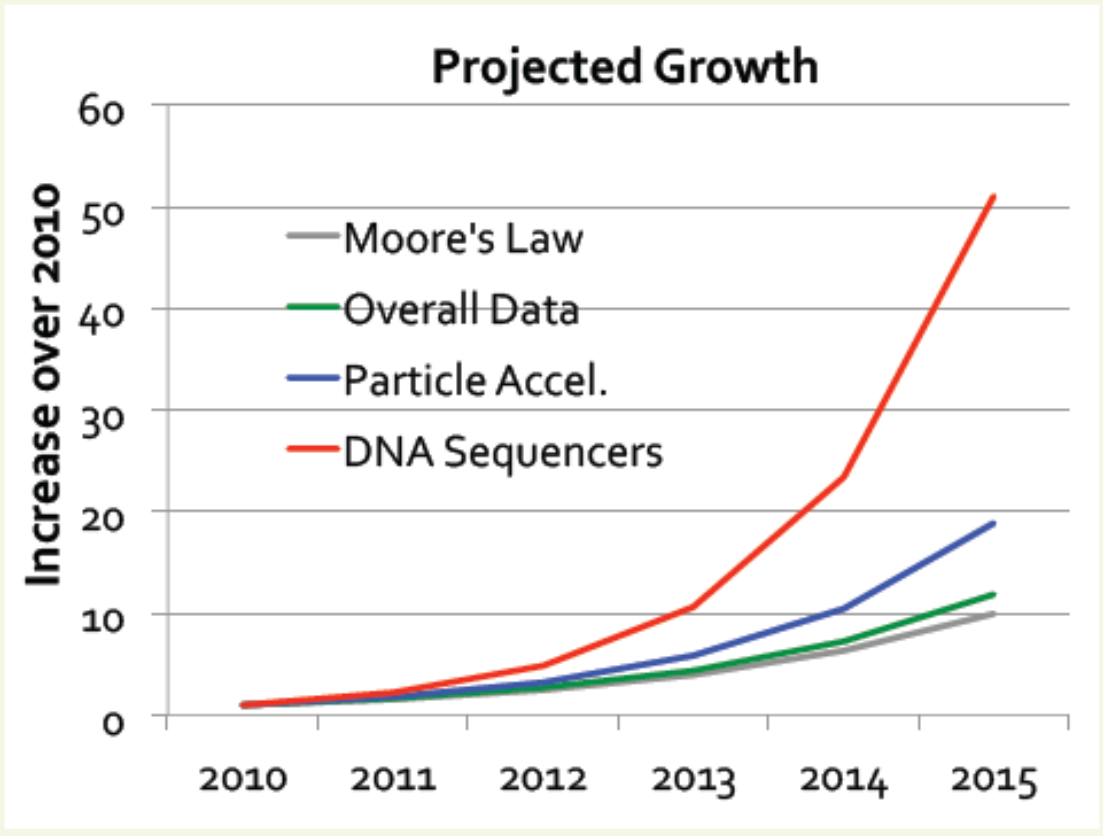
The Power of Choice in Data-Aware Cluster Scheduling

Shivaram Venkataraman , Aurojit Panda
Ganesh Ananthanarayanan,
Michael Franklin, Ion Stoica
UC Berkeley , Microsoft Research
OSDI’14

