BigDataBench: A Dwarf-based Big Data and AI Benchmark Suite

Wanling Gao

http://prof.ict.ac.cn

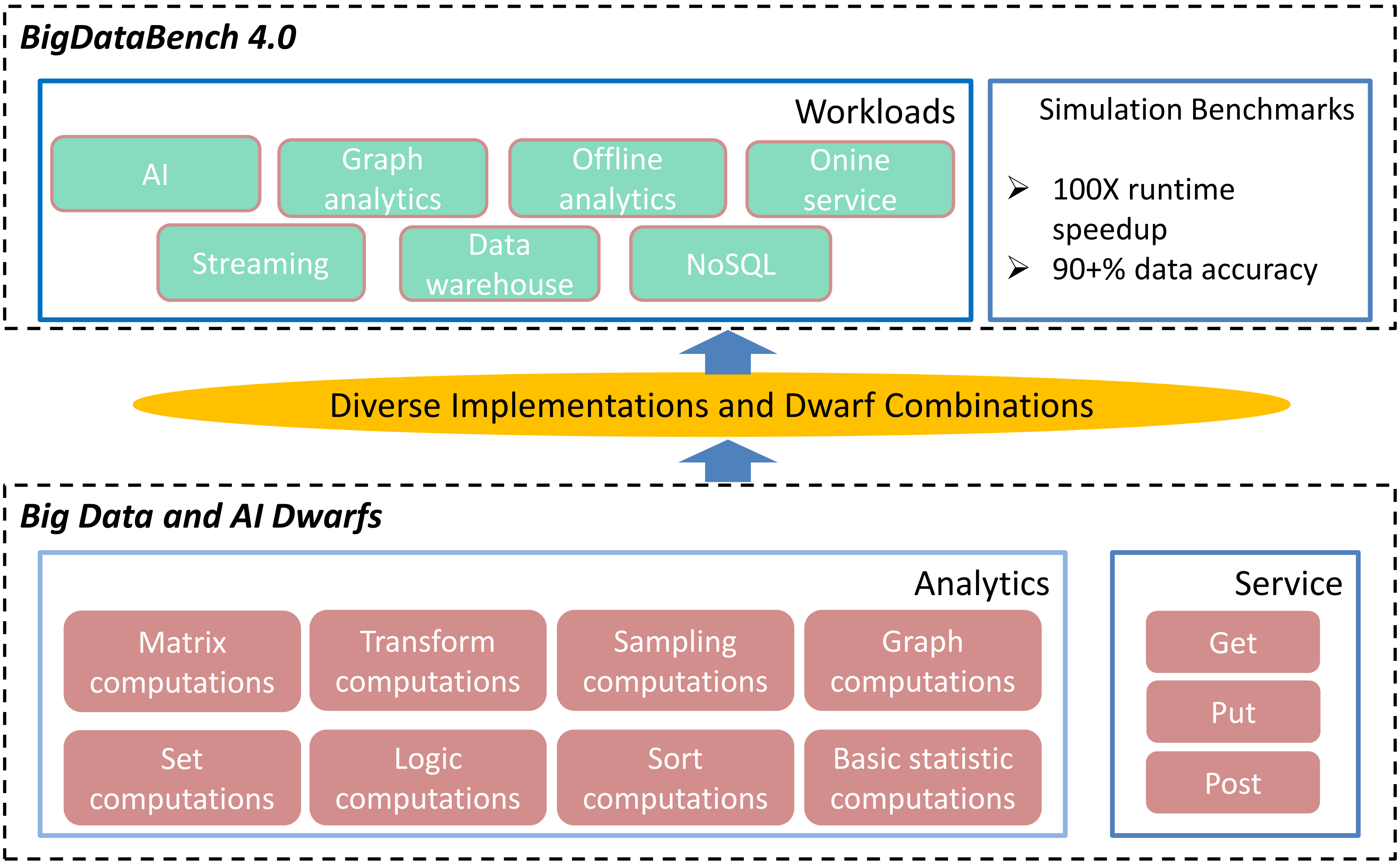
ICT, Chinese Academy of Sciences

ASPLOS 2018, Williamsburg, VA, USA

中科院计算所

INSTITUTE OF COMPUTING TECHNOLOGY

Executive summary



BigDataBench Publications

- BigDataBench: a Big Data Benchmark Suite from Internet Services. 20th IEEE International Symposium On High Performance Computer Architecture (**HPCA-2014**).
- Understanding Big Data Analytics Workloads on Modern Processors. **TPDS 2017**. https://arxiv.org/pdf/1504.04974.pdf
- Characterizing data analysis workloads in data centers. 2013 IEEE International Symposium on Workload Characterization (**IISWC 2013**)（Best paper award）
- BigDataBench: a Dwarf-based Big Data and AI Benchmark Suite. Technical Report. https://arxiv.org/pdf/1802.08254.pdf
- BOPS, Not FLOPS! A New Metric, Measuring Tool, and Roofline Performance Model For Datacenter Computing. Technical Report. https://arxiv.org/pdf/1801.09212.pdf
- Big Data Dwarfs: Towards Fully Understanding Big Data Analytics Workloads. Technical Report. https://arxiv.org/pdf/1802.00699.pdf

BigDataBench 4.0 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	CIFAR-10	LSUN
TED Talks	SoGou Data	MovieLens Dataset
	MNIST	

13 Real-world Data Sets

AI workloads
Offline analytics
Online service
Streaming
Graph analytics
Data warehouse
NoSQL workloads

- Micro benchmarks
- Component benchmarks
- Application benchmarks

47 Workloads with 7 types

JStorm, Flink, GraphLab, *NoSql*, *Impala*, GraphX, Caffe, Hive, Oracle, MySQL, *Shark*, TensorFlow, *Hadoop RDMA*, MVAPICH, *MPI*, Spark, hadoop, *DataMPI*

16 Software Stacks

What's New in BigDataBench 4.0

Dwarf-based benchmarking methodology

- Using dwarf combinations to represent big data and AI workloads
- Specification for micro, Component and Application Benchmarks

Seven workload types

- AI, Online service, Offline analytics, Graph analytics, Streaming, Data warehouse, NoSQL

Dwarf-based simulation benchmarks

- 100X runtime speedup, 90+% average accuracy

Overview

- *Dwarf-based Benchmarking Methodology*

- *Workload Characterization*

Benchmark Challenge

- Complexity and diversity of big data and AI systems
  - Complex software stacks
  - Diversity and frequently changed workloads
  - Rapid evolution of big data and AI systems
- Benchmark fairness
  - Benchmarks must include diversity of data and workloads
    - Data and workloads have great impacts on system and architecture evaluation
- Benchmark consistent across different communities
  - Different benchmark requirements for system, architecture and AI community
  - For the co-design of software and hardware

What‘s Dwarf and Why Dwarf

- Dwarf Definition
  - Captures the common requirements of each class of unit of computation
    - Being reasonably divorced from individual implementations
  - ***A minimum set*** to represent ***maximum patterns***

Benchmarking scalability

Portability cost

Better interpretation of performance data

We need to understand What’s the abstractions of frequently-appearing units of computation among big data and AI workloads (big data and AI dwarf)?

Relational Model of Structured Data

- *E. F. Codd, A relational Model of Data for Large shared data banks. Communication of ACM, vol 13. no.6, 1970.*
- *Set* concept : general mathematical meaning
  - General representation of data
  - Basis of relational algebra (theoretical foundation of database)
  - 5 basic operations
    - Select, Project, Product, Union, Difference

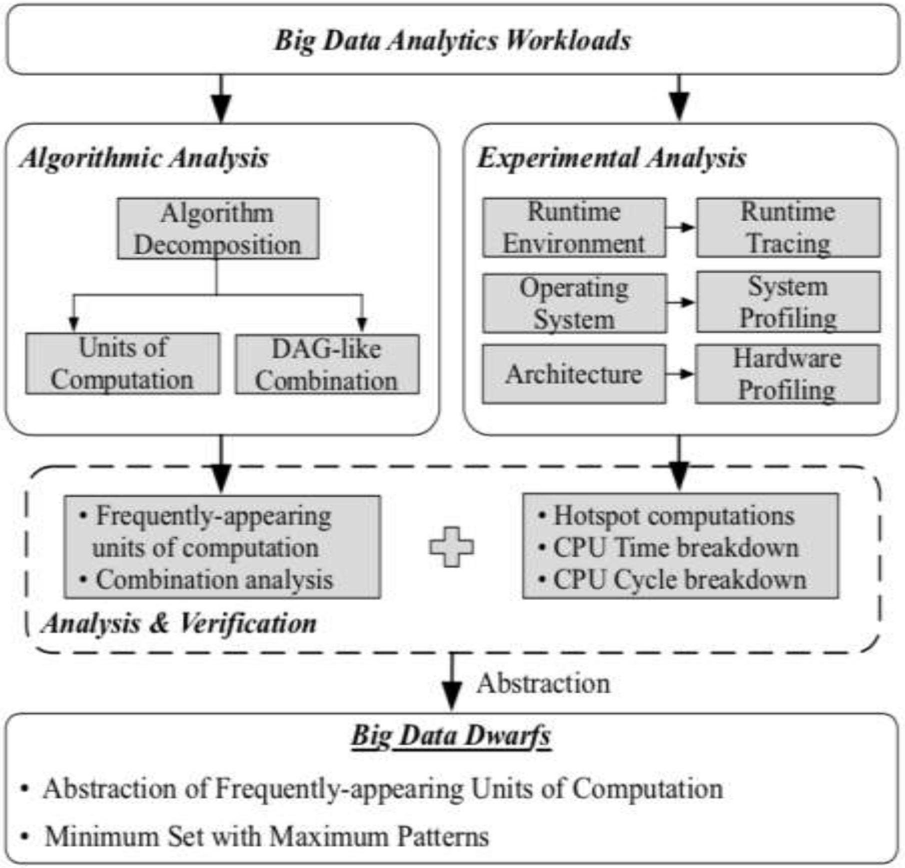
TPC-C Benchmark

- An On-Line Transaction Processing Benchmark

- Units of Computation
  - a mid-weight read-write trans- action (i.e., New-Order)
  - a light-weight read-write transaction (i.e., Payment)
  - a mid-weight read-only transaction (i.e., Order-Status)
  - a batch of mid-weight read-write transactions (i.e., Delivery)
  - a heavy-weight read-only transaction (i.e., Stock-Level)

How to Abstract Big Data and AI Dwarf

- Big Data &AI Dwarf
  - Units of computation
- Dwarf Abstraction
  - Algorithmic analysis
  - Experimental analysis



Big Data and AI Dwarfs

- Matrix computations
- Sampling computations
- Transform computations
- Graph computations
- Logic computations
- Set computations
- Basic statistic computations
- Sort computations

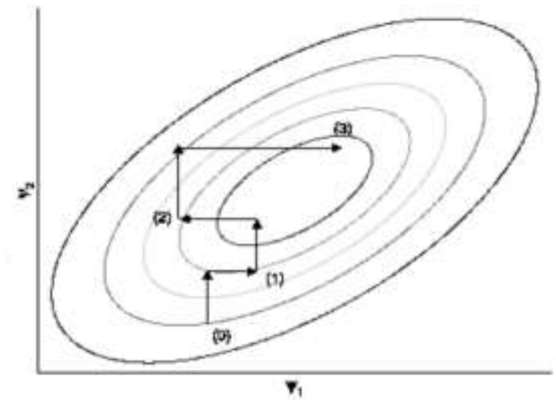
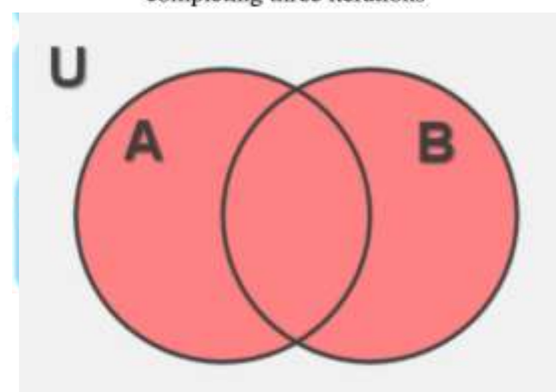


Figure 3.4: Gibbs sampling algorithm in two dimensions starting from an initial point and then completing three iterations



Why Dwarf-based Benchmarking

- Using the combination to represent a wide variety of big data and AI workloads
  - No need to create a new benchmark or proxy for every possible workload

Big Data and AI Dwarf

- Matrix
- Sampling
- Transform
- Graph
- Logic
- Set
- Basic statistics
- Sort

Combinations with different weights

Diverse big data and AI workloads

- Coverage of fundamental units of computation
- Provide the methodology of choosing typical workloads
- Reduce workload redundancy

Methodology Principle

Separating specification from implementation.

