# Big Data Dwarfs: Methodology, Dwarf Library and Simulation Benchmarks

**Wanling Gao**

***Institute of Computing Technology, Chinese Academy of Sciences***

**BPOE 2017 Xi'an, China**


中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

# Simulation for Big Data Analytics

- Challenges
  - Complexity and diversity of big data workloads
  - Rapid evolution of big data systems
  - Complex big data software stack
    - Standalone mode
    - Pseudo-distributed mode
    - Fully-distributed mode
- Existing big data benchmarks cannot run on simulators because of long running time

# Dwarf-based Simulation

- Big data dwarfs
  - a minimum set to represent maximum patterns of big data analytics
- A light-weigh simulation benchmark on the basis of big data dwarfs
  - Shorten the simulation time by 100s times comparing to big data analytics workloads
  - Average micro-architectural data accuracy is above 90% on X86 and ARMv8 processors

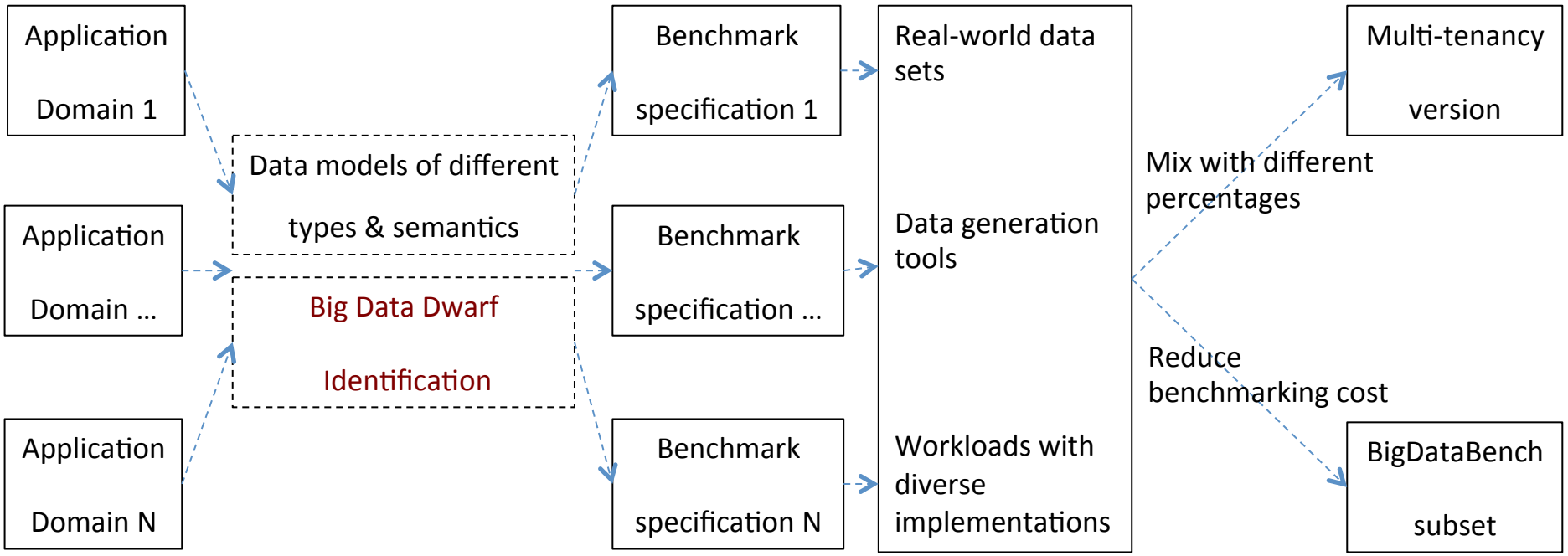
# Outline

- BigDataBench - a big data benchmark suite
- Big Data Dwarfs & Dwarf Library
- Dwarf-based Simulation Benchmarks
- Evaluation on X86 Processor
- Case Study on ARM Processor

# What is *BigDataBench*?

- An open source big data benchmarking project
  - http://prof.ict.ac.cn/BigDataBench
  - Search Google using "**BigDataBench**"

# BigDataBench Methodology



# BigDataBench 3.2 Overview

**BDGS(Big Data Generator Suite) for scalable data**

| | | |
|---|---|---|
| Wikipedia Entries | Amazon Movie Reviews | Google Web Graph |
| Facebook Social Network | E-commerce Transaction | ProfSearch Resumes |
| ImageNet | English broadcasting audio | DVD Input Streams |
| Image scene | Genome sequence data | Assembly of the human genome |
| SoGou Data | MNIST | MovieLens Dataset |

**15 Real-world Data Sets**

Search Engine | Social Network | E-commerce

Multimedia | Bioinformatics

**37 Workloads**

Impala Flink JStorm Alibaba NoSql

GraphX Shark GraphLab Carnegie Mellon HIVE ORACLE MySQL

Hadoop RDMA MVAPICH MPI

Spark Lightning-Fast Cluster Comp hadoop DataMPI

**Software Stacks**

# BigDataBench Users

- http://prof.ict.ac.cn/BigDataBench/users/
- Industry users
  - Accenture, BROADCOM, SAMSUMG, Huawei, IBM
- About 20 academia groups published papers using BigDataBench
- **BigDataBench support for graph, streaming frameworks and Flink**

# Industry Standard: BigDataBench-DCA

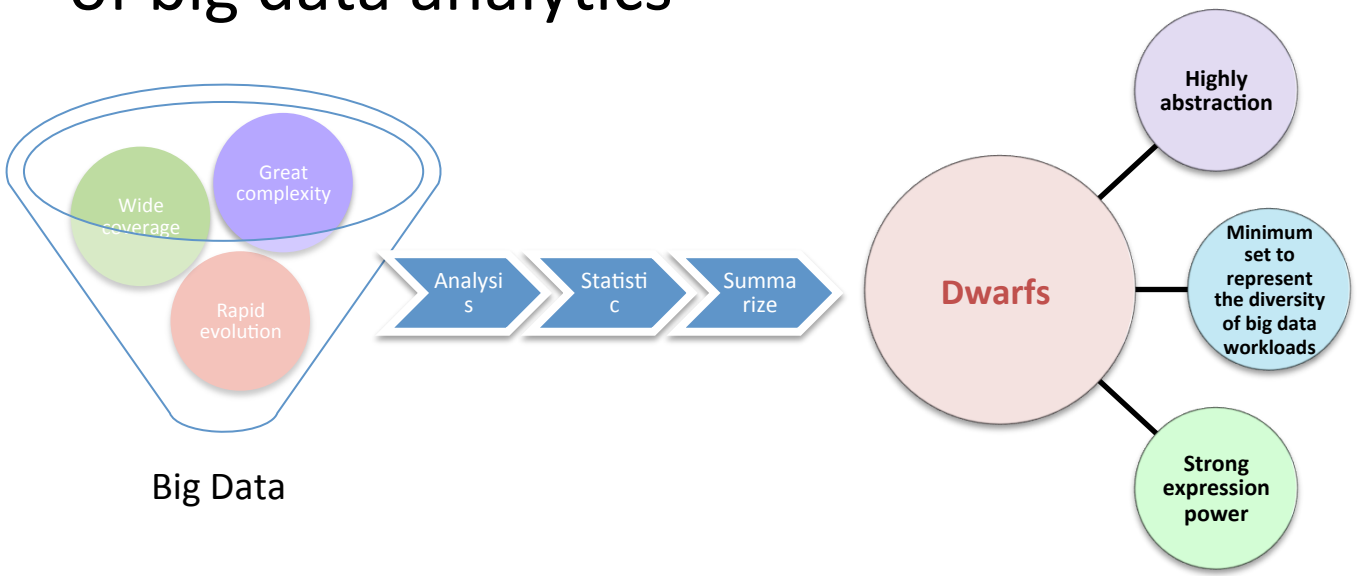
- China's first industry-standard big data benchmark suite
  - http://prof.ict.ac.cn/BigDataBench/industry-standard-benchmarks/
  - Academy of Information and Communications, ICT, CAS, Huawei, China Mobile, Sina, ZTE, Intel (China), Microsoft (China), IBM CDL, Baidu, INSPUR , ZTE, 21viane and UCloud

# Outline

- **BigDataBench - a big data benchmark suite**
- **Big Data Dwarfs & Dwarf Library**
- **Dwarf-based Simulation Benchmarks**
- **Evaluation on X86 Processor**
- **Case Study on ARM Processor**

# Big Data Dwarfs

- Abstractions
  - Frequently-appearing operations
  - ***a minimum set*** to represent ***maximum patterns*** of big data analytics



# Inspiration

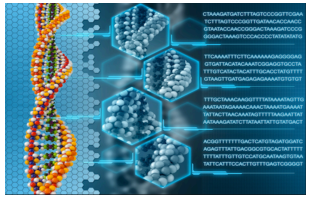
***Successful Compute Abstractions***

- Relational algebra
  - 5 primitive operations
  - Select, Project, Product, Union, Difference
- Parallel computing
  - Computational & communication patterns
  - 13 dwarfs

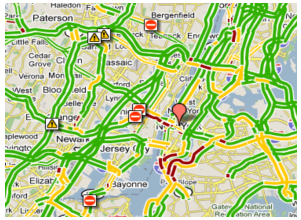
***Successful Benchmarks***

- TPC-C
  - OLTP domain
  - Functions of abstraction
- HPCC
  - High performance computing
  - Seven basically tests

# Challenges #1--Massive Application Domains

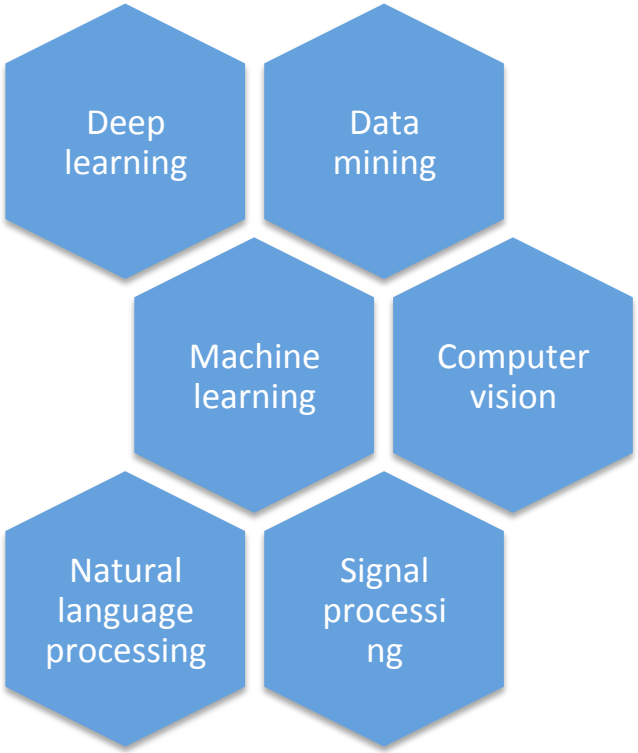



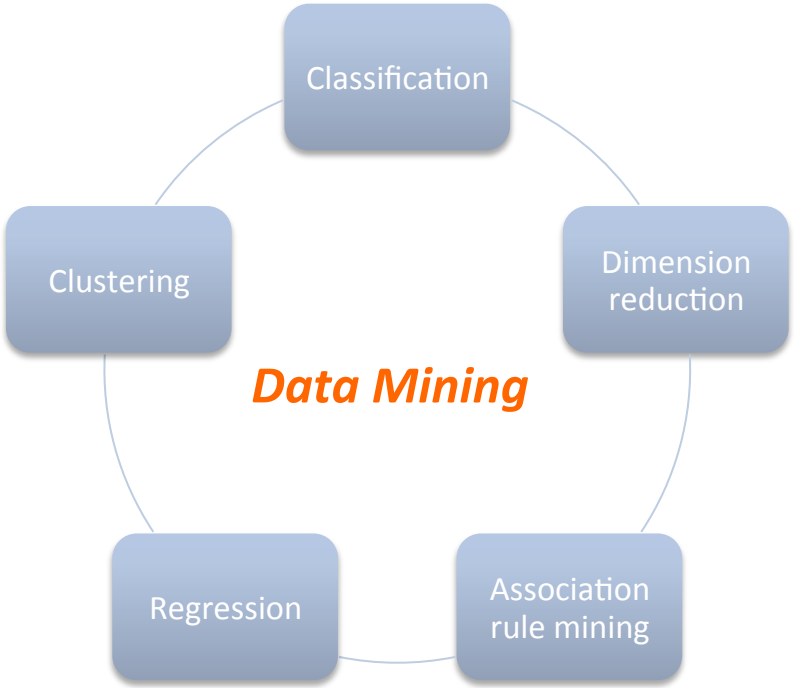
Big Data




Massive application domains make us wonder where to start or how to achieve a wide range of coverage

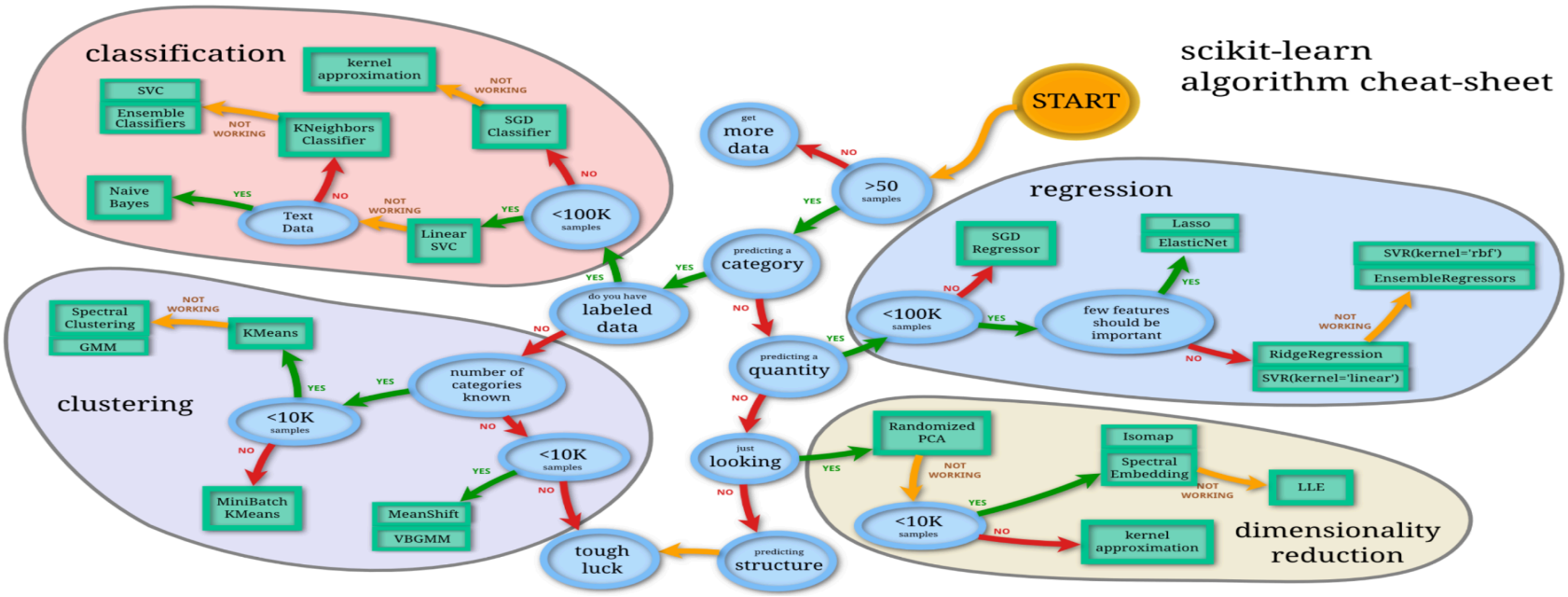
# Challenges #2—Multiple Research Fields and Techniques




Many techniques for processing big data exist, which bring greater complexity for identifying dwarfs workloads

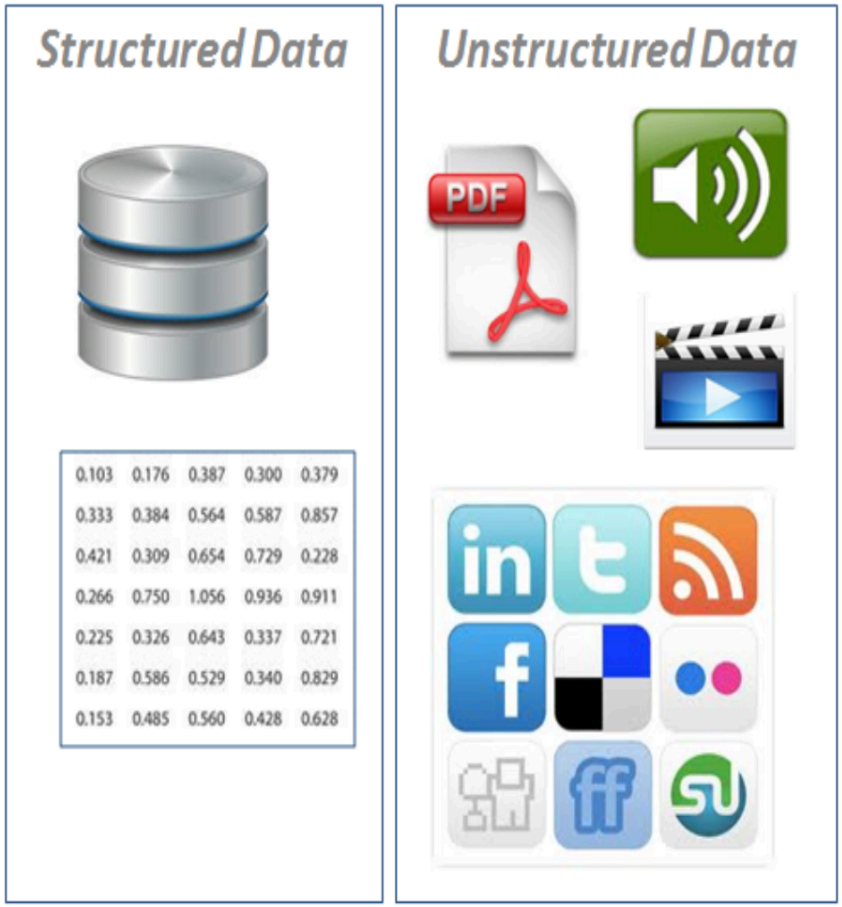
# Challenges #3—Large Numbers of Algorithms and Variants




