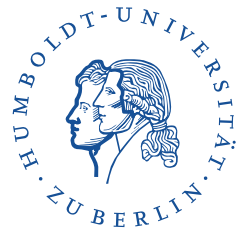
HUMBOLDT-UNIVERSITÄT ZU BERLIN



Starting Workflow Tasks Before They're Ready

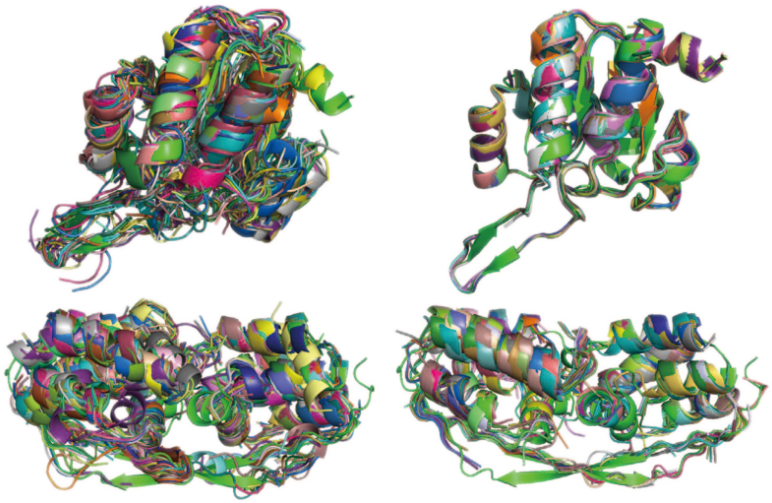
Wladislaw Gusew, Björn Scheuermann

Computer Engineering Group, Humboldt University of Berlin

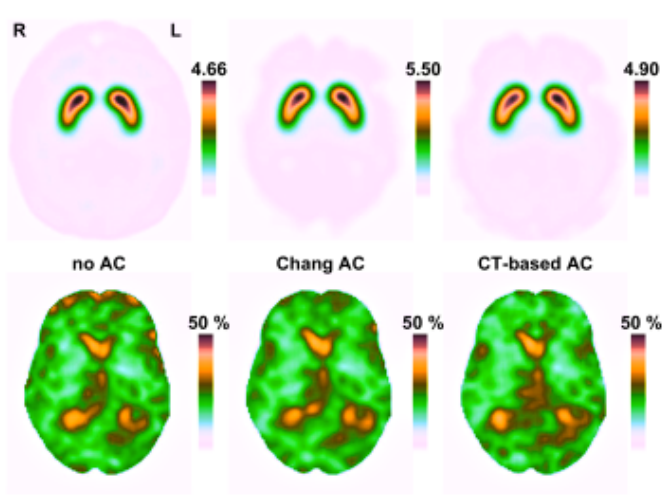
Agenda

- Introduction
- Execution semantics
- Methods and tools
- Simulation results
- Experimental results
- Conclusion

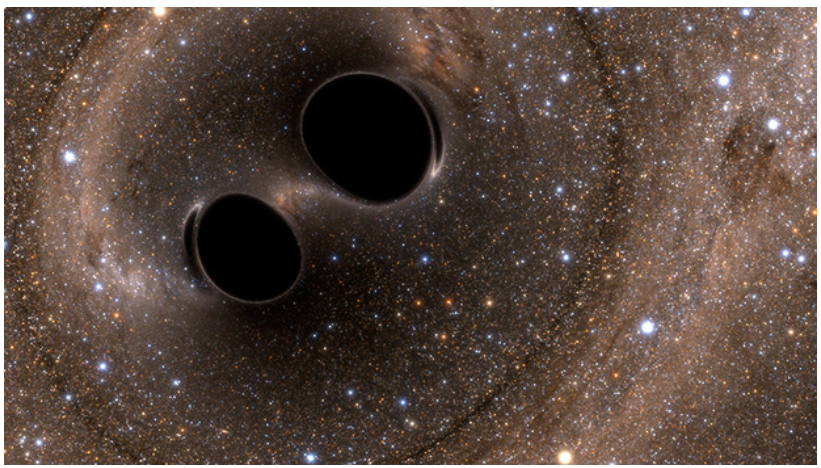
Big data in research



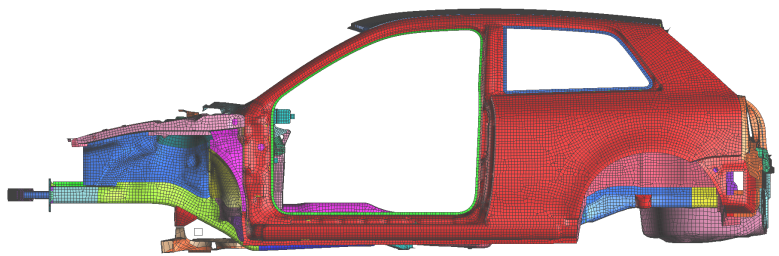
Life sciences



Medicine

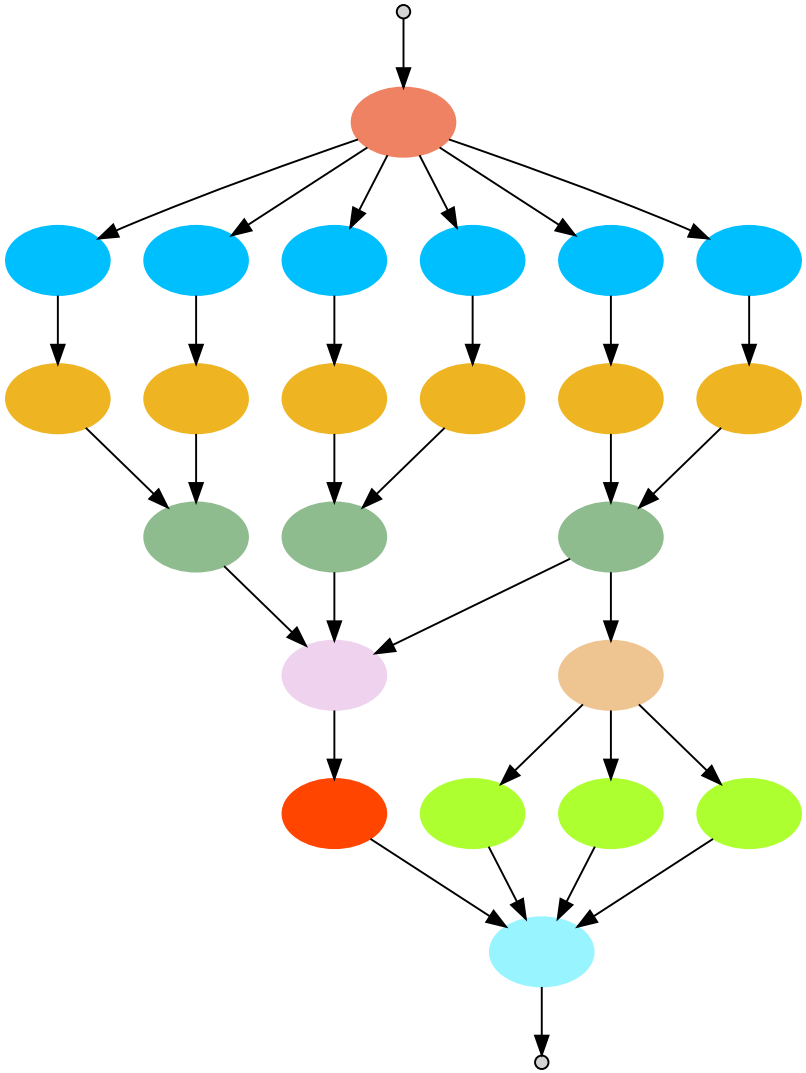


Natural sciences



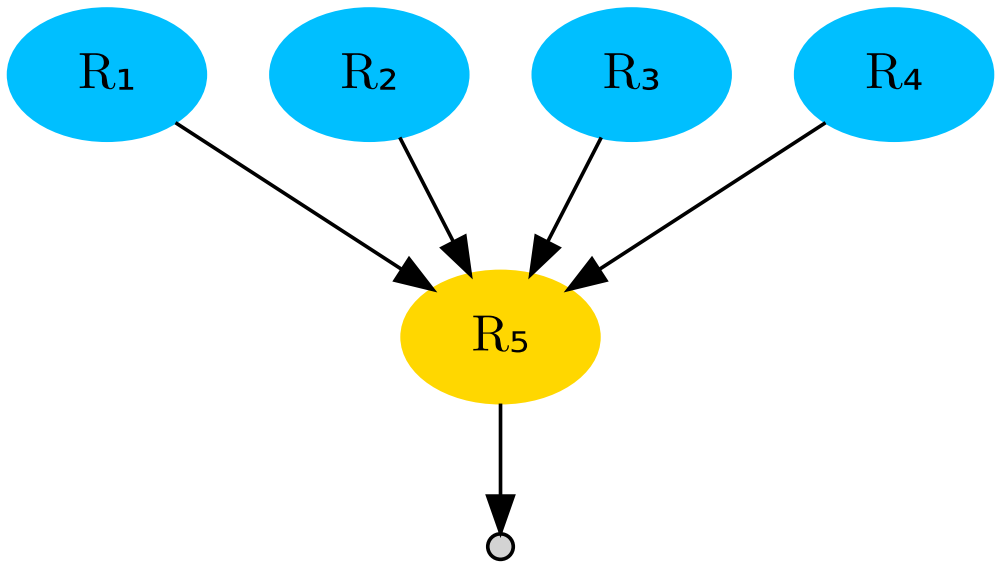
Engineering

Scientific workflow example

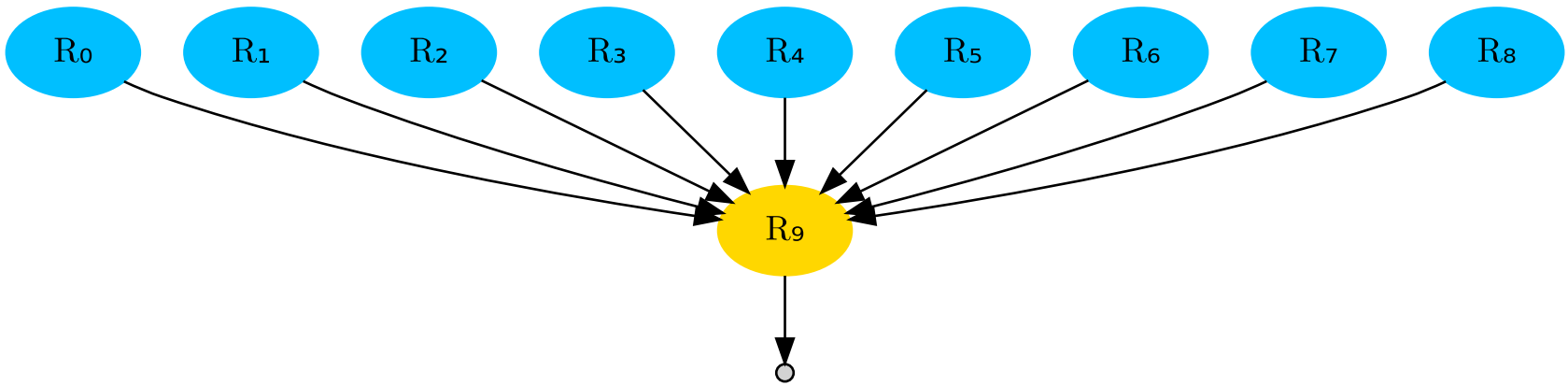


- Directed Acyclic Graph (DAG)
- Executed on distributed systems
- Aggregation and broadcast types of tasks
- Demanding for network resources

