A Hybrid Scheduling Approach for Scalable Heterogeneous Hadoop Systems

Authors:

Aysan Rasooli

Douglas G. Down

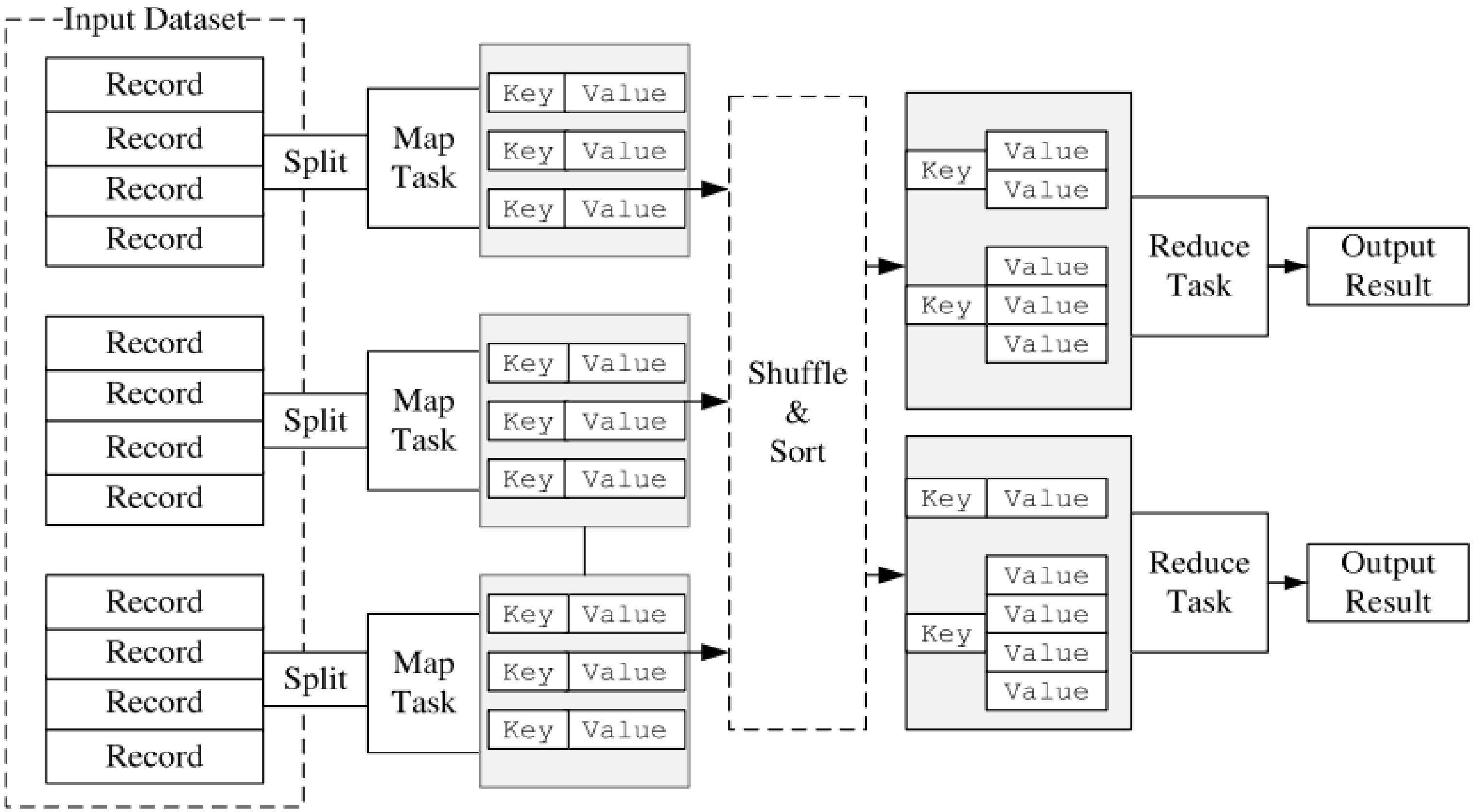
McMaster University

November 2012

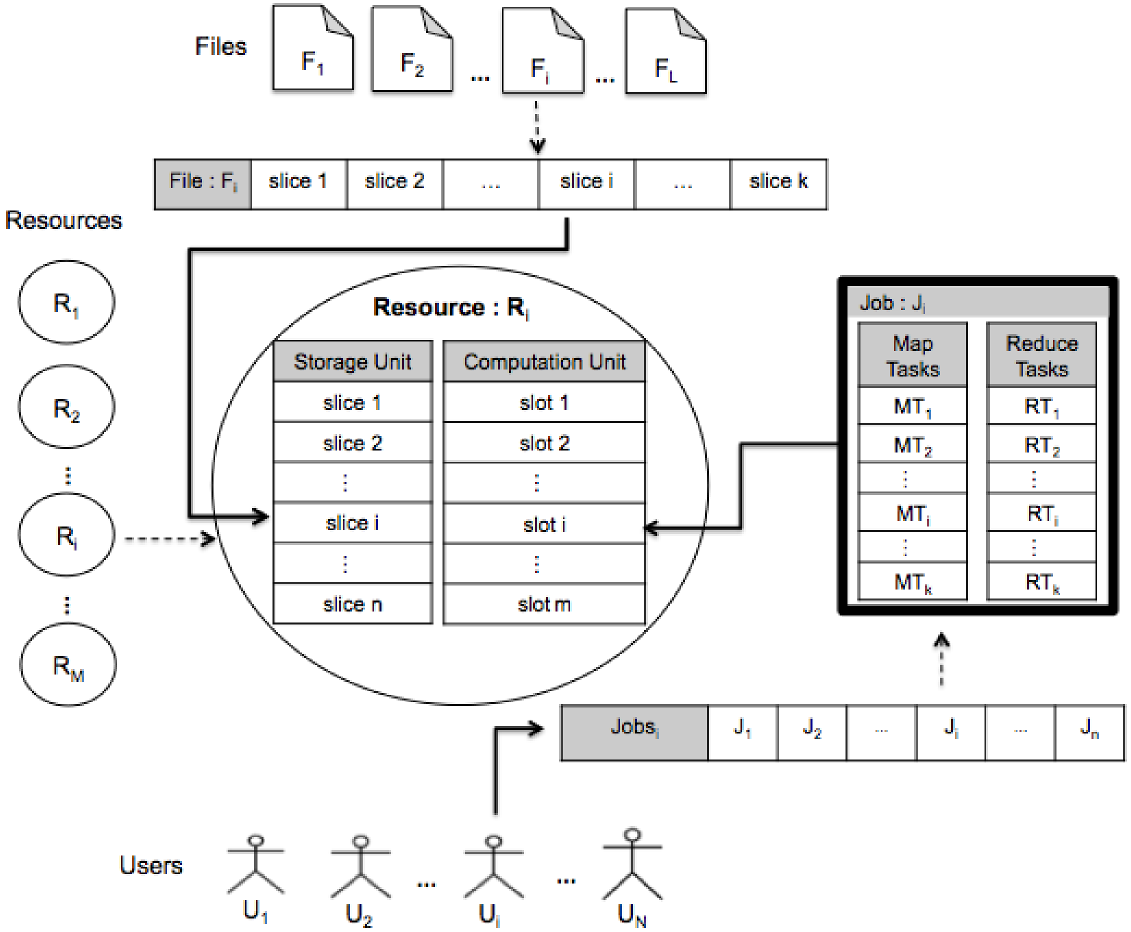
Agenda

- Introducing the Hadoop System
- Heterogeneity and Scalability in Hadoop
- Performance Issues of Existing Hadoop Schedulers
- Proposed Hybrid Scheduling System
- Evaluation
- Conclusion

