Parallel DBSCAN Clustering Algorithm using Apache Spark

Dianwei Han, Ankit Agrawal, Wei−keng Liao, Alok Choudhary

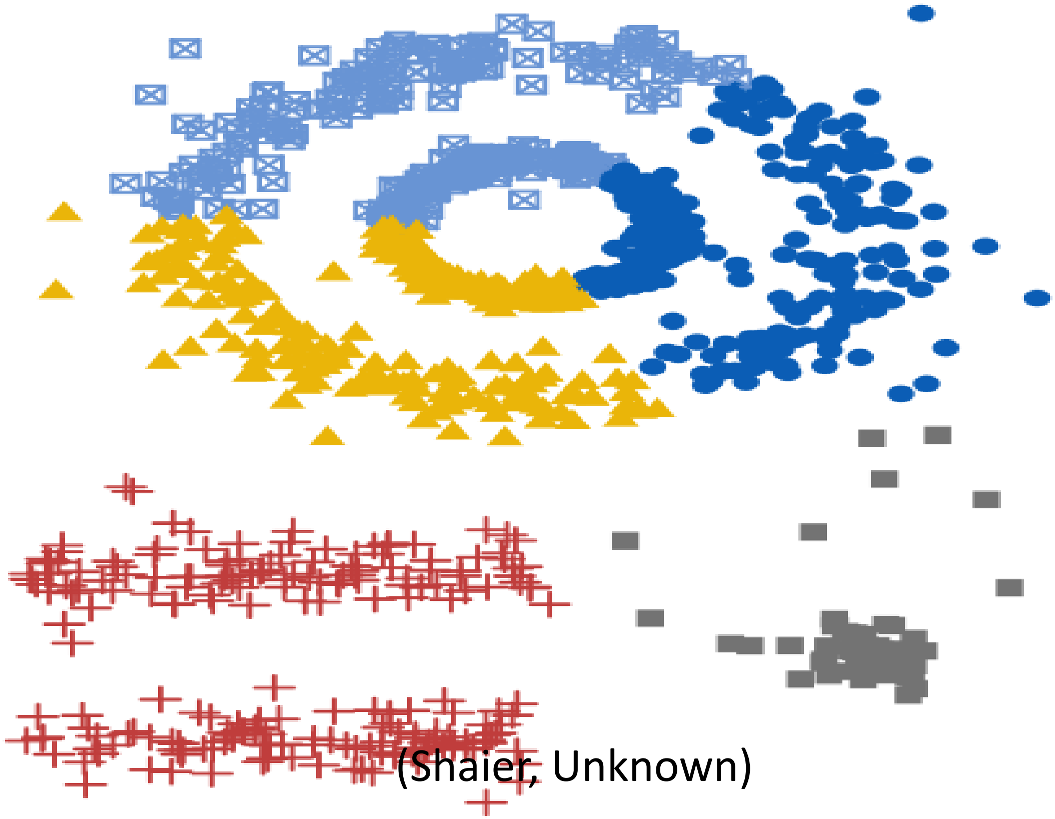
Presented by Anousheh Shahmirza

Overview

- Introduction to DBSCAN algorithm
- Problem definition
- Introduction to MapReduce algorithm
- Description of Apache Spark
- A novel scalable DBSCAN algorithm with Spark
- Conclusion
- Question

