IoT Big Data Processing

Distributed Streaming Classification

Albert Bifet(@abifet)

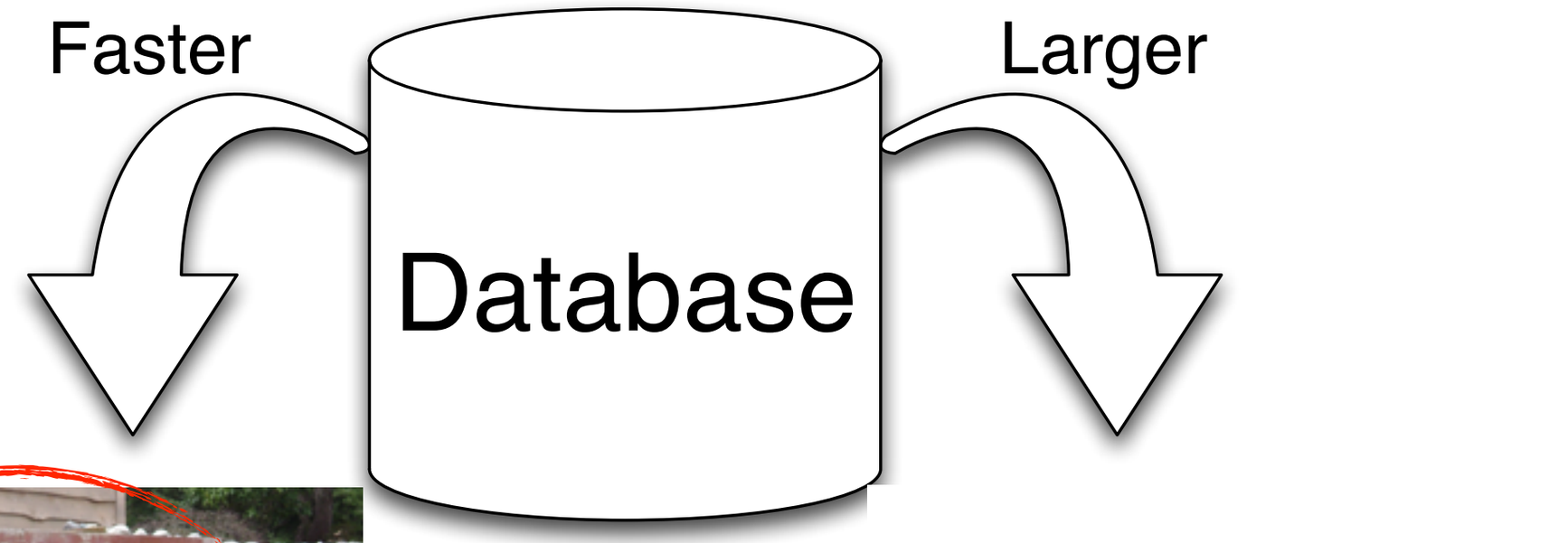
Motivation

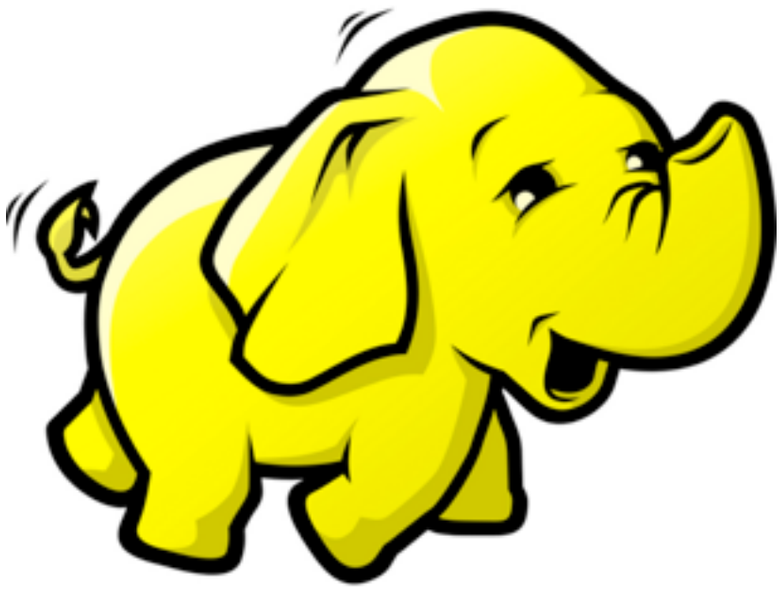
- Datasets already stored on clusters
  - Don’t want to move everything to single powerful machine
- Clusters ubiquitous and cheap (e.g., see TOP500), supercomputers expensive and monolithic
  - Clusters easily shared, leverage economy of scale
- Largest problem solvable by single machine constrained by hardware
  - How fast can you read from disk or network

