# Characterizing and subsetting Big Data workloads

Zhen Jia, Jianfeng Zhan , Lei Wang, Rui Han, Sally A. McKee, Qiang Yang,
Chunjie Luo, and Jingwei Li

1. Select representative workloads in big data fields

2. Identify a set of microarchitecture metrics that can directly or indirectly reflect program behavior

3. Standard statistical methods in characterizations

Identical Algorithms

---all selected applications have two implementations of the same algorithm

---offline analytics application: Hadoop & Spark

---interactive analytics application: Hive & Shark

Identical Data Sets

---For two applications, both data formats and data size are identical

---input data size for each workload is determined by the Spark

---For each workloads, data size ranging from several gigabytes to more than 100 gigabytes

Same Infrastructure

# Methodology—Workload

TABLE I. REPRESENTATIVE DATA ANALYSIS WORKLOADS

| Category | Workload | Problem Size | Data Type | Software Stack |
|---|---|---|---|---|
| Offline Analytics | Sort | 80 GB | unstructured sequence file | Hadoop & Spark |
| | WordCount | 98 GB | unstructured text | |
| | Grep | 98 GB | unstructured text | |
| | Naive Bayes | 84 GB | semi-structured text | |
| | K-means | 44 GB | unstructured text | |
| | PageRank | $2^{24}$ vertices | unstructured graph | |
| Interactive Analytics | Projection | 420 million records | structured table (e-commerce transaction data set) | Hive & Shark |
| | Filter | 420 million records | | |
| | Order By | 420 million records | | |
| | Cross product | 100 million records | | |
| | Union | 420 million records | | |
| | Difference | 100 million records | | |
| | Aggregation | 420 million records | | |
| | JoinQuery | 100 million records | | |
| | AggQuery | 420 million records | | |
| | SelectQuery | 420 million records | | |

A broad set of metrics of different types that cover all major characters
---particularly focus on factors that may affect data movement or calculation

Nine Categories of 45 metrics

Eliminate correlated metrics before analysis
---Principle Component Analysis(PCA)
---only keep the top few PC's which have eigenvalues greater than or equal to one

Using Hierarchical clustering to analyze the similarity among workloads in two software stack

---Hierarchical clustering can quantitatively show the similarity among workloads via a dendrogram

Eight PCs

---retain 91.12% variance

All big data workloads(32)

Dendrogram

---drawing a U-shaped link between a non-singleton cluster and its children

---the length of the top of the U-link is the distance between its children

---the shorter the distance, the more similar the children

---linkage distance between two clusters is made by element pair, namely those two elements(one in each cluster) that are closest to each other