DBSCAN algorithm

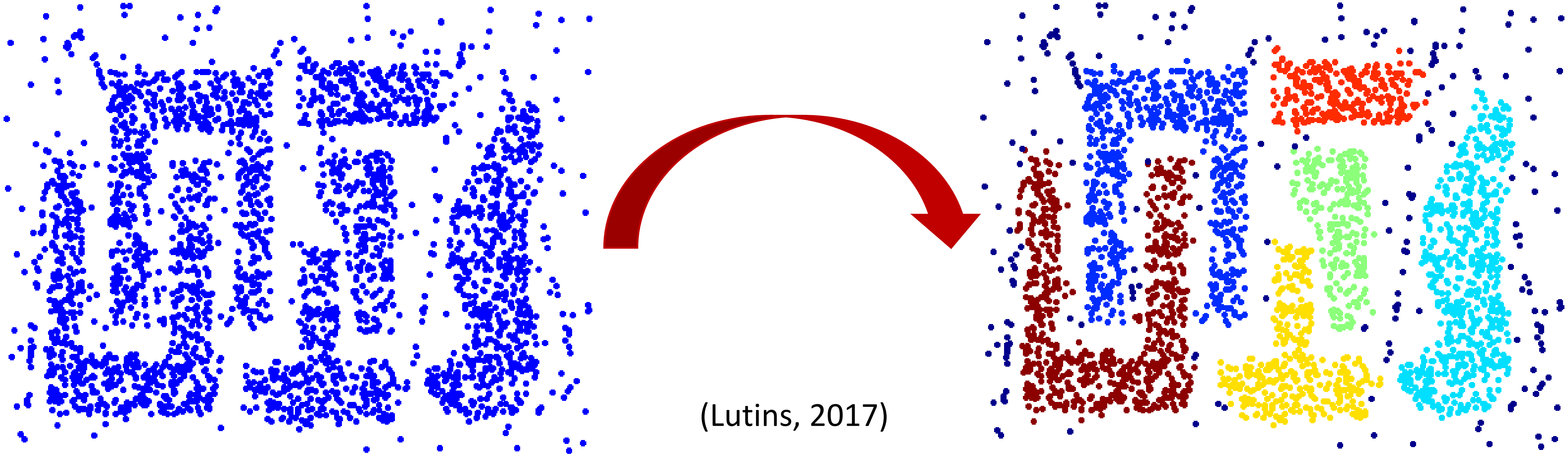
- Density-based spatial clustering
- An unsupervised learning data clustering approach



(Shaier, Unknown)

DBSCAN algorithm

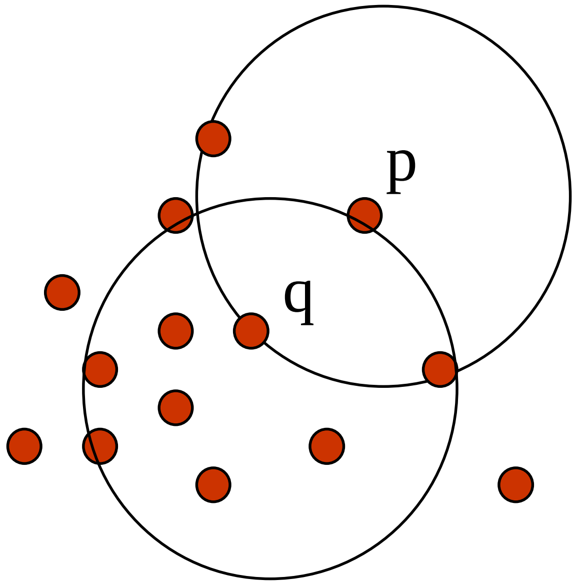
- Discover clusters of arbitrary shape and size
- Resistant to noise



(Lutins, 2017)

DBSCAN algorithm

- Density: number of points within a specified radius (Eps)
- Two parameters*:*
  - Epsilon (Eps): Maximum radius of the neighbourhood
  - MinPts: Minimum number of points in an Epsilon-neighbourhood of that point

