Adaptive Execution of Continuous and Data-intensive Workflows with Machine Learning

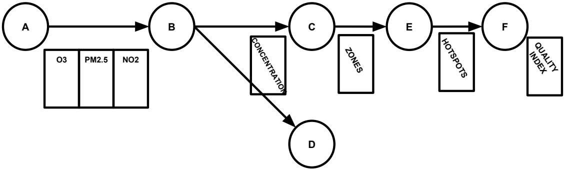
Sérgio Esteves, Helena Galhardas, and Luís Veiga

INESC-ID, Instituto Superior Técnico, University of Lisbon, Portugal

Context

- There is a growing need to analyze, process and store data
- Organizations frequently resort to the composition of data processing workflows
- Workflow applications can be continuous
- Examples:
  - measuring the impact of social business
  - assessing fire risk
  - detecting gravitational-waves
  - predicting earthquakes
- WMS orchestrates the execution of wf applications
  - traditionally enforce strict temporal synchronization
  - not the most desirable in a no. of cases

Motivation - Assessing Air Quality Health Index

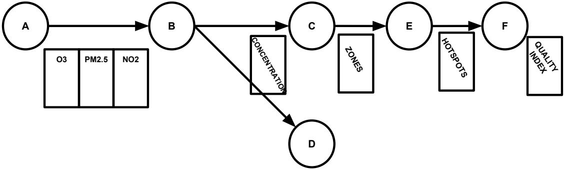


- Executed every fixed interval with a wave of data fed from sensors
- Most part of the time, sequential waves would not change the workflow result
  - pollution stable during most part of the day and night, changing more significantly at rush hours
- Resources are wasted

Contribution

- Adaptive workflow model and framework to enable asynchrony in continuous and incremental processing
- Machine learning to assess the worthiness of executing a processing step
- Trade-off result accuracy with resource savings

Workflow Model - Asynchrony



- step B is only executed after step A produces sufficient output that makes step B execution meaningful

wave	A	B
1	execute	
2	execute	execute
3	execute	

Workflow Model - Output error

- Postponing the execution of processing steps **introduces divergence** (error) in the values, as opposed to the synchronous model

wave	sync output	async output	error
1	10	10	0
2	20	10	ε
3	30	30	0

- Error functions can be provided through an API
  - general implementations are provided
- To guarantee correctness, users should define max_{ε}, so that $\varepsilon \leq max_{\varepsilon}$

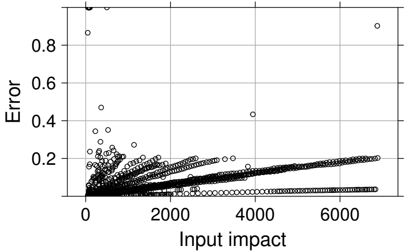
Workflow Model - Input impact

- Error ε of a step is generally correlated with the input of that step
- We define the input impact ι as a metric to characterize the input
- We provide an API through which custom functions can be defined
  - general implementations are provided

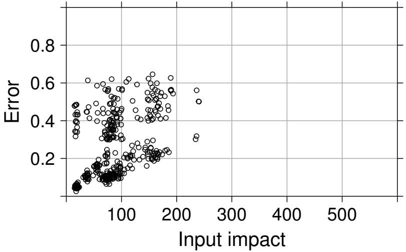
Workflow Model - Generality of the Model

- Applications that exhibit regular input patterns over a period of time (no random or uncorrelated input/output over time)
  - This class of applications is actually commonplace in continuous workflow processing

Workflow Model - Correlation between input impact/error



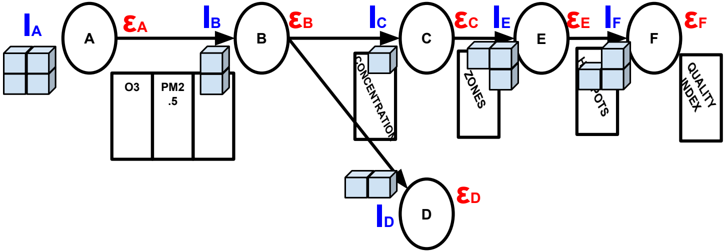
r=0.065



r=0.31

- Pearson correlation coefficient r (1: total linear correlation, 0: no linear correlation)
- Many tasks do not exhibit linear nor trivial correlations
- Machine Learning to handle a wide spectrum of patterns

