Carbyne

Altruistic Scheduling in Multi-Resource Clusters

Robert Grandl, Mosharaf Chowdhury,
Aditya Akella, Ganesh Ananthanarayanan



Performance of Cluster Schedulers

We observe that:

- *Existing cluster schedulers focus on instantaneous fairness*

Performance of Cluster Schedulers

We observe that:

- *Existing cluster schedulers focus on instantaneous fairness*
- *Long-term fairness enables larger scheduling flexibility*

Performance of Cluster Schedulers

We observe that:

- *Existing cluster schedulers focus on instantaneous fairness*
- *Long-term fairness enables larger scheduling flexibility*
- *Data-parallel jobs provide ample opportunities for long-term optimizations*

Performance of Cluster Schedulers

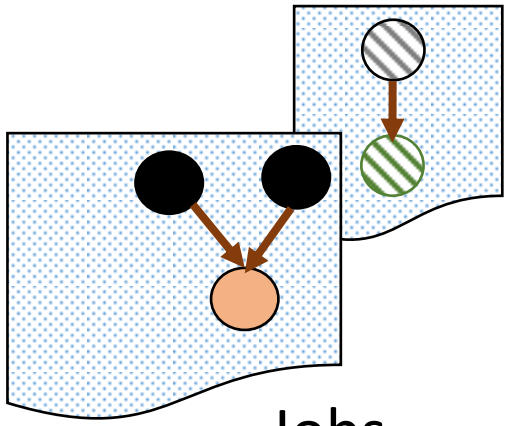
We observe that:

- *Existing cluster schedulers focus on instantaneous fairness*
- *Long-term fairness enables larger scheduling flexibility*
- *Data-parallel jobs provide ample opportunities for long-term optimizations*

Carbyne

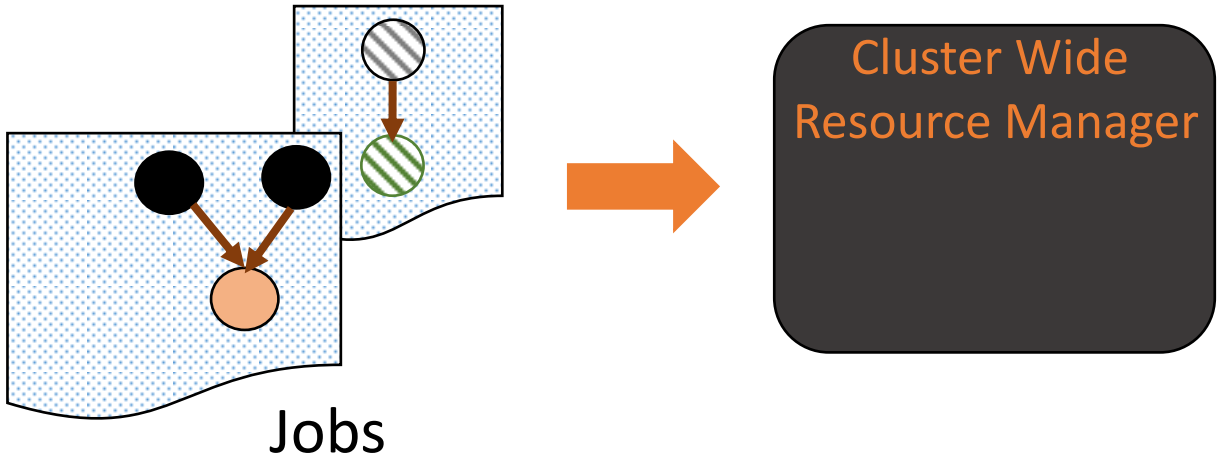
1.3x higher cluster efficiency;
1.6x lower average job completion time;
near-perfect fairness

Scheduling in Data Analytics Clusters

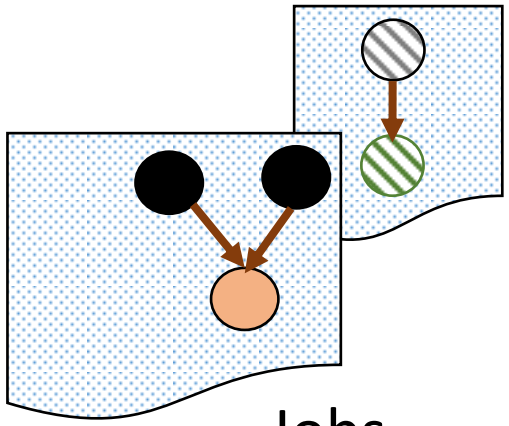


Jobs

Scheduling in Data Analytics Clusters

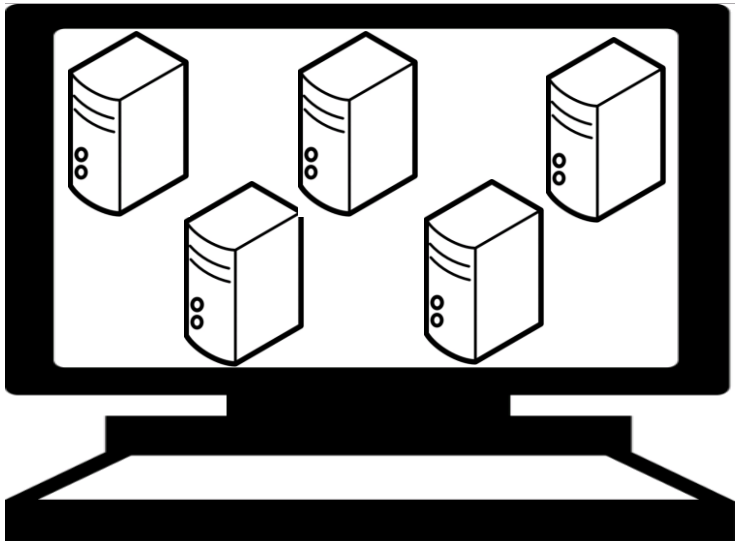


Scheduling in Data Analytics Clusters



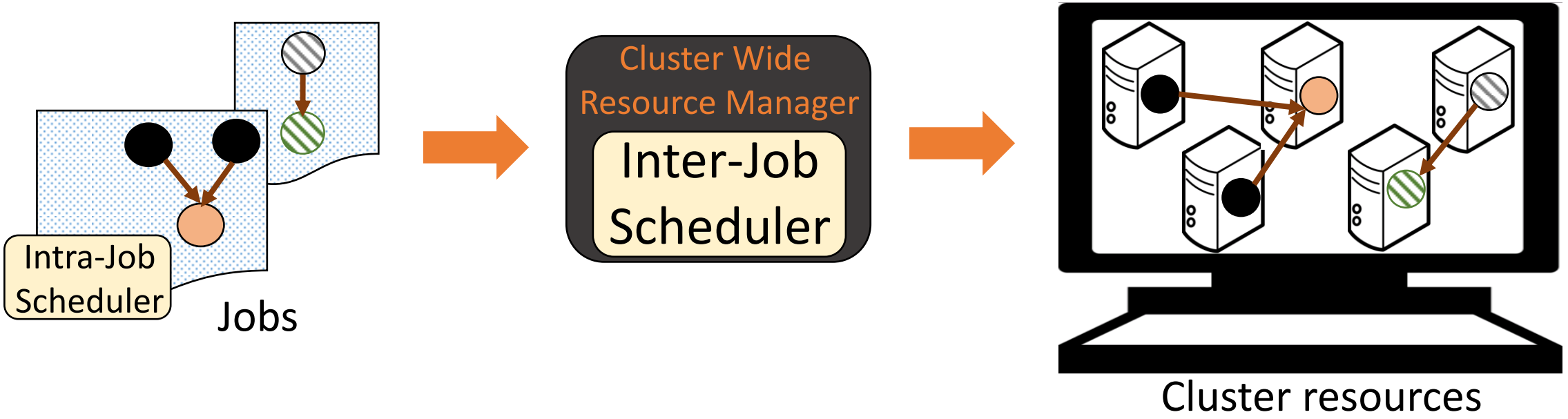
Jobs



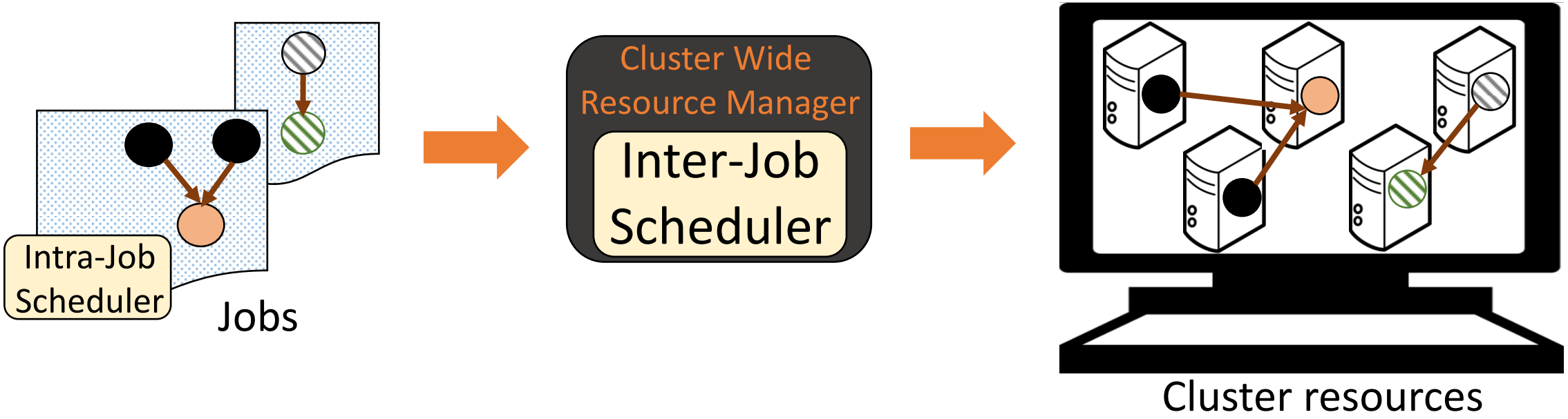


Cluster resources

Scheduling in Data Analytics Clusters



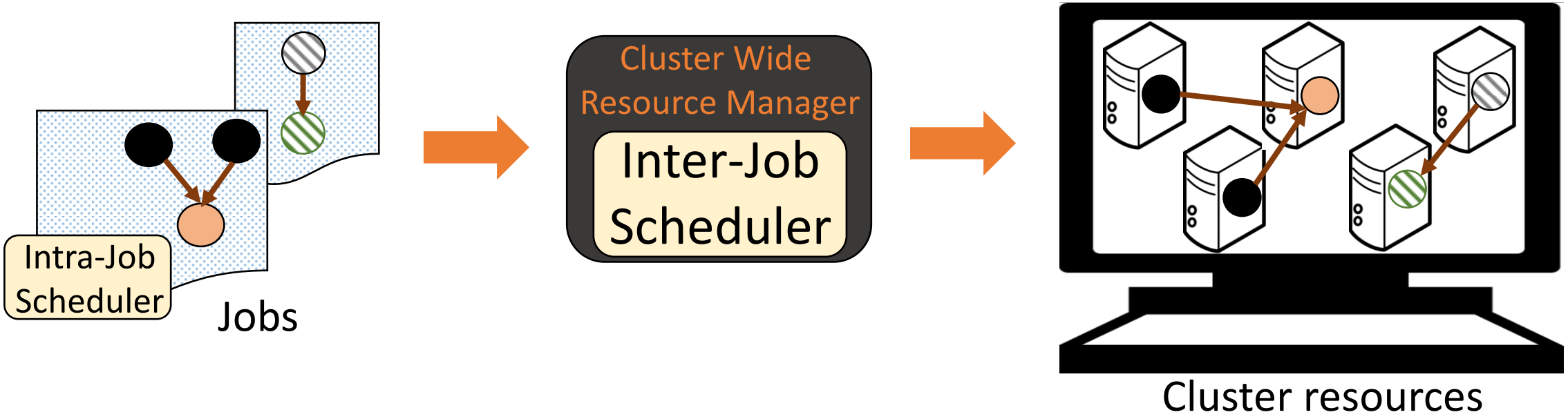
Scheduling in Data Analytics Clusters



Objectives:

#1: Fairness

Scheduling in Data Analytics Clusters



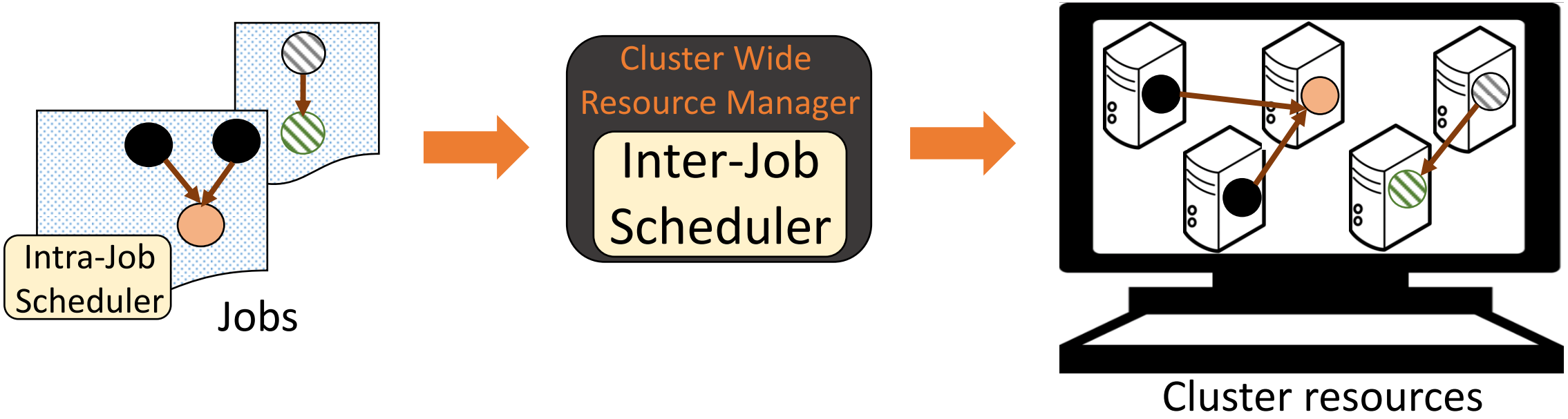
Objectives:

#1: Fairness

#2: Job Performance

#3: Cluster Efficiency

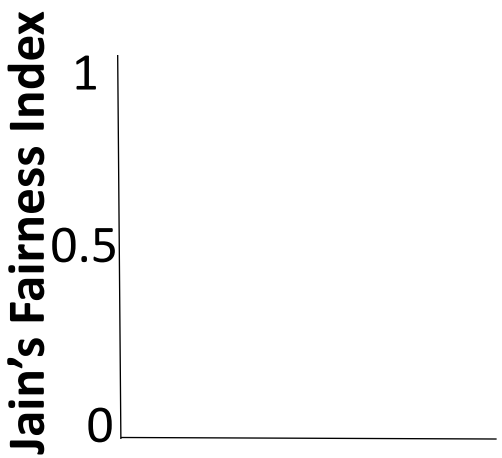
Scheduling in Data Analytics Clusters

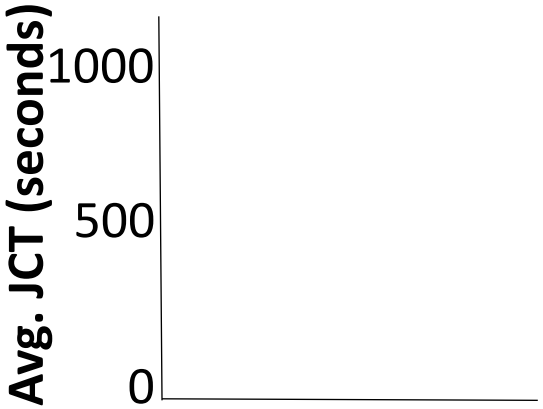


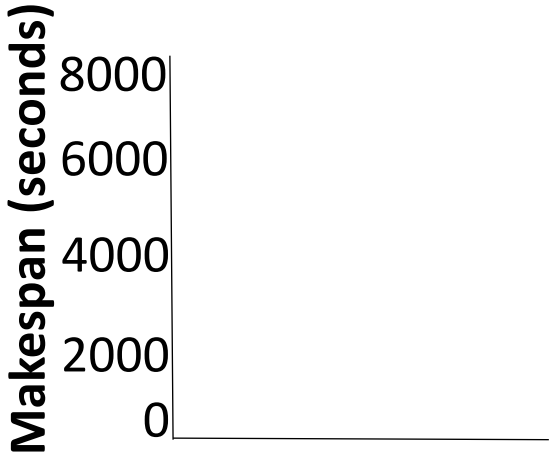
Objectives:

#1: Fairness + #2: Job Performance + #3: Cluster Efficiency = is difficult!

Scheduling in Data Analytics Clusters







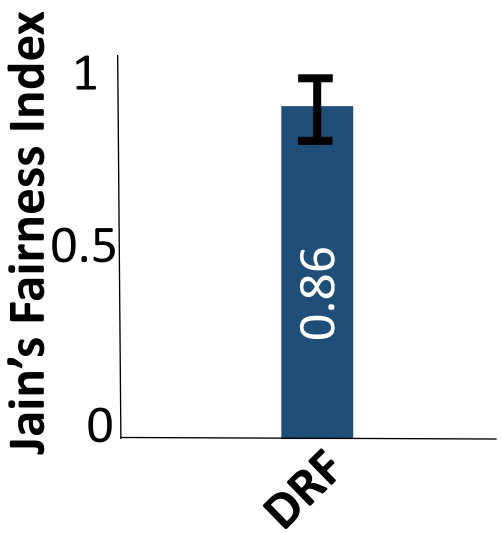
Inter-job fairness **Job Performance** **Cluster efficiency**

Objectives:

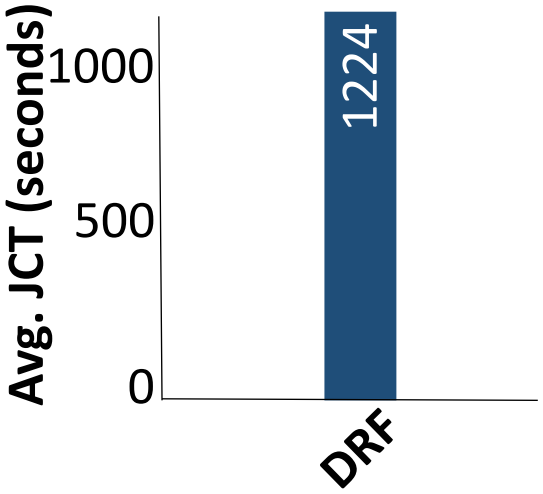
#1: Fairness + #2: Job Performance + #3: Cluster Efficiency = is difficult!

TPC-DS workload on a 100-machine cluster

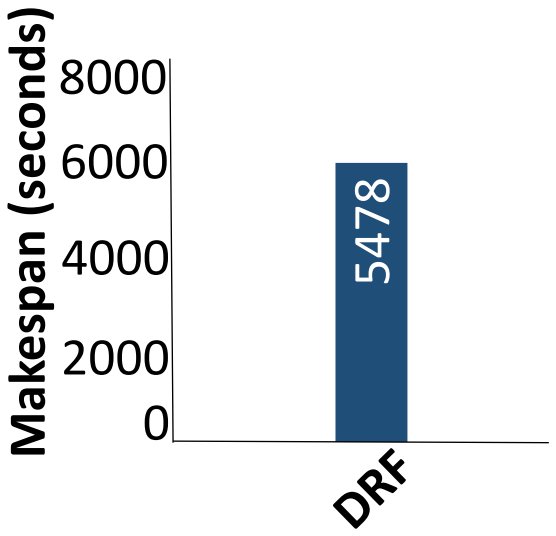
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

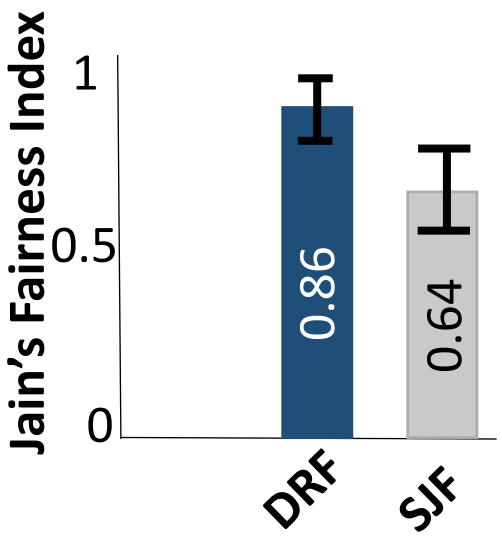
#1: Fairness ✚ #2: Job Performance ✚ #3: Cluster Efficiency = is difficult!

❑ DRF [NSDI'11]

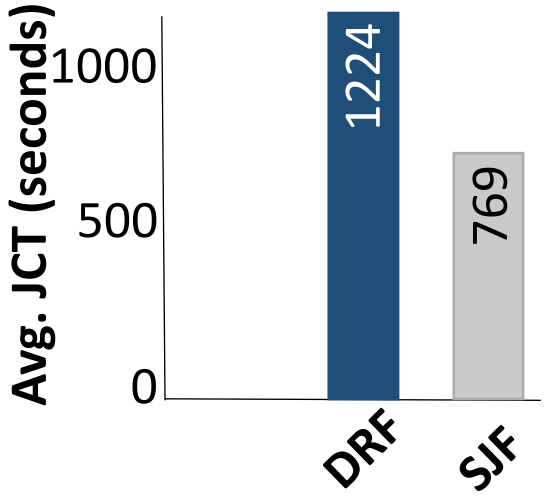
Dominant Resource Fairness
- Max-min fair sharing across multiple dimensions

TPC-DS workload on a 100-machine cluster

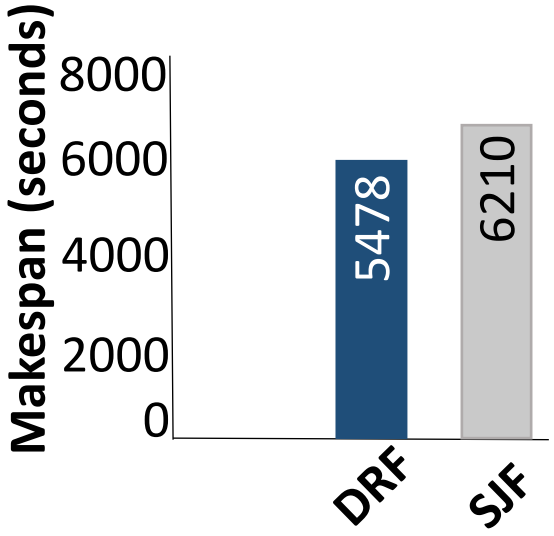
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)
  - Schedule jobs near completion first

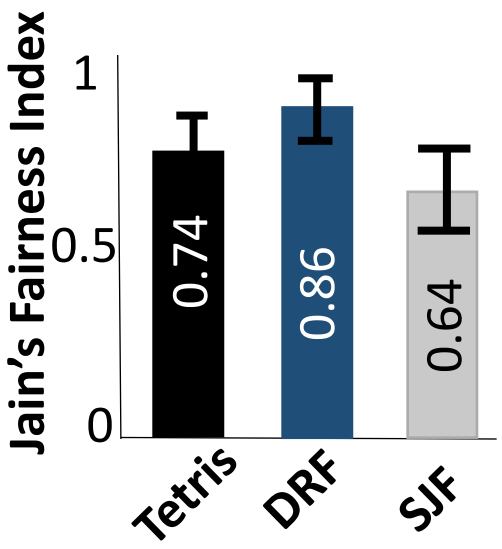
+

#3: Cluster Efficiency

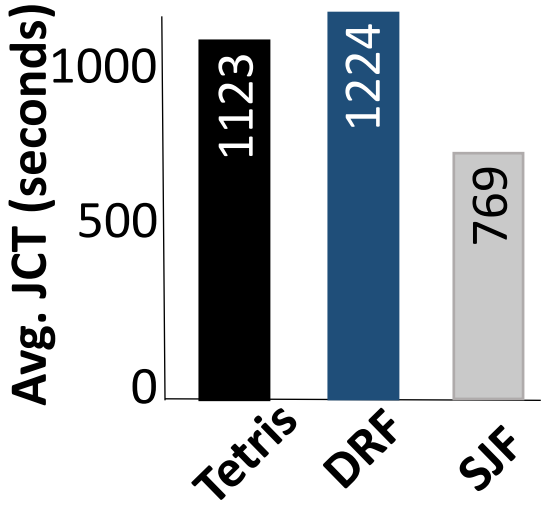
= is difficult!

TPC-DS workload on a 100-machine cluster

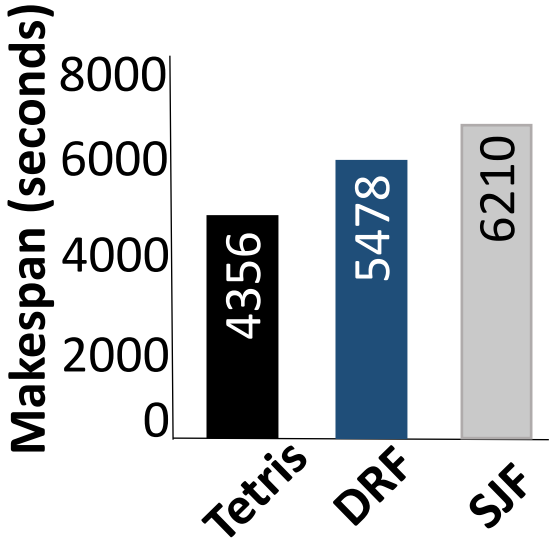
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)

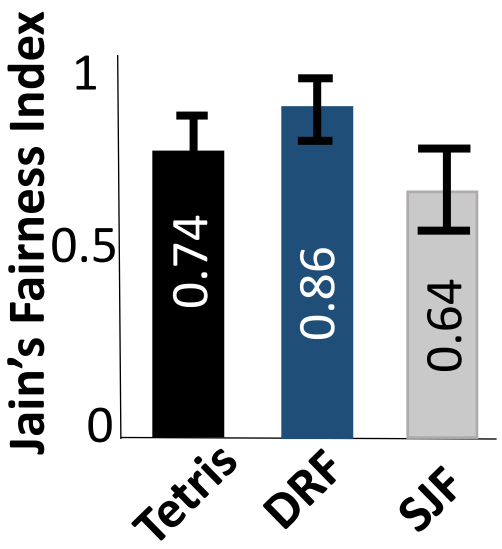
+

#3: Cluster Efficiency
- ❑ Tetris [SIGCOMM'14]
  - Efficiently pack resources

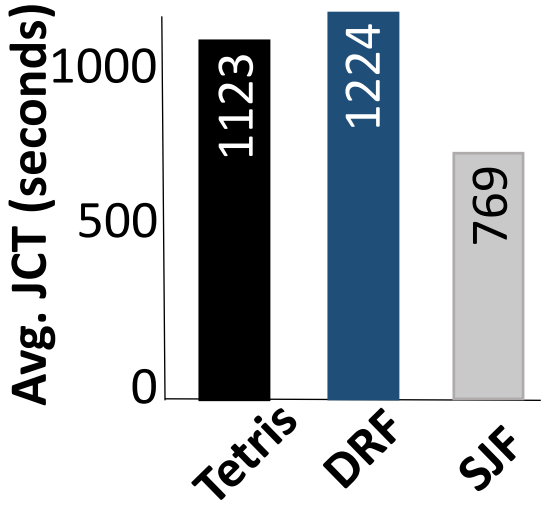
= is difficult!

TPC-DS workload on a 100-machine cluster

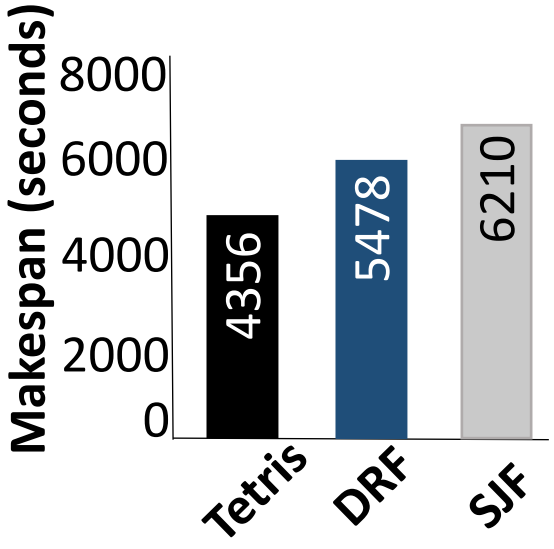
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)

+

#3: Cluster Efficiency
- ❑ Tetris [SIGCOMM'14]

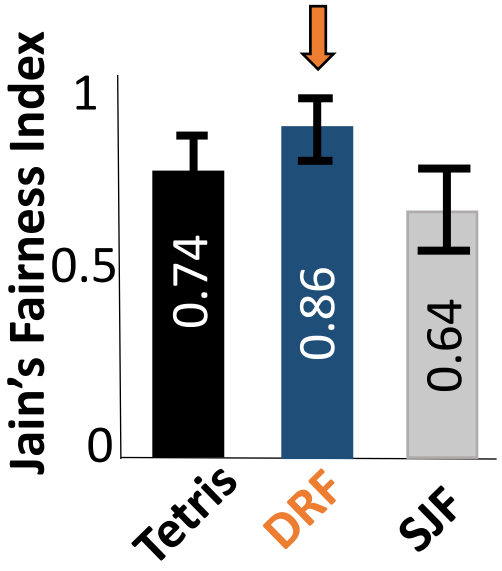
= is difficult!

