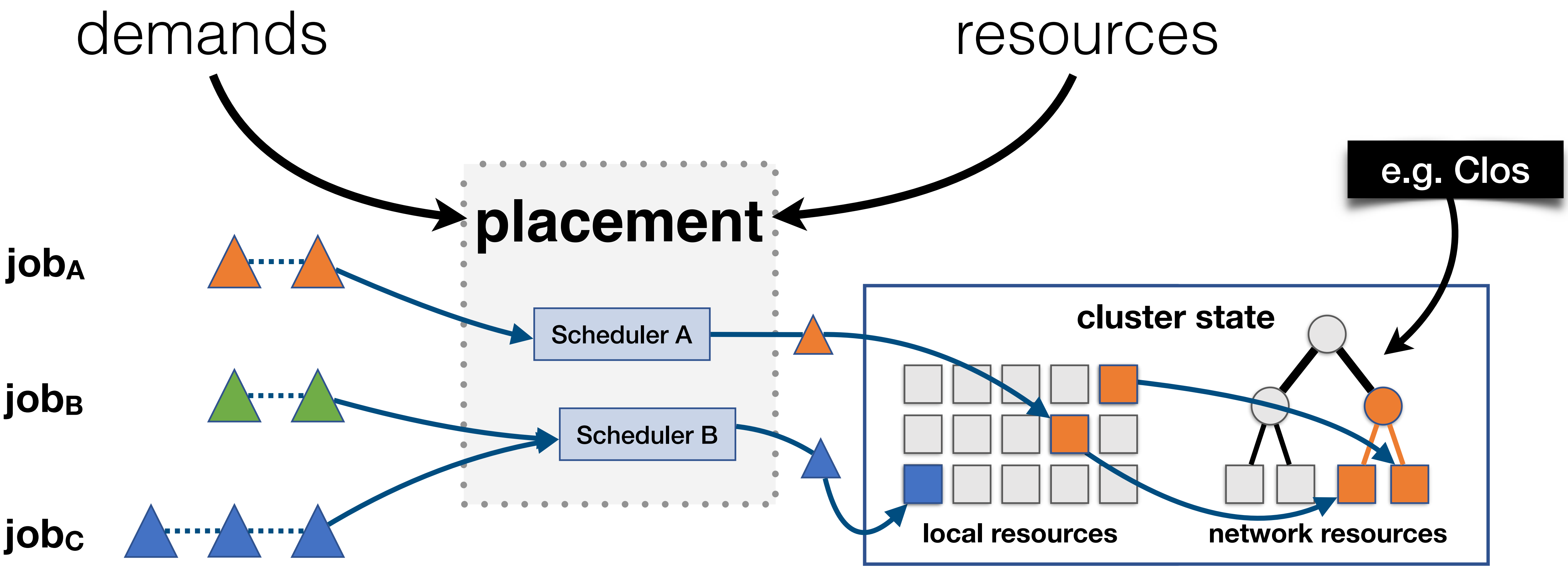
The Grand CRU Challenge

Marcel Blöcher[1], Malte Viering[1],
Stefan Schmid[2], and Patrick Eugster[1&3]

