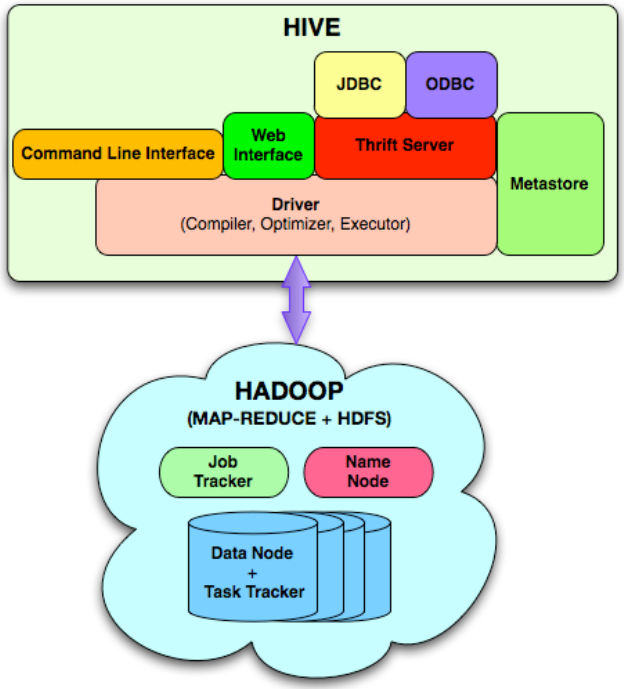
Semantics-Aware Prediction for Analytic Queries in MapReduce Environment

Weikuan Yu, Zhuo Liu, Xiaoning Ding

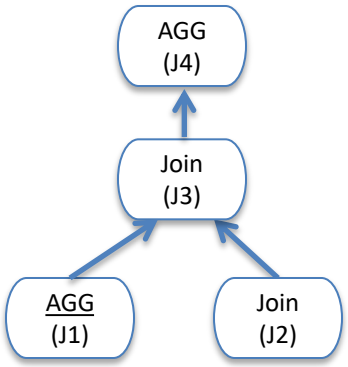
Florida State University
Auburn University
New Jersey Institute of Technology

Background

- MapReduce is a popular data-centric programming model.
- Hive and Pig are popular data warehouse systems.
  - More than 40% of Hadoop jobs at Yahoo! were Pig programs back in 2009, more with Hive now.
  - In Facebook, 95% of MR jobs are generated by Hive.
- In Hive, each SQL query is compiled and translated into a DAG (Directed Acyclic Graph) of MapReduce jobs with inner-dependencies.



