Optimizing Machine Learning Workloads with Experiment Databases

Behrouz Derakhshan
behrouz.derakhshan@dfki.de

Database Systems and Information Management Group
TU Berlin

https://www.dima.tu-berlin.de/

Intelligent Analytics for Massive Data
German Research Center for Artificial Intelligence

https://www.dfki.de/

Motivation

- Online data science platforms allow
  - Write solutions (typically via Jupyter Notebooks)
  - Share solutions
  - Learn from existing solutions

- Examples
  - Google Collaboratory
  - Kaggle
  - Coursera

kaggle

coursera

Motivation

- Users incorporate existing solutions
- Repetition of (similar) operations

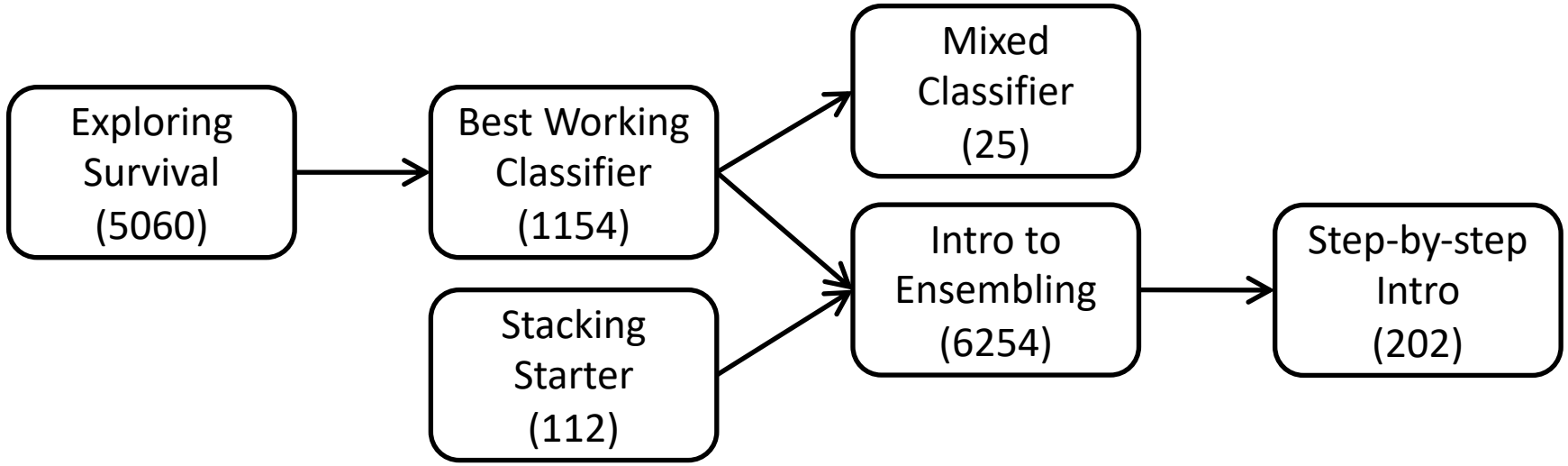


Figure 1: Kaggle's notebook relation[1]

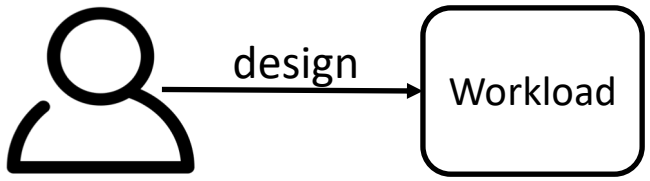
1. https://www.kaggle.com/c/titanic

Goal

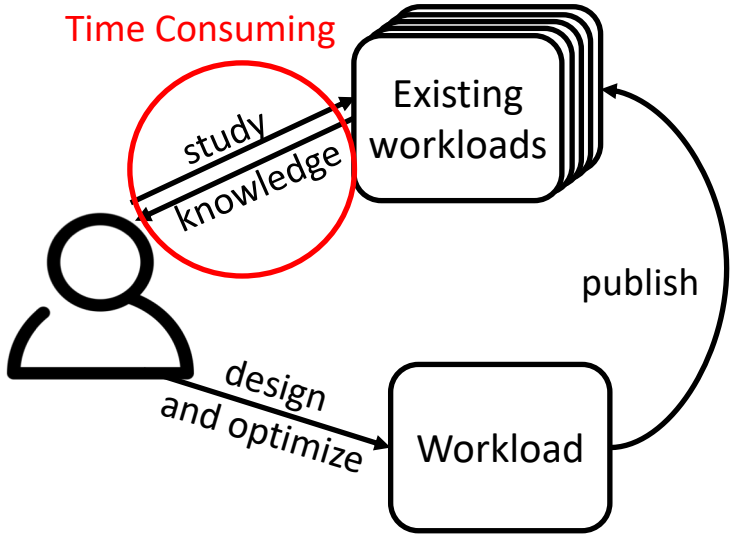
Efficient design and execution of machine learning (and data science) workloads in collaborative environments

Problem with Existing Solutions

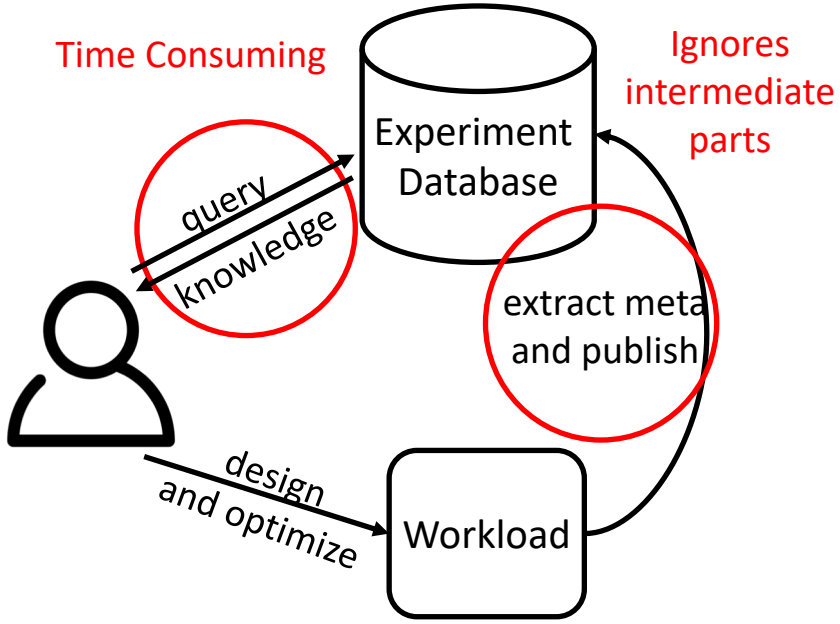
Ignore existing knowledge



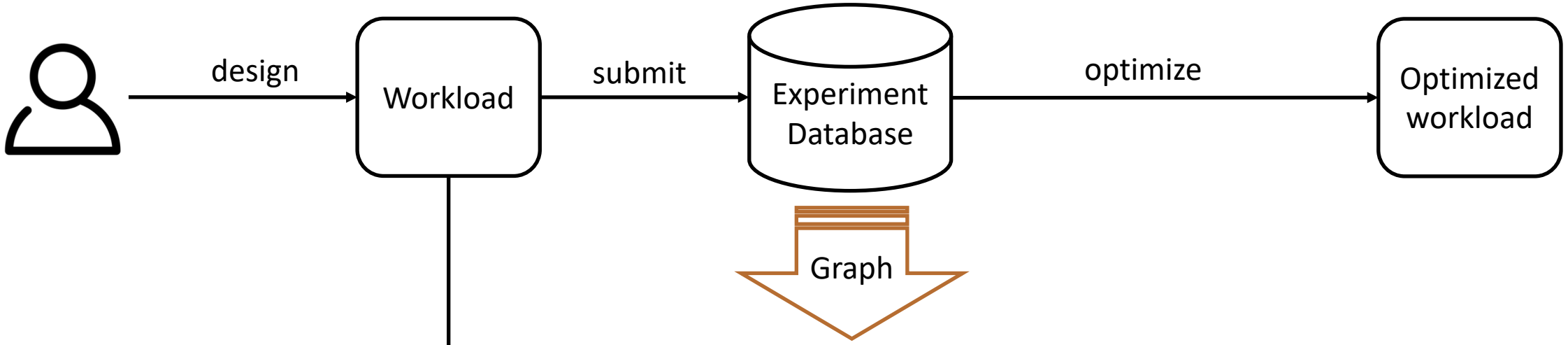
Learn by reading



Through Experiment Databases (e.g., OpenML)