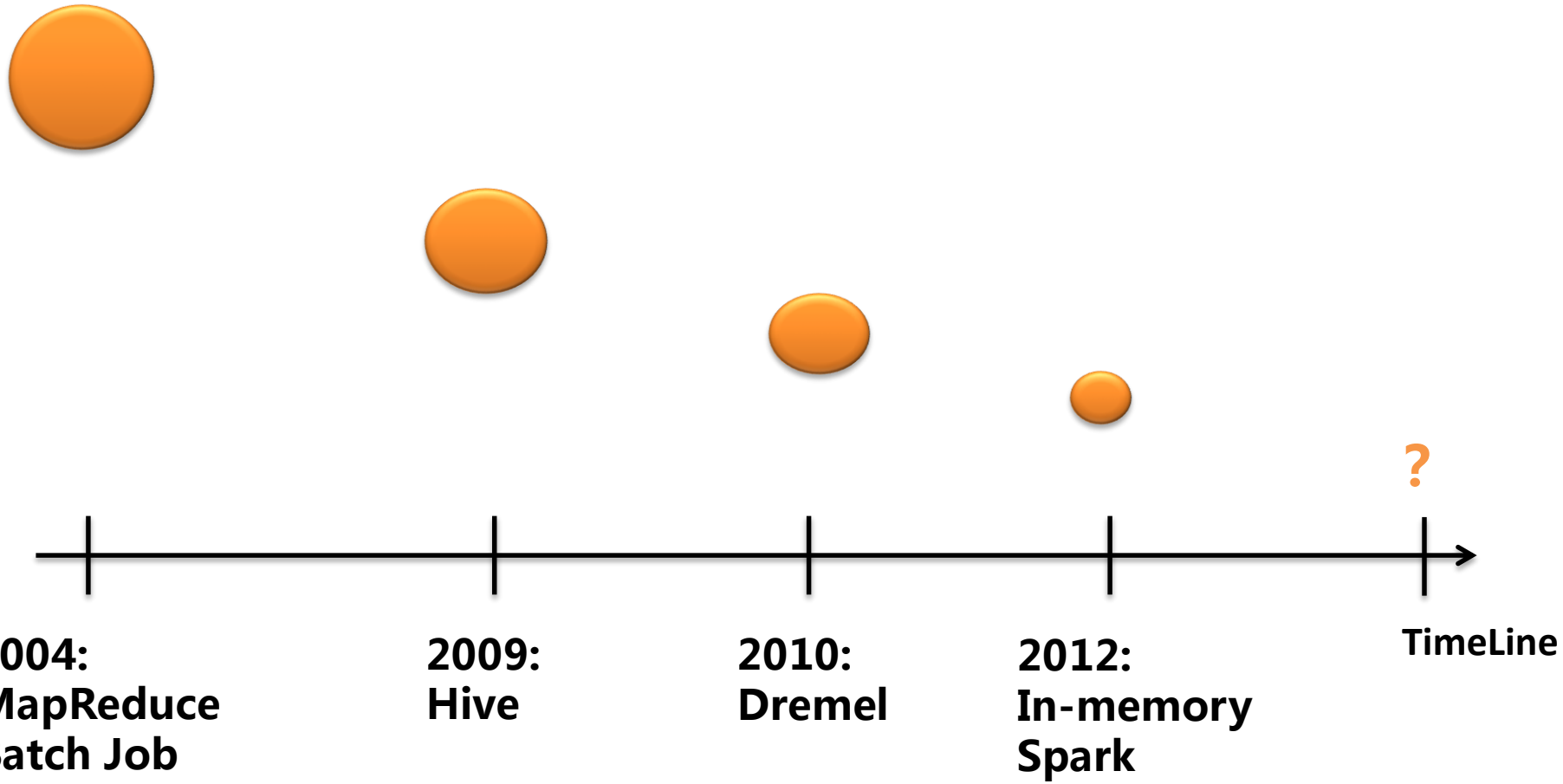
By Pan Fengfeng

Trends: Big Data



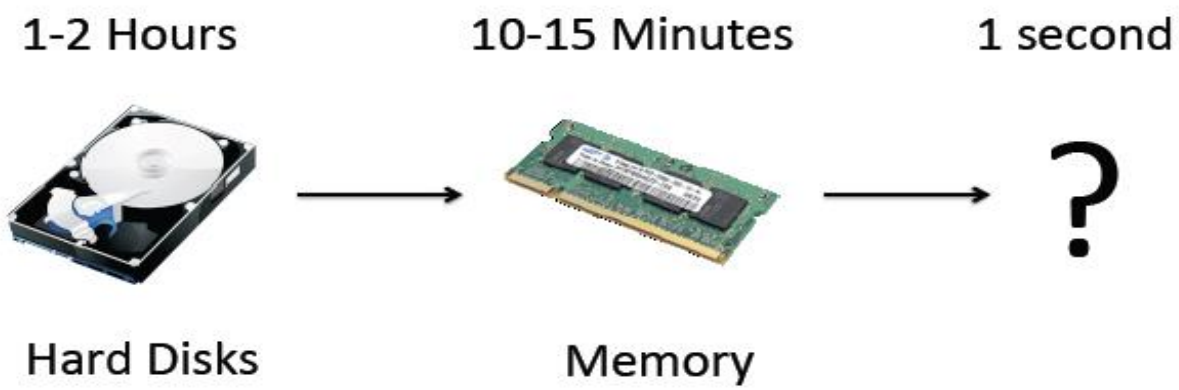
Data grows faster than Moore's Law

Trends: Low Latency



Big Data or Low Latency

- **100TB on 1000 machines**



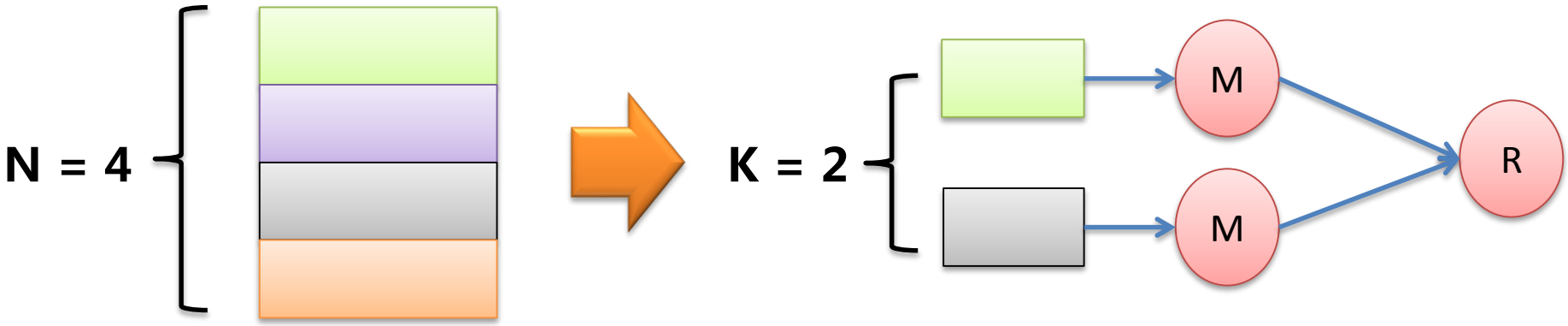
- **Even if no communication and all data in memory, query may take tens of sec**
  - Just scanning 200-300GB RAM may take 10sec

Using a subset of input data

Applications

- **Approximate Query Processing**
  - BlinkDB
  - Presto
- **Machine Learning**
  - Stochastic gradient
  - Coordinate descent