- Model relevant domains

State-of-the-art algorithms and technologies

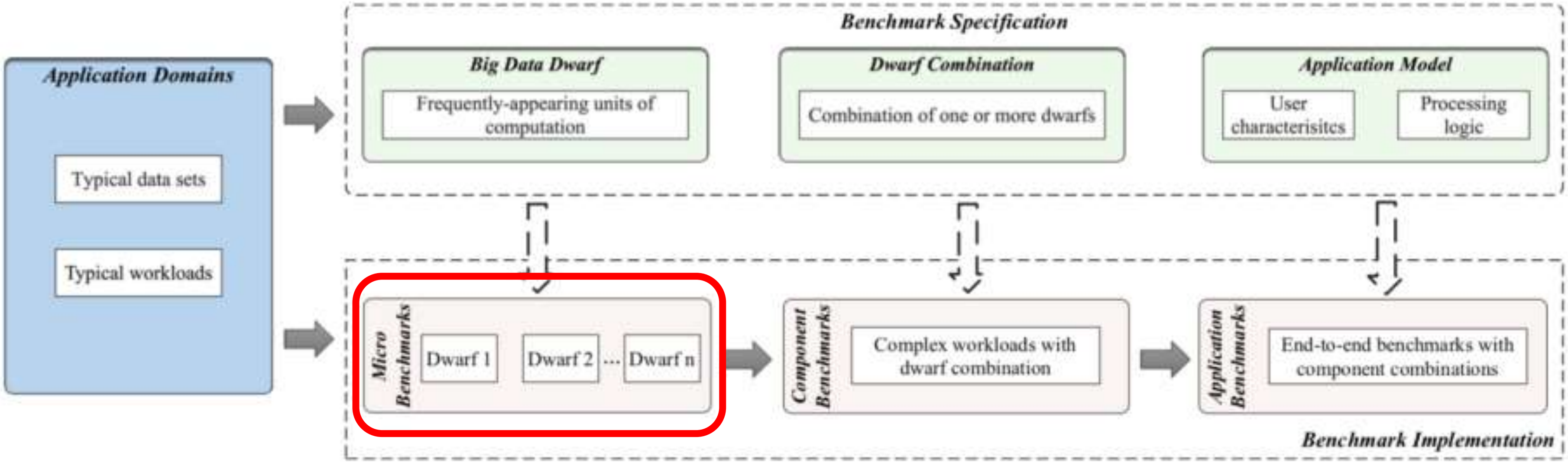
- Implementation keep in pace with the improvement

Data impact

- Representative data sets considering typical types and sources

Benchmarking Methodology

- Circling around the dwarfs
  - Specification of micro, component and application benchmark



Micro Benchmarks

Offline analytics

Graph analytics

Streaming

NoSQL

AI

Micro Benchmark	Involved Dwarf	Application Domain	Workload Type	Data Set	Software Stack
Sort	Sort	SE, SN, EC, MP, BI	Offline analytics	Wikipedia entries	Hadoop, Spark, Flink, MPI
Grep	Set		Offline analytics	Wikipedia entries	Hadoop, Spark, Flink, MPI
			Streaming	Random Generate	Spark streaming
WordCount	Basic statistics		Offline analytics	Wikipedia entries	Hadoop, Spark, Flink, MPI
MD5	Logic		Offline analytics	Wikipedia entries	Hadoop, Spark, MPI
Connected Component	Graph	SN	Graph analytics	Facebook social network	Hadoop, Spark, Flink, GraphLab, MPI
RandSample	Sampling	SE, MP, BI	Offline analytics	Wikipedia entries	Hadoop, Spark, MPI
FFT	Transform	MP	Offline analytics	Two-dimensional matrix	Hadoop, Spark, MPI
Matrix Multiply	Matrix	SE, SN, EC, MP, BI	Offline analytics	Two-dimensional matrix	Hadoop, Spark, MPI
Read	Set	SE, SN, EC	NoSQL	ProfSearch resumes	HBase, MongoDB
Write	Set	SE, SN, EC	NoSQL	ProfSearch resumes	HBase, MongoDB
Scan	Set	SE, SN, EC	NoSQL	ProfSearch resumes	HBase, MongoDB
Convolution	Transform	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Fully Connected	Matrix	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Relu	Logic	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Sigmoid	Matrix	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Tanh	Matrix	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
MaxPooling	Sampling	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
AvgPooling	Sampling	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
CosineNorm [36]	Basic Statistics	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
BatchNorm [37]	Basic Statistics	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Dropout [38]	Sampling	SN, EC, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch

Component Benchmarks

Online service

Streaming

Offline analytics

Graph analytics

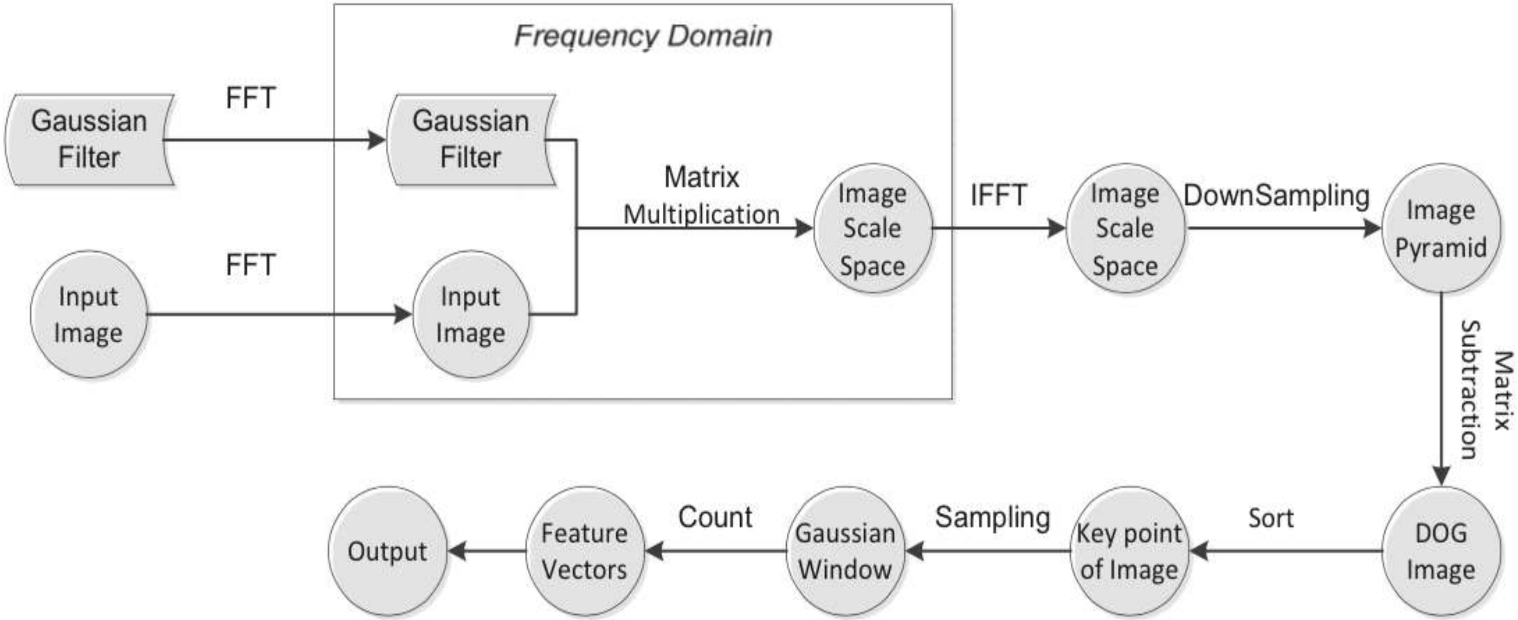
Data warehouse

AI

Component Benchmark	Involved Dwarf	Application Domain	Workload Type	Data Set	Software Stack
Xapian Server	Get, Put, Post	SE	Online service	Wikipedia entries	Xapian
PageRank	Matrix, Sort, Basic statistics, Graph	SE	Graph analytics	Google web graph	Hadoop, Spark, Flink, GraphLab, MPI
Index	Logic, Sort, Basic statistics, Set	SE	Offline analytics	Wikipedia entries	Hadoop, Spark
Rolling top words	Sort, Basic statistics	SN	Streaming	Random generate	Spark streaming, JStorm
Kmeans	Matrix, Sort, Basic statistics	SE, SN, EC, MP, BI	Offline analytics	Facebook social network	Hadoop, Spark, Flink, MPI
			Streaming	Random generate	Spark streaming
Collaborative Filtering	Graph, Matrix	EC	Offline analytics	Amazon movie review	Hadoop, Spark
		EC	Streaming	MovieLens dataset	JStorm
Naive Bayes	Basic statistics, Sort	SE, SN, EC	Offline analytics	Amazon movie review	Hadoop, Spark, Flink, MPI
SIFT	Matrix, Sampling, Transform, Sort	MP	Offline analytics	ImageNet	Hadoop, Spark, MPI
LDA	Matrix, Graph, Sampling	SE	Offline analytics	Wikipedia entries	Hadoop, Spark, MPI
OrderBy	Set, Sort	EC	Data warehouse	E-commerce transaction	Hive, Spark-SQL, Impala
Aggregation	Set, Basic statistics	EC		E-commerce transaction	Hive, Spark-SQL, Impala
Project, Filter	Set	EC	Data warehouse	E-commerce transaction	Hive, Spark-SQL, Impala
Select, Union	Set	EC		E-commerce transaction	Hive, Spark-SQL, Impala
Alexnet	Matrix, Transform, Sampling, Logic, Basic statistics	SN, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Googlenet		SN, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Resnet		SN, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
Inception Resnet V2		SN, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
VGG16		SN, MP, BI	AI	Cifar, ImageNet	TensorFlow, Caffe, pyTorch
DCGAN		SN, MP, BI	AI	LSUN	TensorFlow, Caffe, pyTorch
WGAN		SN, MP, BI	AI	LSUN	TensorFlow, Caffe, pyTorch
GAN	Matrix, Sampling, Logic, Basic statistics	SN, MP, BI	AI	LSUN	TensorFlow, Caffe, pyTorch
Seq2Seq		SE, EC, BI	AI	TED Talks	TensorFlow, Caffe, pyTorch
Word2vec	Matrix, Basic statistics, Logic	SE, SN, EC	AI	Wikipedia entries, Sogou data	TensorFlow, Caffe, pyTorch