Solution



Optimizations:

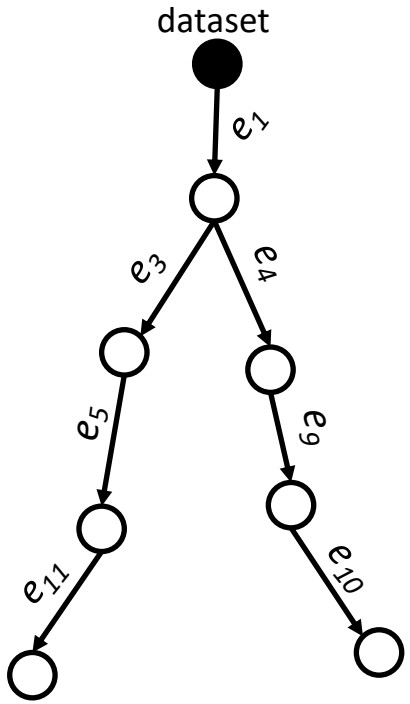
1. **Reuse**
2. **Warmstarting**
3. **Improved Hyperparameter tuning**

- Directed graph
- Vertices = artifacts
- Edges = operations

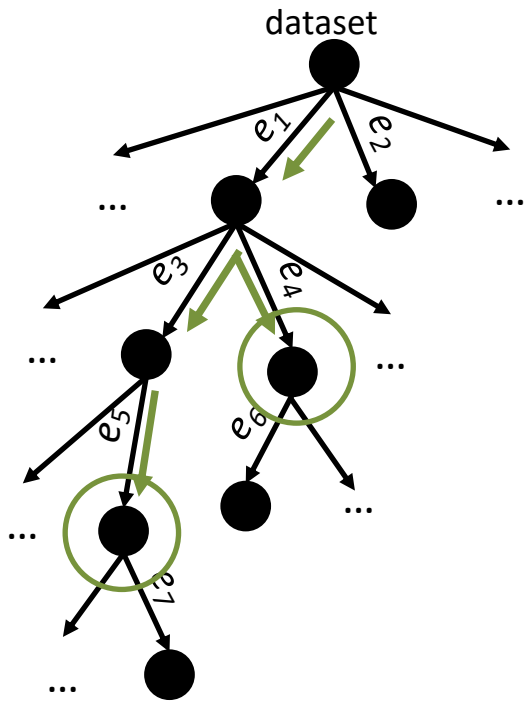
Reuse Optimization

- Edges are uniquely identified
- For new workloads, traverse the Experiment Graph

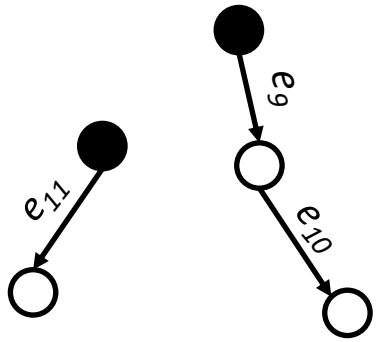
Workload



Experiment Graph



Optimized workload

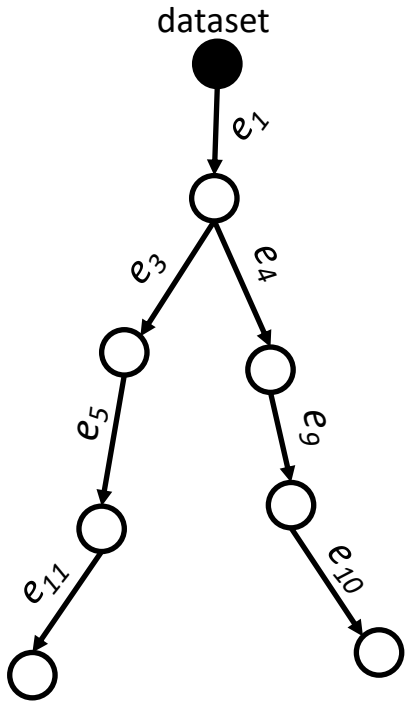


Improves total execution time

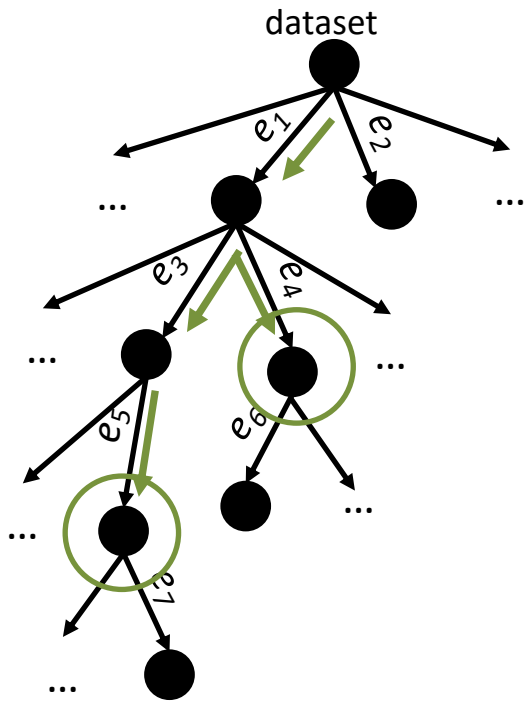
Model Warmstarting

- Warmstart a training operations with the best model from the same model group
- **Model group:** set of all models (of same type) trained on one artifact

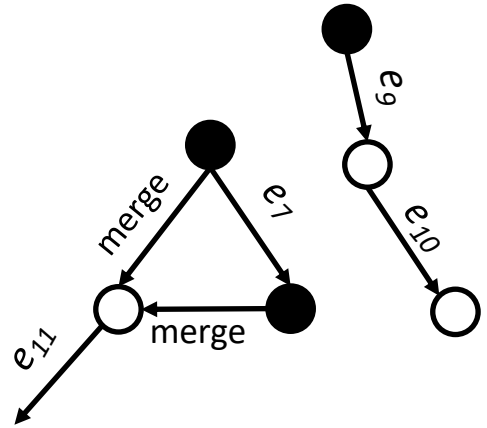
Workload



Experiment Graph



Optimized workload

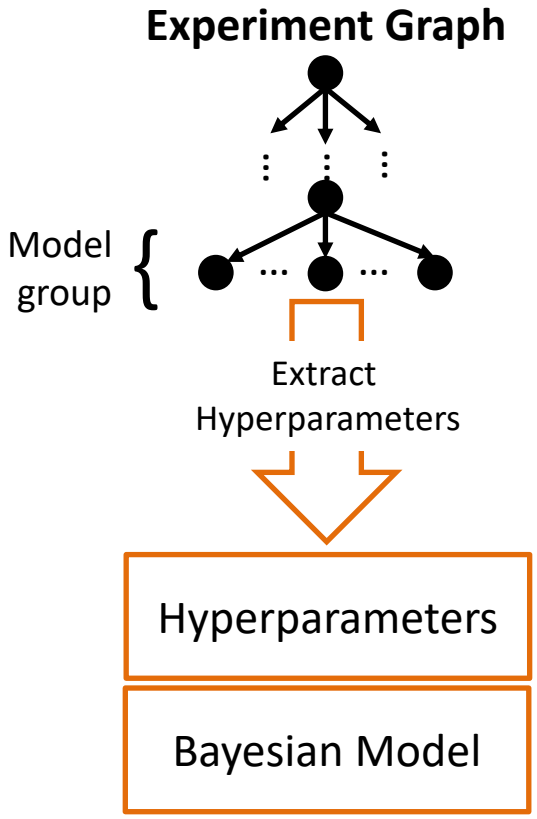


Improves total execution time and *model quality*

e_7 and e_{11} are both model training operations

Improved Hyperparameter Tuning

- Challenges in Hyperparameter tuning
  - Search space definition (grid, random, and Bayesian tuning)
  - Requires several initial runs until promising models are trained (Bayesian tuning)



