Distributed String Matching

Using the MapReduce Framework

Problem Statement

Given a string s

A large input such as:

```
MISSISSIPI
0123456789
```

Given a string t

A set of smaller input such as:

```
SS
```

Find t in s

The program should output:

```
2, 5
```

Procedure

Partition Large Input → Process each partition → Collate Results

Split the string

We split the string into overlapping segments.

Run Sequential Algorithm

We process each segment independently in parallel using KMP algorithm.

Count the number of occurrences

In each partition, if there is a match, we output the byte offset and the total number of occurrences using Reducer.

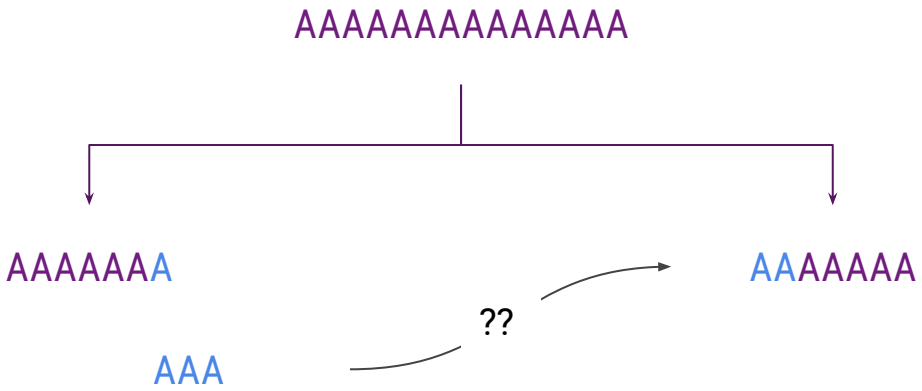
Partitioning the Input

~~Disjoint Partitions~~

~~Disjoint Partitions~~

~~Disjoint Partitions~~

~~Disjoint Partitions~~

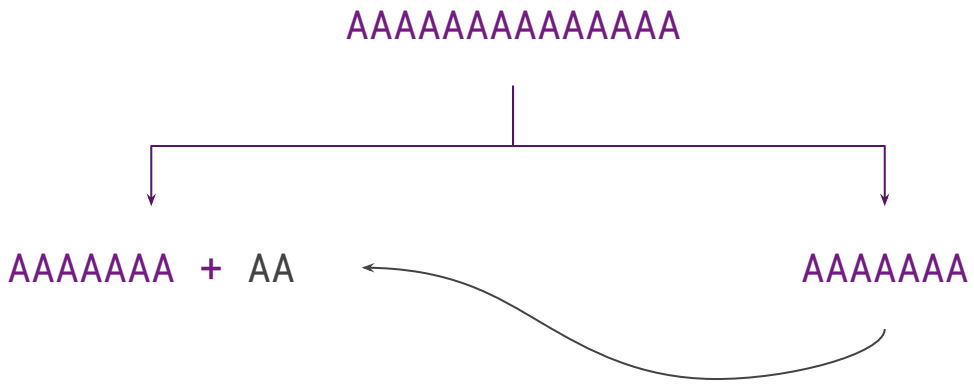


Solution

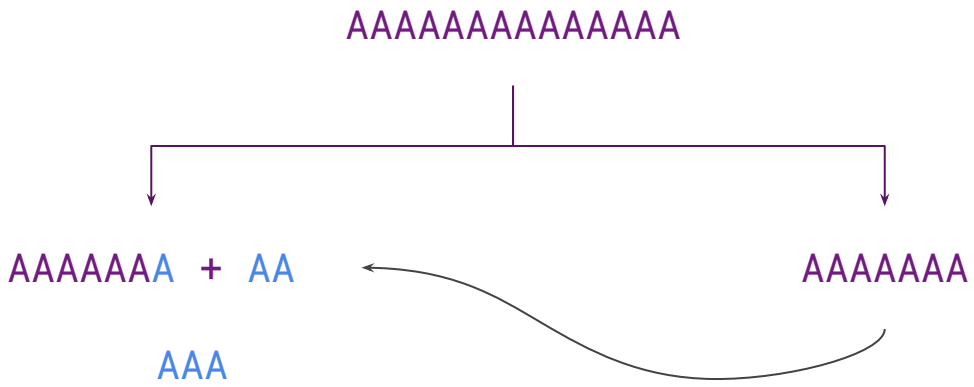
Use overlapping segments

In each segment, include |t|-1 characters of the next segment

Overlapping Segments



Overlapping Segments



Parallel Processing

KMP Algorithm

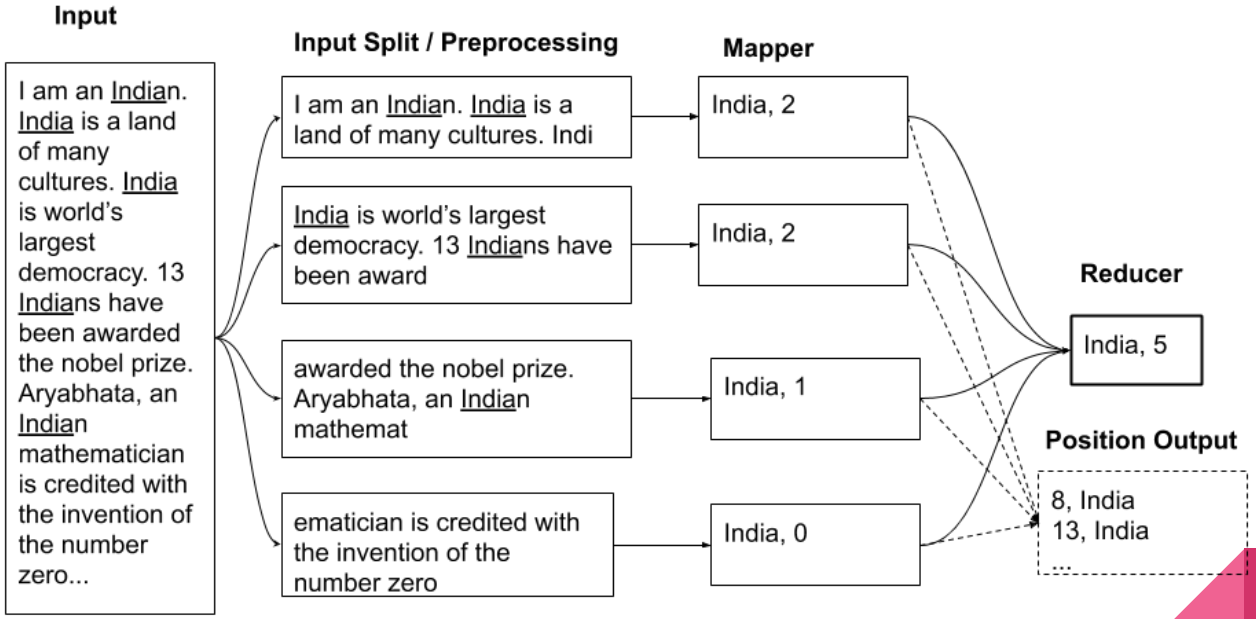
Finds a substring in a string in O(n)

Input: string s & t

Output: offset of t in s

Map-Reduce Implementation

Overview of Map Reduce Implementation



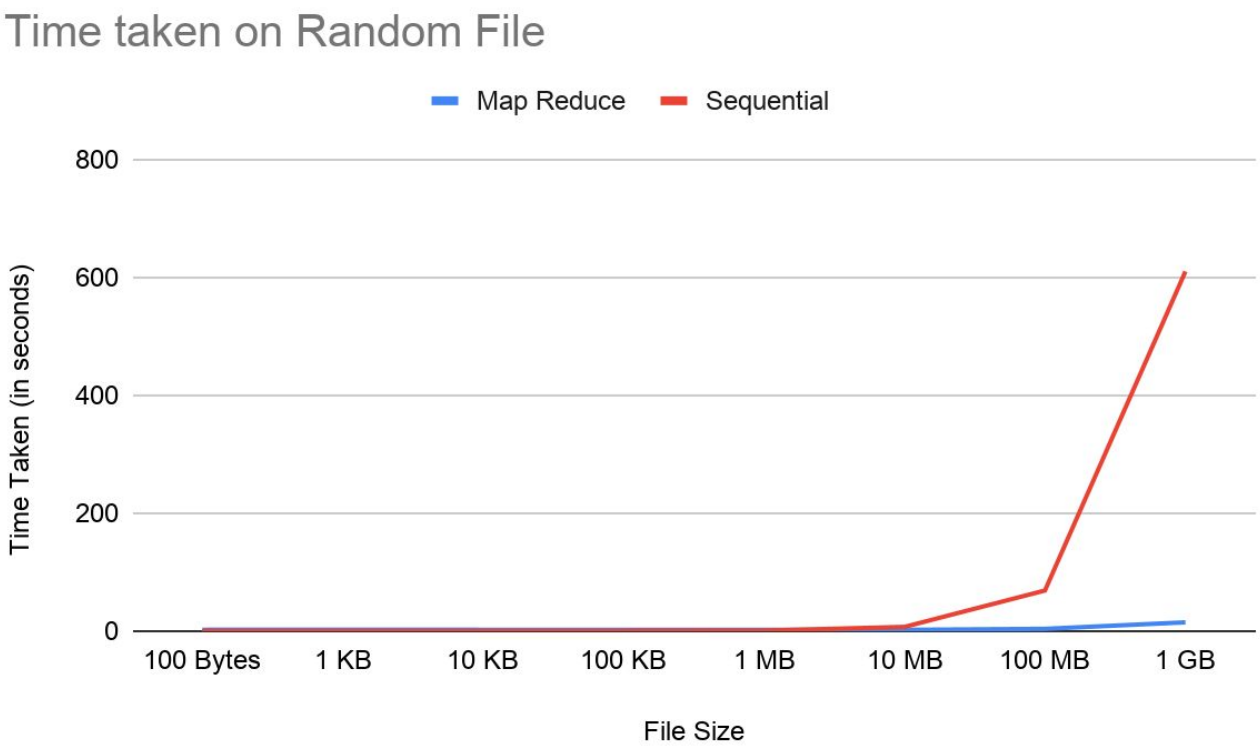
How to calculate positions?

