ANANKE: a Q-Learning based Portfolio Scheduler for Complex Industrial Workflows

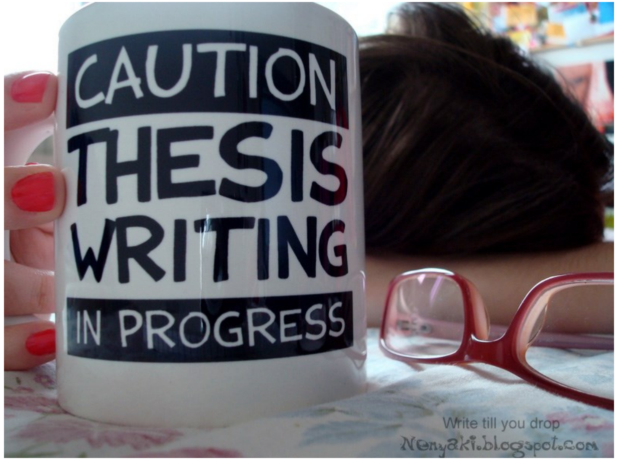
Presentation: Laurens Versluis

Slides: Shenjun Ma

Shenjun Ma
Alexey Ilyushkin
Alexander Stegehuis
Alexandru Iosup
@Large research: https://atlarge-research.com/

Concept: Workflow

- A process/application modeled as tasks with precedence constraints between them
- Example: A thesis
  1. Find a supervisor/topic
  2. Define research questions
  3. Implement a prototype
  4. Perform experiments
  5. Document the results
  6. Defend your work

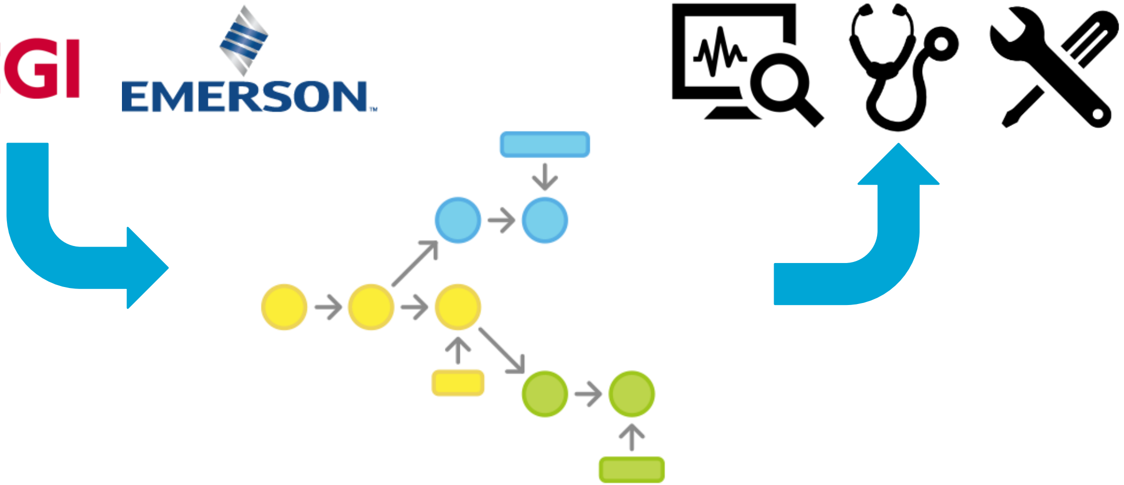


TU Delft

Workflows are used in industrial applications

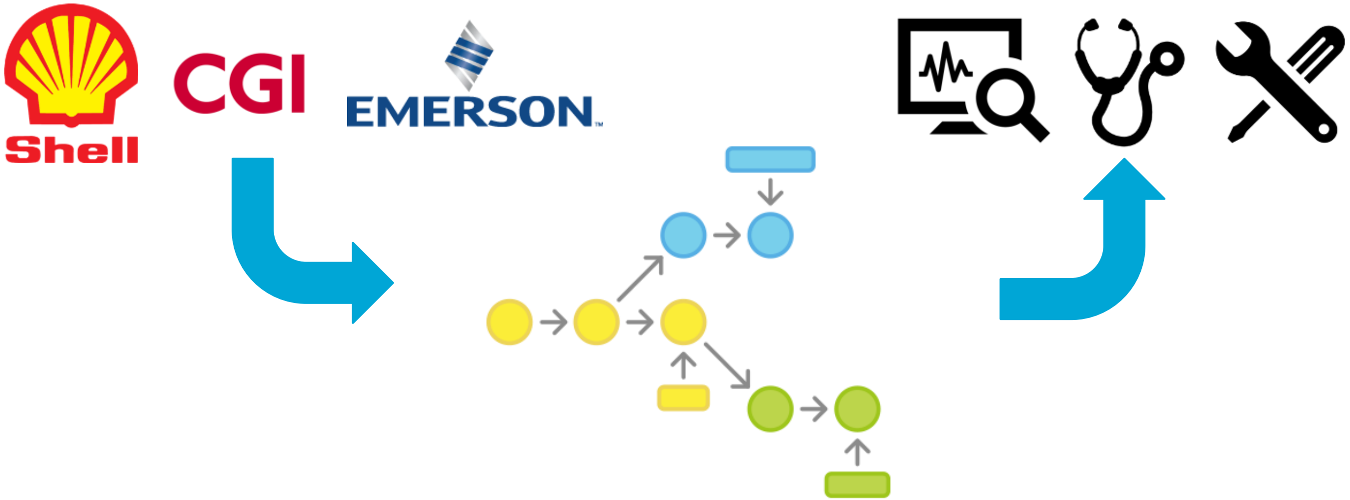
- Sensor data processing is commonly defined as workflows
- Example use-case: monitoring and diagnosis

Shell CGI EMERSON



Workflows are used in industrial applications

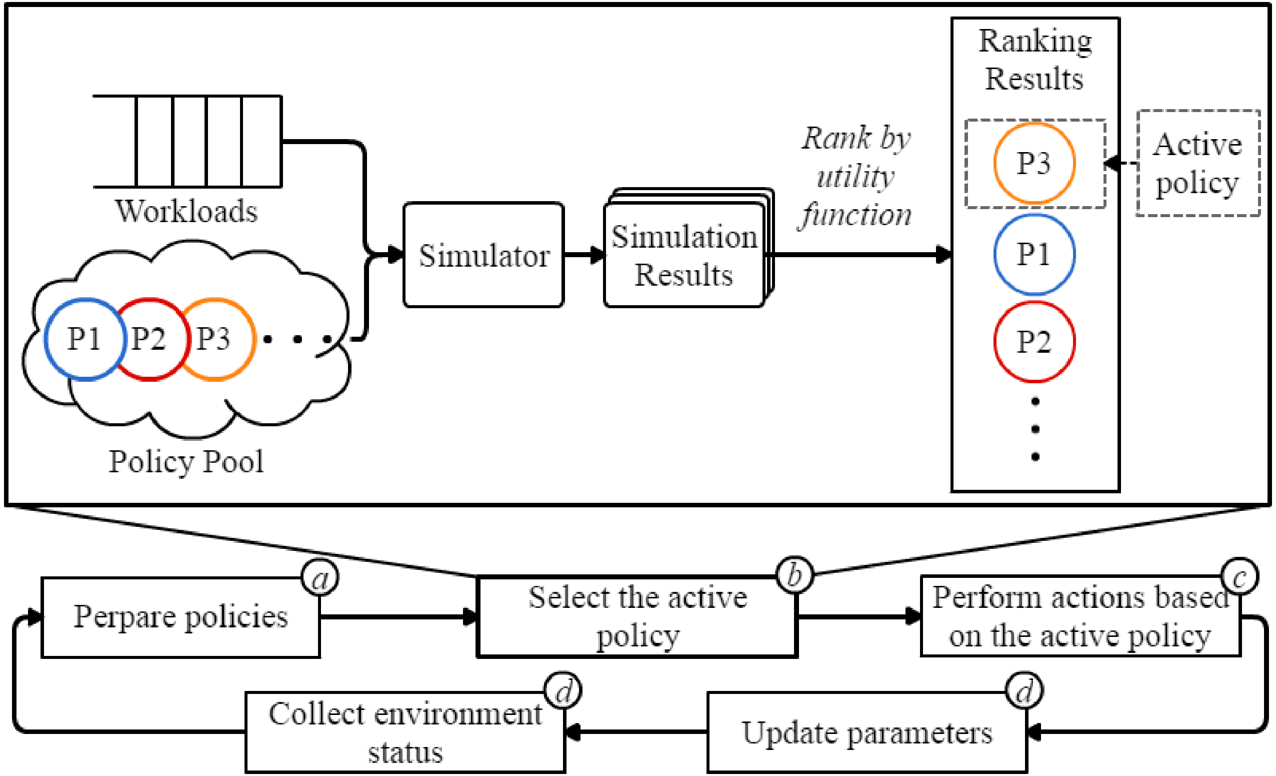
- Sensor data processing is commonly defined as workflows
- Example use-case: monitoring and diagnosis



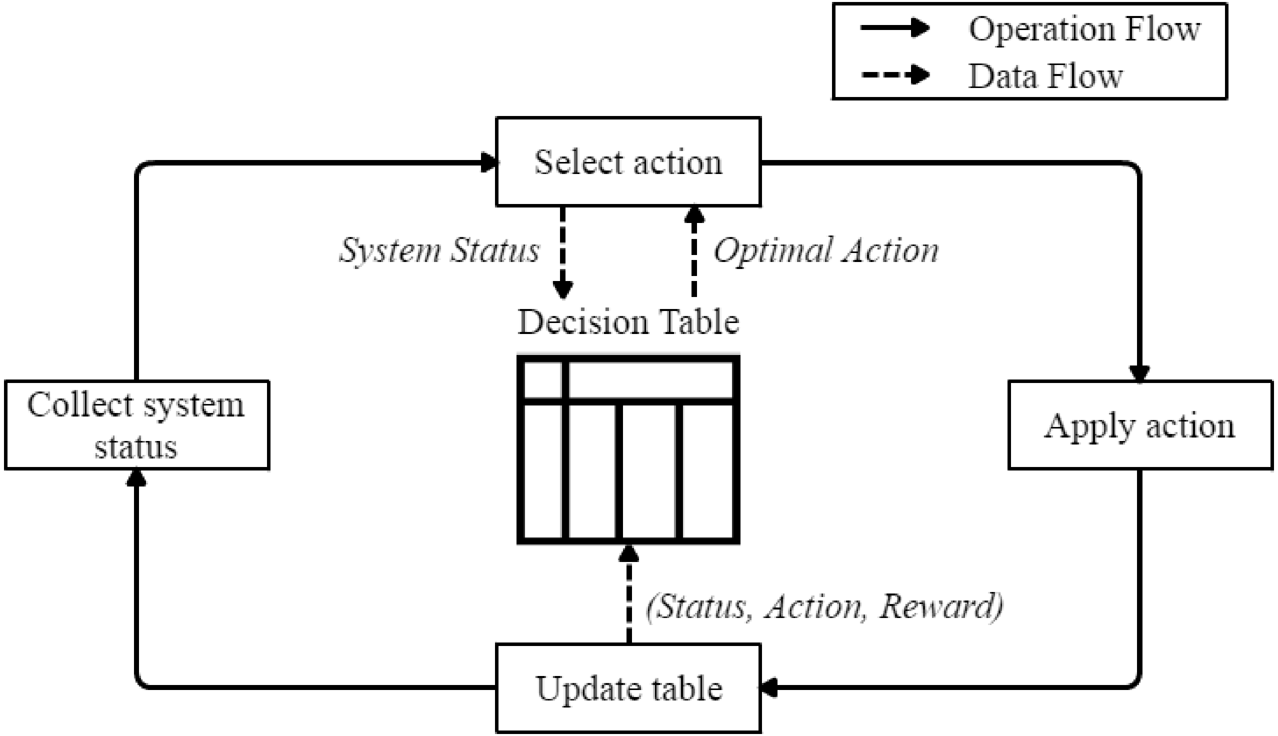
Industrial workflows have special features and requirements

- Analyze and process workflows in production (= in real-time)
- Deadline constraints
  - Monitor asserts in real time
- Exhibit recurrent patterns
  - Sensors collect data with a constant rate
- Workloads may evolve over time
  - New type of sensor data is introduced
  - New type of sensor is applied

Concept: Portfolio Scheduling



Concept: Standard Q-Learning Algorithm



Our approach:

Q-Learning + Portfolio Scheduling

Our approach:

Q-Learning + Portfolio Scheduling

Take the advantage of recurrent patterns

TUDelft

Our approach:

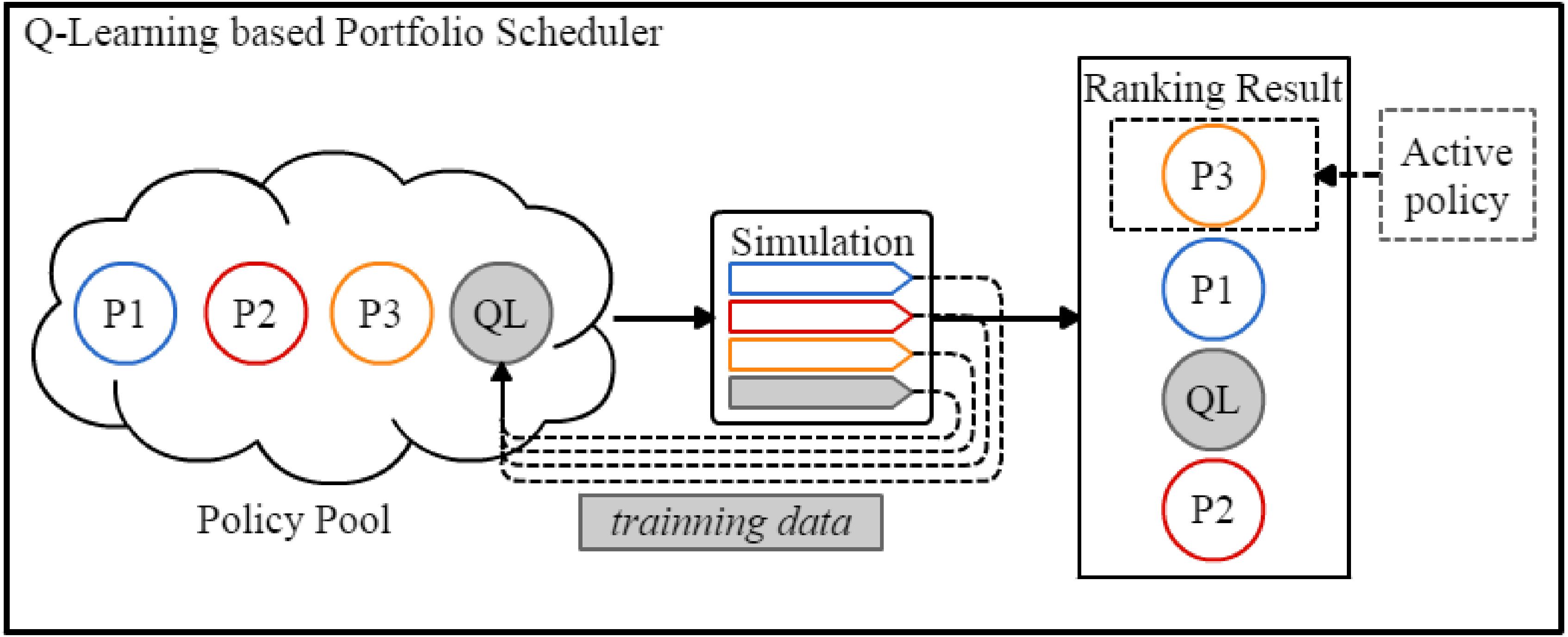
Q-Learning + Portfolio Scheduling

Take the advantage of recurrent patterns

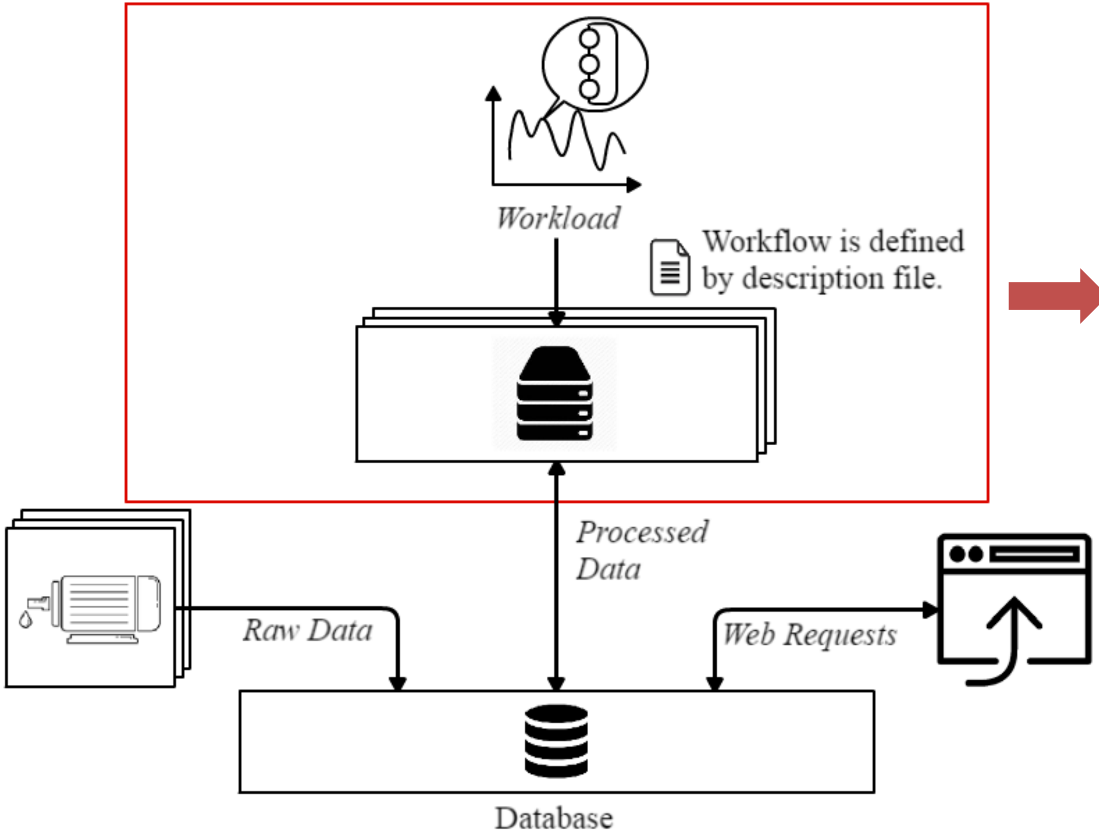
Address the workload evolution and deadline constraints