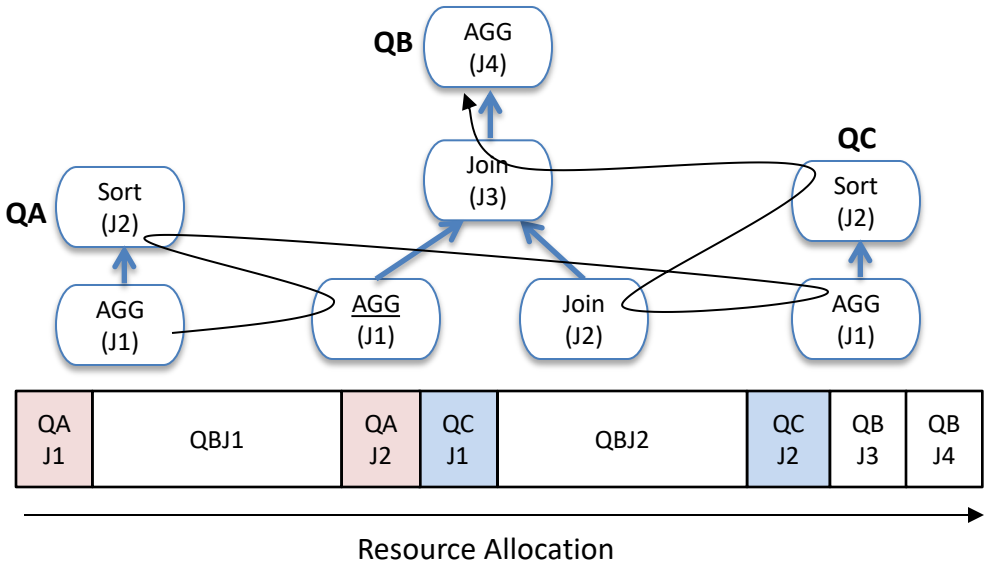
Source: ICDE'10 by Facebook



Motivation

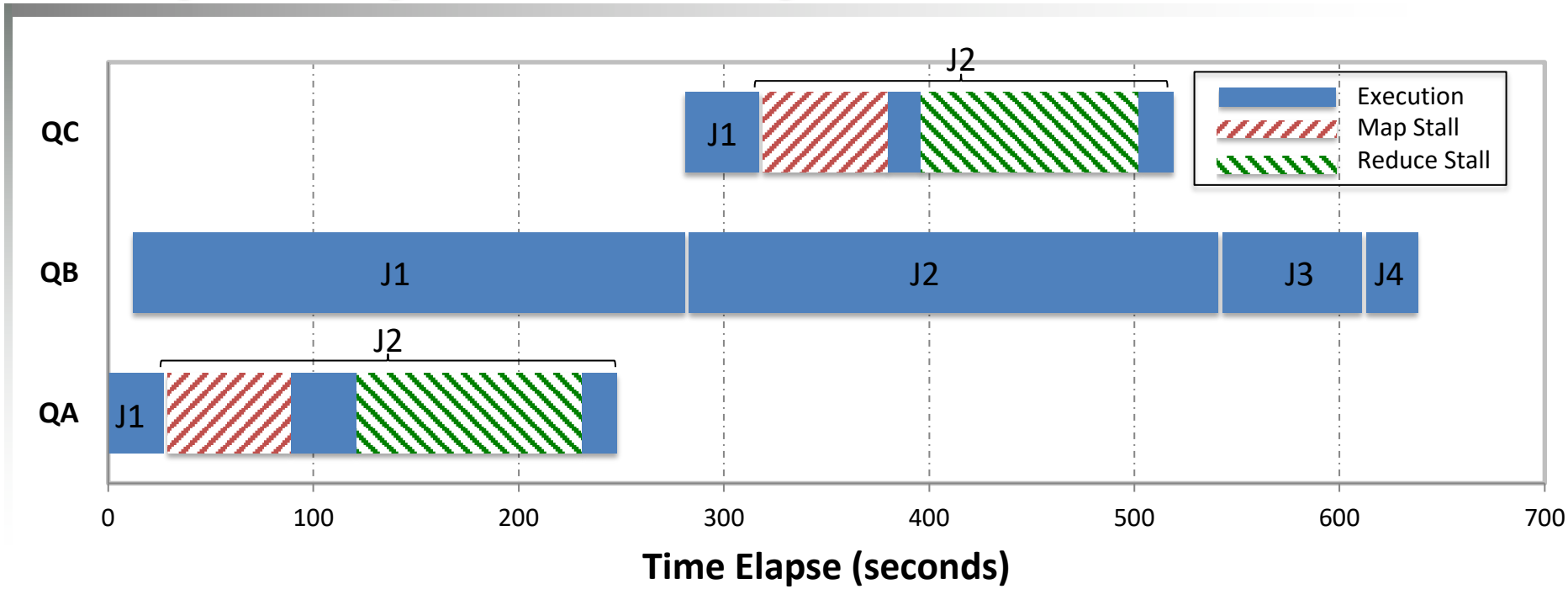
- Semantic gap: between Hive and Hadoop
  - Hadoop is un-aware of such dependency and inter-job relationship, just treating all jobs as the same.
  - Without such awareness, it will be difficult for Hadoop to schedule jobs that belong to a query efficiently.
- Problems:
  - Suboptimal query response time
  - Unfairness among queries

Efficiency issue for queries with varied sizes



- In this test, QA, QB and QC are issued in sequence, where QB is a large query.
- Under HCS, interleaved execution happens among queries' jobs.