Example:
Using a subset of input data



Data-Aware Scheduling I

- **Input Stage**
  - To schedule the task on a machine that stores its input

Less than 60% of tasks achieve locality even with three replicas

Locality for input Tasks

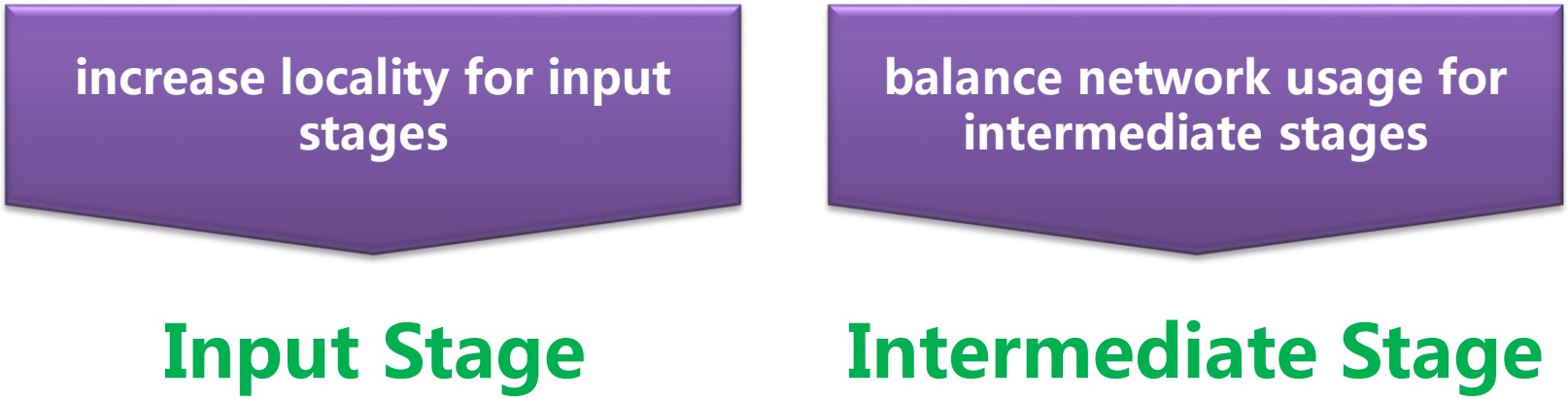
Data-Aware Scheduling II

- **Intermediate Stage**
  - To schedule the task at a machine that minimizes the time it takes to transfer all remote inputs

Balanced Network for intermediate Tasks

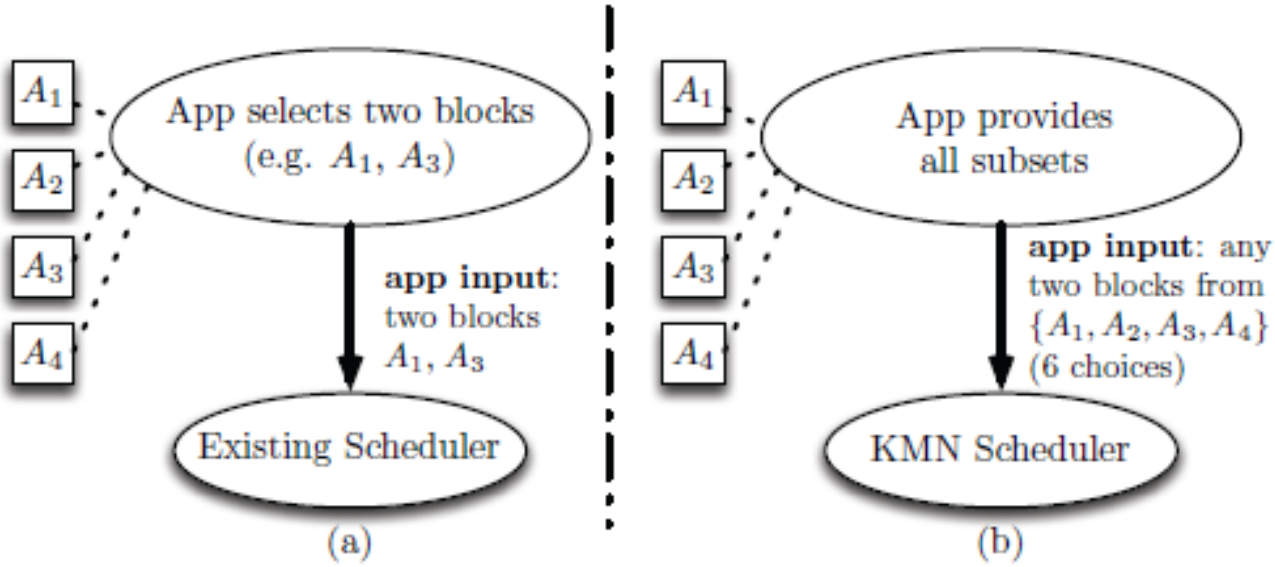
KMN Scheduler

- **A scheduling framework**
  - exploits the available choices to improve performance



Input Stage

- **Two scenarios**
  - **Jobs which can use any K of the N input blocks**
  - **Jobs which use a custom sampling functions**