MapReduce

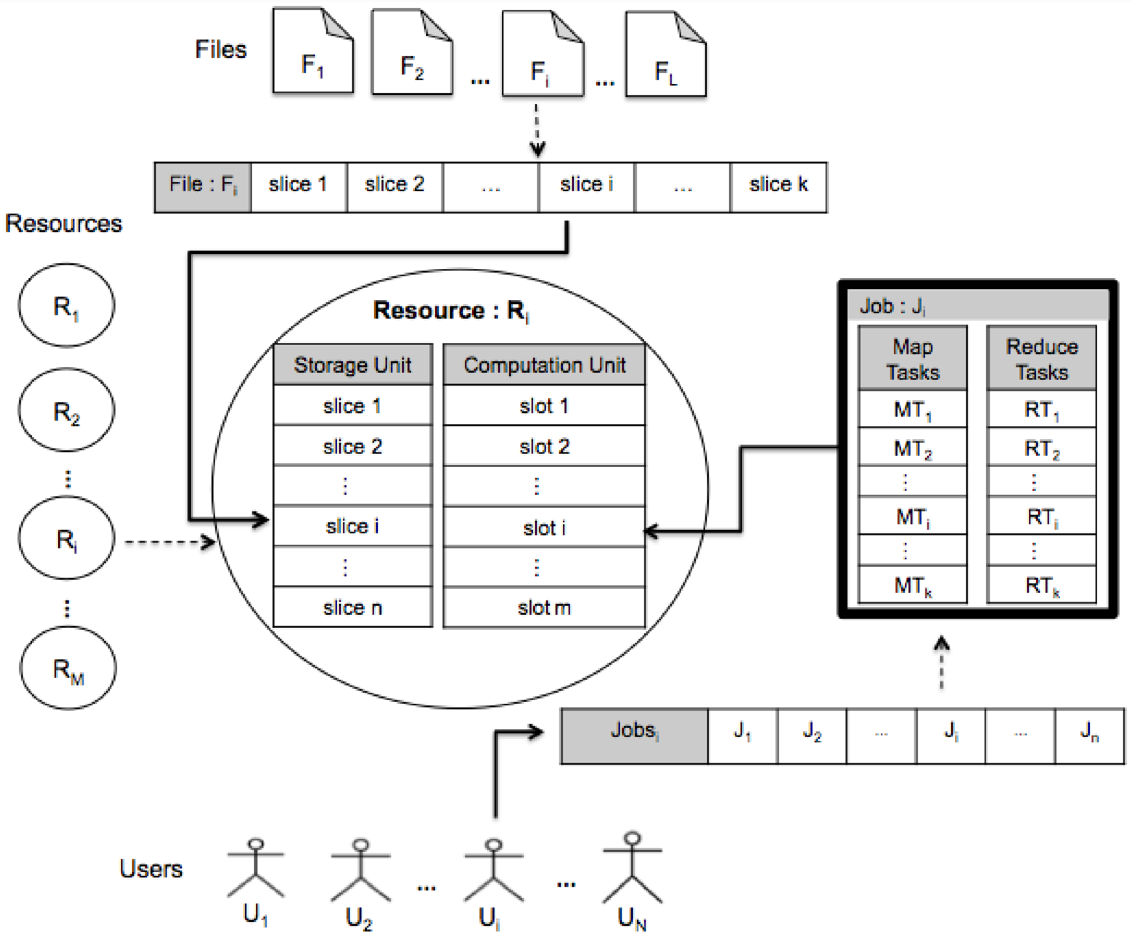

Input Dataset
Record
Record
Record
Record
Split
Map Task
Key Value
Key Value
Key Value
Record
Record
Record
Record
Split
Map Task
Key Value
Key Value
Key Value
Record
Record
Record
Record
Split
Map Task
Key Value
Key Value
Key Value
Shuffle & Sort
Key Value Value
Key Value Value Value
Reduce Task
Output Result
Key Value
Key Value Value Value Value
Reduce Task
Output Result

Hadoop System



Heterogeneity and Scalability in Hadoop

- Cluster
- Workload
- User

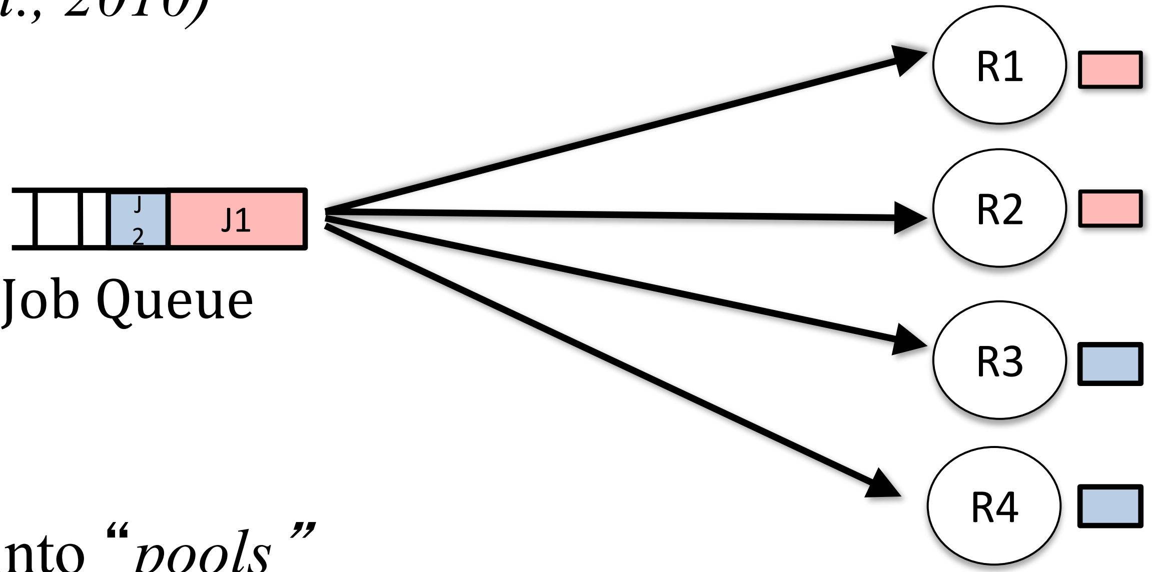


Hadoop Schedulers

- FIFO
- Fair Sharing
- COSHH

Fair Sharing

(Zaharia et al., 2010)



- Group jobs into "*pools*"
- Assign each pool a guaranteed *minimum share*
- Divide excess capacity evenly between pools

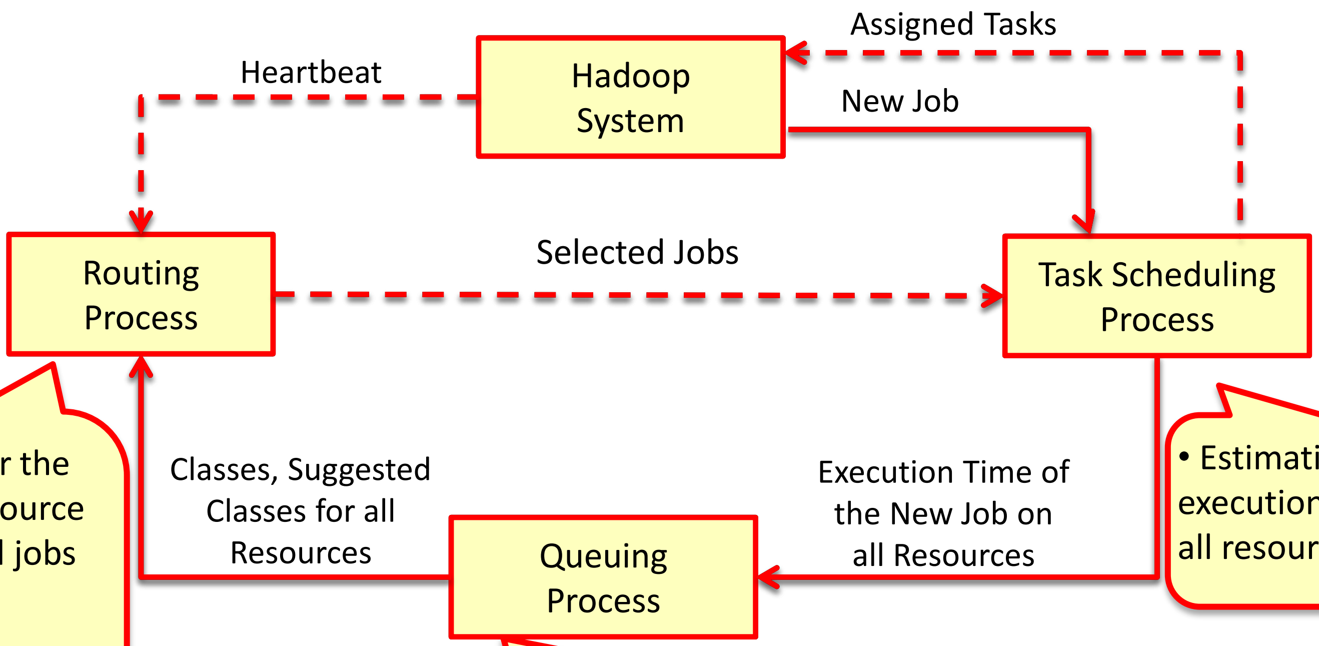
Fair Sharing

- Goal: fast response times for small jobs, guaranteed service levels for long jobs
- Considers Minimum Share satisfaction, Fairness