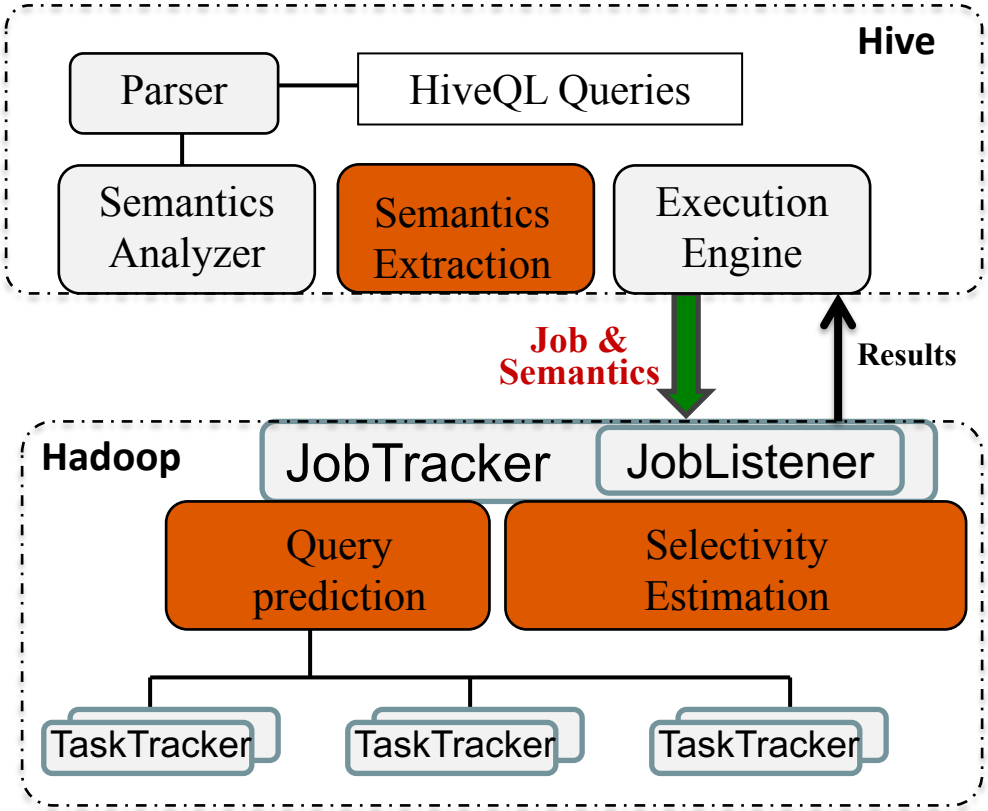
Query Delays shown by GANTT Chart



- QA arrives first with its J1 job, QB and QC afterwards with their jobs listed accordingly.
- QA-J2 gets delayed by QB-J1. QC-J2 gets delayed by QB-J2.
- Query response time can be improved if the scheduler is aware of the query semantics, therefore the relationship among the jobs.

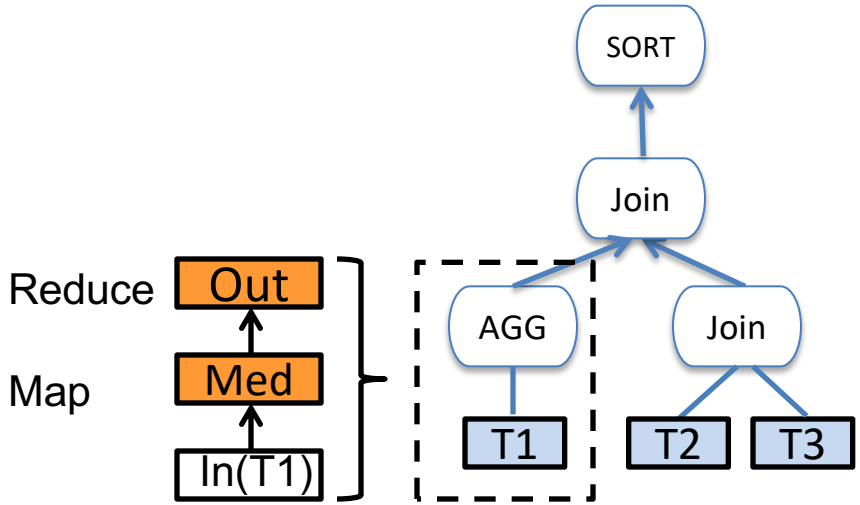
Semantics-Aware Query Prediction

- Three main techniques
  - Semantics extraction (DAG, operator type, predicates, etc.)
  - Selectivity Estimation
  - Query prediction



