BigDataBench:

a Big Data Benchmark Suite from Web Search Engines

Wanling Gao, Yuqing Zhu, Zhen Jia, Chunjie Luo, Lei Wang, Jianfeng Zhan, Yongqiang He, Shiming Gong, Xiaona Li, Shujie Zhang, and Bizhu Qiu

http://prof.ict.ac.cn/BigDataBench

Third Workshop on Architectures and Systems for Big Data (ASBD 2013) held in conjunction with ISCA-40

Acknowledgements

- This work is supported by the Chinese 973 project (Grant No.2011CB302502), the Hi-Tech Research and Development (863) Program of China (Grant No.2011AA01A203, No.2013AA01A213), the NSFC project (Grant No.60933003, No.61202075) , the BNSFproject (Grant No.4133081), and Huawei funding.

Requirements for Big Data Benchmarking

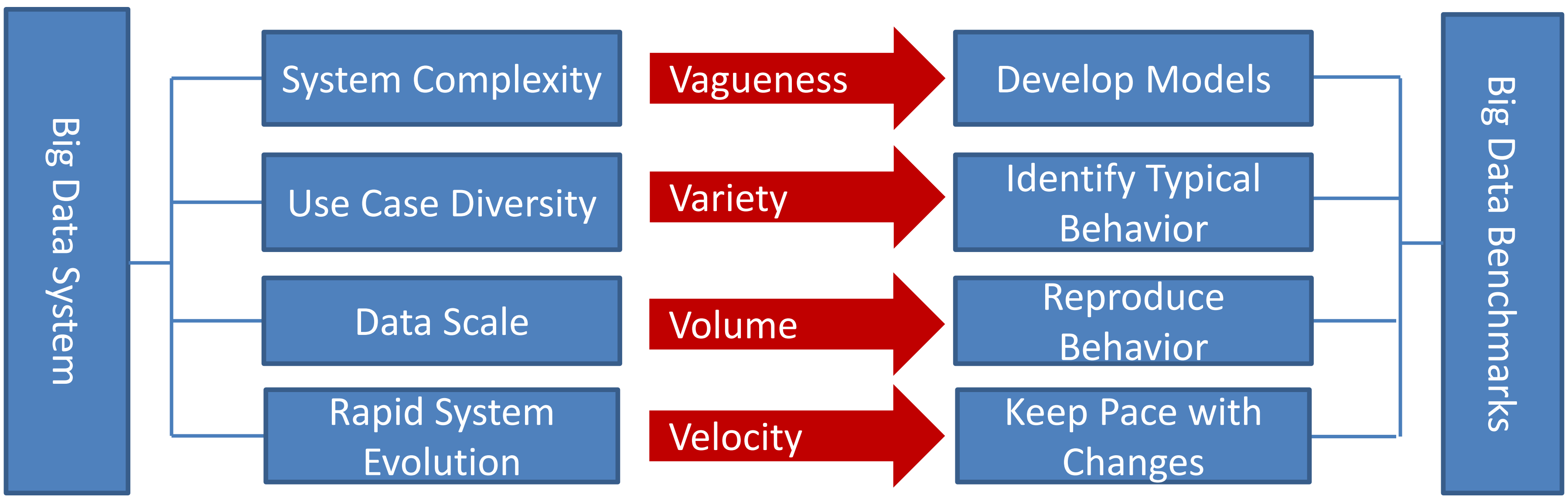
To truly reflect	To rapidly evolve	To widely cover
• Use cases • Requirements	• New workloads • New use cases	• Application domains • Data types

<<Propoal for a Big Data Benchmark Repository>>--Andries Engelbrecht . WBDB2012

Challenges

- Current State--Immature
  - "We Don't Know Enough to Make a Big Data Benchmark Suite"

An Academia-Industry View, Yanpei Chen, UC Berkeley/Cloudera WBDB2012



State-of-Practice and State-of-Art

- Sort : MinuteSort, JouleSort, TeraByte Sort
  - Only one application: one-fits-all solution?
- GridMix, Hadoop microbenchmark
  - Generating data set randomly
  - Ignoring the characteristics of real-world data
- BigBench
  - From electronic commerce
  - Only describe a specification without implementation

BigDataBench: an open source project

- Available from http://prof.ict.ac.cn/BigDataBench
- Data generation tools
  - Generate GB, TB, or PB-scale data from seed data
- BigDataBench 1.0 Beta version
  - Six benchmarks from web search engines

Outline

- Motivation
- Benchmarking Methodology
- Case Studies
- Future Work

Benchmarking Methodology

- Following An Incremental Approach
- Considering Variety of Workloads
- Methodology of Generating Big Data

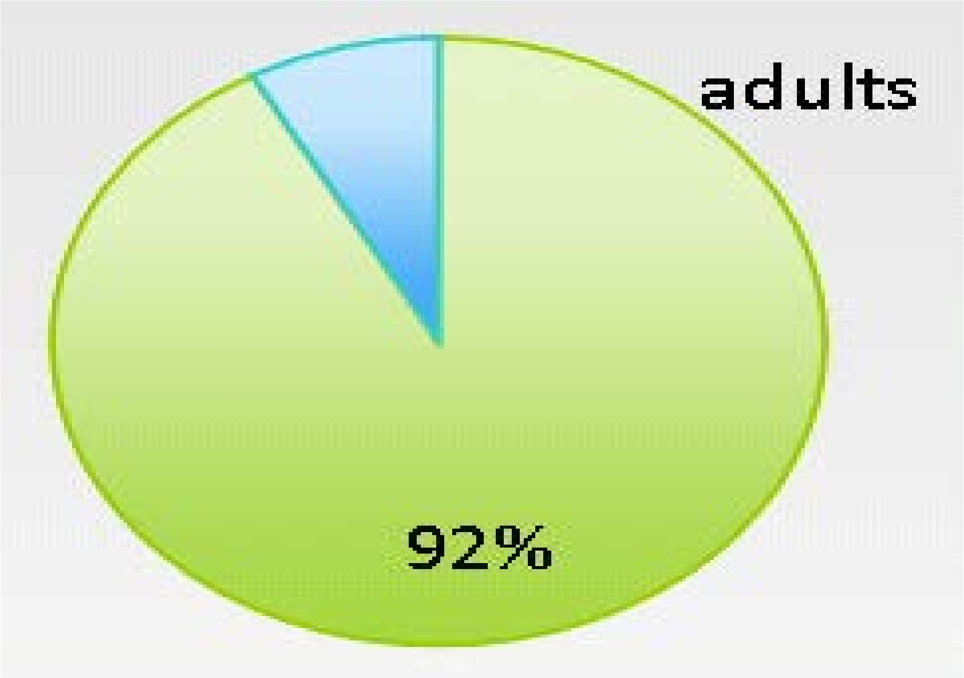
Investigate Application Domain

So many application domains

Why we choose Search Engine for the first step?