Improves total execution time and model quality

- Propose search space
- Propose new/existing hyperparameters
- Initialize Bayesian tuning

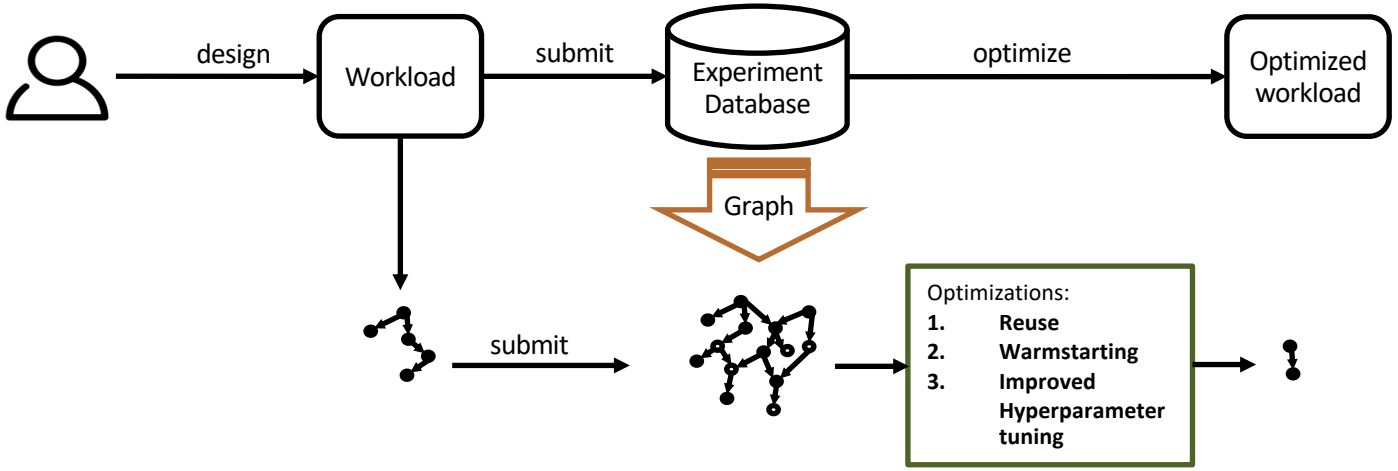
Progress

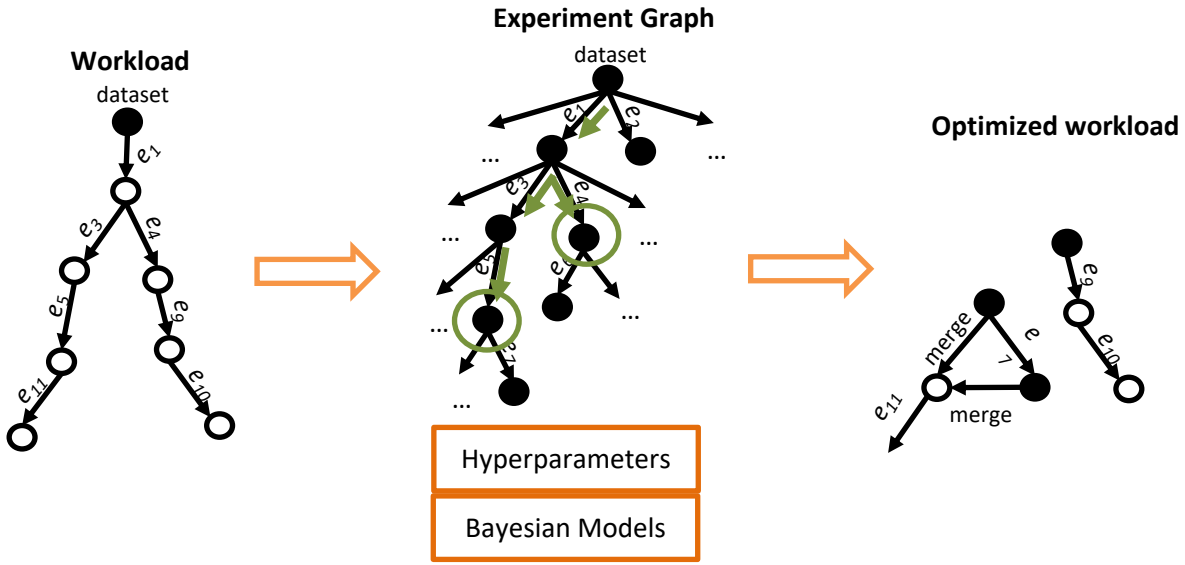
- Mar18–Aug18: Initial experiments and first draft of Experiment Databases paper
- Aug18-Oct18: Submission of Continuous Deployment Paper to EBDT 2019
- Dec19: Acceptance of Continuous Deployment paper at EDBT 2019
- Nov19-Feb19: Prototype including Pandas and Scikit-learn

Next Steps

- Feb19-May19: Finalize the paper
  - When to reuse
  - When to warmstart
- Feb19-May19: Finalize the experiments
  - 2 Kaggle Challenges
  - Entire OpenML (scikit-learn pipelines)
- **May19 or Jul19: Submit to VLDB 2020**
- Aug19-Oct19: On leave
- Oct19-Jun20: Distributed Experiment Graph and Automatic Machine Learning
- ***Jul20: Submit the next work to VLDB/SIGMOD 2021***

Summary





- Optimizing machine learning workloads in collaborative environments by
  - Reuse
  - Model warmstarting
  - Improved Hyperparameter tuning
- By using a graph representation of the database
  - *Graph Construction*
  - *Materialization Strategy*

References

1. https://data.world
2. https://colab.research.google.com
3. https://www.kaggle.com/
4. https://www.coursera.org/
5. Vanschoren, Joaquin, et al. "OpenML: networked science in machine learning." *ACM SIGKDD Explorations Newsletter*15.2 (2014): 49-60.
6. Vartak, Manasi, et al. "Model DB: a system for machine learning model management." *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. ACM, 2016.
7. Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." *International Conference on Learning and Intelligent Optimization*. Springer, Berlin, Heidelberg, 2011.
8. Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms In Advances in neural information processing systems, pages 2951–2959, 2012.
9. Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." *Journal of Machine Learning Research* 13.Feb (2012): 281-305.
10. Vartak, Manasi, et al. "MISTIQUE: A System to Store and Query Model Intermediates for Model Diagnosis." *Proceedings of the 2018 International Conference on Management of Data*. ACM, 2018.
11. Souvik Bhattacherjee, Amit Chavan, Silu Huang, Amol Deshpande, and Aditya Parameswaran. Principles of dataset versioning: Exploring the recreation/storage tradeo. Proceedings of the VLDB Endowment, 8(12):1346–1357, 2015.