Selectivity estimation for a query's jobs

- Selectivity estimation
  - Predict each job node's intermediate (Med) and output (Out) data sizes recursively along the DAG (from bottom to top).
  - For different job types, e.g., Groupby, Join, Select, we reply on certain formulas and offline-built histograms to estimate their selectivities.
- Logic: Selectivity estimation=>Job/query resource estimation and time modeling =>Used for efficient query scheduling



Selectivity Estimation

- IS is used for estimating MOF size
  - $IS = D_{Med}/D_{In}$
- Final Selectivity (FS) is defined as:
  - $FS = |Out|*W_{Out}/D_{In}$
- Predicate selectivity (ratio of selected rows to input rows)
  - $S_{pred} = |Med| / |In|$
- Projection selectivity (ratio of selected cols to tuple width)
  - $S_{proj} = \sum Width_{col\ i\ selected} / TupleWidth$

Intermediate Selectivity - IS

- For extract job such as select and order by,
  $IS = S_{pred} * S_{proj}$

- For join job:
  $IS = S_{pred\ 1} * S_{proj\ 1} * r_1 + S_{pred\ 2} * S_{proj\ 2} * (1 - r_1)$

- Groupby can involve local combine: $IS = S_{comb} * S_{proj}$
  - For clustered keys,
    $S_{comb} = \min(1, \frac{T.dxy}{|T| * Spred}) * S_{pred} = \min(S_{pred}, \frac{T.dxy}{|T|})$
  - For randomly distributed keys,
    $S_{comb} = \min(S_{pred}, \frac{T.dxy}{|T|/N_{Maps}})$

Final Selectivity – Output

- For extract job,
  - For "top k" job, $|Out|=\min(|In|, k)$
  - For "order by" job, $|Out|=|In|$
- For groupby job,
  - $|Out|= \min(|T|*S_{pred}, T.dxy)$
- For join job,
  - Equ-join with uniform keys:

    $$|Out| = |T_1 \bowtie T_2| = |T_1|*|T_2|*\frac{1}{\max(T_1.dx,\ T_2.dx)}$$

  - Chained joins:

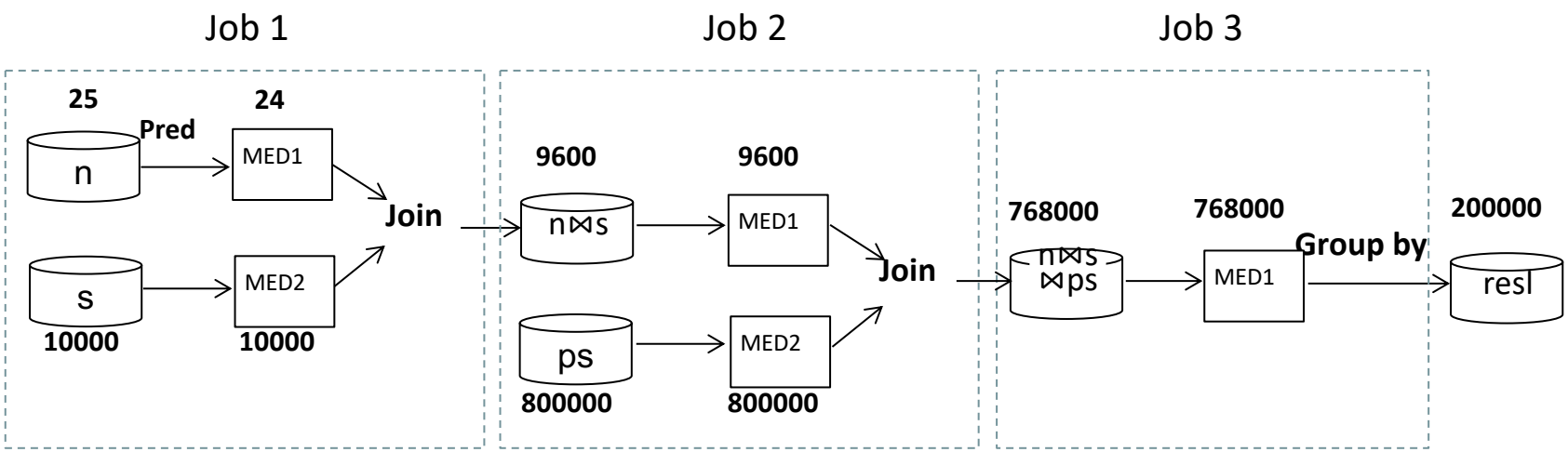
    $$|Out| = |T_1.\mathrm{pred}_1 \bowtie T_2.\mathrm{pred}_2 \bowtie T_3.\mathrm{pred}_3|$$