Distributed Stream Processing Engines

A Tale of two Tribes

M. Stonebraker U. Çetintemel: "'One Size Fits All': An Idea Whose Time Has Come and Gone". ICDE '05

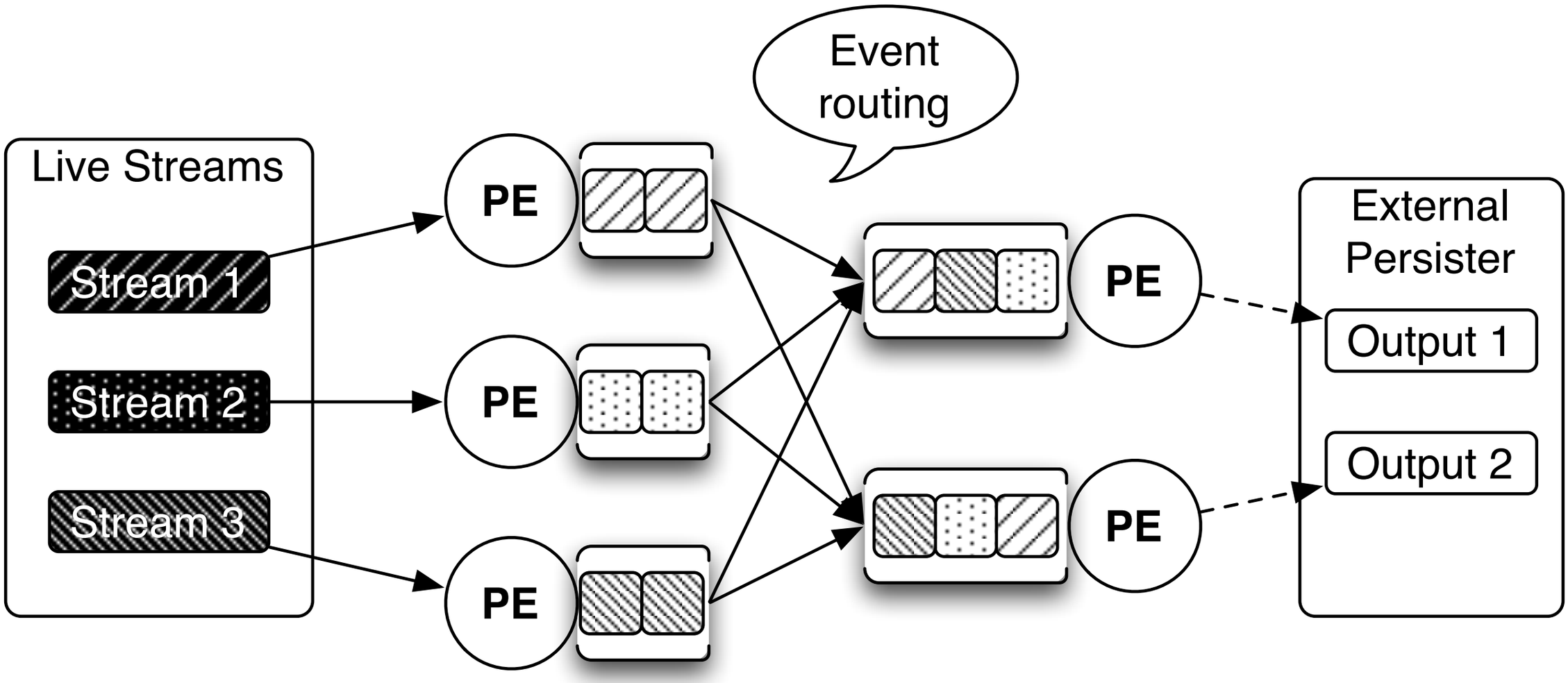




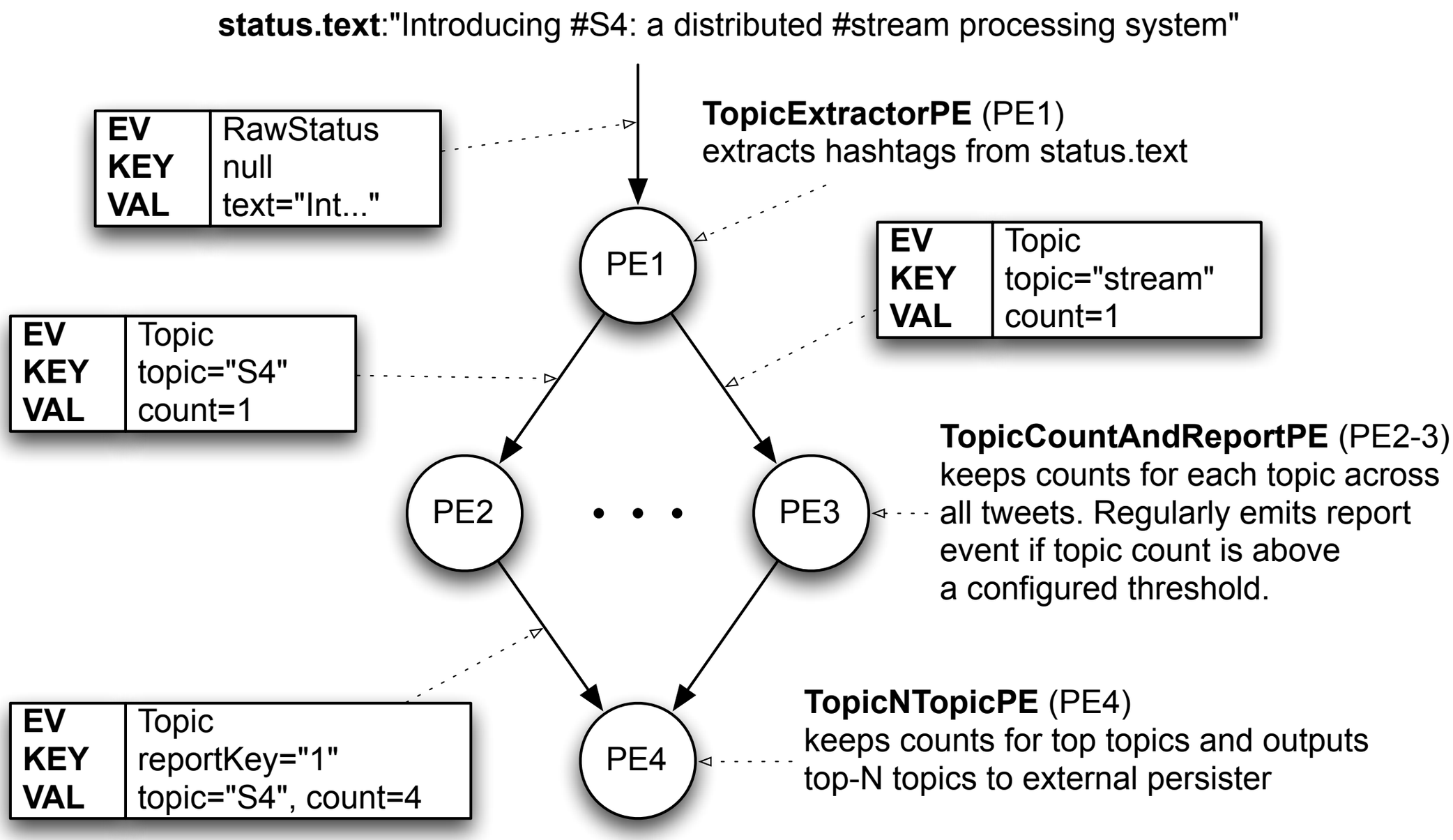
SPE Evolution

1st generation	–2003	Aurora	Abadi et al., "Aurora: a new model and architecture for data stream management," VLDB Journal, 2003
	–2004	STREAM	Arasu et al., "STREAM: The Stanford Data Stream Management System," Stanford InfoLab, 2004.
2nd generation	–2005	Borealis	Abadi et al., "The Design of the Borealis Stream Processing Engine," in CIDR '05
	–2006	SPC	Amini et al., "SPC: A Distributed, Scalable Platform for Data Mining," in DMSSP '06
	–2008	SPADE	Gedik et al., "SPADE: The System S Declarative Stream Processing Engine," in SIGMOD '08
3rd generation	–2010	S4	Neumeyer et al., "S4: Distributed Stream Computing Platform," in ICDMW '10
	–2011	Storm	http://storm.apache.org
	–2013	Samza	http://samza.incubator.apache.org

Actors Model

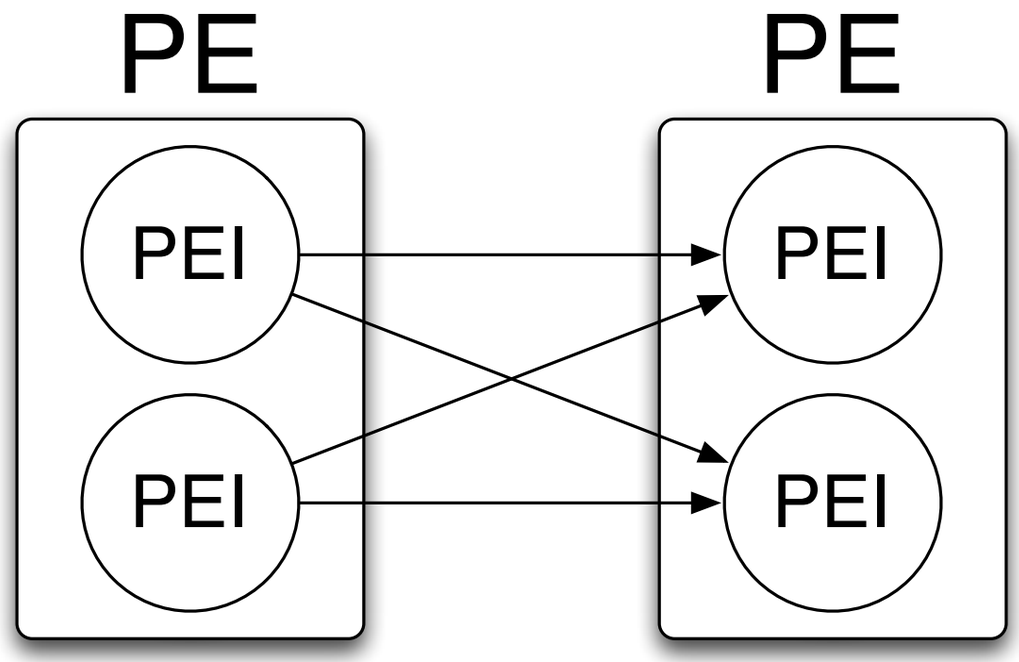


S4 Example


status.text:"Introducing #S4: a distributed #stream processing system"
EV RawStatus
KEY null
VAL text="Int..."
TopicExtractorPE (PE1)
extracts hashtags from status.text
PE1
EV Topic
KEY topic="stream"
VAL count=1
EV Topic
KEY topic="S4"
VAL count=1
PE2
PE3
TopicCountAndReportPE (PE2-3)
keeps counts for each topic across all tweets. Regularly emits report event if topic count is above a configured threshold.
EV Topic
KEY reportKey="1"
VAL topic="S4", count=4
PE4
TopicNTopicPE (PE4)
keeps counts for top topics and outputs top-N topics to external persister

