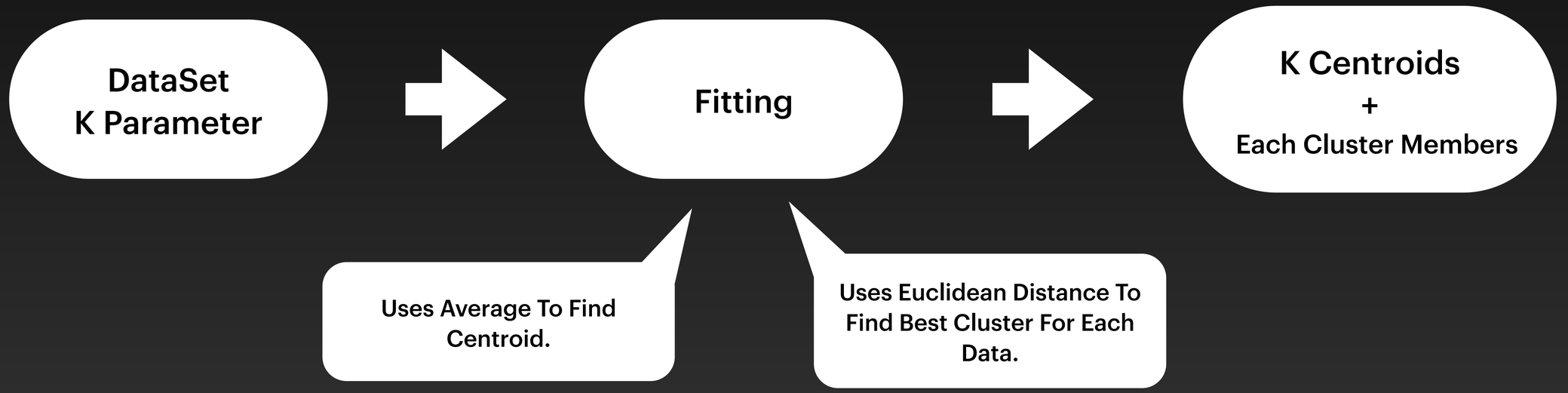
KMeans With OMP

Parallel Programing
Spring 2022
Dr. Savadi

Kiarash Vosough
Amin Erfanian
Mahsa AkhvanFard

How KMeans Operate?