TUDelft

Obtaining a Q-learning based portfolio scheduler

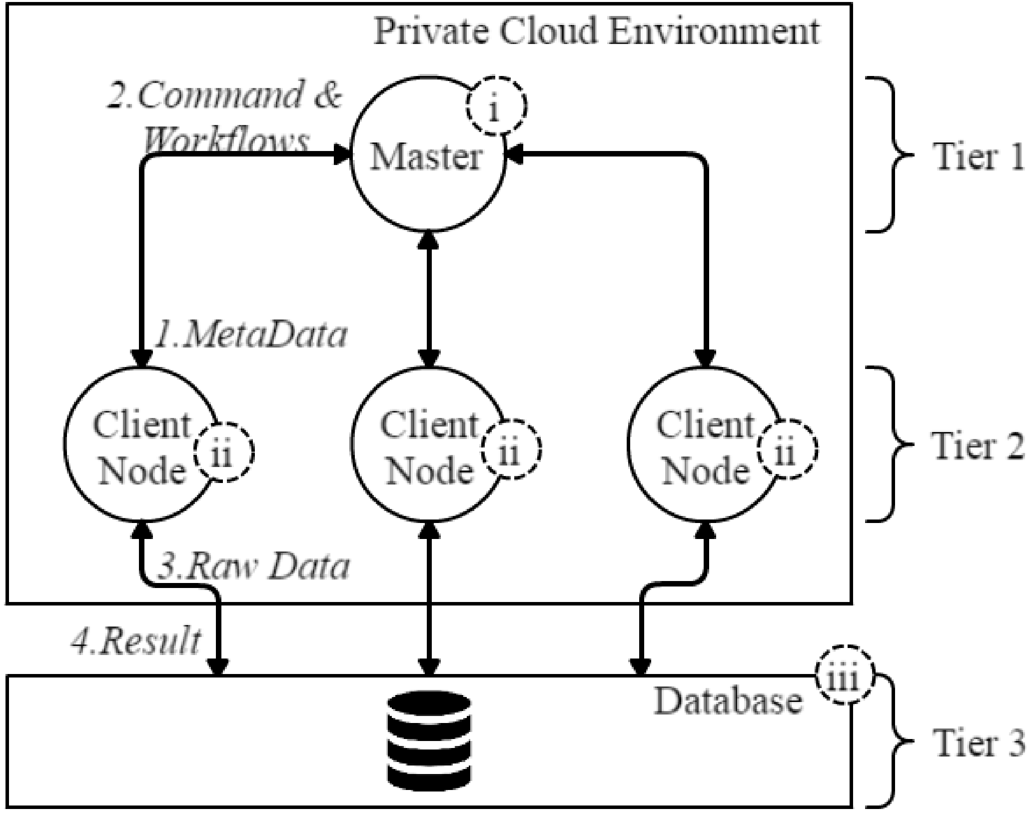


The Smart Connect Framework at Shell



Workflow and Resource Management

The Smart Connect Framework at Shell



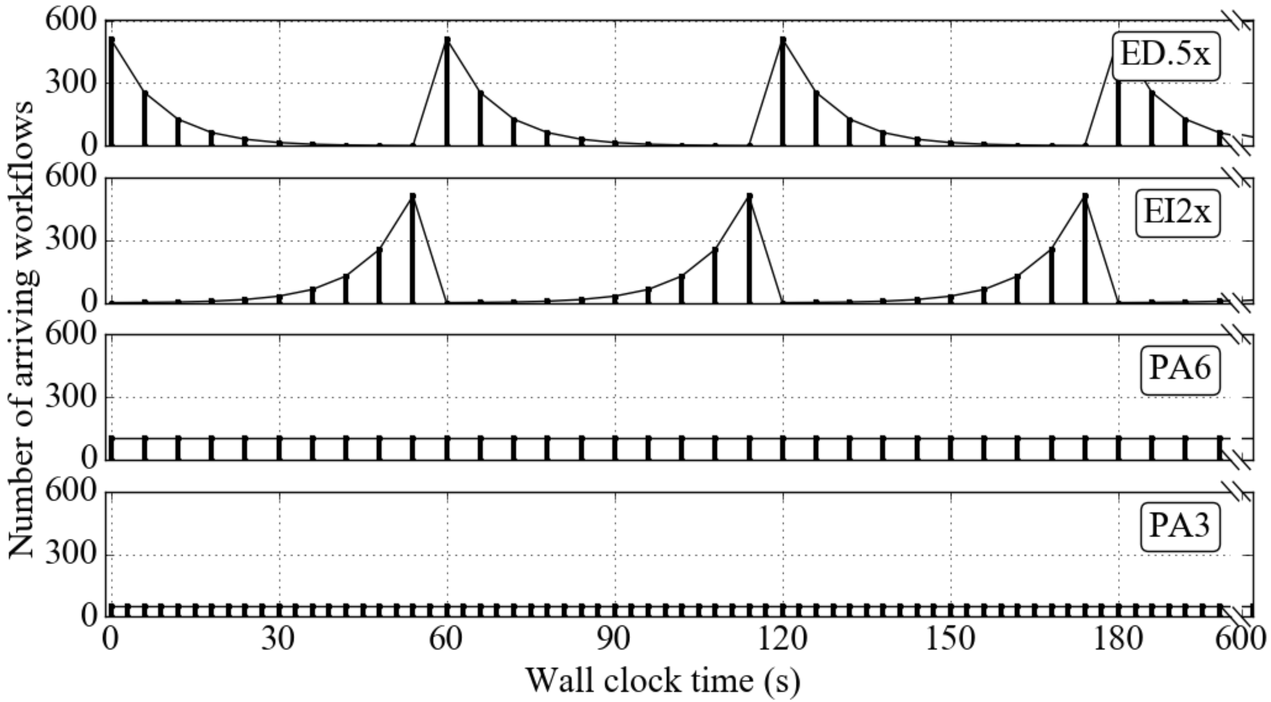
Workflow and Resource Management

TUDelft

Evaluation

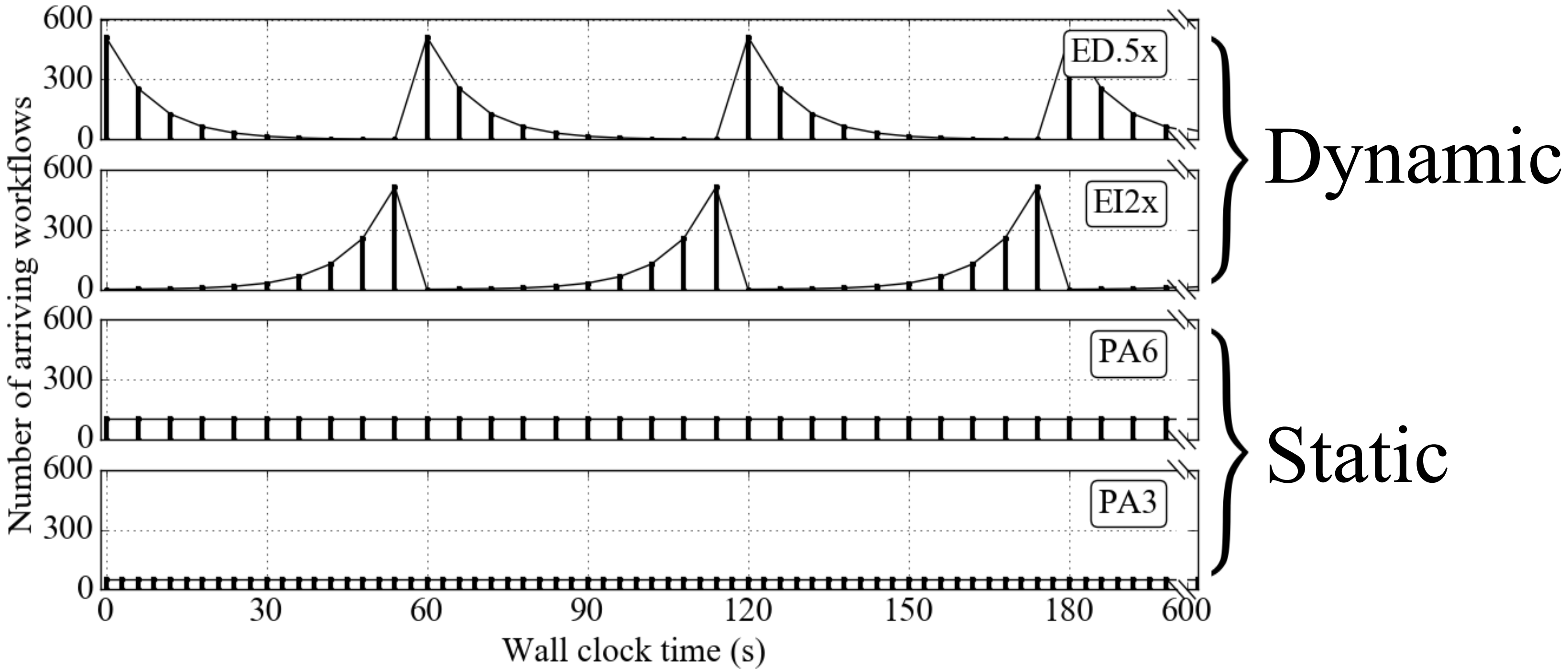
- Real-world experiments based on a prototype implementation
- Realistic experimental environment: DAS-5
  - DAS is often used to emulate cloud environments
  - 1 Master node; 3-50 Client nodes
  - CPU: Intel E5-2630v3 2.4GHz
  - RAM: 64 GB
  - Network: 1 GB/s Ethernet links

Synthetic Experimental workload, with realistic parameters derived from real-world deployment

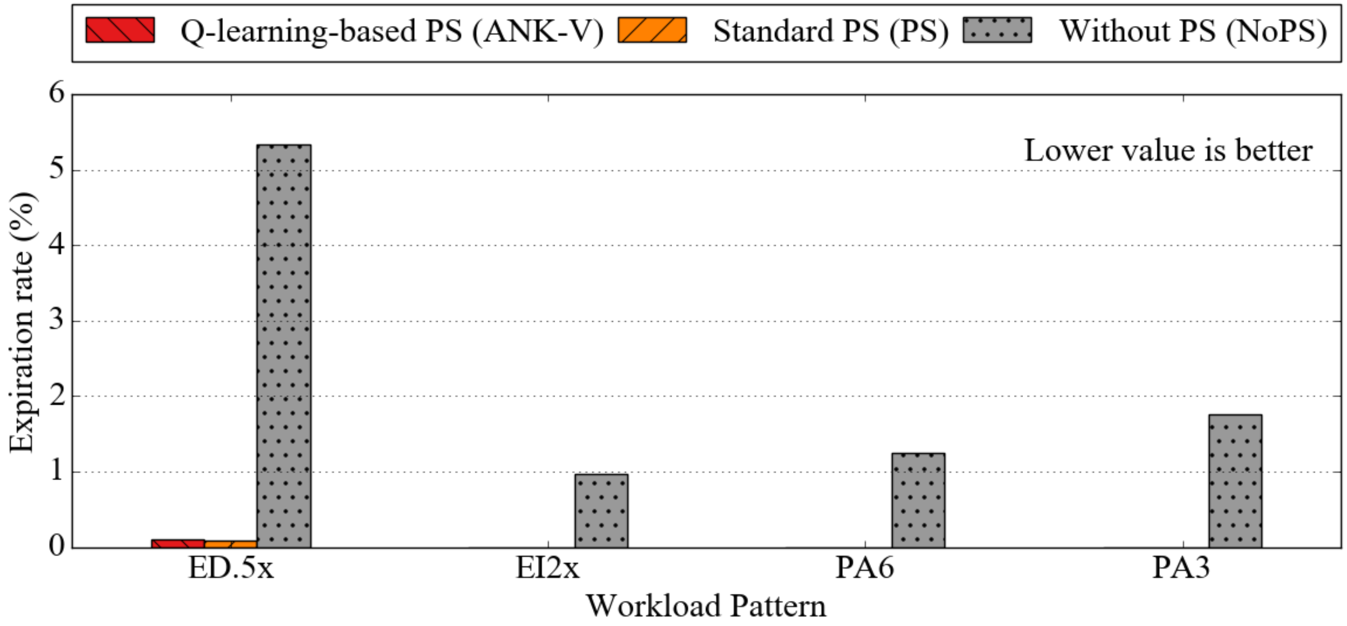


TUDelft

Synthetic Experimental workload, with realistic parameters derived from real-world deployment

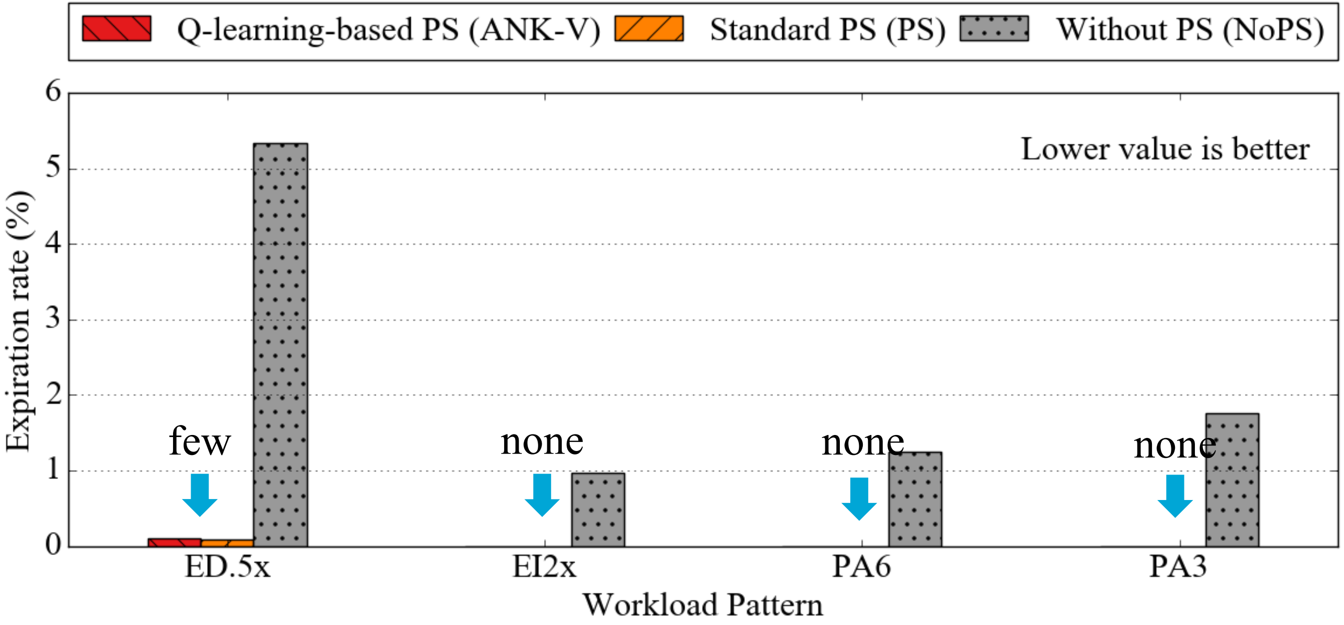


Key Deadline Metrics: Expired Workflows (#Clients Nodes: 3)



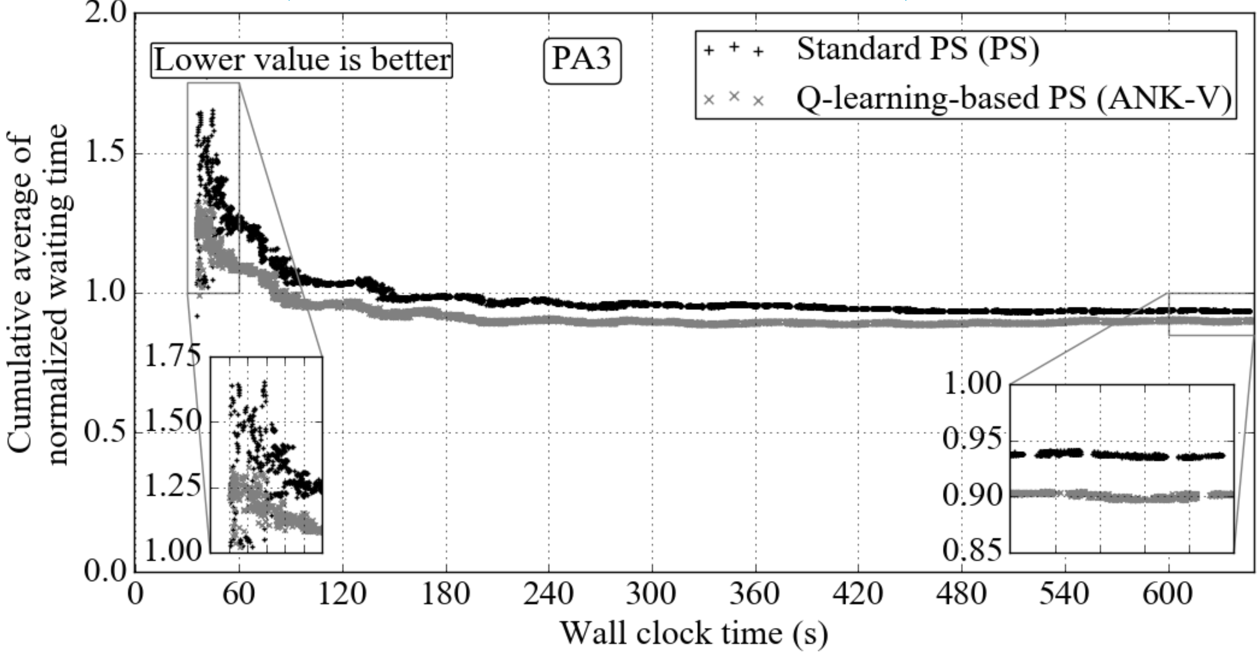
TU Delft

Key Deadline Metrics: Expired Workflows (#Clients Nodes: 3)



