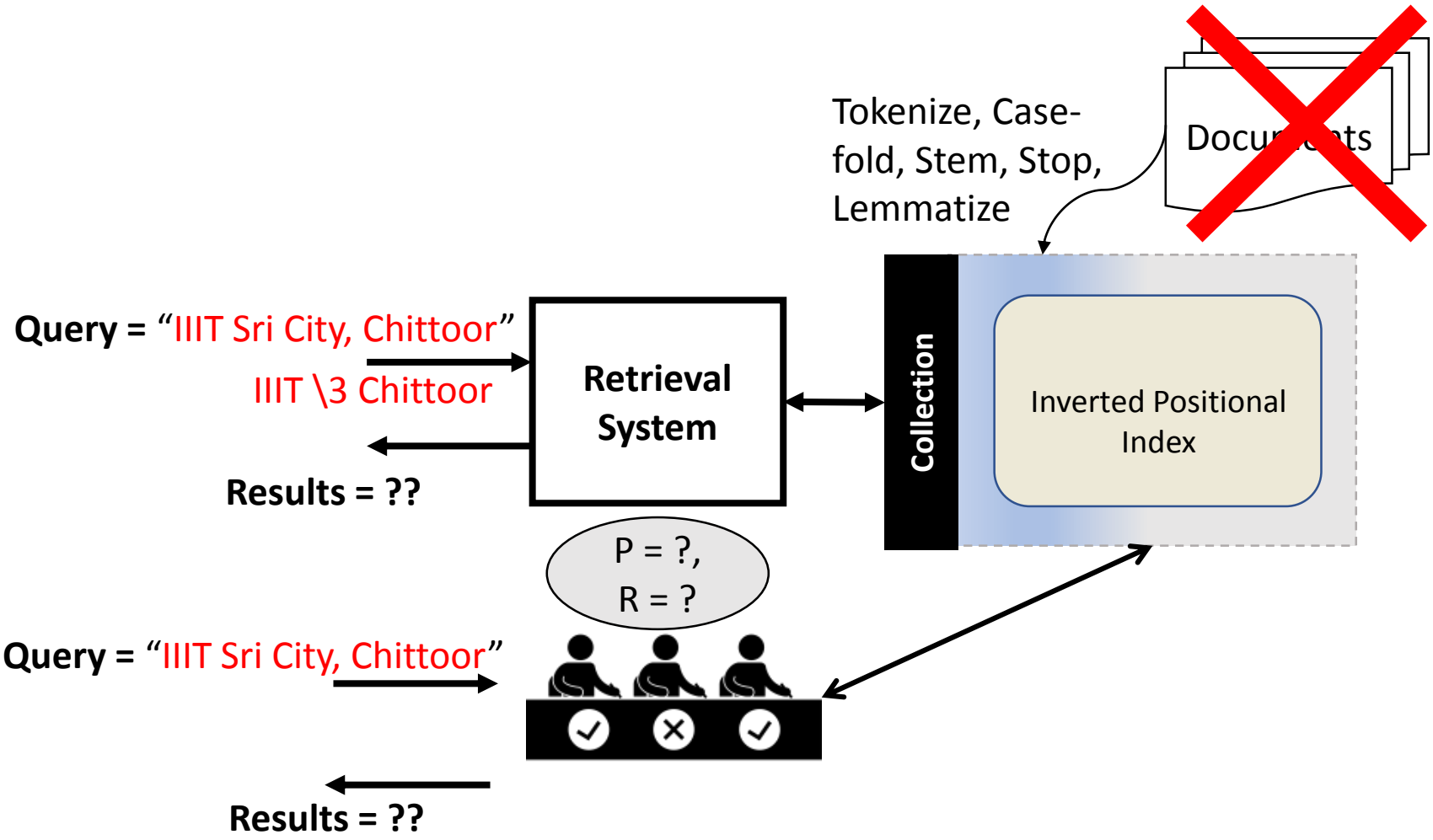
Dynamic Indexing

When the collection gets updated, how to update our index?



How to deal with changes in the collection?

- Periodically re-index from scratch!

Can we do better?

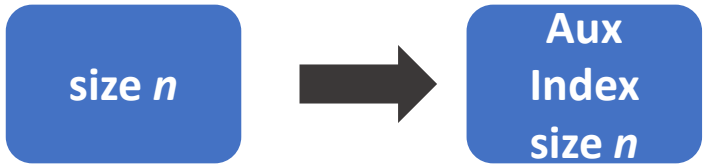
What if new documents must be indexed quickly?

Auxiliary Index

- Keep two indexes
  - Main Index
    - Keep invalidation bit to indicate deletion.
  - Auxiliary Index (for new files)
    - Periodically (or when Aux. Index becomes too large) merge auxiliary index with main index.
- Search in both. Merge the results.

Can we do better?

