Department of Computing

**Hong Kong Polytechnic University**

**Comp 5527 Mobile Computing and Data Management**

**Tutorial Three Sample Solutions**

Q1: Sample Solutions:

The overall average waiting time (**)

= (33/2) \* ( 4/21\* 3+ 1/21 \* 6 + 1/168\*24)

= 16.5

The overall average access time (**)

= ** +1

= 17.5

Q2. Sample Solutions:

* The access probabilities for items in groups A, B and C are

qa=4/21, qb=1/21, qc=1/168.

* The qa:qb:qc=32:8:1
* The square root of qa:qb:qc= 5.6569: 2.8284:1
* Choose the frequencies: fa=6, fb=3, fc=1
* Compute the cycle length which is the number of broadcasted data items when each original data item has been send at least once:

h=6\*3+3\*6+1\*24=60

* Spacing of items in A, B and C are 10, 20, 60
* Assigning the copies into the broadcast cycle positions with the objective of matching the equal-spacing criterion.
* The broadcast schedule is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 5 | 8 | 1 | 26 | 12 | 15 | 9 | 6 | 10 | 2 | 18 | 22 | 25 | 26 |
| 9 | 6 | 10 | 2 | 3 | 13 | 16 | 26 | 7 | 11 | 9 | 19 | 23 | 2 | 5 |
| 26 | 7 | 11 | 9 | 4 | 14 | 2 | 5 | 8 | 17 | 26 | 20 | 24 | 9 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 11 | 9 | 28 | 31 |
| 8 | 21 | 26 | 29 | 32 |
| 10 | 2 | 27 | 30 | 33 |

Q3. Sample Solutions:

* Organize the data items with same access probability into three groups A, B and C . Each of them is a disk.

qa=4/21, qb=1/21, qc=1/168.

* The qa:qb:qc=32:8:1
* The square root of qa:qb:qc= 5.6569: 2.8284:1
* Choose the frequencies: fa=6, fb=3, fc=1
* Maxinum number of chunks on a disk=6
* Numbers of chunks for each disk are 1, 2 and 6.
* Chunk size for A, B and C are 3, 3, 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 5 | 1 | 12 | 26 | 11 | 15 | 9 | 6 | 18 | 2 | 8 | 22 | 24 | 26 |
| 9 | 6 | 3 | 2 | 8 | 13 | 16 | 26 | 7 | 19 | 9 | 10 | 23 | 2 | 5 |
| 26 | 7 | 4 | 9 | 10 | 14 | 2 | 5 | 17 | 20 | 26 | 11 | 24 | 9 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 28 | 9 | 10 | 31 |
| 25 | 29 | 26 | 11 | 32 |
| 27 | 2 | 8 | 30 | 33 |

Q4. Sample Solutions:

Instead of replicating the entire path in *(1,m)* indexing, distributed indexing replicates only a part of it. Notice that root *I* is no longer replicated many times. The **offset** at the data bucket *3* will direct the client to ***second\_a1***. However, to make up for the lack of root preceding *second\_a1*, there is a small index called the control index within *second\_a1*. If the local index (in *second\_a1*) does not have a branch that would lead to the required record, then the control index is used to direct the client to a proper branch in the index tree. The **control index** in *second\_a1*, directs the client to ***i2***. At *i2* the root is available and the client makes the following probes: ***first\_a3****,* ***b8****,* ***c23***, and **bucket** ***66***.



In another case, a record in bucket *11* may be searched by the client, reading the bucket *second\_a1* would provide the client with the required information to successively tune in at *b2, c4*, and bucket *11*. In this case, having the root just before second\_a1 would have been a waste of space (this is true if the search key was in any data buckets *9* through *26*).

The control index in this example is like this:

