Multi-Resource Packing for Cluster Schedulers

Robert Grandl
Aditya Akella

Srikanth Kandula
Ganesh Ananthanarayanan
Sriram Rao

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Microsoft Research

Diverse Resource Requirements

Tasks need varying amounts of each resource

- E.g., Memory [100MB to 17GB]
  CPU [2% of a core to 6 cores]

Demands for resources are not correlated

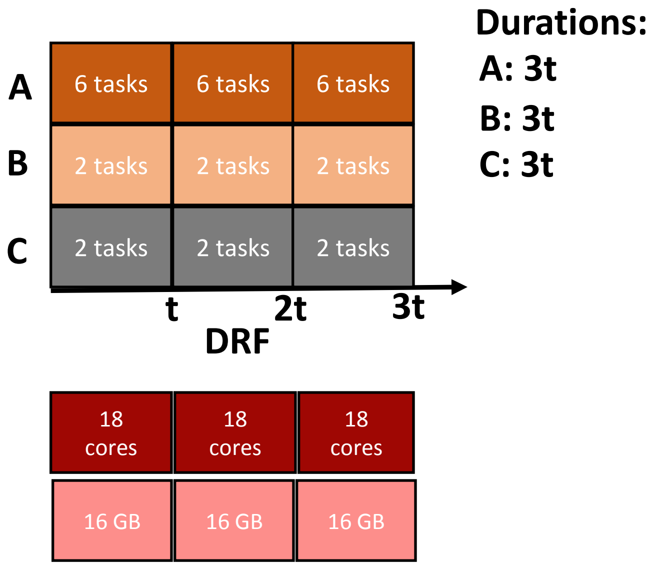
- Correlation coefficient across resource demands $\in$ [–0.11, 0.33]

Need to match tasks with machines based on resource

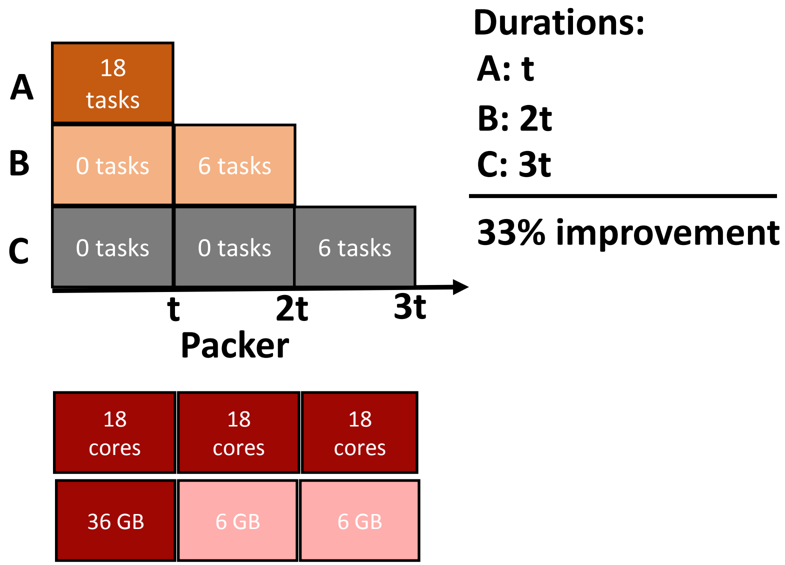
Current Schedulers do not Pack

Slots allocated purely on fairness considerations

Cluster [18 Cores, 36 GB] / Job: [Task Prof.], # tasks	
A	[1 Core, 2 GB], 18
B	[3 Cores, 1 GB], 6
C	[3 Cores, 1 GB], 6



Resources used: DRF share = 1/3



Resources used: Packer

Current Schedulers

Packer Schedulers

It is all about packing ?

Multi-dimensional bin packing is NP-hard for #dimens. ≥ 2

- Several heuristics proposed
- But they do not apply here …

  size of the ball, contiguity of allocation,
  resource demands are elastic in time

Will perfect packing suffice ?