Drawbacks:

- Does not take into account locality
- Does not take into account heterogeneity

COSHH Scheduler



- Select a job for the current free resource using suggested jobs of the Queuing Process
- Consider the fairness and the minimum share satisfaction in the system

- Classify the jobs
- Calculate the best set of suggested job classes for each resource

- Estimation of Job execution time across all resources

COSHH Scheduler

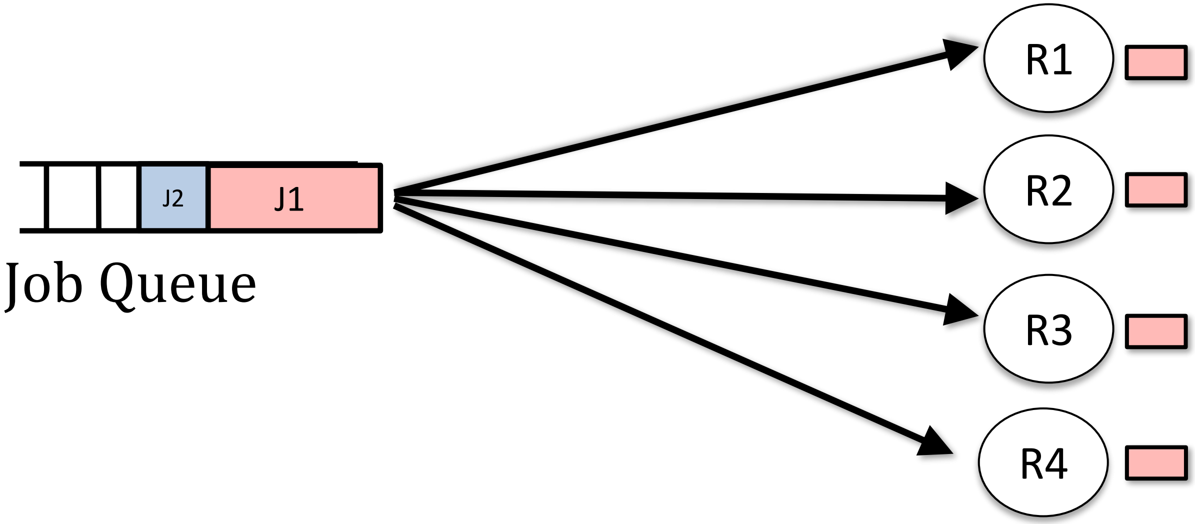
- Considering the heterogeneity in the Hadoop system
- Improves Mean Completion Time
- Considers:
    - Minimum Share Satisfaction
    - Fairness
    - Locality

Performance Issues of Existing Schedulers

McMaster University

Problem I. Small Jobs Starvation

FIFO :

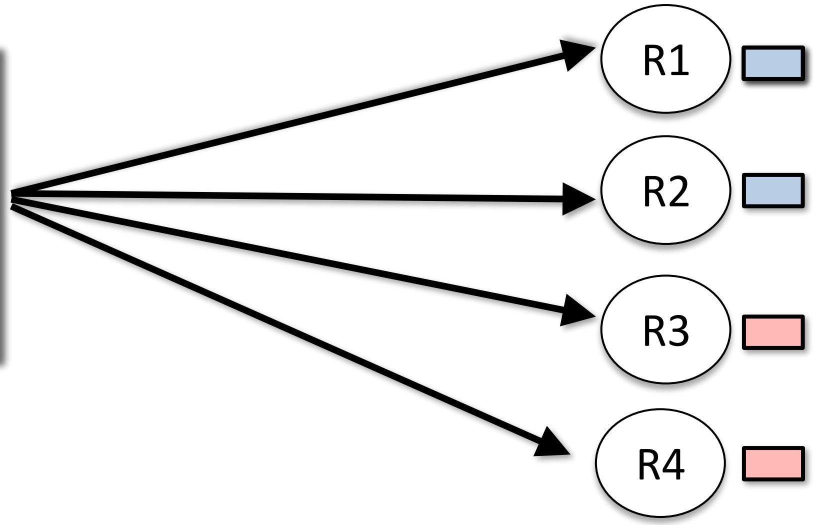


Performance Issues of Existing Schedulers

Problem II. Sticky Slots

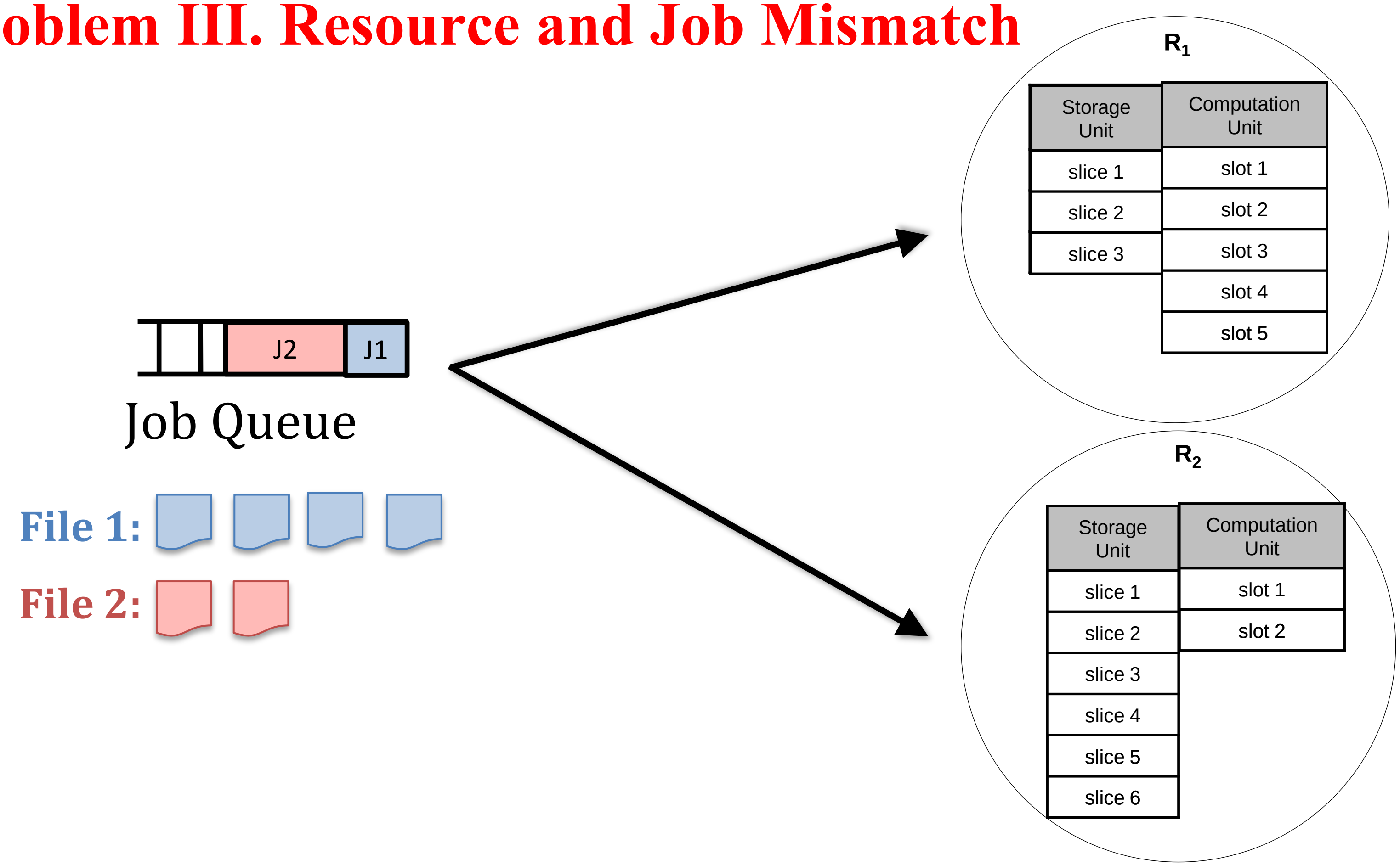
Fair Sharing:

Job	Fair Share	Running Tasks
Job 1	2	**1**
Job 2	2	2



Performance Issues of Existing Schedulers

Problem III. Resource and Job Mismatch



R_1

Storage Unit	Computation Unit
slice 1	slot 1
slice 2	slot 2
slice 3	slot 3
	slot 4
	slot 5

R_2

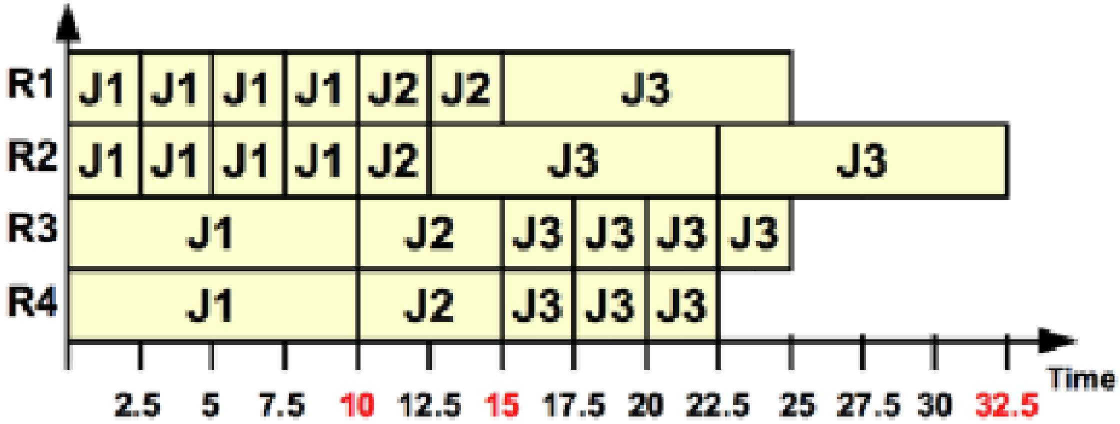
Storage Unit	Computation Unit
slice 1	slot 1
slice 2	slot 2
slice 3	
slice 4	
slice 5	
slice 6	

Problem I. Small Jobs Starvation

FIFO :

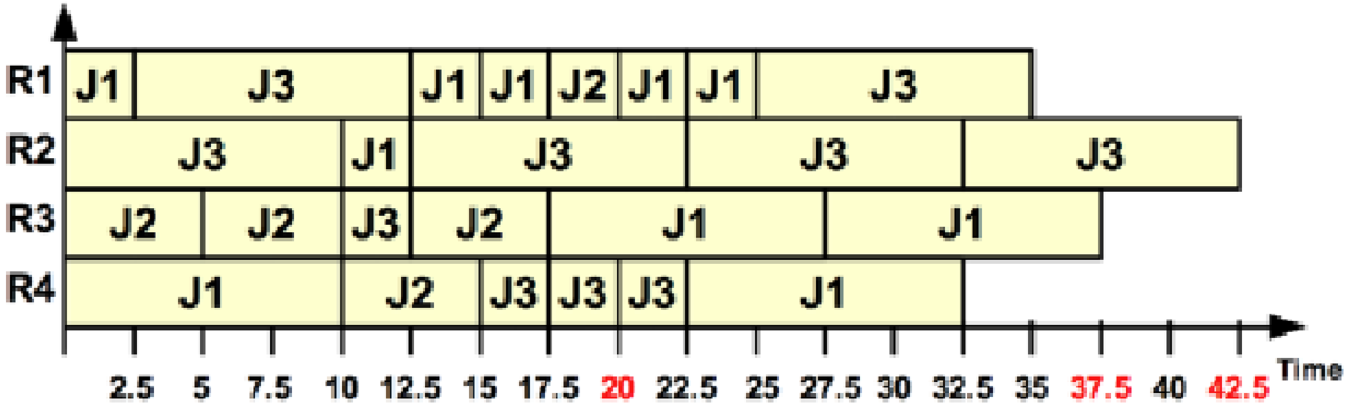
User1: Job1 (consists of 10 Task1)
User2: Job3 (consists of 10 Task3)
User3: Job2 (consists of 5 Task2)

$$m_t = \begin{bmatrix} 2.5 & 2.5 & 10 & 10 \\ 2.5 & 2.5 & 5 & 5 \\ 10 & 10 & 2.5 & 2.5 \end{bmatrix}$$



Problem II. Sticky Slots

Fair Sharing :



User1: Job1 (consists of 10 Task1)
User2: Job3 (consists of 10 Task3)
User3: Job2 (consists of 5 Task2)

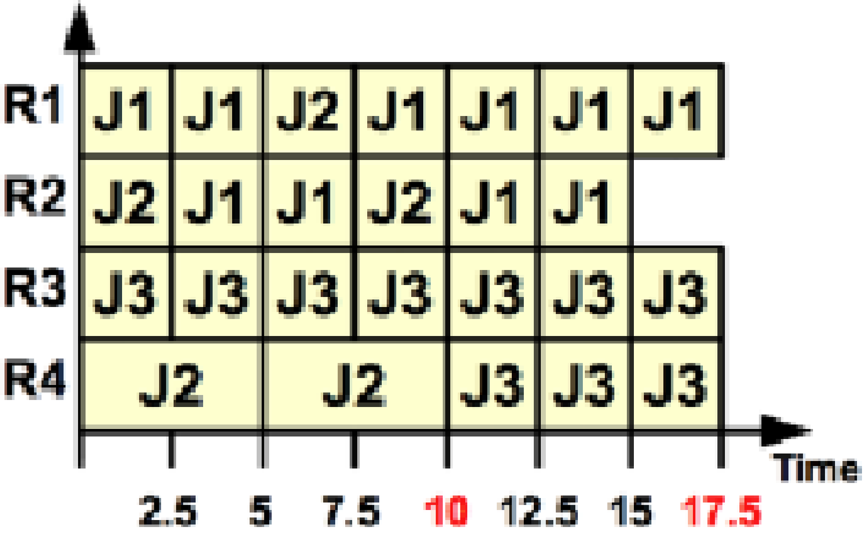
$$m_t = \begin{bmatrix} 2.5 & 2.5 & 10 & 10 \\ 2.5 & 2.5 & 5 & 5 \\ 10 & 10 & 2.5 & 2.5 \end{bmatrix}$$

Problem III. Resource and Job Mismatch

COSHH:

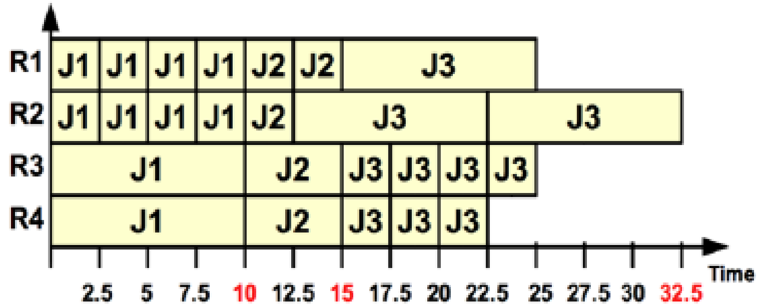
User1: Job1 (consists of 10 Task1)
User2: Job3 (consists of 10 Task3)
User3: Job2 (consists of 5 Task2)

$$m_t = \begin{bmatrix} 2.5 & 2.5 & 10 & 10 \\ 2.5 & 2.5 & 5 & 5 \\ 10 & 10 & 2.5 & 2.5 \end{bmatrix}$$

