Master Seminar August, 11

USENIX NSDI 2012

PACMan: Coordinated Memory Caching for Parallel Jobs

Ganesh Ananthanarayanan, Ali Ghodsi, Andrew Wang, Scott Shenker, Ion Stoica

University of California, Berkeley

Srikanth Kandula

Microsoft Research

Dhruba Borthakur

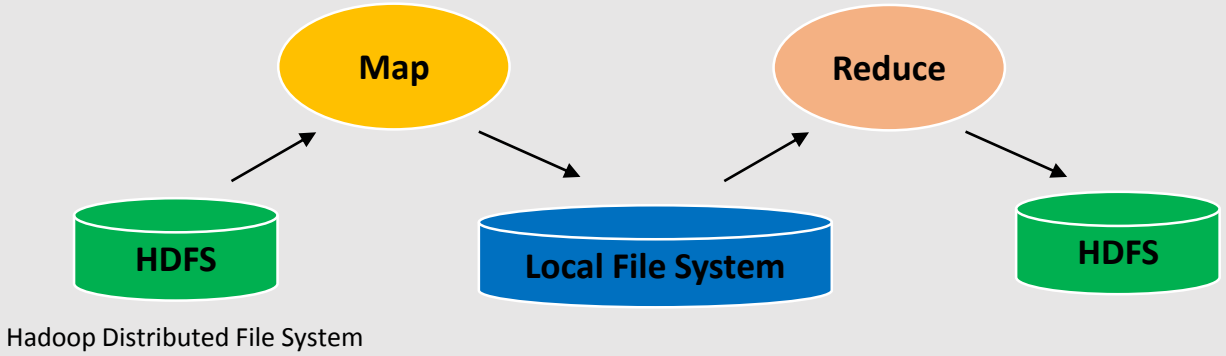
Facebook

Presented By HaeJoon Lee

1 Motivation

Rapid innovation in cluster computing frameworks

IO-intensive phase 79% of job's duration, consumes 80% resources [Facebook trace]



1 Motivation

- **Underutilization memory [Facebook trace]**

  -Median : 95th percentile (memory utilization) = 19% : 42%

- **Possible to save big data in memory ?**

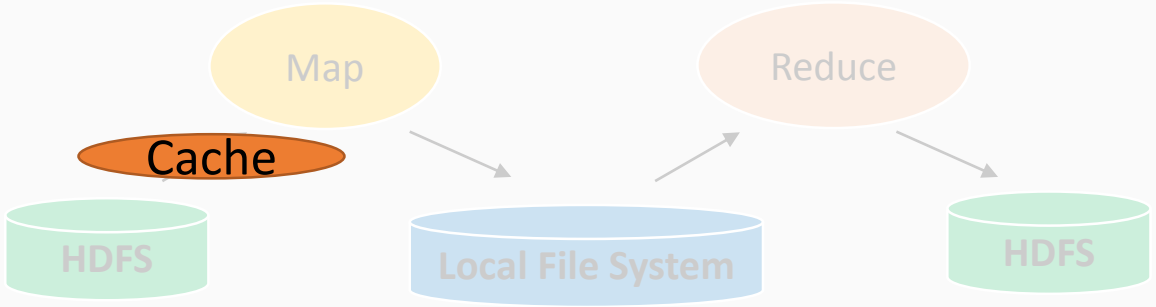
  -96% of a job in Facebook can have their data fit in 32GB memory.

- **Same file is frequently accessed**

2 Where to Go

By caching in memory
Reduction in completion time,
Improvement of cluster efficiency

