EC2 instance r6id.8xlarge **Input size** 44GB

Fusion v2.2.6 scratch=false workdir=s3
AWS v1.0.2 scratch=false workdir=local

		AWS Mountpoint			Fusion					
process	**step**	**read**	**write**	**seconds**	**read**	**write**	**seconds**	**speed-up**	**comments**	**command**
CHECK_AND_UNZIP	read1	S3	-	36.551	S3	-	46.026	-25.92%	1,2	dd if=file.fastq.gz of=/dev/null iflag=direct bs=1M
	read2	S3	-	34.426	S3	-	18.919	45.04%	1,3	dd if=file.fastq.gz of=/dev/null iflag=direct bs=1M
	check	S3	-	529.096	S3	-	79.504	84.97%	4	echo ${checksum} file.fastq.gz | md5sum --check --status
	uncompress	S3	local	993.755	S3	S3	921.271	7.29%	5	gzip -d -c file.fastq.gz > file.fastq
	count	local	-	90.792	S3	-	98.079	-8.03%	6	grep -c '^+\$' file.fastq
ZIP_AND_COMPARE	compress	local	local	2217.792	S3	S3	2037.19	8.14%	7,8,9	gzip --fast -c file.fastq > file.fastq.gz
	compare	local/s3	-	1054.855	S3	-	910.741	13.66%	10	zcmp file.fastq.gz original.fastq.gz
	runtime			**4957.267**			**4111.73**	**17.06%**	11,12,13,14	

Comments

1. Due to an important problem that has AWS with Kernel page cache, here we used a special crafted sequential read with page cache disable to get a more fair comparison
2. On a pure sequencial that does nothing with what is reading we see that AWS is around 20% - 25% faster. This make sense, because Fusion do not uses memory and has the extra work to write the cache to disk.
3. We see a big speed-up on Fusion when a second pass is done. Compensating the extra effort of doing a disk cache on the first run.
4. Here we start to see that if the process that is reading the file is doing some CPU task, then the slighlty slower throwput of Fusion is not a problem
5. Here AWS is writing to local disk while Fusion is indirectly writing to S3. Even dought, we get very similar speed, because Fusion is uploading the output on the background.
6. We get similar speed, but here AWS is reading from local disk and Fusion from the previously generated file that is still on cache but also on S3.
7. We start a new process and Fusion do not has anything on cache
8. If the process that is reading is CPU intensive we see that reading from S3 and writting to S3 with Fusion can be even faster than than doing the same from local disk
9. Fusion can be faster than the local disk because on the background is doing a more agressive prefetching than the Kernel itself, so if the process is doing some CPU when it requests more data it is already on the page cache and is served faster than if read directly from disk.
10. Here we see another example than if the process that is reading the data is doing some CPU, Fusion can be even faster than local disk.
11. In this setup (that we run all the pipeline in a singel node) AWS is using local NVMe disk as working directory. On a normal setup we will need to use a bit slower parallel filesystem like Lustre.
12. In this AWS setup you still need to upload to S3 the final results
13. Overall we see a ~17% speed up running on Fusion this pipeline. A more CPU intensive pipeline could even get better results.
14. We see similar results running on top of Fusion versus running directly on local NVMe disk