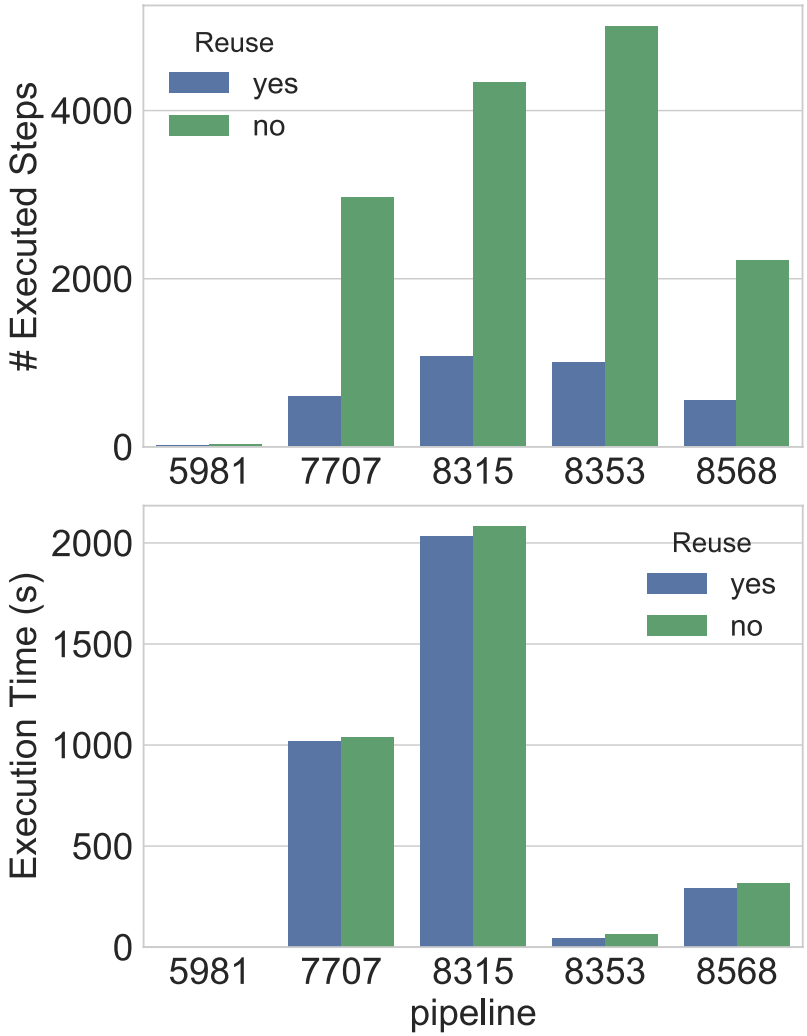
Scope of the optimizations

- Optimizations are limited to a task
- A task defines the goal of the machine learning or data science work
- Examples of tasks and workloads:
  - A competition on the Kaggle,
  - An assignment in Coursera
- Each task is associated with one connected component in the experiment graph

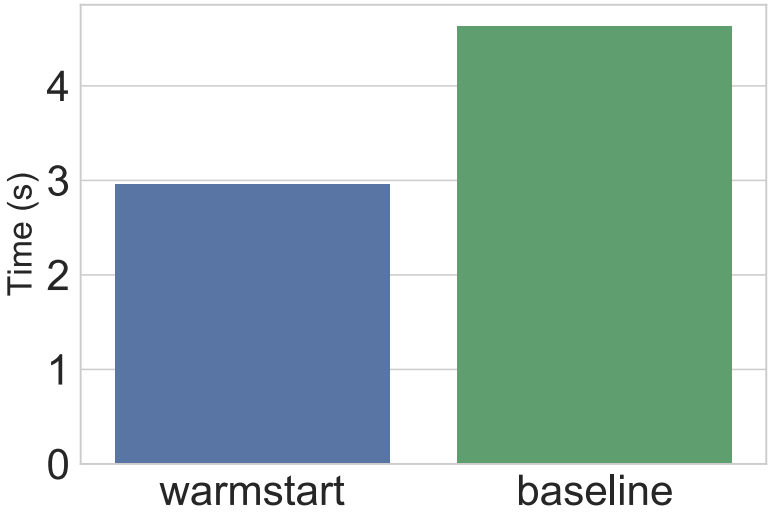
OpenML Pipelines

id	operations	#exec
5981	Impute→ Scale → logistic regression	11
7707	Impute → Onehot encode → Scale → Variance Threshold → SVM	594
8315	Impute → Onehot encode → Variance Threshold → Random Forest	1084
8353	Impute → Onehot encode → Variance Threshold → SVM Forest	1000
8568	Impute → Onehot encode → Variance Threshold → Random Forest	555

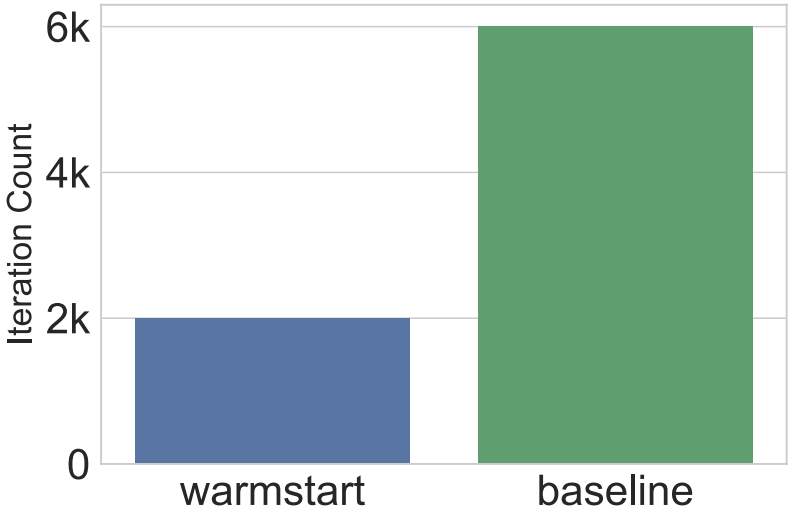
Reuse Optimization



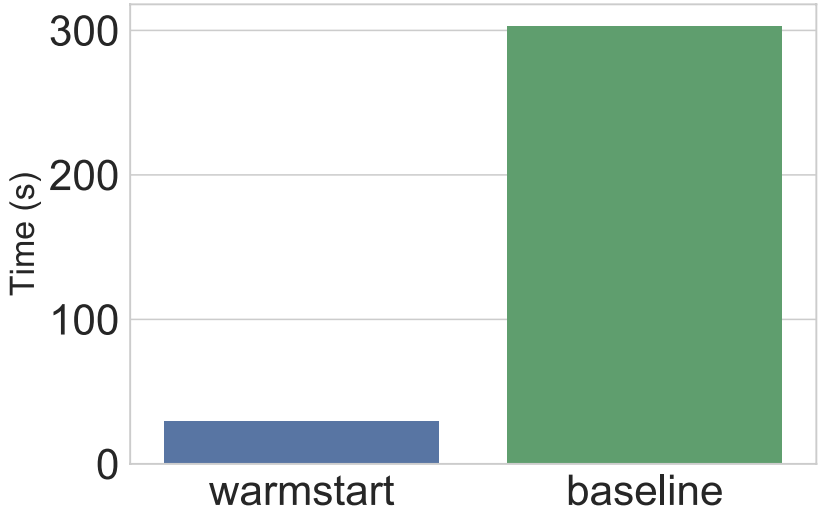
Model warm-starting Optimizations



(a) Total training time (#5981)



(a) Total iteration count (#5981)