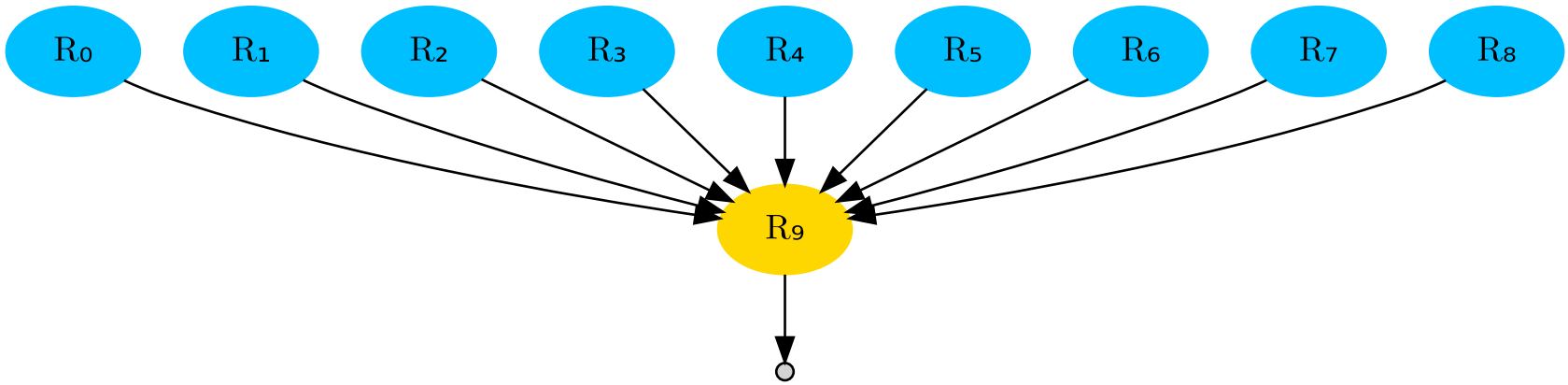
Execution semantics



Execution semantics

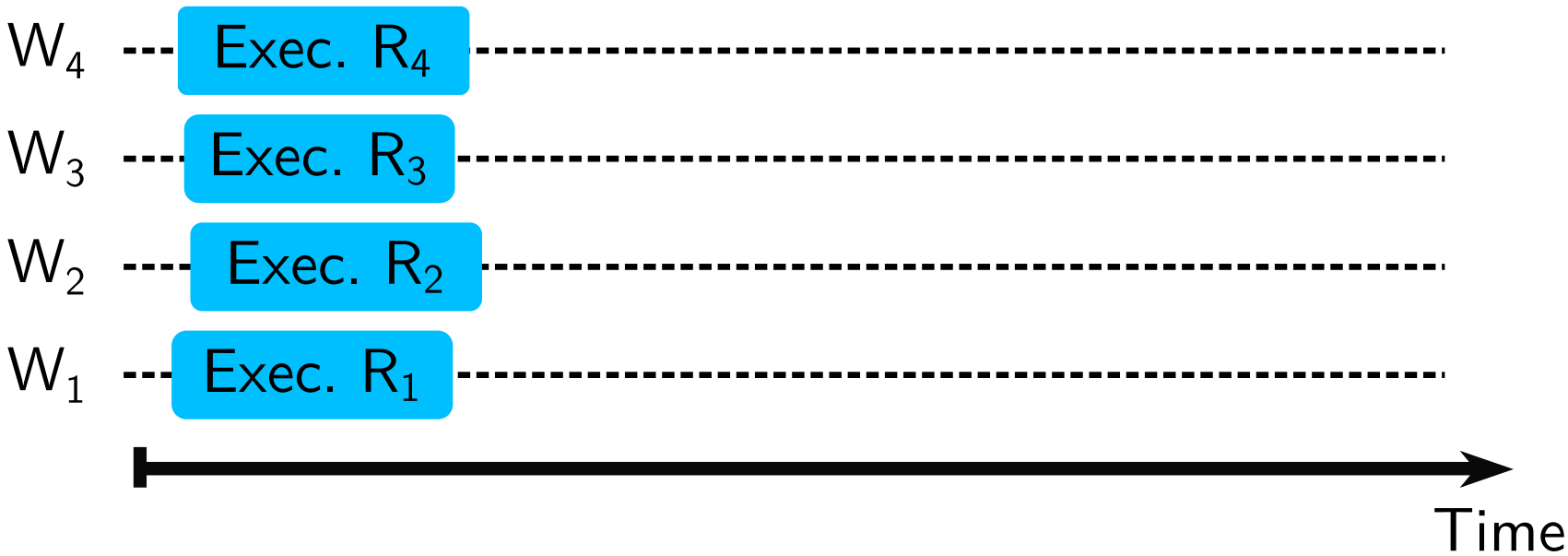


Execution semantics

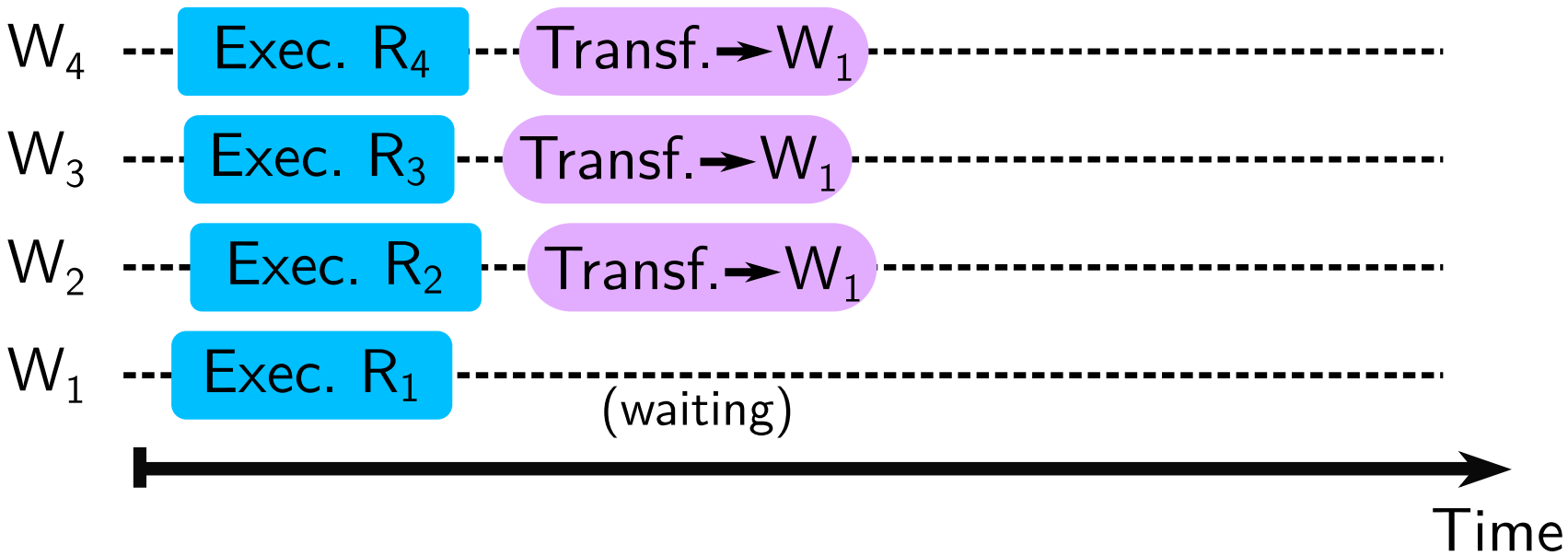


- But in reality resources are limited
- Execute only a subset of parent tasks concurrently (insufficient number of workers)
- Congestion of network (all parent tasks have the same priority)