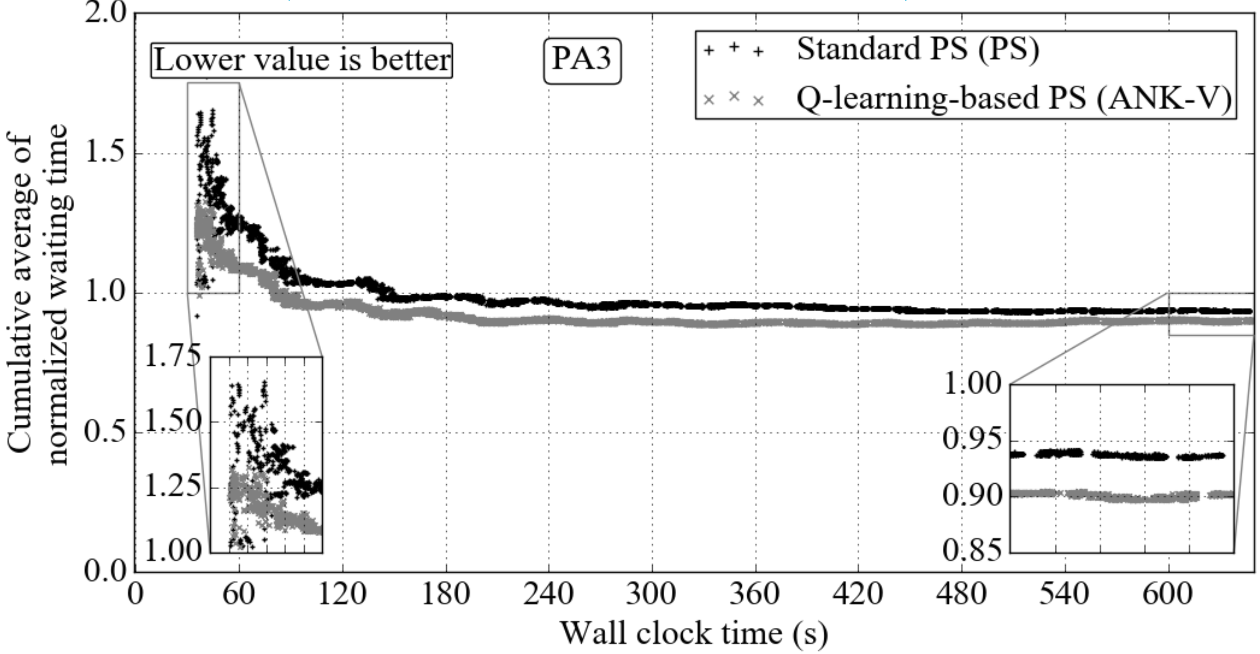
Both meet the deadline-constraints

Key User Metrics: Workflow performance across portfolio schedulers (#Clients Nodes: 3)



TU Delft

Key User Metrics: Workflow performance across portfolio schedulers (#Clients Nodes: 3)



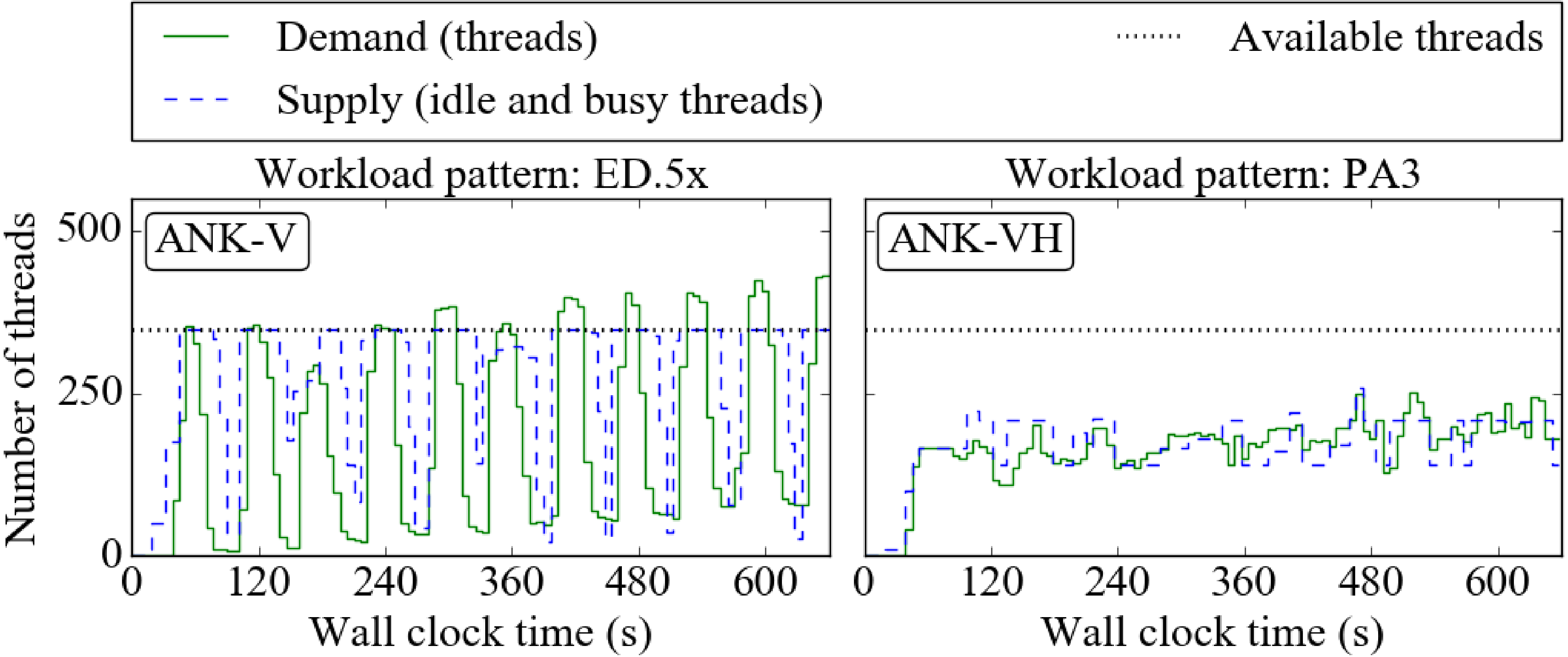
ANANKE has 5-20% better performance with static workloads, 0 – 8.3% for dynamic.

TUDelft

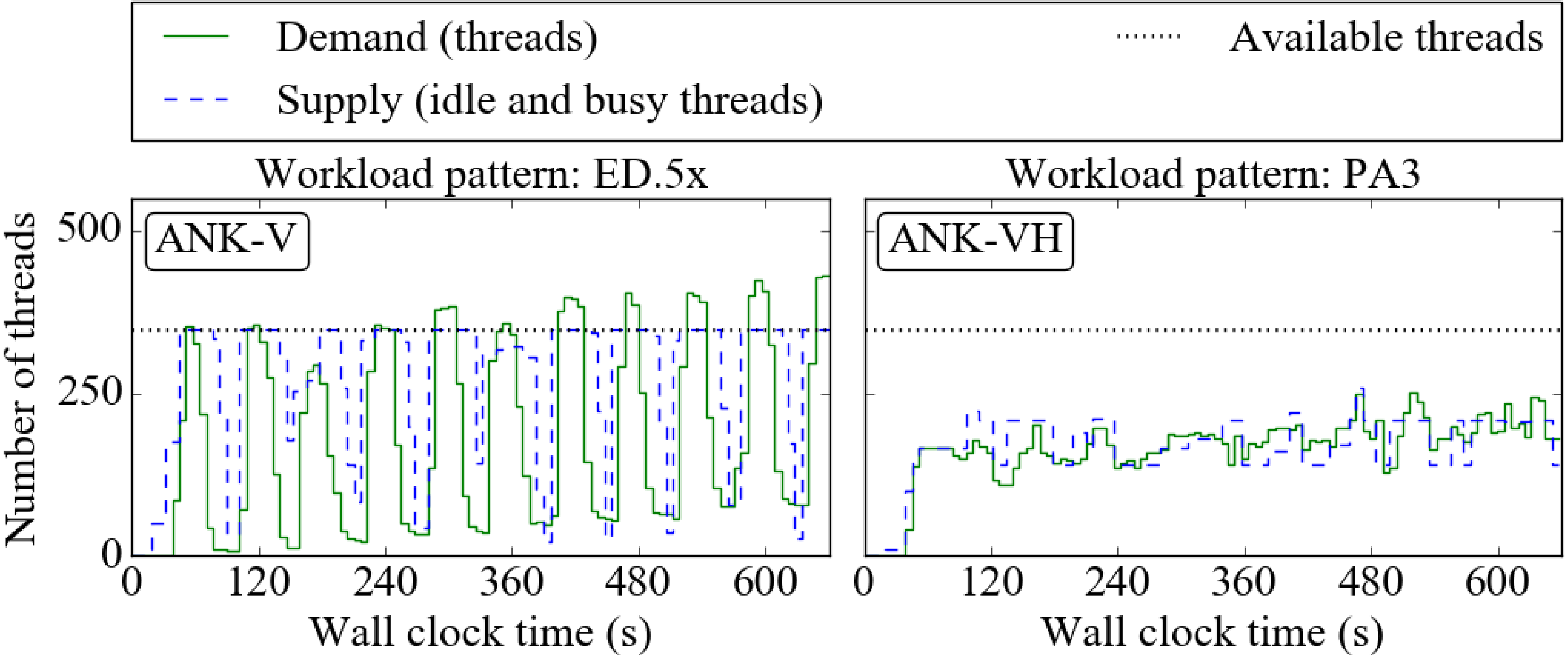
Elasticity: Supply and Demand

- 5 Auto Scalers
  - 2 ANANKE implementations
  - 3 baselines
- Supply: Number of active threads
- Demand: Number of running and near deadline workflows