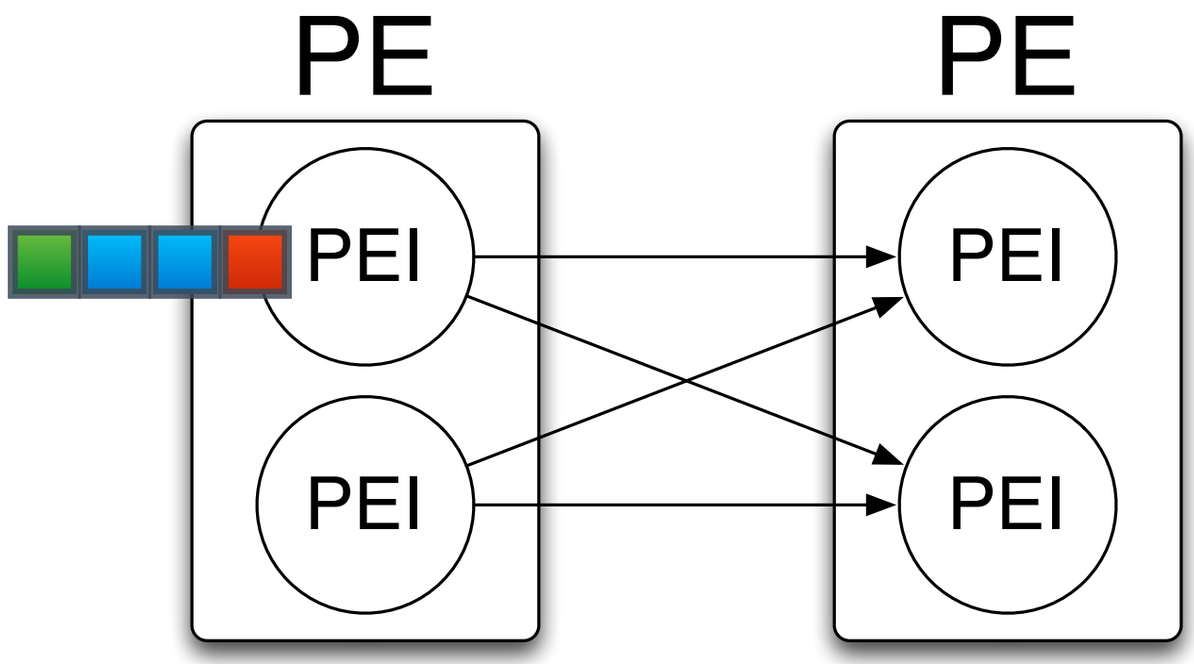
Groupings

- Key Grouping (hashing)
- Shuffle Grouping (round-robin)
- All Grouping (broadcast)



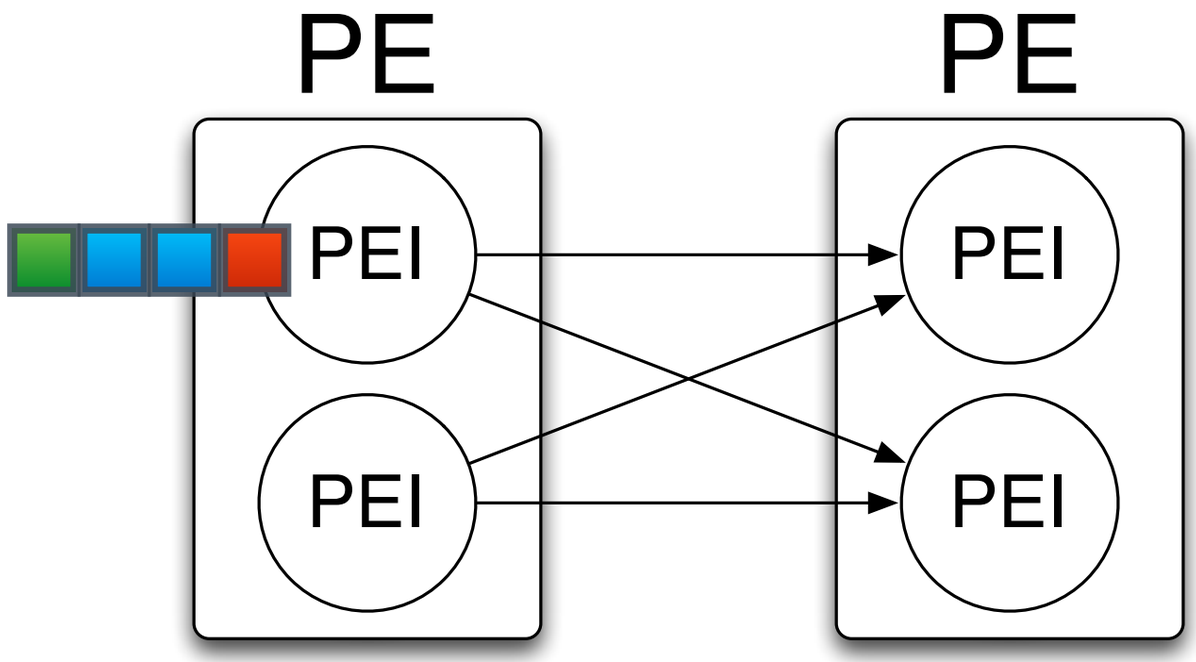
Groupings

- **Key Grouping (hashing)**
- Shuffle Grouping (round-robin)
- All Grouping (broadcast)



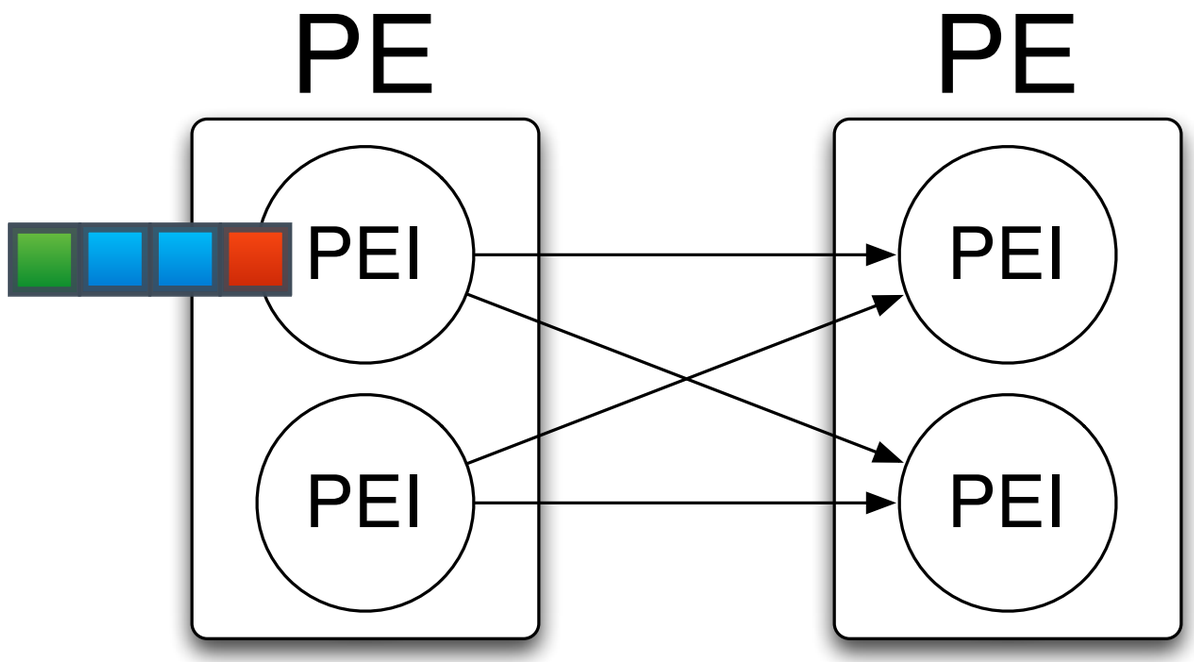
Groupings

- Key Grouping (hashing)
- **Shuffle Grouping (round-robin)**
- All Grouping (broadcast)