Popularity of Search Engines

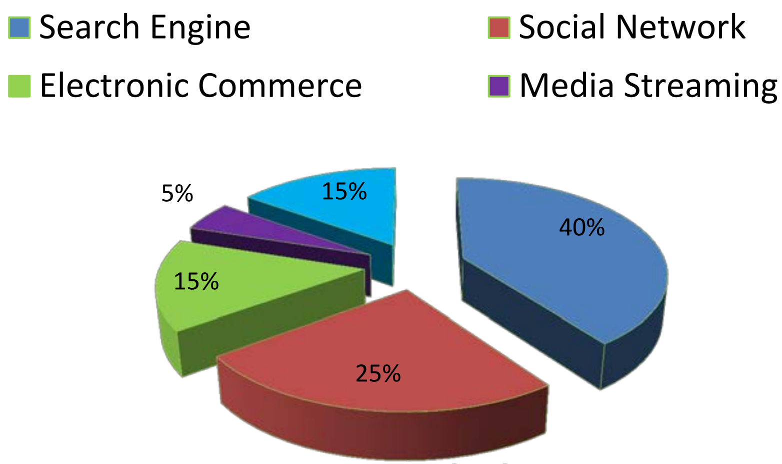
92% of online adults use ***search engines*** to find information on the web.



(***pew internet study***)

Page Views & Daily Visitors

- Search Engine is the most important Internet service workload.
- 40% of top 20 websites



