Enhancing the Performance of Data Intensive Systems: Filtering and Indexing

Shikharesh Majumdar, PhD, PEng, FIET
Chancellor's Professor

Director, Real Time and Distributed Systems Research Centre

Dept. of Systems & Computer Engineering,
Carleton University,
Ottawa, CANADA.
email: majumdar@sce.carleton.ca

Acknowledgments

Carleton University:

Abdulla Kalandar Mohideen
Bannya Chanda
Marc St-Hilaire

Telus:

Ali-El-Haraki

Financial Support:

- Natural Sciences and Engineering Research Council of Canada (NSERC)
- Telus

Outline of Presentation

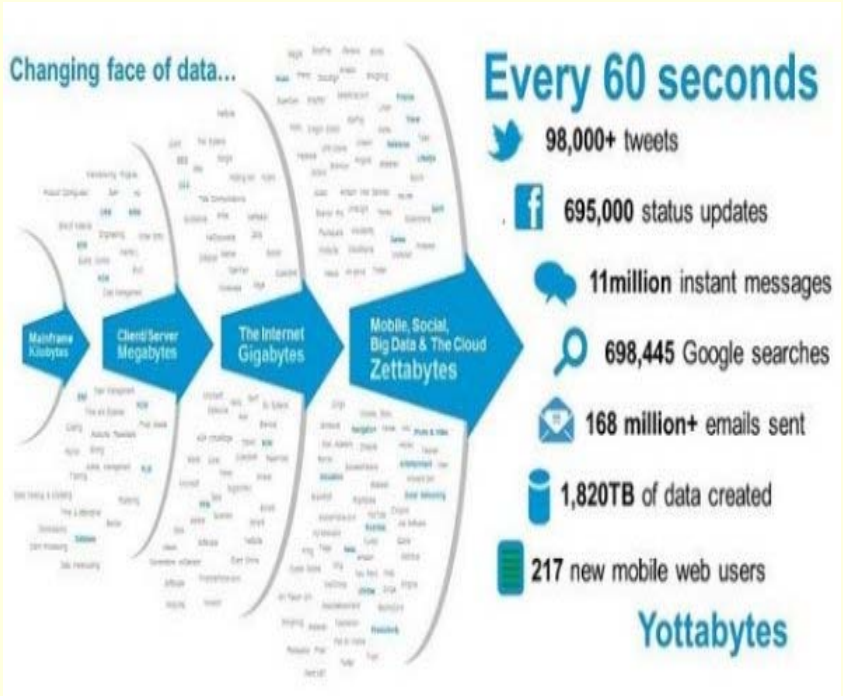
- Background
- Sources of Big Data
  - Data Analytics in Enterprises, Smart Systems (Smart Facilities Management) and Social networks
- Challenges
- Approaches to Performance Improvement
  - Data Indexing
  - Data Filtering
- Summary and Conclusions

Sources of Data

Analysis of Enterprise Data

https://www.indiamart.com/proddetail/business-intelligence-and-big-data-analysis-service-19722247930.html

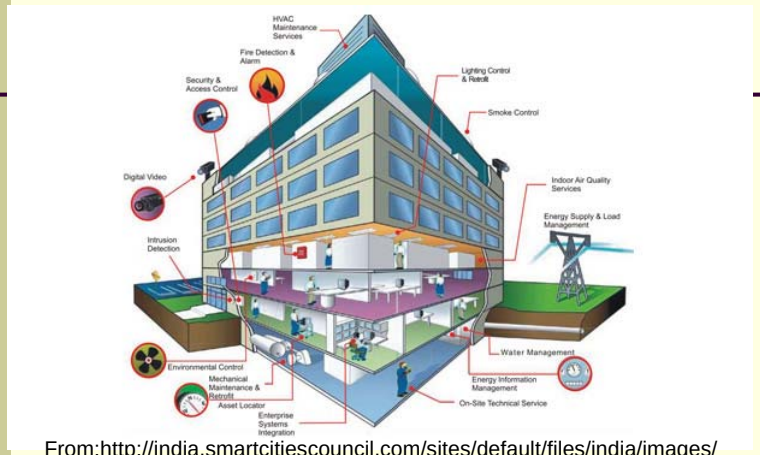
https://twitter.com/Ronald_vanLoon/status/922284106200870912/photo/1