A machine learning library – scikit learn, implements so many algorithms, which is still much less than the total number of algorithms

http://scikit-learn.org/stable/tutorial/machine_learning_map/

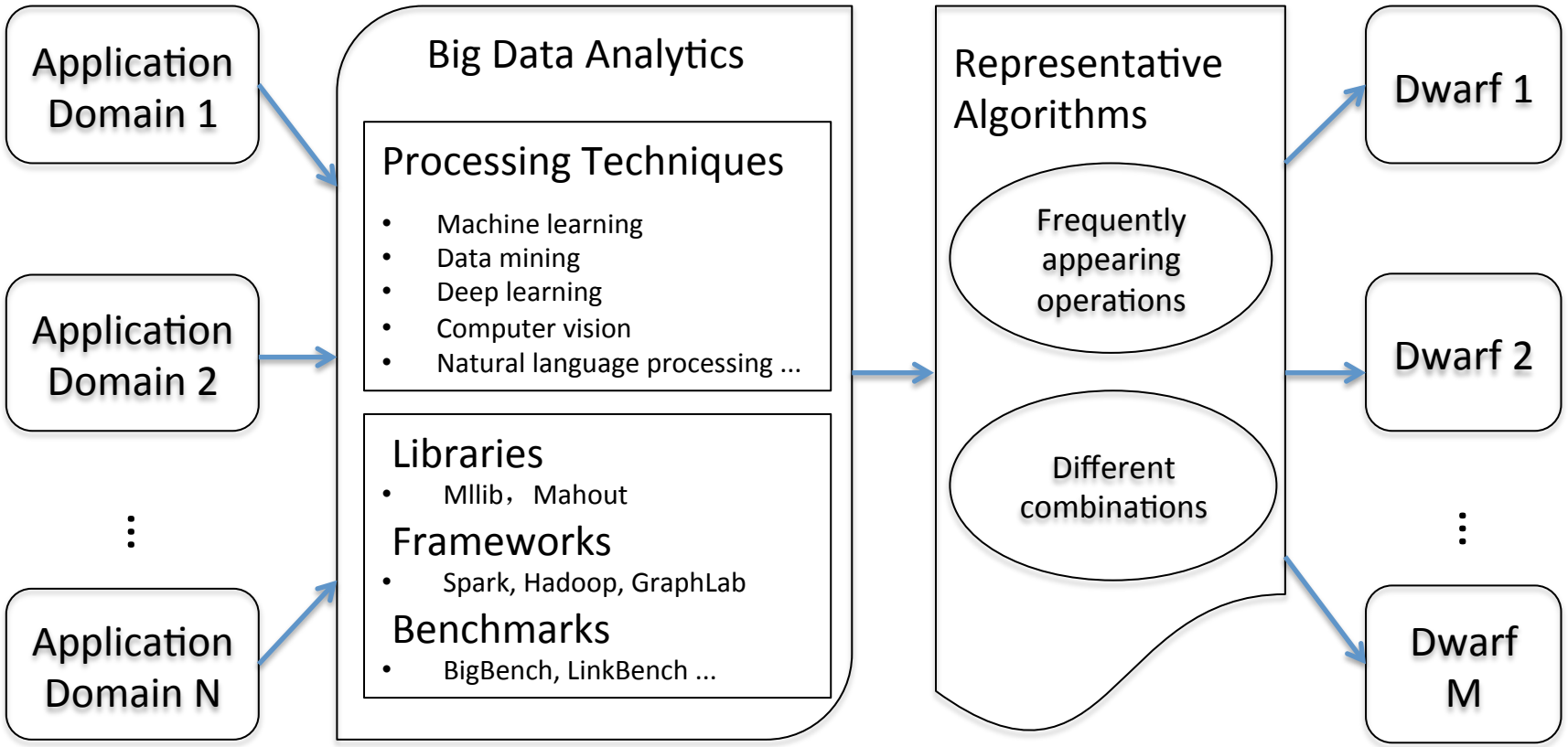
# Challenge #4—Unstructured Data with Complicated Operations



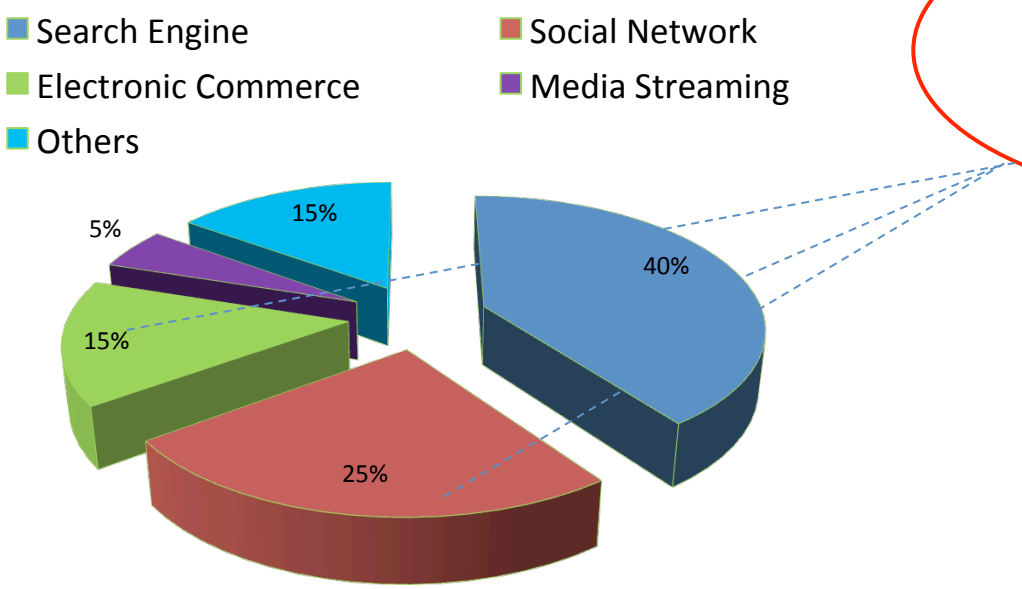

- *80%* data growth are unstructured data
- Operations on big data are *complicated*
  - Pipeline? Parallel?

https://www.capgemini.com/blog/capping-it-off/2014/07/are-you-effectively-using-big-data

# Methodology of Dwarfs



# Internet Services




Taking up 80% of internet services according to page views and daily visitors

Top 20 websites

http://www.alexa.com/topsites/global;0

# The Explosive Growth of Multimedia Data

**600+** new **VIDEOS** on YouTube every minute

**13000+** hours **MUSIC** streaming on PANDORA every minute

**6600+** new **PHOTOS** on FLICKR every minute

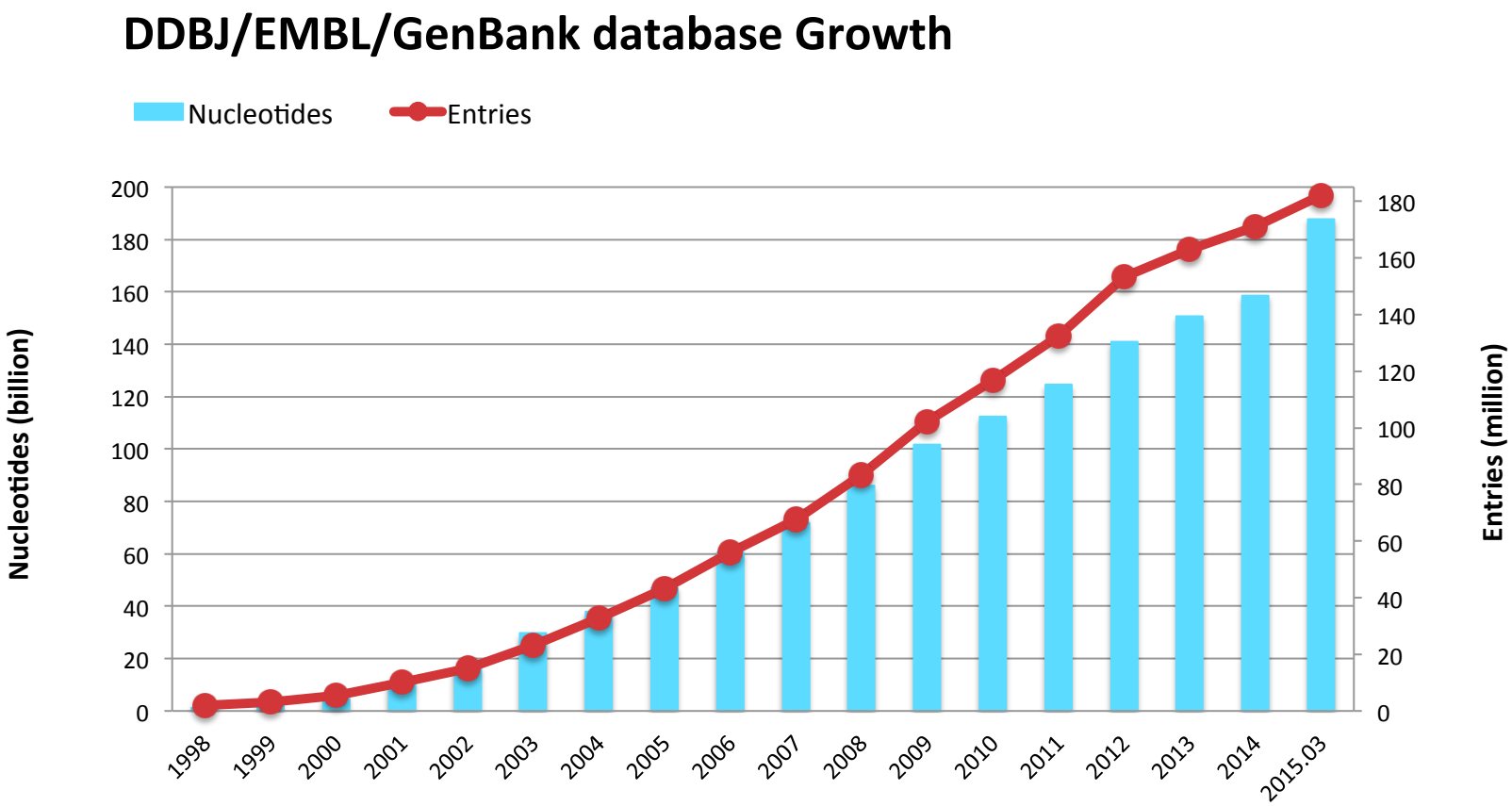
**100's** **VIDEO** feeds from surveillance cameras

**370000+** minutes **VOICE** calls on Skype every minute

**80%** data growth are **IMAGES, VIDEOS**, documents, …

*http://www.oldcolony.us/wp-content/uploads/2014/11/whatisbigdata-DKB-v2.pdf*

# The Explosive Growth of Bioinformatics Data




*http://www.ddbj.nig.ac.jp/breakdown_stats/dbgrowth-e.html#dbgrowth-graph*

# Five Application Domains


Top 20 websites

**Internet Service**

*Search engine, Social network, E-commerce*

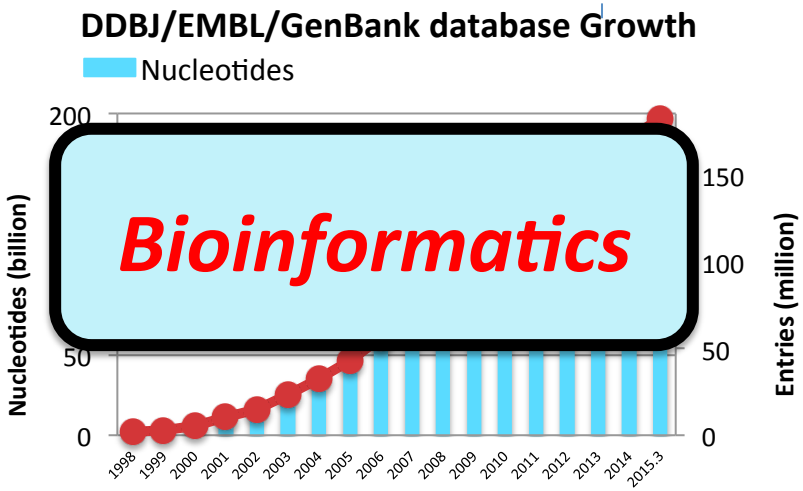
**600+** new **VIDEOS** on YouTube every minute

**100's** VIDE feeds from surveillance cameras

minutes **VOICE** calls on Skype every minute

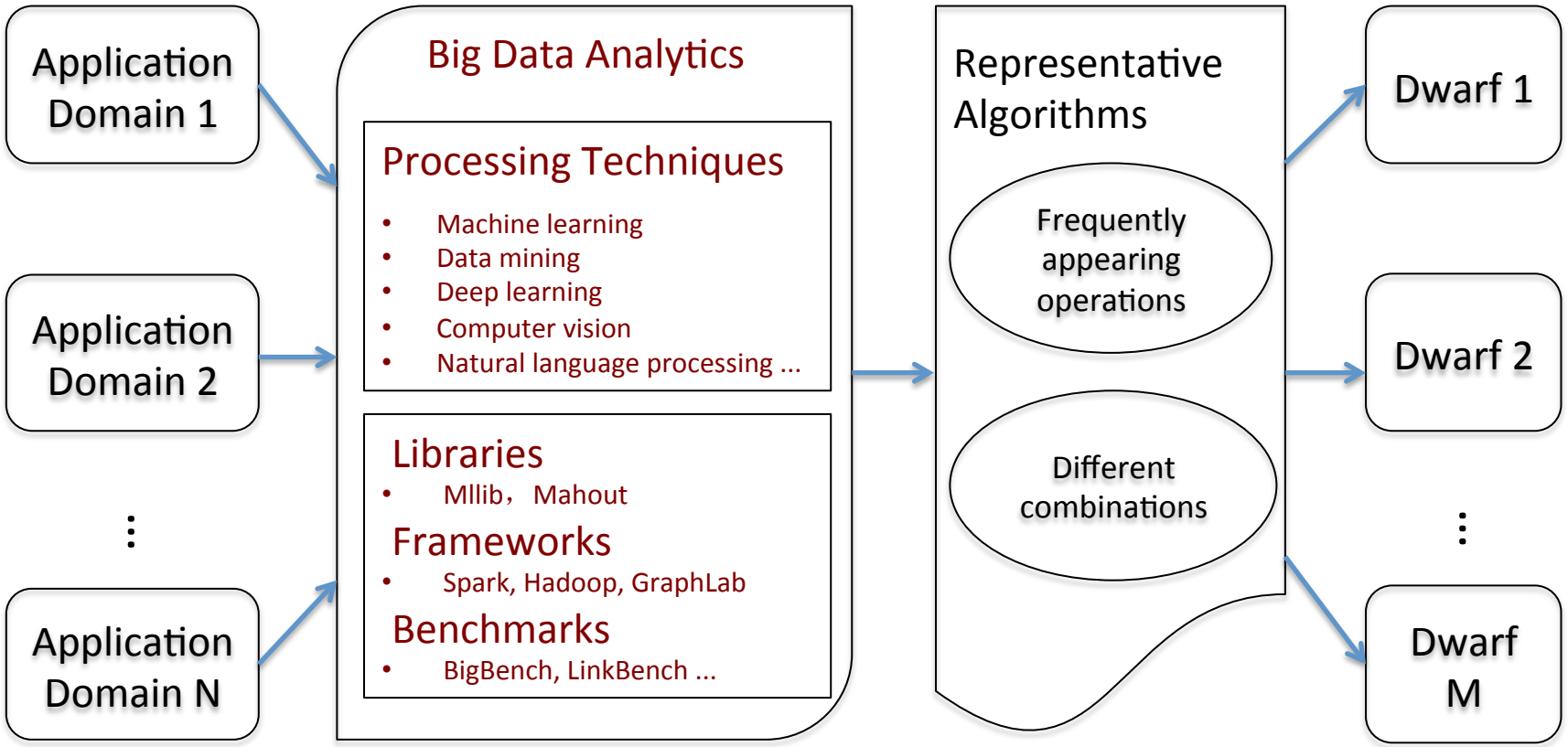
are **IMAGES**, **VIDEOS**, documents, …

