GraphBench: A Benchmark Suite for Graph Computing Systems

Presented by Lei Wang

Institute of Computing Technology, Chinese Academy of Sciences

Bench 2019, Denver, USA

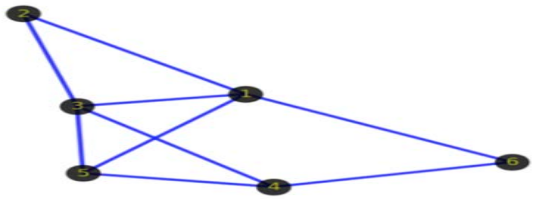
中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

Outline

- **Motivations**
- The GraphBench Benchmark Suite
  - Methodology
  - Basic operations of graph computing
  - The implementations
- Evaluations
- Conclusions

Graph Data and Its Processing

- **Graph Data**
  - ✓ A kind of structural data that defined entities as vertices and described dependencies between different entities as edges.



- **Processing large-scale graph data is a big challenge**
  - ✓ Facebook pushes advertisements to more than nine hundreds million users
  - ✓ The PageRank application of Google determines the index quality of more than one trillion Web pages.

Social Media **Web** **Advertising**

Graph Computing Systems

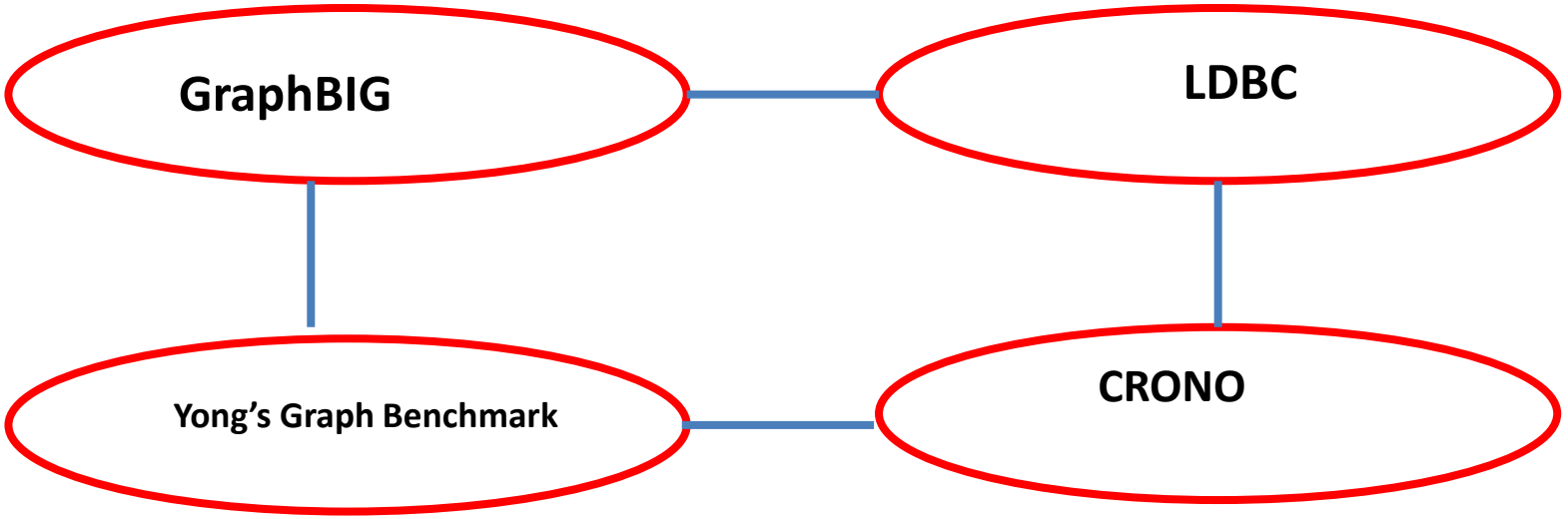
- **Diversity of The Design Pattern**

- **Lots of Implementations**

Pregel
oogle
GraphLab
PowerLyra
GraphX
Gemini
Apache Hama

How to quantitatively evaluate graph computing systems?

State-of-Practice Graph Benchmarks

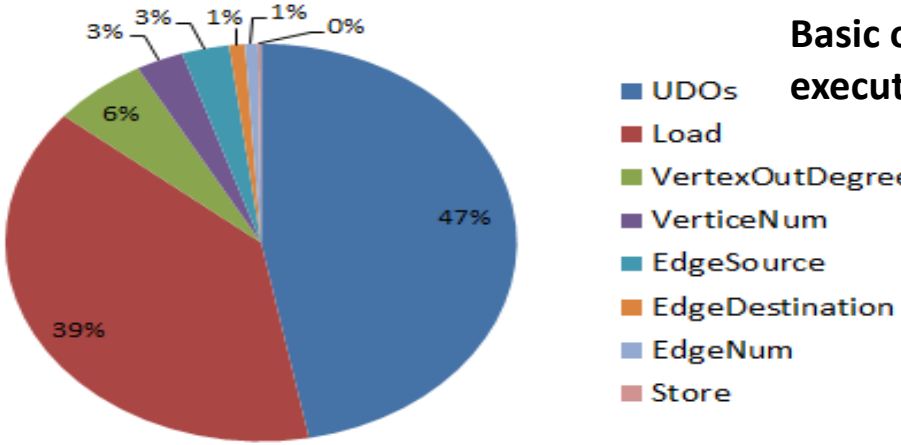


- Existing graph computing benchmarks are all constructed with prevalent graph computing workloads, and take graph computing algorithm workloads as a whole for evaluation.