Groupings

- Key Grouping (hashing)
- Shuffle Grouping (round-robin)
- **All Grouping (broadcast)**

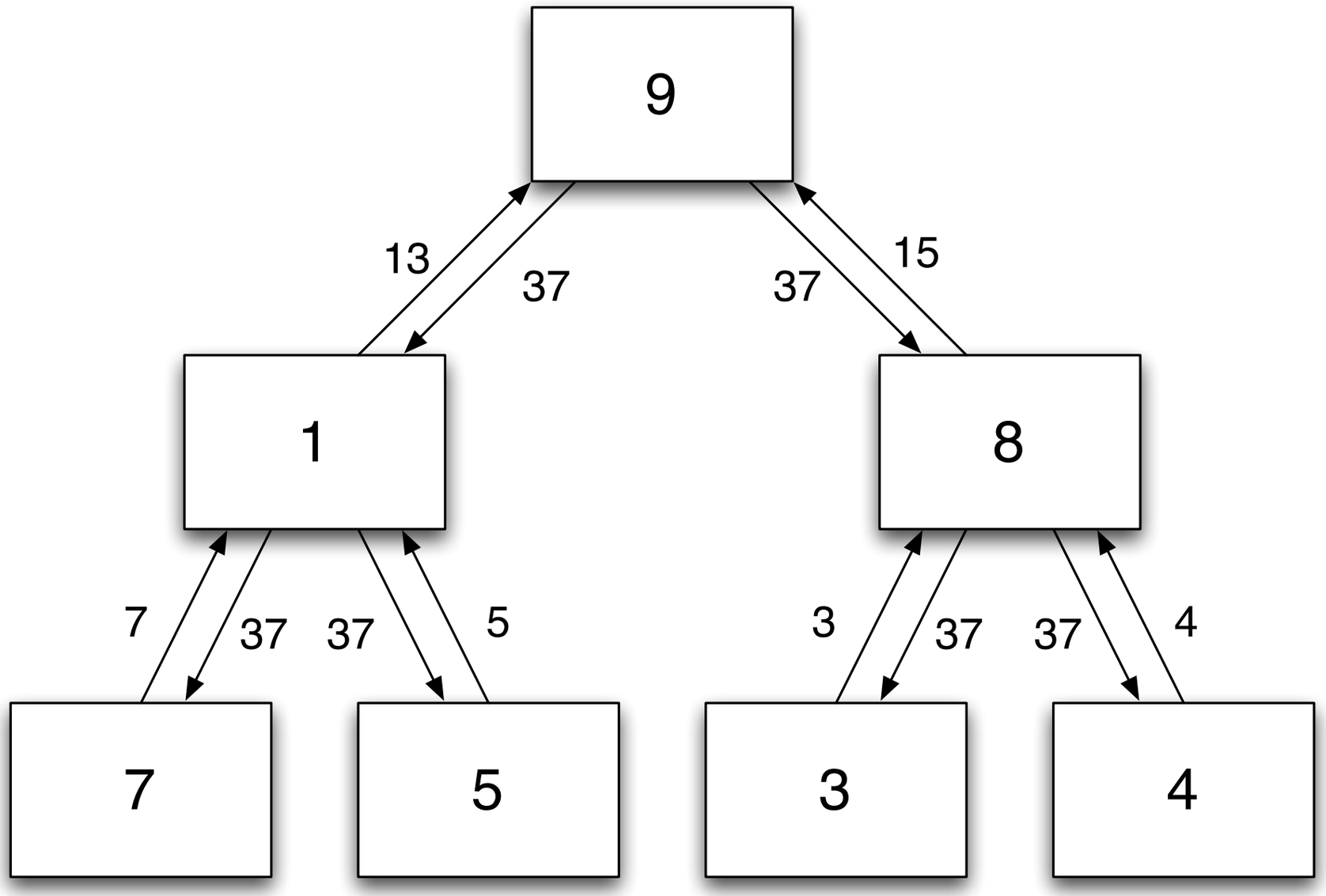


Classification

Hadoop AllReduce

A. Agarwal, O. Chapelle, M. Dudík, J. Langford: “A Reliable Effective Terascale Linear Learning System”. JMLR (2014)

- MPI AllReduce on MapReduce
- Parallel SGD + L-BFGS
- Aggregate + Redistribute
  - Each node computes partial gradient
  - Aggregate (sum) complete gradient
  - Each node gets updated model
- Hadoop for data locality (map-only job)



