Web-Scale *k*-means++ Clustering on PowerPS

Author: Zhanhao (Jasper) Liu Supervisor: Prof. James Cheng
Dept. of Computer Science & Engineering, The Chinese University of Hong Kong
zhliu6@cse.cuhk.edu.hk

Introduction

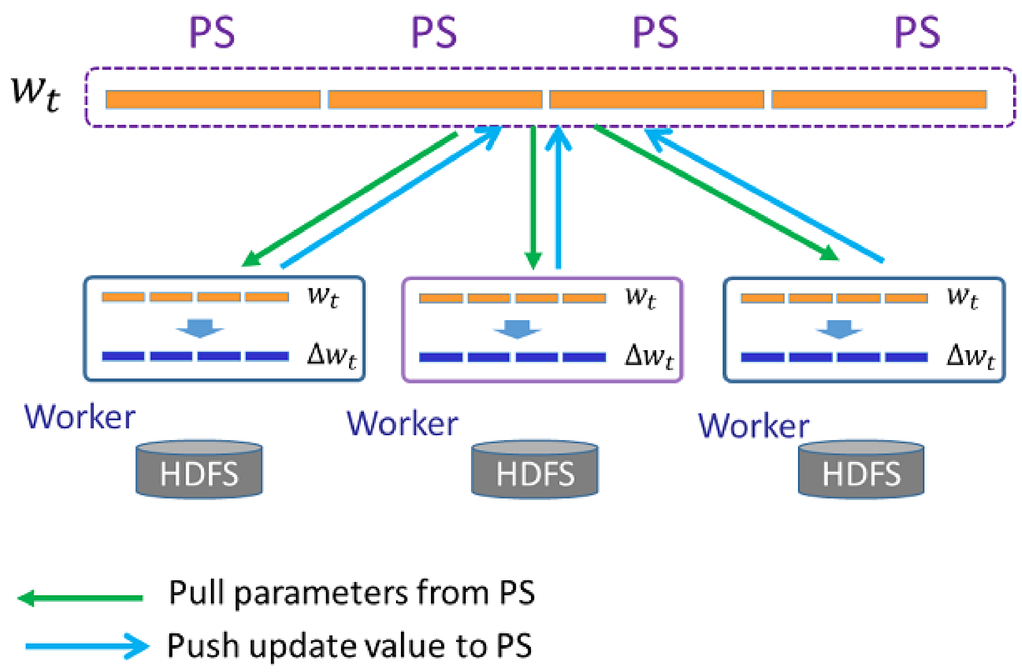
- PowerPS is a general and scalable Parameter Server based system for distributed machine learning. It provides flexible control over computing resources with a novel multi-stage design.
- *k*-means algorithm is the most popular clustering method and was identified as one of the top 10 algorithms in data mining. However, the traditional k-means algorithm encounters certain performance bottlenecks when it comes to the huge data and model size. In this research, we presented an efficient, scalable and distributed implementation of *k*-means++ Clustering algorithm on PowerPS.

Scalable *k*-means++ Initialization[1]

$C \leftarrow$ sample a point uniformly at random from X
$\psi \leftarrow \phi_X(C)$ ▷ Within set sum of squared errors
for $O(\log \psi)$ **do**
  $C' \leftarrow$ sample each point $x \in X$ independently with probability $p_x = \frac{l \cdot d^2(x, C)}{\phi_X(C)}$
  $C \leftarrow C \cup \{C'\}$
end for

Parameter Server

Model Storage We store the features of all the centers **C** and a vector **v** of size k recording the number of data in each cluster in the parameter server adopting chunk-based characteristic of PowerPS. (Each center and vector as a chunk)