Outperforms
- Preferred metric

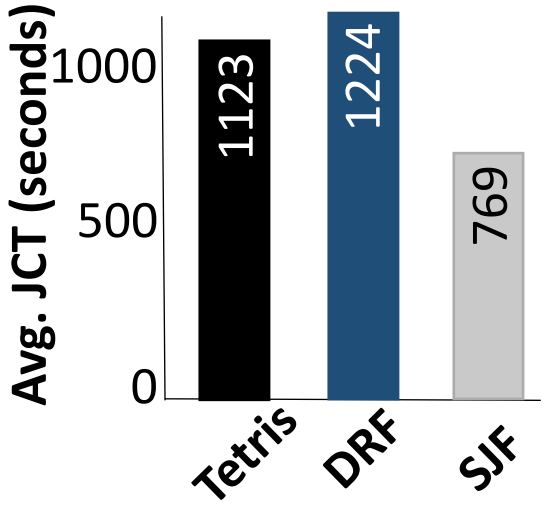
Underperforms
- Secondary metrics

TPC-DS workload on a 100-machine cluster

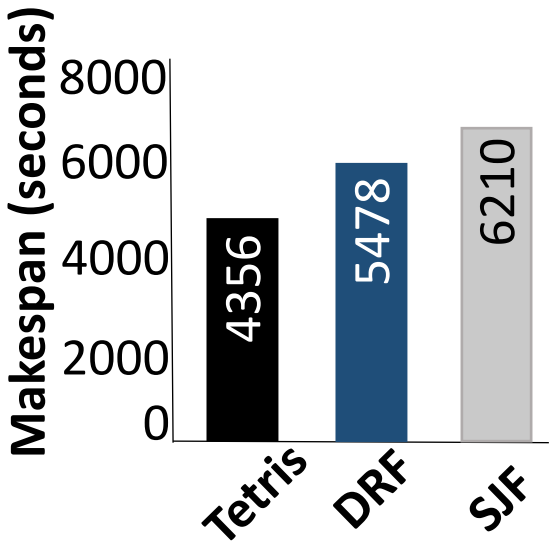
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)

+

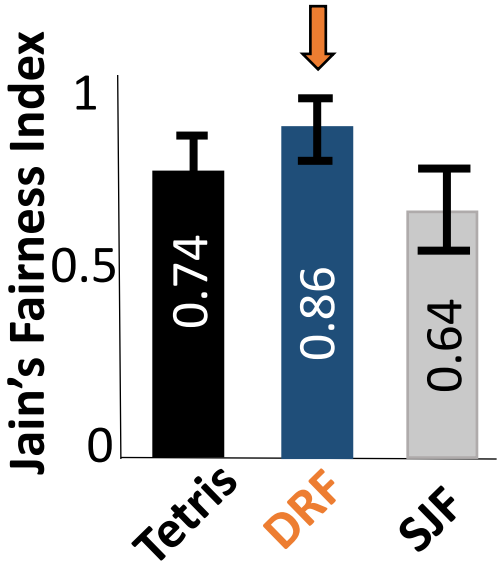
#3: Cluster Efficiency
- ❑ Tetris [SIGCOMM'14]

= is difficult!

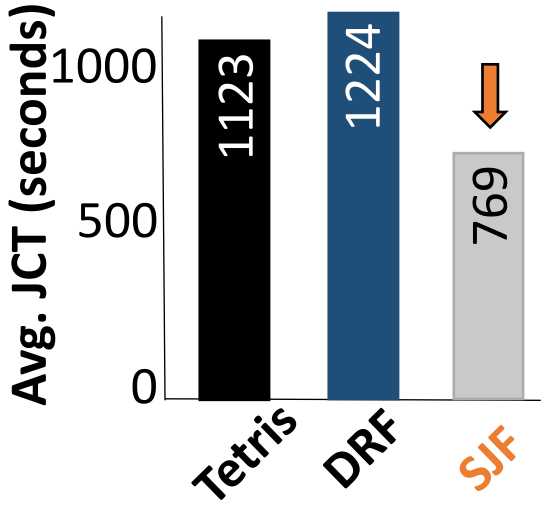
Outperforms	Underperforms
▪ Preferred metric	▪ Secondary metrics

TPC-DS workload on a 100-machine cluster

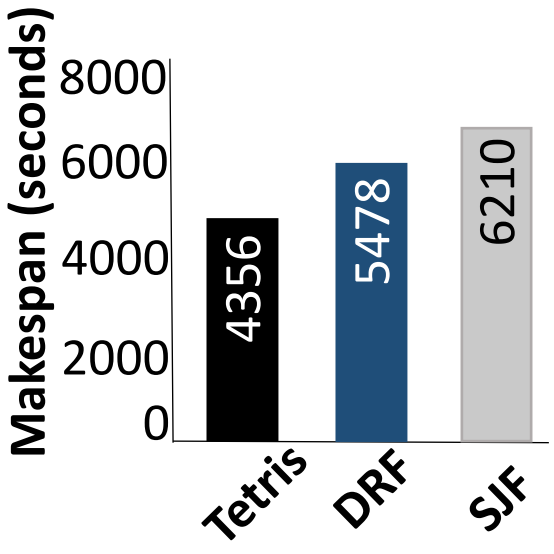
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)

+

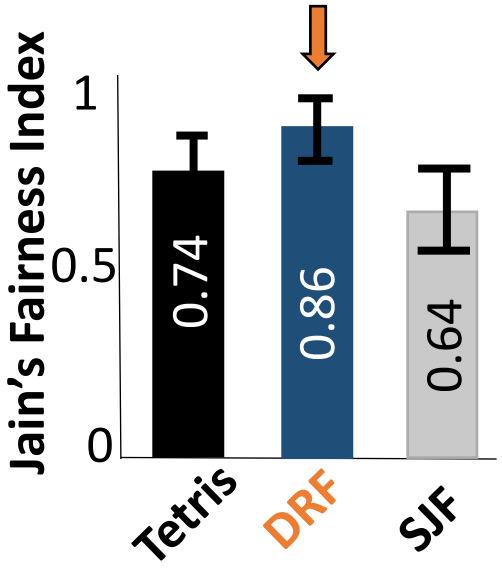
#3: Cluster Efficiency
- ❑ Tetris [SIGCOMM'14]

= is difficult!

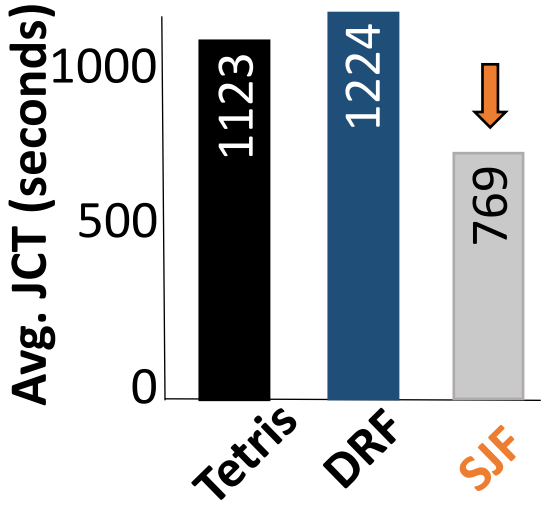
Outperforms	Underperforms
▪ Preferred metric	▪ Secondary metrics

TPC-DS workload on a 100-machine cluster

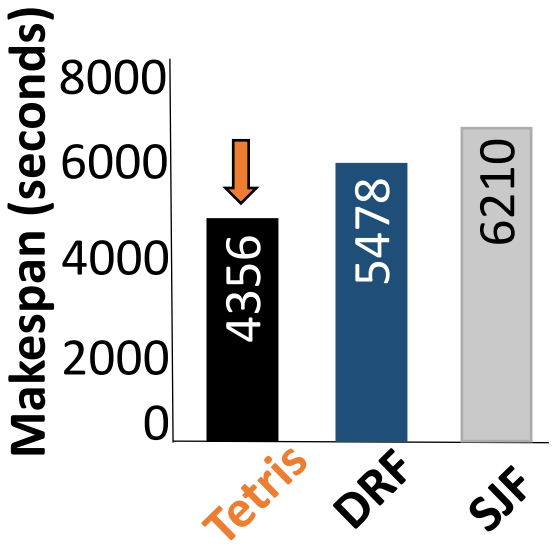
Scheduling in Data Analytics Clusters



Inter-job fairness



Job Performance



Cluster efficiency

Objectives:

#1: Fairness
- ❑ DRF [NSDI'11]

+

#2: Job Performance
- ❑ Shortest Job First (SJF)

+

#3: Cluster Efficiency
- ❑ Tetris [SIGCOMM'14]

= is difficult!

Outperforms	Underperforms
▪ Preferred metric	▪ Secondary metrics

TPC-DS workload on a 100-machine cluster

Scheduling in Data Analytics Clusters



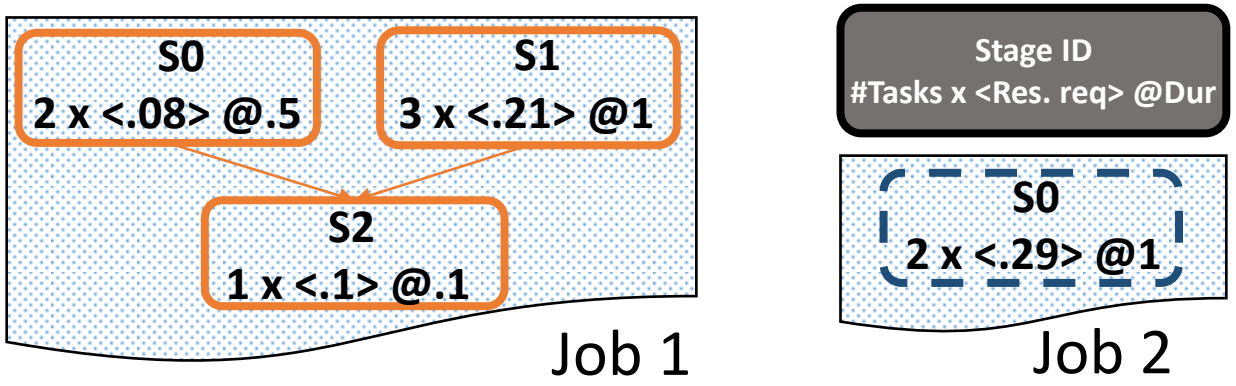
Is it possible to ensure fairness and still be competitive with the best approaches for the secondary metrics (job performance and cluster efficiency)?

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

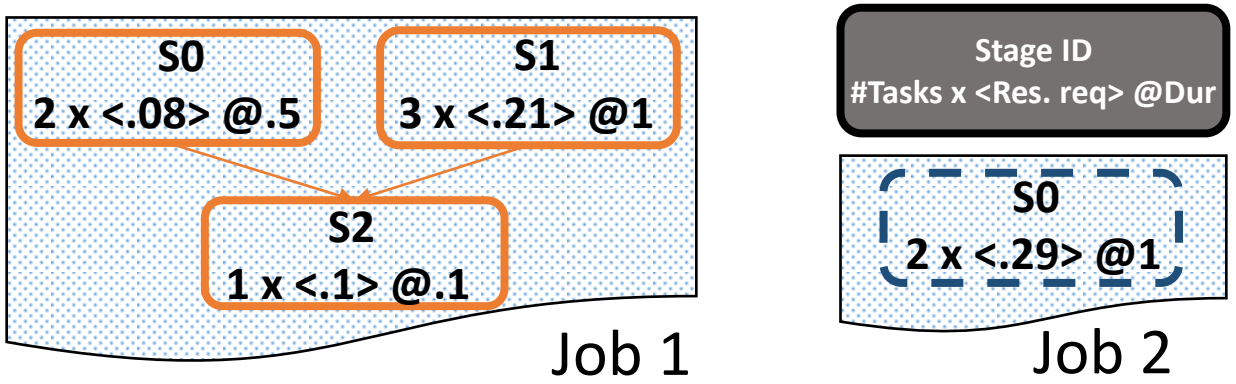
Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*



Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

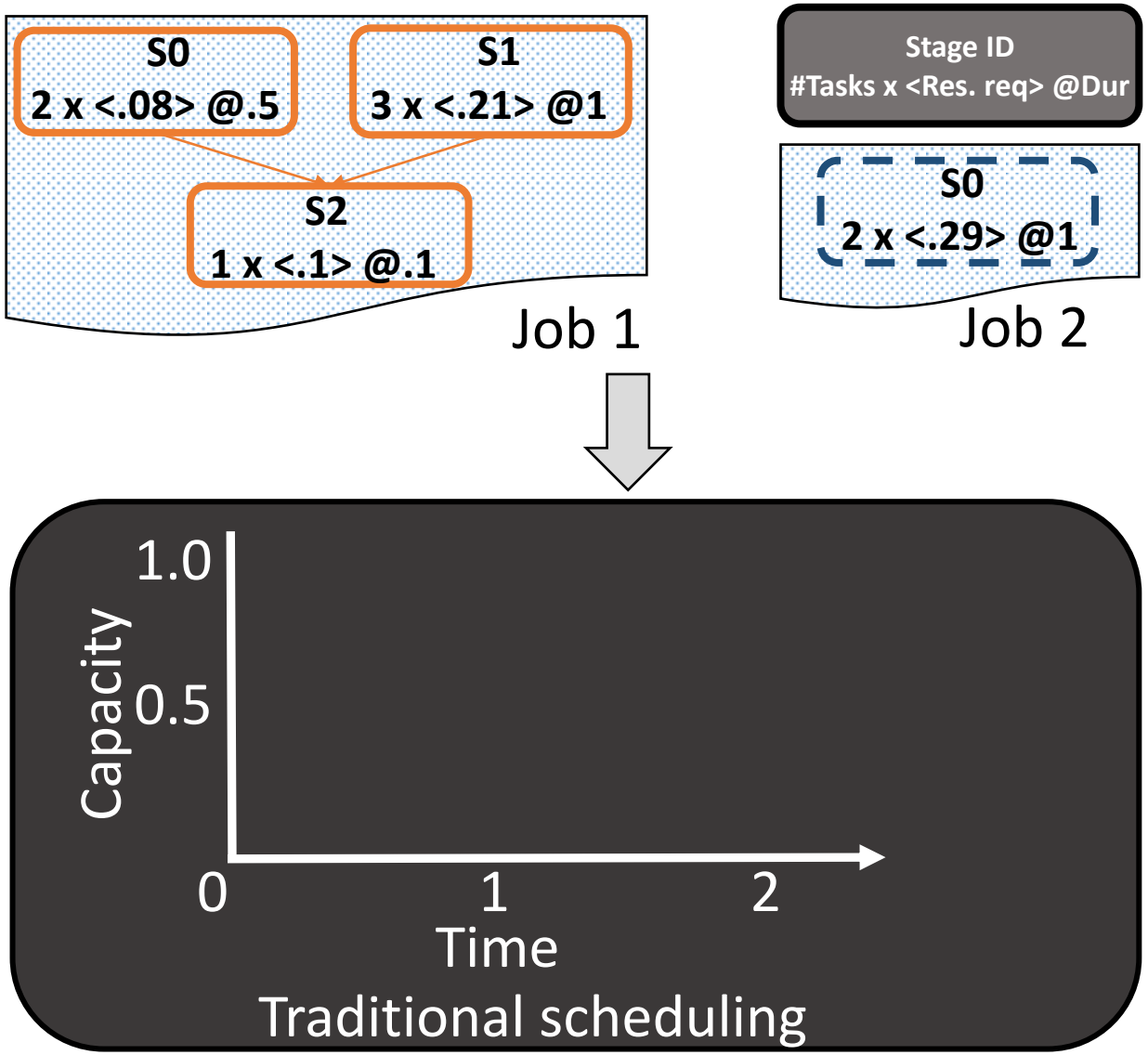


Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

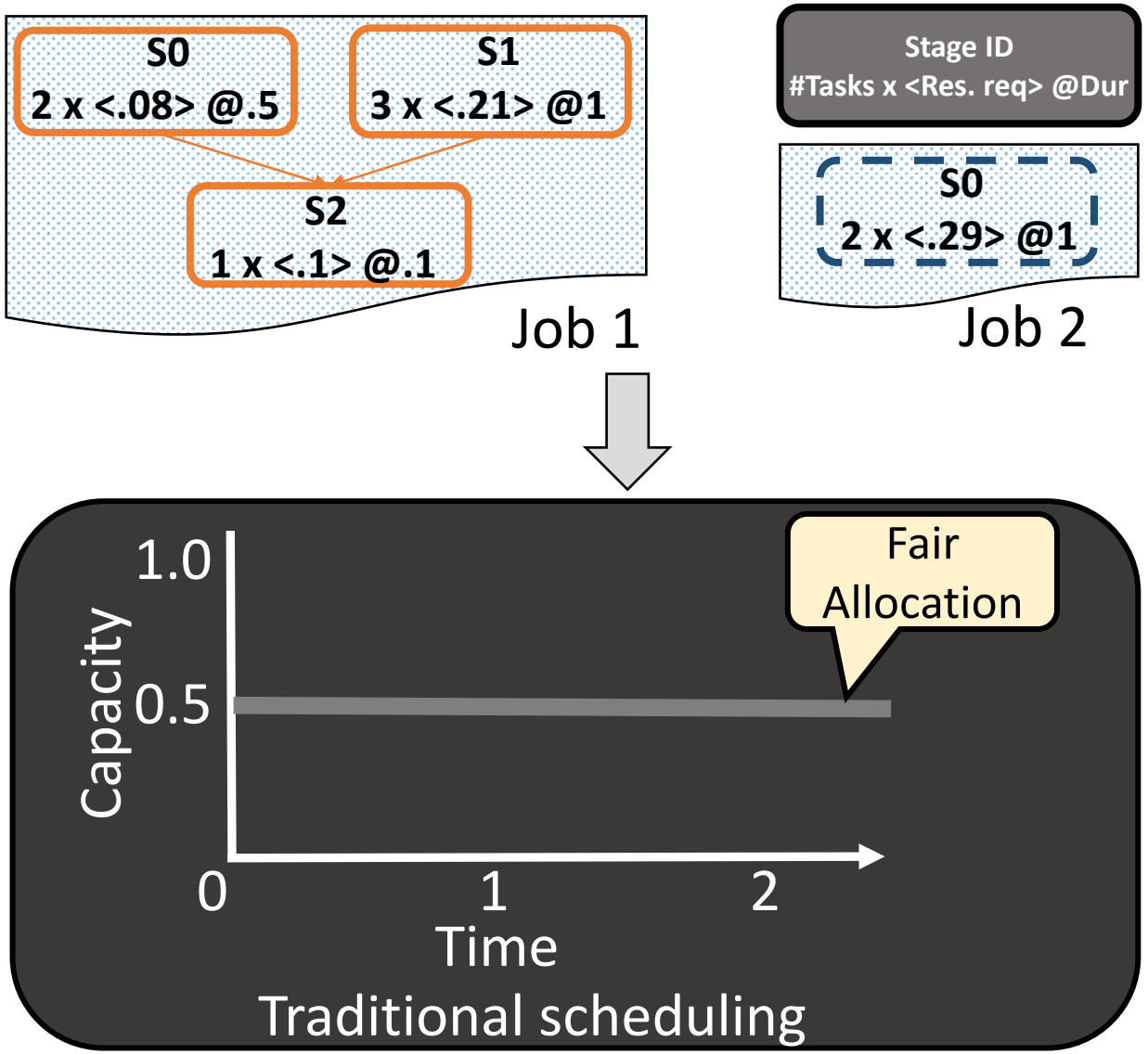


Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

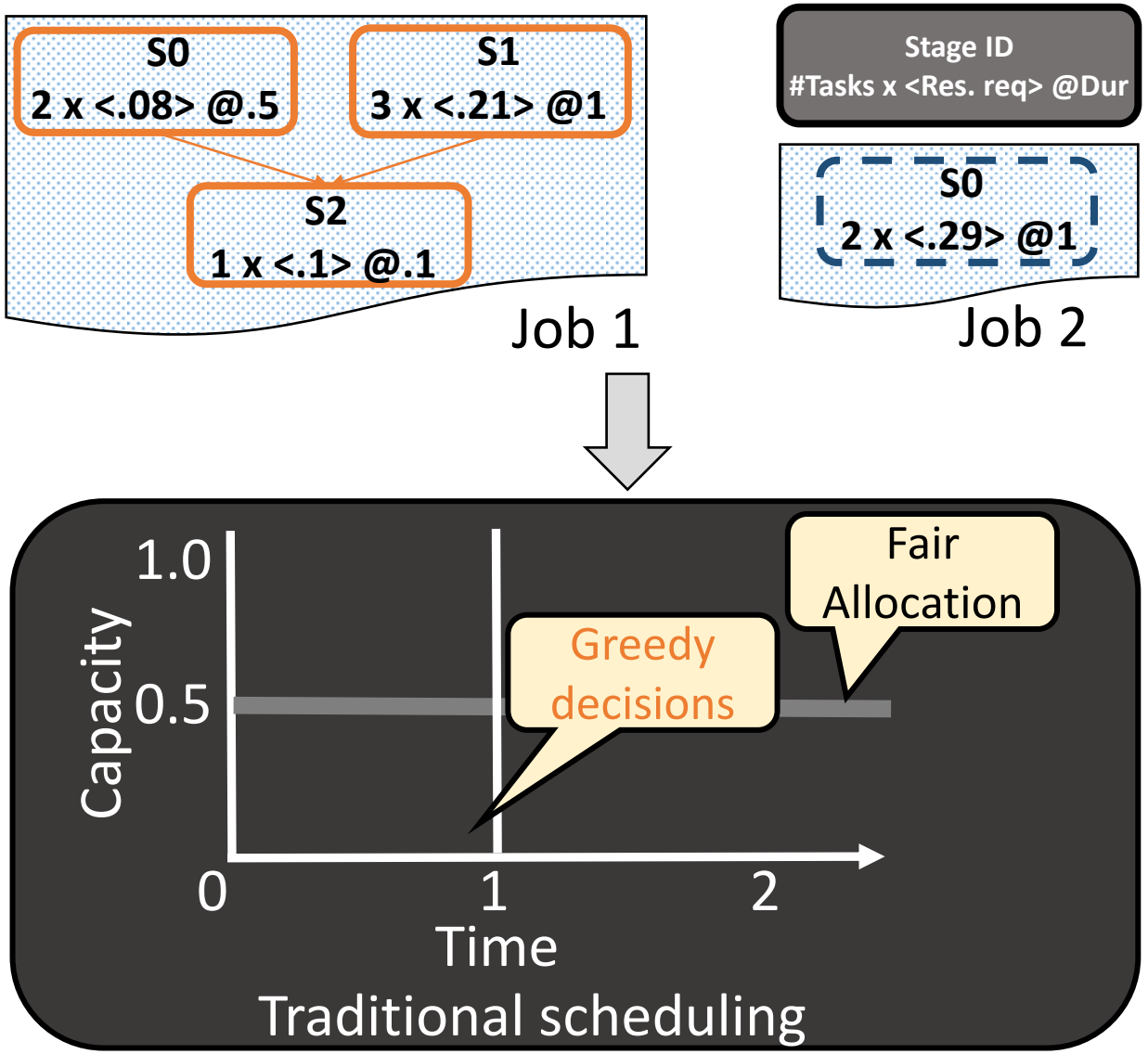


Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

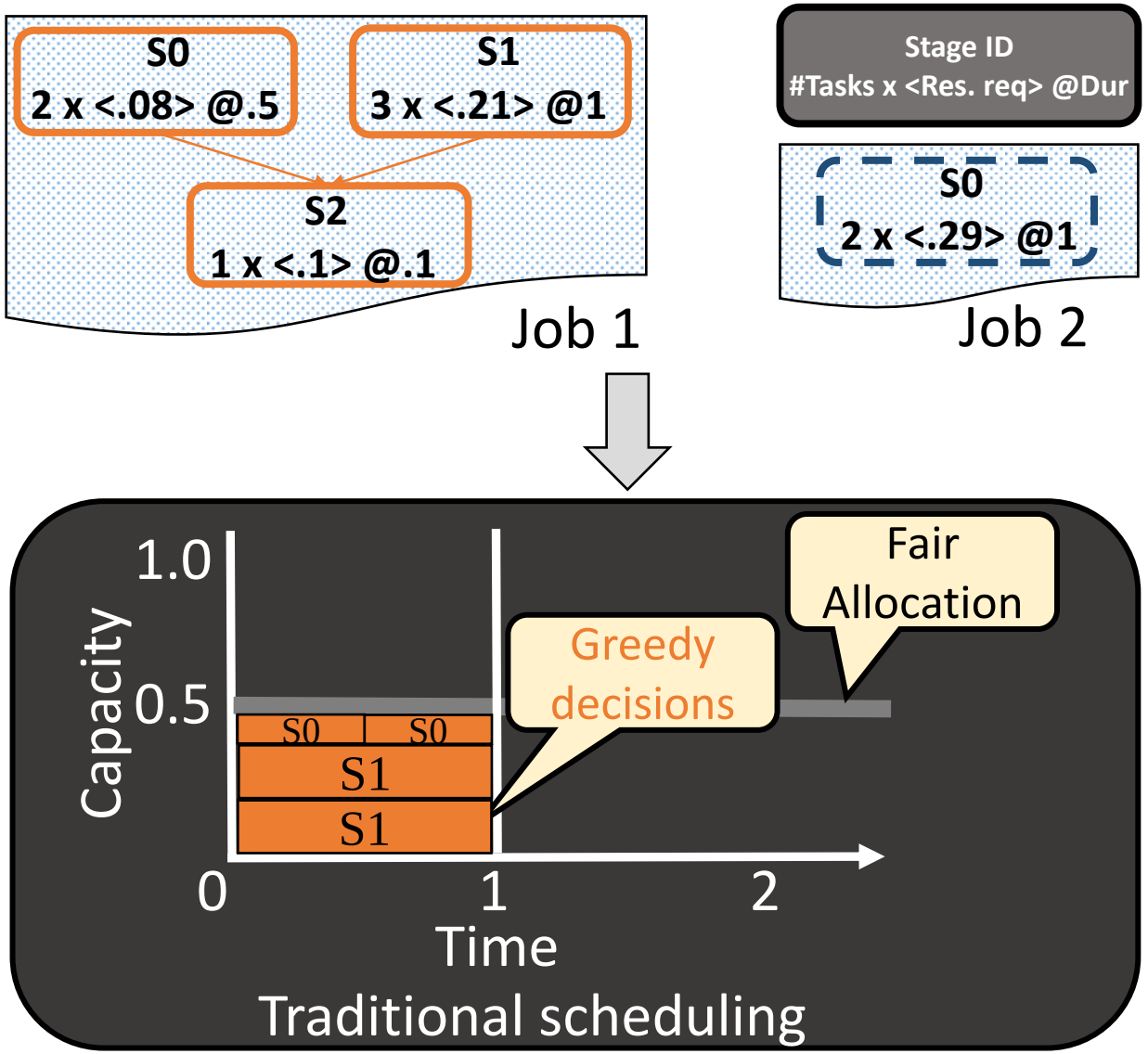


Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

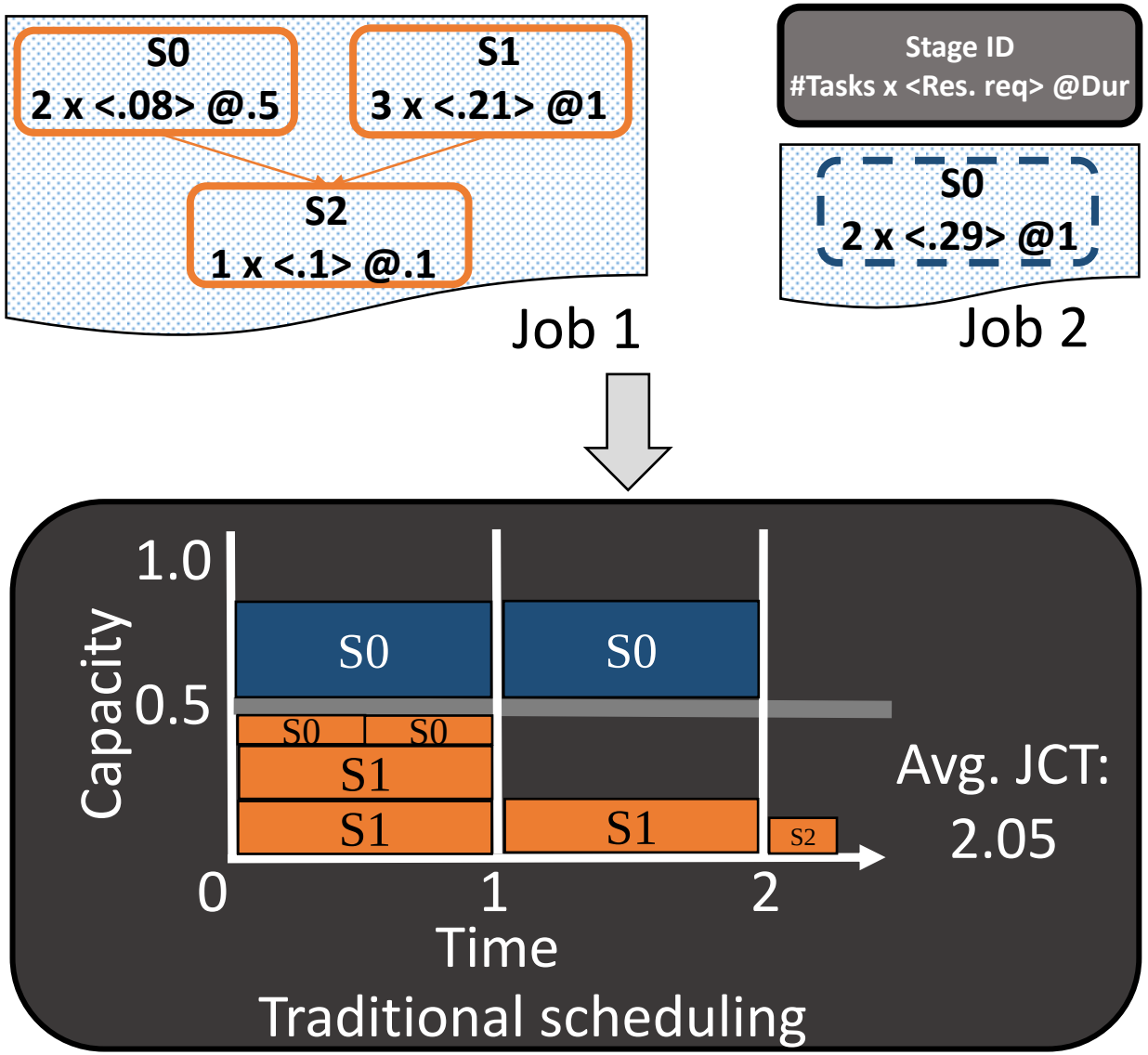


Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*



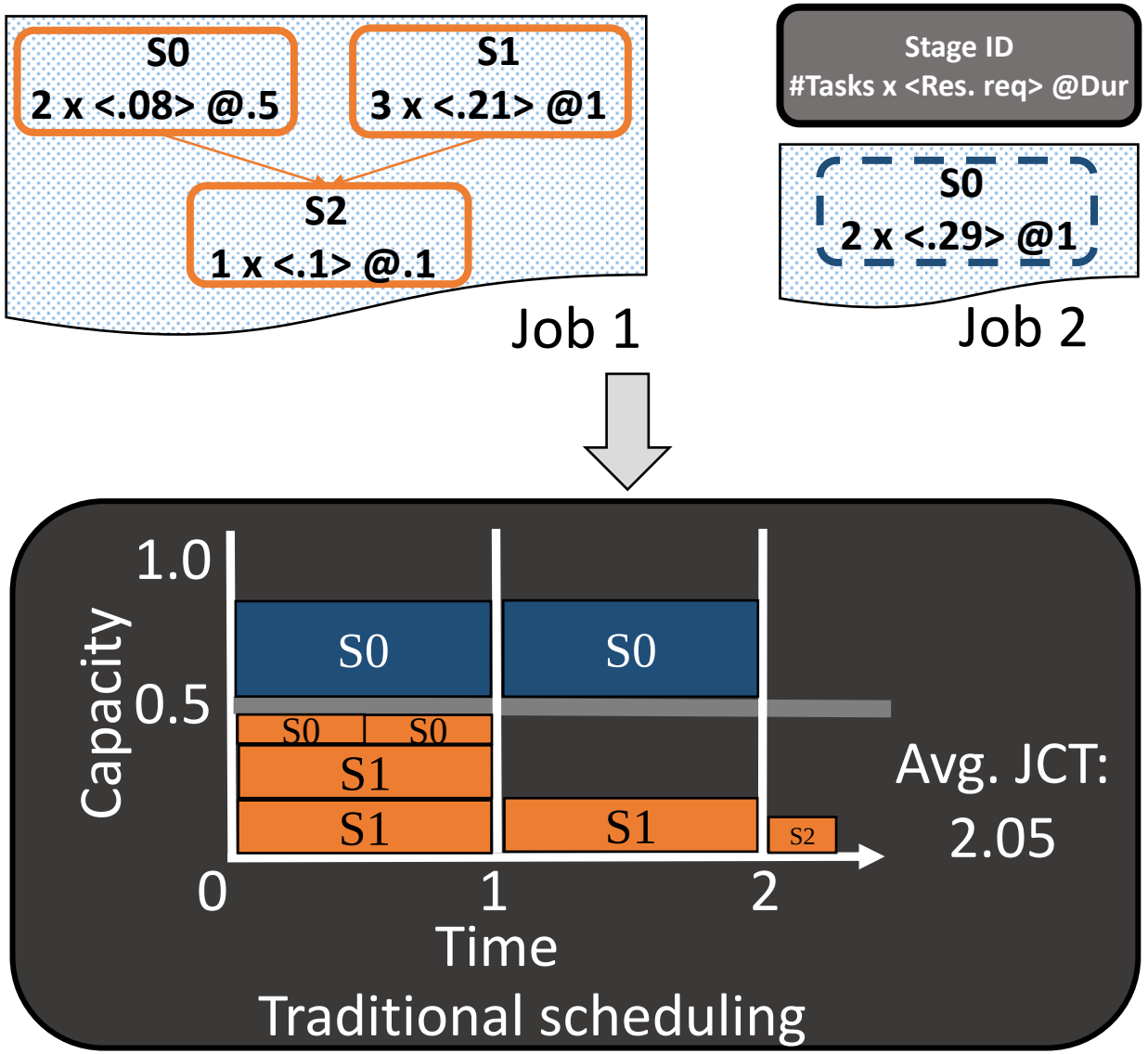
Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

Users care less for instantaneous fairness



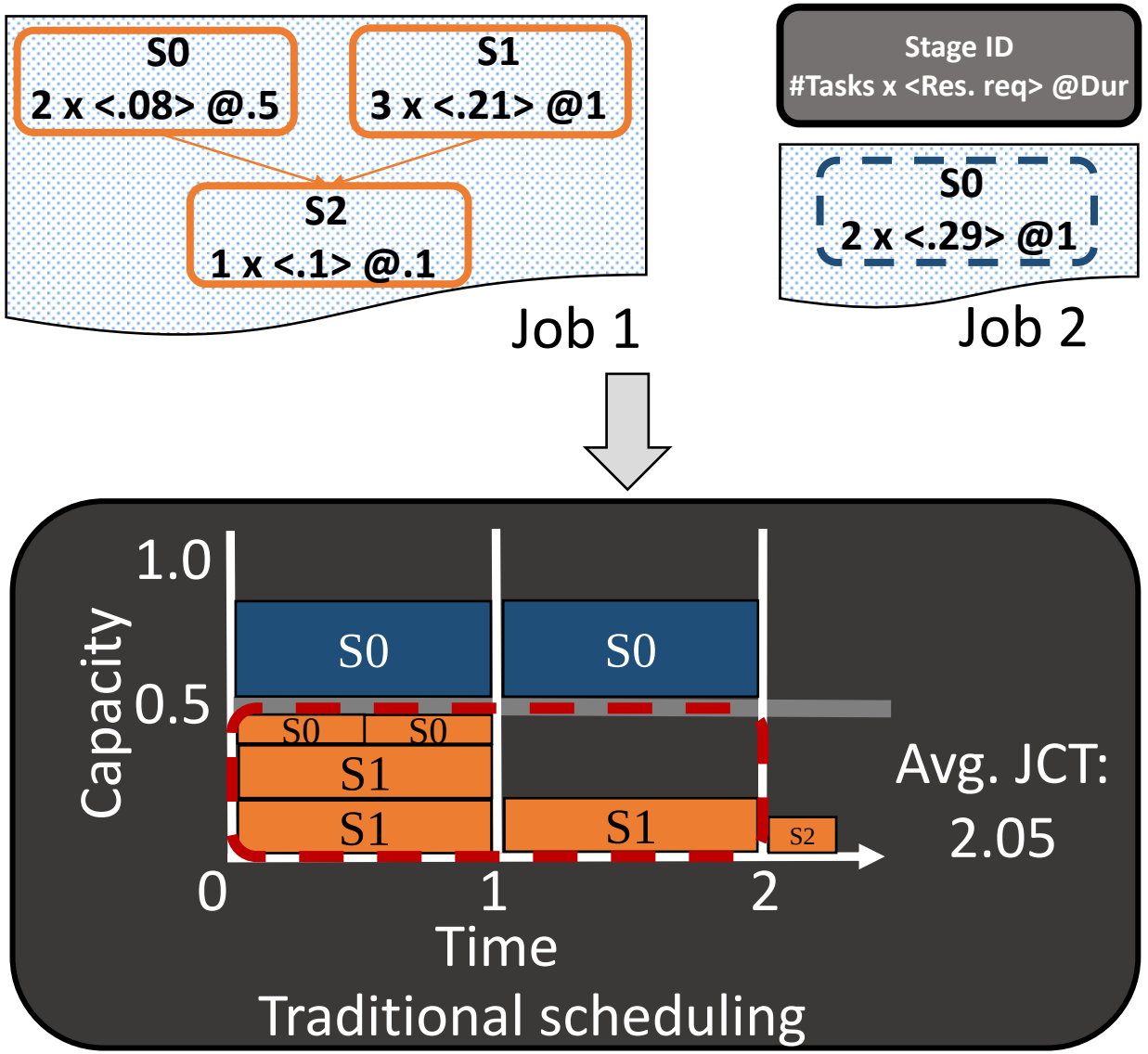
Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

Users care less for instantaneous fairness



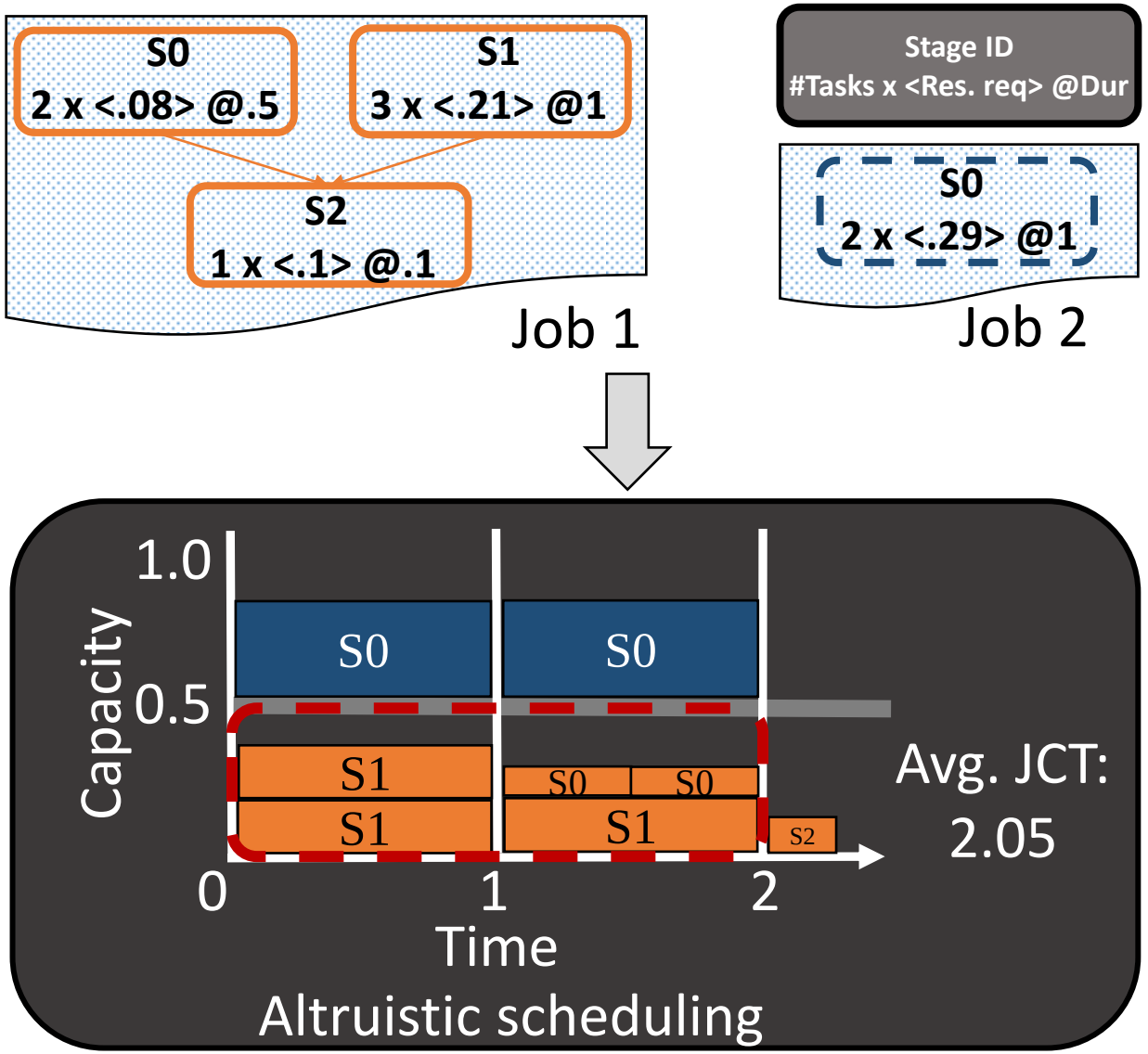
Assumption:

- Know tasks demands and durations

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

Users care less for instantaneous fairness



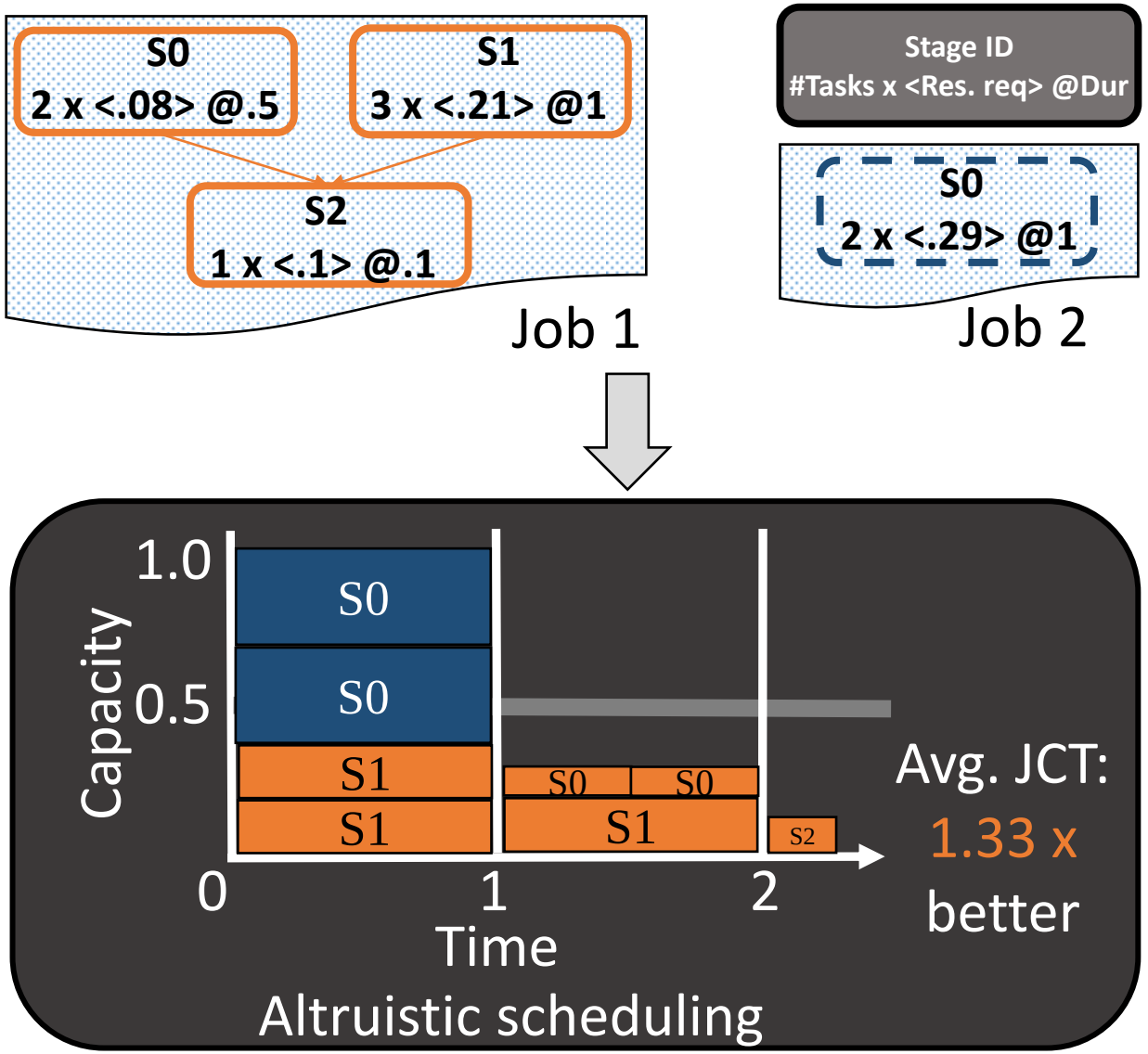
Leftover: donated unnecessary resources

Altruism: an action to contribute leftover resources

Key observation #1

Modern cluster schedulers focus on *instantaneous fairness* and force *short-term optimizations*

Users care less for instantaneous fairness



Leftover: donated unnecessary resources

Altruism: an action to contribute leftover resources

Key observation #2

Jobs in data analytics clusters have ample *opportunities for altruism*

Key observation #2

Jobs in data analytics clusters have ample *opportunities for altruism*

What increases opportunities?

Key observation #2

Jobs in data analytics clusters have ample *opportunities for altruism*

What increases opportunities?

- Complex DAG structures
- Longer DAGs

Key observation #2

Jobs in data analytics clusters have ample *opportunities for altruism*

What increases opportunities?

- Complex DAG structures
- Longer DAGs

How much opportunities?

Key observation #2

Jobs in data analytics clusters have ample *opportunities for altruism*

What increases opportunities?

- Complex DAG structures
- Longer DAGs

How much opportunities?

50% of the time at least 20% of the resources can be used as leftover

Carbyne

Altruistic multi-resource scheduling technique

Carbyne

Altruistic multi-resource scheduling technique

Leftover

Carbyne

Altruistic multi-resource scheduling technique

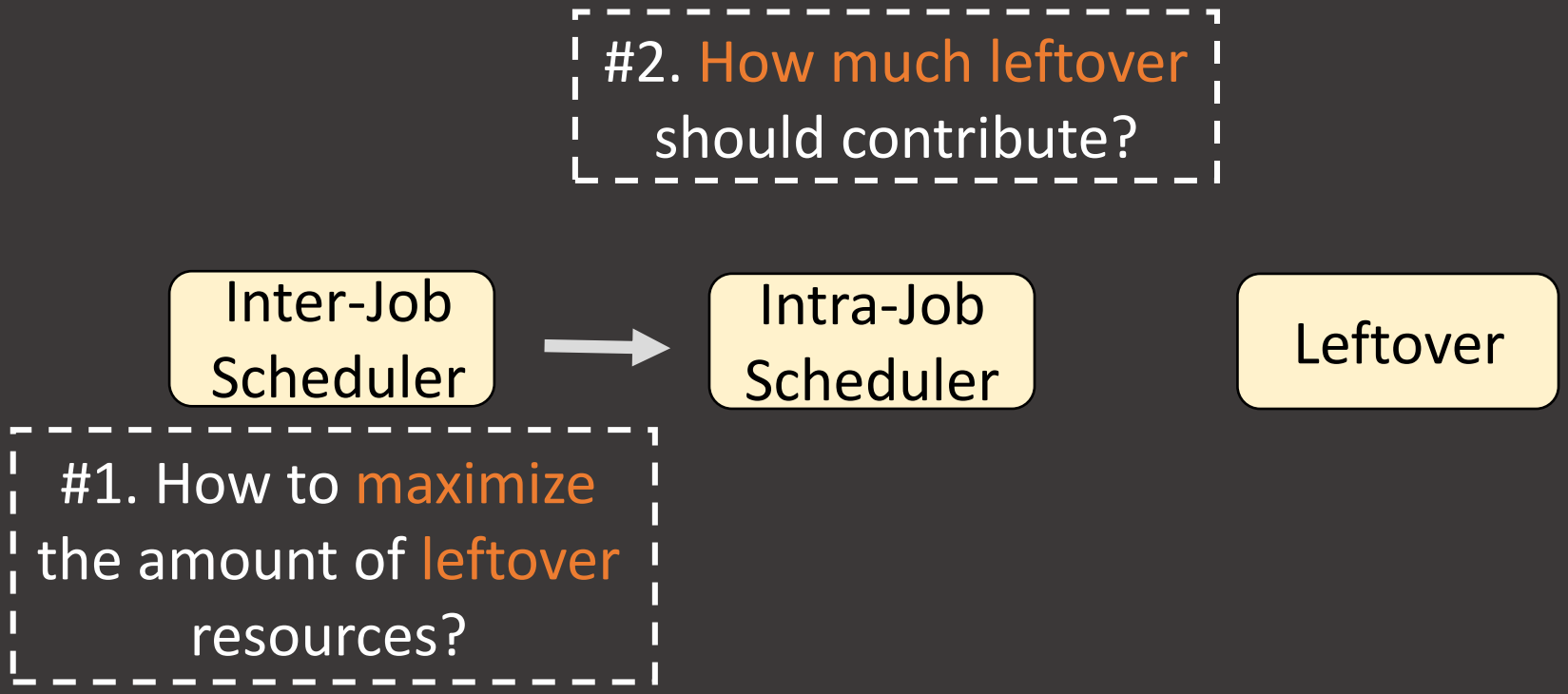
Inter-Job Scheduler

#1. How to maximize the amount of leftover resources?

Leftover

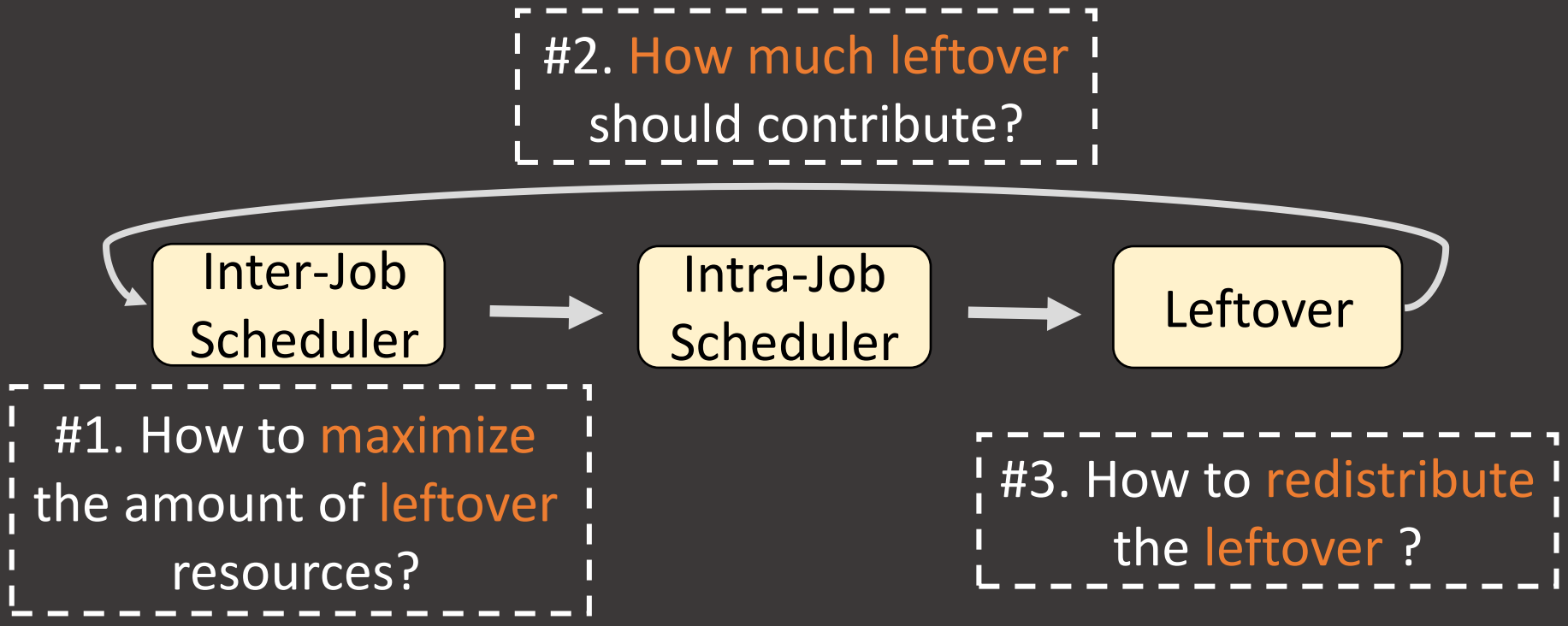
Carbyne

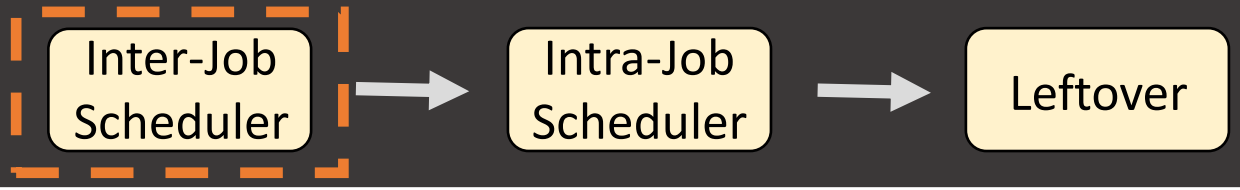
Altruistic multi-resource scheduling technique



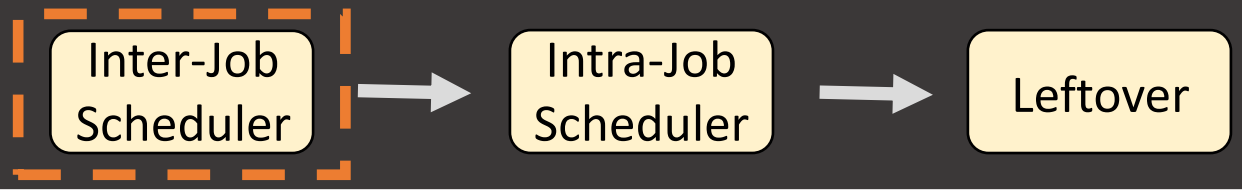
Carbyne

Altruistic multi-resource scheduling technique



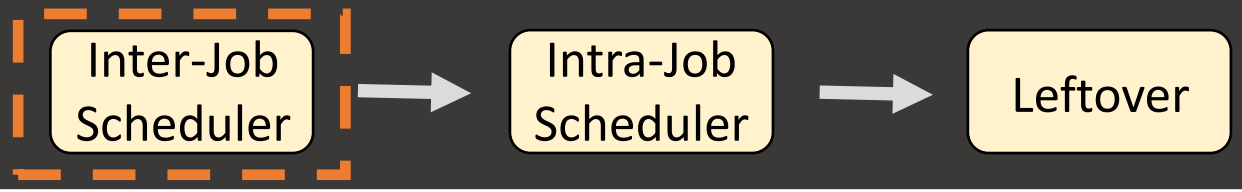


Maximize the amount of leftover resources



Maximize the amount of leftover resources

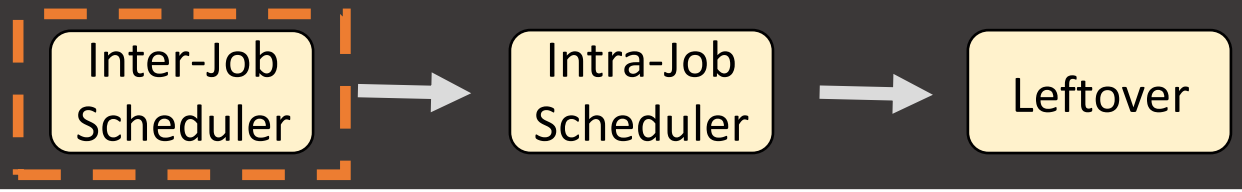
Instantaneous fairness elongates job completion time the most and increases altruism opportunities



Maximize the amount of leftover resources

Instantaneous fairness elongates job completion time the most and increases altruism opportunities

Carbyne uses DRF for inter-job scheduling



Maximize the amount of leftover resources

Instantaneous fairness elongates job completion time the most and increases altruism opportunities

Carbyne uses DRF for inter-job scheduling

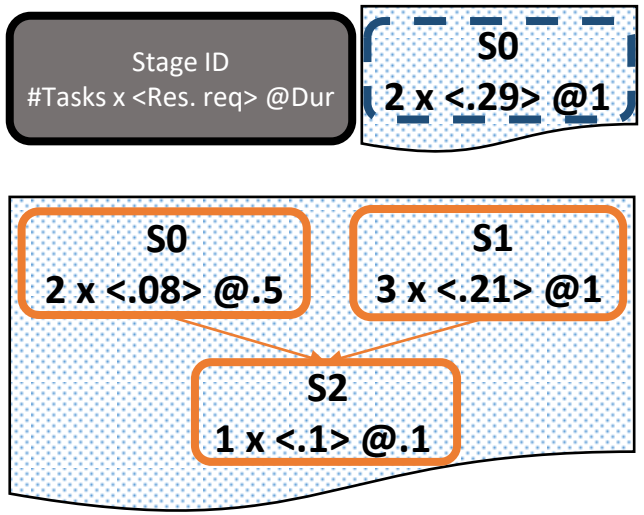
- Any fair scheduler technique can be used

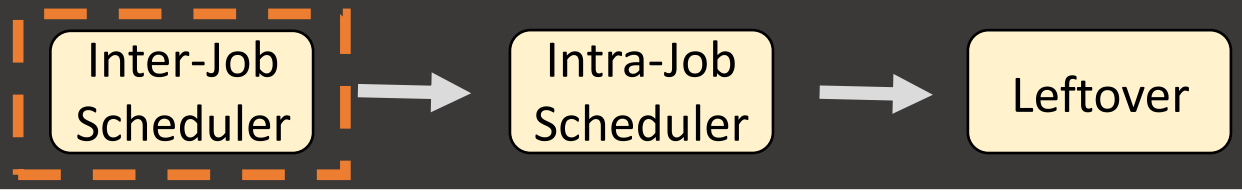
Maximize the amount of leftover resources

Instantaneous fairness elongates job completion time the most and increases altruism opportunities

