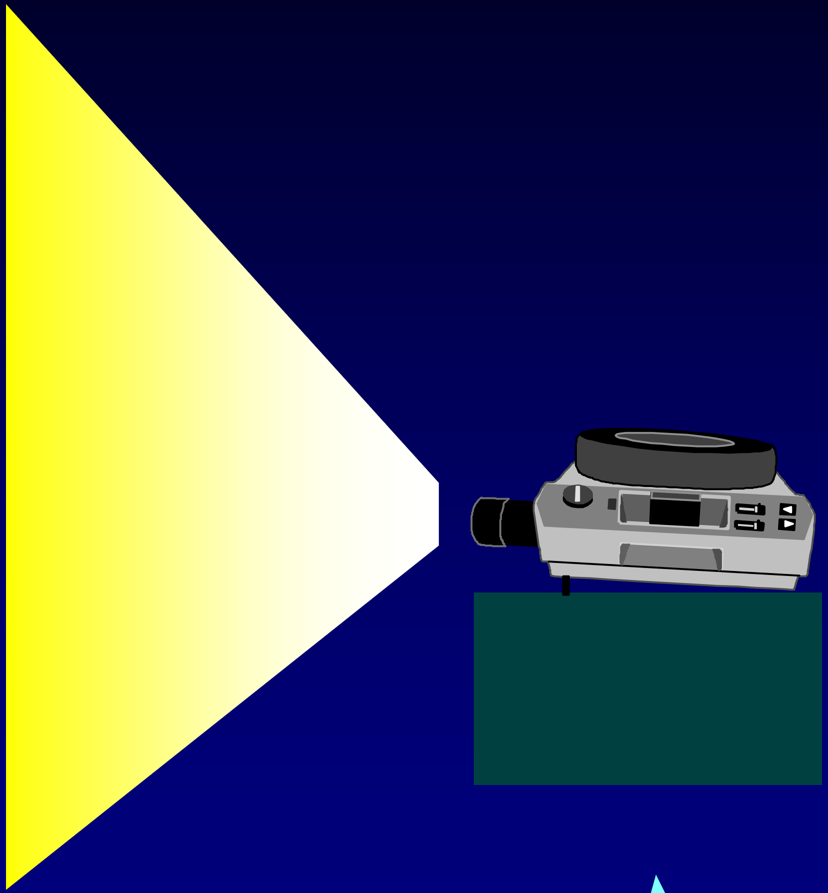
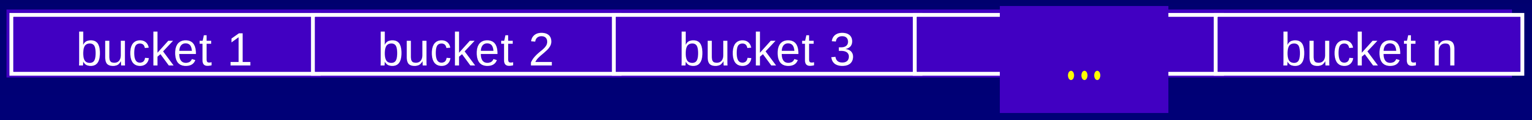
Direct Files : Outline

- Introduction
- Key-to-Address
- Hashing
- Collision
- Overflow Management
- Coalesced Chaining
- Extendible Hashing
- Linear Hashing



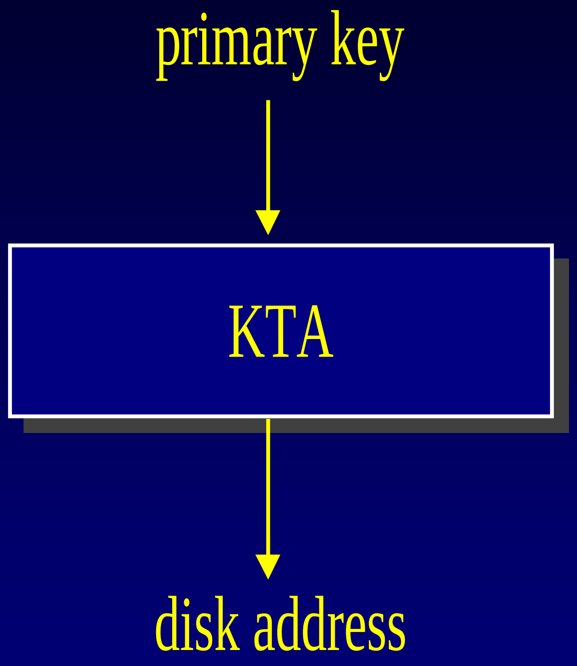
Direct Files

- Physical sequential files
  - 1 + log *NBLK* *rba*
  - (*NBLK*+1) / 2 *sba*
- Direct files
  - 1 *rba*