# Methodology—Experiment A similarity analysis

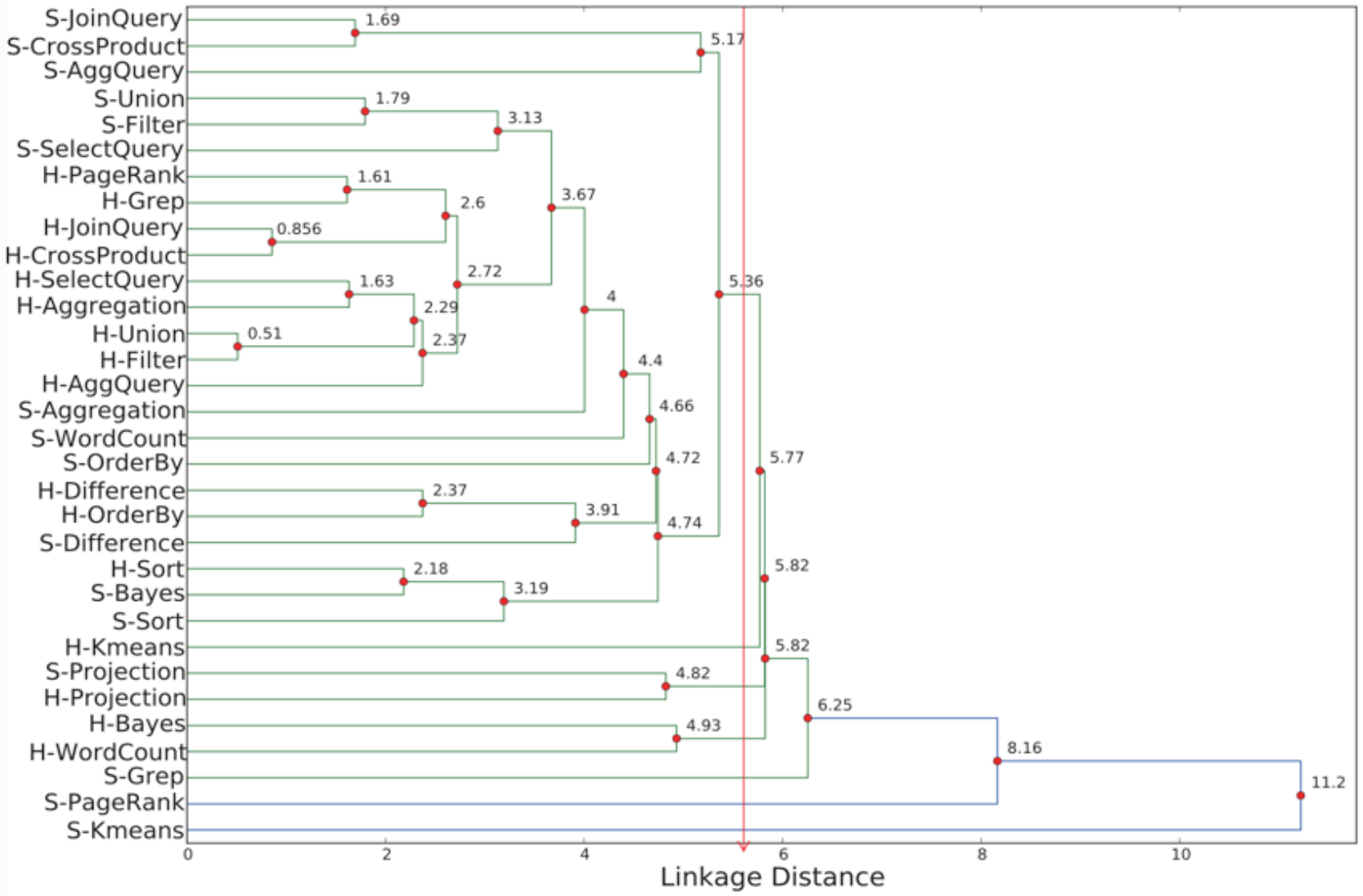


Fig. 1. Similarity of Hadoop (H) and Spark (S) workloads.

Observation 1:
At first clustering iteration, 80% of clusters consist of workloads that are based on the same software stack.

Observation 2:
Two workloads implementing the same algorithm on the different software stacks do not get clustered together in the first clustering iteration- the only exception is Projection

Observation3:
After the first iteration, workloads based on the same software stack are easier to cluster together

Conclusion from Observation1&2&3:
software stacks have significant impacts on workload behaviors — even greater than that of the benchmark algorithms

