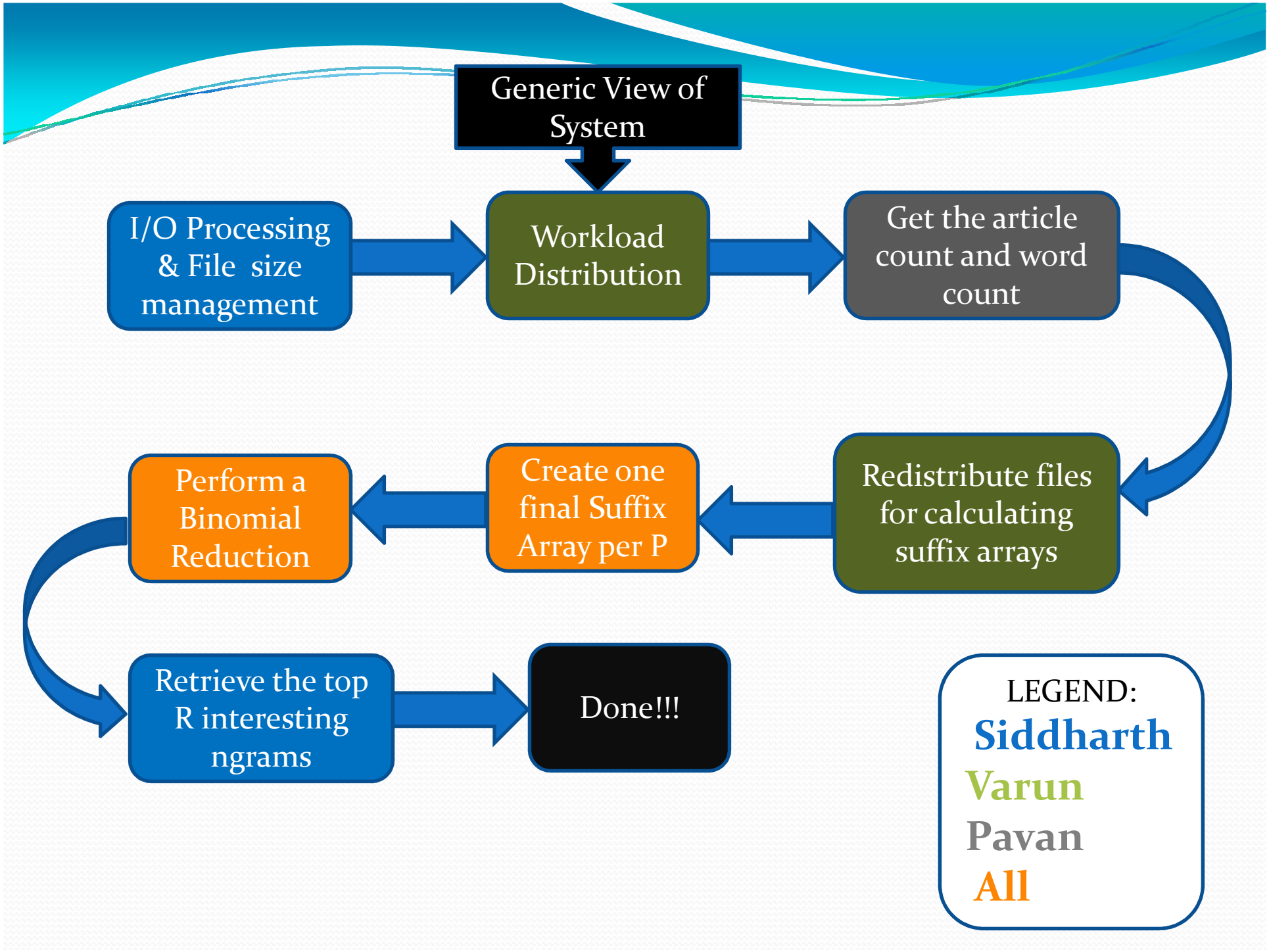
Team Tikal

Pavan Poluri
Siddharth Deokar
Varun Sudhakar


Generic View of System
I/O Processing & File size management
Workload Distribution
Get the article count and word count
Redistribute files for calculating suffix arrays
Create one final Suffix Array per P
Perform a Binomial Reduction
Retrieve the top R interesting ngrams
Done!!!
LEGEND:
Siddharth
Varun
Pavan
All

