

Scarlett: Coping with Skewed Content Popularity in MapReduce Clusters

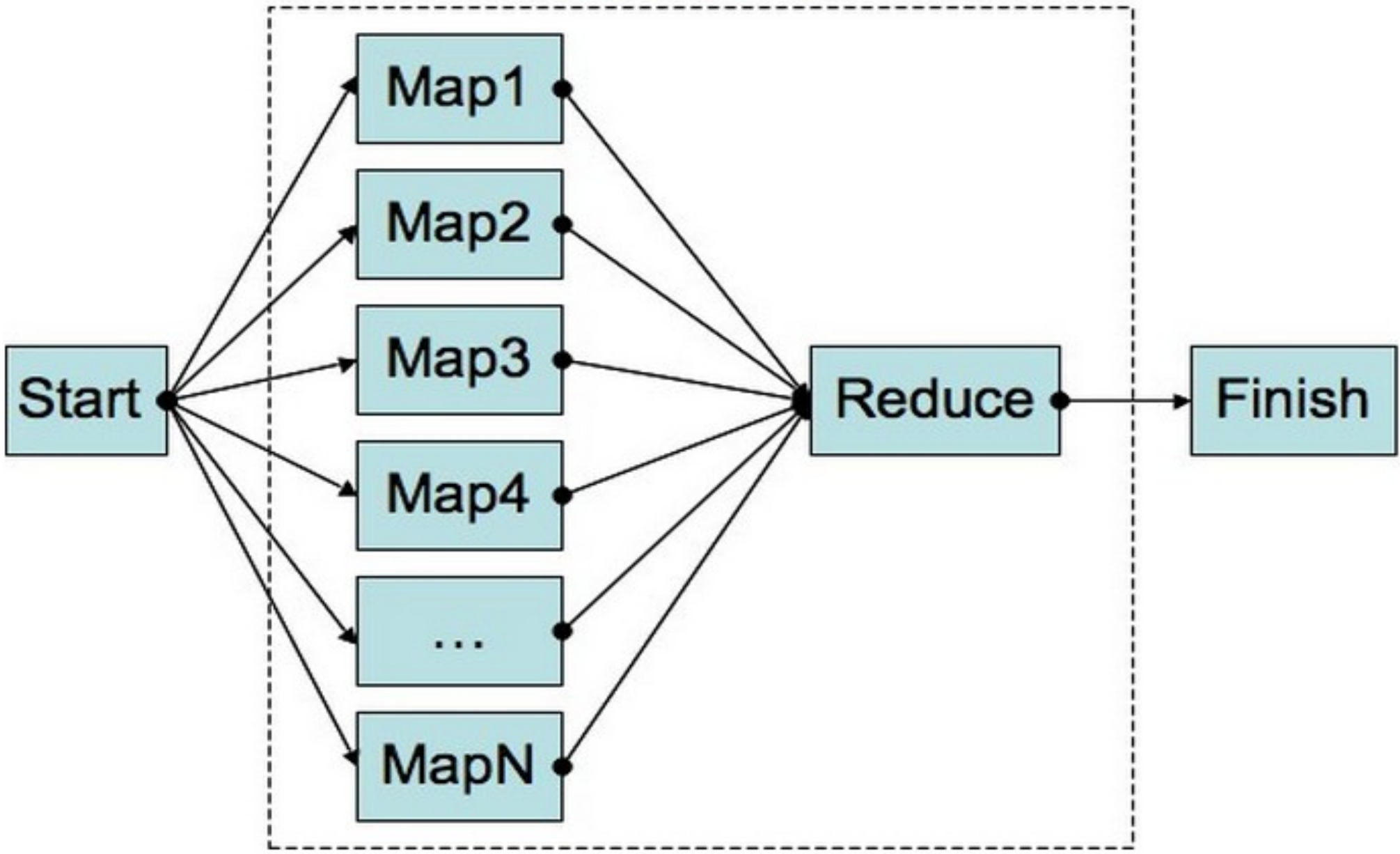
Ganesh Ananthanarayanan, Sameer Agarwal, Srikanth
Kandula, Albert Greenberg, Ion Stoica, DukeHarlan, Ed
Harris