Supply and Demand Analysis

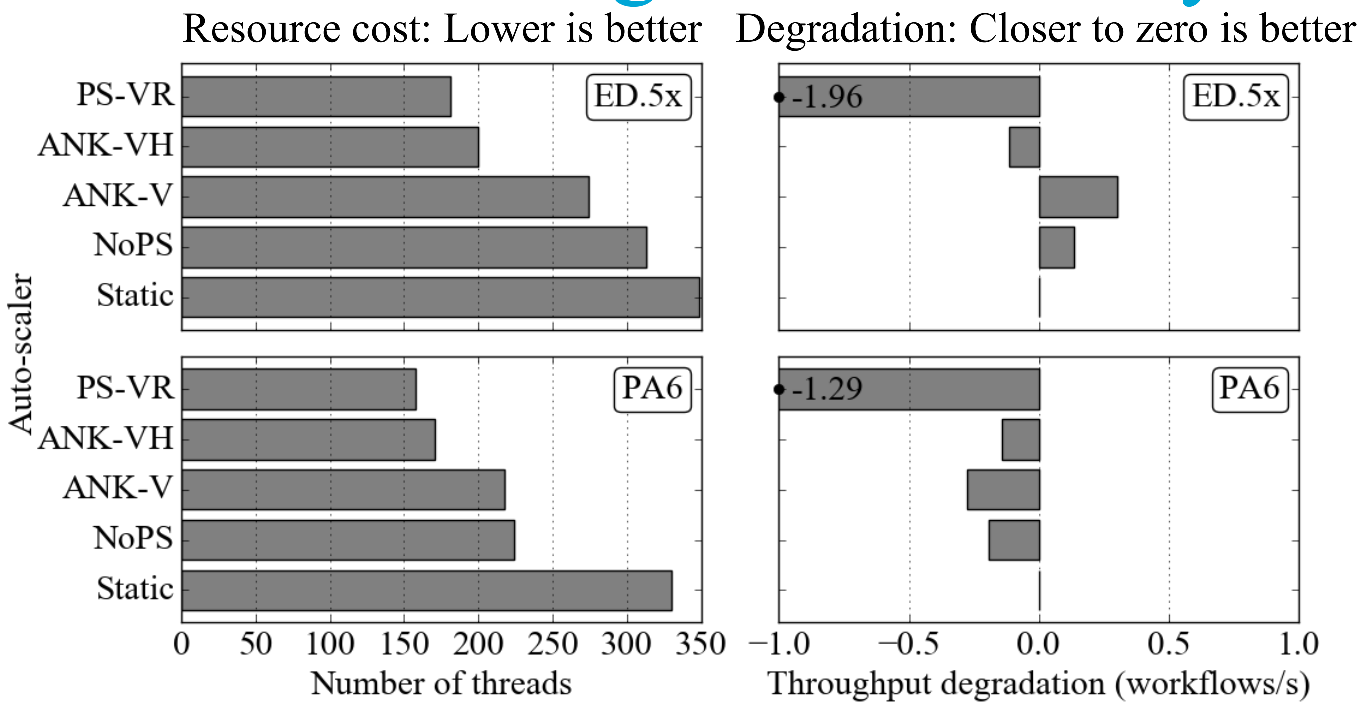


Supply and Demand Analysis

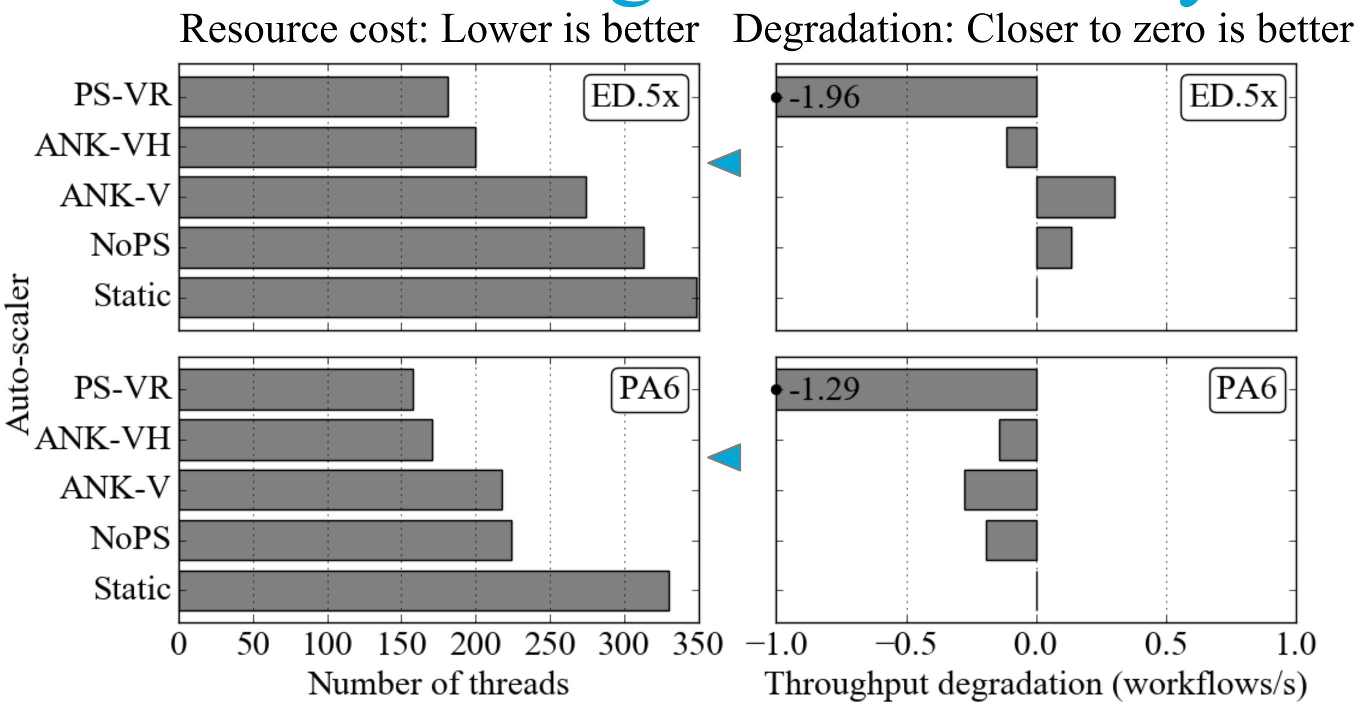


ANANKE has the best elasticity among the candidates

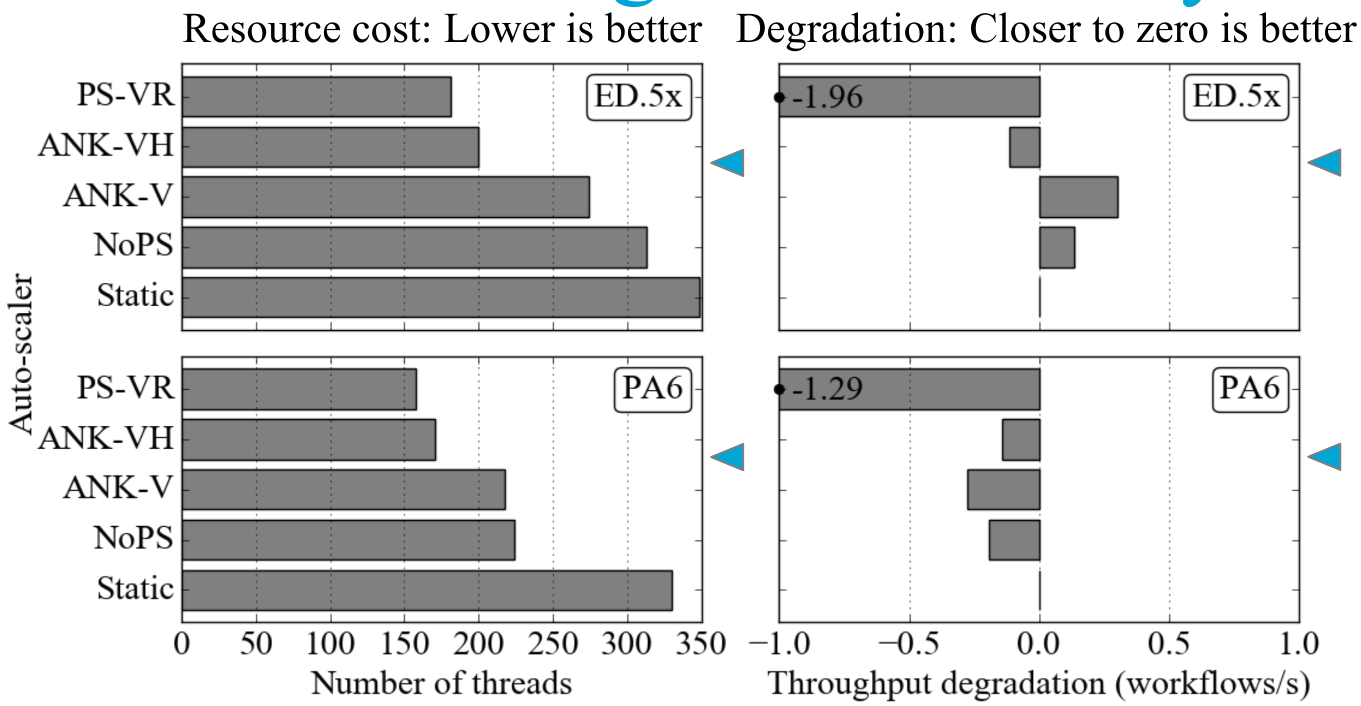
Performance Degradation Analysis



Performance Degradation Analysis

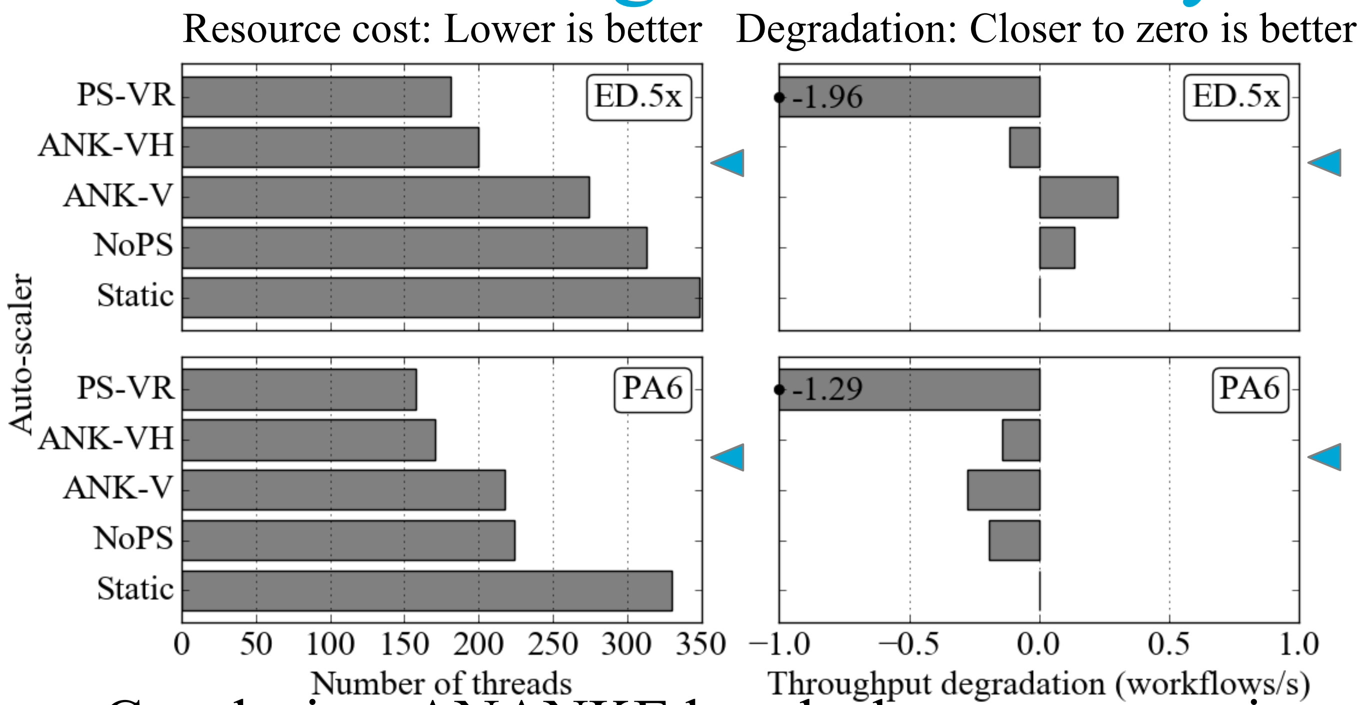


Performance Degradation Analysis



TU Delft

Performance Degradation Analysis



Conclusion: ANANKE has the best user-experience performance with the lowest resource cost. 24 – 36% resource savings at the cost of at most 1.4% throughput.

TU Delft

Conclusion

- Design and implement ANANKE
  - a Q-learning based portfolio scheduler
  - Complex industrial workflows
- Evaluate through real-world experiments
- Better user-experience performance, resource utilization and elasticity for relatively static workloads
- For highly dynamic workloads, using Q-learning is less beneficial, but still positive

TUDelft

Future Work

- Different type of simulators
- Different learning techniques
  - GeneRec (error-driven reinforcement learning)
- Advanced mechanism to control/restrict the simulation time
- Apply ANANKE in hybrid cloud environments

ANANKE: a Q-Learning based Portfolio Scheduler for Complex Industrial Workflows

Presentation: Laurens Versluis

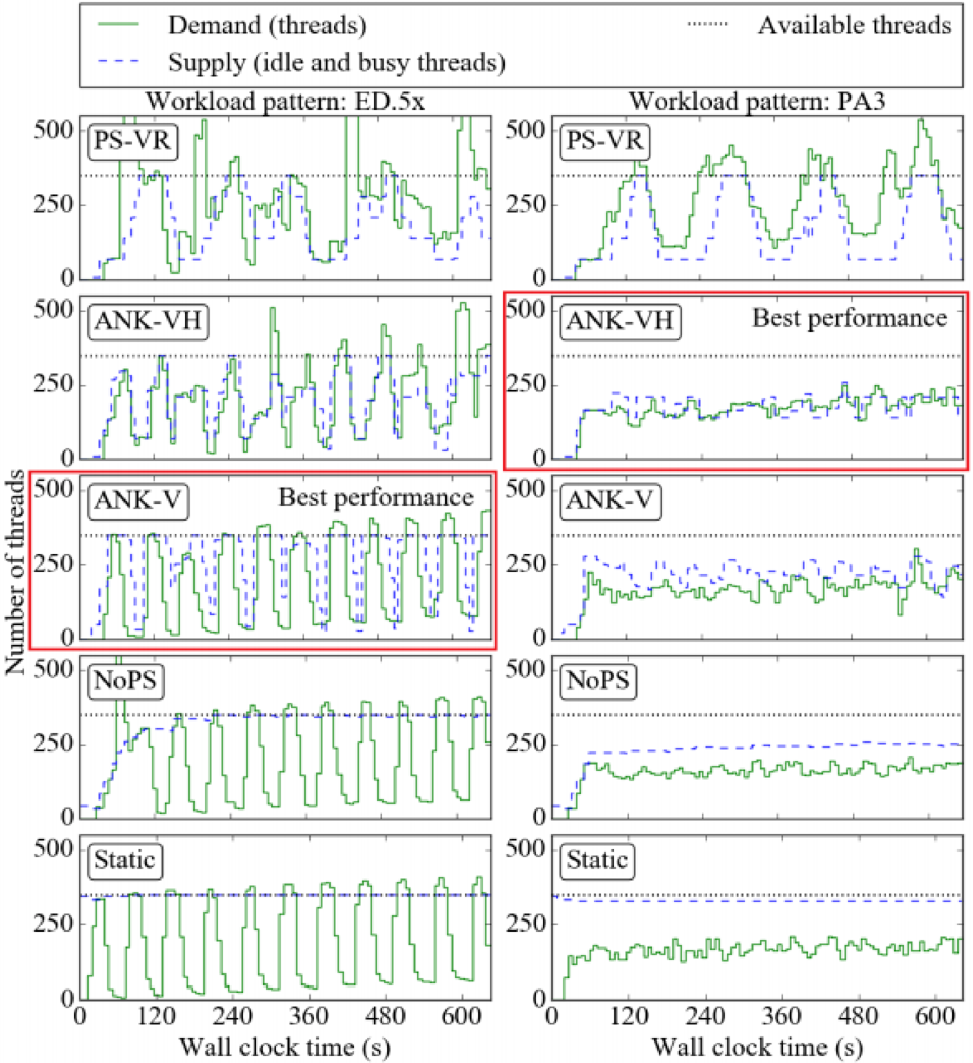
Slides: Shenjun Ma

Shenjun Ma
Alexey Ilyushkin
Alexander Stegehuis
Alexandru Iosup
@Large research: https://atlarge-research.com/

