A Case for Packing and Indexing in Cloud File Systems

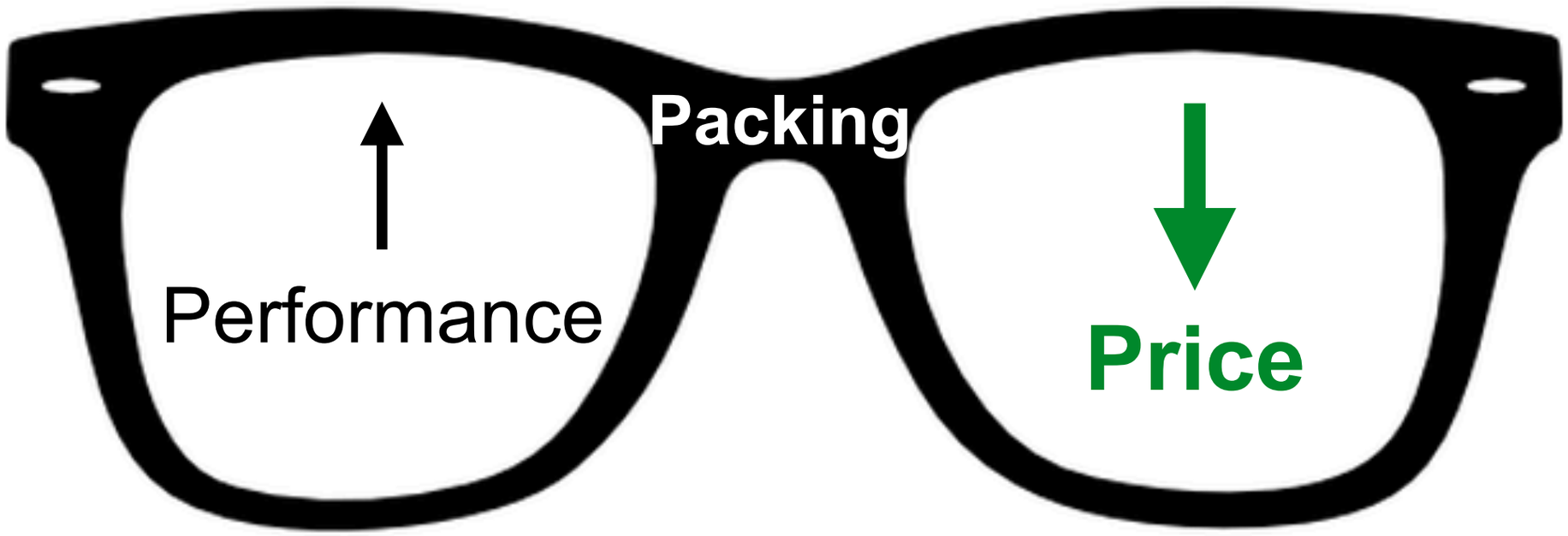
Saurabh Kadekodi, Bin Fan*, Adit Madan*, Garth Gibson and Greg Ganger

Parallel Data Laboratory , *Alluxio Inc.
Carnegie Mellon University

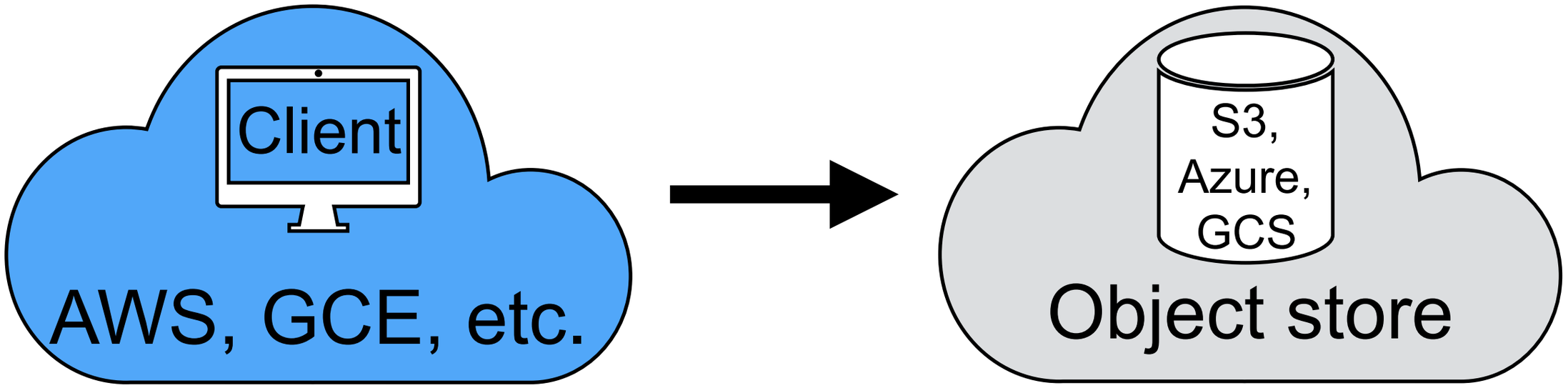
Carnegie Mellon
Parallel Data Laboratory

In a nutshell

- Cloud object stores have a per-operation pricing structure
- Small object workloads have poor performance and high cost

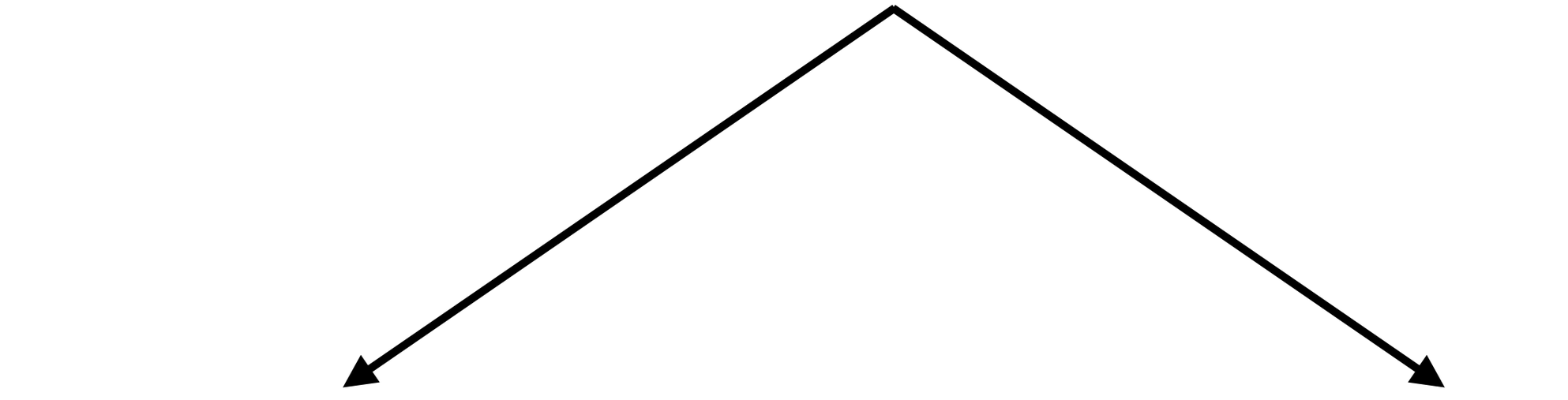


Cloud file systems

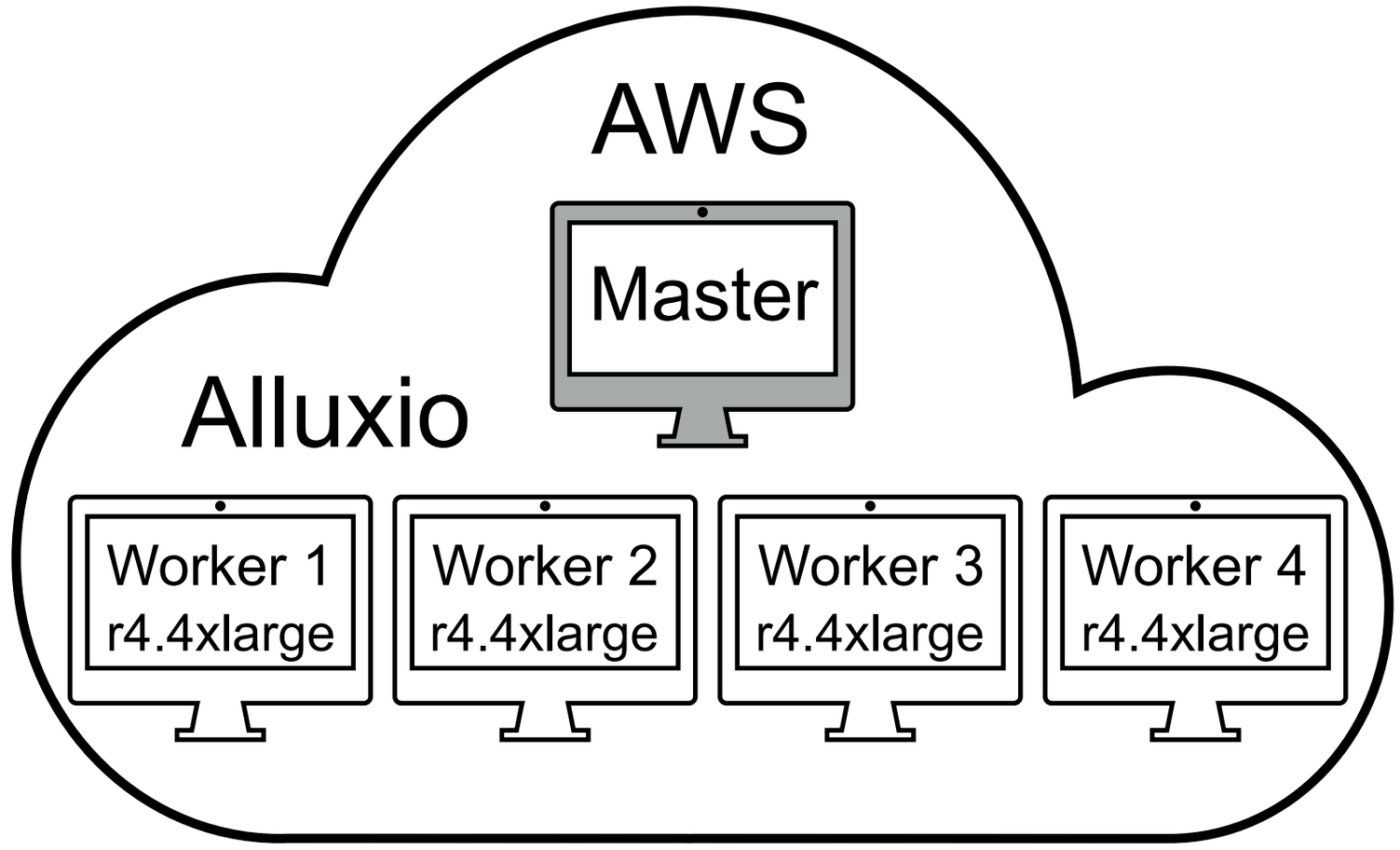


- Natural approach is 1:1 file to object mapping; bad for small files
- Packing (batching or coalescing) small files improves local FS performance
- How much does packing help in cloud file systems?

