S3 Performance Optimization

S3 is designed to support very high request rates. However, if your S3 buckets are routinely receiving >100 PUT / LIST/ DELETE or >300 GET requests per second, then there are some best practice guidelines that will help optimize S3 performance.

The guidance is based on the type of workload you are running:

- GET-Intensive Workloads - use CloudFront content delivery service to get best performance. CloudFront will cache your most frequently accessed objects and will reduce latency for your GET requests

S3 Performance Optimization

- Mixed Request Type Workloads - a mix of GET, PUT, DELETE, GET Bucket) - the key names you use for your objects can impact performance for intensive workloads
- S3 uses the key name to determine which partition an object will be stored in
- The use of sequential key names e.g. names prefixed with a time stamp or alphabetical sequence increases the likelihood of having multiple objects stored on the same partition.
- For heavy workloads this can cause I/O issues and contention
- By using a random prefix to key names, you can force S3 to distribute your keys across multiple partitions, distributing the I/O workload.

S3 Key Name Example

A CLOUD GURU

- The following sequential Key Names are not optimal:
  - mybucket/*2018-03-04-15-00-00*/cust1234234/photo1.jpg
  - mybucket/*2018-03-04-15-00-00*/cust3857422/photo2.jpg
  - mybucket/*2018-03-04-15-00-00*/cust1248473/photo2.jpg

- For optimal performance, introduce some randomness into the key name, e.g. prefix the key name with a 4-character hexadecimal hash:
  - mybucket/7eh4-*2018-03-04-15-00-00*/cust1234234/photo1.jpg
  - mybucket/h35d-*2018-03-04-15-00-00*/cust3857422/photo2.jpg
  - mybucket/o3n6-*2018-03-04-15-00-00*/cust1248473/photo2.jpg

S3 Performance Optimization - Exam Tips

A CLOUD GURU

- Remember the 2 main approaches to Performance Optimization for S3:
  - GET-Intensive Workloads - Use CloudFront
  - Mixed-Workloads - Avoid sequential key names for your S3 objects. Instead, add a random prefix like a hex hash to the key name to prevent multiple objects from being stored on the same partition.