We cannot fine-grained analyze the graph computing system.

Motivations of the GraphBench

- There are lots of basic operations in the graph computing
  - Loading , Counting the numbers of vertices or edges, and so on.



Basic operations take 53% execution time of the PageRank.

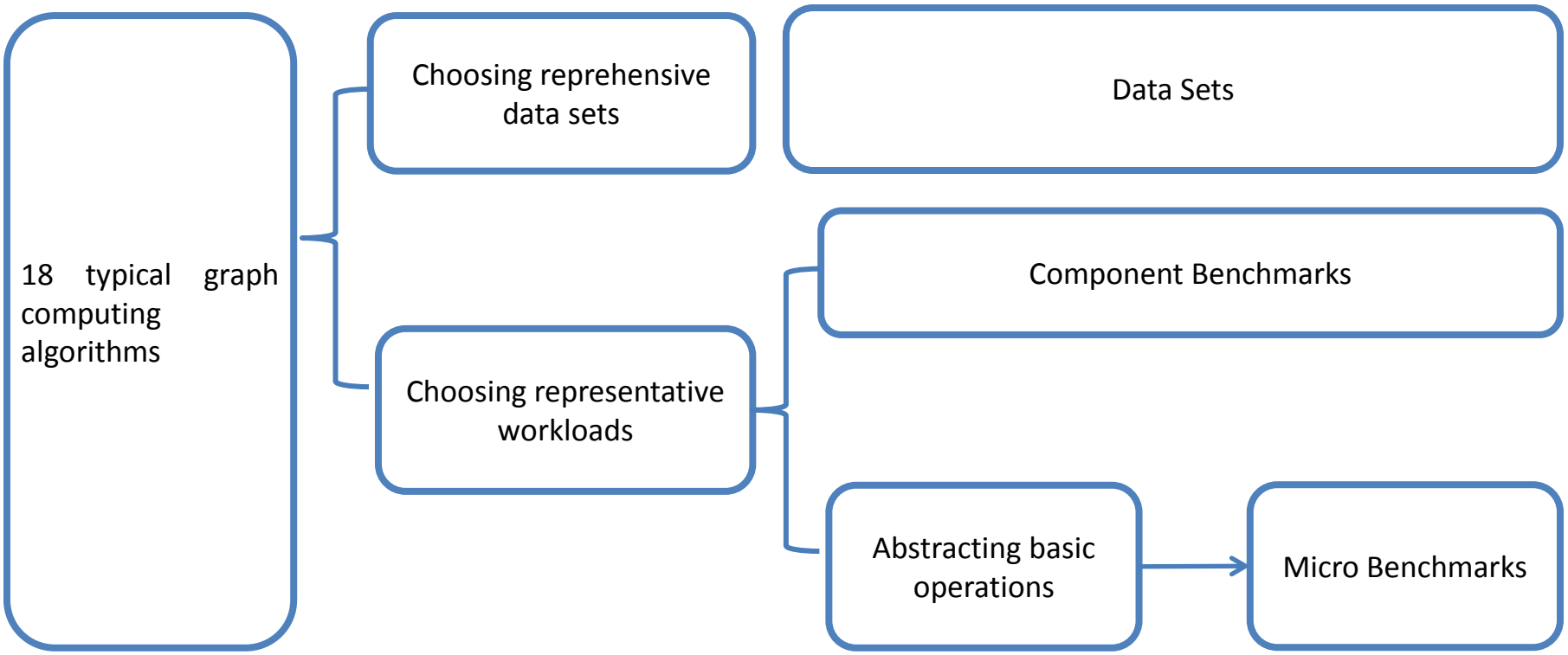
The graph computing workload = GBOs + UDOs

GBOs: graph basic operations, UDOs: user-defined operations

Outline

- Motivations
- **The GraphBench Benchmark Suite**
  - **Methodology**
  - Basic operations of graph computing
  - The implementations
- Evaluations
- Conclusions

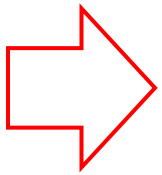
The Methodology of GraphBench



Representative workloads

Single Source Shortest Path
All Pairs Shortest Path
Minimum Spanning Tree
Breadth-First Search
Depth-First Search
the Traveling Salesman Problem
Connected Component
Strongly Connected Component
Weakly Connected Component
Community Detection
Triangle Counting
K-core
Degree Centrality
Closeness Centrality
Betweenness centrality
PageRank
Graph Coloring
Topological Sort