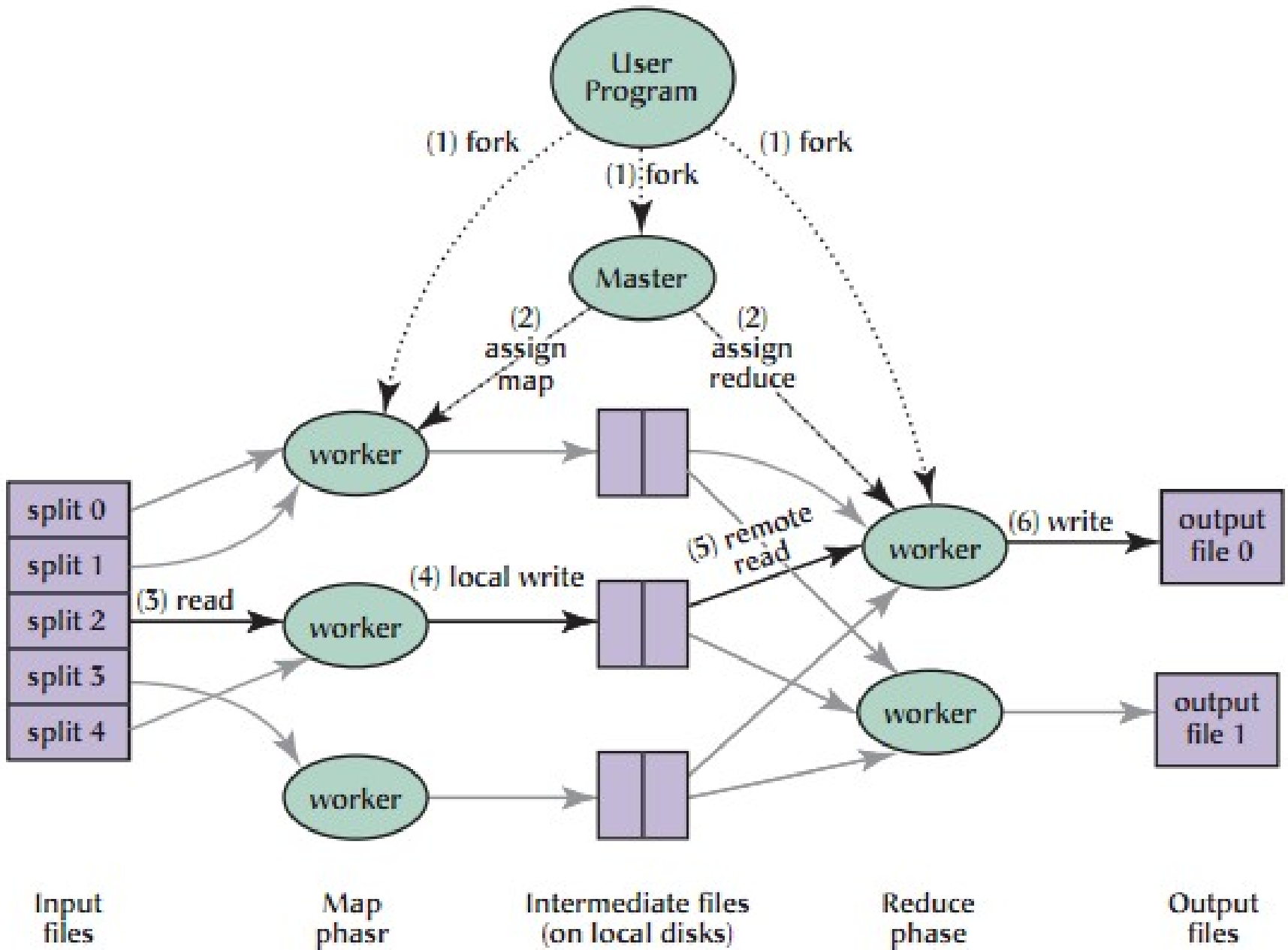
presented by

Paweł Posieleşny



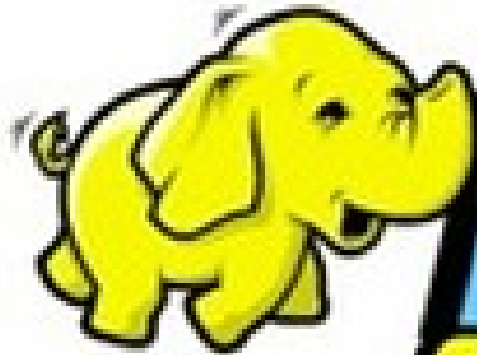
mapreduce





```
map(String key, String value):  
    // key: document name  
    // value: document contents  
    for each word w in value:  
        EmitIntermediate(w, "1");
```

```
reduce(String key, Iterator values):  
    // key: a word  
    // values: a list of counts  
    int result = 0;  
    for each v in values:  
        result += ParseInt(v);  
    Emit(AsString(result));
```



hadoop
Map Reduce





DRYAD



Why Scarlett?



Scarlett uses

- historical usage statistics
- online predictors based on recent past
- information about the jobs that have been submitted for execution



Dates	Phases (x10³)	Jobs (x10³)	Data (PB)	Network (PB)
May 25,29	44.3	23.4	35.5	1.55
Aug 20,24	77.1	48.8	47.7	2.10
Sep 15,19	74.4	40.1	54.0	1.82
Oct 15,19	49.0	33.0	69.5	2.17
Nov 16,20	96.4	45.3	54.4	1.70
Dec 10,14	46.4	42.4	51.9	1.40



