Final report

Lei Zhong, Hantian Zhang, Jian Zhang

Abstract

We did more experiment with different parameters during the Milestone 4. We optimized the code and visualized the results as a video on youtube[7]. In this final report, we will summarize over the whole three-month project, demonstrate the results and point out some future work

1 Introduction

2 Contribution

to add In addition to coreset sampling, we could also run on the whole feature,

3 Performance Measurements and Results

3.1 Performance metrics

There are three common performance metrics, we have our own way to measure:

- Execution time: one is run-time, the other is normalized instance hour;
- Memory-consumption & number of machines: peak memory of mapper and reducer, see Table 3.1. We use m3.2xlarge and one master, two cores;
- Solution quality: our project is quite innovative and there is no existing work. We can only compare our results with differnt parameters by topographical world map[8].

Generally, it would take four hours to run on the whole datasets, which is more than 100 instance hours.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Table 1: Instance types

3.2 Result of recent four decades

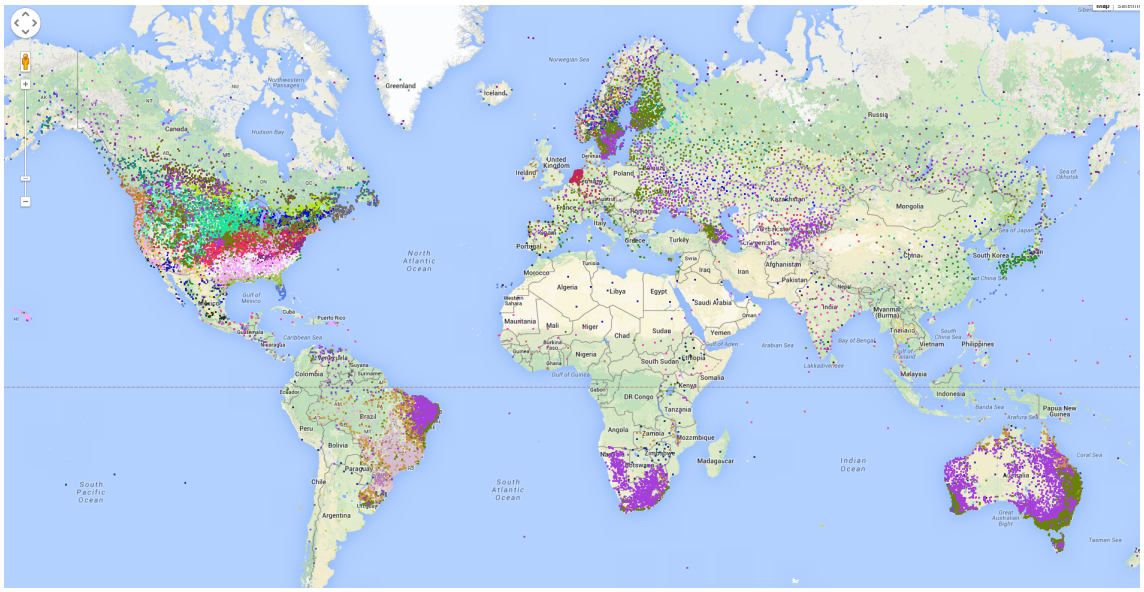
Same as Milestone 3, we selected the year 1983, 1993, 2003 and 2013 to show the results. The size of raw compressed dataset is as showing in the Table 3.2.

We kept adjusting the parameters until we found the result is resonable. The following pictures are snapshots from 150 clusters, 500 iterations. From the pictures we can see that there do exist some pattern in the same region in different years. But generally, the shape of clusters would note be the same.