Novelty

- Conceptual contriution: combining Q-learning with portfolio scheduling
- Conceptual contribution: A QL-based scheduling policy

TUDelft



Demand (threads)
Supply (idle and busy threads)
Available threads
Workload pattern: ED.5x
Workload pattern: PA3
PS-VR
ANK-VH
ANK-V
NoPS
Static
Best performance
Best performance
Number of threads
500
250
0
0
120
240
360
480
600
Wall clock time (s)
Wall clock time (s)

Comprehensive Comparison between auto-scalers

TU Delft

Comprehensive Comparison between auto-scalers

- Numerical methods
  - Pairwise Comparison
  - Fractional Difference Comparison

Comprehensive Comparison between auto-scalers

- Numerical methods
  - Pairwise Comparison
  - Fractional Difference Comparison
- 9 related metrics
  - system- and user-oriented metrics

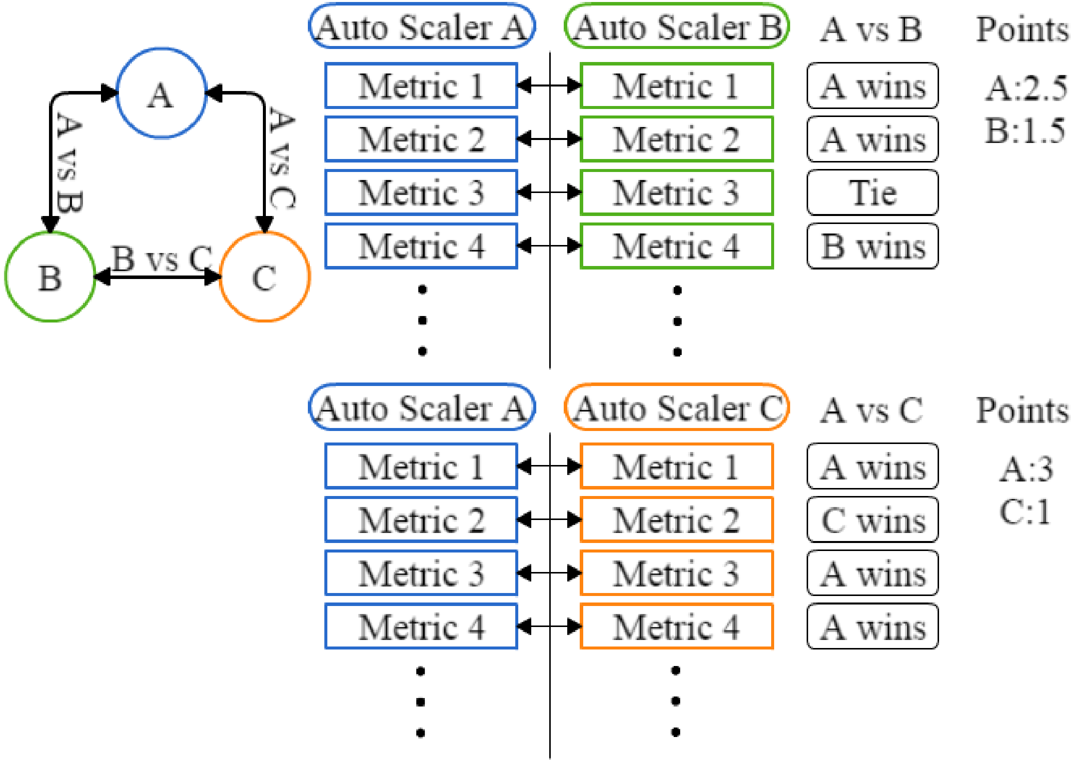
Comprehensive Comparison between auto-scalers

- Numerical methods
  - Pairwise Comparison
  - Fractional Difference Comparison
- 9 related metrics
  - system- and user-oriented metrics
- Full results are in the technical report

TUDelft

Comprehensive Comparison between auto-scalers

- Numerical methods
  - Pairwise Comparison
  - Fractional Difference Comparison
- 9 related metrics
  - system- and user-oriented metrics
- Full results are in the technical report

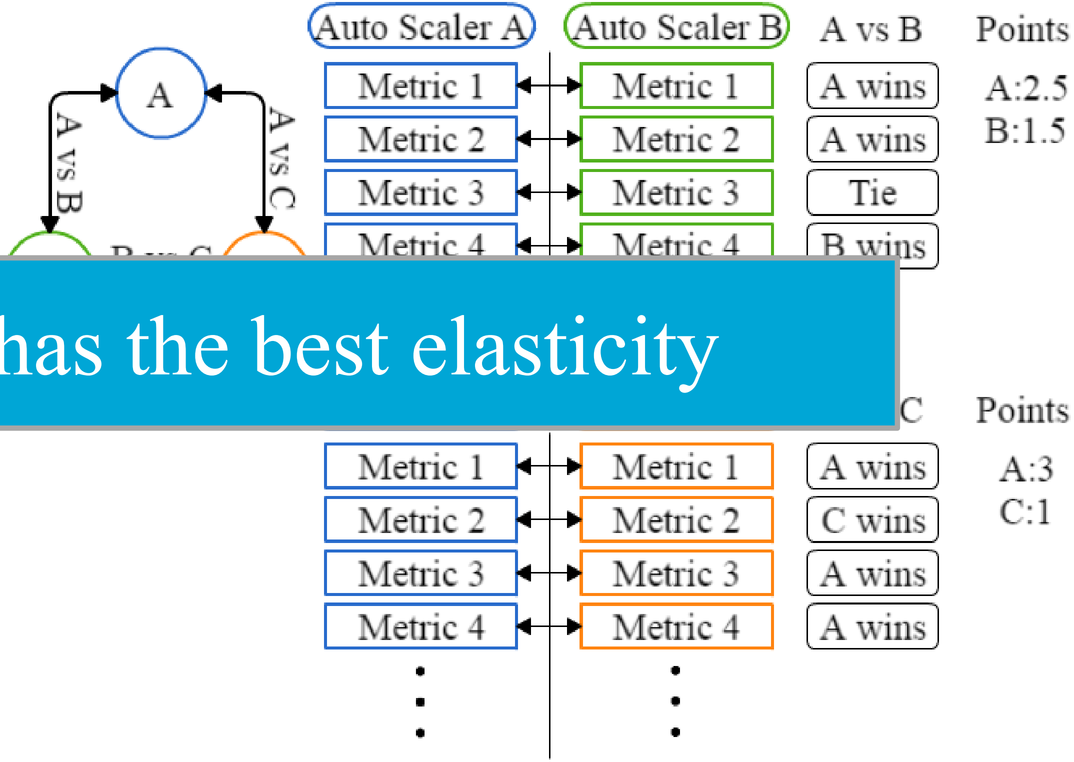


TUDelft

Comprehensive Comparison between auto-scalers

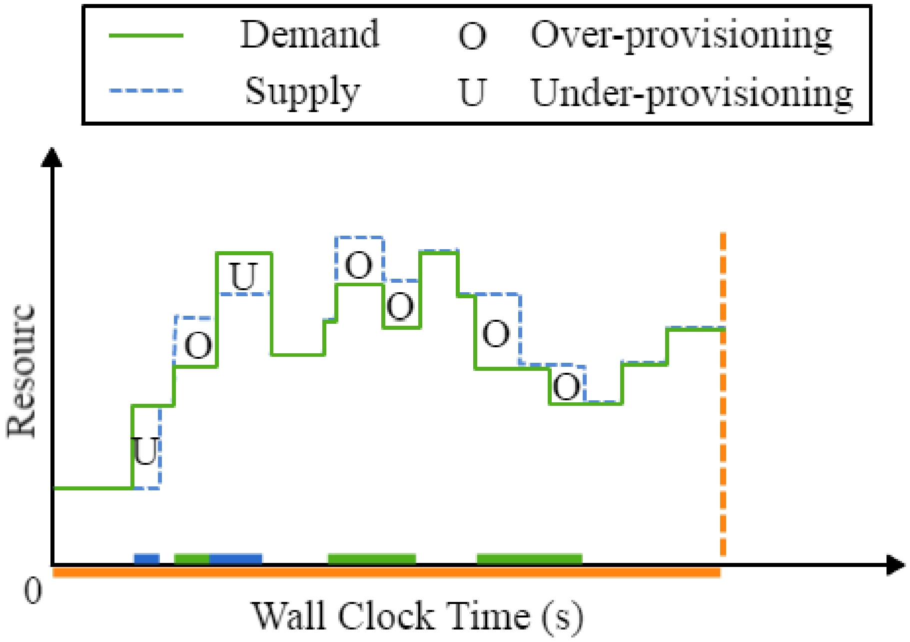
- Numerical methods
  - Pairwise Comparison
  - Fractional Difference
- 9 re
  - system- and user-oriented metrics
- Full results are in the technical report

ANANKE has the best elasticity



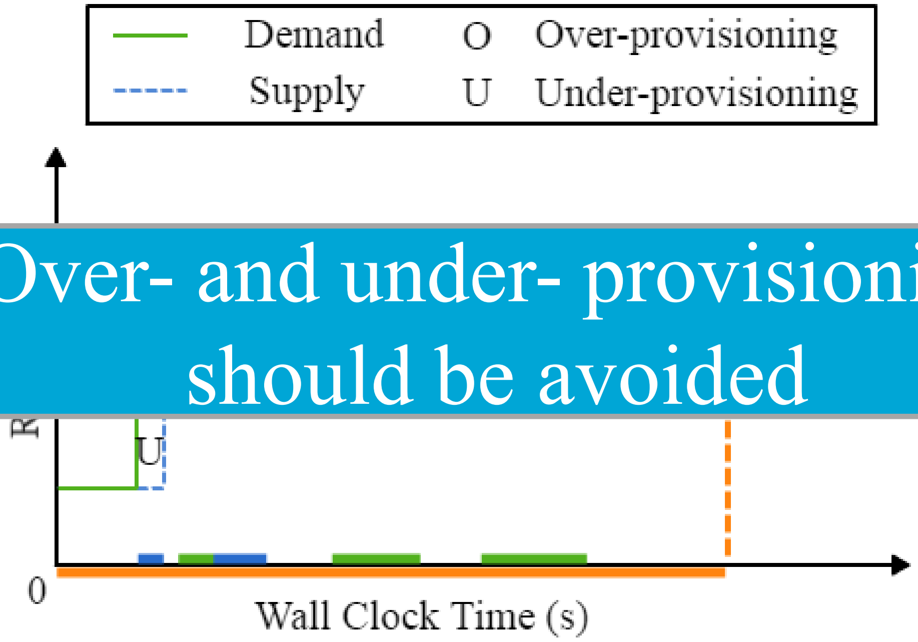
TUDelft

Key concept for auto scaling: Demand and Supply



Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A. V., Bogdan, G., Epema, D., & Iosup, A. (2016). An Experimental Performance Evaluation of Autoscaling Algorithms for Complex Workflows. In ACM Symposium on Cloud Computing 2016 (SOCC 2016).