Eighteen typical graph computing algorithms



Single-Source Shortest Path (SSSP) of path planning

Breadth-first search (BFS) of search

Connected Components (CC) of social analysis

K-core (K-core) of network analysis

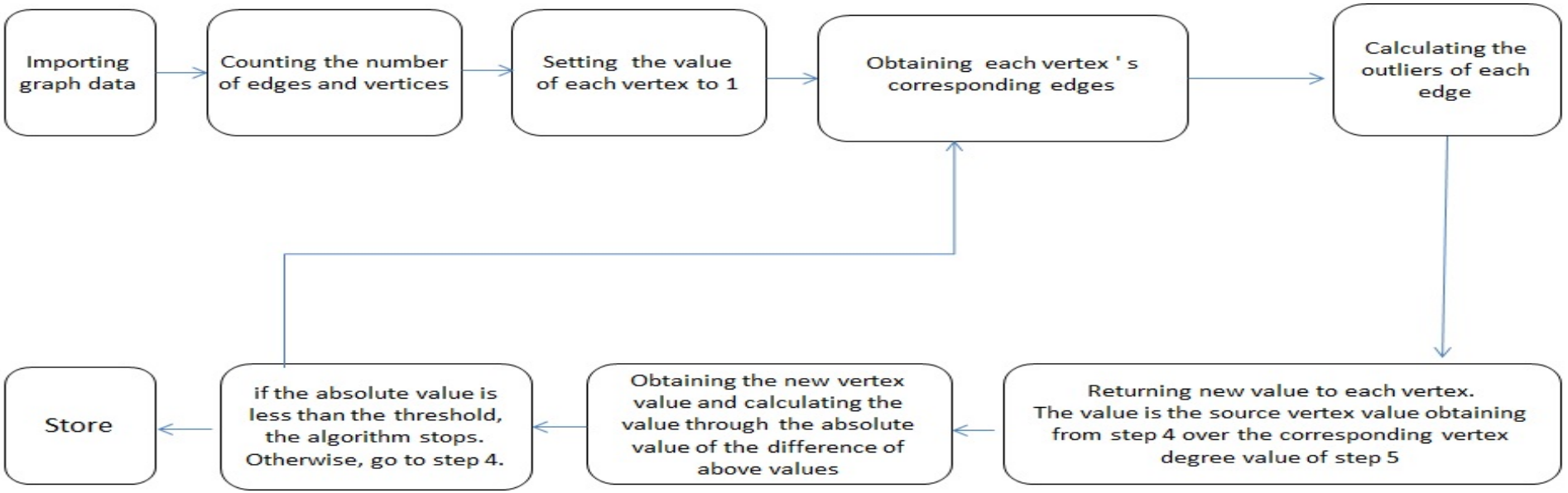
PageRank (PageRank) of graph analysis

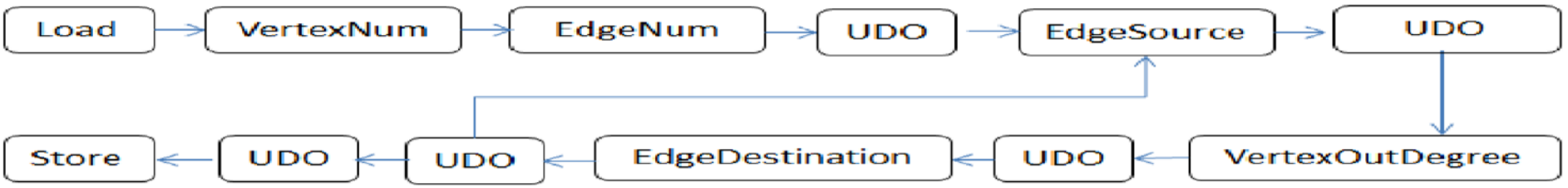
Five Component Benchmarks

Outline

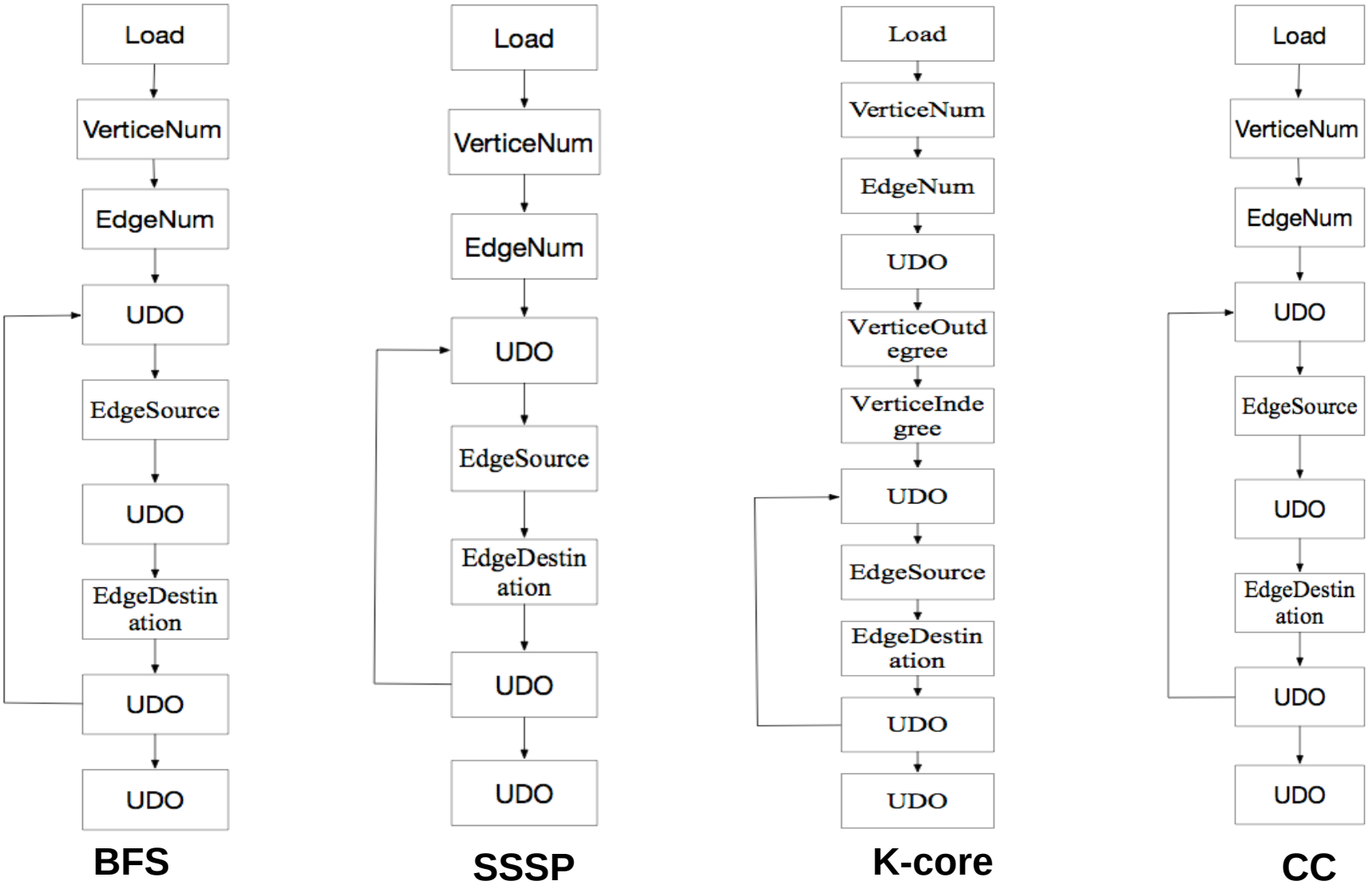
- Motivations
- **The GraphBench Benchmark Suite**
  - Methodology
  - **Basic operations of graph computing**
  - The implementations
- Evaluations
- Conclusions

Basic Operations of the PageRank Workload





Basic Operations of Other Workloads



Basic Operations Summary

1) Loading graph data (**Load**).

Load is the operation that imports the data into memory to build the specific graph data structure.

2) Counting the number of vertices (**VerticeNum**).

VerticeNum is the operation that counts the number of imported vertices of the graph data.

3) Counting the number of edges (**EdgesNum).**

EdgesNum is the operation that counts the number of imported edges of the graph data.

4) Counting the out-degree of the specific vertex (**VerticeOutDegree**).

VerticeOutDegree is the operation that counts the Out-degree of the specific vertex.

5) Counting the in-degree of the specific vertex (**VerticeInDegree)**.

VerticeInDegree is the operation that counts the In-degree of the specific vertex.

6) Obtaining the source vertex of the specific edge (**EdgeSource**).

EdgeSource is the operation that returns the source vertex of the specific edge.

7) Obtaining the destination vertex of the specific edge (**EdgeDestination**).

EdgeDestination is the operation that returns the destination vertex of the specific edge.