    $$=S_{pred\ 1}*S_{pred\ 2}*S_{pred\ 3}\ \max(|T_1|, |T_2|, |T_3|)$$

An example for selectivity estimation

- Predict jobs' selectivity recursively in a query.

```
SELECT ps_partkey, sum(ps_supplycost*ps_availqty)
FROM nation n JOIN supplier s ON
  s.s_nationkey=n.n_nationkey AND n.n_name<>'CHINA'
JOIN partsupp ps ON
  ps.ps_suppkey=s.s_suppkey
GROUP BY ps_partkey;
```



|Out|= 0.96*25*10000 *1/max(25,25)

|Out|= 0.96 * max(25, 10000, 800000)

|Out| = min(768000,200000)

Multivariate Time Prediction

- List of Considered Input Features
  - Operators
  - Input Data
  - Output Data
  - Data Growth

Table 1: Input Features for the Model

Name	Description
O	The Operator Type: 1 for Join, 0 for others
D_{in}	The Size of Input Data
$D_{avg\,med}$	Avg Intermediate Data Per Reduce Task
D_{out}	The Size of Output Data
$P(1-P)D_{med}$	The Data Growth of Join Operators

Job Time Prediction Model

- Model job execution time based on selectivity estimation
- Training on over 5647 MR jobs, about 1000 queries from TPC-DS and TPC-H of different scales.
- θ is trained for extract, groupby and join jobs respectively.

$$P = \frac{max(|T_1|S_{pred1}, |T_2|S_{pred2})}{|T_1|S_{pred1} + |T_2|S_{pred2}}, \; 0 < P < 1$$

$$ET = \theta_0 + \theta_1 D_{in} + \theta_2 D_{avg\,med} + \theta_3 D_{out} + \theta_4 O * P(1-P) D_{med}.$$

Task Time Prediction Model

- Data size:
  - TD_{In_i} and TD_{Out_i}
- The predicted time for the i-th task: ET_i
  - $ET_i = k0 + k1\ TD_{In\_i} + k2\ TD_{Out\_i} + k3 * P\ (1-P)\ TD_{In\_i}$

Scheduling with Semantics Awareness

- Semantics-Aware Resource Demand
  - Weight Resource Demand (WRD): aggregate the demand from all map tasks (MT_i) and Reduce tasks (RT_i)
  - WRD = $\sum(MTi * NMi) + \sum(RT_i * NRi)$
- Experimented with a simple greedy scheduling policy
  - Prioritizing smallest Queries for fast turnaround
  - Smallest WRD First (SWRD) query scheduling

Evaluation setup

- Benchmarks.
  - Built with TPC-H, TPC-DS queries and Terasort/Grep/Wordcount MapReduce jobs.
  - Submitted in Poisson interval

