CS 561 Data Systems Architecture

Project 0

Implementation of a Zone Map

1. Workloads

Average execution time for the workloads are as follows :-

W1

Time taken to perform point queries from zonemap = 310.333 microseconds

Time taken to perform range query from zonemap = 0 microseconds

W2

Time taken to perform point queries from zonemap = 0 microseconds

Time taken to perform range query from zonemap = 635773 microseconds

W3

Time taken to perform point queries from zonemap = 0 microseconds

Time taken to perform range query from zonemap = 6.33522e+06 microseconds

W4

Time taken to perform point queries from zonemap = 1.2587e+07 microseconds

Time taken to perform range query from zonemap = 0 microseconds

W5

Time taken to perform point queries from zonemap = 0 microseconds

Time taken to perform range query from zonemap = 1.18472e+06 microseconds

W6

Time taken to perform point queries from zonemap = 0 microseconds

Time taken to perform range query from zonemap = 6.82124e+06 microseconds

```
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 10000 -R 0 -s 0 --sort
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main true
Time taken to perform point queries from zonemap = 310.333 microseconds
Time taken to perform range query from zonemap = 0 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 0 -R 1000 -s 0.001 --sort
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main true
Time taken to perform point queries from zonemap = 0 microseconds
Time taken to perform range query from zonemap = 635773 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 0 -R 1000 -s 0.1 --sort
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main true
Time taken to perform point queries from zonemap = 0 microseconds
Time taken to perform range query from zonemap = 6.33522e+06 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 10000 -R 0 -s 0
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main false
Time taken to perform point queries from zonemap = 1.2587e+07 microseconds
Time taken to perform range query from zonemap = 0 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 0 -R 1000 -s 0.001
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main false
Time taken to perform point queries from zonemap = 0 microseconds
Time taken to perform range query from zonemap = 1.18472e+06 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./workload_generator -N 5000000 --UB 3000000 -
P 0 -R 1000 -s 0.1
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2> ./main false
Time taken to perform point queries from zonemap = 0 microseconds
Time taken to perform range query from zonemap = 6.82124e+06 microseconds
PS C:\files\2_Subjects\Spring_2023\CS_561_Data_System_Architecture\Project_0\cs561_templatezonemaps2>
```

2. Research Questions

2.1 Expected memory footprint to build the zone map

The implemented Zone Map has the following attributes.

a) **elements**

size1 = N * sizeof(T) bytes where T is the data type of each member of the **element** vector.

(b **zones**

Each object of zone data structure is of size = (**d * sizeof(T1)) + (3 * sizeof(T2)**) where T1 and T2 are the size of each member of the zone and size of (min, max, size) respectively.

size2 = num_zones * (d * sizeof(T1) + 3 * sizeof(T2)) bytes where num_zones is the number of zones in the Zone Map. num_zones = ceil(N/d)

(c **num_zones**

size3 = sizeof(T) bytes where T is the size of the data type of num_zones.

(d **num_elements_per_zone**

size4 = sizeof(T) bytes where T is the size of the data type of num_elements_per_zone.

(e **sorted_flag**

size5 = sizeof(T) bytes where T is the size of the data type of sorted_flag.

Total size = size1 + size2 + size3 + size4 + size5 bytes

2.2 If we only have a limited memory budget to build the zone map and when the memory budget of M bytes is smaller than the expected memory footprint, we can build the zone map by using the following methods :-

a) **Partitioning** - By partitioning technique, we can divide the dataset into smaller and manageable parts and we can build the zone map incrementally by loading a portion of data set into memory at any given time and repeating the process until the entire dataset has been processed.

b) **Downsampling** – By downsampling the data, we can select a subset of the data in such a way that the subset is a good representation of the whole dataset and it fits memory budget of M bytes. We can build the zone map using the subset of the data. This may result in a loss of accuracy.

c) **Divide-and-Conquer algorithms** – We can divide the zone map into smaller parts and we can build the zone map for each part separately. The resulting zone map of each part can be combined to form the final zone map.

d) **External Sorting** – We can sort the data into small subsets and the zone map can be built in stages.