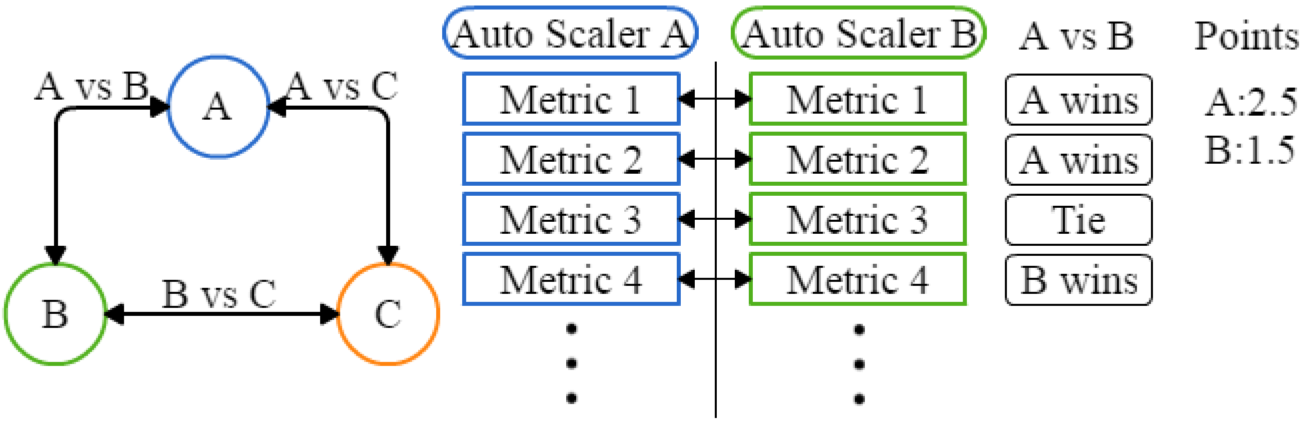
Key concept for auto scaling: Demand and Supply



Over- and under- provisioning should be avoided

Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A. V., Bogdan, G., Epema, D., & Iosup, A. (2016). An Experimental Performance Evaluation of Autoscaling Algorithms for Complex Workflows. In ACM Symposium on Cloud Computing 2016 (SOCC 2016).

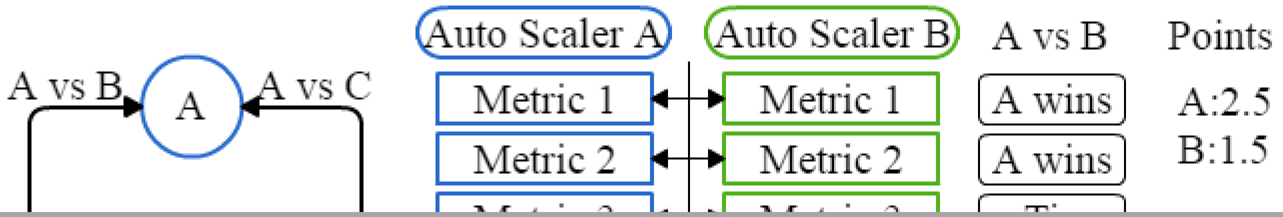
Pairwise Comparison



- Higher value is better

Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A. V., Bogdan, G., Epema, D., & Iosup, A. (2016). An Experimental Performance Evaluation of Autoscaling Algorithms for Complex Workflows. In ACM Symposium on Cloud Computing 2016 (SOCC 2016).

Pairwise Comparison

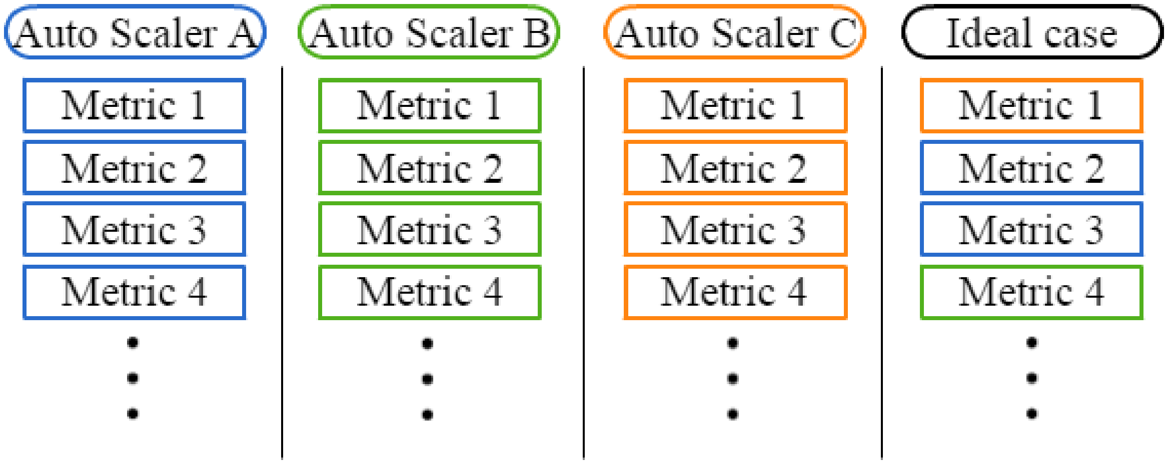


ANANKE with vertical scaling outperforms the candidates in static and dynamic workloads

- Higher value is better

Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A. V., Bogdan, G., Epema, D., & Iosup, A. (2016). An Experimental Performance Evaluation of Autoscaling Algorithms for Complex Workflows. In ACM Symposium on Cloud Computing 2016 (SOCC 2016).

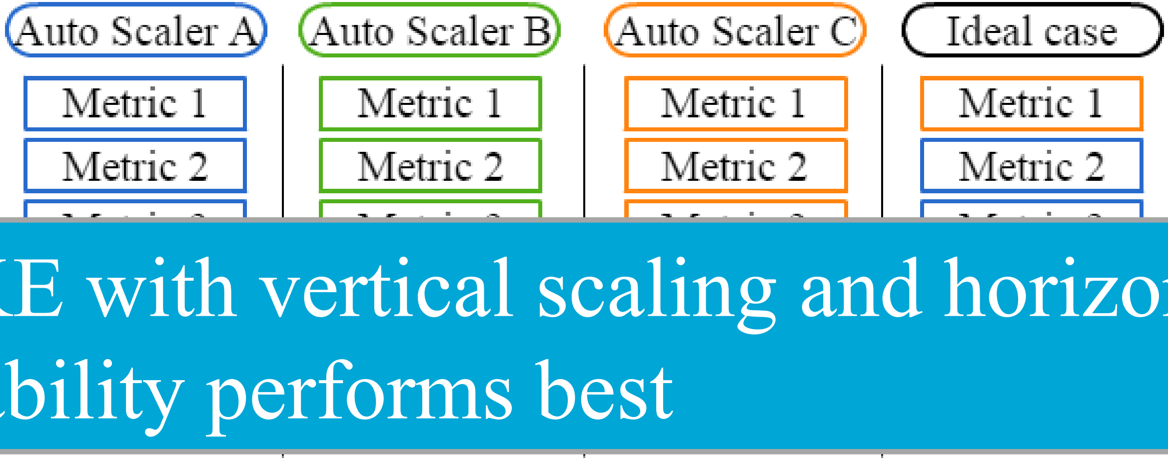
Fractional Difference Comparison



- Lower value is better

TU Delft

Fractional Difference Comparison



ANANKE with vertical scaling and horizontal scaling ability performs best

- Lower value is better

Appendix: Comprehensive Comparison Results

Table 6.6: The results of the pairwise and fractional comparisons. The winners are highlighted in bold. AS stands for auto-scaler.

Auto Scaler	Pairwise (points)				Fractional (frac.)			
	ED.5x	EI2x	PA6	PA3	ED.5x	EI2x	PA6	PA3
PS-(VR)	11	13	13	13	3.00	2.88	3.65	3.58
ANK-VH	17	16	**19**	15	**2.89**	**2.70**	**1.80**	**1.94**
ANK-V	**20**	**22**	15	**19**	6.78	5.53	4.00	3.95
NoPS	19	16	17.5	17	6.02	5.82	4.08	3.92
Static	13	13	15.5	16	13.44	15.91	14.51	14.46

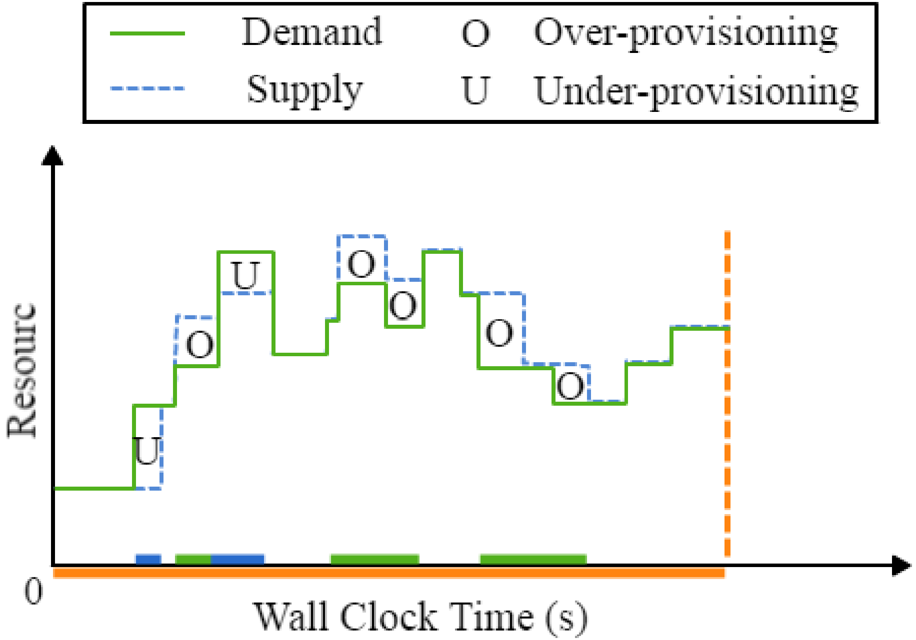
TU Delft

Appendix: Research Questions

1. How to adapt portfolio scheduling?
2. How to use learning technique and historical information?
3. How to evaluate the learning-based portfolio scheduler?

TUDelft

Appendix: Auto scaling metrics (Accuracy)



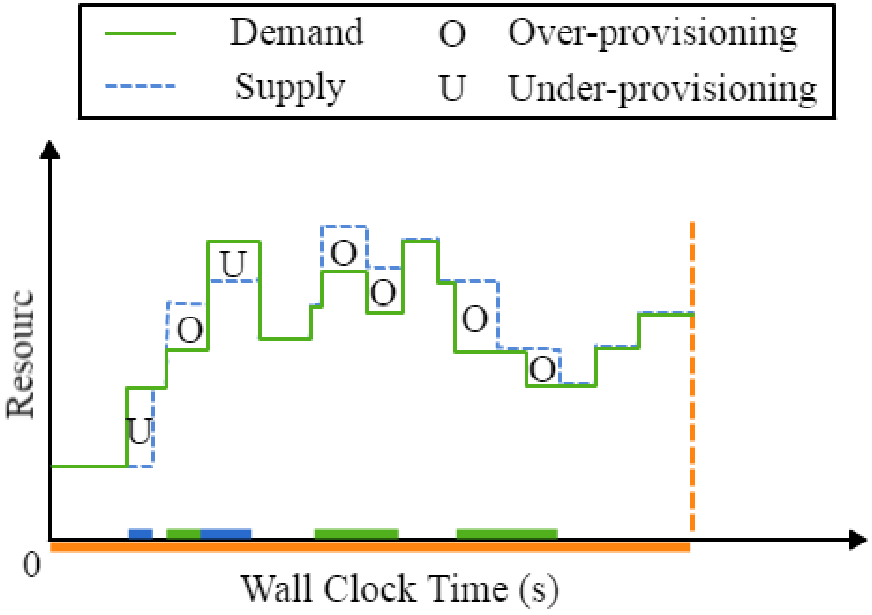
Under-provisioning Accuracy

$$= \frac{resource\ under\_provisioning}{Total\ resource}$$

Over-provisioning Accuracy

$$= \frac{resource\ over\_provisioning}{Total\ resource}$$

Appendix: Auto scaling metrics (Timeshare)



