Data on Air: Organization and Access

T. Imielinski, S. Viswanathan, and B.R. Badrinath

Presented by Qinhai Xia

Motivation

Power conservation:

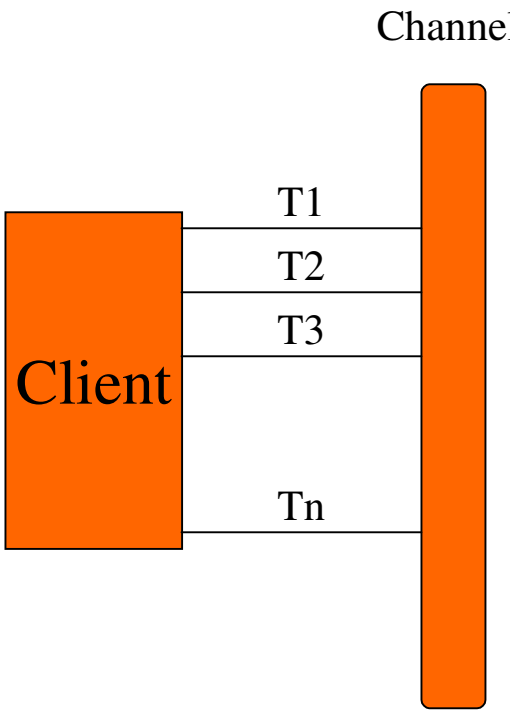
•Processor: AT&T Hobbit chip

Active mode -- 25mw, Doze mode -- 50 uw

•Broadcasting:

Quotrex system over FM channel

Parameter of Concern



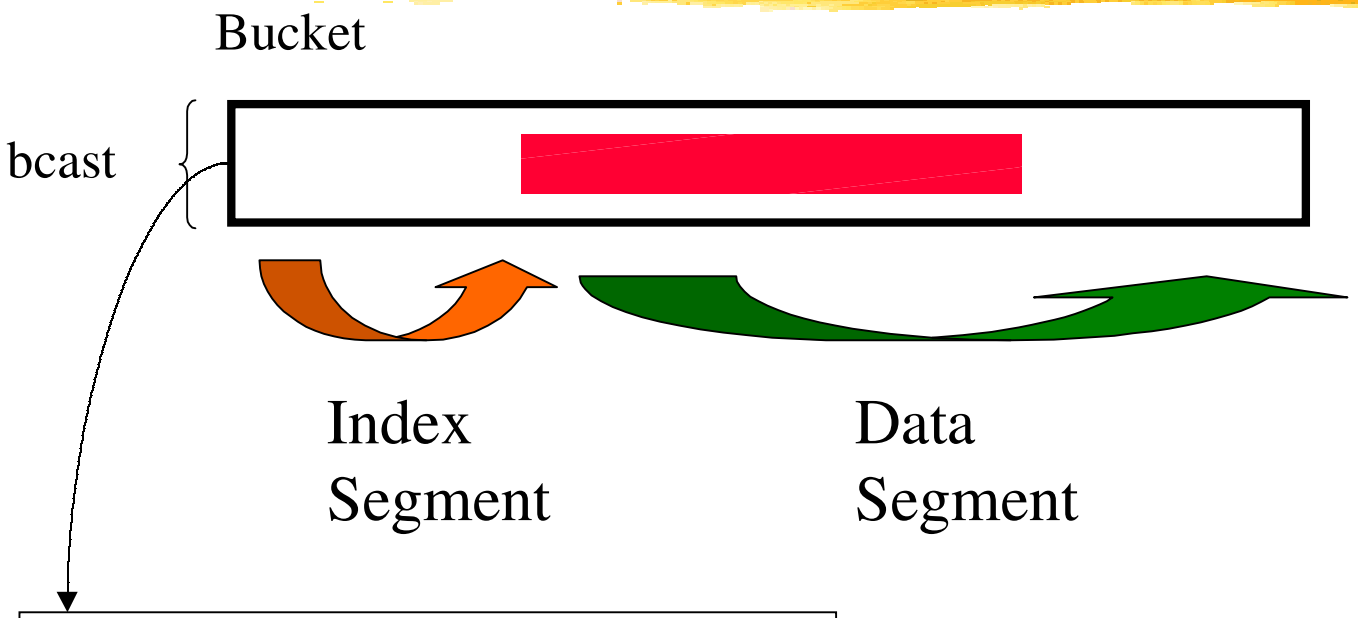
Tuning time :

T1 + T2 + T3 + … + Tn

Latency:

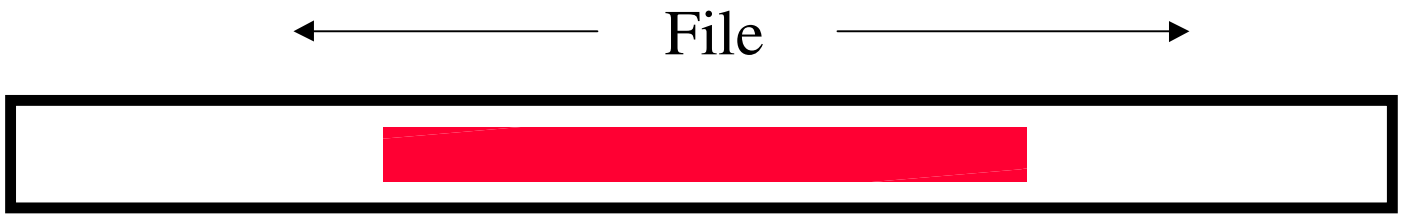
Tn - T1

Some Definitions



- Bucket ID
- Bcast pointer
- Index pointer
- Bucket type

Latency OPT

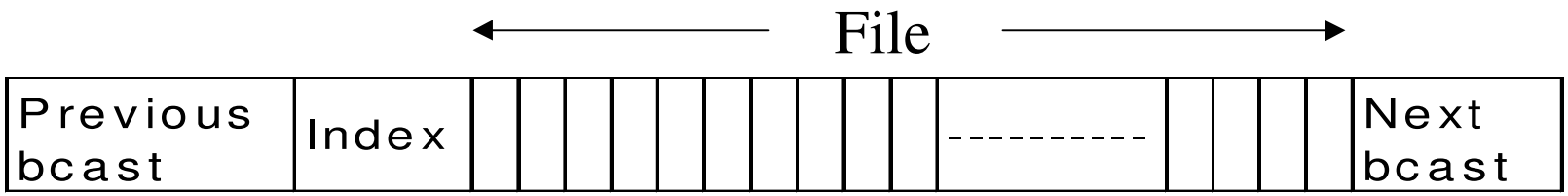


Latency is the best: No overhead for index

Latency = Data/2 + C

Tuning time = Data/2 + C

Tuning OPT



Tuning is the best:

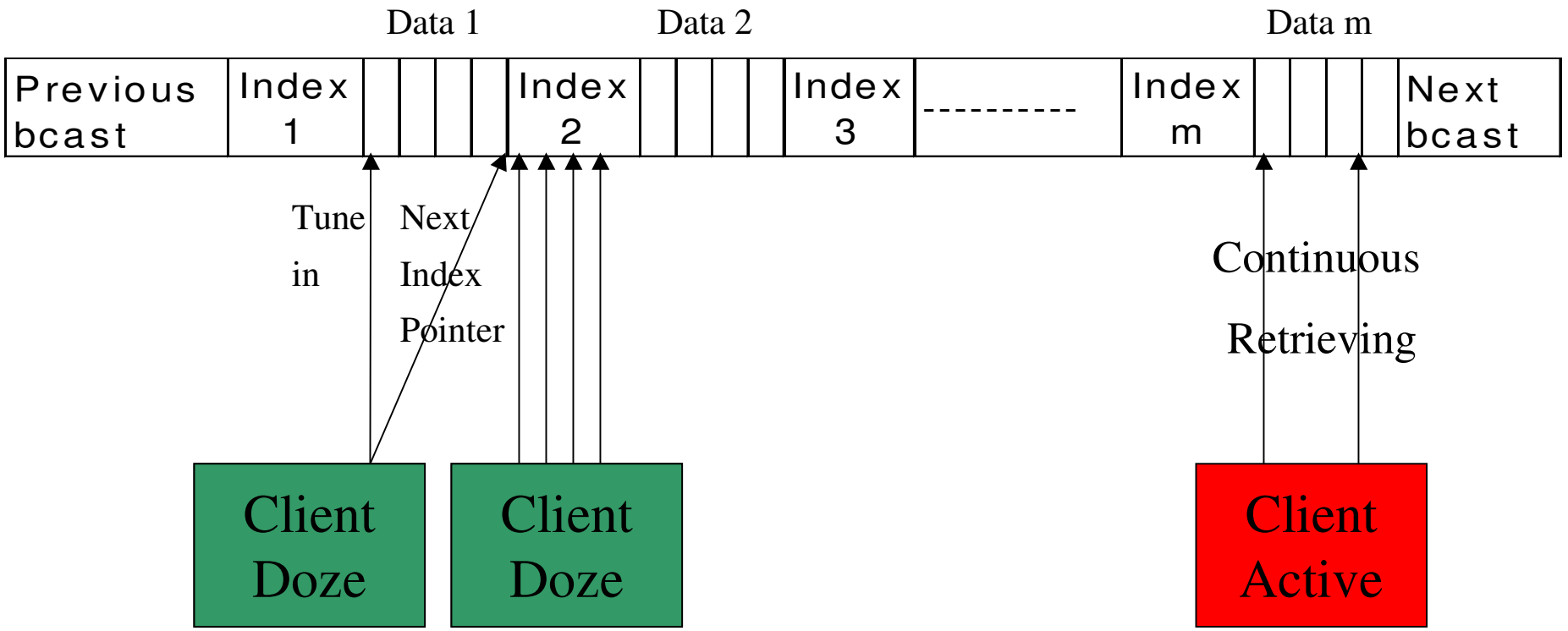
Latency = (Data + Index) / 2 + (Data + Index) / 2 + C

