Tetris: Multi-Resource Packing for Cluster Schedulers

Robert Grandl*, Ganesh Ananthanarayanan†, Srikanth Kandula†,
Sriram Rao†, Aditya Akella*

**University of Wisconsin, †Microsoft Research*

Motivation

Diverse Resource Requirements

Tasks in modern data parallel clusters have highly diverse resource requirements along CPU, memory, disk and network

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

Any of these resources may become bottlenecked

- *Demand for different resources are not correlated*

	Cores	Memory	Disk	Network
Cores	—	0.33	0.22	0.29
Memory	—	—	−0.11	0.04
Disk	—	—	—	0.26
Network	—	—	—	—

	Cores	Memory	Disk	Network
Cores	—	0.41	0.12	0.23
Memory	—	—	0.28	−0.1
Disk	—	—	—	−0.07
Network	—	—	—	—

Correlation matrix of task resource demands for Bing(left) and Facebook(right).

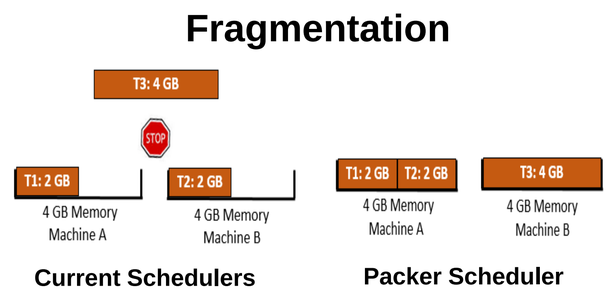
- *Utilization of different resources peaks at different times*

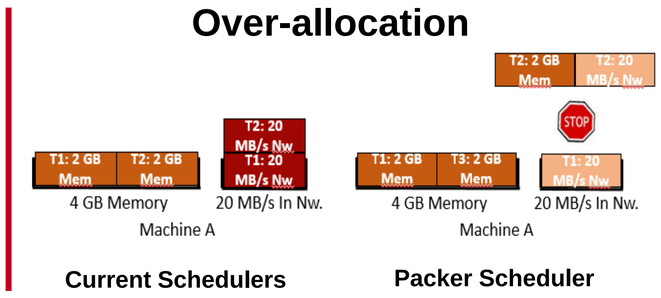
	> 75% used	> 90% used	> 95% used
CPU	0.58	0.35	0.28
Memory	0.68	0.41	0.22
Disk in	0.11	0.02	0.003
Disk out	0.26	0.04	0.006
Network in	0.22	0.01	0.008
Network out	0.44	0.28	0.05

Tightness of resources. Probability that a type of resource is used at above a certain fraction of its capacity in the Facebook cluster.

Current Schedulers Do Not Pack

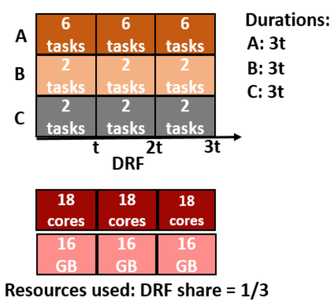
Today's schedulers allocate resources to tasks in units of slots, each slot corresponding to some amount of memory or cores. Slots based allocation leads to several problems.

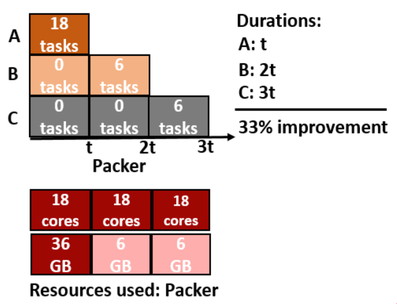




Slots allocated purely on fairness considerations

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





Given such diversity, we seek to build *a* cluster scheduler that ***packs tasks*** to machines based on their requirements of multiple resources so as to increase cluster efficiency. Our objective in packing is not only to ***maximize the task throughput*** but also to ***speed up job completions***. While ***fair allocations*** do not improves cluster efficiency, a practical solution should enable it.

Tetris

Multi-dimensional bin-packing problem

- APX-hard for more than two dimensions

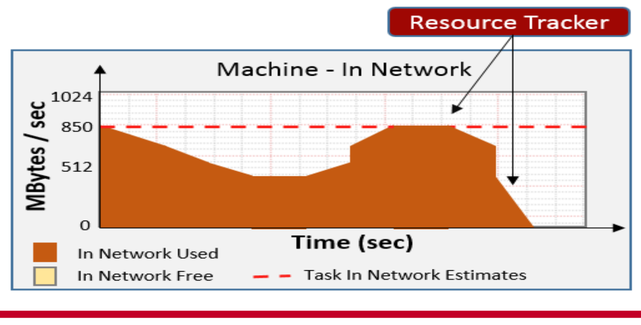
- Several heuristics proposed but they do not apply
  size of the ball, contiguity of allocation, resource demands are elastic in time

Competing objectives in practice

Cluster utilization vs. Job completion times vs. Fairness

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



Improves Cluster Efficiency

Pack tasks along multiple resources

Cosine similarity between task demand vector and machine resource vector — Score A

Improves Job Completion Time

Multi-resource version of SRTF

Favor jobs with small remaining duration and small resource consumption — Score T

Incorporate Fairness

Fairness knob $\in (0, 1]$

$f \to 0$ close to perfect fairness

$f = 1$ most efficient scheduling

Score F

(simplified) **Scheduling procedure**

```
1: while (resources R are free)
2:  among ⌈FJ⌉ jobs furthest from fair share
3:    score (j) =
4:          max task t in j, demand(t) ≤ R A(t, R) + ε T(j)
5:  pick j*, t* = argmax score(j)
6:  R = R − demand(t*)
7: end while
```

Evaluation

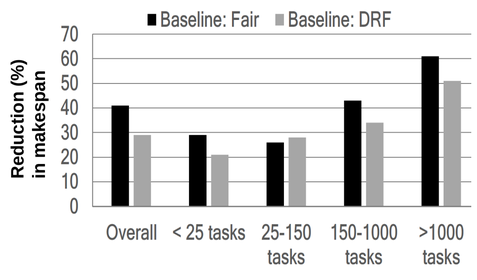
Prototype atop Hadoop Yarn 2.3

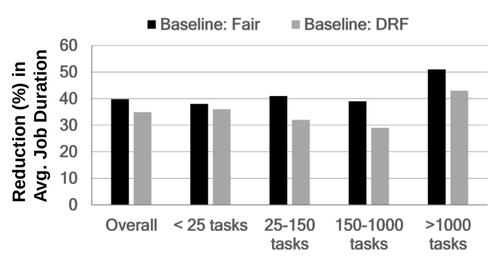
Large scale evaluation

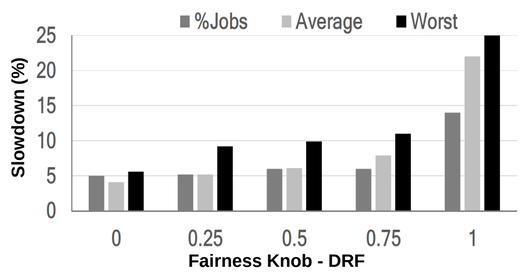
- Cluster capacity: 250 nodes
- 4 hour synthetic workload

Trace-driven simulation

- Facebook production traces analysis







Speeds up jobs by 40% and 35%(Fair, CS)
Reduces makespan by 41% and 29%(Fair, DRF)
Fairness knob: fewer than 6% of jobs slow down