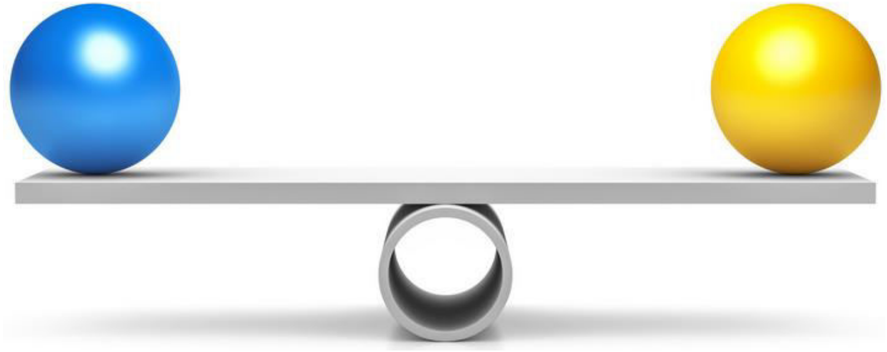
Competing objectives:

Cluster utilization vs.
Job completion times vs.
Fairness

Intuition behind the solution

Something reasonably simple and which can be applied

Cluster efficiency Job completion time



Performance Fairness

Tetris

Pack tasks along multiple resources

- *Cosine similarity between task demand vector and machine resource vector*

A

Multi-resource version of SRTF

- *Favor jobs with small remaining duration and small resource consumption*

T

Incorporate Fairness

- *Fairness knob $\in (0, 1]$*
  $f \rightarrow 0$ close to perfect fairness
  $f = 1$ most efficient scheduling

F

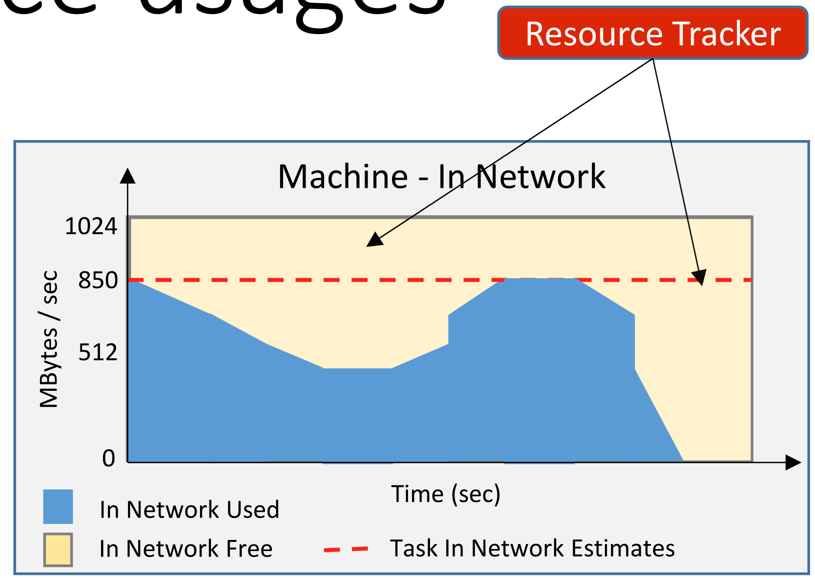
(simplified) **Scheduling procedure**

```
while (resources R are free)
 among ⌈FJ⌉ jobs furthest from fair share
   score (j) =
          max task t in j, demand(t) ≤ R A(t, R) + ε T(j)
 pick j*, t* = argmax score(j)
 R = R – demand(t*)
end while
```

Task Requirements and resource usages

Learning task requirements

- From tasks that have finished in the same *phase*
- Coefficient of variation $\in$ *[*0.022, 0.41]
- Collecting *statistics from recurring* jobs
- Peak usage demands estimates for tasks