# Methodology—Experiment A similarity analysis

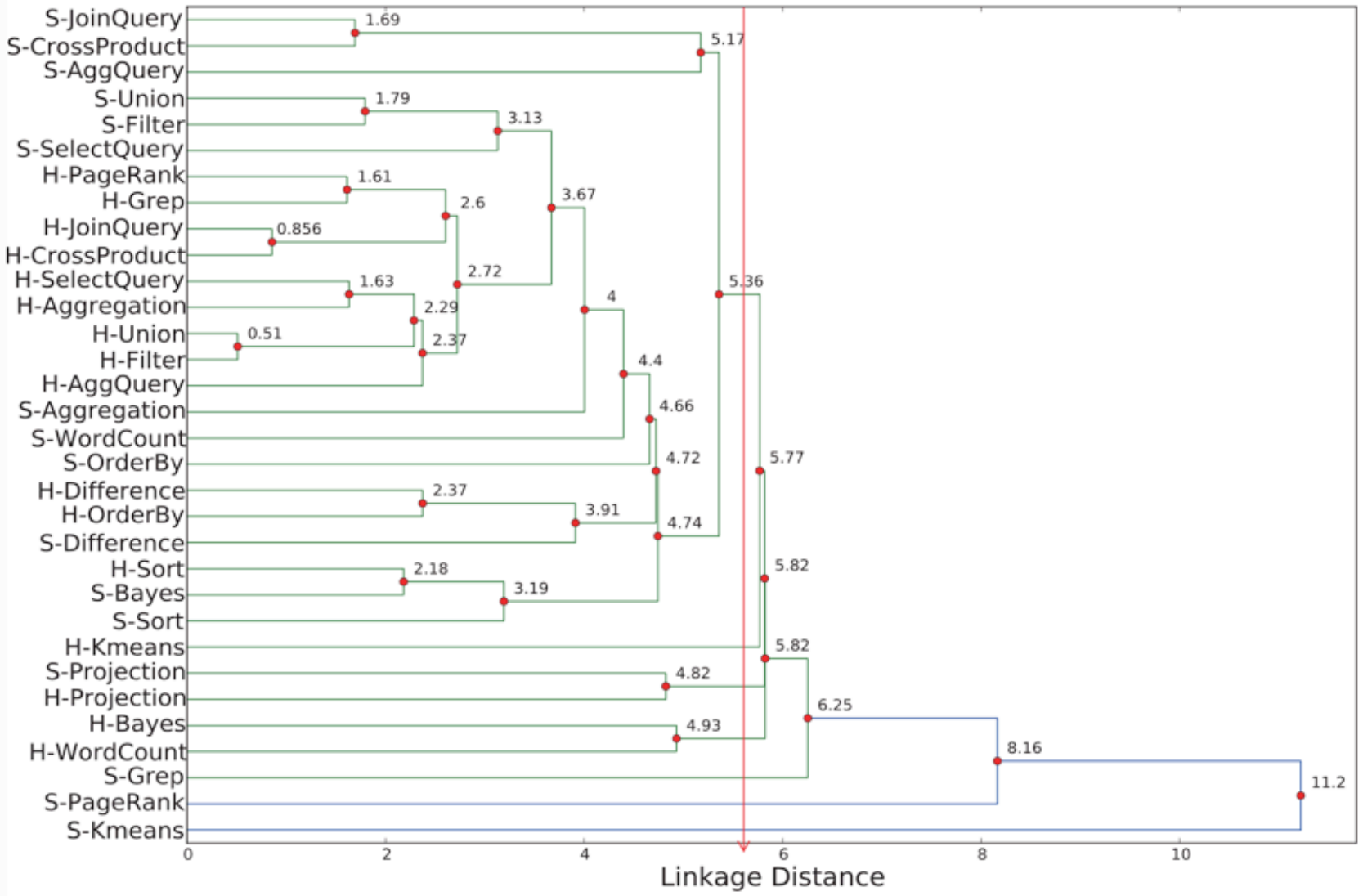


Fig. 1. Similarity of Hadoop (H) and Spark (S) workloads.

Observation 4:
H-Union/H-Filter, S-Union/S-Filter, H-JoinQuery/H-CrossProduct, and S-JoinQuery/S-CrossProduct, are grouped, respectively. Workloads using the same software stack to implement similar algorithms have similar behaviors

Observation 5:
There are nine Hadoop workloads clustered within linkage distances of 2.72.Only three Spark-based workloads are clustered within a similar distance. Hadoop-based workloads have more similarities with each other than their Spark-based counterparts.

Conclusion from Observation4&5:
Hadoop has a larger impact on microarchitectural behaviors than Spark

The first four PCs

---covering 71.45% variance

All big data workloads(32)

Scatter Plot

---Workloads appearing close in these figures may in fact be farther away when all PCs

--- workloads that appear far apart indeed exhibit significantly different behaviors