bucket 1 | bucket 2 | bucket 3 | ... | bucket n

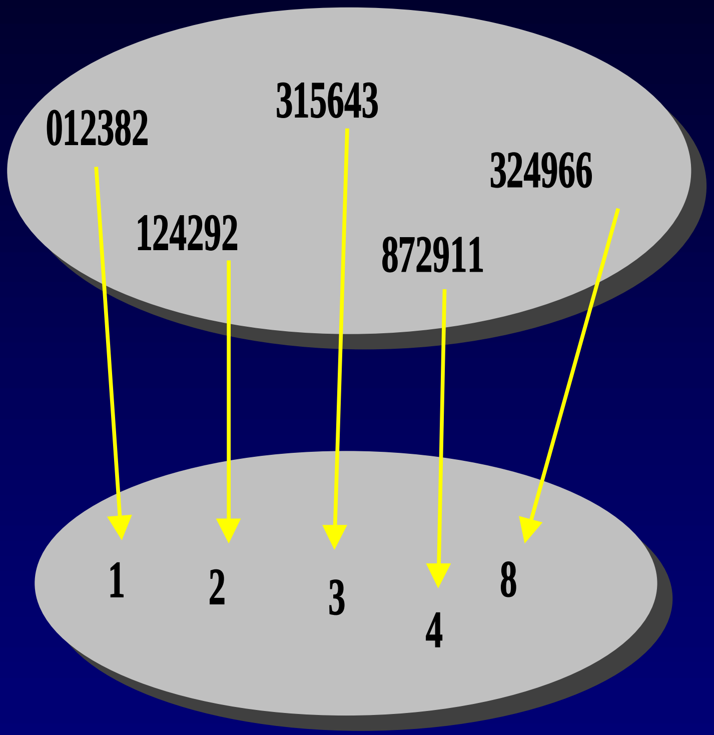
Key-to-Address Transformation



primary key

KTA

disk address



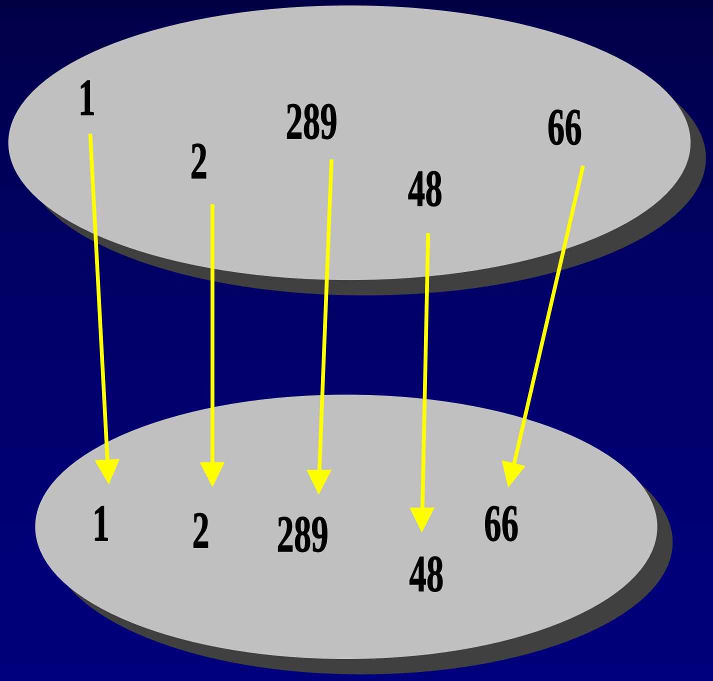
Notations

- NR : the number of records
- T : the number of allocated buckets
- B : bucket size
- LF : load factor