FOSTER'S DESIGN IN OUR PROJECT

- Partitioning: Domain Decomposition
- Communication : Broadcasting, Point to Point Communication and Customized Communication
- Agglomeration: Gathering of suffix arrays
- Mapping: Cyclic Mapping Strategy

Data Structure

- Customized suffix array1 to hold the following data
  - Position of ngram in the file
  - File index to identify the file
  - Term Frequency
  - Document Frequency
- Customized suffix array2 to hold the following data
  - Position of ngram in the file
  - File index to identify the file
  - Term Frequency
  - TF*IDF value

Algorithm

- I/O processing
  - Reading directory and storing file information
- File size Management
  - Partitioning files
  - Communication
- Workload Distribution
  - Interleaved Allocation

Contd……

- Alpha Requirement:
  - Calculating the number of words and articles
  - Reduction
    - MPI_Reduce()

Contd…

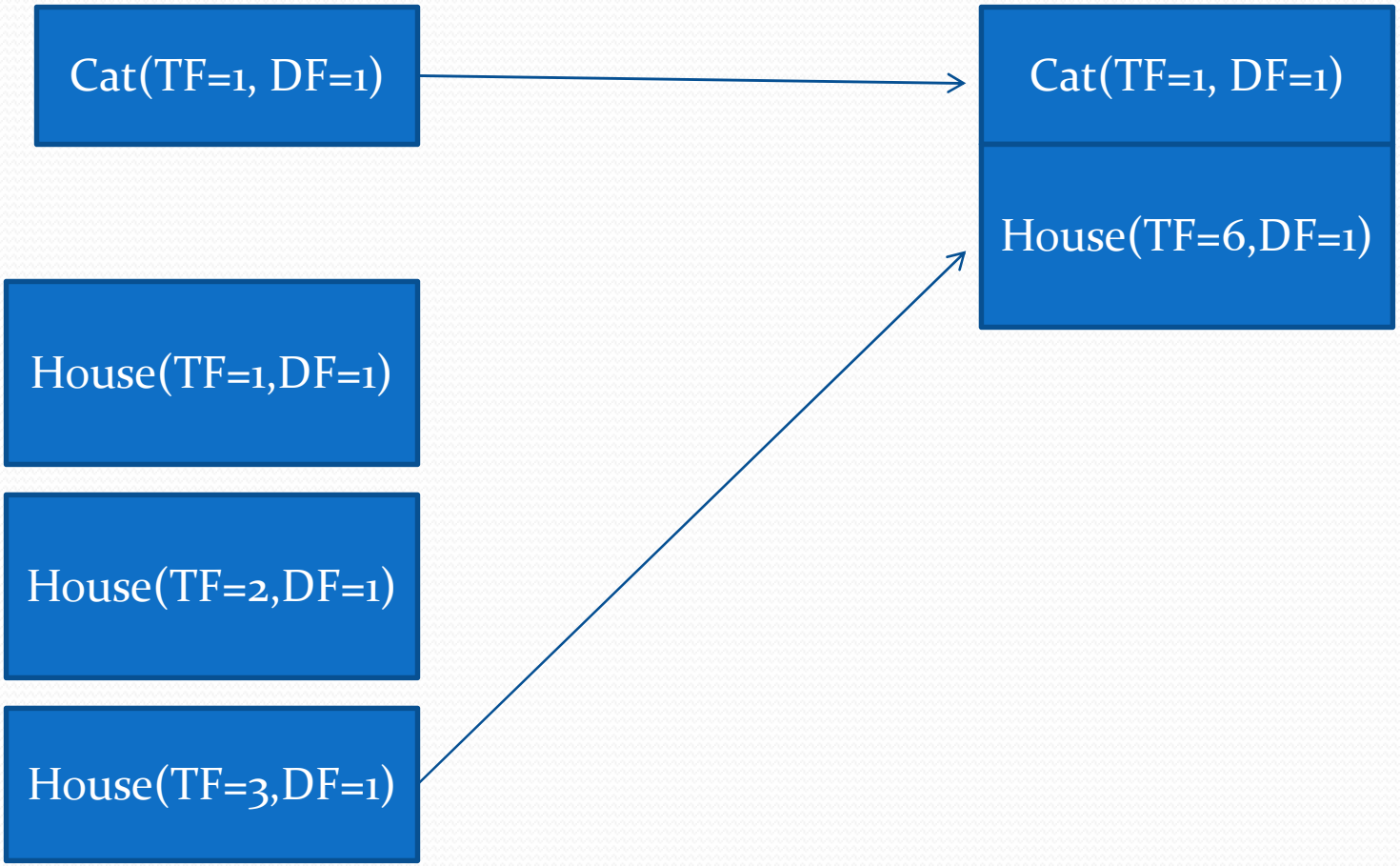
- Suffix Array Calculation
  - Every word has a suffix array associated with it
  - Allocating memory to suffix array based on the alpha output
  - Filling the details of suffix arrays of all words
    - Getting the position of the word in the file
    - Getting the file index of the file the word is in
    - Assigning term frequency
    - Assigning document frequency