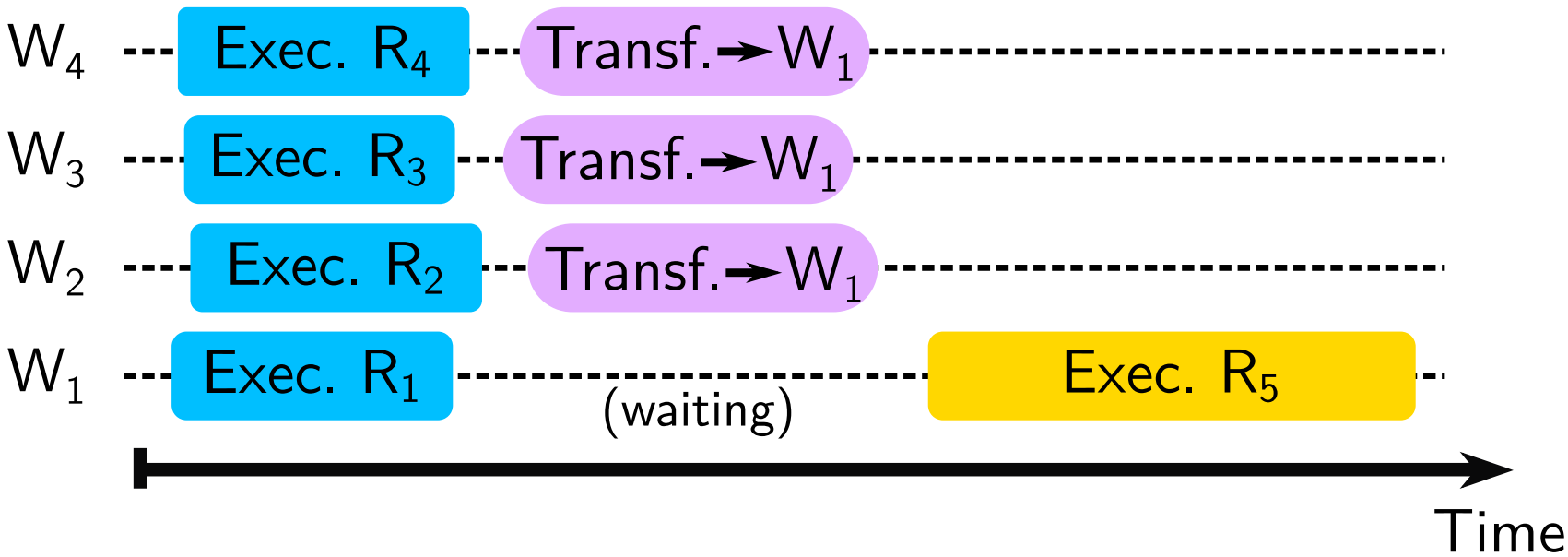
Example execution



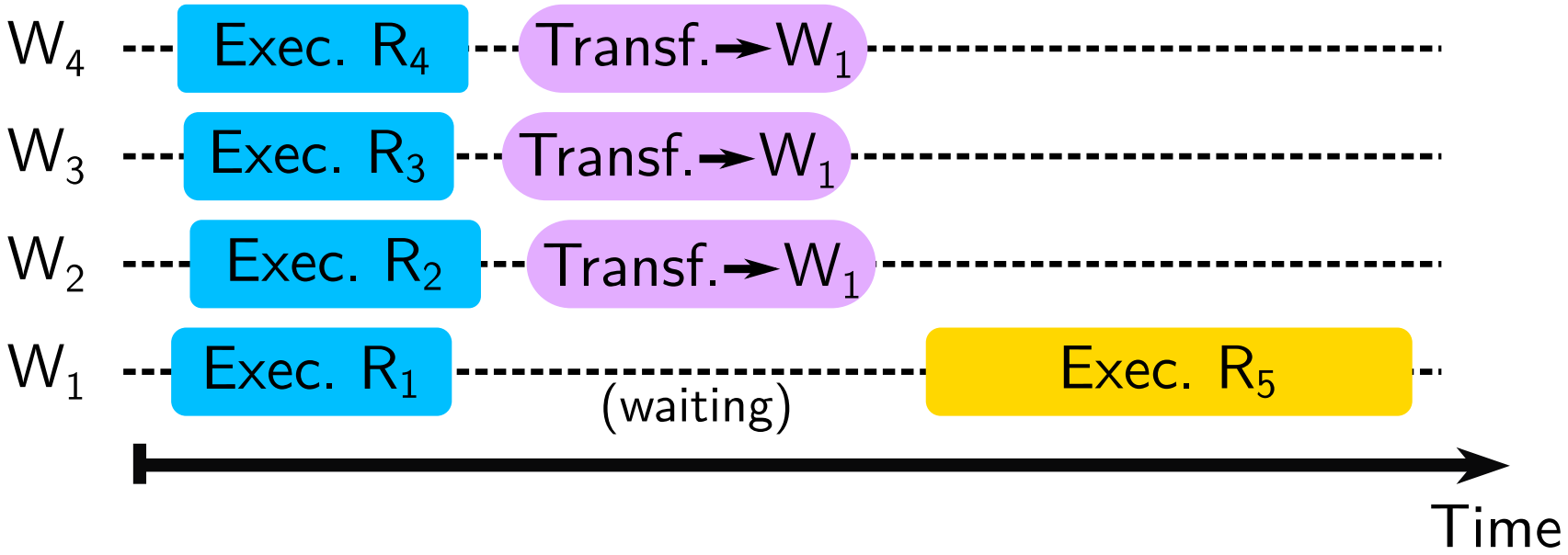
Example execution



Example execution



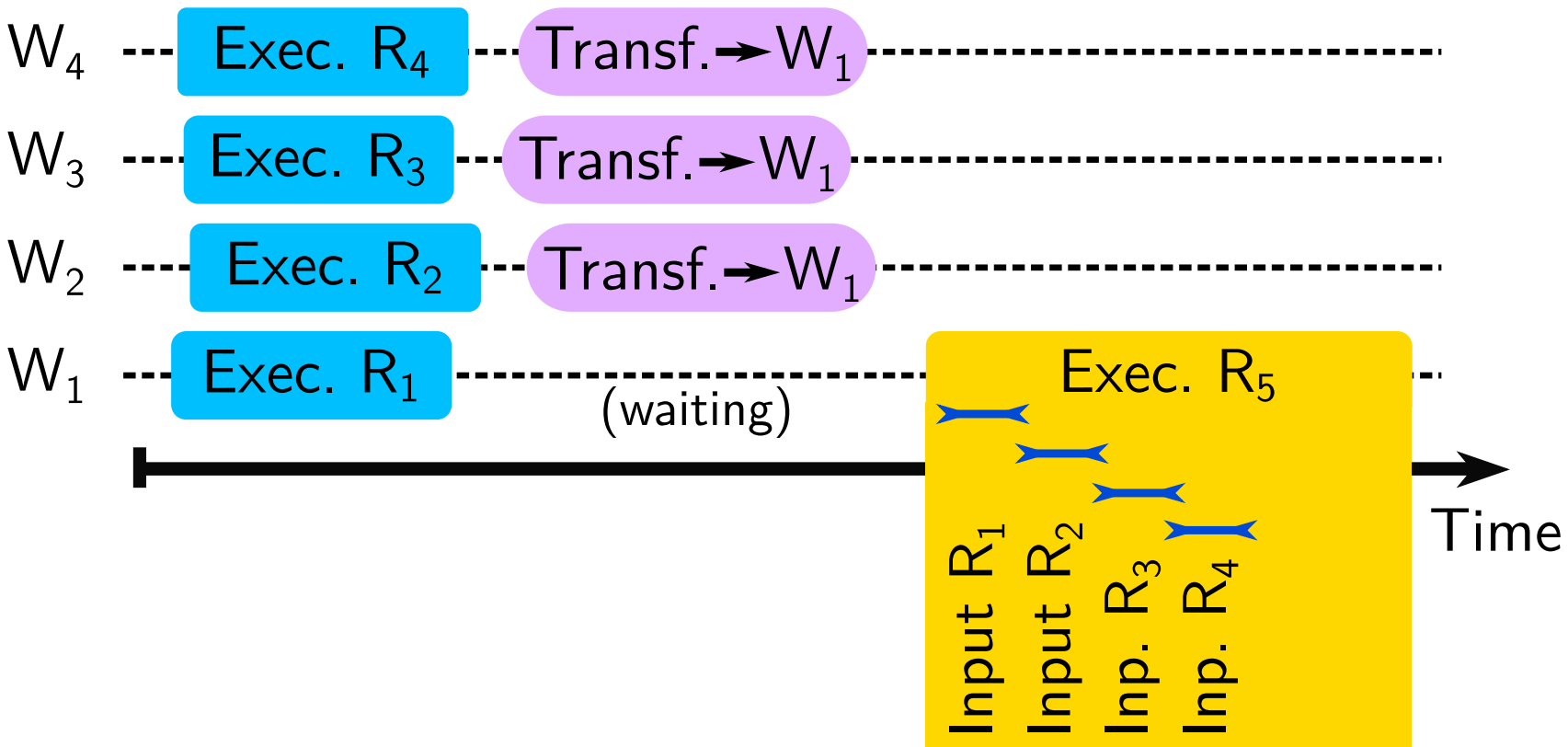
Example execution



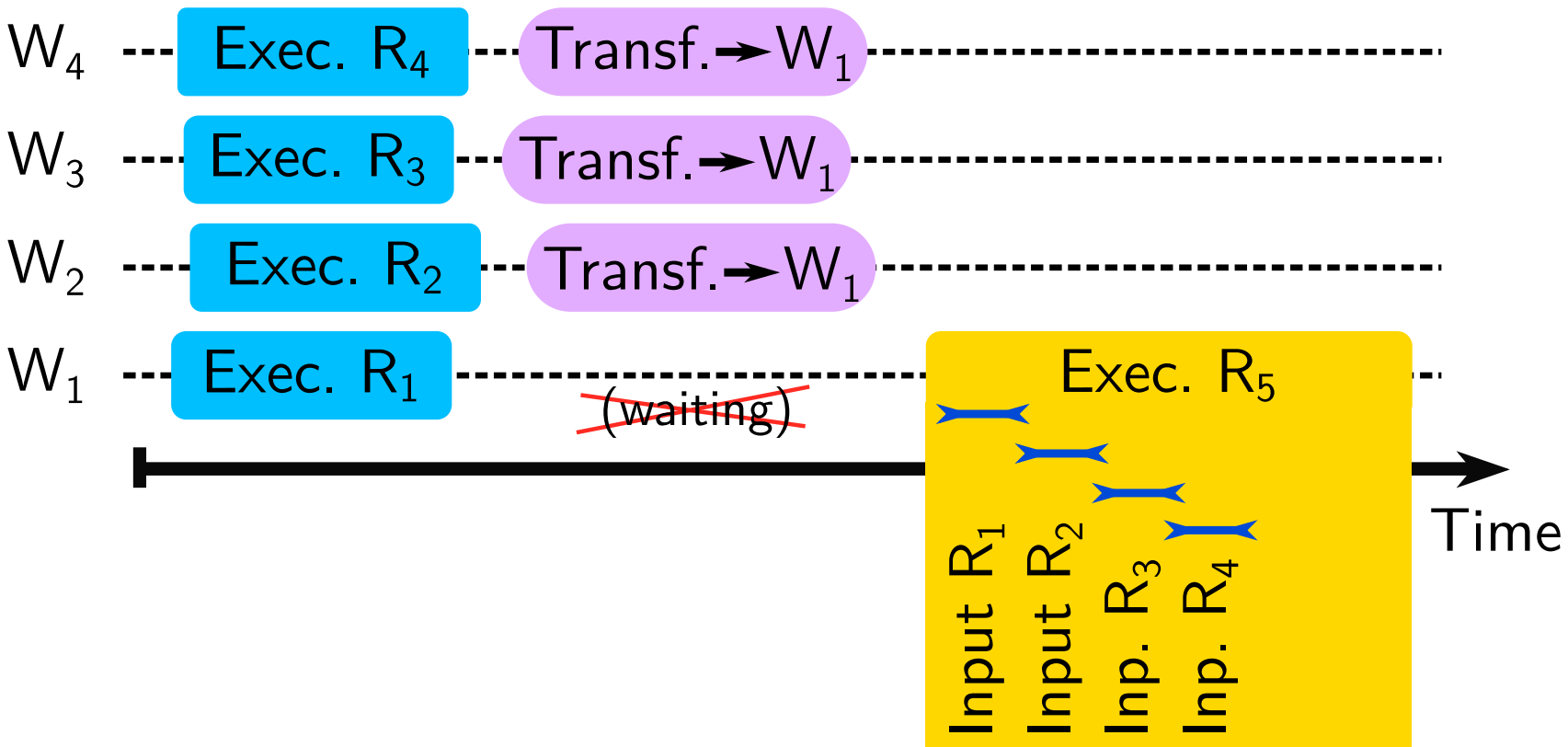
- Network congestion can slow down processing even further (effects of data losses at the transport protocol layer)
- High delay to the start of the aggregation task
- Low performance and high execution costs (e.g., in computation clouds)

What can we do to improve this?

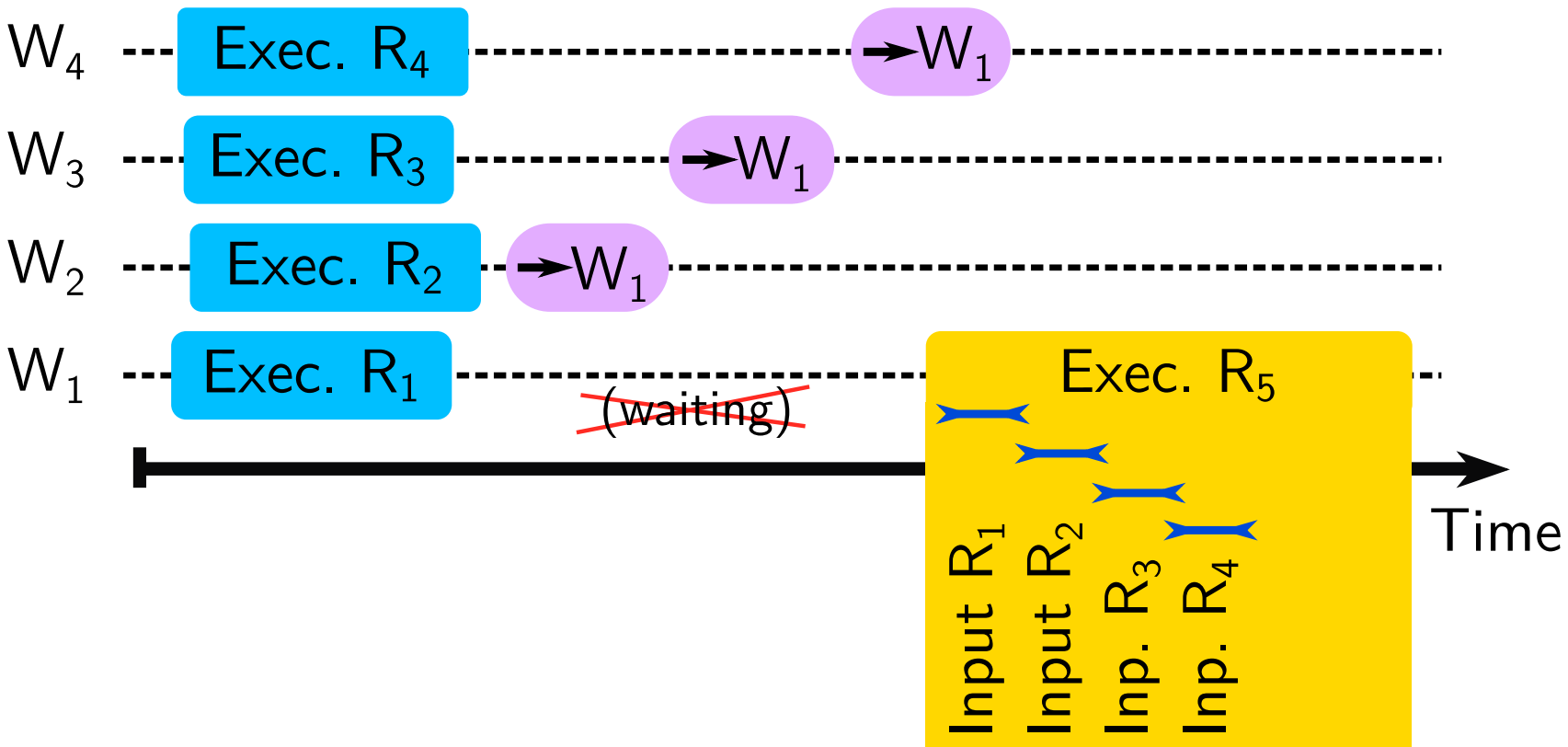
What can we do to improve this?