$$LF = \frac{NR}{T \cdot B}$$

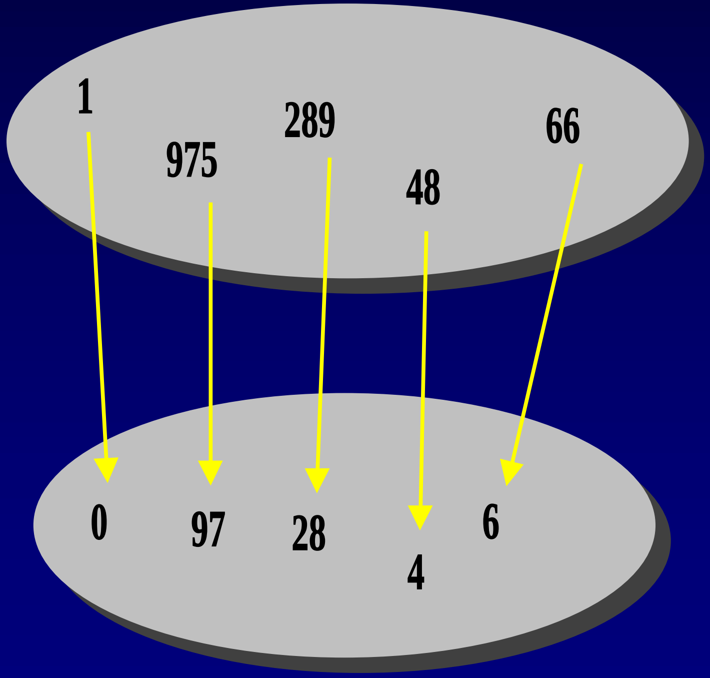
Key-to-Address

- A file of 1,000 records
- Primary keys are interger 0 to 999
- $f(key) = key$
- $T = 1{,}000,\ B = 1$
- $LF = 100\%$



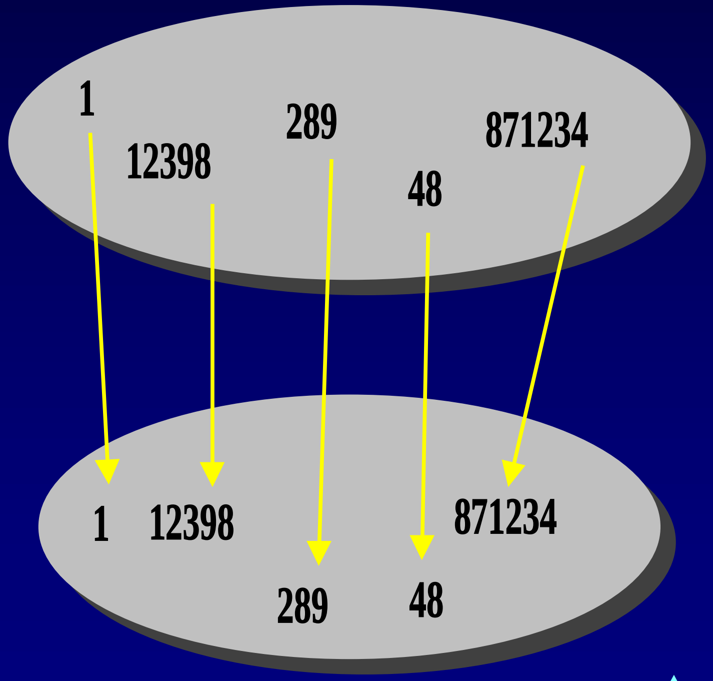
Key-to-Address

- A file of 1,000 records
- Primary keys are interger 0 to 999
- $f(key) = key \div 10$
- $T = 100$, $B = 10$
- $LF = 100\%$



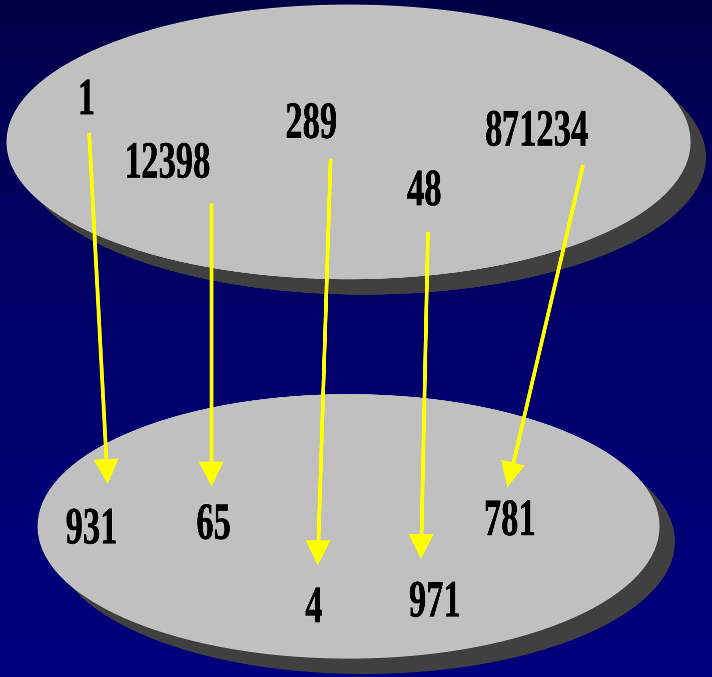
Key-to-Address

- ▼A file of 1,000 records
- ▼Primary keys are interger 0 to 999999
- ▼$f(key) = key$
- ▼T = 1,000,000, B=1
- ▼LF = 0.1%