= Data + Index + C

Tuning time = k + C

k: number of levels in the index tree

(1, m) Indexing


Data 1
Data 2
Data m
Previous bcast
Index 1
Index 2
Index 3

Index m
Next bcast
Tune in
Next Index Pointer
Continuous Retrieving
Client Doze
Client Doze
Client Active

(1, m) Indexing Continue

Analysis:

Latency = (Index + Data/m)/2 + (m*Index + Data)/2 + C
= ((m+1) * Index + (1/m + 1) * Data)) / 2 + C

Tuning Time = 1 + K + C

K: level of the index tree

Distributed Indexing

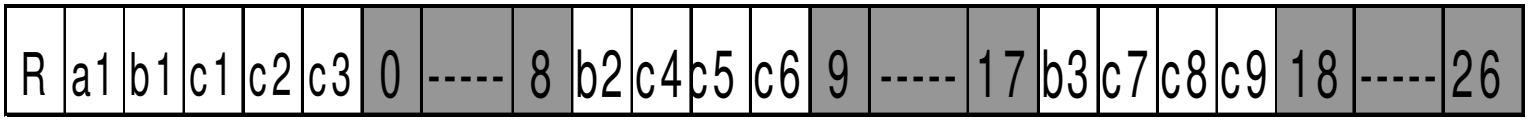
- Nonreplicated Distribution :

  Different index segments are disjoint
- Entire Path Replication:

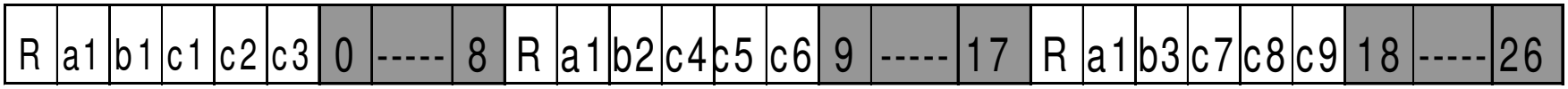
  The path from the root to an index bucket B is replicated
- Partial Path Replication (Distributed Indexing)

  Between two index buckets B and B’, it is enough to replicate just the path from the least common ancestor

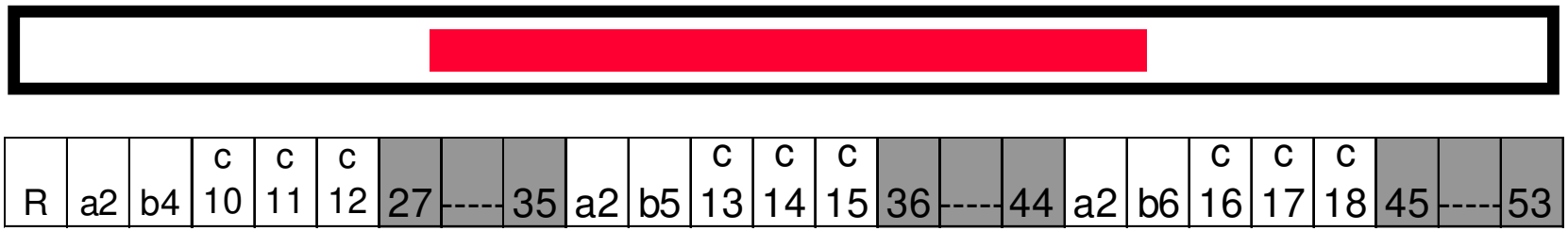
Comparison



Nonreplicated Distribution



Full Path Replication



Partial Path Replication (Distributed Indexing)