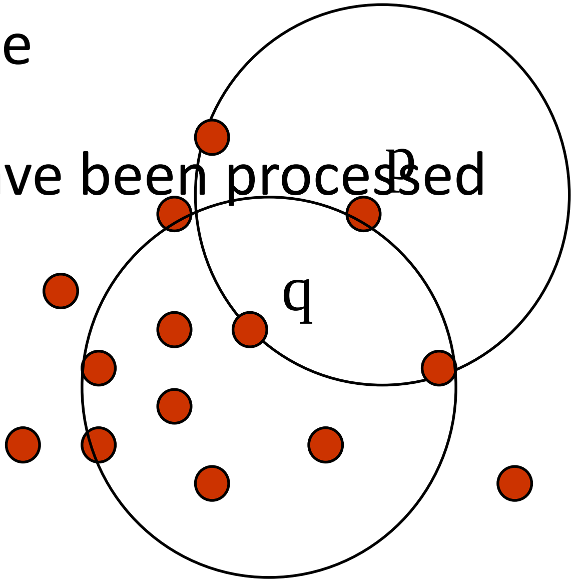


MinPts = 4

Eps = 1 cm

DBSCAN algorithm

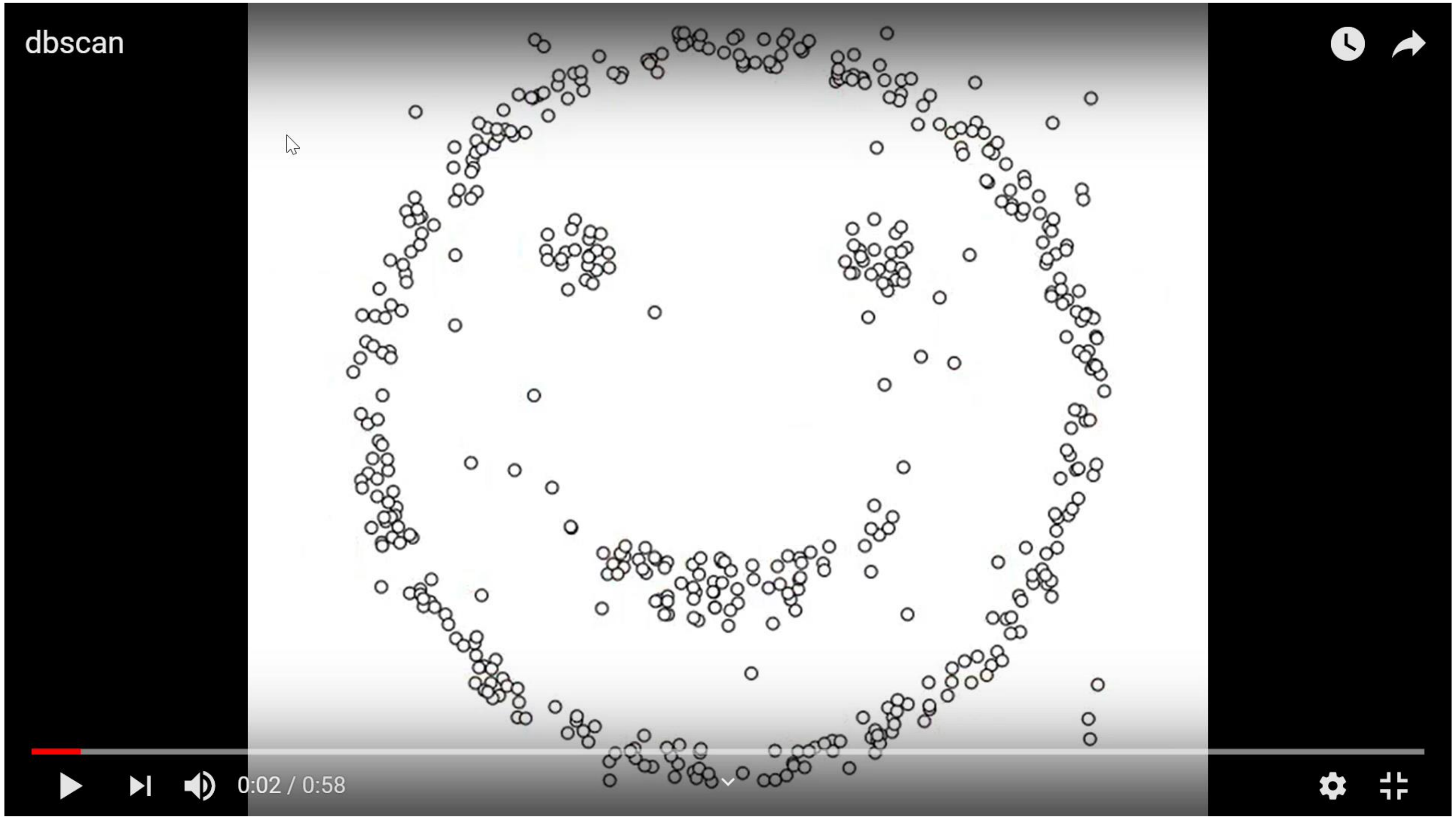
1. Select an arbitrary point **p**, insert that to a queue
2. Retrieve all neighbor points of **p** wrt **Eps**
3. If the number of points are greater than or equal to **MinPts**, a cluster is formed, all neighbours are inserted to the queue
4. Repeate steps 2 to 3 for all points in the queue
5. Continue the process until all of the points have been processed
6. Noise points do not belong to any clusters