Top 20 websites

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

Benchmarking Methodology

Following An Incremental Approach

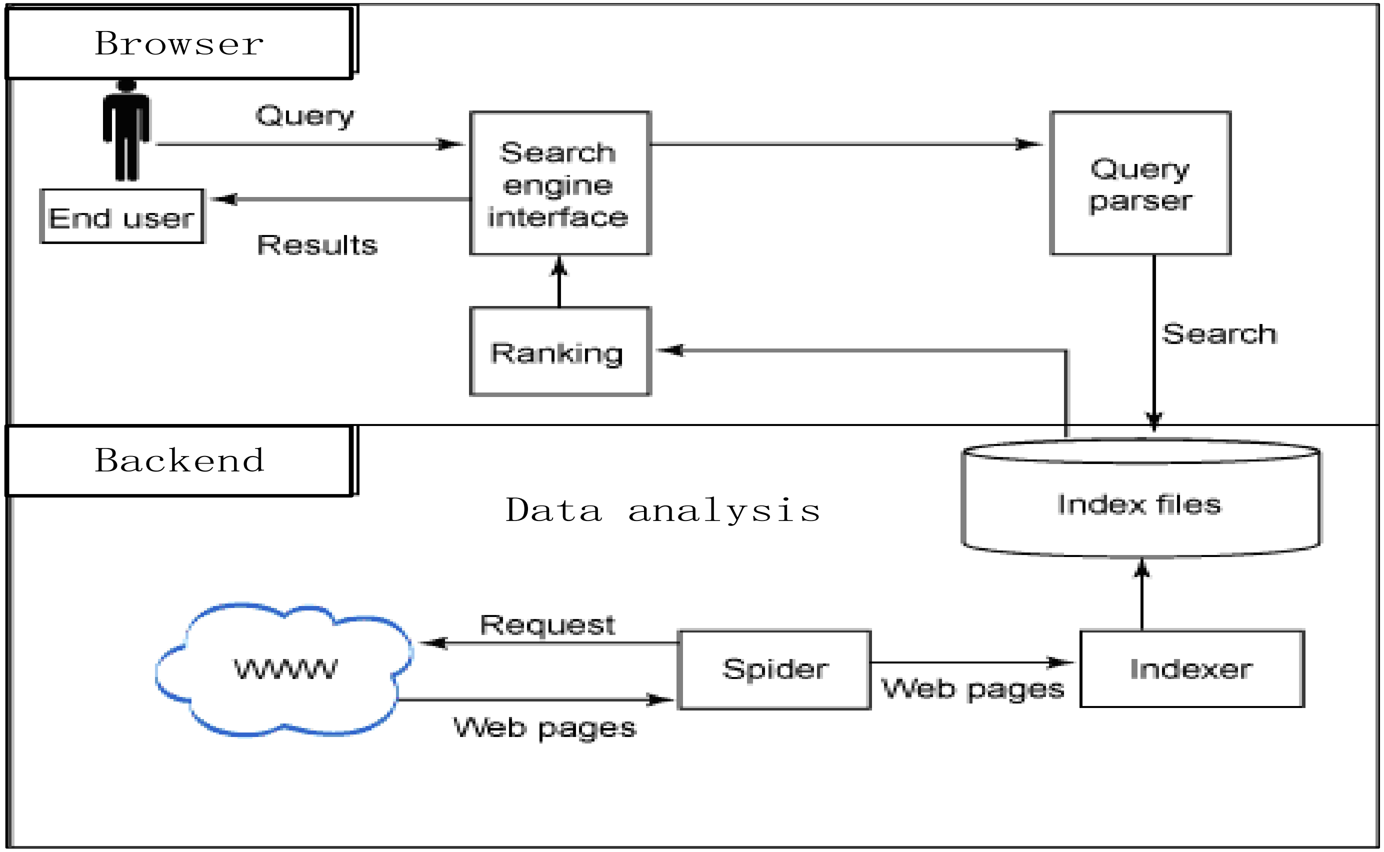
Considering Variety of Workloads

Methodology of Generating Big Data

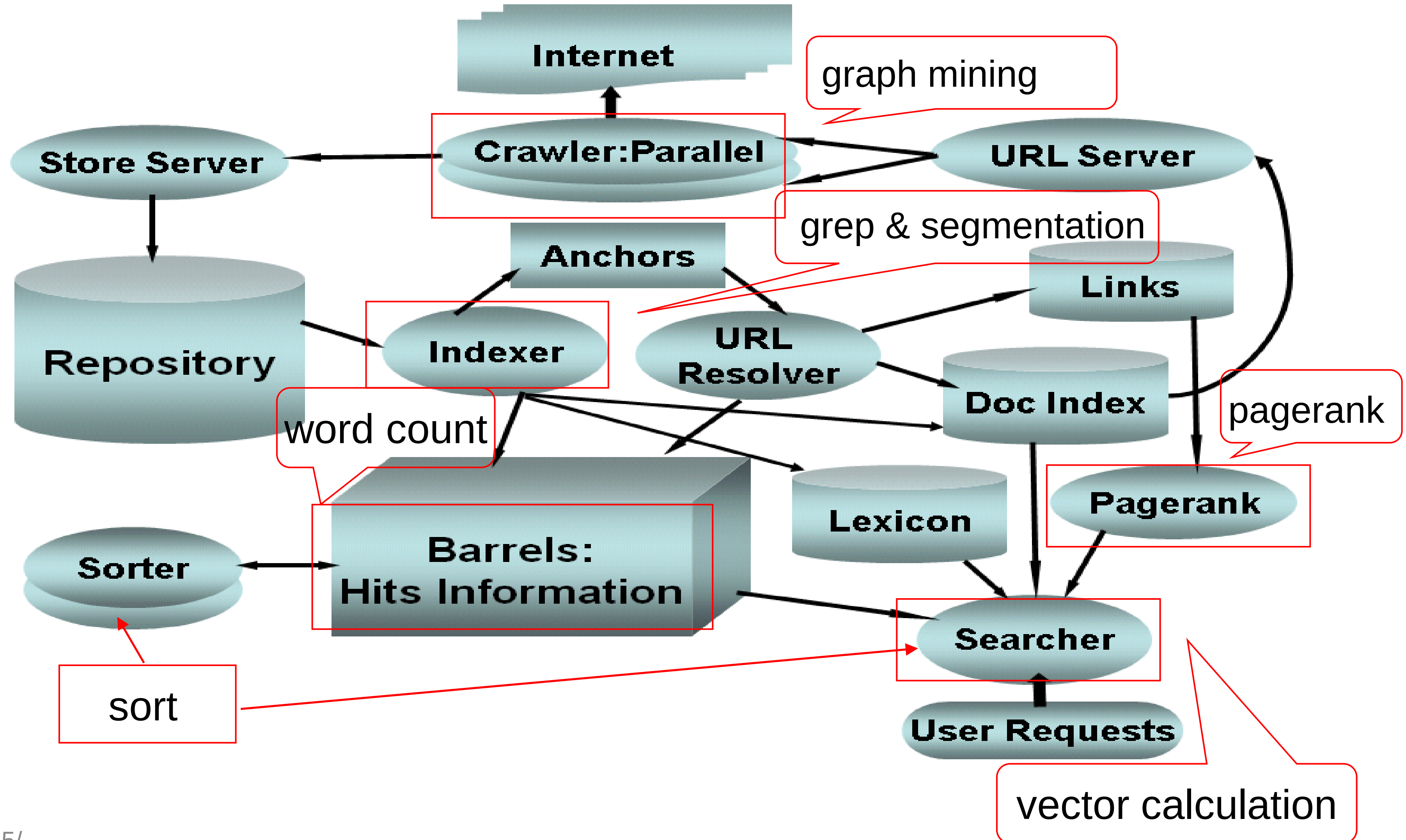
Our two practices

- Surveying Open-Source Common Search Engines
  - Backed by practioners of several industry partners
    - From facebook YAHOO! HUAWEI Baidu 百度 Sogou 搜狗
- Building a sematic search engine (Chinese)
  - ProfSearch
    - Search scientists or professionals
    - 251,564 researchers across 260 universities and institutes
    - http://prof.ict.ac.cn/

Overview of a Typical Search Engine



Algorithms in a Typical Search Engine



ProfSearch

Crawler Workloads

- Scrapy

Analysis Workloads

- SVM, Naïve Bayes, K-means, HMM, CRFs, LSA, LDA

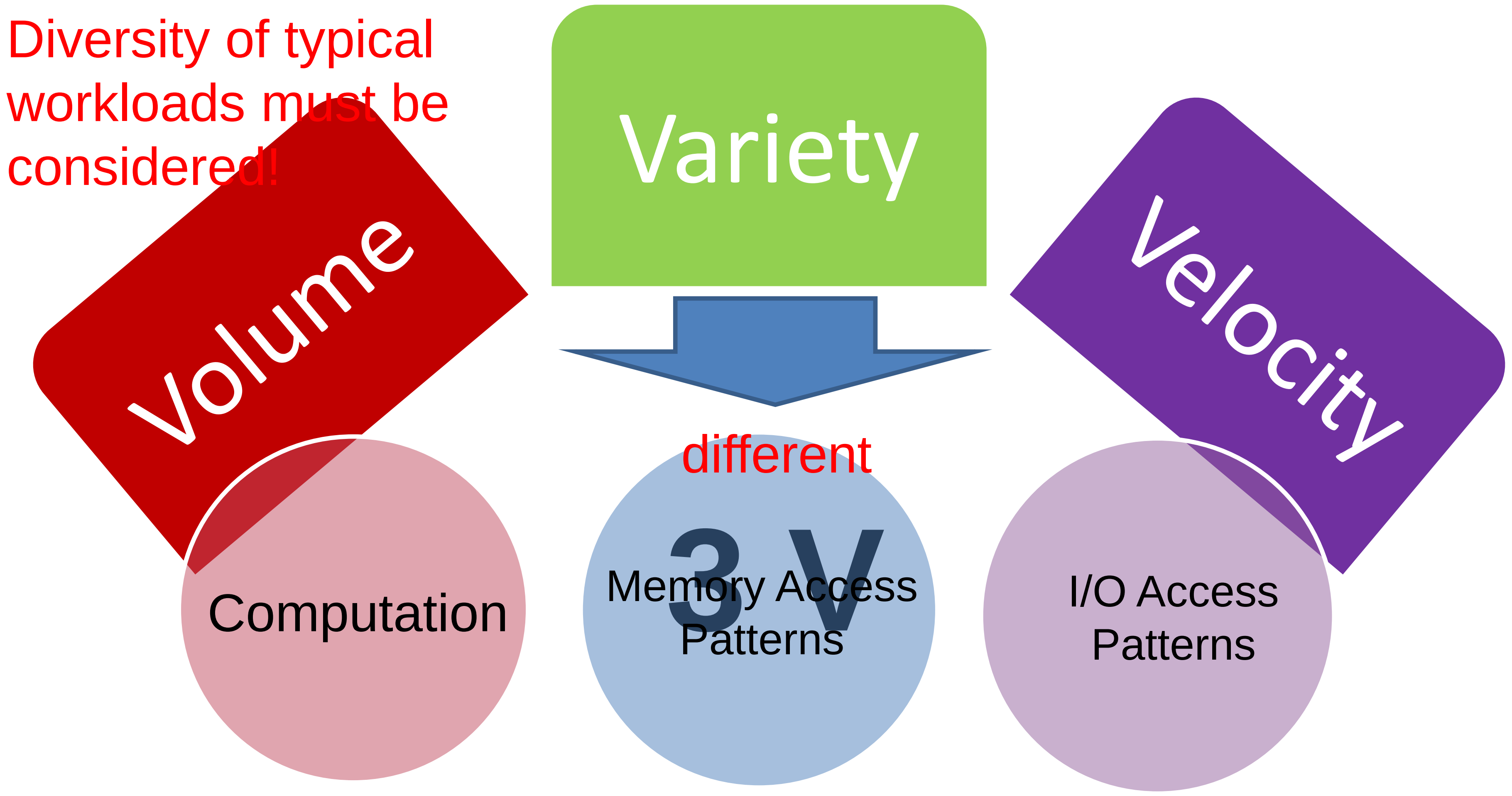
Store and Management Workloads

- HDFS – Storing unstructured web pages
- HIVE – Storing semi-structured intermediate data
- MySQL – Storing structured data extracted from the web