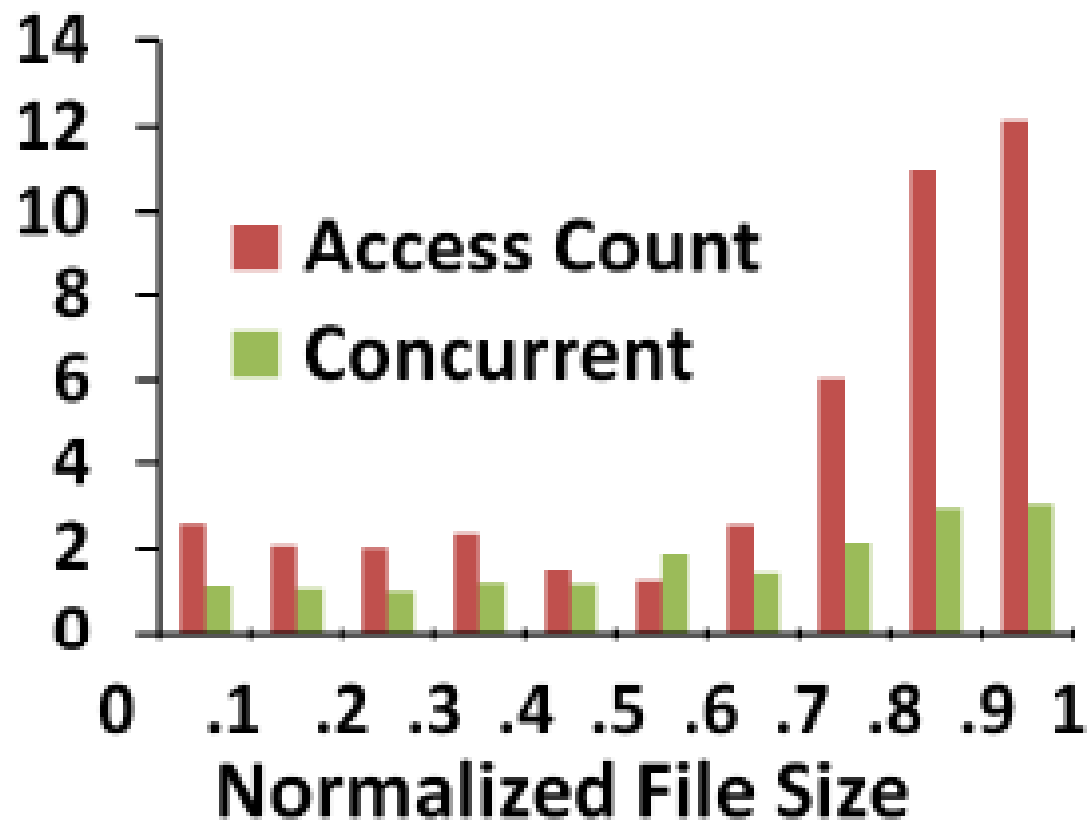


Figure 2: Popularity of files as a function of their sizes, normalized to the largest file; the largest file has size 1. The columns denote the average value (# accesses, concurrence) of files in each of the ten bins.

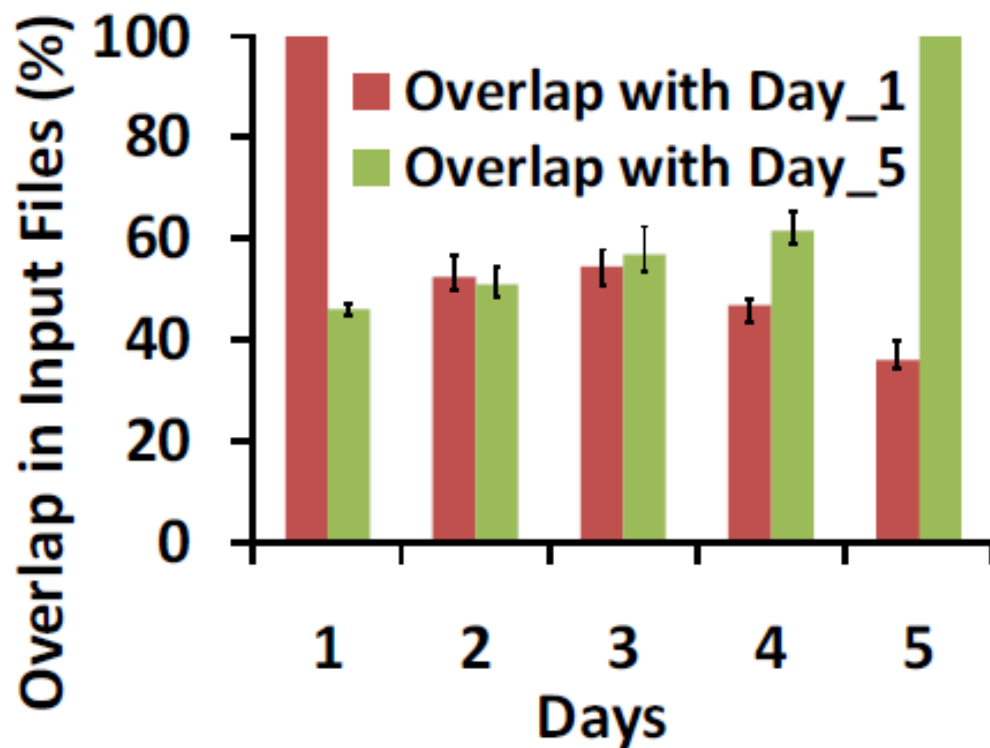


Figure 3: **Overlap in files accessed across five days, for each month listed in Table 1. With the first and fifth day as references, we plot the fraction of bytes accessed on those days that were also accessed in the subsequent and preceding days, respectively.**