Carbyne uses DRF for inter-job scheduling

- Any fair scheduler technique can be used



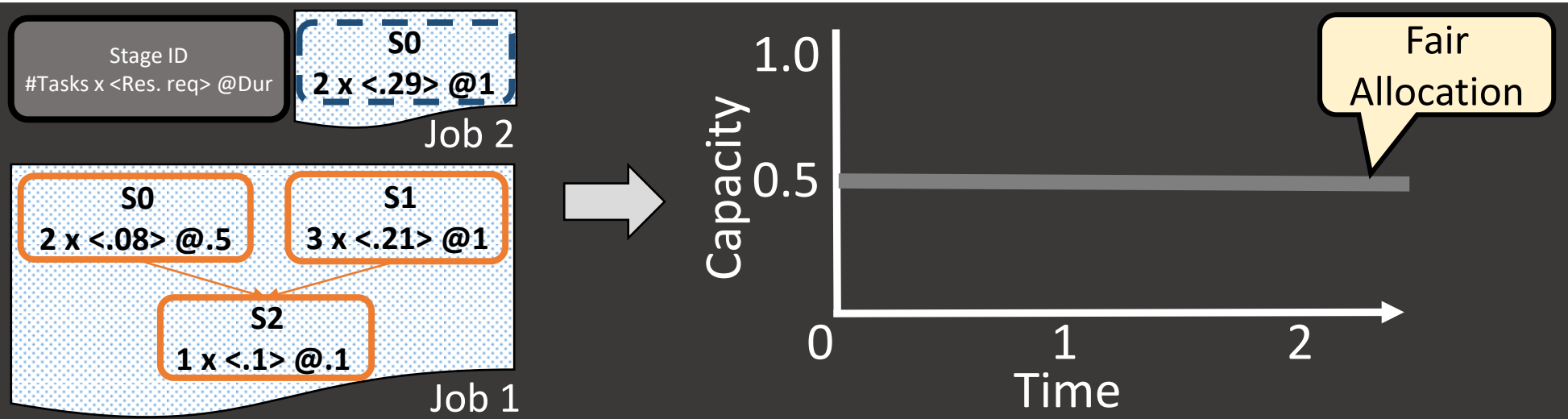


Maximize the amount of leftover resources

Instantaneous fairness elongates job completion time the most and increases altruism opportunities

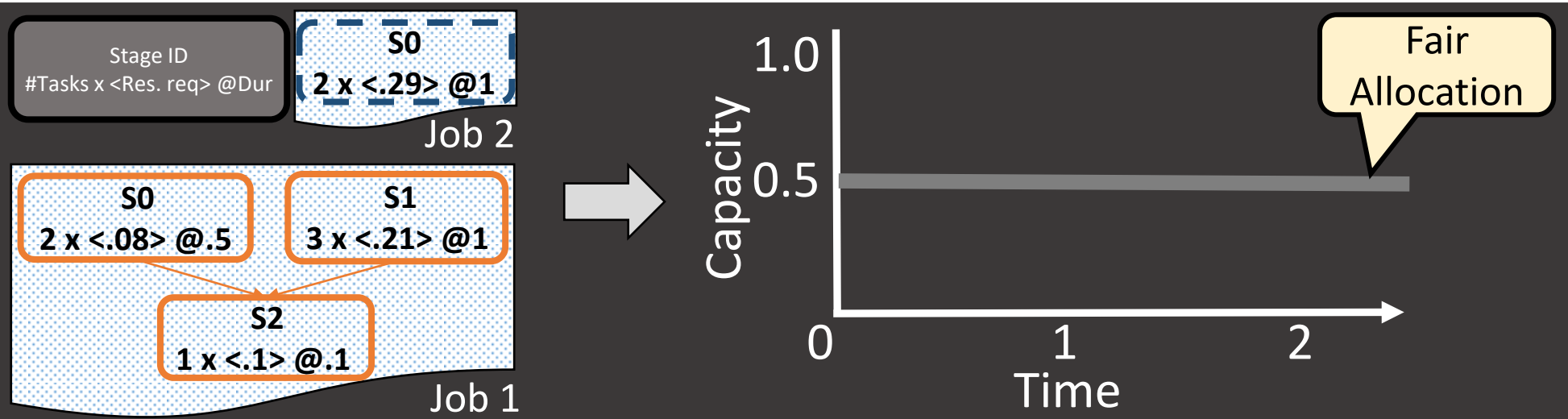
Carbyne uses DRF for inter-job scheduling

- Any fair scheduler technique can be used





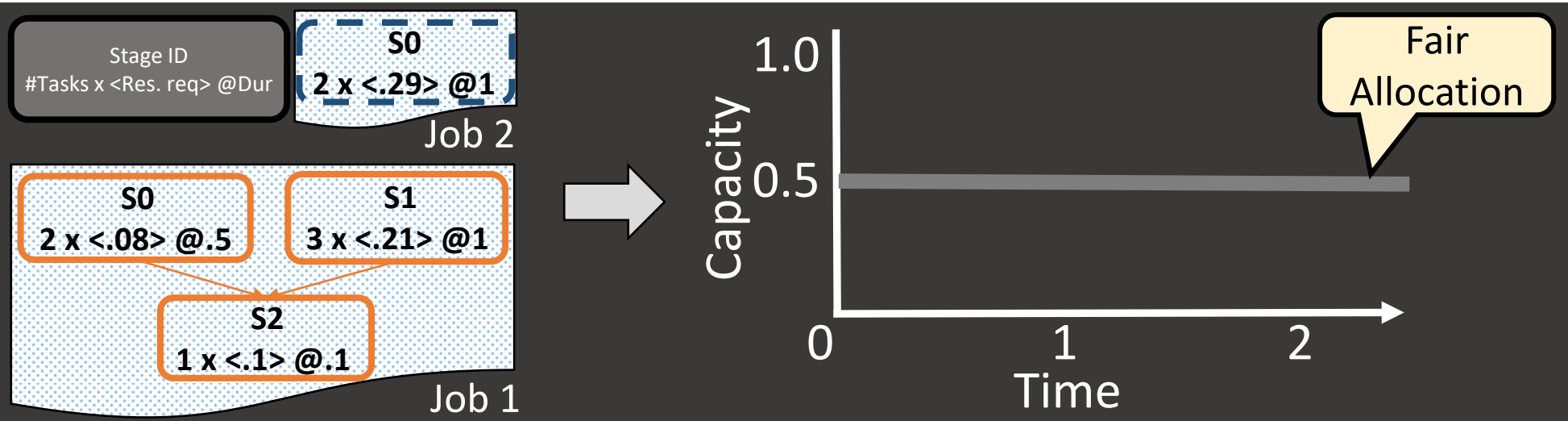
How much leftover to contribute?





How much leftover to contribute?

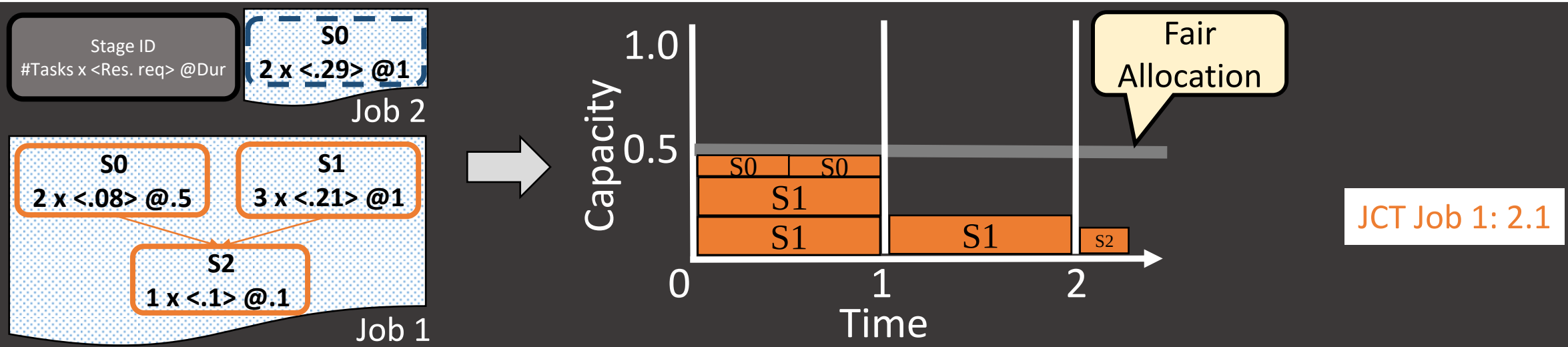
Traditional scheduling to compute expected completion time





How much leftover to contribute?

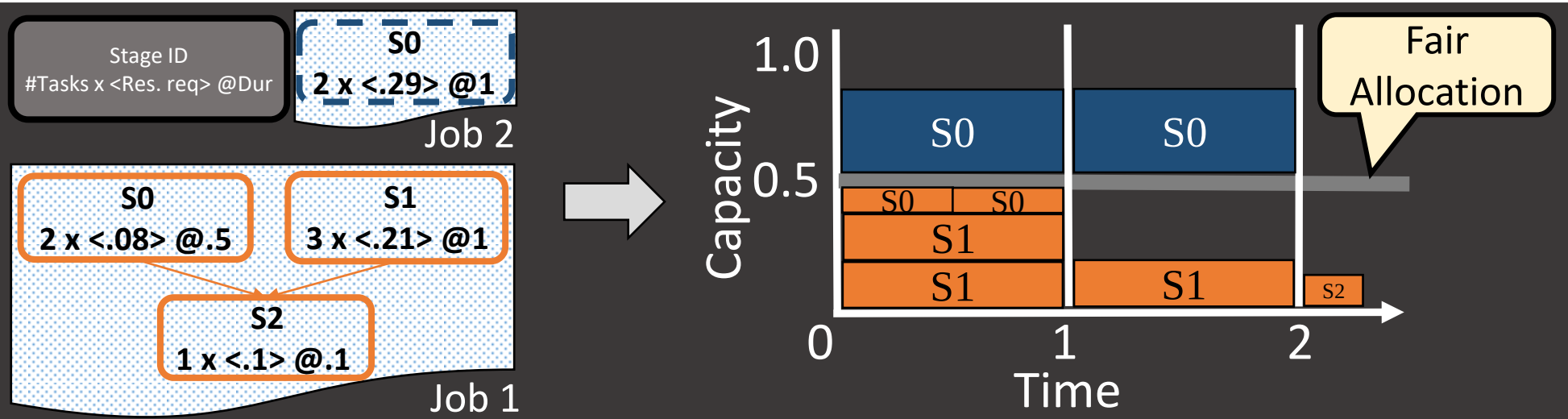
Traditional scheduling to compute expected completion time





How much leftover to contribute?

Traditional scheduling to compute expected completion time



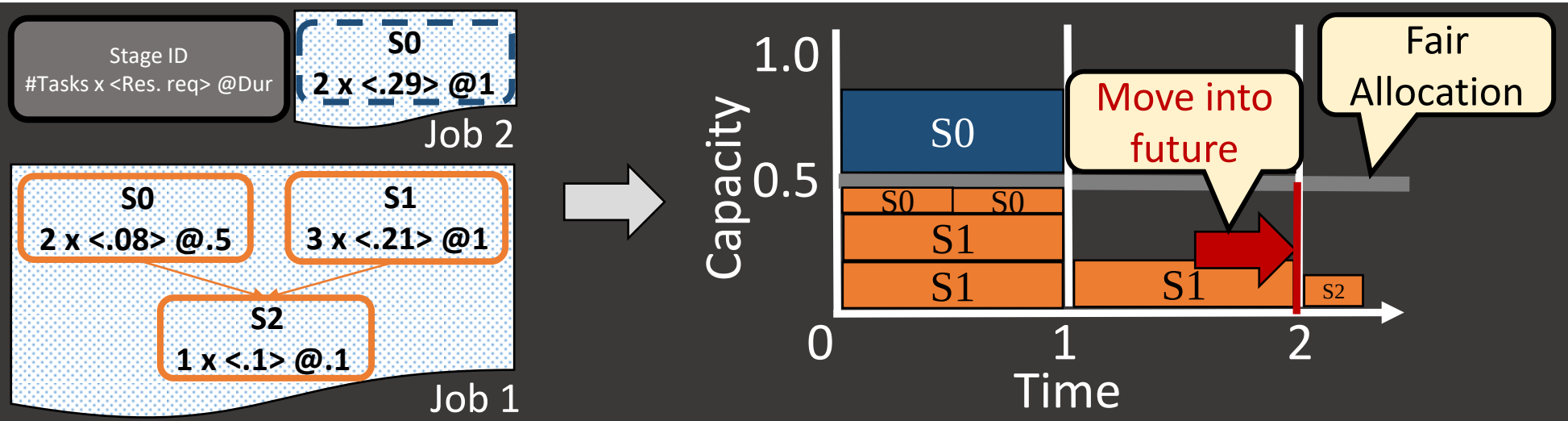
JCT Job 1: 2.1

JCT Job 2: 2.0



How much leftover to contribute?

Traditional scheduling to compute expected completion time



JCT Job 1: 2.1

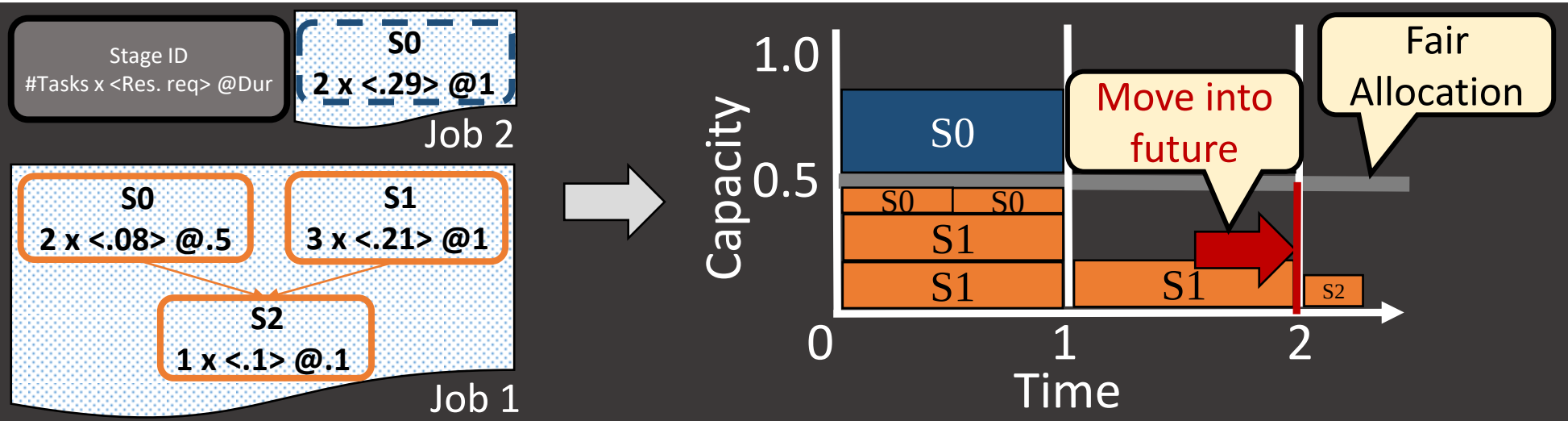
JCT Job 2: 2.0



How much leftover to contribute?

Traditional scheduling to compute expected completion time

Scheduling in the future from finish to current time



JCT Job 1: 2.1

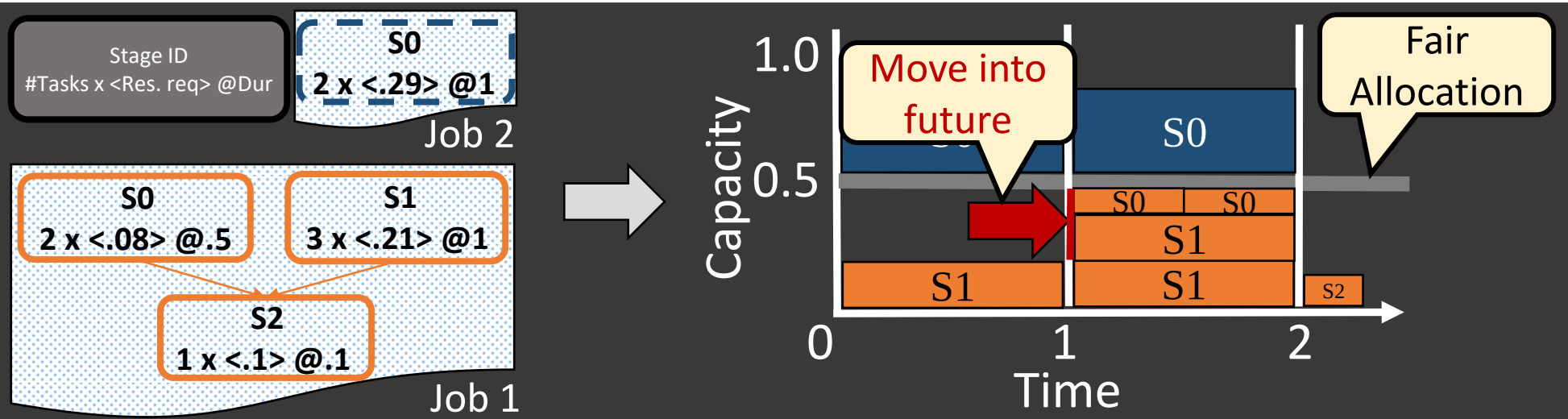
JCT Job 2: 2.0



How much leftover to contribute?

Traditional scheduling to compute expected completion time

Scheduling in the future from finish to current time



JCT Job 1: 2.1

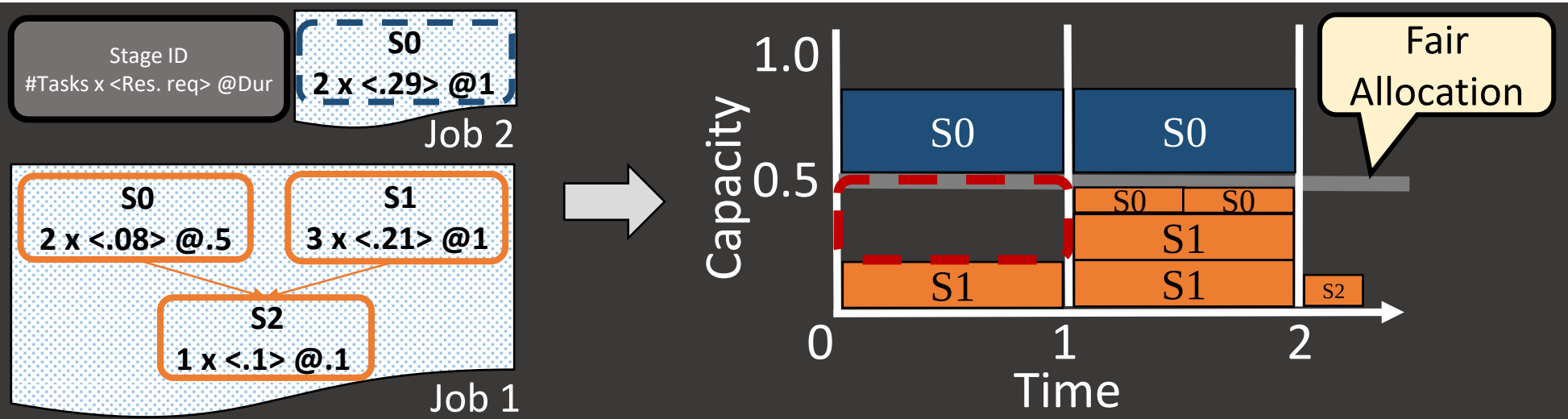
JCT Job 2: 2.0



How much leftover to contribute?

Traditional scheduling to compute expected completion time

Scheduling in the future from finish to current time



JCT Job 1: 2.1

JCT Job 2: 2.0

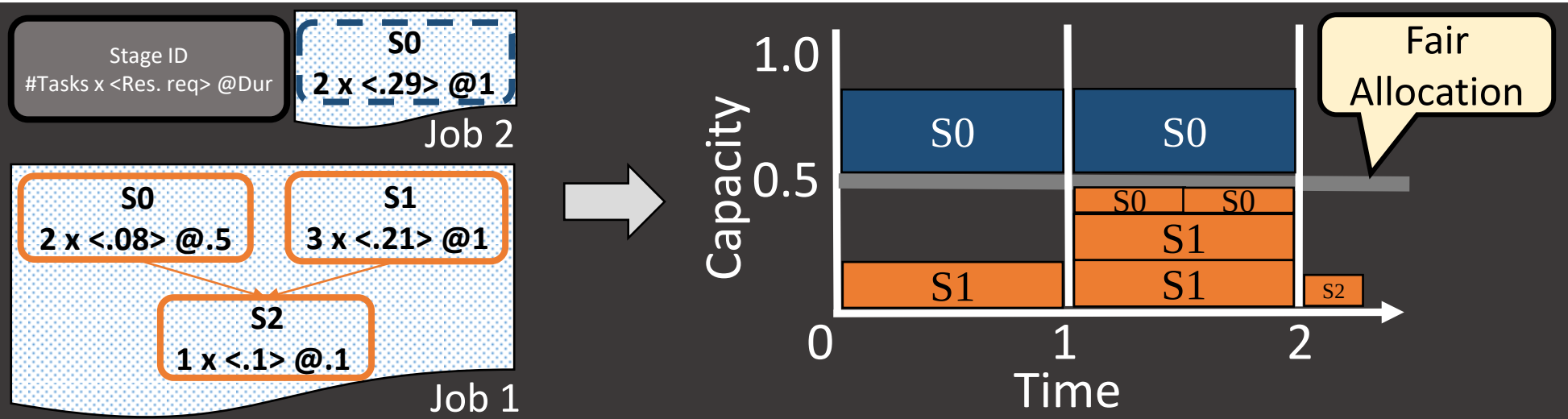


How much leftover to contribute?

Traditional scheduling to compute expected completion time

Scheduling in the future from finish to current time

Donate leftover resources through altruism



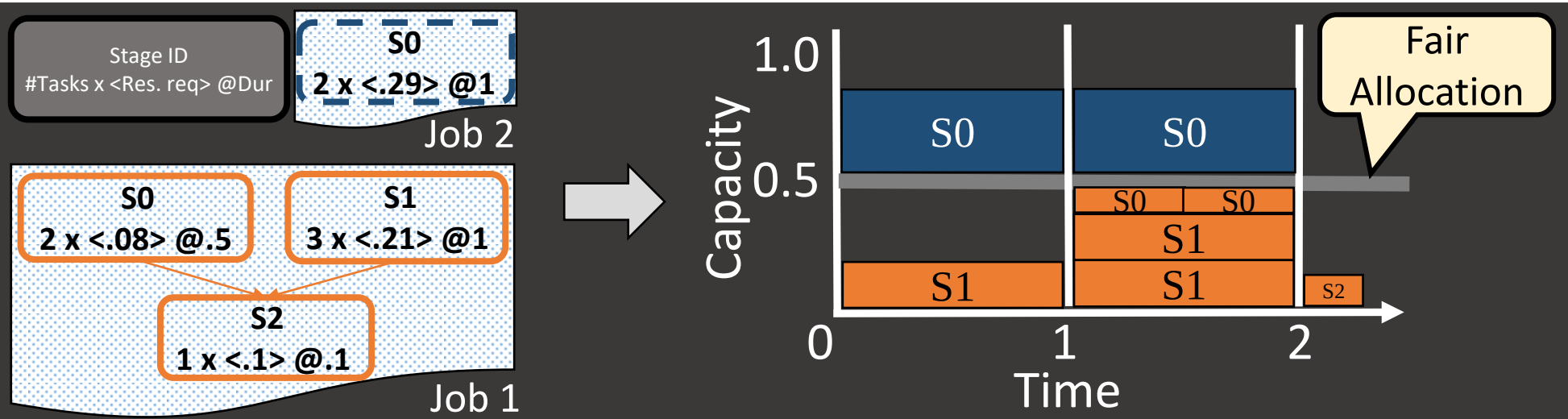


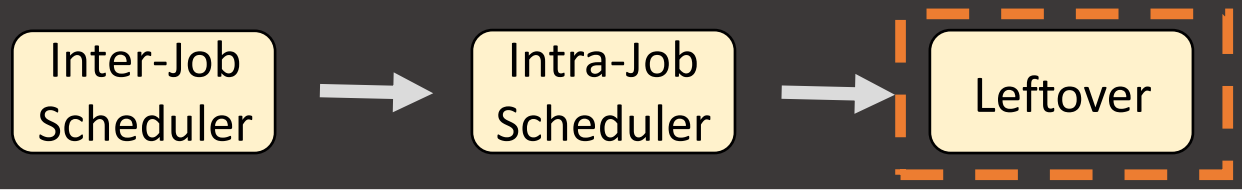
How much leftover to contribute?

Traditional scheduling to compute expected completion time

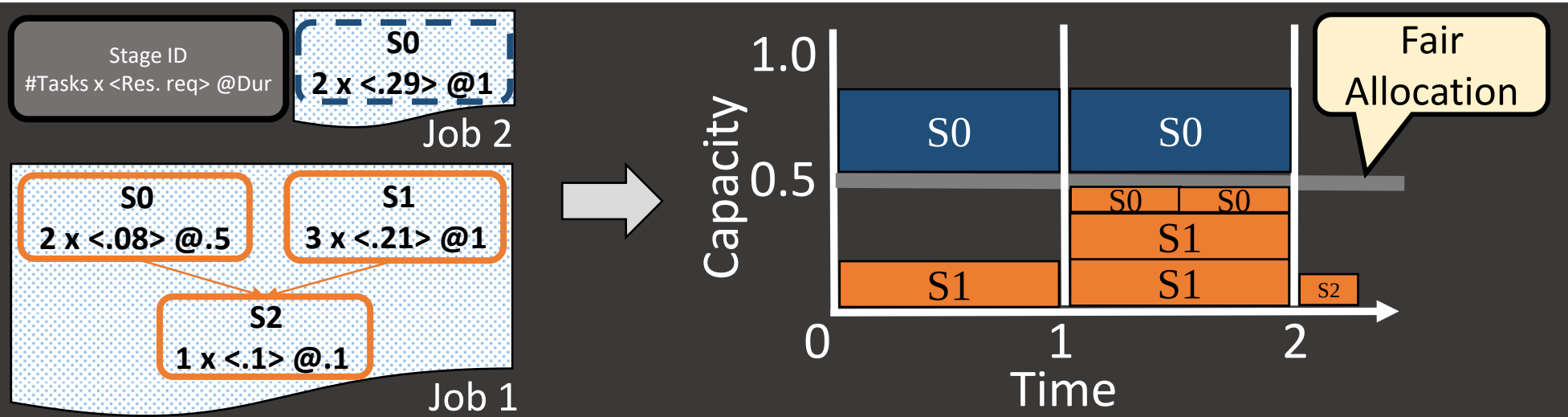
Scheduling in the future from finish to current time

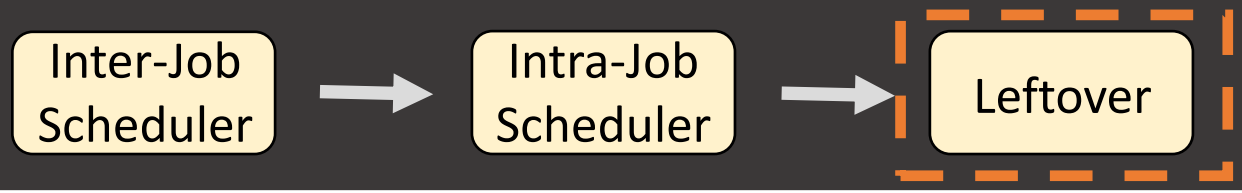
Donate leftover resources through altruism





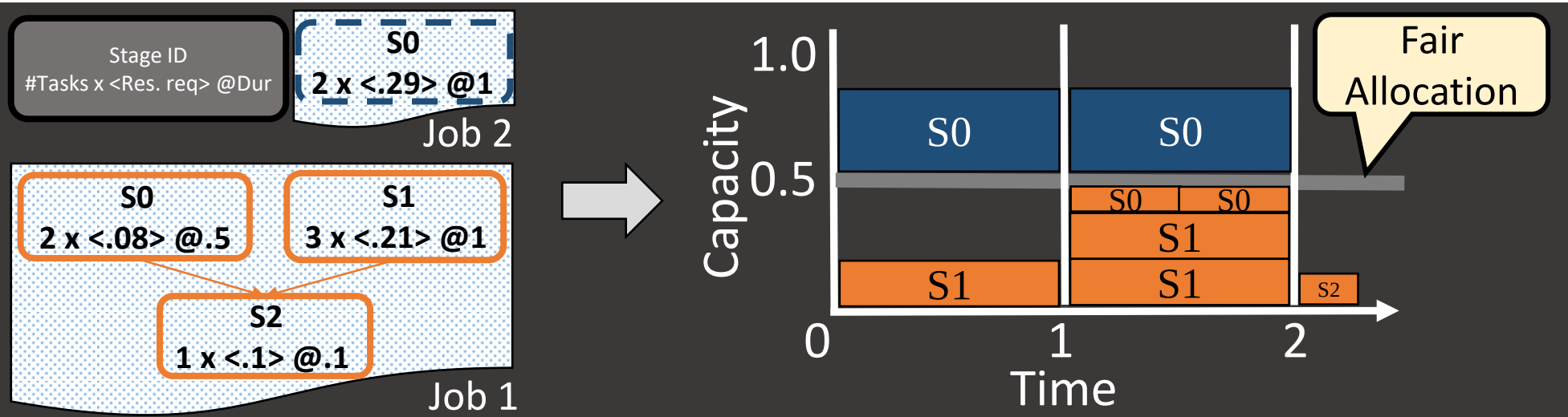
How to redistribute the leftover resources?

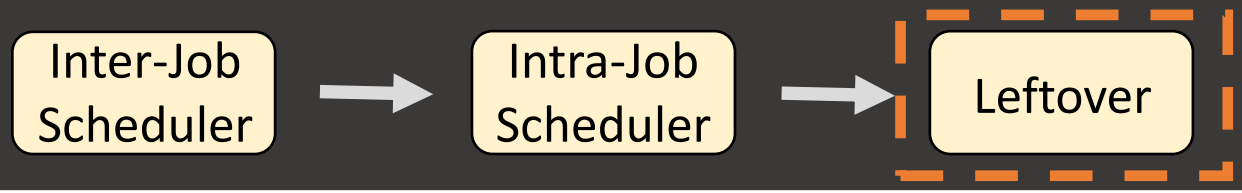




How to redistribute the leftover resources?