Input Stage

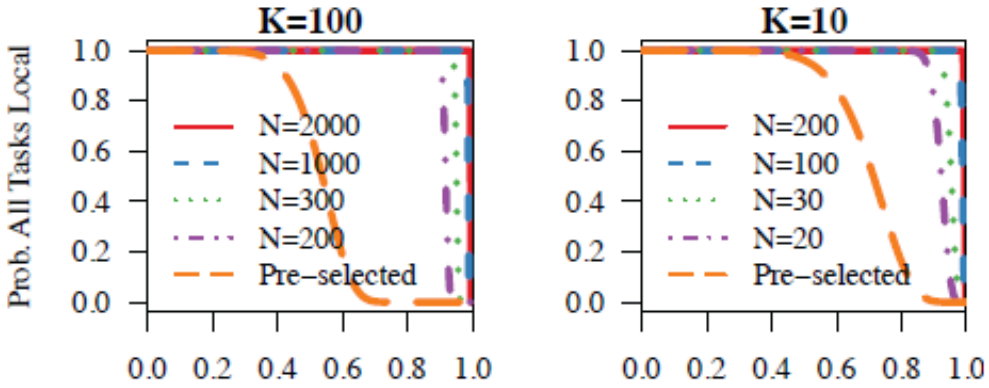
- **Main idea**
  - leverage the combinatorial choice of inputs
- **Potential Benefit (Quantization)**

$$p_t = 1 - u^s$$

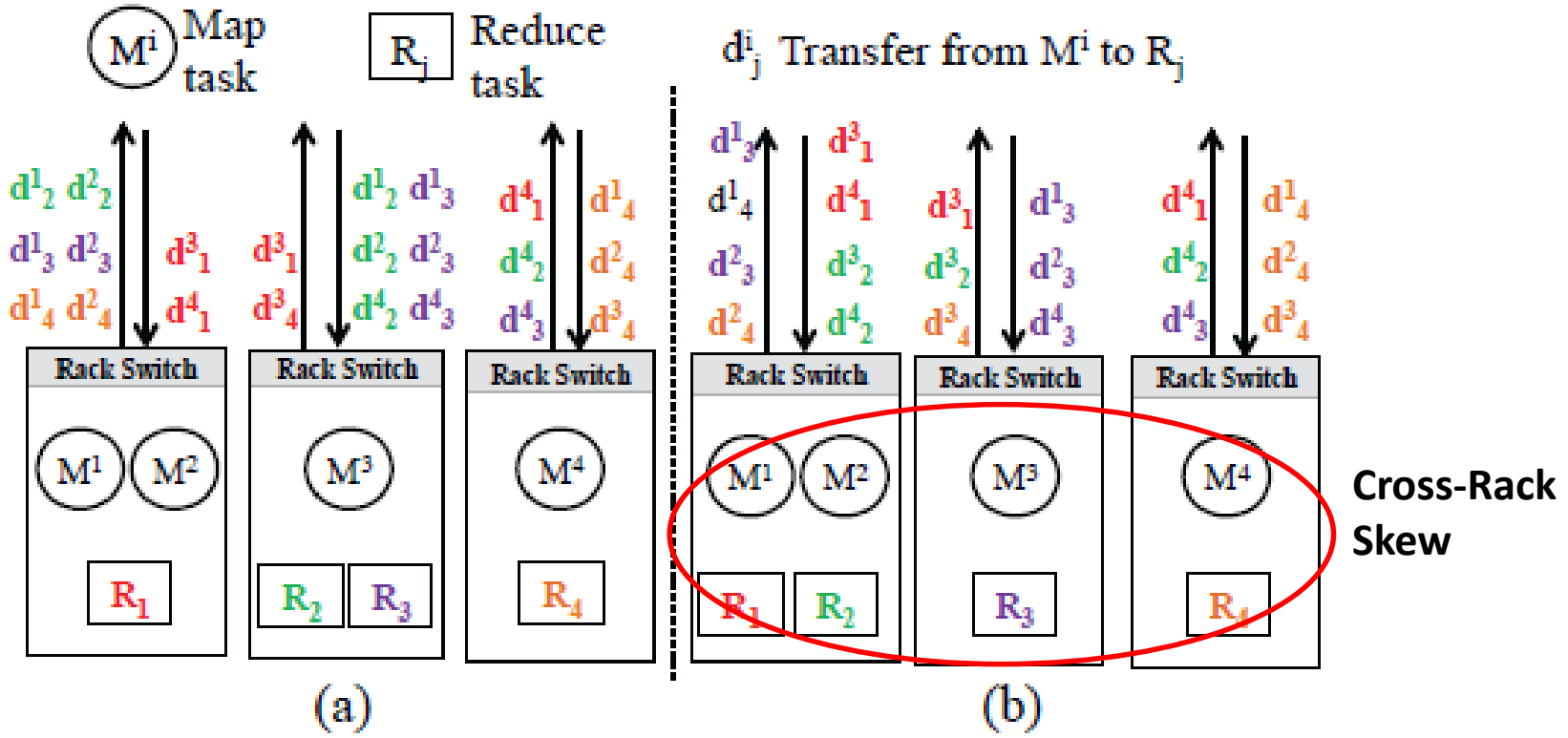
u is the cluster utilization
s is the number of compute slots per machine