Distributed Indexing Analysis

- r: level of index tree
- Level[r]: number of nodes on the rth level of the index tree
- Index[r]: the size of top r level of the index tree
- Δ $Index_r$: additional index overhead

$$\Delta Index = Level[r+1] - 1$$

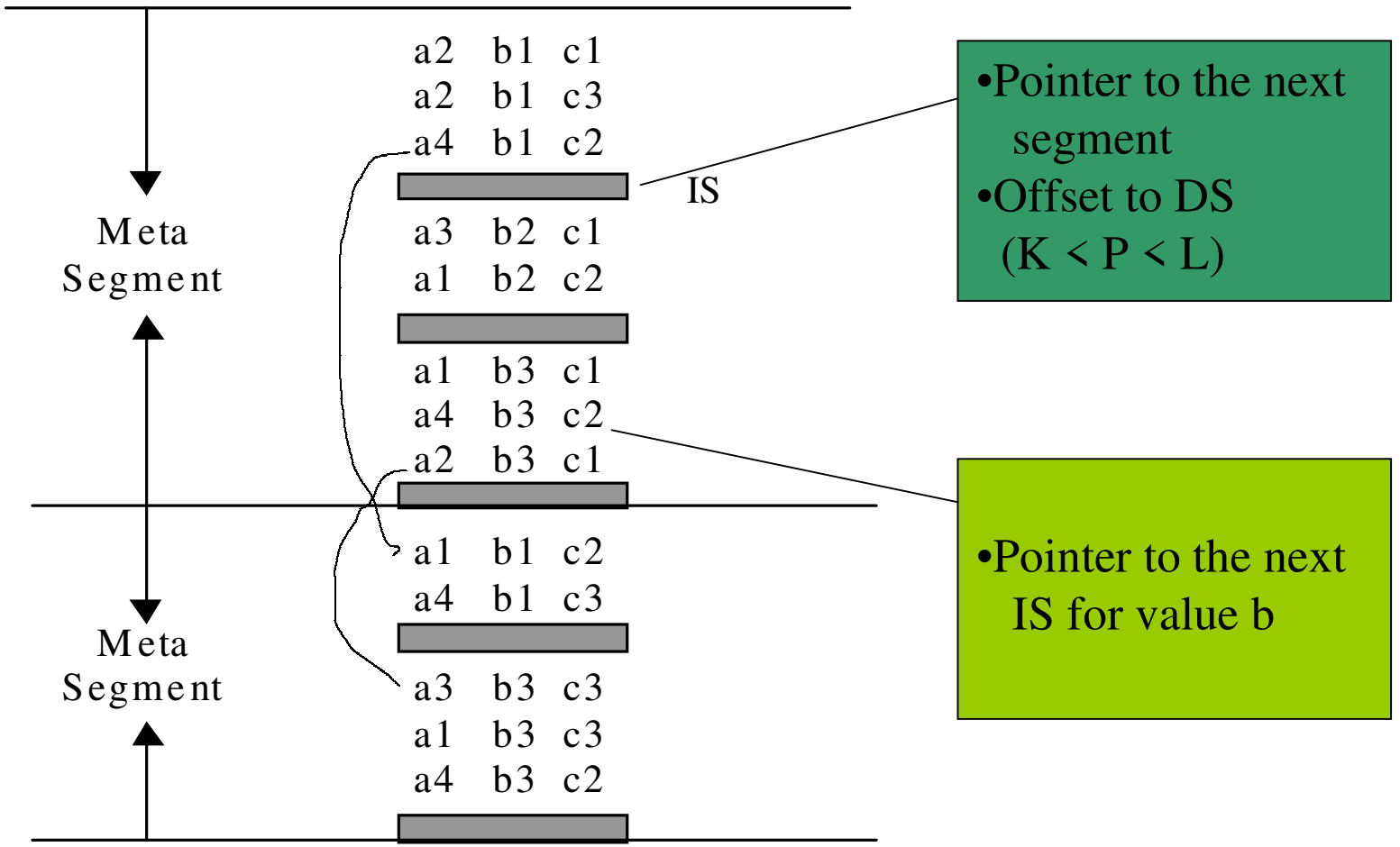
$$Latency = ((Index - Index[r])/ Level[r+1] + Data/Level[r+1] + Data + Index + \Delta Index_r) /2 + C$$