**Multimedia**

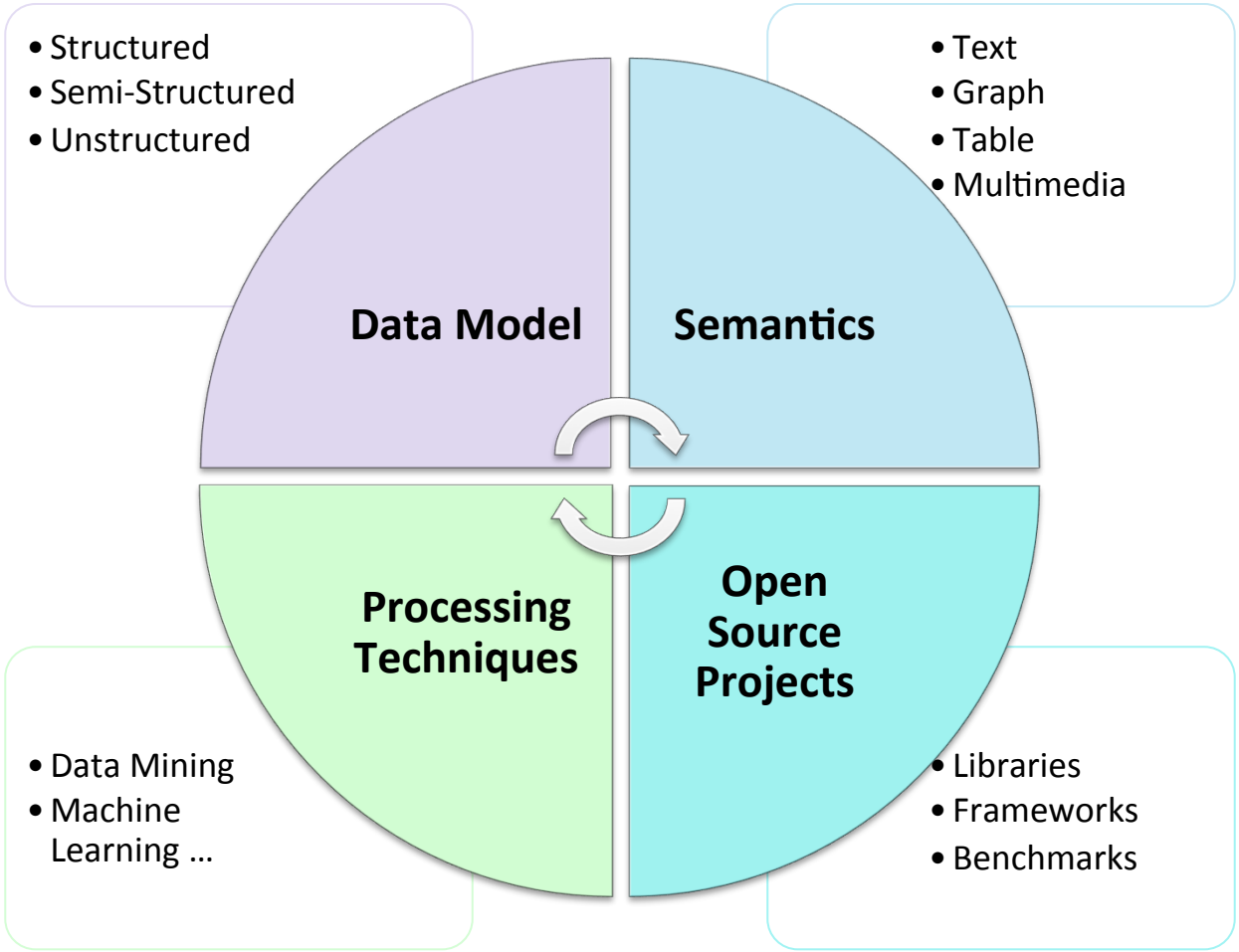



**Bioinformatics**

# Methodology of Dwarfs



# Big Data Analytics



# Libraries & Frameworks & Benchmarks

## Libraries

- Opencv, Mllib, Weka …

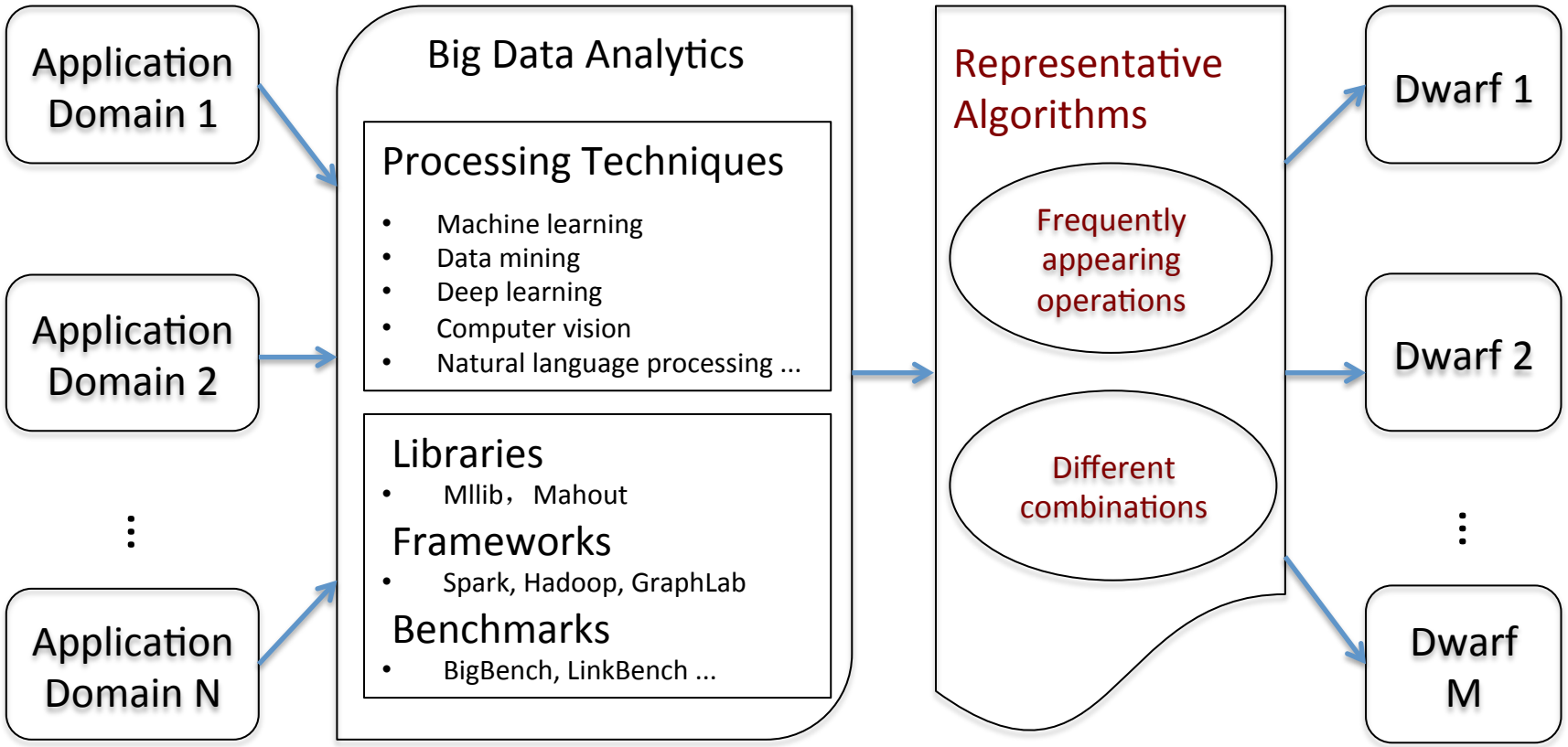
## Frameworks

- Spark, Hadoop, Graphlab …

## Benchmarks

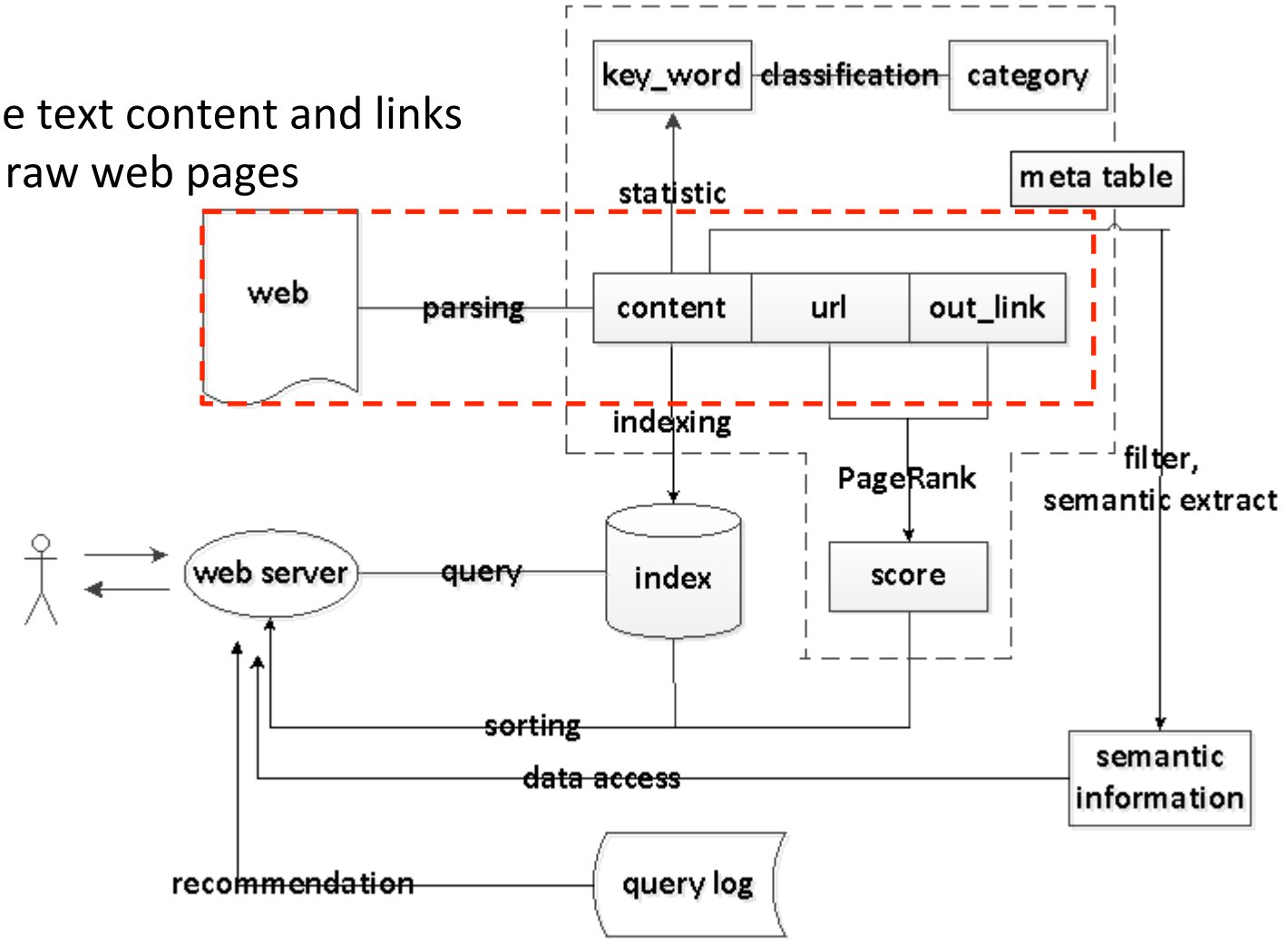
- BigDataBench, Linkbench …

# Methodology of Dwarfs



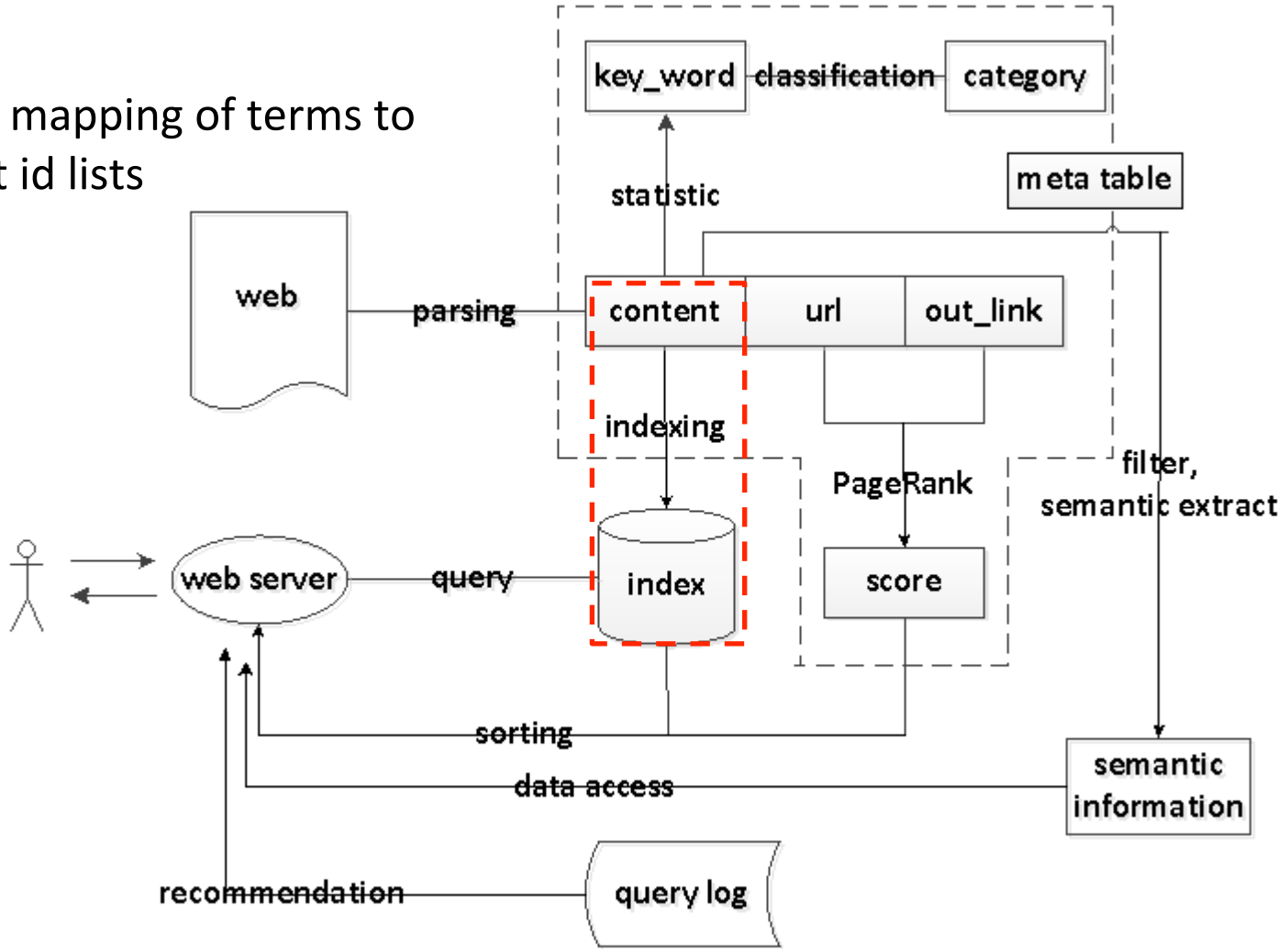
# Search Engine: Parsing

- Parsing:
  - Extract the text content and links from the raw web pages



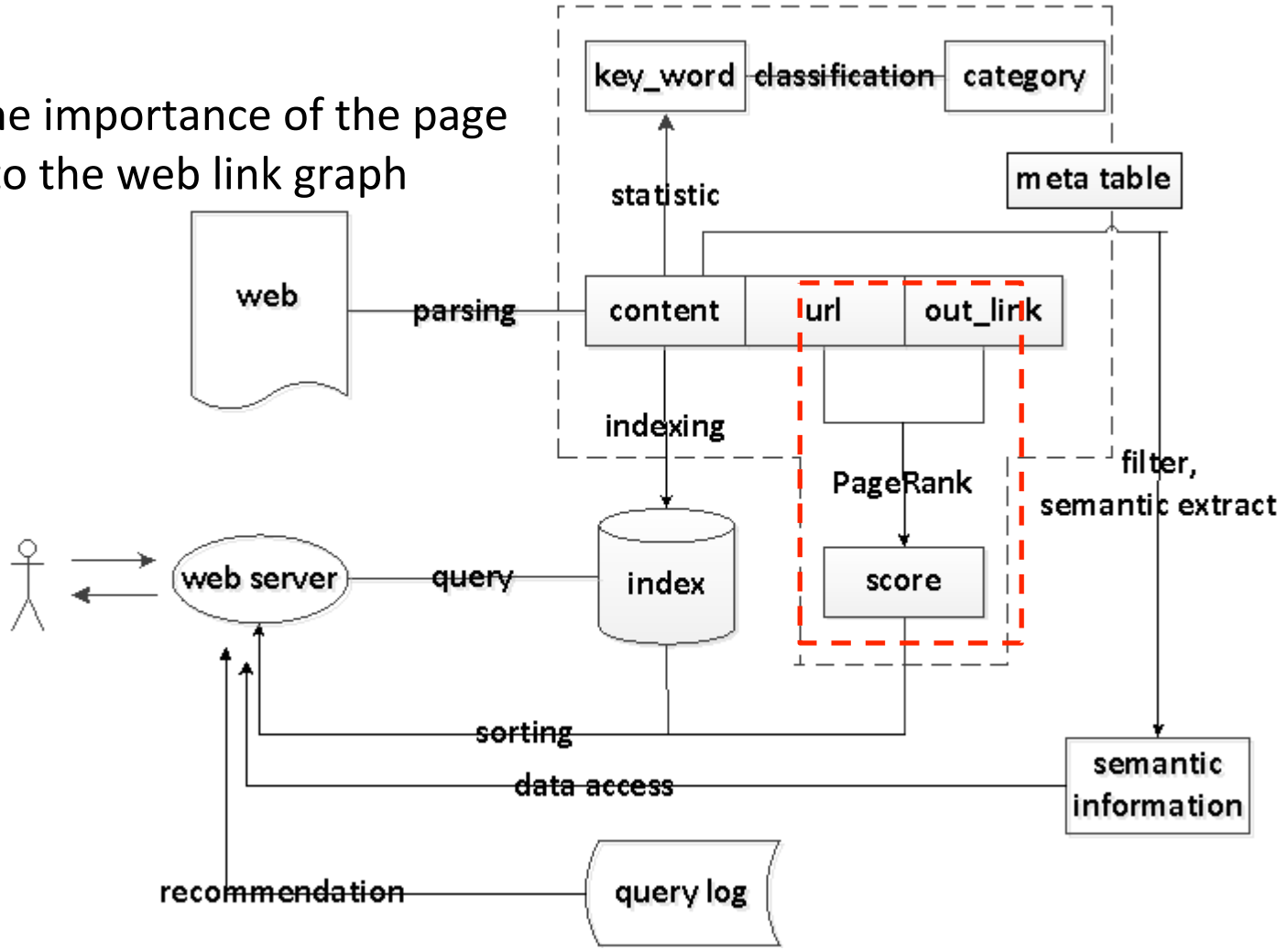
# Search Engine: Indexing

- Indexing
  - create the mapping of terms to document id lists



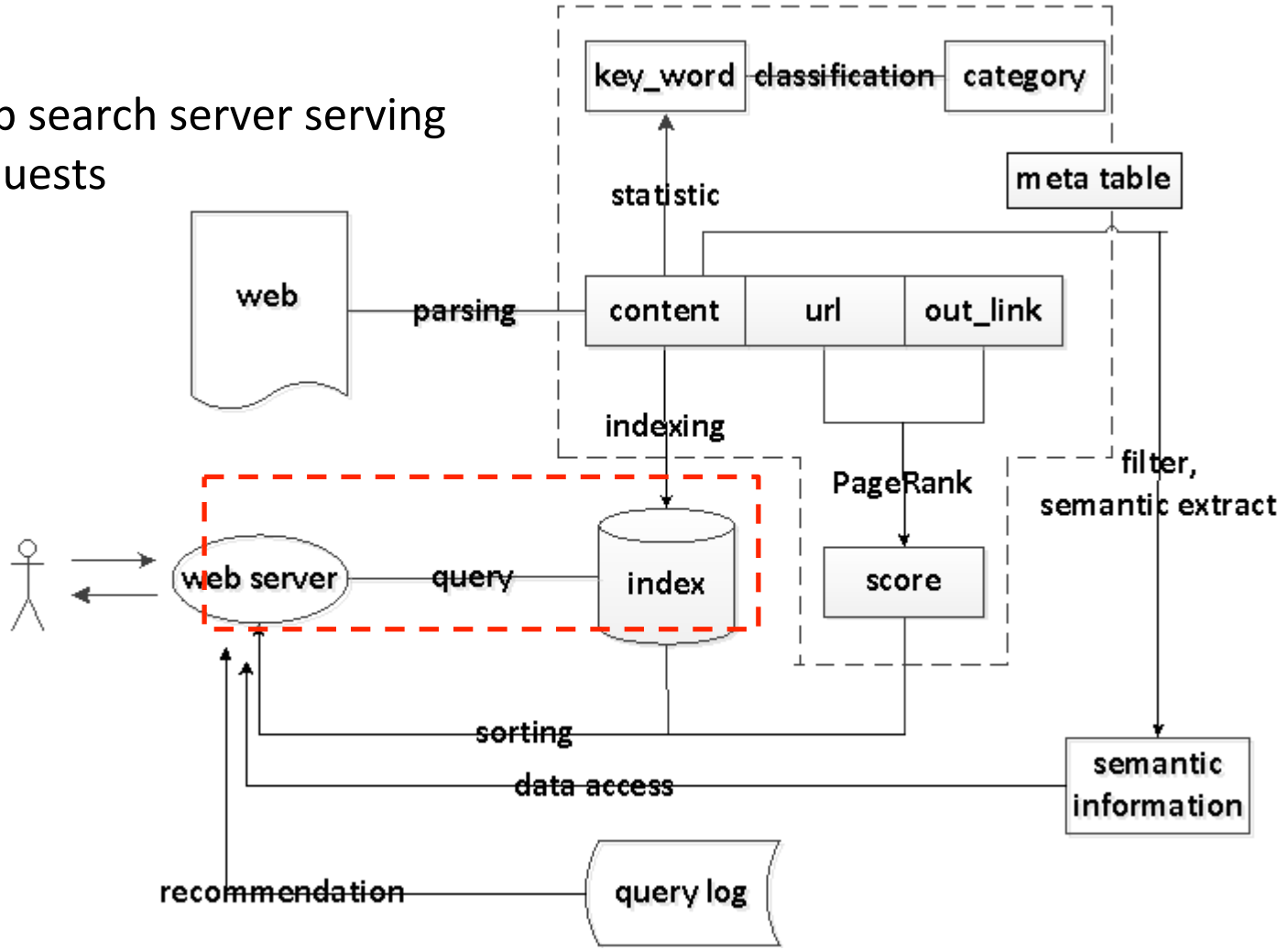
# Search Engine: PageRank

- PageRank
  - Compute the importance of the page according to the web link graph