Goal 1: Improve average JCT



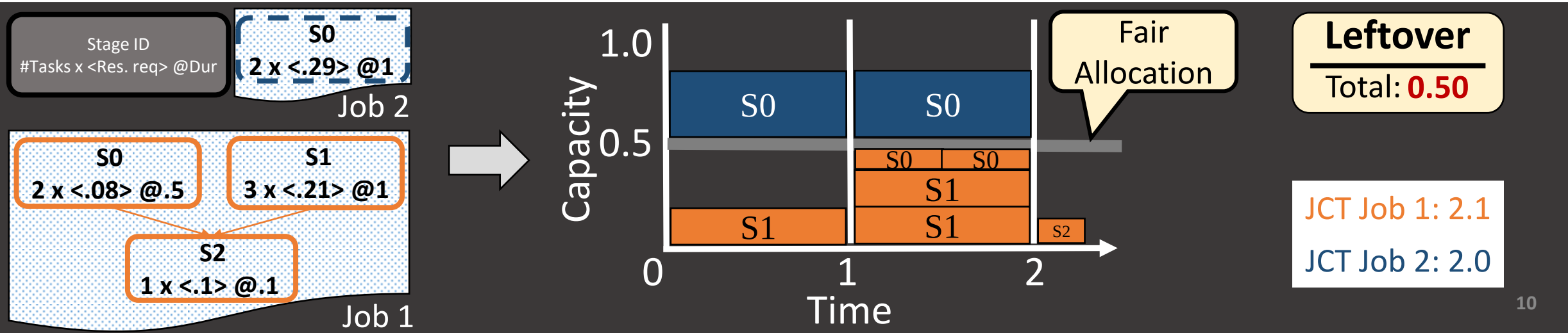


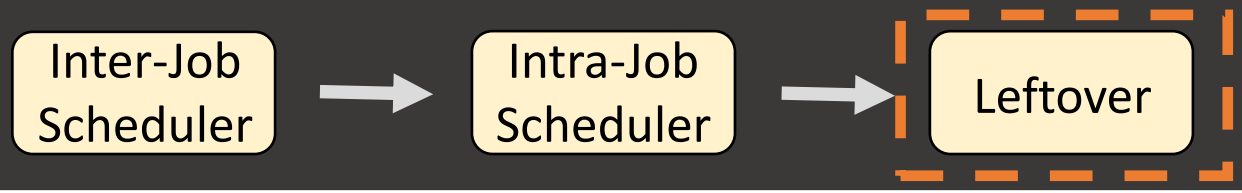
How to redistribute the leftover resources?

Goal 1: Improve average JCT

Goal 2: Maximize efficiency

Goals 1 and 2 can be interchanged





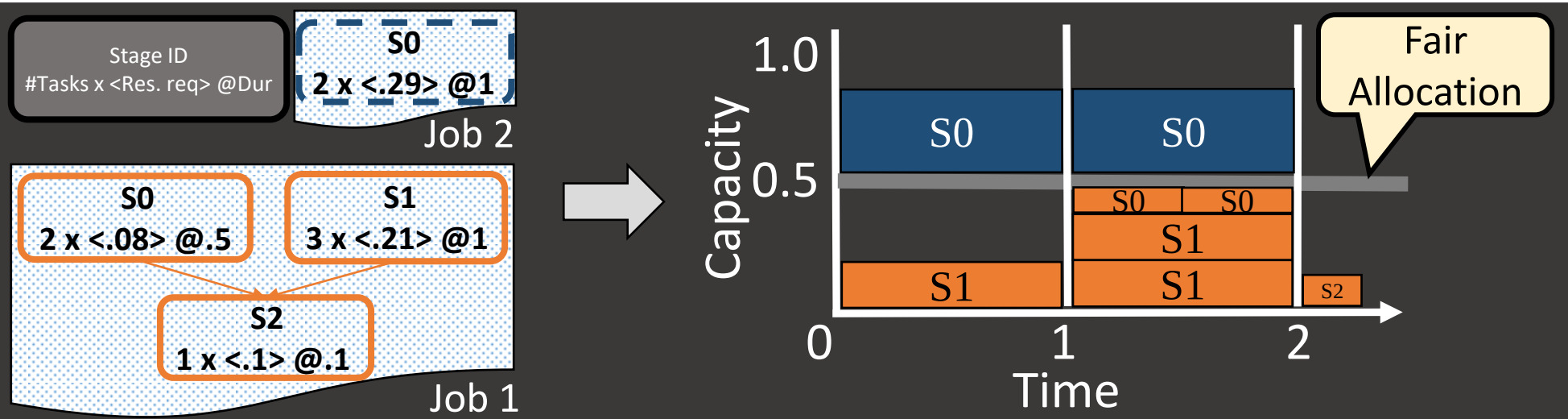
How to redistribute the leftover resources?

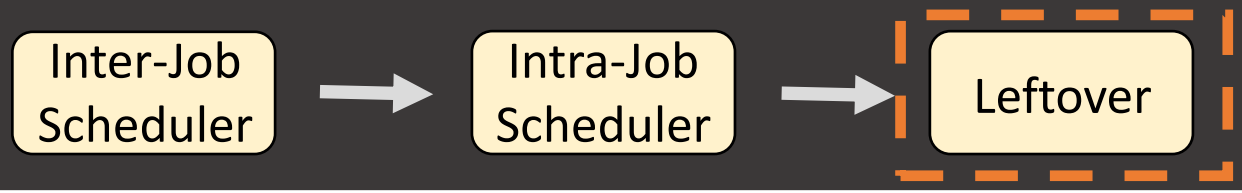
Goal 1: Improve average JCT

- *Schedule jobs closest to completion time first*

Goal 2: Maximize efficiency

Goals 1 and 2 can be interchanged





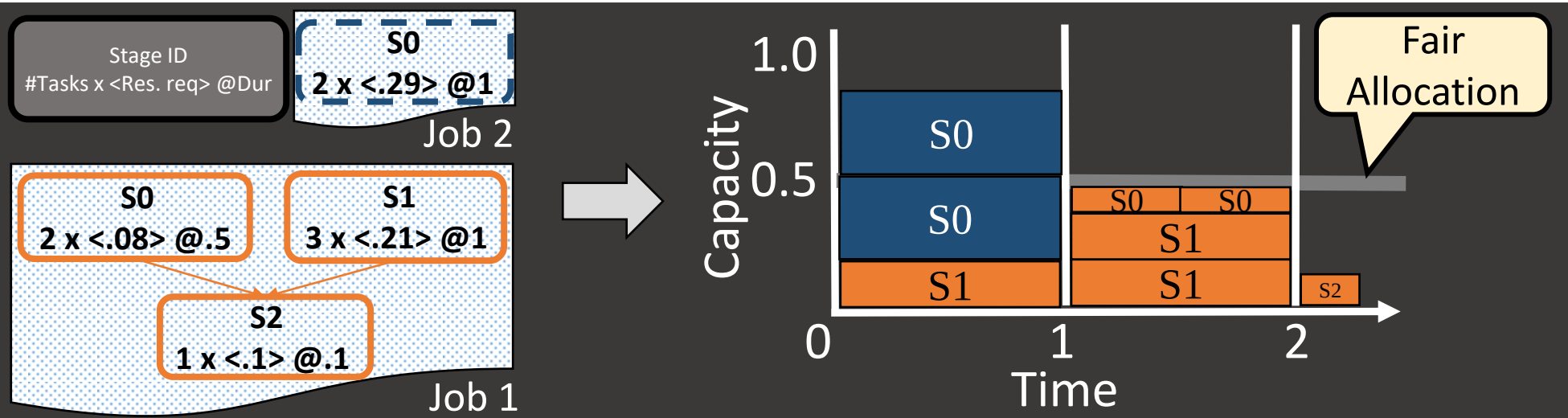
How to redistribute the leftover resources?

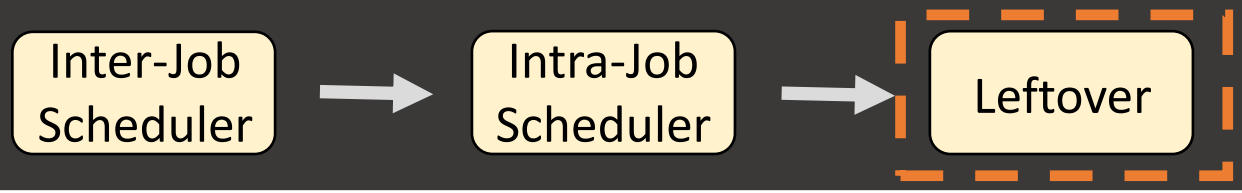
Goal 1: Improve average JCT

- *Schedule jobs closest to completion time first*

Goal 2: Maximize efficiency

Goals 1 and 2 can be interchanged





How to redistribute the leftover resources?

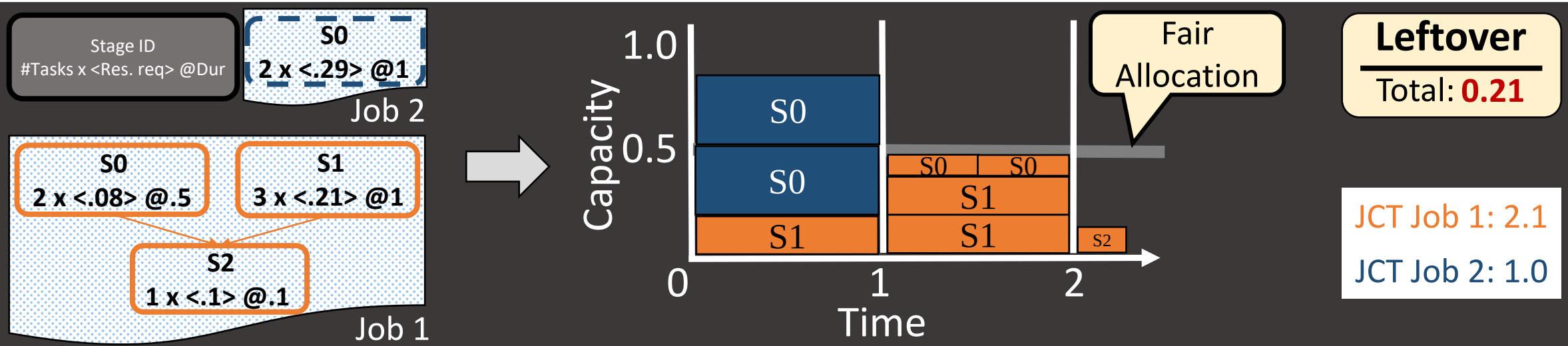
Goal 1: Improve average JCT

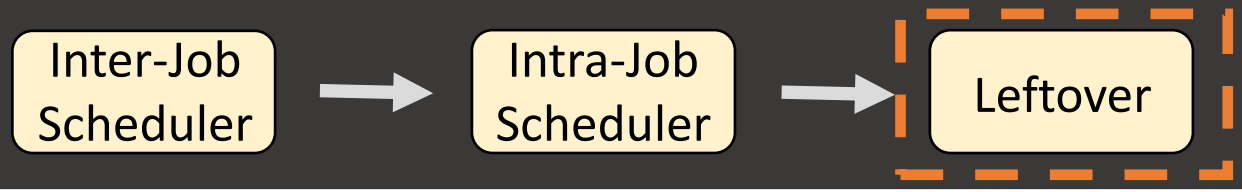
- *Schedule jobs closest to completion time first*

Goal 2: Maximize efficiency

- *Pack as many unscheduled tasks are possible*

Goals 1 and 2 can be interchanged





How to redistribute the leftover resources?

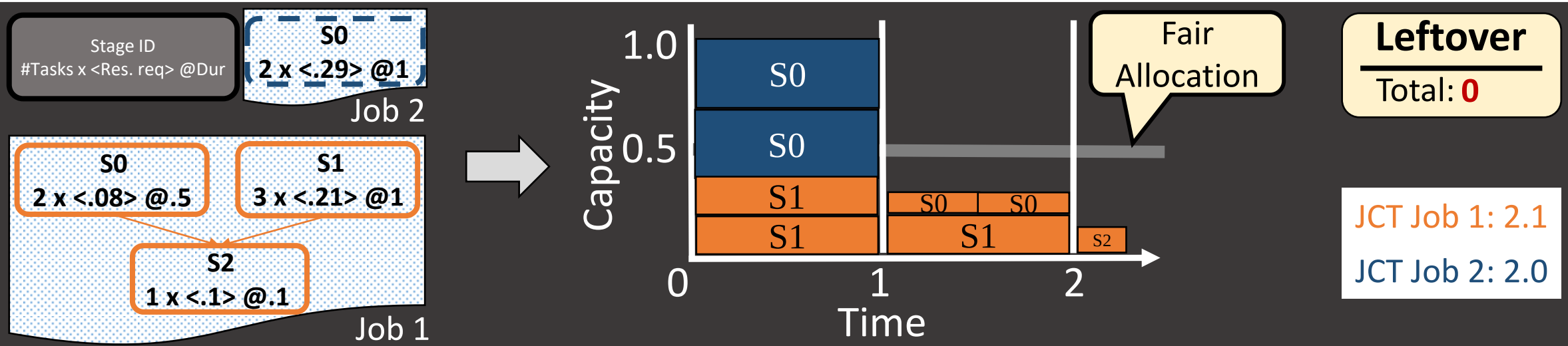
Goal 1: Improve average JCT

- *Schedule jobs closest to completion time first*

Goal 2: Maximize efficiency

- *Pack as many unscheduled tasks are possible*

Goals 1 and 2 can be interchanged



Putting it all together

We saw

Increase leftover via Inter-Job Scheduling

- Adopting best fair schedulers

Putting it all together

We saw

Increase leftover via Inter-Job Scheduling

- Adopting best fair schedulers

Compute leftover via Intra-Job Scheduling

Putting it all together

We saw

Increase leftover via Inter-Job Scheduling
- Adopting best fair schedulers

Compute leftover via Intra-Job Scheduling

Leftover redistribution
- Improve JCT and cluster efficiency

Putting it all together

We saw

Increase leftover via Inter-Job Scheduling
- Adopting best fair schedulers

Compute leftover via Intra-Job Scheduling

Leftover redistribution
- Improve JCT and cluster efficiency

Other things in the paper

Bounding altruism with *P(Altruism)*

Putting it all together

We saw

Increase leftover via Inter-Job Scheduling
- Adopting best fair schedulers

Compute leftover via Intra-Job Scheduling

Leftover redistribution
- Improve JCT and cluster efficiency

Other things in the paper

Bounding altruism with *P(Altruism)*

Resource estimation

Putting it all together

We saw

Increase leftover via Inter-Job Scheduling
- Adopting best fair schedulers

Compute leftover via Intra-Job Scheduling

Leftover redistribution
- Improve JCT and cluster efficiency

Other things in the paper

Bounding altruism with *P(Altruism)*

Resource estimation

Data locality

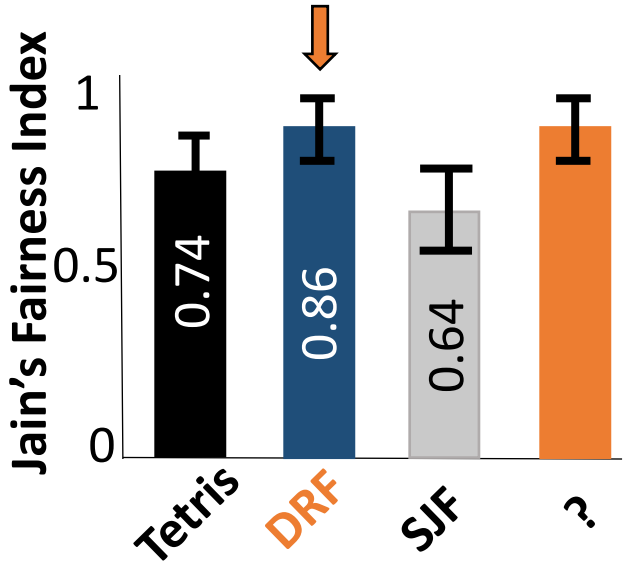
Straggler mitigation

Task failures

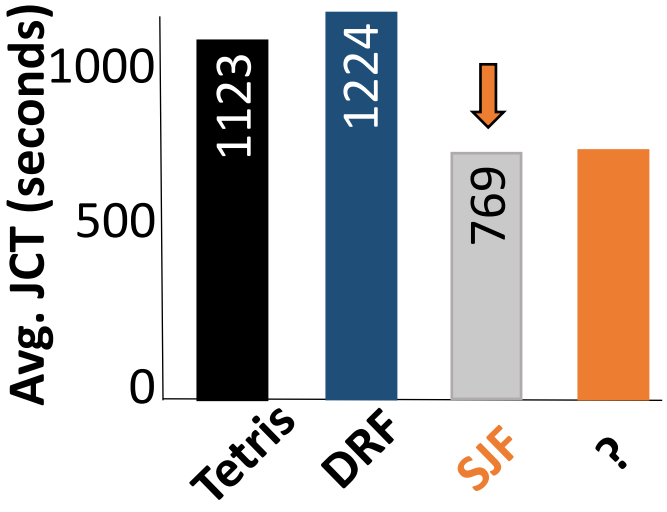
Evaluation

- Implemented in Yarn and Tez
- 100 machine cluster deployment
- Replay Bing / Facebook traces and TPC-DS / TPC-H workloads