Slot: the number of resources on which tasks execute in distributed system



Outline

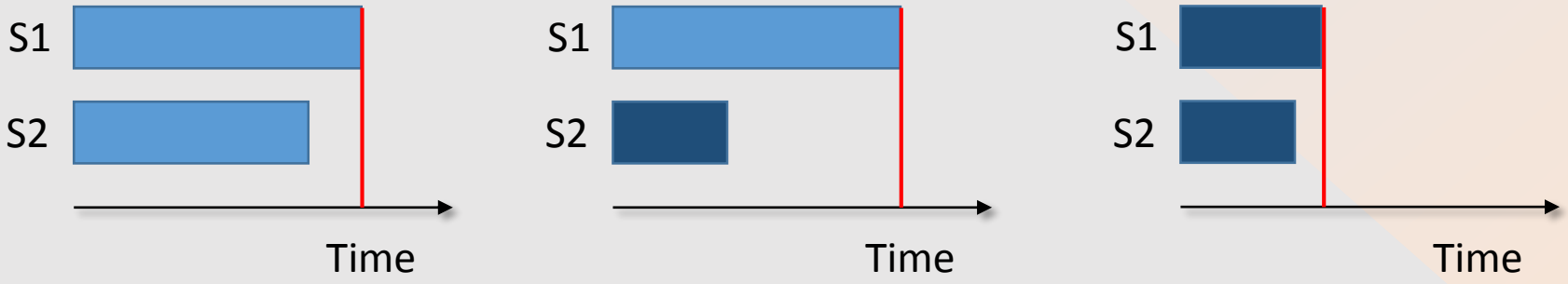
PACMan Ideas & Architecture

Implementation

Results

Conclusion

3 All or Nothing Property