# Methodology—Experiment B Principle Component Space analysis

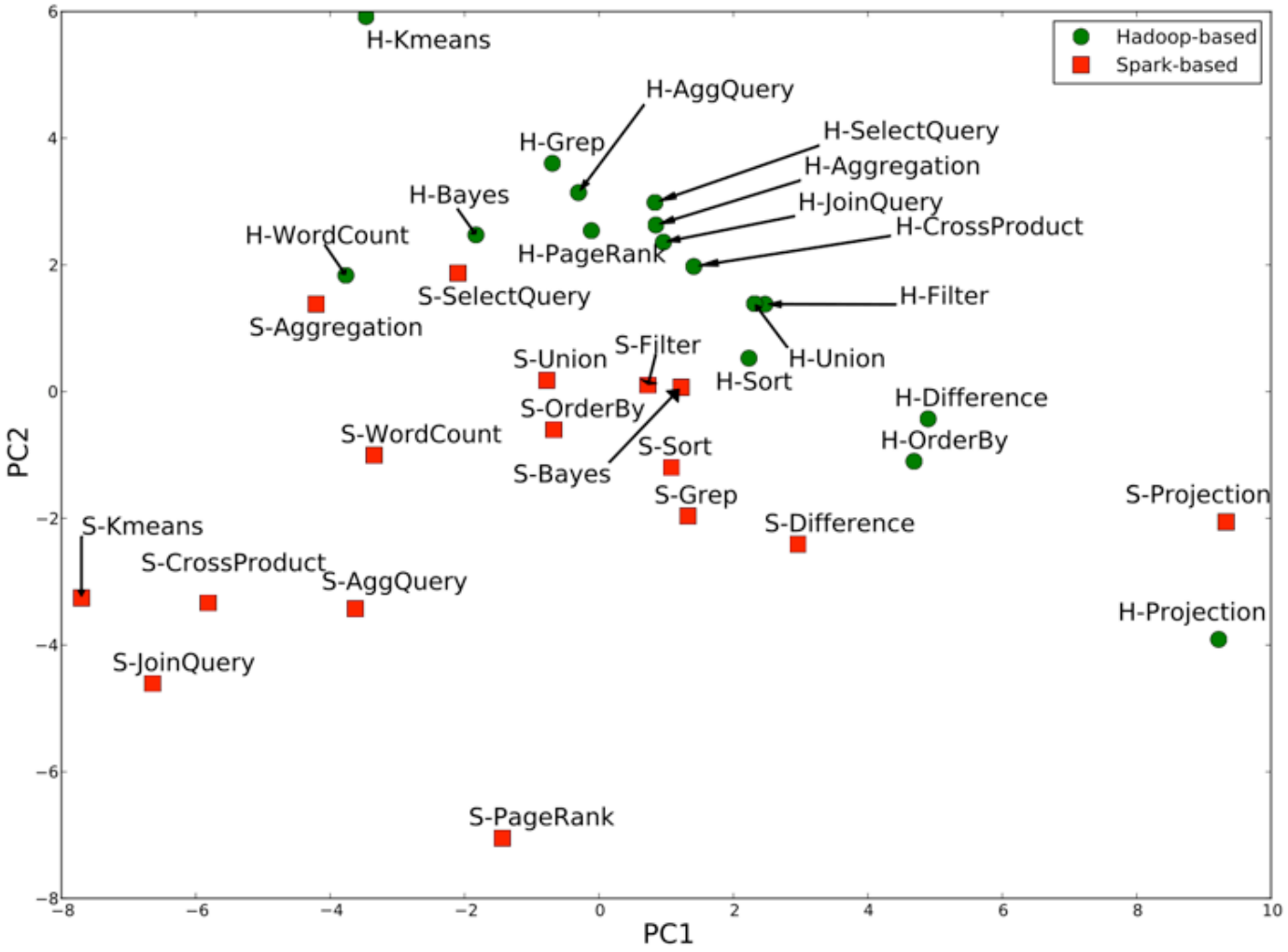


Fig. 2. Scatter plot of workloads using the first and second principle components

Spark-based workloads are spread widely along the x-axis (PC1) in Figure 2.

Hadoop-based workloads are grouped in the middle of the chart.

Conclusion: Spark- based workloads exhibit more diversity than their Hadoop- based counterparts with respect to PC1.