Motivation



Performance Motivation



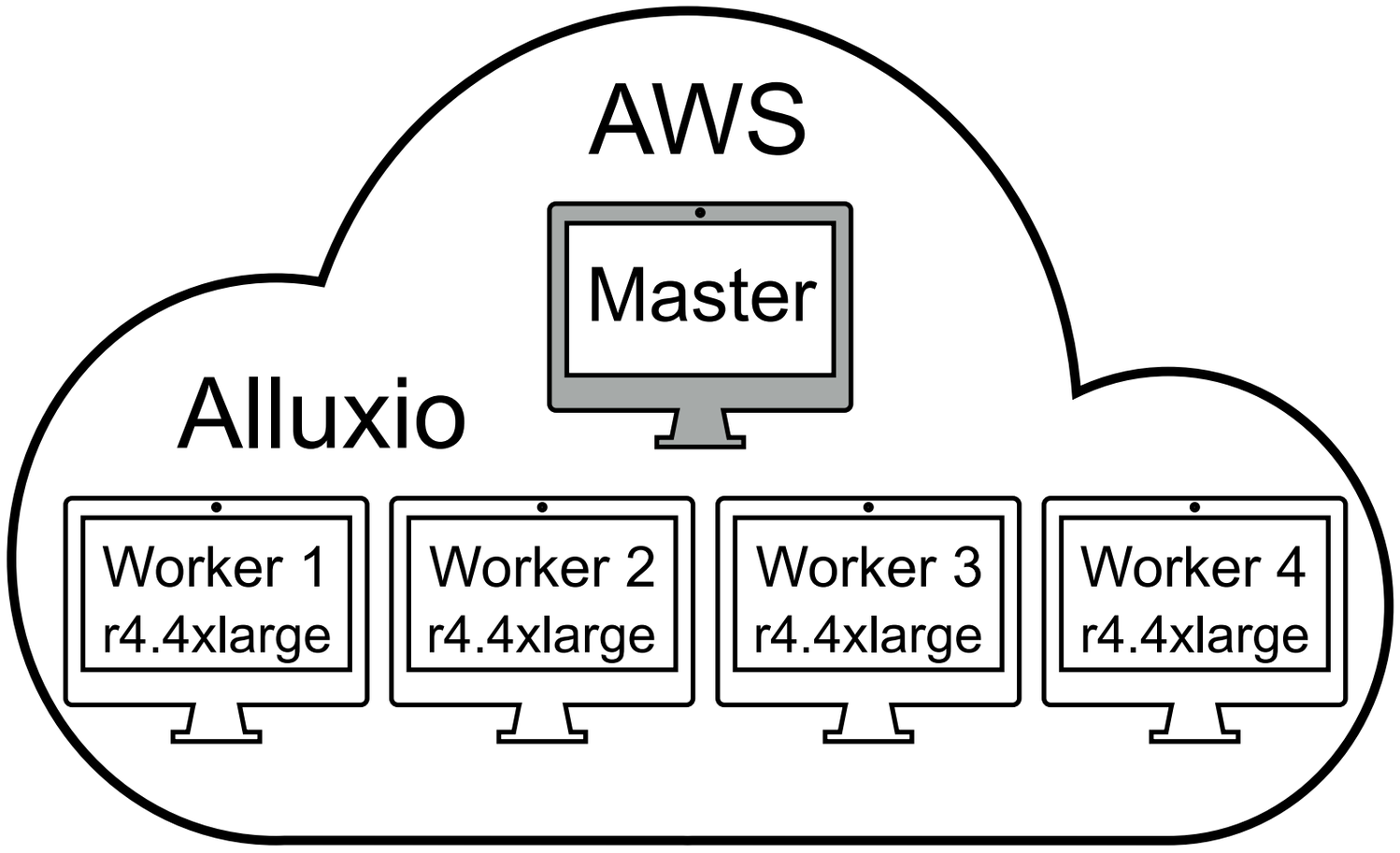
4KiB objects

1 object per 1 file

to S3

3.2M objects of 4KiB each = 12.5GiB

Performance Motivation



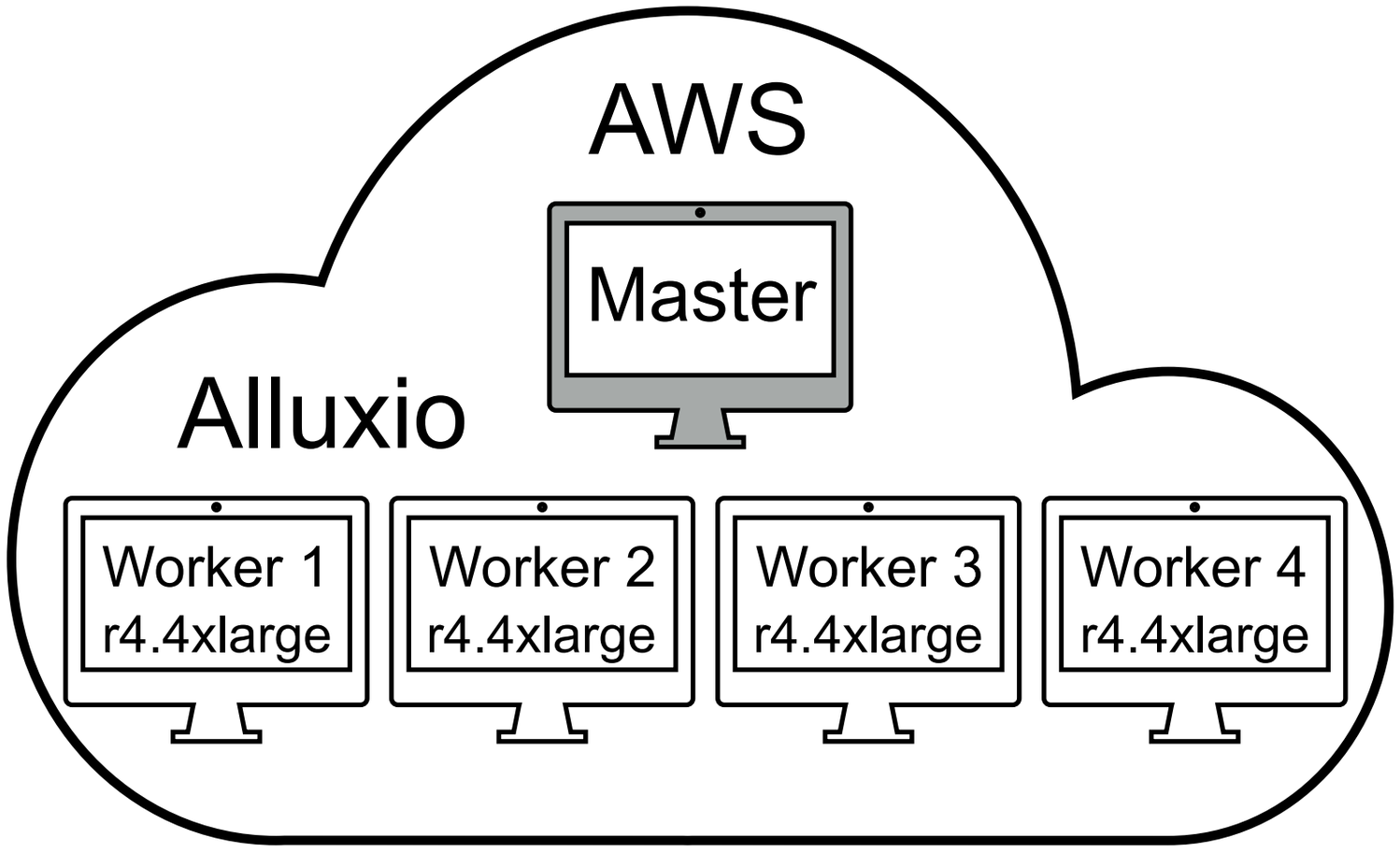
4MiB objects

1 object per 1K files

to S3

3.2K objects of 4MiB each = 12.5GiB

Performance Motivation



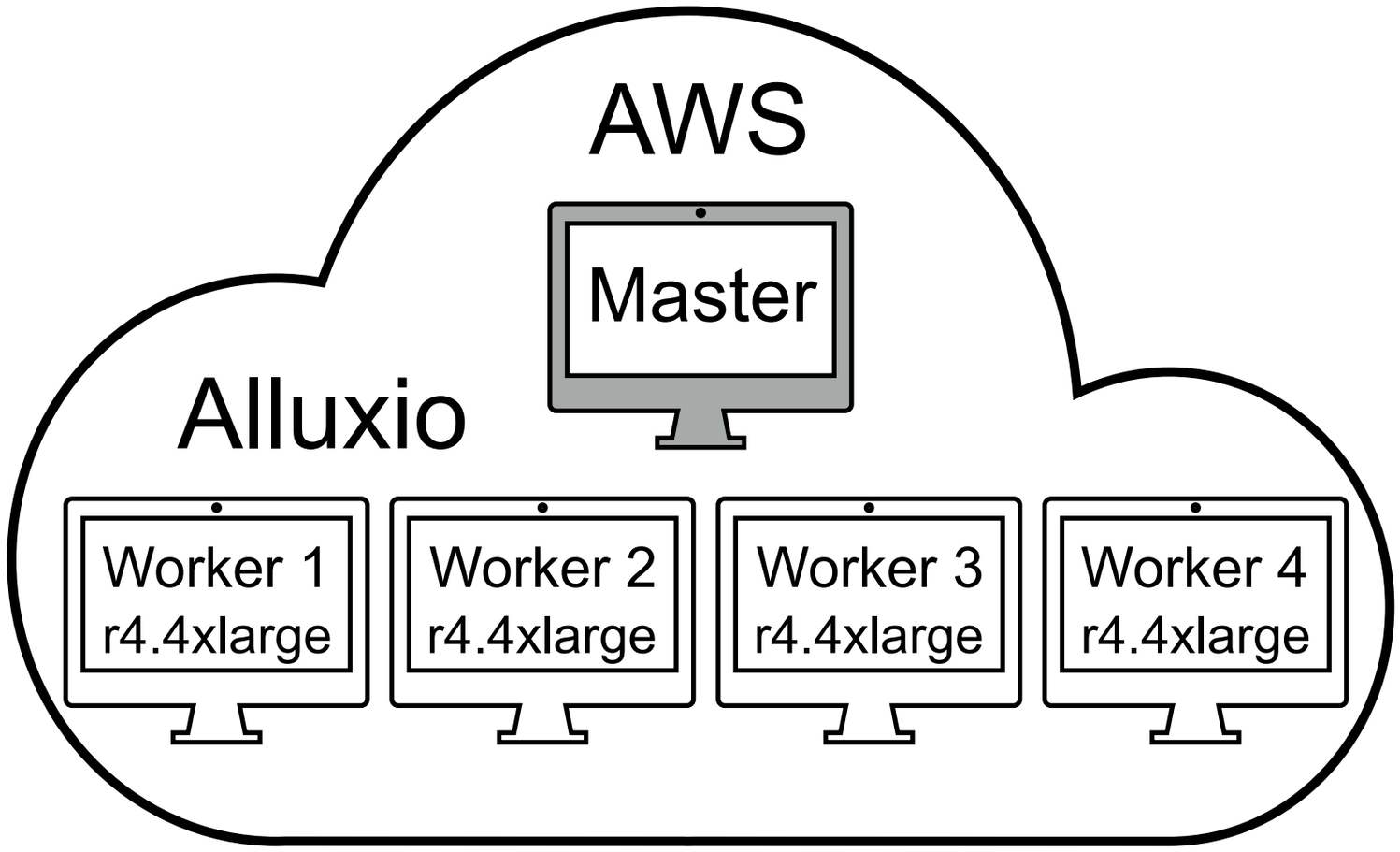
40MiB objects

1 object per 10K files

to S3

320 objects of 40MiB each = 12.5GiB

Performance Motivation



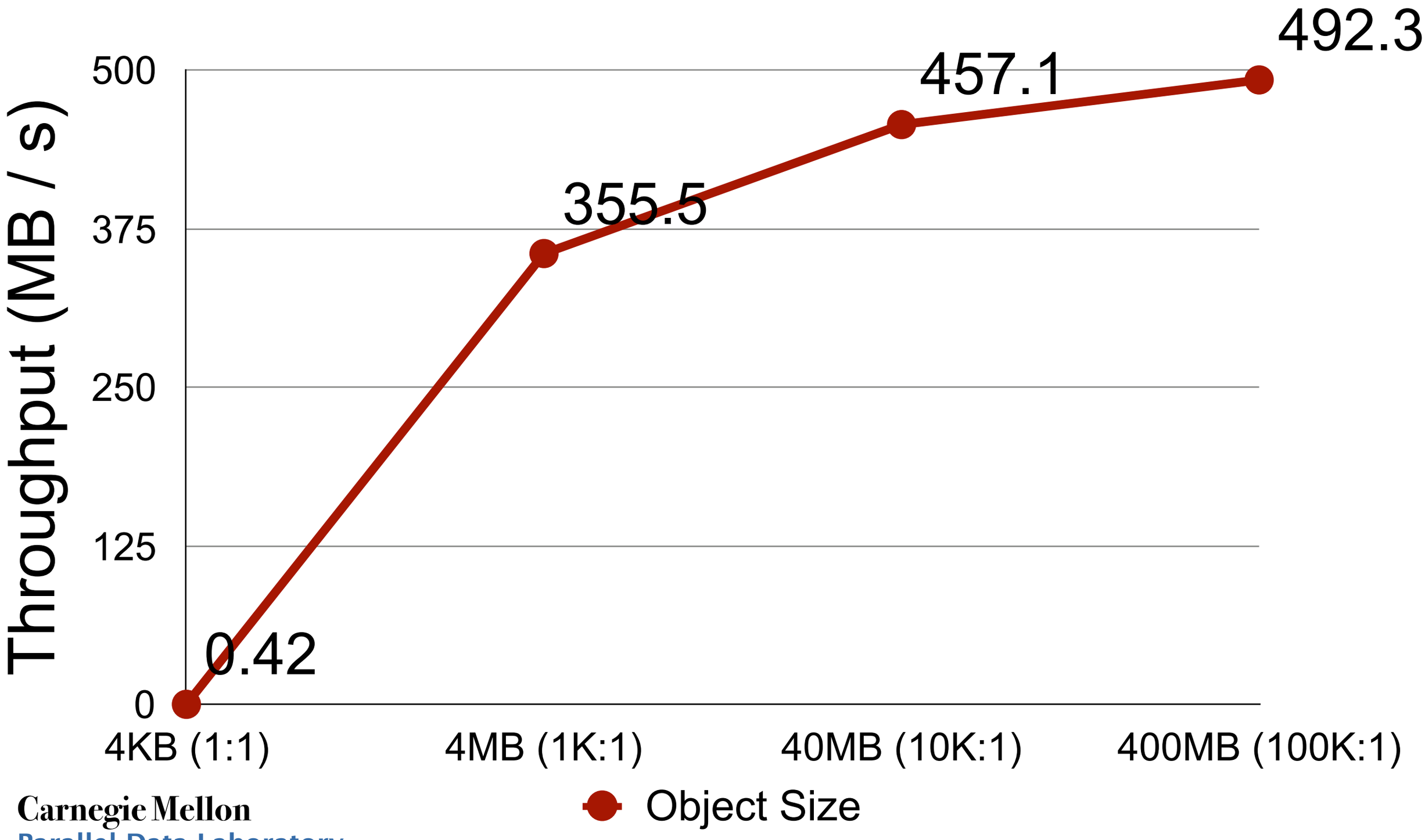
400MiB objects

1 object per 100K files

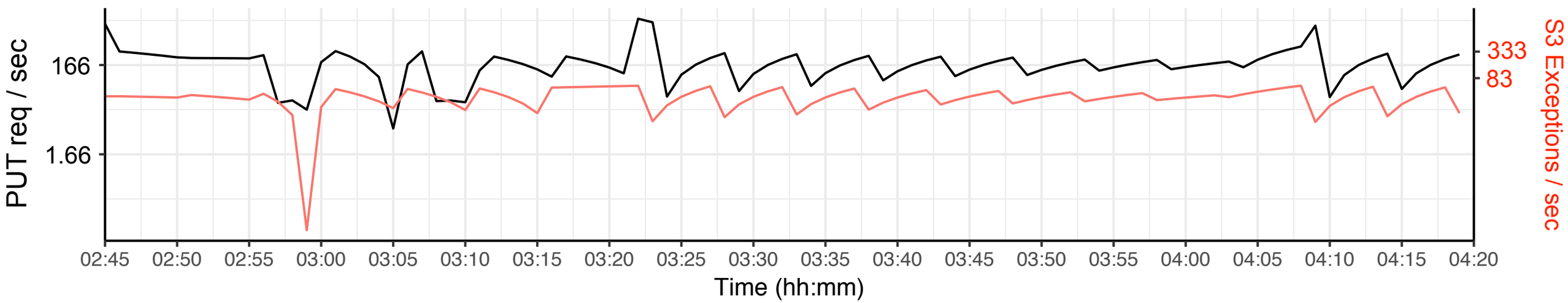
to S3

32 objects of 400MiB each = 12.5GiB

Throughput By Object Size



S3 Throttling in Action

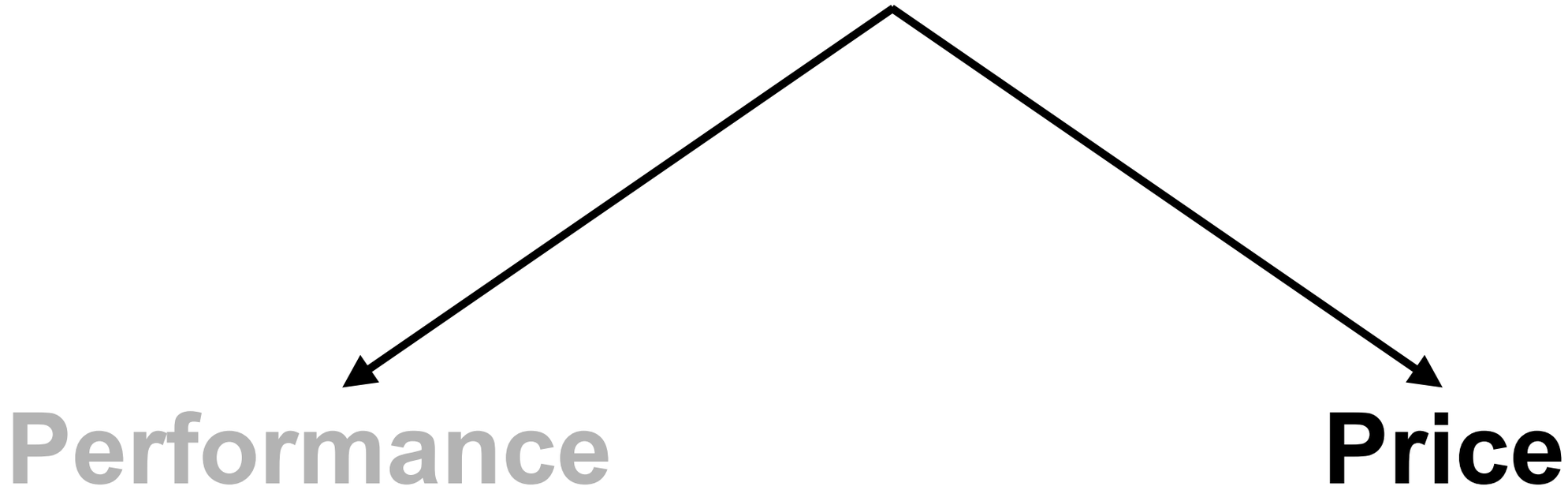


