

The prediction techniques we developed in this project are advancing the state-of-the-art in three ways:

- i) by providing prediction mechanisms for a class of iterative analytics that were not empirically addressed before and are widely used today in analytical workflows;
- ii) by providing hybrid prediction models for different categories of data analytics (i.e., iterative machine learning, data pre-processing, and reporting SQL) and by analyzing the trade-offs at varying levels of model granularities;
- iii) by providing mechanisms to reduce the training cost (in terms of running benchmark queries) while maintaining a competitive level of accuracy for the models.

PREDICT improves the accuracy of analytical upper bounds for estimating iterations for PageRank from a relative error of [104, 168] % to [0,11] %. Overall, the runtime estimates have an error of [10-30] % for all scale-free graph analyzed. Our prediction techniques proposed in the context of automated workload deployment for reporting analytics can be used to answer resource allocation questions, i.e., identifying the resource allocation(s) that can satisfy a target performance goal."

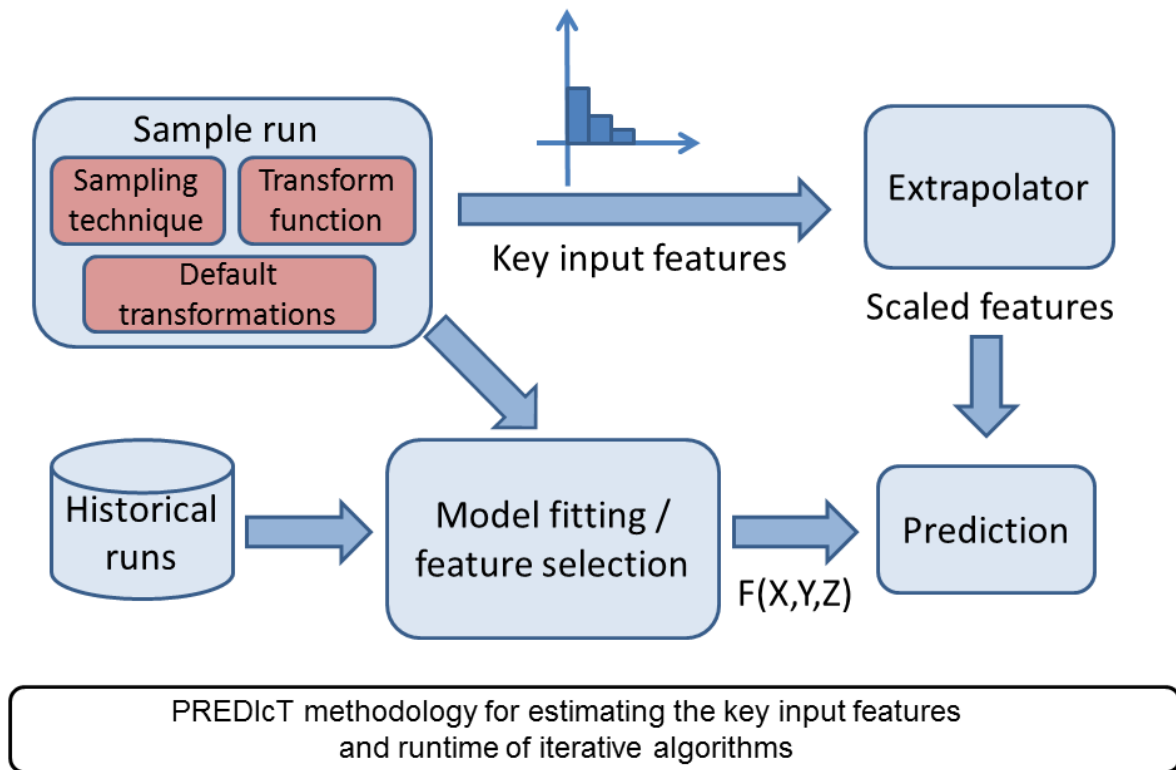
Today's analytical workloads include a mix of declarative queries and workflows of iterative, machine learning algorithms executing on large-scale infrastructures. While SQL-like query languages offer the possibility to execute traditional analytical queries at large scale (e.g., JAQL, Pig, Hive), user defined functions written in high level languages give users the opportunity to execute customized operators. Within the last category, machine learning algorithms are prevalent today and used to filter out irrelevant data or to find correlations into the ever-increasing datasets. For instance, Facebook uses machine learning to order stories in the news feed (i.e., ranking), and to group users with similar interests together (i.e., clustering).