# Methodology—Experiment B Principle Component Space analysis

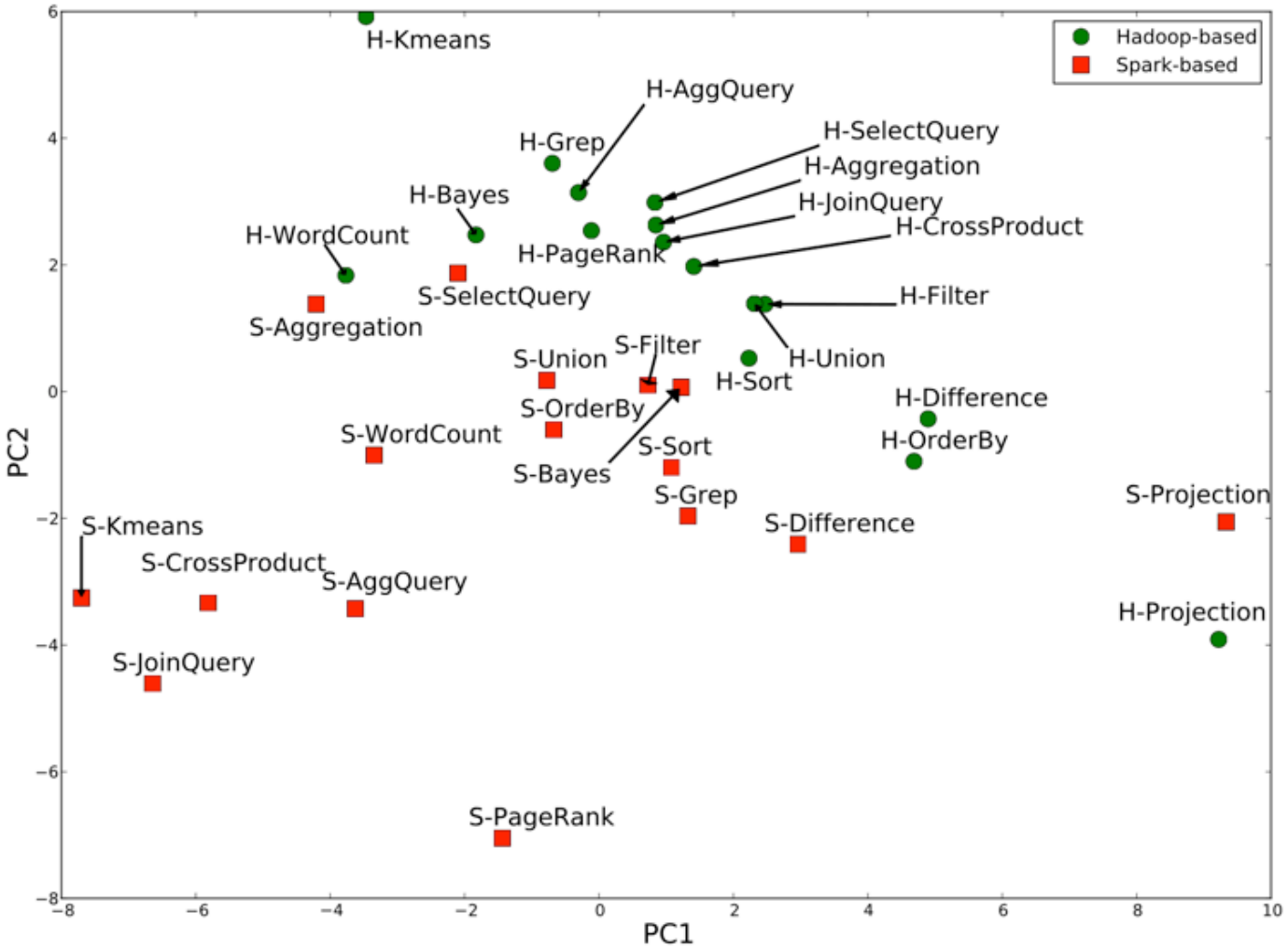


Fig. 2. Scatter plot of workloads using the first and second principle components

On the y-axis (PC2) we find that most of the Hadoop-based workloads are located towards the top, whereas most of Spark-based workloads are located in the middle or bottom parts of the chart.

Conclusion: PC2 is the main component that differentiates the Hadoop-based workloads from the Spark-based workloads

# Methodology—Experiment B Principle Component Space analysis

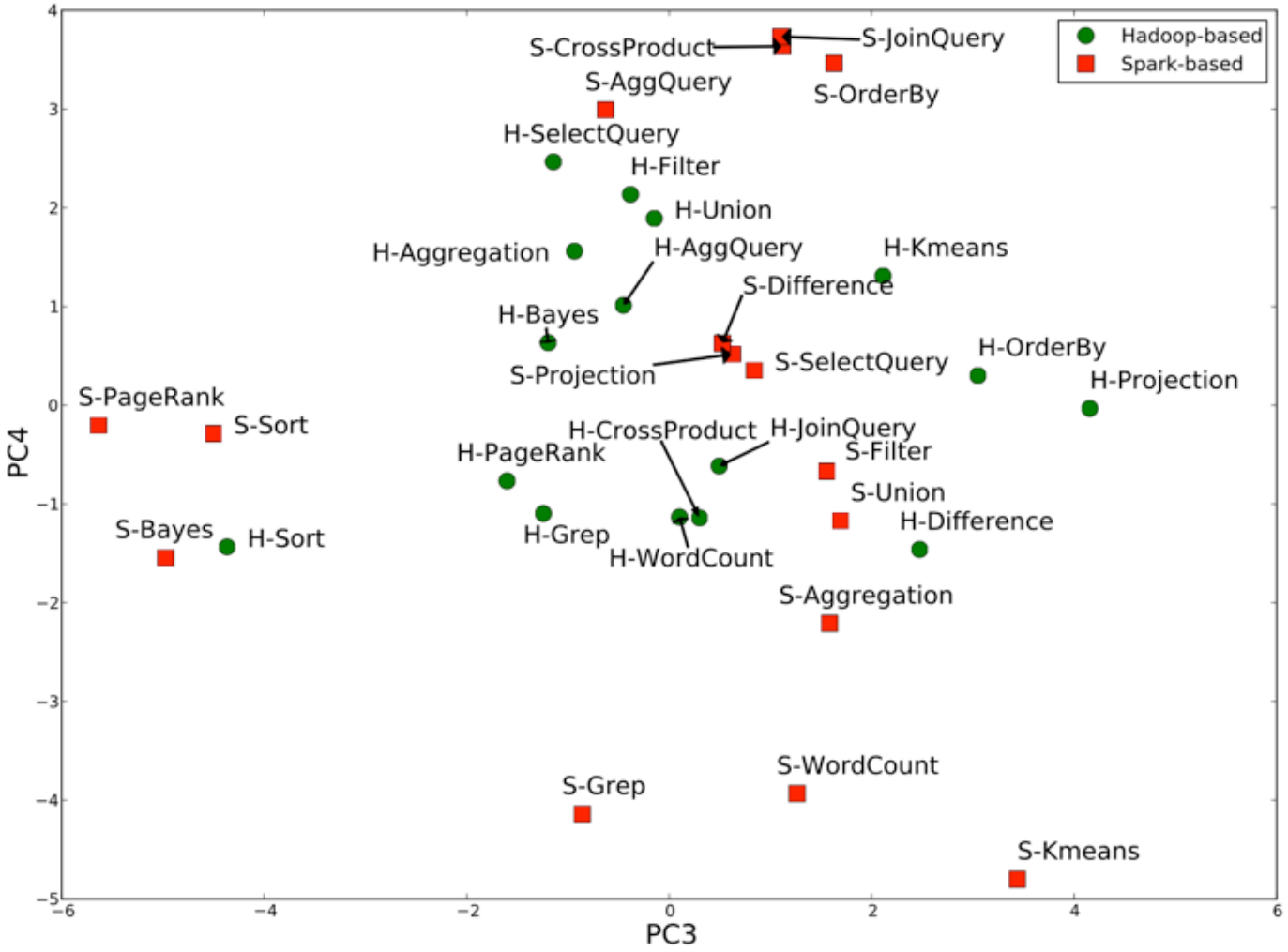


Fig. 3. Scatter plot of workloads using the third and fourth principle components

Spark-based workloads nearly cover all of the space for PC3 and PC4.