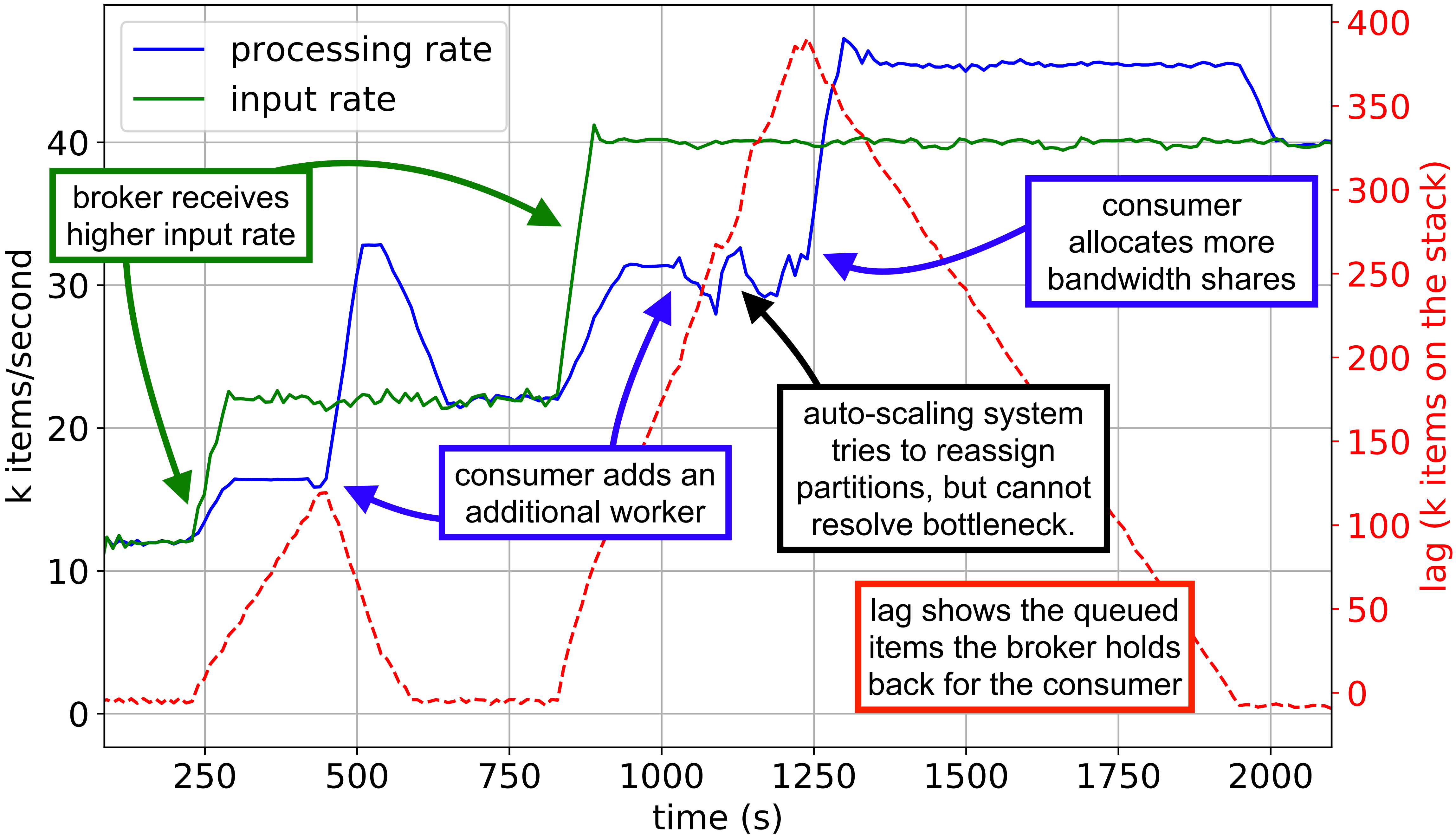
[1]TU Darmstadt, Germany
[2]Aalborg University, Denmark & TU Berlin, Germany
[3]Purdue University, USA

Cluster Resource Scheduling



Scheduling information is distributed!

Why bother?



broker receives higher input rate

consumer adds an additional worker

auto-scaling system tries to reassign partitions, but cannot resolve bottleneck.

consumer allocates more bandwidth shares

lag shows the queued items the broker holds back for the consumer

Two-dimensional resource scaling: an Apache Kafka streaming case study

Cluster **R**esource **U**tilization

What is required for taking informed resource scheduling decisions?

CRU dilemma

Without knowledge of both roles' information, scheduling decisions are likely to be suboptimal.

