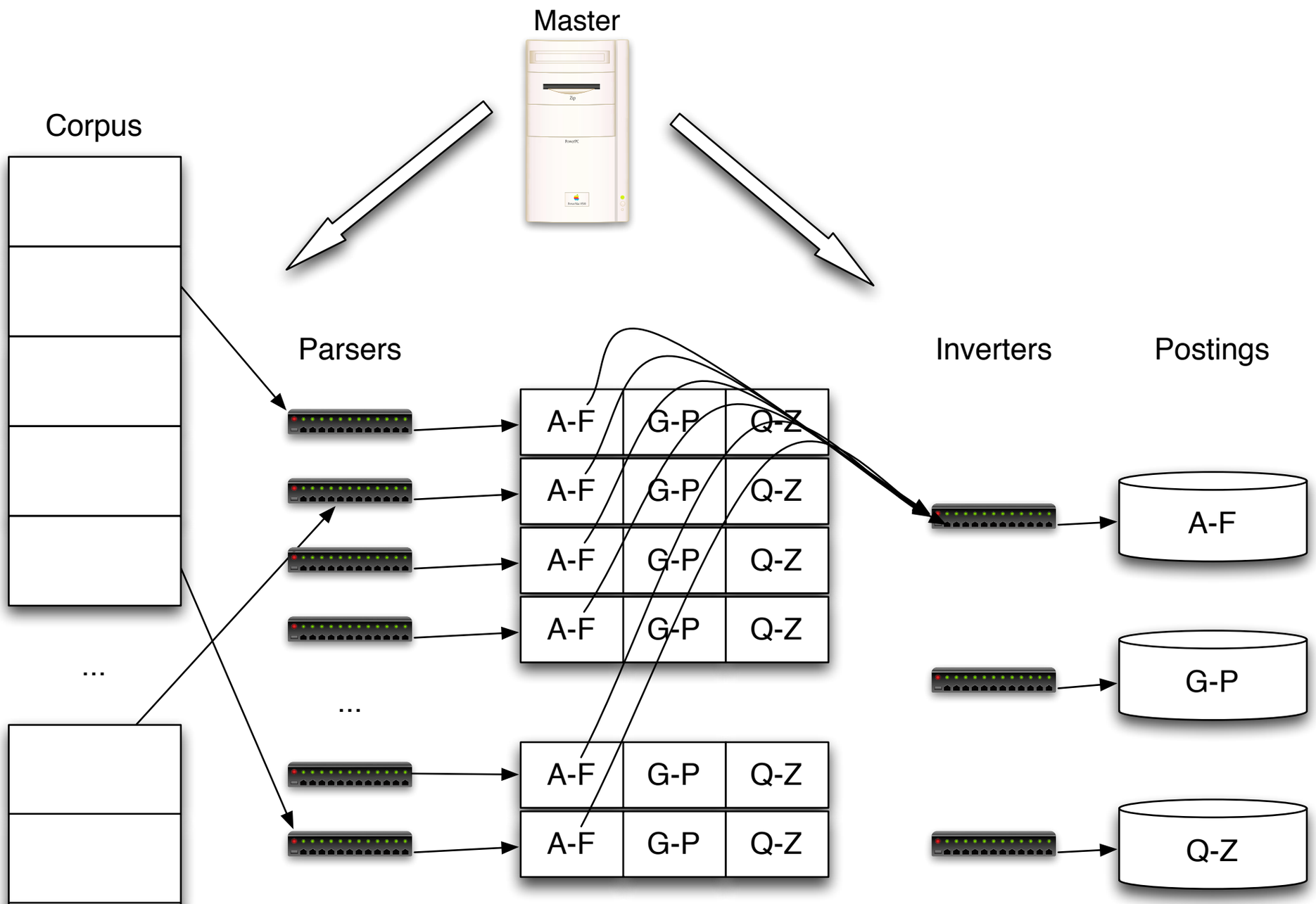
Index Construction

Introduction to Information Retrieval
INF 141
Donald J. Patterson

Content adapted from Hinrich Schütze
http://www.informationretrieval.org

Distributed Indexing - Architecture



Distributed Indexing - Architecture

- Parsers and Inverters are not separate machines
  - They are both assigned from a pool
  - It is separate software
- Intermediate files are stored on a local disk
  - Part of the “merge” task is to talk to the parser machine and get the data. (master coordinates)
- MapReduce has different architectures for different data manipulation tasks besides this one.