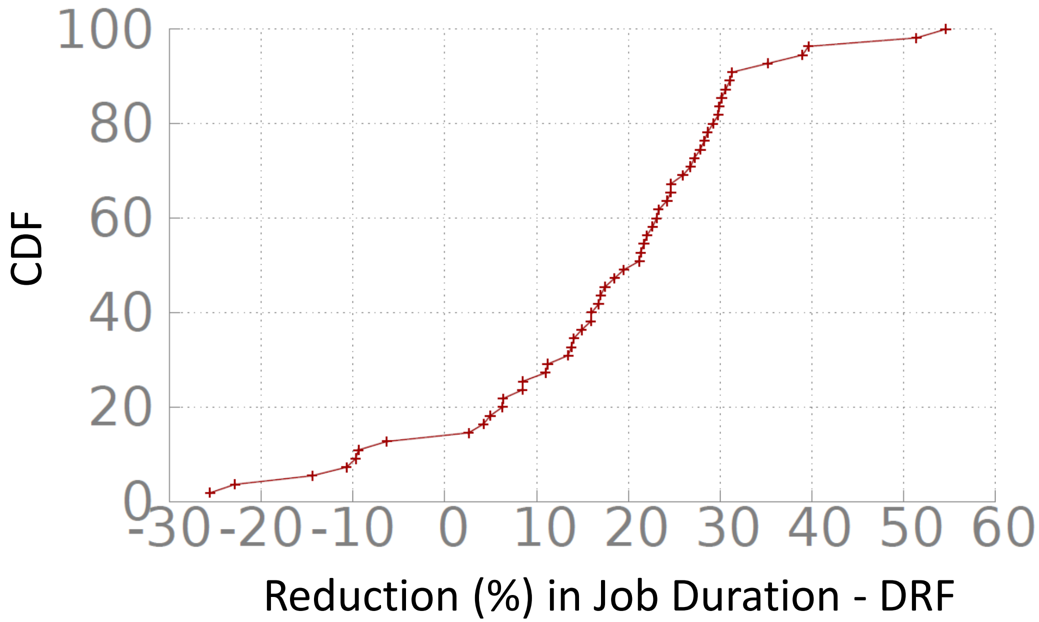
Resource Tracker

- measure actual usage of resources
- enforce allocations
- aware of activities on the cluster other than tasks assignment: *ingest and evacuation*

Evaluation

Prototype atop Hadoop 2.3

- Tetris as a pluggable scheduler to RM
- Implement RT as a NM service
- Modified AM/RM resource allocation protocol