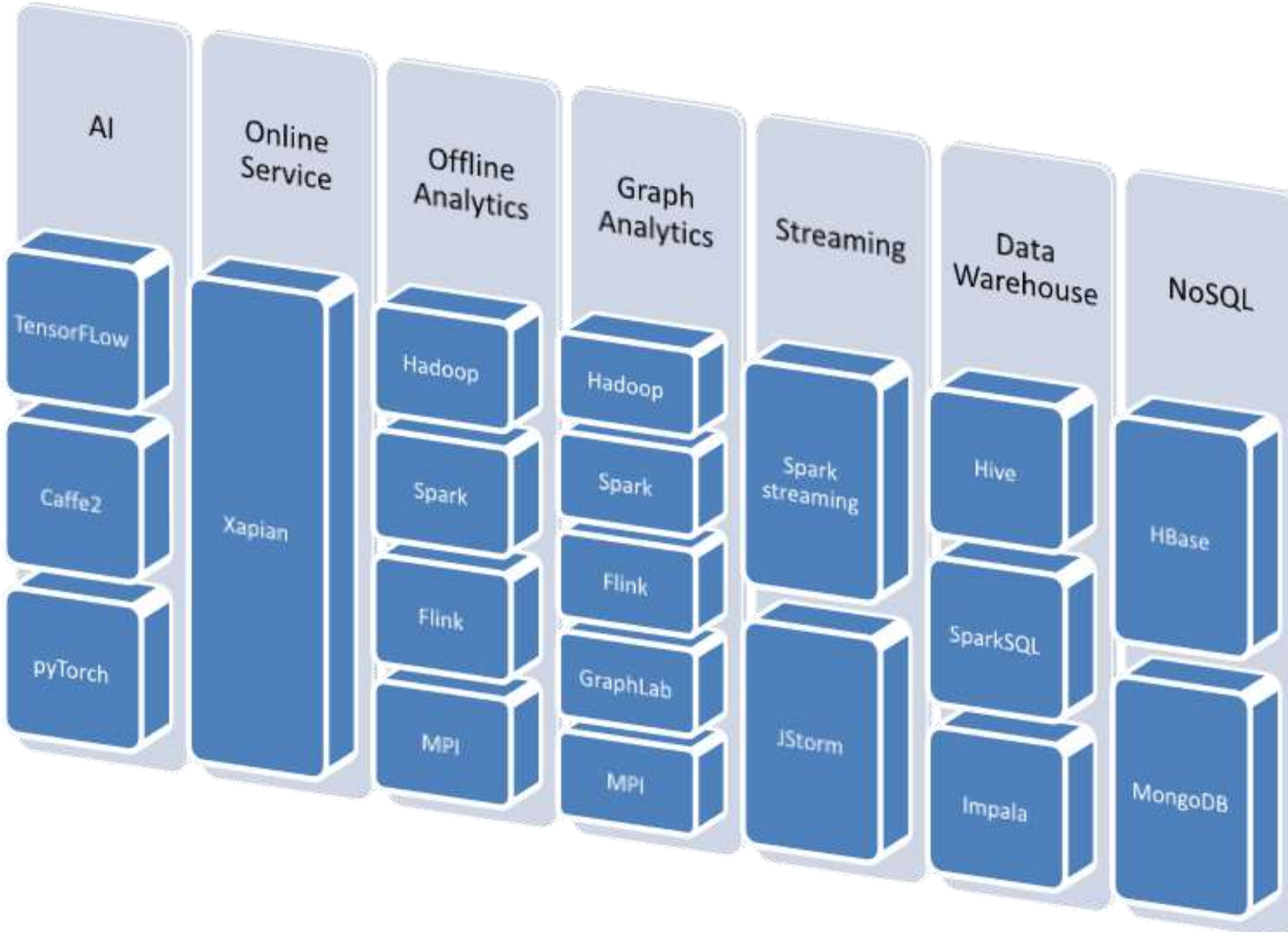
One Combination Example

- Feature extraction – SIFT Workload



Several dwarfs: ***transform*** computations(FFT, IFFT), ***sampling*** computations(downsampling), ***matrix*** computations(matrix multiplication/subtraction), ***sort*** computations(sort), ***basic statistic*** computations(count)

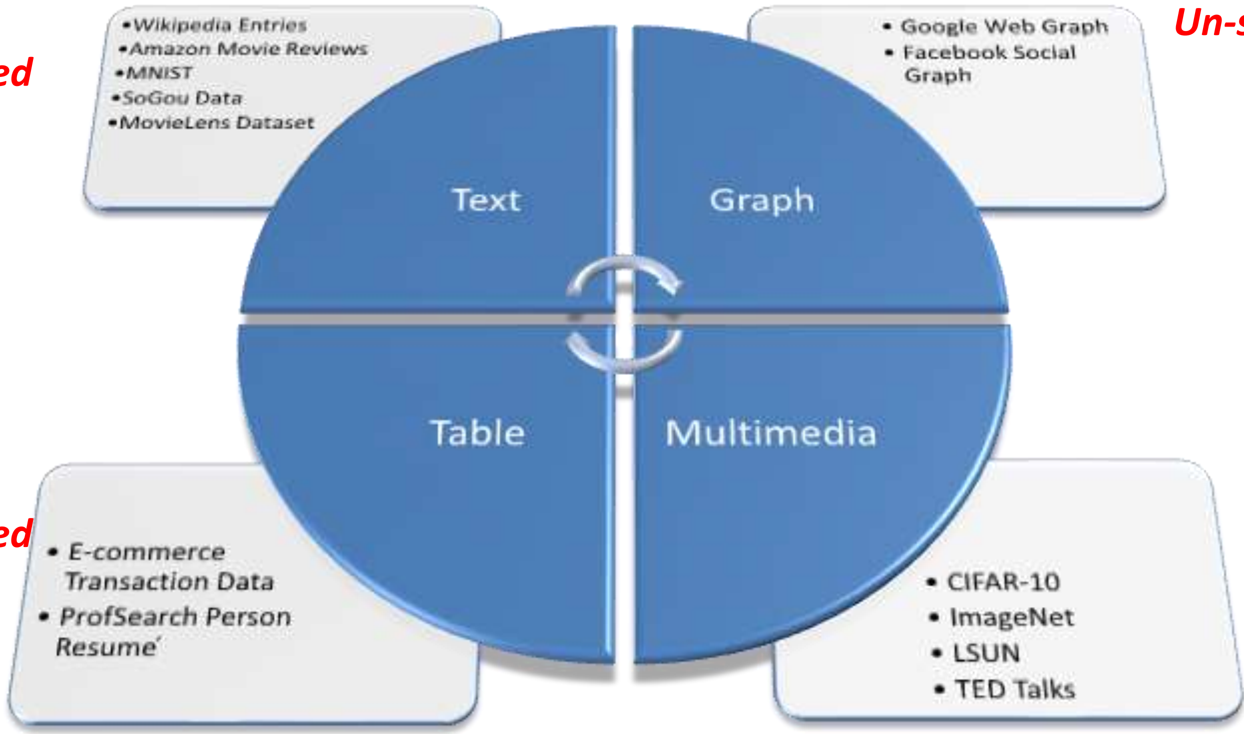
Software Stacks



BigDataBench 4.0 - Dataset

Un-structured
Semi-structured

Un-structured



Structured
Semi-structured

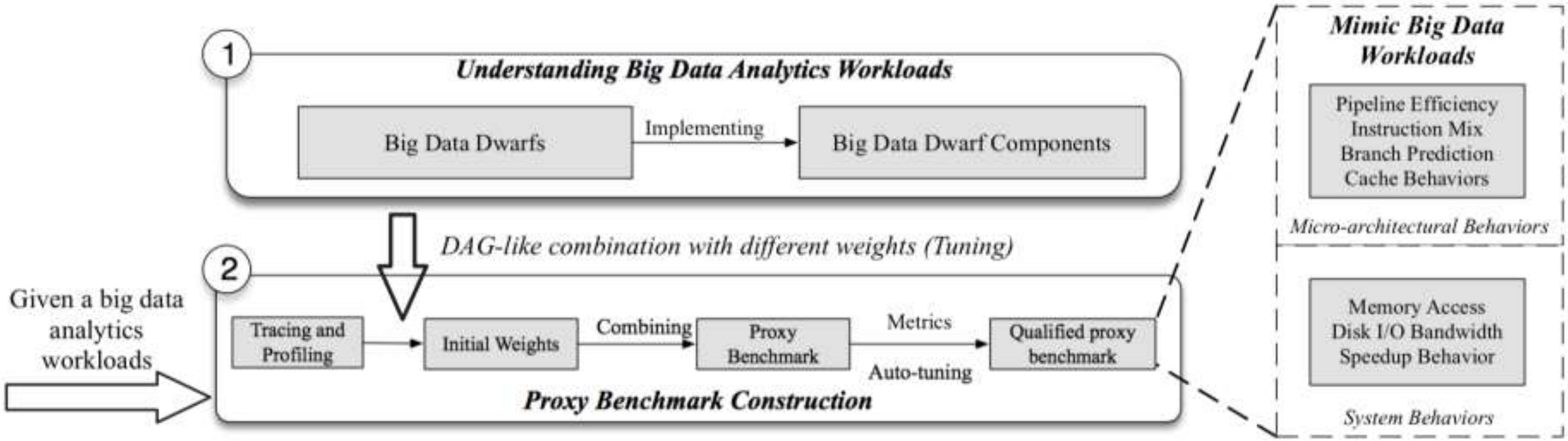
Un-structured

Dwarf-based Simulation

- Simulation Challenge for Big Data and AI Workloads
  - Complex software stacks – limited support of simulators
  - Long running time – Several weeks even months
- A light-weigh simulation benchmark on the basis of big data dwarfs (*OpenMP & Pthreads*)
  - Provide a unified memory management module
  - Shorten the simulation time by <u>100s times</u>
  - Average micro-architectural <u>data accuracy is above 90%</u> on X86 and ARMv8 processors