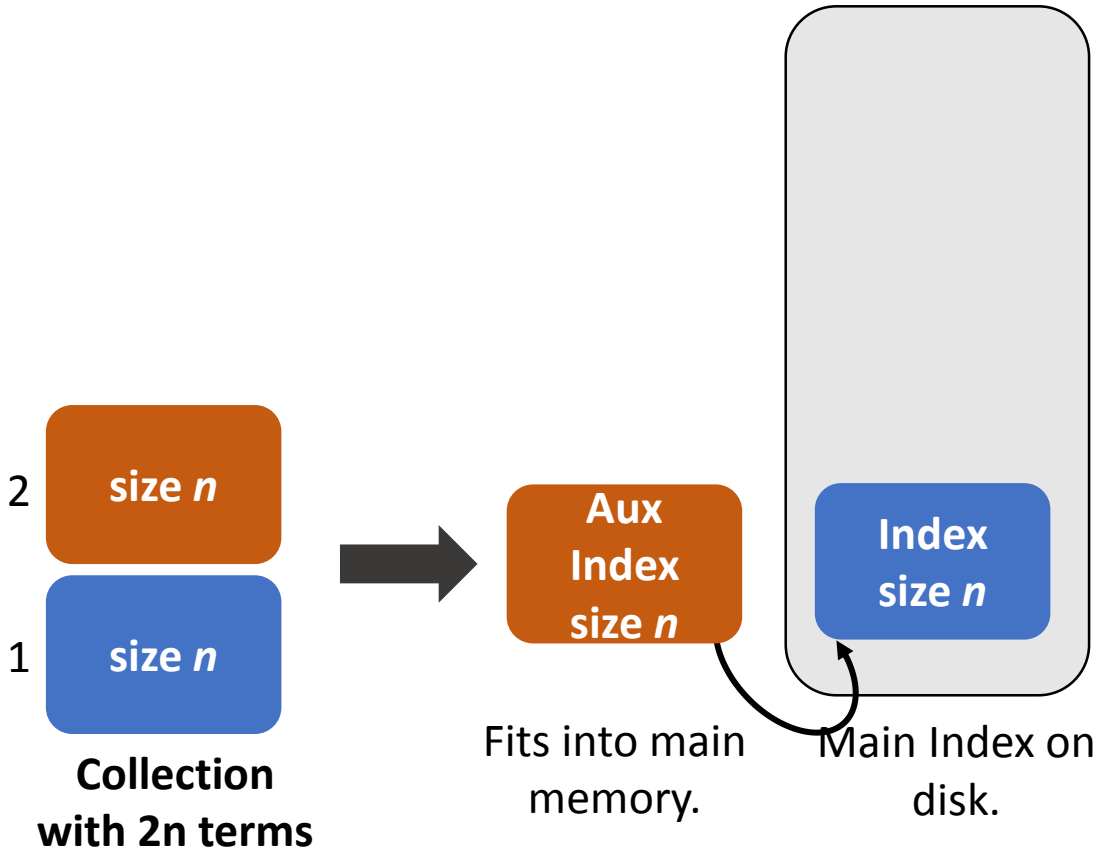
Main Index and Auxiliary Index



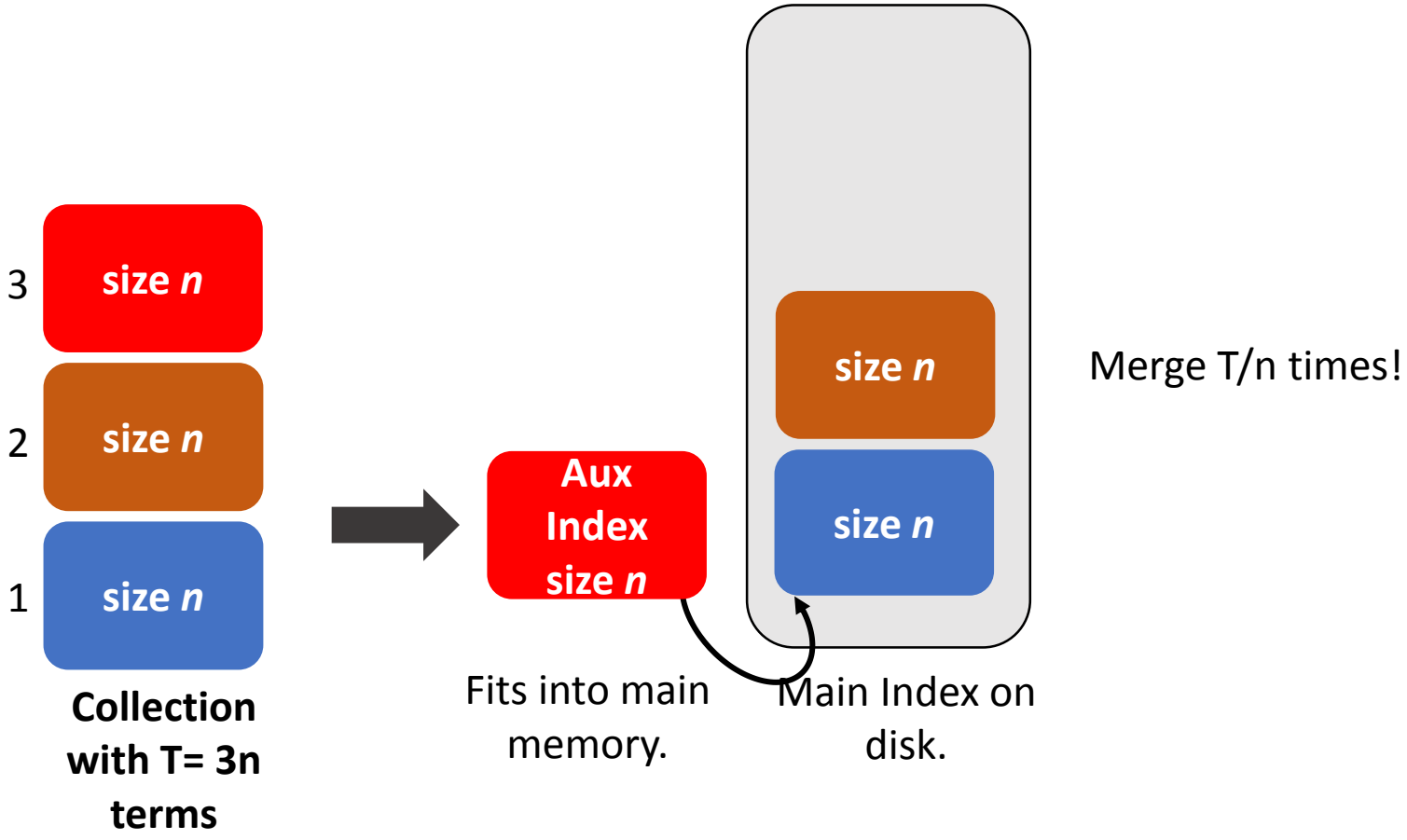
Collection with n terms

Start with a small index that fits into main memory.