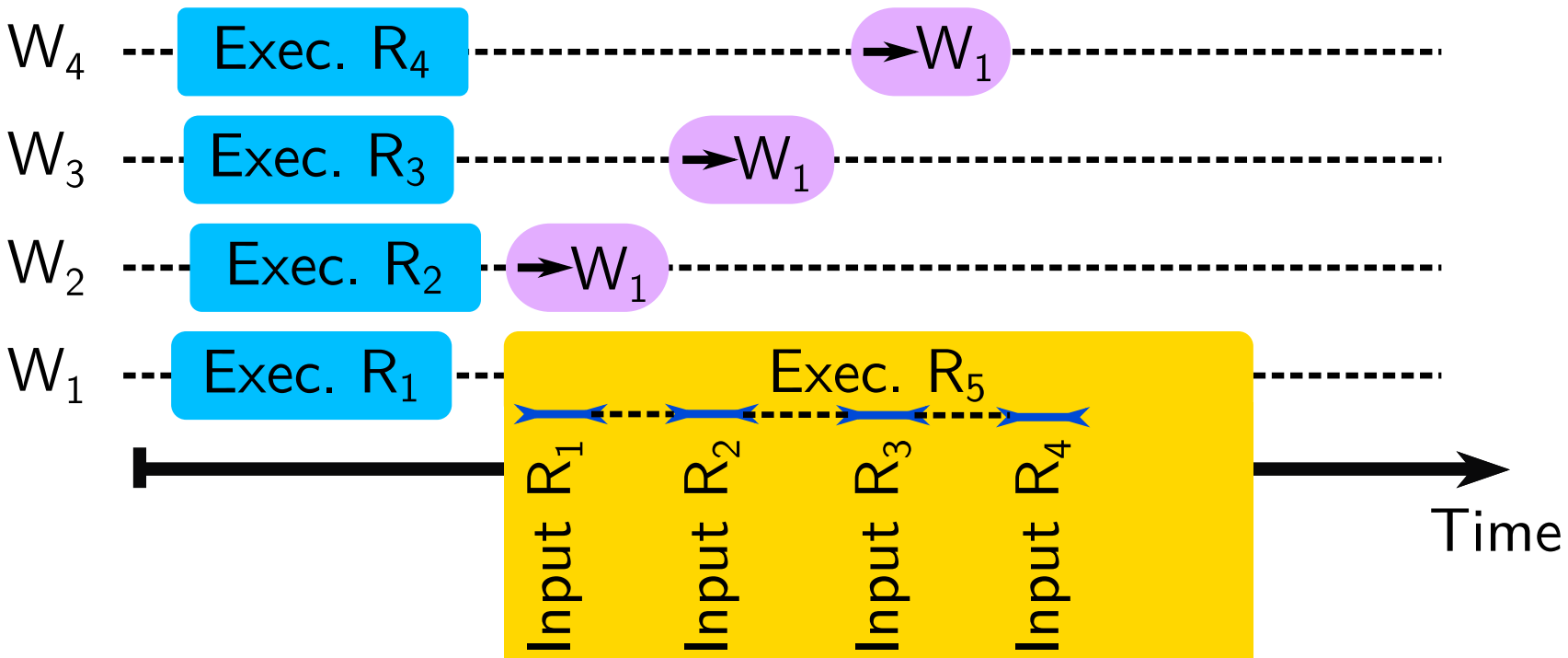
What can we do to improve this?



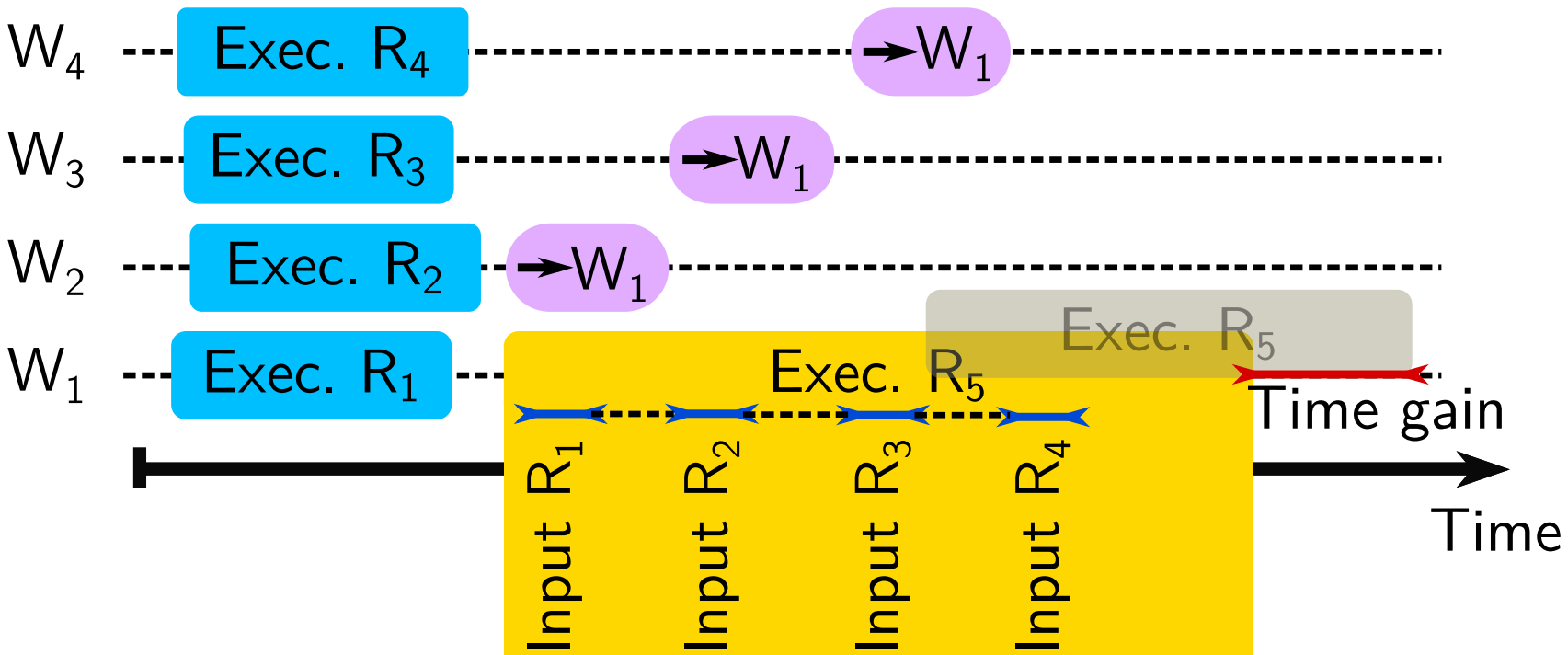
What can we do to improve this?



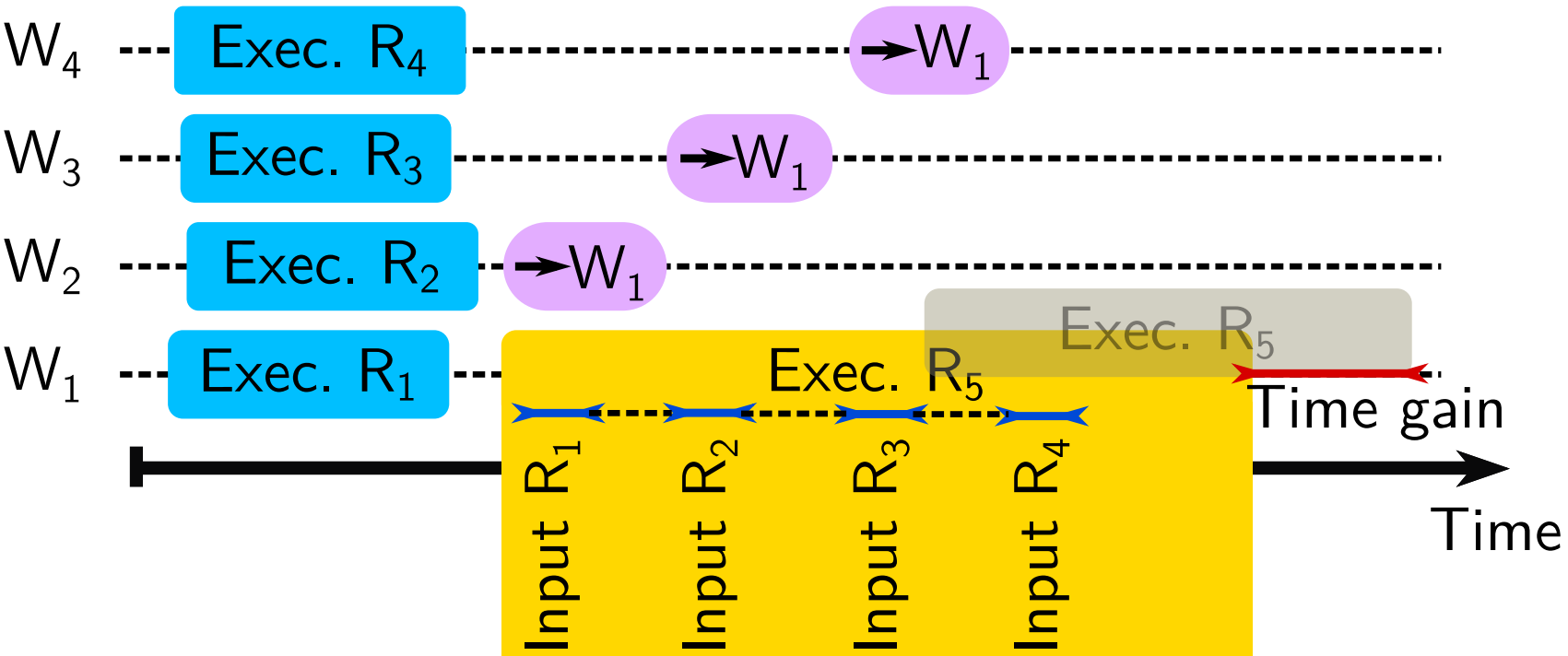
What can we do to improve this?



What can we do to improve this?