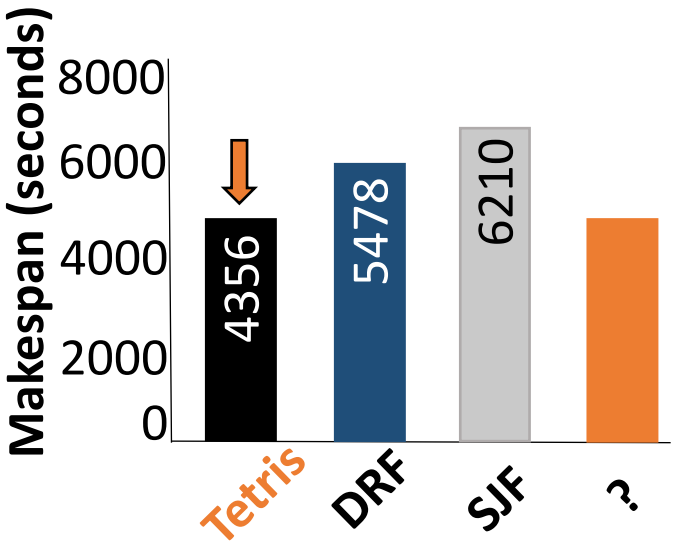
Fairness vs. Performance vs. Efficiency



Inter-job fairness

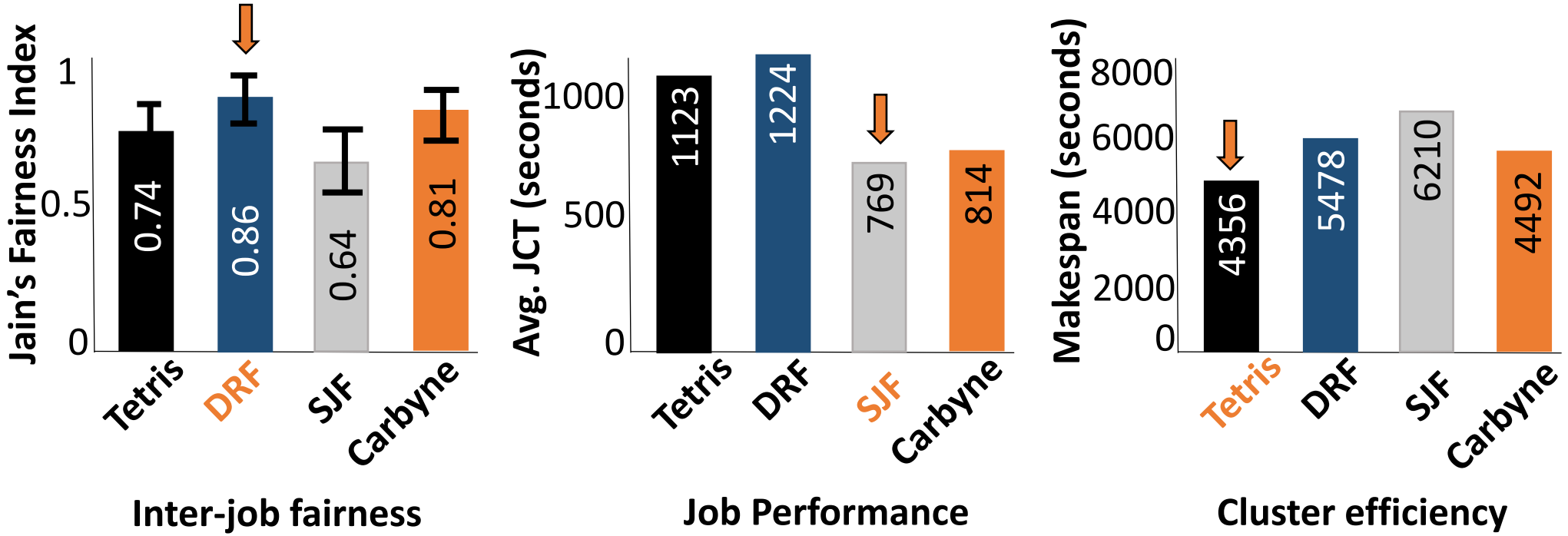


Job Performance



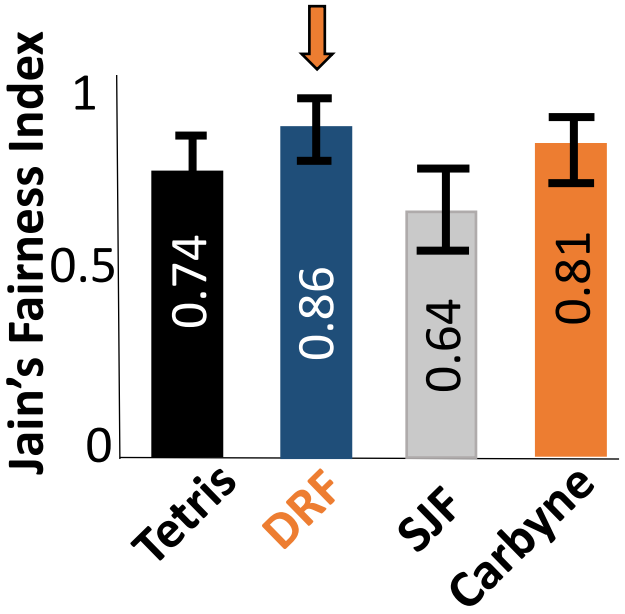
Cluster efficiency

Fairness vs. Performance vs. Efficiency

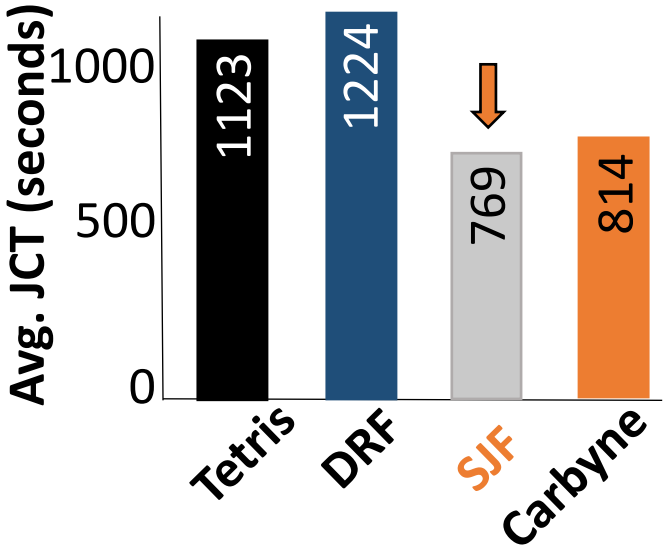


Comparable performance with
best approaches in each metric

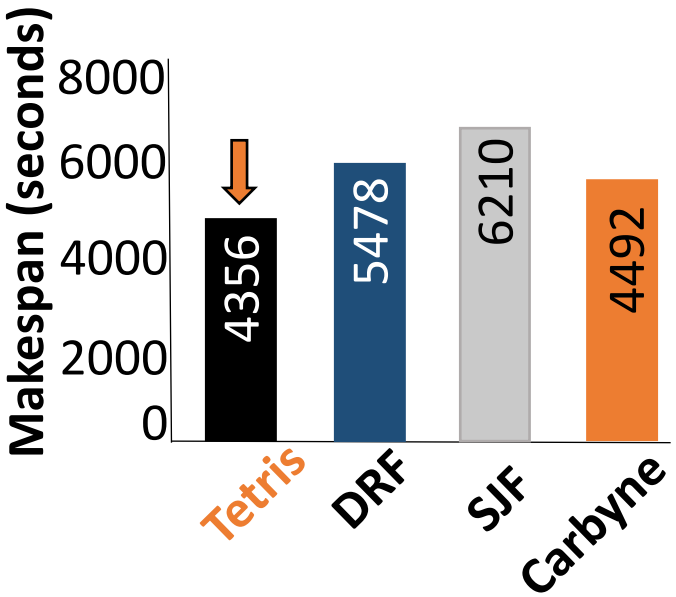
Fairness vs. Performance vs. Efficiency



Inter-job fairness



Job Performance



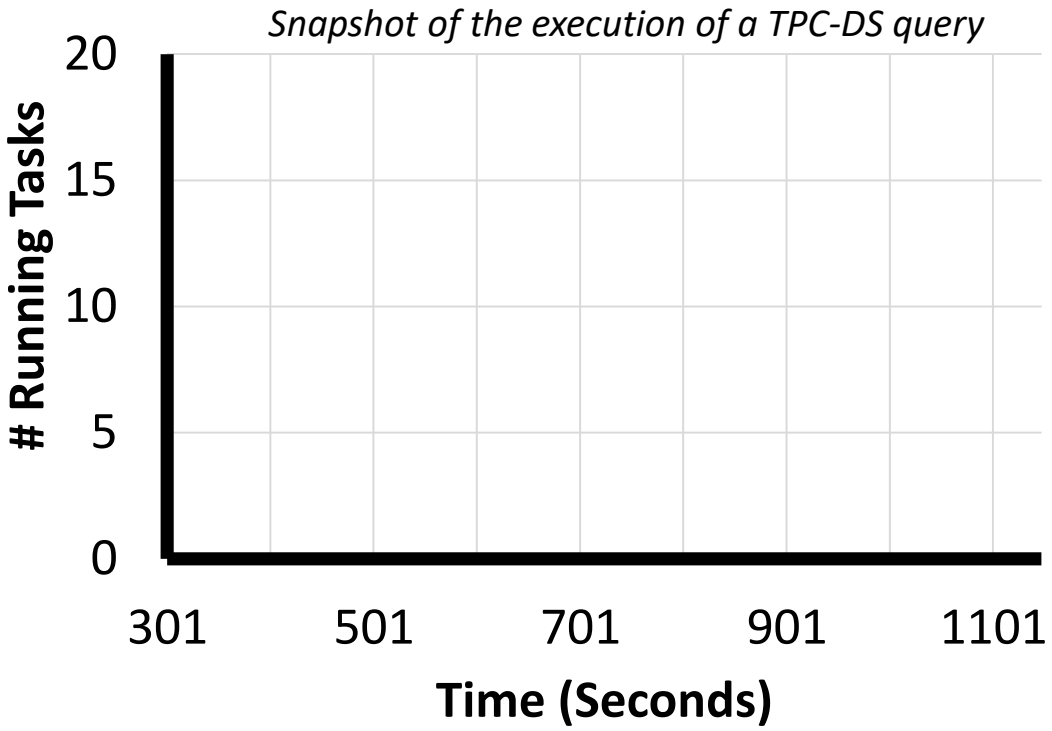
Cluster efficiency

Comparable performance with best approaches in each metric

Gains from

Altruism helps in long run

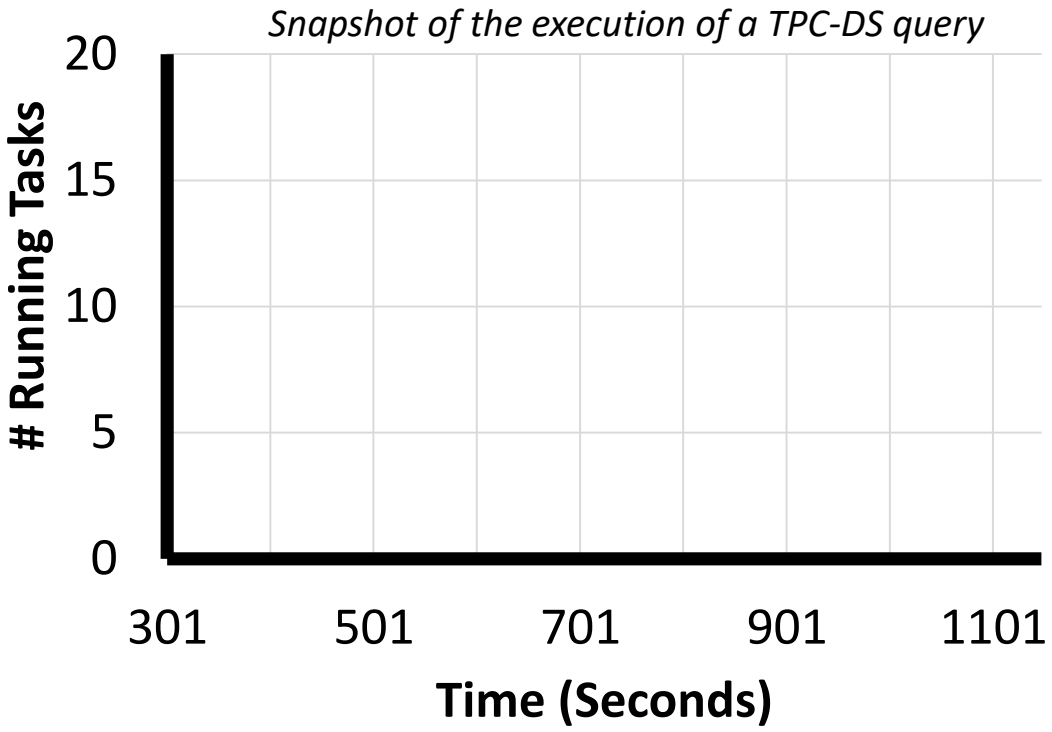
Job Performance



Gains from

Altruism helps in long run

Job Performance



DRF

- Tasks from a DRF allocation

Carbyne w/o leftover

- Only running tasks from being altruistic

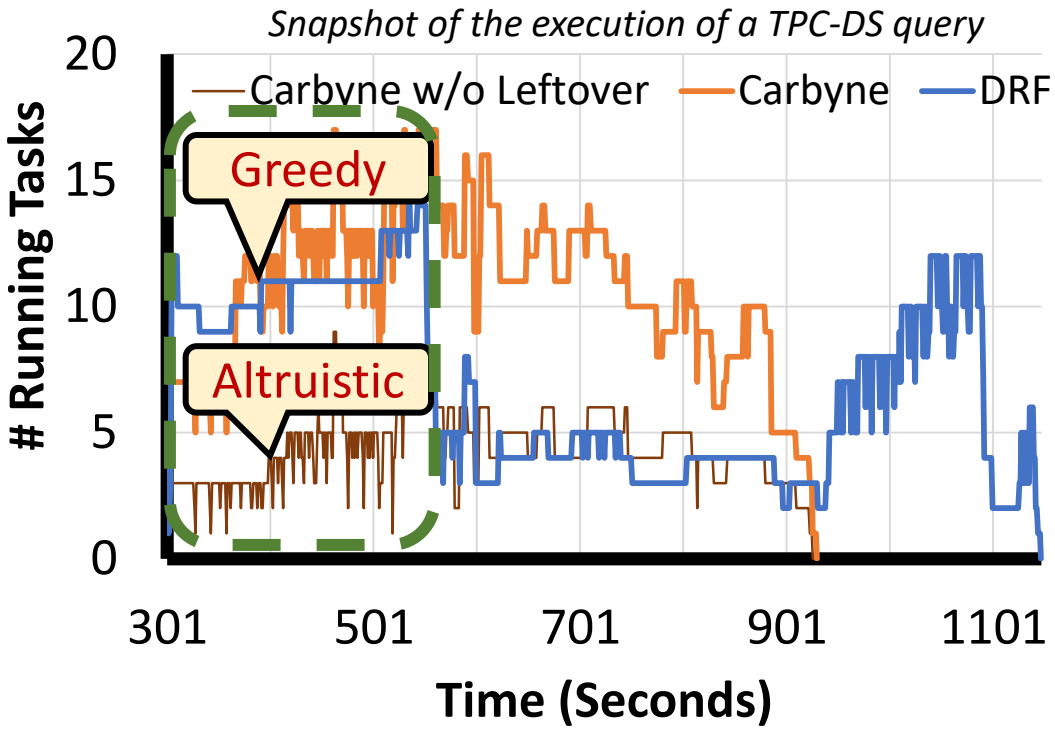
Carbyne

- Tasks from being altruistic and from leftover redistribution

Gains from

Altruism helps in long run

Job Performance



DRF

- Tasks from a DRF allocation

Carbyne w/o leftover

- Only running tasks from being altruistic

Carbyne

- Tasks from being altruistic and from leftover redistribution

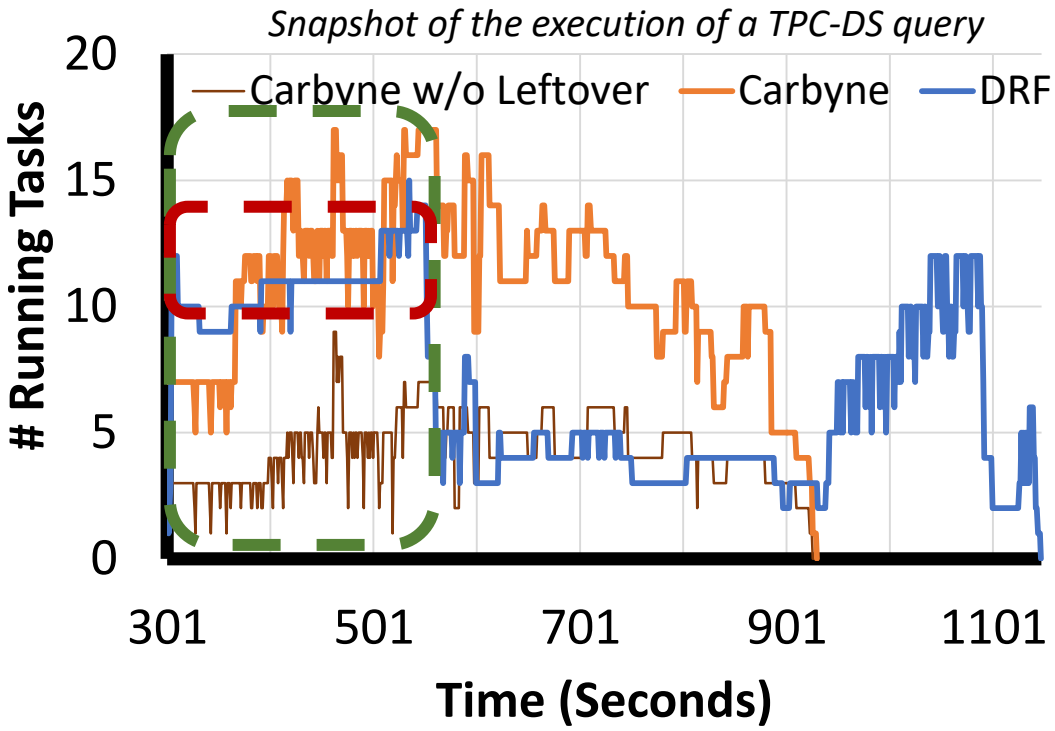
Low resource contention

- DRF takes greedy decisions

Gains from

Altruism helps in long run

Job Performance



DRF

- Tasks from a DRF allocation

Carbyne w/o leftover

- Only running tasks from being altruistic

Carbyne

- Tasks from being altruistic and from leftover redistribution

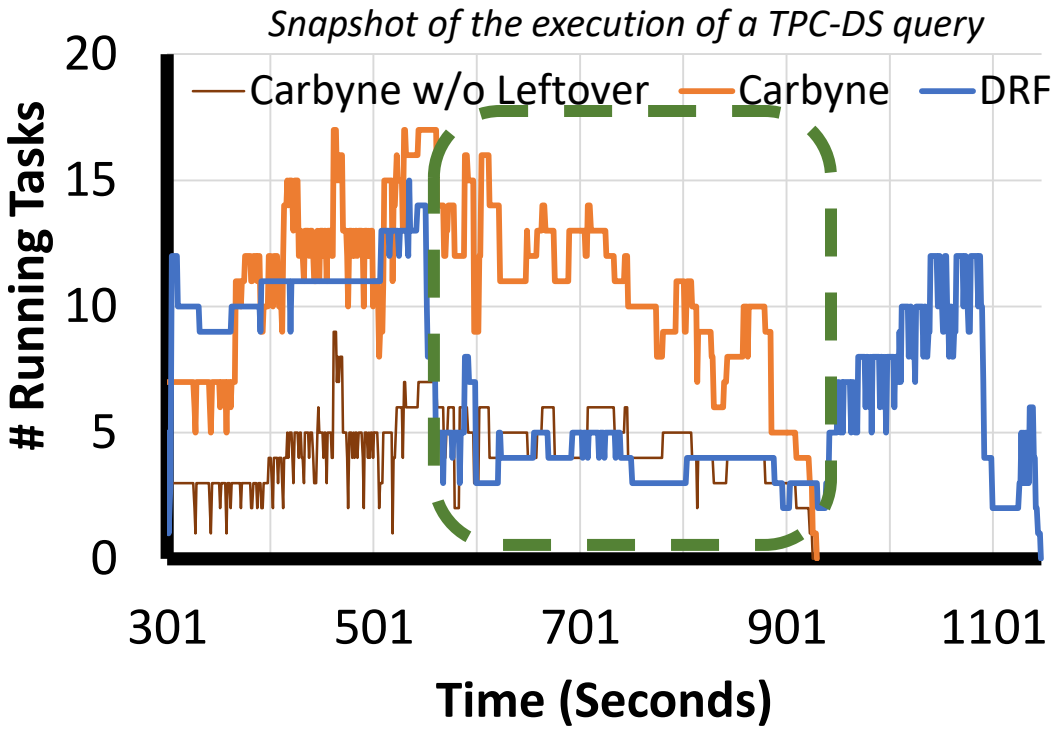
Low resource contention

- DRF takes greedy decisions
- Approximates DRF allocation due to leftover

Gains from

Altruism helps in long run

Job Performance



DRF

- Tasks from a DRF allocation

Carbyne w/o leftover

- Only running tasks from being altruistic

Carbyne

- Tasks from being altruistic and from leftover redistribution

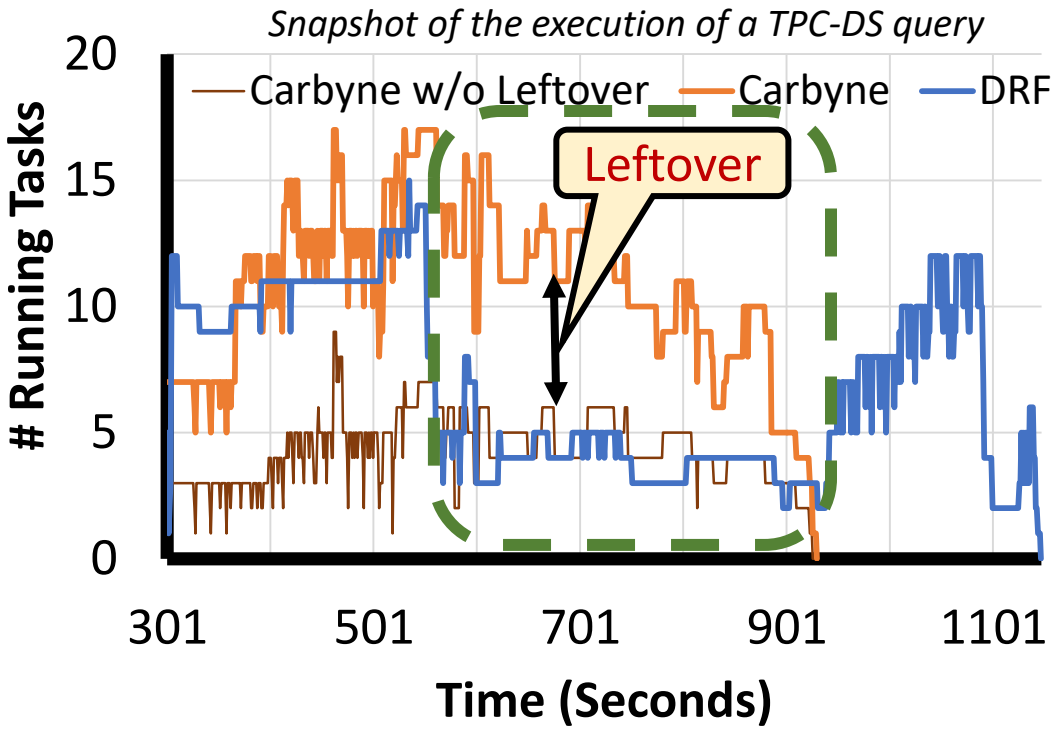
High resource contention

- DRF progress is slowed down

Gains from

Altruism helps in long run

Job Performance



DRF

- Tasks from a DRF allocation

Carbyne w/o leftover

- Only running tasks from being altruistic

Carbyne

- Tasks from being altruistic and from leftover redistribution

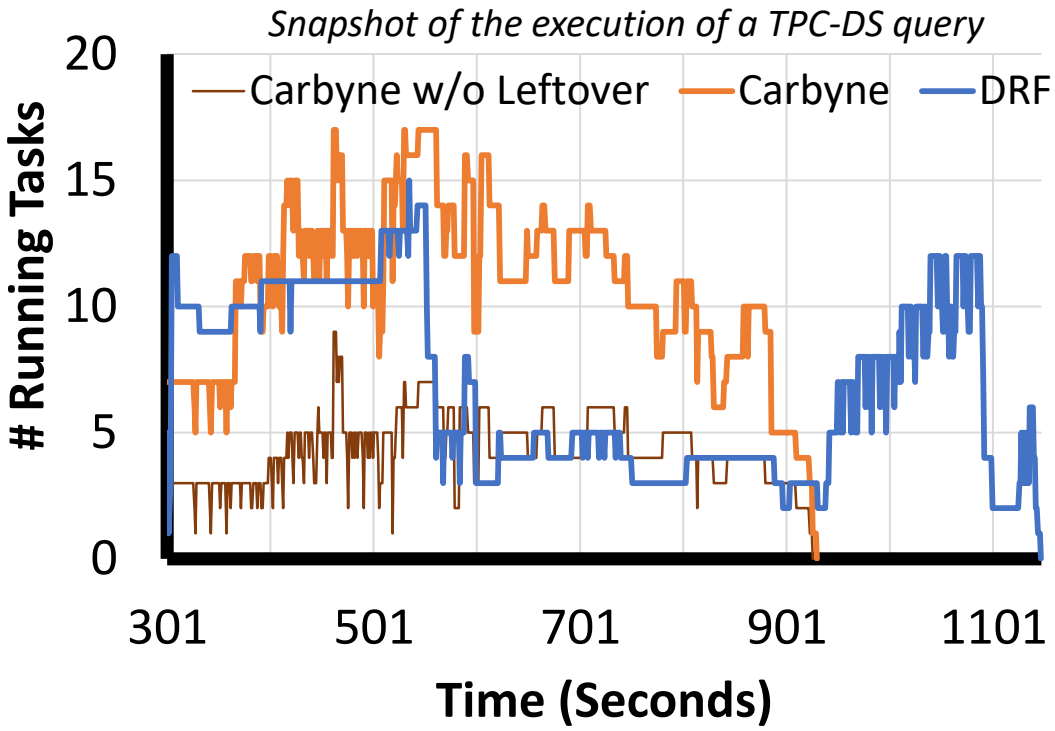
High resource contention

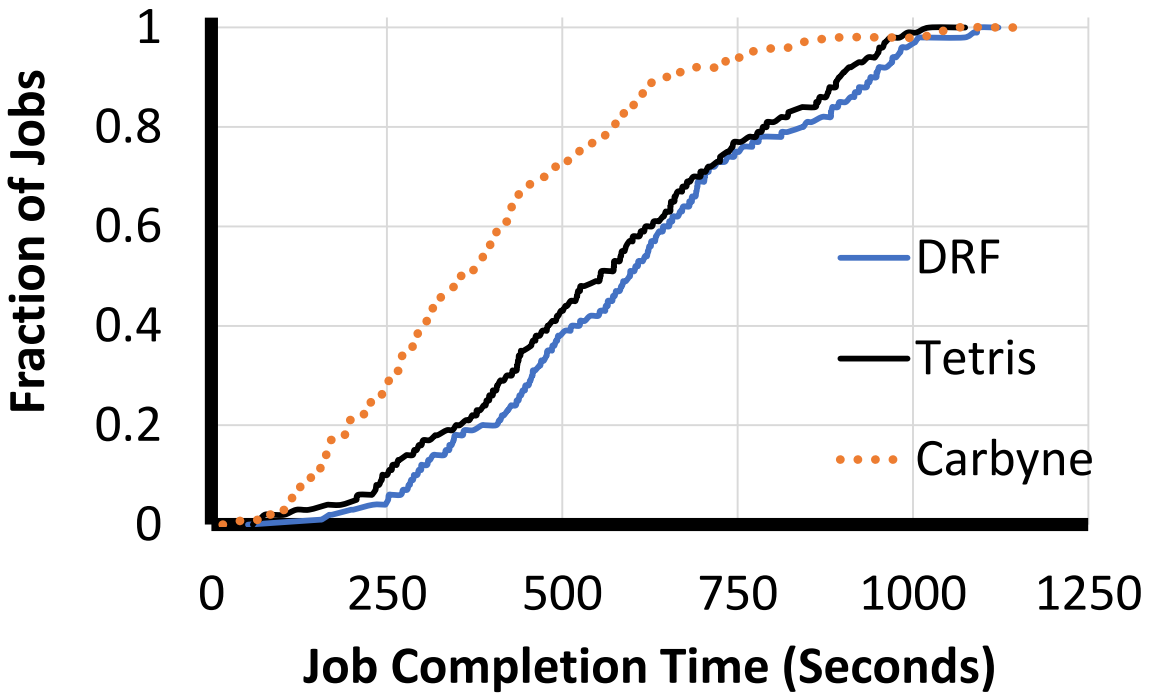
- DRF progress is slowed down
- Carbyne progress faster due to receiving leftover