Contd…

- Sorting the Suffix arrays
  - Based on Quick sort algorithm
    - Timing Complexity of quick sort : O(NlogN) (average case)
    - Memory Requirement : O(NlogN)

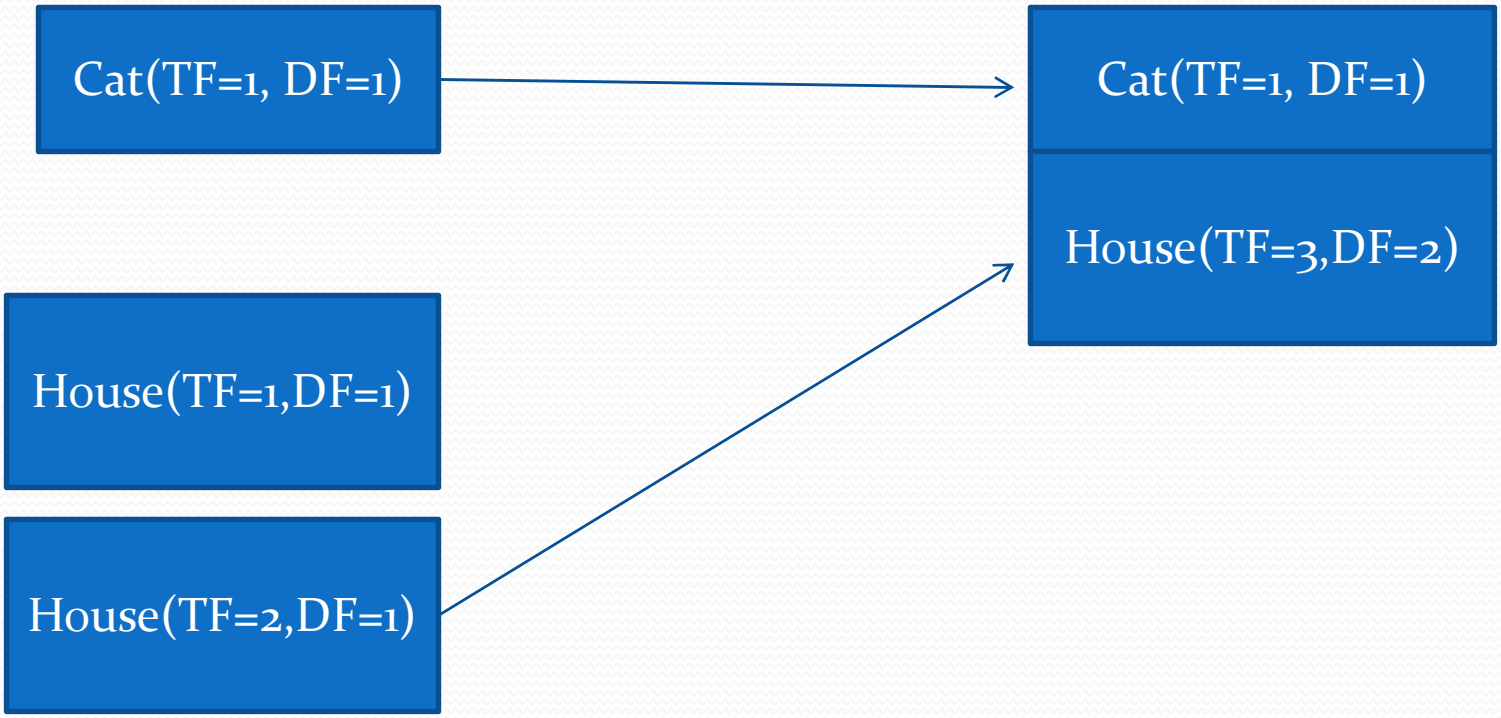
Contd…

- Finding Distinct terms in same article



Contd…

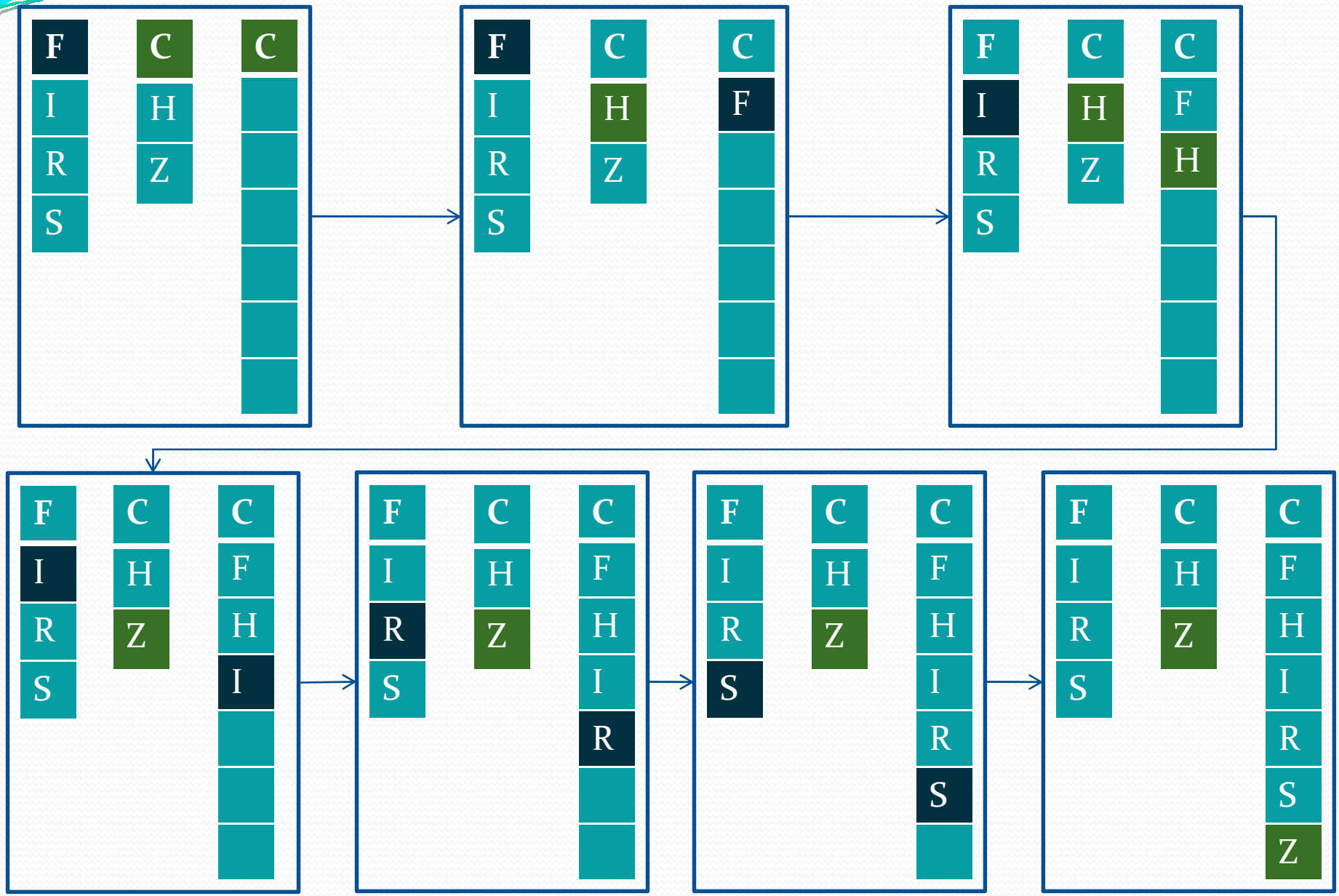
- Finding Distinct terms in different articles