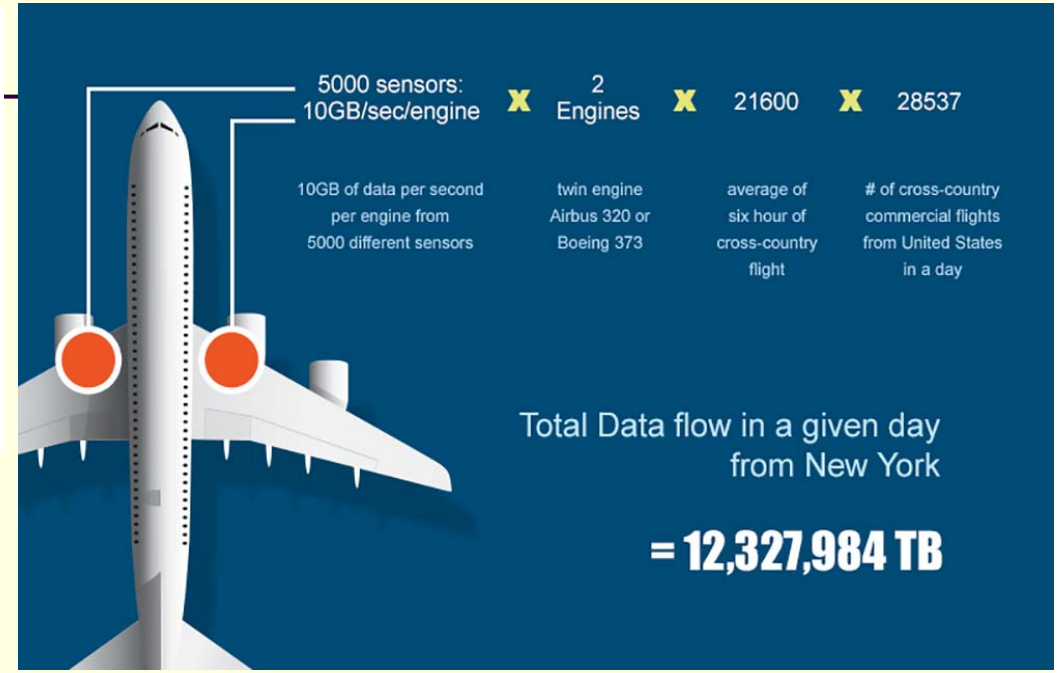
Multiple forms of Data

(From Article by R. vanLoon, 2017)

Smart Facilities: Key to a Smart Society
[Ex: Building/Bridges, Aerospace Systems Cameras]



From:http://india.smartcitiescouncil.com/sites/default/files/india/images/Smart-Buildings-Key-to-smart-cities.jpg



From: https://www.google.com/search?q=bigdata+aeroplane&tbm=isch&ved=2ahUKEwjljOuAs5TsAhWRX80KHdg8CMgQ2cCegQIABAA&oq=bigdata+aeroplane&gs_lcp=CgNpbWcQAzoCCAA6BQgAELEDOgQIABBDOgQIABAeOgYIABAKEBhQgKQRWNXgEWDP4hFoAXAAeACAAVOIAf0JkgECMTiYAQCgAQGqAQtnd3Mtd2l6LWltZ7ABAMABAQ&sclient=img&ei=OFJ2WQL5G_tQbY-aDADA&bih=636&biw=1016&hl=EN#imgrc=EPYADdPo-2Y5dM

