PARMA: A Parallel Randomized Algorithm for Approximate Association Rules Mining in MapReduce

Date : 2013/10/16

Source : CIKM'12

Outline

- **Introduction**
- Approach
- Experiment
- Conclusion

Association Rules

- Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} $\rightarrow$ {Beer},
{Milk, Bread} $\rightarrow$ {Eggs,Coke},
{Beer, Bread} $\rightarrow$ {Milk},

Itemset

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- I={Bread, Milk, Diaper, Beer, Eggs, Coke}
- Itemsets
  - 1-itemsets: {Beer}, {Milk}, {Bread}, …
  - 2-itemsets: {Bread, Milk}, {Bread, Beer}, …
  - 3-itemsets: {Milk, Eggs, Coke}, {Bread, Milk, Diaper},…
- t_1 contains {Bread, Milk}, but doesn't contain {Bread, Beer}

Frequent Itemset

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Support count : $\sigma(X)$**
  - Frequency of occurrence of an itemset X
  - $\sigma(X) = |\{t_i \mid X \subseteq t_i, t_i \in T\}|$
  - E.g. $\sigma(\{Milk, Bread, Diaper\}) = 2$
- **Support**
  - Fraction of transactions that contain an itemset X
  - $s(X) = \sigma(X) / |T|$
  - E.g. $s(\{Milk, Bread, Diaper\}) = 2/5$
- **Frequent Itemset**
  - An itemset X $s(X) \geq$ *minsup*

Association Rule

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Association Rule**
  - $X \rightarrow Y$, where X and Y are itemsets
  - Example: {Milk, Diaper} $\rightarrow$ {Beer}
- **Rule Evaluation Metrics**
  - Support
    - Fraction of transactions that contain both X and Y
    - $s(X\rightarrow Y)= \sigma(X\cup Y)/|T|$
  - Confidence
    - How often items in Y appear in the transactions that contain X
    - $c(X\rightarrow Y)= \sigma(X\cup Y)/ \sigma(X)$

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \{\text{Beer}\}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup*
  - confidence ≥ *minconf*

Goal

- Because :
  - Number of transactions ⇧
  - Cost of the existing algorithm, e.g. Apriori, FP-Tree ⇧
  - What can we do in big data ?
    - Sampling
    - Parallel
- Goal :
  - A MapReduce algorithm for discovering approximate collections of frequent itemsets or association rules

Outline

- Introduction
- **Approach**
- Experiment
- Conclusion

Sampling

Original Data → Sampling → Find FI in Sample

Question : Is the sample always good ?

Definition

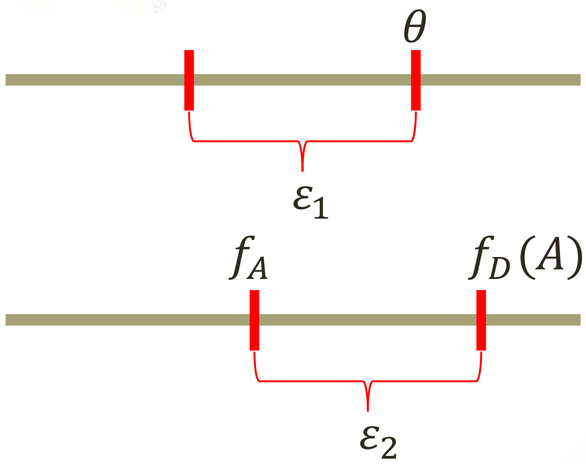
$$\mathsf{FI}(\mathcal{D},\mathcal{I},\theta) = \{(A, f_{\mathcal{D}}(A)) \ : \ A \in 2^{\mathcal{I}} \text{ and } f_{\mathcal{D}}(A) \geq \theta\}.$$

$$\mathsf{TOPK}(\mathcal{D},\mathcal{I},K) = \mathsf{FI}(\mathcal{D},\mathcal{I},f_{\mathcal{D}}^{(K)}). \qquad (1)$$