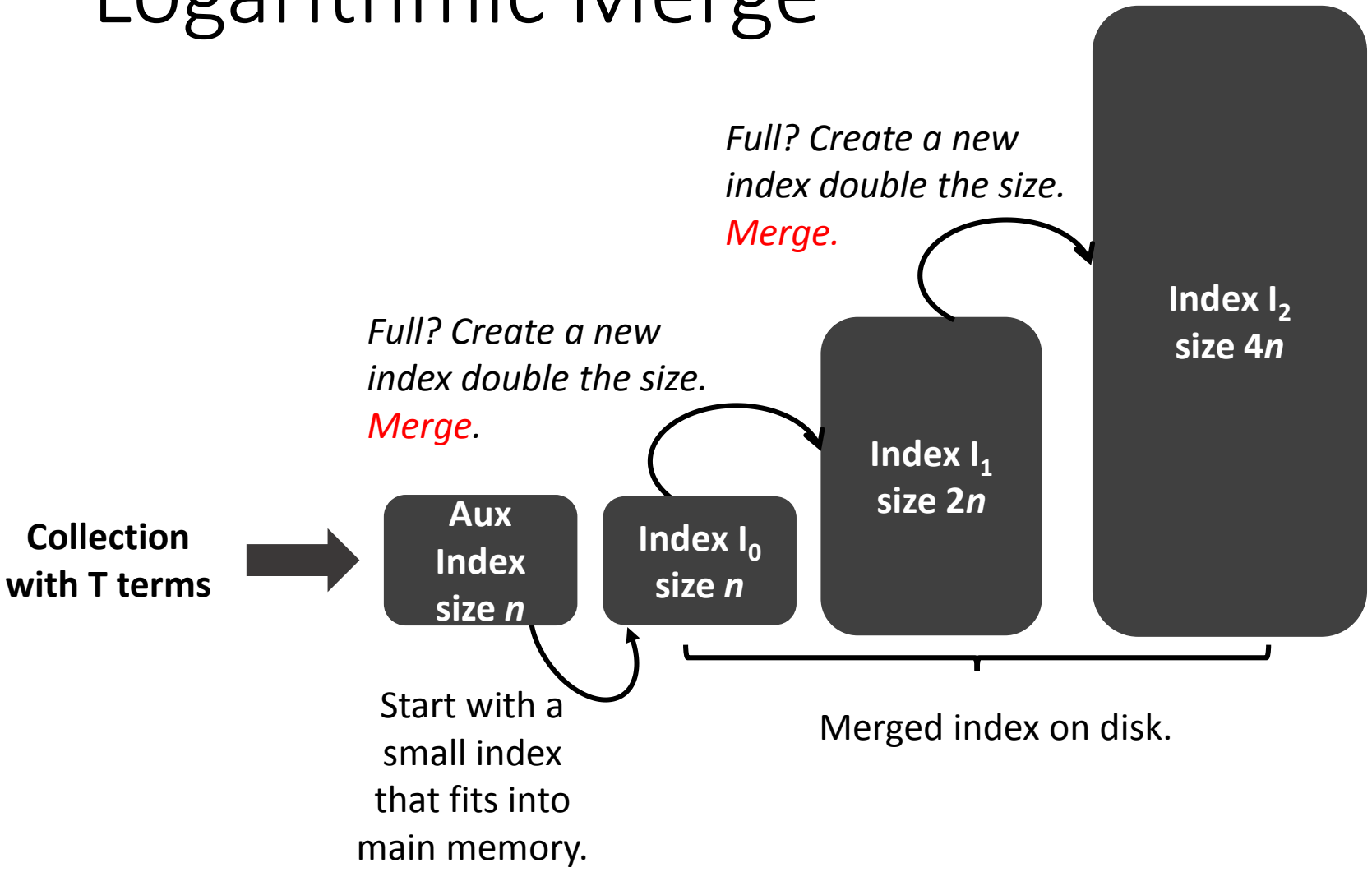
Merge


2
size n
1
size n
Collection with 2n terms
Aux Index size n
Index size n
Fits into main memory.
Main Index on disk.

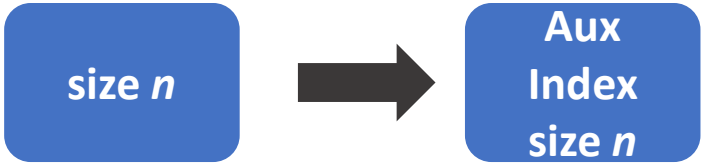
Merge


3
size n
2
size n
1
size n
Collection with T= 3n terms
Aux Index size n
Fits into main memory.
size n
size n
Main Index on disk.
Merge T/n times!