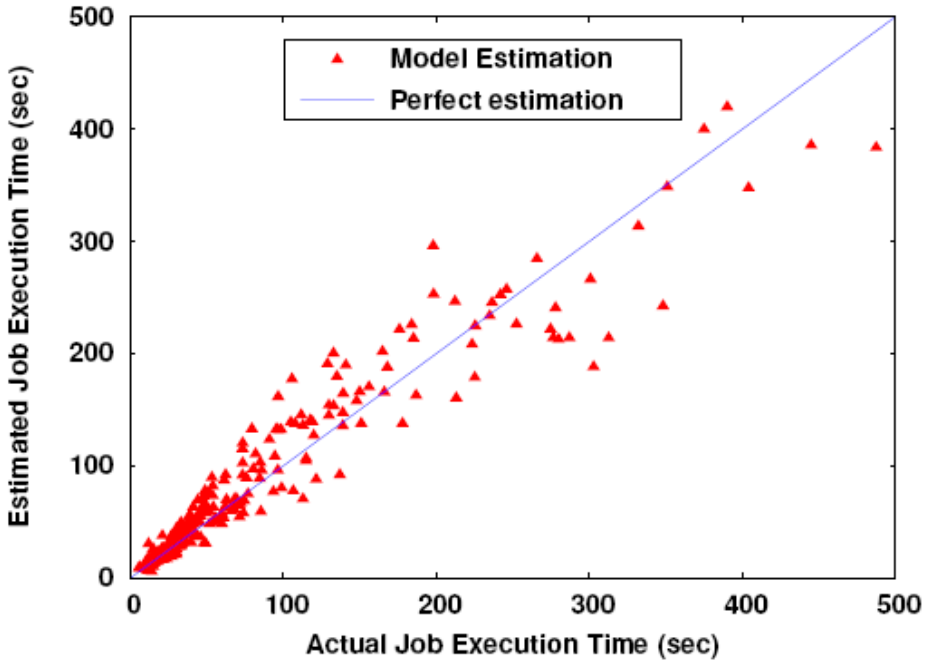
TABLE IV: Workload Composition

Bin	Input Size	Number of Queries		
		Bing	Facebook	QMix
1	1-10 GB	44	85	85
2	20 GB	8	4	4
3	50 GB	24	8	8
4	100 GB	22	2	2
5	>100 GB	2	1	1

- Metrics
  - Accuracy of the prediction via semantics awareness
  - Efficiency: query execution time

Estimation of Job Execution time

- Accuracy
  - On average, 13.98% error rate for the test set of jobs.



Job Time Estimation

Estimation of Task Execution

- Map Task Execution Time
  - Join operators lead to lower accuracy

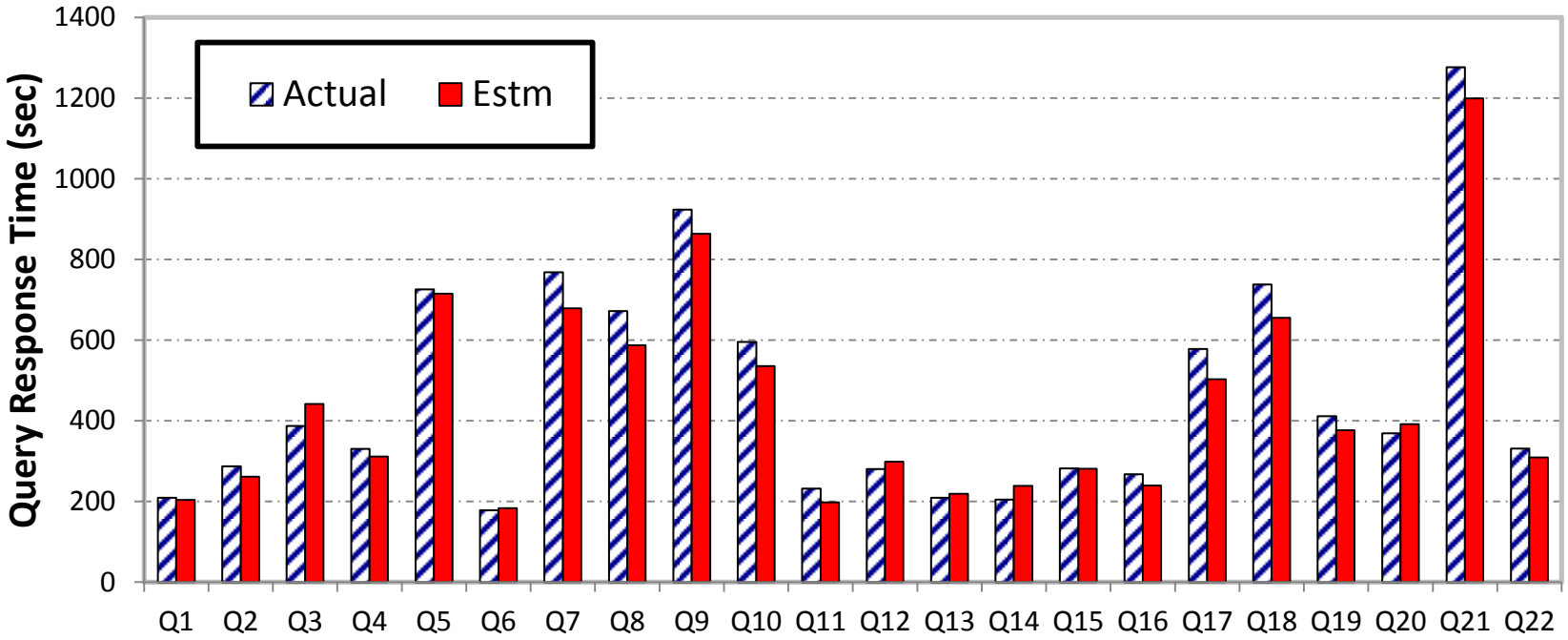
Types	R-squared accuracy	Avg Error
Join	85.6%	16.27%
Groupby	92.4%	24.8%
Extract	92.74%	14.5%
Together	87.05%	20.5%