- Request rate throttling by S3 for 4KiB writes

- S3 warns routinely requesting >100 PUT req / s subject to throttling

- NW bandwidth of r4.4xlarge instances ~ 10 Gb / sec → minimum 13MB writes to avoid being throttled → packing

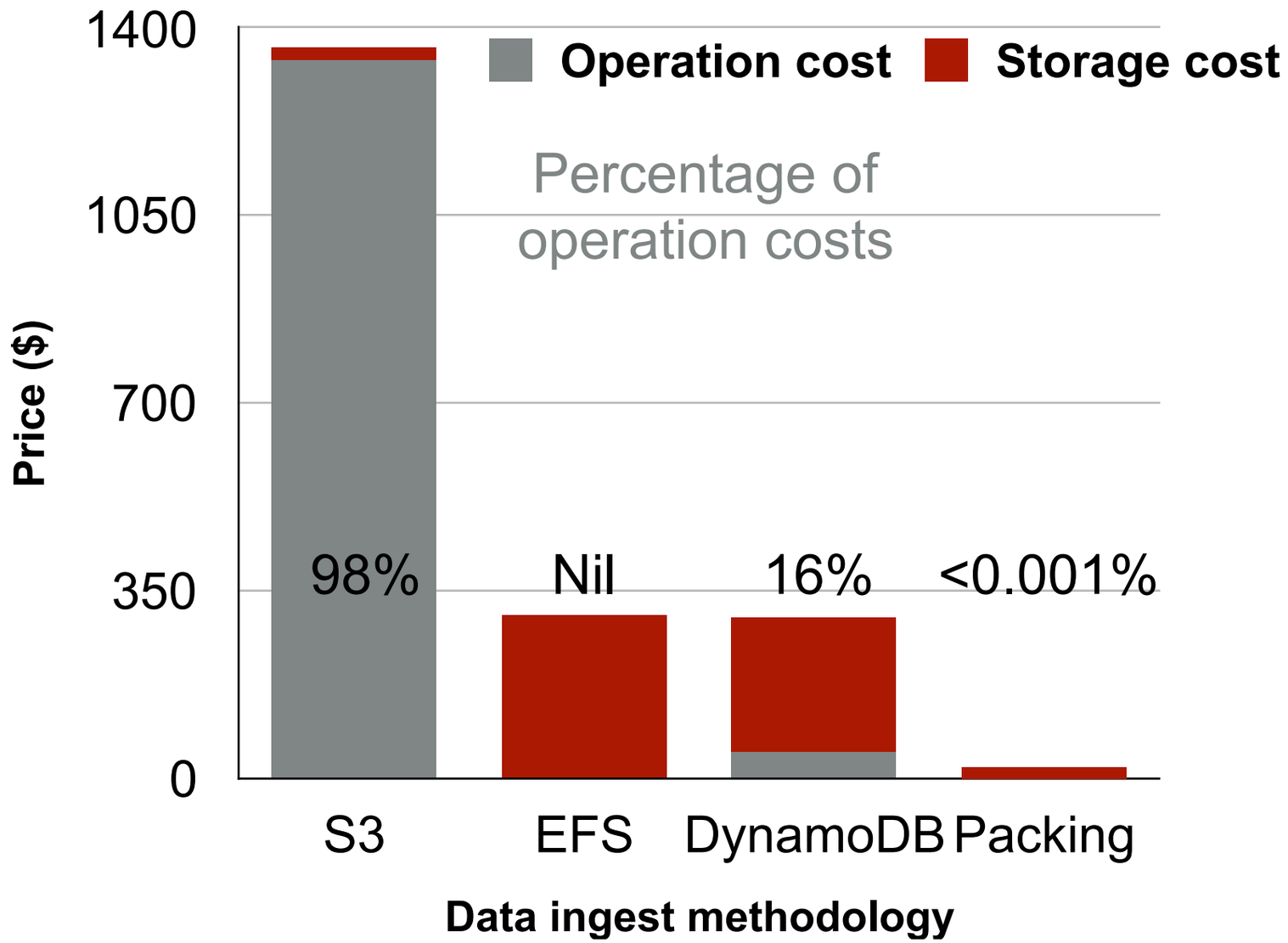
Motivation



Price Motivation

S3	PUT, COPY, POST	GET	Data Retrieval	Data Storage
Standard	$0.05 / 10K req	$0.04 / 100K req	Free (for certain data center locations)	$0.023 / GB-month

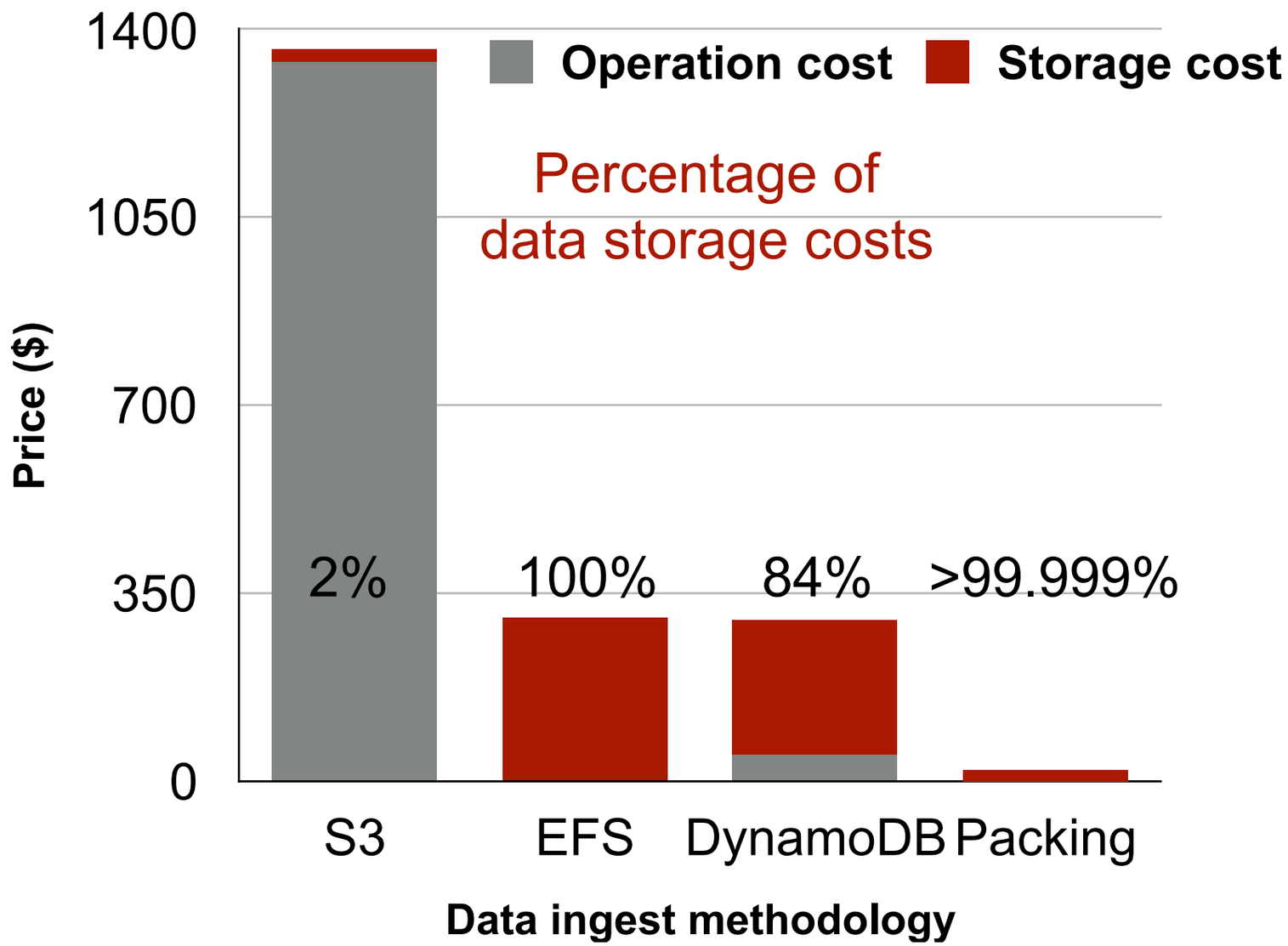
- Say we want to store 1TB per month as 4KiB objects, approx 100 req/s for a month



Price Motivation

S3	PUT, COPY, POST	GET	Data Retrieval	Data Storage
Standard	$0.05 / 10K req	$0.04 / 100K req	Free (for certain data center locations)	$0.023 / GB-month