Contd…

- Merging Suffix Arrays
  - Input: Two sorted suffix arrays
  - Reading ngrams from file
  - Output: One sorted suffix array

MERGE EXAMPLE

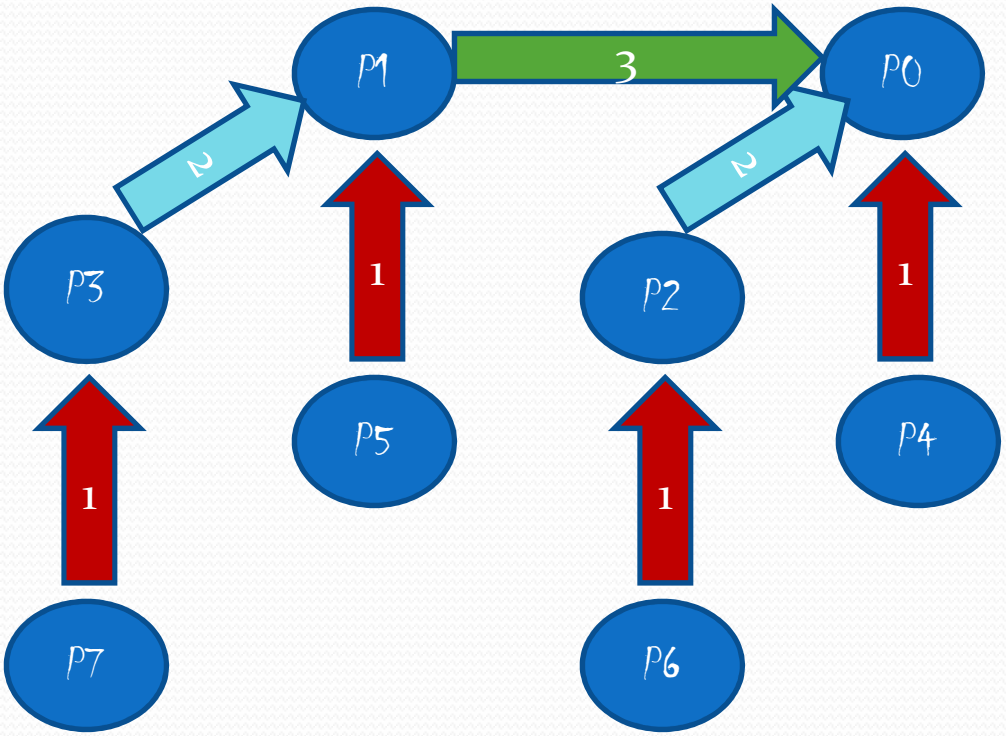


Contd…

- Communication Strategies
  - Reading and Writing files (Strategy 1 - deprecated)
    - Binomial Tree Reduction and Nomenclature
      - Use of MPI_Barrier
    - Single file corresponding to suffix array
  - Communicating Structures (Strategy 2)
    - Binomial Tree Reduction
      - Use of MPI_Pack, MPI_Unpack()