Hadoop-based workloads are again grouped in the center

Conclusion from Figure 2&3:

Verify the the result in experiment A

Spark-based workloads show more diversity than the Hadoop-based workloads with respect to microarchitectural behaviors because the Hadoop software stack minimizes the impact of the user application code

investigate which microarchitectural-level metrics dominate PC2 values

---Select 9 metrics negatively dominate PC2's value

---Select 5 metrics positively dominate PC2's value

--- calculate the mean for each metric for each kind of big workload and present normalized Hadoop-based values using the Spark-based workloads as the baseline

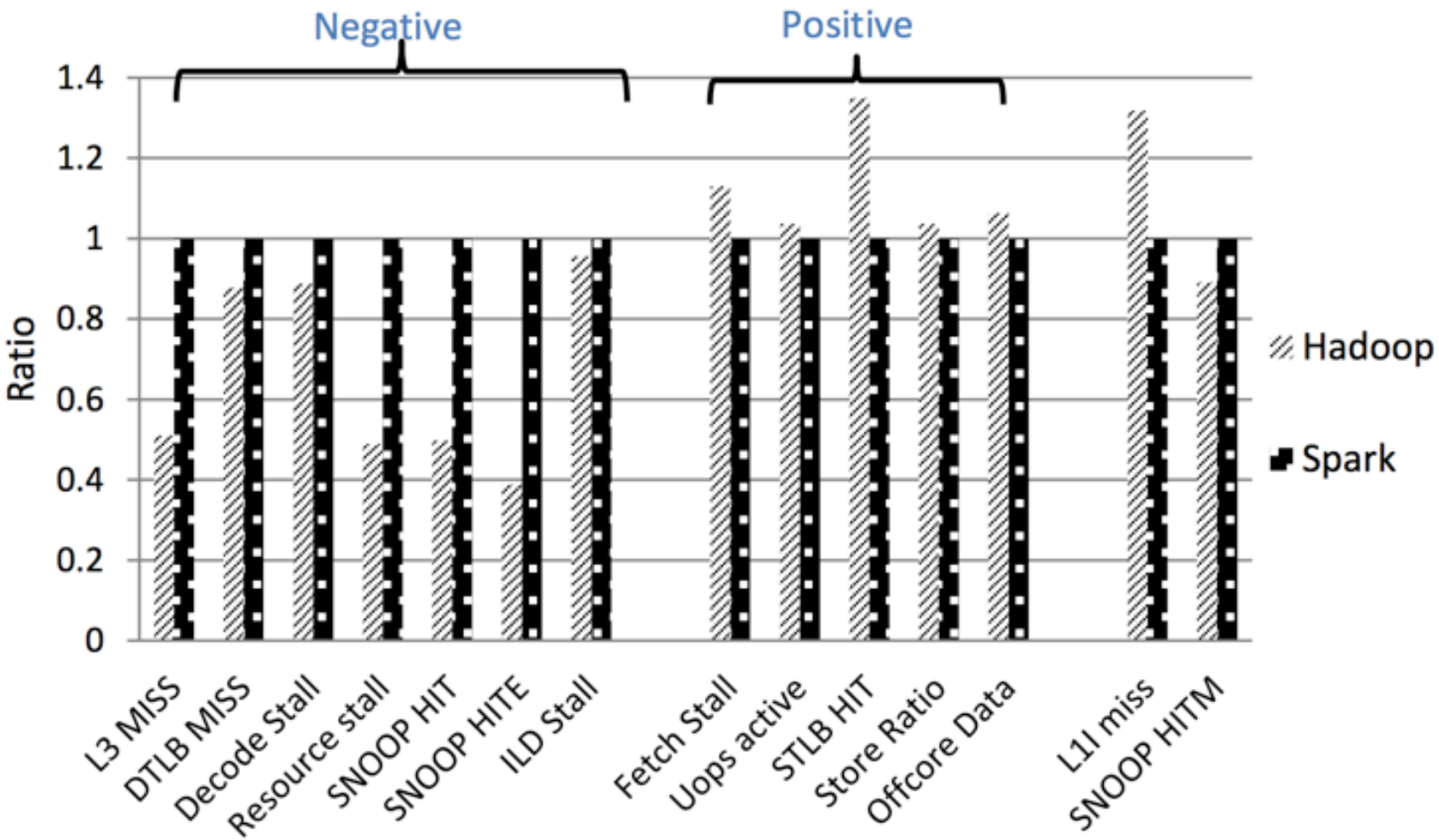


Fig. 5. Metrics causing Hadoop and Spark to behave differently

Observation 6&7&8:
the large number of L3 cache and data TLB misses cause more backend stalls for the Spark-based work- loads, and then high instruction cache misses cause more fron-tend stalls for the Hadoop-based workloads