Learning approach

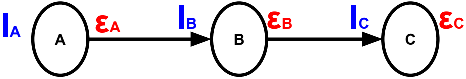


- Learn dataflow patterns: correlation between input variation (ι) and corresponding output error (ϵ)
- Predict when ι causes ε to be above the maximum tolerated error threshold (max_{ε})
- If so, input is significant and task should be triggered

Learning approach - Classification

- Classification algorithms to predict when $\varepsilon > max_{\varepsilon}$.
  - Bayes Network, J48 tree, Logistic, Neuronal Network, Random Forest (RF), and Support Vector Machine (SVM).
- SVMs and RF outperformed all others in most scenarios
  - RF performs better with default parameterization, especially in the presence of unbalanced datasets with variable relation patterns
- Problem falls into supervised learning and multi-label classification
- During a training phase, the WF is run synchronously to build a model

Learning approach - Training phase



$$h\left(\underbrace{\begin{array}{c} \begin{array}{ccc} I_A & I_B & I_C \end{array} \\ \begin{bmatrix} 694.86 & 601.6 & 498.3 \\ 191.24 & 886.1 & 498.3 \\ 278.13 & 1071.4 & 498.3 \\ 433.78 & 233.78 & 664.24 \\ 551.53 & 523.8 & 956.52 \end{bmatrix} \end{array}}_{X}\right) = \underbrace{\begin{array}{c} \begin{array}{ccc} \varepsilon_A > \max_\varepsilon ? & \varepsilon_B > \max_\varepsilon ? & \varepsilon_C > \max_\varepsilon ? \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}}_{Y}$$

Learning approach - Test phase

- During a test phase, we assess the quality of the produced model
- Our model can be adjusted to favor results on either **recall** or **precision**

High recall means that we are avoiding the existence of false negatives; i.e., the percentage of times the model estimated incorrectly that $\varepsilon <= max_{\varepsilon}$

- leads to higher error compliance

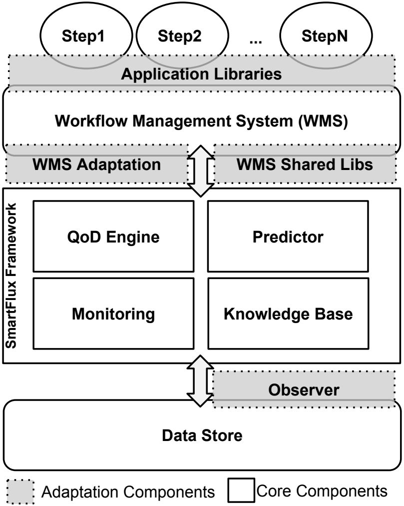
High precision means that we are avoiding to estimate incorrectly that $\varepsilon > max_{\varepsilon}$

- leads to higher resource savings

Learning approach - Execution Phase

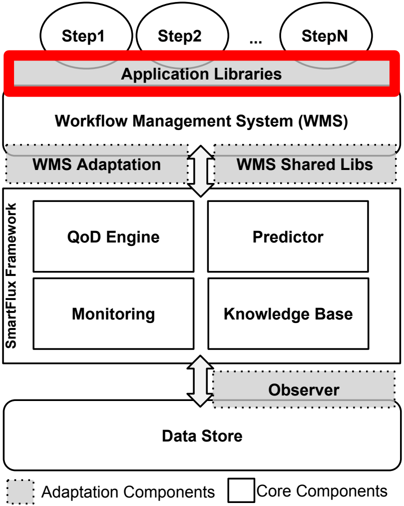
- After accurate model is built, the execution phase takes place
  - WF starts running asynchronously in an adaptive way
- At each wave, the input impact of each step is given to the classifier, which in return indicates the steps that should be executed