Properties

Structured

Multi Dimension DataSet

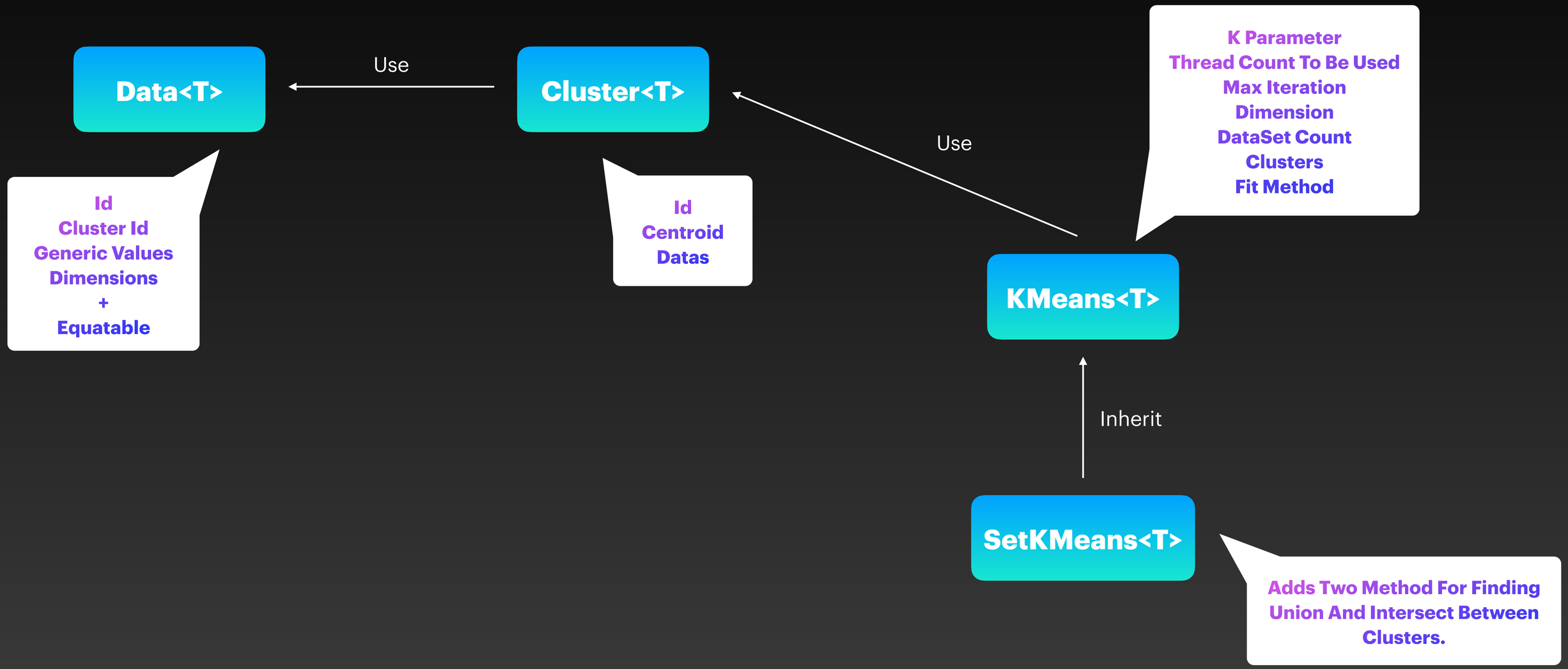
Generic DataSet

Documented In Code

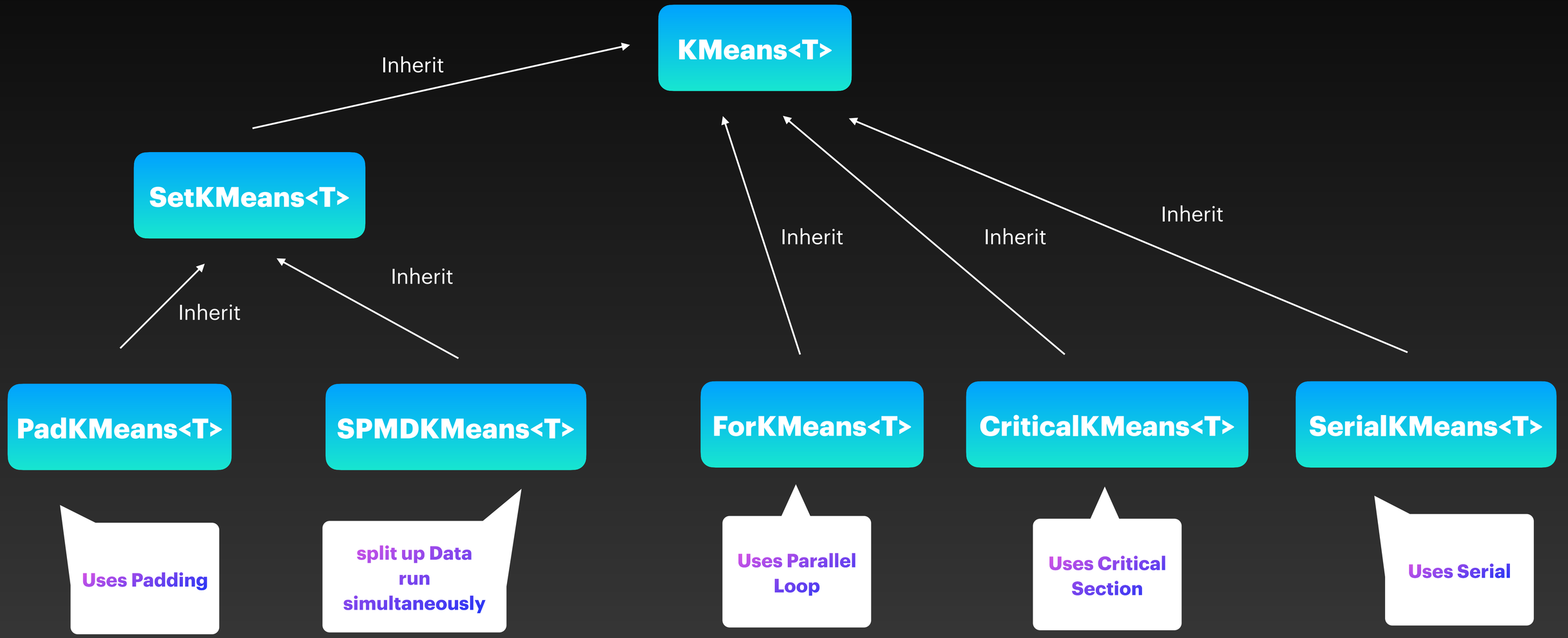
Open-Source

Multiple Parallel Approaches

Implementation



Implementation



Data<T>

```
20
21  template <typename T> class Data {
22
23  private:
24
25      int Id;
26
27      int clusterId;
28
29      int dimensions;
30
31      vector<T> values;
32
33      void loadData(string dataLine);
34
35  public:
36
37      const vector<T>& getValues() const;
38
39      Data(int id, string dataLine);
40
41      int getDimensions() const;
42
43      int getClusterId() const;
44
45      int getID() const;
46
47      void setClusterId(int id);
48
49      T getValueByIndex(int index) const;
50
51      bool operator != (Data const &obj) const;
52
53      bool operator == (Data const &obj) const;
54  };
```

Cluster<T>

```
57
58  template <typename T> class Cluster {
59
60  private:
61
62      int clusterId;
63
64      vector<T> centroid;
65
66      vector<Data<T>> datas;
67
68  public:
69
70      Cluster(int clusterId, Data<T> centroid);
71
72      Cluster ();
73
74      void addData(Data<T> data);
75
76      bool removeData(int dataId);
77
78      int getId() const;
79
80      Data<T> getData(int position) const;
81
82      int getSize() const;
83
84      vector<T> getCentroid() const;
85
86      T getCentroidByIndex(int index) const;
87