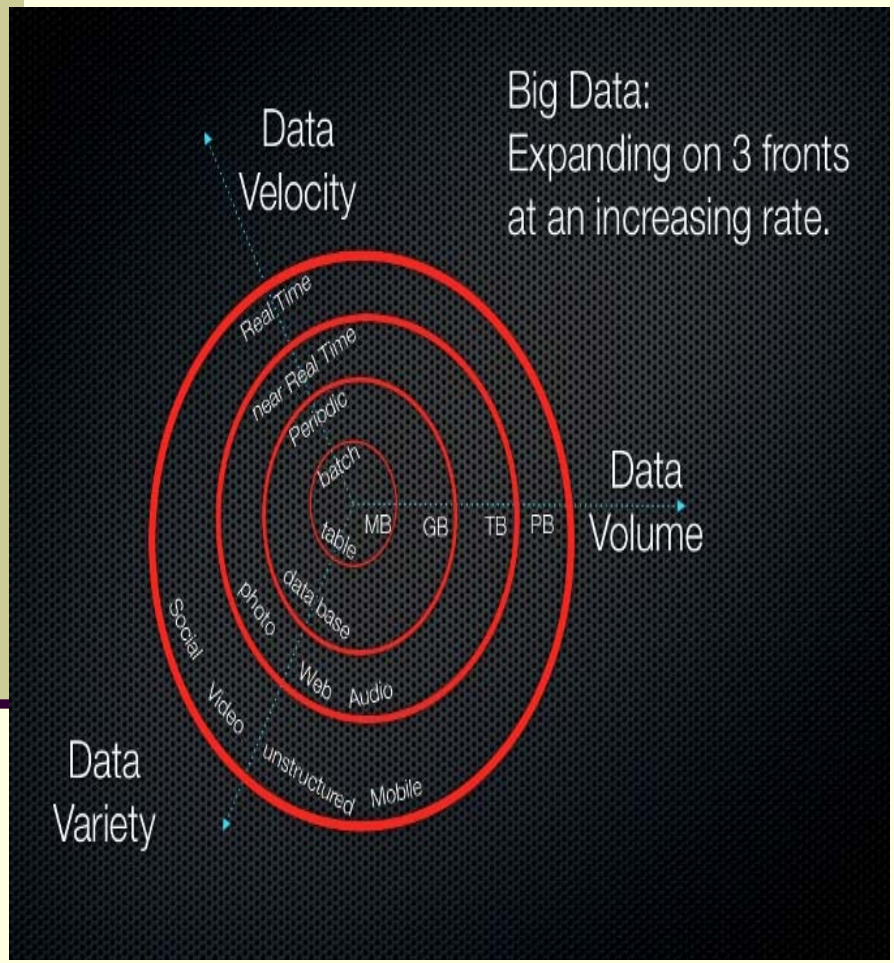
Street and Building & Street Cameras

Smart Buildings

- Responsive to user needs
- measure, monitor, control,
- and optimise building operations and maintenance
- Internal and external data streams
- Sensor-data based real time control
- Security and access control
- Intrusion detection
- Optimization of building performance (including energy optimization)

Both ***Stored*** and ***Streaming*** Data

Data Intensive Systems : Challenges



- ***Volume: Very large scale data files***
- ***Variety: Many Types***
  - Text, Images, numeric
- ***Velocity:*** Flows of data streams
  - Internet of Things (IoT)
  - Twitter feeds
  - Timeliness of response
- ***Data Processing: can be time consuming***

Information → *Knowledge*

Two Techniques:

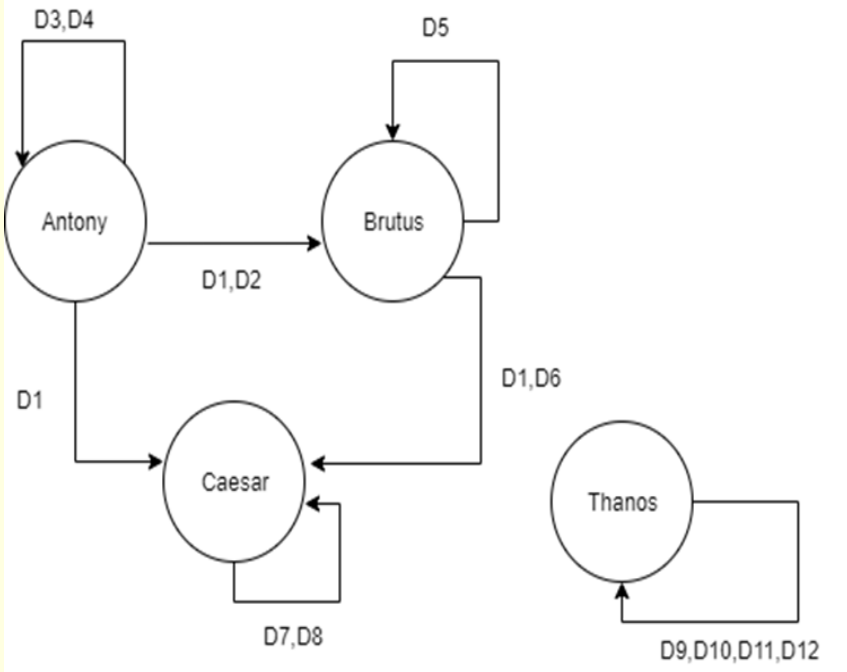
- Data Indexing
- Selective Data Filtering