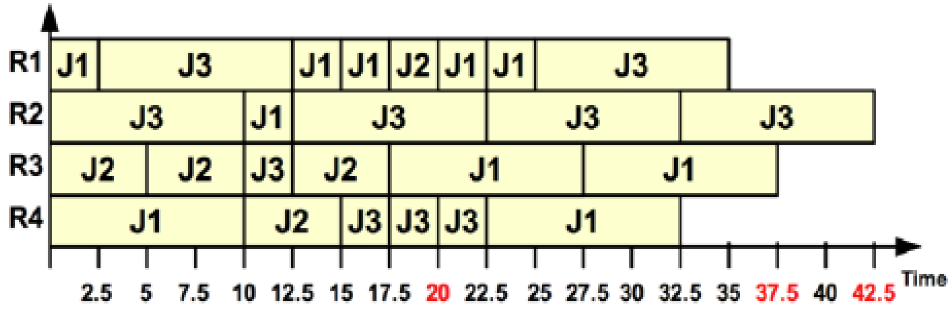


Performance Issues of Existing Schedulers

a) FIFO:

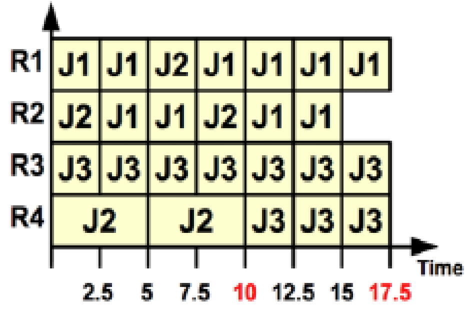


b) Fair Sharing:

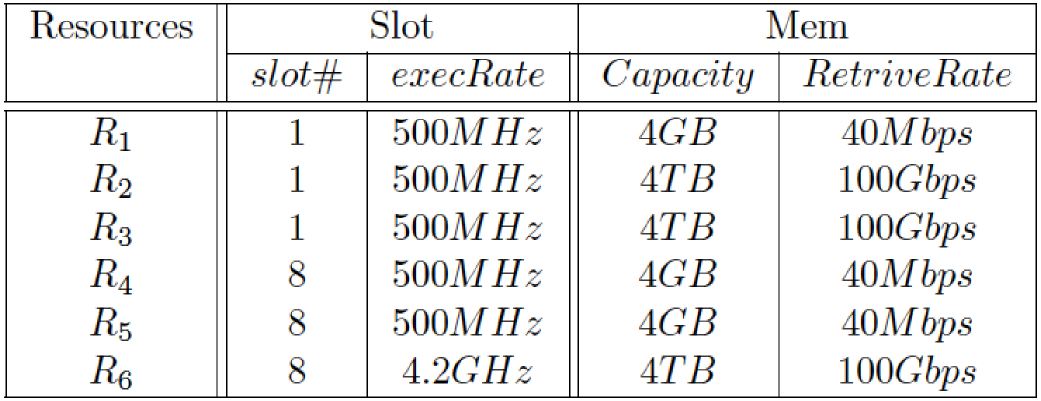


Scheduler	*Job*	*Completion Time*	*Average Completion Time*
FIFO	*J1* *J2* *J3*	*10* *15* *32.5*	*19.17*
Fair Sharing	*J1* *J2* *J3*	*37.5* *20* *42.5*	*33.33*
COSHH	*J1* *J2* *J3*	*17.5* *10* *17.5*	*15*

c) COSHH:



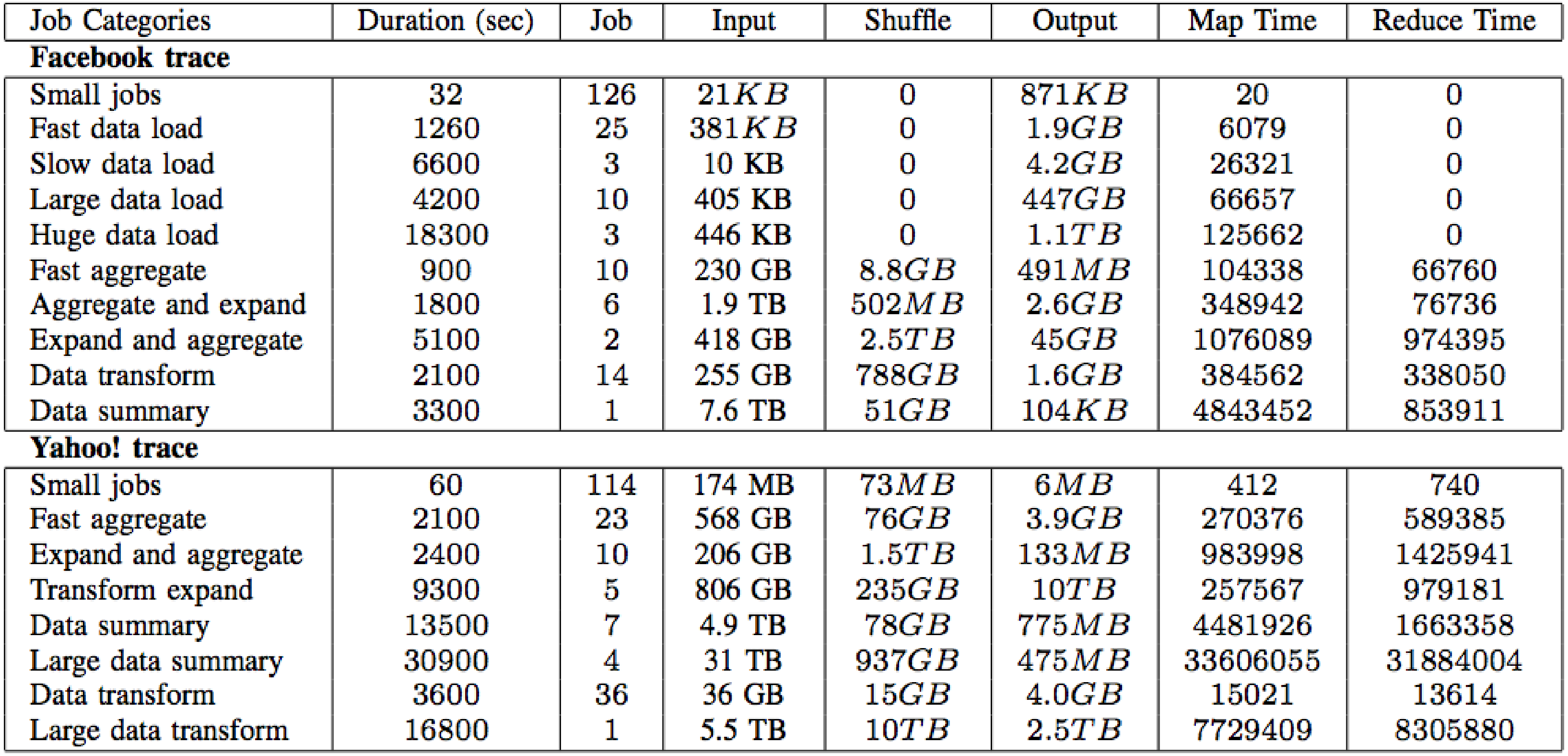
Experimental Environment



Resources	Slot		Mem	
	$slot\#$	$execRate$	$Capacity$	$RetriveRate$
R_1	1	$500MHz$	$4GB$	$40Mbps$
R_2	1	$500MHz$	$4TB$	$100Gbps$
R_3	1	$500MHz$	$4TB$	$100Gbps$
R_4	8	$500MHz$	$4GB$	$40Mbps$
R_5	8	$500MHz$	$4GB$	$40Mbps$
R_6	8	$4.2GHz$	$4TB$	$100Gbps$