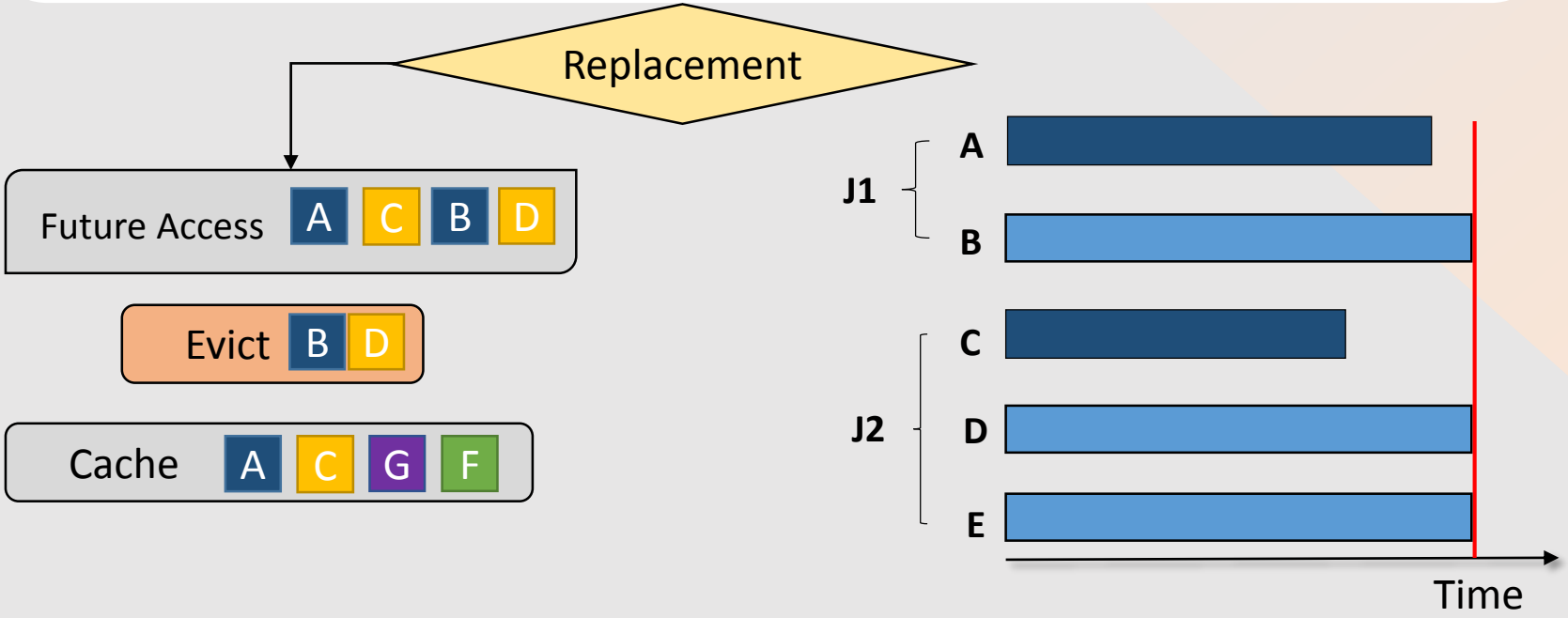


Job speeds up only if the inputs of all tasks are cached

3 Traditional Policy

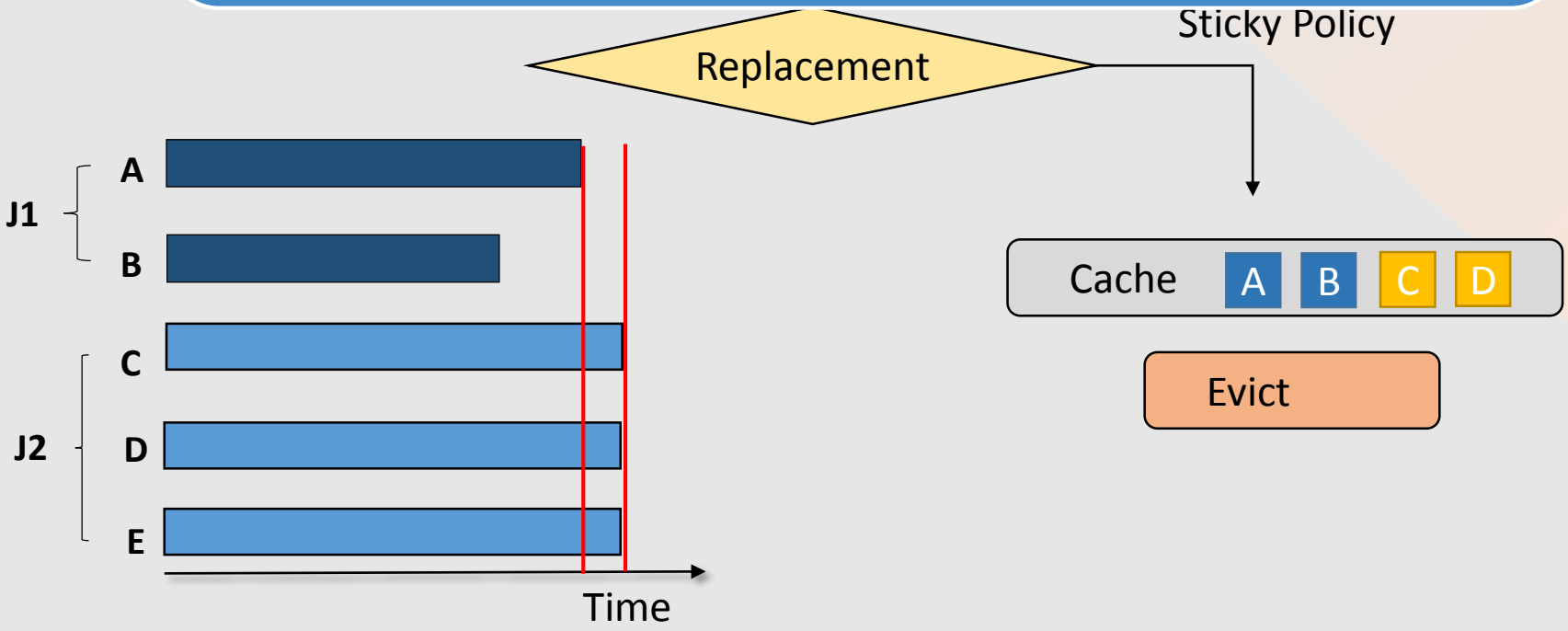
MIN replacement to increase cache hit ratio