$$1 - \sum_{i=0}^{K-1} \binom{N}{i} p_t^i (1 - p_t)^{N-i}$$

the probability for K out of N tasks getting locality



Intermediate Stage



Better placement of both upstream and downstream tasks

Intermediate Stage

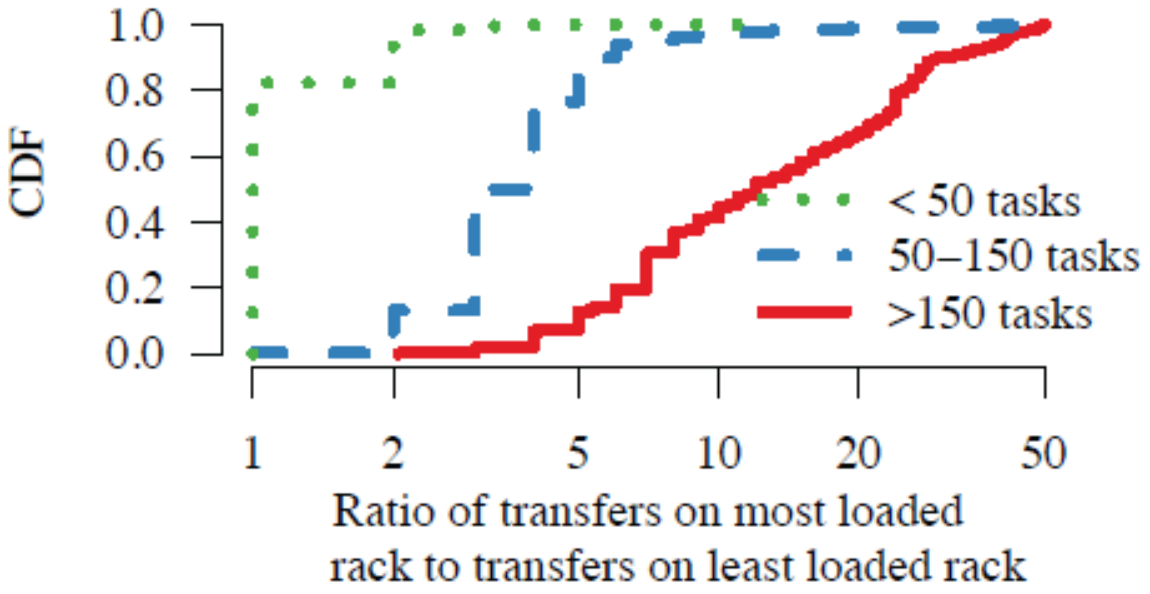


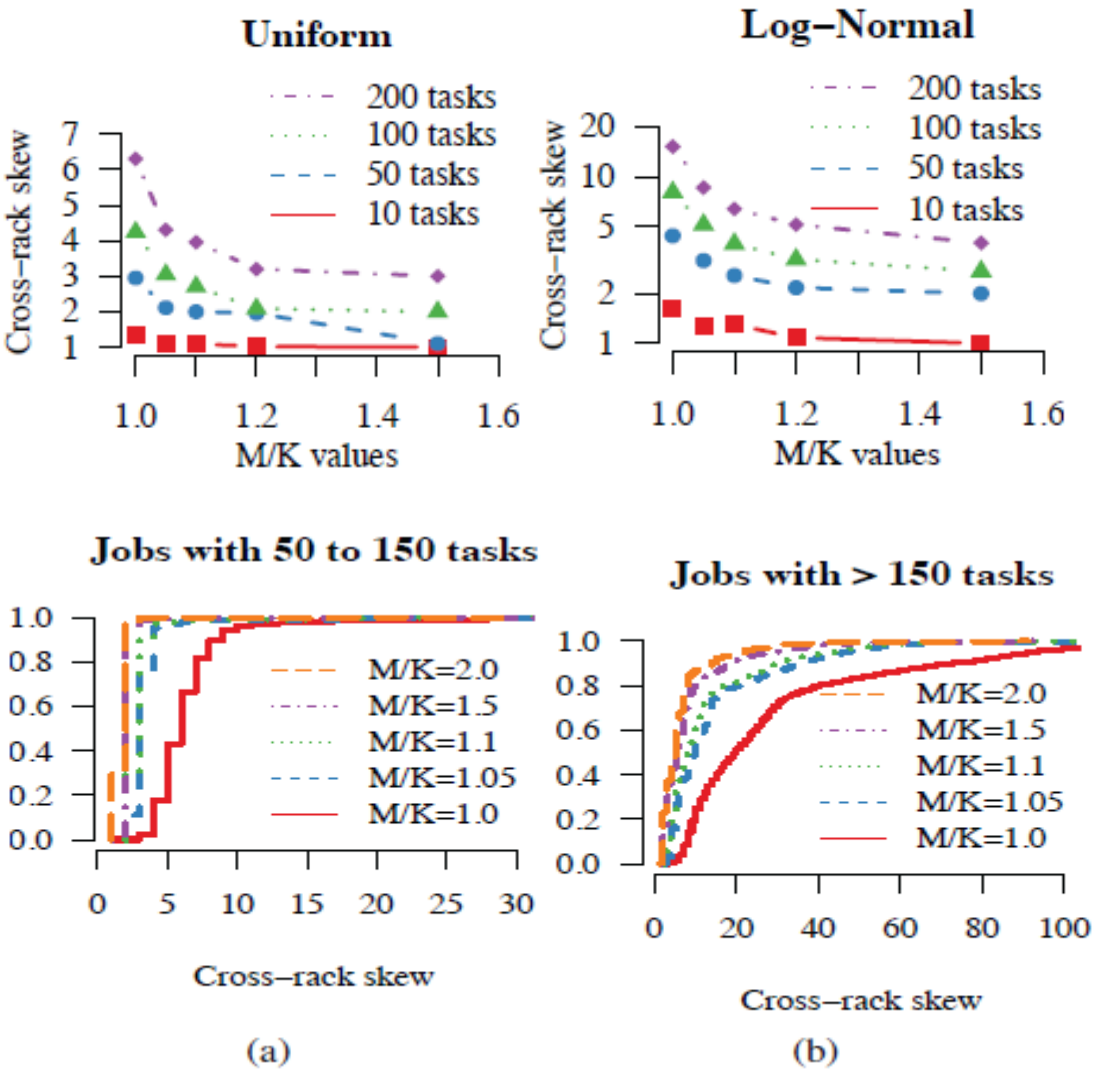
Figure 3: **CDF of cross-rack skew for the Facebook trace split by number of map tasks. Reducing cross-rack skew improves intermediate stage performance.**