8) Storing graph data(**Store**).

Store is the operation that exports the result to the file on the disk.

Outline

- Motivations
- **The GraphBench Benchmark Suite**
  - Methodology
  - Basic operations of graph computing
  - **The implementations**
- Evaluations
- Conclusions

Data Sets

- Considering the power law characteristic of the data.
  - the average clustering coefficient as the metric to evaluate the power law of the graph data.
- Considering graph data structure diversity.
  - the directed graph structure & the un-directed graph structure.

	Graph Structure	Vertices	Edges	Clustering Coefficient
Email	directed	265,214	420,045	0.07
Wikipedia	directed	2,394,385	5,021,410	0.05
Pokec	directed	1,632,803	30,622,564	0.1
Live Journal	un-directed	3,997,962	34,681,189	0.3

The Summary of GraphBench

Types	*Workloads*	*Data Sets*	*Frameworks*
Component	SSSP	Pokec	PowerGraph, PowerLyra, Gemini, GraphX, C++
	BFS	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	CC	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	K-core	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	PageRank	Wikipedia	PowerGraph, PowerLyra, Gemini, GraphX, C++
Micro	Load	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	VerticeNum	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	EdgesNum	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++
	VertexOutDegree	Email	PowerGraph, PowerLyra, Gemini, GraphX, C++
	VertexInDegree	Email	PowerGraph, PowerLyra, Gemini, GraphX, C++
	EdgeSource	Email	PowerGraph, PowerLyra, Gemini, GraphX, C++
	EdgeDestination	Email	PowerGraph, PowerLyra, Gemini, GraphX, C++
	Store	LJ	PowerGraph, PowerLyra, Gemini, GraphX, C++