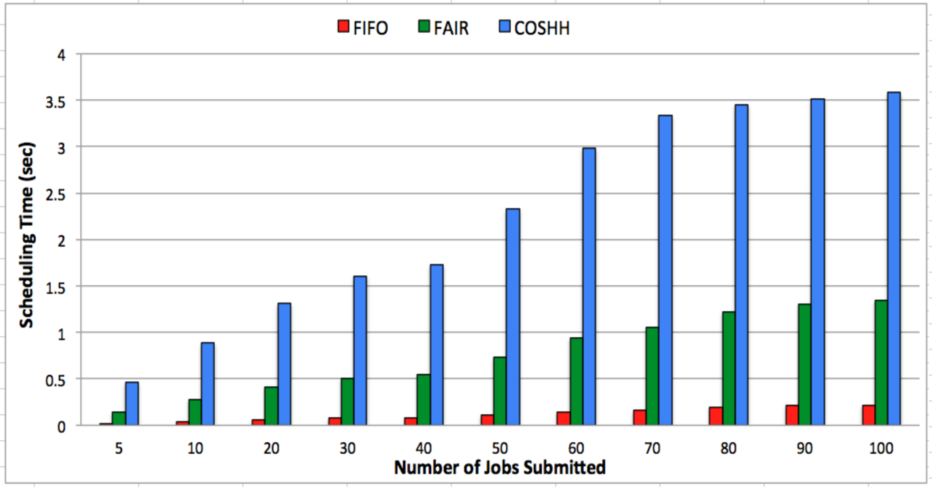
Real Hadoop Workloads

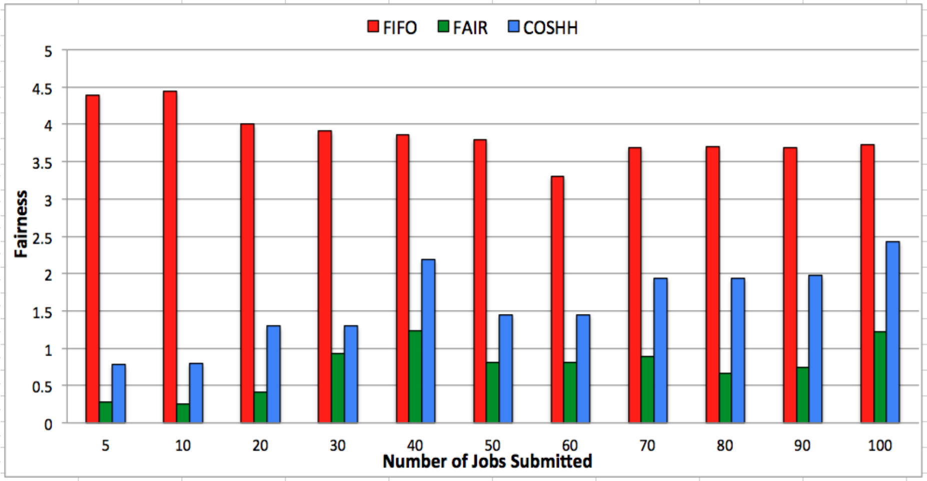
(Chen et al., 2011)

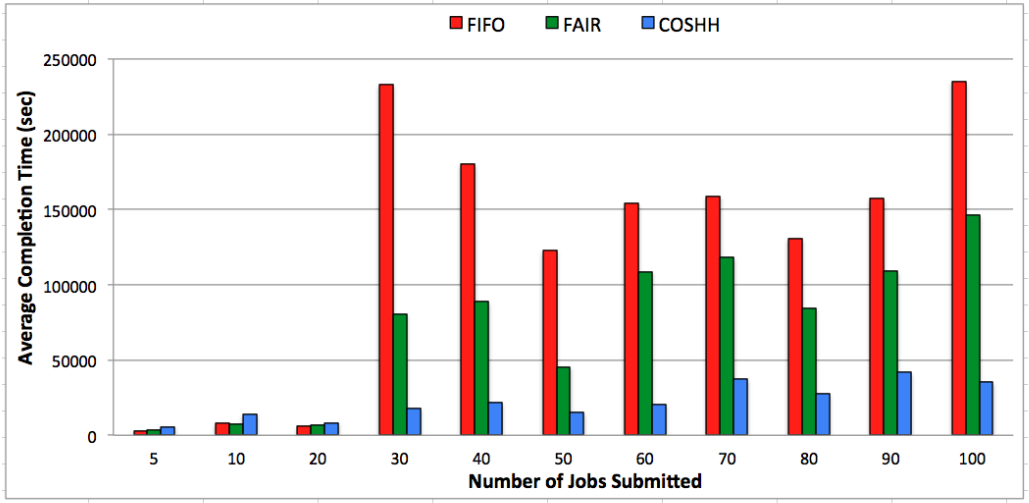


Job Categories	Duration (sec)	Job	Input	Shuffle	Output	Map Time	Reduce Time
Facebook trace							
Small jobs	32	126	21KB	0	871KB	20	0
Fast data load	1260	25	381KB	0	1.9GB	6079	0
Slow data load	6600	3	10 KB	0	4.2GB	26321	0
Large data load	4200	10	405 KB	0	447GB	66657	0
Huge data load	18300	3	446 KB	0	1.1TB	125662	0
Fast aggregate	900	10	230 GB	8.8GB	491MB	104338	66760
Aggregate and expand	1800	6	1.9 TB	502MB	2.6GB	348942	76736
Expand and aggregate	5100	2	418 GB	2.5TB	45GB	1076089	974395
Data transform	2100	14	255 GB	788GB	1.6GB	384562	338050
Data summary	3300	1	7.6 TB	51GB	104KB	4843452	853911
Yahoo! trace							
Small jobs	60	114	174 MB	73MB	6MB	412	740
Fast aggregate	2100	23	568 GB	76GB	3.9GB	270376	589385
Expand and aggregate	2400	10	206 GB	1.5TB	133MB	983998	1425941
Transform expand	9300	5	806 GB	235GB	10TB	257567	979181
Data summary	13500	7	4.9 TB	78GB	775MB	4481926	1663358
Large data summary	30900	4	31 TB	937GB	475MB	33606055	31884004
Data transform	3600	36	36 GB	15GB	4.0GB	15021	13614
Large data transform	16800	1	5.5 TB	10TB	2.5TB	7729409	8305880

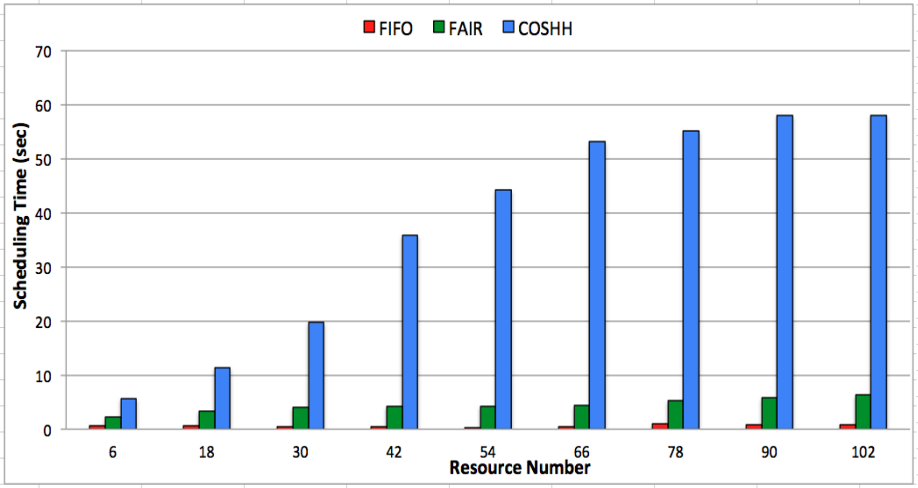
Scalability Analysis- Results
Job Number Scalability