Middleware Framework



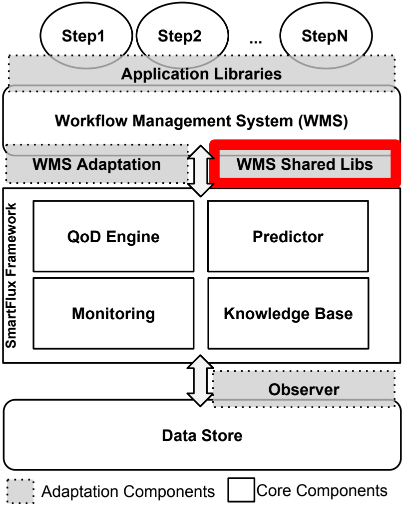
- Operates between WMS and data store
- Steps must share data through the underlying data store
- Adaptation components connect SmartFlux with WMS and data store

Middleware Framework - Adaptation



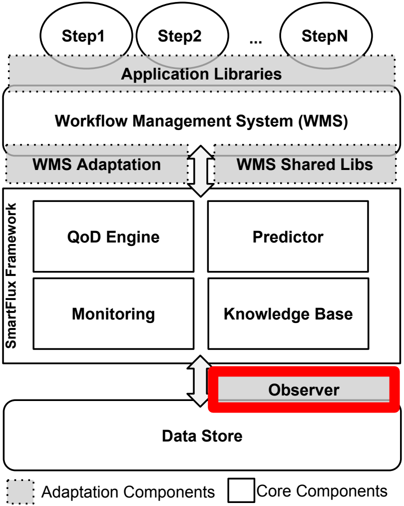
- Adapted database client libraries
- Applications need to be slightly modified (eg, changing package names in the imports of Java classes)

Middleware Framework - Adaptation



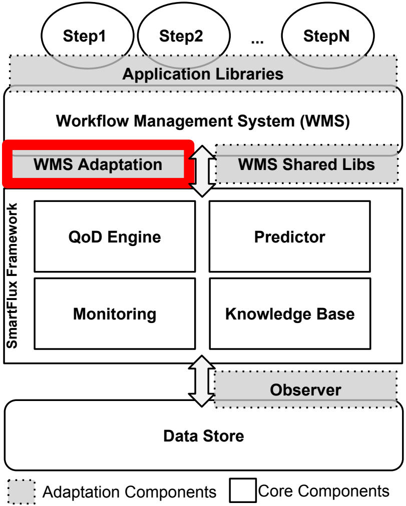
- Libraries at the WMS level (e.g., pig scripts or any other high-level language that must be interpreted/compiled by the WMS)
- Provides transparency to applications

Middleware Framework - Adaptation



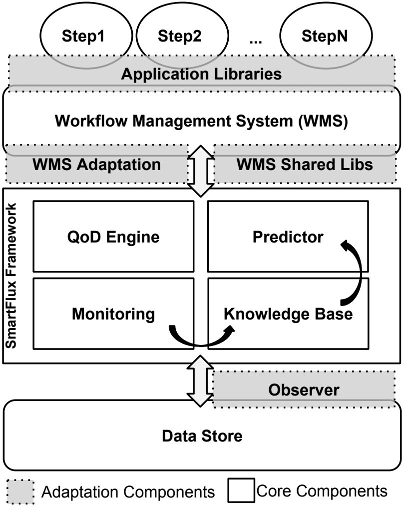
- Custom code that is executed at the database level (e.g., co-processors in HBase or triggers in Cassandra)
- Transparent to applications

Middleware Framework - Adaptation



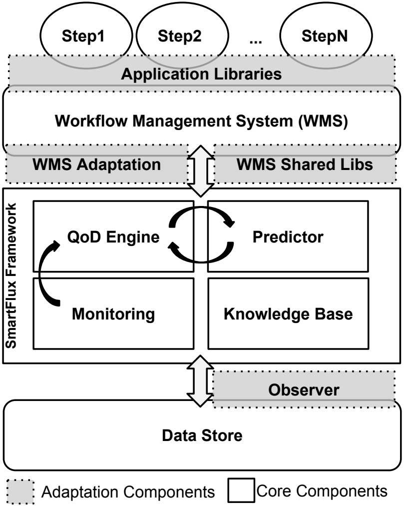
- SmartFlux issues triggering notifications
- WMS notifies of new waves and steps completion