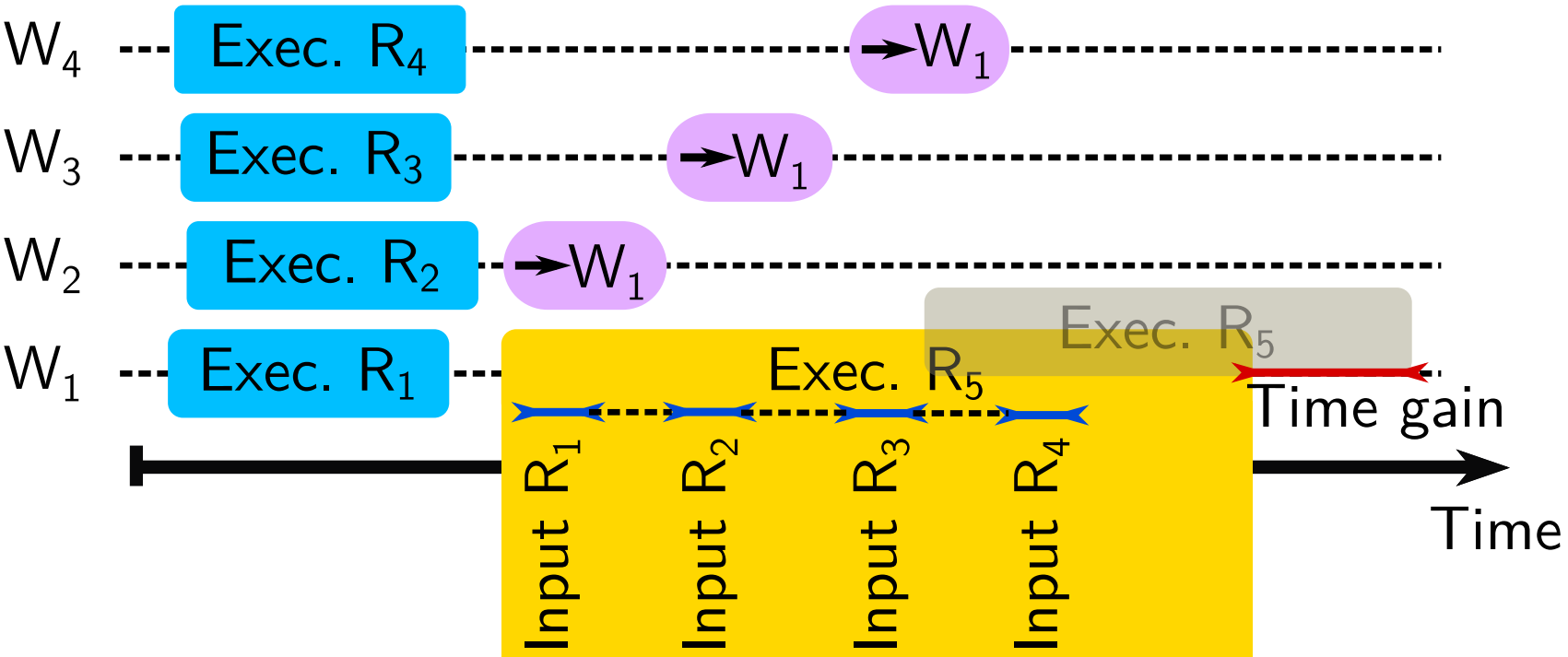
What can we do to improve this?



List of actions:

1. Obtain information on task's input characteristics
2. Refine the workflow and inform the execution engine
3. Let the aggregation task "feel comfortable" in changed setting

What can we do to improve this?



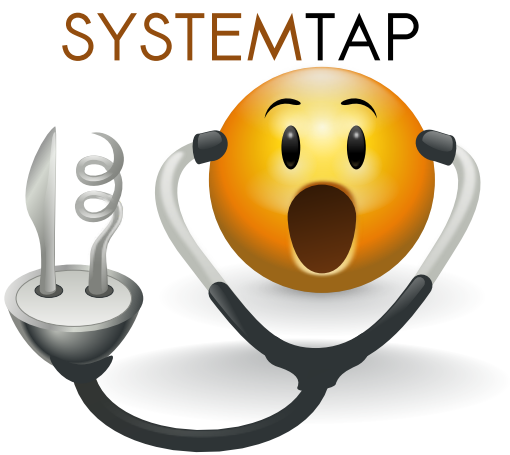
List of actions:

1. Obtain information on task's input characteristics
2. Refine the workflow and inform the execution engine
3. Let the aggregation task "feel comfortable" in changed setting

Obtaining input characteristics

1. Annotations to workflows
2. Manual code review
3. Automated profiling

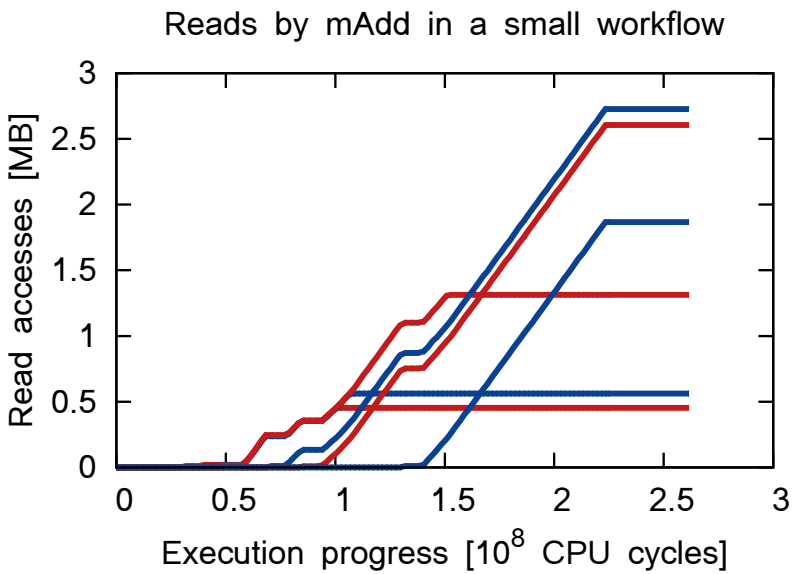
Automated profiling

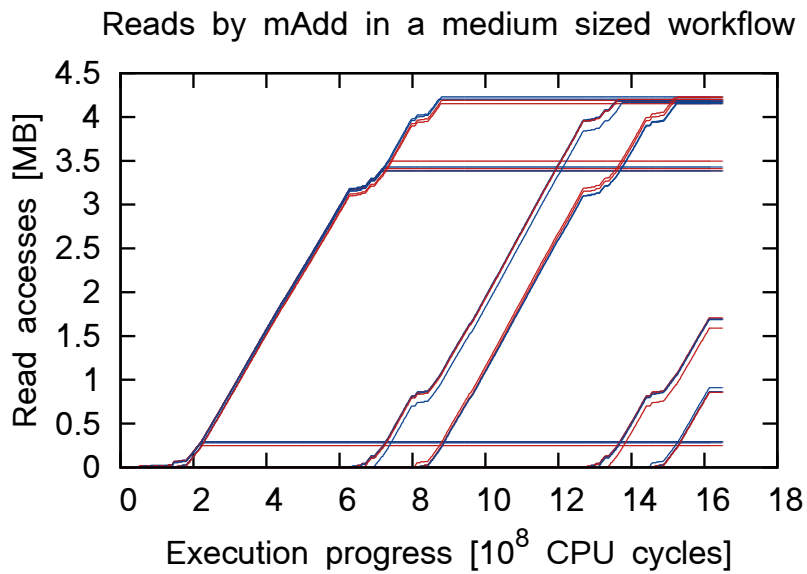


- Operating system instrumentation tool
- Enables interception of system calls (file open, read/write, file close)
- Record and evaluate logfiles with traces of conducted file accesses.

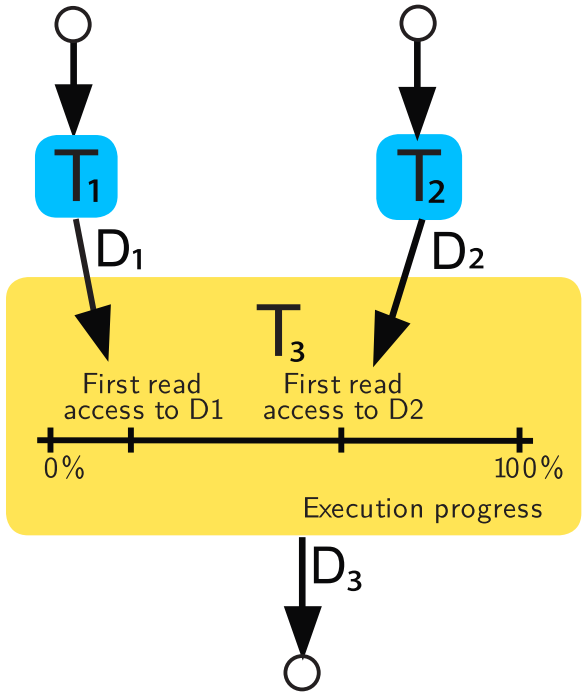
Automated profiling

- Operating system instrumentation tool
- Enables interception of system calls (file open, read/write, file close)
- Record and evaluate logfiles with traces of conducted file accesses.