Distributed Indexing - Architecture

- MapReduce/Hadoop in particular (hadoop.apache.org/core):
  - Scalable: Hadoop can reliably store and process petabytes.
  - Economical: It distributes the data and processing across clusters of commonly available computers. These clusters can number into the thousands of nodes.
  - Efficient: By distributing the data, Hadoop can process it in parallel on the nodes where the data is located. This makes it extremely rapid.
  - Reliable: Hadoop automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.

Distributed Indexing - Architecture

- Basic steps for running MapReduce on Hadoop:
  - Get some computers
  - Install Hadoop on all the computers
  - Identify two masters
    - NameNode
      - Manages the distributed file system
    - JobTracker
      - Manages the MapReduce work

Distributed Indexing - Architecture

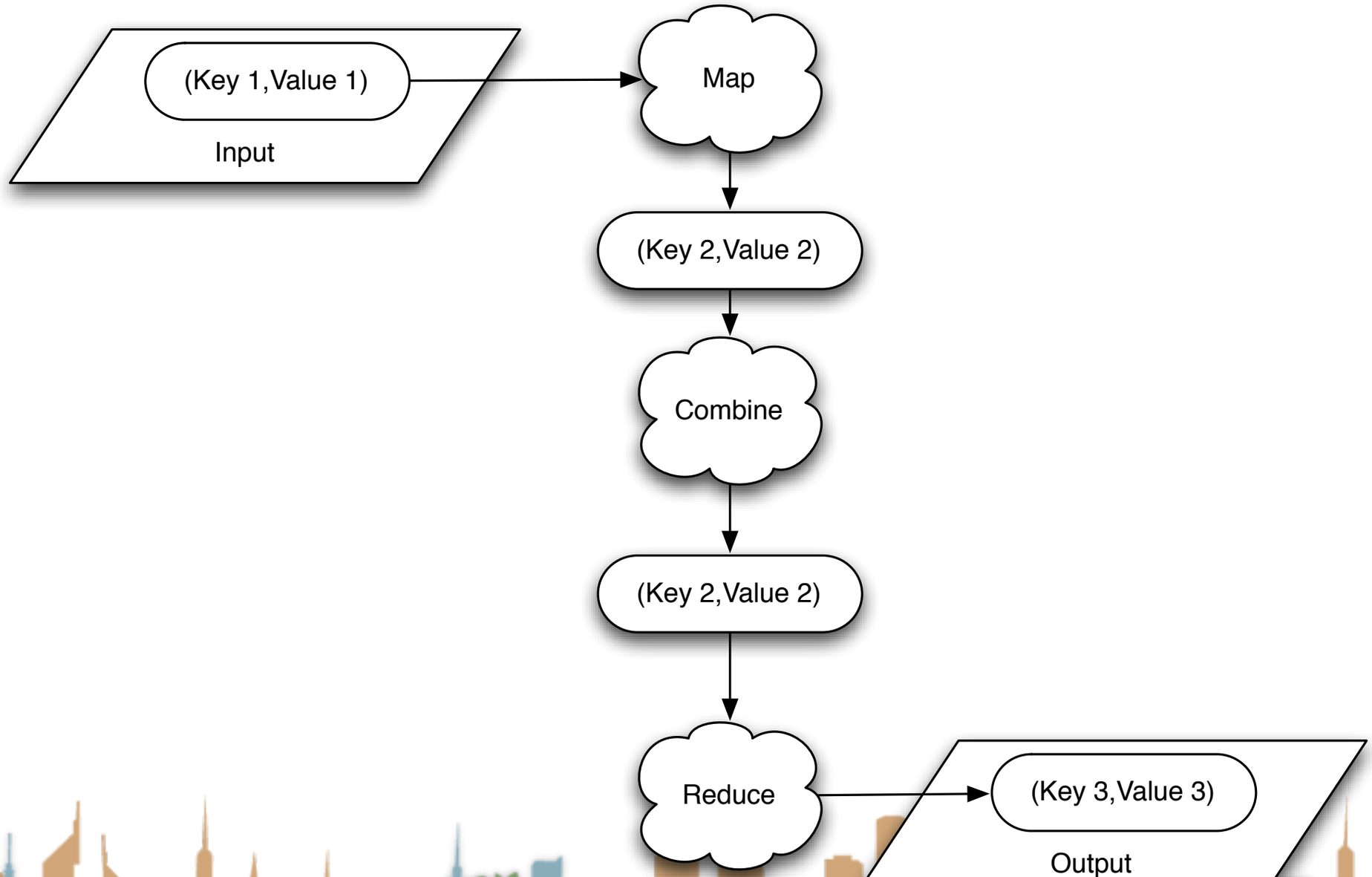
- Basic steps for running MapReduce on Hadoop:
  - Identify slaves, which act as both
    - DataNodes
      - A piece of the distributed file system
    - TaskTrackers
      - A resource for running Map and Reduce jobs
  - The JobTracker tells the TaskTrackers what to do