Data Indexing

- Indexing: Mapping of keywords to documents they appear in
- Speeds up searching operations
- Example: ***Inverted Index***
- Popular technique (e.g. used in Elastic Search)
- Example
  - Antony -> D1,D2,D3,D4
  - Brutus -> D1,D2,D5,D6
  - Caesar -> D1,D6,D7,D8
  - Thanos > D9, D10, D11, D12

  Implies:
  - Keyword “Antony” appears in document D1,D2,D3,D4 and Brutus appears in document D1,D2,D5,D6 and so on.
  - Problem: retrieving all matched document ids at the same time
  - Performing conjunction of keywords
- Searching for documents containing multiple keywords
  - For handling Boolean search operations

- Our Solution:
- Graph-Based Indexing Technique (GBIT) for text data

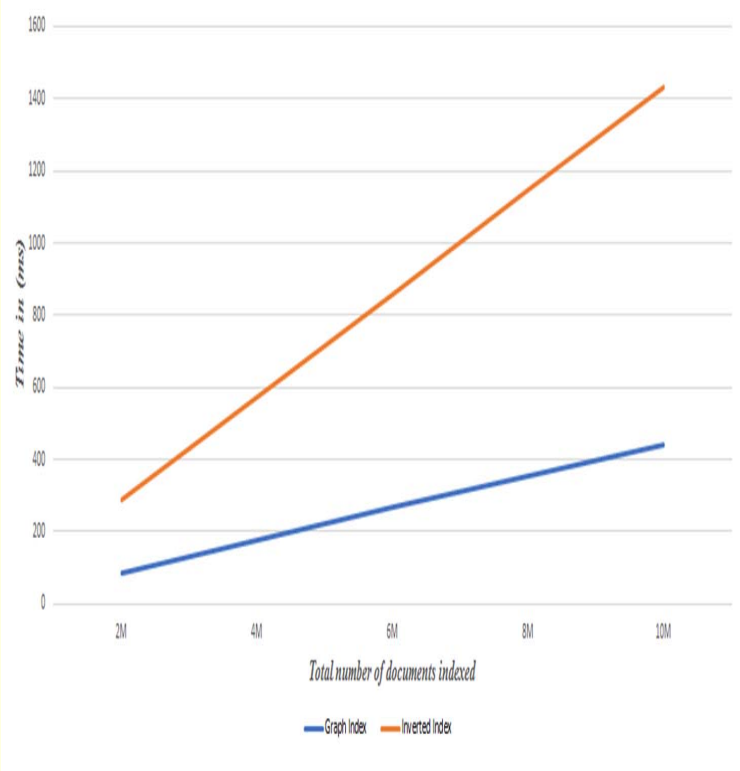


- Node: Keyword. Edge: Document Id

Search Times for Boolean AND Queries

- Boolean AND (Inverted Index vs G-BIT):

Search Time comparison (for different number of documents indexed)



- GBIT - significantly lower search times
- Overhead: GBIT has a higher indexing time

Reference:

Ref: A. K. Mohideen, S. Majumdar, M. St-Hilaire and A. El-Haraki, "A Data Indexing Technique to Improve the Search Latency of AND Queries for Large Scale Textual Documents," *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, 2020, pp. 37-46, doi: 10.1109/BDCAT50828.2020.00019.