MinPts = 4

Eps = 1 cm

DBSCAN algorithm



(source: https://www.youtube.com/watch?v=h53WMIImUuc)

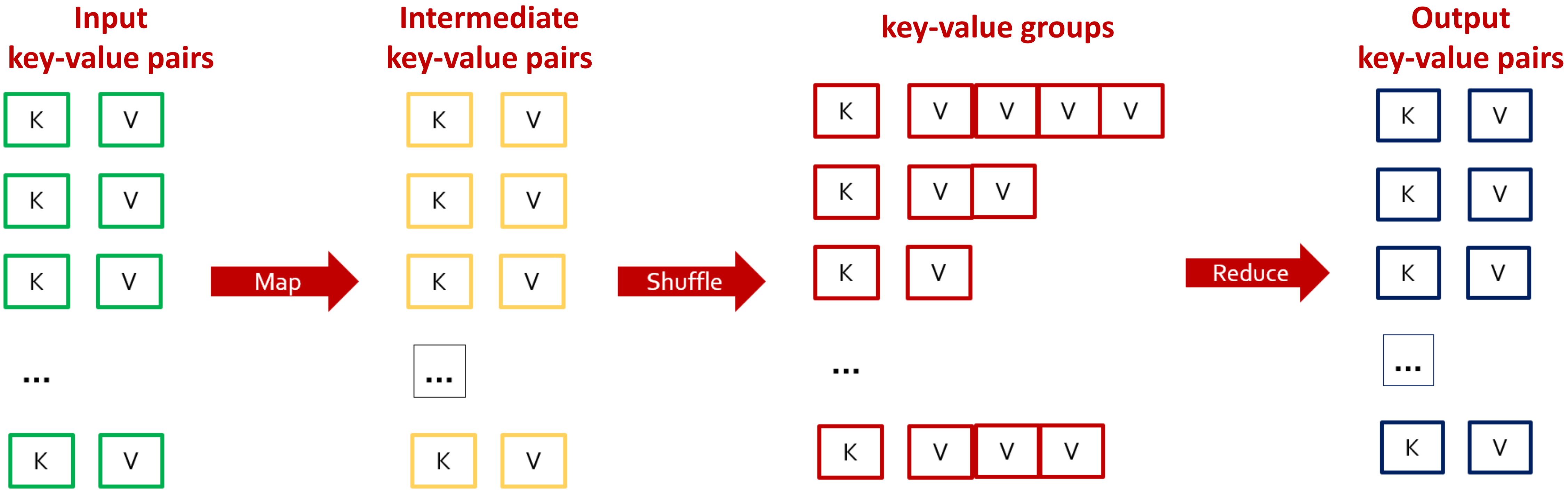
Problem

- Algorithm goes through each point of the database multiple times
- O(nlog(n)) Best case, using kd-tree
- $O(n^2)$ Worst case

MapReduce

- MapReduce is a framework for data processing
- The goal is to process massive data by connecting many cluster nodes to work in parallel
- Map function and reduce function suppose to be programmed
- In MapReduce data elements are always structured as key-value (i.e., (K, V)) pairs