No reduction in completion time,
Maximizing cache hit ratio is insufficient



3 Sticky Policy of PACMan

J1 Reduction in completion time



3 Sticky Policy of PACMan

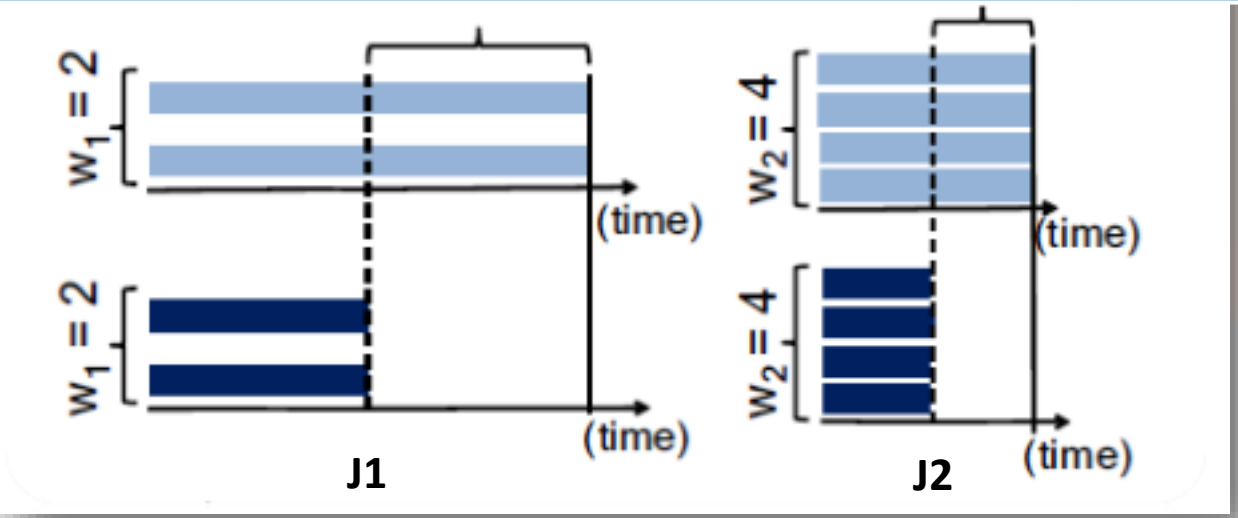
- Conventional policy: MIN
  - Doesn’t consider constraint of all-or-nothing parallel jobs
  - Result in no reduction in completion time

- Developed policy: Sticky
  - Considers constraint of all-or-nothing parallel jobs
  - Rids a block of incomplete file in cache
  - LIFE & LFU-F cache replacement based on stick policy

3 LIFE for reduction in completion time

Wa ob

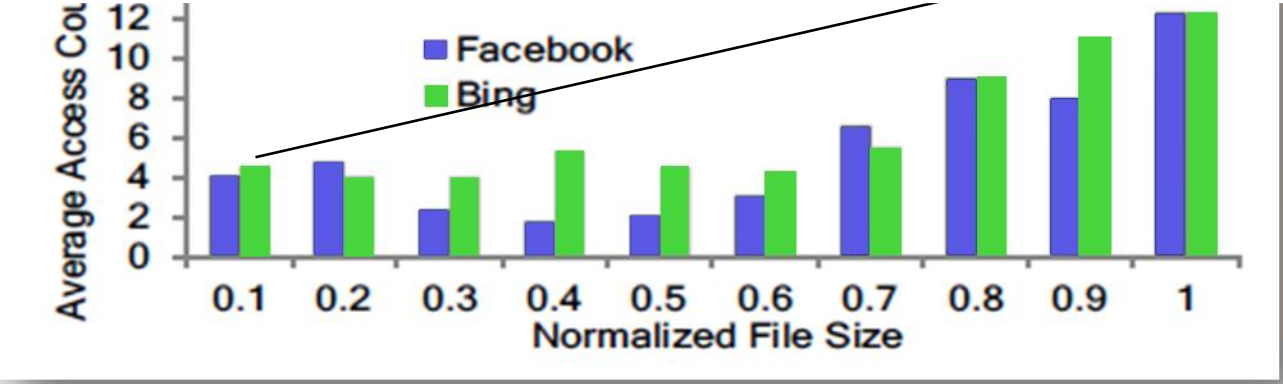
Reduction in average completion time by favoring jobs with smallest wave-width



Gains in average completion time due to caching increase as wave-width decreases.

3 LFU-F for Improvement Efficiency

Retaining frequently accessed files maximizes efficiency of the cluster



The larger file is, the more the file consumes resource.
The larger file is, the more the file is accessed.