- GBIT: Can also handle Other operations e.g. OR and NOT

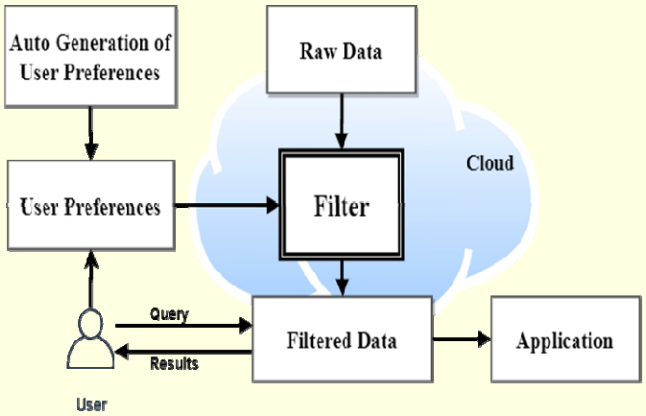
Carleton Research Computing and Development Cloud (RCDC): IBM POWER8E processor with 64 GB memory running Ubuntu

Selective Filtering

- Users often need to search through large volumes of text data.
  - i.e., articles in newspapers or journals, meeting minutes, medical records, and tweets.

https://mercomcapital.com/tag/electronic-medical-records/

https://www.google.com/search?q=news+paper+data+sets&tbm=isch&ved=2ahUKEwir1tmQkqfxAhWGf6wKHTeaDzsQ2-cCegQIABAA&oq=news+paper+data+sets&gs_lcp=CgNpbWcQA1CEdljThQFg_Z4BaABwAHgAgAGBAYgBjgOSAQM0LjGYAQCgAQGqAQtnd3Mtd2I6LWltZ8ABAQ&sclient=img&ei=O7HPYOvMD4b_sQW3tL7YAw&bih=657&biw=1024&hl=EN#imgrc=DXuvDZuZ0kjqpM



Raw Data:
one or multiple text files.

User Preferences:
Key words, sentences .e.g., dates, names, products

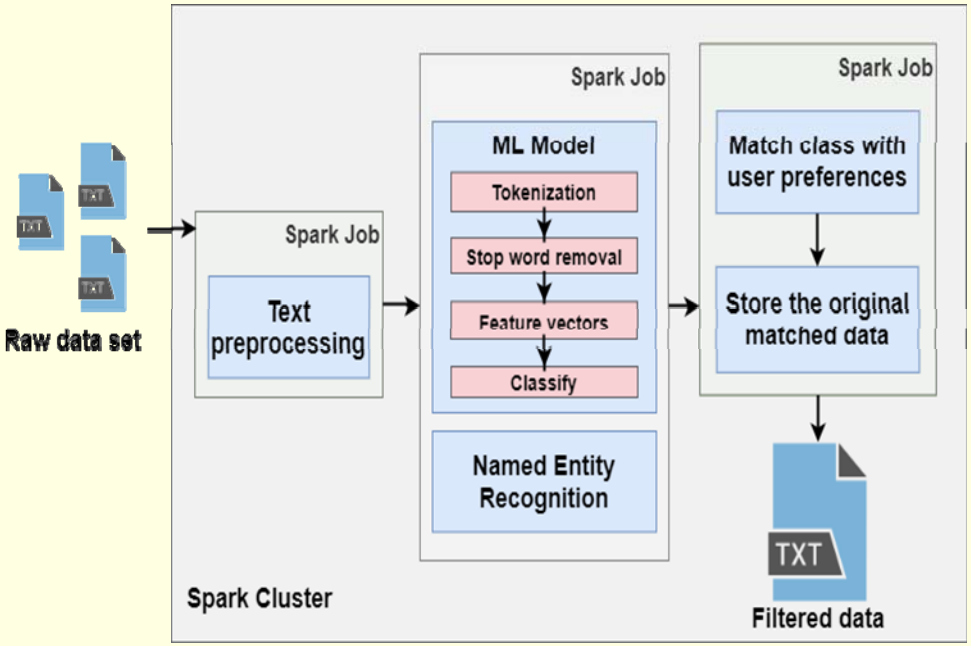
Filtered Data:
The output data for the filtering algorithm applied on the raw data set.

User Queries:
Get data related to given keywords

Application:
A program that processes the filtered data for a specific purpose.