But both options speak against a clear separation

Application Information

- performance goals
- resources usage

enrich the application

applications *learn more about the underlying infrastructure ➜ schedule an entire "graph" of containers*

Cluster Information

- node-local resources
- network resources

enrich the resource manager

resource manager *understand more of the applications' semantics and performance goals*

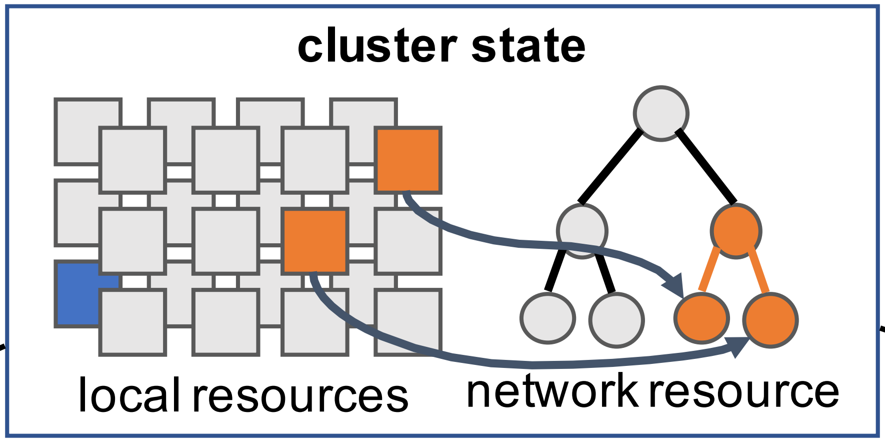
The Grand CRU Challenge

Idea: Share slightly more information but

- respect separation of different roles
- naïve approach (expose all information) becomes combinatorial and expensive
- resources are different in nature - shared vs local resources

Challenge: Find a cluster scheduling architecture which provides efficient information sharing mechanisms