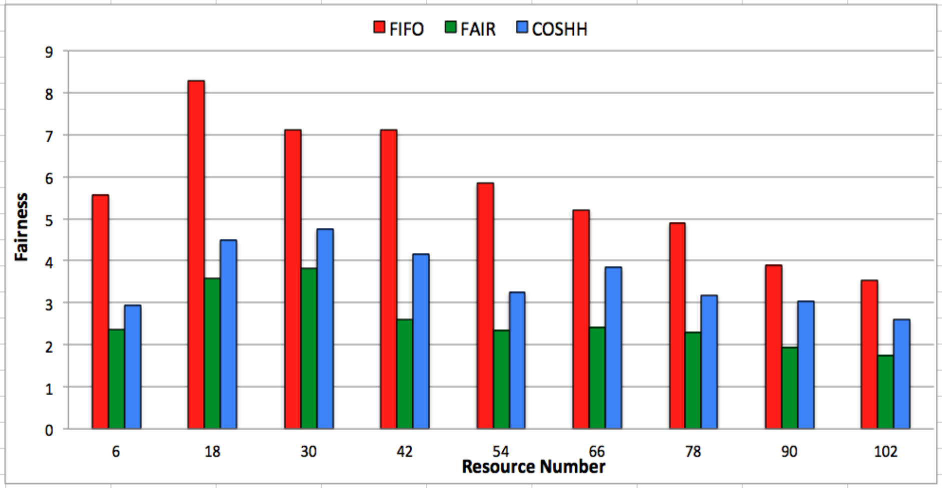


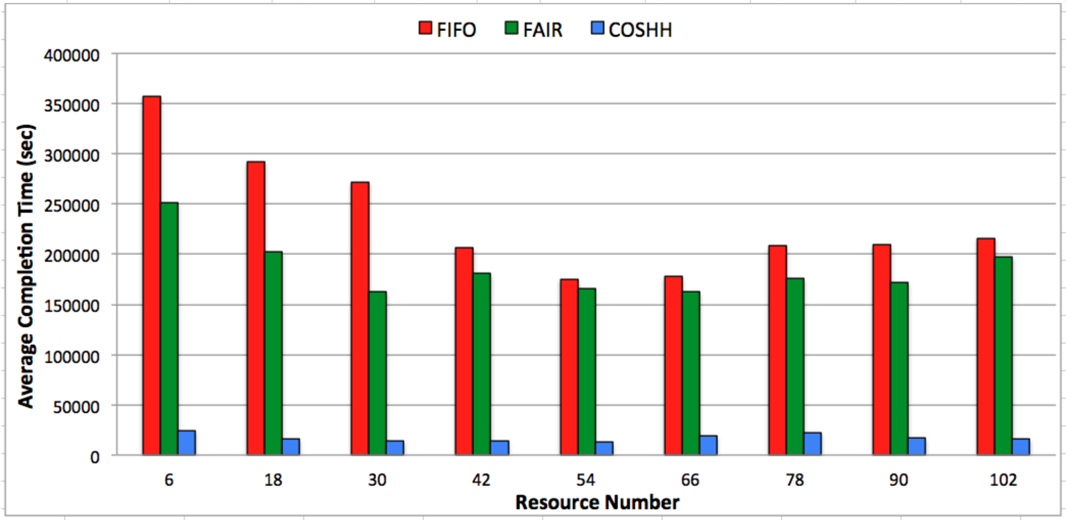




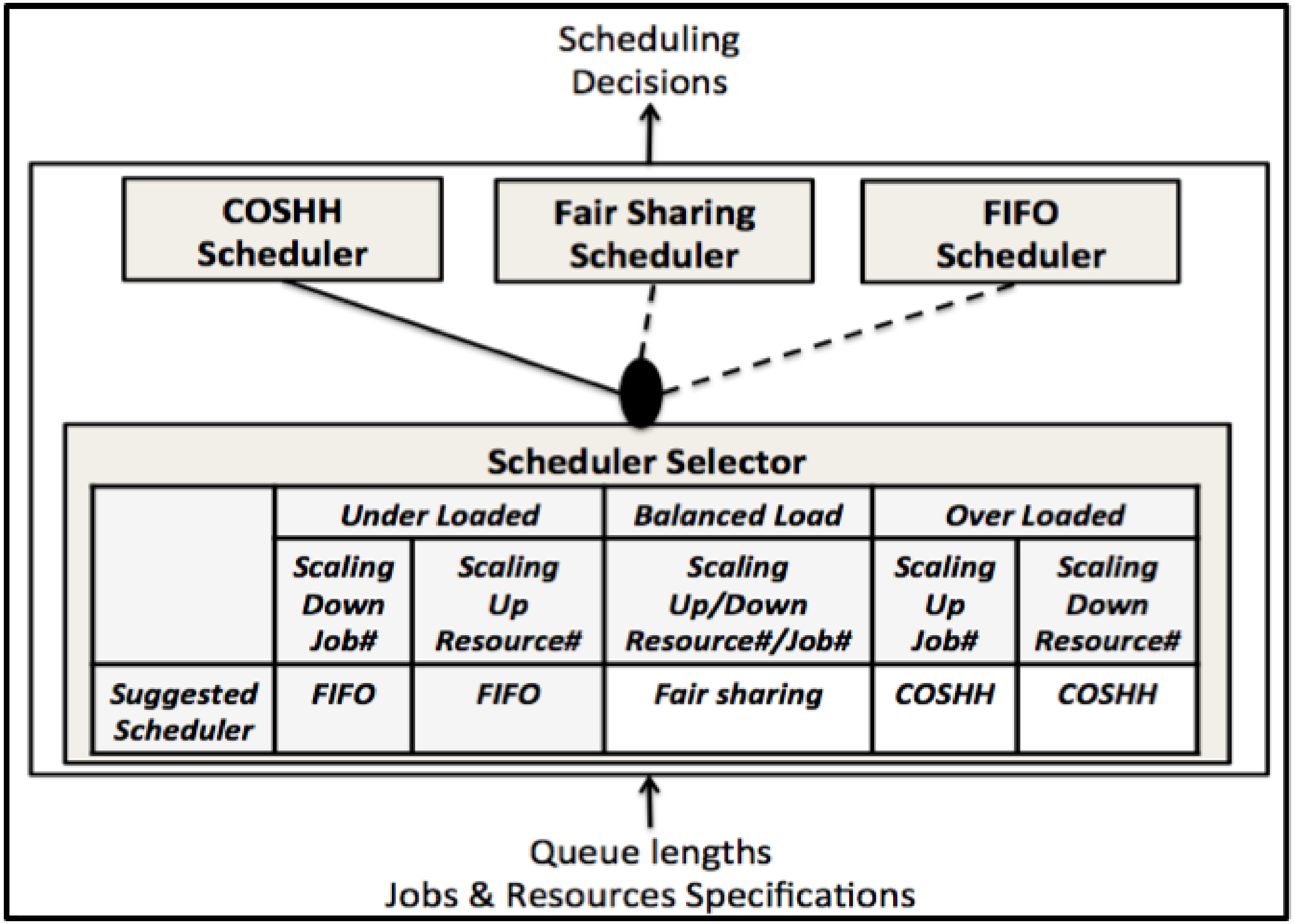
Scalability Analysis- Results
Resource Number Scalability