88      void setCentroidByIndex(int index, T newValue);
89
90      vector<Data<T>> getDatas() const;
91
92      void removeAllData();
93  };
```

KMeans<T>

```
95  template <typename T> class KMeans {
96
97  protected:
98
99      int threadCountToBeUsed;
100
101     int demandClusterNumber;
102
103     int maxIterations;
104
105     int dimensions;
106
107     int totalDataNumber;
108
109     vector<Cluster<T>> clusters;
110
111     vector<T> clusterCenters[100];
112
113     int findClosestClusterId(Data<T> data);
114
115     void initializeClusters(vector<Data<T>> &inputData);
116
117     int detectDataDimensions(vector<Data<T>> &inputData) const;
118
119     void storeCentroidWithUsedIteration(int usedIteration);
120
121     void reviseCentroidsOfClusters();
122
123     bool checkForCompletion(bool doneFlag, int iterationUntilNow);
124
125     void saveStringToFileAndPrintOnConsole(ofstream& outputStream, string message);
126 };
```

KMeans<T>

```
95  template <typename T> class KMeans {
96
97  public:
98
99      KMeans(int demandClusterNumber, int iterations, int threadCountToBeUsed);
100
101     const vector<double>* getClusterCenters() const;
102
103     vector<Cluster<T>> getClusters() const;
104
105     virtual int fit(vector<Data<T>>& inputData) = 0;
106
107     virtual void printResults(int usedIteration, bool saveToFile, string beginOutput);
108
109     virtual  void saveCSV(string type, string test);
110 };
```