Contd…

- Binomial Tree Reduction



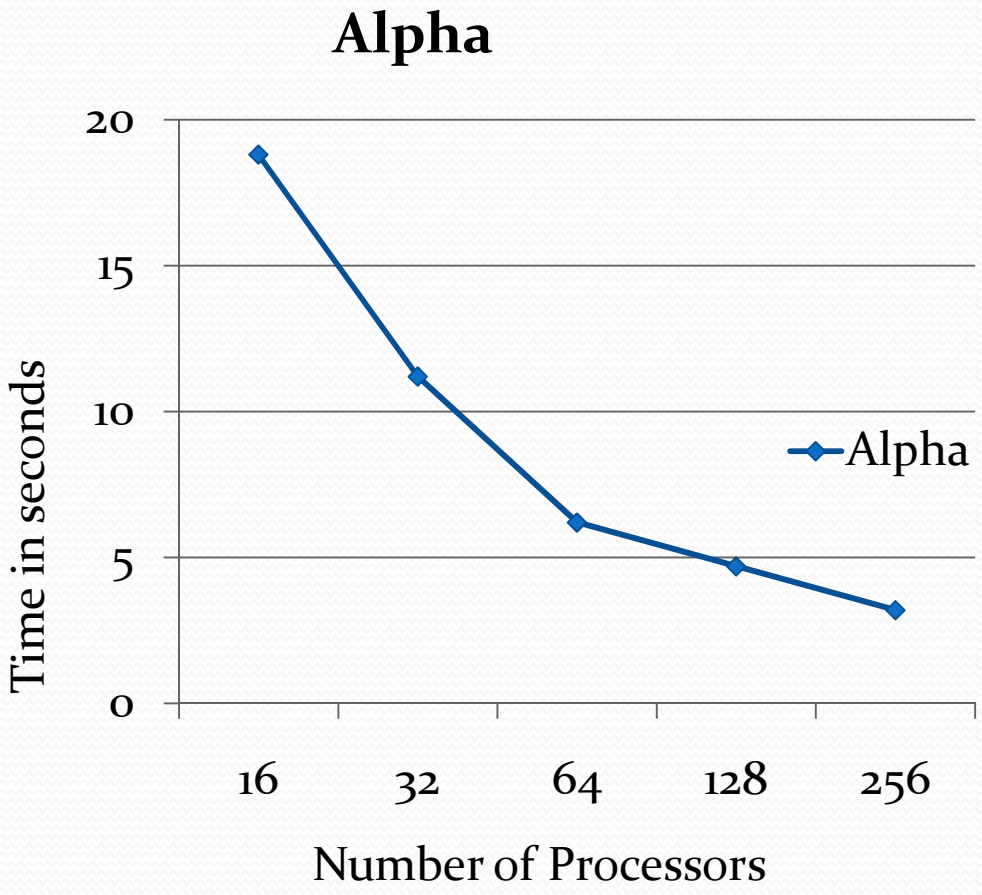
Contd…

- Finding top R interesting terms
  - Calculation and Storage
    - New suffix array structure with IDFTF measure
  - Sorting
  - Merging

