1) Unfortunately, credits ran out last moment before I could complete running distributed training. The Paleo Results are as follows (I have attached the outputs generated as files):

#Workers	time_fwd	time_bwd	time_apply	time_comp	TreeAllReduce
1	343.644	720.385	0.9320	1064.962	-
2	166.893	380.851	0.9320	548.677	210.972
4	83.499	179.7774	0.9320	264.209	632.915
8	41.731	94.968	0.9320	37.631	1476.802

2) The assumption in general that a GPU always works at its maximum throughput, with its I/O saturated is not true. It requires highly personalised code for that particular GPU, highly specialised libraries to implement the same. Additionally, there is no guarantee that the computation done is for the model alone and nothing else- there might be other system processes (example: Spark workers run inherent MapReduce daemons and operations to maintain parallelism) which are used to manage and maintain the states of the GPUs. These processes are not accounted for when we claim the GPU is operating on the job with 100% efficiency.

Thus, given the observed and estimated throughputs, platform performance peak is a parameter that is learnt by Paleo. This learned parameter is then subsequently used to estimate the realistic amount of time taken for a job.

4)

a) K80 Tflops: (assuming double precision operations) - 1.87 Terra FLOPS/sec

   VGG16 Flops: 15503M.
   Therefore, time = 15003 M / 1.87T = 0.008s = 8ms
   Theoretical peak Throughput = 1/0.00801 = 125 images/sec

   GoogleNet Flops : 1606M
   Time = 1606 M / 1.87T = 0.0008s = 0.8ms
   Theoretical peak Throughput = 1/0.0008 = 1250 images/s

   ResNet: 3922M
   Time = 3922M / 1.87T = 0.00209 = 2.09 ms

Theoretical peak Throughput = 478 images/sec

b) Paleo forward pass time (batch size=64, num. workers=1)
VGG16: 531.0148336550836
Resnet: 214.18089576707573
InceptionV3 (substitutes GoogleNet): 343.6448843145265

Other cases- 2-8 workers attached in the file below.

c)