Gains from

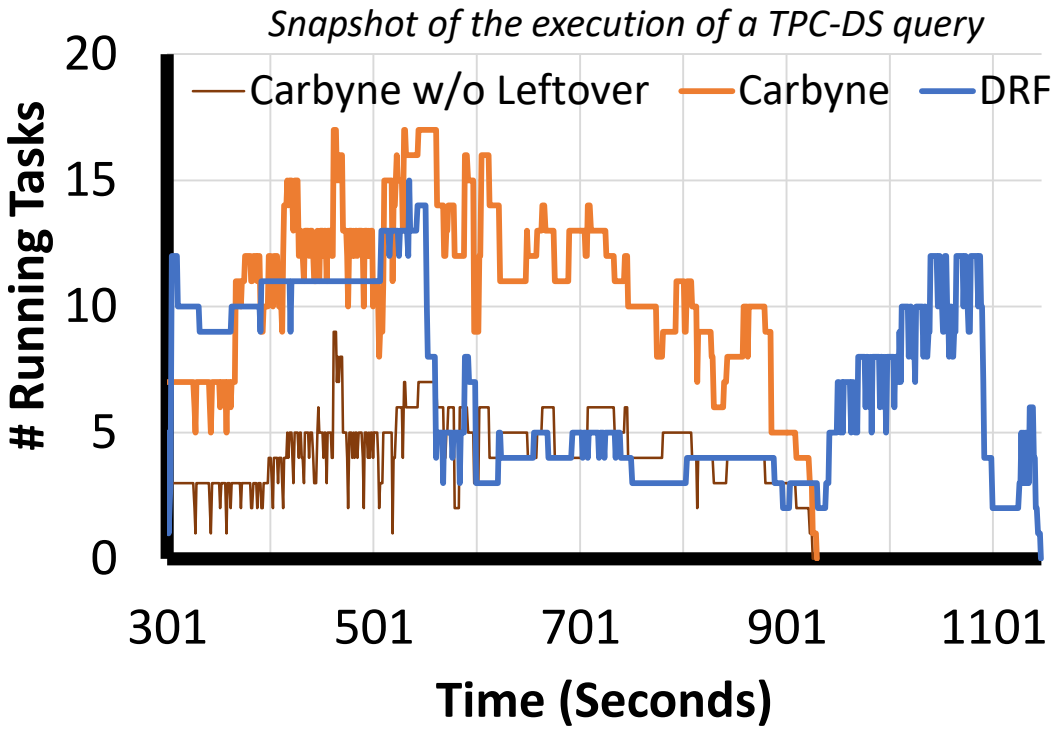
Altruism helps in long run

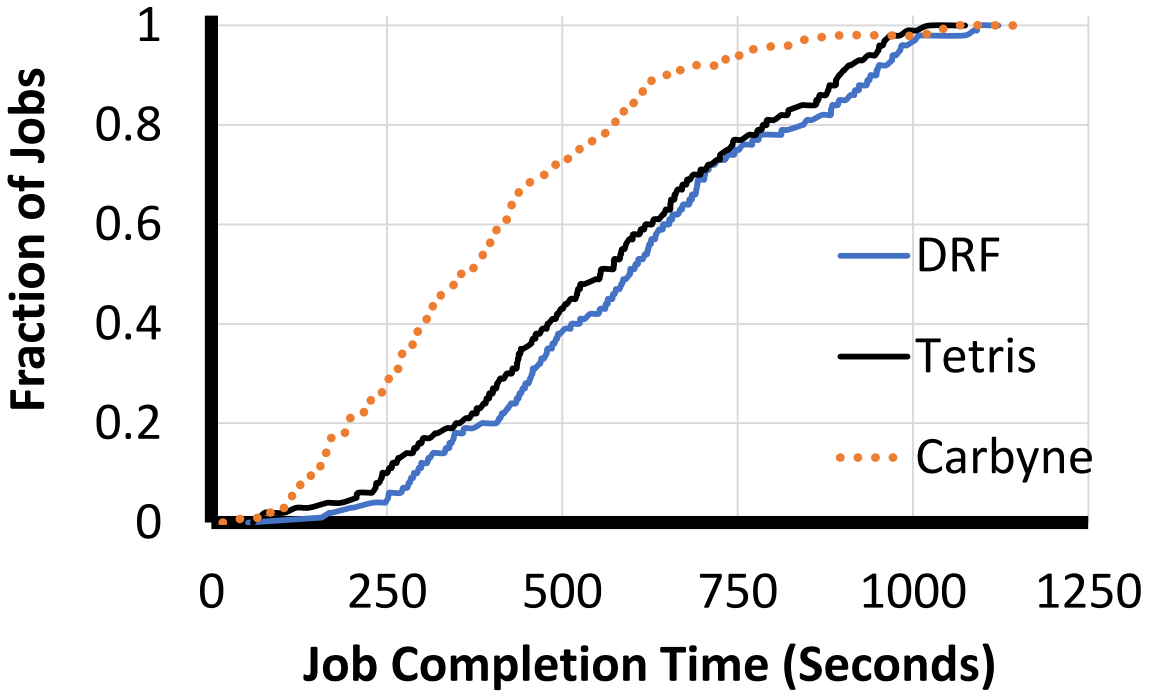
Job Performance





Job Performance

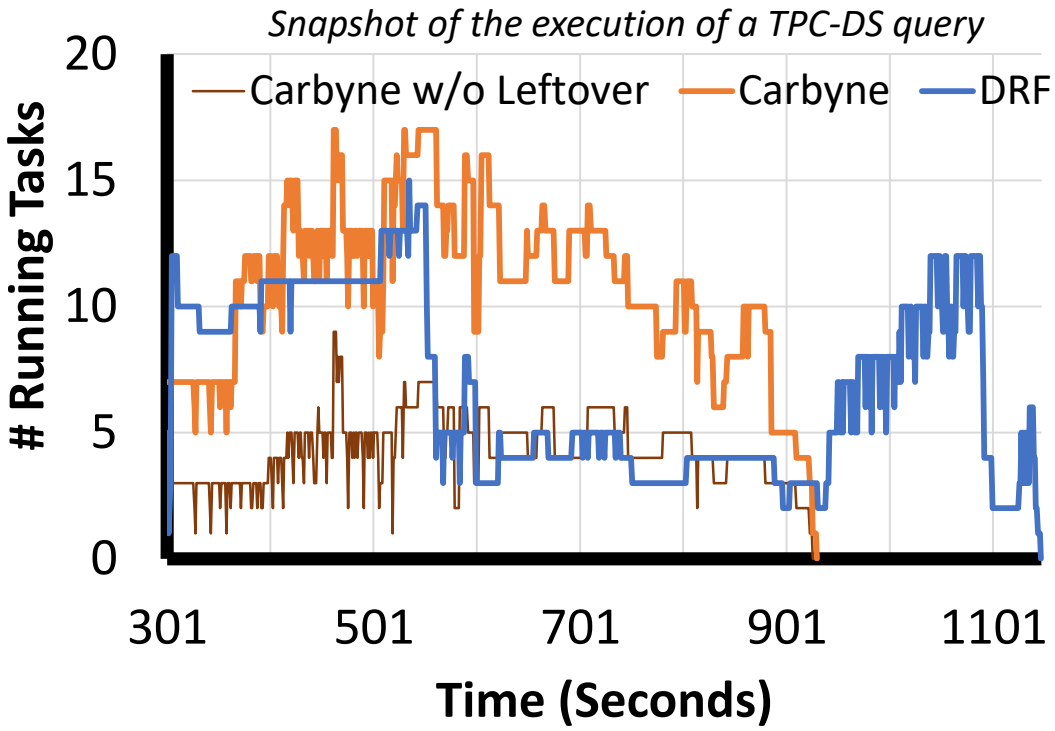


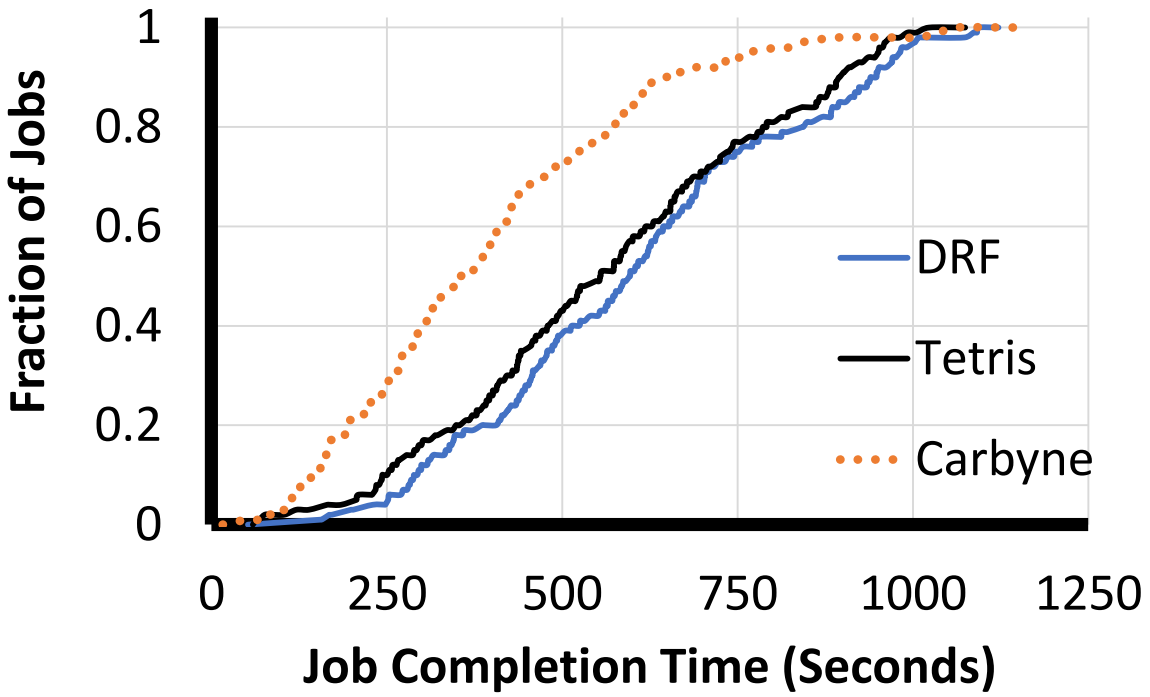


Performance

- Better jobs completion time

Job Performance

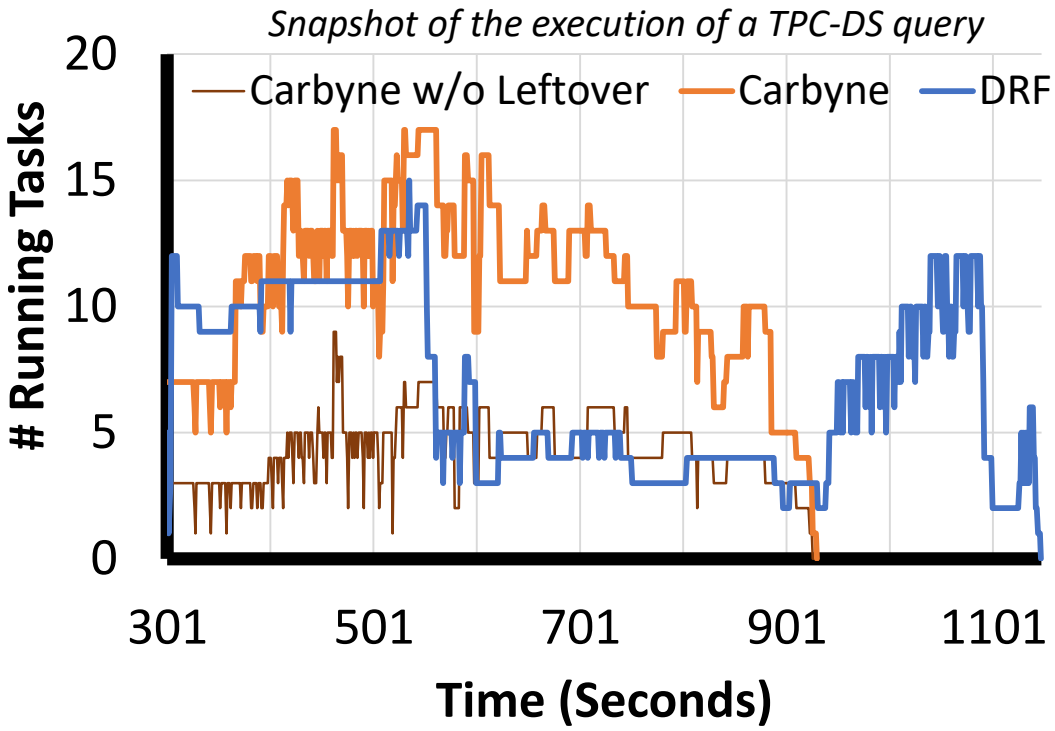


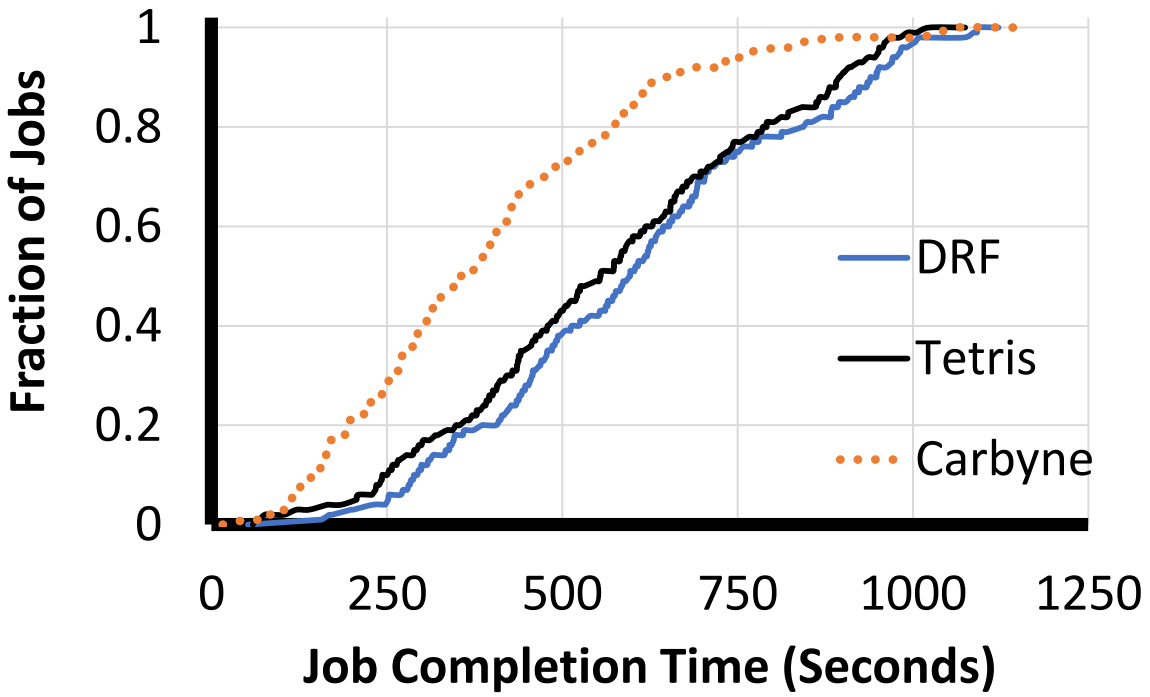


Performance

- Better jobs completion time
- 16% jobs slowed down
- 4% only by more than 0.8x

Job Performance



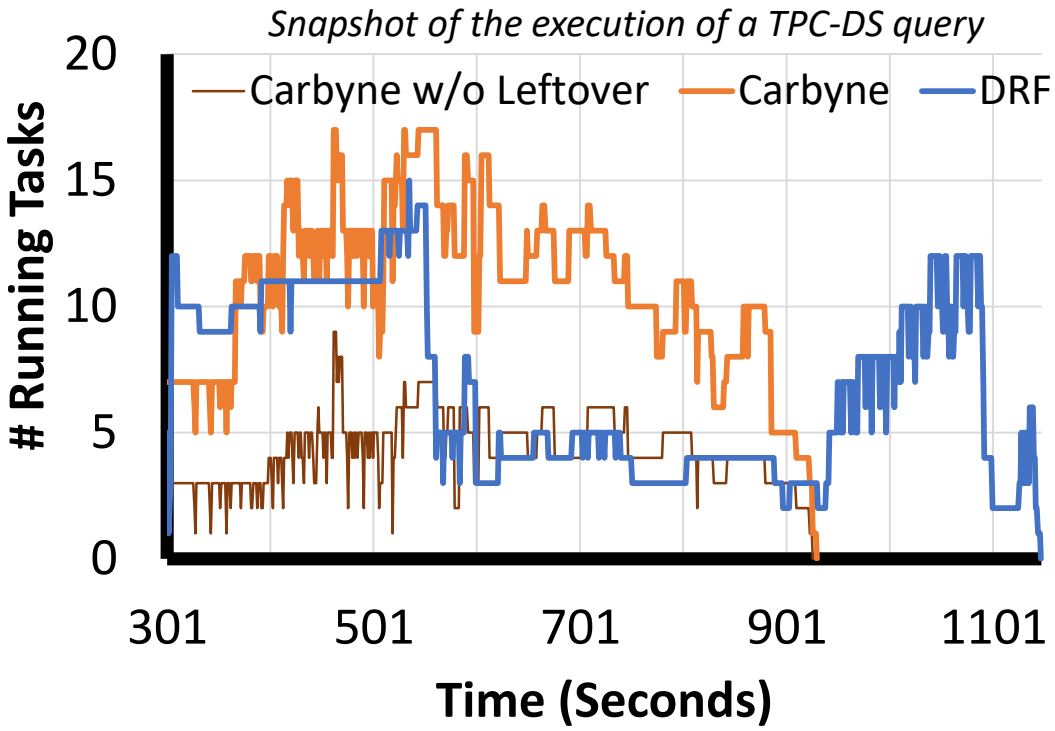


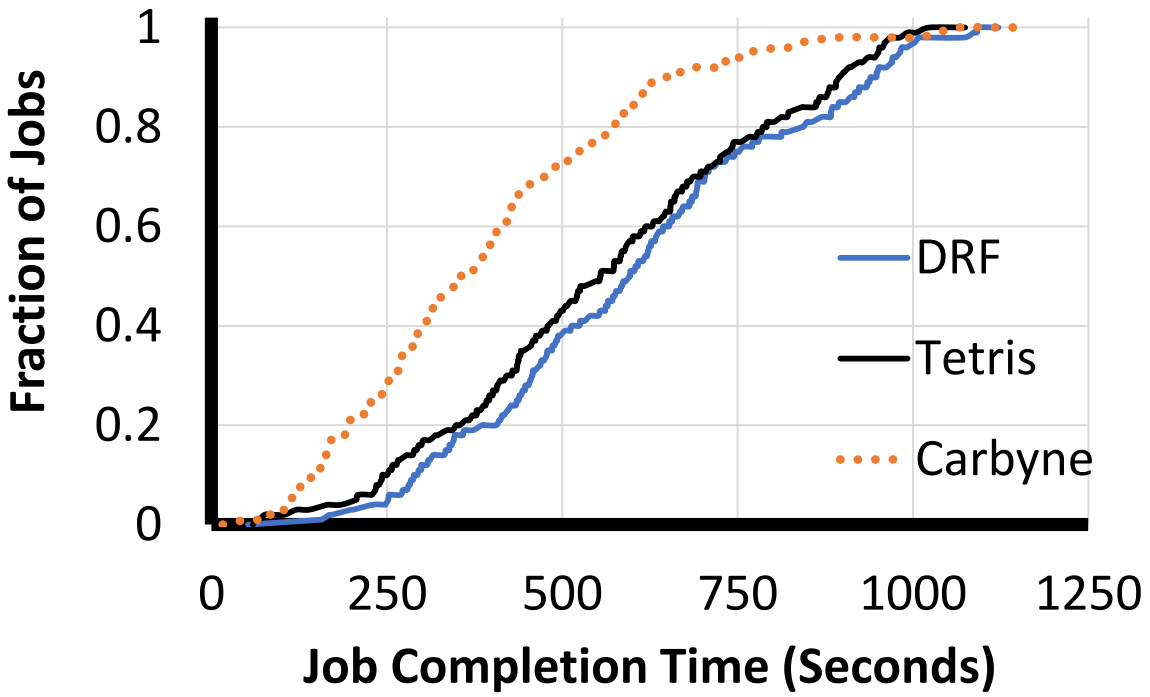
Performance

- Better jobs completion time
- 16% jobs slowed down
- 4% only by more than 0.8x

Which jobs slow down?

Job Performance





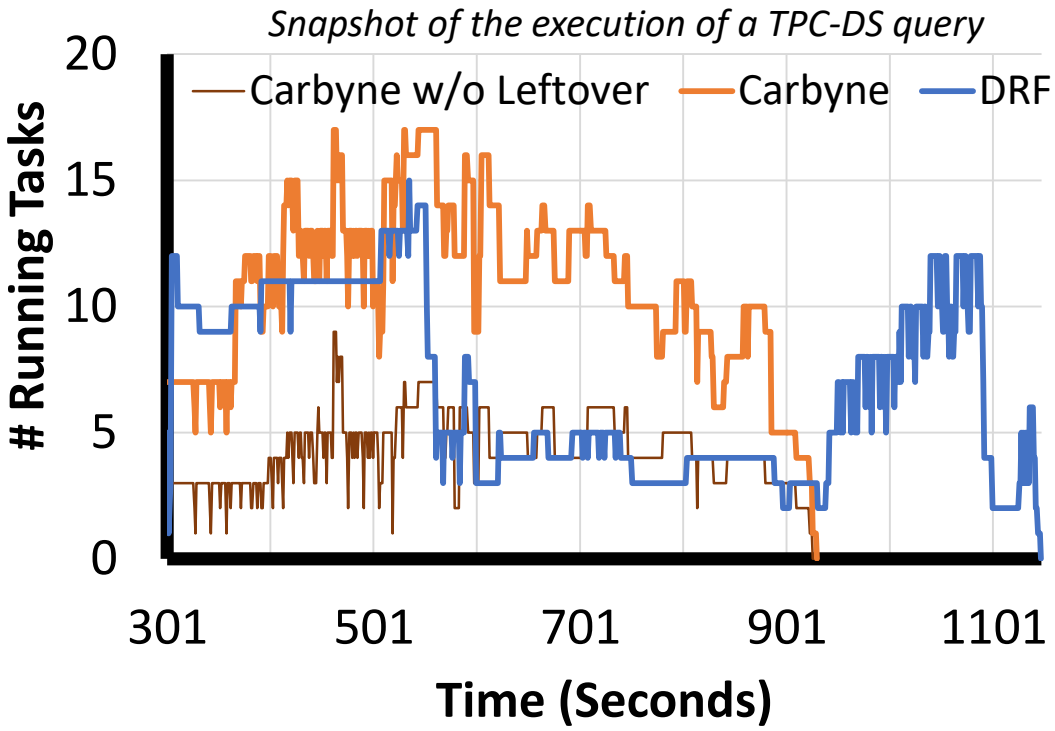
Performance

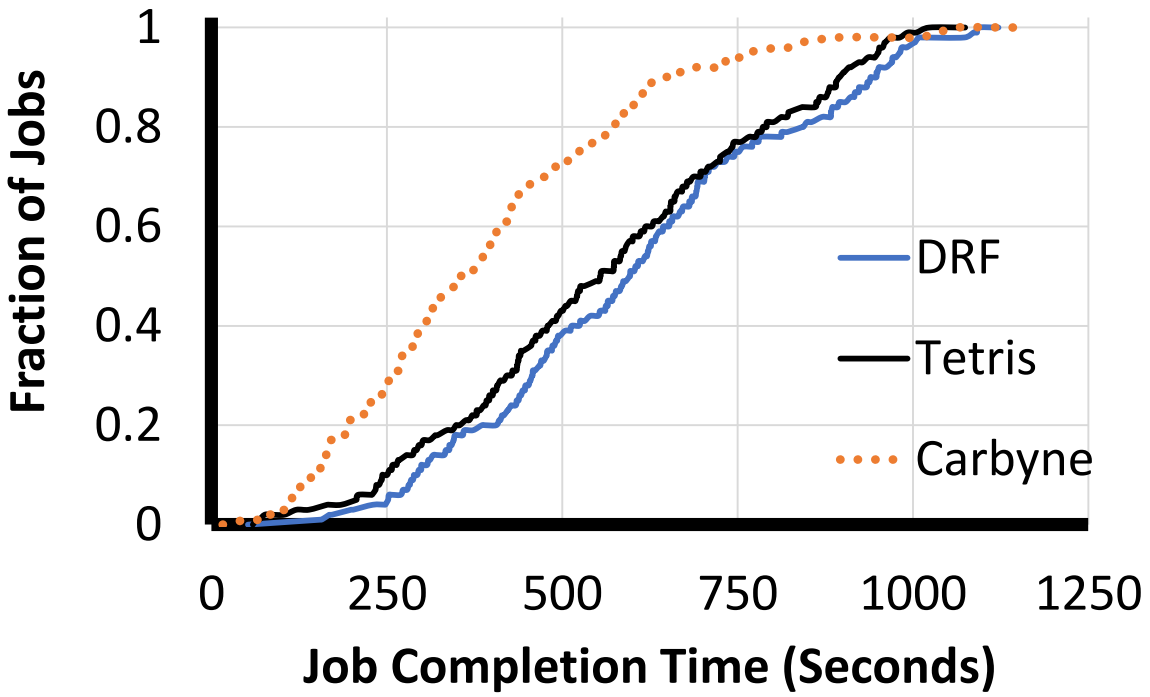
- Better jobs completion time
- 16% jobs slowed down
- 4% only by more than 0.8x

Which jobs slow down?

- Longer jobs

Job Performance





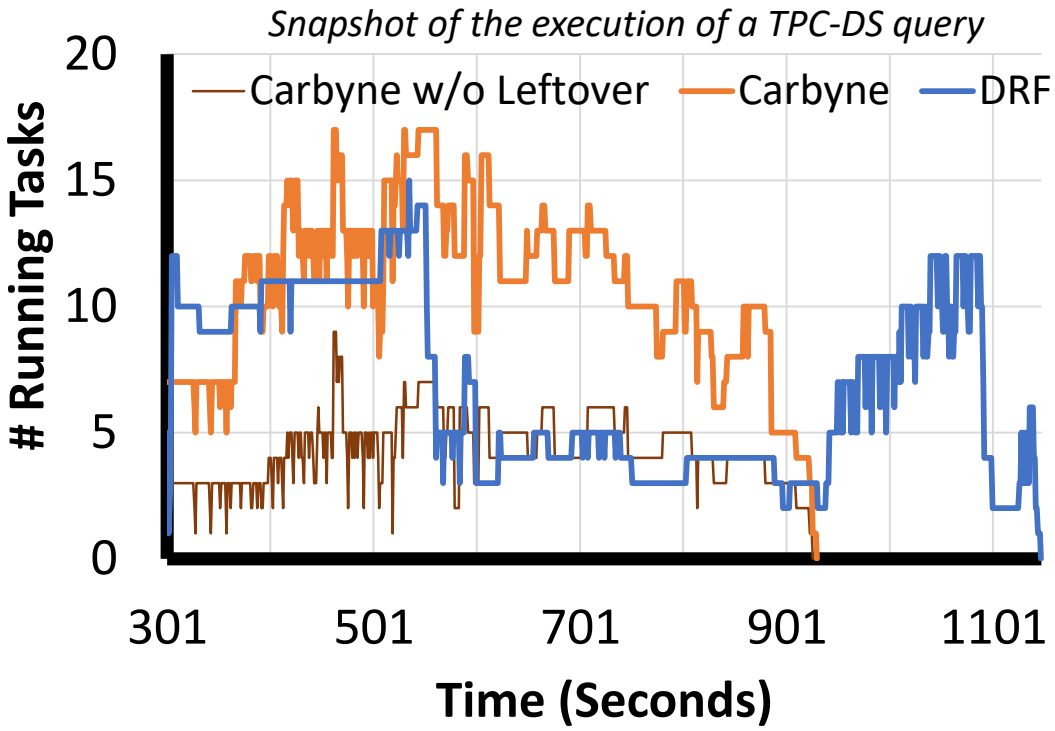
Performance

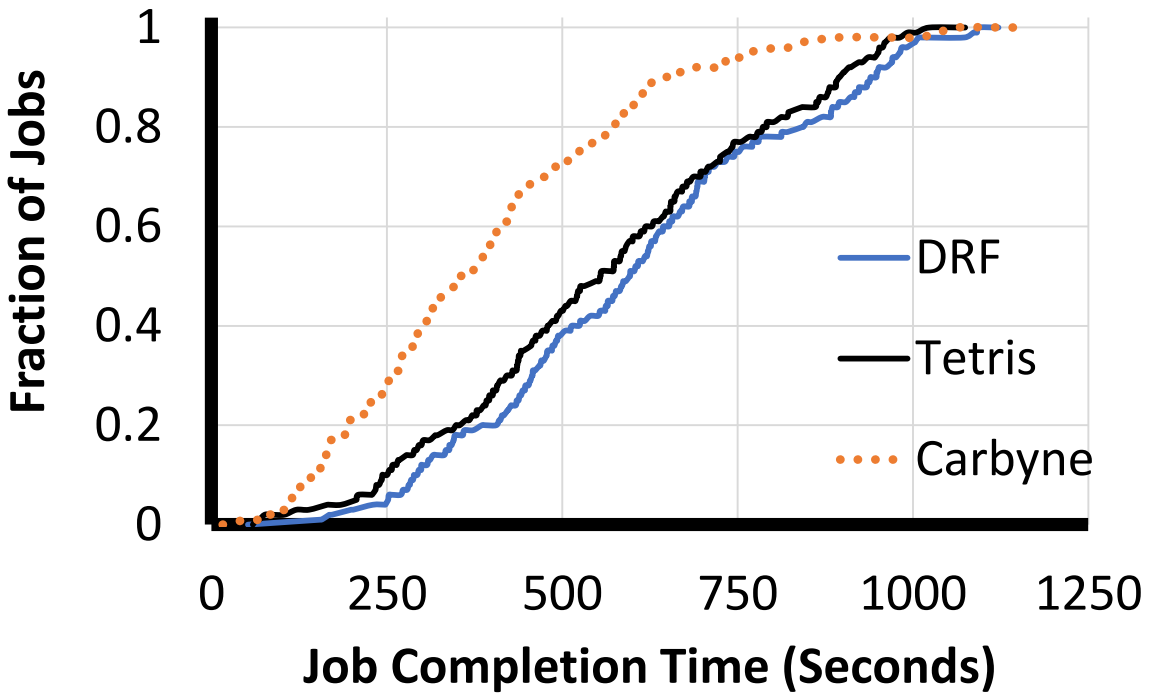
- Better jobs completion time
- 16% jobs slowed down
- 4% only by more than 0.8x

Which jobs slow down?

- Longer jobs
- No bias towards shorter jobs

Job Performance





Performance

- Better jobs completion time
- 16% jobs slowed down
- 4% only by more than 0.8x

Which jobs slow down?

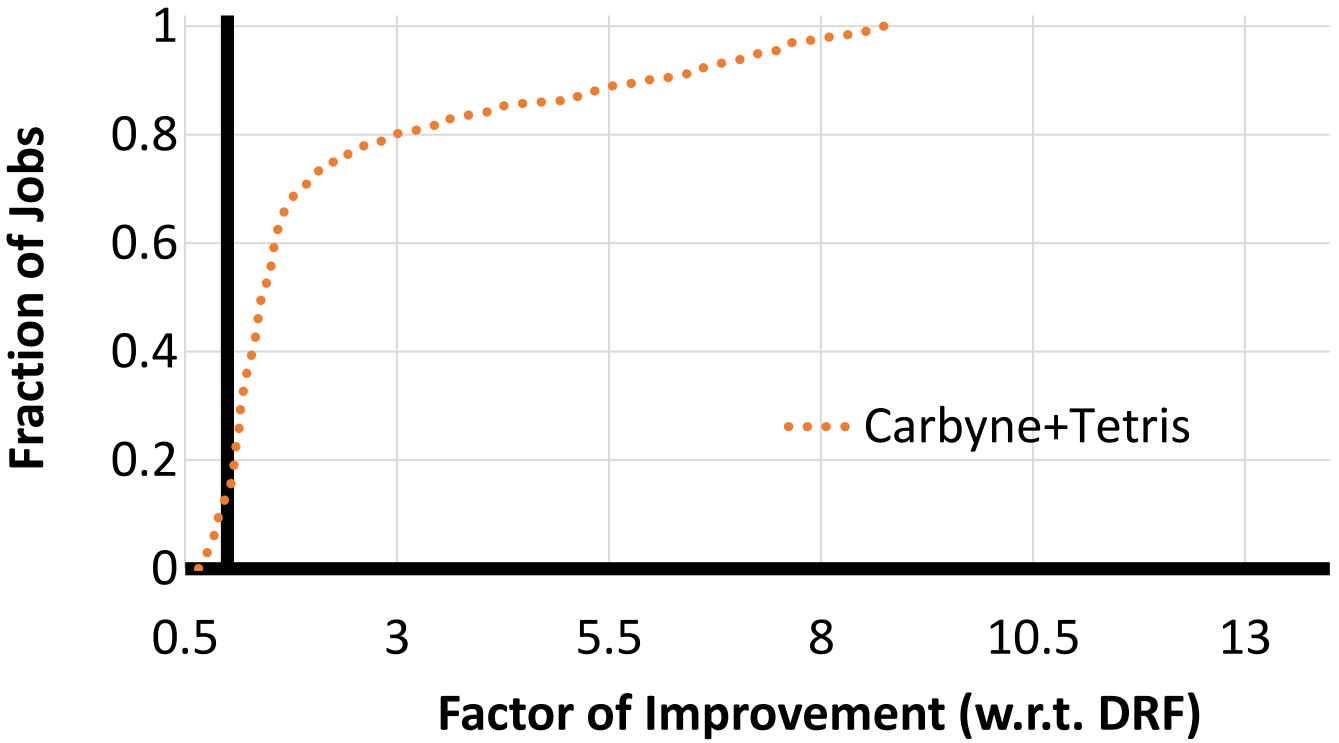
- Longer jobs
- No bias towards shorter jobs
- Most jobs benefit from leftover

Impact of a Better Intra-Job Scheduler

Default intra-job scheduler
- Tetris

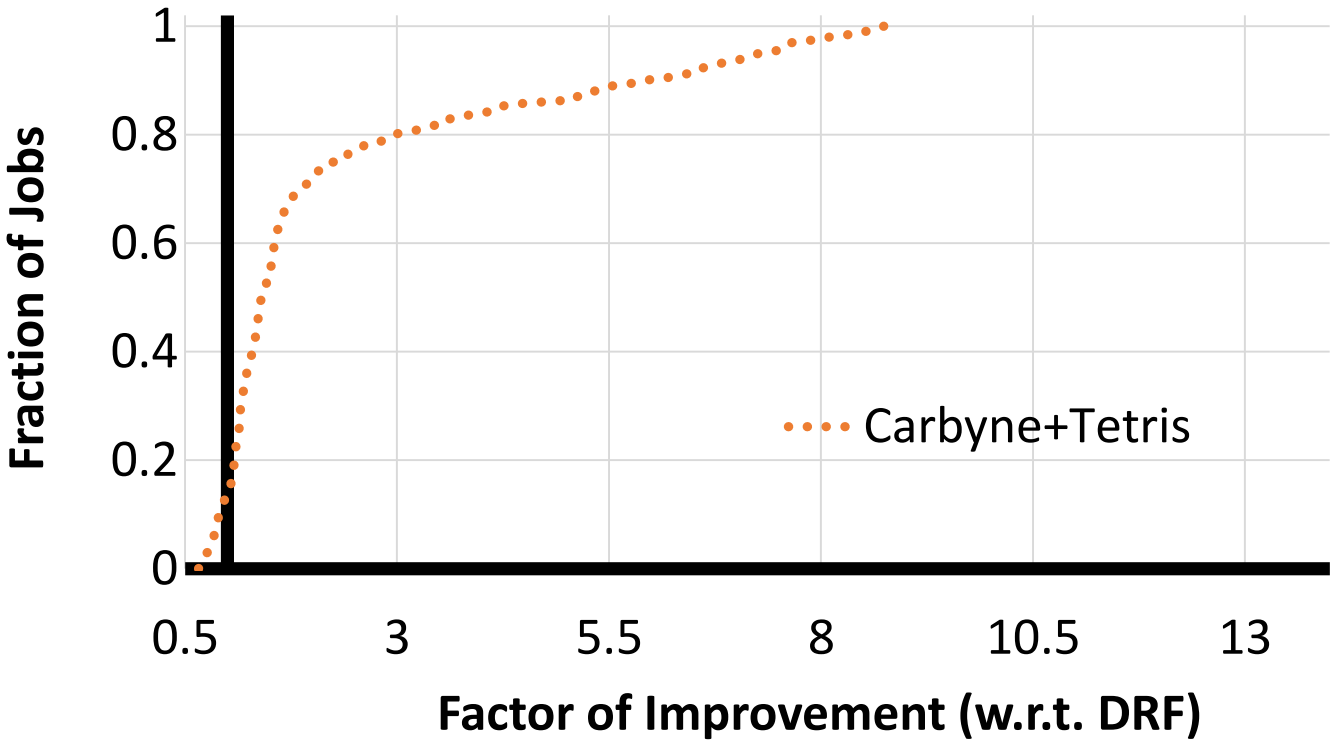
Impact of a Better Intra-Job Scheduler

Default intra-job scheduler
- Tetris



Impact of a Better Intra-Job Scheduler

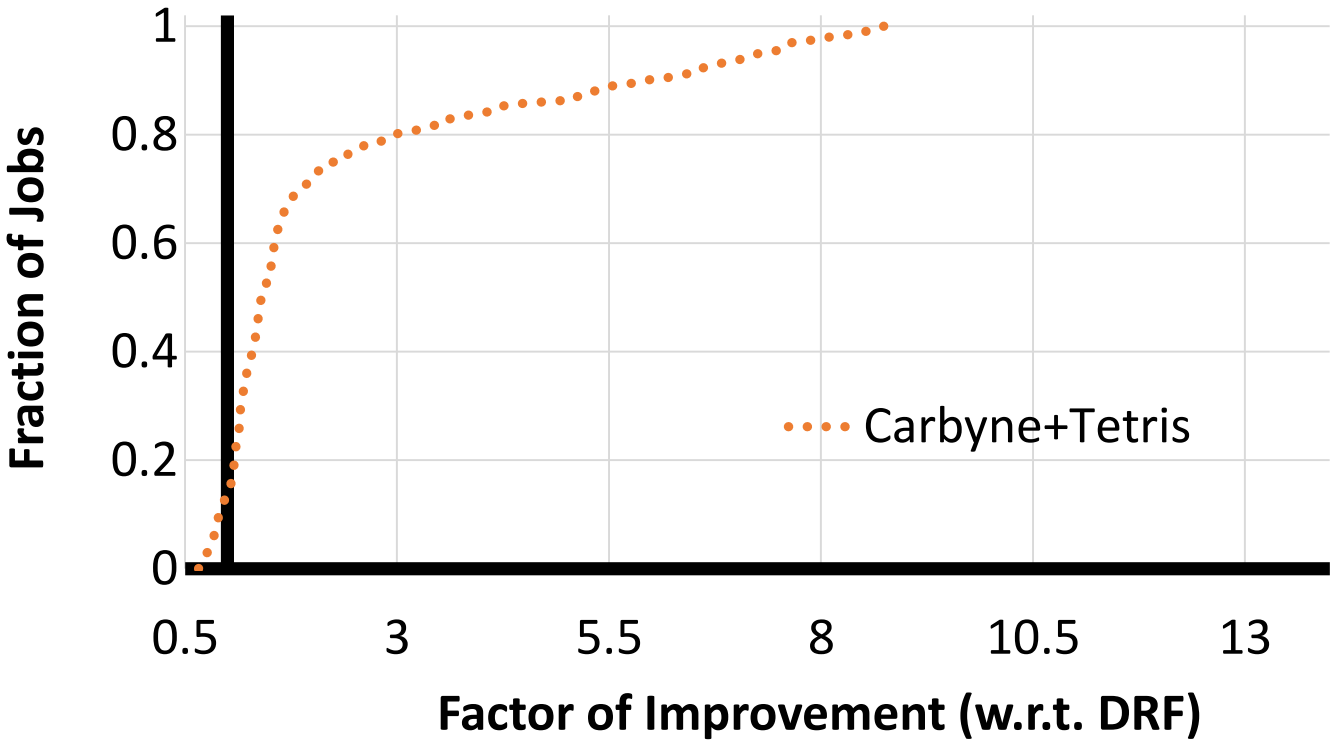
Default intra-job scheduler
- Tetris
- Limited view of the job's DAG



Impact of a Better Intra-Job Scheduler

Default intra-job scheduler
- Tetris
- Limited view of the job's DAG

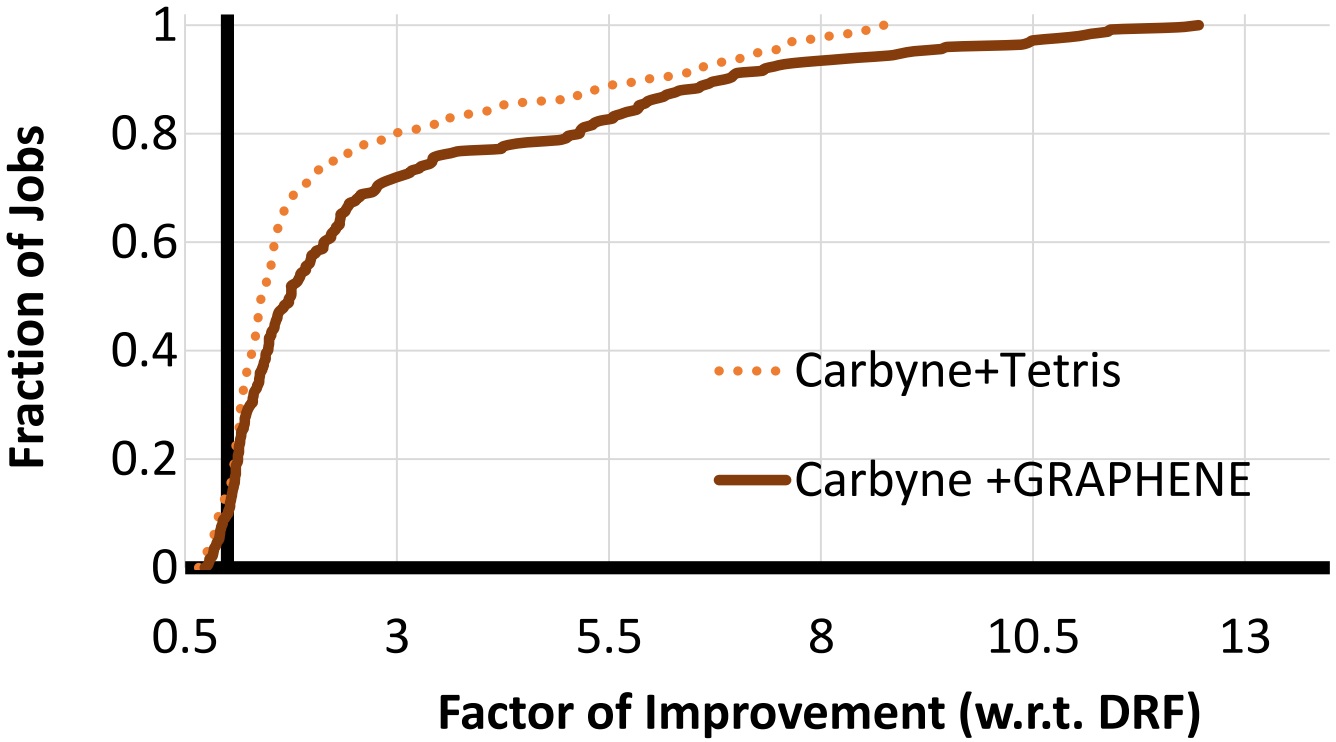
Better intra-job scheduler
- Graphene – DAG-wide view