Middleware Framework - Training



- Monitoring intercepts database requests and updates Knowledge Base with statistical information
- Predictor builds a classification model based on input impact/error metrics stored in Knowledge Base

Middleware Framework - Executing



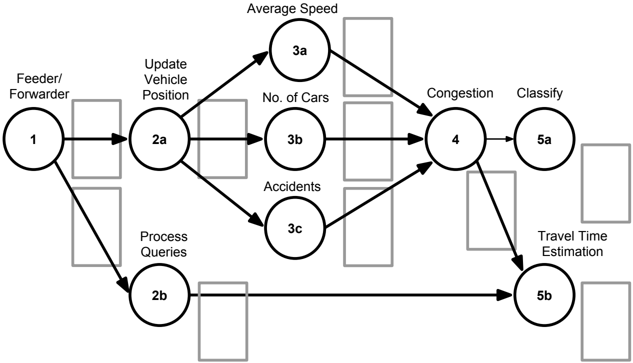
- Monitoring intercepts database requests, computes the input impact and sends it to the QoD Engine
- QoD Engine queries the Predictor and gets in return the subset of steps that should be executed

Evaluation

- SmartFlux was integrated with a widely deployed WMS, Apache Oozie
- As data store, we adopted Apache HBase
- SmartFlux uses MEKA, a multi-label classification library based on the well known WEKA Toolkit
- 6 machines with commodity hardware

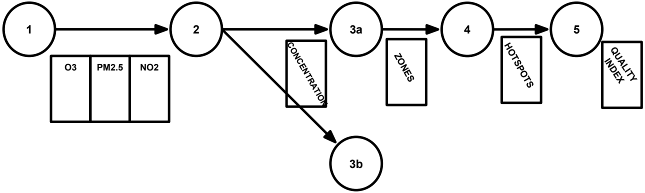
Evaluation - Scenarios

Linear Road Benchmark

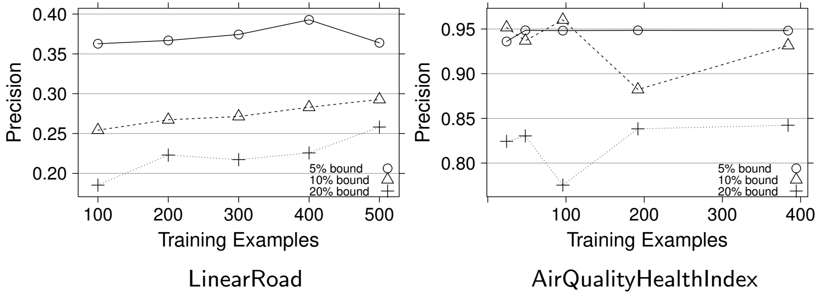


Evaluation - Scenarios

Air Quality Health Index

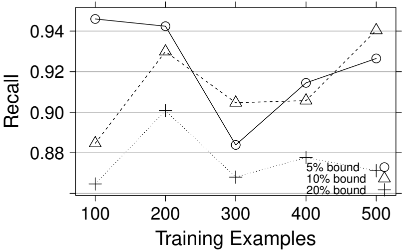


Evaluation - Precision

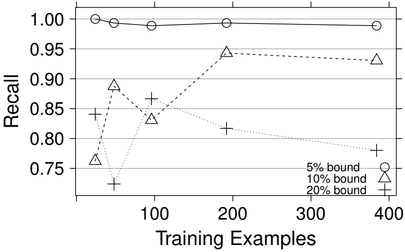


- AirQuality outperformed LinearRoad (higher resource efficiency)
- Classifier generalized better input/error relations in AirQuality