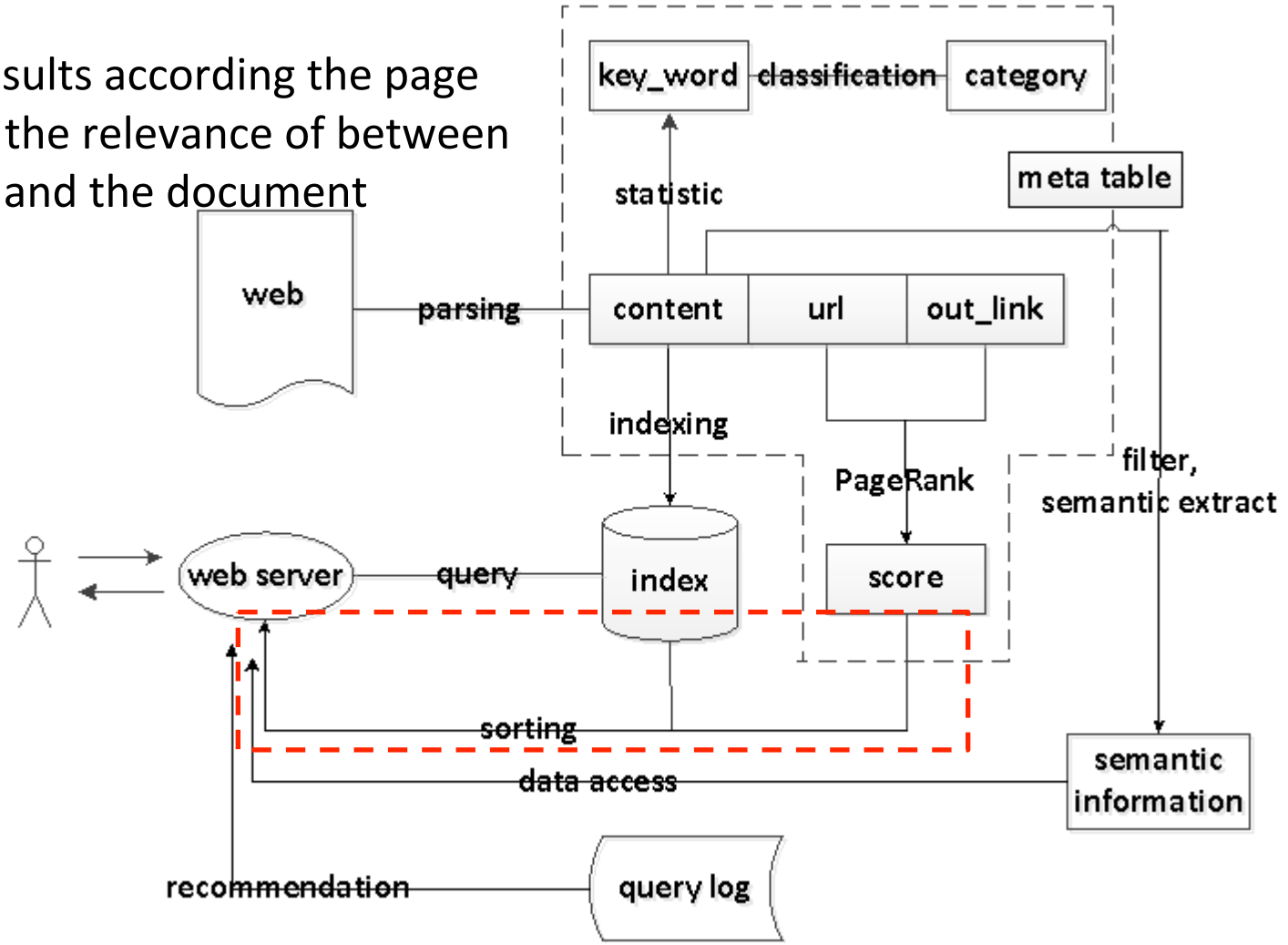
# Search Engine: Search query

- Querying
  - online web search server serving users' requests



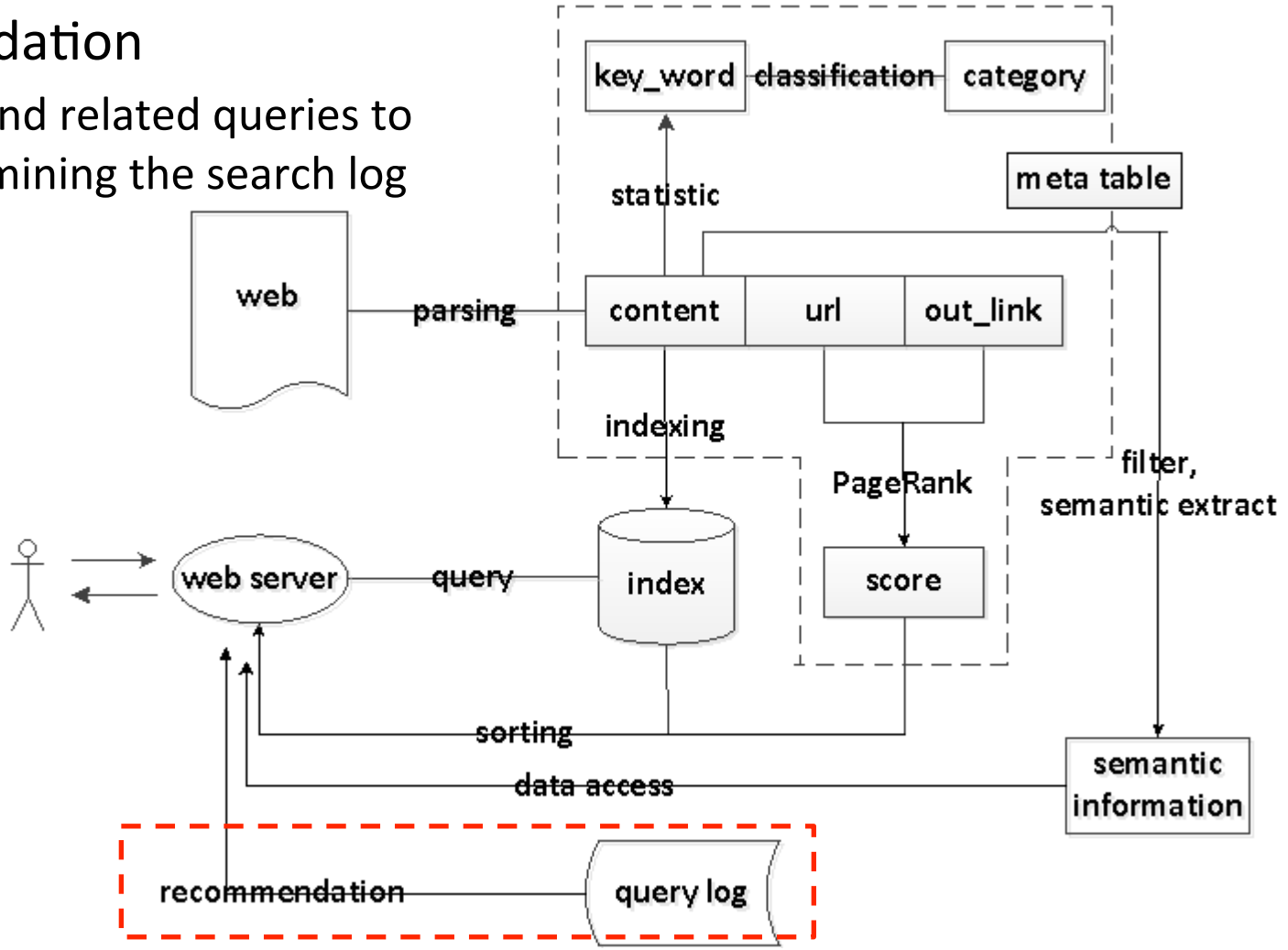
# Search Engine: Sorting

- Sorting
  - Sort the results according the page ranks and the relevance of between the query and the document



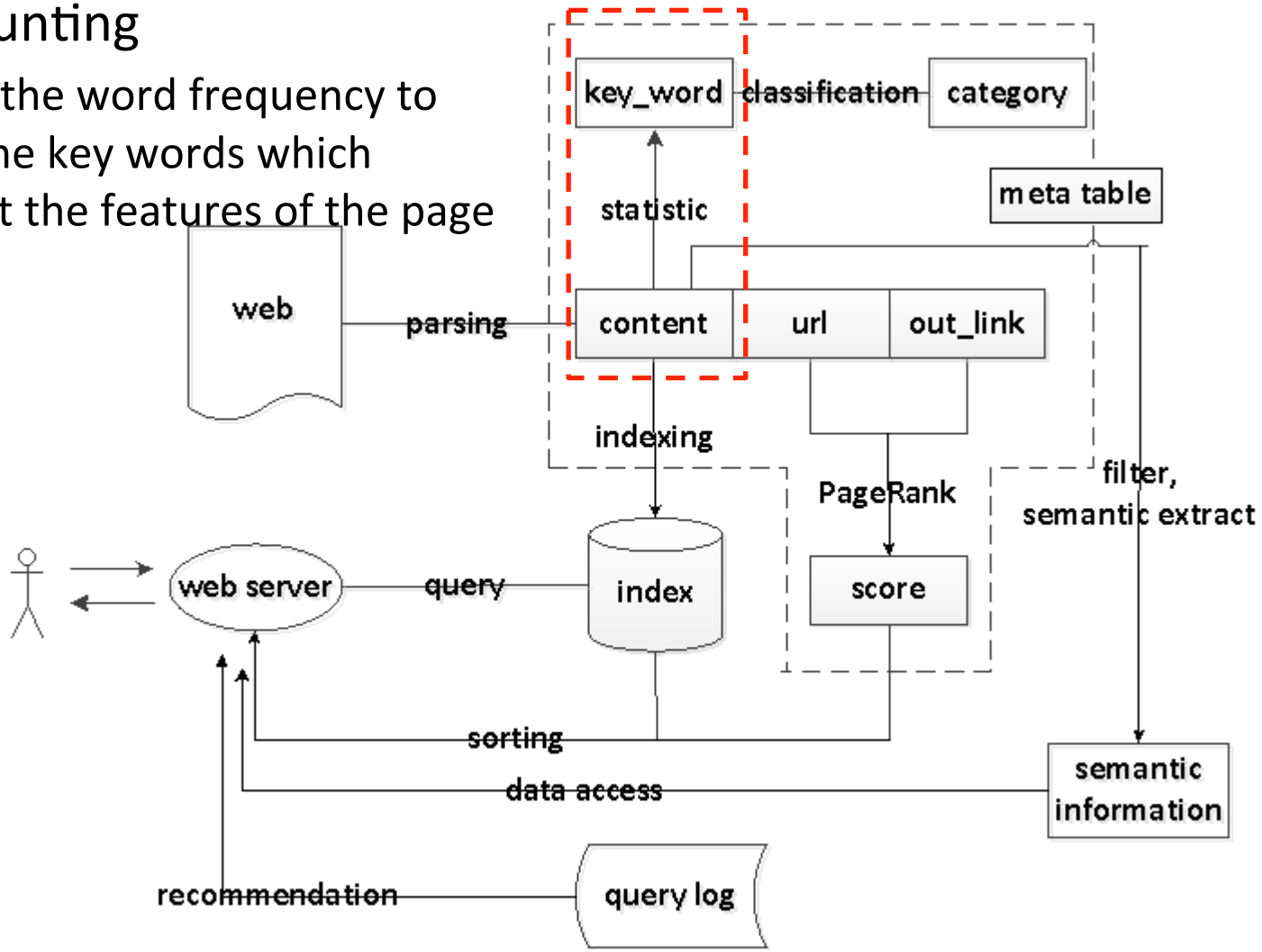
# Search Engine: Recommendation

- Recommendation
  - Recommend related queries to users by mining the search log



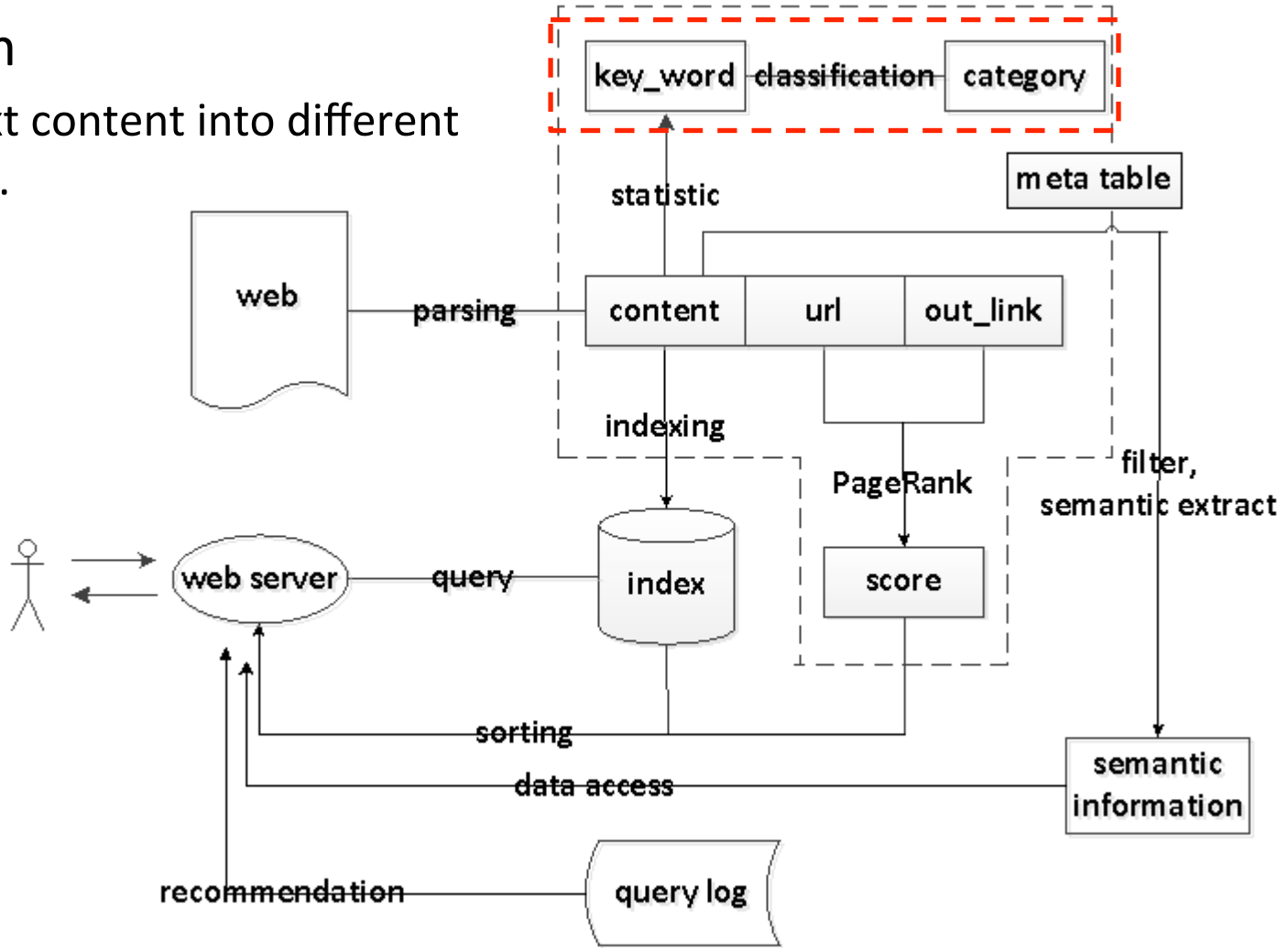
# Search Engine: Statistic counting

- Statistic counting
  - Counting the word frequency to extract the key words which represent the features of the page



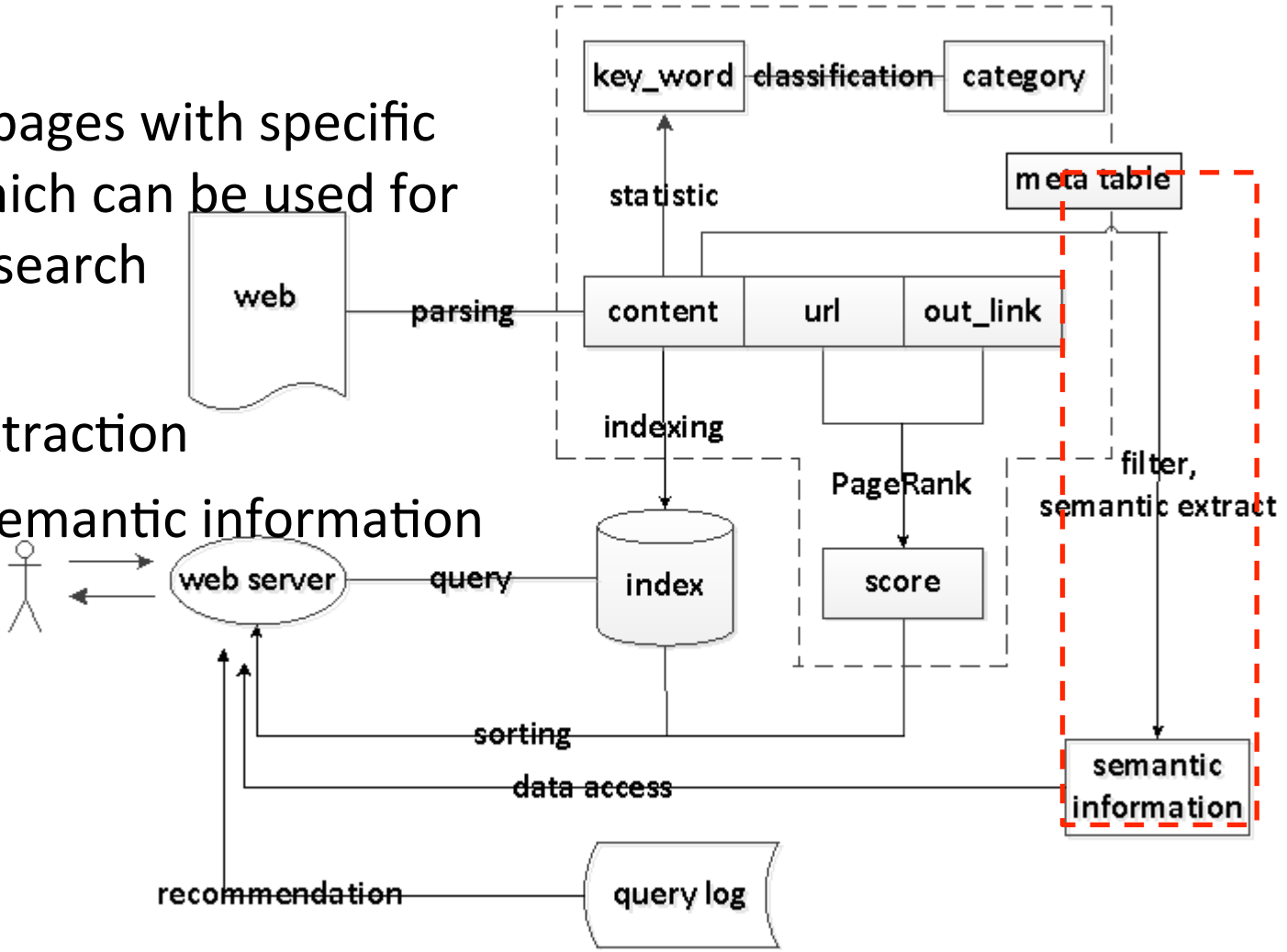
# Search Engine: Classification

- Classification
  - Classify text content into different categories.

