

# **Implementation of KMeans Clustering using OpenMP**

**RESEARCH PAPER :** A Hybrid MPI/OpenMP Parallelization of K -Means Algorithms Accelerated Using the Triangle Inequality

**PUBLISHER :** IEEE Access

**YEAR OF PUBLICATION :** 2019

## **TEAM MEMBERS:**

**19PD04:** Akshaya L  
**19PD10:** Divya Sivaraman  
**19PD19:** Krithika V  
**19PD35:** Sriram Siddhartha  
**19PD39:** Thiruvenkata Krishnan

## **TABLE OF CONTENTS:**

- Abstract
- Lloyd's algorithm for K-means clustering
- Elkan's Algorithm for K-means clustering
- OpenMP Directives used
- Code
- Inference
- Contribution
- Research Paper

## **ABSTRACT:**

We have implemented a parallelized K-means clustering algorithm using OpenMP. Here, we have implemented Lloyd's and Elkan's algorithm for K-means clustering

## **LLOYD'S ALGORITHM FOR K- MEANS CLUSTERING:**

- Initialize N points
- Initialize K centroids that represents K clusters
- Compute the distance between each point using euclidean distance and all of the centroids and assign the point to the nearest centroid which represents a cluster
- Update the centroids and repeat the process until reaching a maximum number of iterations or until the centroids don't change.

We implemented the above process using openmp instead of using normal loops. Using openmp will decrease the time to find clusters than the normal K-means clustering algorithm.

## **PSEUDOCODE FOR LLOYD'S ALGORITHM:**

```
repeat {
    {assignment step}
    for i ←1 to N do {
        a(i)←arg_mind(x(i),c(j))
    }
    {update step}
    for j←1 to K do y(j)←0, z(j)←0{
        for i←1 to N do {
            y(a(i))←y(a(i))+x(i)
            z(a(i))←z(a(i))+1
        }
        for j←1 to K do c(j)←y(j)/z(j)
    }
Until Termination Condition
```

## ELKANS'S ALGORITHM FOR K- MEANS CLUSTERING:

- Initialize N points
- Initialize K centroids that represents K clusters
- We set the upper bound and lower bound to manage the clusters. ➤ Before the assignment step, the algorithm pre-computes the inter-centroid distance matrix and, for each k, one-half the shortest distance from centroid to any other centroid.
- We take an index of an arbitrary centroid and ensure that in the first iteration, all the lower bounds overlap the corresponding upper bound. This initialization guarantees that tight lower and upper bounds in the first assignment step will be obtained.

We shift from Lloyd's algorithm to Elkan's algorithm because computing the distance is not necessary. We implemented the above process using openmp instead of using normal loops. Using openmp will decrease the time to find clusters than the normal K-means clustering algorithm.

## PSEUDOCODE FOR ELKAN'S ALGORITHM:

```
{initialization of bounds}
for i←1 to N do{
    a(i)←1, u(i)←∞
    for k←1 to K do l(i,k)←0
}
repeat {
    Compute inter-centroid distance matrix for k←1 to K
    do s(k)←min C(j,k)/2
    for i←1 to N do{
        If u(i)>s(a(i)) then
            r←true
        for k←1 to K do{
            z←1 to K do max(l(i,k),C(a(i),k)/2) If k=a(i)
            or u(i)<=z, then continue If r then
                u(i)←d(x(i),c(a(i)))
                r←false
            If u(i) <= z then continue
            l(i,k)←d(x(i),c(k))
            If l(i,k) <u(i) then
                a(i)←k
                u(i)←l(i,k)
        }
    c'← c, c←UpdateStep
    for k←1 to K do δ(k)=d(e(k),c'(k))
    for i ←1 to N do
        u(i) ←u(i) + δ(u(i))
        for k ← 1 to K do l(i,k) ← l(i,k) ←δ(k) until
TerminationCondition
```

## OPENMP DIRECTIVES USED:

- parallel
- section /sections
- schedule
- atomic
- for

## CODE:

### Lloyd's Algorithm without openmp:

⌚ Lloyd's Algorithm without openmp:

**Time:** 0.133723 seconds

### Output:

```
-bash-4.3$ g++ -o pdc -fopenmp pdc_package_lloyd_normal.cpp
-bash-4.3$ ./pdc
Number of processors: 4
Initializing..
Reading your data set
Points initialized!!
Your clusters are getting ready..
Clusters initialized!!
Points and clusters generated in: 2.004070 seconds
Iteration started...
Iteration no 1 done
Iteration no 2 done
Iteration no 3 done
Iteration no 4 done
Iteration no 5 done
Iteration no 6 done
Iteration no 7 done
Iteration no 8 done
Iteration no 9 done
Iteration no 10 done
Iteration no 11 done
Iteration no 12 done
Iteration no 13 done
Iteration no 14 done
Iteration no 15 done
Iteration no 16 done
Iteration no 17 done
Iteration no 18 done
Iteration no 19 done
Iteration no 20 done
Iteration no 21 done
Iteration no 22 done
Iteration no 23 done
Iteration no 24 done
Iteration no 25 done
Iteration no 26 done
Iteration no 27 done
Iteration no 28 done
Iteration no 29 done
Iteration no 30 done
Iteration no 31 done
Iteration no 32 done
```

Iteration no 86 done  
Iteration no 87 done  
Iteration no 88 done  
Iteration no 89 done  
Iteration no 90 done  
Iteration no 91 done  
Iteration no 92 done  
Iteration no 93 done  
Iteration no 94 done  
Iteration no 95 done  
Iteration no 96 done  
Iteration no 97 done  
Iteration no 98 done  
Iteration no 99 done  
Iteration no 100 done  
Centroid no: 1 14.5302,71.0323,59.56465 , 99 , 120 , 187 , 205 , 222 , 228 , 233 , 236 , 242 , 263 , 293 , 311 , 356 , 372 , 407 , 442 , 455 , 460 , 464 , 490 , 492 , 505 , 523 , 524 , 539 , 554 , 560 , 562 , 585 , 607 , 613 , 634 , 660 , 664 , 666 , 667 , 696 , 709 , 745 , 769 , 782 , 787 , 799 , 807 , 821 , 831 , 886 , 892 , 899 , 920 , 940 , 971 , 973 ,  
Centroid no: 2 23.1299,14.2296,24.3132 , 21 , 70 , 81 , 106 , 119 , 151 , 189 , 202 , 212 , 251 , 252 , 276 , 283 , 287 , 334 , 358 , 384 , 477 , 491 , 506 , 509 , 532 , 544 , 558 , 581 , 649 , 683 , 702 , 713 , 724 , 772 , 793 , 815 , 838 , 864 , 874 , 891 , 905 , 928 , 930 , 937 , 958 ,  
Centroid no: 3 47.7422,49.696,18.231 , 12 , 16 , 58 , 82 , 91 , 97 , 105 , 133 , 134 , 138 , 217 , 291 , 305 , 328 , 343 , 350 , 368 , 371 , 374 , 382 , 386 , 424 , 428 , 441 , 551 , 571 , 602 , 628 , 644 , 648 , 677 , 767 , 804 , 809 , 836 , 837 , 861 , 866 , 893 , 938 , 952 , 954 , 959 , 968 ,  
Centroid no: 4 55.3561,11.6692,19.15710 , 11 , 17 , 28 , 32 , 53 , 67 , 127 , 158 , 218 , 304 , 332 , 341 , 426 , 482 , 515 , 534 , 555 , 574 , 595 , 596 , 639 , 665 , 675 , 684 , 699 , 721 , 731 , 755 , 784 , 832 , 850 , 880 , 923 , 978 , 979 , 987 ,  