Evaluation - Recall

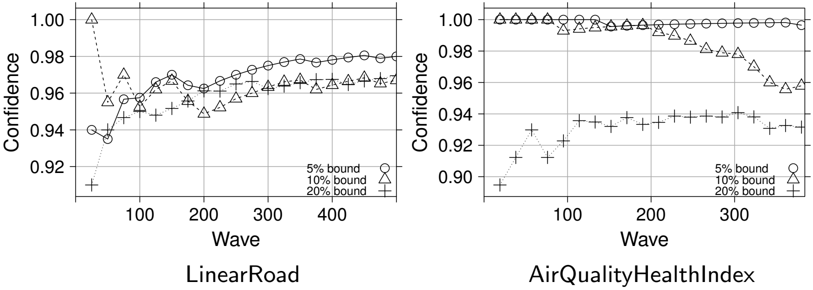


LinearRoad



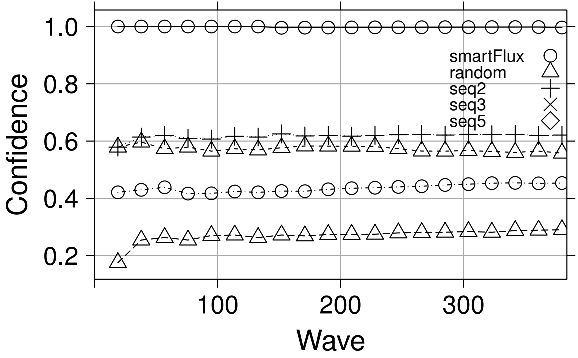
AirQualityHealthIndex

Evaluation - Confidence in respecting error bounds



- In steady state:
  - over 95% with an error bound of 5%
  - over 90% overall

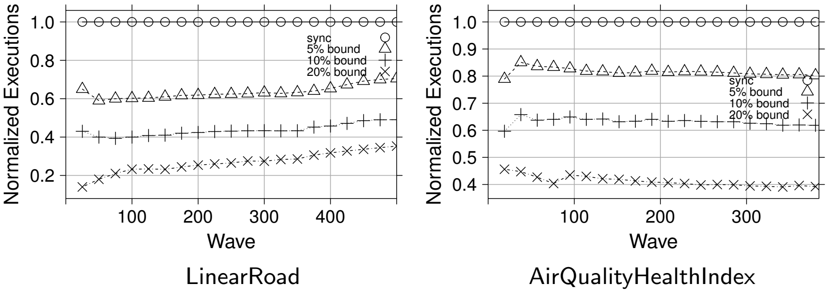
Evaluation - Confidence with simpler techniques



AirQuality

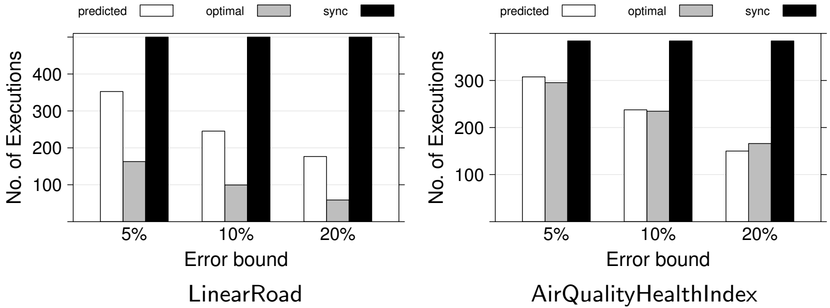
- random: probability of executing or not a step is equal
- step: executes steps at every 2, 3, or 5 waves
- None can provide confidence level close to that of SmartFlux

Evaluation - Savings



- With a 5% error bound: 40 and 20% of saved executions
- With a 20% error bound: upto 80 and 60% of savings

Evaluation - Savings



- Higher efficiency in AirQuality, reflecting the high precision obtained

Conclusion

- Presented an adaptive workflow model and framework for continuous processing
  - explores trade-off between result accuracy and resource savings
  - provides probabilistic guarantees
- Our solution makes use of Machine Learning to learn correlation patterns between input and output error and guide WF execution in a resource-efficient manner
- Evaluation indicates substantial resource savings in exchange of allowing small errors to exist
  - up to 40% savings with a maximum error of 5%

Thanks for your attention.

Questions?

Email: sesteves@gsd.inesc-id.pt