Multi-Dimensional Scheduling

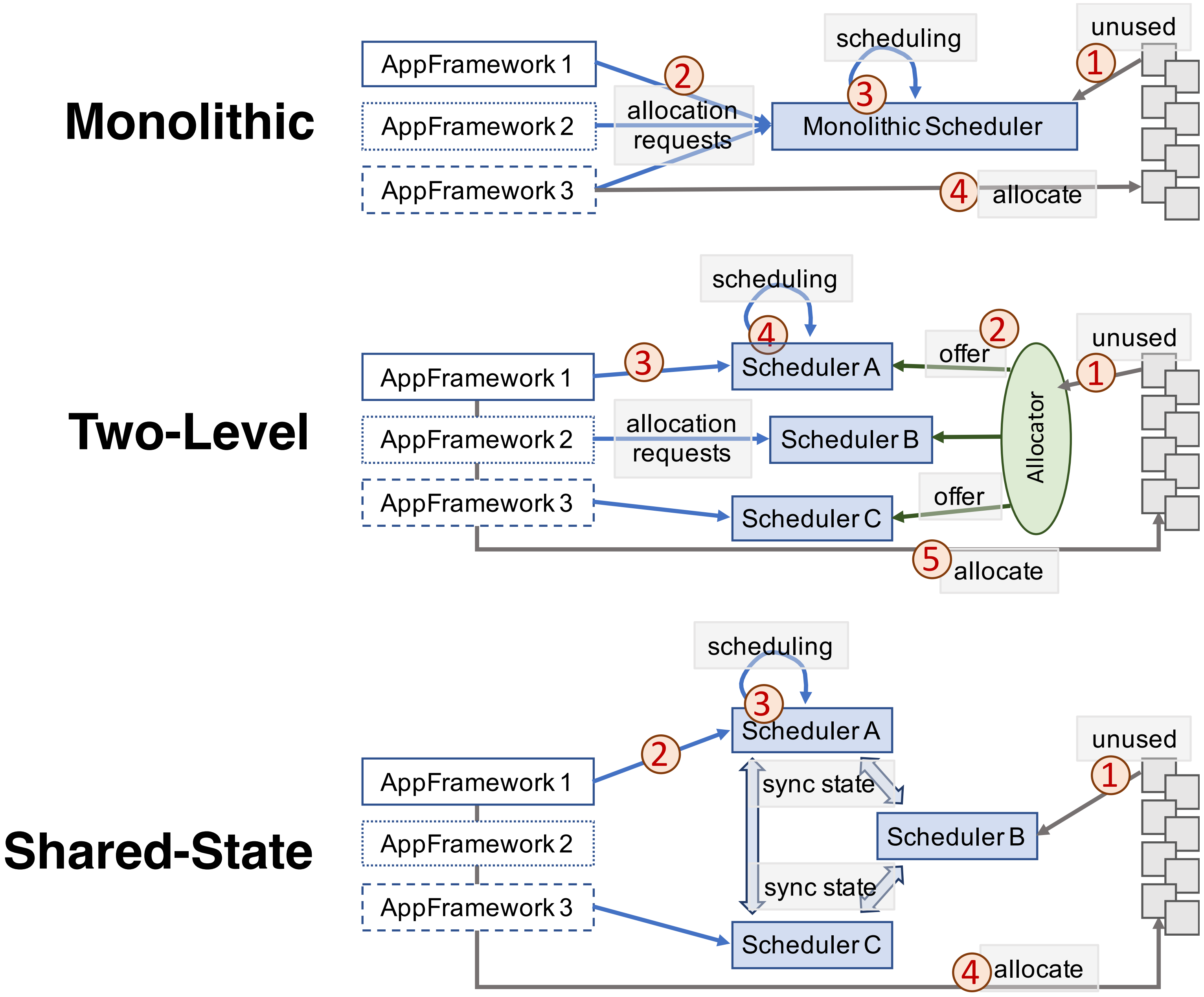


<u>local</u> resources can be handled in an <u>isolated</u> fashion

network resources are <u>shared</u> and allocation is <u>intertwined</u> with that of local resources