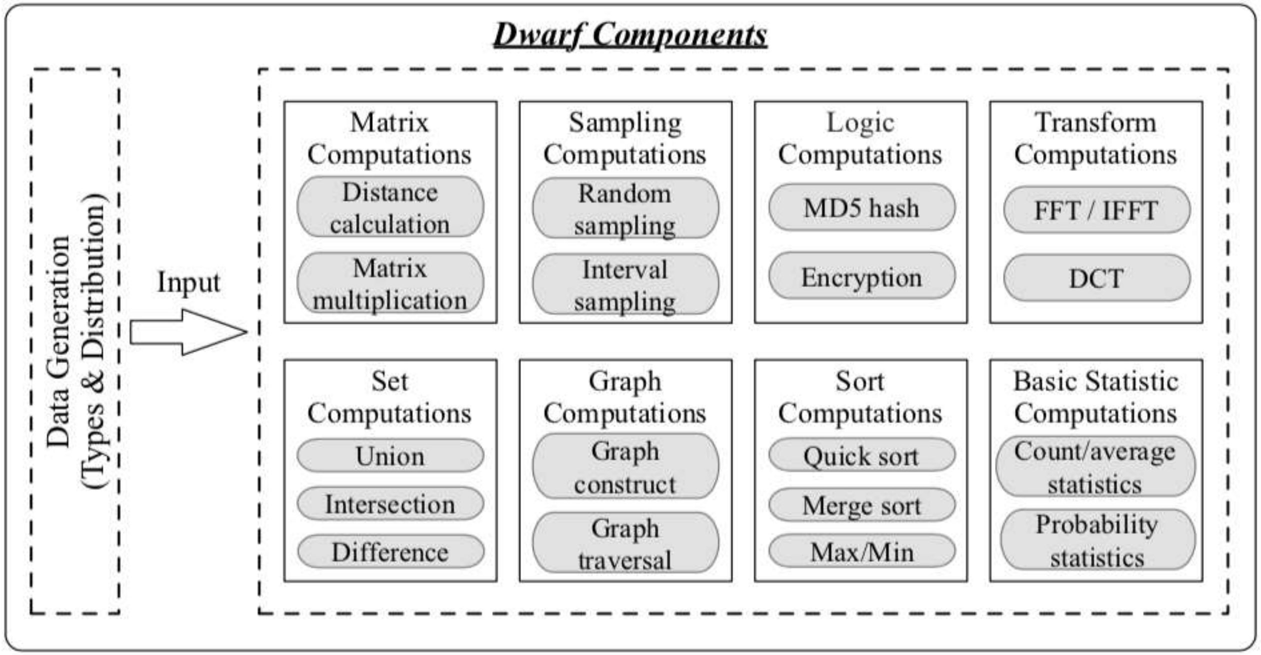
Dwarf-based Simulation Methodology

- DAG-like combinations of dwarfs
  - Different weights
  - Computation logic



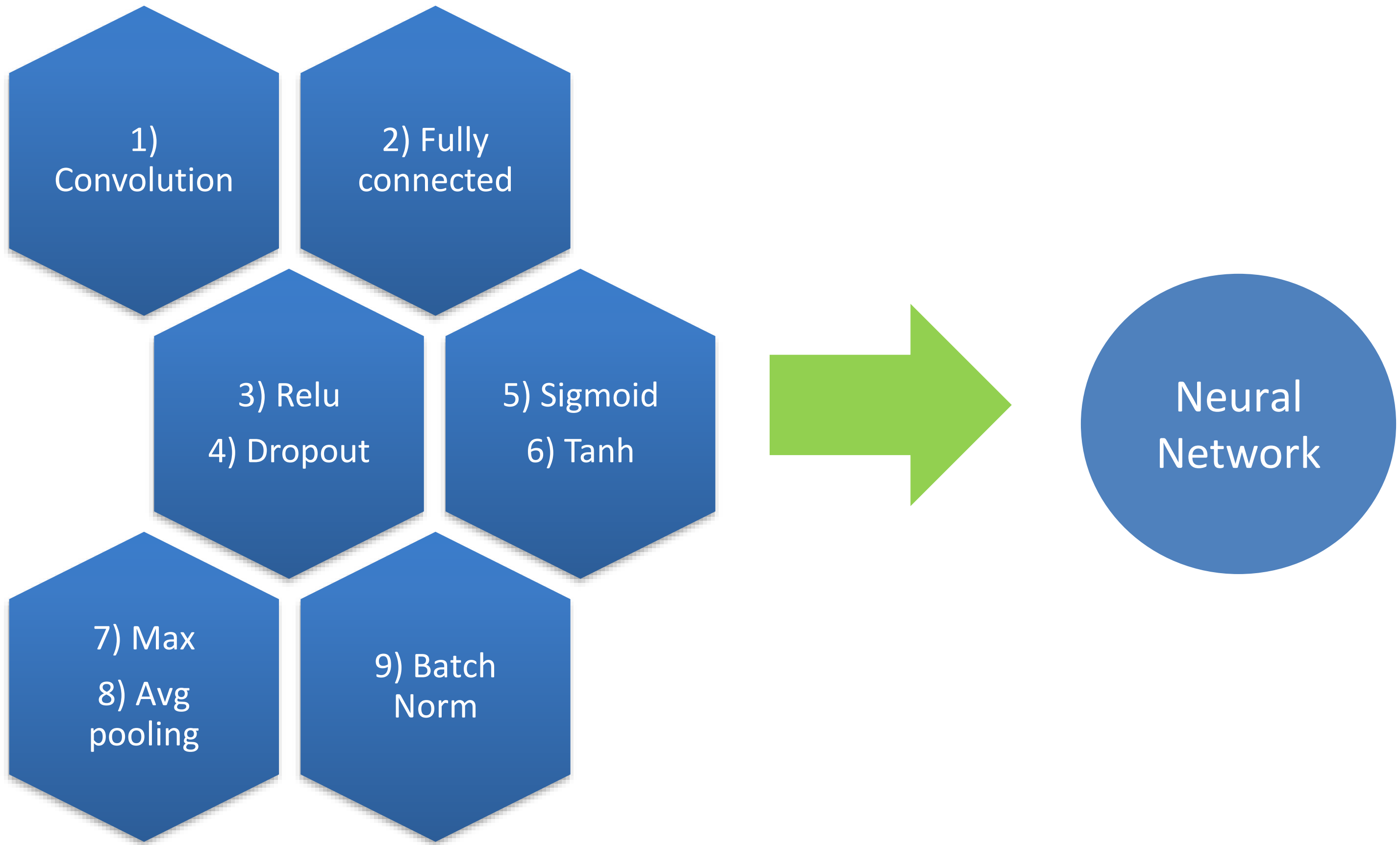
Simulation for Big Data

- Data generation tools
- Dwarf implementations (OpenMP & Pthreads)



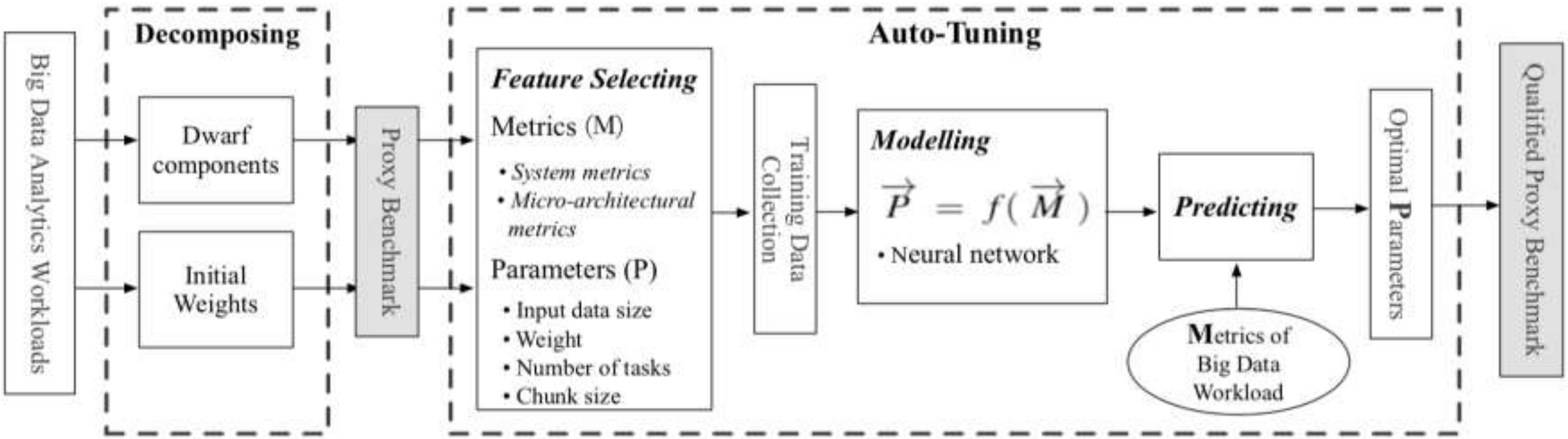
Simulation for AI

- Dwarf implementations (OpenMP & Pthreads)



Dwarf Combination & Tuning

- Modelling and Tuning
  - Metric: $\overrightarrow{M} = (\,runtime,\ IPC,\ MIPS,\ L1D\ hitR,\ L2\ hitR,\ \ldots\ldots)$
  - Parameter: $\overrightarrow{P} = (\,dataSize,\ chunkSize,\ numTasks,\ weight\,)$



Methodology Comparison

- Traditional simulation methodology
  - Kernel benchmark
  - Synthetic trace
  - Synthetic benchmark

Methodology	Typical Benchmark	Input Data	Different Micro-architecture	Multi-core Scalability	System Evaluation	Accuracy
Kernel Benchmark	NPB [17]	Fixed	Recompile	Yes	Yes	Low
Synthetic Trace	SimPoint [19]	Fixed	Regenerate	No	No	High
Synthetic Benchmark	PerfProx [20]	Fixed	Regenerate	No	No	High
Dwarf-Based Proxy Benchmark	Dwarf Benchmark	On-demand	Auto-tuning	Yes	Yes	High

Dwarf Benchmarks

- Four representative big data workloads

Big Data Benchmark	Workload Patterns	Data Set	Involved Dwarfs	Involved Dwarf Components
Hadoop TeraSort	I/O Intensive	Text	Sort computations Sampling computations Graph computations	Quick sort; Merge sort Random sampling; Interval sampling Graph construction; Graph traversal
Hadoop Kmeans	CPU Intensive	Vectors	Matrix computations Sort computations Basic Statistic	Vector euclidean distance; Cosine distance Quick sort; Merge sort Cluster count; Average computation
Hadoop PageRank	Hybrid	Graph	Matrix computations Sort computations Basic Statistic	Matrix construction; Matrix multiplication Quick sort; Min/max calculation Out degree and in degree count of nodes
Hadoop SIFT	CPU Intensive Memory Intensive	Image	Matrix computations Sort computations Sampling computations Transform computations Basic Statistic	Matrix construction; Matrix multiplication Quick sort; Min/max calculation Interval sampling FFT/IFFT Transformation Count Statistics

Overview

- *Dwarf-based Benchmarking Methodology*

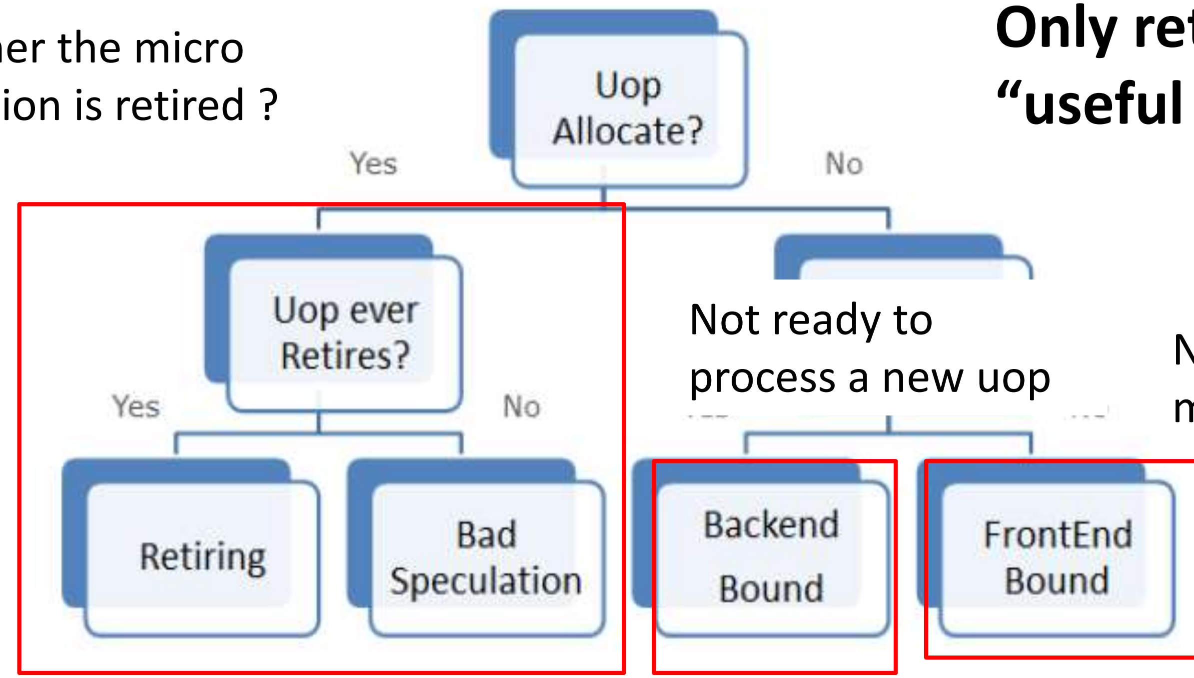
- *Workload Characterization*

Top-Down Method

- Issue point as the dividing point

Whether the micro operation is retired ?