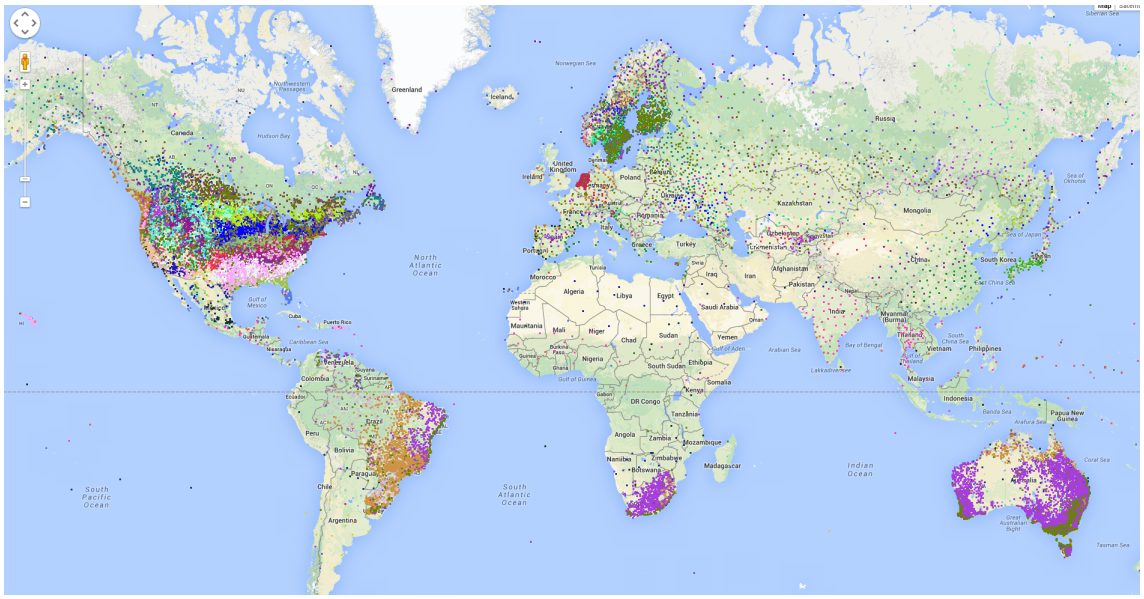
Compared with results in milestone 3, there are two major improvements.

Year	1983	1993	2003	2013
Compressed Size	155MB	152MB	159MB	166MB
Original	912.4MB	899.7MB	960.7MB	987.2MB