Graph Benchmarks Comparison

Benchmarks	Workloads	Workload types	Software stacks
GraphBench	**13**	**Component+Micro**	**5**
CRONO	10	Component	1
GraphBIG	13	Component	1
LDBC	6	Component	2
Yong's Graph Benchmark	3	Component	3

Outline

- Motivations
- The GraphBench Benchmark Suite
  - Methodology
  - Basic operations of graph computing
  - The implementations
- **Evaluations**
- Conclusions

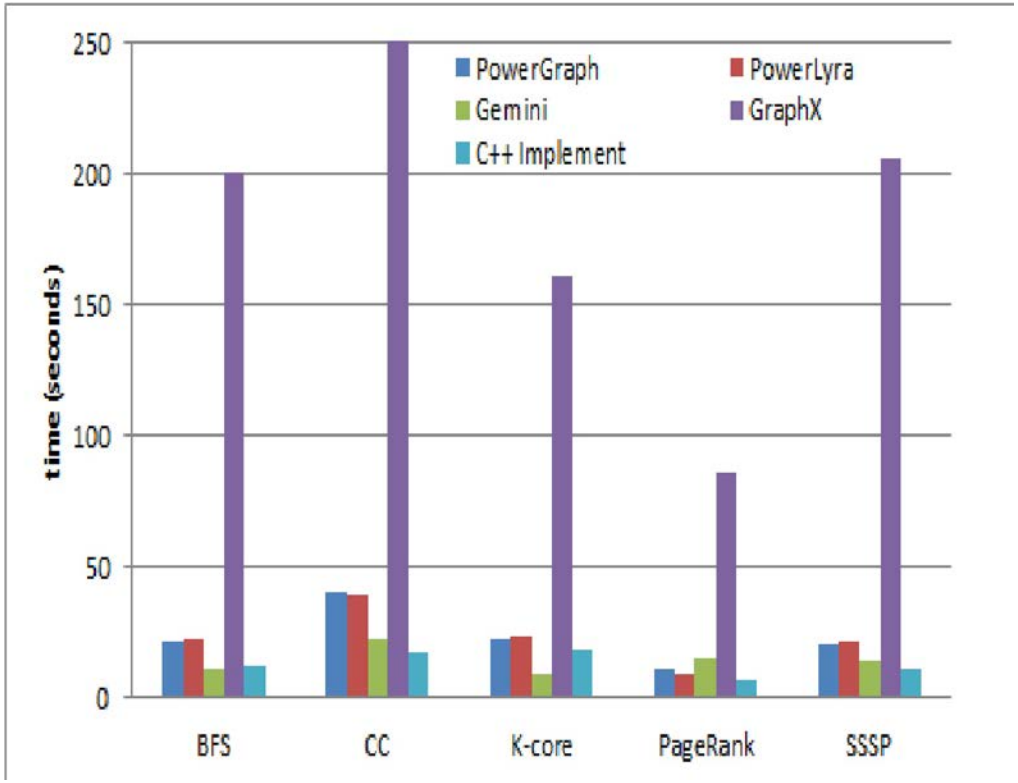
Experimental Configurations

- Platforms

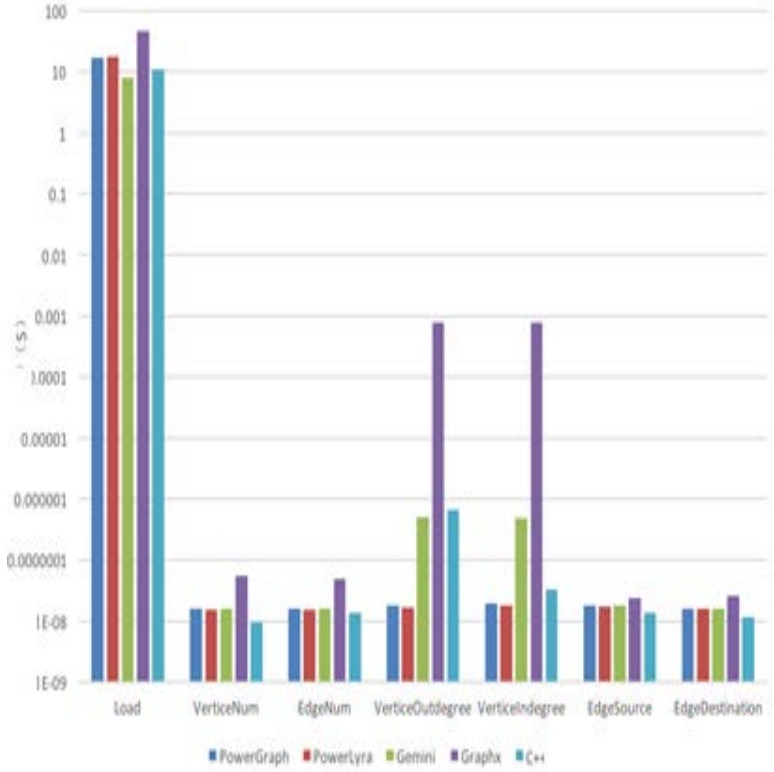
CPU	Intel(R) Xeon(R) E5645 2.40G
Memeory	96GB DDR3 1333MHz bandwidth:8GB/s
Network	Ethernet 1G bandwidth:943Mbits/s
Disk	SATA 1T bandwidth:154.82MB/s
OS	Ubuntu 16.04 and the kernel is 4.13.0-43-generic
GCC	4.3
Redis	4.2.5

- Workloads
  - We use GraphBench as the experimental workloads.