MapReduce


Input
key-value pairs
K V
K V
K V
...
K V
Map
Intermediate
key-value pairs
K V
K V
K V
...
K V
Shuffle
key-value groups
K V V V V
K V V
K V
...
K V V V
Reduce
Output
key-value pairs
K V
K V
K V
...
K V

MapReduce

- Rather than sending data to where the application or logic resides, the logic is executed on the server where the data already resides
- A work performed by each task is done independetly

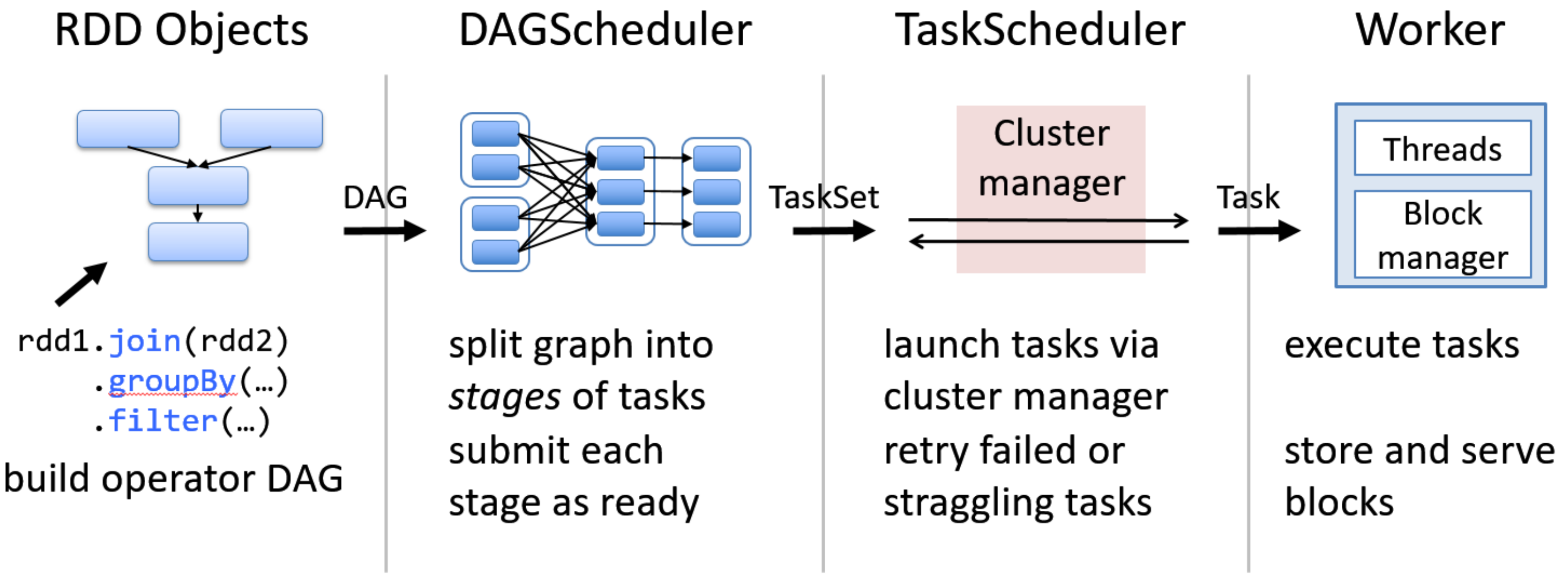
Hadoop

- Open source software framework designed for storage and processing of large scale data on clusters

- Use multiple machines for a single task
- Divided into Data Nodes and Compute Nodes
- At compute time, data is copied to the Compute Nodes
- A master program allocates work to individual nodes

Spark

- Not limited to map and reduce function, defines a large set of operations (transformations & actions)