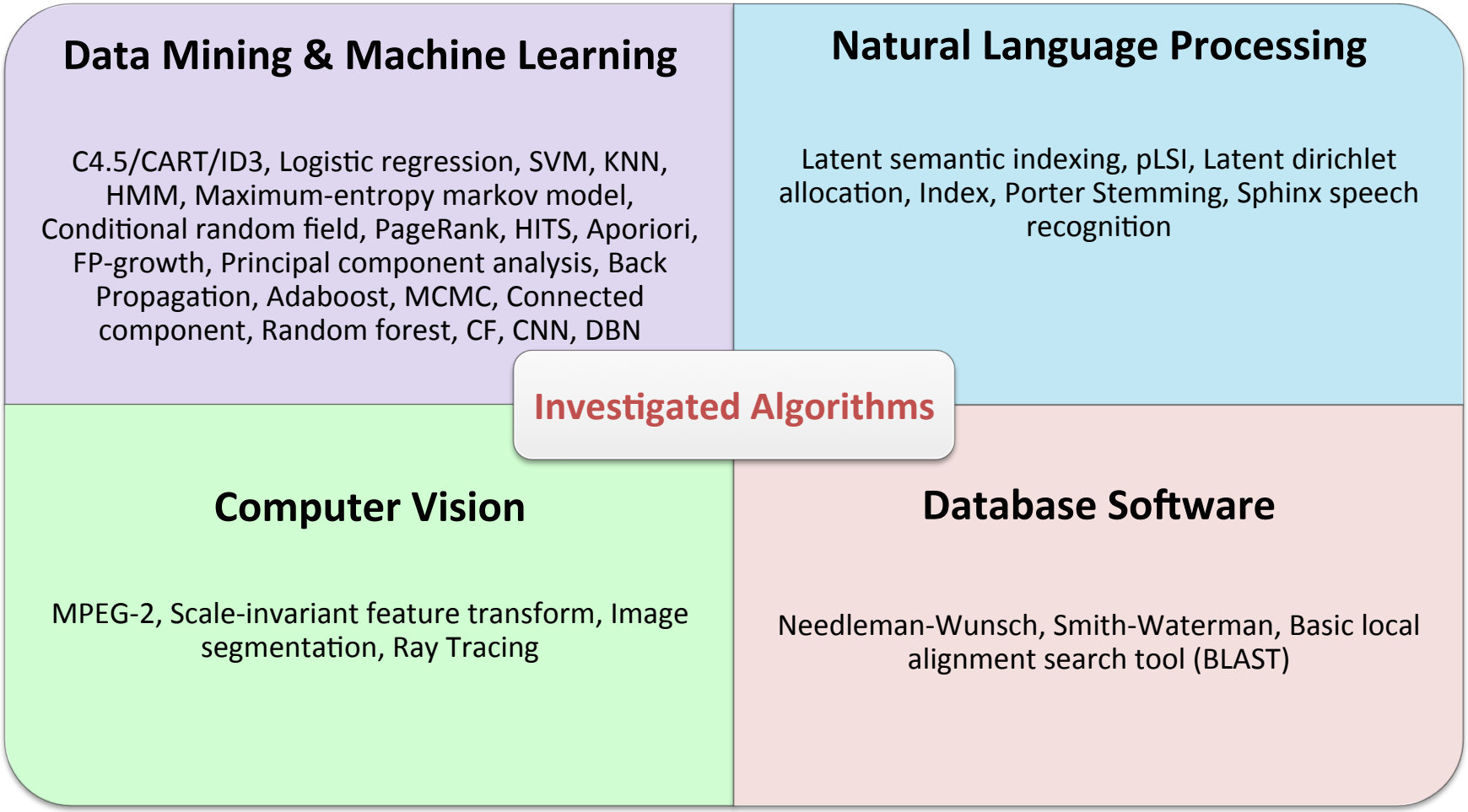


# Search Engine: Filter & Semantic extraction

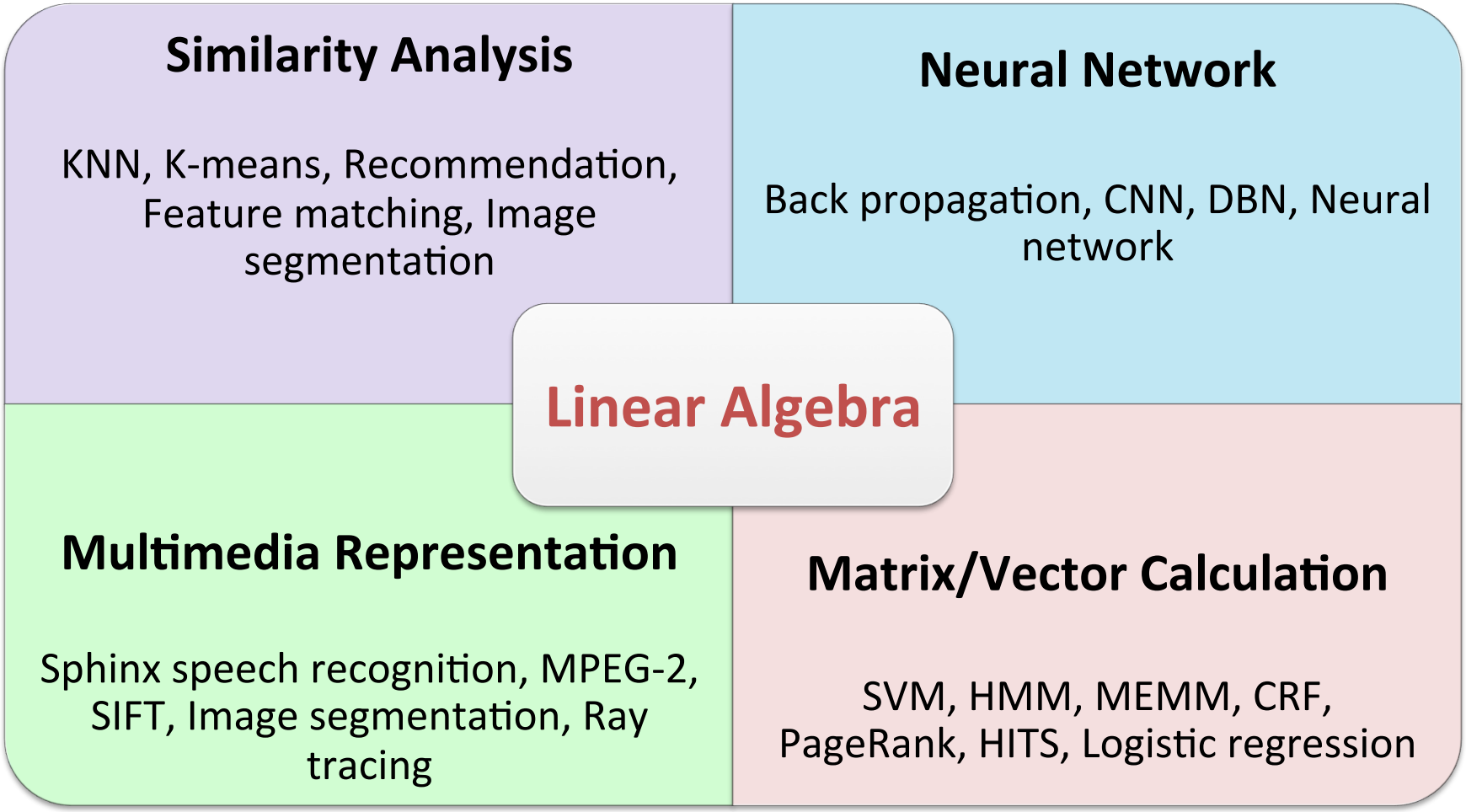
- Filter
  - Identify pages with specific topic which can be used for vertical search
- Semantic extraction
  - extract semantic information