<u>What are the consequences for the scheduler architecture?</u>

Design Space: Scheduling Architectures



Monolithic

Two-Level

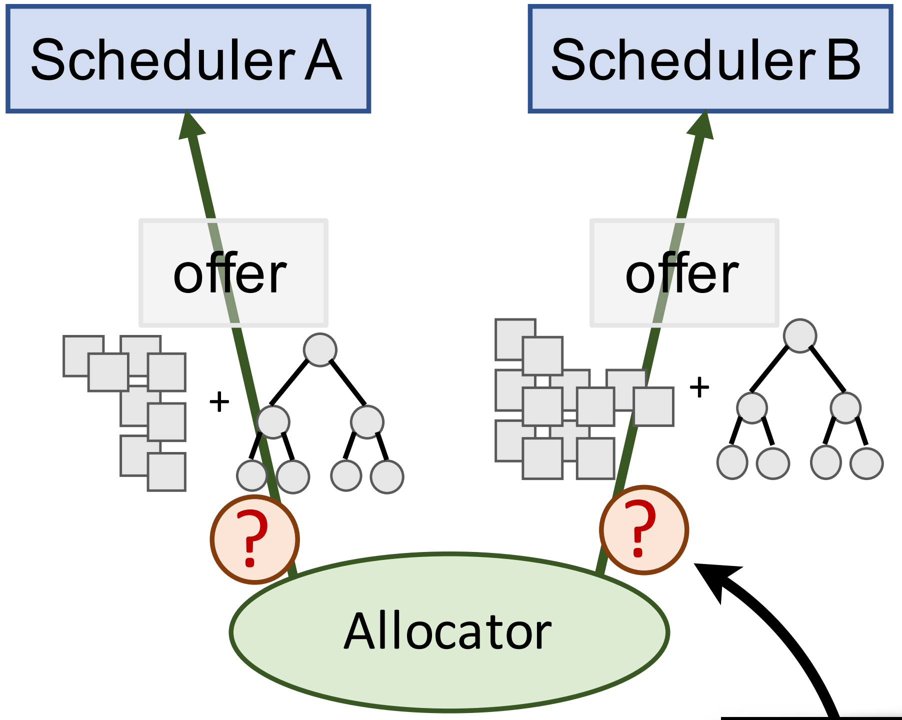
Shared-State

CRU Dilemma - Evaluation
Two-Level Architecture

1. Resource Hoarding Issue

2. Resource Offer Conflict

➜ *next slide*

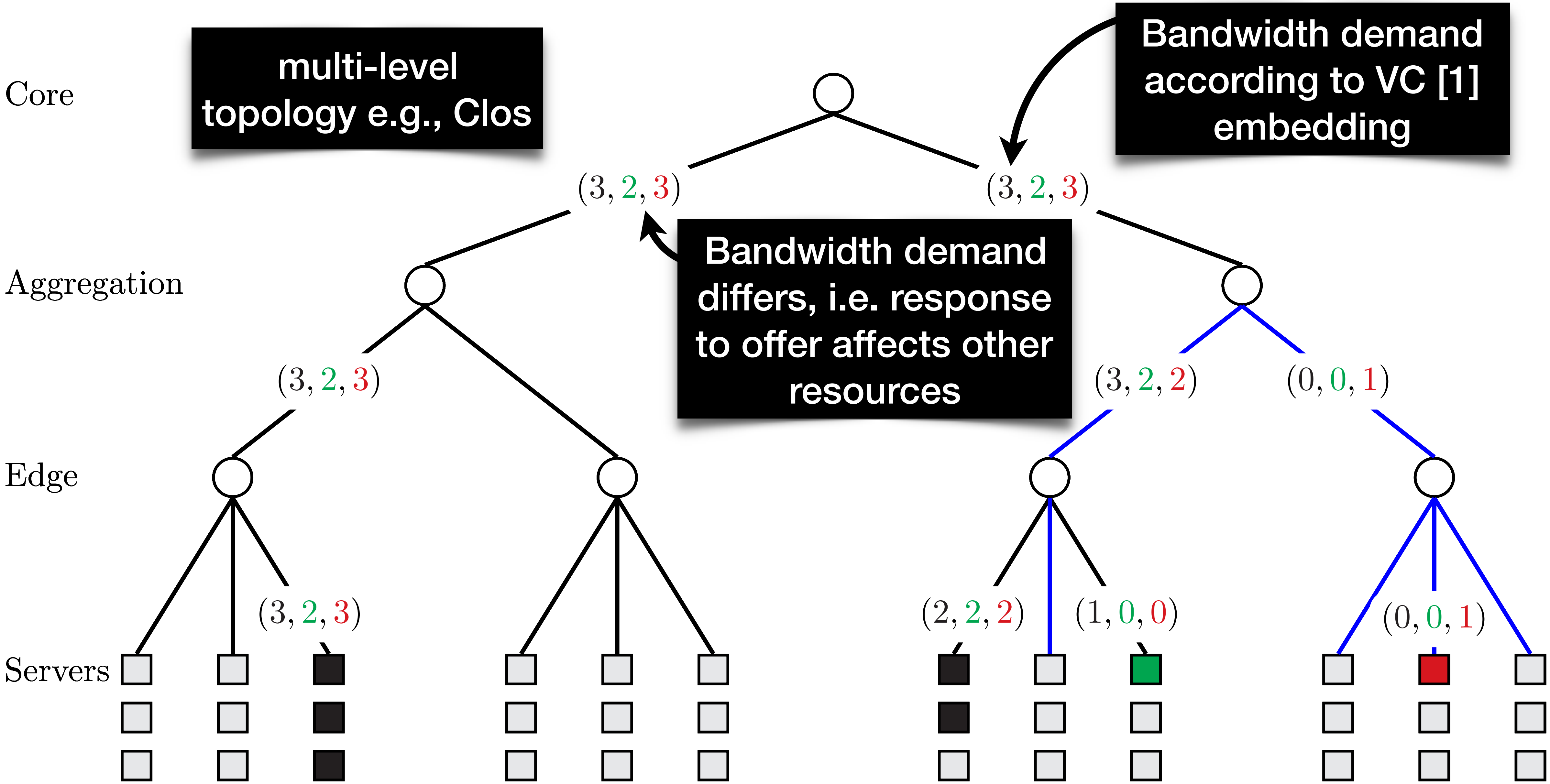


network shares in an offer affect more than a single node

Resource Offer Conflict

