COMP-6521

Advanced Database Technology and Applications

Lab Mini-Project 1

Two-Phase, Multiway Merge-Sort Method (Bag-based union*)*

Submitted To: Dr. Nematollaah Shiri

Submitted By: -

Name	Student ID	Email
Manpreet Singh Rana	40227463	rsingh.manpreet@gmail.com
Firas El Rachidi	40202702	firasrachidi1@gmail.com
Chuanyan Hu	40168030	amy.chuanyanhu@gmail.com
Karandeep Singh	40197407	karansaini591@gmail.com

Steps to run the program.

- Create a folder named “src” and place all the three .java files named Phase1.java, Phase2.java and Main.java inside it.
- To the same ‘src’ folder copy the ‘r1_large.txt’ and ‘r2_large.txt’ relations.
- Compile all the three java files using the command line:

  javac Phase1.java

  javac Phase2.java

  javac Main.java
- After compiling, run the Main.java source file using the below command.

  java Main

Program Description.

Phase 1 (Phase1.java):

Below steps represents the first phase of an external merge sort algorithm. It reads the input data in chunks, sorts each chunk in memory, and writes each sorted chunk to a temporary file on disk. The resulting sorted sublists will be merged in the second phase to produce the final sorted output file.

- The Phase1 class contains a static method sort that takes three parameters: the name of the input file to sort, a boolean flag to indicate whether to refresh the sublists directory, and a string that identifies the source of the input data.
- The MEMORY_SIZE constant defines the number of blocks available in memory, and SUBLIST_SIZE defines the size of each sublist on disk.
- The method starts by checking the refreshDirectory flag. If it's true, the sublists directory is emptied of any existing files, and the directory itself is created.
- The input file is read in chunks of SUBLIST_SIZE tuples, and each chunk is stored in a two-dimensional memory array with MEMORY_SIZE rows.
- Once the memory array is full, or the end of the input file is reached, each subarray is added to a list of sublists, which will be sorted and written to disk in the next step.
- Each sublist is sorted using the Arrays.sort method and then written to a temporary file on disk in the sublists directory. The filenames are generated based on the source parameter and a fileNumberIndex counter.
- The method returns the number of I/O operations performed during the sorting process.
- The fileNumberIndex, total_write_blocks, and output_records variables are used to keep track of the number of temporary files created, the total number of sublists written to disk, and the total number of records in the output file, respectively. These variables are defined as static class variables and are initialized to zero.

Phase 2 (Phase2.java):

Below steps represent the implementation of the second phase of External Merge Sort algorithm, which performs sorting of large data sets that cannot fit into the memory at once.

- In the run function, the input sublists are read from the 'sublists' directory created in phase 1 and stored in a list of sorted sublists.
- In a loop, each sublist in the sortedSublists list is merged with other sublists (up to MEMORY_SIZE - 1 sublists at a time) and a new sorted sublist is created.
- This sublist is passed to 'mergeSublists' method.
- The 'mergeSublists' method merges a set of sublists using a priority queue and writes the result to a new file in the 'mergedSublists_' + run directory.
- The merge process is performed by reading a block of data from each sublist, storing it in the main memory buffer, and adding the first element of each buffer to the priority queue.
- The minimum element from the priority queue is removed and written to the output file, and the corresponding buffer is updated by removing the written element and adding the next element from the same sublist, if available.
- The process continues until all elements from all sublists are written to the output file.
- The method return from mergeSublists is pointed to the directory that contains the new sorted sublists, to be merged again in the next iteration of run operation.
- This process continues until all sublists are merged into a single sublist.
- A BagEntry class is defined to store the value of a record and the index of the sublist it belongs to, which is used in the priority queue to keep track of the minimum value.
- The total number of IO operations performed during the merge process is returned.

Time Calculations

The running time for Phase1 and Phase2 is calculated in two ways:

- Using System class of Java – This method involves starting a system timer just before the phase starts executing and stopping it after the phase ends running. The difference between stop and start is the recorded time.
- Assuming Megatron 747 metrices – The project requires to calculate time using average seek time, rotational delay and block transfer time. Since SSD’s don’t have spinning disks or moving parts, we took Megatron 747 as base and took its average seek time, rotational delay and block transfer time for time calculations. The blocks are assumed to be at random positions, so time is calculated with the formula x(s+r+t) where x is the number of blocks used in a phase, s is average seek time, r is average rotational delay and t is average block transfer time.