In this project, we address the problem of estimating performance of mixed analytical workflows with the ultimate goal of providing the set of prediction models and techniques that can assist end users to perform automatic workload deployment in a cloud setting. Runtime estimates are a pre-requisite for optimizing cluster resource allocations in a similar manner as query cost estimates are a pre-requisite for DBMS optimizers. In particular, resource managers and schedulers that are used to optimize resource provisioning require as input runtime estimates for quantifying alternative resource allocations.

### **PREDICT: Estimating the Runtime of Large Scale Iterative Algorithms**

Analytical tasks often include machine learning and graph mining algorithms executed on large input datasets. Part of these algorithms are iterative: one or more processing steps are executed repetitively until a convergence condition is met (e.g., ranking, clustering, distributed graph processing). Optimizing cluster resource allocations among multiple workloads of iterative algorithms motivates the need for estimating their runtime, which in turn requires: i) predicting the number of iterations, and ii) predicting the processing time of each iteration. As both parameters depend on the characteristics of the dataset and on the convergence function, estimating their values before execution is difficult.

In this context we propose PREDICT, an experimental methodology for estimating the runtime of a class of iterative algorithms operating on graphs, including algorithms with very different runtime patterns among subsequent iterations (i.e., constant and variable). PREDICT uses sample runs for capturing the algorithm's convergence trend and per-iteration key input features (such as function call counters, message byte counters) that are well correlated with the actual processing requirements of the algorithm on the complete input dataset. After key input features are profiled during the sample run, and extrapolated to the scale of the complete dataset, a cost model is used for translating key input features into runtime estimates. For this purpose, PREDICT introduces a framework for building customizable cost models for network intensive iterative algorithms executing on top of the Bulk Synchronous Parallel (BSP) execution model.

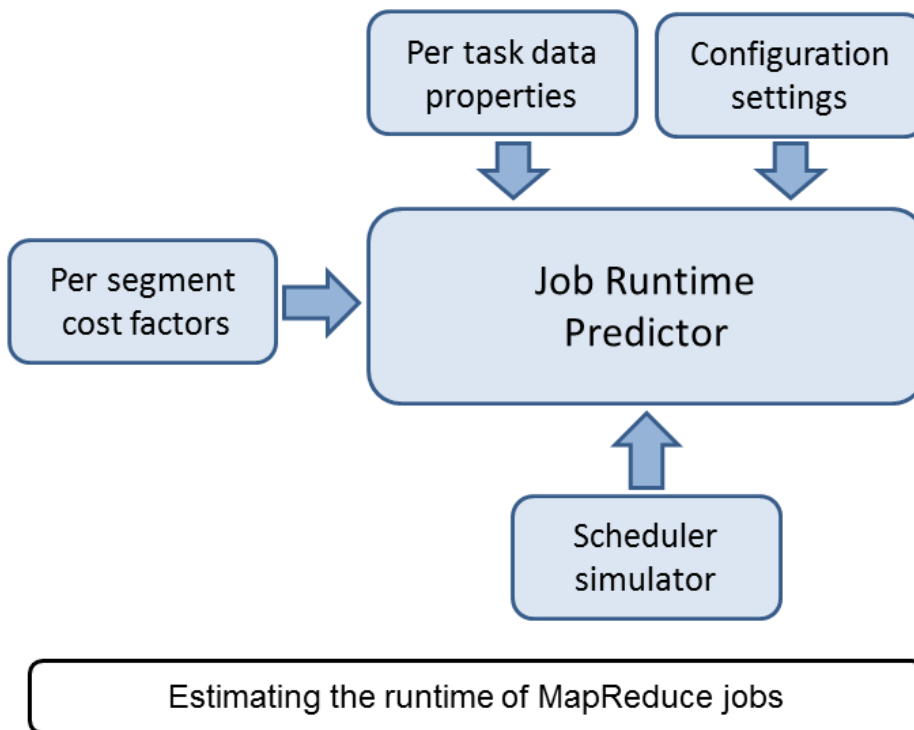


### What-If Analysis and Automated Workload Deployment

With the prevalence of using hardware infrastructure as a service (IaaS) for data management tasks, answering cluster sizing questions and feasibility analysis questions for hypothetical configurations is increasingly important. In this spectrum, questions like: "What instance types and how many machines are needed to meet the performance demands of my analytical application?", and "What hardware configuration can increase my current workload performance by 2x?" are common, especially when transiting workloads from one deployment to another. This project aims on exploiting query profiles in the process of query runtime prediction and automatic workload deployment for big data analysis.