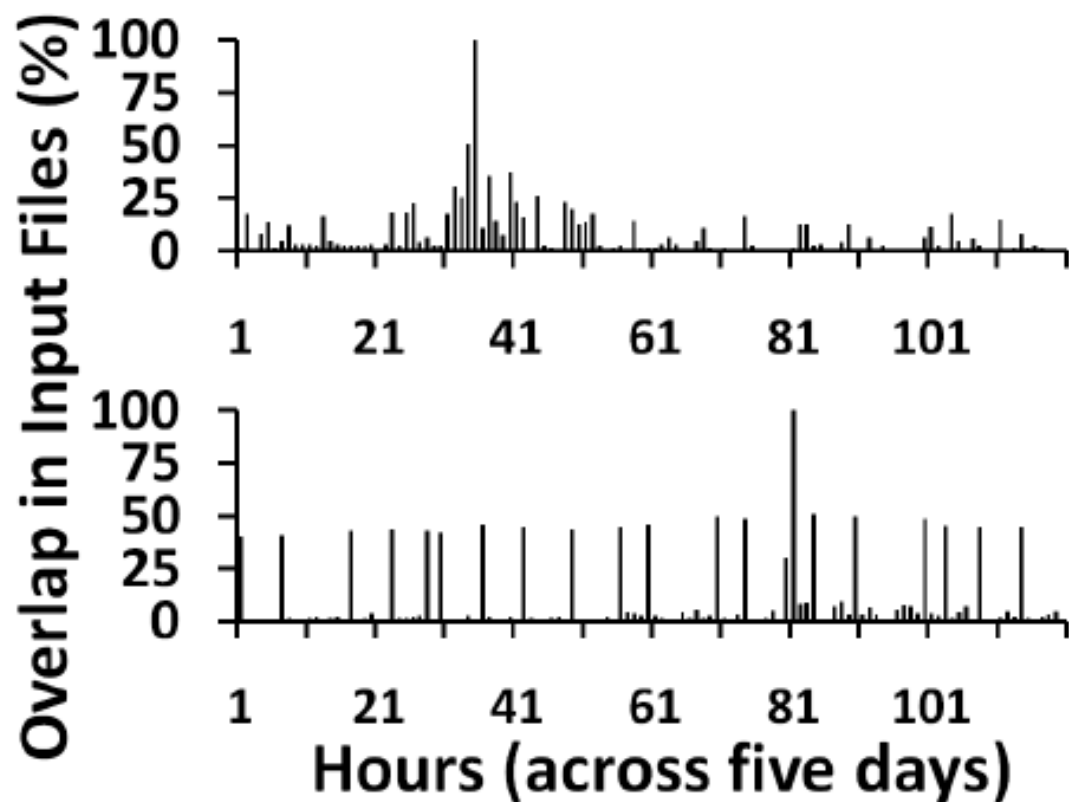


Figure 4: Hourly overlap in the set of files accessed with two sample reference hours (*hour-35* and *hour-82*). The graph on the top shows a gradual change while the bottom graph shows periodically accessed files.

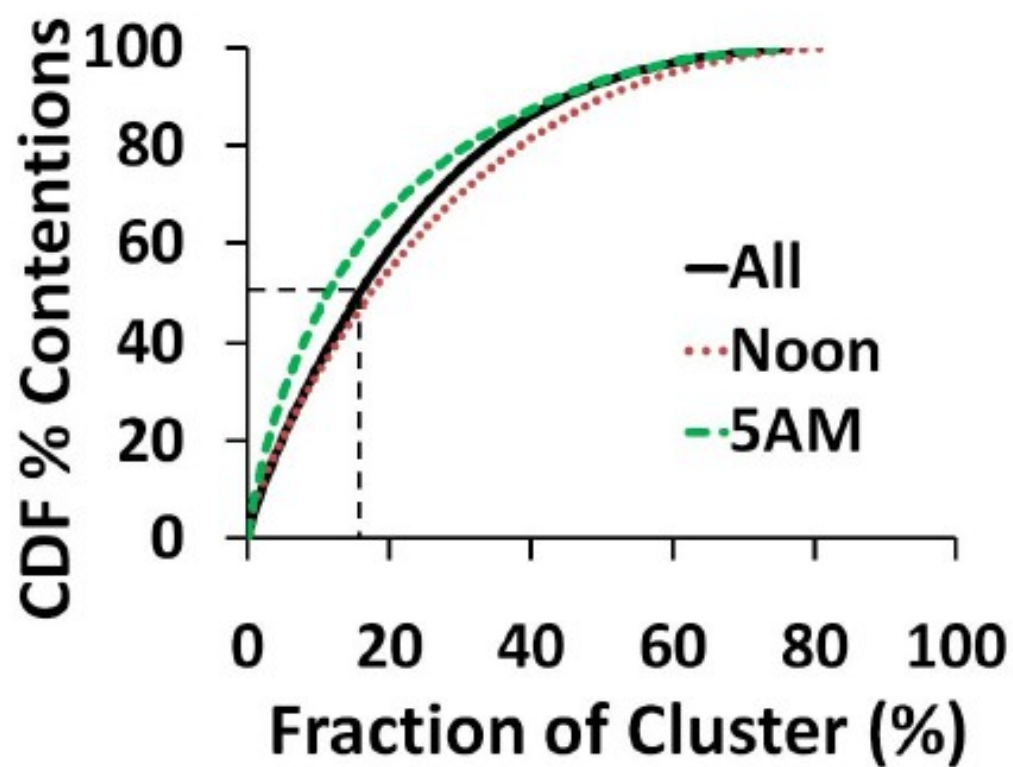


Figure 5: Hotspots: One-sixth of the machines account for half the contentions in the cluster.

Logs summary

- The number of concurrent accesses is a sufficient metric to capture popularity of files.
- Large files contribute to most accesses in the cluster, so reducing contention for such files improves overall performance.
- Recent logs are a good indicator of future access patterns.
- Hotspots in the cluster can be smoothed via appropriate placement of files.