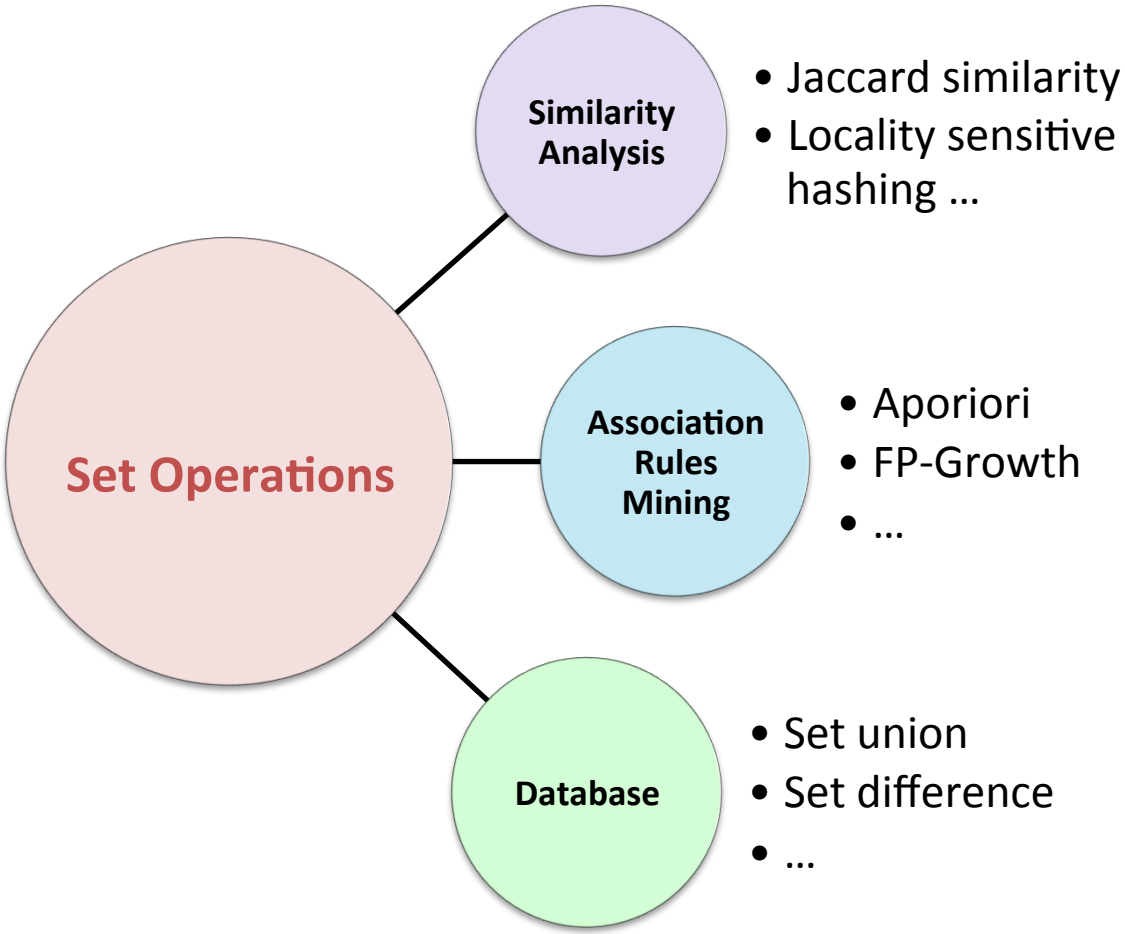
# Investigated Algorithms



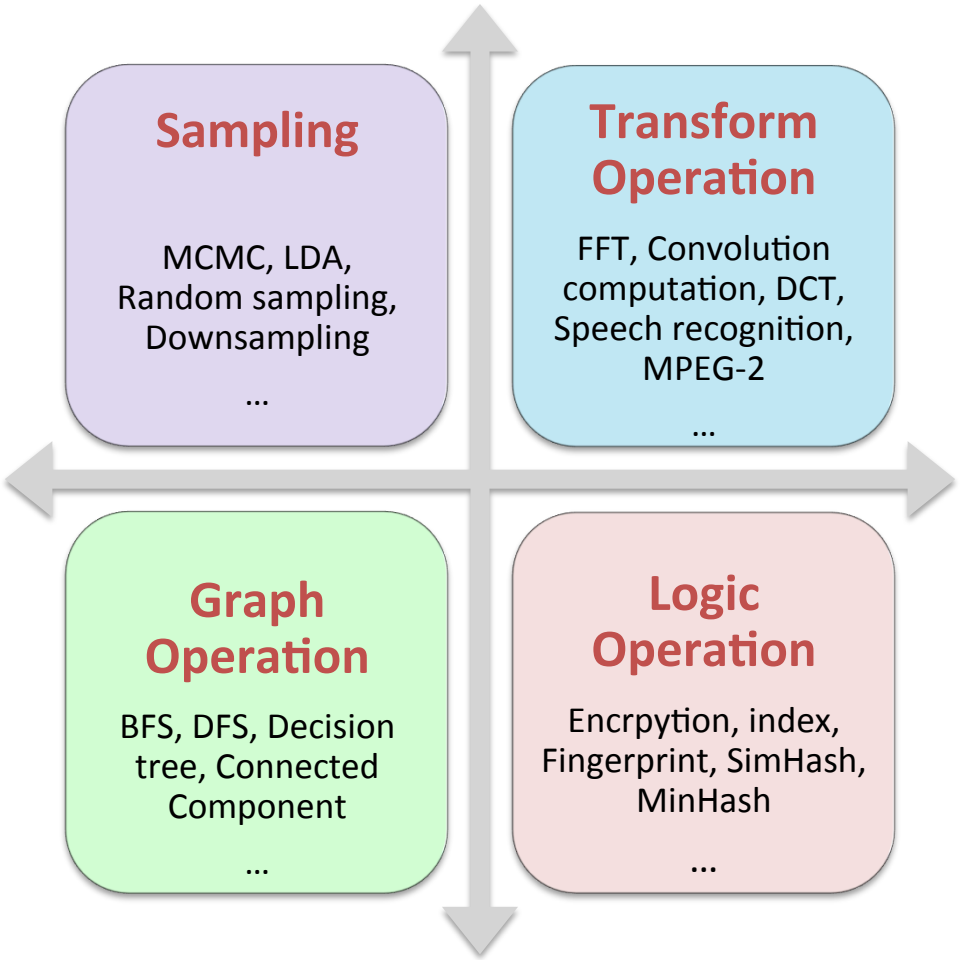
# Frequently-appearing Operations



# Frequently-appearing Operations(cont')



# Frequently-appearing Operations(cont')



# Frequently-appearing Operations(cont')

- Two more primitive operations

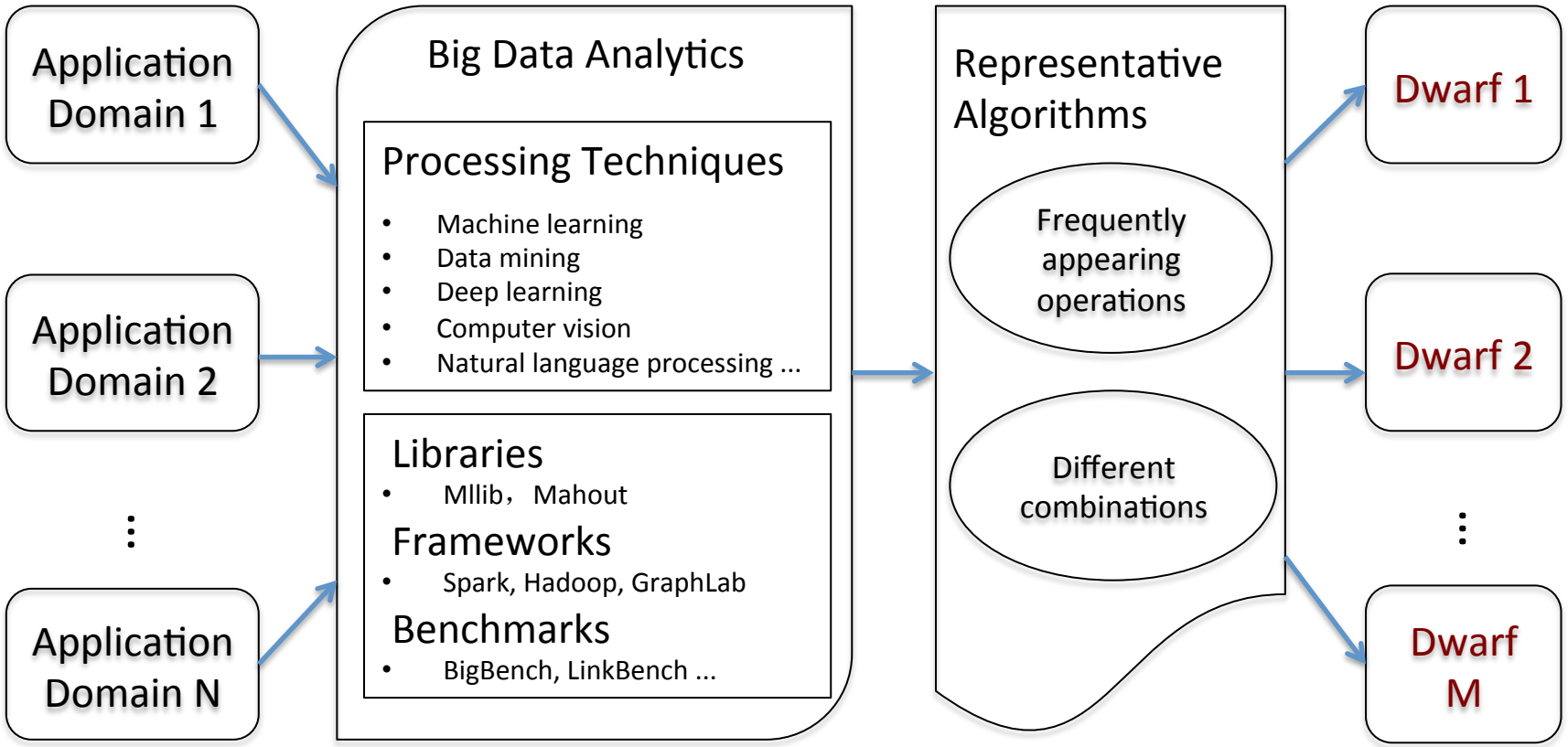
**Statistic Operation**

- Probability calculation
- LSI, pLSI, Latent dirichlet allocation …

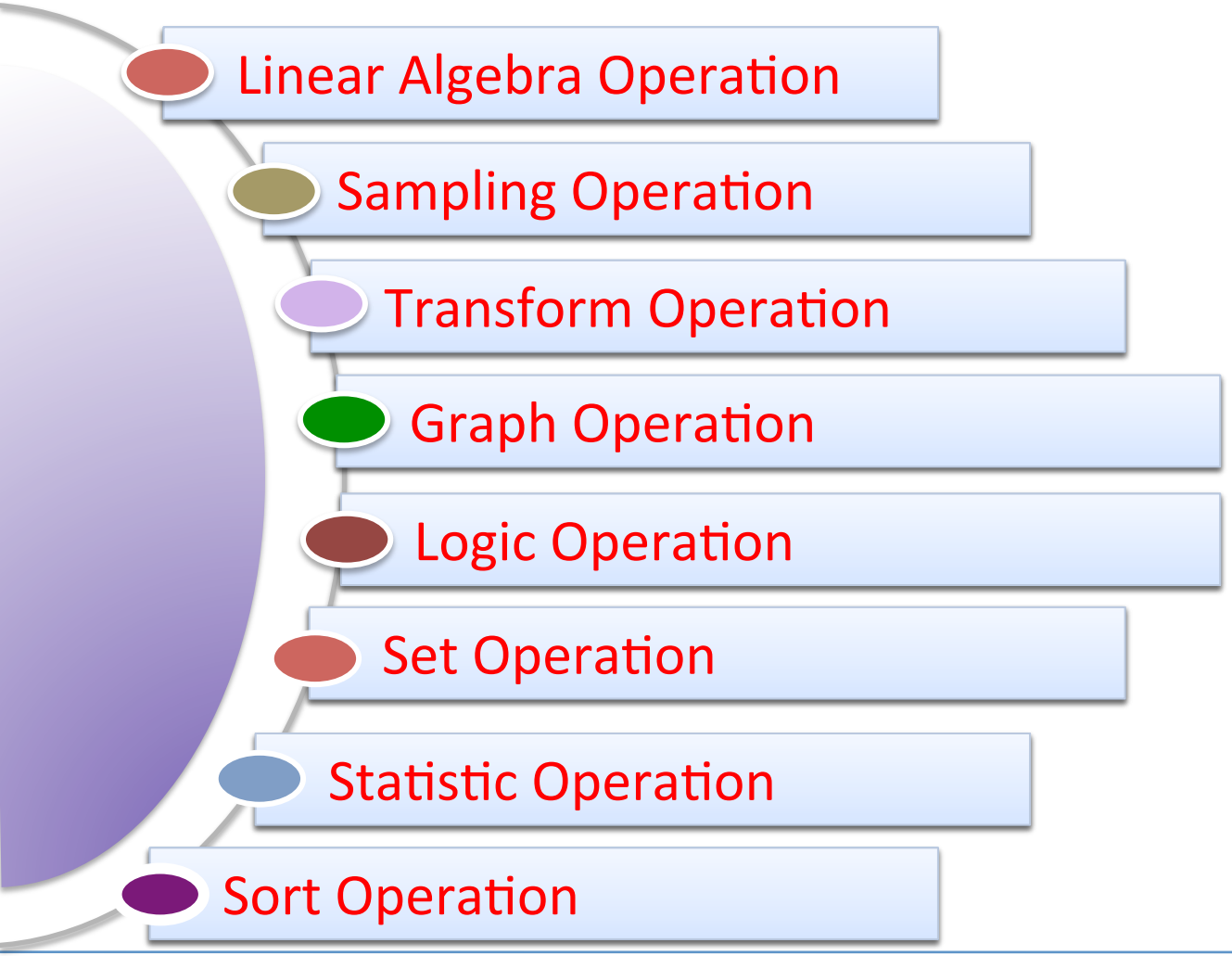
**Sort**

- Partial sort, quick sort, Top k sort…
- K-means, Decision tree …

# Methodology of Dwarfs



# Dwarfs in Big Data Offline Analytics



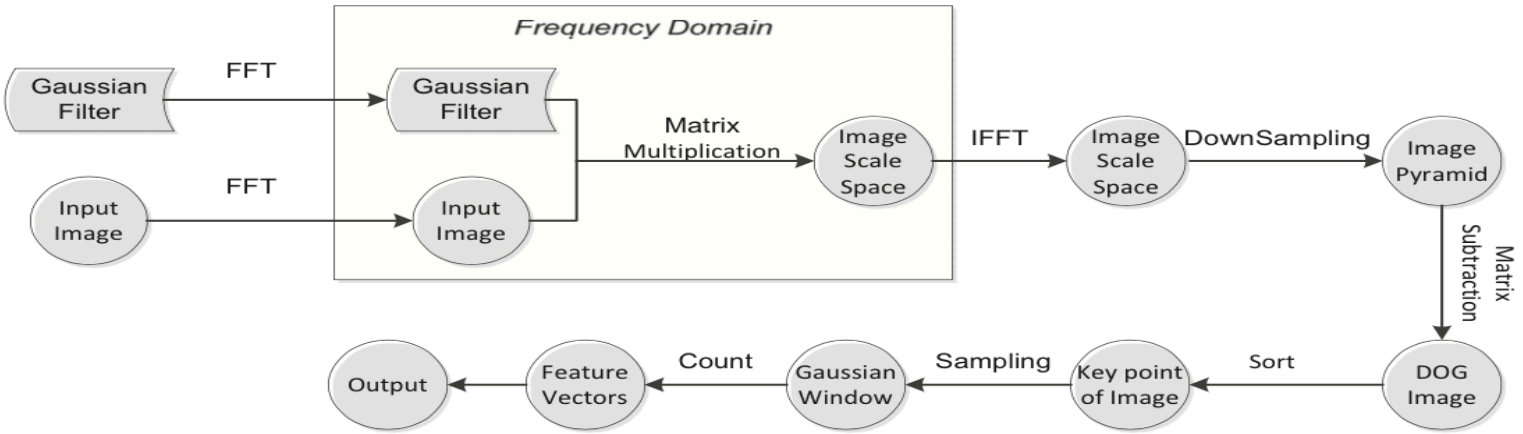
# Dwarf Summary

| Dwarf | Scenario | Algorithm |
|---|---|---|
| Linear Algebra Operations | Classification and recommendation in internet services;<br>Multimedia feature extraction;<br>Neural network;<br>Similarity analysis | Support vector machine (SVM), Hidden markov model (HMM), Conditional random field (CRF), Kmeans, Collaborative filtering (CF), Convolution neural network (CNN), Image Segmentation (GrabCut) |
| Sampling Operations | sample web search engine query results;<br>Mine sentiments from online reviews;<br>Detect sequence patterns | Markov chain monte carlo (MCMC), Latent dirichlet allocation (LDA), Deep belief network (DBN), Scale-invariant feature transform (SIFT) |
| Logic Operations | Inverted indexing;<br>Hashtag recommendation;<br>Digital signature;<br>Find similar images | Encrpytion, Index, SimHash, MinHash, Fingerprint, Locality-sensitive hashing (LSH) |
| Transform Operations | Social Link Prediction;<br>Protein–protein docking;<br>Multimedia signal processing | Sphinx speech recognition, MPEG-2, Convolution, Scale-invariant feature transform (SIFT) |
| Set Operations | Extract semantic information;<br>Data warehouse processing;<br>Image union and reflection | Association rules mining (Aporiori, FP-Growth), Grep, Ray tracing, Jaccard index, Data warehouse (Project, Filter) |
| Graph Operations | Web link graph computing;<br>Community detection;<br>Multimedia indexing | Decision tree (C4.5/CART/ID3), Connected component, Random forest, Breadth first search (BFS) |
| Sort Operations | Rank websites results;<br>Active user ranking;<br>Compare image pixel | k-nearest neighbors (KNN), Kmeans, PageRank, TeraSort |
| Statistic Operations | Count the word frequency;<br>Probability statistics;<br>Model statistical structure | Naïve bayes, WordCount, Latent semantic indexing (LSI), Term frequency–inverse document frequency (TF-IDF) |

# Dwarf Combination

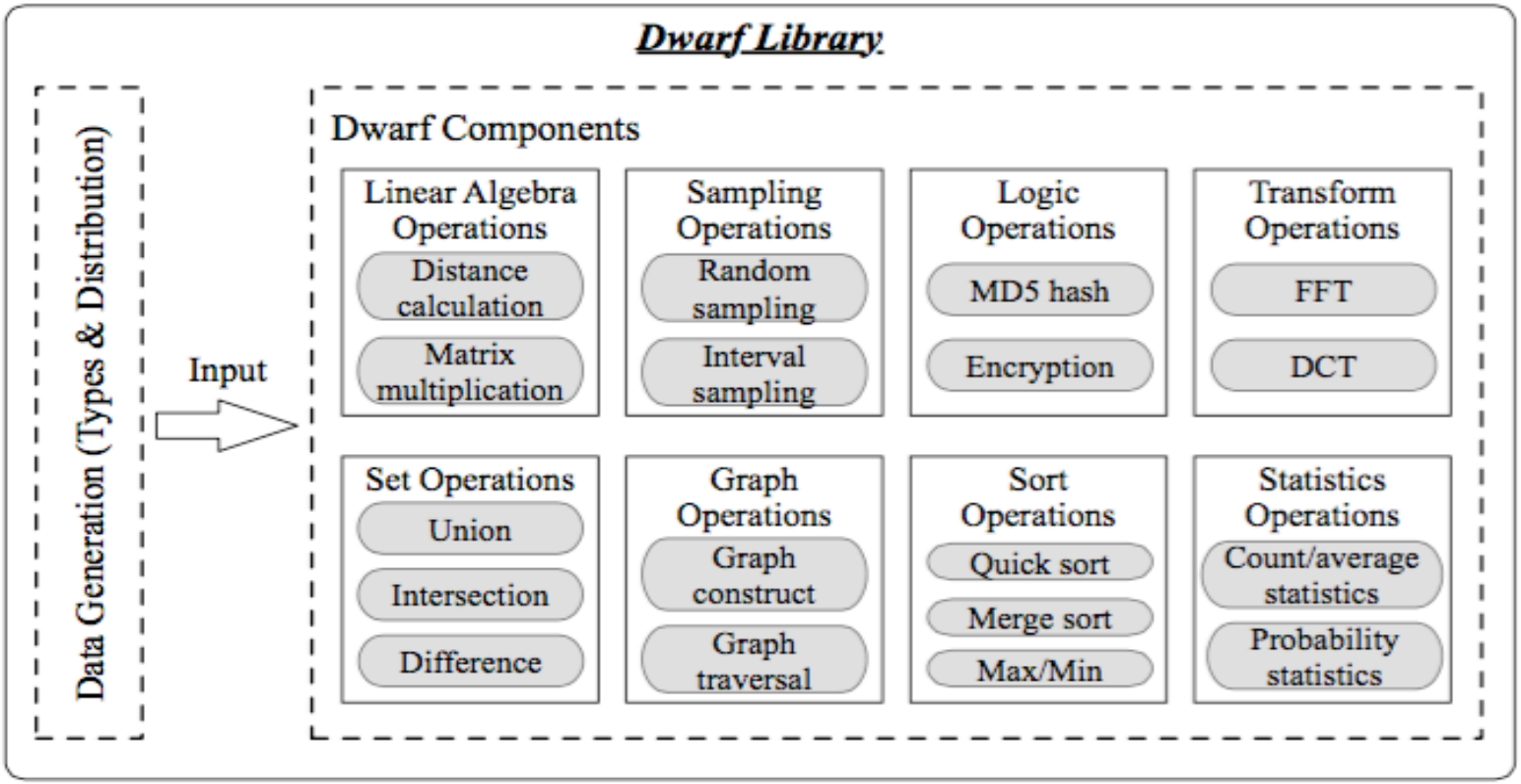
- DAG-like graph of one or more dwarfs


*Feature extraction – SIFT Algorithm*



# Dwarf Library

- Data generator & Data component



# Outline

- BigDataBench - a big data benchmark suite
- Big Data Dwarfs & Dwarf Library
- Dwarf-based Simulation Benchmarks
- Evaluation on X86 Processor
- Case Study on ARM Processor

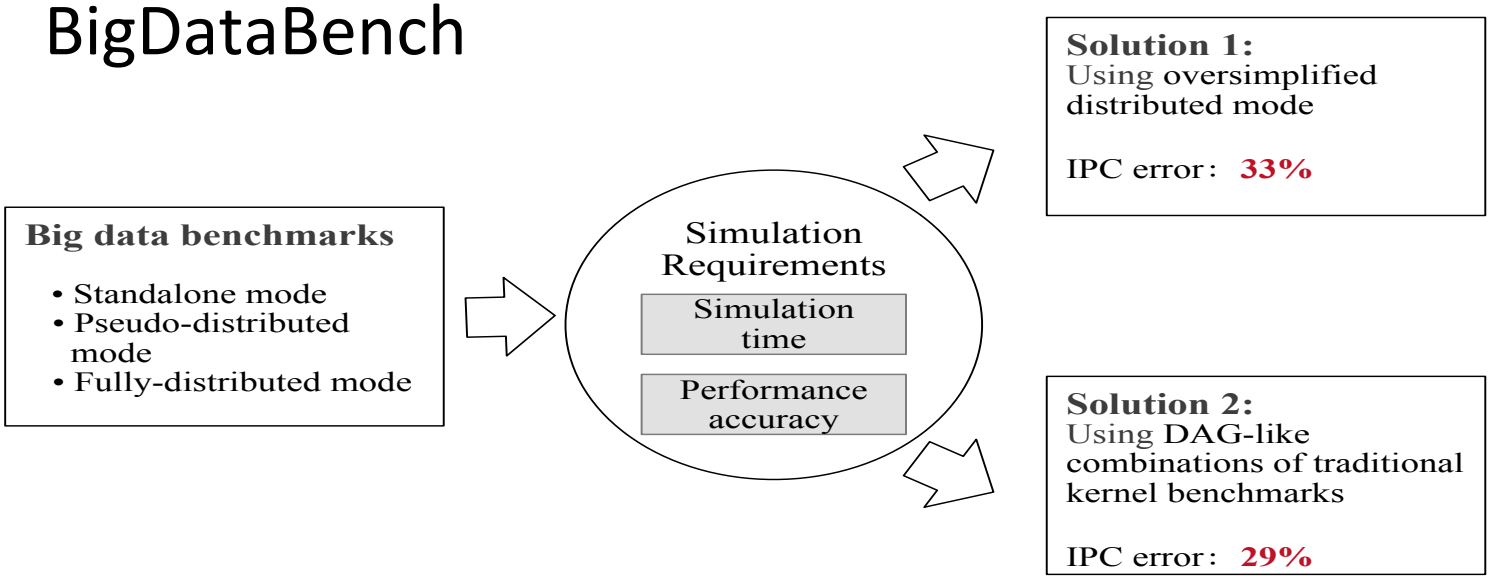
# Motivation

- Traditional simulation method
  - A case-by-case simulation
    - Sampled trace or synthetic trace
  - Focus on specific workload and architecture

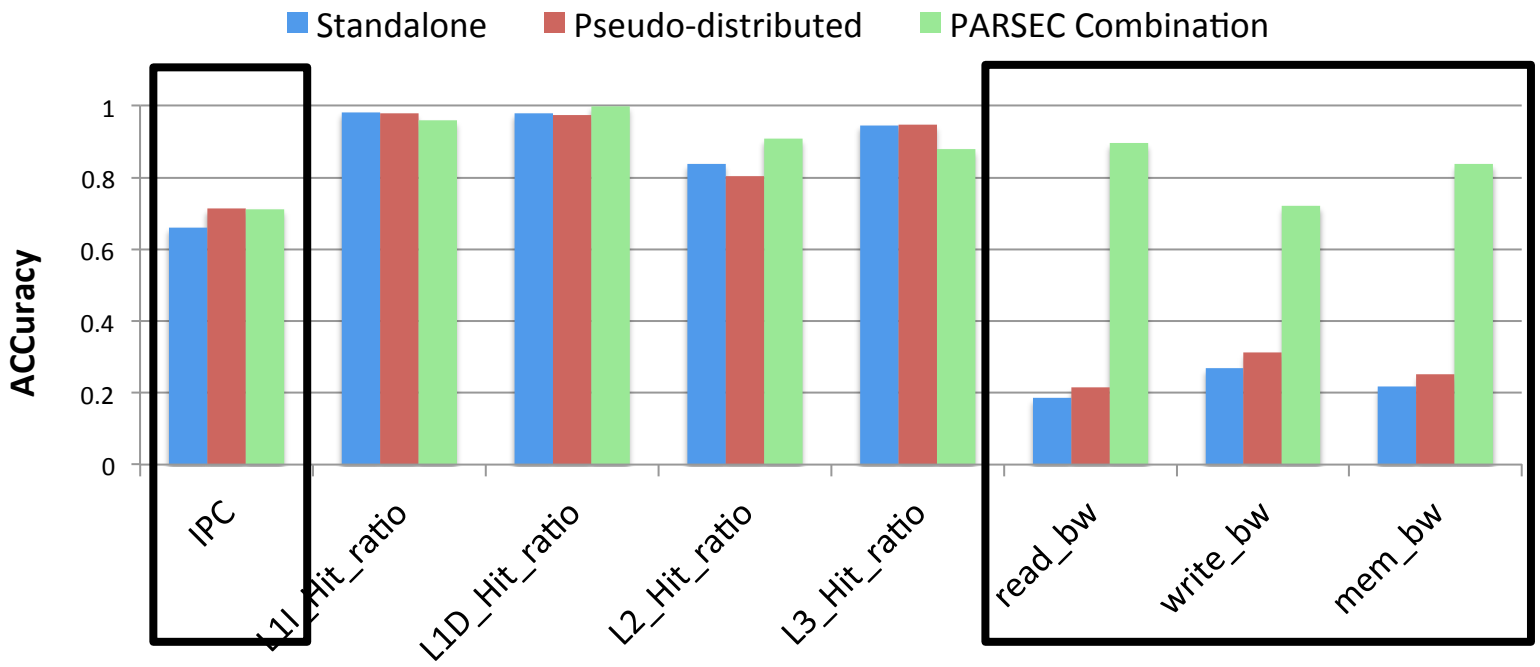
***Not suit for big data scenario***

# Motivation (Cont')

- Two previous simulation solutions for big data analytics workloads
  - Using a fully-distributed mode Sort workload in BigDataBench