Impact of a Better Intra-Job Scheduler

Default intra-job scheduler
- Tetris
- Limited view of the job's DAG

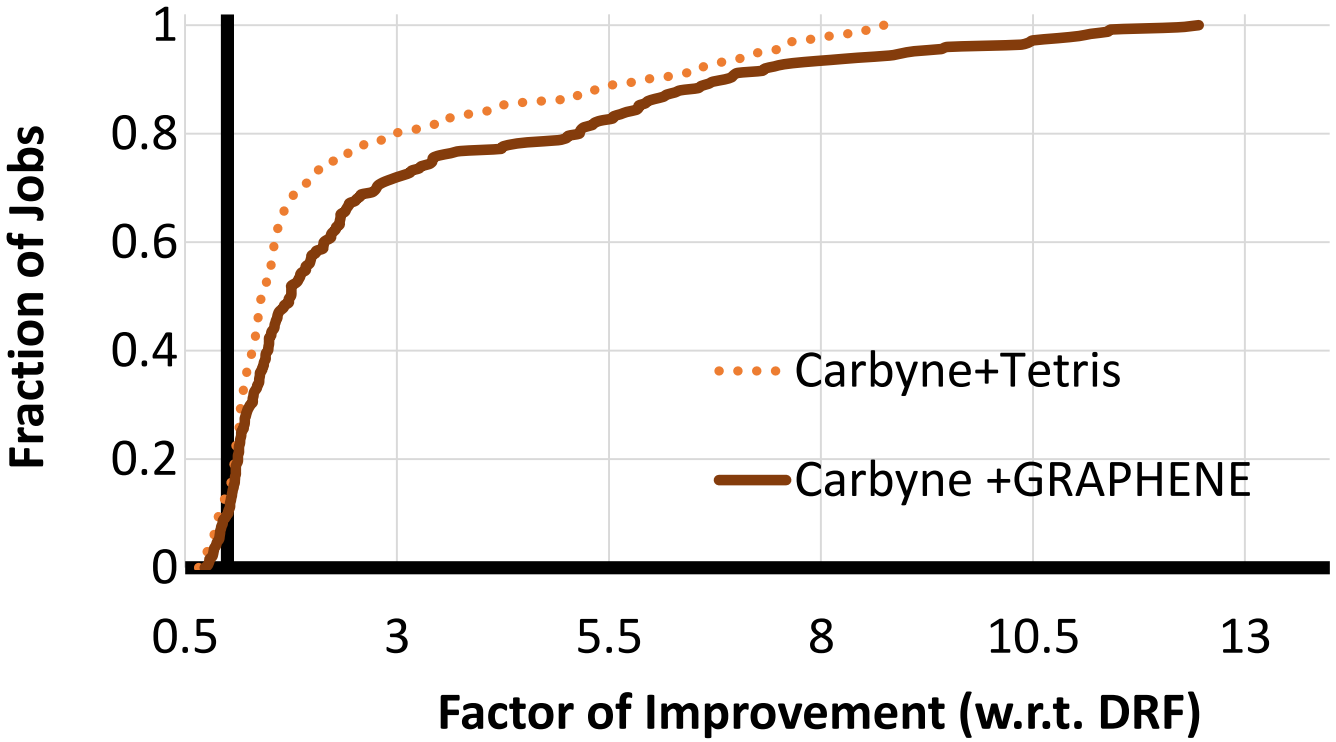
Better intra-job scheduler
- Graphene – DAG-wide view



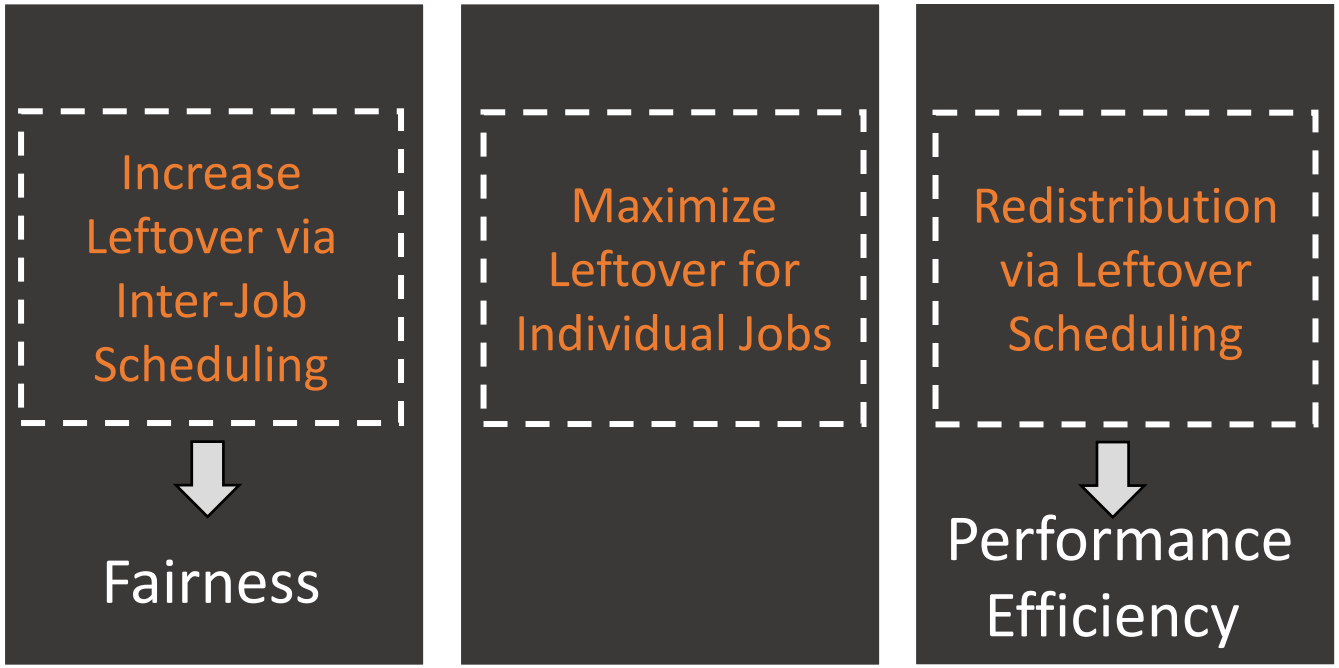
Impact of a Better Intra-Job Scheduler

Default intra-job scheduler
- Tetris
- Limited view of the job's DAG

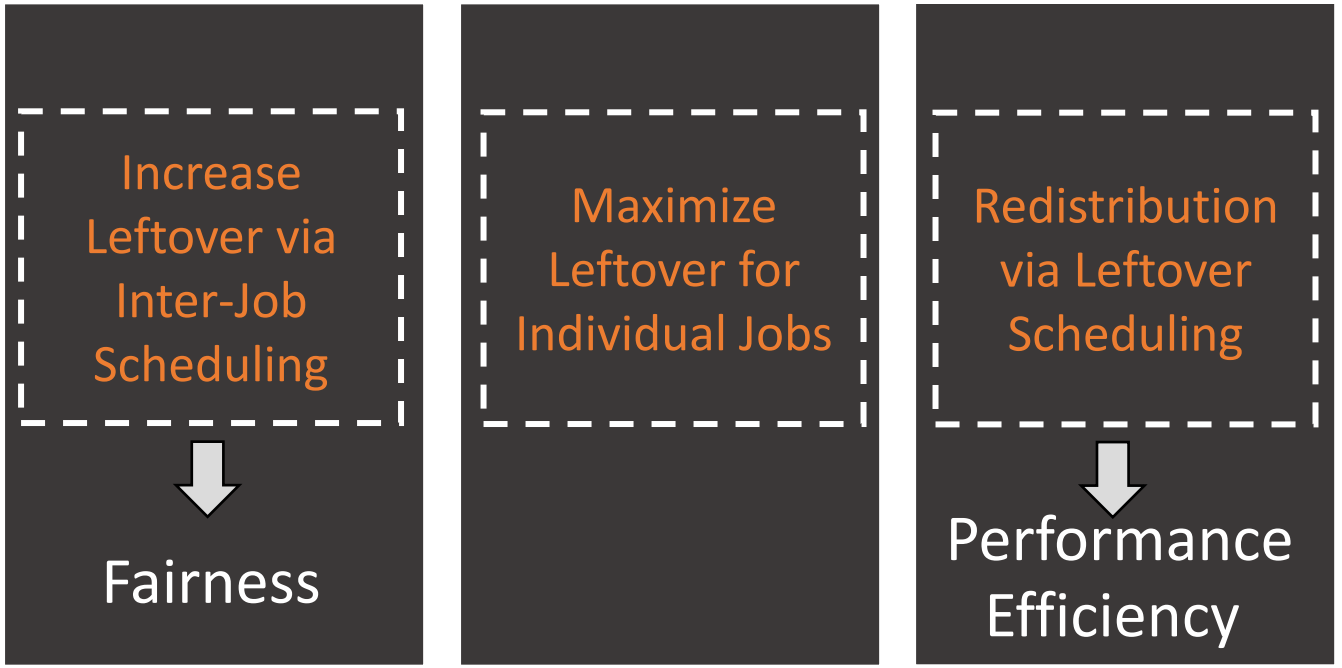
Better intra-job scheduler
- Graphene – DAG-wide view
- Extracts more leftover
- Further increase performance



Carbyne

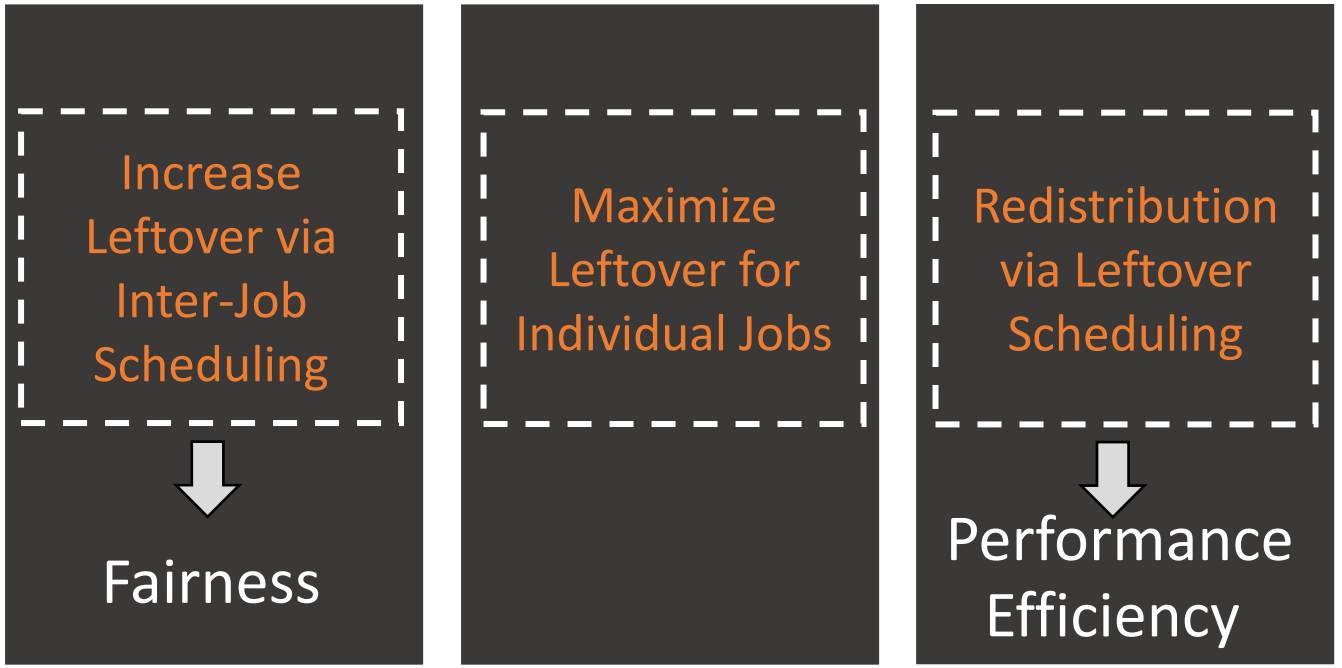

Increase Leftover via Inter-Job Scheduling
Fairness
Maximize Leftover for Individual Jobs
Redistribution via Leftover Scheduling
Performance Efficiency

Carbyne



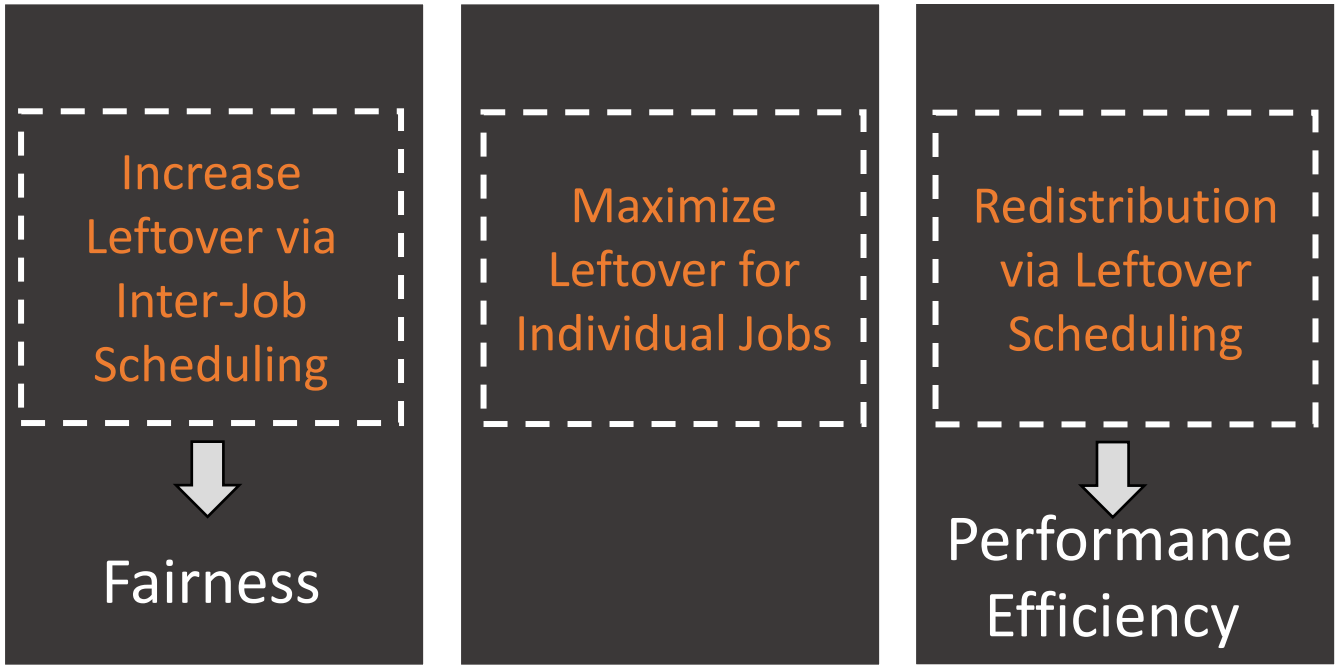
- **Long-term altruistic view** of Carbyne **outperforms** existing cluster schedulers which focus on instantaneous fairness

Carbyne



- **Long-term altruistic view** of Carbyne **outperforms** existing cluster schedulers which focus on instantaneous fairness
- Implemented inside YARN and Tez

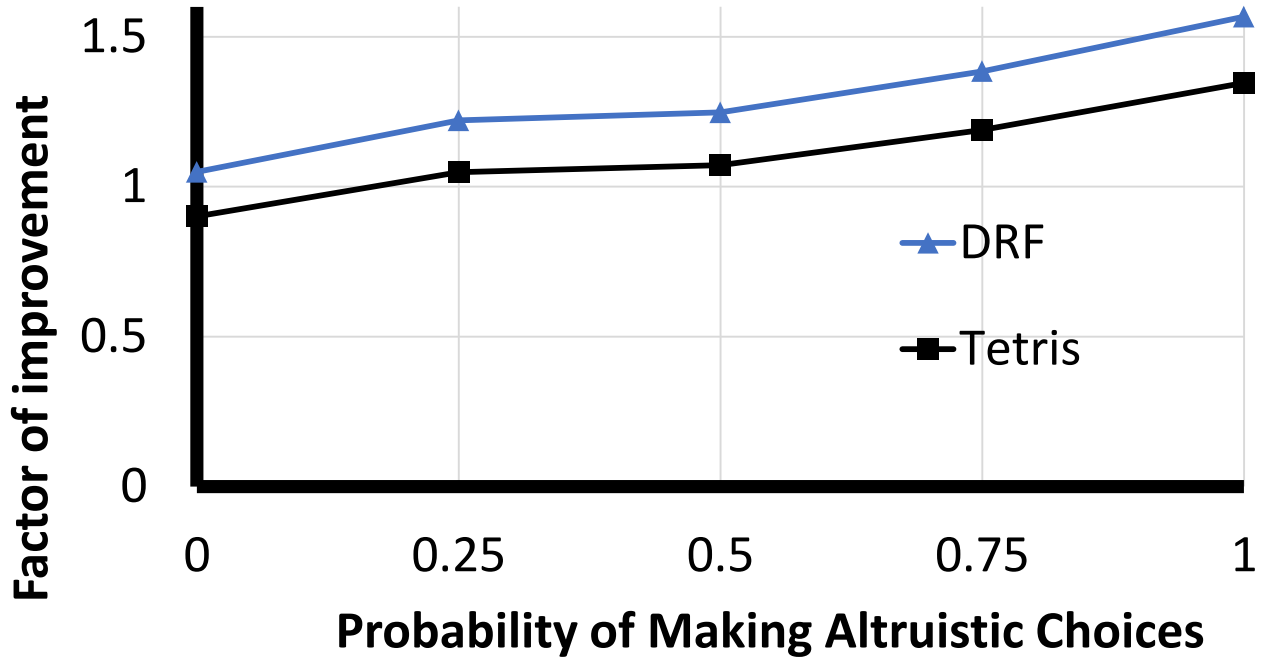
Carbyne



- **Long-term altruistic view** of Carbyne **outperforms** existing cluster schedulers which focus on instantaneous fairness
- Implemented inside YARN and Tez
- Performance comparable with best approaches in terms of fairness **and** job completion time **and** cluster efficiency

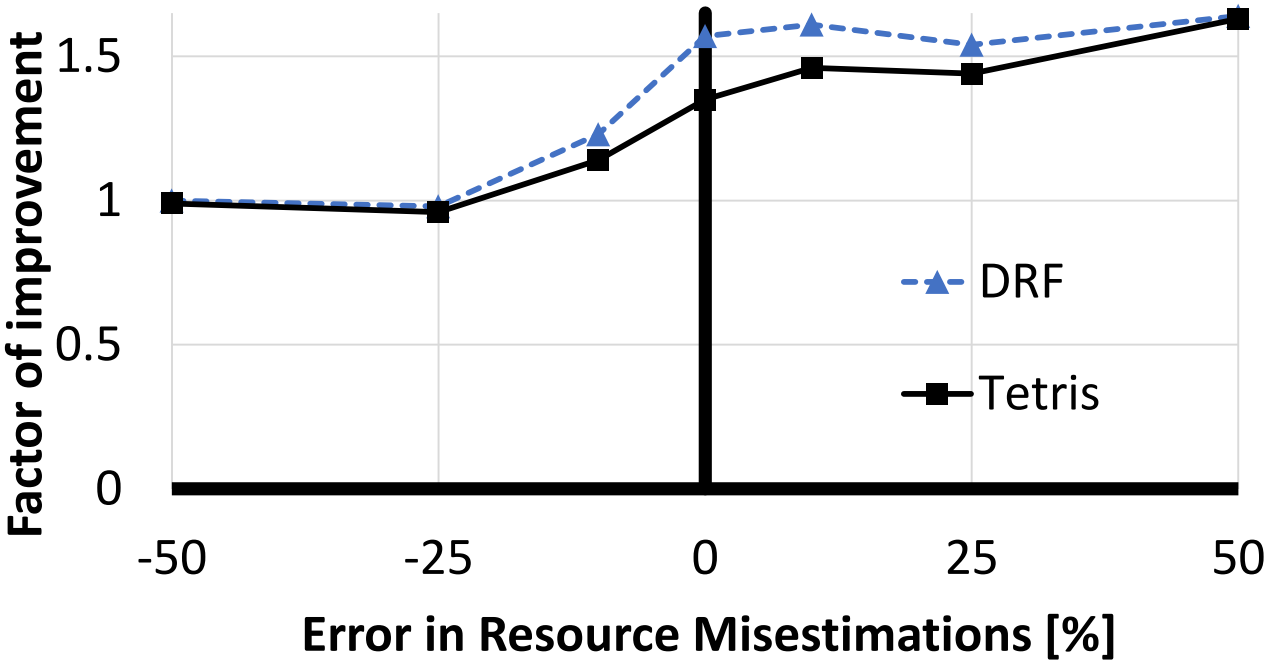
Backup slides

Impact of Altruism Levels



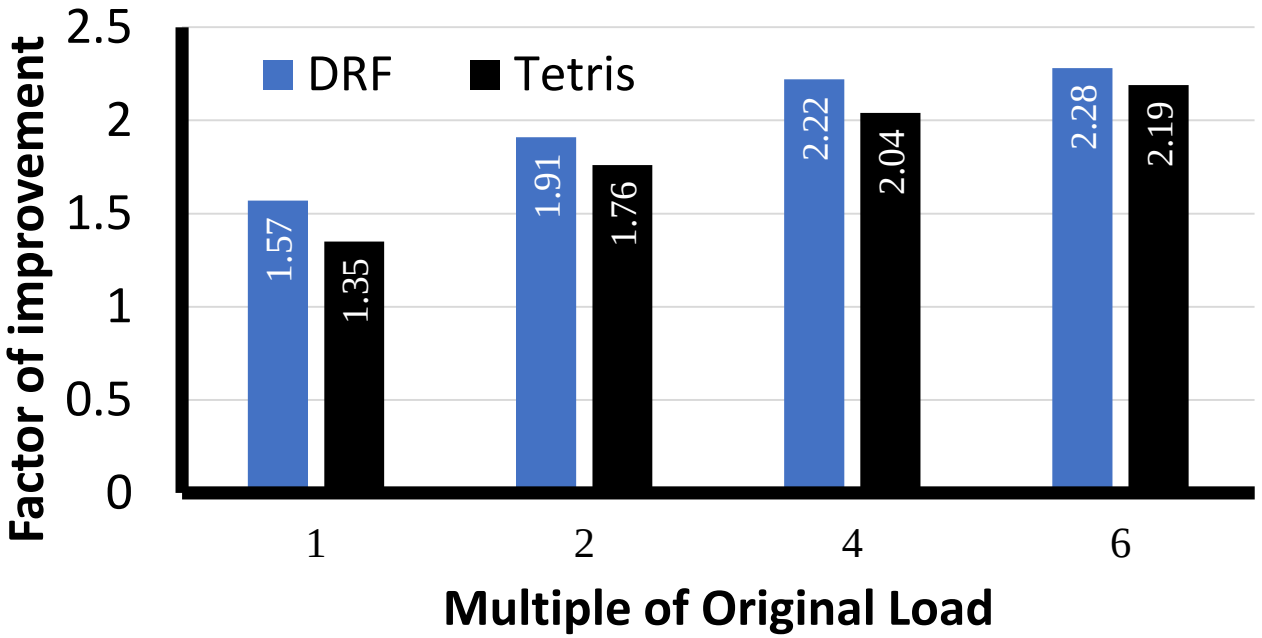
- Increasing levels of altruism increase performance
- Comparable when P(Altruism) == 0

Impact of Misestimations



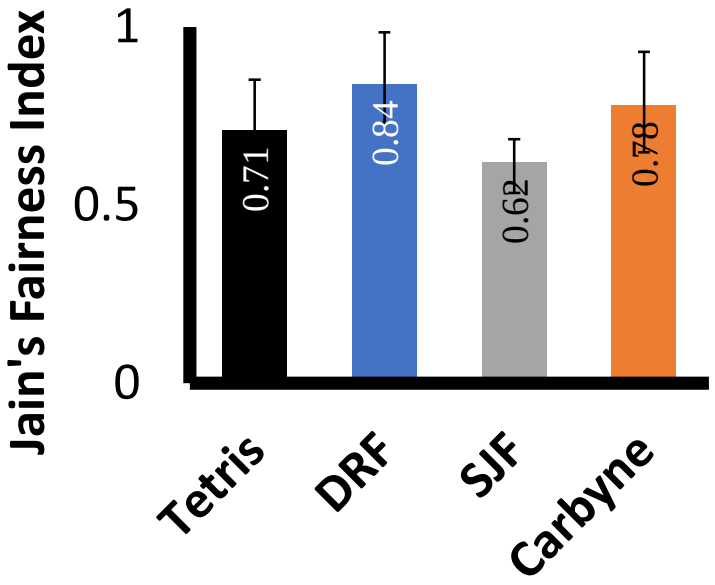
- Consistently better performance
- Comparable when resources are underestimated

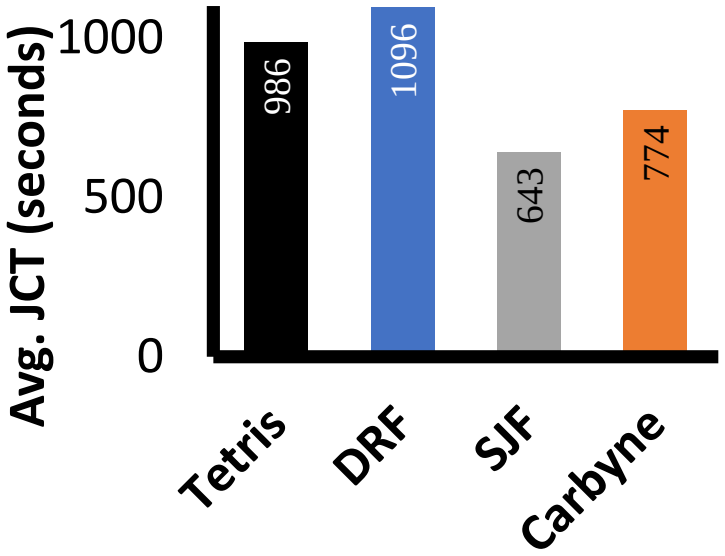
Impact of Contention

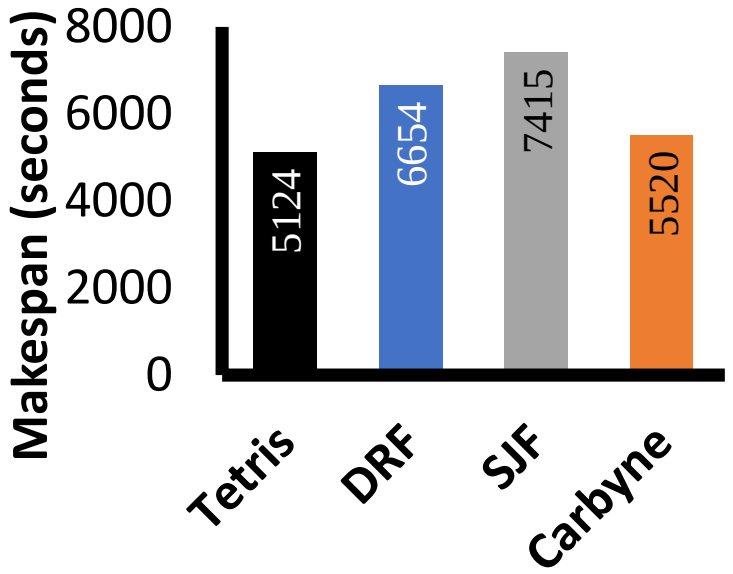


- More contention increases the need for carefully rearranging tasks
- Too much contention saturates the cluster; not much room for leftover allocations

Fairness vs. Performance vs. Efficiency - Online







- Even in online case, Carbyne comes closely to the best metric in each metric

Data Locality vs. Straggler Mitigation vs. Task Failures

Data Locality

- Altruistically giving up resources for data-local task may have adverse effects
- An altruistically delayed data-local task is likely to find data locality when it is eventually scheduled

Straggler Mitigation

- Likely to prioritize speculative tasks during leftover scheduling because it selects jobs in the SRTF order

Handling Task Failures

- Does not distinguish between new and restarted tasks
- in case of task failures, it has to recalculate the expected completion time for the job