T1 a job runs 6 tasks
T2 blue offer, spawn new task

T3 in the meantime, a tasks finishes
T4 response to offer



multi-level topology e.g., Clos

Bandwidth demand according to VC [1] embedding

Bandwidth demand differs, i.e. response to offer affects other resources

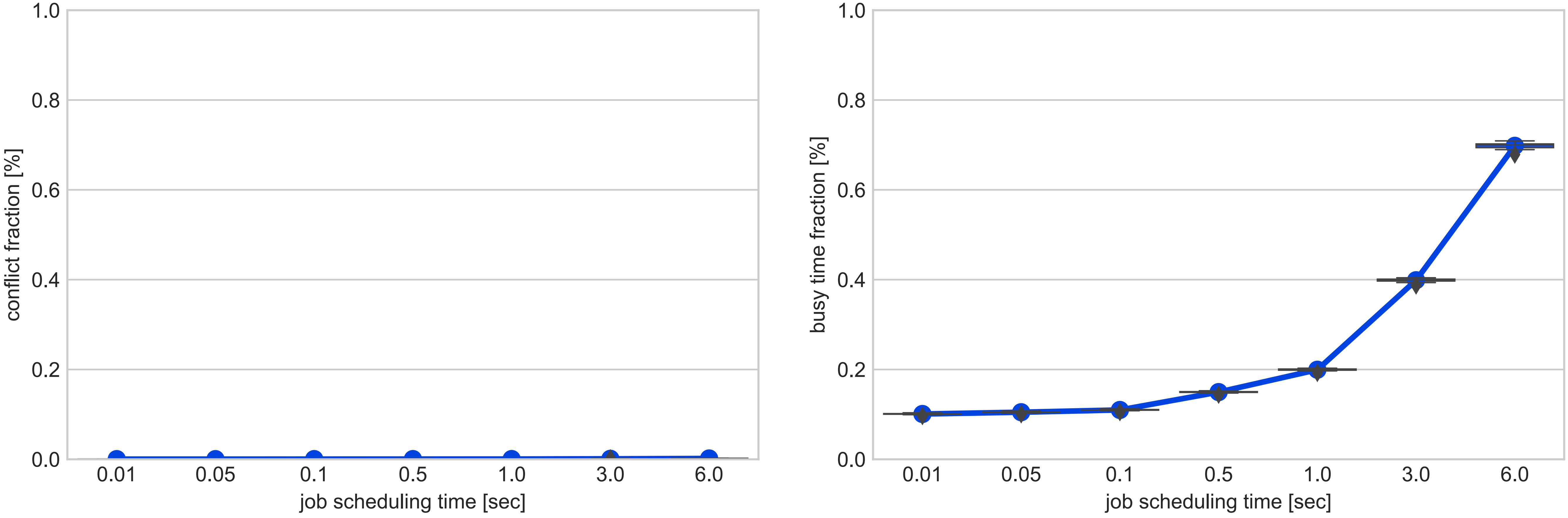
[1] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. 2011. To- wards predictable datacenter networks. In ACM SIGCOMM.

