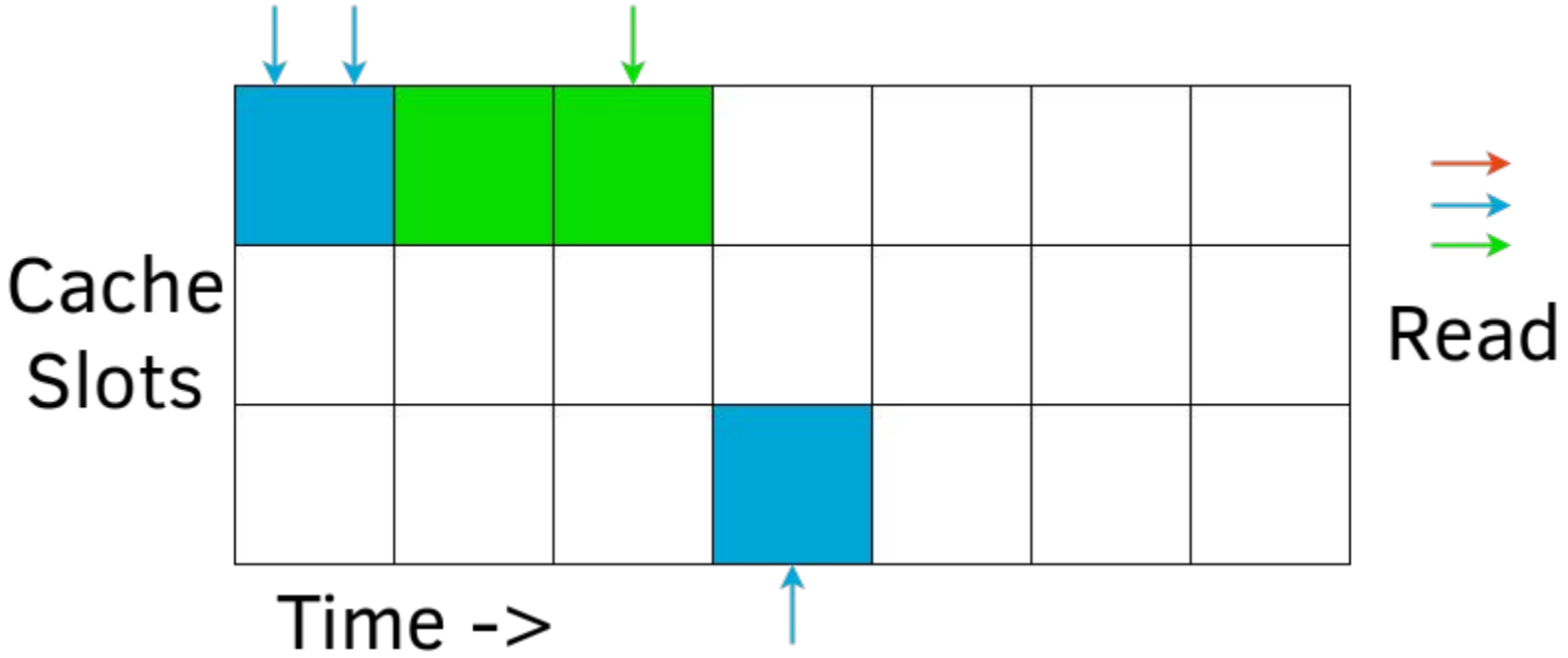
Design of the *Approximate Read Density* Cache Policy

Read Density Family of Utility Metrics



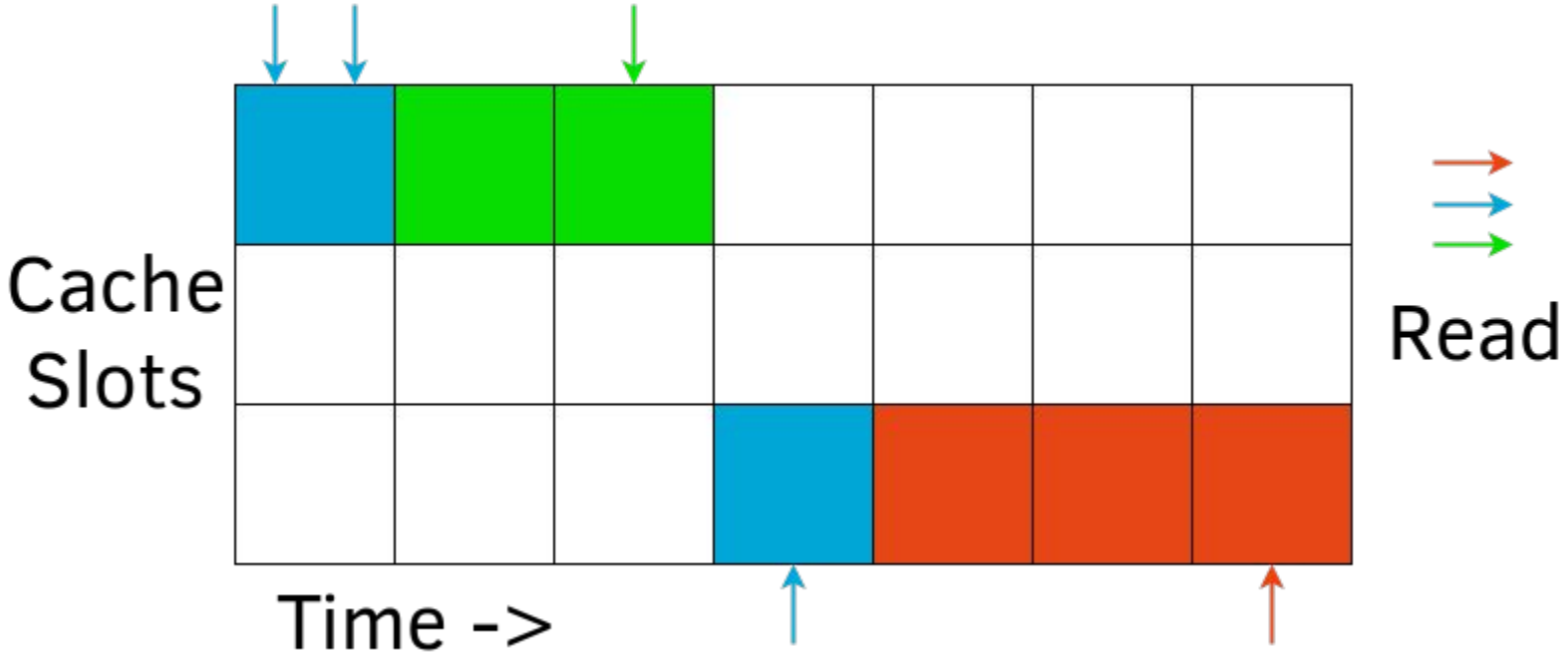
Design of the *Approximate Read Density* Cache Policy