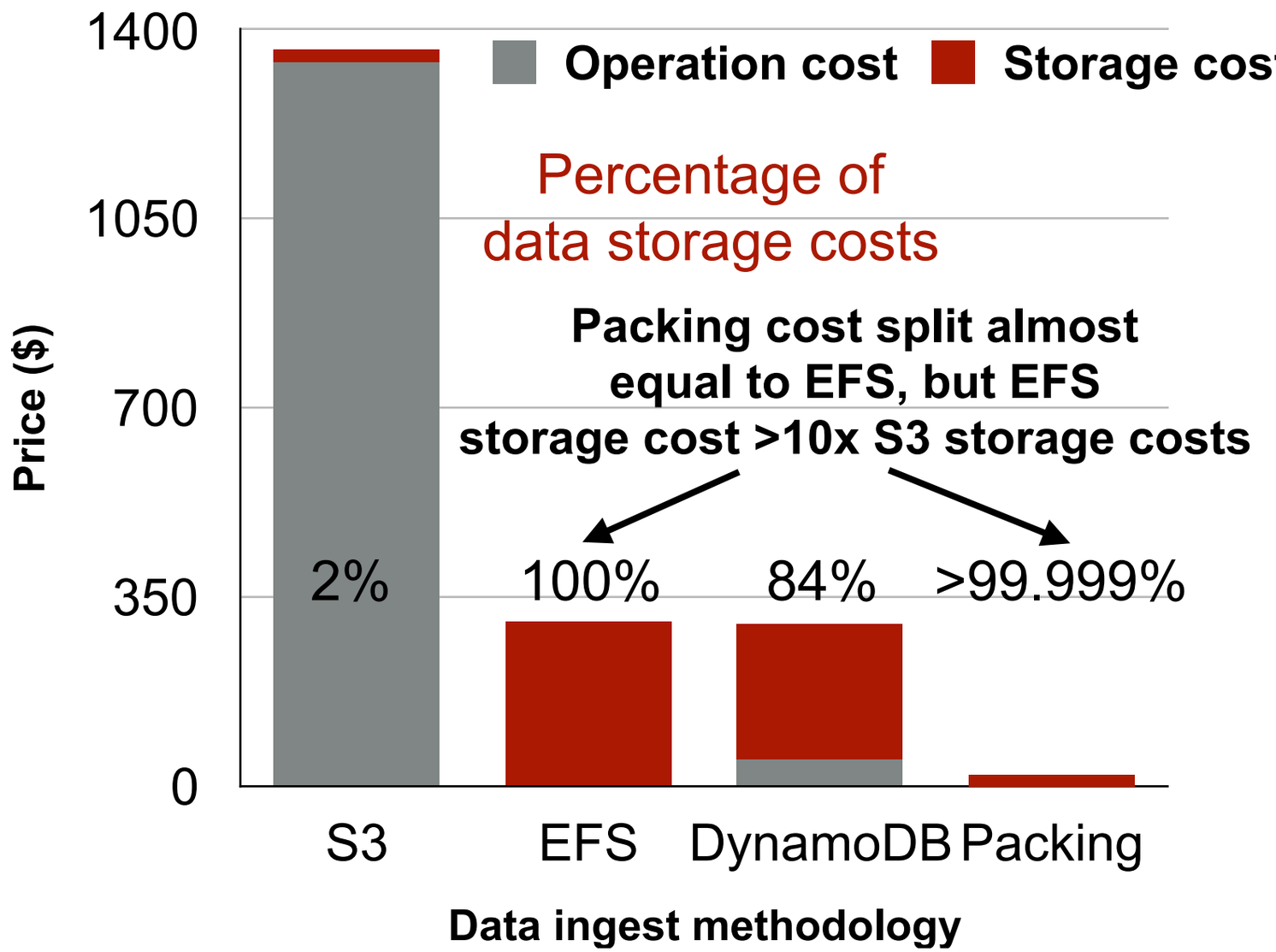
- Say we want to store 1TB per month as 4KiB objects, approx 100 req/s for a month



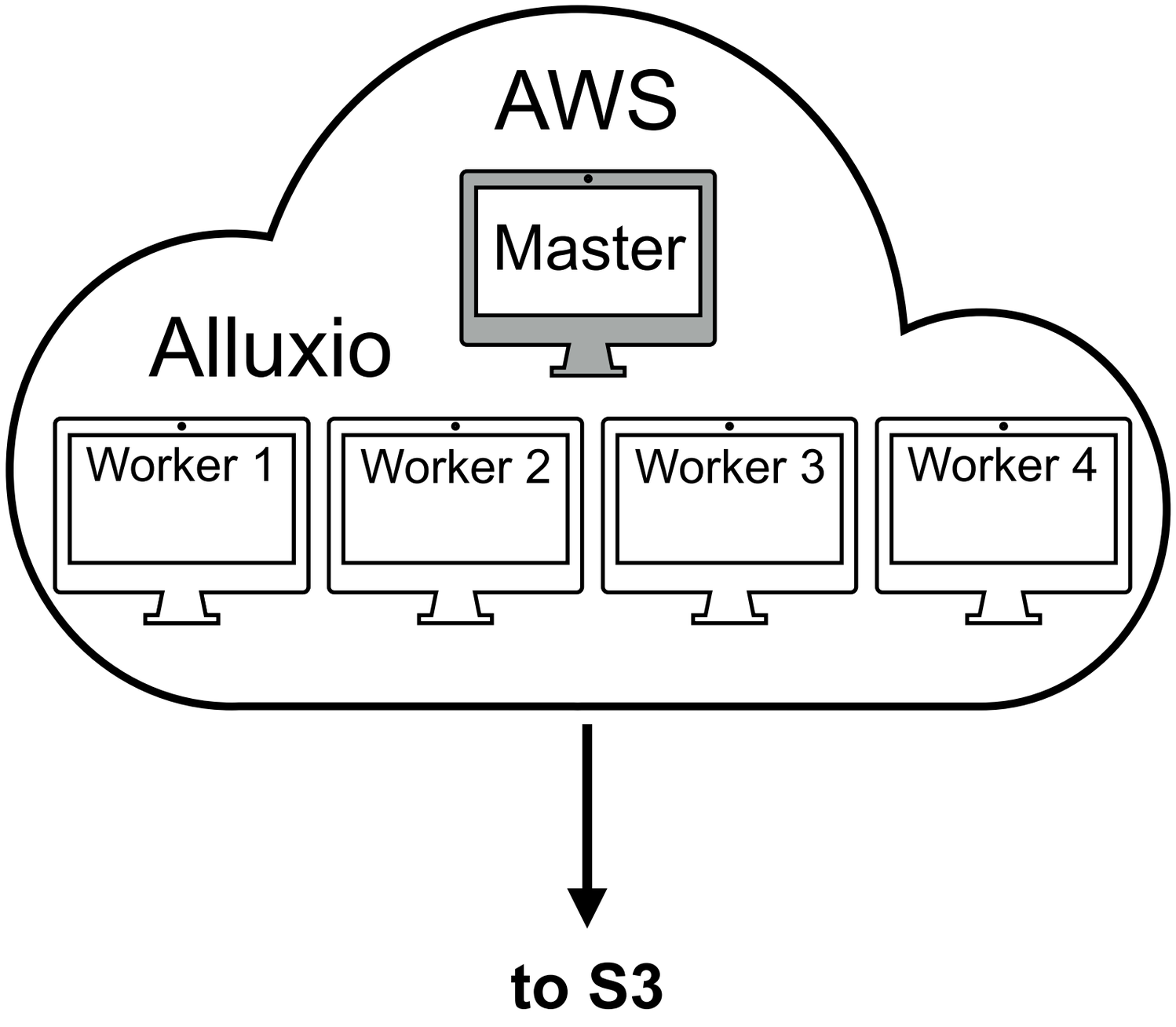
Price Motivation

S3	PUT, COPY, POST	GET	Data Retrieval	Data Storage
Standard	$0.05 / 10K req	$0.04 / 100K req	Free (for certain data center locations)	$0.023 / GB-month

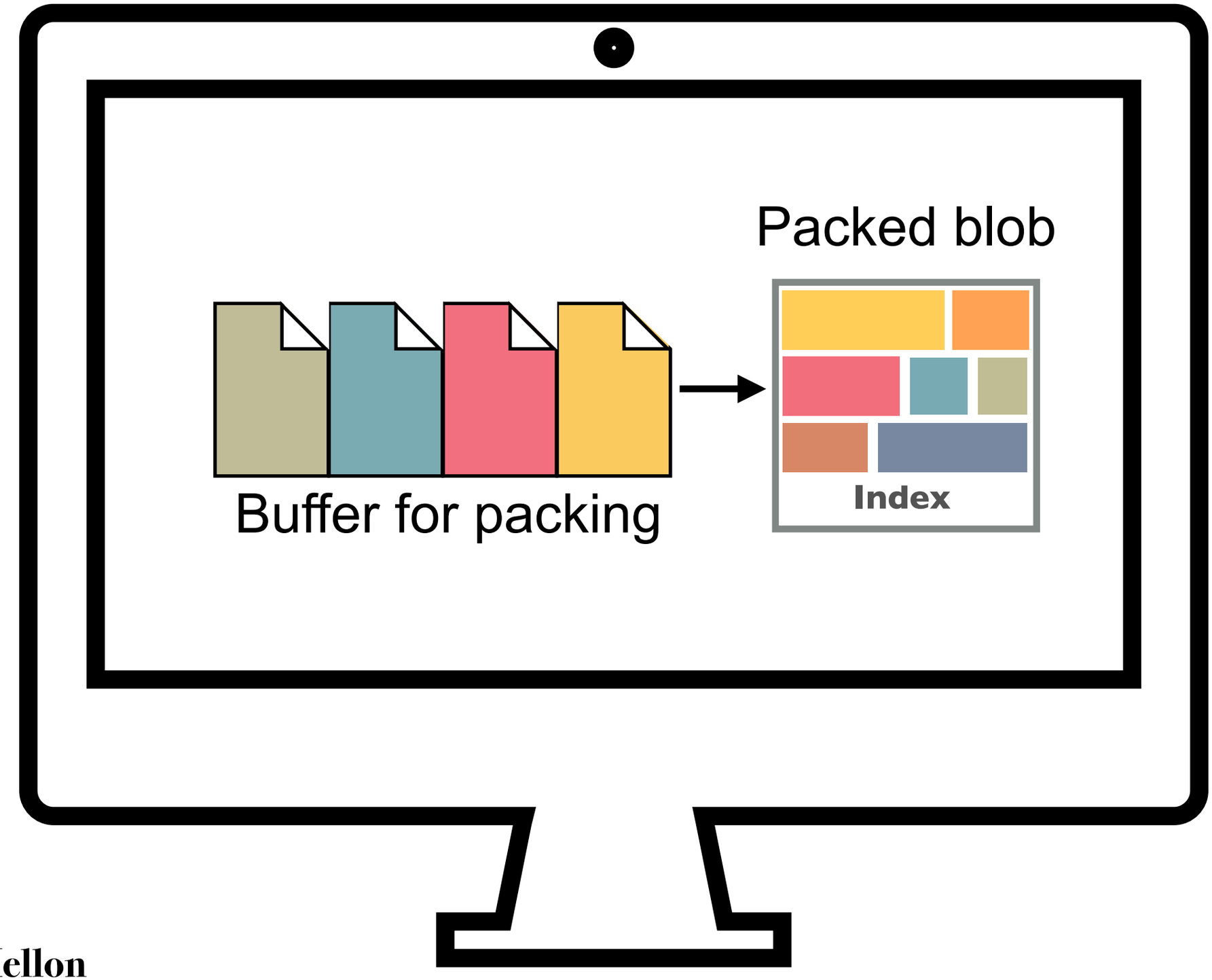
- Say we want to store 1TB per month as 4KiB objects, approx 100 req/s for a month



