Deep Learning Service for Efficient Data Distribution Aware Sorting

Presenter: Xiaoke Zhu

Beihang University

November 23, 2024

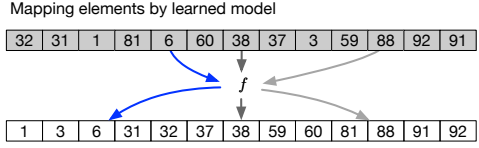
Background

- Fundamental CS problem
- Traditional sorting algorithm: Quick Sort, Merge Sort, QuickX Sort ...
- ML-enhanced algorithms: Learned Data Structures & Algorithms, SageDB Sort ...
- Sorting is a well-studied topic, but we argue that leveraging ML models offers a way to further accelerate it.

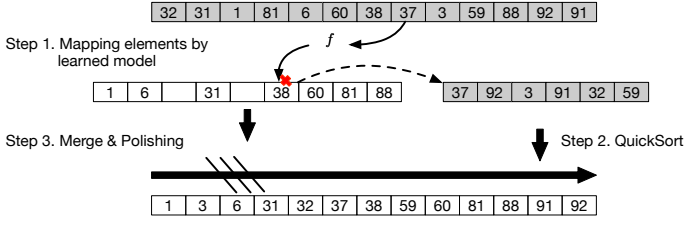
Motivation

Learned Sorting

- Train a model f.
- Use f to predict the final position of each key in the sorted output.



SageDB Sort



Limitations of SageDB Sort

A toy example

- QuickDraw game dataset Google Creative Lab

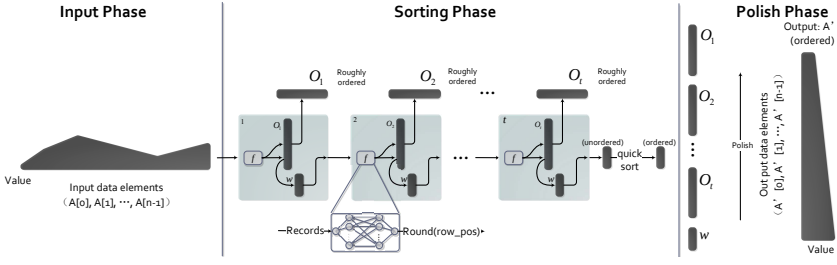
  50,426,265 records
  schema: 'key-id', 'word', 'country code', 'timestamp', 'recognized'

- Near 10% elements are conflicts
- SageDB Sort addresses collisions using traditional sorting algorithms, which incurs computational overhead when collisions is too large

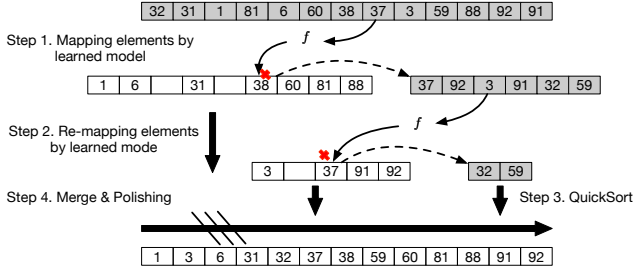
Algorithm name	Time (sec.)	Sorting Rate (elements/sec.)	Conflicting rate(%)
std::heap sort	13.46	3746.44	-
std::sort	23.71	2127.19	-
SageDB Sort	10.53	4790.125	9.16