Conclusion: Hadoop-based workloads have larger instruction footprints than their Spark-based counterparts. Spark-based workloads have larger data footprints than Hadoop-based workloads

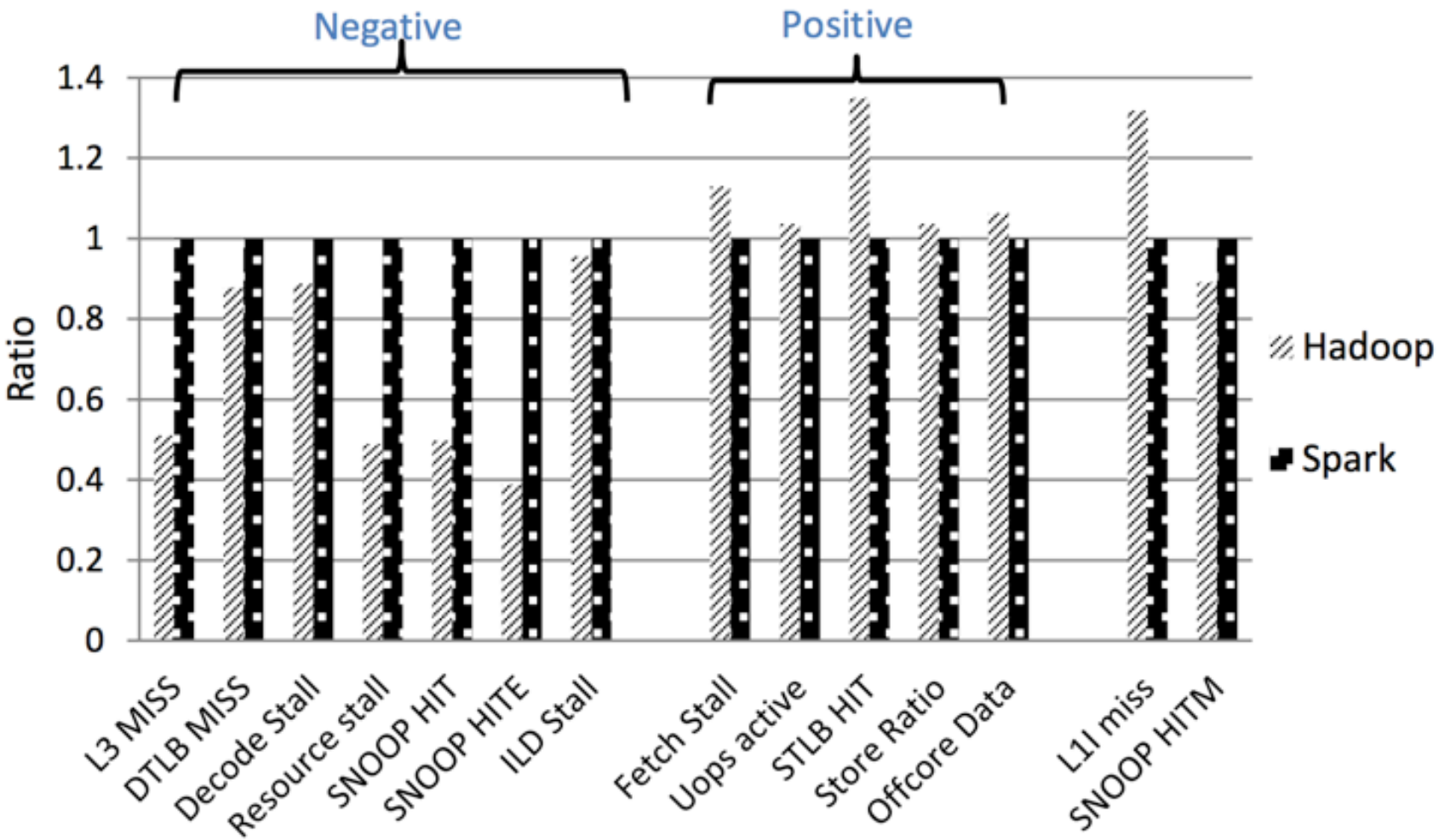


Fig. 5. Metrics causing Hadoop and Spark to behave differently

Observation 9:
Spark-based workloads have more SNOOP HIT and SNOOP HITE responses than Hadoop-based workloads.