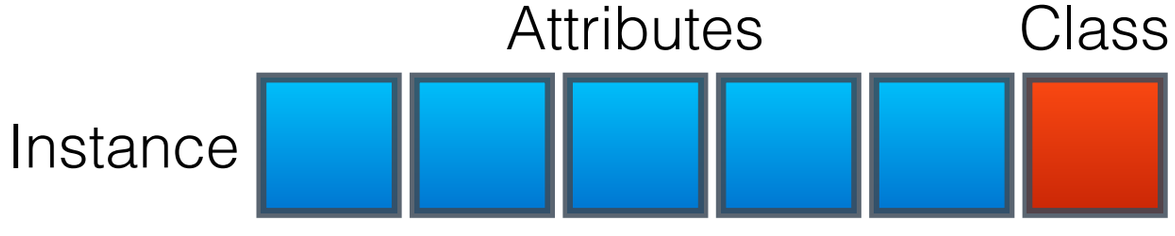
AllReduce

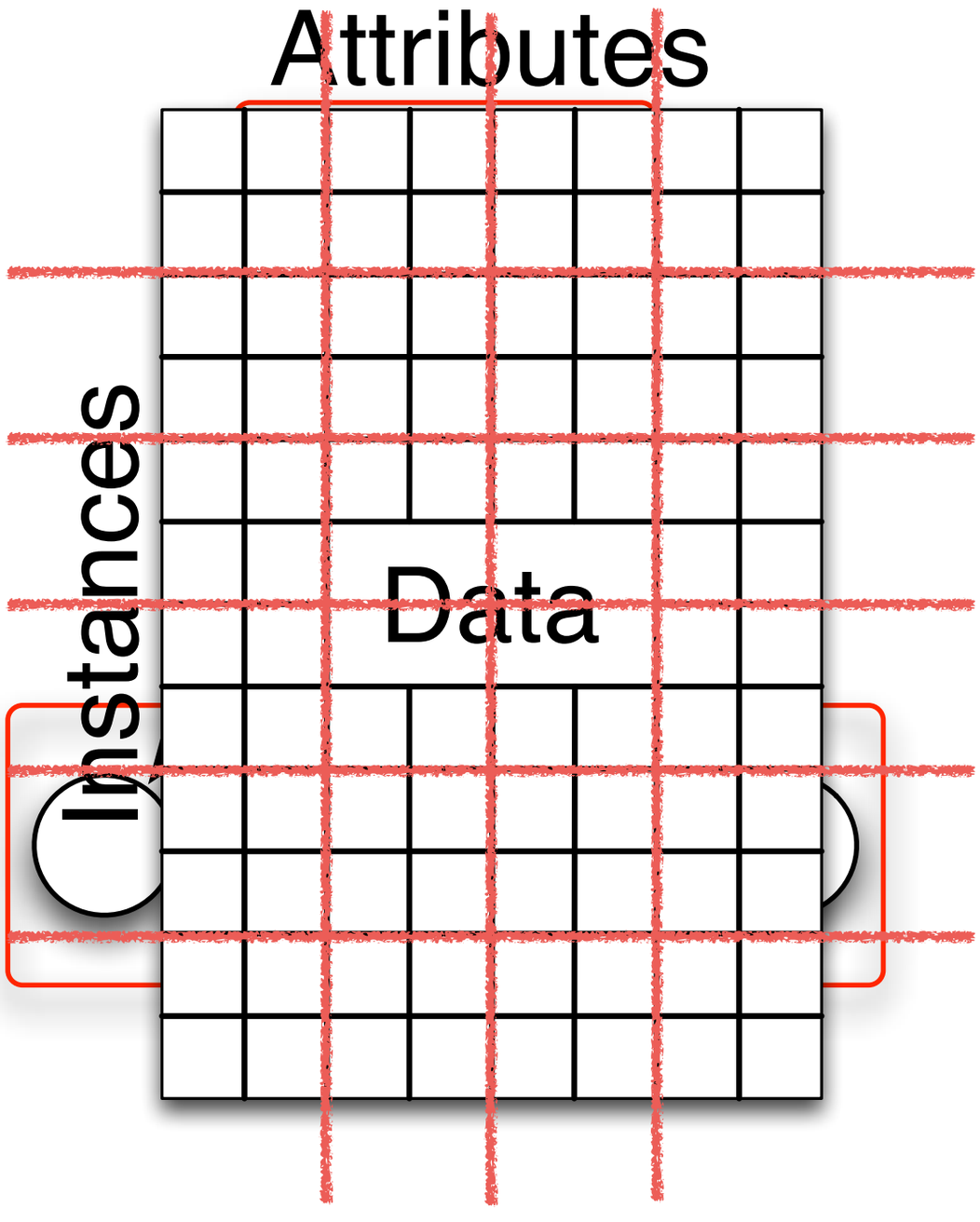
Reduction Tree

Upward = Reduce Downward = Broadcast (All)

Parallel Decision Trees

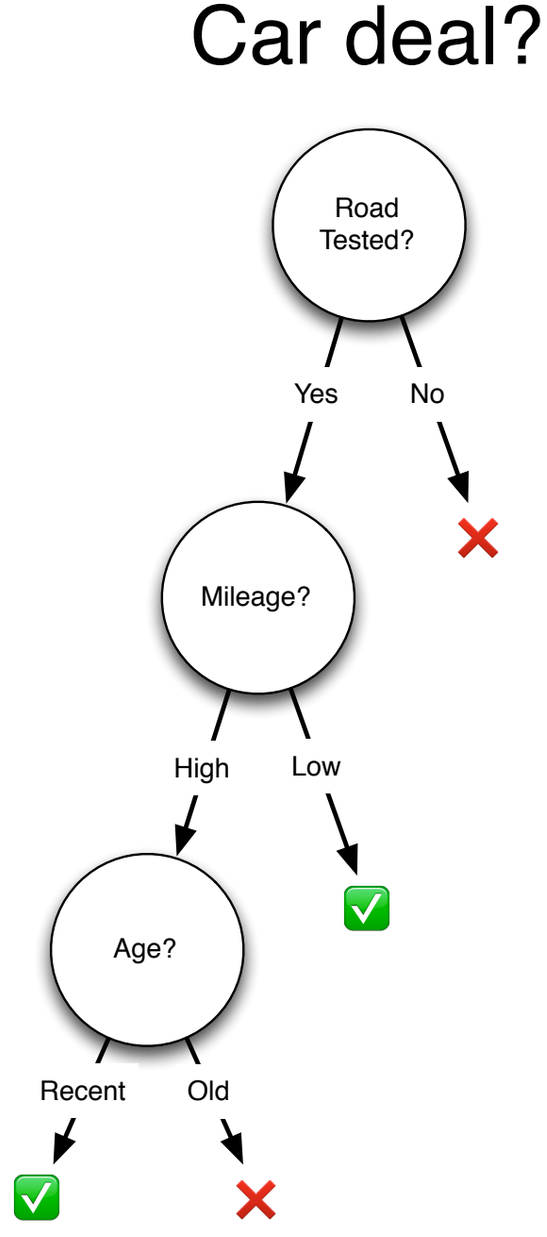
- Which kind of parallelism?
  - Task
  - Data
    - Horizontal
    - Vertical





Decision Tree

- Each node tests a features
- Each branch represents a value
- Each leaf assigns a class
- Greedy recursive induction
  - Sort all examples through tree
  - x_i = most discriminative attribute
  - New node for x_i, new branch for each value, leaf assigns majority class
  - Stop if no error | limit on #instances



Very Fast Decision Tree

Pedro Domingos, Geoff Hulten: “Mining high-speed data streams”. KDD ’00

- AKA, Hoeffding Tree
- A small sample can often be enough to choose a near optimal decision
- Collect sufficient statistics from a small set of examples
- Estimate the merit of each alternative attribute
- Choose the sample size that allows to differentiate between the alternatives

Leaf Expansion

- When should we expand a leaf?
- Let x_1 be the most informative attribute, x_2 the second most informative one
- Is x_1 a stable option?
- Hoeffding bound
  - Split if $G(x_1) - G(x_2) > \varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$

HT Induction

HT(*Stream*, δ)

1 ▷ Let HT be a tree with a single leaf(root)
2 ▷ Init counts n_{ijk} at root
3 **for** each example (x, y) in Stream
4 **do** HTGROW$((x, y), HT, \delta)$

HTGROW$((x, y), HT, \delta)$

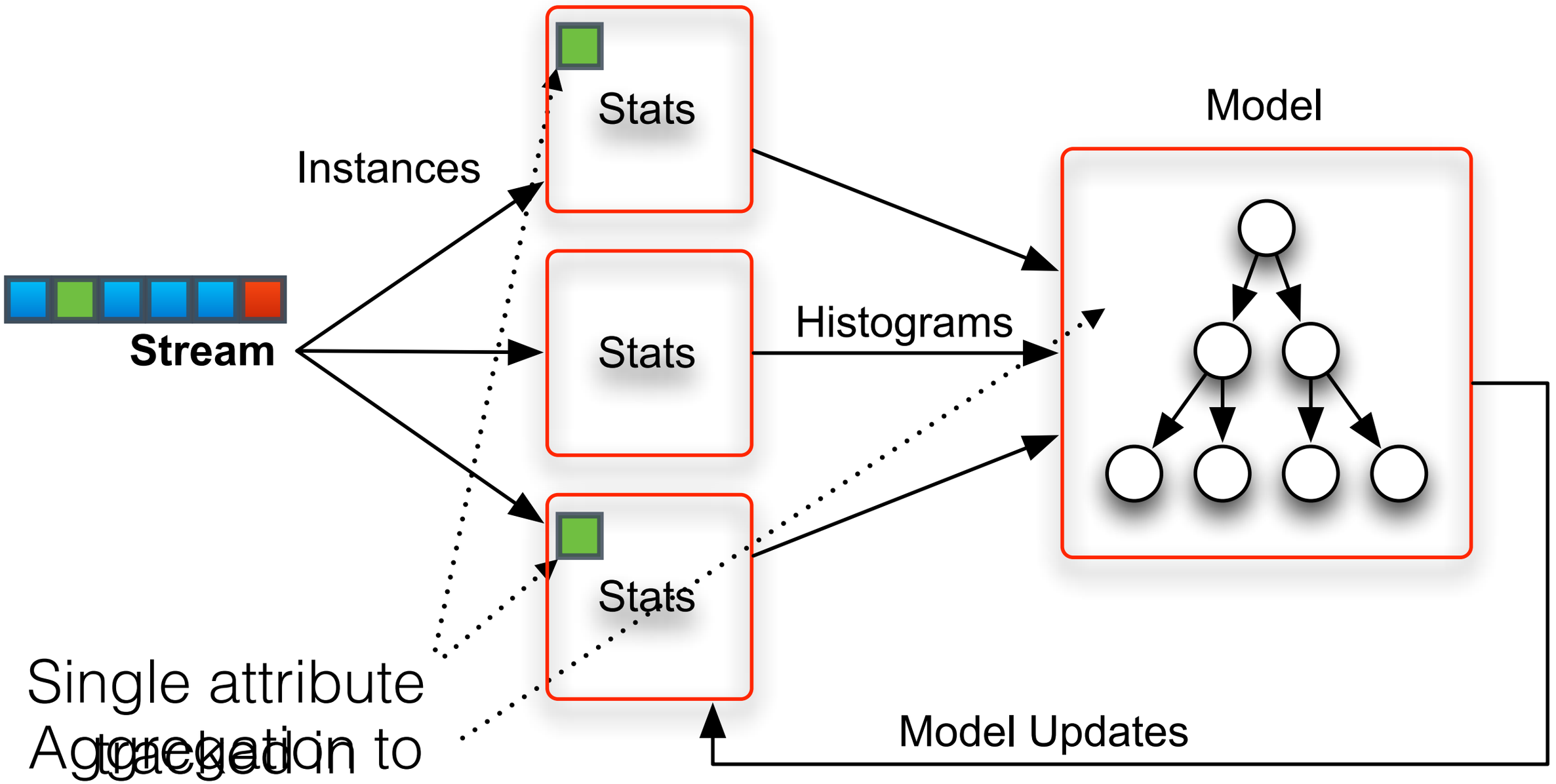
1 ▷ Sort (x, y) to leaf l using HT
2 ▷ Update counts n_{ijk} at leaf l
3 **if** examples seen so far at l are not all of the same class
4 **then** ▷ Compute G for each attribute
5 **if** G(Best Attr.)$-G$(2nd best) $> \sqrt{\frac{R^2 \ln 1/\delta}{2n}}$
6 **then** ▷ Split leaf on best attribute
7 **for** each branch
8 **do** ▷ Start new leaf and initiliatize counts