Scarlett: System Design

- Scarlett considers replicating content at the smallest granularity at which jobs can address content (file)
- Scarlett replicates files based on predicted popularity.



File Replication Factor

- maintains a count of the maximum number of concurrent accesses (cf) in a learning window of length TL
- Once every rearrangement period, TR , Scarlett computes appropriate replication factors for all the files.
- $TL = 24$ hours
- $TR = 12$ hours

replication factor: $rf = \max(cf + \delta, 3)$.



Scarlett employs two approaches.

- the priority approach
- round-robin approach



```

Used Budget,  $B_{used} \leftarrow 0$ 
 $F \leftarrow$  Set of files sorted in descending order of size
Set  $r_f \leftarrow 3 \quad \forall f \in F$  ▷ Base Replication
for file  $f \in F$  do
     $r_f \leftarrow \max(c_f + \delta, 3)$  ▷ Increase  $r_f$  to  $c_f + \delta$ 
     $B_{used} \leftarrow B_{used} + f_{size} \cdot (r_f - 3)$ 
    break if  $B_{used} \geq B$ 
end for

```

Pseudocode 1: Scarlett **computes the file replication factor r_f based on their popularity and budget B . c_f is the observed number of concurrent accesses. Here files with larger size have a strictly higher priority of getting their desired number of replicas.**

```

Used Budget,  $B_{used} \leftarrow 0$ 
 $F \leftarrow$  Set of files sorted in descending order of size
Set  $r_f \leftarrow 3 \quad \forall f \in F$  ▷ Base Replication
while  $B_{used} < B$  do
    for file  $f \in F$  do
        if  $r_f < c_f + \delta$  then
             $r_f \leftarrow r_f + 1$  ▷ Increase  $r_f$  by 1
             $B_{used} \leftarrow B_{used} + f_{size}$ 
            break if  $B_{used} \geq B$ 
        end if
    end for
end while

```

Pseudocode 2: Round-robin distribution of the replication budget B among the set of files F .

Desirable properties Scarlett's strategy

- Files that are accessed more frequently have more replicas to smooth their load over.
- Together, δ , TR and TL track changes in file popularity while being robust to short-lived effects.
- Choosing appropriate values for the budget on extra storage B and the period at which replication factors change TR can limit the impact of Scarlett on the cluster.



Smooth Placement of Replicas

place the desired number of replicas of a block on as many distinct machines and racks as possible while ensuring that the expected load is uniform across all machines and racks.



Smooth Placement of Replicas

- load factor for each machine - l_m
- The load factor for each rack – l_r (the sum of load factors of machines in the rack)
- Each replica is placed on the the rack with the least load and the machine with the least load in that rack.
- Placing a replica increases both these factors by the expected load due to that replica ($= cf/rf$).




```

for file  $f$  in  $F$  do
  if  $r_f > r_f^{desired}$  then
    Delete Replicas ▷ De-replicate
    Update  $l_m$  accordingly
  end if
end for
for file  $f$  in  $F$  do
  while  $r_f < r_f^{desired}$  do
    for blocks  $b \in f$  do
       $m^* \leftarrow \arg \min(l_m) \forall$  machines  $m$ 
      Replicate( $b$ ) at  $m^*$ 
       $l_{m^*} \leftarrow l_{m^*} + \frac{c_f}{r_f}$  ▷ Update load
    end for
     $r_f \leftarrow r_f + 1$ 
  end while
end for

```

Creating Replicas Efficiently

- While Replicating, Read From Many Sources
- Compress Data Before Replicating
- Lazy Deletion