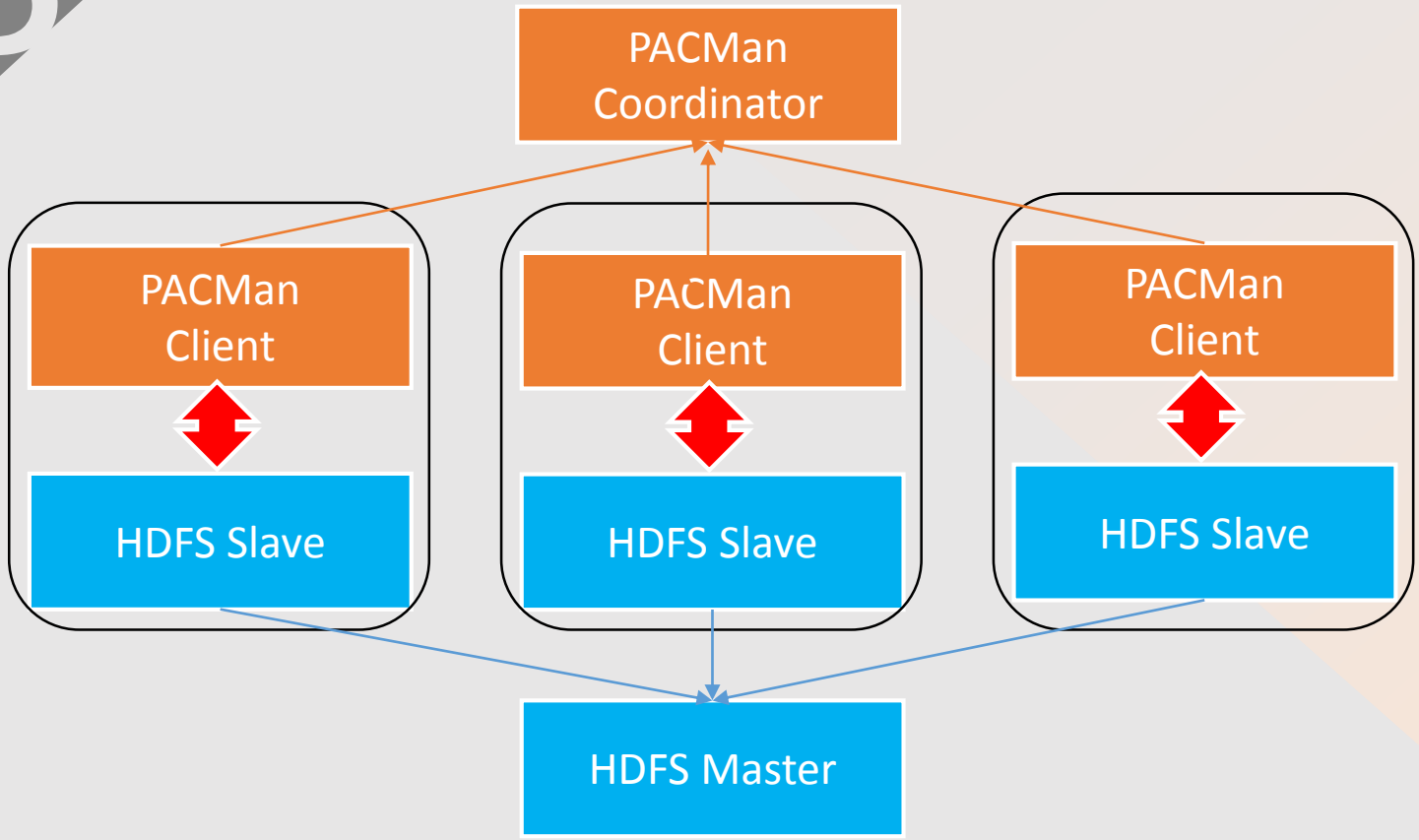
More accessed file -> Larger file -> Saving much resource

3 Policy- LIFE & LFU-F within PACMan

- Completion time policy: LIFE
  - Evicts from incomplete file with highest wave-width
  - Favors smaller wave-width files
  - Sticky: fully evicts file before going to next file (all-or-nothing)

- Improvement of efficiency: LFU-F
  - Evicts from incomplete file with lowest frequency
  - Favors higher frequency files
  - Sticky: fully evicts file before going to next file (all-or-nothing)