Properties

- Number of examples to expand node depends only on Hoeffding bound (ε decreases with √n)
- Low variance model (stable decisions with statistical support)
- Low overfitting (examples processed only once, no need for pruning)
- Theoretical guarantees on error rate with high probability
  - Hoeffding algorithms asymptotically close to batch learner. Expected disagreement δ/p (p = probability instance falls into a leaf)
- Ties: broken when ε < τ even if ΔG < ε

Horizontal Partitioning

Y. Ben-Haim, E. Tom-Tov: “A Streaming Parallel Decision Tree Algorithm”. JMLR (2010)

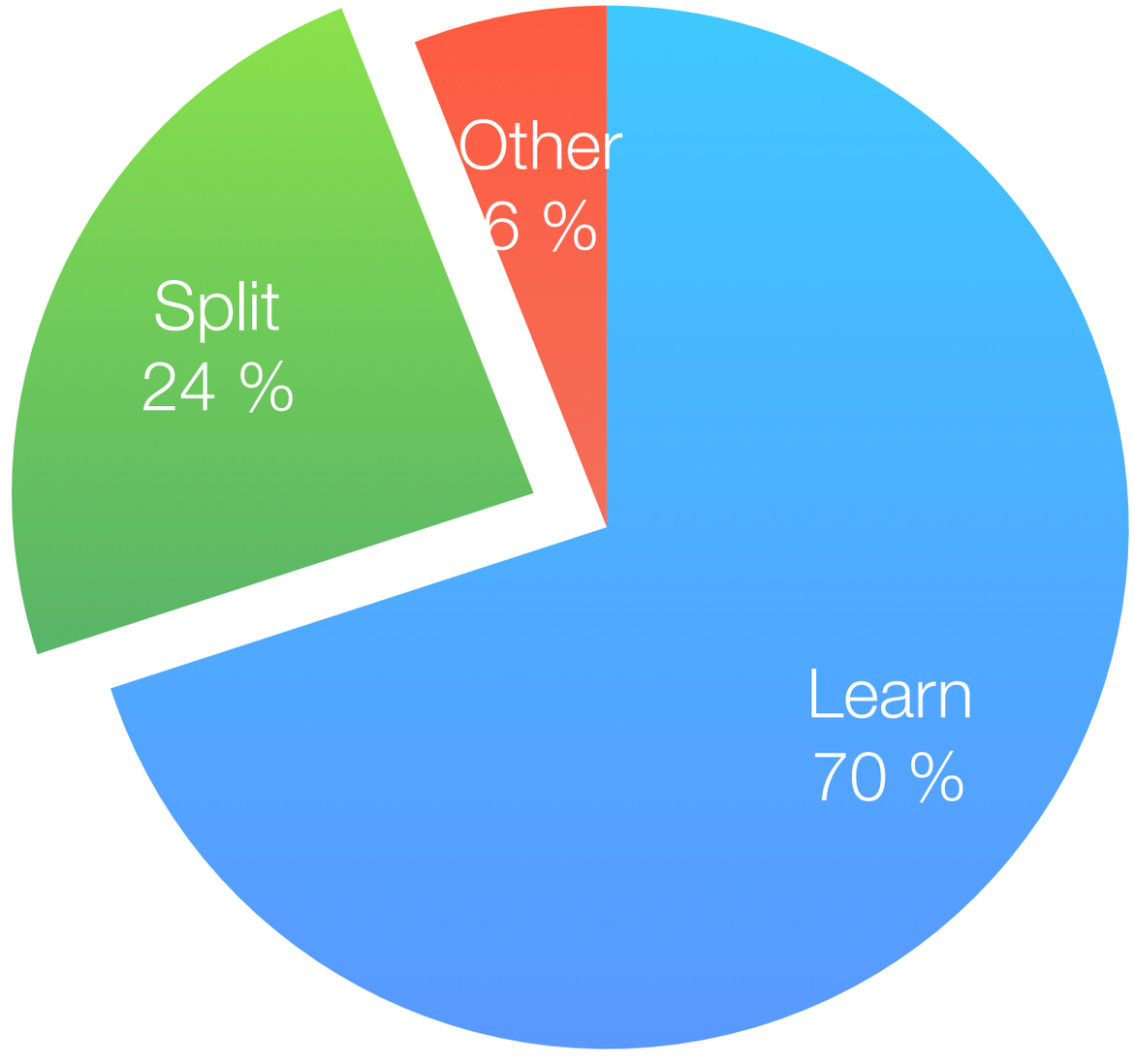


