Hash-Based Indexes

(From Chapter 11)

Introduction

- As for any index, 3 alternatives for data entries $k^*$:
  - Hash-based indexes are best for equality selections.
  - Static and dynamic hashing techniques exist

Static Hashing

- # primary pages fixed, allocated sequentially, never de-allocated; overflow pages if needed.
- $h(k) \mod N$ = bucket to which data entry with key $k$ belongs. ($N = \#$ of buckets)

- Primary bucket pages
- Overflow pages
Static Hashing (Contd.)

- Buckets contain data entries.
- Hash fn works on search key field(s) of record \( r \).
  - Must distribute values over range 0 \(...\) N-1.
    - \( h(\text{key}) = \)
- Long overflow chains can develop and degrade performance

Extendible Hashing

- Main idea: If bucket (primary page) becomes full, why not re-organize file by doubling \# of buckets?
- But reading and writing all buckets is expensive!
  - Idea:

Insert \( h(r)=14 \)

![Diagram of Extendible Hashing]

Database Management Systems, R. Ramakrishnan and Johannes Gehrke
Insert \( h(r) = 20 \)

Database Management Systems, R. Ramakrishnan and Johannes Gehrke

Insert \( h(r) = 32 \)

Database Management Systems, R. Ramakrishnan and Johannes Gehrke
**Insert $h(r)=16$**

```

```

**Insert $h(r)=20$**

```

```

**Insert $h(r)=5, 15, 7, 19$**

```

```

```
Deletions

- Inverse of insertion
- If removal of data entry makes bucket empty, merge with 'split image'
- If each directory element points to same bucket as its split image, can halve directory

Comments on Extendible Hashing

- If directory fits in memory, equality search answered with ______ I/O; else ______
- 100MB file, 100 bytes/rec, 4K pages contain 1,000,000 records (as data entries) and 25,000 directory elements; chances are high that directory will fit in memory.
- Directory grows in spurts, and, if the distribution of hash values is _________, directory can grow large

Linear Hashing

- This is another dynamic hashing scheme, an alternative to Extendible Hashing
- LH handles the problem of long overflow chains without using a directory, and handles duplicates
- Main idea:
Inserting $h(r) = 43$

Example (Inserting $h(r) = 43$)

Inserting $h(r) = 50$ (End of a Round)
Overview of LH File

- In the middle of a round.

Summary

- Hash-based indexes: best for _______ searches, cannot support _______ searches.
- Static Hashing can lead to ________________.
- Extendible Hashing uses directory doubling to avoid ________________.
  - Duplicates may require ________________.
- Linear hashing avoids directory by splitting in rounds
  - Naturally handles ________________
  - Uses overflow buckets (but not very long in practice)