Centroid no: 5 66.841,26.2607,86.9865 , 13 , 26 , 51 , 64 , 73 , 75 , 76 , 142 , 148 , 152 , 155 , 168 , 214 , 224 , 231 , 243 , 277 , 298 , 338 , 354 , 362 , 369 , 394 , 402 , 421 , 432 , 435 , 444 , 454 , 486 , 510 , 548 , 580 , 586 , 590 , 594 , 599 , 611 , 636 , 670 , 672 , 689 , 718 , 732 , 744 , 797 , 798 , 844 , 851 , 860 , 891 , 932 , 939 , 967 , 981 ,  
Centroid no: 6 84.59498,19.49988,58.4822 , 60 , 92 , 101 , 111 , 210 , 211 , 215 , 220 , 221 , 241 , 260 , 269 , 280 , 290 , 299 , 307 , 337 , 366 , 376 , 390 , 392 , 401 , 409 , 448 , 488 , 503 , 528 , 559 , 589 , 603 , 605 , 606 , 609 , 647 , 655 , 719 , 726 , 728 , 768 , 803 , 805 , 817 , 856 , 868 , 877 , 881 , 887 , 888 , 894 , 896 , 934 , 935 , 942 , 946 , 984 , 993 ,  
Centroid no: 7 81.6601,49.4202,37.3277 , 103 , 147 , 157 , 165 , 167 , 179 , 191 , 197 , 198 , 284 , 285 , 320 , 321 , 344 , 346 , 357 , 389 , 399 , 400 , 416 , 436 , 439 , 502 , 508 , 514 , 522 , 531 , 533 , 561 , 615 , 616 , 617 , 626 , 651 , 690 , 695 , 708 , 737 , 753 , 875 , 929 , 931 , 985 , 990 ,  
Centroid no: 8 87.8242,78.6777,62.20619 , 44 , 59 , 96 , 110 , 115 , 131 , 139 , 141 , 160 , 173 , 185 , 234 , 254 , 256 , 257 , 288 , 296 , 316 , 326 , 333 , 347 , 377 , 395 , 414 , 418 , 420 , 423 , 466 , 474 , 489 , 501 , 518 , 529 , 541 , 614 , 658 , 688 , 717 , 735 , 736 , 739 , 741 , 746 , 747 , 751 , 812 , 818 , 825 , 829 , 841 , 845 , 883 , 933 , 934 , 974 ,  
Centroid no: 9 14.2224,28.1362,32.71814 , 45 , 46 , 71 , 83 , 88 , 117 , 124 , 132 , 140 , 171 , 186 , 194 , 239 , 244 , 249 , 278 , 289 , 349 , 398 , 415 , 422 , 451 , 483 , 537 , 575 , 588 , 622 , 646 , 653 , 661 , 663 , 749 , 760 , 763 , 764 , 791 , 808 , 820 , 823 , 961 , 964 , 977 ,  
Centroid no: 10 20.8851,20.1052,84.22375 , 37 , 40 , 41 , 42 , 48 , 55 , 62 , 78 , 87 , 98 , 114 , 118 , 129 , 135 , 136 , 153 , 190 , 208 , 219 , 248 , 250 , 262 , 272 , 273 , 275 , 295 , 300 , 313 , 331 , 359 , 367 , 405 , 430 , 431 , 453 , 456 , 458 , 466 , 467 , 471 , 476 , 478 , 507 , 526 , 576 , 623 , 631 , 669 , 681 , 693 , 754 , 771 , 774 , 777 , 792 , 802 , 827 , 858 , 870 , 915 , 944 , 951 , 972 , 976 , 982 , 986 , 991 , 995 , 999 ,  
Centroid no: 11 86.05597,24.74719,12.25424 , 52 , 85 , 89 , 122 , 137 , 156 , 169 , 170 , 178 , 201 , 203 , 309 , 335 , 351 , 353 , 447 , 470 , 475 , 493 , 530 , 545 , 547 , 568 , 579 , 620 , 629 , 674 , 682 , 686 , 714 , 716 , 742 , 743 , 748 , 795 , 801 , 811 , 822 , 824 , 853 , 873 , 882 , 900 , 906 , 909 , 920 , 925 , 930 , 935 , 940 , 945 , 950 , 955 , 958 ,  
Centroid no: 12 24.4269,89.608,76.68218 , 39 , 69 , 112 , 143 , 161 , 193 , 195 , 204 , 207 , 232 , 240 , 294 , 312 , 314 , 319 , 327 , 336 , 378 , 381 , 406 , 408 , 412 , 425 , 440 , 450 , 487 , 549 , 565 , 583 , 601 , 680 , 687 , 794 , 814 , 830 , 841 , 862 , 897 , 956 , 992 ,  
Activate Windows  
Copyright © Microsoft Corporation. All rights reserved.  
Windows