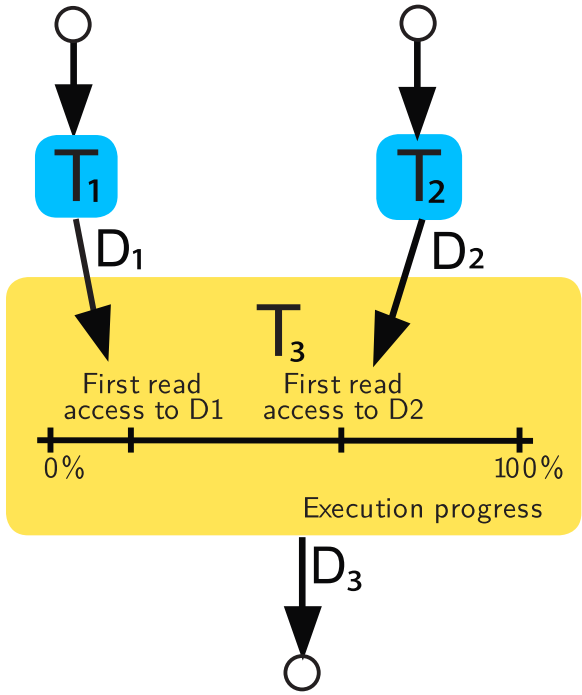


Refining workflow by transforming DAG

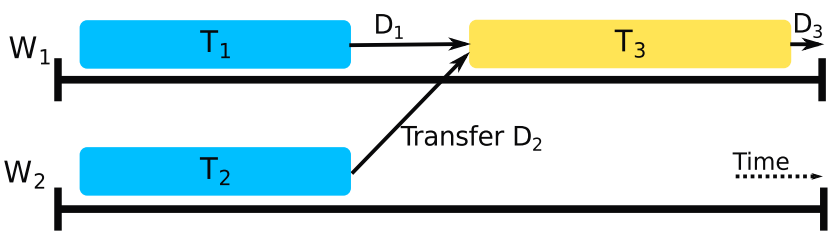


Original DAG

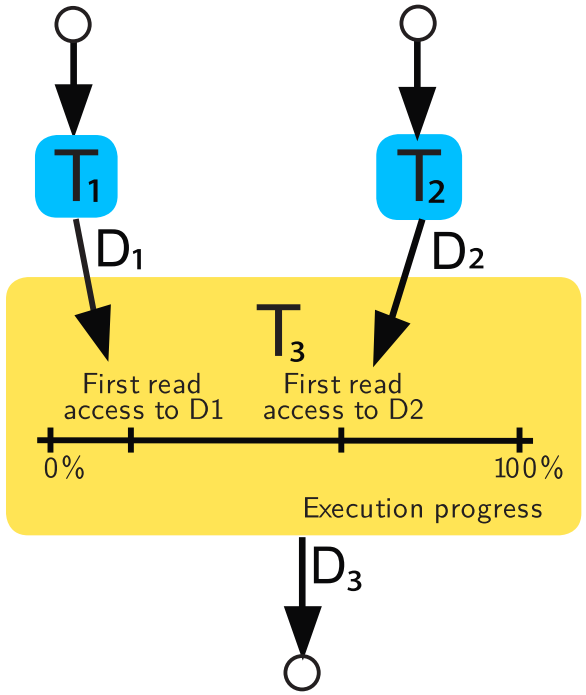
Refining workflow by transforming DAG



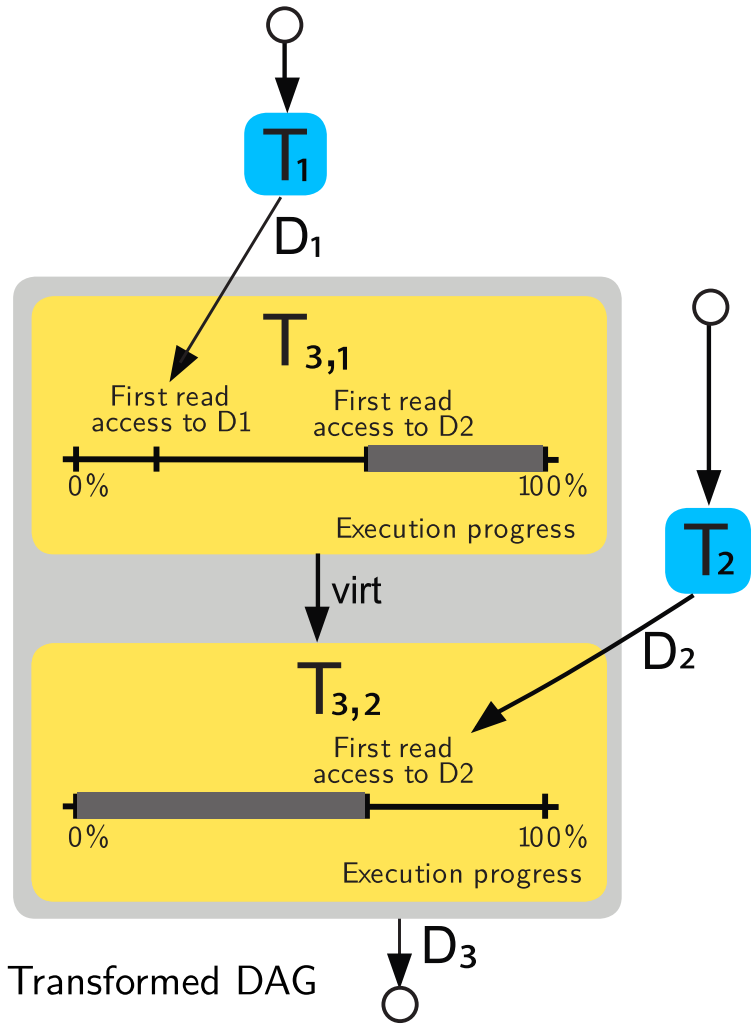
Original DAG



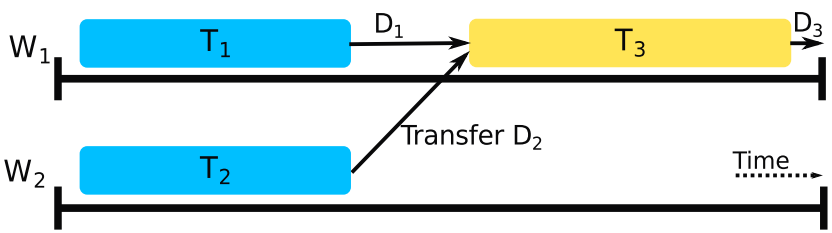
Refining workflow by transforming DAG



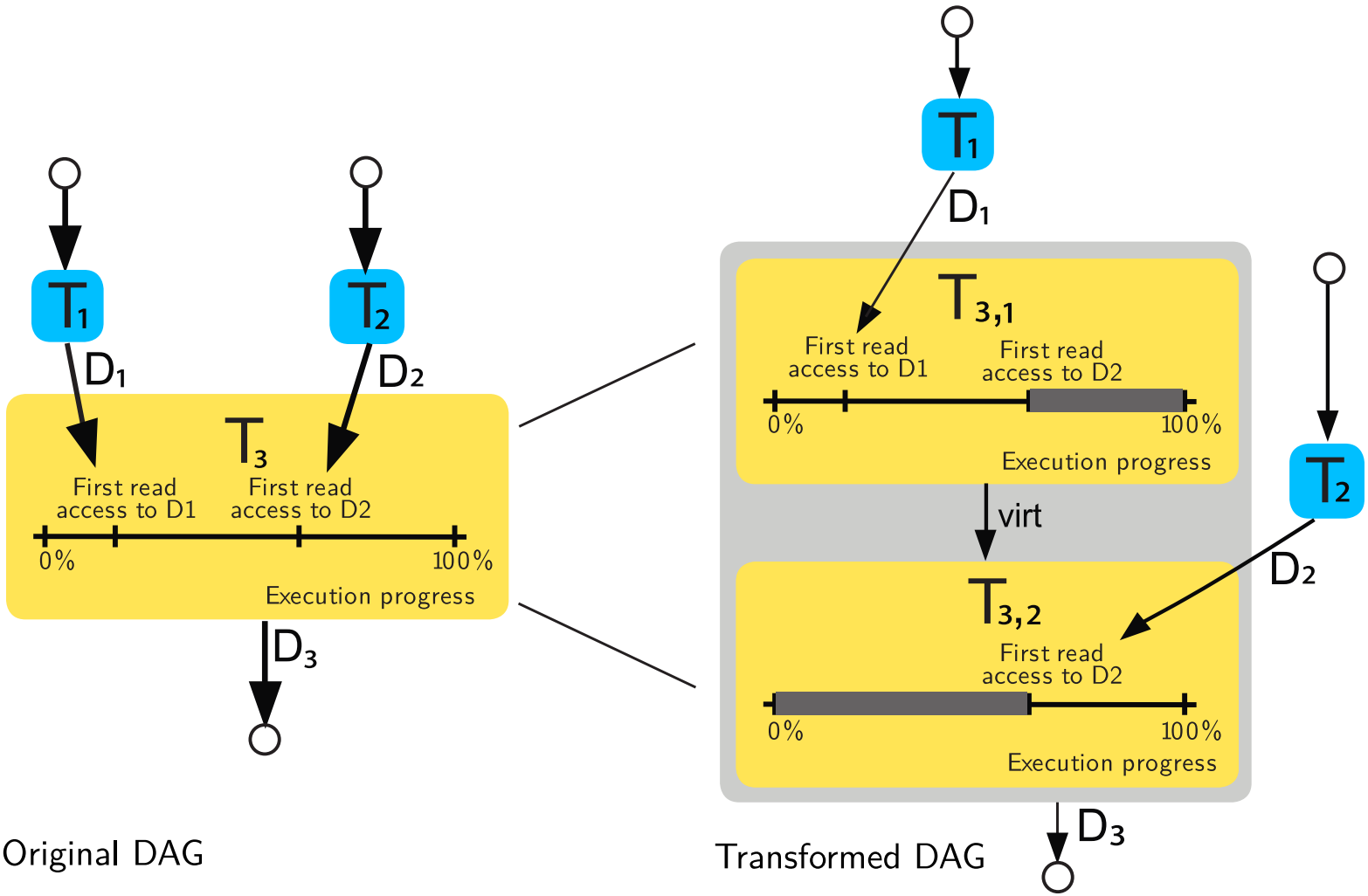
Original DAG

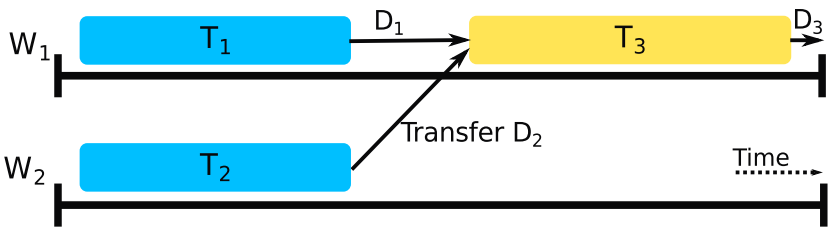


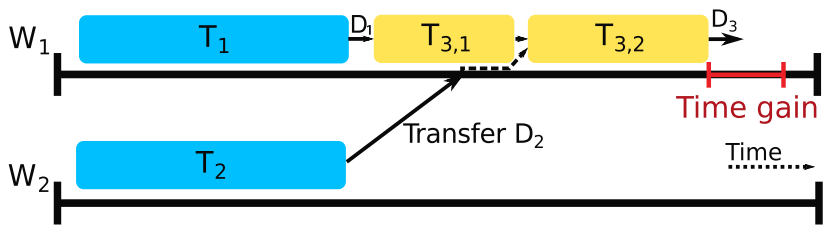
Transformed DAG



Refining workflow by transforming DAG



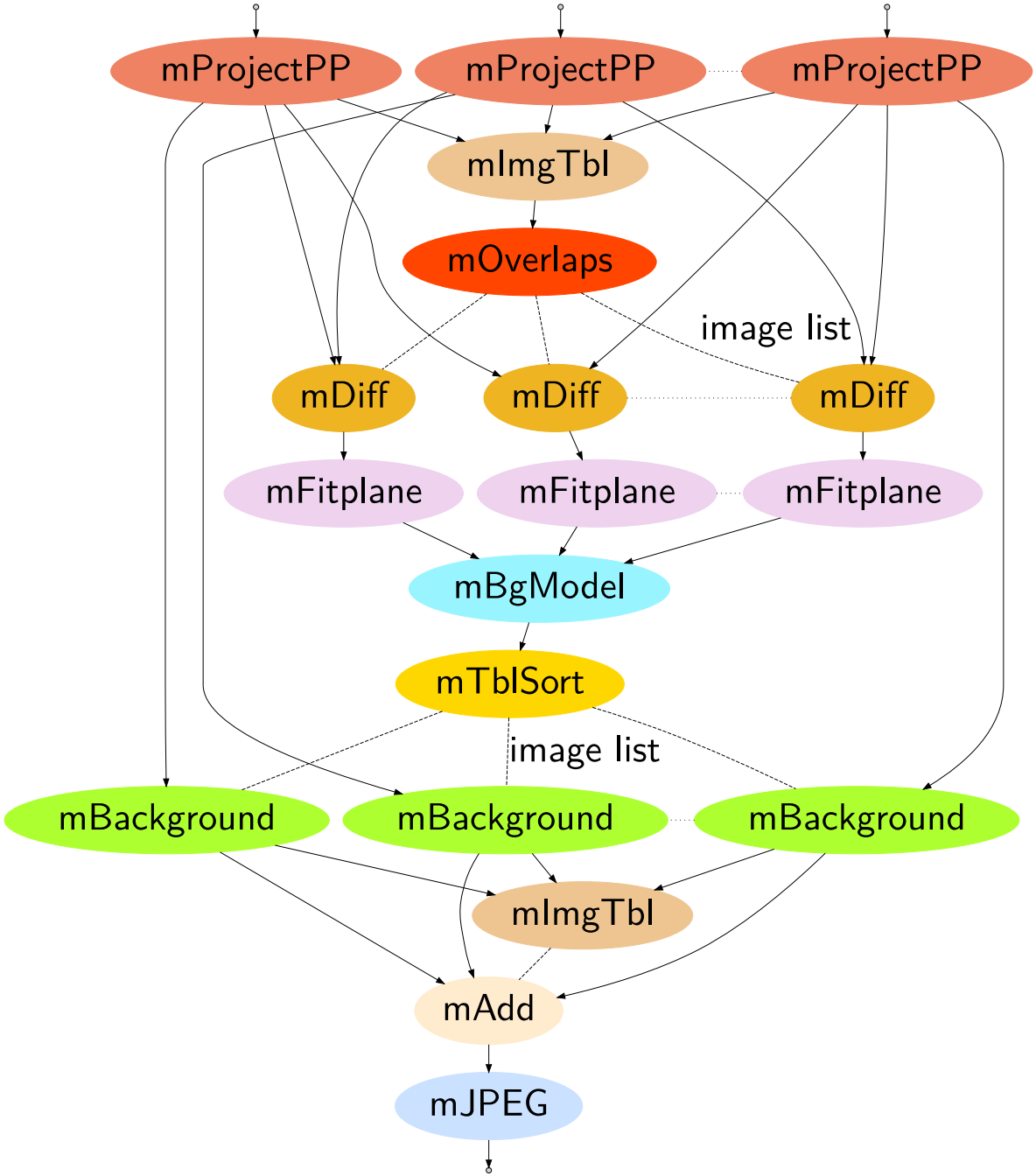




Realizing virtual task split

- Real task is transparently wrapped
- FUSE enables the setup of a virtual File system in USEr space
- Access to input files is performed through our wrapper
- Wrapper is responsible for maintaining the correct execution logic