Analysis

- Alpha

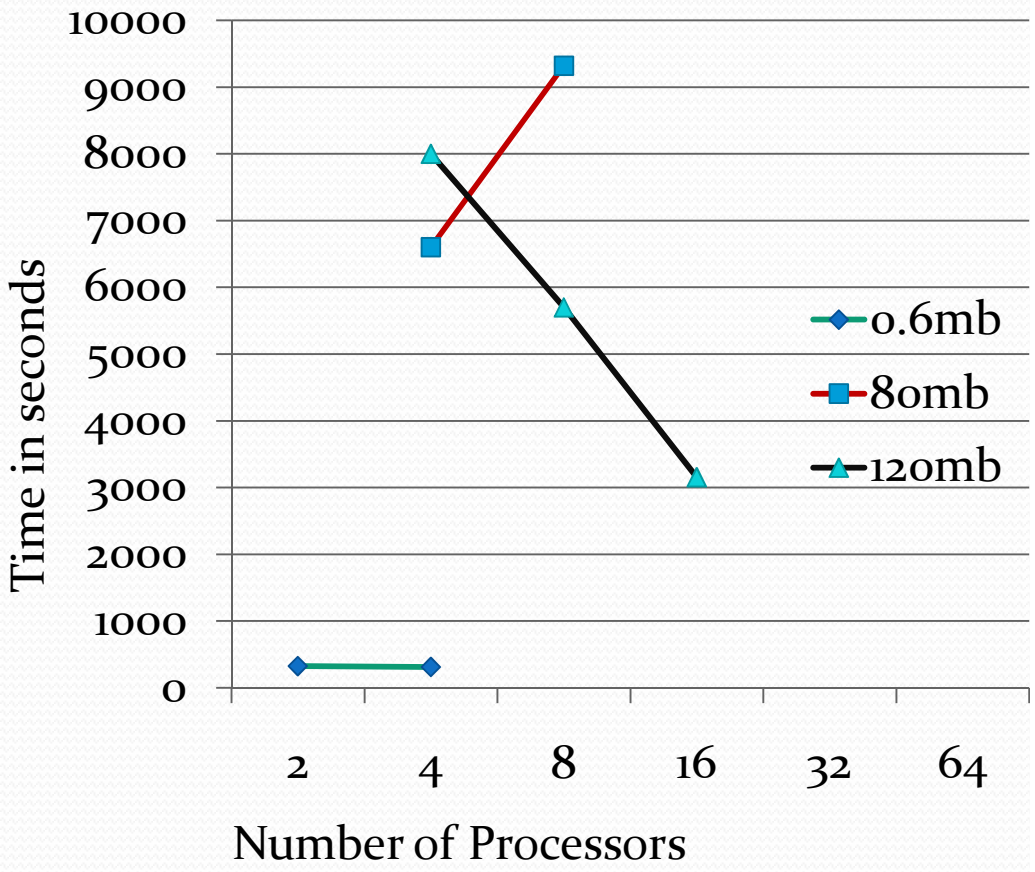
#P	Time in secs
16	18.8
32	11.2
64	6.2
128	4.8
256	3.2