The Execution Time



Component Benchmarks

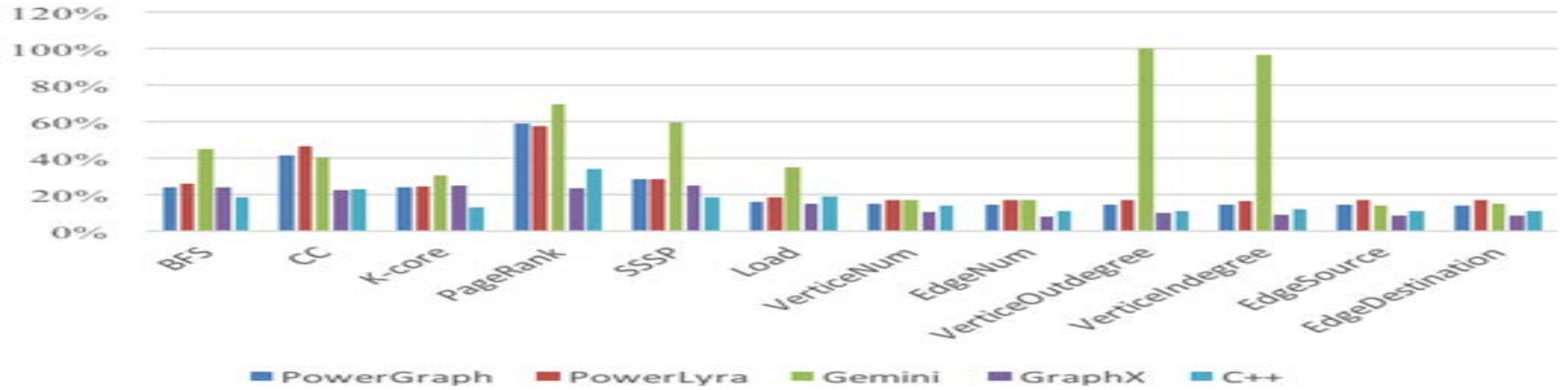


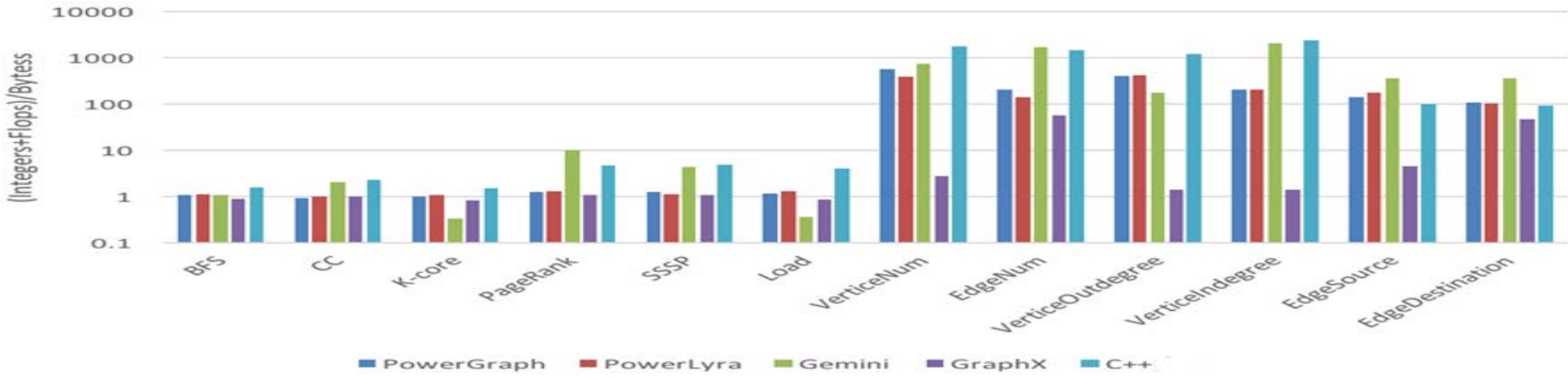
Micro Benchmarks

The Fine-Grained Analysis of CC Workload

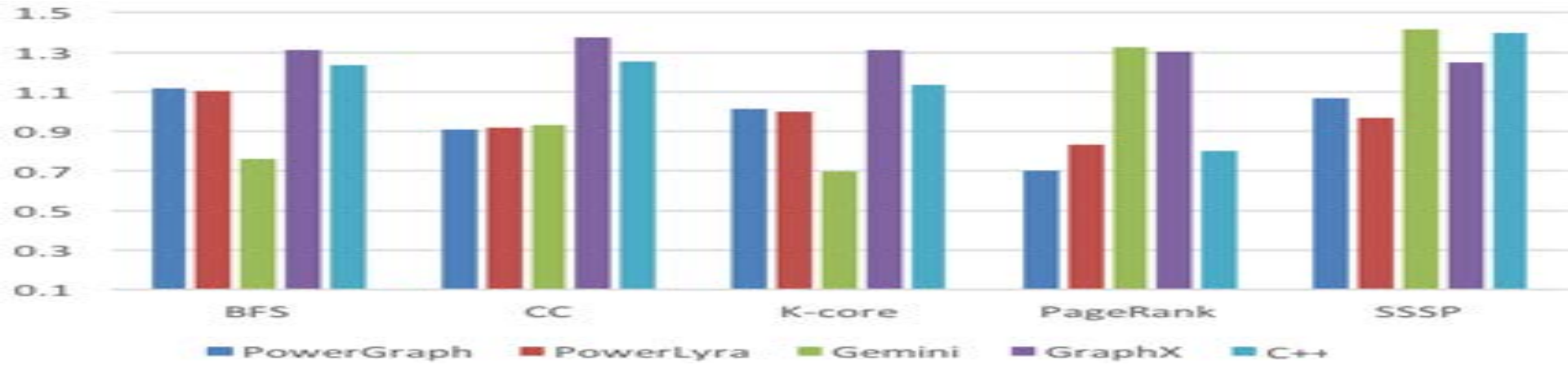
	The PowerLyra CC Workload			The Gemini CC Workload		
	Execution time	Times	Time Ratio	Execution time	Times	Time Ratio
Total	38.7	-	100%	22	-	100%