In this work we consider MapReduce workloads that are produced by analytics applications. In contrast to ad hoc query workloads, analytics applications are comprised of fixed data flows that are run repetitively over newly arriving data sets or on different portions of an existing data set. Examples of such workloads include Extract Transform Load (ETL), machine learning pre-processing, and social media analytics. In this context, we propose a prediction approach that predicts the runtime performance for a fixed set of queries running over different input data sets (i.e., same queries, different data).

In contrast to traditional DBMS, modeling query runtime performance for MapReduce data flows using pure analytical models (as in traditional query optimization) is still an open problem. We are investigating approaches that model query performance by combining local machine learning models, which have a powerful mechanism of extracting correlations from historical data, with global analytical models. In particular, our prediction technique splits each query into several segments, where each segment's performance is estimated using machine learning models. Then, these per-segment estimates are plugged into a global analytical model to predict the overall query runtime.



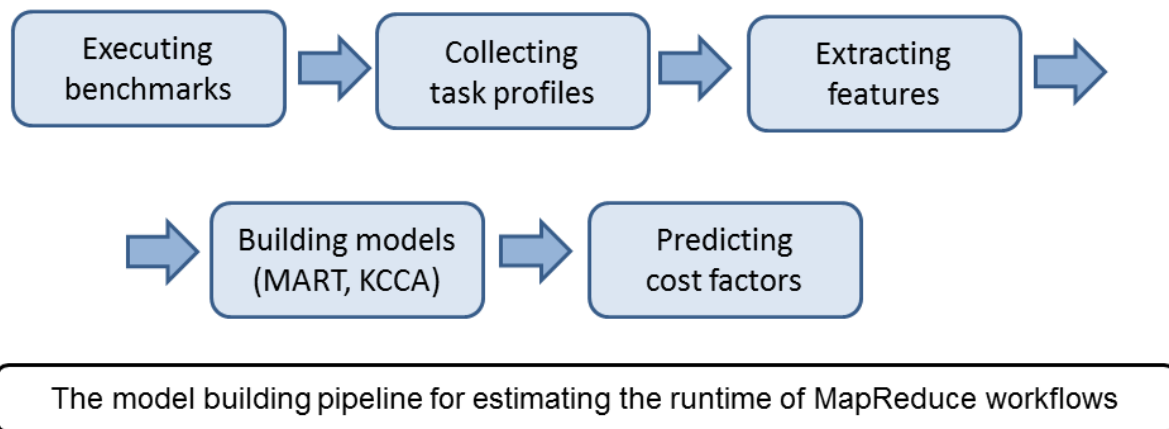
The model building pipeline used in the process of performing What-If analysis consists of the following key modules:

1- Synthetic benchmarks: We built benchmarks for generating the training data starting from the characteristics of the real workload: schema and data properties, operator types, and configuration settings. Synthetic benchmarks synthesize the characteristics of real workloads by generating the minimum number of queries that ensure a good coverage of the query parameter input space.

2- Collecting real workload profiles: We collect real workload profiles as workloads are being executed, and we use both synthetic benchmark profiles and real workload profiles (if available) in training the models.

3- Profiling and feature extraction: We are collecting a variety of input features that are useful in the prediction process from the query profiles generated through non-intrusive dynamic instrumentation.

4- Building models: We have developed the infrastructure for building non-linear regressive models that can estimate the runtime of analytical queries given the input features we collected in the profiling phase. We use algorithms such as Multiple Additive Regression Trees (MART) and Kernel Canonical Correlation Analysis (KCCA) for building prediction models at multiple levels of query segment granularities.



**People involved:** Dr Adrian Popescu