Scalability Analysis- Hybrid Scheduler



Scheduling Decisions

COSHH Scheduler

Fair Sharing Scheduler

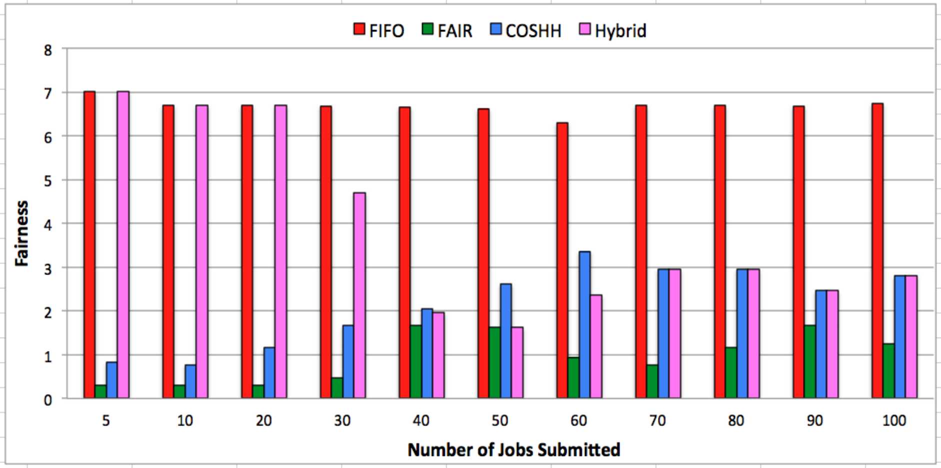
FIFO Scheduler

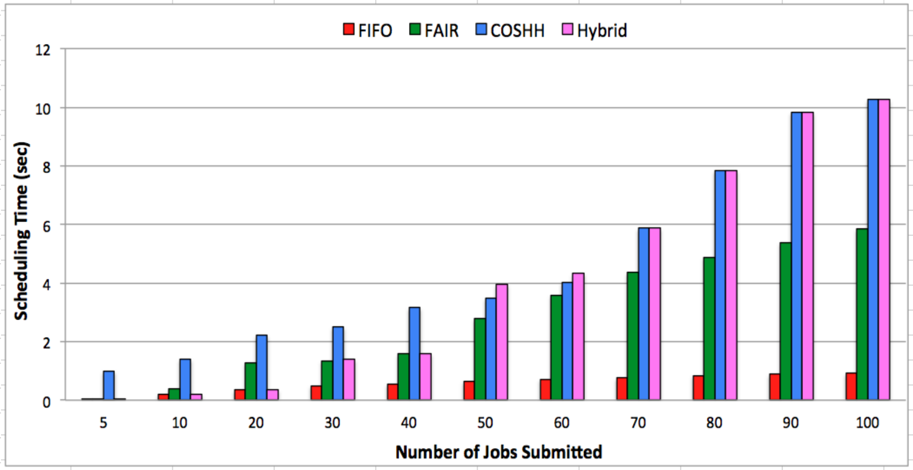
Scheduler Selector

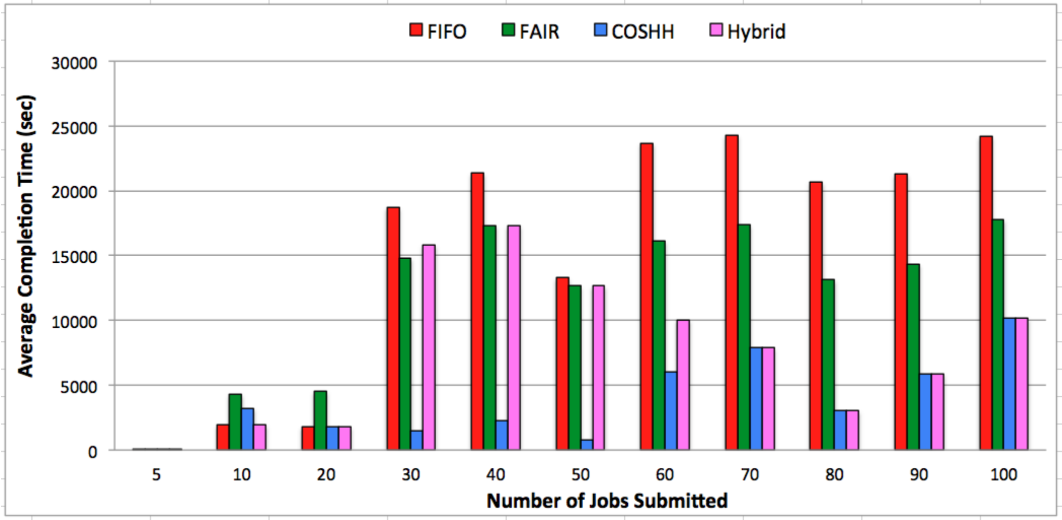
	Under Loaded		*Balanced Load*	*Over Loaded*	
	Scaling Down Job#	*Scaling Up Resource#*	*Scaling Up/Down Resource#/Job#*	*Scaling Up Job#*	*Scaling Down Resource#*
Suggested Scheduler	*FIFO*	*FIFO*	*Fair sharing*	*COSHH*	*COSHH*

Queue lengths
Jobs & Resources Specifications

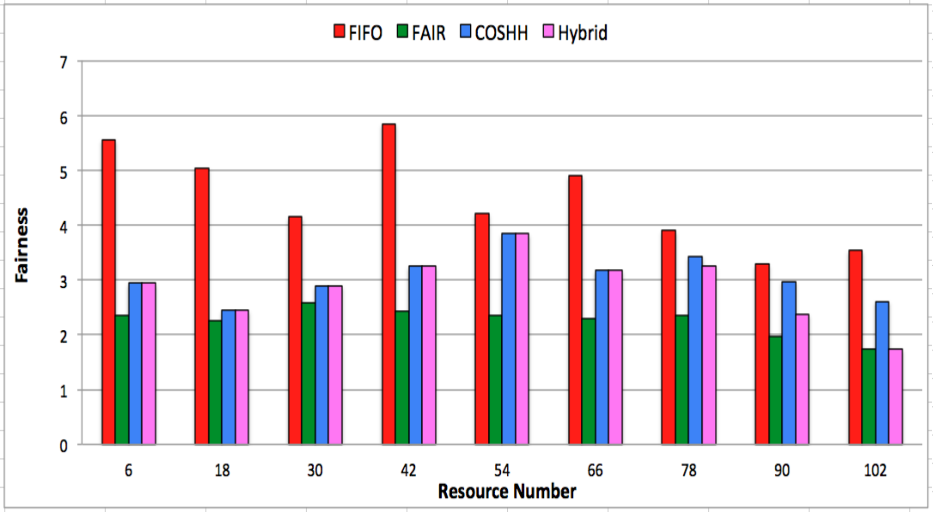
Scalability Analysis- Hybrid Scheduler Job Number Scalability

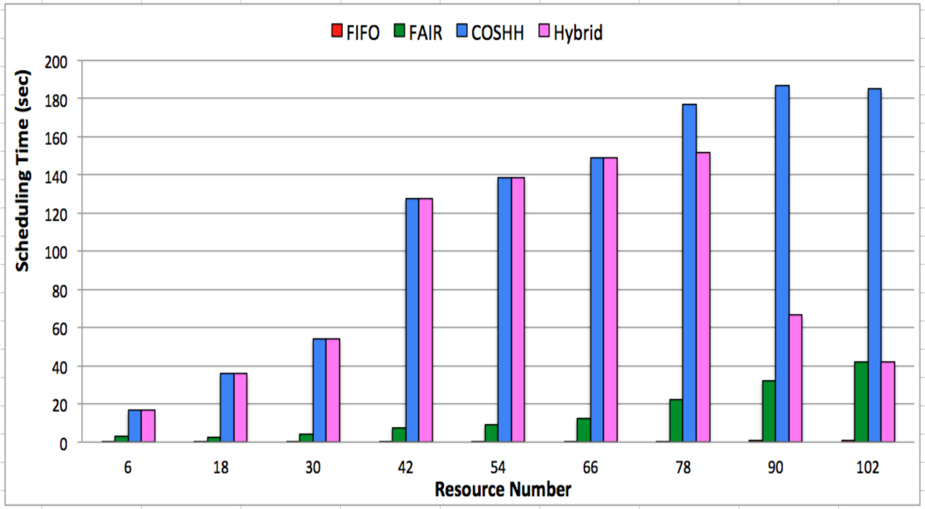


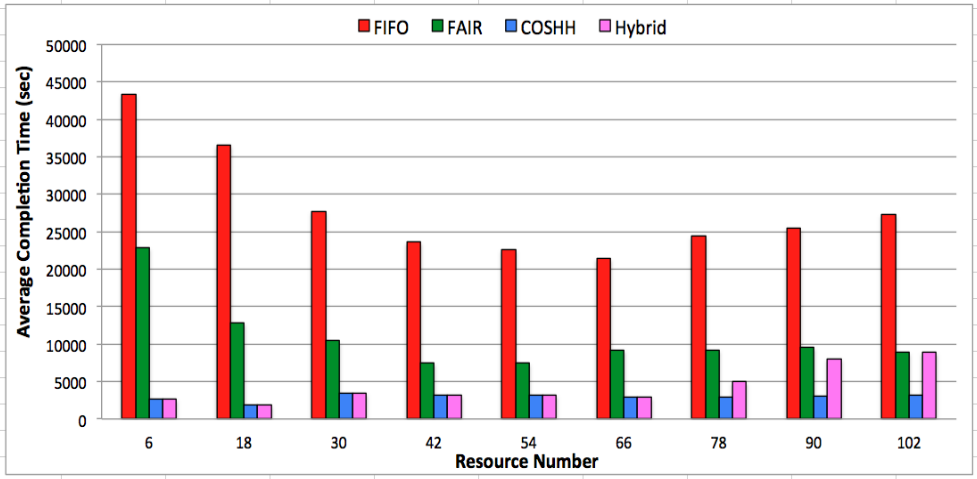




Scalability Analysis- Hybrid Scheduler Resource Number Scalability







Conclusion

- Performance Issues of Hadoop Schedulers:
  - Small Jobs Starvation
  - Sticky Slots
  - Resource and Jobs Mismatch
- Propose a Hybrid Hadoop Scheduler

Thanks