Key-to-Address

- A file of 1,000 records
- Primary keys are interger 0 to 999999
- Let $T = 1{,}000$, $B=1$
- $f(key)$ = ???
- LF = 100%

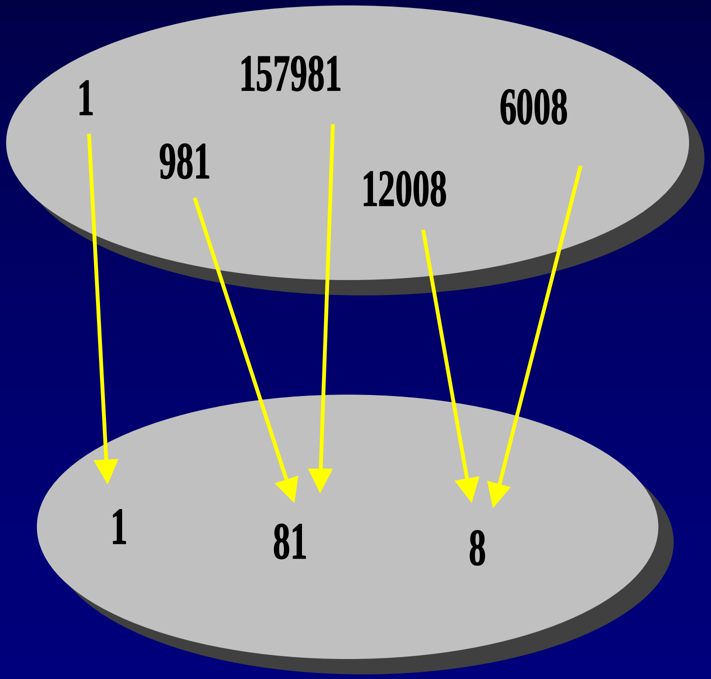


Hashing

- ▼A file of 1,000 records
- ▼Primary keys are interger 0 to 999999
- ▼Let T = 100, B=10
- ▼$f(key) = key \bmod 100$

Collision

Record overflow



Hashing

- To avoid collision
  - perfect hashing function
- To reduce *NOR*
  - spread out the records
  - bigger *T*
  - bigger *B*

NOR = the number of overflow records

T, *B*, *NOR*, and *LF*

- Increase T and B ⇨ decrease *NOR*, *LF*
- Decrease LF ⇨ decrease space utilization
- Unacceptable, if $LF < LF_0$
- *NOR* depends on *NR* / *T*
- Reduce *NOR*, by adjusting *T* and *B* while maintaining *LF*

Number of Collisions

$$LF = \frac{NR}{T \cdot B}$$

Random distribution

$Dh(k)$ is the expected number of buckets that receive exactly k records under the transformation h.