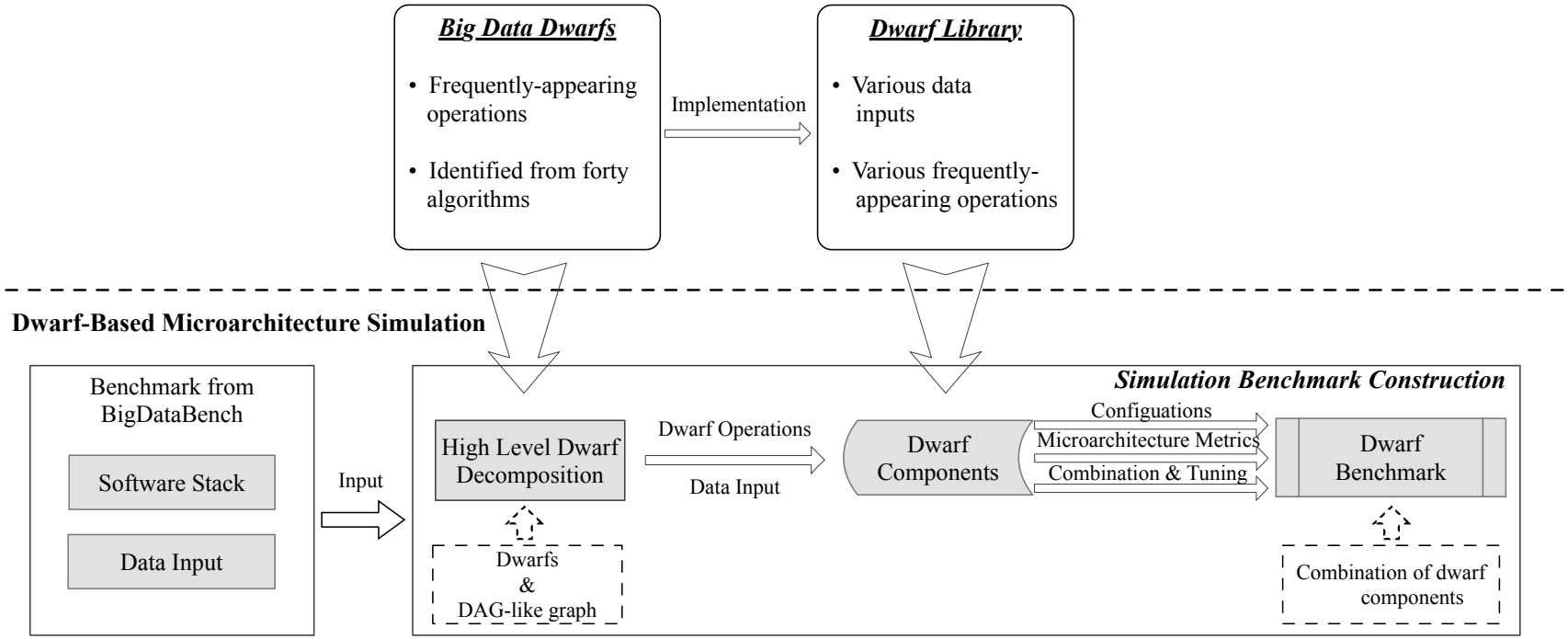


# Data Accuracy of Previous Solution



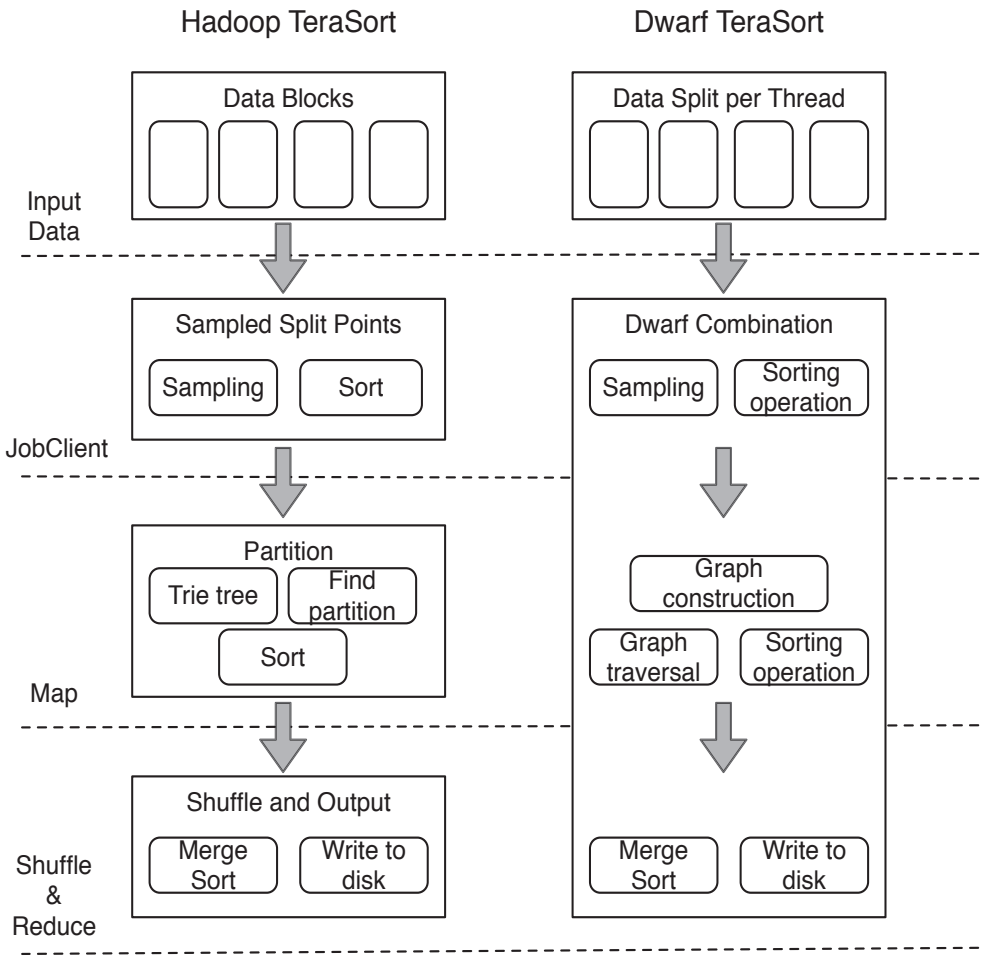

- *read_bw: read memory bandwidth*
- *write_bw: write memory bandwidth*
- *mem_bw: all memory bandwidth*

# Dwarf-based Simulation Methodology



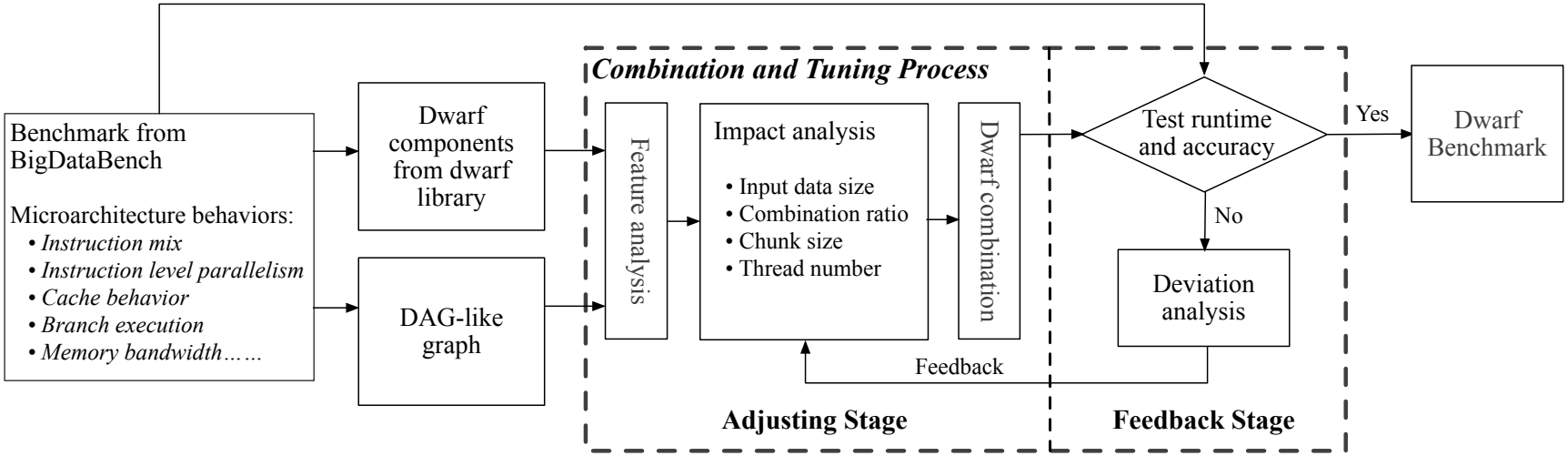
# How to Map a Benchmark

- Hadoop TeraSort
  - Sampling
  - Trie tree
- Dwarf TeraSort
  - Dwarf component
    - Operations combination



# Dwarf Combination & Tuning

- Adjusting stage
  - Generate a dwarf combination
- Feedback stage
  - Testing, analyzing and feedback



# Dwarf Benchmarks

- Three representative big data analytics workloads

| BigDataBench Benchmark | Data Set | Involved Dwarfs | Dwarf Components in Dwarf Library | Dwarf Benchmark Implementation |
|---|---|---|---|---|
| Hadoop TeraSort | Records | Sort operations<br>Sampling operations<br>Graph operations | Quick sort; Merge sort<br>Random sampling; Interval sampling<br>Graph construction; Graph traversal | Dwarf Components; OpenMP Framework; Combination and Tuning |
| Hadoop Kmeans | Vectors | Linear algebra operations<br>Sort operations<br>Statistic operations | Vector euclidean distance; Cosine distance<br>Quick sort; Merge sort<br>Cluster count; Average computation | |
| Hadoop PageRank | Graph | Linear algebra operations<br>Sort operations<br>Statistic operations | Matrix construction; Matrix multiplication<br>Quick sort; Min/max calculation<br>Out degree and in degree count of nodes | |

# Outline

- BigDataBench - a big data benchmark suite
- Big Data Dwarfs & Dwarf Library
- Dwarf-based Simulation Benchmarks
- Evaluation on X86 Processor
- Case Study on ARM Processor

# Evaluation on X86 Processor

- Configurations

| Hardware Configurations | | | |
|---|---|---|---|
| CPU Type | | Intel CPU Core | |
| Intel ®Xeon E5645 | | 6 cores@2.40G | |
| L1 DCache | L1 ICache | L2 Cache | L3 Cache |
| 6 × 32 KB | 6 × 32 KB | 6 × 256 KB | 12MB |
| Memory | Ethernet | Hyper-Threading | |
| 32GB,DDR3 | 1Gb | Disabled | |
| **Software Configurations** | | | |
| Operating System | Linux Kernel | JDK Version | Hadoop Version |
| CentOS 6.4 | 3.11.10 | 1.7.0 | 2.7.1 |

# Experiment Setup

- One-master-four-slave cluster
- Hadoop Workloads
  - Fully-distributed workloads from BigDataBench
    - TeraSort: 100GB records
    - Kmeans: 100GB dense vector data
    - PageRank: 2^26-vertex graph data

# Metrics

- Accuracy
  - Micro-architectural metrics
    - Processor performance
    - Cache behavior
    - Instruction mix
    - Branch prediction
    - Memory bandwidth

| Category | Metric Name | Description |
|---|---|---|
| Processor Performance | IPC | Instructions per cycle |
| | MIPS | Million instructions per second |
| Cache Behavior | L1I Hit Ratio | L1 instruction cache hit ratio |
| | L1D Hit Ratio | L1 data cache hit ratio |
| | L2 Hit Ratio | L2 cache hit ratio |
| | L3 Hit Ratio | L3 cache hit ratio |
| Instruction Mix | Instruction ratios | Instruction ratios of floating-point, load, store, branch and integer instructions |
| Branch Prediction | Branch Miss | Branch miss prediction ratio |
| Memory Bandwidth | Memory Read Bandwidth | Data load rate from memory |
| | Memory Write Bandwidth | Data store rate into memory |
| | Memory Bandwidth | Data load/store rate from/into memory |

# Runtime Speedup on Xeon E5645

- Speedup comparing to Hadoop benchmarks
  - 136X for TeraSort
  - 355X for Kmeans
  - 160X for PageRank

| Workloads | Execution Time (Second) | |
|---|---|---|
| | Hadoop version | Dwarf version |
| TeraSort | 1500 | 11.02 |
| Kmeans | 3564 | 10.03 |
| PageRank | 1444 | 9.03 |

# Performance Accuracy

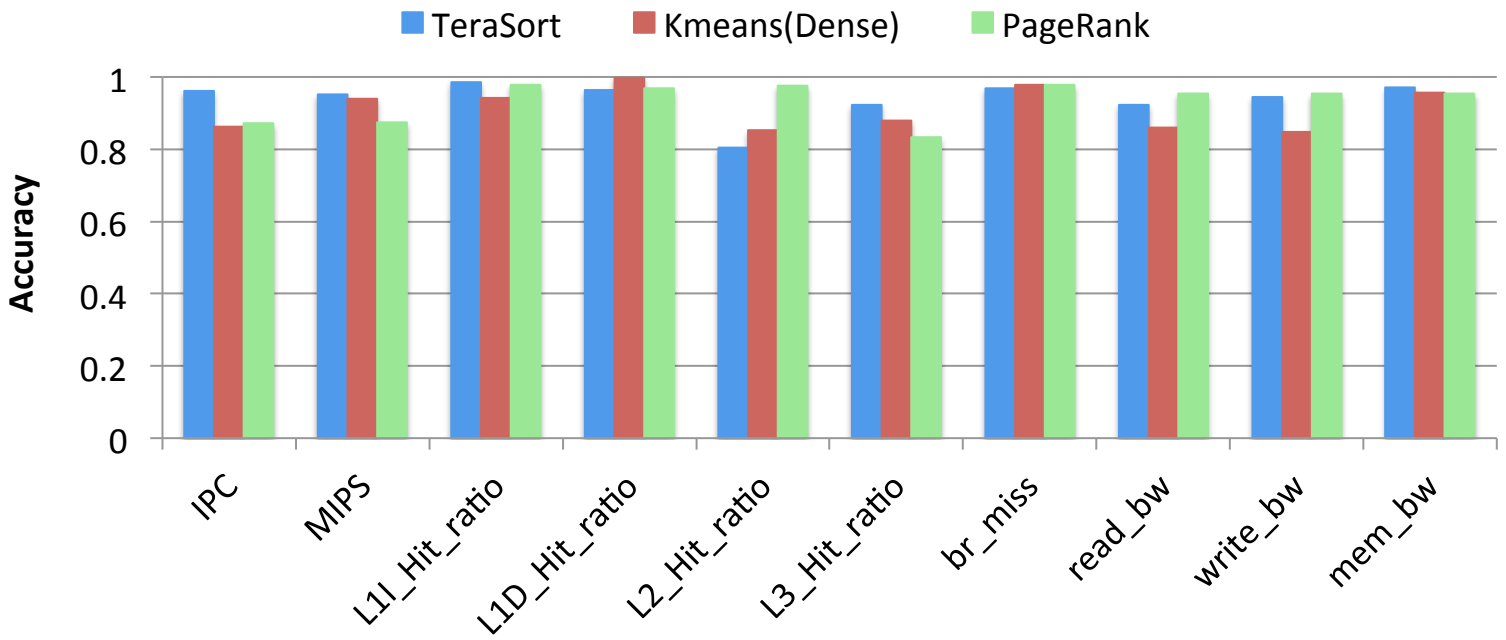
- The accuracy of all selected metrics

$$Accuracy(Val_H, Val_D) = 1 - \left| \frac{Val_D - Val_H}{Val_H} \right|$$

- $Val_H$ – average value of Hadoop benchmark on all slaves
- $Val_D$ – average value of corresponding dwarf benchmark

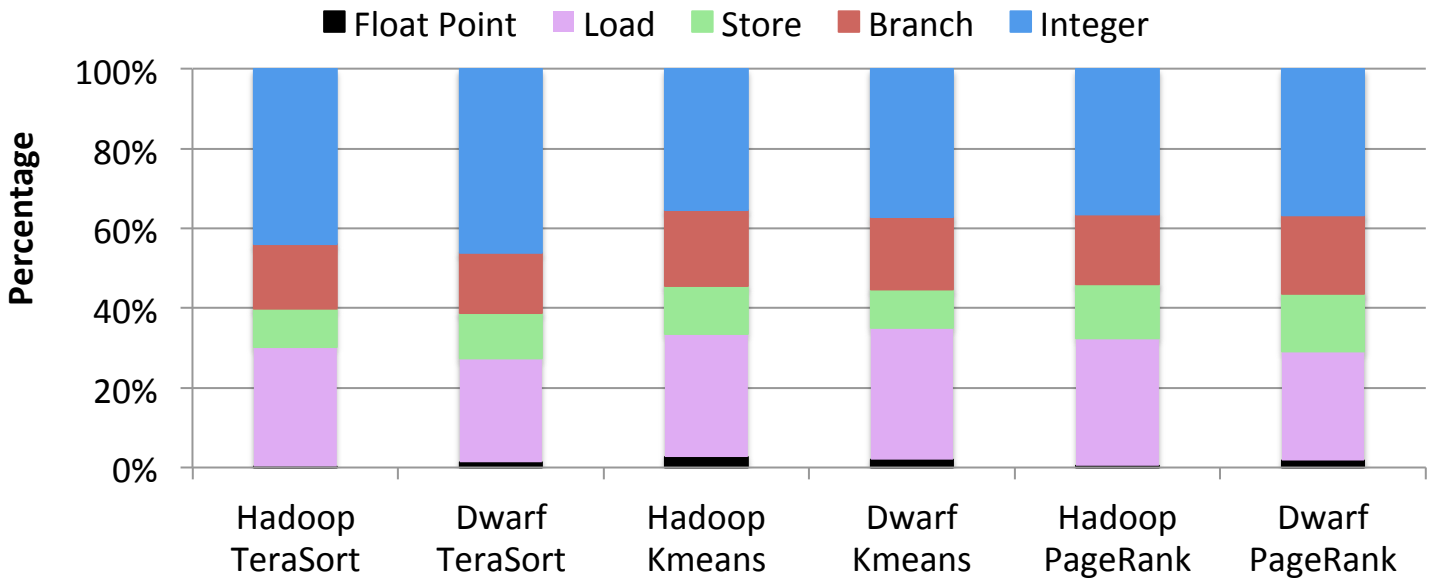
# Micro-architectural Data Accuracy on Xeon E5645