- Reduce Task Execution Time

Types	R-squared accuracy	Avg Error
Join	85.83%	14.23%
Groupby	98.82%	4.67%
Extract	90.03%	6.18%
Together	90.68%	7.4%

Validation of job and query time estimation

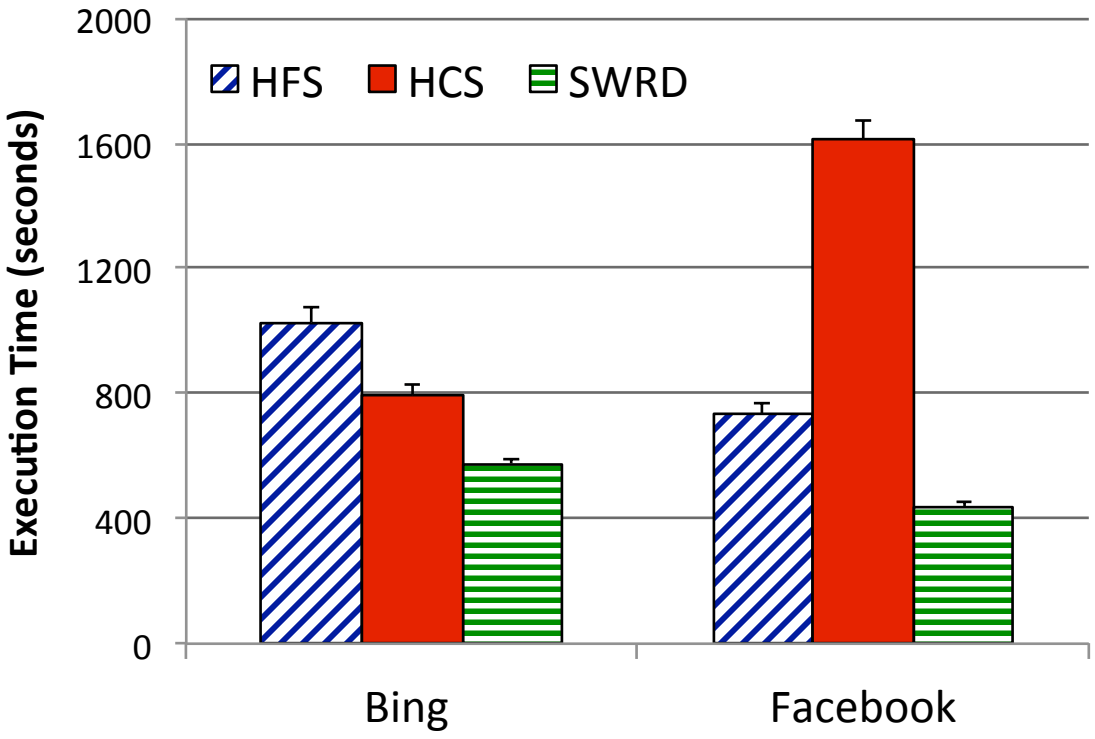
- Predicted time accuracy for queries
  - Error rate is 8.3% on average for 22 100G TPC-H queries.



Query Time Estimation

Benefits of Semantics-Aware Scheduling

- Execution time of queries
  - Compared to HFS, SWRD improves the execution of Bing and Facebook workloads by 44% and 40%, respectively.
  - Compared to HCS, SWRD improves by 27.4% and 72.8%, respectively.



Conclusion and Future Work

- Introduced cross-layer semantics extraction and percolation to increase the semantics awareness of the Hadoop job scheduler
- Formalized the estimation of selectivity for intermediate data and final output
- Developed a multivariate prediction model for job and task execution time and validated the accuracy
  - Leveraged semantics awareness for efficient query scheduling in HIVE

- Plan to pursue further integration of semantics awareness in complex query scheduling and other data analytics systems.

Acknowledgement