Centroid no: 4 55.3561,111.6692,19.15710 , 11, 17, 28, 32, 53, 67 , 127, 158, 218, 304, 332, 341, 426, 482, 515, 534, 555, 574, 595, 596, 639, 665, 675  
684 , 699, 721 , 731, 755, 784 , 796, 832, 850, 880, 923, 978, 979, 987 ,  
Centroid no: 5 66.841,36.26207,86.9865 , 13, 26, 51, 54, 73 , 75, 76, 142, 148 , 152, 155, 168, 214, 224, 231, 243, 277, 298, 338, 354, 362, 365 , 39  
402 , 421, 432, 435, 444, 454, 486, 510, 548, 580, 586, 590, 594, 599, 611, 636, 670, 672, 689, 718, 732, 744 , 797, 798, 844, 851, 860, 89  
5, 932 , 939 , 967, 981 ,  
Centroid no: 6 84.5948,19.4888,58.4222 , 60, 92, 101, 111, 210, 211, 215, 220, 221, 241, 260, 269, 280, 290, 299, 307, 337, 366, 376, 390, 392, 40  
109 , 448, 488, 503, 528, 559, 589, 603, 605, 606, 609, 647, 655, 719, 726, 728, 768, 803, 805, 817, 856, 866, 877, 881, 888, 894, 90  
3, 835 , 942 , 946, 984, 983 ,  
Centroid no: 7 81.6601,49.4202,37.3277 , 103, 147, 157, 165, 167, 179, 191, 197 , 198, 284, 285, 320, 321, 344, 346, 357, 389, 399, 400, 416, 436, 439  
502, 508 , 514, 522, 531, 533, 561, 615, 616, 617, 626, 651, 690, 695, 708, 737, 753, 875, 929, 931, 985, 990 ,  
Centroid no: 8 87.82428,78.7777,62.20619 , 44, 59, 96, 110, 115, 131, 139, 141, 160, 173, 185, 234, 254, 256, 257, 288, 296, 316, 326, 333, 347, 37  
7, 395 , 414, 418, 420, 423, 466, 474, 489, 501, 518, 529, 541, 614, 658, 688, 717, 735, 736, 739, 741, 746, 747 , 751, 812, 818, 825, 829, 84  
7, 844, 859, 872, 883, 883, 934, 974 ,  
Centroid no: 9 14.2224,28.1362,32.71814 , 45, 46, 71, 83, 88, 117, 124, 132, 140, 171, 186, 194, 239, 244, 249, 278, 289, 349, 398, 415, 422, 451  
483, 537, 575, 588, 622, 646, 653, 661, 663, 749, 760, 763, 764, 791, 808, 820, 823, 961, 964, 977 ,  
Centroid no: 10 20.8851,20.1052,84.22735 , 37, 40, 41, 42, 48, 55, 62, 78, 87, 98, 114, 118, 126, 129, 135, 136, 153, 190, 208, 219, 248, 250, 262  
, 272, 273, 275, 295, 308, 313, 331, 359, 367, 405, 430, 431, 453, 456, 458, 465, 467, 471, 476, 478, 507, 526, 576, 623, 631, 669, 681, 693  
, 754, 771, 774, 777, 792, 802, 827, 854, 870, 915, 944, 951, 972, 976, 982, 986, 991, 995, 998, 999 ,  
Centroid no: 11 86.05945,24.4749,12.25424 , 52, 85, 89, 122, 137, 156, 169, 170, 178, 201, 203, 309, 335, 351, 353, 447, 469, 475, 530, 545, 547, 56  
8, 579, 620, 629, 674, 682, 686, 714, 716, 742, 743, 748, 795, 801, 811, 822, 824, 853, 873, 882, 900, 906, 909, 913, 919, 926 ,  
Centroid no: 12 24.424689,69.608,7.68218 , 39, 69, 112, 143, 161, 193, 195, 204, 207, 232, 240, 254, 312, 314, 319, 327, 336, 378, 381, 406, 408, 412  
, 425, 440, 450, 487, 549, 565, 583, 601, 680, 687, 794, 814, 830, 841, 862, 897, 956, 992 ,  
Centroid no: 13 69.9013,76.8615,79.73215 , 25, 29, 63, 72, 74, 80, 84, 95, 121, 123, 130, 162, 172, 176, 183, 192, 223, 230, 255, 268, 281, 302, 3  
66, 322, 355, 361, 380, 383, 443, 473, 499, 540, 542, 566, 598, 633, 635, 643, 654, 656, 659, 685, 694, 701, 703, 722, 733, 757, 761, 778, 7  
79, 780, 788, 811, 855, 869, 871, 884, 889, 901, 908, 911, 943, 948, 970, 988, 989 ,  
Centroid no: 14 -nan,-nan,79.732 ,  
Centroid no: 15 16.316,60.00502,14.8676 , 23, 50, 54, 154, 159, 181, 196, 225, 238, 271, 282, 301, 324, 325, 329, 363, 364, 381, 396, 410, 413, 419  
429, 433, 449, 452, 457, 479, 480, 481, 498, 500, 521, 546, 570, 587, 621, 624, 638, 662, 712, 720, 752, 762, 776, 790, 813, 833, 835, 867  
885, 902, 910, 925, 963, 996 ,  
Centroid no: 16 80.0858,78.2169,19.820 , 3, 7, 20, 34, 57, 79, 93, 113, 116, 145, 166, 177, 209, 216, 226, 227, 235, 253, 264, 292, 303, 317, 330  
339, 340, 345, 352, 370, 379, 397, 403, 417, 427, 434, 472, 485, 493, 494, 513, 564, 573, 584, 592, 593, 597, 610, 657, 676, 678, 697, 692  
, 697, 700, 704, 706, 707, 710, 734, 750, 759, 765, 773, 842, 845, 898, 916, 918, 924, 945, 971 ,  
Centroid no: 17 52.877895,30.6733,43.62638 , 56, 61, 66, 68, 90, 100, 125, 144, 149, 182, 184, 188, 245, 258, 279, 310, 365, 438, 446, 461, 468, 496  
, 497, 511, 519, 552, 557, 569, 578, 582, 591, 642, 668, 671, 715, 740, 756, 770, 786, 800, 806, 890, 907, 950, 965, 998 ,  
Centroid no: 18 30.53244,49.4472,67.77 , 30, 31, 43, 49, 107, 108, 109, 146, 163, 175, 180, 200, 213, 247, 261, 274, 286, 315, 318, 373, 404, 437,  
459, 463, 512, 517, 543, 550, 563, 567, 650, 652, 711, 723, 729, 730, 766, 785, 789, 828, 846, 852, 863, 927, 955, 960, 969, 975, 980, 994,  
997 ,  
Centroid no: 19 24.71599,84.7619,77.8164 , 8, 36, 86, 94, 104, 199, 225, 237, 265, 267, 297, 300, 323, 361, 375, 385, 387, 388, 393, 411, 462, 470,  
516, 520, 538, 558, 556, 600, 608, 612, 625, 630, 632, 645, 673, 698, 705, 727, 738, 781, 810, 826, 834, 839, 840, 843, 857, 876, 878, 896,  
904, 914, 917, 936, 941, 949, 957, 962, 966, 983 ,  
Centroid no: 20 49.0180,83.9932,40.57627 , 33, 47, 102, 120, 150, 164, 174, 206, 246, 259, 266, 270, 342, 348, 445, 484, 865, 879, 912, 921, 922 ,  
933Settings to activate Windows.  
Number of iterations: 100, total time: 0.133723 seconds, time per iteration: 0.001337 seconds  
-bash-4.3\$

```
Number of iterations: 100, total time: 0.060704 seconds, time per iteration: 0.000607 seconds  
-bash-4.3$
```

