## Chapter 4.5 Two - Pass Algorithm based on Hash

***Basic Thought:***

* If the data is too much to be put into main memory block, then using an available ***hash - key*** to hash all tuples of one or more Operation Objects. For all normal operations, there exists one hash - key method, it enables us to assign all tuples into one same bucket when under consideration.
* Operate one bucket each time. *(Under the situation of Binary Operation, deal with a pair bucket with the same hash value.)* Actually, the size of Operation Object has been decreased, the ratio of decreasing equals to the number of bucket. Its size is M.

### Chapter 4.5.1 Divide Relation Through Hash

***Assumption:***

* There exist Relation R, and M main memory blocks.
* h is hash function, hash function takes the whole Relation as the parameter.

***Algorithm:***

*Using M - 1 buffer blocks to initialize M - 1 buckets;*

*For each block b in Relation R DO BEGIN*

*Read block b into Mth buffer block;*

*For each tuple t in block b DO BEGIN*

*If the h(t) buffer block has no more space for the current tuple t THEN*

*BEGIN*

*Copy the h(t) buffer block to disk;*

*Using a new empty buffer block to initialize this buffer block;*

*END;*

*Copy tuple t into the h(t) buffer block;*

*END;*

*END;*

*END;*

*For each bucket DO*

*If the current bucket has tuples THEN*

*Write the current bucket back to disk;*

### Chapter 4.5.2 Remove Duplication Algorithm based on Hash

***Principle:***

* Divide Relation R into M - 1 buckets, two copies of the same tuples t will be put into the same bucket.
* Check one bucket for one time, and execute Remove Duplication Algorithm, and set the result as *Ri*.
* Merge all *Ri* collections and write the final collection back to disk.

***Cost:***

* B(R) <= M \* M.
* The total disk I/O is 3 \* B(R).

### Chapter 4.5.3 Grouping and Aggregation Algorithm based on Hash

***Procedure:***

1. Divide Relation R into M - 1 buckets, and the chosen hash function is decided by Grouping Key.
2. Then use *One - Pass Grouping Algorithm* to deal with each bucket.
3. In the second pass, only need one tuple in each group. So, even the size of bucket is larger than M, as long as the number of tuple in all groups is smaller than M, then we can use one pass to deal with the bucket.

***Cost:***

1. As long as the *B(R) <= M \* M*, then we can deal with each bucket in the main memory.
2. The total disk I/O is *3 \* B(R)*.

### Chapter 4.5.4 Union, Intersection and Difference Algorithm based on Hash

### Chapter 4.5.5 Hash Join Algorithm

Chapter 4.5.6 Save Disk I/O

Chapter 4.5.7 Conclusion on Algorithm based on Hash