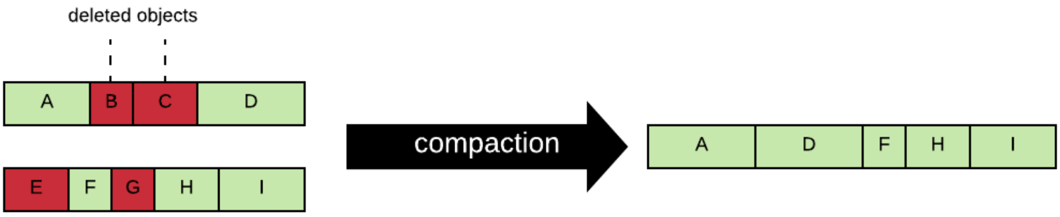
Garbage Collection in a storage system with billions of objects

Description

Writing to a hard disk is much faster if you write large sequential data in one part of the disk, instead of doing “random writes” i.e. writing small bits of data all over the disk. Therefore we group many small objects together in one large container, and write this large container in one big sequential write operation to the hard disk.

However this creates a problem when deleting the small objects – how can you reclaim the free space on-disk. This is a typical garbage collection problem. Once in a while you want to rewrite containers by joining containers together and removing the deleted pieces such that you keep nice big containers on disk but you reclaim free space regularly. This process is called compaction:



Technology

- Inventing scalable algorithms, heuristics, or machine learning
- Developing in a language of choice (Java, C++, Python)
- Produce a demo or a simulator

Goals

- Identify millions of deleted objects in a global system of billions (or trillions) of objects. How to find out “low hanging fruit” i.e. where can we reclaim most capacity with the least effort. How to predict what is going to be deleted.
- How to do incremental compaction rather than a full scan of all deleted objects.
- How to be smart & proactive by grouping objects together that probably have the same age and lifecycle policy i.e. they probably will be deleted at the same time, or will never be deleted.
- If multiple containers have free space to reclaim, how to select which containers you group together as to have a long term sustainable set of large containers with not too much garbage.