Table 2: Size of dataset in selected years

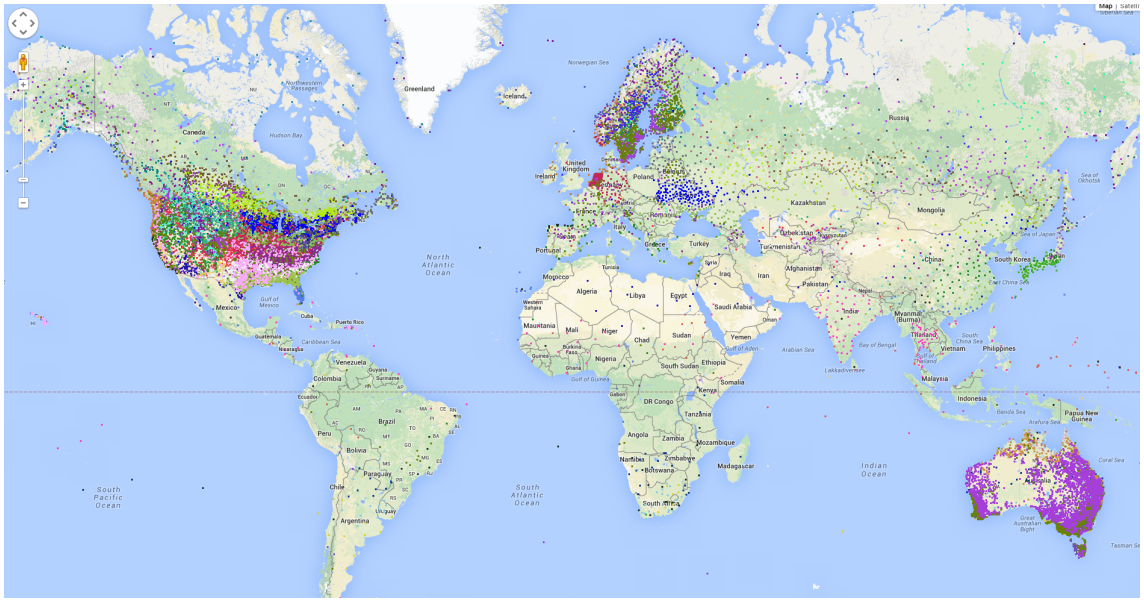


1983

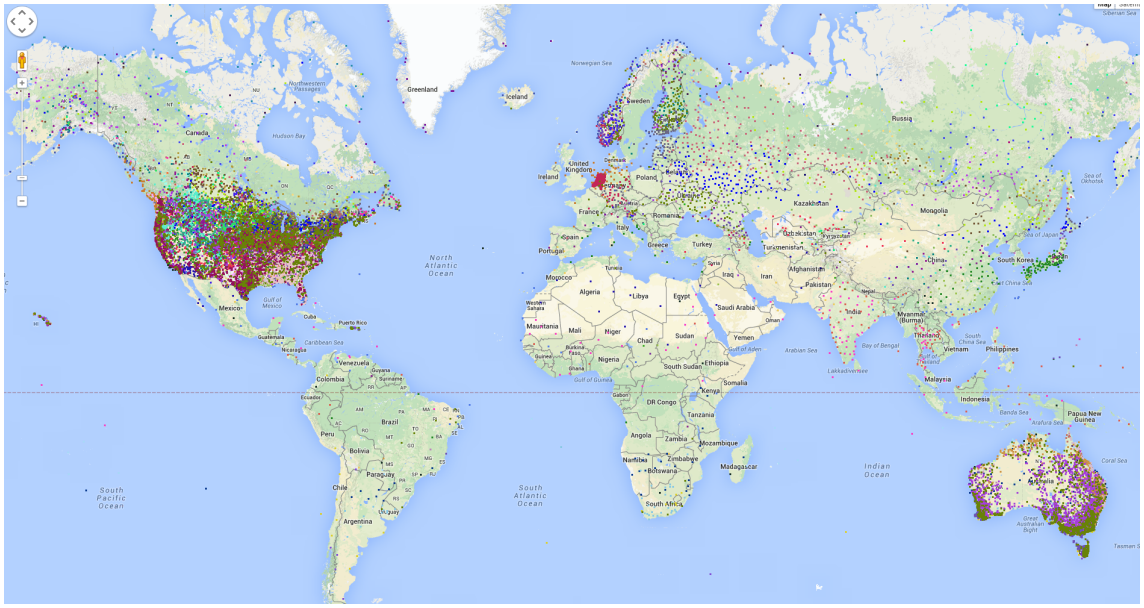


1993

Figure 1: Clustering results of 1983 and 1993.



2003



2013

Figure 2: Clustering results of 2003 and 2013.

- More clusters (which are different colors) are observed;
- Less mising points on the map;

3.3 Result of more than a century

At next step, we adjusted the parameter to less clusters. Because the former years have really small recorded weawther information, it's unnecessary to run many clusters. When we run with 90 clusters, 500 iterations to check if there is any similar pattern to the extent of a century. We have chosen the years 1876, 1896, 1904, 1940, and 2010 with the size shown in Table 3.3. The values are varying from $900K$ to $184M$[1].

Year	1876	1896	1904	1940	2010
Compressed	900KB	18MB	32MB	73MB	184MB
Original	5.5MB	110.2MB	201.7MB	441.4MB	1GB

Table 3: Size of dataset in selected years

Here we observed a similar result as before.

3.4 Result of recent 11 years

We also extracted the result from 2003 to 2013, made a video by Python which is now available on Youtube[7].