Conclusion:
Spark-based workloads have more data sharing among different cores.

Using eight principle components obtained in experiment A as metrics

K-means clustering to group the workloads

Choose a representative workload from each cluster

TABLE IV. THE RESULT OF K-MEANS CLUSTERING ALGORITHM

| Cluster | Workloads | Number |
|---|---|---|
| 1 | H-PageRank, H-Grep, H-JoinQuery, H-Sort, H-CrossProduct, S-Bayes, S-Grep, S-Sort | 8 |
| 2 | H-AggQuery, S-Filter, S-Union, S-SelectQuery, S-OrderBy, H-Kmeans | 6 |
| 3 | S-Aggregation, S-WordCount, S-Kmeans, H-Wordcount, H-Bayes | 5 |
| 4 | H-OrderBy, H-Difference, S-Difference, S-PageRank | 4 |
| 5 | H-Aggregation, H-Filter, H-Union, H-SelectQuery | 4 |
| 6 | S-CrossProduct, S-JoinQuery, S-AggQuery | 3 |
| 7 | H-Projection, S-Projection | 2 |

Two Method in K-means Clustering:

--- choose the workload that is as close as possible to the center of the cluster it belongs to

--- select an extreme workload situated at the “boundary” of each cluster

Using hierarchical clustering to select seven representative workloads, four of result workload s also included in the second approach in K-means while not in the the first approach

Conclusion: the second approach superior in selecting representative workloads

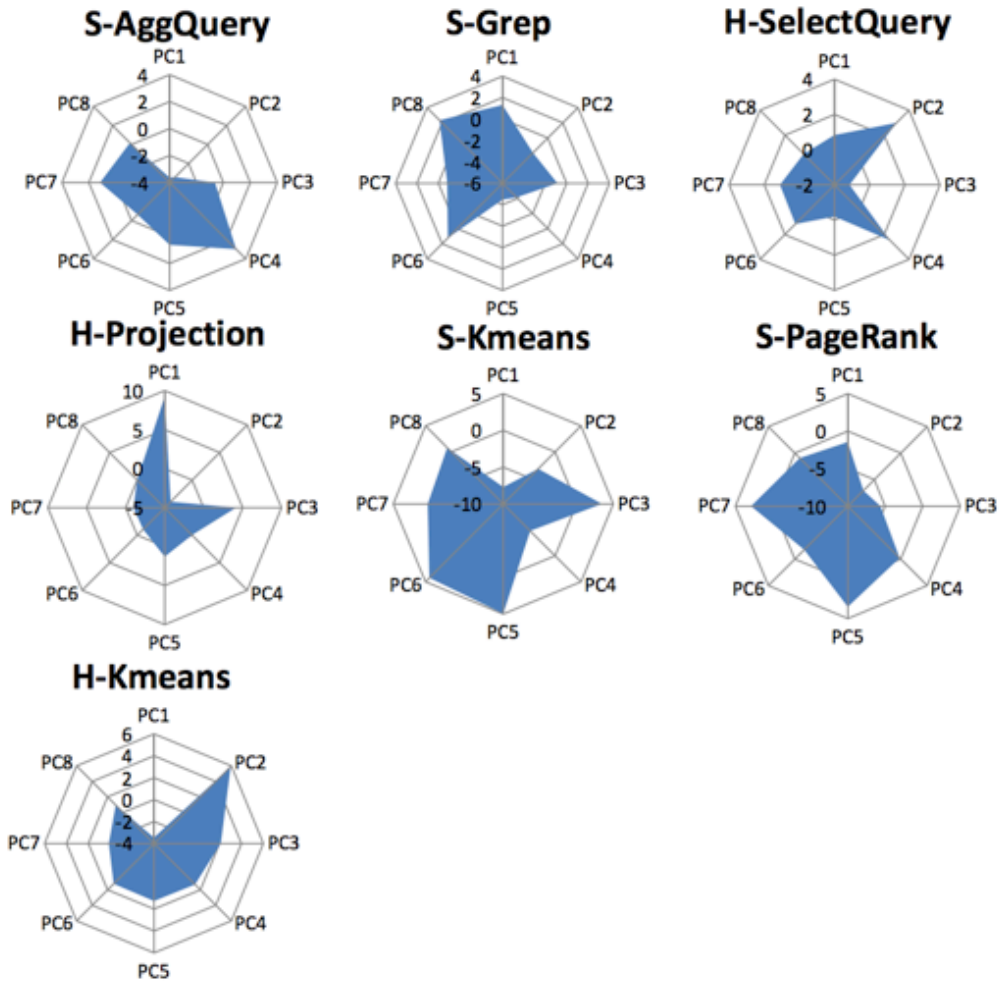


Fig. 6. Kiviat diagrams of the representative workloads