$(\varepsilon_1, \varepsilon_2)$ *approximation of* $FI(D, I, \theta)$ *is a set*

$$C = \{(A, f_A, K_A) : A \in 2^I, f_A \in K_A \subseteq [0,1]\}$$

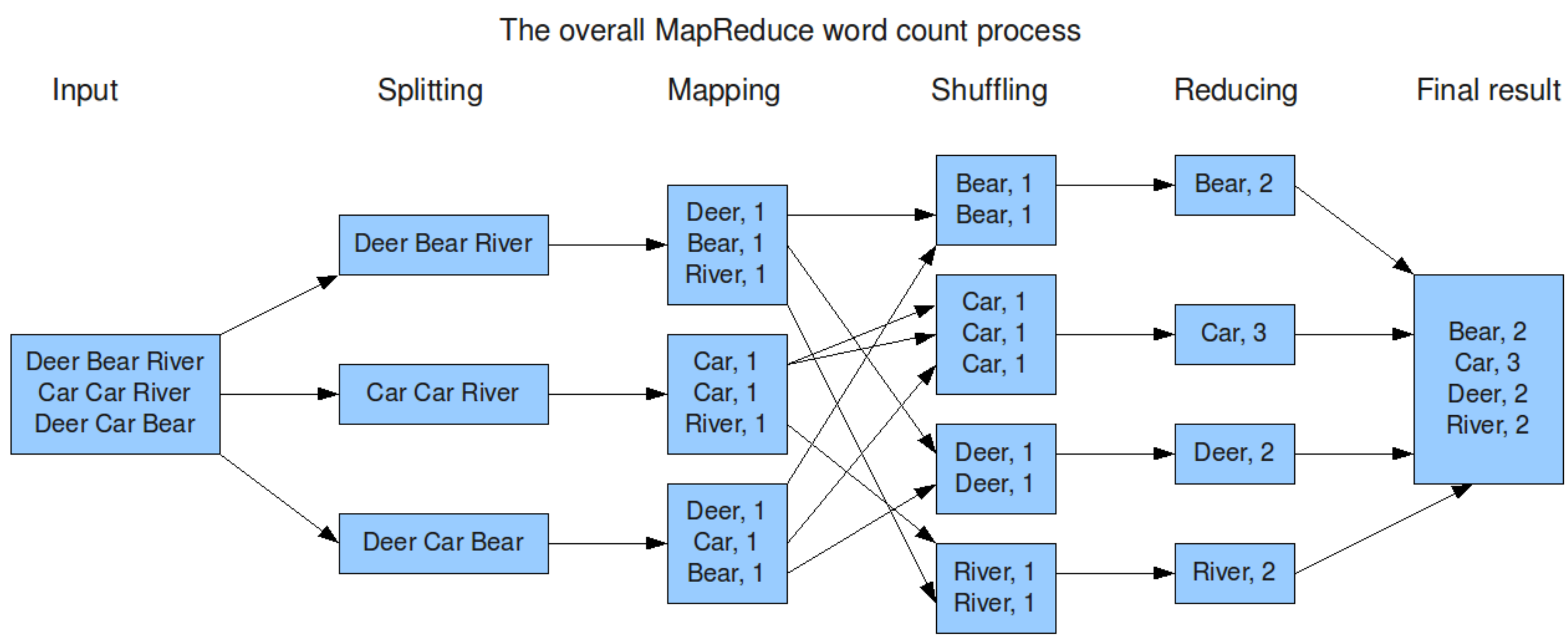
1. $\mathcal{C}$ contains all itemsets appearing in $\mathsf{FI}(\mathcal{D},\mathcal{I},\theta)$;
2. $\mathcal{C}$ contains no itemset A with frequency $f_{\mathcal{D}}(A) < \theta - \varepsilon_1$;
3. For every triplet $(A, f_A, \mathcal{K}_A) \in \mathcal{C}$, it holds
   (a) $|f_{\mathcal{D}}(A) - f_A| \leq \varepsilon_2$.
   (b) f_A and $f_{\mathcal{D}}(A)$ belong to $\mathcal{K}_A$.
   (c) $|\mathcal{K}_A| \leq 2\varepsilon_2$.