Web Service Workloads

- Sphinx

Three "V" of Big Data



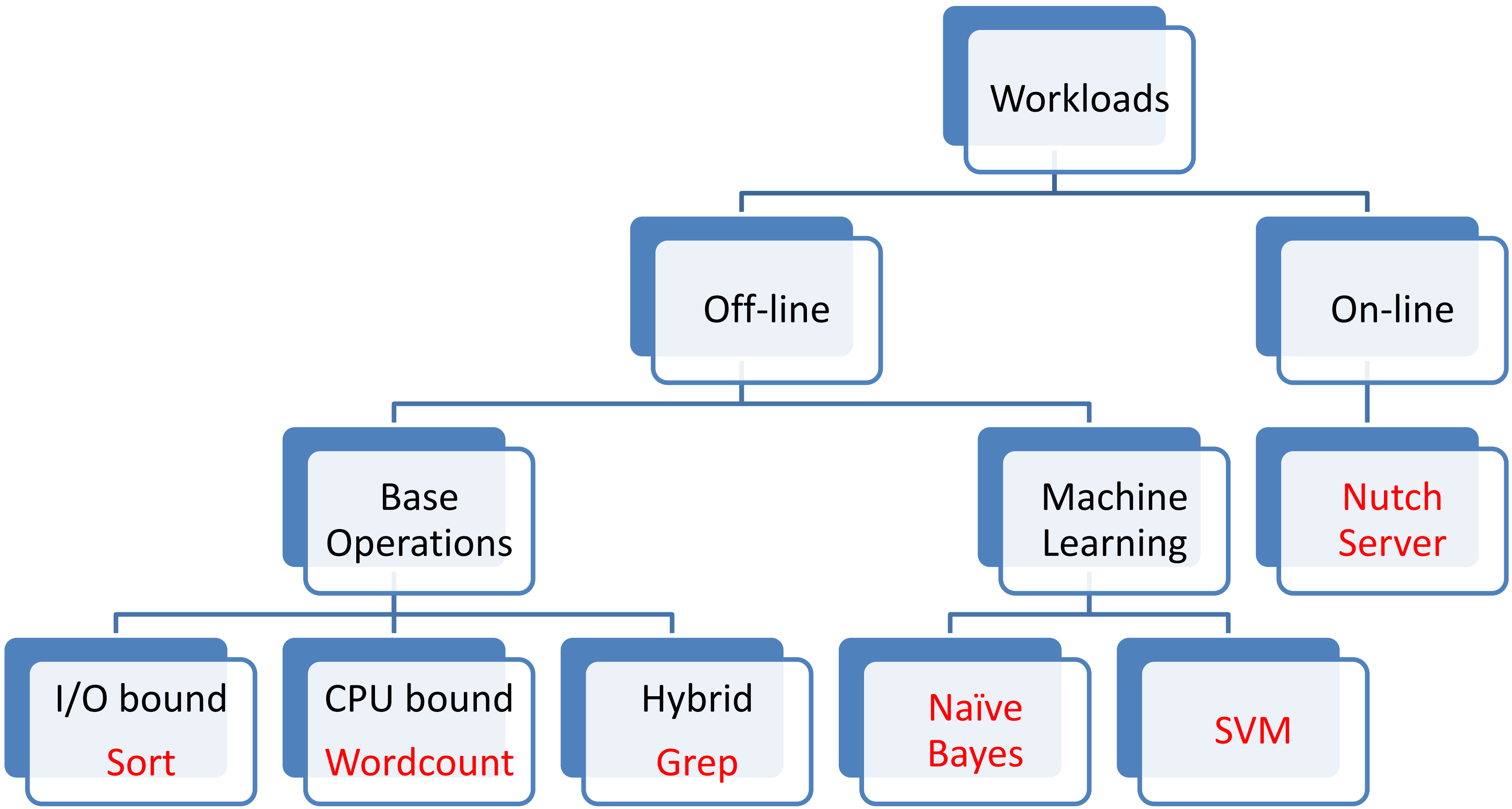
Workloads in BigDataBench 1.0 Beta

- Analysis Workloads
  - Simple but representative operations
    - Sort, Grep, Wordcount
  - Highly recognized algorithms
    - Naïve Bayes, SVM
- Search Engine Service Workloads
  - Widely deployed services
    - Nutch Server

Features of Workloads

Workloads	Resource Characteristic	Computing Complexity	Instructions
Sort	I/O bound	O(n*lgn)	Integer comparison domination
Wordcount	CPU bound	O(n)	Integer comparison and calculation domination
Grep	Hybrid	O(n)	Integer comparison domination
Naïve Bayes	/	O(m*n) [m: the length of dictionary]	Floating-point computation domination
SVM	/	O(M*n) [M: the number of support vectors * dimension]	Floating-point computation domination
Nutch Server	I/O & CPU bound		Integer comparison domination