Intermediate Stage

- **downstream tasks**
  - limited by the locations of the outputs of upstream tasks
- **Scheduling upstream tasks**
  - **more important and complicated**

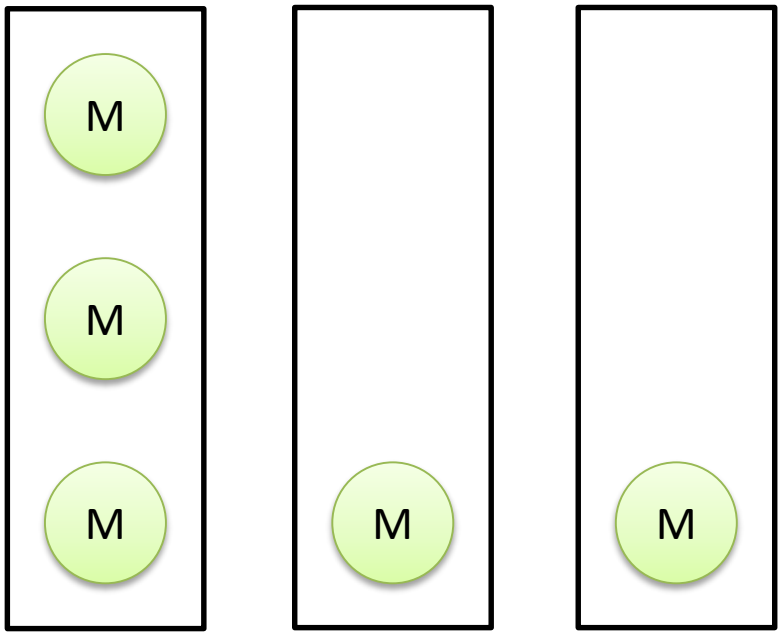
Main idea
scheduling a few additional upstream tasks

Running a few extra tasks is an effective strategy to reduce skew



Cross-rack skew :
ratio of the rack with largest and smallest number of upstream tasks