RDD Objects

```
rdd1.join(rdd2)
    .groupBy(…)
    .filter(…)
```

build operator DAG

DAGScheduler

split graph into *stages* of tasks
submit each stage as ready

TaskScheduler

launch tasks via cluster manager
retry failed or straggling tasks

Worker

execute tasks

store and serve blocks

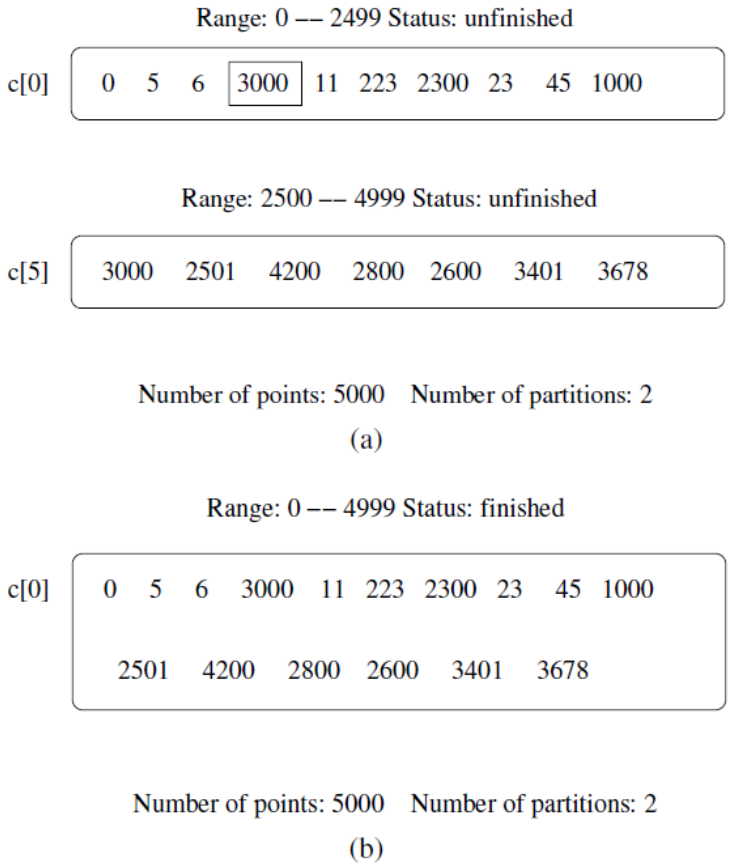
source: https://cwiki.apache.org/confluence/display/SPARK/Spark+Internals