Read Density Family of Utility Metrics



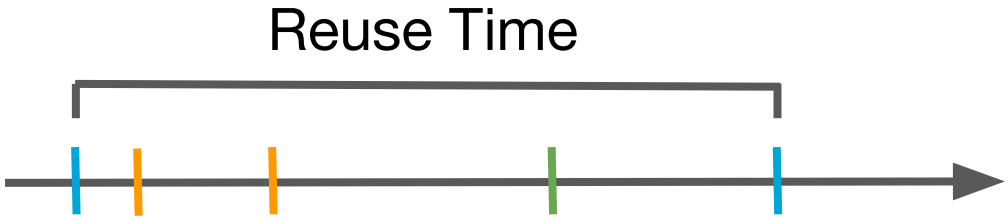
Design of the *Approximate Read Density* Cache Policy

Read Density Family of Utility Metrics

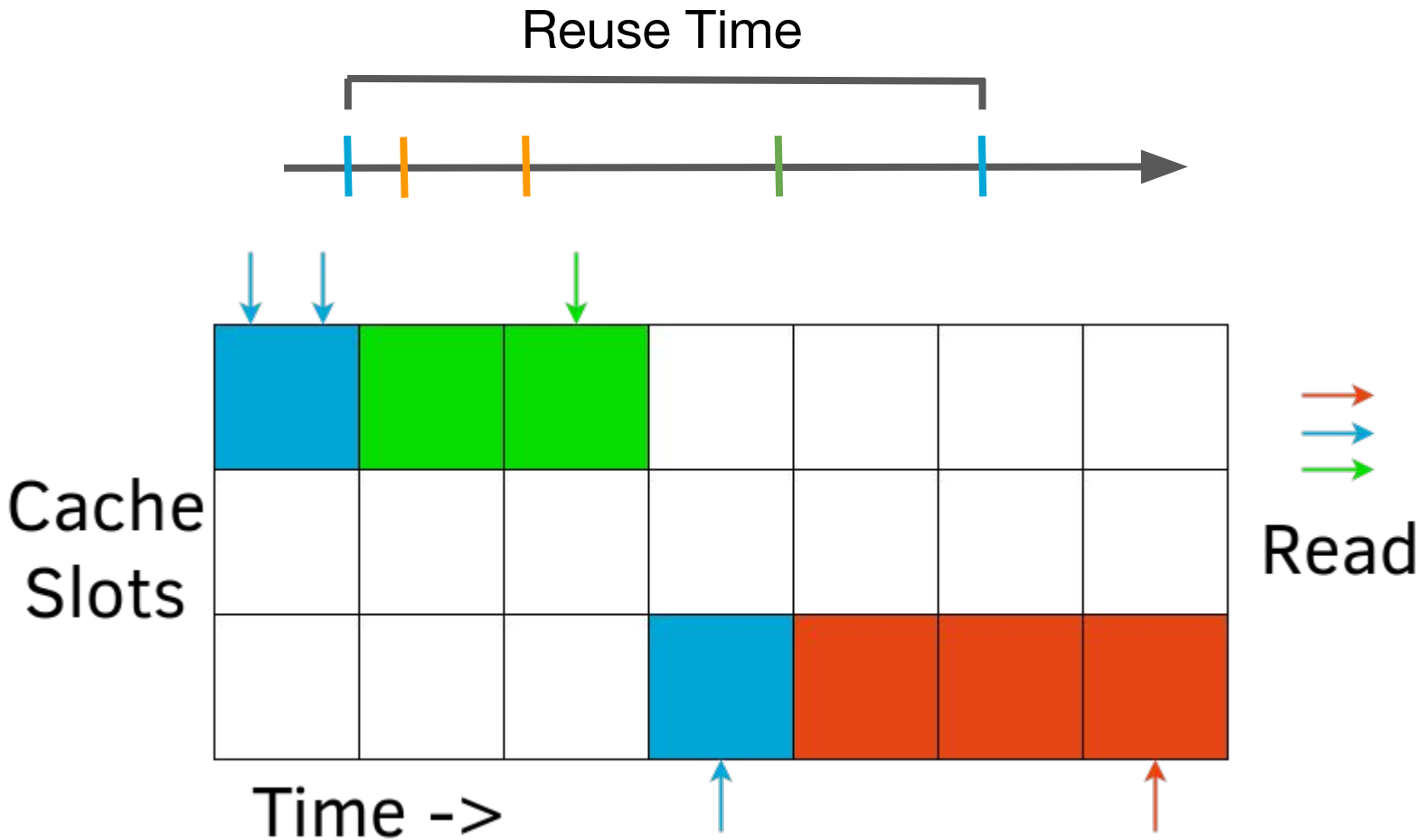


Design of the *Approximate Read Density* Cache Policy

Reuse time

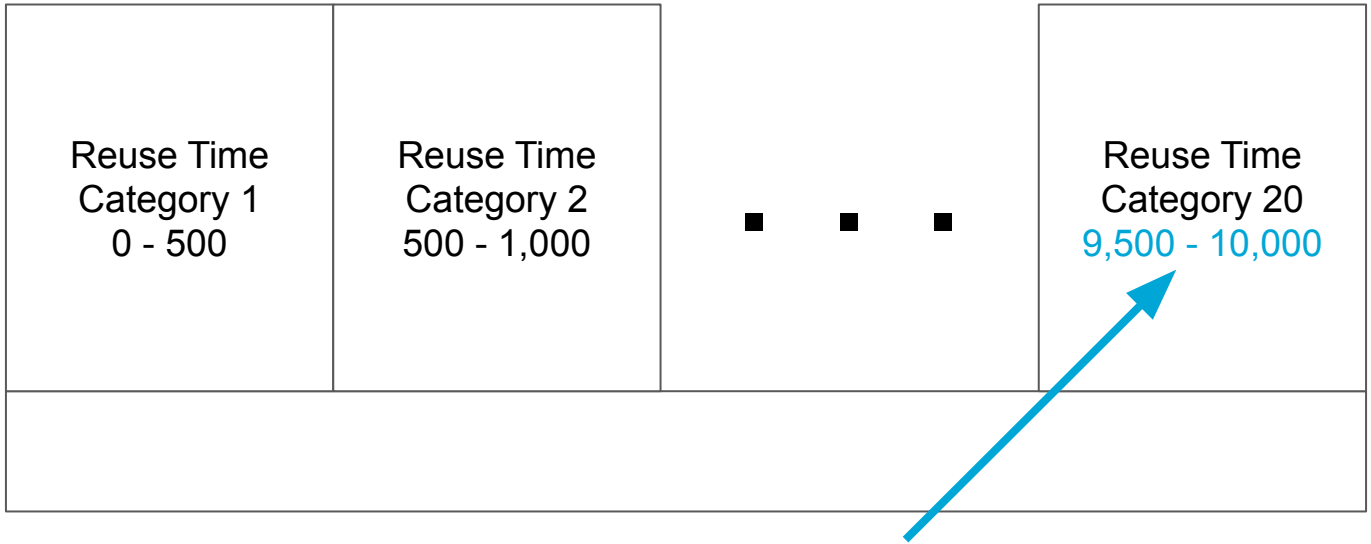