CRU Dilemma - Evaluation Shared-State Architecture

- Simulation based evaluation
  - modified Omega simulator, network perspective added
  - each job’s task ➜ VC bandwidth demand
  - 6000 node Fat-Tree, avg. 200 tasks per job
  - 2 schedulers running simultaneously
- Metrics
  - scheduler busy time
  - conflict fraction of scheduling decisions

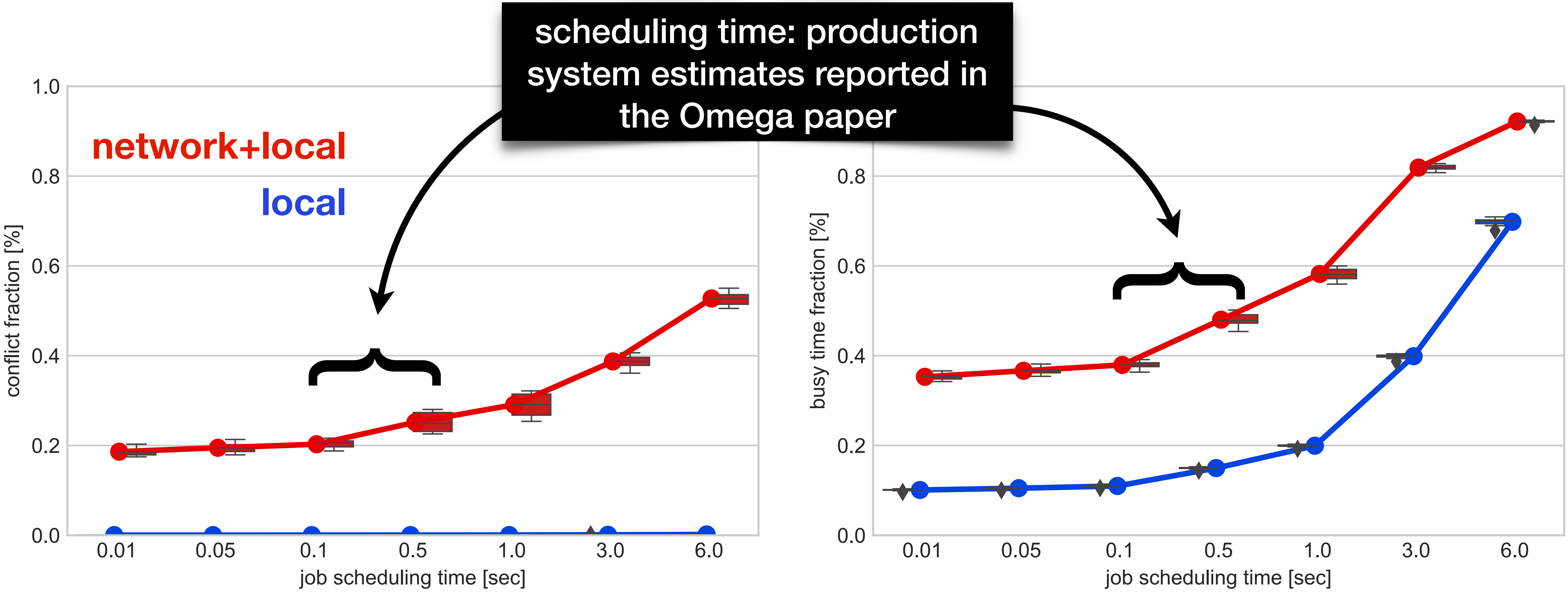
Shared-State Scheduler

Experiment A: only **node-local** resources



Shared-State Scheduler

Experiment B: **network+local** resources



scheduling time: production system estimates reported in the Omega paper

Conclusion

- We make the case for multi-dimensional resource scheduling
- Scheduling information is distributed ➡ CRU dilemma

Open Question - Grand CRU Challenge:
How to maximize CRU when networking enters the picture?

	Issue	**Core Design Principle**
Monolithic	does not scale / multi-path issue	single point which holds all information
Two-Level	too pessimistic	distributed, by small disjoint information shares
Shared-State	too many conflicts	distributed write access by conflict resolution

None of the investigated architectures tackles the CRU dilemma

We advocate an architecture that combines all three design principle

Thank You

contact: bloecher@dsp.tu-darmstadt.de

Marcel Blöcher[1], Malte Viering[1],
Stefan Schmid[2], and Patrick Eugster[1&3]

[1]TU Darmstadt, Germany
[2]Aalborg University, Denmark & TU Berlin, Germany
[3]Purdue University, USA