Single attribute tracked in each node

Aggregation to compute splits

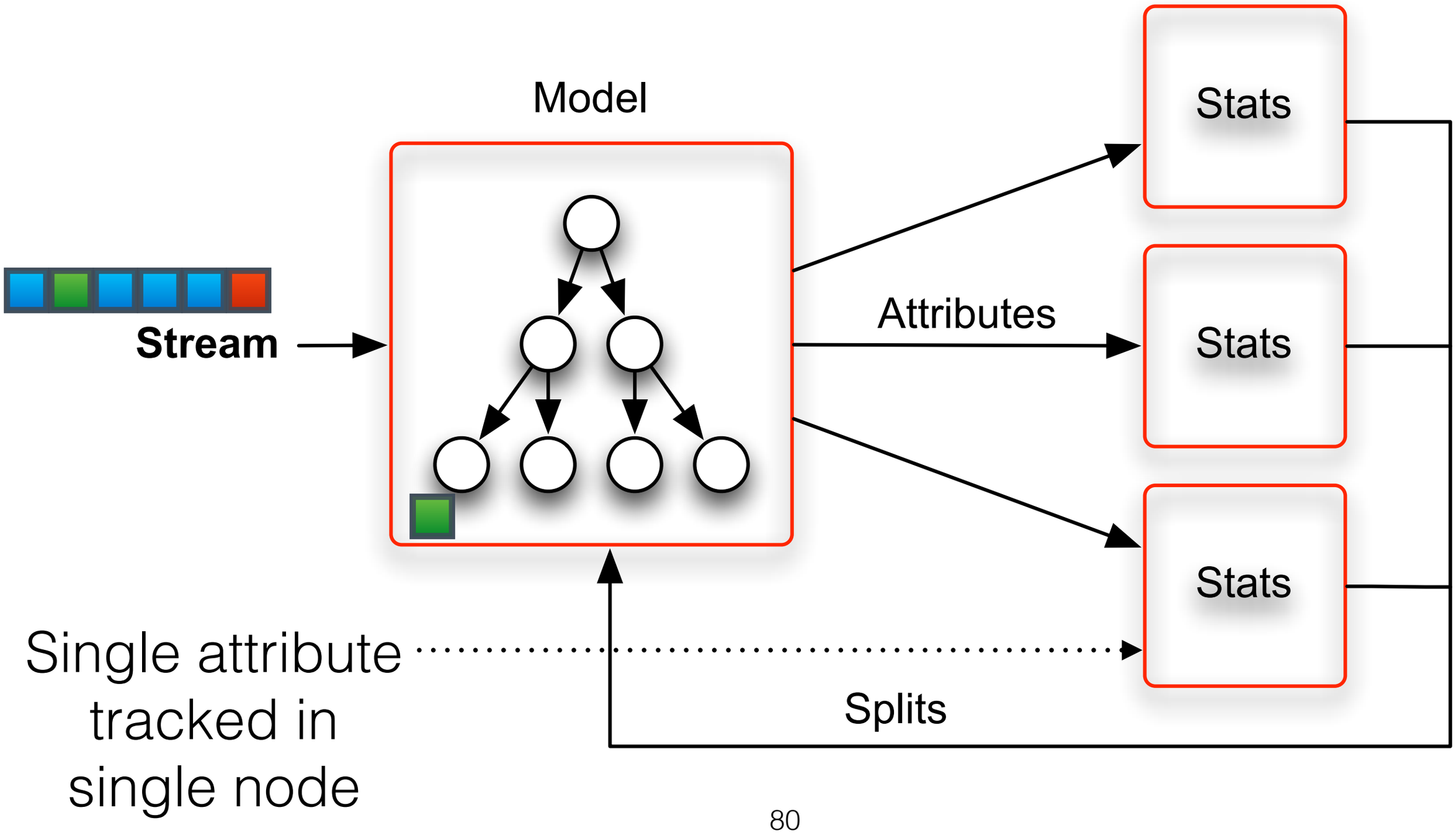
Hoeffding Tree Profiling

Training time for 100 nominal + 100 numeric attributes

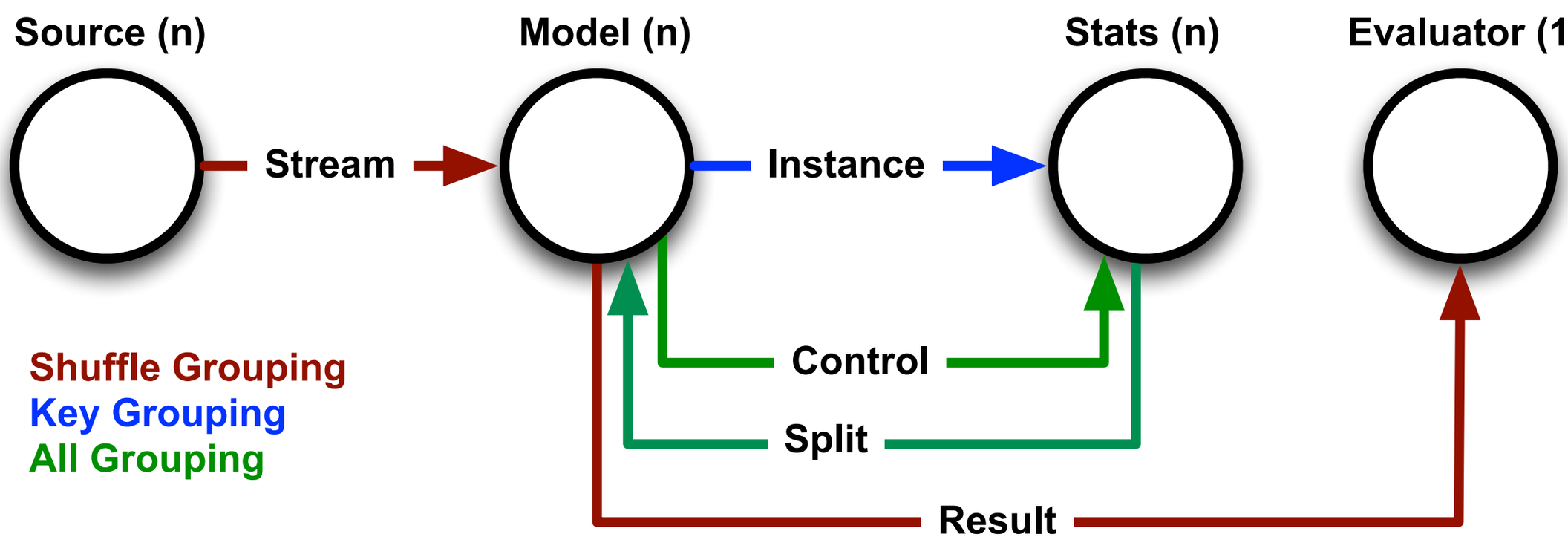


Vertical Partitioning

A. Murdopo, A. Bifet, G. De Francisci Morales, N. Kourtellis: “VHT: Vertical Hoeffding Tree”. Working paper (2014)



Vertical Hoeffding Tree



Advantages of Vertical Parallelism

- High number of attributes => high level of parallelism (e.g., documents)

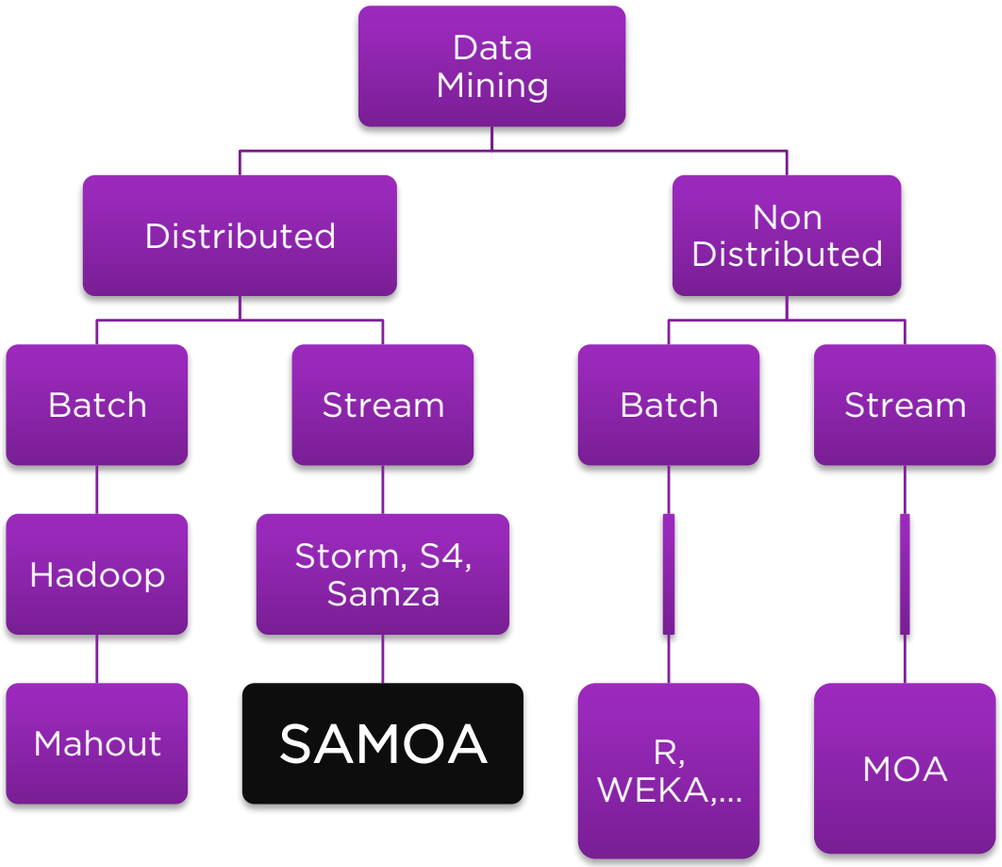
- vs. task parallelism
  - Parallelism observed immediately