Let’s See The Result

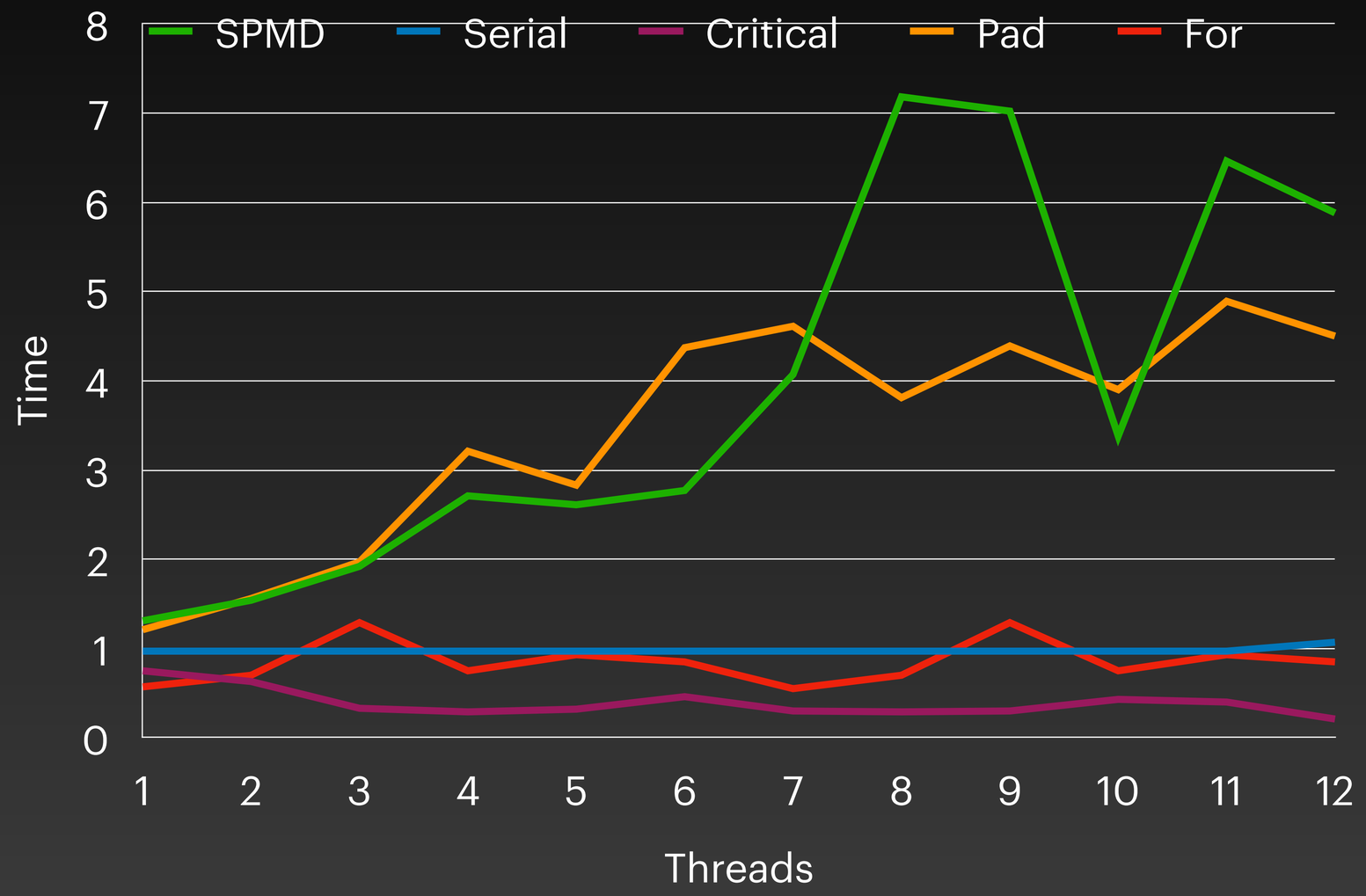
Used A Computer With 4 CPU Core i7 And 12 Threads.