3 PACMan Architecture



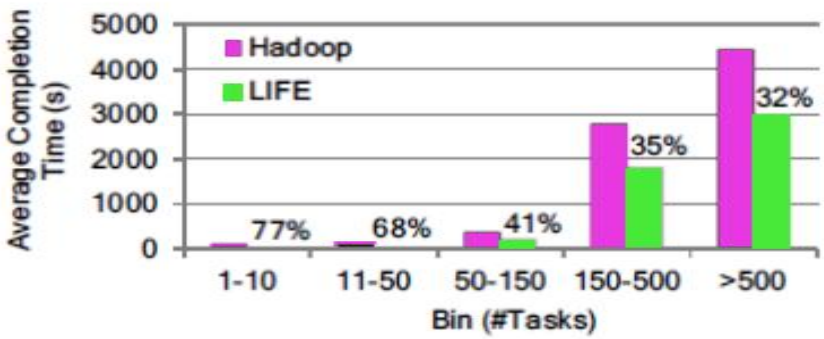
- Client updates the coordinator about the state of cache changes
- Coordinator's global view of all caches
- Memory locality by knowing where cached block is across machines

4 Experimental Setup

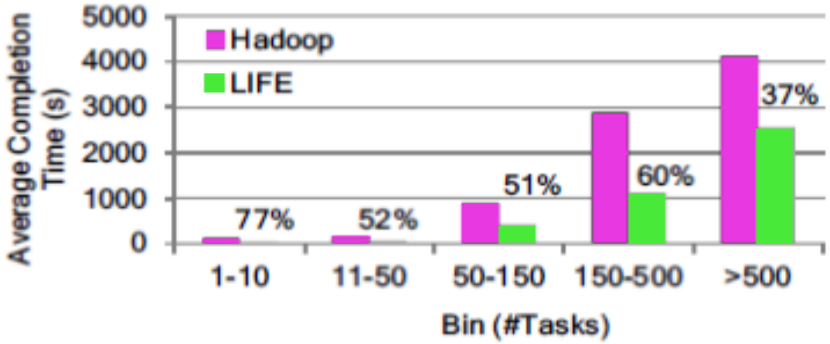
- **Workloads** - Facebook: HDFS, 150PB Input Data
  - Microsoft Bing: Cosmos, 310PB Input Data
- **Cluster** - 100 Amazon EC2 nodes with 34.2GB Memory
  - PACMan is allotted 20GB of cache per machine
- **Job Bins**

Bin	Tasks	% of Jobs		% of Resources	
		Facebook	Bing	Facebook	Bing
1	**1 – 10**	**85%**	**43%**	8%	6%
2	11 – 50	4%	8%	1%	5%
3	51 – 150	8%	24%	3%	16%
4	151 – 500	2%	23%	12%	18%
5	> 500	1%	2%	**76%**	**55%**