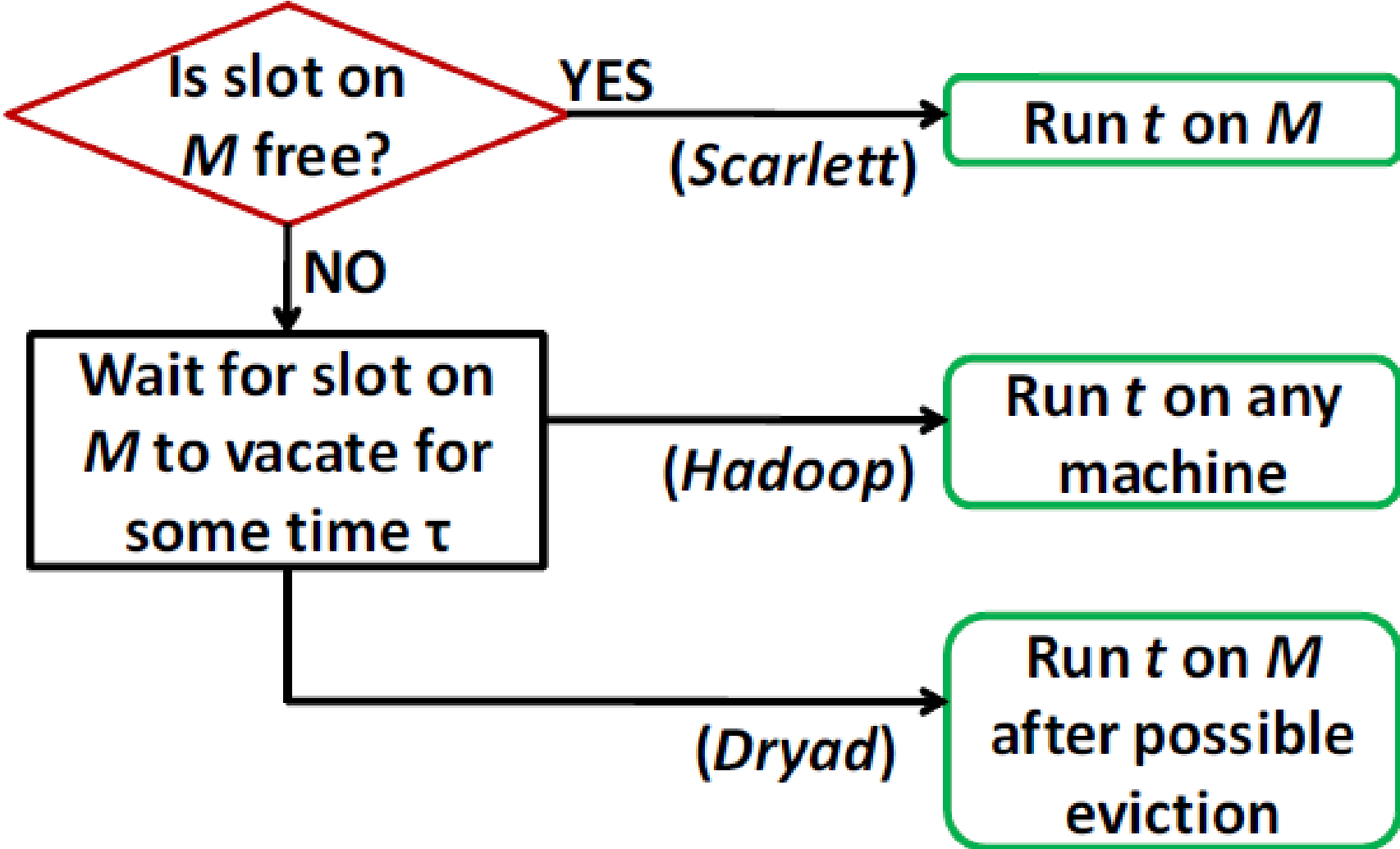
Case Studies of Frameworks

How to deal with a task that cannot run at the machine(s) that it prefers to run at?

- less preferred tasks can be evicted to make way
- the newly arriving task can be forced to run at a suboptimal location in the cluster
- one of the contending tasks can be paused until contention passes



Task t wants to run on machine M for data locality



Evictions in Dryad

- Evicted task is given a 30s notice period before being evicted.
- Of all tasks that began running on the cluster, 21.1% of them end up being evicted



Loss of Locality in Hadoop

- achieve only 5% node locality and 59% rack locality.
(data from Facebook's Hadoop's logs)



Evaluation

Methodology:

- using an implementation of Hadoop
- using an extensive simulation of Dryad
- sensitivity analysis
- budget size and distribution
- compression techniques



Does data locality improve in Hadoop?

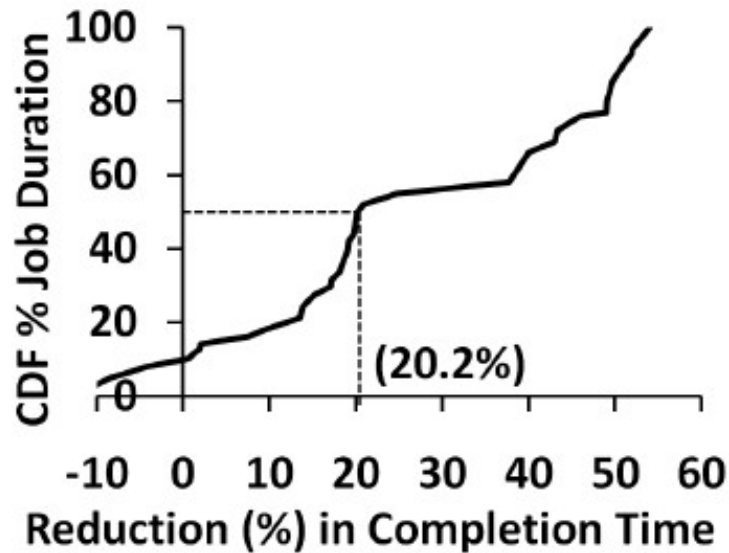


Figure 10: Improvement in data locality for tasks leads to median and third-quartile improvements of 20.2% and 44.6% in Hadoop job completion times.

$$\delta = 1$$

TL range from 6 to 24 hours

TR \geq 10 hours

$$B = 10\%$$

completion times of 500 jobs.



Is eviction of tasks prevented in Dryad?