Only retiring is **"useful work"**

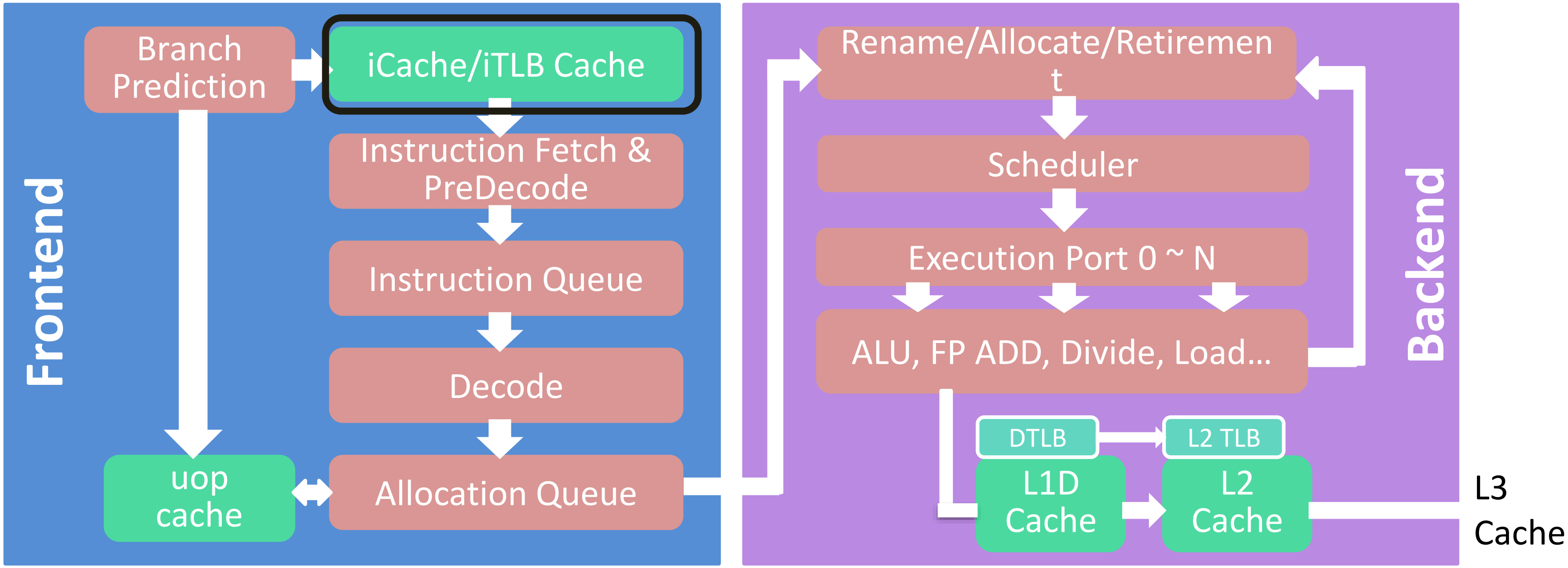


Not ready to process a new uop

Not ready with more uops

From "A Top-Down Method for Performance Analysis and Counters Architecture"

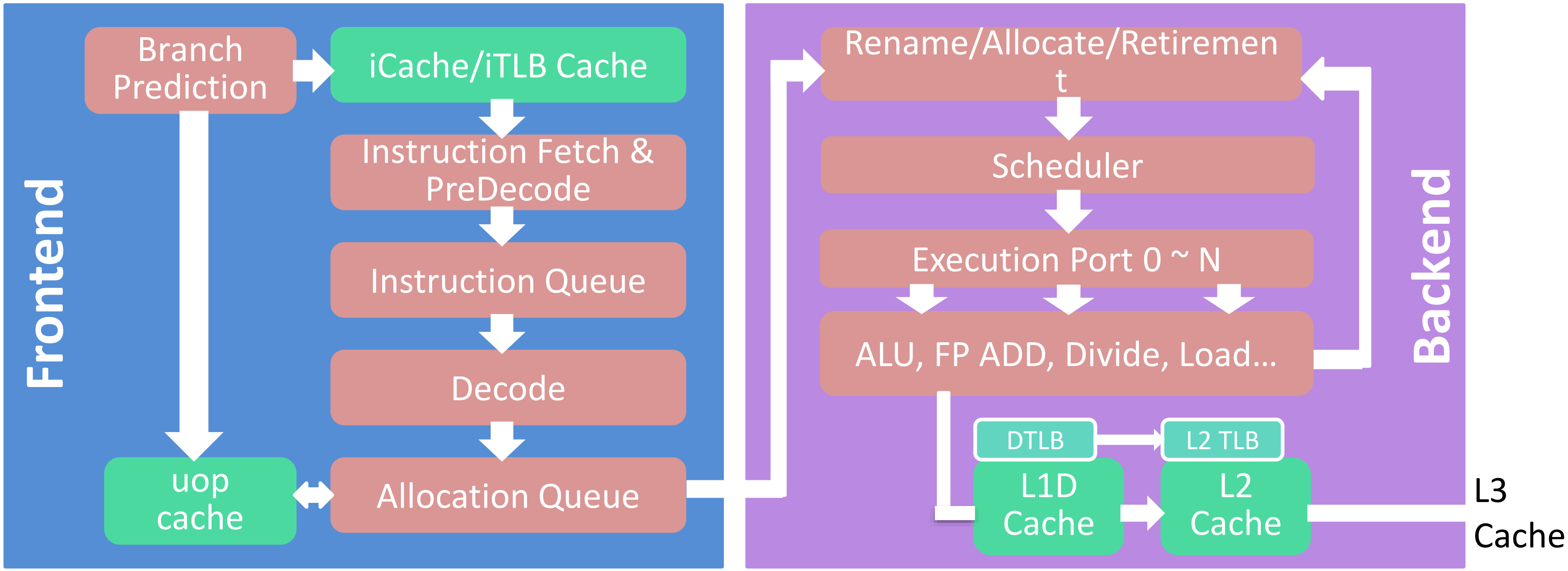
Modern Processor Architecture



- Frontend bound
  - Frontend latency bound
    - Frontend delivers no uops in a cycle, while the Backend was ready to consume them

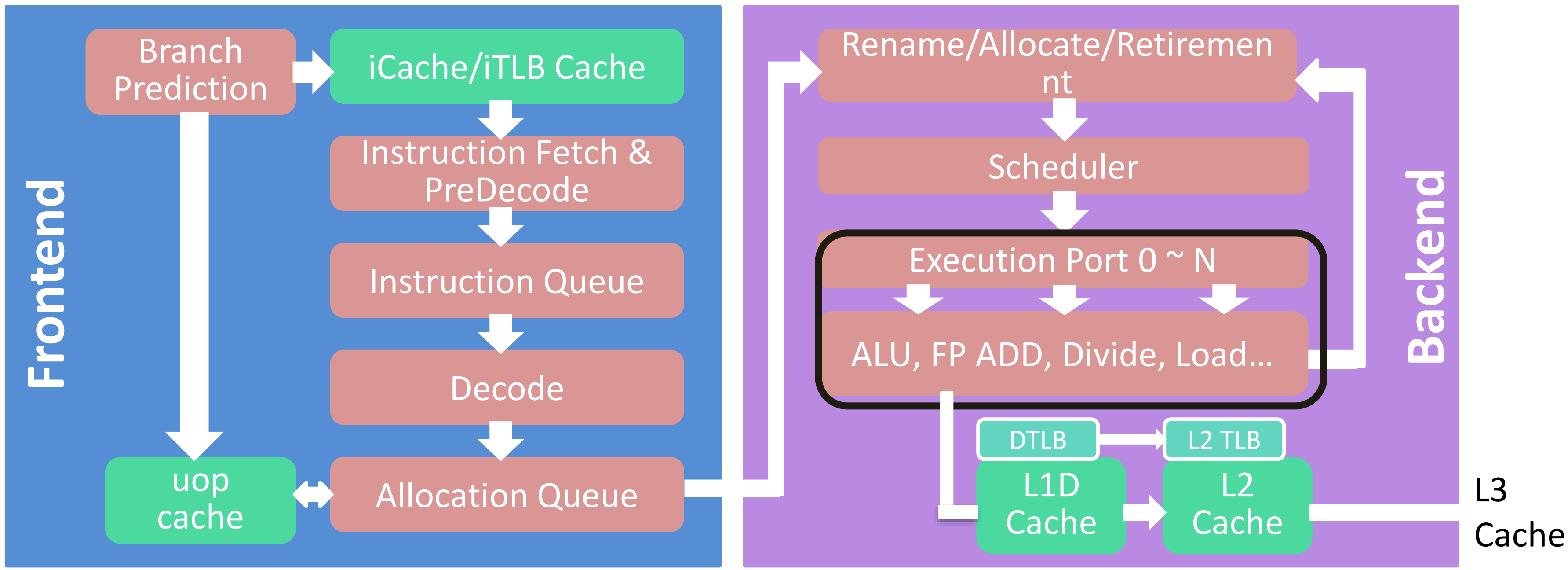
iCache miss/iTLB miss

Modern Processor Architecture



- Frontend bound
  - Frontend bandwidth bound
    - Issued uops less than theoretical value (4 for Sandy Bridge) in a cycle, representing an inefficient use of the Frontend's capability

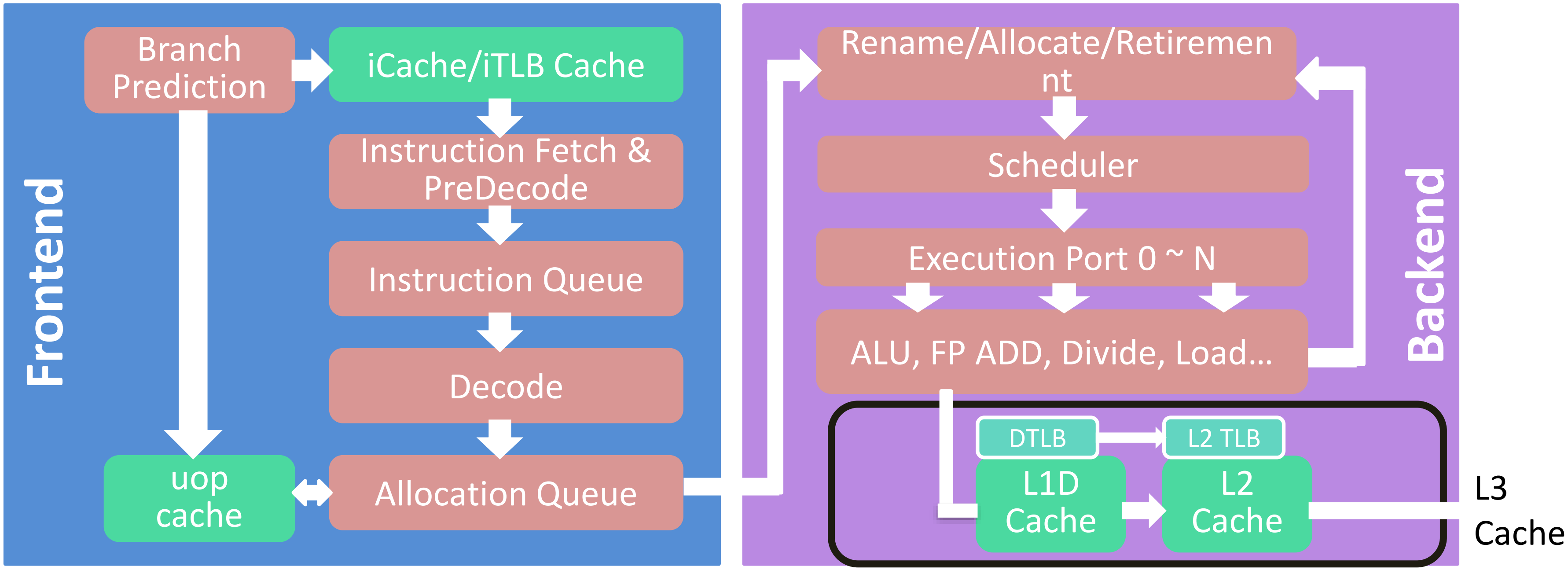
Modern Processor Architecture



- Backend bound
  - Core bound
    - Cycles the back end is bound on core non-memory issues (i.e. Out of Order (OOO) resource and execution)