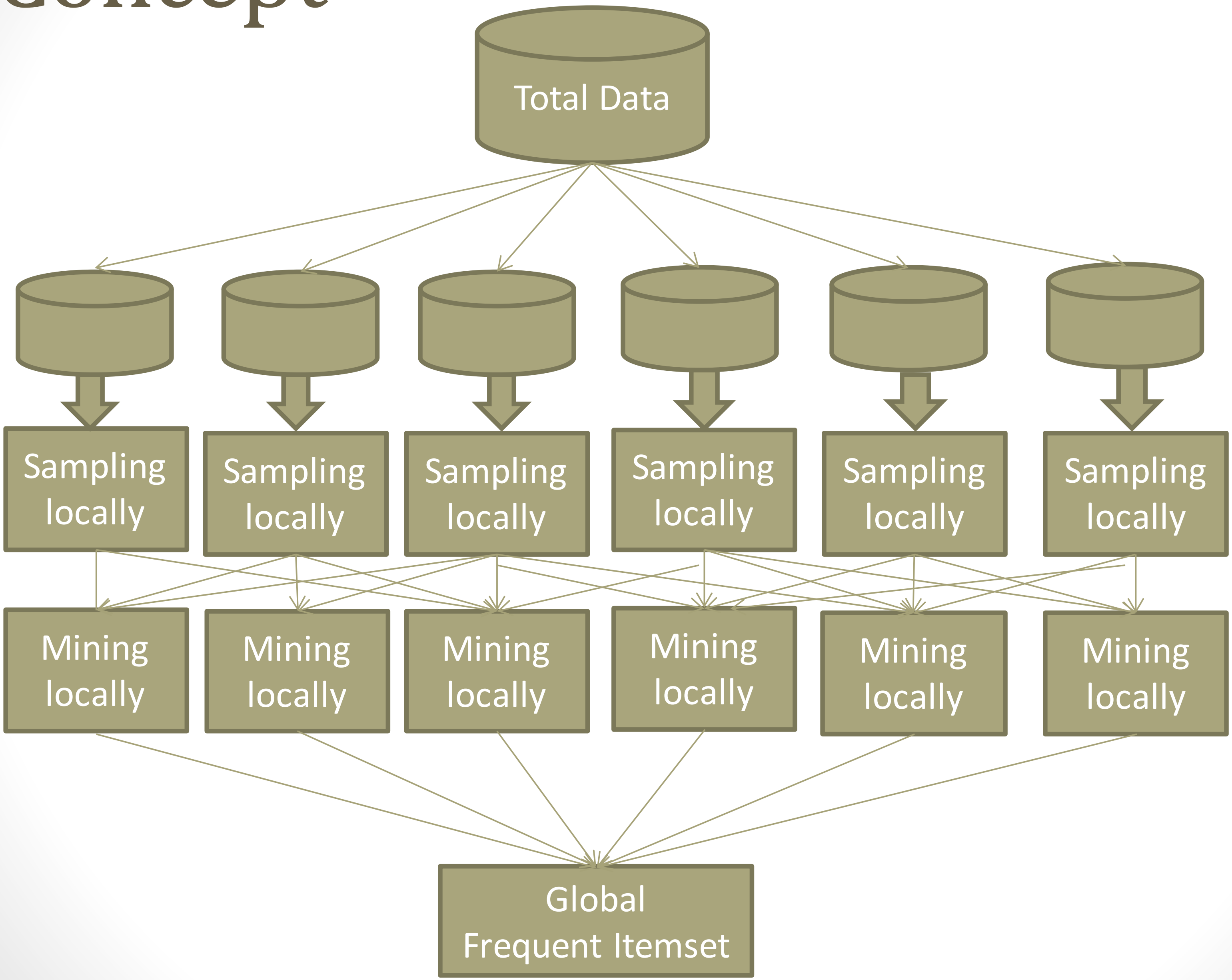
How many samples do we need?

LEMMA 1. *[29, Lemma 1] Let $\mathcal{D}$ be a dataset with transactions built on an alphabet $\mathcal{I}$, and let* d *be the maximum integer such that $\mathcal{D}$ contains at least* d *transactions of size at least* d. *Let $0 < \varepsilon, \delta, \theta < 1$. Let $\mathcal{S}$ be a random sample of $\mathcal{D}$ containing $|\mathcal{S}| = \boxed{\frac{2}{\varepsilon^2}\left(d + \log\frac{1}{\delta}\right)}$ transactions drawn uniformly and independently at random with replacement from those in $\mathcal{D}$, then with probability at least $1 - \delta$, the set $\mathsf{FI}(\mathcal{S}, \mathcal{I}, \theta - \frac{\varepsilon}{2})$ is a $(\varepsilon, \varepsilon/2)$-approximation of $\mathsf{FI}(\mathcal{D}, \mathcal{I}, \theta)$.*

Introduction of MapReduce



Concept



Total Data
Sampling locally
Sampling locally
Sampling locally
Sampling locally
Sampling locally
Sampling locally
Mining locally
Mining locally
Mining locally
Mining locally
Mining locally
Mining locally
Global Frequent Itemset