Distributed Indexing - Architecture

- Basic steps for running MapReduce on Hadoop:
  - Job Configuation (aka, Define your work):
    - Where and what is your input data?
    - Where and what is your output data?
    - Write a Map function:
      - Takes chunks of input data and processes it in parallel
  - Write a Reduce function:
    - Takes output of Map function and aggregates it

Distributed Indexing - Architecture

- Doing Job Configuration
  - Remember this works on <key,value> pairs



Distributed Indexing - Architecture

- Doing Job Configuration
  - WordCount

```
public class WordCount {
  public static class Map{
     public void map(Type1 key, Type2 value,OutputCollector
    output){
        process key, value;
        output.write(newKey, newValue)
  }

  public static class Reduce{
   public void reduce(Type3 key, Iterator<Type4>
  values,OutputCollector output)
        process key and values;
        output(newKey, newValue);
  }
}
```

  - http://hadoop.apache.org/core/docs/current/mapred_tutorial.html

Overview

- Introduction
- Hardware
- BSBI - Block sort-based indexing
- SPIMI - Single Pass in-memory indexing
- Distributed indexing
- Dynamic indexing
- Miscellaneous topics

Dynamic Indexing

- Documents come in over time
  - Postings need to be updated for terms already in dictionary
  - New terms need to get added to dictionary
- Documents go away
  - Get deleted, etc.