Ex. T=2000, NR=1500, hashing function is randomly distributed

Dh(0) = 945 Dh(1) = 708

Dh(2) = 266 Dh(3) = 66

Dh(4) = 12 Dh(5) = 2

Dh(6) = 1 Dh(k) = 0 , k>6

Number of Overflow Records

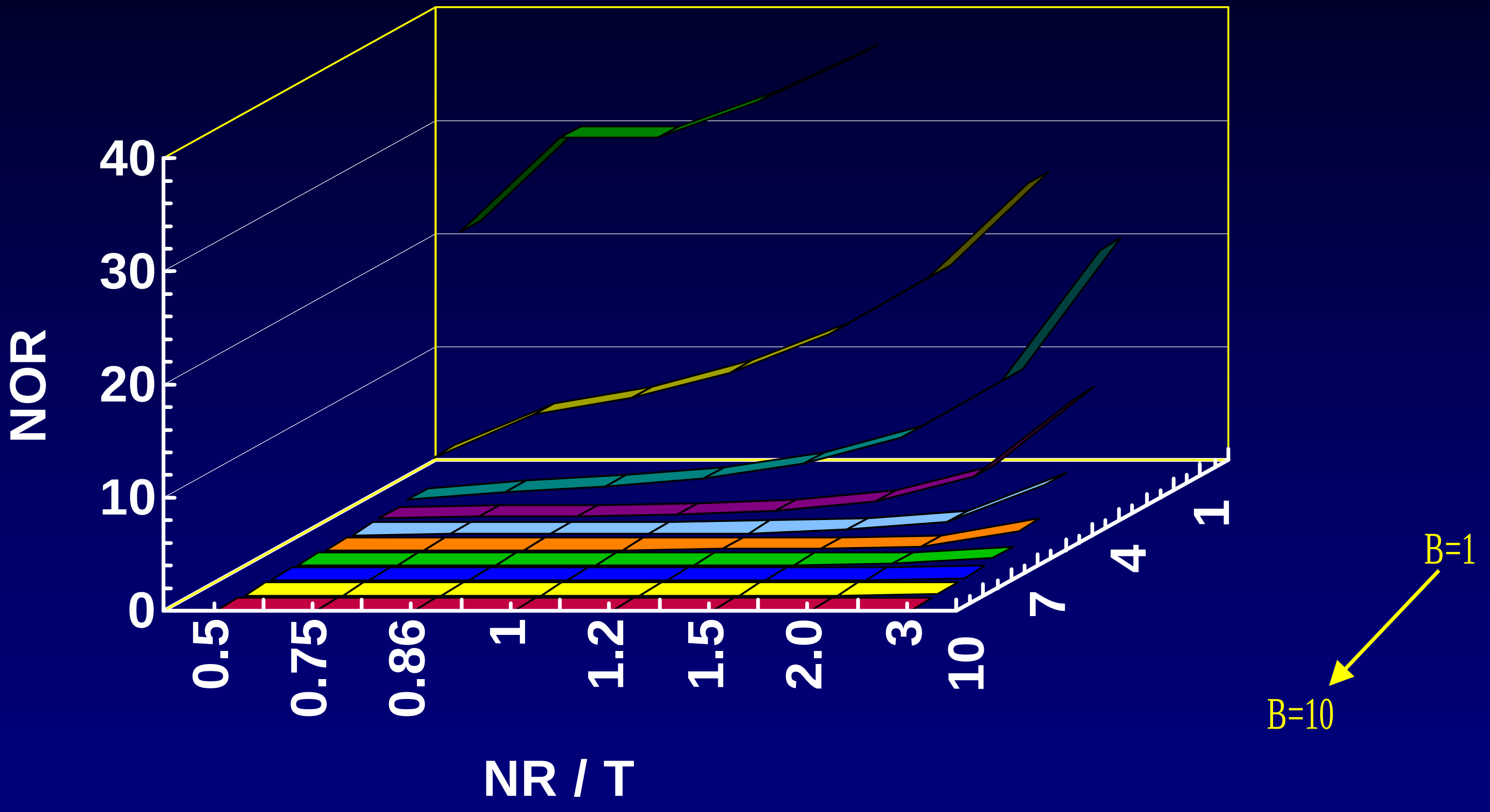
$$LF = \frac{NR}{T \cdot B}$$

Random distribution

Ex. T=2000, NR=1500, hashing function is randomly distributed and B=2.