PARMA

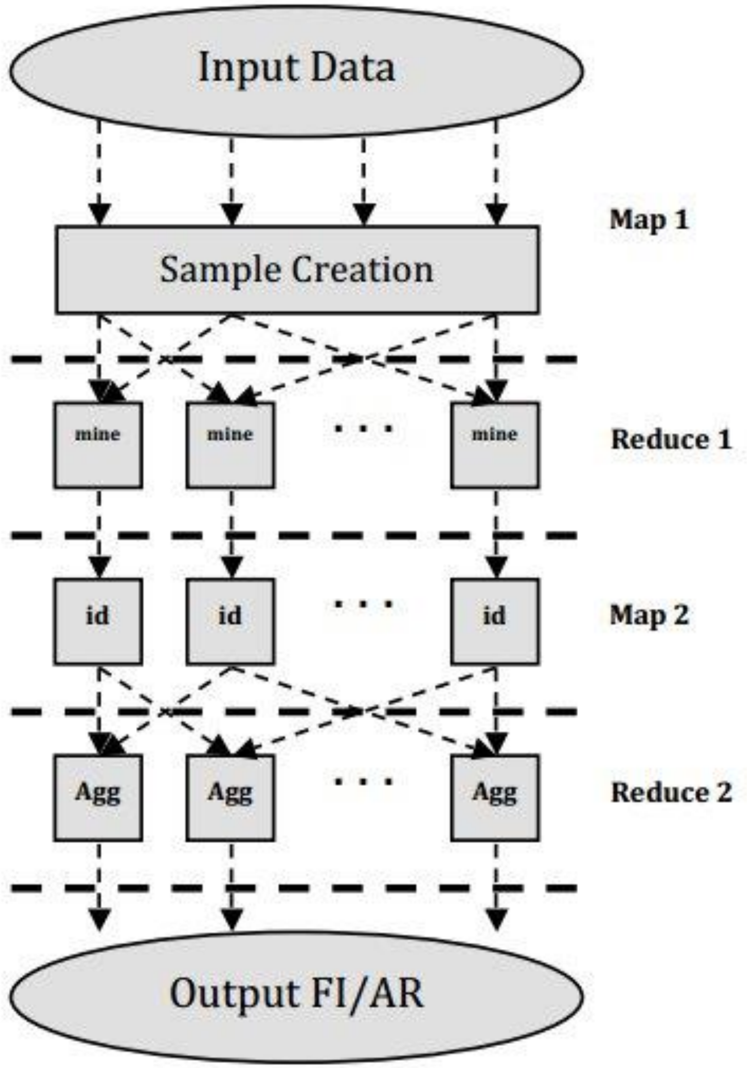


Figure 1: A system overview of PARMA. Ellipses represent data, squares represent computations on that data and arrows show the movement of data through the system.

Parameter Space

- **p:** number of processors/nodes
- **m:** memory within each node
- **w:** sample size
- **N:** number of samples
- **ε:** error probability
- **δ:** confidence bound

Given a fixed **ε** and **δ** value we can measure the sample size using Lemma1. If the sample size is greater than **m** we have to increase the number of samples.

Trade-offs

Number of samples

Probability to get the wrong approximation

- **Variables:** non-negative integer $\boxed{N}$, real $\boxed{\phi} \in (0, 1)$,
- **Objective:** minimize $2N/\varepsilon^2(d + \log(1/\phi))$.

$$N \leq p$$

$$\phi \geq e^{-m\varepsilon^2/2+d}$$

$$N(1-\phi) - \sqrt{N(1-\phi)2\log(1/\delta)} \geq N/2 + 1$$

In Reduce 2

- For each itemset, we have

$$\mathcal{F}_A = (f\mathcal{S}_i(A), [f\mathcal{S}_i(A) - \varepsilon/2, f\mathcal{S}_i(A) + \varepsilon/2])$$

- Then we use

$$R = N(1-\phi) - \sqrt{N(1-\phi)2\log(1/\delta)}. \qquad (5)$$

Result

- The itemset A is declared globally frequent and will be present in the output if and only if $|\mathcal{F}_A| \geq R$
- Let $[a_A, b_A]$ be the shortest interval such that there are at least N-R+1 elements from $\mathcal{F}_A$ that belong to this interval.