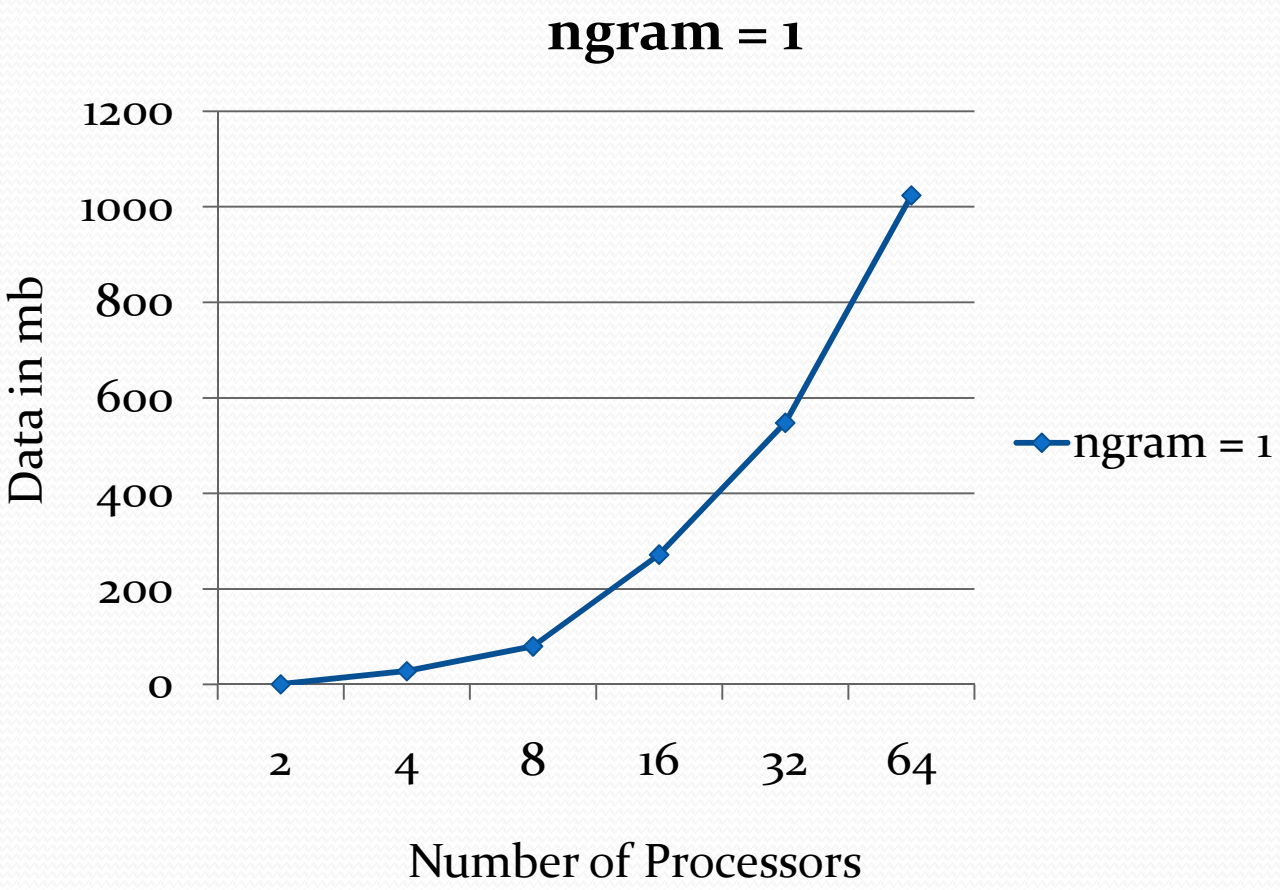
#P	Time in secs	Data
2	326	0.6MB
4	314	0.6MB

#P	Time in secs	Data
4	6600	80MB
8	9317	80MB

#P	Time in secs	Data
4	8000	120MB
16	3156	120MB



#P	Data in MB	Execution Time in sec
2	0.6	326
4	28	1620
8	80	9317
16	272	7297
32	548	12000
64	1024	14694



Formula

- Amdahl's Law
  - $\Psi <= 1/f+(1-f)/p$
    - where f is the serial component and p is the number of processors
    - Ψ is the speedup
- Gustafson's Law
  - $\Psi <= p+(1-p)s$
    - Ψ is the scaled speed up
    - s is the serial component and p is the number of processors

Contd…

- Using our results for data of size 120 MB
  - Speed up = 7680/3156=2.4
    - Considering the case where 4 processors as serial and 16 processors as parallel
  - Using the formula for Amdahl's Law and substituting Ψ as 2.4 we get f = 0.22
- According to Gustafson's Law using s = 0.22, Ψ (scaled speed up) = 3.34

Contact Info

- Project web page: giga word corpus
- Email
- Pavan Poluri: poluro07@d.umn.edu
- Siddharth Deokar: deoka001@d.umn.edu
- Varun Sudhakar: sudha002@d.umn.edu