4 Conclusion

Althgouh the result of our project is quite subjective, there is an optimal number of clusters. After enough iterations, the result remains a good quality. For the ones with less number of clusters, it would take less time to compute while for bigger number of clusters, it would take more time. From [5]'s website, there is a note for the complexity of algorithm:

> The k-means problem is solved using Lloyds algorithm.
>
> The average complexity is given by $O(knT)$, were n is the number of samples and T is the number of iteration.
>
> The worst case complexity is given by $O(n^{(k+2/p)})$ with n = n_samples, p = n_features. (D. Arthur and S. Vassilvitskii, 'How slow is the k-means method?' SoCG2006)
>
> In practice, the k-means algorithm is very fast (one of the fastest clustering algorithms available), but it falls in local minima. Thats why it can be useful to restart it several times.

Where in our experiment, we choose 150 clusters and 500 iterations. The solution quality would improve if we use more 'useful' data. We throw away the station which have huge number of missing values. The more 'useful' data we use, the more computational effort would take. So that's the trade-off.

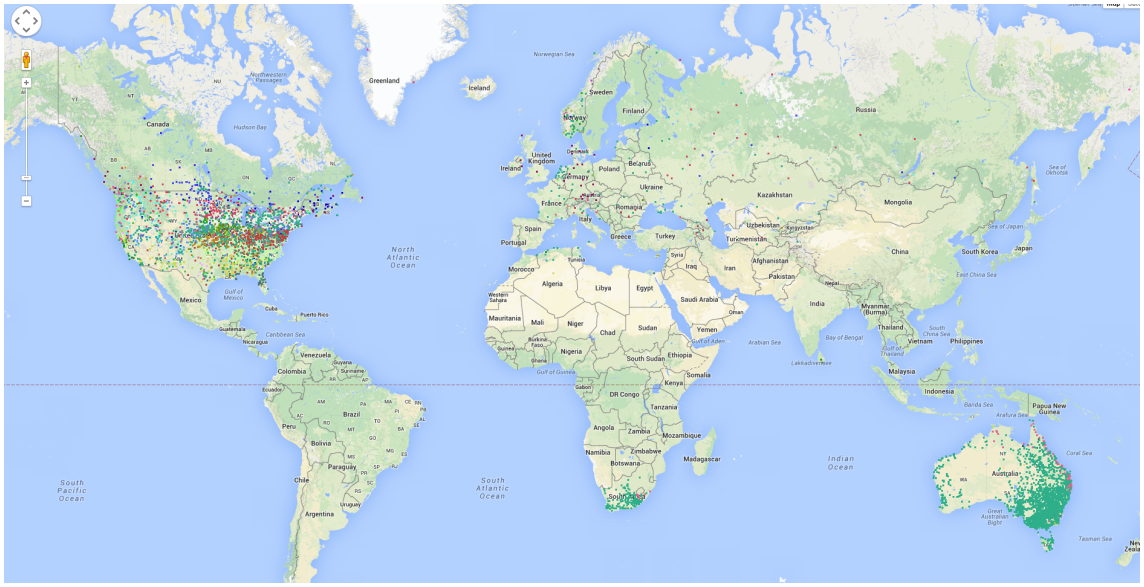
4.1 Difficulties

We encountered several difficulties during the projects. The top major ones is following:

- The size of Dataset is too big to fit in the memory;
- Too in-efficient to run K-means on the whole dataset;
- Install Python tools in the remote machine.

[1]The results before 1876 is meaningless since there were not enough data.