4 Evaluation- PACMan with LIFE



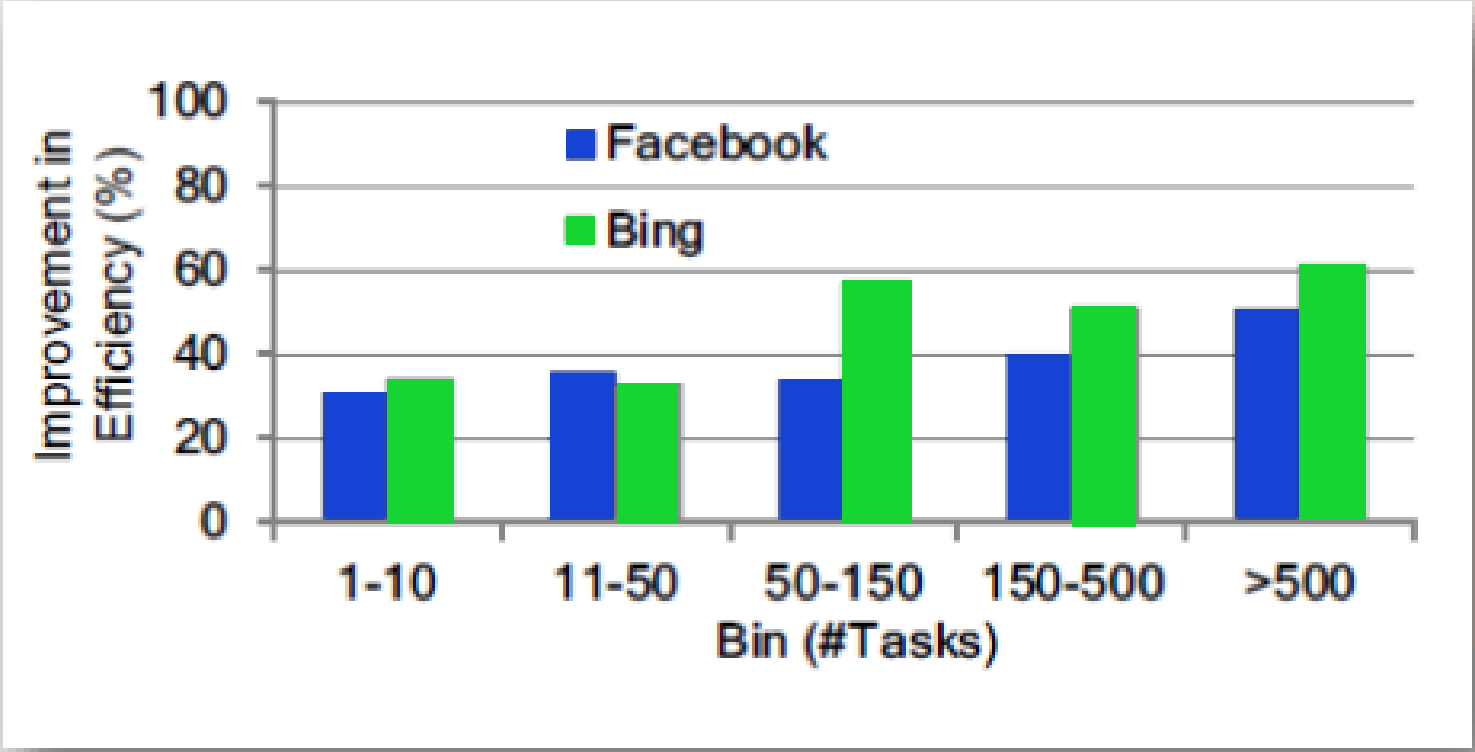
(a) Facebook Workload



(b) Bing Workload

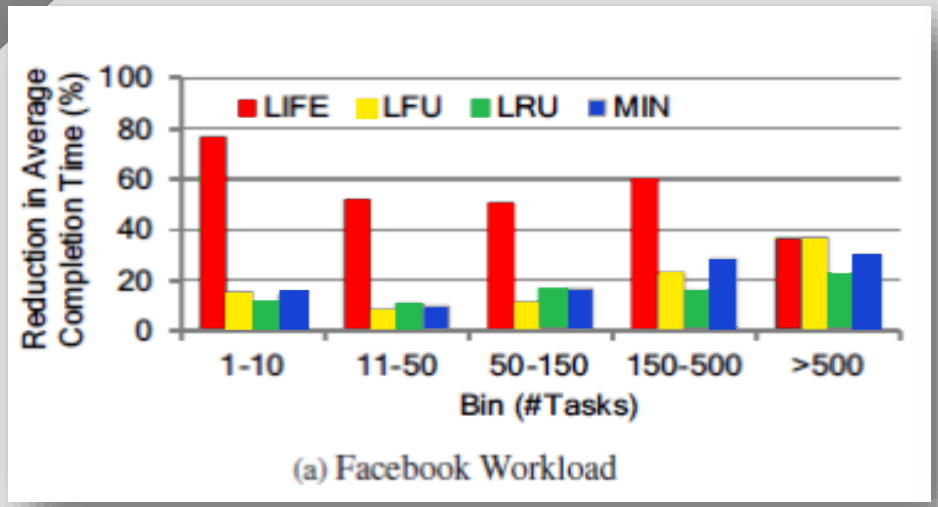
LIFE: Completion time reduced by 77 % in Bin-1