Load	17.8	1	46%	8	1	36.4%
EdgeSource	1.8E-8	376713258	17.2%	1.81E-8	208087132	16.6%
EdgeDestination	1.6E-8	376713258	15.6%	1.61E-8	312130698	22.7%
UDO	8.2	-	21%	5.3	-	24.2%
Others	0.08	-	0.2%	0.02	-	0.1%

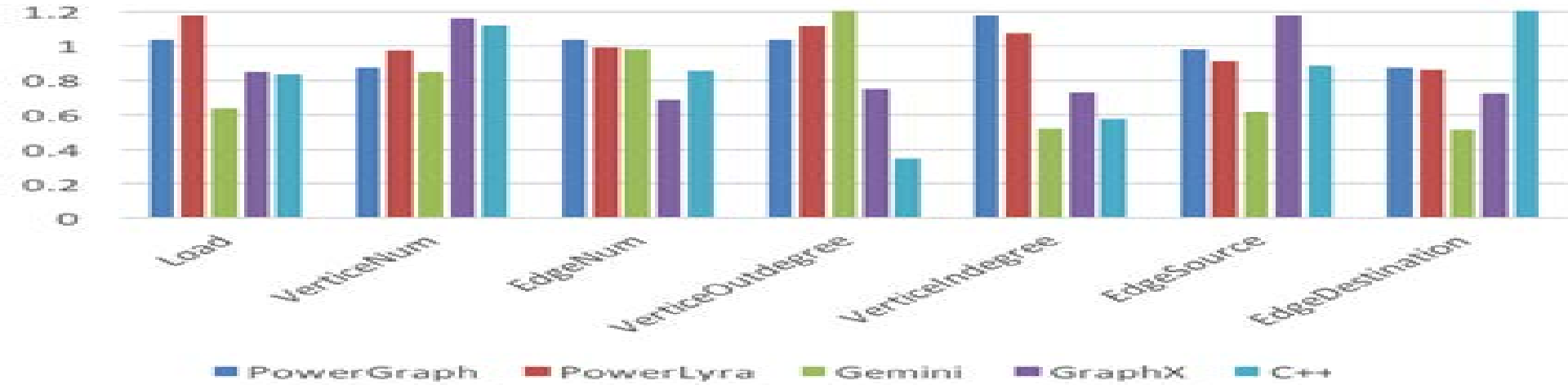
CPU Utilizations & Computation Intensity



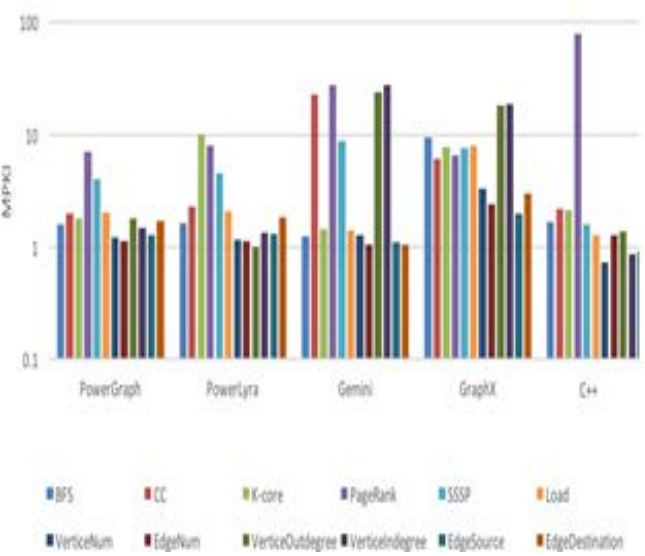


IPC (Instructions Per Cycle)