Logarithmic Merge



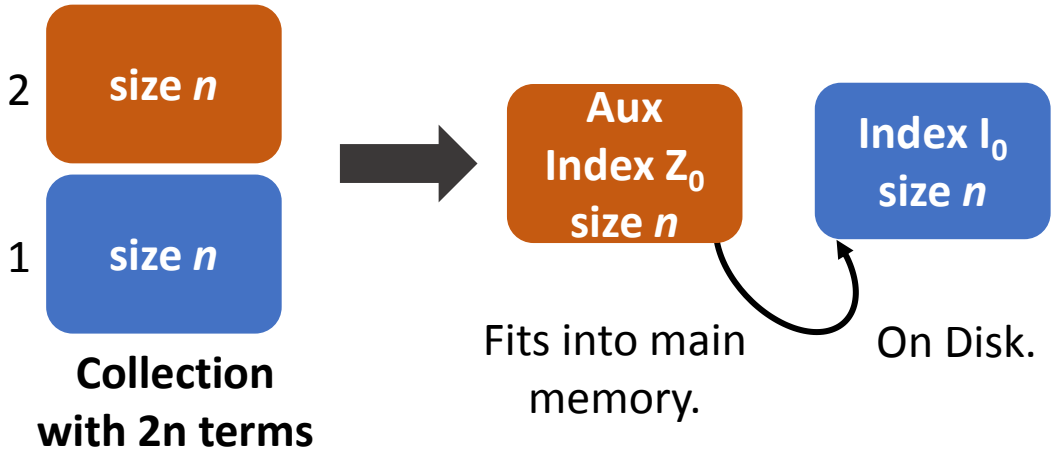
Logarithmic Merge



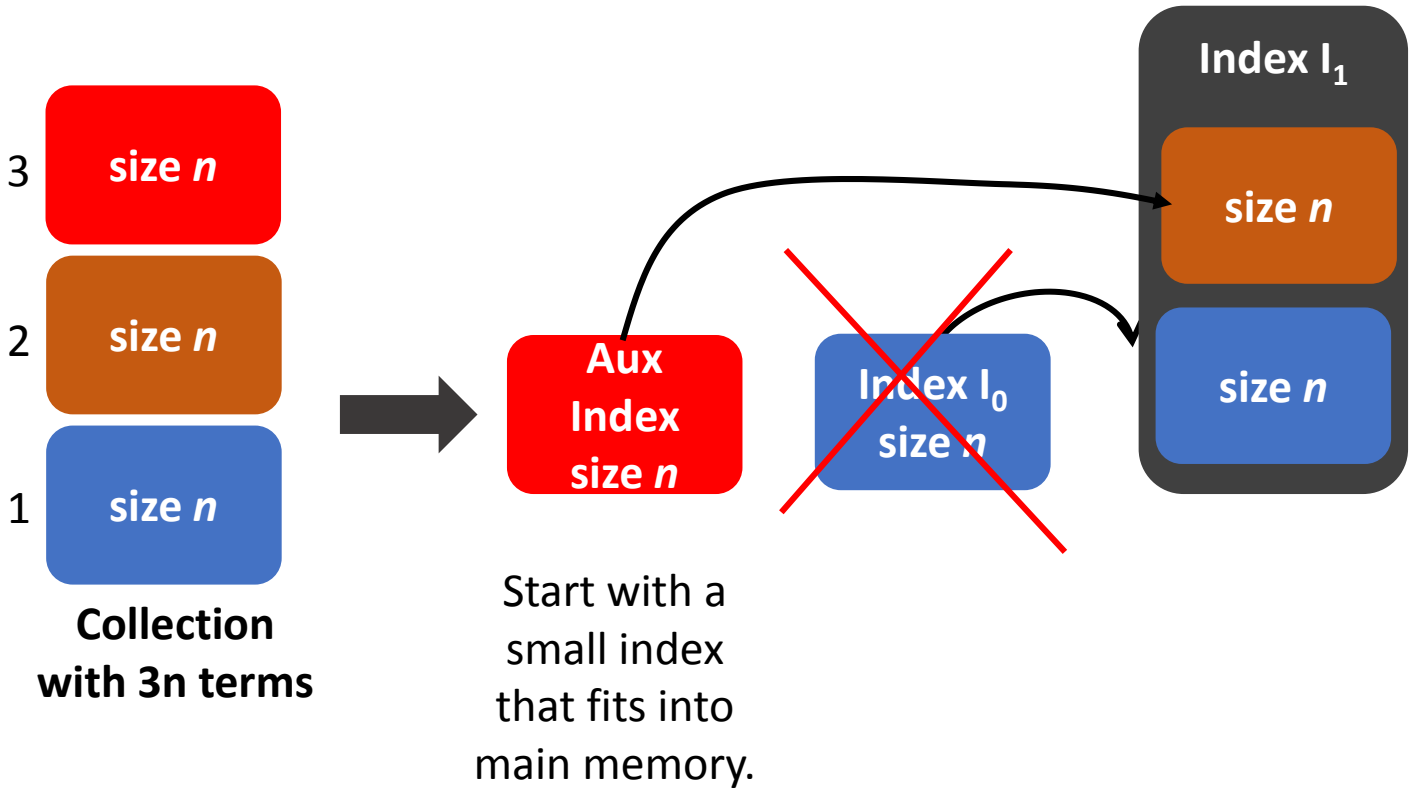
Collection with n terms

Start with a small index that fits into main memory.