Experimental Results

Main Memory Size = 101 blocks

Tuple Count	***2,00,000***
Number of Disk I/O for phase 1	10,000
Number of Output Blocks written to the disk for phase 1	5000
Elapsed Time for Phase 1 (using System class)	1039 milliseconds.
Elapsed Time for Phase 1 (assuming Megatron 747 disk metrices and random R/W)	107600 milliseconds.
Number of Disk I/O for phase 2	25049
Number of Output Blocks written to the disk for phase 2	10000
Elapsed Time for Phase 2(using System class)	37505 milliseconds
Elapsed Time for Phase 2 (assuming Megatron 747 disk metrices and random R/W)	269538 milliseconds.
Total number of output blocks written on disk during both the phases	15000
Total time together taken by phase 1 and phase 2 to produce the output (using System class)	38544 milliseconds
Total time together taken by phase 1 and phase 2 to produce the output (assuming Megatron 747 disk metrices and random R/W)	377138.0 milliseconds

Main Memory Size = 51 blocks

Tuple Count	***2,00,000***
Number of Disk I/O for phase 1	10,000
Number of Output Blocks written to the disk for phase 1	5000
Elapsed Time for Phase 1 (using System class)	1067 milliseconds.
Elapsed Time for Phase 1 (assuming Megatron 747 disk metrices and random R/W)	107600 milliseconds.
Number of Disk I/O for phase 2	35101
Number of Output Blocks written to the disk for phase 2	15000
Elapsed Time for Phase 2 (using System class)	39490 milliseconds

Elapsed Time for Phase 2(assuming Megatron 747 disk metrices and random R/W)	377697 milliseconds.
Total number of output blocks written on disk during both the phases	20000
Total time together taken by phase 1 and phase 2 to produce the output (using System class)	40557 milliseconds
Total time together taken by phase 1 and phase 2 to produce the output (assuming Megatron 747 disk metrices and random R/W)	485297.52 milliseconds

Main Memory Size = 101 blocks

Tuple Count	***20,000***
Number of Disk I/O for phase 1	1000
Number of Output Blocks written to the disk for phase 1	500
Elapsed Time for Phase 1 (using System class)	188 milliseconds.
Elapsed Time for Phase 1 (assuming Megatron 747 disk metrices and random R/W)	10760 milliseconds.
Number of Disk I/O for phase 2	2504
Number of Output Blocks written to the disk for phase 2	1000
Elapsed Time for Phase 2 (using System class)	4792 milliseconds.
Elapsed Time for Phase 2 (assuming Megatron 747 disk metrices and random R/W)	26953 milliseconds.
Total number of output blocks written on disk during both the phases	1500
Total time together taken by phase 1 and phase 2 to produce the output (using System class)	4980 milliseconds.
Total time together taken by phase 1 and phase 2 to produce the output (assuming Megatron 747 disk metrices and random R/W)	37713.799999 milliseconds

Main Memory Size = 51 blocks

Tuple Count	***20,000***
Number of Disk I/O for phase 1	1000

Number of Output Blocks written to the disk for phase 1	500
Elapsed Time for Phase 1 (using System class)	273 milliseconds.
Elapsed Time for Phase 1 (assuming Megatron 747 disk metrices and random R/W)	10760 milliseconds.
Number of Disk I/O for phase 2	2509
Number of Output Blocks written to the disk for phase 2	1000
Elapsed Time for Phase 2 (using System class)	5003 milliseconds.
Elapsed Time for Phase 2 (assuming Megatron 747 disk metrices and random R/W)	27007 milliseconds.
Total number of output blocks written on disk during both the phases	1500
Total time together taken by phase 1 and phase 2 to produce the output (using System class)	5276 milliseconds.
Total time together taken by phase 1 and phase 2 to produce the output (assuming Megatron 747 disk metrices and random R/W)	37767.6 milliseconds