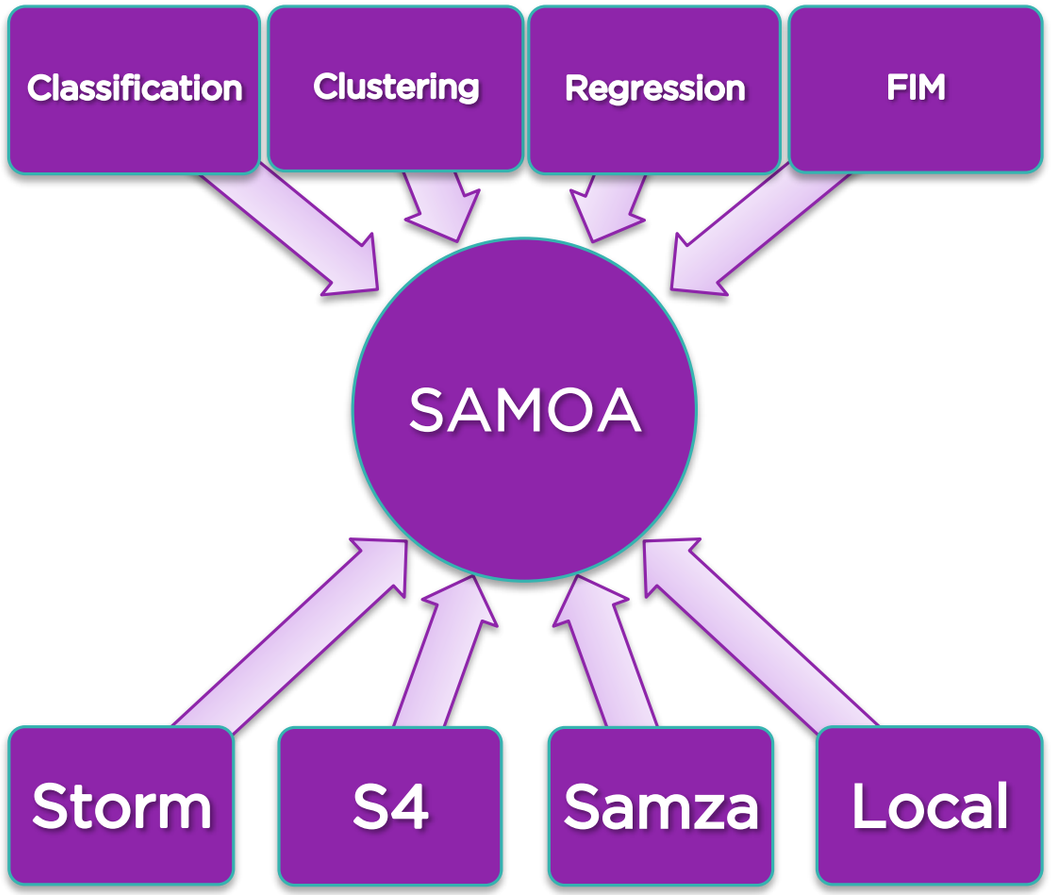
- vs. horizontal parallelism
  - Reduced memory usage (no model replication)
  - Parallelized split computation

http://samoa-project.net

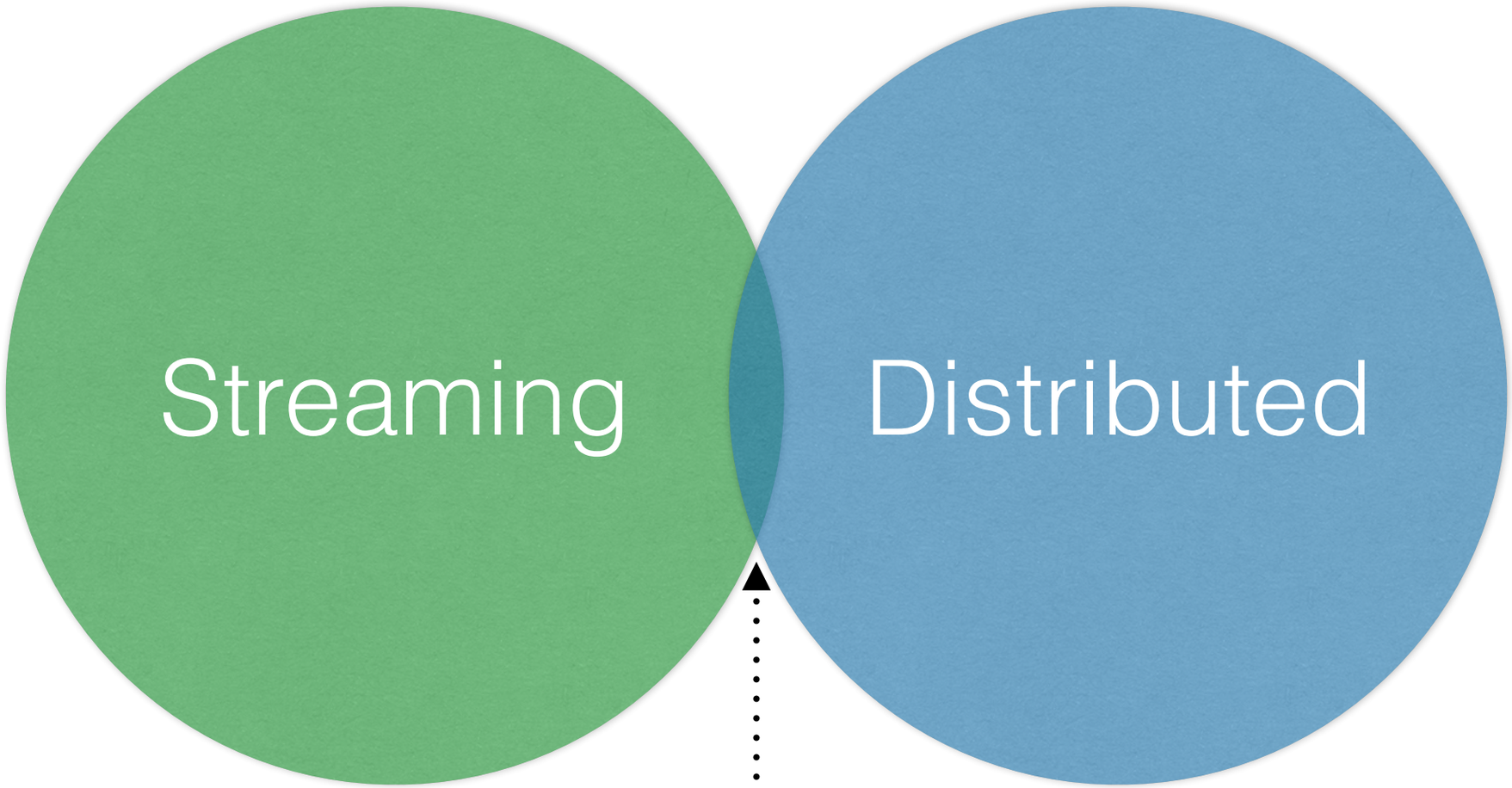
SAMOA

G. De Francisci Morales, A. Bifet: “SAMOA: Scalable Advanced Massive Online Analysis”. JMLR (2014)





Vision



Big Data Stream Mining