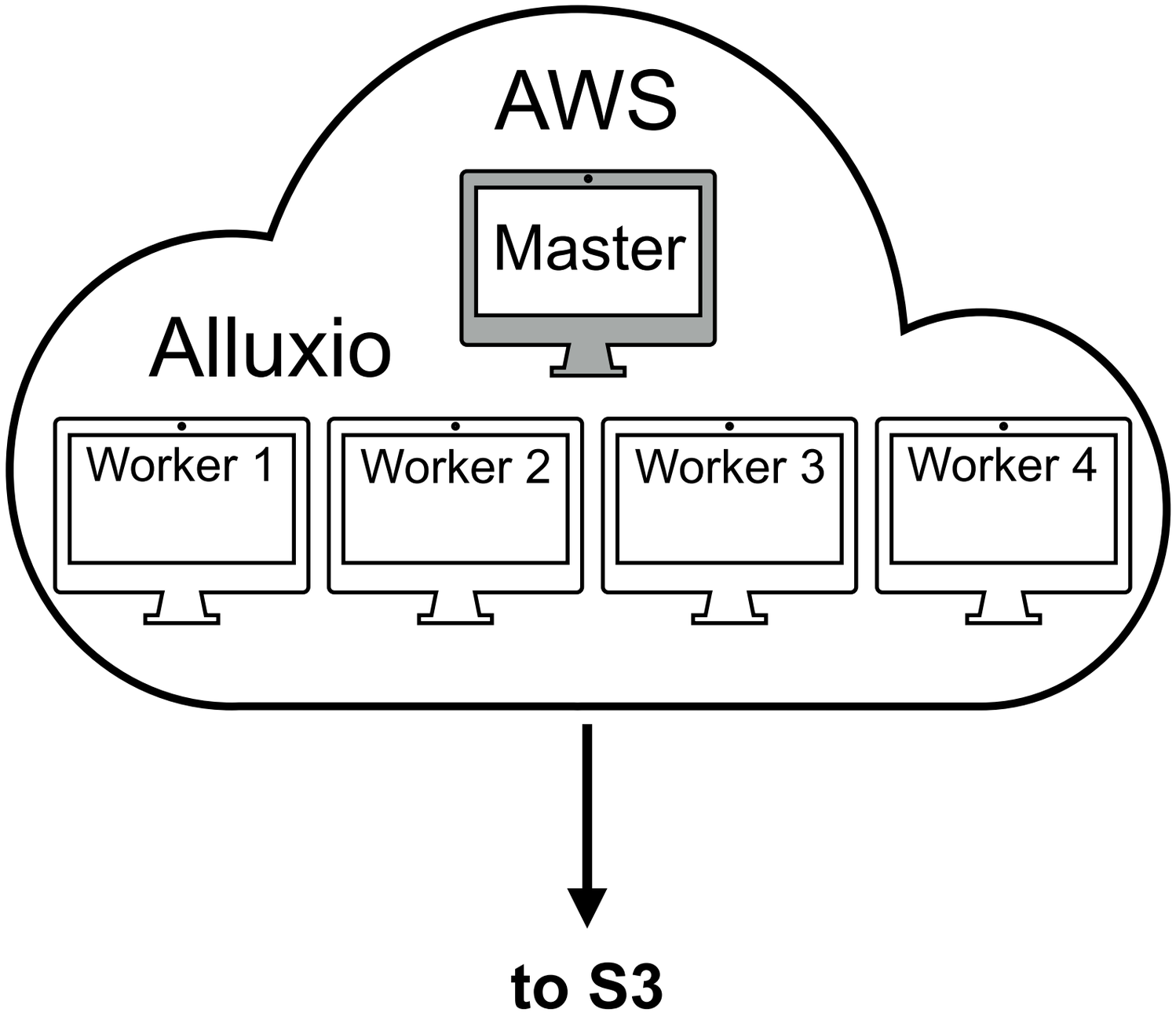
Packing in Alluxio



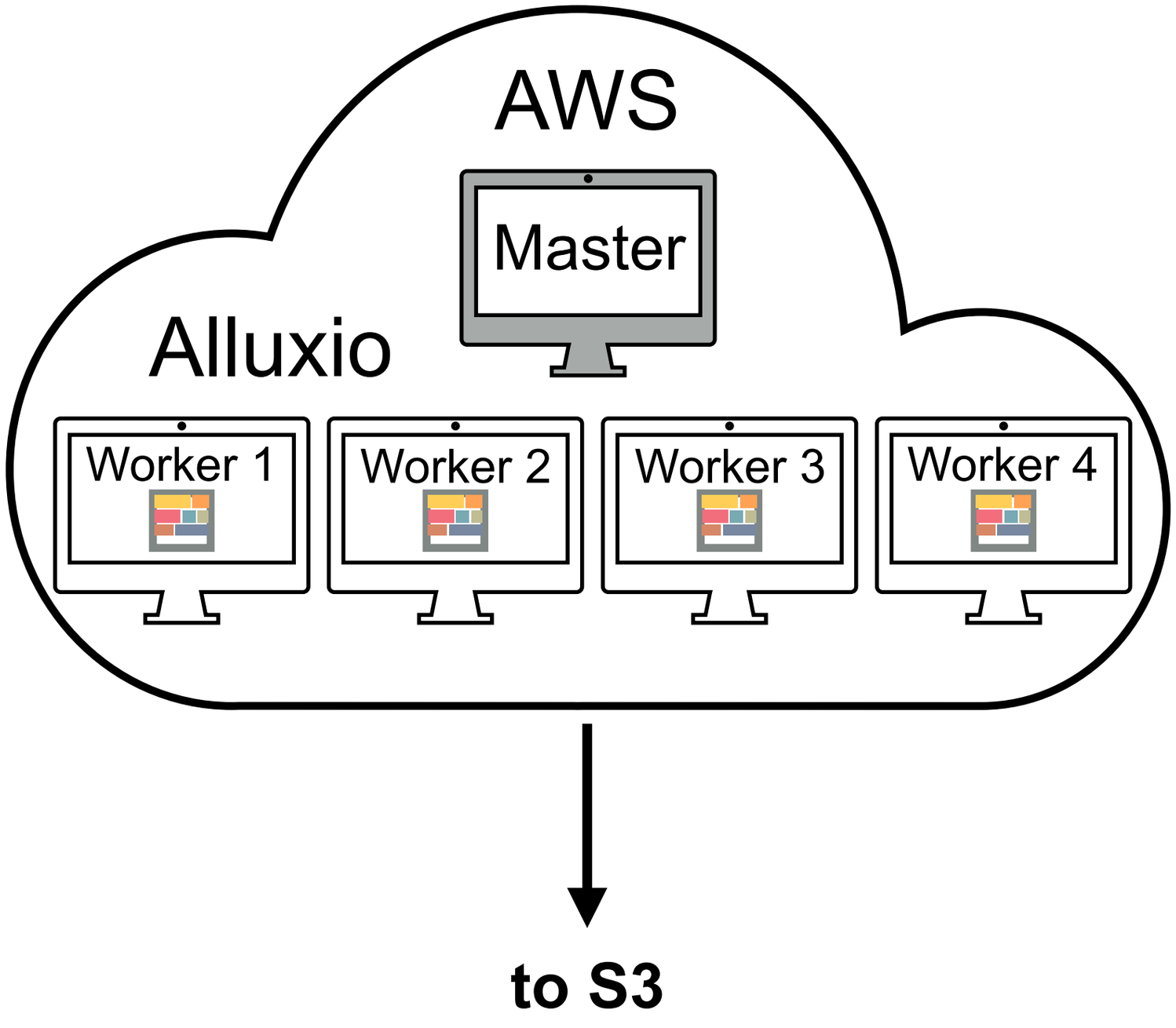
Packing in Alluxio



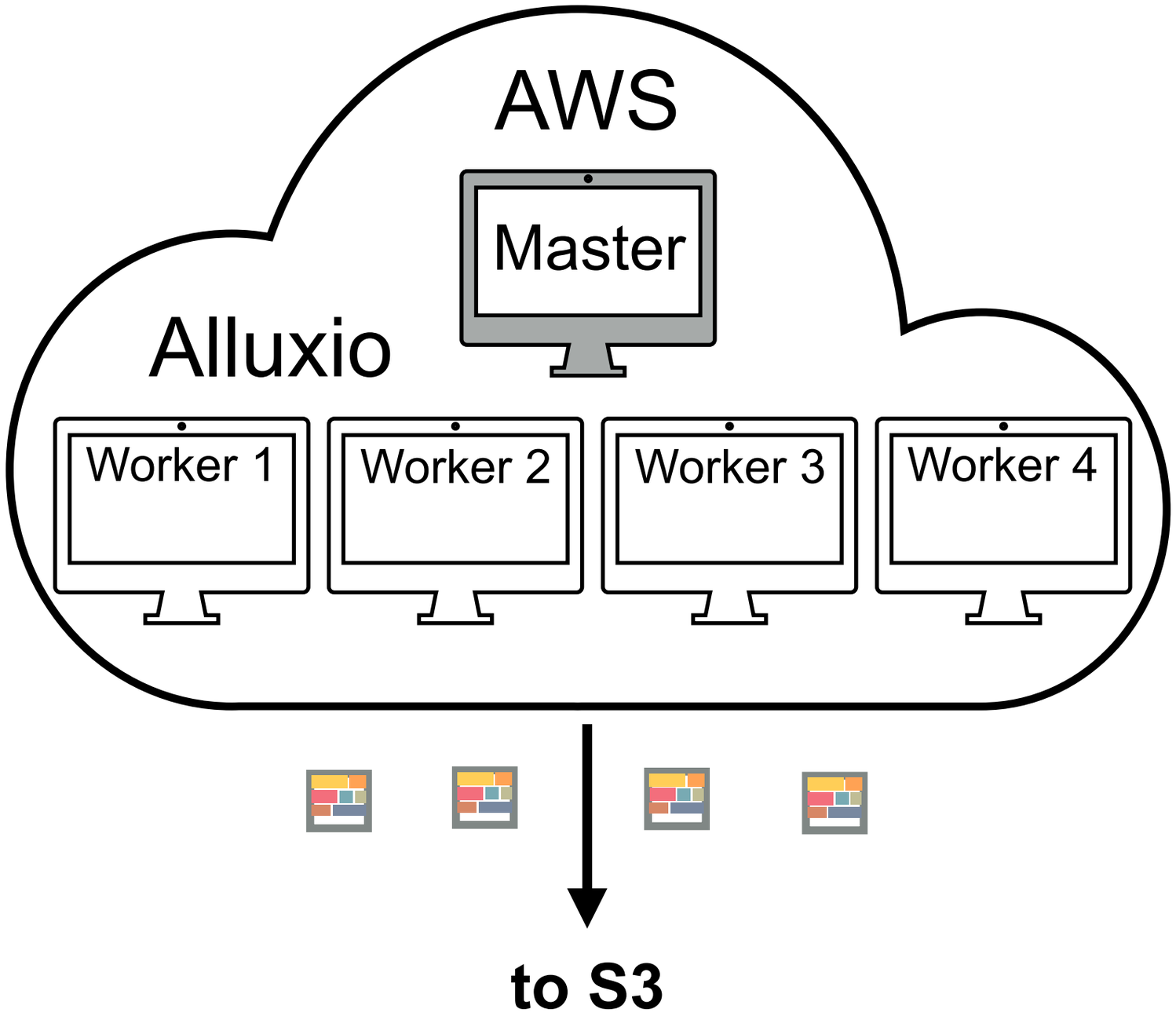
Packing in Alluxio



Packing in Alluxio



Packing in Alluxio



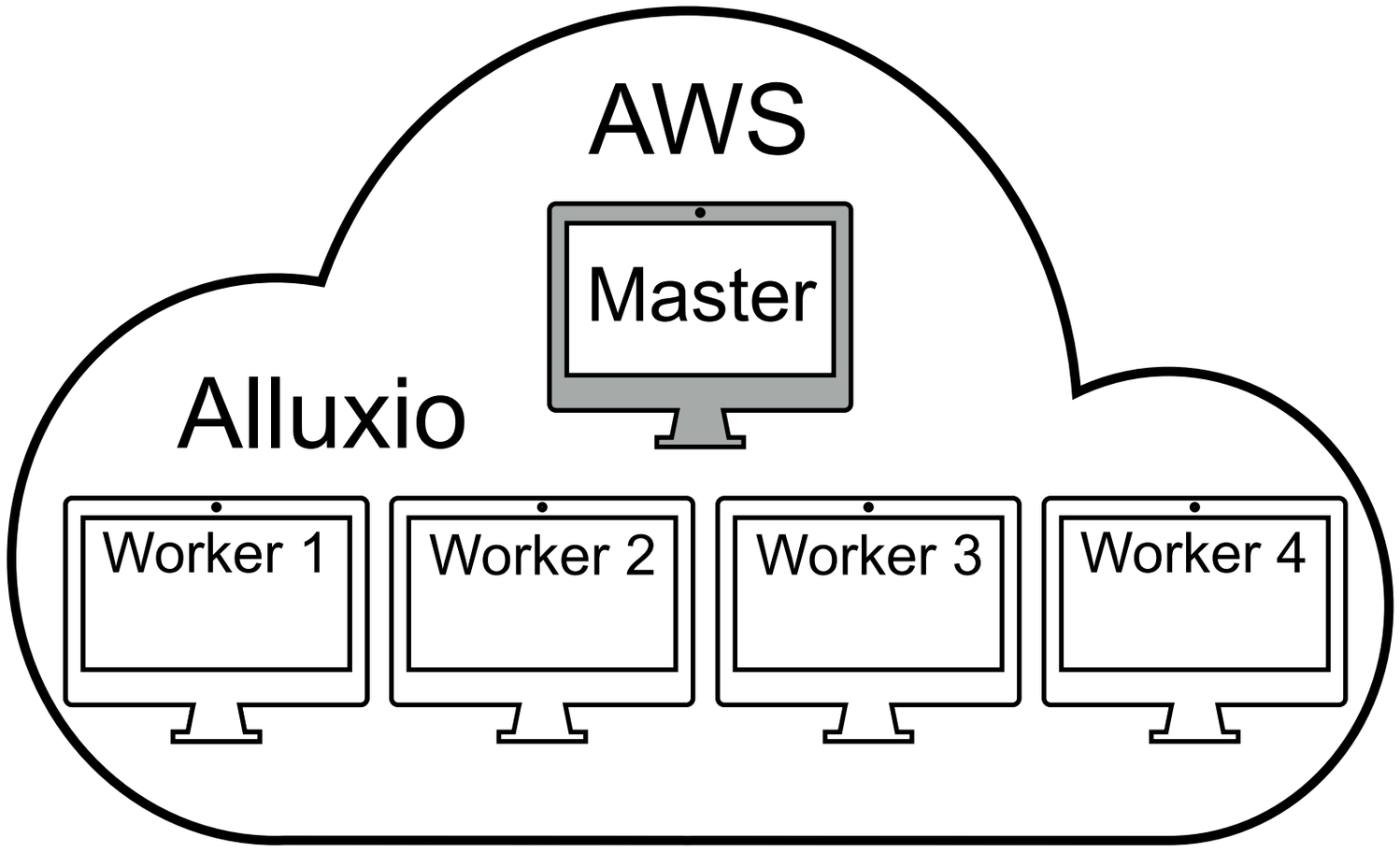
A Packed Blob

Blob Extent: *alluxio-path:logical-offset:physical-offset:length*

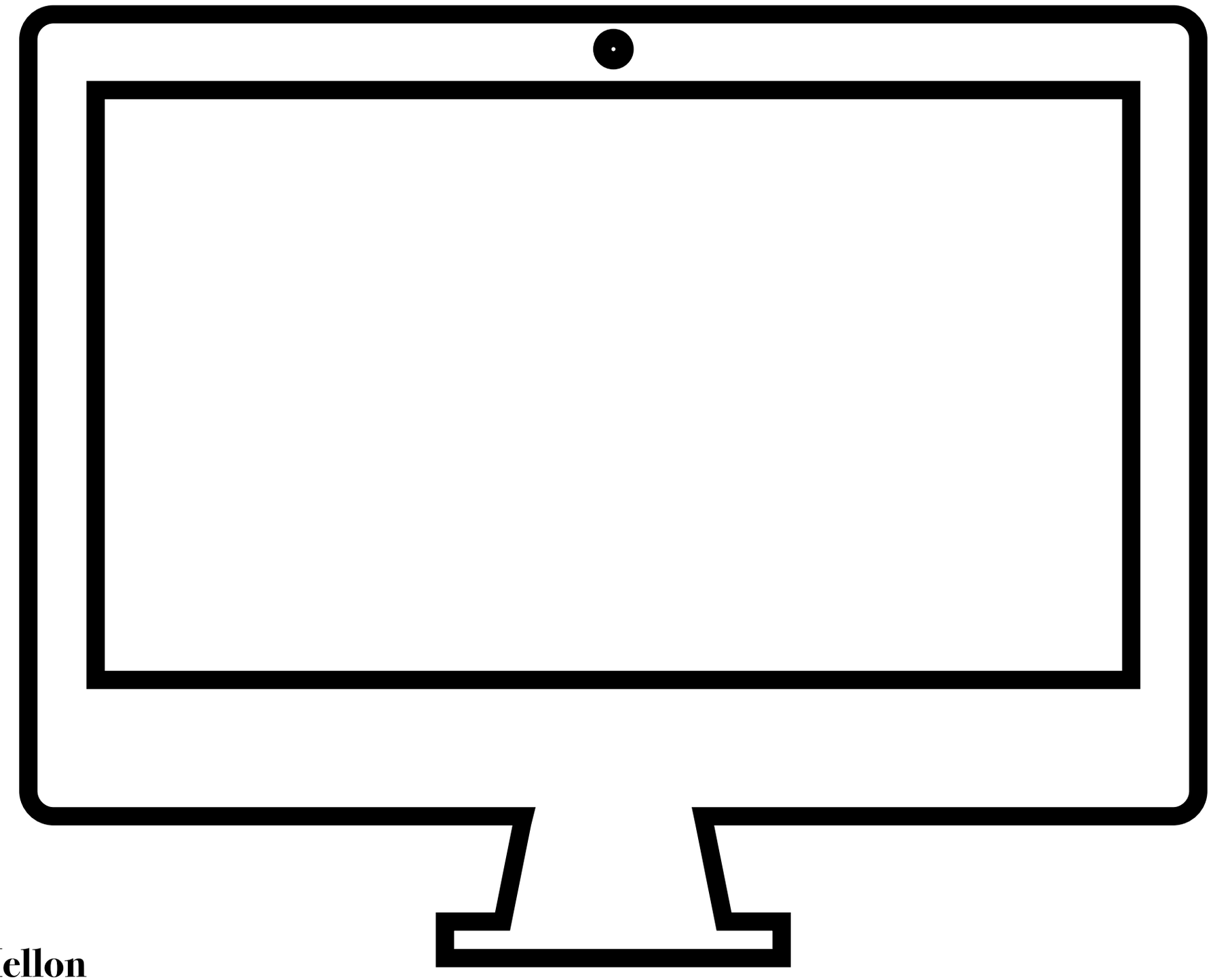
- Packing policy determines what to pack
- Triggered by dirty bytes & timeout