Results Screenshots

```
------------------------------------------- Input Configurations ---------------------------------
Memory Configuration : 101 blocks
Total # of input Tuples : 200000
------------------------------------------- Phase 1 ----------------------------------------------
Total # of disk I/o's during phase1: 10000
Number of output blocks written to disk for Phase1: 5000
Total # of output records: 200000
Elapsed time (using System class): 1039 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 107600.0 milliseconds.
------------------------------------------- Phase 2 ----------------------------------------------
Total # of disk I/o's during phase2: 25049
Number of output blocks written to disk for Phase2: 10000
Total # of output records: 200000
Elapsed time(using System class): 37505 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 269538 milliseconds.
-------------------------------------Phase1 & Phase2 summary ---------------------------------------
Total # of output blocks written on disk during both the phases: 15000
Total time together taken to produce the output: 38544 milliseconds(Using System class).
Total time together taken to produce the output: 377138.0 milliseconds(Using assuming Megatron 747 disk metrices and random R/W).
```

101 blocks – 200,000 Tuples

```
------------------------------------------- Input Configurations ----------------------------------
Memory Configuration : 51 blocks
Total # of input Tuples : 200000
------------------------------------------- Phase 1 -----------------------------------------------
Total # of disk I/o's during phase1: 10000
Number of output blocks written to disk for Phase1: 5000
Total # of output records: 200000
Elapsed time (using System class): 1067 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 107600.0 milliseconds.
------------------------------------------- Phase 2 -----------------------------------------------
Total # of disk I/o's during phase2: 35101
Number of output blocks written to disk for Phase2: 15000
Total # of output records: 200000
Elapsed time(using System class): 39490 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 377697 milliseconds.
--------------------------------------Phase1 & Phase2 summary ---------------------------------------
Total # of output blocks written on disk during both the phases: 20000
Total time together taken to produce the output: 40557 milliseconds(Using System class).
Total time together taken to produce the output: 485297.52 milliseconds(Using assuming Megatron 747 disk metrices and random R/W).
```

51 blocks – 200,000 Tuples

```
------------------------------------------- Input Configurations ----------------------------------
Memory Configuration : 101 blocks
Total # of input Tuples : 20000
------------------------------------------- Phase 1 -----------------------------------------------
Total # of disk I/o's during phase1: 1000
Number of output blocks written to disk for Phase1: 500
Total # of output records: 20000
Elapsed time (using System class): 188 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 10760.0 milliseconds.
------------------------------------------- Phase 2 -----------------------------------------------
Total # of disk I/o's during phase2: 2504
Number of output blocks written to disk for Phase2: 1000
Total # of output records: 20000
Elapsed time(using System class): 4792 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 26953 milliseconds.
--------------------------------------Phase1 & Phase2 summary ---------------------------------------
Total # of output blocks written on disk during both the phases: 1500
Total time together taken to produce the output: 4980 milliseconds(Using System class).
Total time together taken to produce the output: 37713.799999999996 milliseconds(Using assuming Megatron 747 disk metrices and random R/W).
```

101 blocks – 20,000 Tuples

```
------------------------------------------- Input Configurations ------------------------------------
Memory Configuration : 51 blocks
Total # of input Tuples : 20000
------------------------------------------- Phase 1 -------------------------------------------------
Total # of disk I/o's during phase1: 1000
Number of output blocks written to disk for Phase1: 500
Total # of output records: 20000
Elapsed time (using System class): 273 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 10760.0 milliseconds.
------------------------------------------- Phase 2 -------------------------------------------------
Total # of disk I/o's during phase2: 2509
Number of output blocks written to disk for Phase2: 1000
Total # of output records: 20000
Elapsed time(using System class): 2211 milliseconds.
Elapsed time (assuming Megatron 747 disk metrices and random R/W): 27007 milliseconds.
---------------------------------------Phase1 & Phase2 summary ----------------------------------------
Total # of output blocks written on disk during both the phases: 1500
Total time together taken to produce the output: 2484 milliseconds(Using System class).
Total time together taken to produce the output: 37767.6 milliseconds(Using assuming Megatron 747 disk metrices and random R/W).
```

51 blocks – 20,000 Tuples