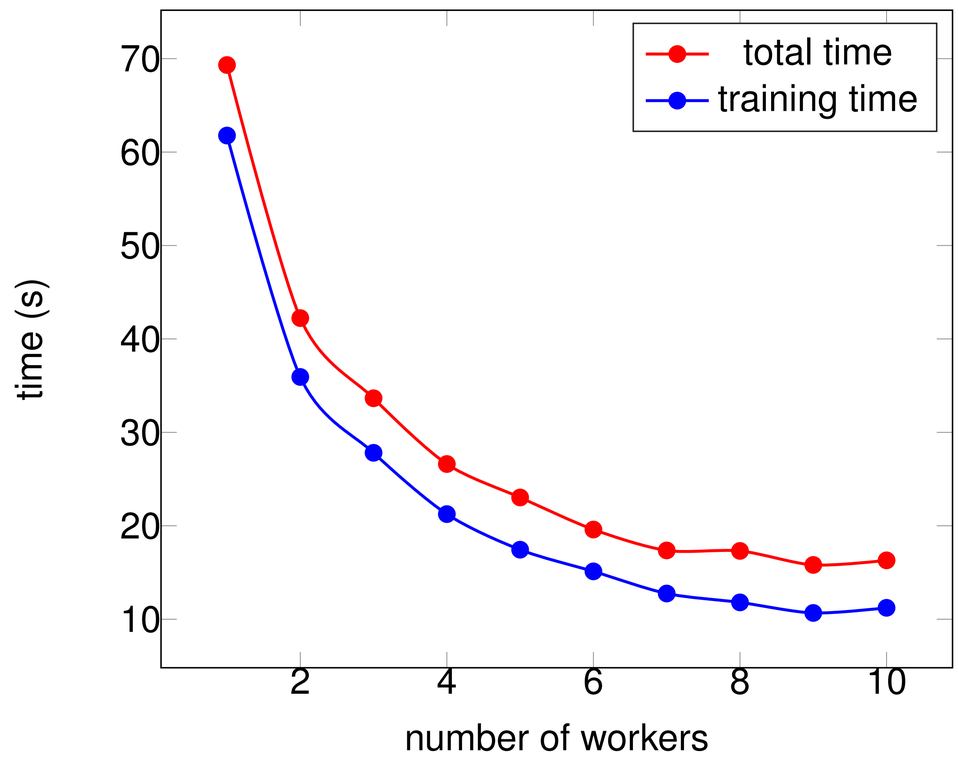


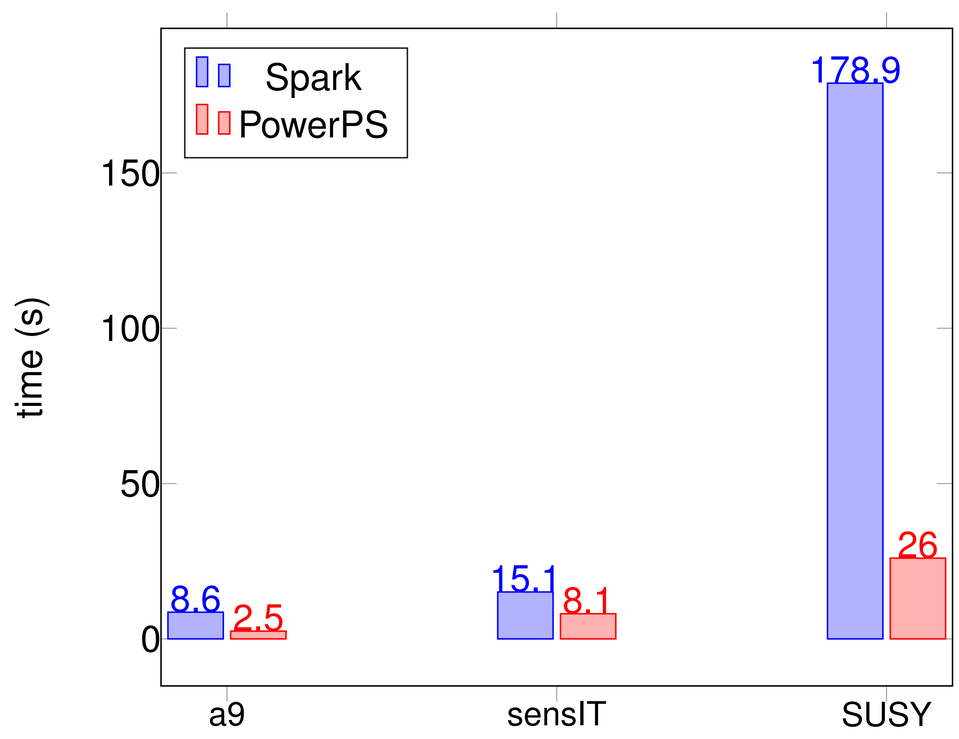
Algorithm

Given: k, mini-batch size b, iterations t, dataset X, $\mathbf{v} \leftarrow 0$
Initialize each $\mathbf{c} \in C$ using Scalable *k*-means++ algorithm
KV-Worker $r = 1, ..., m$:
for $i = 1$ to t **do**
  Pull initial cluster centers C and $\mathbf{v}$ from KV-Servers
  $M \leftarrow b$ examples picked randomly from X ▷ Mini-batch updating[2]
  for $\mathbf{x} \in M$ **do**
    $\mathbf{c} \leftarrow f(C, \mathbf{x})$ ▷ Cache the center nearest to x
    $\mathbf{v}[\mathbf{c}] \leftarrow \mathbf{v}[\mathbf{c}] + 1$
    $\eta \leftarrow \frac{1}{\mathbf{v}[\mathbf{c}]}$ ▷ Update learning rate
    $\Delta w_r \stackrel{+}{=} -\eta(\mathbf{c} - \mathbf{x})$
  end for
  push Δw_r^i to KV-servers
end for
KV-Server:
Receive initial cluster centers (w^0) from KV-Worker
for $i = 1$ to t **do**
  Send w^{i-1} to each KV-Worker
  Receive Δw^i from KV-Worker and update w^i
end for

Performance

Scalability The scalability of a distributed algorithm can be roughly measured by the linear relationship between the number of workers and the running time for the same data set.





Convergence speed Web-Scale *k*-means++ Clustering on PowerPS outperforms the *k*-means library on Spark[3] significantly in terms of Convergence time in several experiments on different data sets.

Conclusion

In this project, we presented a distributed Web-Scale *k*-means++ clustering using parameter server and adopted the multi-stage feature of PowerPS to accelerate the computation and make the most of the computing resources. In terms of scalability and convergence speed, this implementation outperforms the state of art MLlib on Spark platform.

Acknowledgements

The author would like to express his special thanks to Tatiana Jin, Yidi Wu, Tommy Tu, Yuzhen Huang and others in Husky Team for their kind guidance and generous assistance. He also likes to thank Prof. James Cheng for giving him such a great opportunity to explore about distributed system and machine learning.

References

[1] Bahmani, Bahman and Moseley, Benjamin and Vattani, Andrea and Kumar, Ravi and Vassilvitskii, Sergei: *Scalable k-means++*
[2] Sculley, D.: *Web-scale k-means Clustering*
[3] Zaharia, Matei and Chowdhury, Mosharaf and Franklin, Michael J. and Shenker, Scott and Stoica, Ion: *Spark: Cluster Computing with Working Sets*, `https://spark.apache.org/`