Logarithmic Merge



Logarithmic Merge



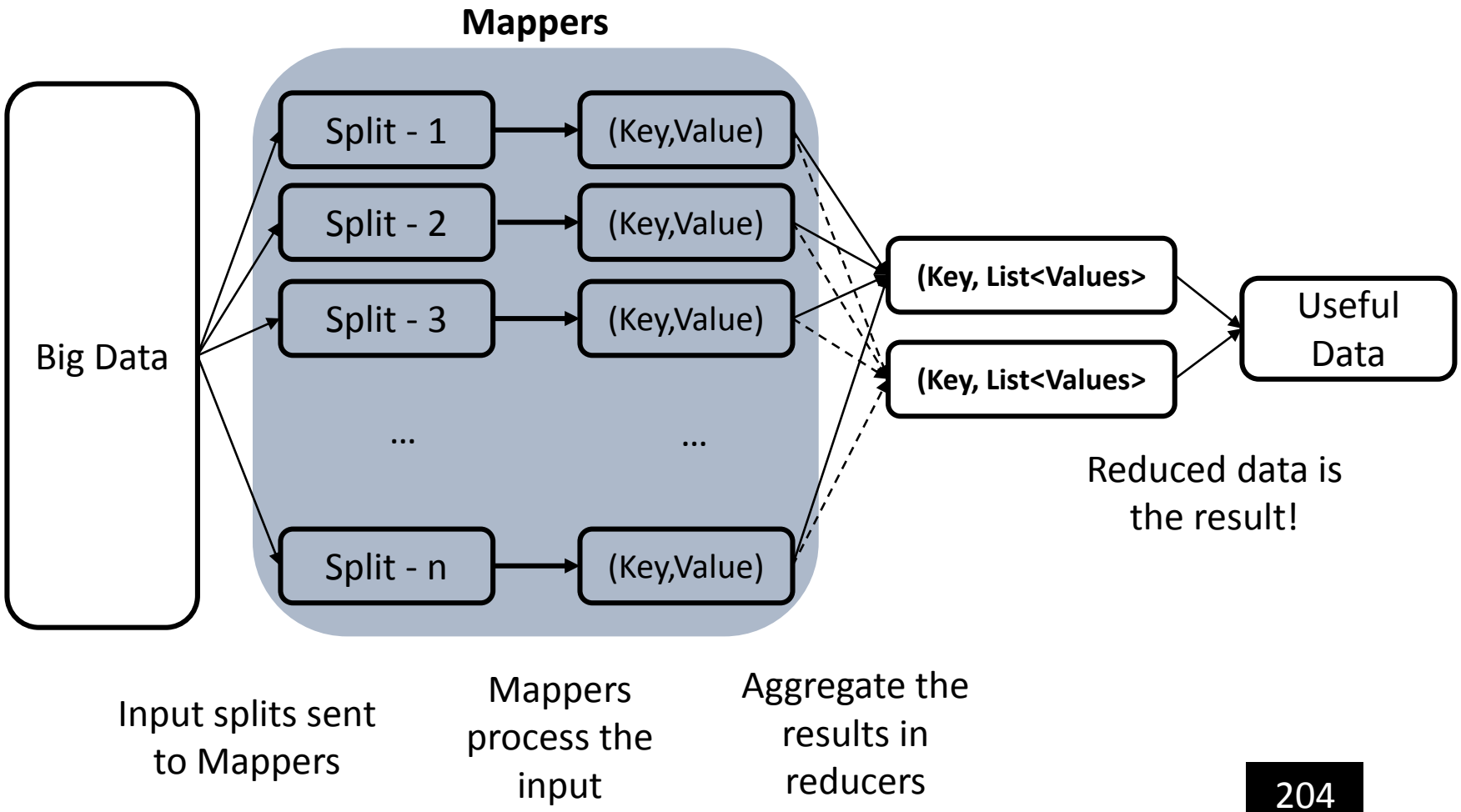
Quiz

- What is the overall index construction time? (choose the best answer)
  - $\theta(1)$
  - $\theta(T/n)$
  - $\theta(T^{2/n})$
  - $\theta(\log(T/n))$
  - $\theta(T \log(T/n))$