Evaluation

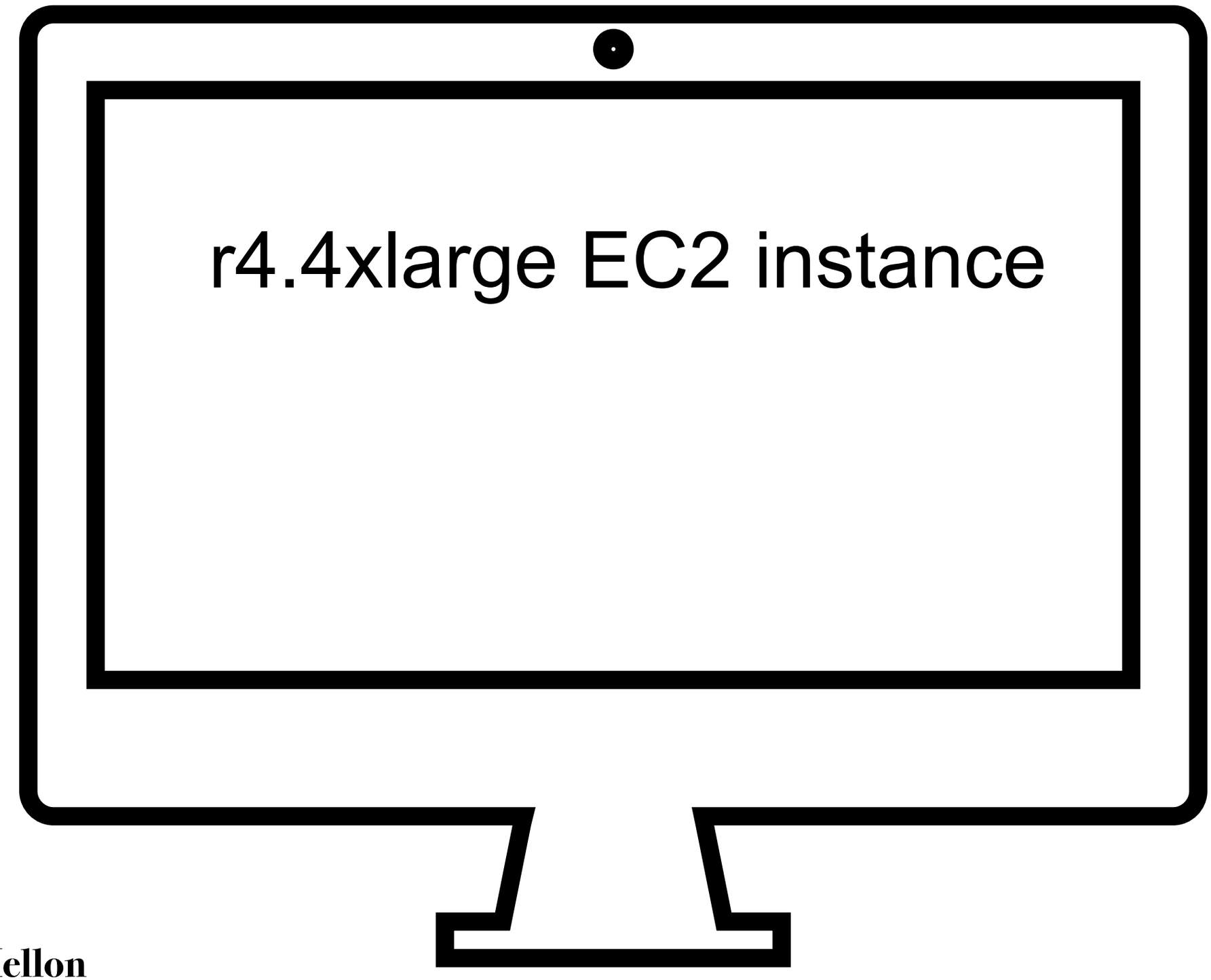
Configuration



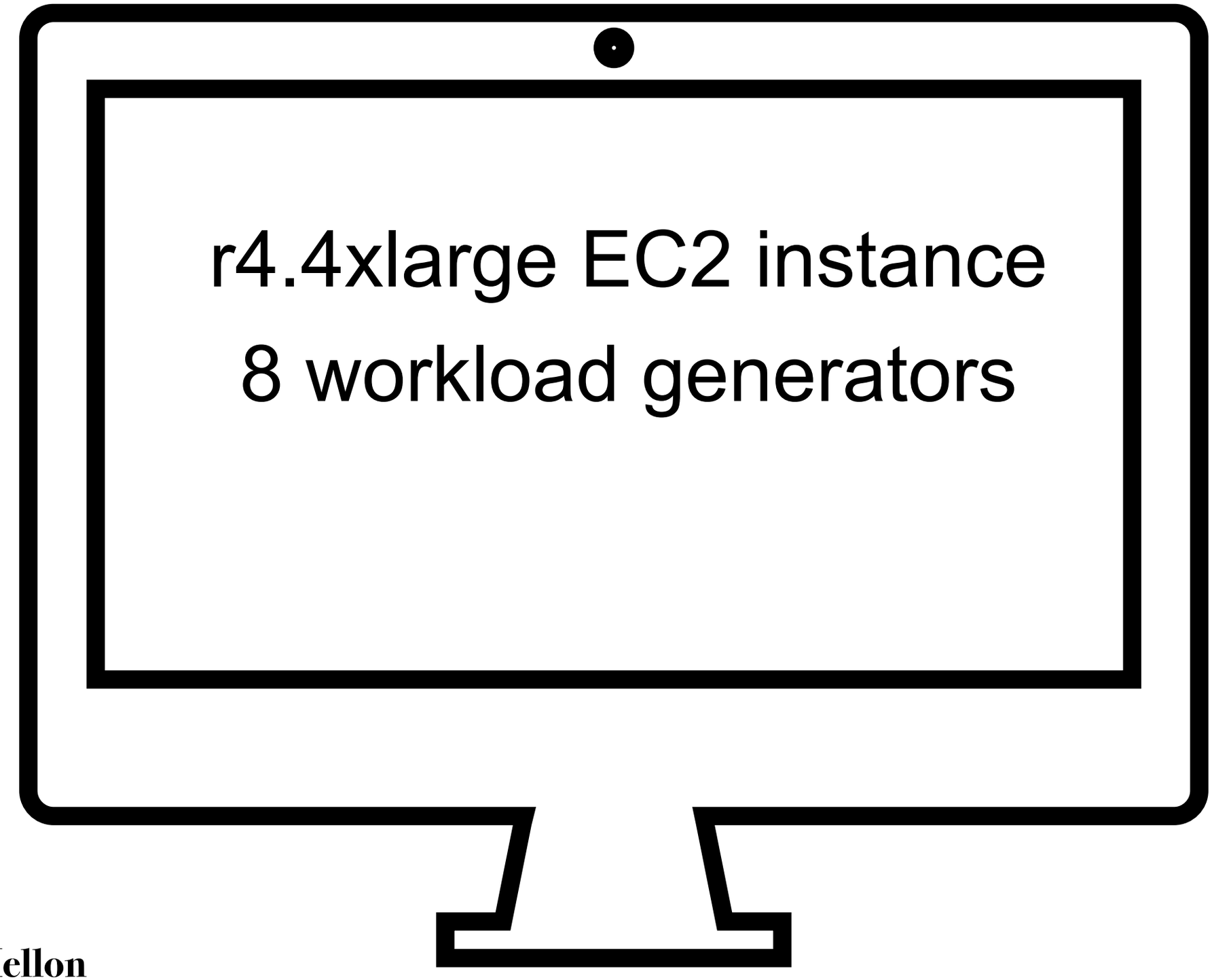
Configuration



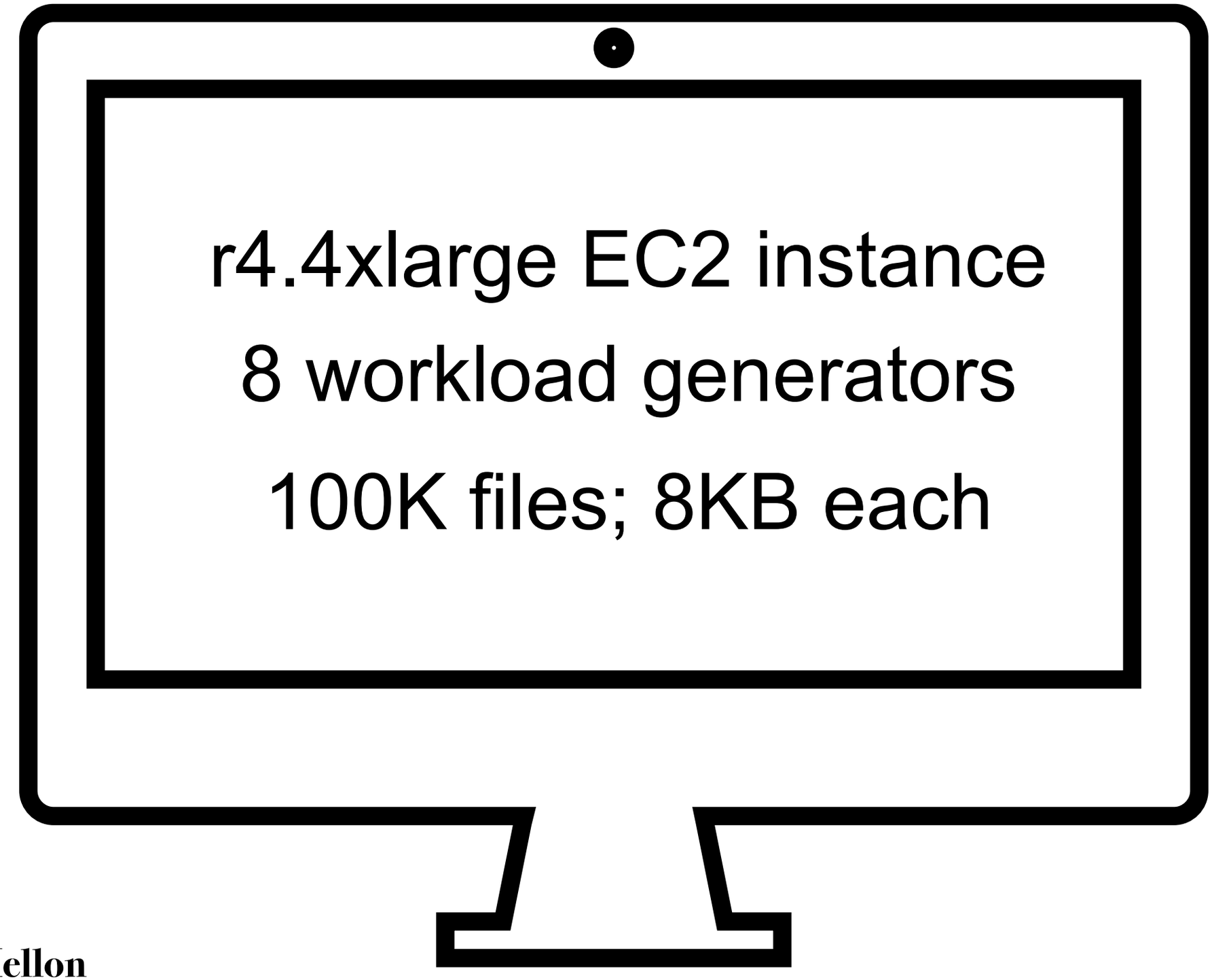
Configuration



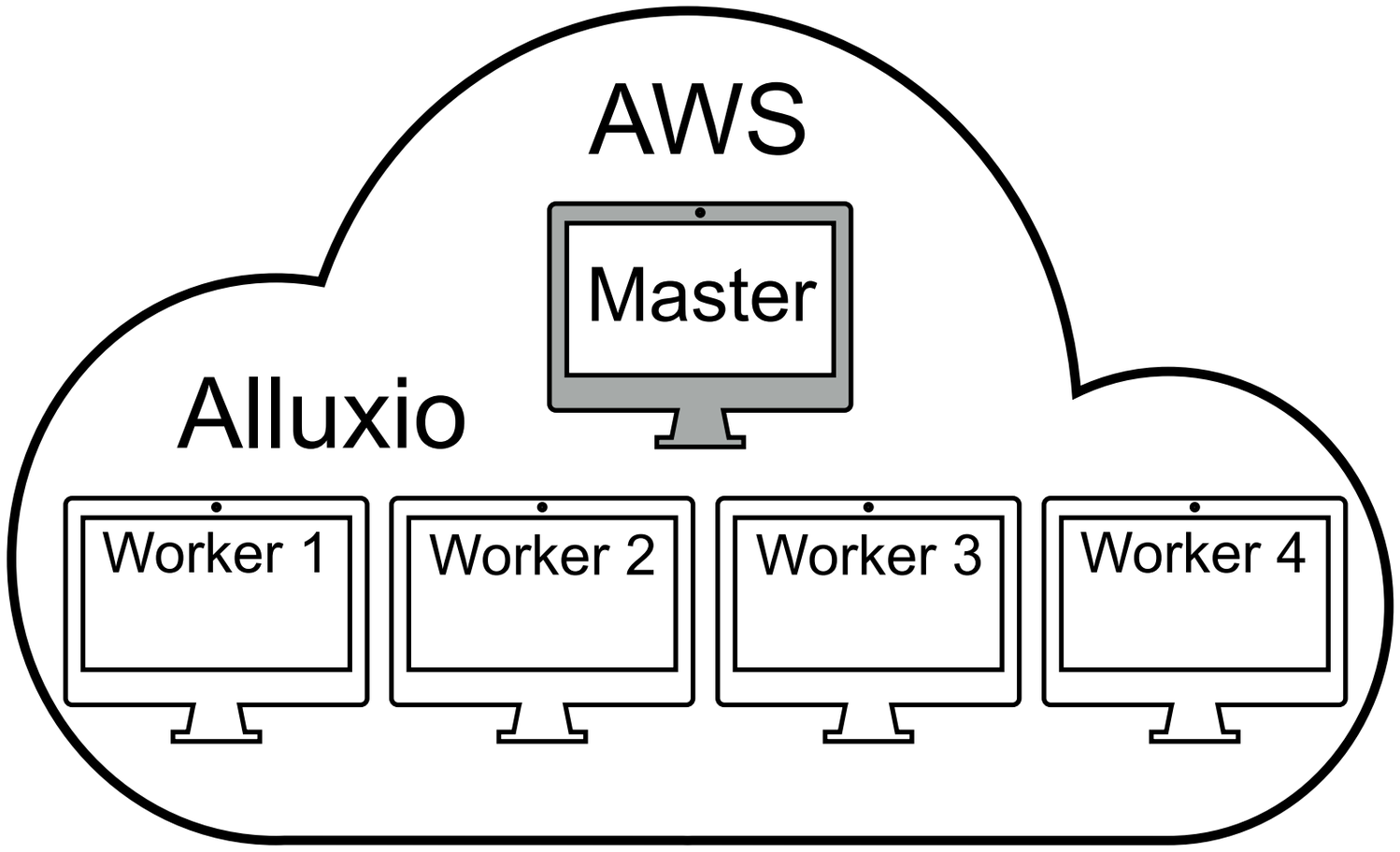
Configuration



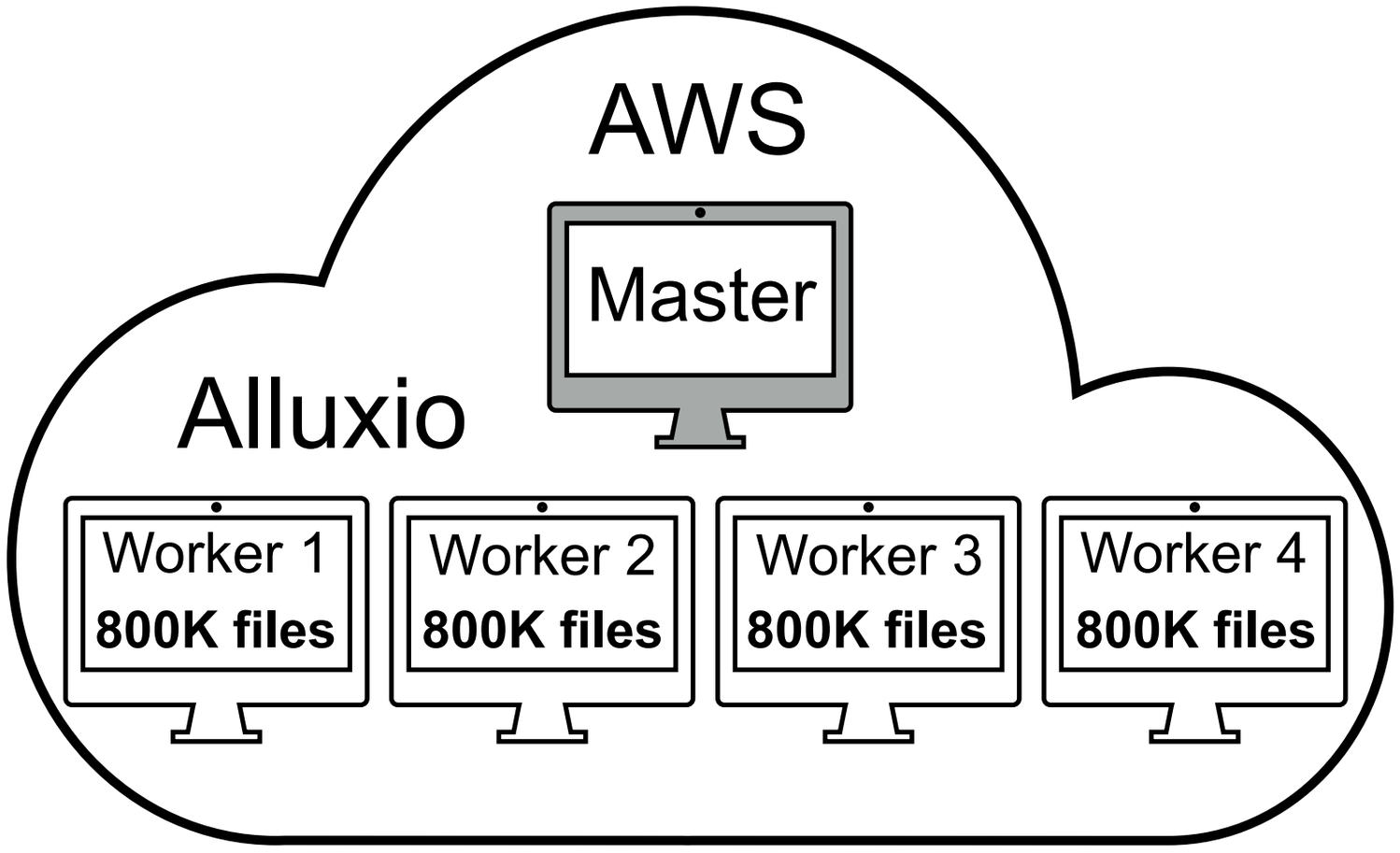
Configuration