Evaluation with the Montage workflow



Simulating workflow execution

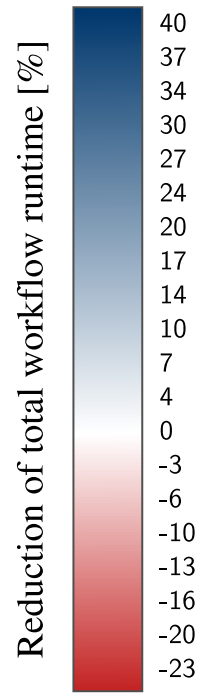
WorkflowSim

- Java-based simulation framework for scientific workflows
- Simulates an execution on a Pegasus/HTCondor stack
- Use provided Montage workflows with 25, 50, 100, 1000 tasks
- Python script conducted DAG transformation of DAX files
- Network configured as bottleneck (by bandwidth limitation)

W. Chen and E. Deelman, "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," in eScience'12.

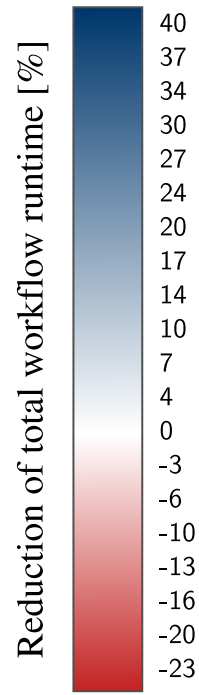
Simulation results

#VMs	#Tasks	Scheduling and planning algorithms: Min-Min	Max-Min	Round-robin	HEFT	DHEFT	Random
5	25	10.5	15.0	12.9	12.7	11.1	40.5
5	50	10.4	13.1	15.5	-22.8	15.2	39.9
5	100	10.1	11.1	12.1	8.7	13.4	12.6
5	1000	11.1	10.3	10.4	7.3	7.5	10.9
10	25	14.5	14.5	15.7	11.1	11.3	7.7
10	50	14.7	18.9	14.8	12.1	13.3	4.8
10	100	14.5	17.2	21.2	10.3	19.5	11.3
10	1000	17.0	16.4	16.1	8.6	10.5	-0.1
50	25	14.5	14.5	15.7	11.1	11.3	20.6
50	50	16.1	19.0	20.0	16.3	13.9	0.0
50	100	25.9	24.6	25.2	25.2	16.7	-0.8
50	1000	31.1	30.7	31.1	29.9	15.1	20.7
100	25	14.5	14.5	15.7	11.1	11.3	-8.4
100	50	16.1	19.0	20.0	16.3	13.9	1.4
100	100	24.8	26.6	27.6	25.1	16.7	33.9
100	1000	34.2	33.5	33.7	33.1	18.8	4.7

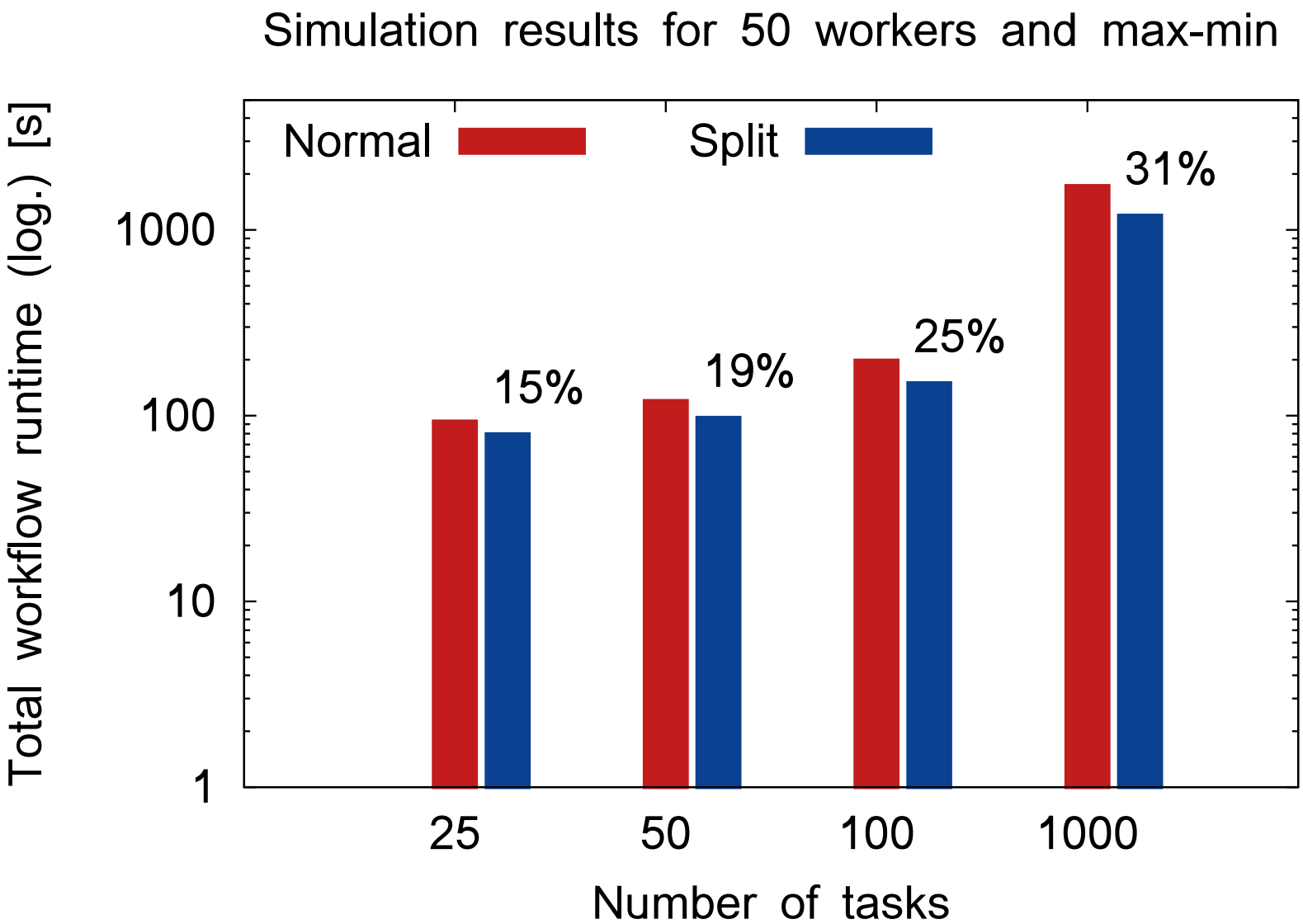


Simulation results

#VMs	#Tasks	Scheduling and planning algorithms: Min-Min	Max-Min	Round-robin	HEFT	DHEFT	Random
5	25	10.5	15.0	12.9	12.7	11.1	40.5
5	50	10.4	13.1	15.5	-22.8	15.2	39.9
5	100	10.1	11.1	12.1	8.7	13.4	12.6
5	1000	11.1	10.3	10.4	7.3	7.5	10.9
10	25	14.5	14.5	15.7	11.1	11.3	7.7
10	50	14.7	18.9	14.8	12.1	13.3	4.8
10	100	14.5	17.2	21.2	10.3	19.5	11.3
10	1000	17.0	16.4	16.1	8.6	10.5	-0.1
50	25	14.5	14.5	15.7	11.1	11.3	20.6
50	50	16.1	19.0	20.0	16.3	13.9	0.0
50	100	25.9	24.6	25.2	25.2	16.7	-0.8
50	1000	31.1	30.7	31.1	29.9	15.1	20.7
100	25	14.5	14.5	15.7	11.1	11.3	-8.4
100	50	16.1	19.0	20.0	16.3	13.9	1.4
100	100	24.8	26.6	27.6	25.1	16.7	33.9
100	1000	34.2	33.5	33.7	33.1	18.8	4.7

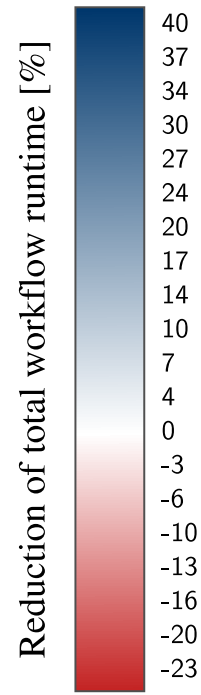


Variation of number of tasks

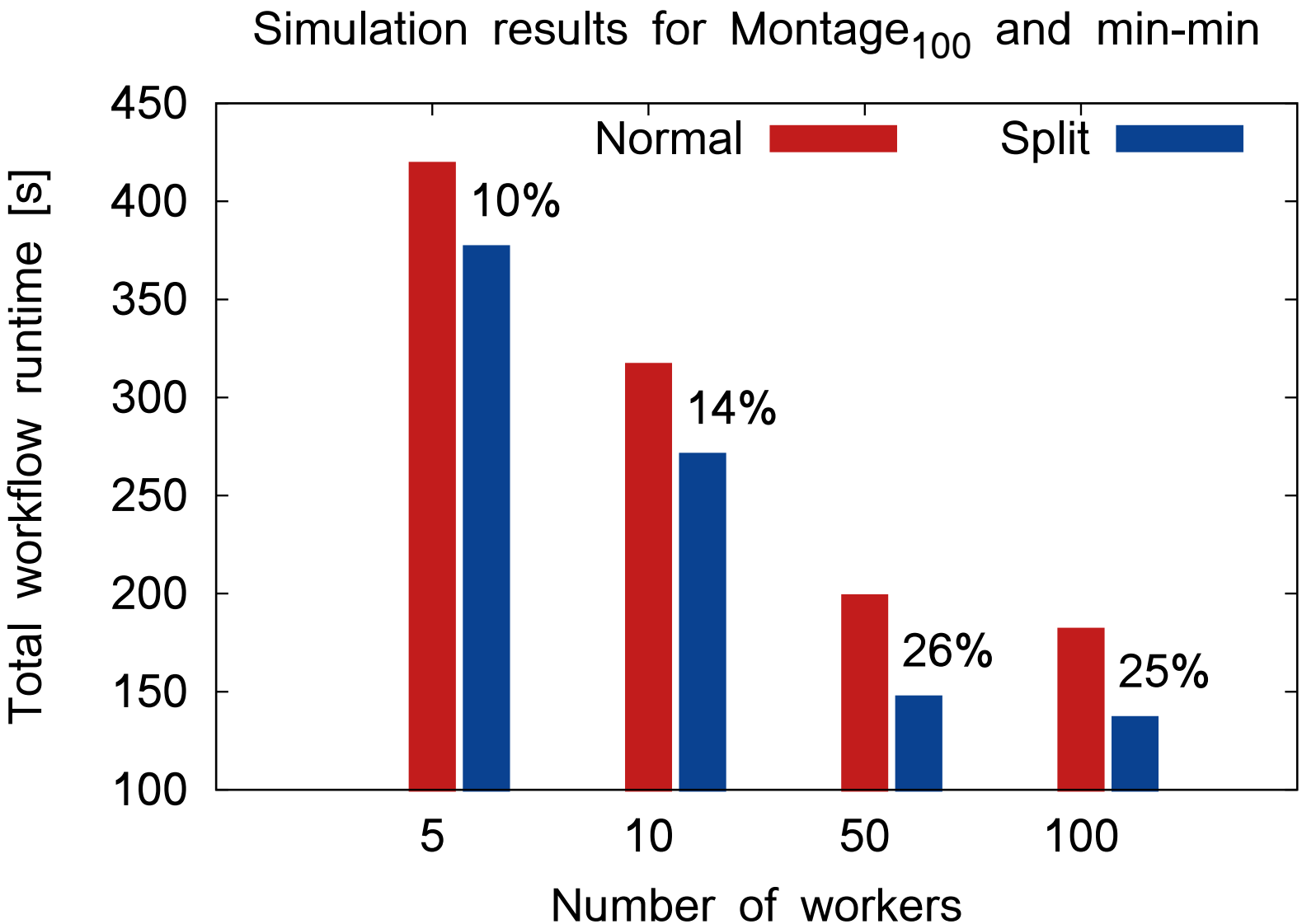


Variation of workers

#VMs	#Tasks	Scheduling and planning algorithms: Min-Min	Max-Min	Round-robin	HEFT	DHEFT	Random
5	25	10.5	15.0	12.9	12.7	11.1	40.5
5	50	10.4	13.1	15.5	-22.8	15.2	39.9
5	100	10.1	11.1	12.1	8.7	13.4	12.6
5	1000	11.1	10.3	10.4	7.3	7.5	10.9
10	25	14.5	14.5	15.7	11.1	11.3	7.7
10	50	14.7	18.9	14.8	12.1	13.3	4.8
10	100	14.5	17.2	21.2	10.3	19.5	11.3
10	1000	17.0	16.4	16.1	8.6	10.5	-0.1
50	25	14.5	14.5	15.7	11.1	11.3	20.6
50	50	16.1	19.0	20.0	16.3	13.9	0.0
50	100	25.9	24.6	25.2	25.2	16.7	-0.8
50	1000	31.1	30.7	31.1	29.9	15.1	20.7
100	25	14.5	14.5	15.7	11.1	11.3	-8.4
100	50	16.1	19.0	20.0	16.3	13.9	1.4
100	100	24.8	26.6	27.6	25.1	16.7	33.9
100	1000	34.2	33.5	33.7	33.1	18.8	4.7