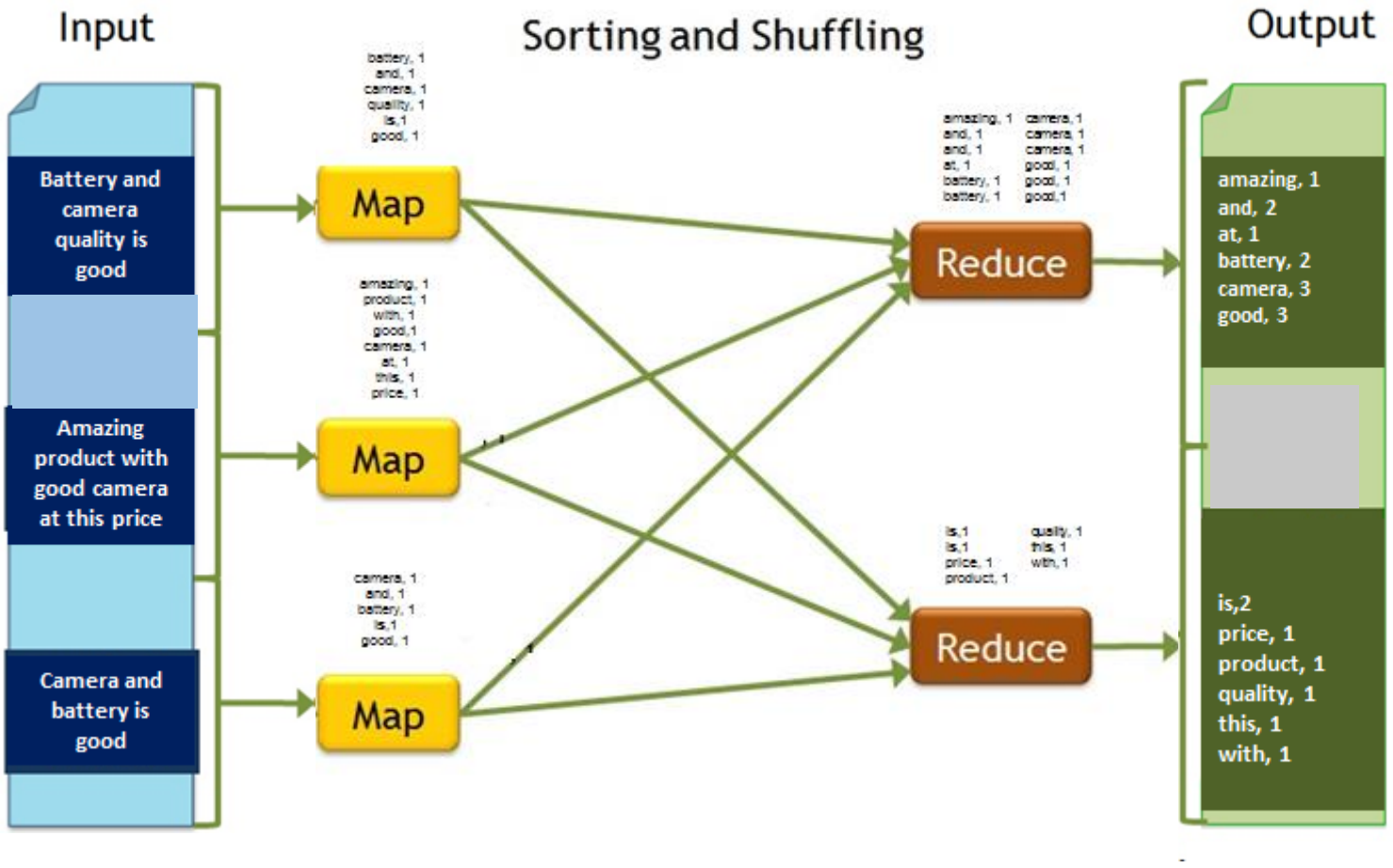
Distributed Indexing

How to leverage cloud computing infrastructure and distributed computing frameworks for indexing?