Design of the *Approximate Read Density* Cache Policy



Design of the *Approximate Read Density* Cache Policy

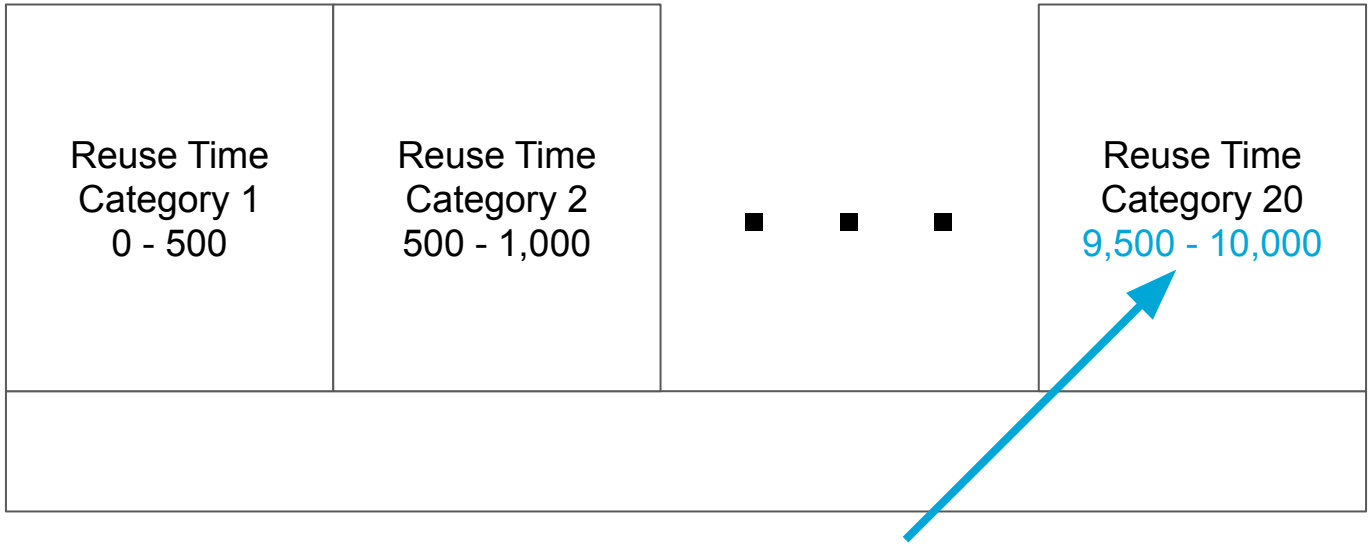
Reuse Time Histogram



Keeps track of reads occurring in this time period since the last read

Design of the *Approximate Read Density* Cache Policy

Reuse Time Histogram

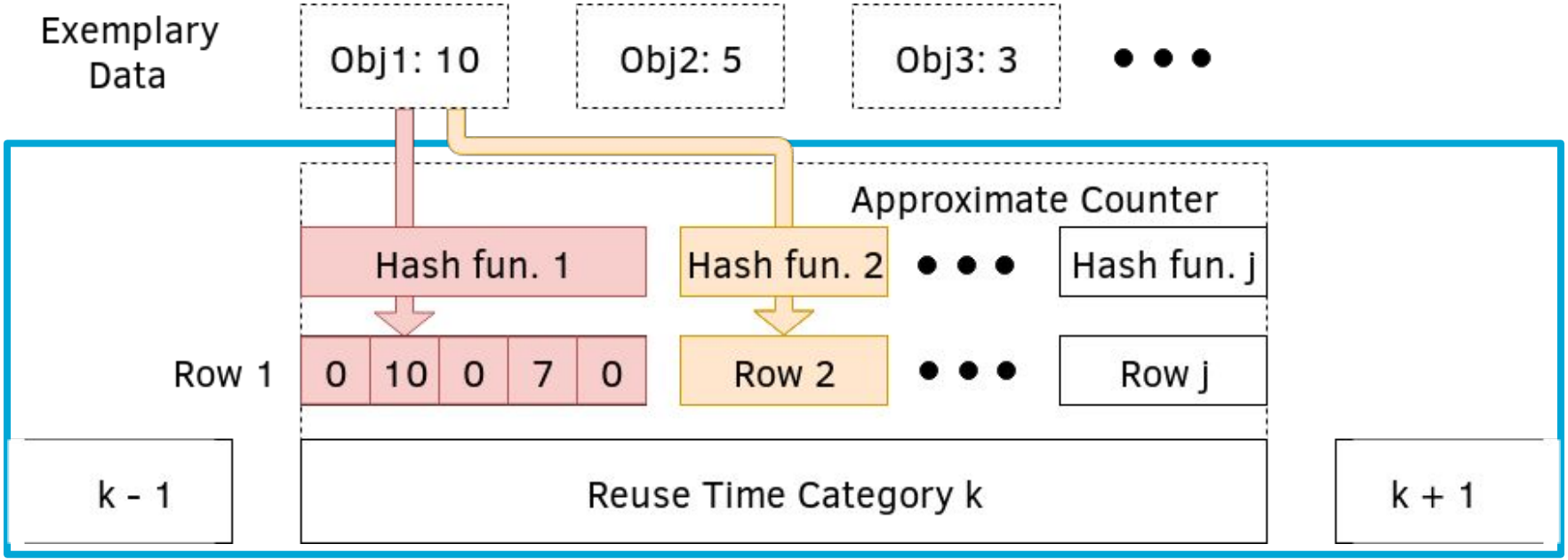


Keeps track of reads occurring in this time period since the last read

Costly to store a histogram for every object

Design of the *Approximate Read Density* Cache Policy

Approximate Histogram



Approx. 10x less space

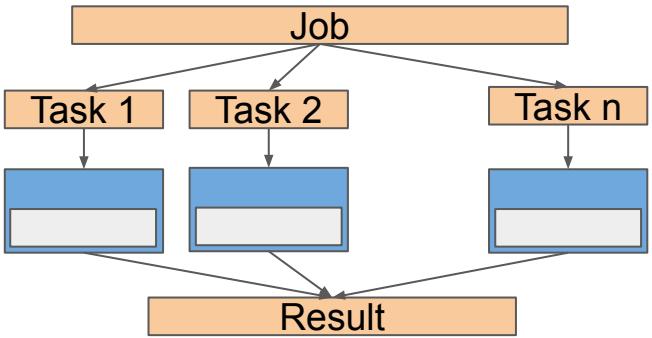
Design of the *Approximate Read Density* Cache Policy