Variety of Workloads are included



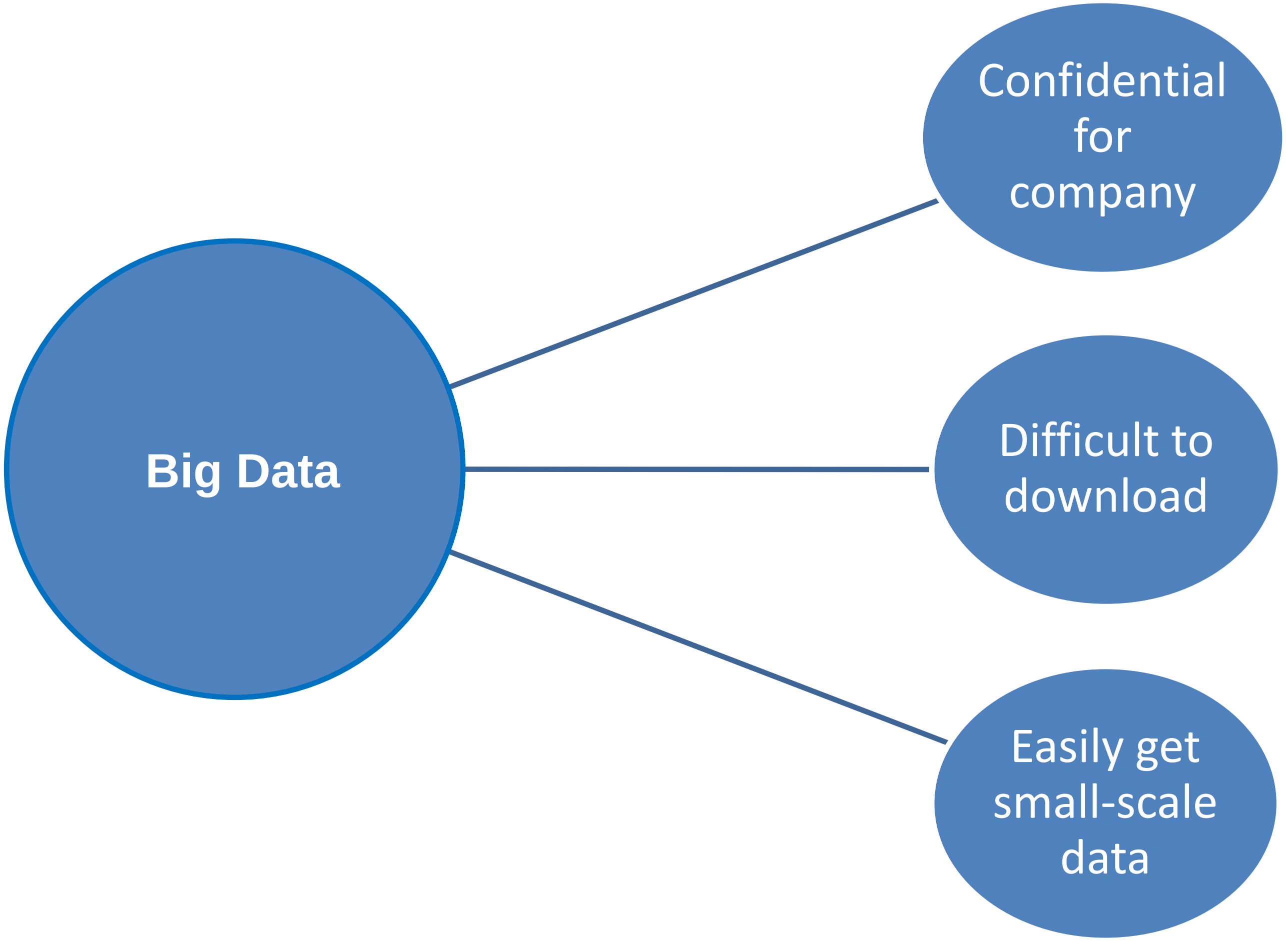
Benchmarking Methodology

Following An Incremental Approach

Considering Variety of Workloads

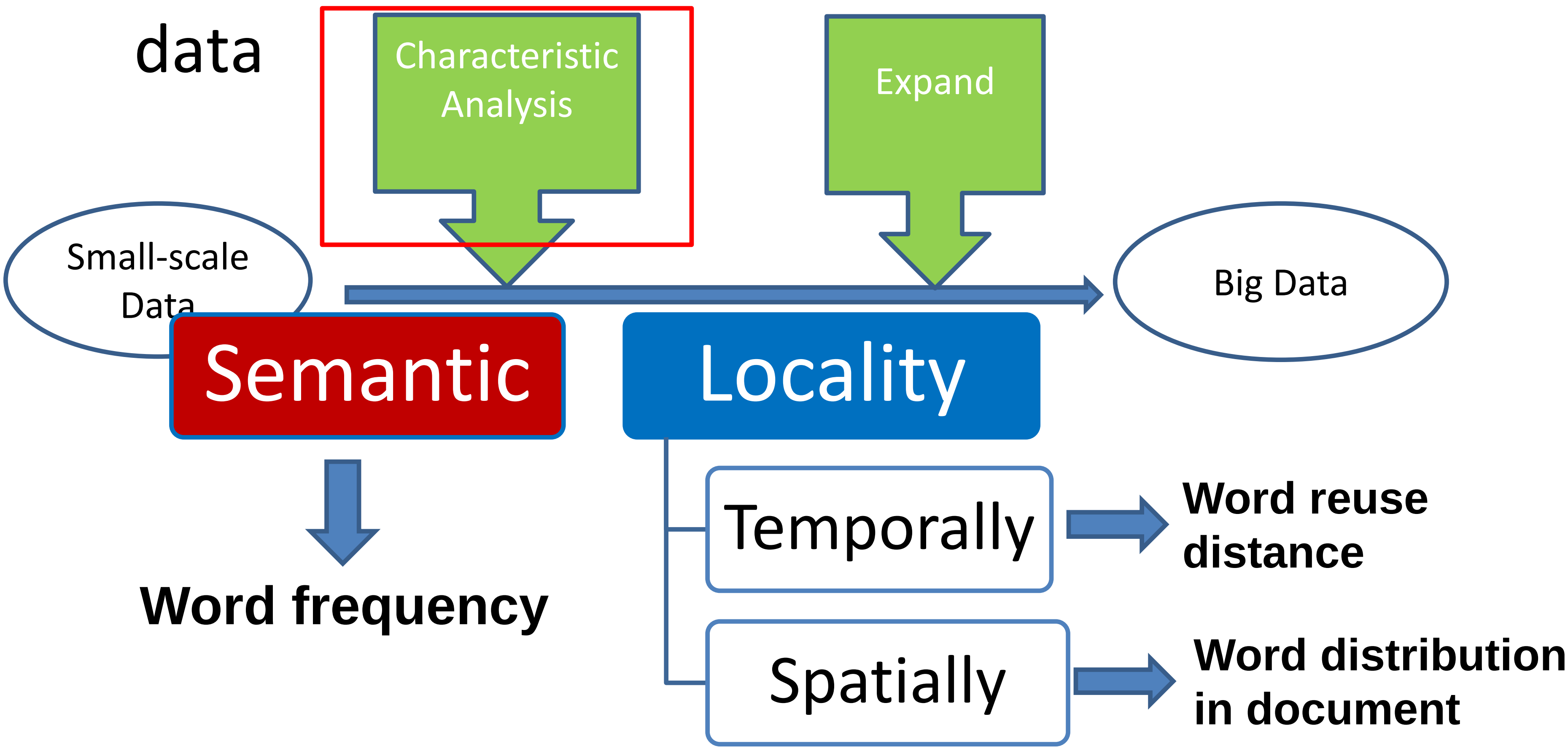
Methodology of Generating Big Data

Big Data Puzzle



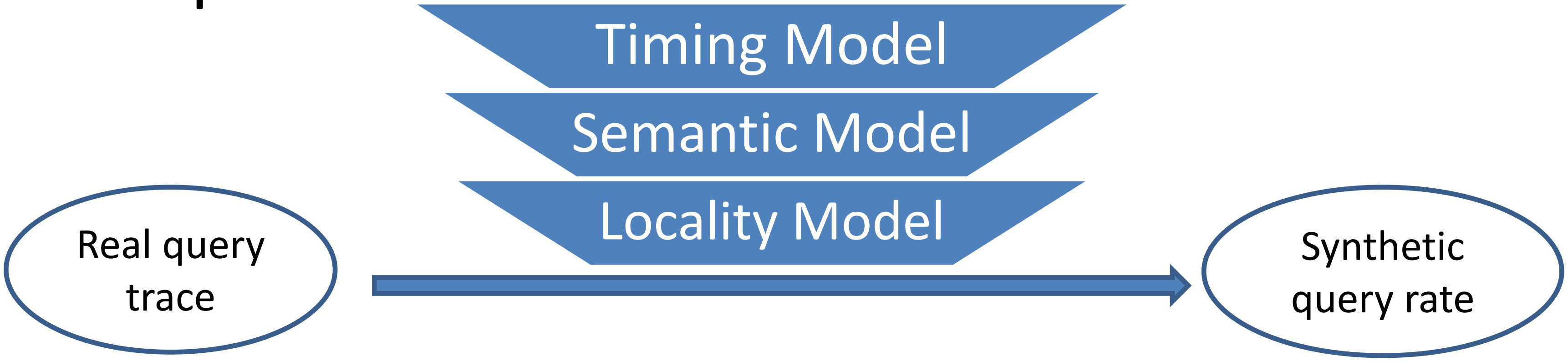
Methodology of Generating Big Data

- To preserve the characteristics of real-world data

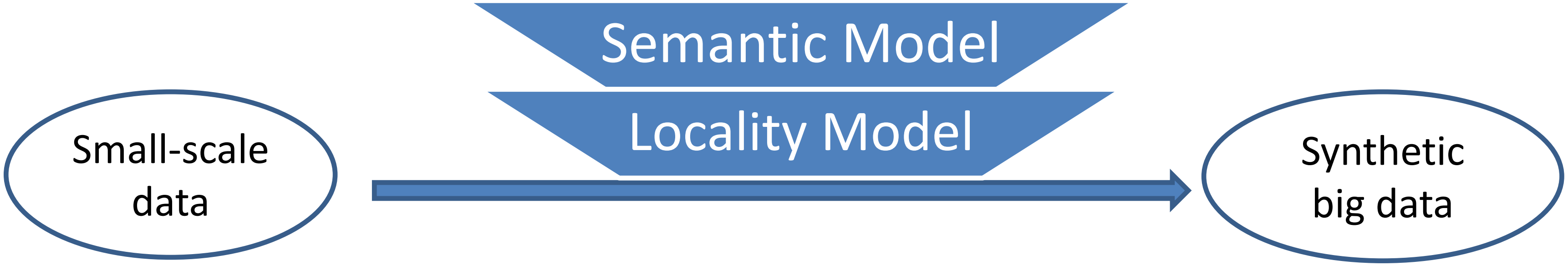


Scalable Data Generation Tool

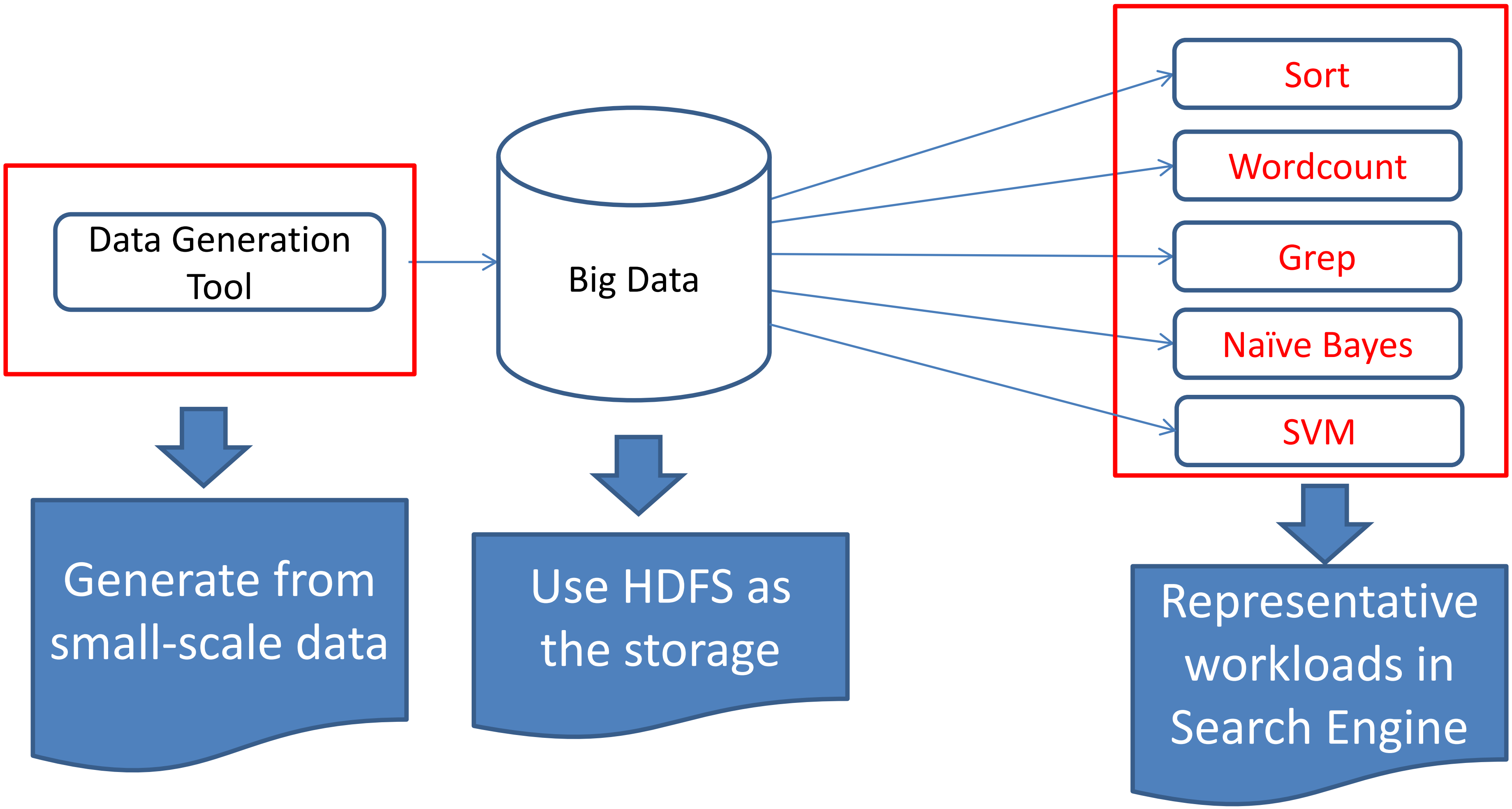
- Request Generation



- Input Data Generation



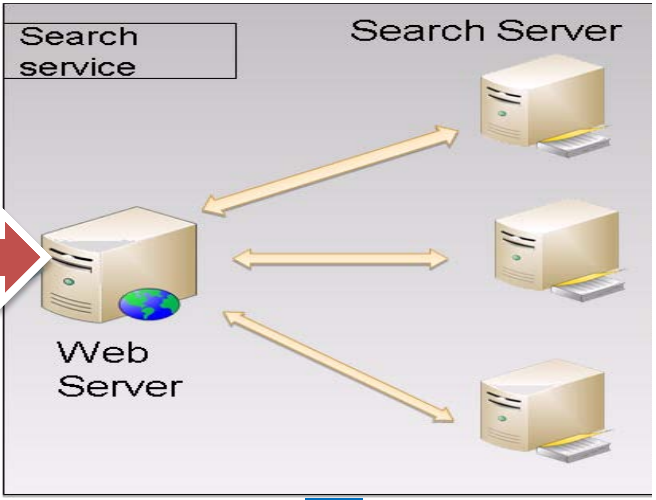
BigDataBench-Data Analysis Workloads



BigDataBench-Service Workload

Data Generation Tool

Massive Requests



Generate from small-scale query sequences and timing sequences

Use local file system as the storage

Search engine

Outline

- Motivation
- Benchmarking Methodology
- Case Studies
- Future Work

Use Case 1: System Evaluation

- Using BigDataBench 1.0 Beta
- Data Scale
  - 10 GB – 2 TB
- Hadoop Configuration
  - 1 master 14 slave node