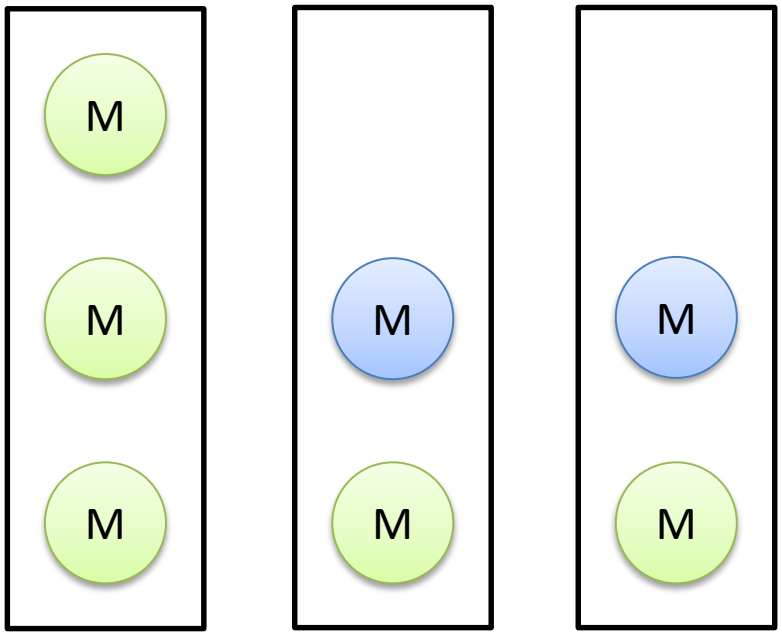
Select best upstream outputs



K = 5
cross-rack skew = 3

Select best upstream outputs

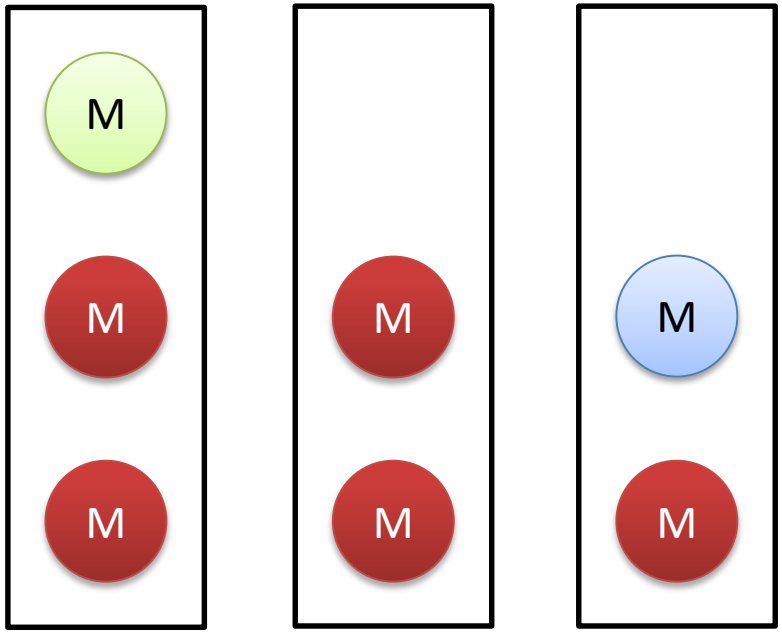
- **After running extra Tasks**



M = 7
K = 5
cross-rack skew = 3

Select best upstream outputs

- **Technique**
  - spreading our choice of *K* outputs across as many racks as possible



M = 7
K = 5
cross-rack skew = 2

Using KMN

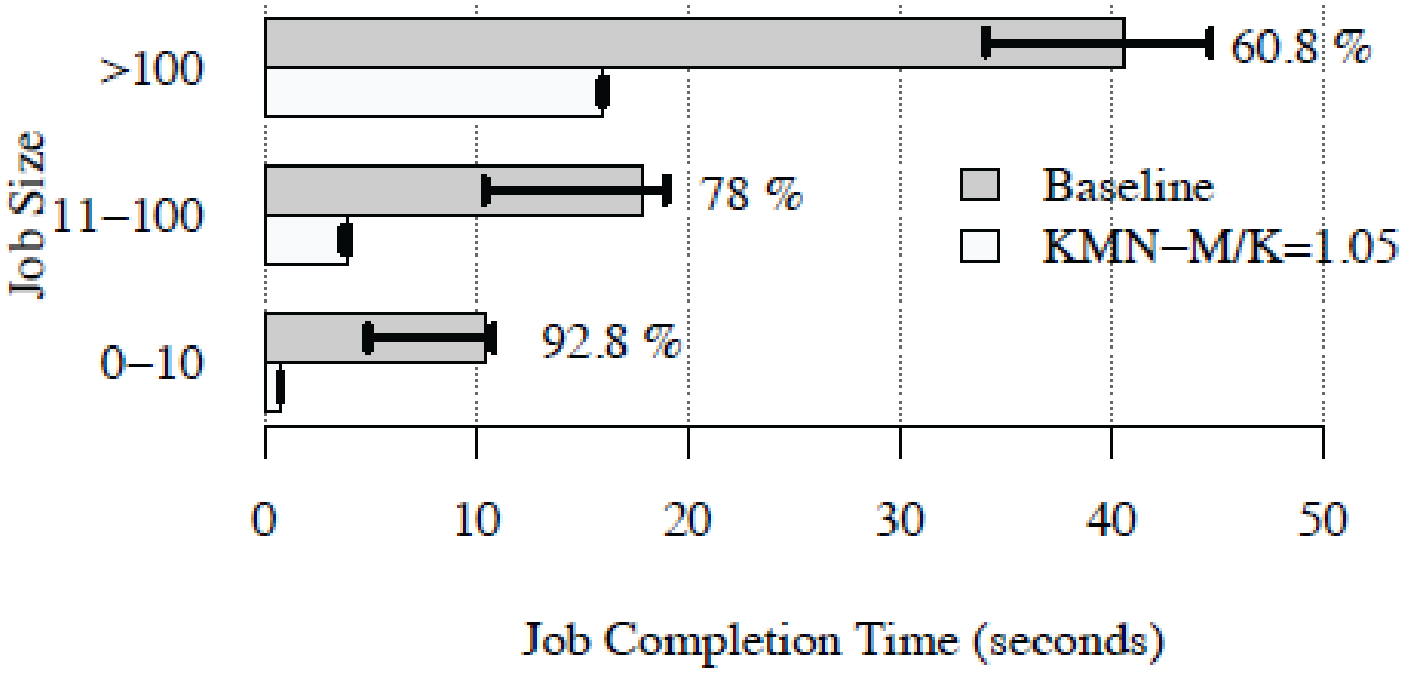
```
// SQL Query
SELECT status, SUM(quantity)
FROM items
GROUP BY status

// Spark Query
kv = file.map{ li =>
  (li.l_linestatus,li.quantity)}
result = kv.reduceByKey{(a,b) =>
  a + b}.collect()

// KMN Query
sample = file.blockSample(0.1, sampler=None)
kv = sample.map{ li =>
  (li.l_linestatus,li.quantity)}
result = kv.reduceByKey{(a,b) =>
  a + b}.collect()
```