NN-sort

Architecture



Example



The complexity

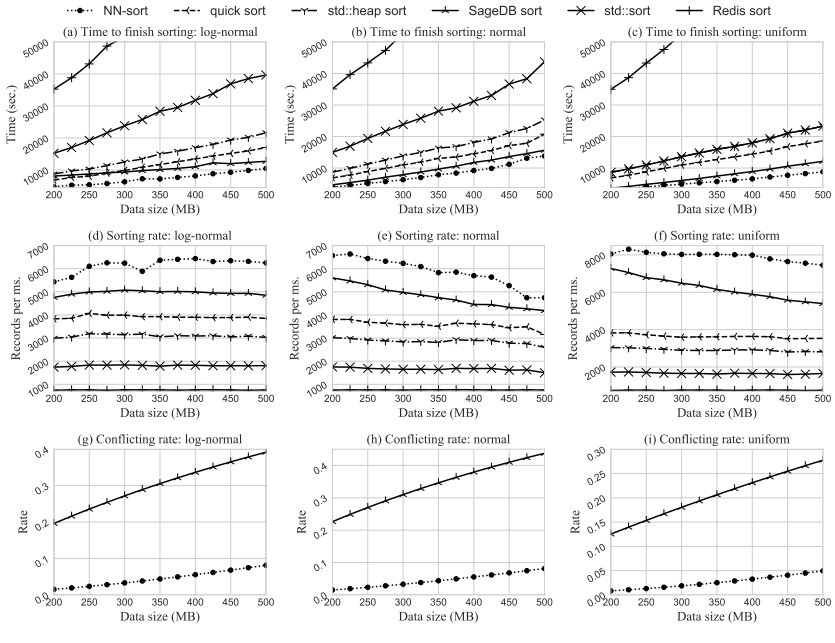
symbols	notations
n	the amount of data elements to be sorted
σ_i	collision rate per iteration
e_i	the number of data elements that were out-of-order in the i-th iteration
ϵ	the predefined limit of iterations
t	the number of completed iterations
θ	The operations required for data to pass through f

$$T(n, e, \sigma, t, \theta) = \begin{cases} 1, & if\ n = 1 \\ C_1 n^2 + C_2 n log n + C_3 n, & if\ n > 1 \end{cases}$$

$$C_1 = [\frac{1}{2} \sum_{i=1}^{t} e_i (1 - \sigma_i)(\prod_{j=1}^{i-1} \sigma_j)^2], C_2 = \prod_{j=1}^{t} \sigma_j$$

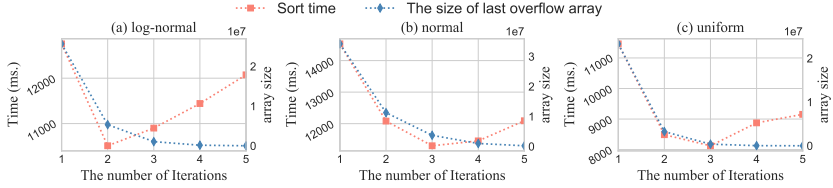
$$C_3 = \sum_{i=1}^{t} [\theta \sum_{j=1}^{i} \sigma_j + (1 - e_i)(1 - \alpha_i) \prod_{j=1}^{i-1} \sigma_j + \prod_{j=1}^{i} \sigma_j] + (\prod_{j=1}^{t} \sigma_j) log(\prod_{j=1}^{t} \sigma_j)$$

Experimental Results: overall performance

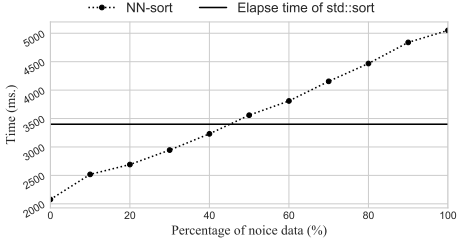


Experimental Results: other results

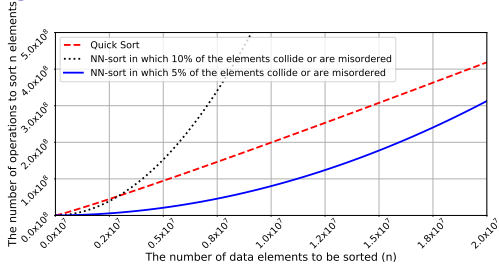
Impact of Iterations



Impact of data distribution



Operations between traditional sorting algorithm and NN-sort



Thanks