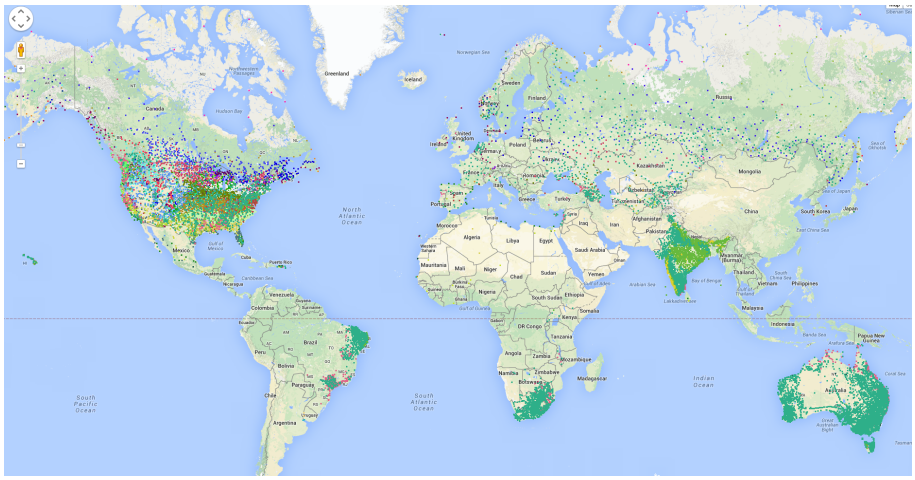
1876



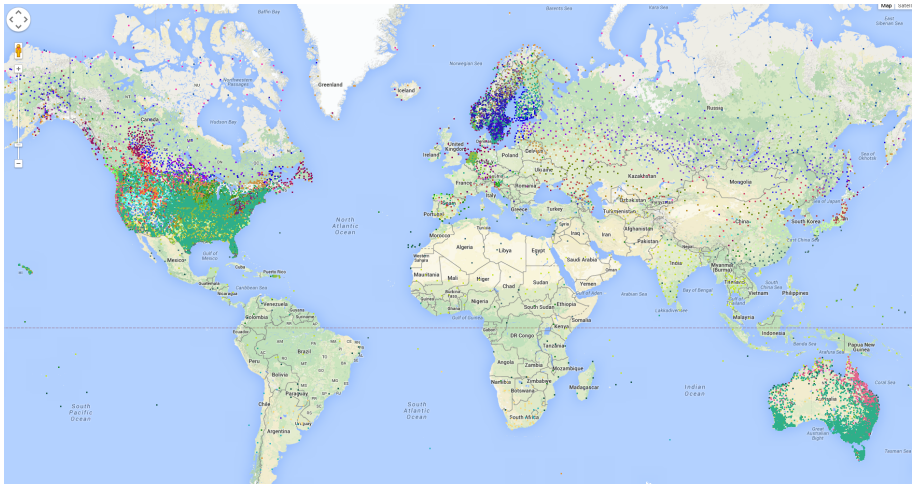
1896

Figure 3: Clustering results of 1876 and 1896.

1904



1940



2010

Figure 4: Clustering results of 1904, 1940 and 2010.

4.2 Lecture Learned

From this project, we learned quite a lot lessons, which could be improved when we do future big data project:

- Debug thoroughly before implementing it in a distributed system;
- Save the log whenever necessary.

5 Future Work

In future, we have two directions to go to improve our system: one is related with features, the other is about the clustering algorithm:

- Create more features and do feature selection;
- Other clustering algorithms, maybe GMM.

References

[1] http://aws.amazon.com/elasticmapreduce/

[2] Peterson, Thomas C., and Russell S. Vose. "An overview of the Global Historical Climatology Network temperature database." Bulletin of the American Meteorological Society 78.12 (1997): 2837-2849.

[3] https://boto.readthedocs.org/en/latest/

[4] http://aws.amazon.com/de/s3/

[5] http://scikit-learn.org/stable/

[6] https://developers.google.com/maps/?hl=de

[7] https://www.youtube.com/watch?v=xQfEk978IE4&list=UUkzQg0KkXhibidSnmizd4Vg&index=2

[8] http://www.ngdc.noaa.gov/mgg/image/color_etopo1_ice_low.jpg