Improving S3 to EC2 transfer speeds

M. Yeung, C.Borys (AuriQ Systems Inc.)

Backstory

- Essentia is our home-grown ETL/Analytics engine deployed on AWS, and it streams files directly from s3 for processing on a cluster of EC2 instances
- We process log data for online marketing clients primarily
- Each client has one s3 bucket to store their data.

Problem

- Data is very similar in content/format from client to client, yet processing for some takes much longer on average.
- Issue seems to be with the speed at which the data is being streamed from s3 to an ec2 instance
- Observation: download speed correlated with bucket names and method/timing of upload

Testing the hypothesis

- Create 2 buckets with names that conform to AWS guidelines and 2 that do not. I'll call these the 'good' and 'bad' buckets. Push the test file to one of the good buckets and one to a bad bucket.

- Wait 3 hours and then push the file again to the other good bucket and the other bad bucket.

- In all cases, the file should be pushed from outside of AWS (i.e your desktop)

Results

- Of the 4 bucket tests, ONLY the well-formed bucket where we waited 3 hours to upload supported fast transfers from s3 to our ec2 instances

```
s3://ultest-delay/testdata.bin -> tmp.bin  [1 of 1]
 10485760 of 10485760   100% in    0s    16.14 MB/s  done
s3://ultest-nodelay/testdata.bin -> tmp.bin  [1 of 1]
 10485760 of 10485760   100% in    4s     2.45 MB/s  done
s3://ULTEST_DELAY/testdata.bin -> tmp.bin  [1 of 1]
 10485760 of 10485760   100% in    5s  1925.31 kB/s  done
s3://ULTEST_NODELAY/testdata.bin -> tmp.bin  [1 of 1]
 10485760 of 10485760   100% in    5s  2005.19 kB/s  done
```

~10x transfer speed !

- If the bucket creation & upload is performed from an EC2 instance (i.e. inside the AWS infrastructure), this isn't an issue. Seems any bucket name and even no delay to fill a bucket is fine

Final Notes

- (If you need to push files to s3 from outside of AWS)
- create an s3 bucket using following recommended guidelines for names: (a-z, 0-9, - ONLY. 3<= length <=63)
- Wait about 3 hours before uploading anything to the bucket
- Caveat 1: This test was done on us-east-1. Other zones may have different (or no) issues.
- Caveat 2: all bucket creation and data transfer was done with s3cmd.

Questions? Feel free to contact me at
colin.borys@gmail.com

1. Run from outside AWS (i.e. desktop)

```
#!/bin/bash

# create 10Mb test file
tfile=testdata.bin
dd if=/dev/zero of=${tfile} bs=10485760 count=1

# create buckets
s3cmd mb s3://ULTEST_DELAY
s3cmd mb s3://ULTEST_NODELAY
s3cmd mb s3://ultest-delay
s3cmd mb s3://ultest-nodelay

# upload the test file to our 'nodelay' buckets
s3cmd put ${tfile} s3://ULTEST_NODELAY
s3cmd put ${tfile} s3://ultest-nodelay

# wait for 3 hours and then upload the test data to the 'delay'
buckets
sleep 3h
s3cmd put ${tfile} s3://ULTEST_DELAY
s3cmd put ${tfile} s3://ultest-delay
rm testdata.bin
```

2. Run from an EC2 instance

```
#!/bin/bash
tfile=testdata.bin

# fetch the files back and look at the download times.
# for fairness you should do this multiple times and average,
# but you should see the same result
s3cmd get s3://ULTEST_NODELAY/${tfile} tmp.1
s3cmd get s3://ultest-nodelay/${tfile} tmp.2
s3cmd get s3://ULTEST_DELAY/${tfile} tmp.3
s3cmd get s3://ultest-delay/${tfile} tmp.4
```

3. Cleanup buckets and files

```
#### CLEANUP
# delete files
s3cmd del s3://ULTEST_DELAY/${tfile}
s3cmd del s3://ULTEST_NODELAY/${tfile}
s3cmd del s3://ultest-delay/${tfile}
s3cmd del s3://ultest-nodelay/${tfile}

# delete buckets
s3cmd rb s3://ULTEST_DELAY
s3cmd rb s3://ULTEST_NODELAY
s3cmd rb s3://ultest-delay
s3cmd rb s3://ultest-nodelay

# delete tmp files
rm -f tmp.*
```