- Average data accuracy
  - 94% for TeraSort, 91% for Kmeans, 93% for PageRank



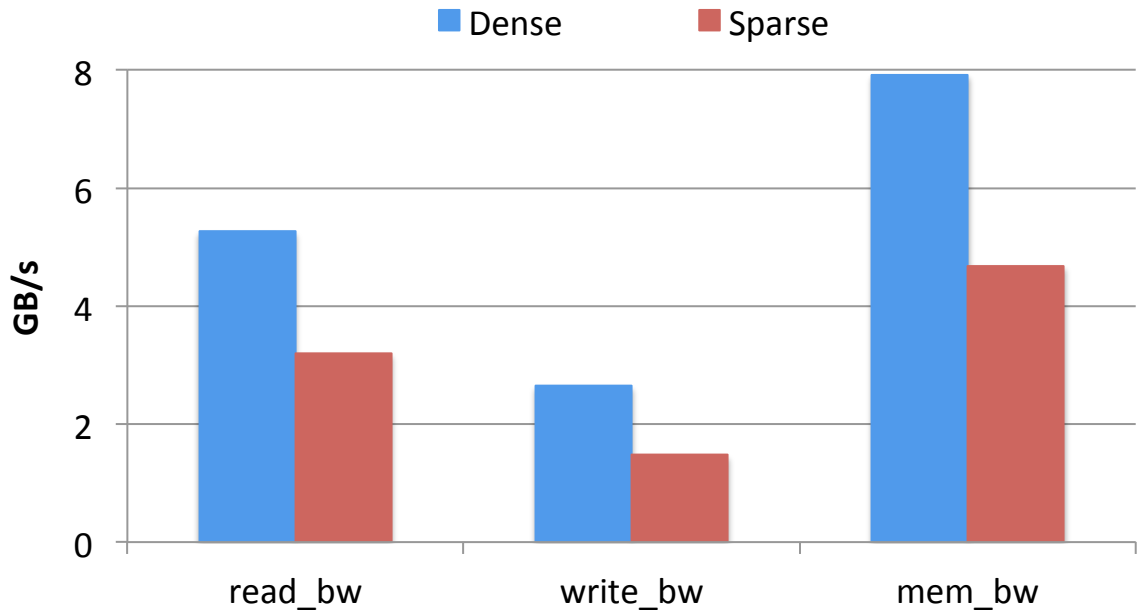
# Instruction Mix Breakdown

- Preserve the instruction mix characteristics
  - less floating-point instructions
  - more integer operations
  - higher demand for data movement than instruction executions



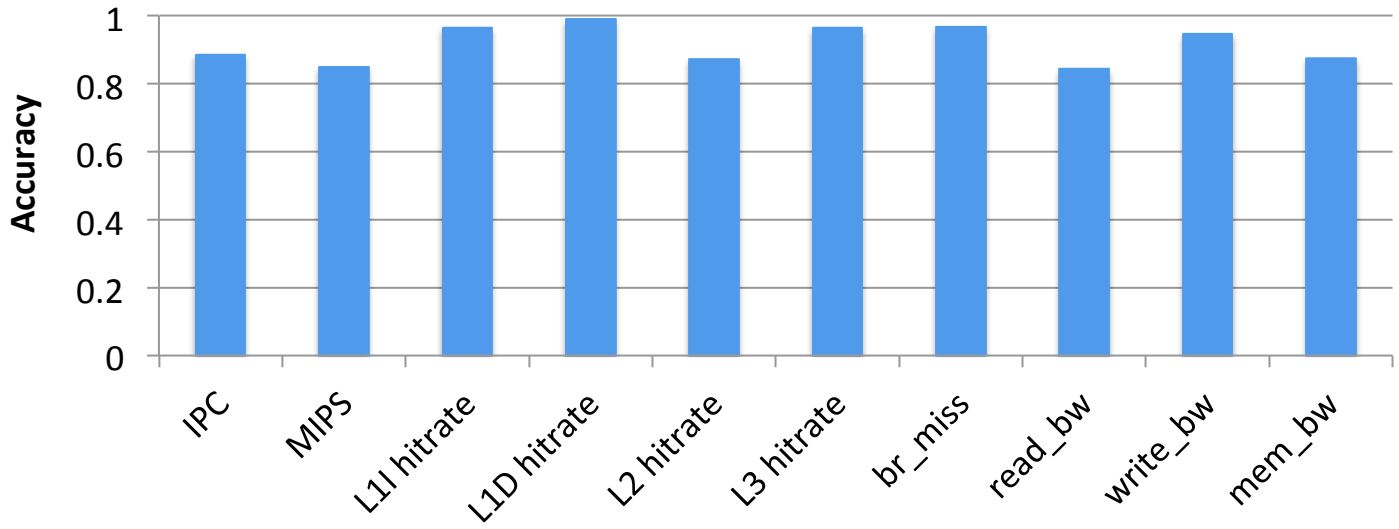
# Impact of Input Data

- Data impact on memory bandwidth for Hadoop Kmeans
  - Dense vector: all elements are none-zero
  - Sparse vector: 90% elements are zero



# Data Accuracy using Sparse data

- Hadoop Kmeans and Dwarf Kmeans
  - Only change the input data sparsity
  - Average data accuracy above 92%



# Outline

- **BigDataBench - a big data benchmark suite**
- **Big Data Dwarfs & Dwarf Library**
- **Dwarf-based Simulation Benchmarks**
- **Evaluation on X86 Processor**
- **Case Study on ARM Processor**

# Case Study on ARMv8 Processor

- Runtime speedup comparing to Hadoop benchmarks
- Performance accuracy
- Multi-core scalability
- Runtime speedup across different processors
  - ARMv8 and Xeon E5-2690 V3 (Haswell)

# Configurations

- ARMv8 and Haswell processors

| Hardware Configurations | | |
|---|---|---|
| Model | ARMv8 | Xeon E5-2690 V3 |
| Number of Processors | 1 | 1 |
| Number of Cores | 32 | 12 |
| Frequency | 2.1GHz | 2.6GHz |
| L1 Cache(I/D) | 48KB/32KB | 32KB/32KB |
| L2 Cache | 8 x 1024KB | 12 x 256KB |
| L3 Cache | 32MB | 30MB |
| Architecture | ARM | X86_64 |
| Memory | 64GB, DDR4 | 64GB, DDR4 |
| Ethernet | 1Gb | 1Gb |
| Hyper-Threading | None | Enabled |
| **Software Configurations** | | |
| Operating System | EulerOS V2.0 | Red-hat Enterprise Linux Server release 7.0 |
| Linux Kernel | 4.1.23-vhulk3.6.3.aarch64 | 3.10.0-123.e17.x86-64 |
| GCC Version | 4.9.3 | 4.8.2 |
| JDK Version | jdk1.8.0_101 | jdk1.7.0_79 |
| Hadoop Version | 2.5.2 | 2.5.2 |

# Experiment Setup

- One-master-one-slave cluster
- Hadoop Workloads
  - Fully-distributed workloads from BigDataBench
    - TeraSort: 50GB records
    - Kmeans: 50GB dense vector data
    - PageRank: 2^24-vertex graph data

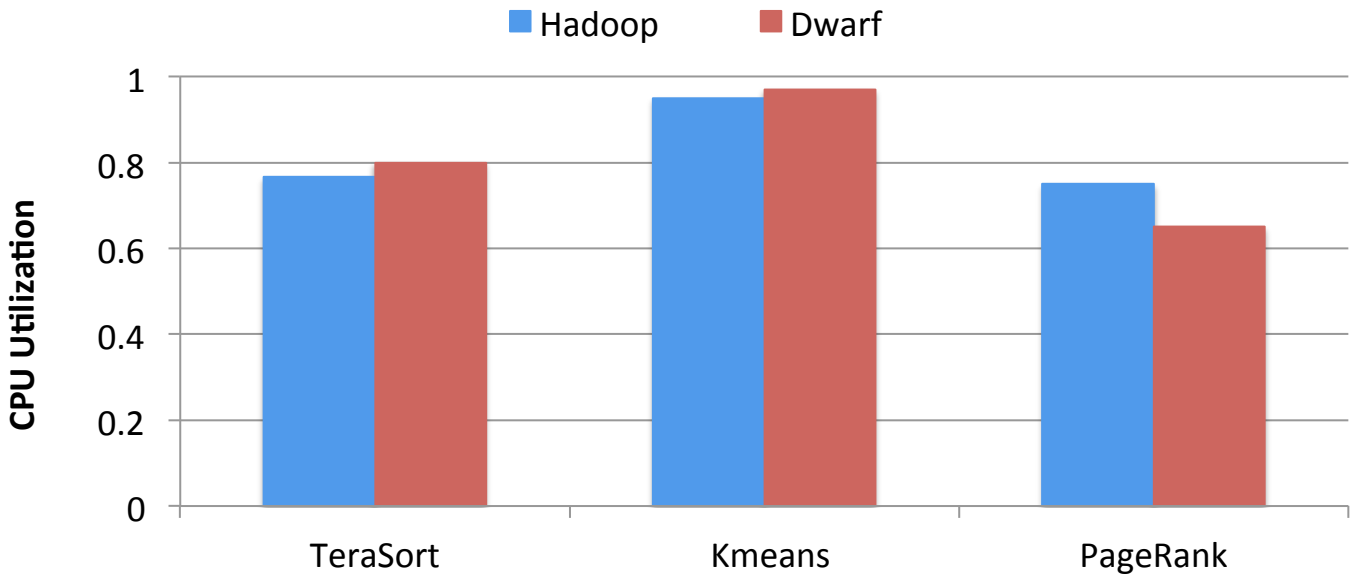
# Runtime Speedup on ARMv8

- Speedup comparing to Hadoop benchmarks
  - 336X for TeraSort
  - 386X for Kmeans
  - 690X for PageRank

| Workloads | Execution Time (Second) | |
|---|---|---|
| | Hadoop version | Dwarf version |
| TeraSort | 1378 | 4.10 |
| Kmeans | 3347 | 8.68 |
| PageRank | 4291 | 6.22 |

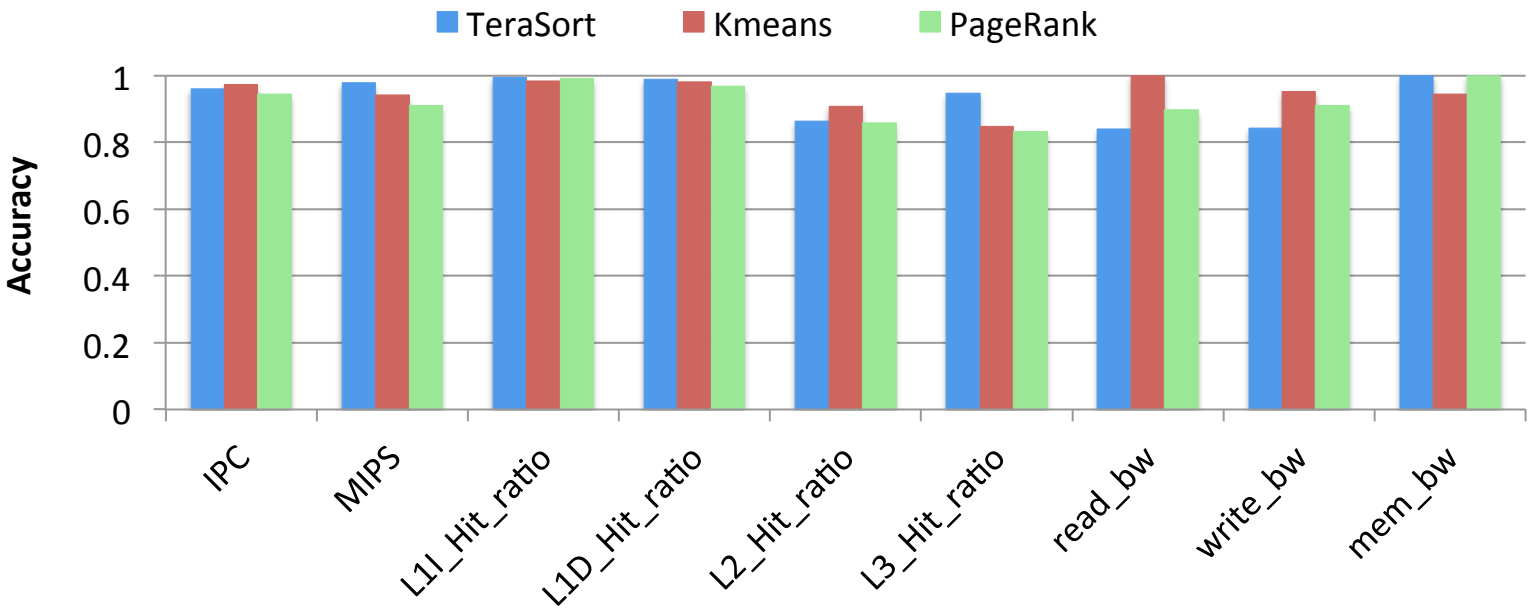
# CPU Utilization on ARMv8

- The similar utilization of CPU resources



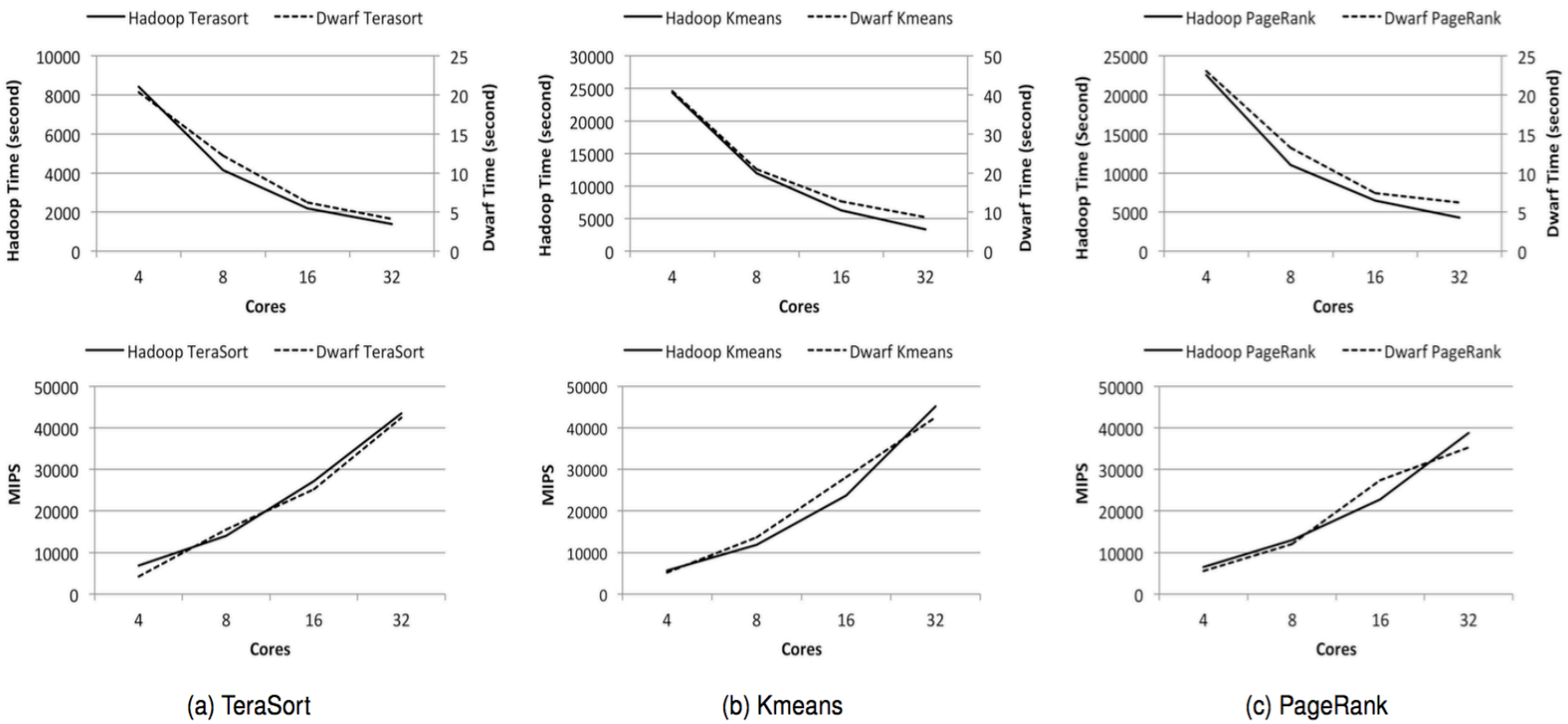
# Data Accuracy on ARMv8

- Average data accuracy
  - 93% for TeraSort, 95% for Kmeans, 92% for PageRank



# Multi-core Scalability on ARMv8

- Similar multi-core scalability trends



# Runtime Speedup Equation

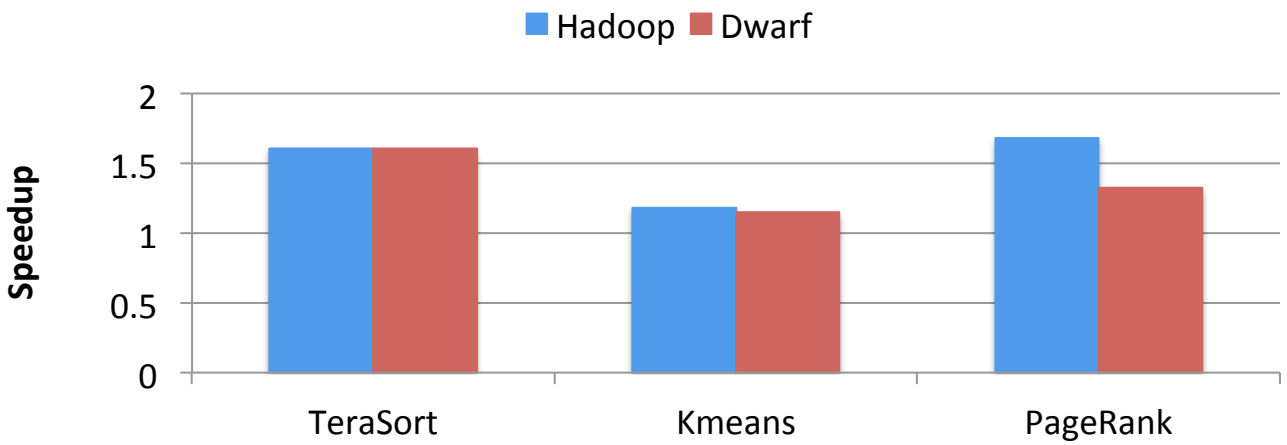
- The runtime speedup across different processors

$$Speedup(Time_{Haswell}, Time_{ARMv8}) = \frac{Time_{ARMv8}}{Time_{Haswell}}$$

- $Time_{Haswell}$ – runtime on Haswell processor
- $Time_{ARMv8}$ – runtime on ARMv8 processor

# Runtime Speedup across Different Processors

- ARMv8 V.S. Xeon E5-2690 V3 (Haswell)
  - Consistent speedup trends



# Future Work

- Expanding the dwarf set
  - We acknowledge our current eight dwarfs may be not enough for other applications we fail to investigate
- More implementations for the dwarf library
- Expanding dwarf-based simulation benchmarks

# Conclusion

- Big data dwarfs & Dwarf library
  - A comprehensive methodology
  - Eight kinds of frequently-appearing operations
- Dwarf-based simulation benchmarks
  - 100s runtime speedup with respect to the benchmarks from BigDataBench
  - average micro-architectural data accuracy is above 90% on both X86_64 and ARM

# Thank You !