### Lloyd's Algorithm with openmp:

≡ Lloyd's Algorithm with openmp

**Time:** 0.060704

**Output:**

```
Number of iterations: 100, total time: 0.133723 seconds, time per iteration: 0.001337 seconds  
-bash-4.3$
```

### Elkan's Algorithm without openmp:

≡ Elkan's Algorithm without openmp

**Time:** 0.13489

**Output:**

```
Number of iterations: 100, total time: 0.050569 seconds, time per iteration: 0.000506 seconds  
-bash-4.3$
```

### Elkan's Algorithm with openmp:

≡ Elkan's Algorithm with openmp

**Time:** 0.050569

**Output:**

```
Number of iterations: 100, total time: 0.134829 seconds, time per iteration: 0.001348 seconds  
-bash-4.3$
```

### DATA SET:

≡ data\_set1

## **INFERENCE:**

- Lloyd's with openmp takes more time than Lloyd's without openmp.
- Elkan's with openmp takes more time than Elkan's without openmp.
- Elkan's without openmp is faster than Lloyd's without openmp.
- So, **Elkan's algorithm without openmp** is better than all four methods for K-means clustering.

## **CONTRIBUTION:**

1. Lloyd's Algorithm with and without OpenMP  
Akshaya L and Krithika V
2. Elkan's Algorithm with and without OpenMP  
Divya Sivaraman, Sriram Siddhartha and Thiruvenkata Krishnan