Input to the Mapper function is the byte offset of the partition (s) and the contents of the partition.

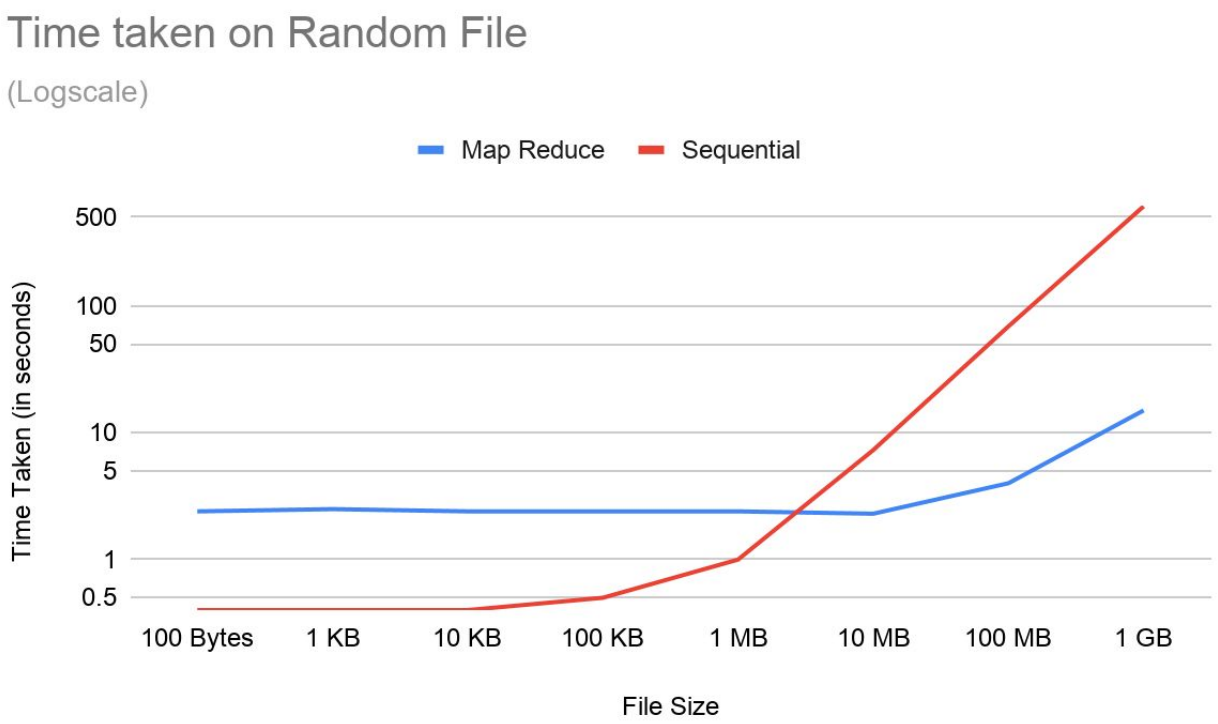
Let us say we find a string starting at o and p is the partition size.

Position = $$\left(\frac{s}{p}\right) \times (p - |t| + 1) + o$$

Performance Analysis



Performance Analysis on Random Text File



Performance Analysis on Random Text File

Human DNA

Finding TATA-boxes
86,46,037 Occurrences

2 minutes 36 seconds

Human DNA has ~3 Billion base pairs,
File Size: 3.2 GB

Wikipedia Abstracts

Finding “India”
3,64,964 Occurrences

2 minutes 24 seconds

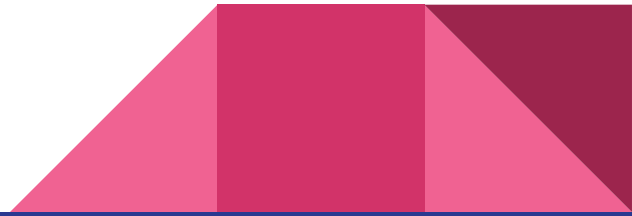
File Size: 6.2 GB

Conclusion

1. Map Reduce can be an effective tool for matching patterns in Big Data
2. Pattern matching can have many real-world applications

Future Work:

1. Extend to match RegEx instead of plain strings



Thank You