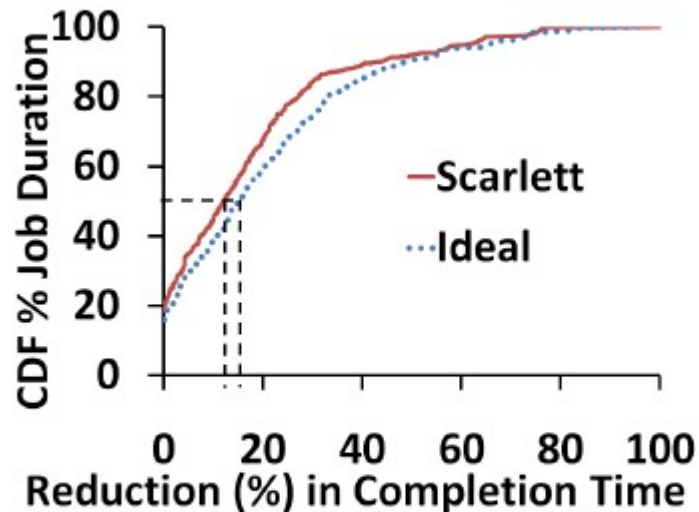


Figure 11: Increased replication reduces eviction of tasks and achieves a median improvement of 12.8% in job completion times, or 84% of ideal.

$$\delta = 1$$

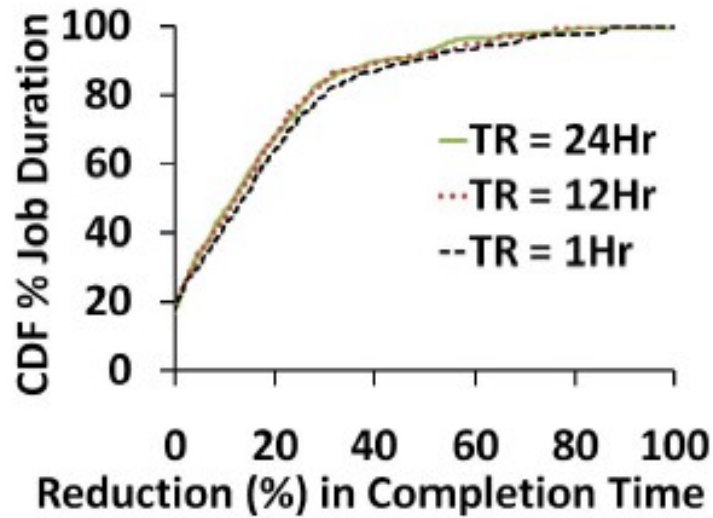
TL range from 6 to 24 hours

TR = 12 hours

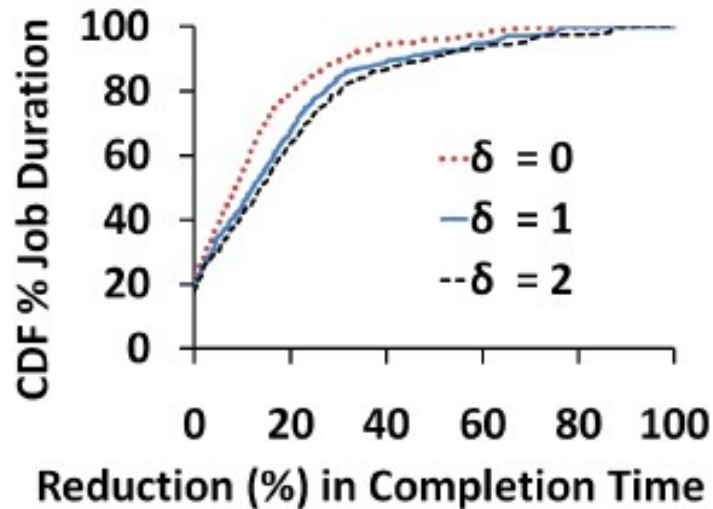
B = 10%



Sensitivity Analysis



(a) Rearrangement Window (T_R)



(b) Replication Allowance (δ)

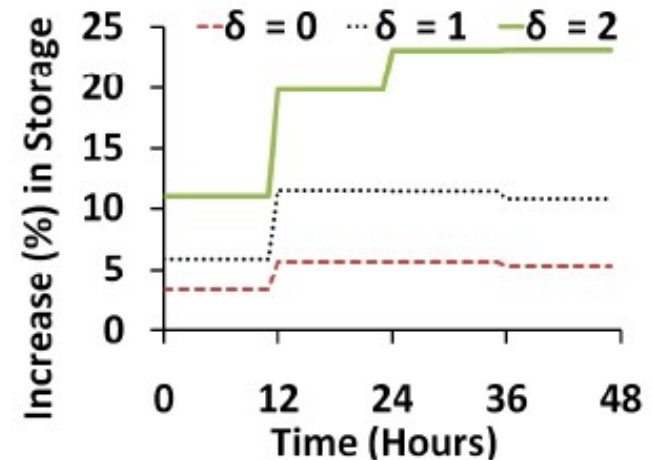
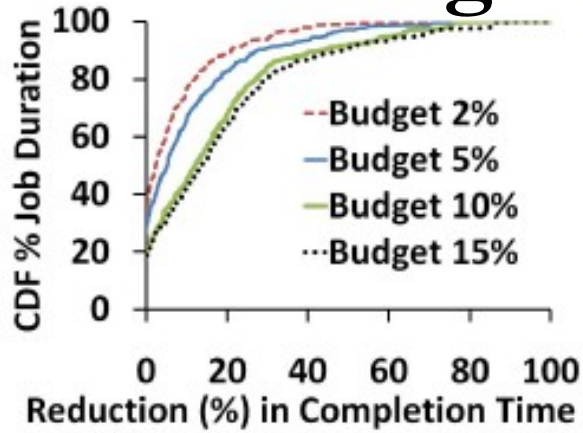
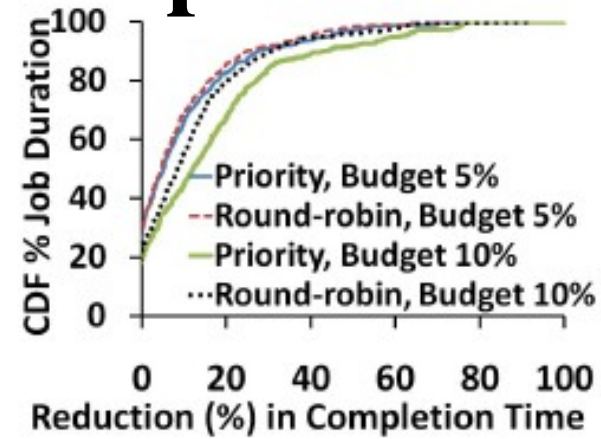