$$Tuning\ Time = 2 + k + C$$

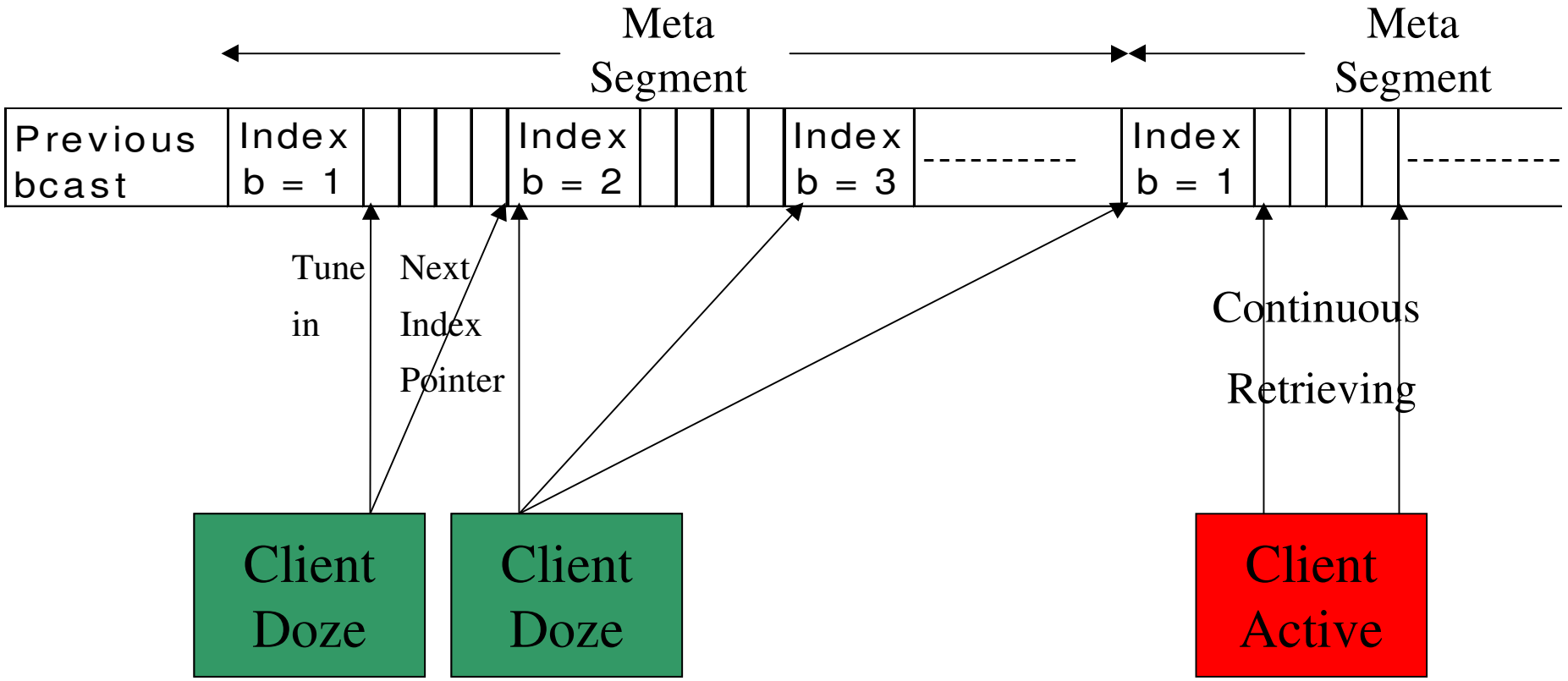
(1, M) vs. Distributed Indexing

- Distributed indexing usually has a much lower latency then (1,m)
- Distributed indexing just has one more bucket overhead than (1, m)

Nonclustered Indexing



Nonclustered Indexing Protocol



Nonclustered Indexing Analysis

$$Latency = \frac{1}{2} * \left(\frac{Index - Index_{[r]}}{Level_{[r+1]}} + \frac{Data}{M * Level_{[r+1]}} + M * (Index + Index) + Data \right)$$

$$TuningTime = 1 + (K + 1) + C + M$$

Real Example

Quotrex System

Information: 160,000 bytes

Bucket length: 128 bytes

Bandwidth: 10 kbps

Algorithm	Latency Seconds	Tuning Time Active Mode	Tuning Time Doze Mode	Power (J) Consumption
Latency_OPT	62.5	62.5	0	15.625
Tuning_OPT	130.3	0.4	129.9	0.106
(1, m) m = 5	90.9	0.5	90.4	0.13
Distributed Indexing	68.9	0.6	68.3	0.153
Non Cluster Latency_OPT	125	125	0	31.25
Non Cluster Tuning_OPT	187.5	2.2	185.3	0.559
Non Cluster Indexing	132.4	2.4	130	0.607