Port under-utilization
Divider bound

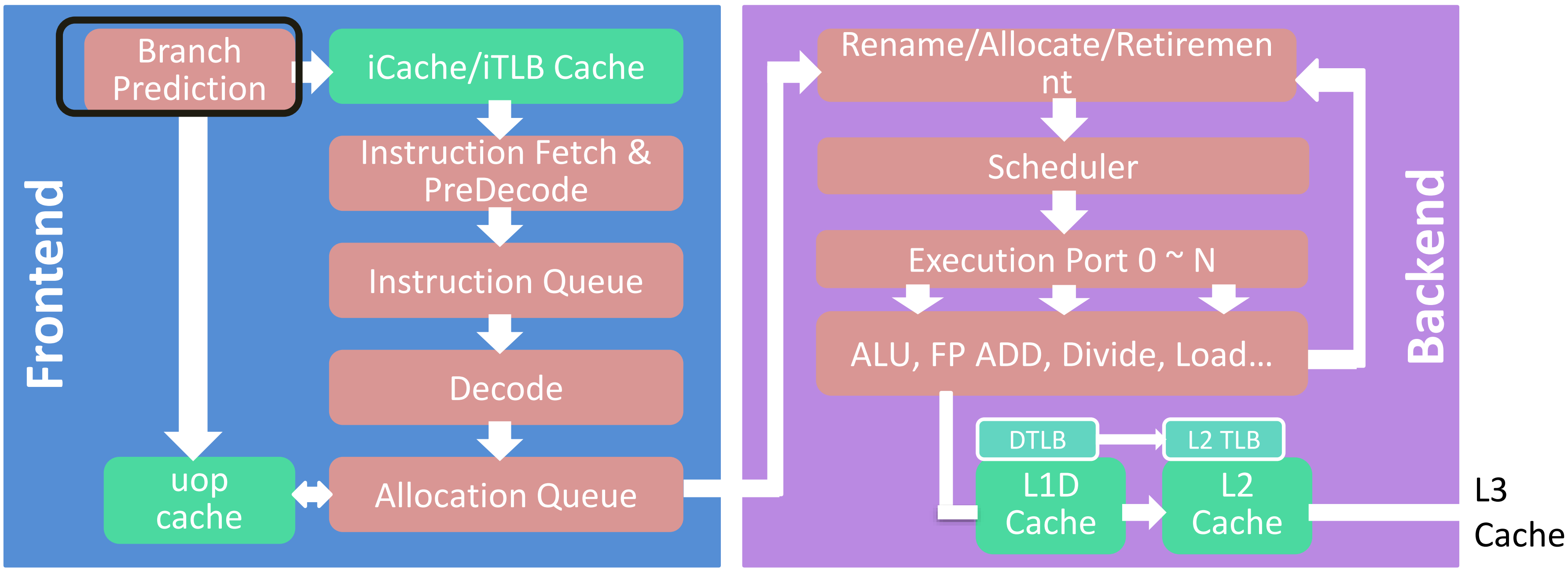
Modern Processor Architecture



- Backend bound
  - Memory bound
    - bound in the memory hierarchy

Data cache misses

Modern Processor Architecture



- Bad speculation
  - cycles wasted because of incorrect predictions

Configurations

Hardware

- 3-node Hadoop cluster
  - Network: 1 Gb Ethernet network
  - Processor: Intel Xeon E5-2620 v3 (Haswell)

Hardware Configurations			
CPU Type		Intel CPU Core	
Intel ®Xeon E5-2620 V3		12 cores@2.40G	
L1 DCache	L1 ICache	L2 Cache	L3 Cache
12 × 32 KB	12 × 32 KB	12 × 256 KB	15MB
Memory		64GB,DDR4	
Disk		SATA@7200RPM	
Ethernet		1Gb	
Hyper-Threading		Disabled	

Software Configurations	
Operating System	CentOS 7.2
Linux Kernel	4.1.13
JDK Version	1.8.0_65
Hadoop Version	2.7.1
Hive Version	0.9.0
HBase Version	1.0.1
Spark Version	1.5.2
Tensorflow Version	1.0

Software

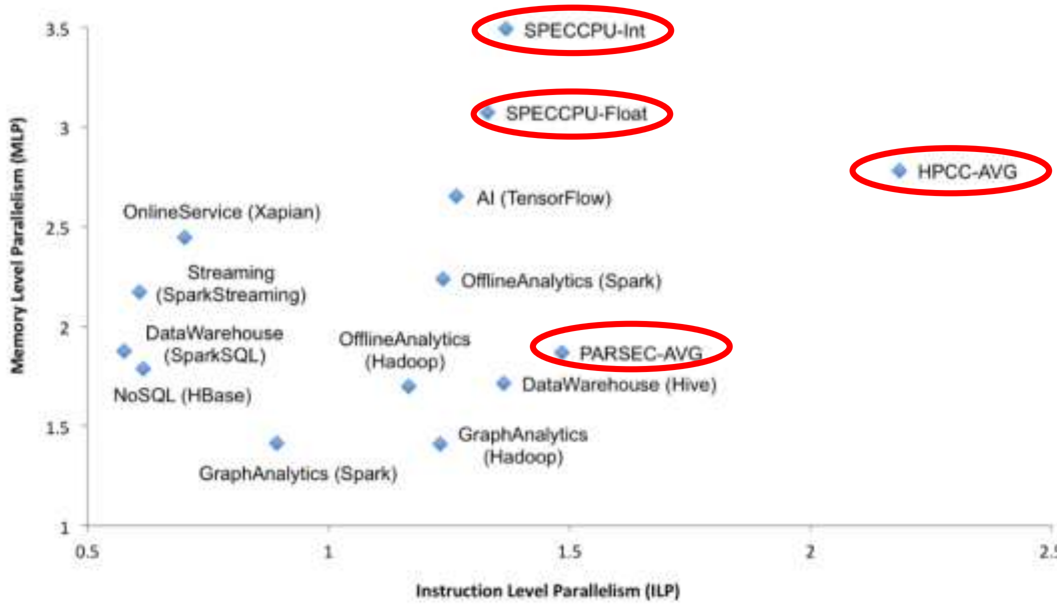
- Software version
  - CentOS 7.2, Kernel 4.1.13
  - JDK version: 1.8.0_65
  - Hadoop version: 2.7.1
- Compared benchmarks
  - SPEC CPU2006
  - HPCC 1.4.0
  - PARSEC 2.0

Benchmark

- Seven workload types

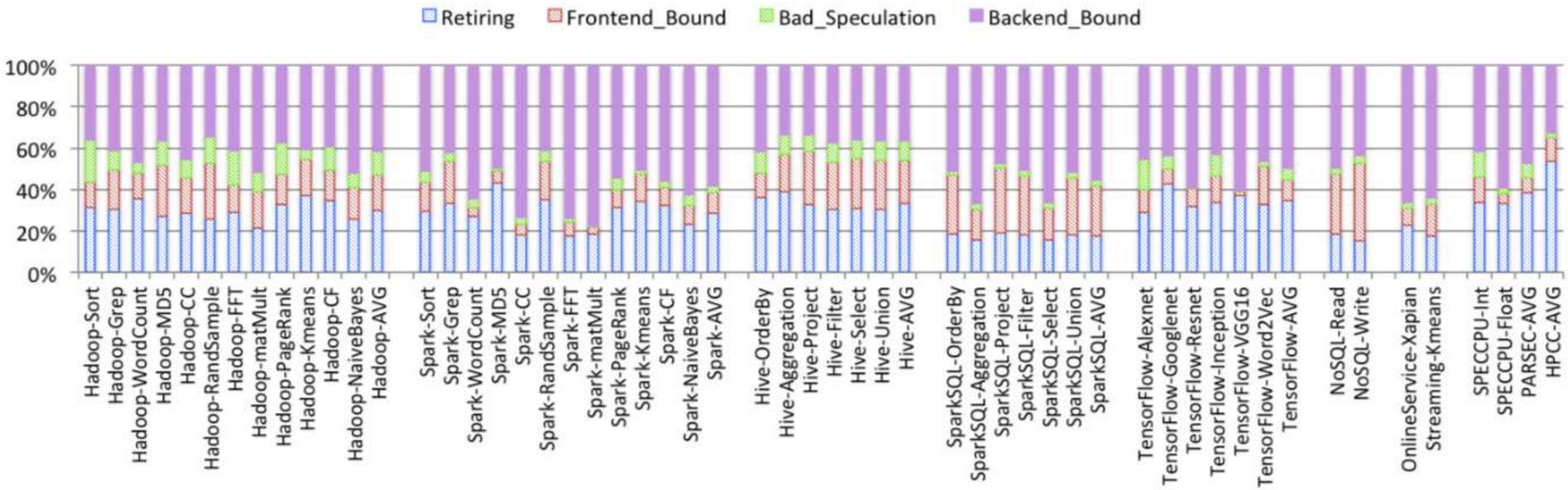
Execution Performance

- ILP and MLP
  - AI: ILP slightly lower than SPECCPU, MLP similar with HPCC
  - Big data has lower ILP and MLP than AI for almost all types, except Hive based data warehouse type



Pipeline Efficiency (Level 1)

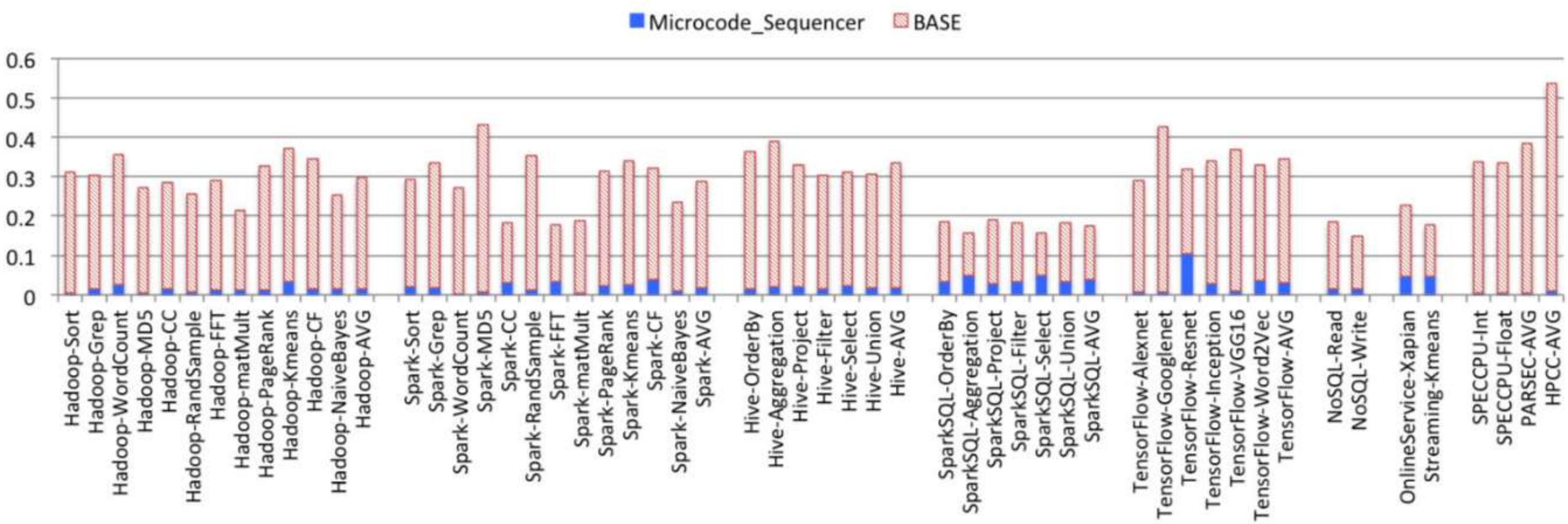
- AI reflect similar pipeline behaviors with the traditional benchmarks
  - retiring (35% v.s. 39.8%), bad speculation (6.3% v.s. 6.1%), frontend bound (both about 9%), and backend bound (49.7% v.s. 45.1%)
- Big data and AI have a small fraction of bad speculation