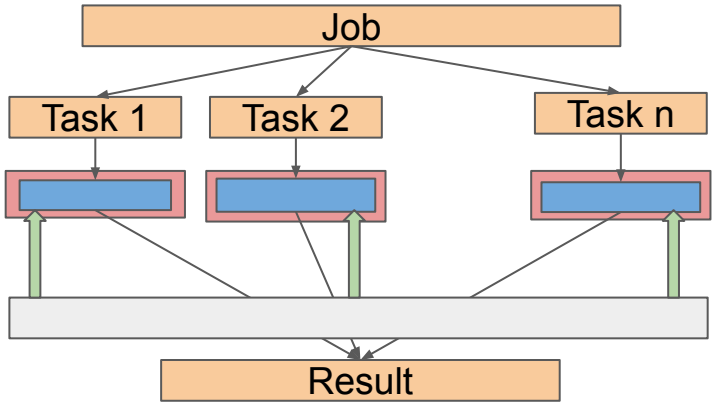
Comparison to LHD

- No special categories
  - Keeps track of history of each object
- Less space
- Does not try to evict based on predicted eviction time
  - Simpler to reason about

Experimental Setup - Infrastructure

- Simulation using simulator component of Caffeine
  - Previously used in high quality work
- Two Workloads
  - Databricks[1]
  - Yahoo Webscope 3
- From Two different System Models



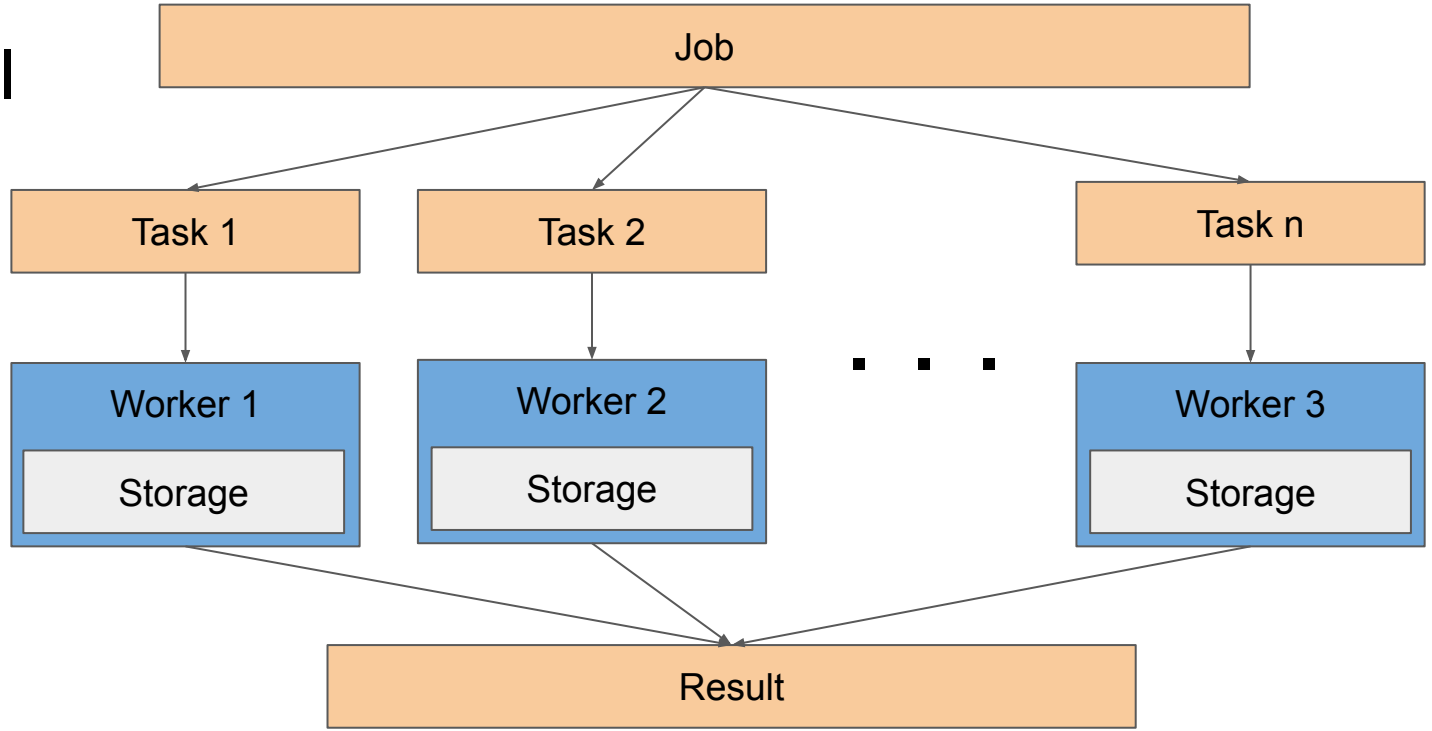


[1] Talluri, et. al., Characterization of a Big Data Storage Workload, ICPE'19 - https://atlarge-research.com/pdfs/db-big-data-characterization-icpe-2019.pdf