A novel scalable DBSCAN algorithm with Spark

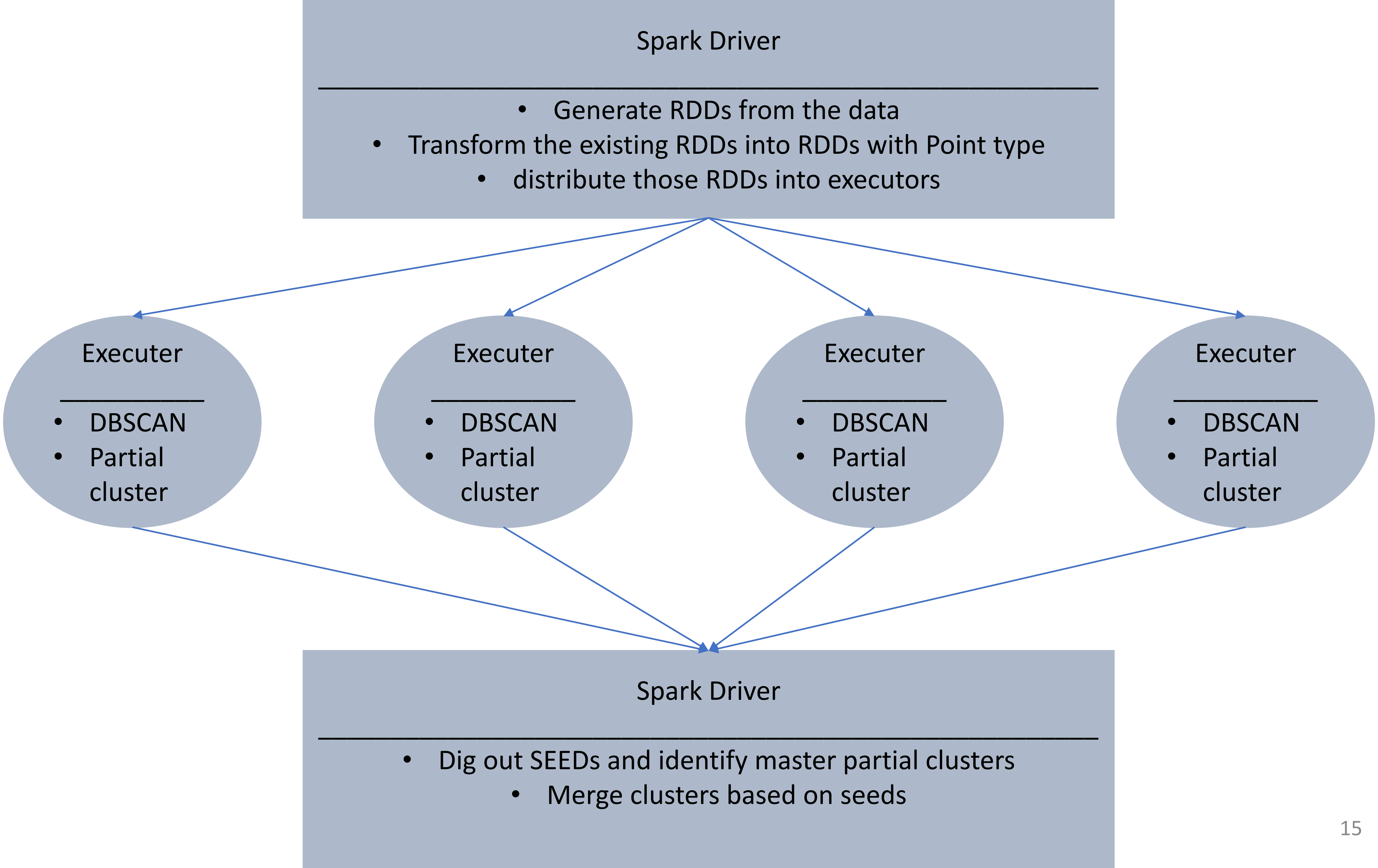
- SEEDs: points that do not belong to the current partition

- shuffle operations are prevented which costs a lot

- Generates the same results as the serial algorithm


Range: 0 –– 2499 Status: unfinished
c[0]
0 5 6 3000 11 223 2300 23 45 1000
Range: 2500 –– 4999 Status: unfinished
c[5]
3000 2501 4200 2800 2600 3401 3678
Number of points: 5000 Number of partitions: 2
(a)
Range: 0 –– 4999 Status: finished
c[0]
0 5 6 3000 11 223 2300 23 45 1000
2501 4200 2800 2600 3401 3678
Number of points: 5000 Number of partitions: 2
(b)

(Han, et.al, 2016)



Spark Driver

- Generate RDDs from the data
- Transform the existing RDDs into RDDs with Point type
- distribute those RDDs into executors

Executer

- DBSCAN
- Partial cluster

Executer

- DBSCAN
- Partial cluster

Executer

- DBSCAN
- Partial cluster

Executer

- DBSCAN
- Partial cluster

Spark Driver

- Dig out SEEDs and identify master partial clusters
- Merge clusters based on seeds

Question

- What was is the time complexity of DBSCAN?

Question

As I explained we can use Kd-tree algorithm in order to reduce the time complexity of the DBSCAN to (nlog(n))

- Who can explain how does RD-tree work on two dimension?

References

- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd, volume 96, pages 226–231, 1996
- Dianwei Han, Ankit Agrawal, Wei-Keng Liao, and Alok Choudhary. 2016. A Novel Scalable DBSCAN Algorithm with Spark. In Proc. 2016 IEEE Int'l Sympo. on Parallel and Distributed Processing. 1393–1402
- Kyuseok Shim. Mapreduce algorithms for big data analysis. Proceedings of the VLDB Endowment, 5(12):2016–2017, 2012.