$$\tilde{f}(A) = a_A + \frac{b_A - a_A}{2}$$

$$\mathcal{K}_A = \left[a_A - \frac{\varepsilon}{2}, b_A + \frac{\varepsilon}{2}\right]$$

$$(A, (\tilde{f}(A), \mathcal{K}_A))$$

Association Rules

Lemma 2. *[29, Lemma 6] Let $\mathcal{D}$ be a dataset with transactions built on an alphabet $\mathcal{I}$, and let* d *be the maximum integer such that $\mathcal{D}$ contains at least* d *transactions of size at least* d*. Let* $0 < \varepsilon, \delta, \theta, \gamma < 1$ *and let* $\varepsilon_{\mathrm{rel}} = \frac{\varepsilon}{\max\{\theta,\gamma\}}$*. Fix* $c > 4 - 2\varepsilon_{\mathrm{rel}}$, $\eta = \frac{\varepsilon_{\mathrm{rel}}}{c}$*, and* $p = \frac{1-\eta}{1+\eta}\theta$*. Let $\mathcal{S}$ be a random sample of $\mathcal{D}$ containing* $\frac{1}{\eta^2 p}(d \log \frac{1}{p} + \log \frac{1}{\delta})$ *transactions from $\mathcal{D}$ sampled independently and uniformly at random. Then* $\mathsf{AR}(\mathcal{S}, \mathcal{I}, (1-\eta)\theta, \frac{1-\eta}{1+\eta}\gamma)$ *is an* $(\varepsilon, \varepsilon/2)$ *approximation to* $\mathsf{AR}(\mathcal{D}, \mathcal{I}, \theta, \gamma)$.

Outline

- Introduction
- Approach
- **Experiment**
- Conclusion

Implementation

- Amazon Web Service : ml.xlarge - 17GB
- Hadoop with 8 nodes
- Parameters :

  $\varepsilon = 0.05$ and $\delta = 0.01$
- Compare against DistCount,PFP

number of items	1000
average transaction length	5
average size of maximal potentially large itemsets	5
number of maximal potentially large itemsets	5
correlation among maximal potentially large itemsets	0.1
corruption of maximal potentially large itemsets	0.1

number of items	10000
average transaction length	10
average size of maximal potentially large itemsets	5
number of maximal potentially large itemsets	20
correlation among maximal potentially large itemsets	0.1
corruption of maximal potentially large itemsets	0.1

Compare with other Algorithm

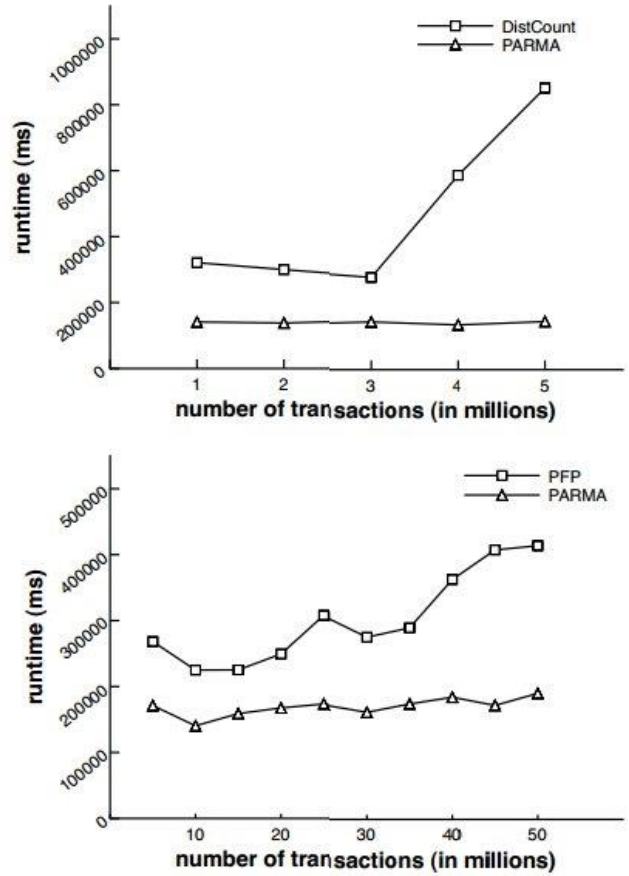


Figure 2: A runtime comparison of PARMA with DistCount (top) and PFP (bottom).