Model 1: Cluster based Big Data Processing

Uses local storage

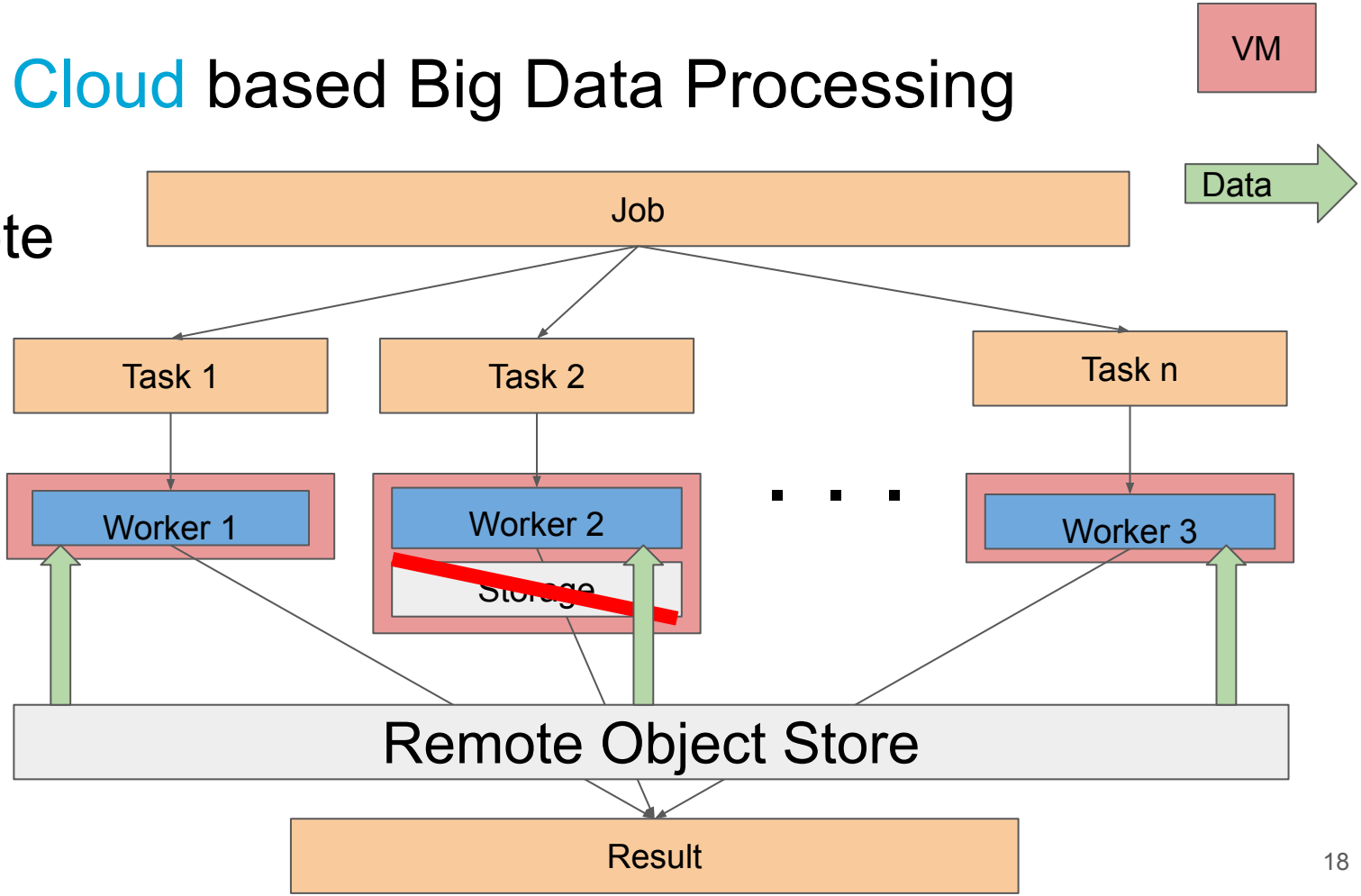
1,000s of servers



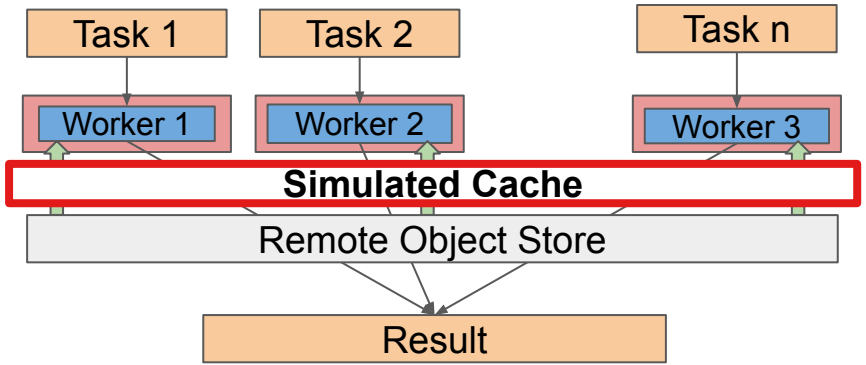
Model 2: Cloud based Big Data Processing