Figure 13: Increasing the value of the replication allowance (δ) leads to Scarlett using more storage space. We fix δ as 1.

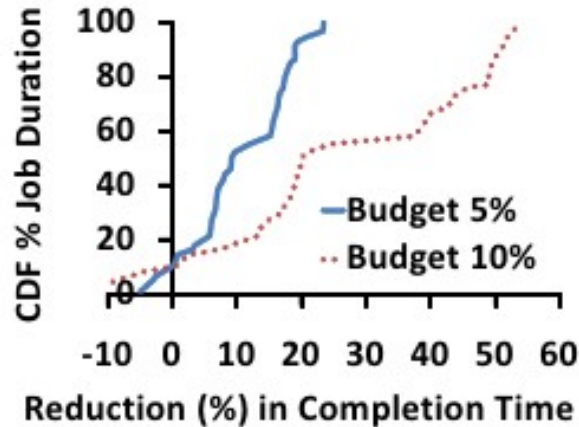
Storage Budget for Replication



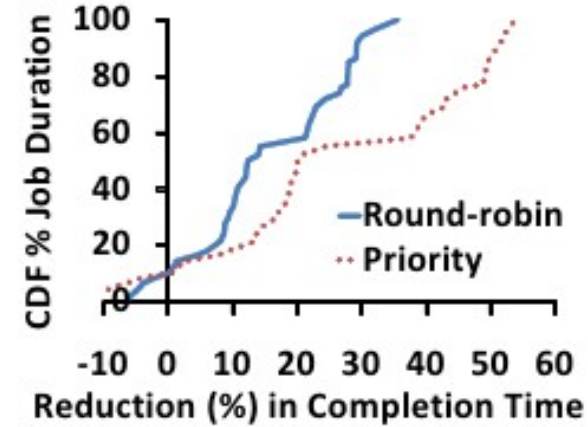
(a) Dryad



(a) Dryad



(b) Hadoop



(b) Hadoop

Figure 14: Low budgets lead to little fruitful replication. On the other hand, as the graph below shows, budgets cease to matter beyond a limit.

Figure 15: Priority distribution of the replication budget among the files improves the median completion time more than round-robin distribution.

Increase in Network Traffic

Scheme	Throughput (Mbps)		Compression Factor
	Compress	De-compress	
<i>gzip</i>	144	413	12-13X
<i>bzip2</i>	9.7	88.2	19-20X
<i>LZMA</i>	3.6	375	22-23X
<i>PPMVC</i>	30.2	31.4	26-27X

Table 2: Comparison of the computational overhead and compression factors of compression schemes.

Benefits from selective replication

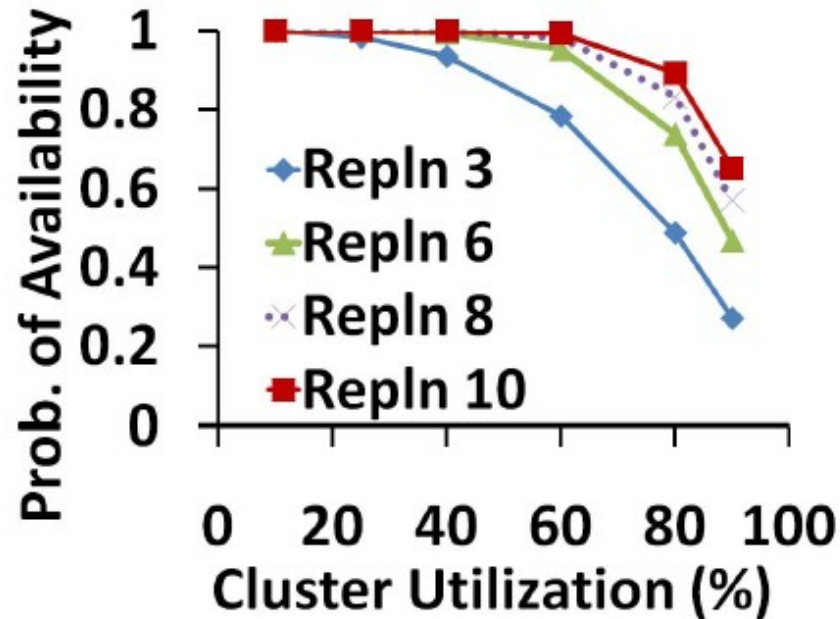


Figure 7: The probability of finding a replica on a free machine for different values of file replication factor and cluster utilization.

Summary

Scarlett uses:

- historical usage statistics
- Scarlett uses online predictors based on recent past
- Scarlett uses information about the jobs that have been submitted for execution

Scarlett replicates files based on predicted popularity.



Thank you



Any questions?