4 Evaluation- PACMan with LFU-F

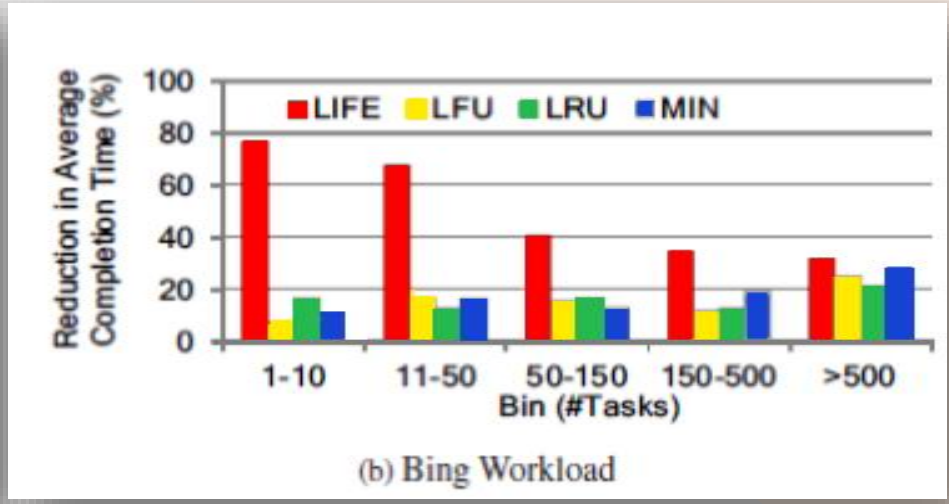


LFU-F: Improves cluster efficiency by 47 %, 54%

4 Evaluation- Comparison Policies



(a) Facebook Workload



(b) Bing Workload

LIFE is more reduced in completion time than other policies

5 Conclusion

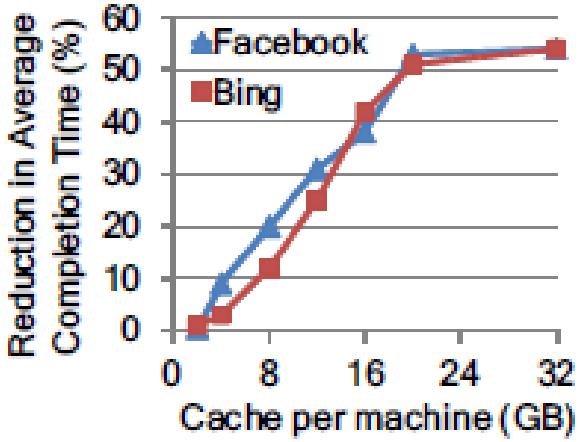
- PACMan, an in-memory coordinated caching system for parallel jobs
  >> Having the all-or-nothing property

- By PACMan Coordinator's global view to the distributed caches,
  PACMan ensures different tasks obtain memory locality

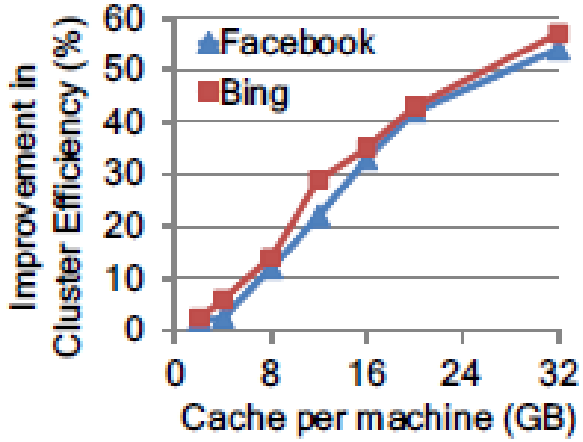
- Two cache sticky replacement policies: LIFE & LFU-F
  Reduction in completion time & Improvement of efficiency in the cluster

Thanks

6 Backup: Cache Size & Scalability



(a) LIFE



(b) LFU-F

Figure 18: **LIFE's and LFU-F's sensitivity to cache size.**

6 Backup: LIFE, reduction in completion time

LIFE

Wave-width: W
Frequency (Future Access): F
Data Read: D
Speedup Factor for cached tasks: μ

- **Cost of caching: WD**
- **Benefit of caching: μDF**
- **Benefit/Cost: μF/W**