Offline analytics & Graph analytics (Hadoop, Spark) **Data warehouse (Hive, Spark sql)** **AI** **NoSQL, Online service, Streaming**

Retiring Breakdown (Level 2)

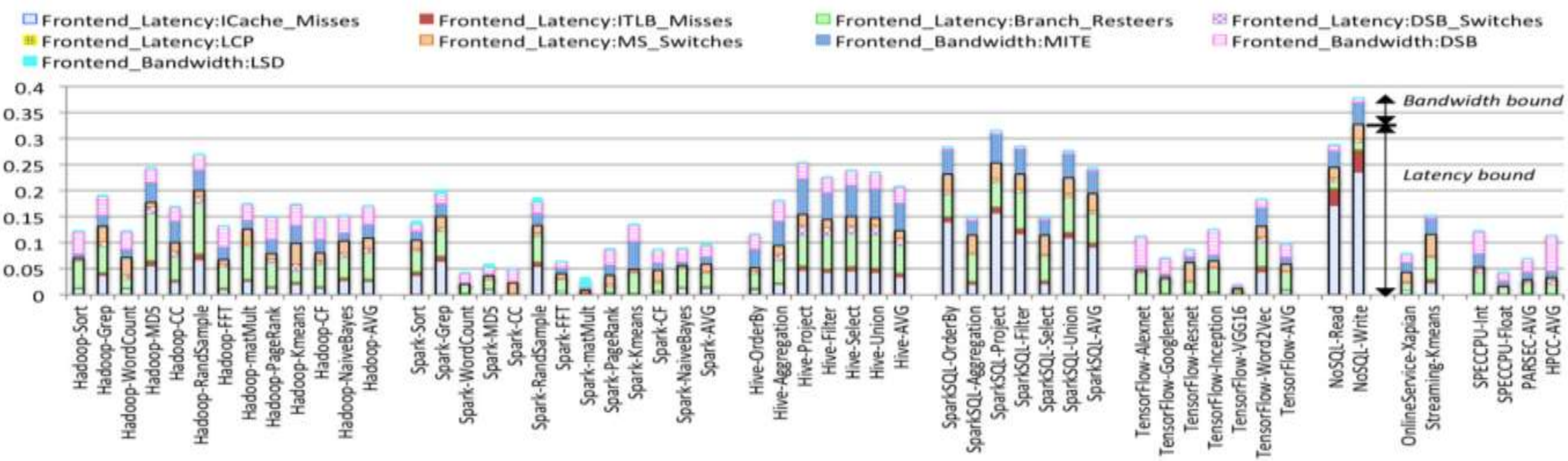
- Retiring instructions from microcode sequencer (MS) unit are about 10 times larger than that of traditional benchmarks
  - Incurring notable penalties due to MS switches and further hurts performance



Offline analytics & Graph analytics (Hadoop, Spark) | **Data warehouse (Hive, Spark sql)** | **AI** | **NoSQL, Online service, Streaming**

Frontend Bound Breakdown (Level 2&3)

- Big data Benchmark: More frontend bound than the traditional benchmarks
- AI Benchmark: nearly equal frontend bound with the traditional benchmarks
- Frontend bound percentage varies across different workload types
  - NoSQL (35%), data warehouse (25%), the others (15%) on average



Offline analytics & Graph analytics (Hadoop, Spark)

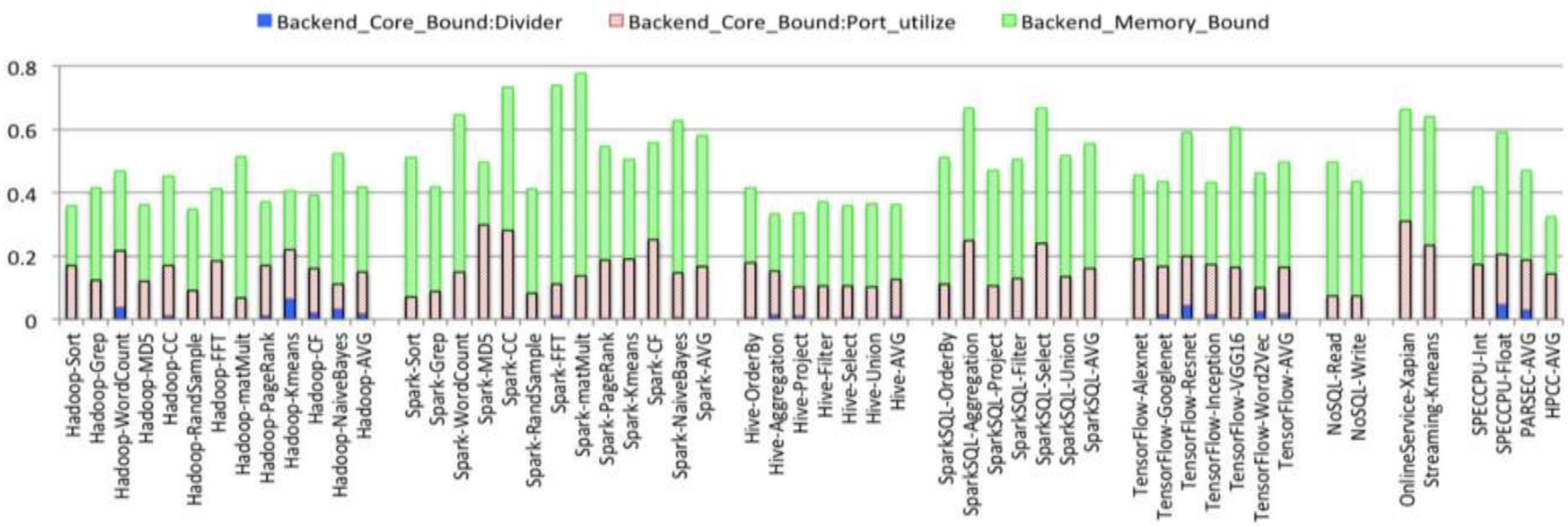
Data warehouse (Hive, Spark sql)

AI

NoSQL, Online service, Streaming

Backend Bound Breakdown (Level 2&3)

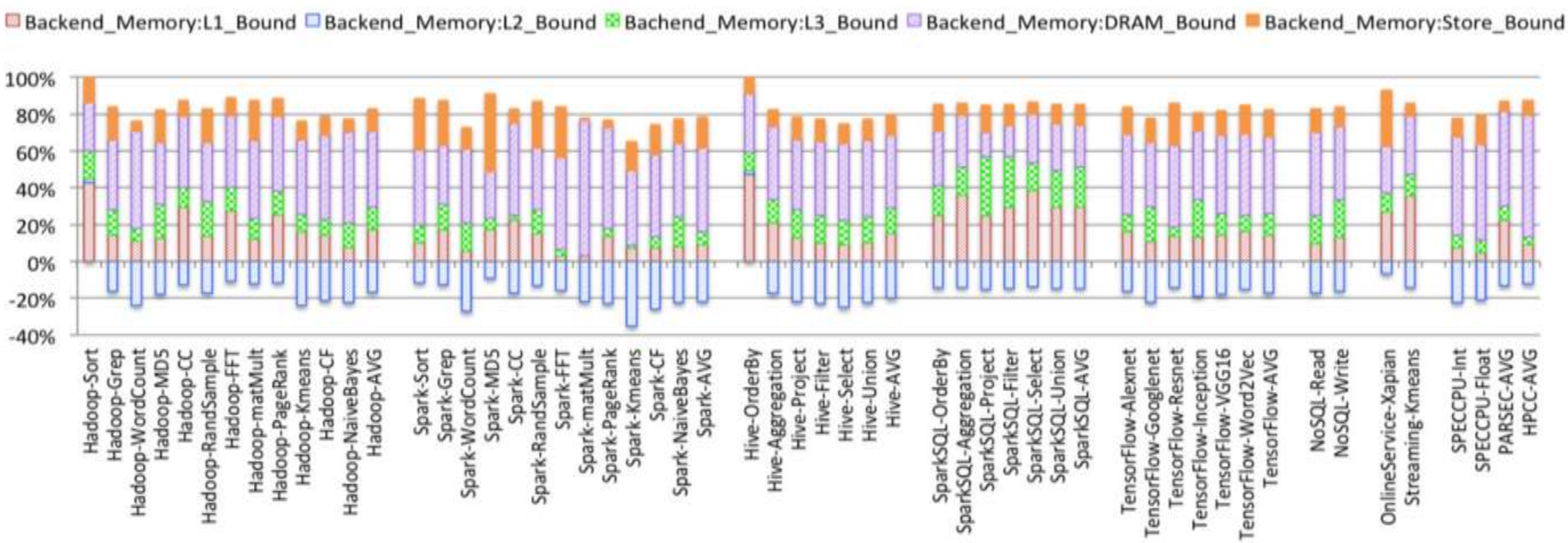
- Memory bound is more severe than core bound
  - Except online service (nearly equal core and memory bound)



Offline analytics & Graph analytics (Hadoop, Spark) **Data warehouse (Hive, Spark sql)** **AI** **NoSQL, Online service, Streaming**

Backend Memory Bound (Level 3)

- Mainly DRAM bound for big data and AI benchmarks
- More stalls due to L1 Bound, L3 Bound and Store Bound than traditional benchmarks



Offline analytics & Graph analytics (Hadoop, Spark)

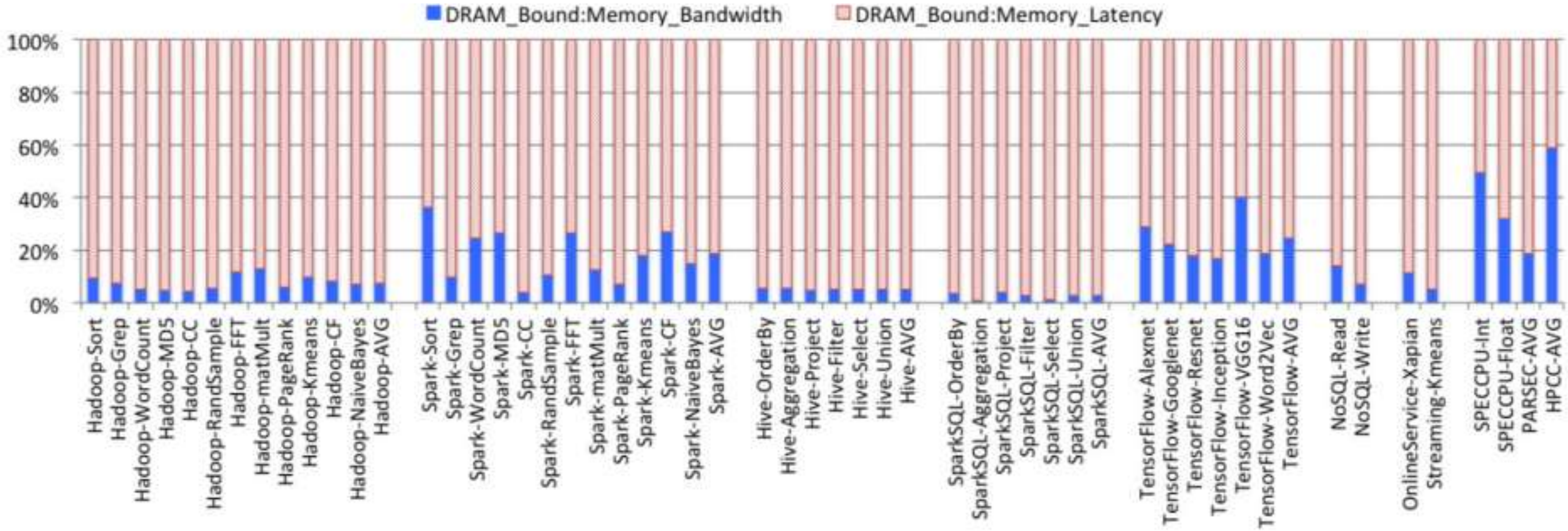
Data warehouse (Hive, Spark sql)