Configuration



Configuration

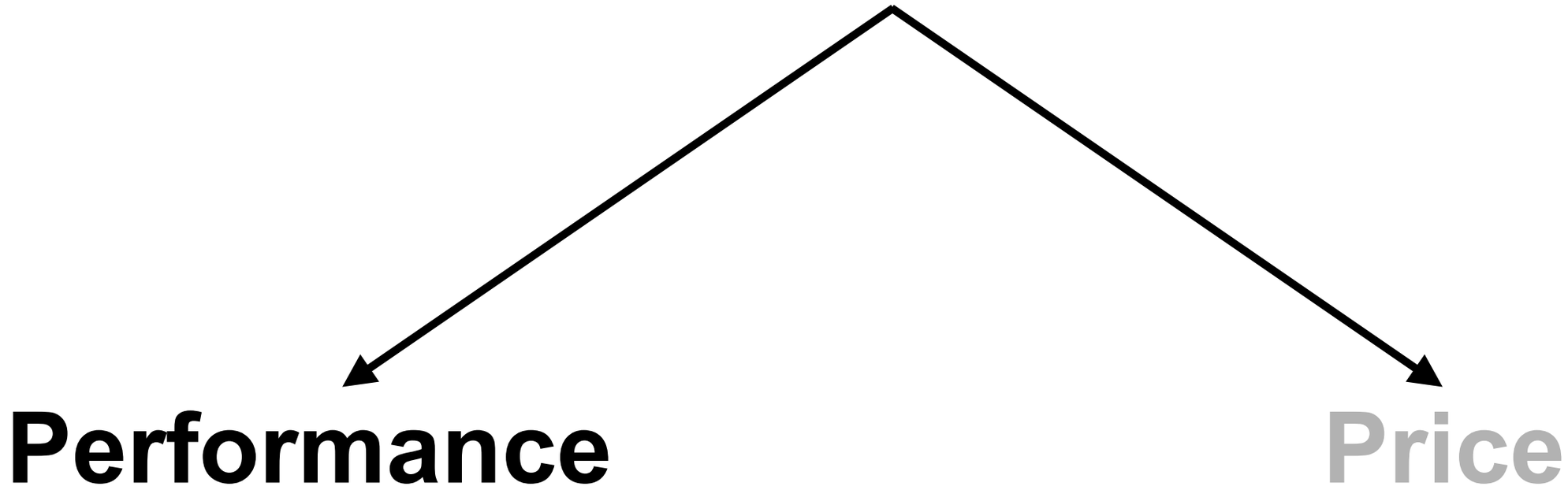


3.2M files of 8KiB = 24.4GB

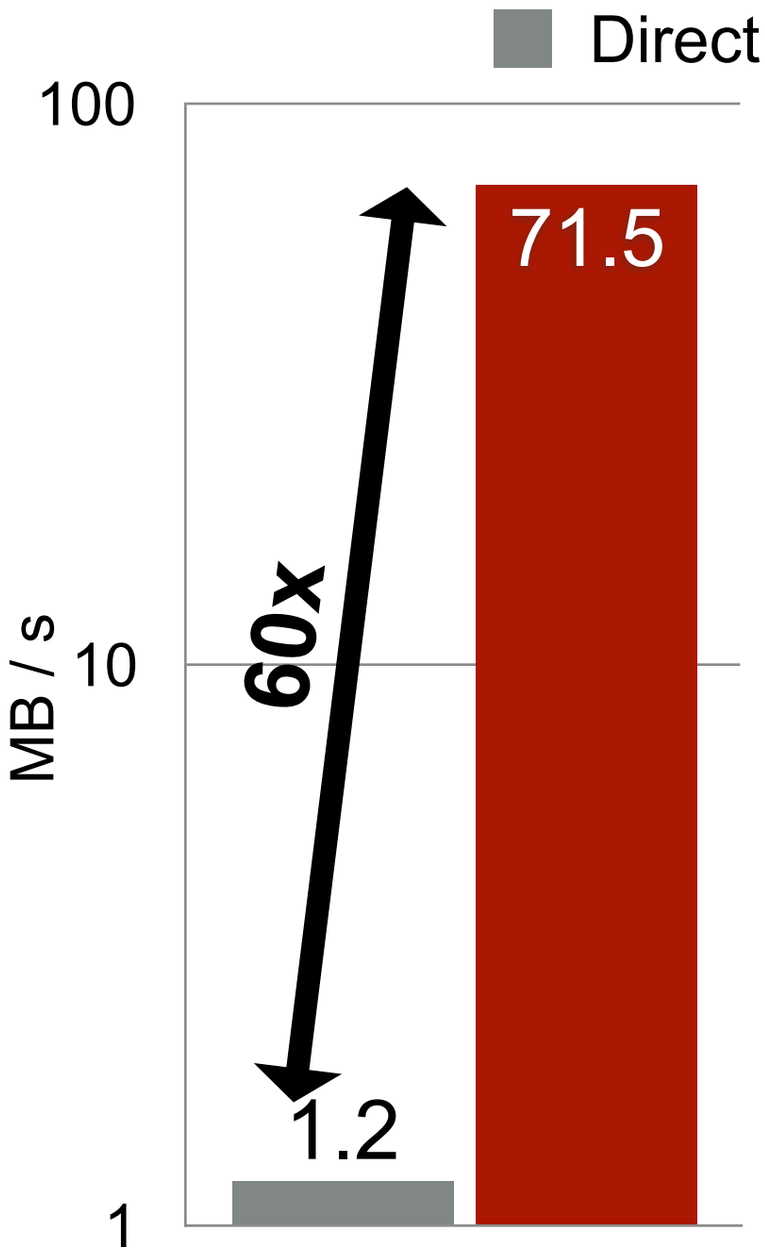
Packing Configuration

- Max blob size: 1 GB
- Packing interval: 5 sec
- # Packing threads: 16
- # Master threads: 16
- Backup interval: 1 min