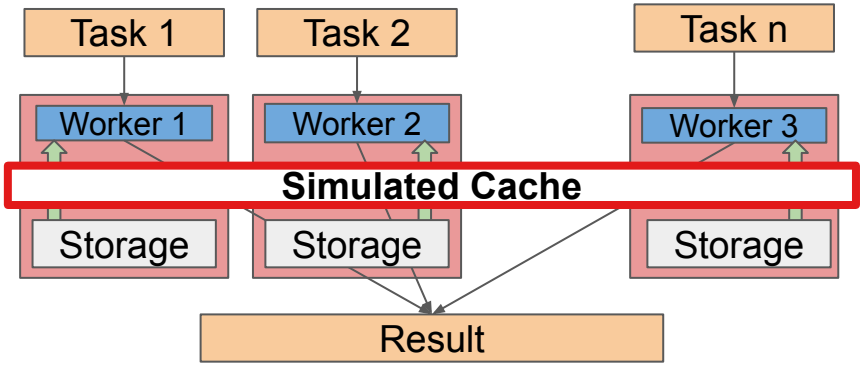
Uses remote storage

10s of ephemeral servers



Simulated Cache Placement



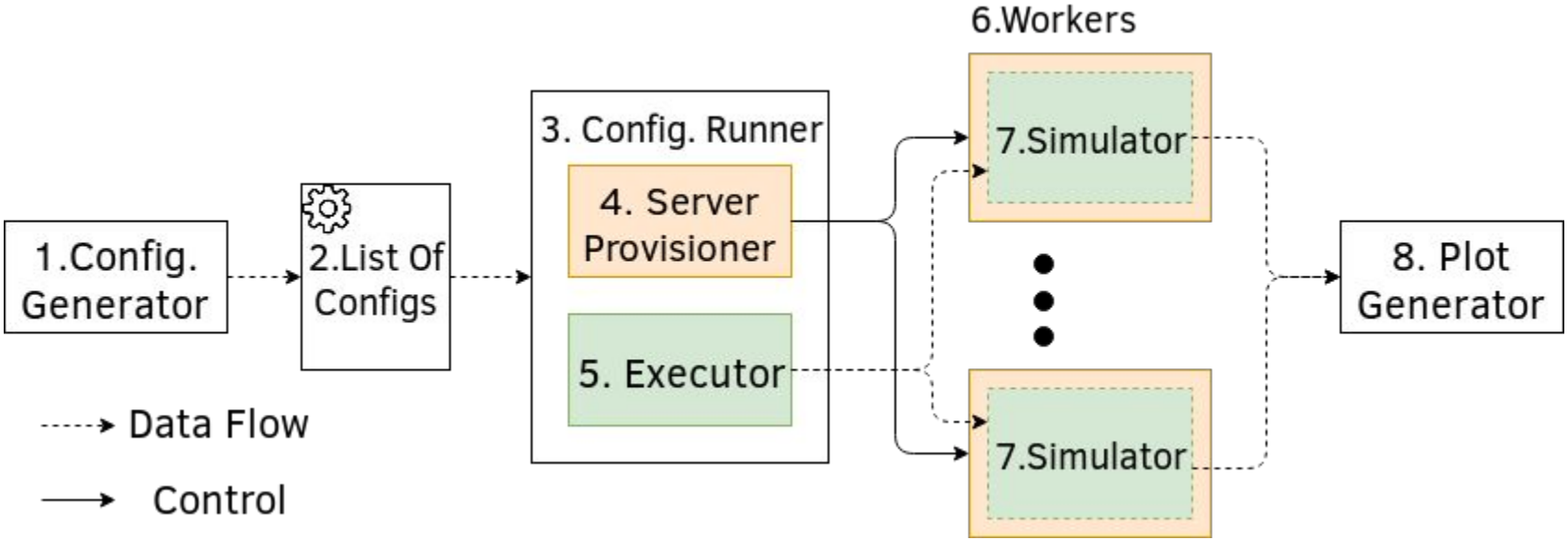


Experimental Setup - Policies

- Traditional policies
  - FIFO
  - LRU, LFU
  - LIRS - Uses reuse time
  - SLRU - Partitioned cache
  - ARC, CART - Adaptive
- Post 2010 algorithms
  - Hyperbolic, W-TinyLFU, LHD
- Representative of different kinds of policies available