Indexing with the Map-Reduce Framework

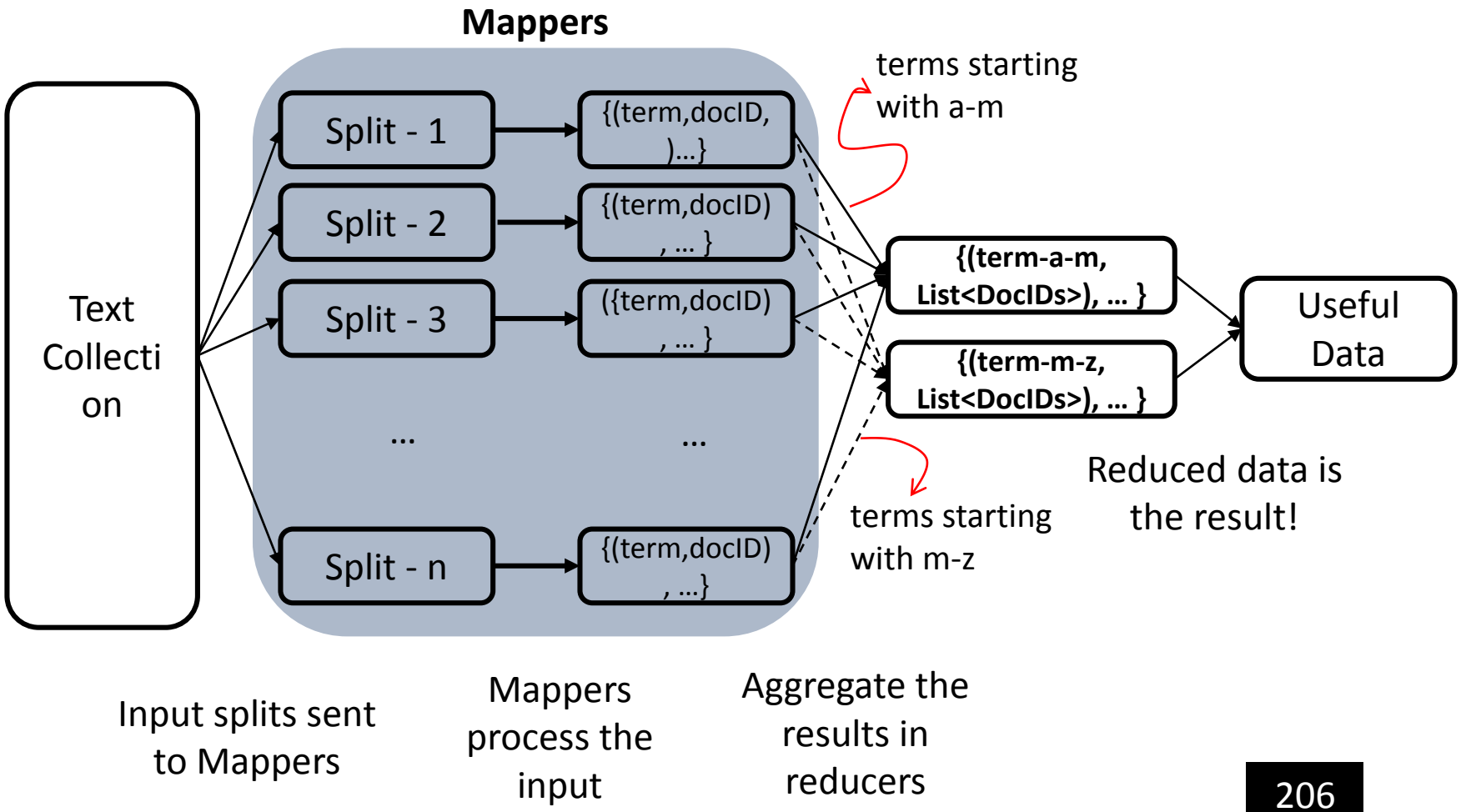


Map-Reduce Framework

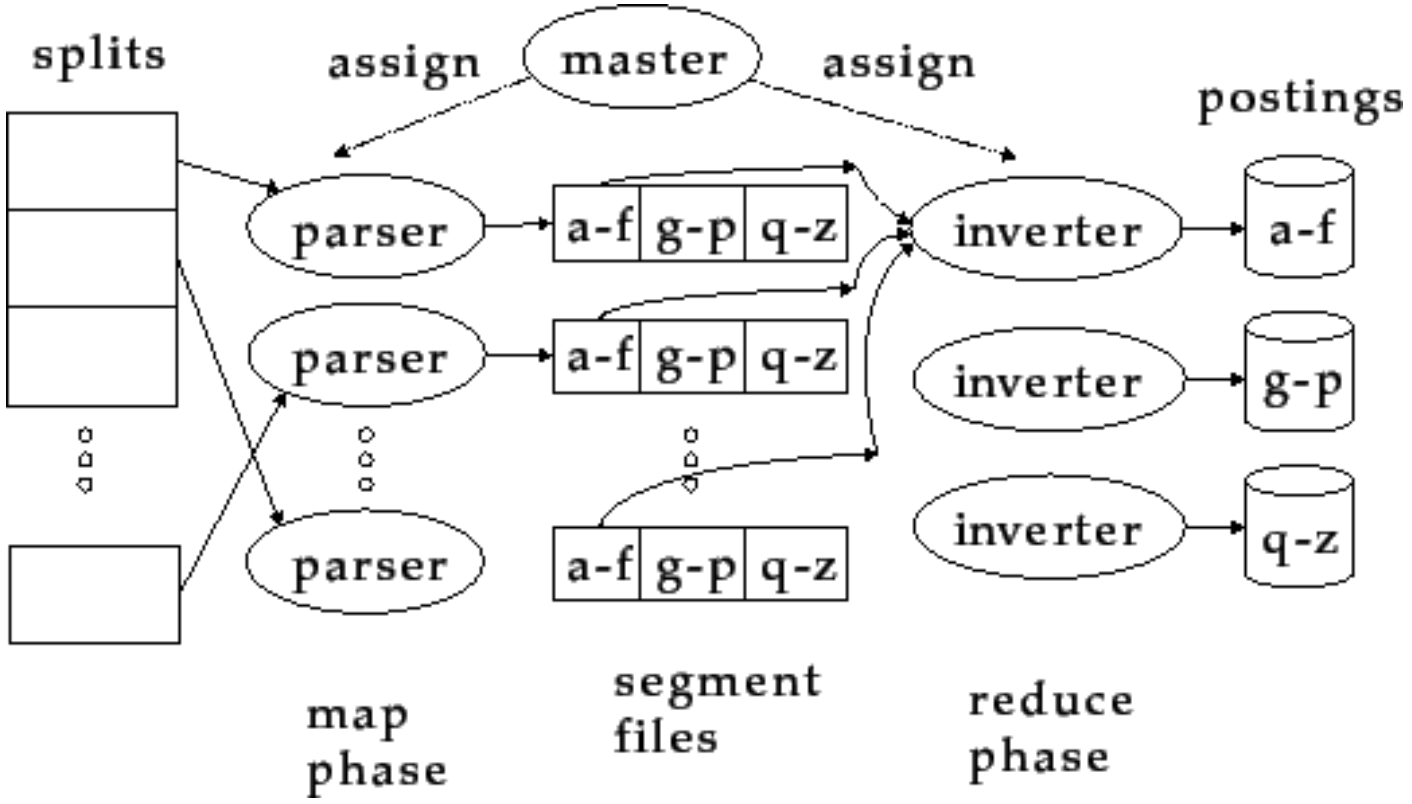


Picture Source: Internet.

Indexing with the Map-Reduce Framework



Indexing


splits
assign
master
assign
postings
parser
parser
parser
a-f g-p q-z
a-f g-p q-z
a-f g-p q-z
inverter
inverter
inverter
a-f
g-p
q-z
map phase
segment files
reduce phase

Summary

- We discussed
  - Blocked Sort-based Indexing
  - Single-pass In-memory Indexing
  - Distributed Indexing
  - Dynamic Indexing
- Advances in hardware, networking and software capabilities have led to wide variety of techniques for efficient indexing.