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

Table 1. Details of node configuration

System Evaluation

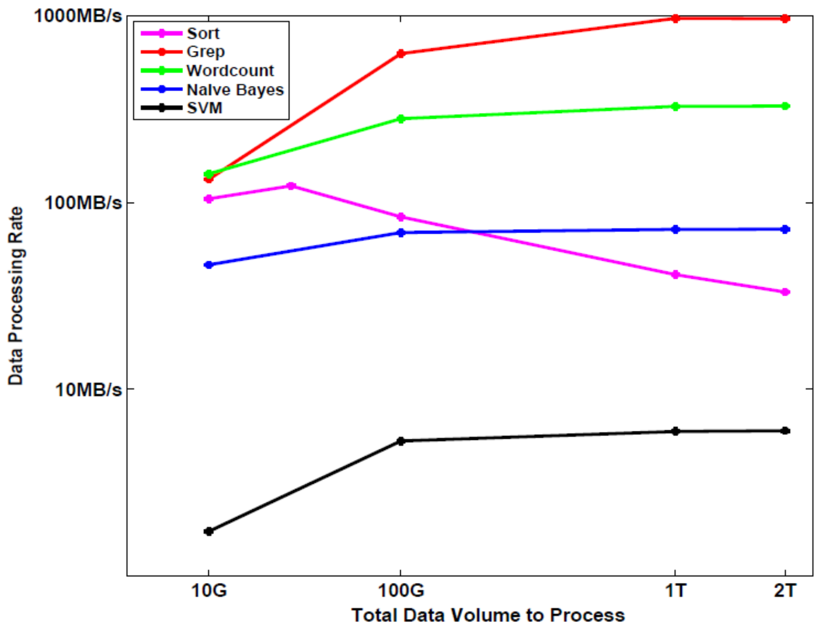


Figure 3. Data Processing Rates for Different Data Volumes to Process.

- a threshold for each workload
  - 100M ~ 1TB
  - System is fully loaded when the data volume exceeds the threshold
- Sort is an exception
  - An inflexion point(10GB ~ 1TB)
  - Data processing rate decreases after this point
  - Global data access requirements
    - I/O and network bottleneck
- System performance is dependent on applications and data volumes.

Use case 2: Architecture Research

- Using BigDataBench 1.0 Beta
- Data Scale
  - 10 GB – 2 TB
- Hadoop Configuration
  - 1 master 14 slave node

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

Table 1. Details of node configuration

Use case 2: Architecture Research

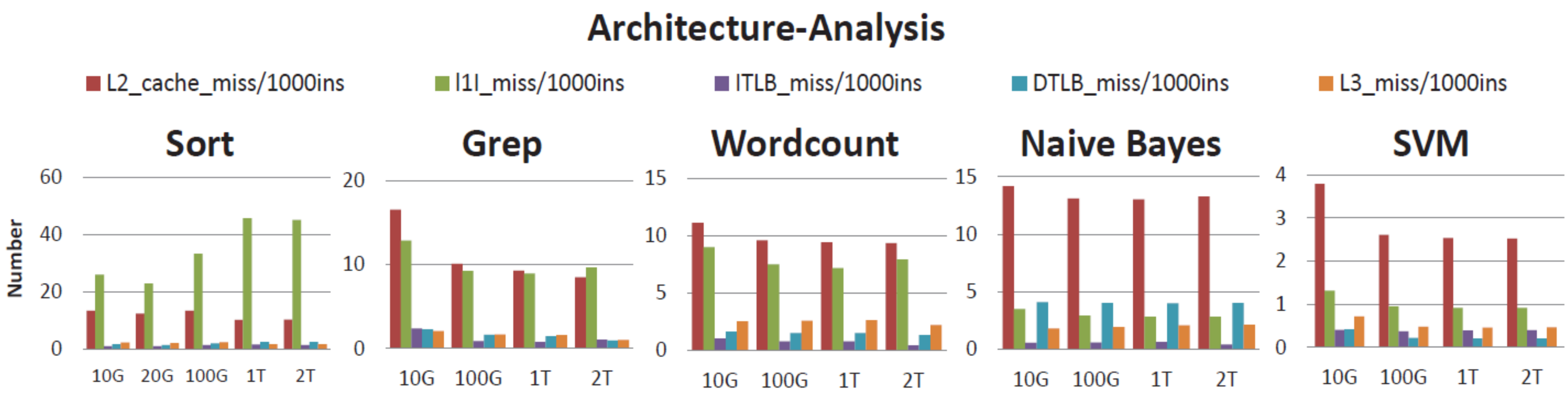


Figure 4. Cache and TLB Behaviors of Data Analysis Applications.

- Some micro-architectural events are tending towards stability when the data volume increases to a certain extent
- Cache and TLB behaviors have different trends with increasing data volumes for different workloads
  - L1I_miss/1000ins: increase for Sort, decrease for Grep

Search engine service experiments

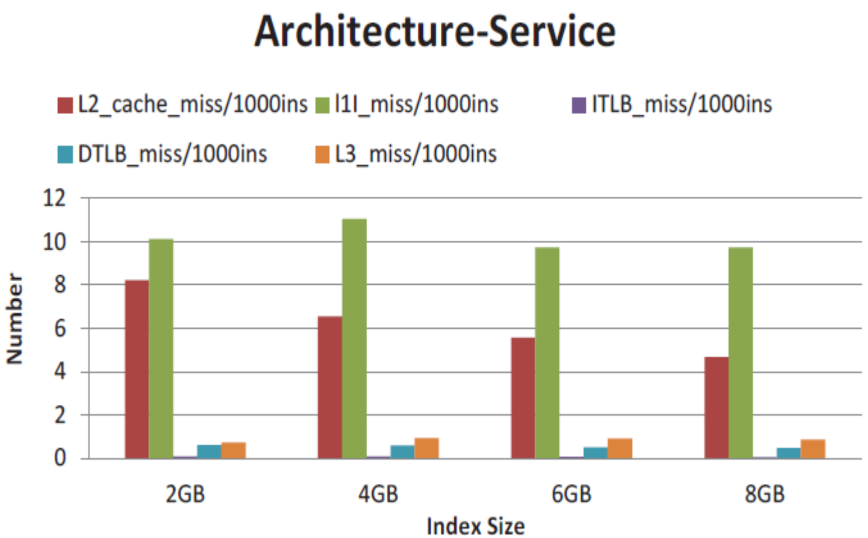


Figure 5. Cache and TLB Behaviors of the Nutch Server.