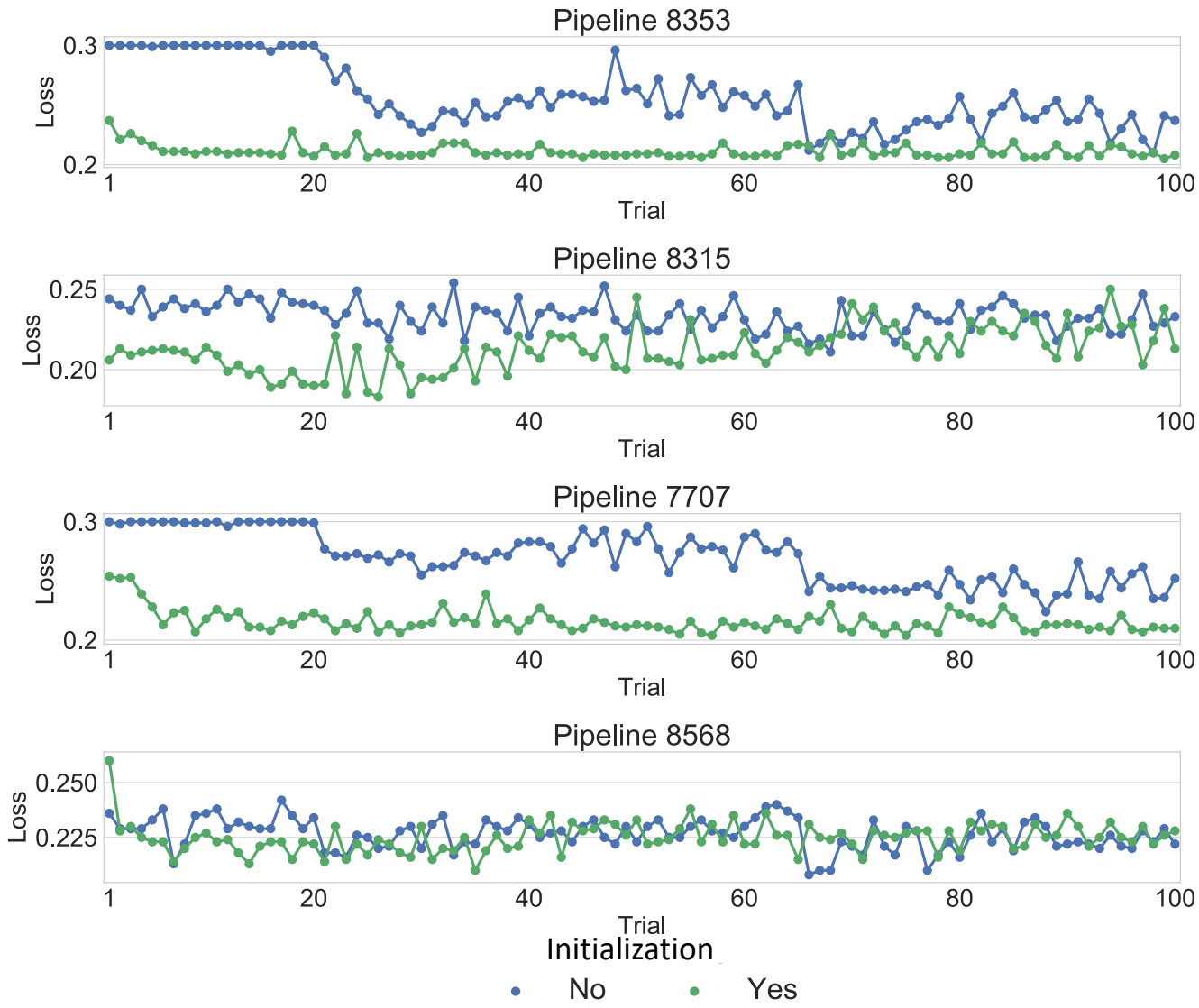
(a) Total training time (#8568)

Effect of the warm-starting optimization on the total training time and iteration count. In (a) and (b) we train 11 logistic regression models and in (c) we train 555 random forest model from the configurations that exist in the experiment data.

Bayesian hyperparameter search

- Initialize Bayesian search with values from the experiment database

- The figure shows the loss value gained for a Bayesian tuning process with budget of 100 (with/without initialization)



Progress

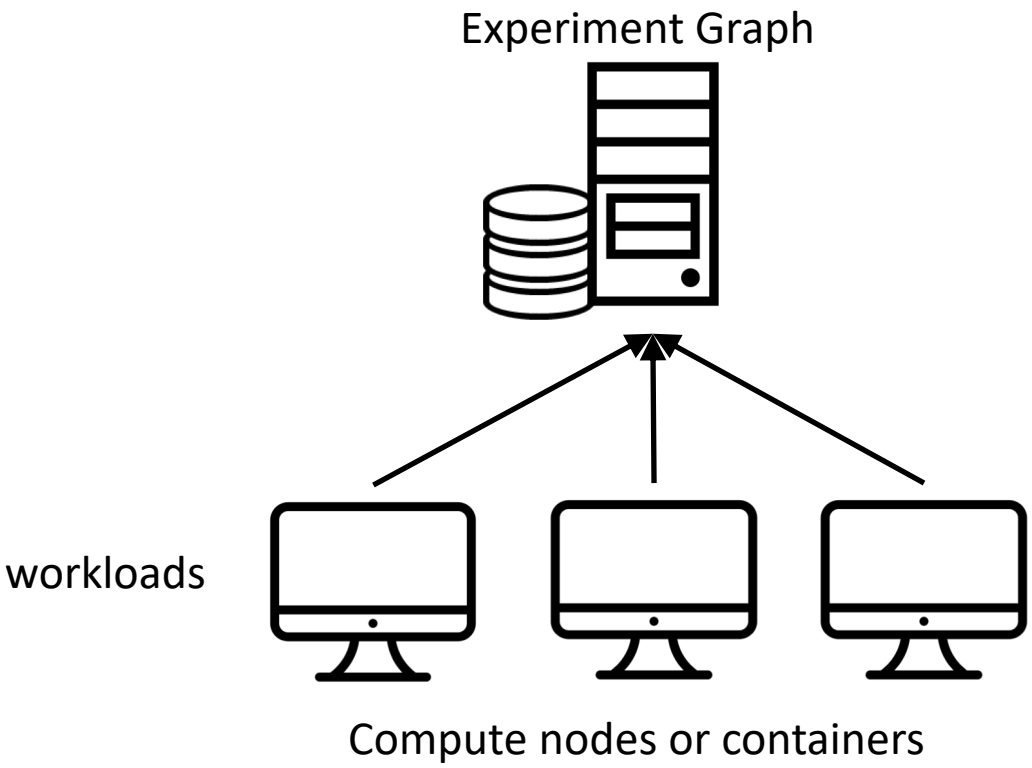
Implementation

- Promising initial experiment results
- Prototype of Experiment Graph with Pandas dataframes and scikit-learn
- Experiments on Kaggle use case
  - Home Credit Default Risk (In progress)
  - Another competition

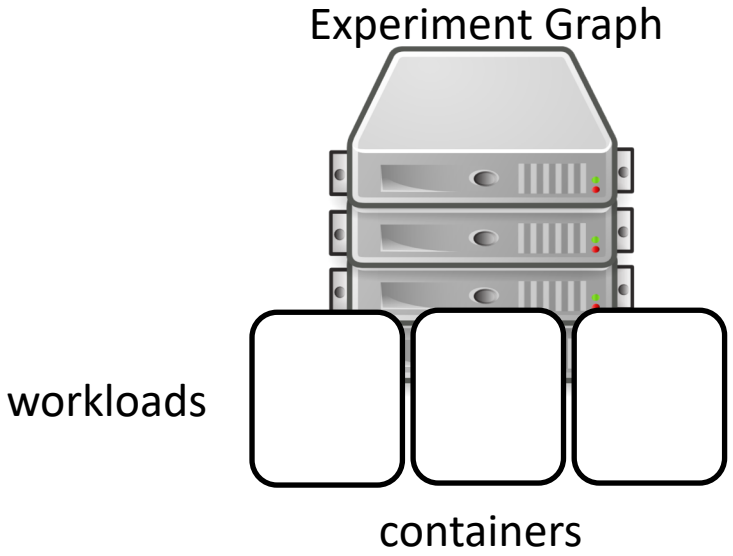
Questions

- Single node with a large memory or a cluster
- Are these experiments adequate
  - OpenML
  - 2 Kaggle competitions