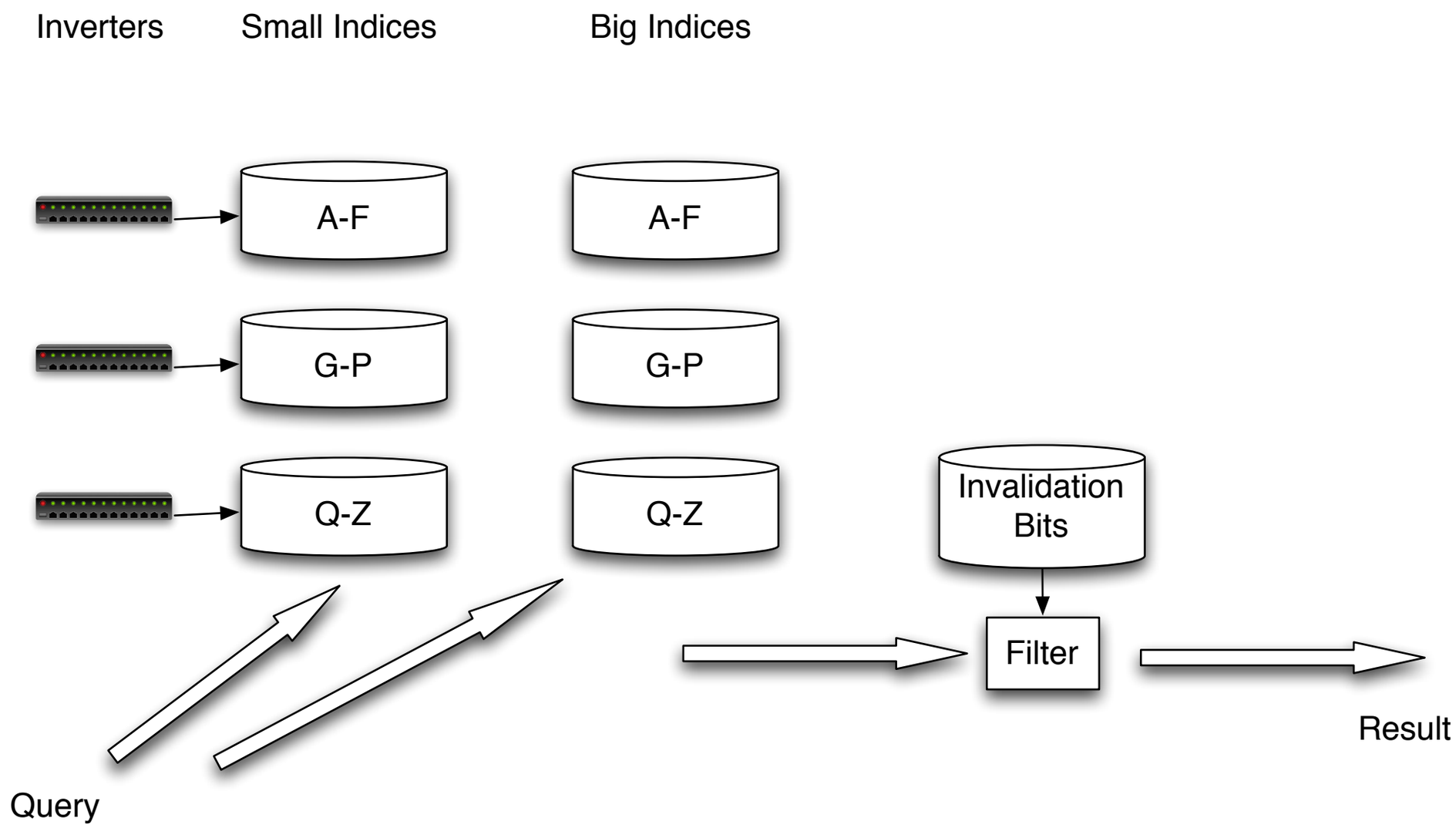
Dynamic Indexing

- Overview of solution
  - Maintain your “big” main index on disk
    - (or distributed disk)
  - Continuous crawling creates “small” indices in memory
  - Search queries are applied to both
  - Results merged

Dynamic Indexing

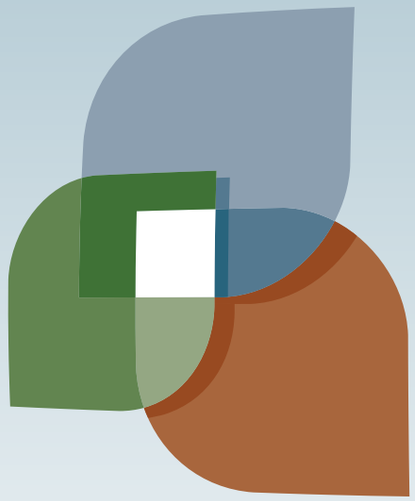
- Overview of solution
  - Document deletions
    - Invalidation bit for deleted documents
    - Just like contextual filtering,
      - results are filtered to remove invalidated docs
        - according to bit vector.
- Periodically merge “small” index into “big” index.

Dynamic Indexing



Dynamic Indexing

- Issues with big *and* small indexes
  - Corpus wide statistics are hard to maintain
    - Typical solution is to ignore small indices when computing stats
  - Frequent merges required
- Poor performance during merge
  - unless well engineered
- Logarithmic merging



LUCI