Experiment Results

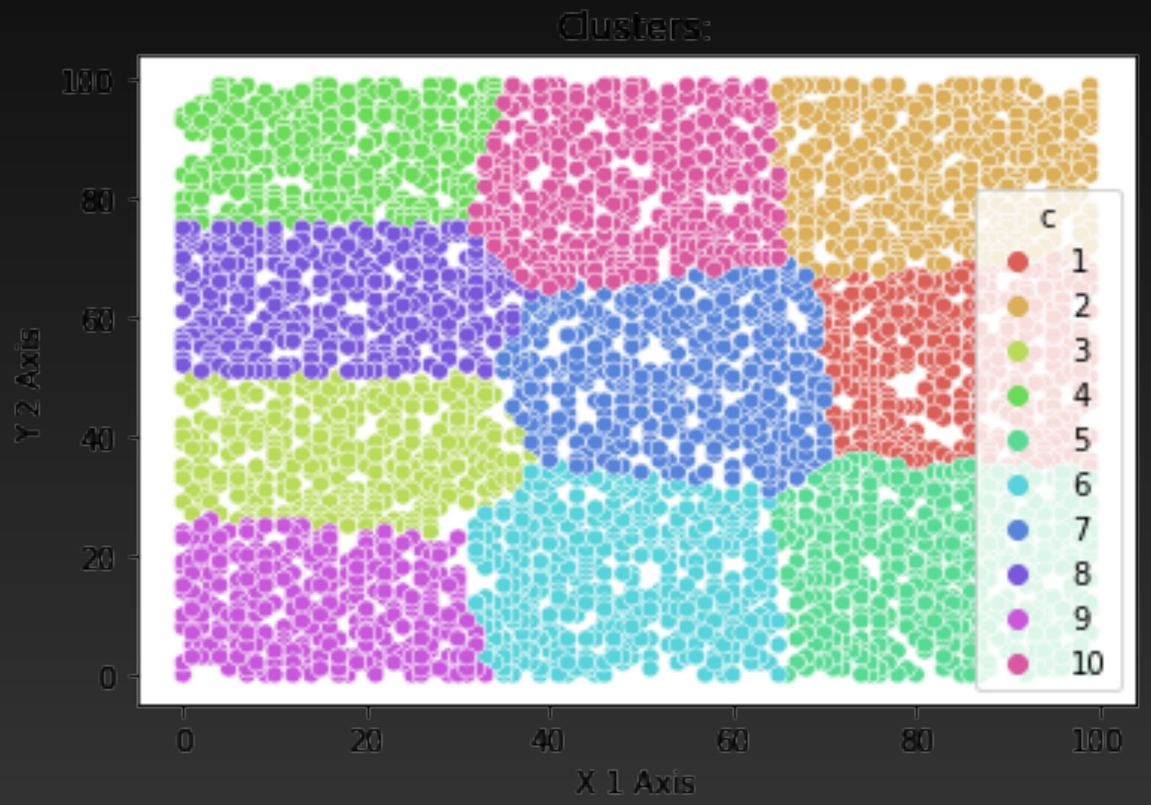
Tested With 5000 2-D Random-Generated DataSet And 30 Clusters

Type \ Thread	1	2	3	4	5	6	7	8	9	10	11	12
SPMD	1.311	1.54	1.92	2.71	2.61	2.77	4.07	7.18	7.02	3.38	6.46	5.88
Serial	1.07	-	-	-	-	-	-	-	-	-	-	-
Critical	0.75	0.63	0.33	0.29	0.32	0.46	0.3	0.29	0.3	0.43	0.4	0.21
Pad	1.21	1.56	1.97	3.21	2.83	4.37	4.61	3.81	4.39	3.90	4.89	4.50
For	0.57	0.70	1.29	0.75	0.93	0.85	0.55	0.70	1.29	0.75	0.93	0.85

Experiment Results



Cluster Visual Result



Thank You For The Attention
And
Feel Free To Ask.

Source Code is Available On My Github.
github.com/kiarashvosough1999/OMPKMeans