Designing an Index for ZooDB

Jonas Nick & Bogdan Vancea

May 28, 2014

Outline

ZooDB

- an open source object database in Java
- JDO standard
- 4 times faster than competitor db4o
- `github.com/tzaeschke/zoodb`

Index

datastructure that allows rapid lookup

- where indices are used in zoodb

B+ Tree

- node fills one disk page
- inner node contains keys and children pointer, leaves contain keys and values
- key unique vs. key-value unique

Example: insert

3908 loc + 2665 loc of test

Goals

- faster
- buffer manager
- prefix sharing

Challenges

- edge cases
- runtime dominated by disk access
  - change nodes infrequently
  - fewer nodes is better
- only parent to child pointer
- determine best when its time to split/redistribute
- key unique vs. key-value unique
- buffer manager lookup takes time
- prefix-sharing encoding/decoding takes time
- prefix-sharing rebalancing takes time
- general optimizations
  - avoid polymorphism
  - bit-level operations

Our B+ Tree

Buffer Manager

Prefix Sharing

Optimizations

Fine grained

- insert, remove, write
- duration, number of nodes
- prefix-sharing vs. no prefix-sharing

Whole system

- test harness
- PolePosition benchmark
- StackOverflow