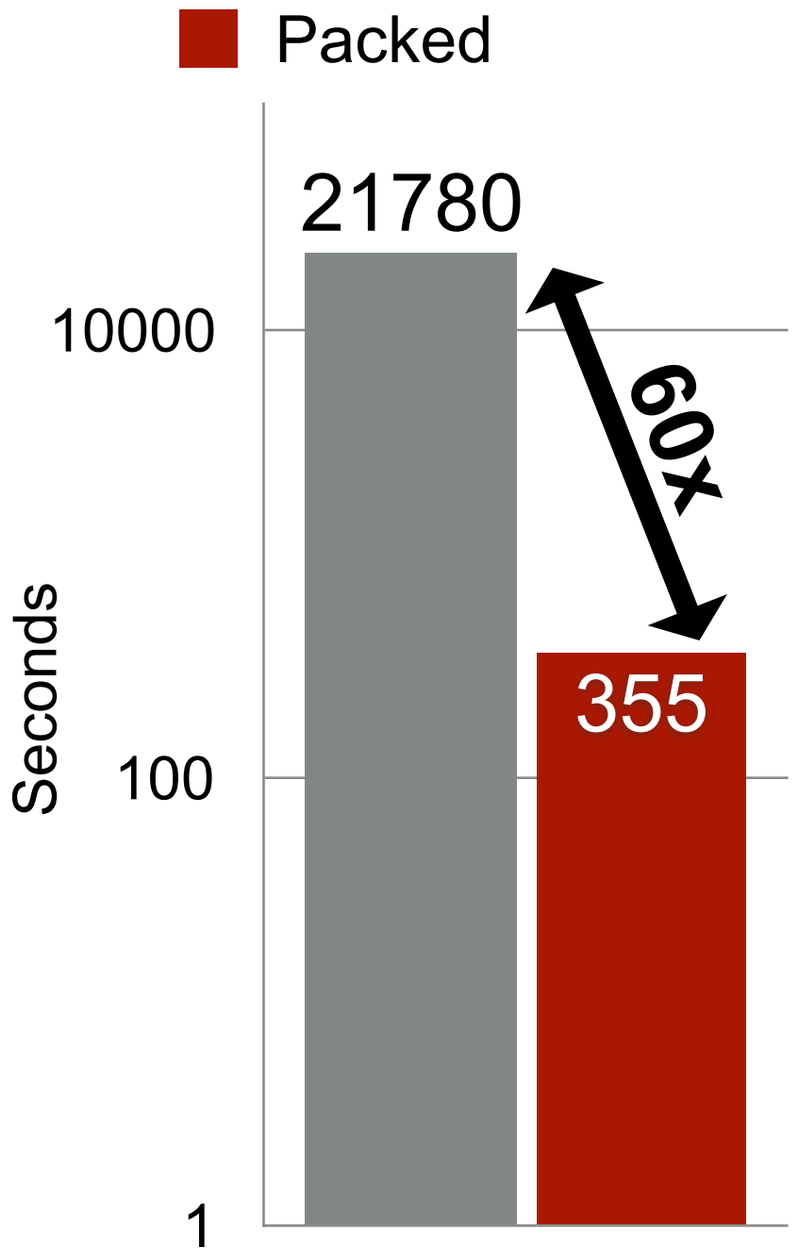
Motivation Revisited



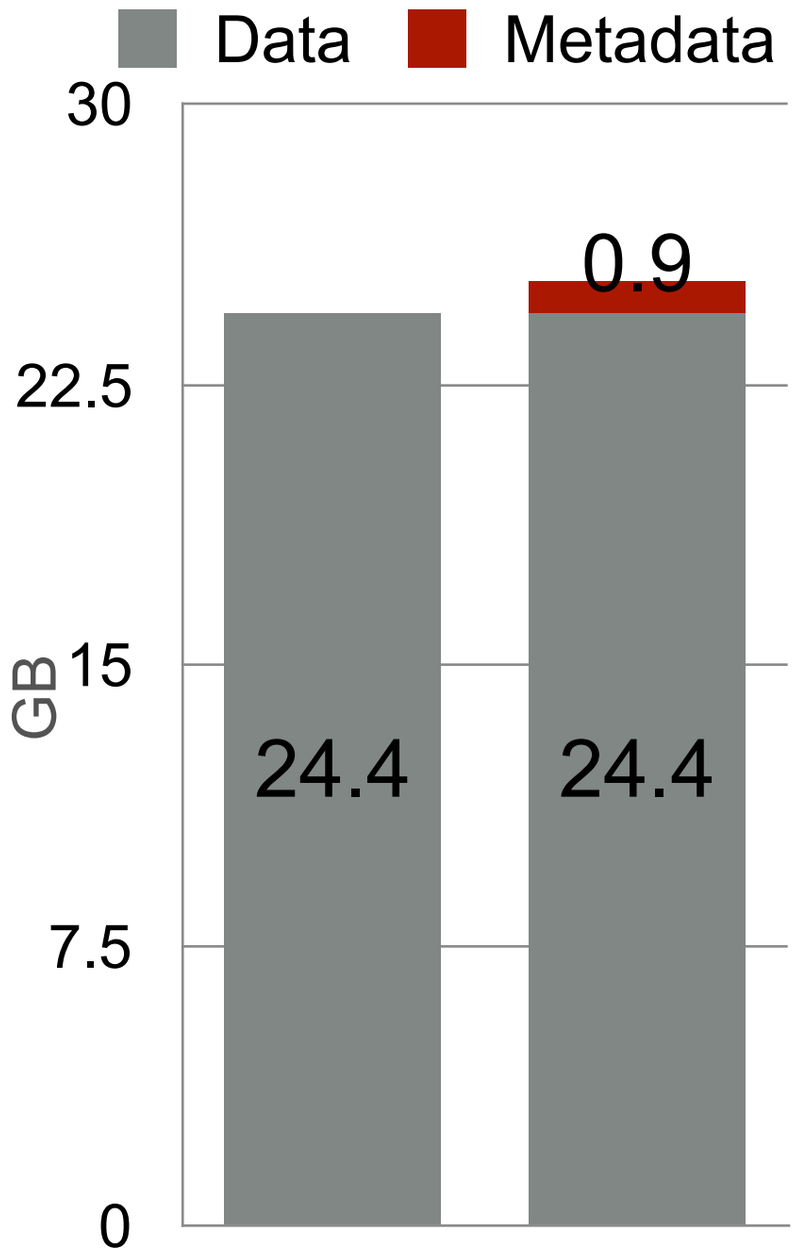
Write Performance Comparison



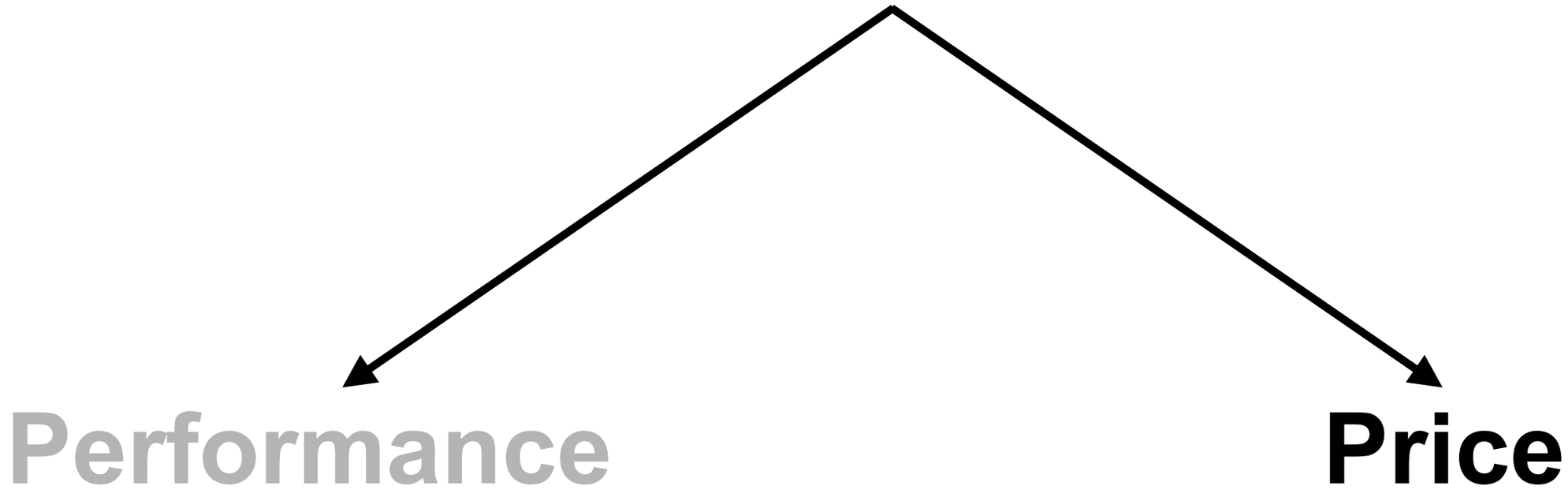
Throughput Comparison



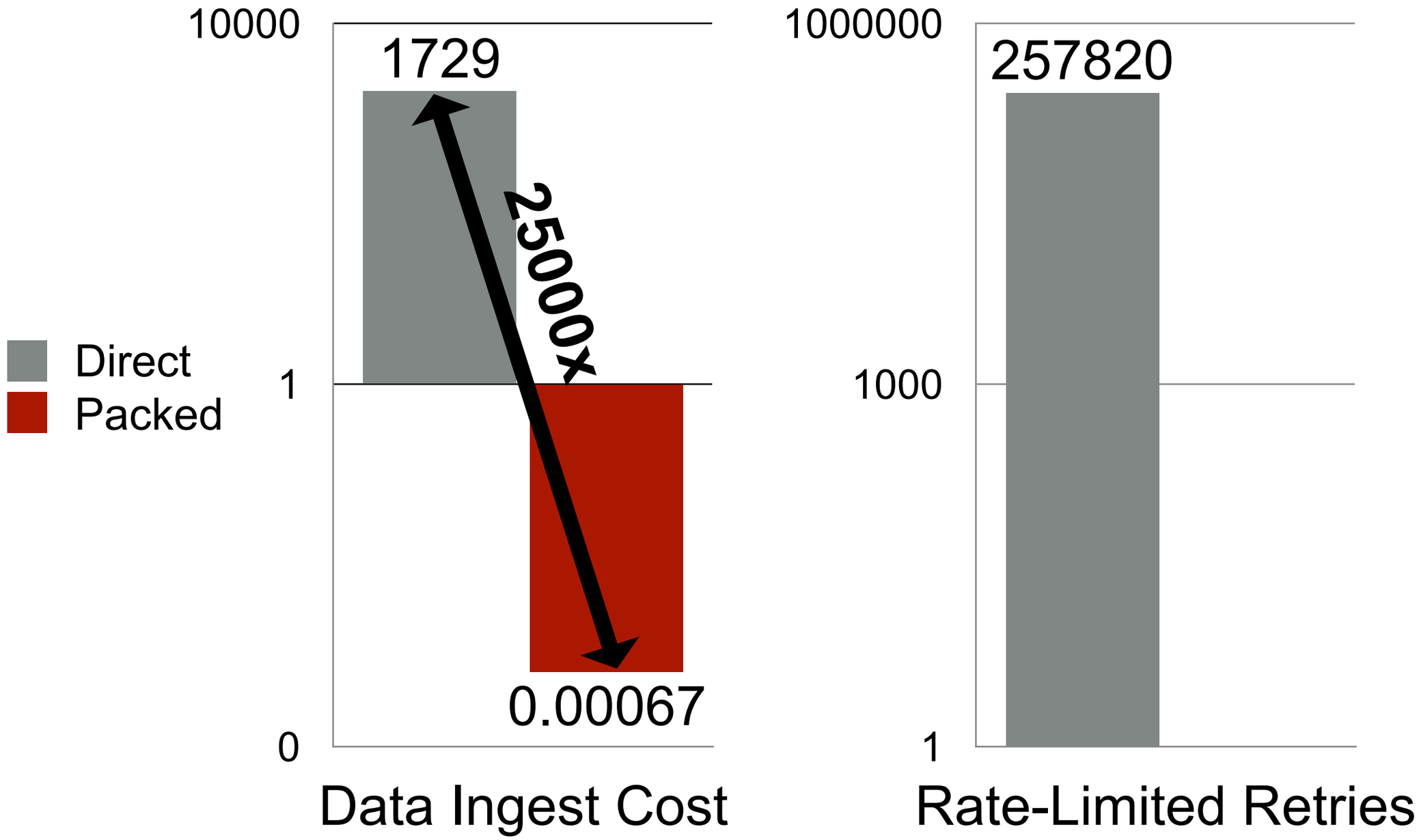
Runtime Comparison



Motivation Revisited



S3 Data Ingest Price



- Request rate is throttled much more than data rate

Research questions (feedback)

- **Price with packing = $\frac{1}{25000}$ price without packing**
  - Will fine-tuning help? By how much?
  - Are there workload specific tuning opportunities / challenges?

- **Garbage collection of packed blobs**
  - Different from LSM Trees, LFS, TableFS, etc.
  - Cost-driven GC policies?
  - Interference with foreground workload?

Conclusion

- Price reduction because of:
  - Matching application write sizes to storage system write sizes
  - Elimination of retries due to aggressive rate-limiting imposed by S3-like services

saukad@cs.cmu.edu

Carnegie Mellon
Parallel Data Laboratory