Runtime in Each Step

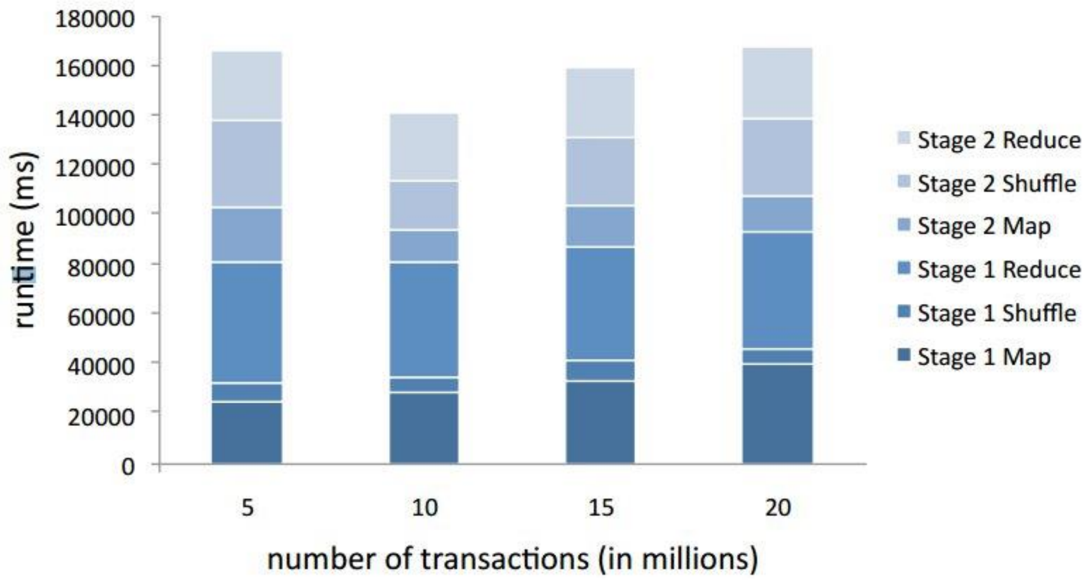


Figure 3: A comparison of runtimes of the map/reduce/shuffle phases of PARMA, as a function of number of transactions. Run on an 8 node Elastic MapReduce cluster.

Acceptable False Positives

θ	Real FI's	Output AFP's	Max AFP's
0.06	11016	11797	201636
0.09	2116	4216	10723
0.12	1367	335	1452
0.15	1053	299	415

Table 3: Acceptable False Positives in the output of PARMA

Error in frequency estimations

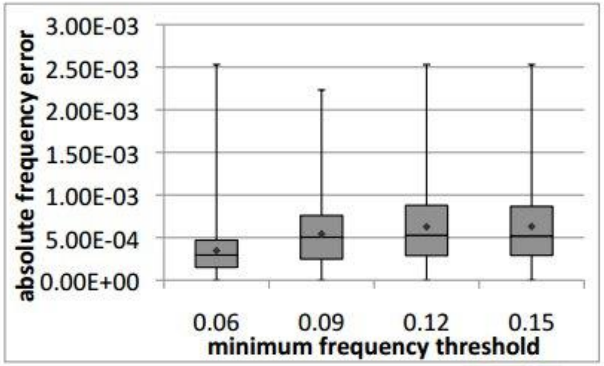


Figure 7: Error in frequency estimations as frequency varies.

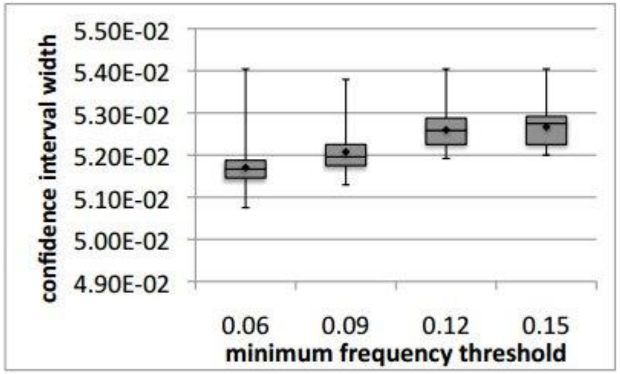


Figure 8: Width of the confidence intervals as frequency varies.

Outline

- Introduction
- Approach
- Experiment
- **Conclusion**

Conclusion

- A parallel algorithm for mining quasi-optimal collections of frequent itemsets and association rules in MapReduce.
- 30-55% runtime improvement over PFP.
- Verify the accuracy of the theoretical bounds, as well as show that in practice our results are orders of magnitude more accurate than is analytically guaranteed.