Large scale evaluation

Cluster capacity: 250 Nodes

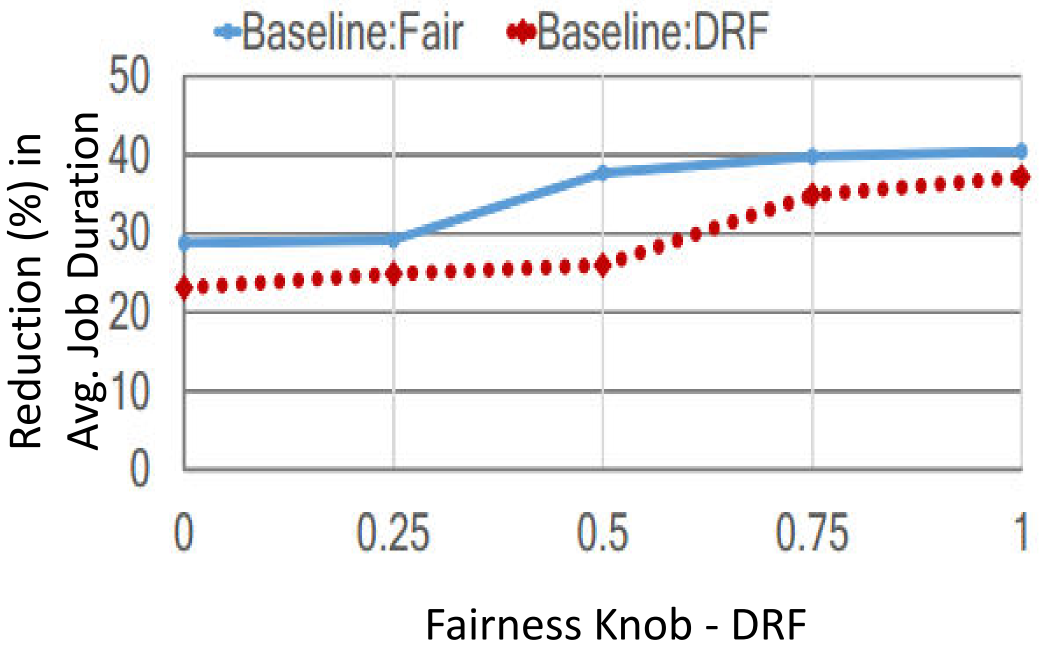
4 hour synthetic workload

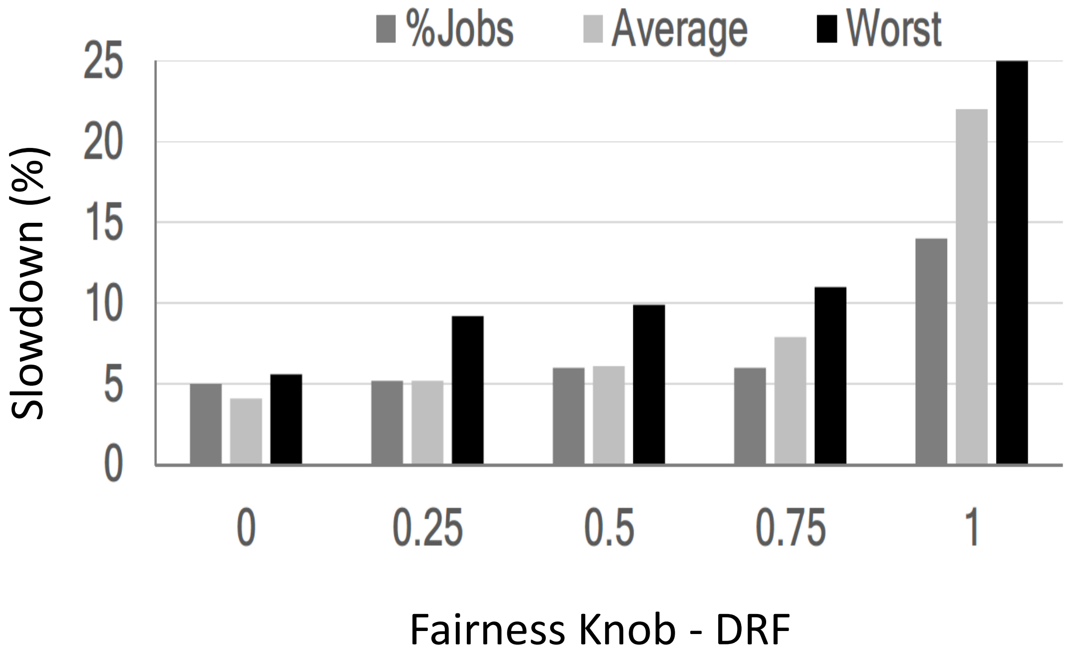
60 jobs with complementary task demands

Reduces average job duration by up to 40%

Reduces makespan by 39%

Evaluation





Trace-driven simulation

Facebook production traces analysis

Fairness knob: fewer than 6% of jobs slow down; by not more than 8% on average

Knob value of 0.75 offers nearly the best possible efficiency with little unfairness

Conclusion

Identify the importance of scheduling all relevant resources in a cluster

Resource Fragmentation + Over-allocation and Interference

New scheduler that pack tasks along multiple resources

Reduce makespan + Job Completion Time

Enable a trade-off between packing efficiency and fairness

Fairness Knob

Come and see our poster !