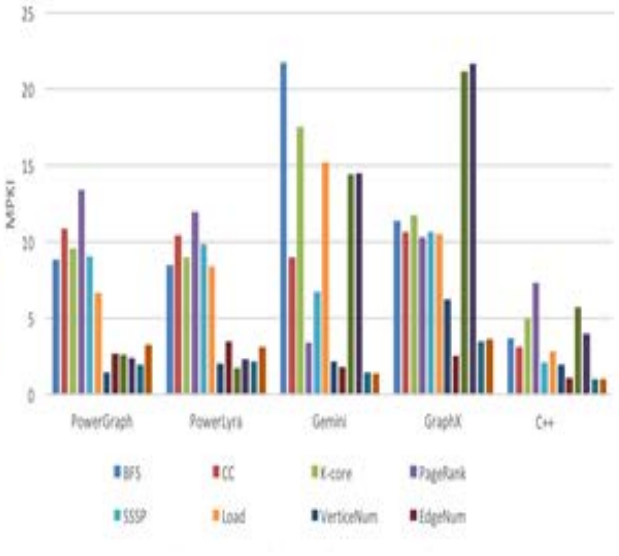




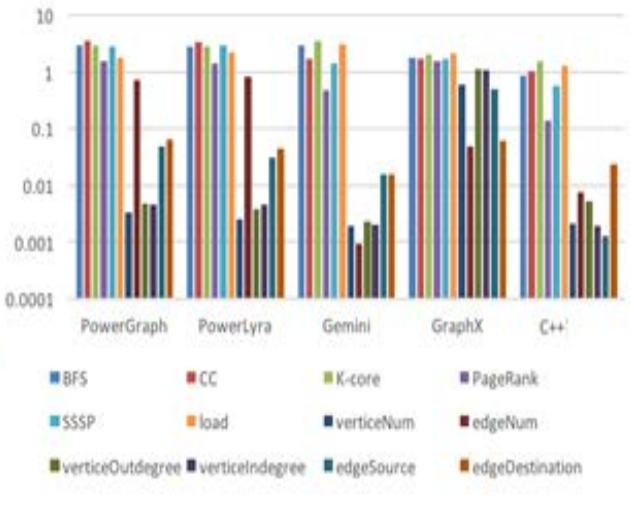
Branch Behaviors & Cache Behaviors



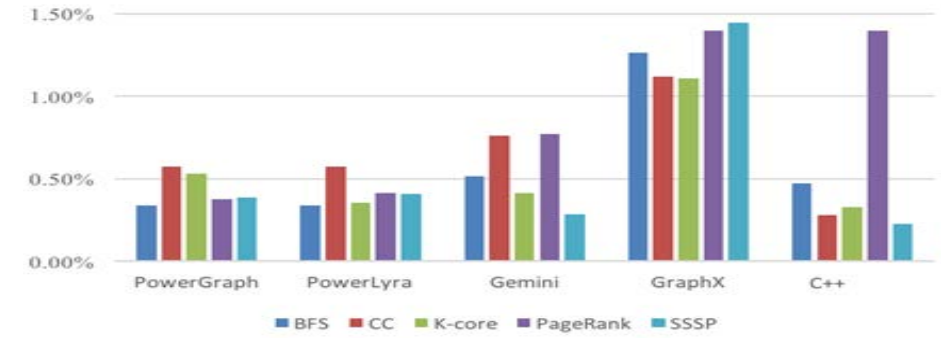
L1I Cache MPKI



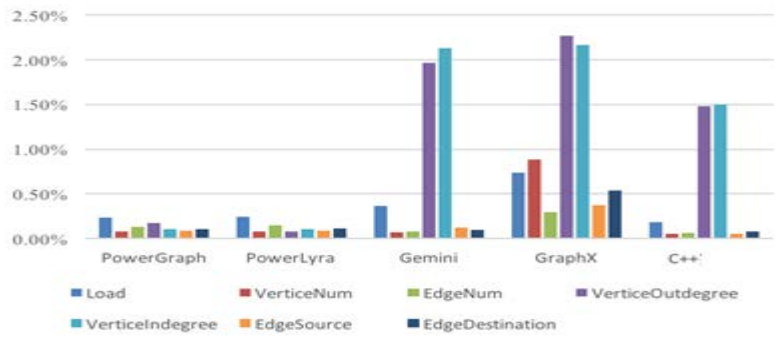
L2 Cache MPKI



L3 Cache MPKI



branch miss ratio



branch miss ratio

Evaluation Summary

- There is no one-size-fits-all solution for the graph computing system.
- Using GraphBench, we can evaluate the graph computing system at the fine-grained level and get more insights.
  - the CPU utilization, the computation intensity and the branch prediction are correlated with the user-observed performance of graph computing system
  - the IPC does not totally conform to the user-observed performance.

Conclusions

- We build the graph computing benchmark suite—GraphBench
  - includes micro-benchmark (graph basic operations) and component benchmarks (graph computing workloads).
- We evaluates the graph computing systems with the GraphBench
  - GraphBench can help people to better understand the graph computing system at the fine-grained level.


QUESTIONS
And
Answers