Remote Experiment Graph



Single Node



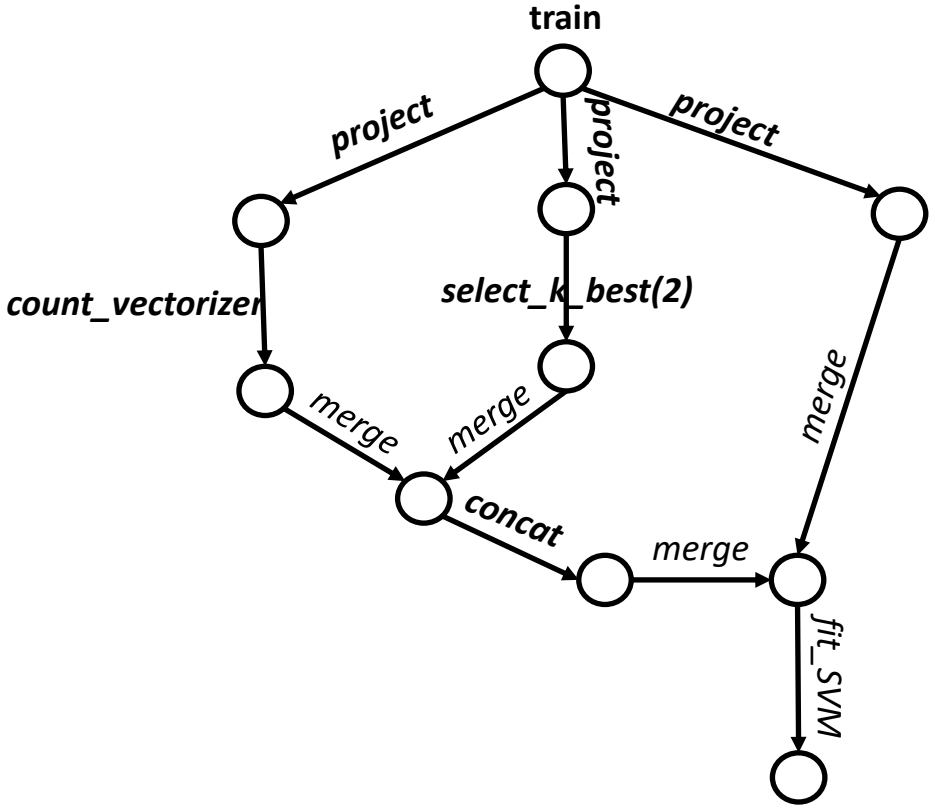
Example script

```
import numpy as np
import pandas as pd

from sklearn import svm
from sklearn.feature_selection import SelectKBest
from sklearn.feature_extraction.text import CountVectorizer

train = pd.read_csv('../input/train.csv')
print train.columns # [ad_description,ts,u_id,price,y]
vectorizer = CountVectorizer()
count_vectorized = vectorizer.fit_transform(train['ad_description'])
selector =  SelectKBest(k=2)
top_features = selector.fit_transform(train[['ts','u_id','price']],
                                      train['y'])
X = pd.concat([count_vectorized,top_features], axis = 1)
model = svm.SVC()
model.fit(X, train['y'])
```

Graph Representation



merge operation:

- Logical operation with a run time of 0 that combines two nodes

Materialization of Artifacts

- Problem statement: given a size budget, T(representing the total storage capacity), materialize the artifacts that result in the smallest weighted recreation cost of the experiment graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the weighted recreation cost is:

$$\mathcal{W}(\mathcal{G}) = \sum_{e \in \{e' \in \mathcal{E} \mid dest(e') \notin \mathcal{MV}\}} e.f \times e.t$$

- Where $\mathcal{MV}$ represents the set of materialized artifacts
- Each edge e is labeled with <*f,t*>, representing the frequency and the average execution time of the operation.
- Vertices are are labeled with <*s*>, representing the size of the of the artifact

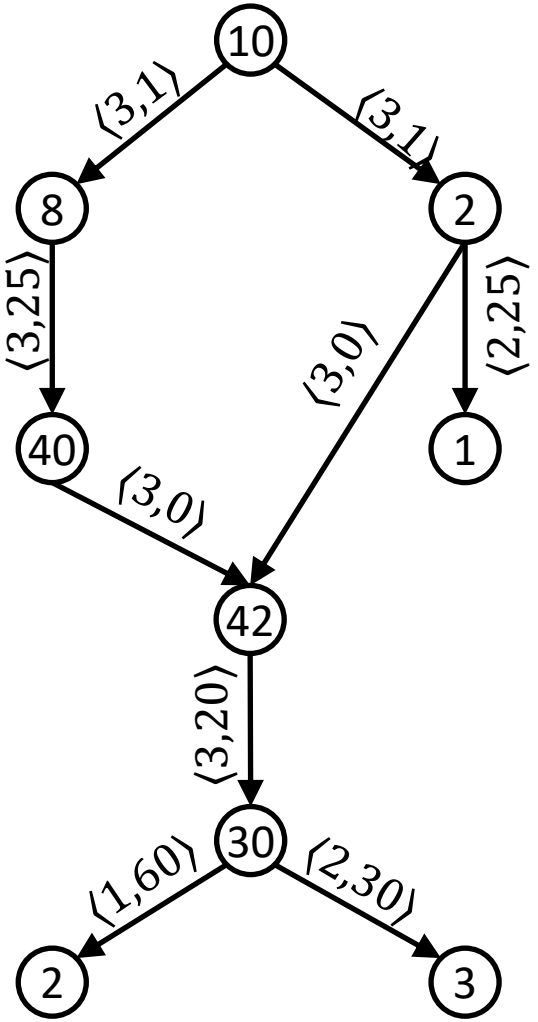
Materialization of Artifacts

- Materialize the root node
- While storage limit is not reached:
  - Find the vertex v, with maximum value of $\frac{\rho(\mathcal{G},v)}{v.s}$

$$v^* = \underset{v \in \mathcal{V}}{\operatorname{argmax}} \frac{\rho(\mathcal{G}, v)}{v.s}$$

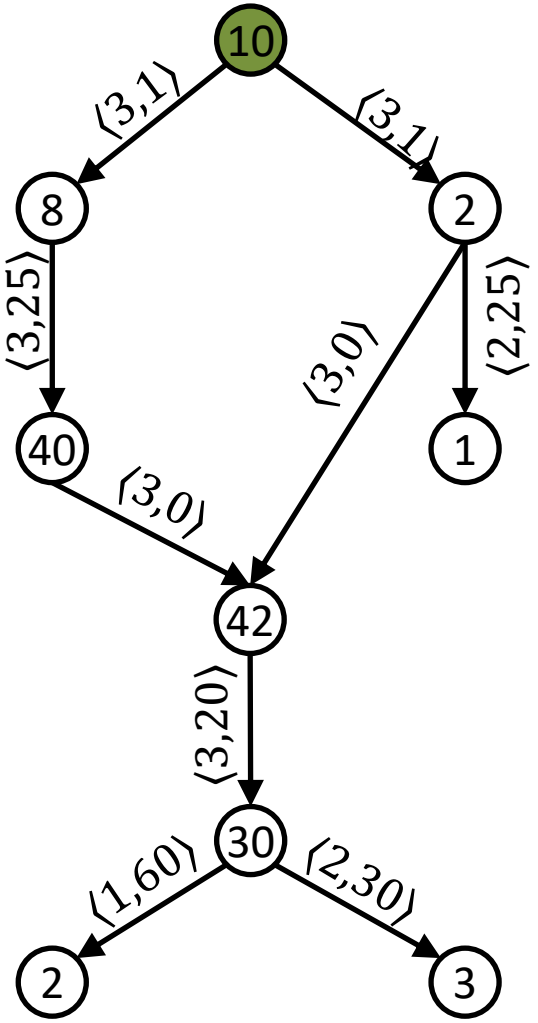
Where,

$$\rho(\mathcal{G}, v) = \alpha(\mathcal{G}, v) \times \sum_{e \in path(\mathcal{G}, v_0, v)} e.t$$



Materialization of Artifacts

$T = 10$ $max\ size = 55$



Materialization of Artifacts

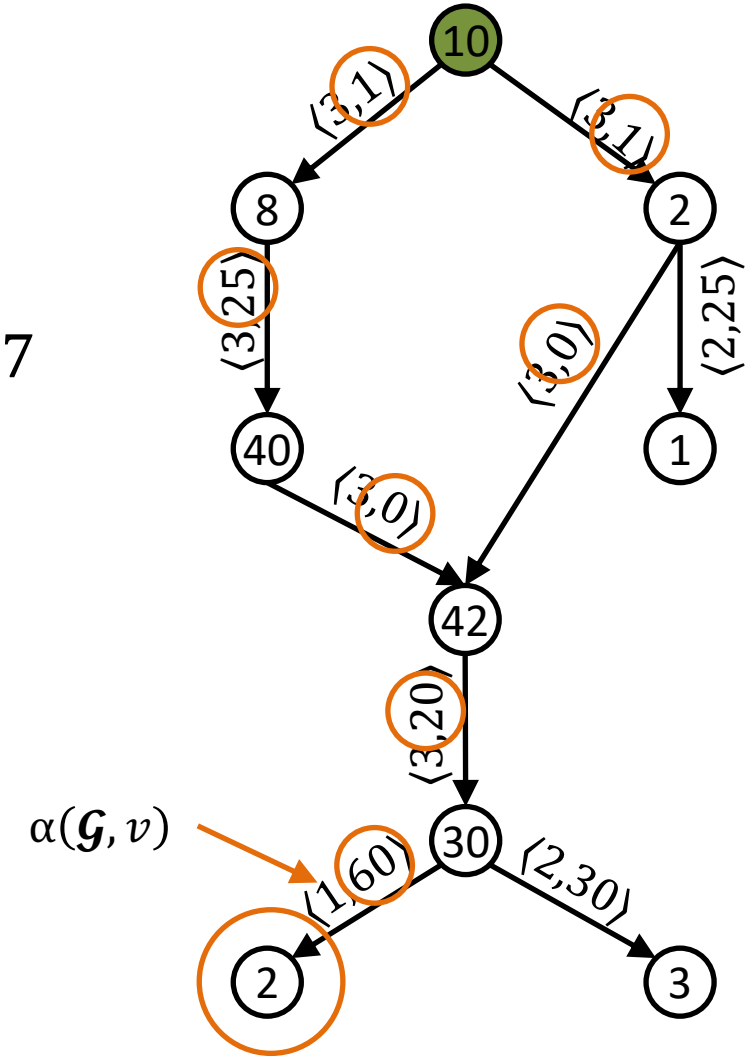
$T = 10$ $\quad max\ size = 55$

$v.s = 2$

$T + v.s < max\ size$

$$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 1 \times (1 + 1 + 25 + 0 + 0 + 20 + 60) = 107$$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 53.5$$



