Multi-Disk B-trees

G22: Rami and Yiqun

Related to: Indexes, Distributed Databases, Disk Arrays
Quick Review on B\(^{+}\)-tree

**B\(^{+}\)-tree properties**
1. search tree;
2. balanced tree;
3. threaded leafs;
4. max. occurrence of a key: 2
5. all keys appear in leafs.

**B\(^ {+}\)-tree operations**
1. search: \(\log_b N\)
2. insert: \(\log_b N\)
3. range query: \(\log_b N + \text{Len\_of\_Result}\)

Insert 1,2,3,4,5,6
Distribution Strategies

- Record Distribution
- Large Page Distribution
- Page Distribution
Record Distribution

Each disk is a node with a local B-tree.

Record distribution based on:
1. key-range partitioning;
2. hash partitioning;
3. round robin assignment.

Performance criteria:
1. balancing;
2. point query;
3. range query (small vs. large).
Large Page Distribution

Every node is a virtual super page which composed of sub-pages distributed to p disks.

Pros: each point query only involves one sub-page.
Cons: each range query requires at least one super page to be loaded.
Page Distribution

- Pages are distributed among disks.
- Assigning pages to disks is done either randomly or in order upon splitting.
- Pros: good for point queries.
- Cons: local load balancing is not guaranteed for range queries.
New Trends

- **ScalaGiST**
  - Similar to Record Distribution.
  - A local b-tree on every node.
  - Master index is composed of the roots of local b-trees.
  - Query can run in parallel.
References