Filtering: Apache Spark Based Approach

- Large Filtering Time:
  - Reduced by using parallel processing



- Text preprocessing
  Preprocess the texts from raw data set by eliminating punctuations and lower casing the capital letters.
- Named entity recognition
  Pre-trained python library “spaCy” is used to extract named entities from the processed raw data.
- Machine learning model
  A multinomial logistic regression classifier classifies the text according to the different categories.
- This classifier has been formed using the training dataset obtained from the Cognitive Computation Group of University of Illinois. [3][4].
-
- After classifying the raw data, the filter method filters data whose class matches with user preferences and stores the filtered data as a comma-separated file.

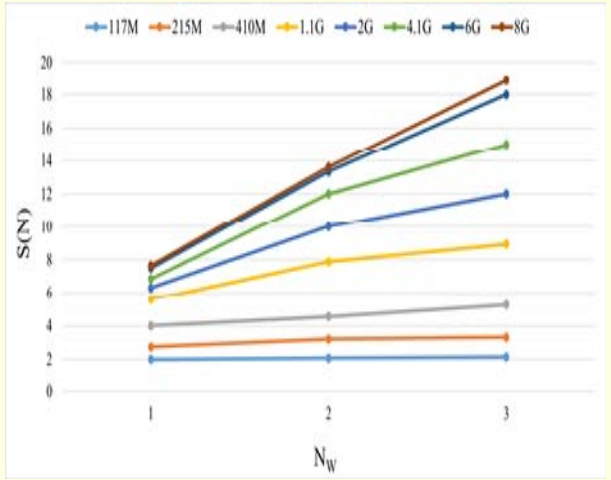
Reference: B. Chanda, S. Majumdar"A Parallel Processing Technique for Extracting and Storing User Specified Data“, **IEEE 8th International Conference on Future Internet of Things and Cloud (FiCloud 2021)**, August 2021

Filtering Time and Filtering Efficiency (Search Latency)

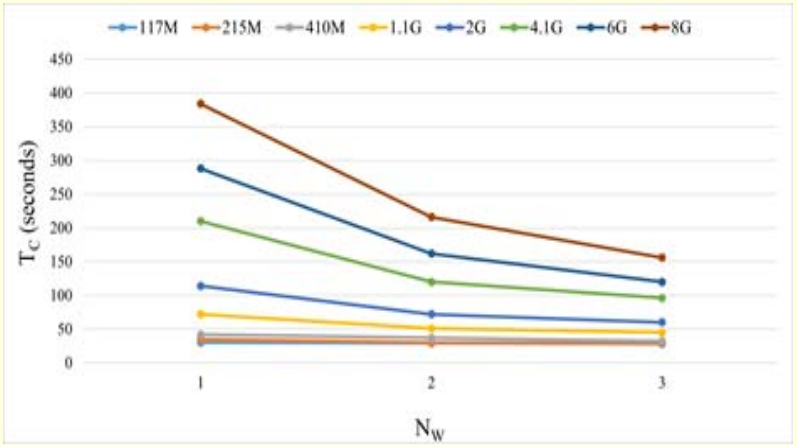
Filtering Efficiency (E_F) =
search latency (Non-Filtered Data)/
search latency (Filtered Data)

Prototype running on Amazon EC2 Cloud

Search method	Search by	E_F
Sequential	Keywords	105
Sequential	Sentences	57.6
Parallel	Keywords	63.6
Parallel	Sentences	29



Each worker node comprises multiple cores



Impact of Parallelism on Filtering Time

Impact of Parallelism on Speedup

N_W = No. of Worker Nodes

Summary & Conclusions

- Processing large text data is resource consuming
  - Requires large storage volume
  - Often gives rise to large search latency
- Data Indexing
  - GBIT- A graph based data indexing technique
    - Useful for retrieving all matches for a search operation at once
    - Efficiently handles boolean operations in search queries
      - AND, OR, NOT
    - Led to a significant performance improvement over inverted index for the synthetic data set experimented with
- Selective Data Filtering
  - Leads to a large reduction in “user preferred” data
  - Filtering time reduced by using a Spark-based parallel processing technique
  - Significant reduction in volume of stored data
  - Large reduction in data search latency (e.g. 10,500%)