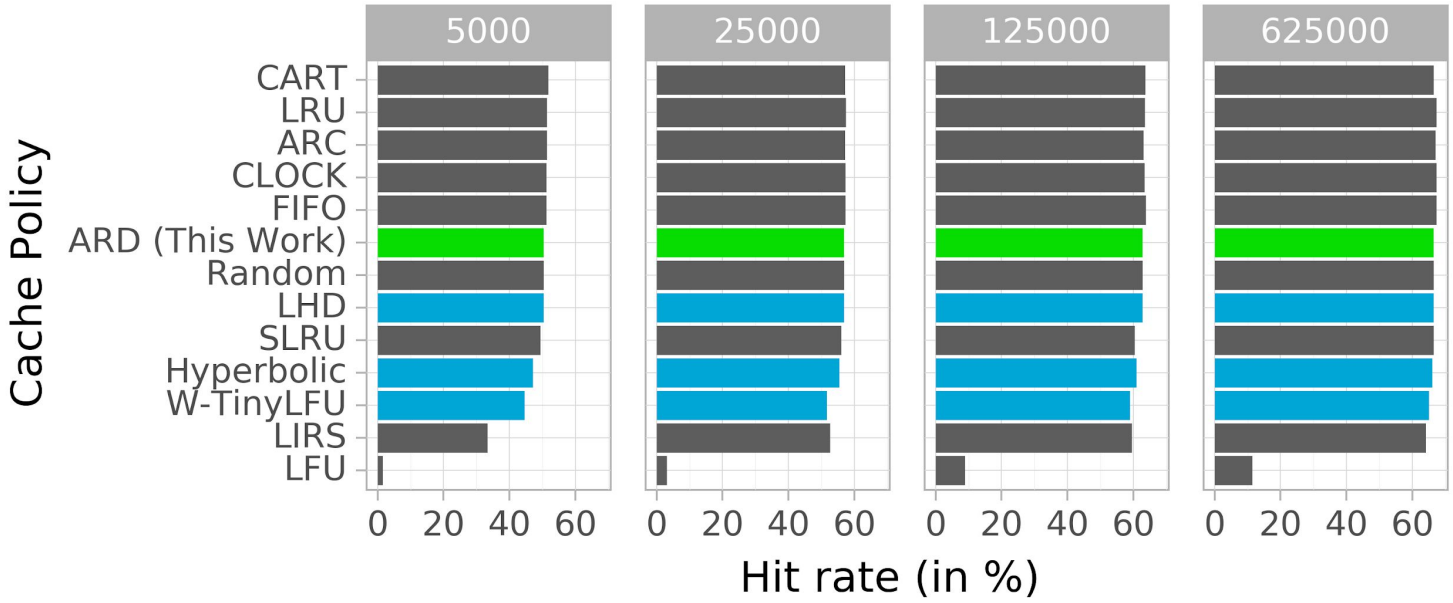
Evaluation of Cache Policies for Big Data Workloads

Using cloud computing for large scale simulation



Evaluation of Cache Policies for Big Data Workloads

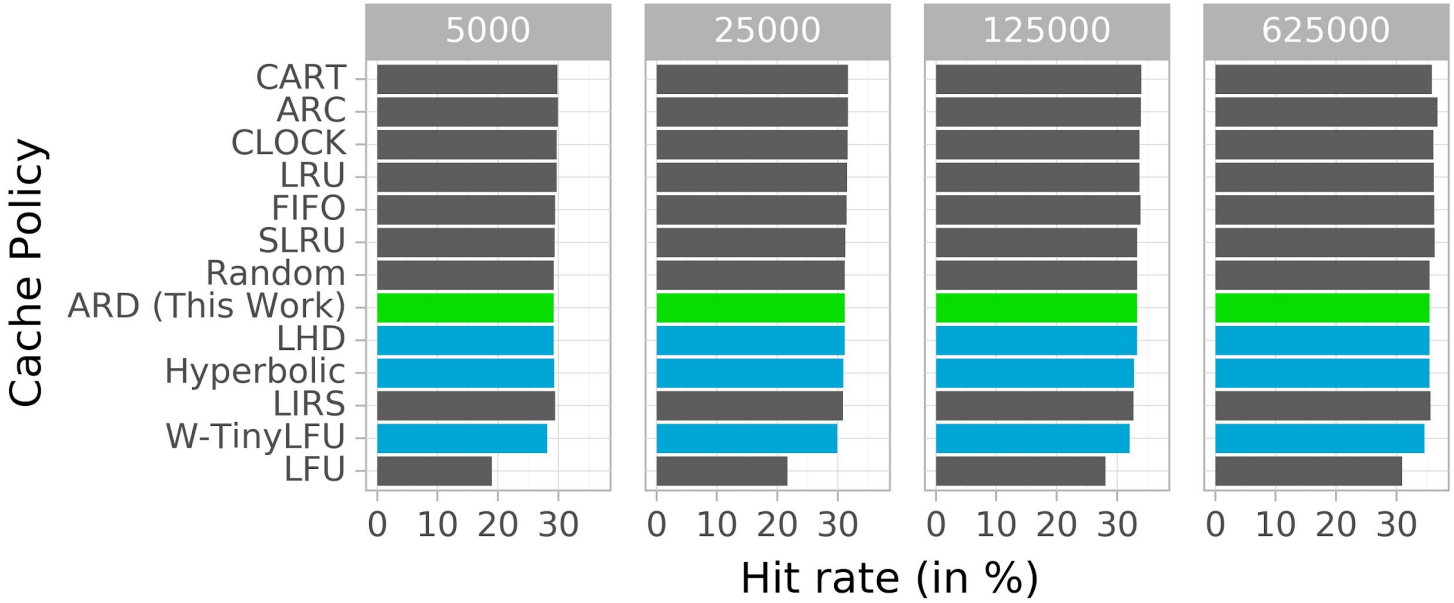
Databricks Trace



Similar performance. No improvement due to recent policies

Evaluation of Cache Policies for Big Data Workloads

Yahoo Trace



Similar performance. No improvement due to recent policies

Take Home Message

1. First simulation-based experiments with modern models, real-world big data traces, and traditional and recent caching policies
2. Caching for big data processing has not received much attention
3. Existing policies all perform about the same

Ideas for future work:
Big data storage accesses show some repetition
Predict times of use prefetch data or even metadata
Reduces time to first byte

Questions?