Under-provisioning Timeshare

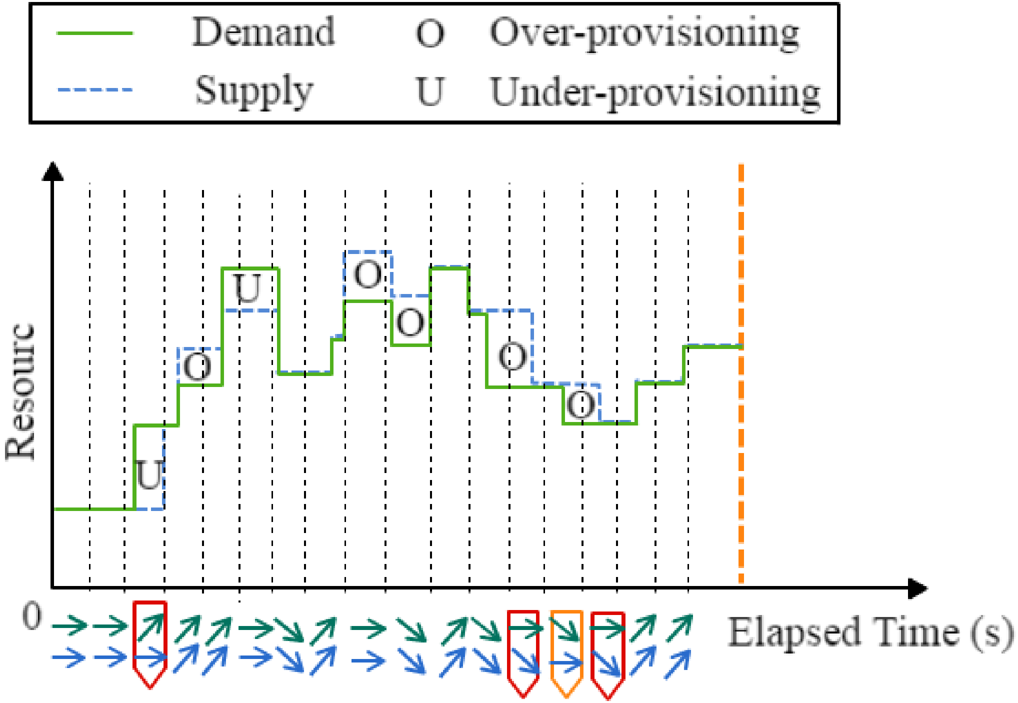
$$= \frac{Time\ of\ under\_provisioning\ state}{Total\ time}$$

Over-provisioning Timeshare

$$= \frac{Time\ of\ over\_provisioning\ state}{Total\ time}$$

TUDelft

Appendix: Auto scaling metrics (instability)



The fraction of time the supply and demand curves move in opposite directions.

The fraction of time the supply and demand curves move towards each others.

TUDelft

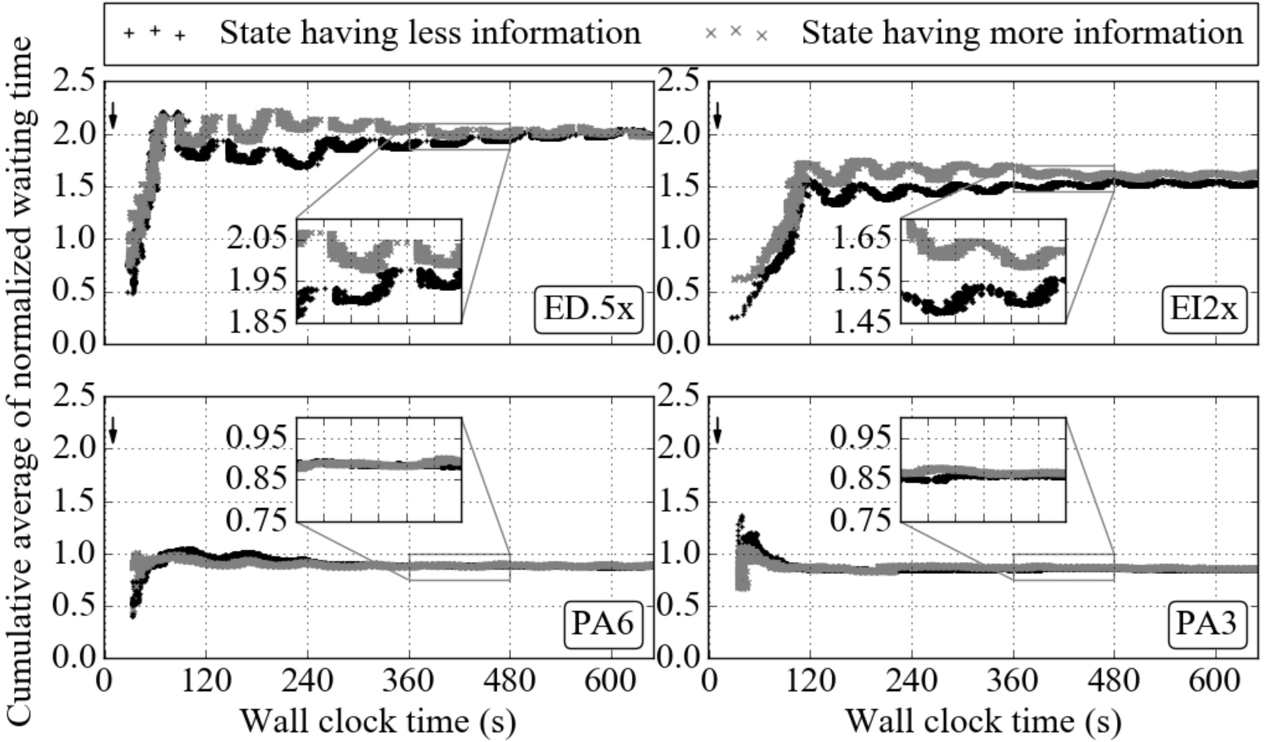
Appendix: Decision Table in Q-Learning based Policy

State \ Action	(0, none)	(1, up)	(3, down)	...	(m, down)
(0, 0, 0)	X	X	X	...	X
(0.1, 0.1, 0.1)	X	X	X	...	X
...	...	...	...	...	...
(u_t, v_t, y_t)	X	X	X	...	X

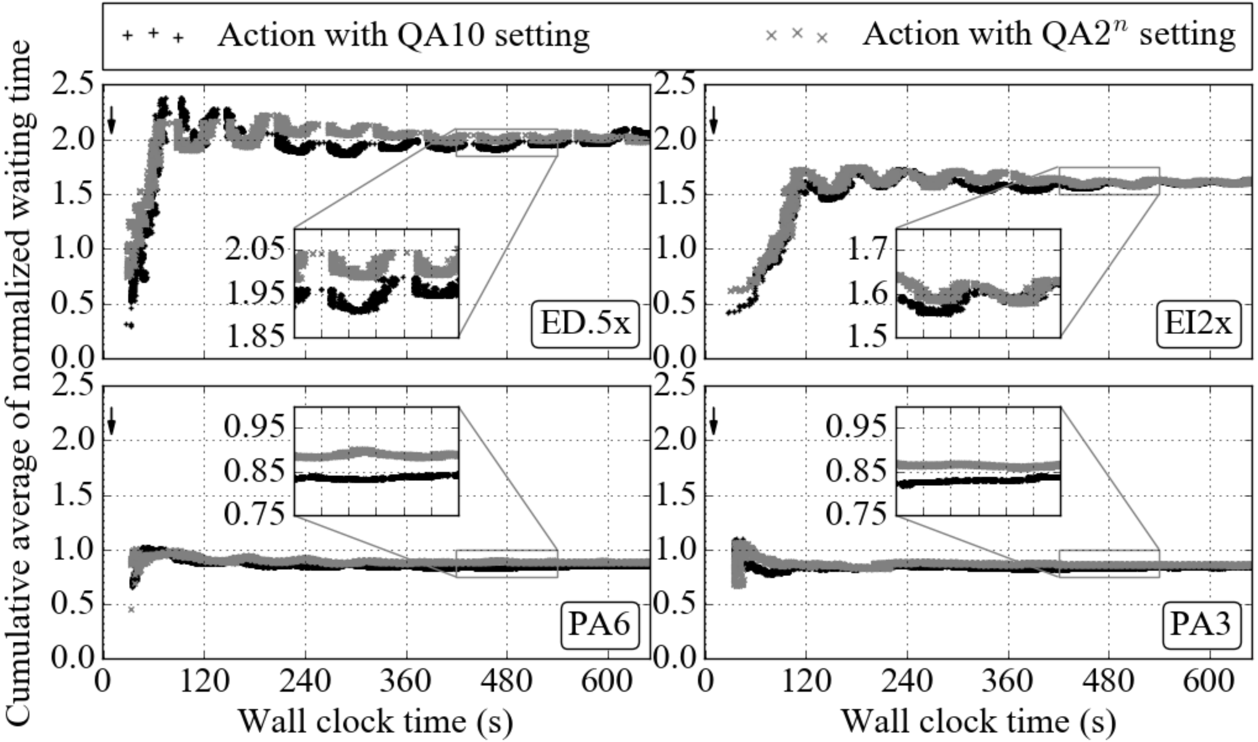
- Different format of State or Action will leads to different Table size.
  - $Table\ Size = size\ of\ action\ set\ \times\ size\ of\ state\ set$

TUDelft

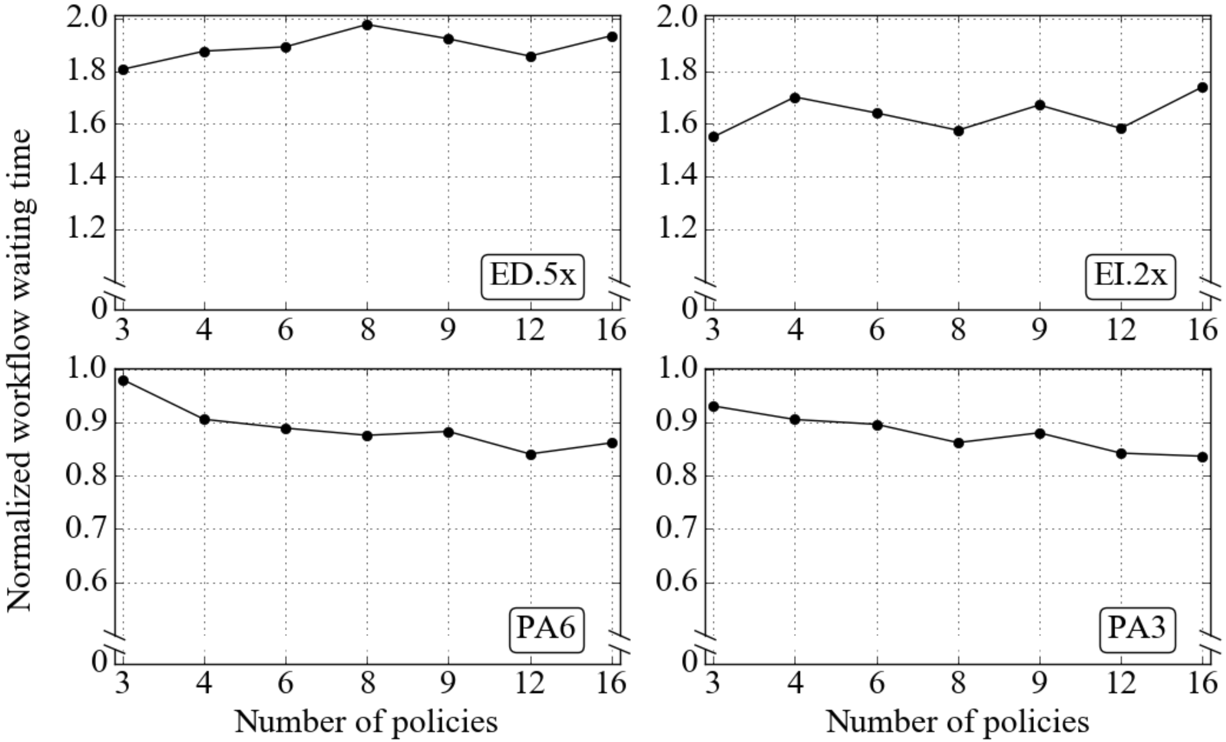
Appendix: Different Configuration Setting in Determining State



Appendix: Different Configuration Setting in Determining Action



Appendix: Policy Pool Size Impact on Workflow Performance



TU Delft