AI

NoSQL, Online service, Streaming

DRAM Bound Breakdown (Level 4)

- First bottleneck: DRAM latency bound
- DRAM bandwidth: AI more than big data



Offline analytics & Graph analytics (Hadoop, Spark)

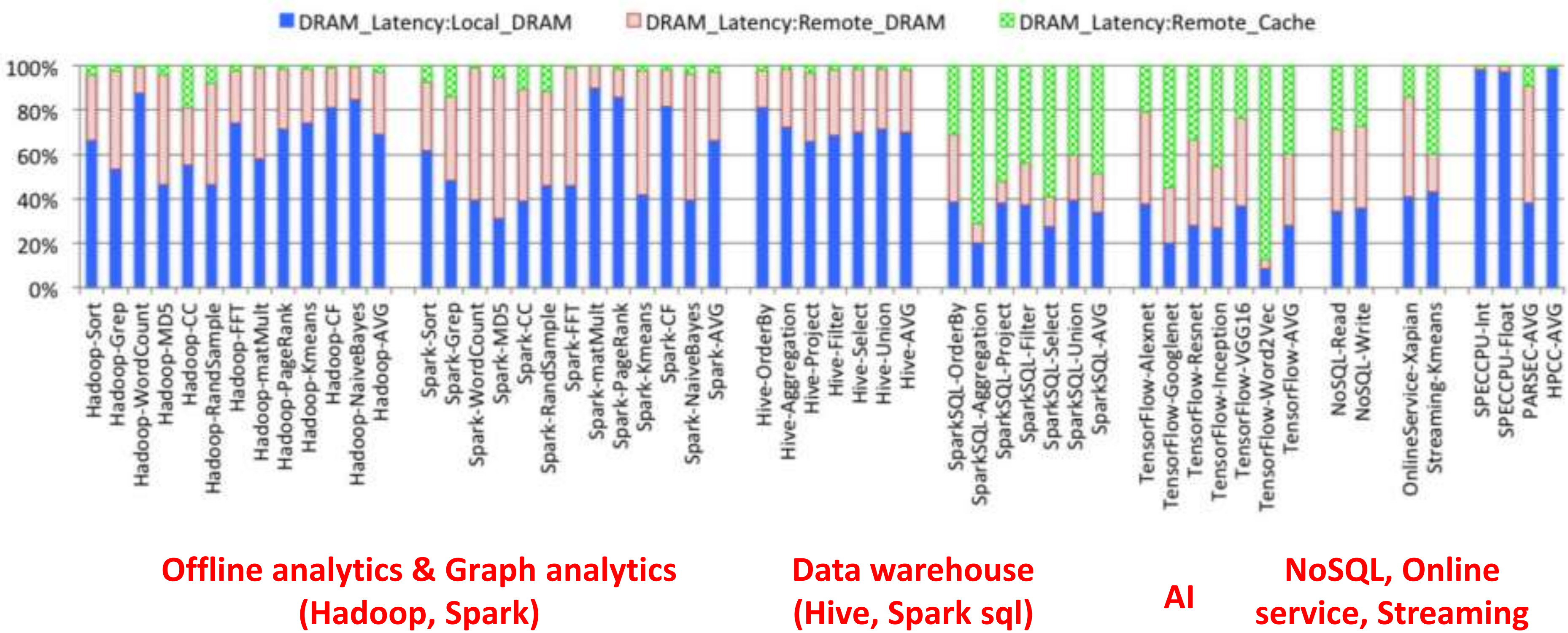
Data warehouse (Hive, Spark sql)

AI

NoSQL, Online service, Streaming

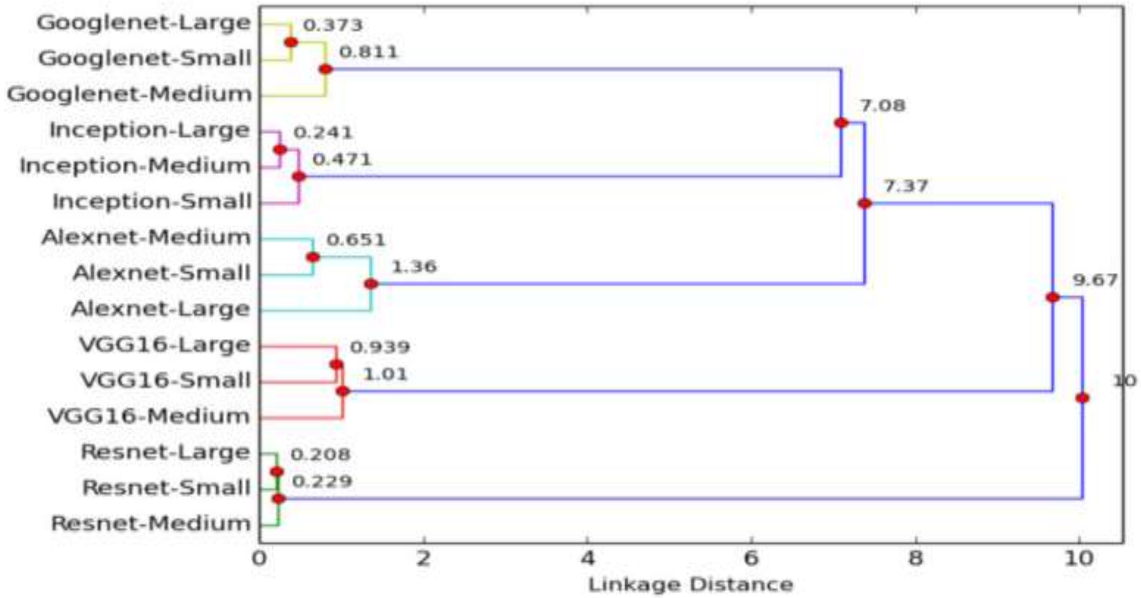
DRAM Latency Bound (Level 5)

- The main bottleneck varies with workload types and software stacks



Iteration Number Impact on AI

- Measure the similarity
  - PCA and hierarchical clustering using fifty micro-architectural metrics
  - Smaller distance means the higher similarity
- A small number of iterations is enough



Summary

- Big Data Benchmark
  - Different workload types reflect diverse pipeline behaviors
  - Backend bound is the first bottleneck
  - Frontend bound is the second bottleneck
    - Frontend bound percentages vary across different workload types and software stacks
  - Software stacks and algorithms both have great impacts on pipeline behaviors

Summary (Cont')

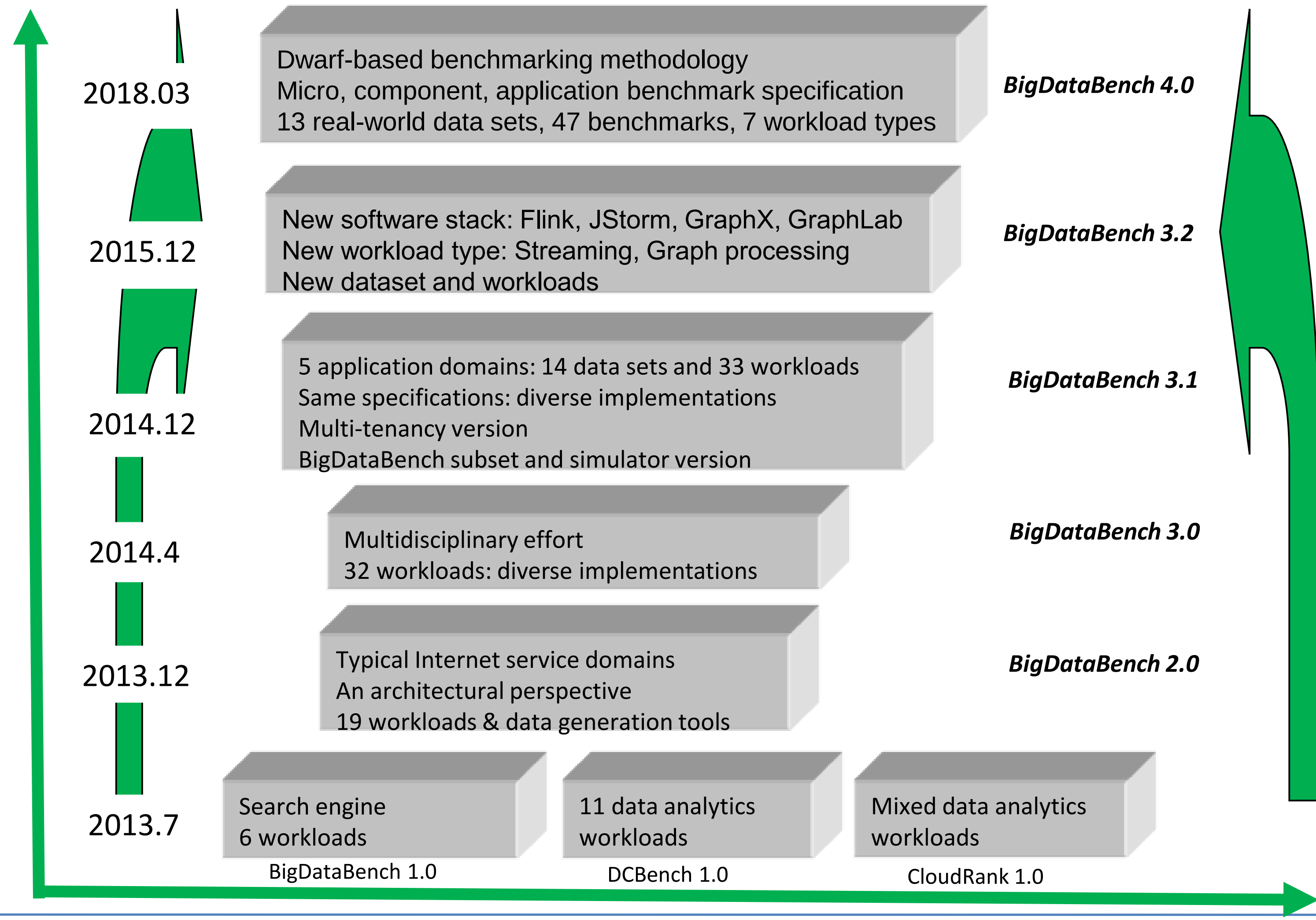
- AI Benchmark
  - Higher retiring percentage than big data
  - Backend bound is the first bottleneck for AI
  - Frontend bound is not always the second bottleneck
  - Neural network structures have a great impact on pipeline behaviors, while iteration number has little impact

Conclusion

- BigDataBench 4.0
  - An open source dwarf-based big data and AI benchmark suite

- Website: http://prof.ict.ac.cn
- Technical Reports:
  - https://arxiv.org/pdf/1802.08254.pdf
  - https://arxiv.org/pdf/1801.09212.pdf

BigDataBench Evolution



BigDataBench Users

- http://prof.ict.ac.cn/BigDataBench/users/
- Industry users
  - Accenture, BROADCOM, SAMSUMG, Huawei, IBM
- About 100 academia groups published papers using BigDataBench or citing BigDataBench papers (800+ citations)
  - VLDB/SIGMOD/ICDE, SC, FAST, ASPLOS, OSDI, ISCA/Micro/ HPCA, ICPP and etc.


QUESTIONS
And
Answers