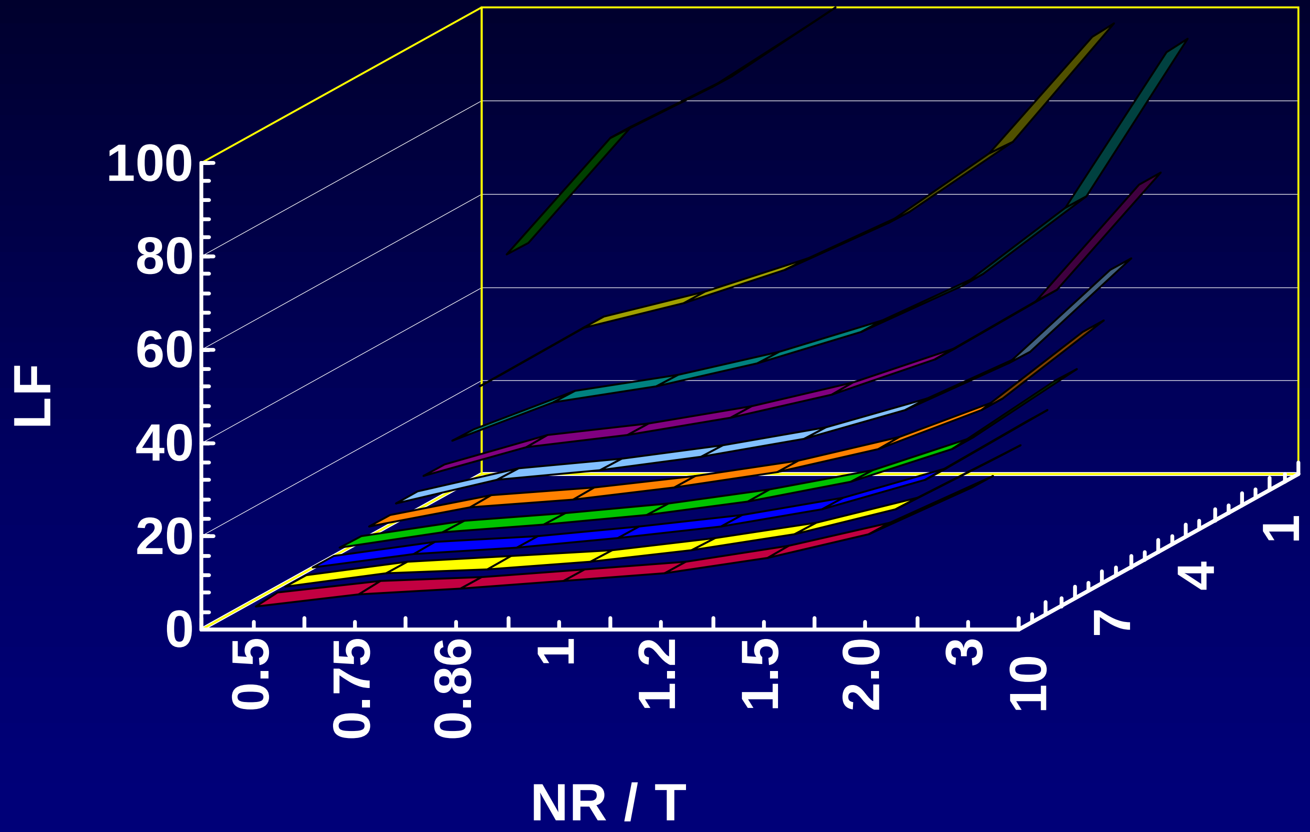
NOR = 3xDh(3) + 4xDh(4) + 5xDh(5) + 6xDh(6)
= 3x66 + 4x12 + 5x2 + 6x1
= 262 records

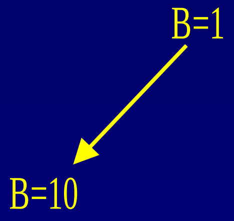
NOR vs. *B* and *T*



increase T

LF vs. *B* and *T*


LF
100
80
60
40
20
0
0.5
0.75
0.86
1
1.2
1.5
2.0
3
10
7
4
1
NR / T


B=1
B=10

increase T

T, B, LF, NOR

	Bucket size								
0.5	0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	2.8
0.75	0	0.0	0.0	0.0	0.0	0.0	0.2	1.1	6.5
0.86	0	0.0	0.0	0.0	0.0	0.0	0.3	1.6	8.1
1	0	0.0	0.0	0.0	0.0	0.1	0.4	2.3	10.3
1.2	0	0.0	0.0	0.0	0.0	0.1	0.8	3.6	13.7
1.5	0	0.0	0.0	0.0	0.1	0.3	1.6	5.9	18.6
2.0	0	0.0	0.0	0.0	0.3	1.1	3.7	10.8	27.0
3	0	0.0	0.1	0.6	1.7	4.5	10.6	22.4	

Overflow Records (%)

	Bucket size									
0.5	5.0	5.6	6.3	7.1	8.3	10.0	12.5	16.7	25.0	50.0
0.75	7.5	8.3	9.4	10.7	12.5	15.0	18.8	25.0	37.5	75.0
0.86	8.6	9.5	10.7	12.2	14.3	17.1	21.4	28.6	42.9	85.7
1	10.0	11.1	12.5	14.3	16.7	20.0	25.0	33.3	50.0	100.0
1.2	12.0	13.3	15.0	17.1	20.0	24.0	30.0	40.0	60.0	
1.5	15.0	16.7	18.8	21.4	25.0	30.0	37.5	50.0	75.0	
2.0	20.0	22.2	25.0	28.6	33.3	40.0	50.0	66.7	100.0	
3	30.0	33.3	37.5	42.9	50.0	60.0	75.0	100.0		

Load Factor (%)