Materialization of Artifacts

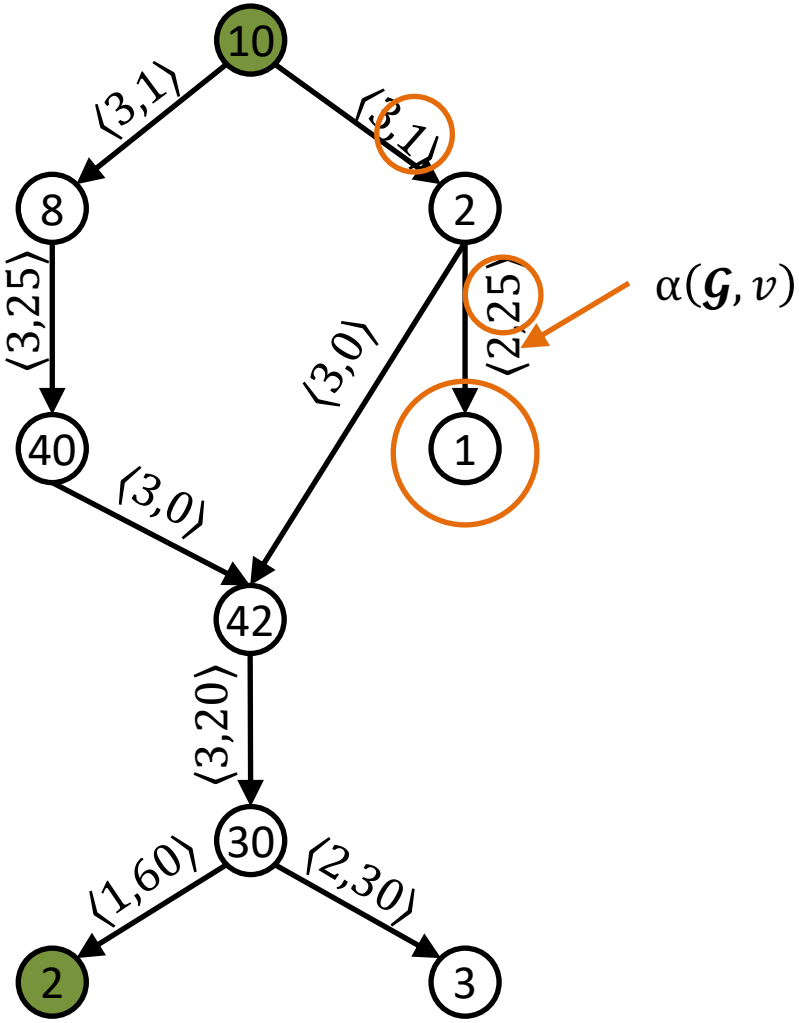
$T = 12$ $\quad max\ size = 55$

$v.s = 1$

$T + v.s < max\ size$

$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 2\times(1 + 25) = 52$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 52$$



Materialization of Artifacts

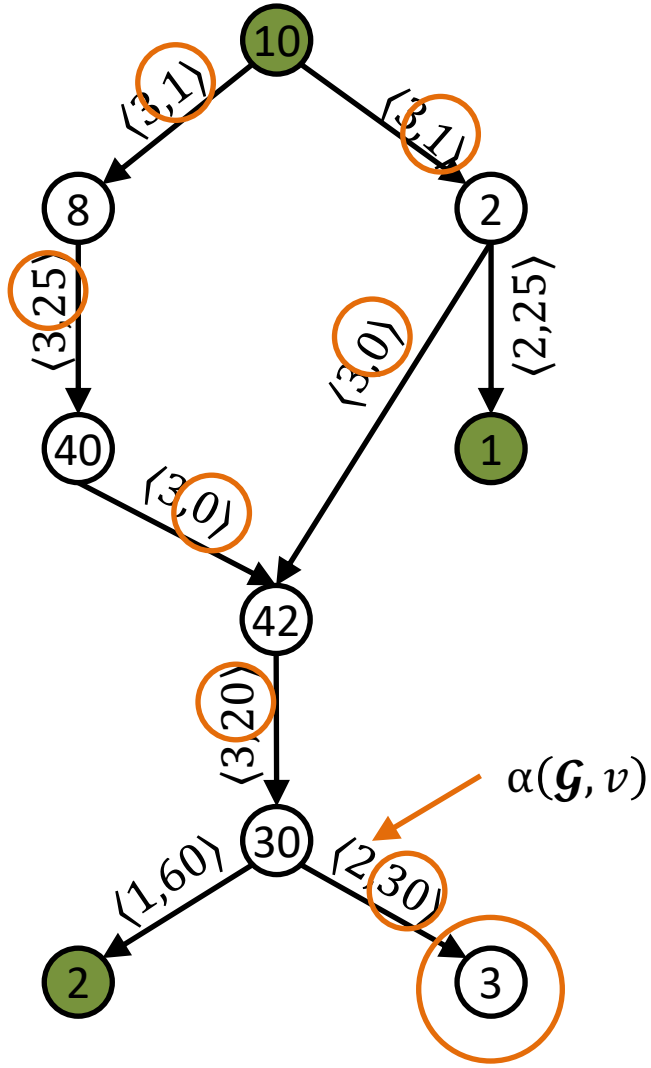
$T = 13$ $\quad\quad$ $max\ size = 55$

$v.s = 3$

$T + v.s < max\ size$

$$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 2\times(1 + 1 + 25 + 0 + 20 + 30) = 154$$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 51.3$$



Materialization of Artifacts

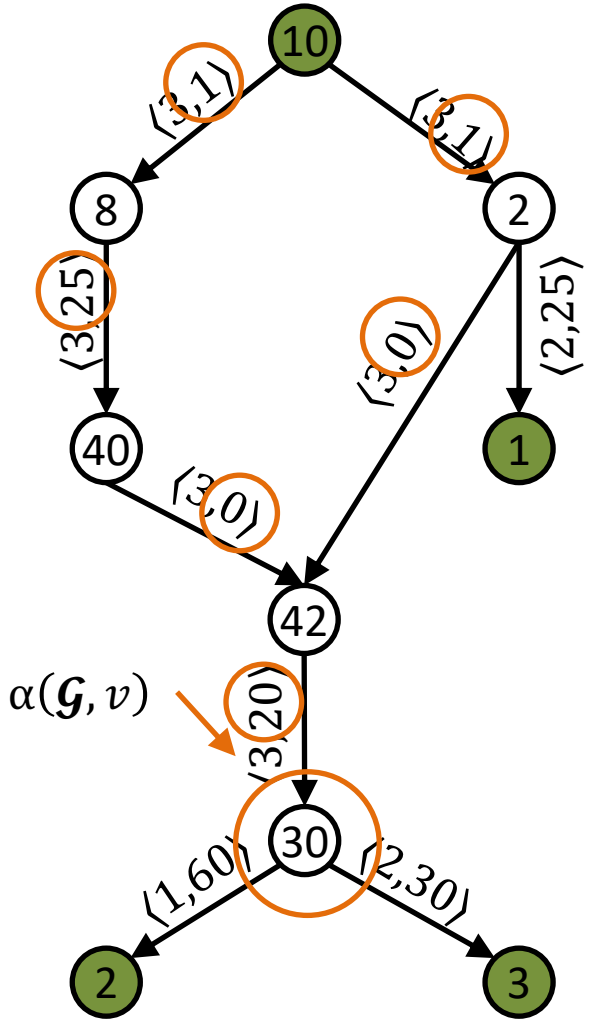
$T = 16$ $\quad max\ size = 55$

$v.s = 30$

$T + v.s < max\ size$

$$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 3\times(1 + 1 + 25 + 0 + 0 + 20) = 141$$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 4.7$$