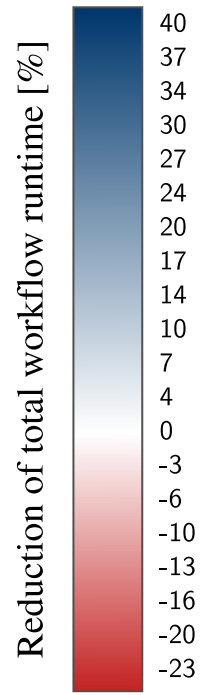


Variation of workers

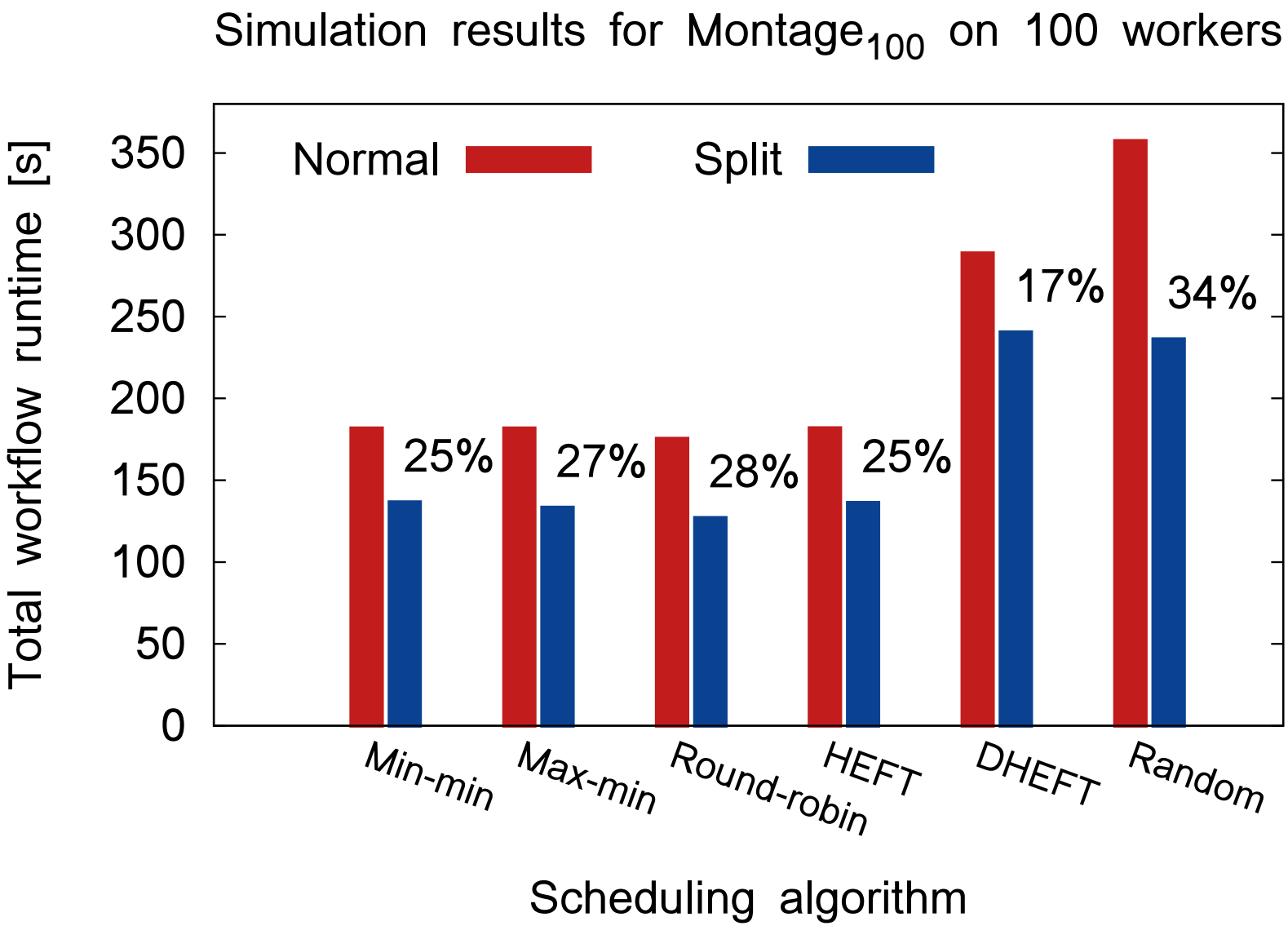


Variation of scheduling algorithms

#VMs	#Tasks	Scheduling and planning algorithms: Min-Min	Max-Min	Round-robin	HEFT	DHEFT	Random
5	25	10.5	15.0	12.9	12.7	11.1	40.5
5	50	10.4	13.1	15.5	-22.8	15.2	39.9
5	100	10.1	11.1	12.1	8.7	13.4	12.6
5	1000	11.1	10.3	10.4	7.3	7.5	10.9
10	25	14.5	14.5	15.7	11.1	11.3	7.7
10	50	14.7	18.9	14.8	12.1	13.3	4.8
10	100	14.5	17.2	21.2	10.3	19.5	11.3
10	1000	17.0	16.4	16.1	8.6	10.5	-0.1
50	25	14.5	14.5	15.7	11.1	11.3	20.6
50	50	16.1	19.0	20.0	16.3	13.9	0.0
50	100	25.9	24.6	25.2	25.2	16.7	-0.8
50	1000	31.1	30.7	31.1	29.9	15.1	20.7
100	25	14.5	14.5	15.7	11.1	11.3	-8.4
100	50	16.1	19.0	20.0	16.3	13.9	1.4
100	100	24.8	26.6	27.6	25.1	16.7	33.9
100	1000	34.2	33.5	33.7	33.1	18.8	4.7



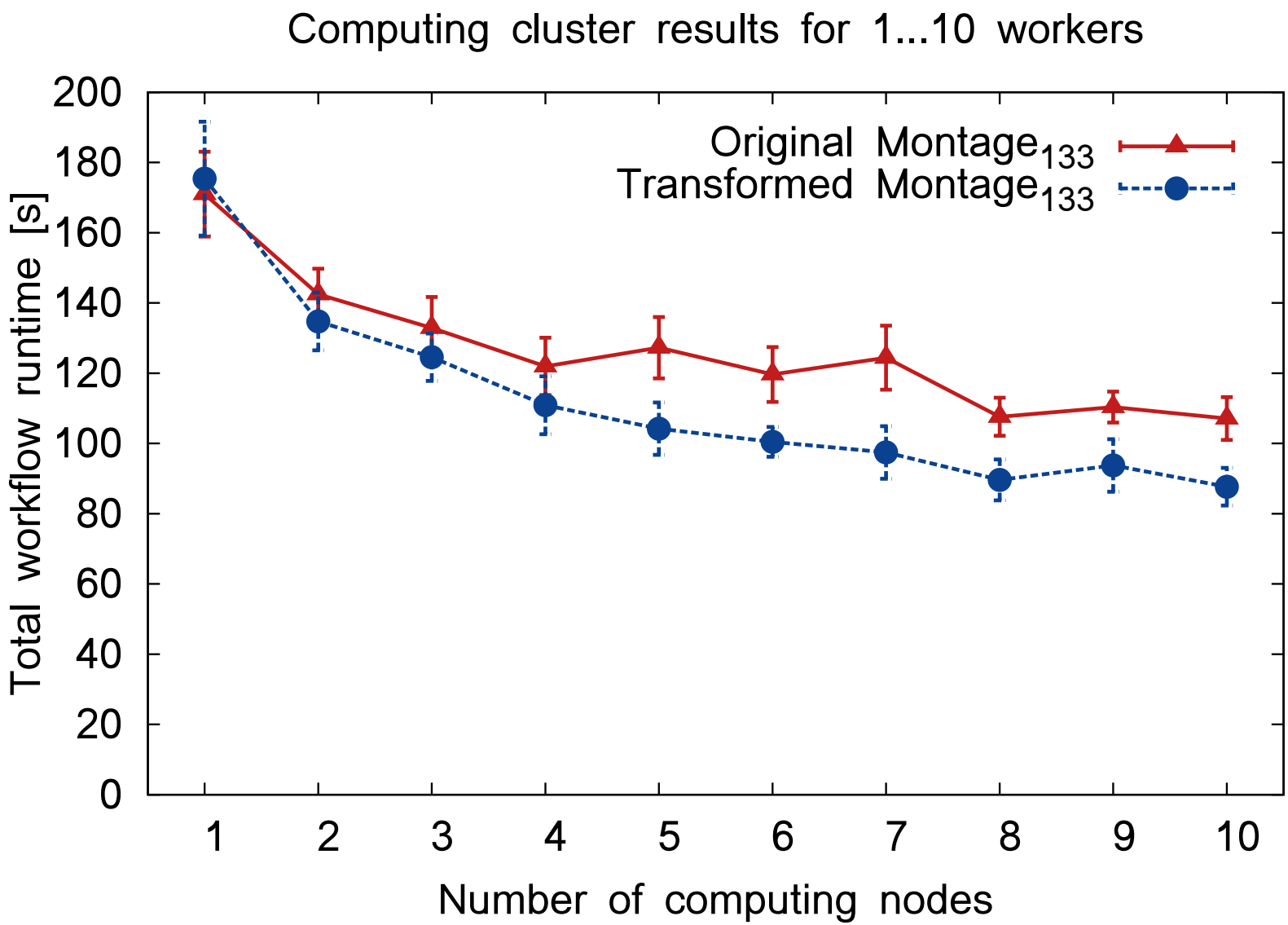
Variation of scheduling algorithms



Evaluation in a computing cluster

- Small cluster of up to 10 compute nodes
- Intel i7 CPU@ 2.5GHz, 8GB RAM, connected to common network switch with 1Gbit/s
- Execute Montage_133 workflow in Pegasus/HTCondor
- Network bandwidth was limited on application layer to 10Mbit/s
- 10 repetitions, mean values with 95% confidence intervals

Measurement results



Conclusion

- Many "legacy" workflows exist which are executed with classic semantics
- Our approach is applicable to aggregation tasks that are often the most time intensive tasks in a workflow
- By using DAG transformation, no changes to task implementations and execution engines are required

Conclusion

- Many "legacy" workflows exist which are executed with classic semantics
- Our approach is applicable to aggregation tasks that are often the most time intensive tasks in a workflow
- By using DAG transformation, no changes to task implementations and execution engines are required
- Simulation and real experiment show that performance can be improved by up to 15%
- Potential of outperforming the original workflow grows with increasing #workers and #tasks