Index size：2GB ~ 8GB
Segment size：4.4GB ~ 17.6GB

- Same phenomena is observed
  - Micro-architectural events are tending towards stability when the index size increases to a certain extent
- Big data impose challenges to architecture researches since large-scale simulation is time-consuming

Outline

- Motivation
- Benchmarking Methodology
- Case Studies
- Conclusion and Future Work

Conclusion (1)

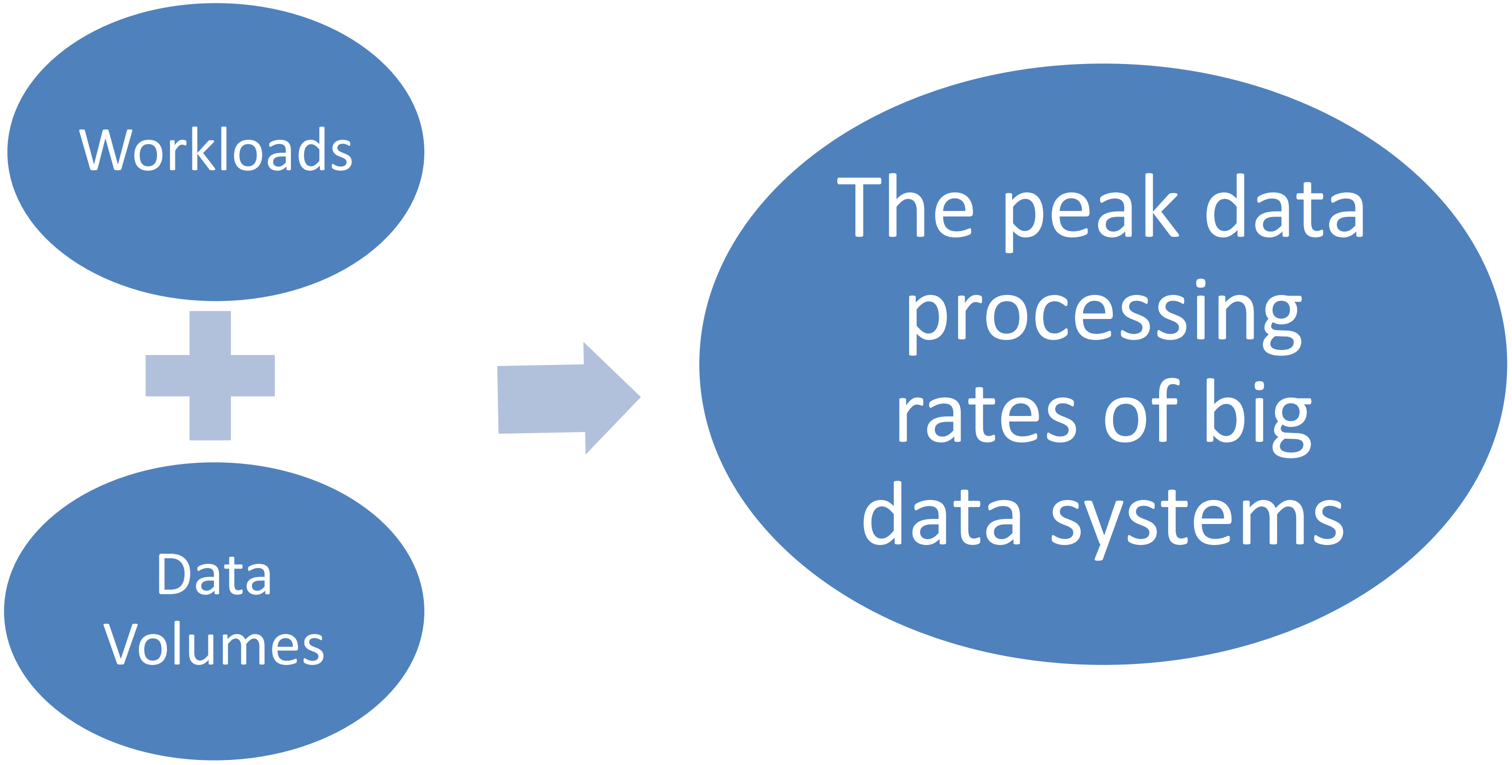
- We create a big data benchmark suite from web search engines
  - Data generation tools and six workloads

- First open-source project on big data benchmarking
  - http://prof.ict.ac.cn/BigDataBench

- Welcome downloading

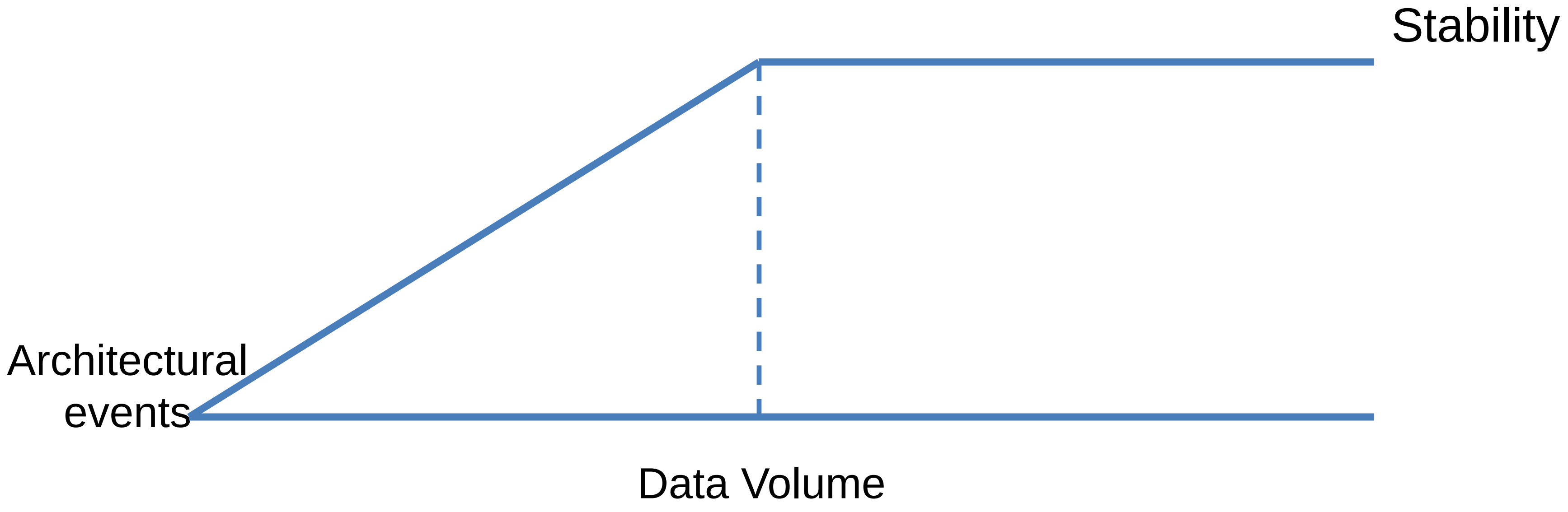
Conclusions (2)

The peak performance of big data systems is dependent on applications and data volumes



Conclusions (3)

Big data impose challenges to experiment methodologies in architecture researches



Future Work (1)

- Release BigDataBench 2.0
  - Consider data variety!
    - Structured, semi-structure, and unstructured data
  - Include multimedia data and applications
  - Include online data analysis applications

Future Work (2)

- Include workloads in other important internet service domains
  - Electronic commence
  - Social network

- Internet services take up only a small part of big data applications
  - A long way to go!!!

Thank you!

Any questions?