Materialization of Artifacts

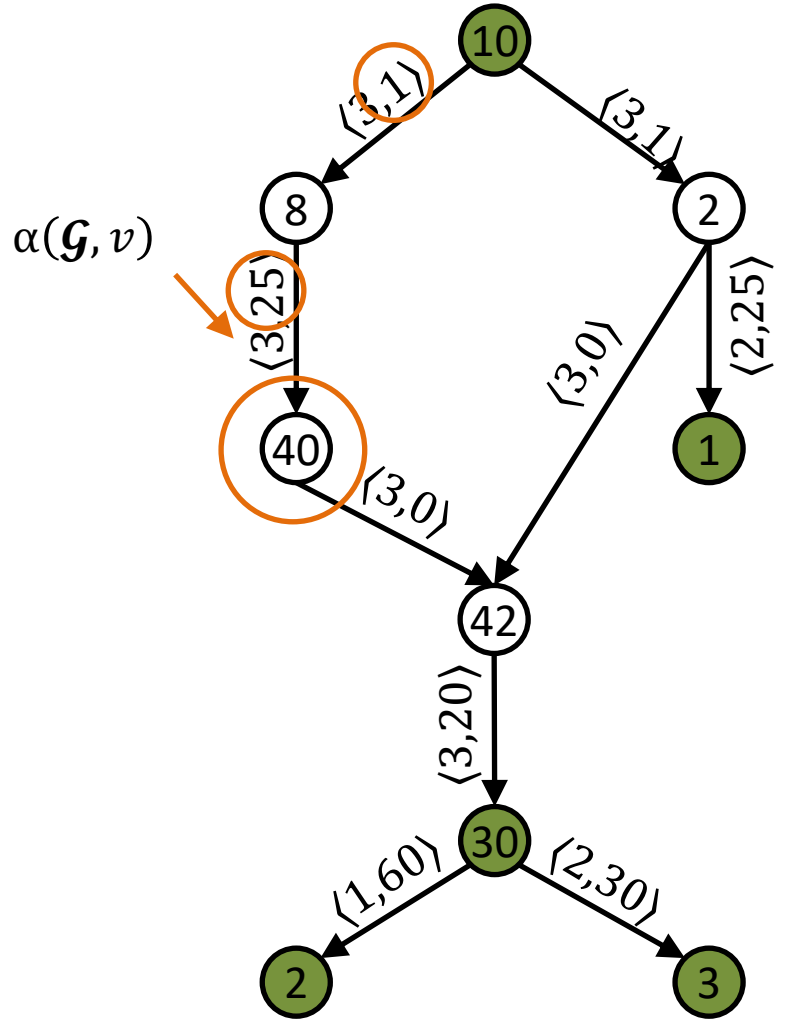
$T = 46$ $max\ size = 55$

$v.s = 40$

~~$T + v.s < max\ size$~~

$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 3 \times (1 + 25) = 78$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 1.95$$



Materialization of Artifacts

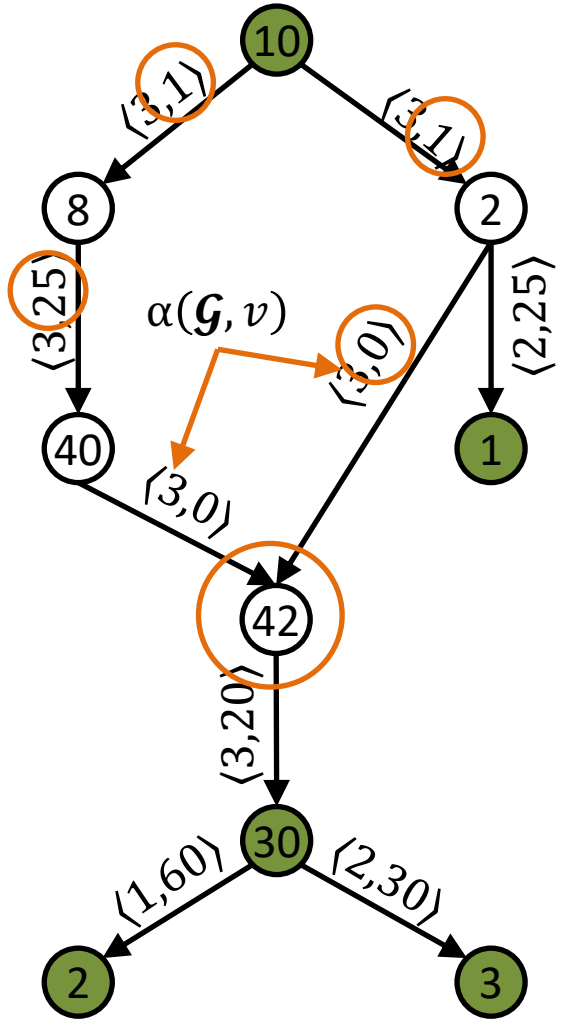
$T = 46$ $max\ size = 55$

$v.s = 42$

~~$T + v.s < max\ size$~~

$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 3\times(1 + 1 + 25 + 0 + 0) = 81$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 1.93$$



Materialization of Artifacts

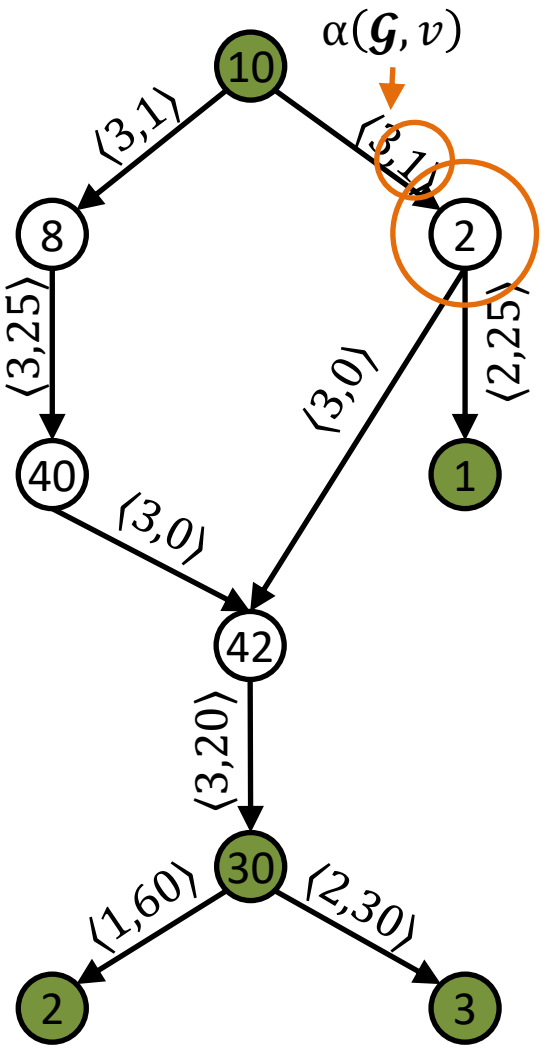
$T = 46$ $\quad max\ size = 55$

$v.s = 2$

$T + v.s < max\ size$

$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 3 \times (1) = 3$

$\dfrac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 1.5$



Materialization of Artifacts

$T = 48$ $max\ size = 55$

$v.s = 8$

~~$T + v.s < max\ size$~~

$\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v) = 3 \times (1) = 3$

$$\frac{\boldsymbol{\rho}(\boldsymbol{\mathcal{G}}, v)}{v.s} = 0.375$$