Evaluation

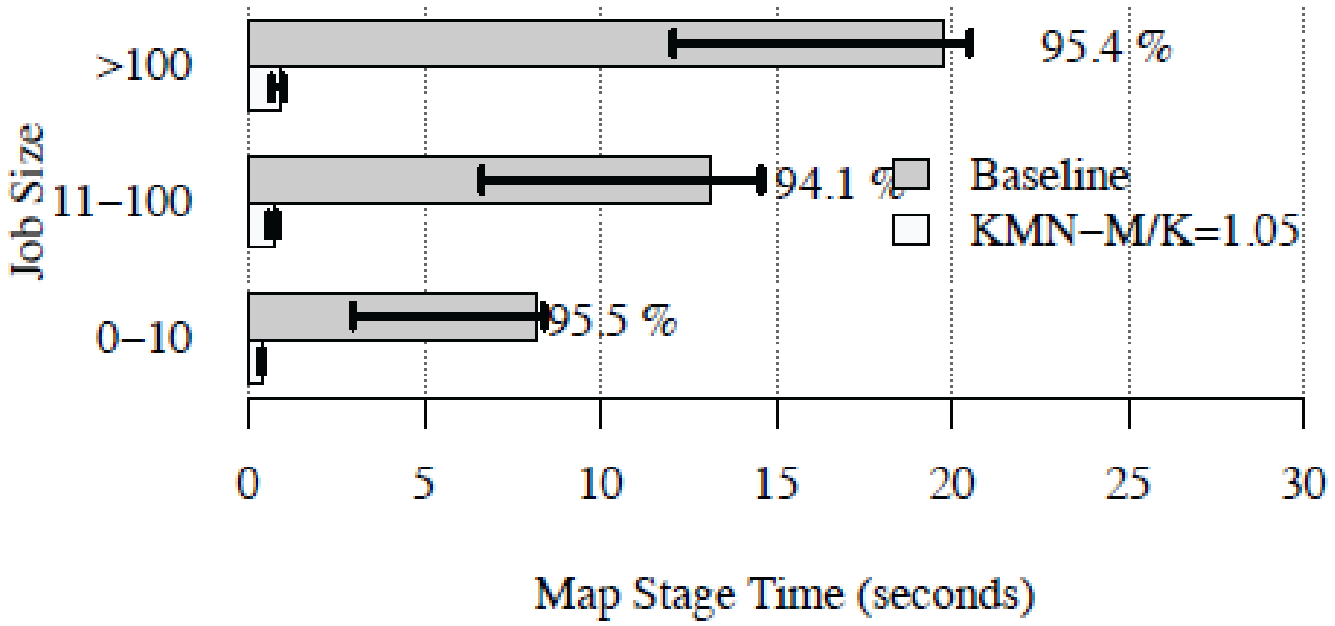
- **Cluster Setup**
  - **100 m2.4xlarge EC2 machines,**
  - **8 cores,60GB RAM, 2 local drives /mc**
- **Workloads**
  - **Facebook trace**
    - A mix of interactive and batch jobs and capture over half a million jobs on a 3500 node cluster
- **Metrics**
  - percentage improvement of job completion time

$$\% \text{ Improvement} = \frac{\text{Baseline Time} - \text{KMN Time}}{\text{Baseline Time}} \times 100$$

Facebook Overall



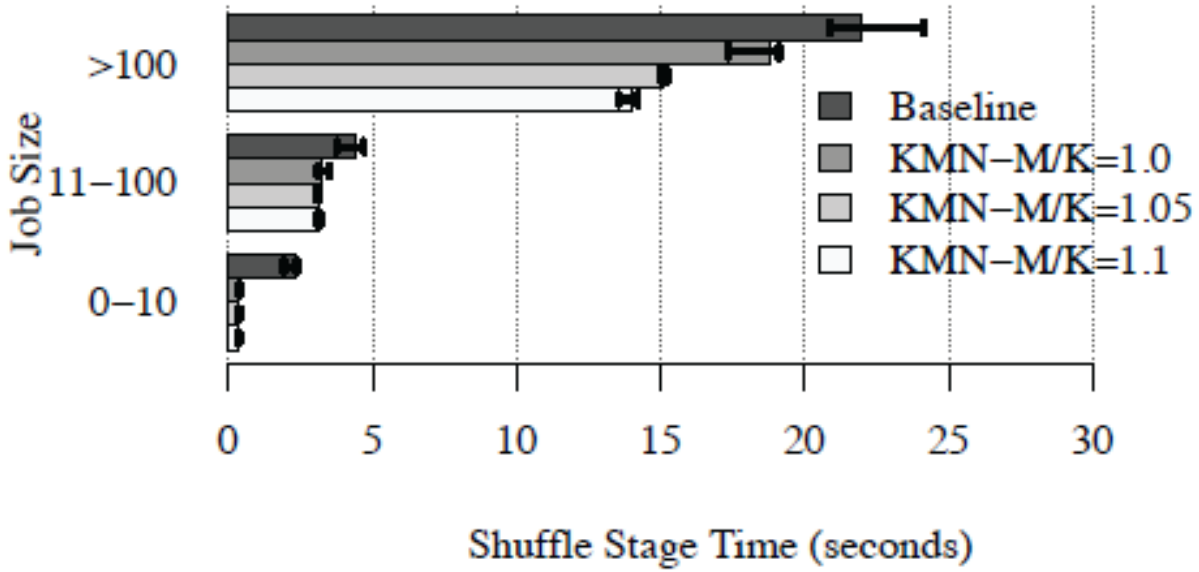
Input locality



Intermediate stage

Job Size	$M/K = 1.0$	$M/K = 1.05$	$M/K = 1.1$
1 to 10	85.04	84.61	83.76
11 to 100	27.5	28.63	28.18
> 100	14.44	31.02	36.35

Table 2: **Shuffle time improvements over baseline while varying M/K**



Conclusion

- **Application trends**
  - **Operate on subsets of data**
- **KMN**
  - **exploit the available choices to improve performance**
    - Increasing locality
    - balancing intermediate data transfers

Thank You