Build Lightning FAST Apps with Docker and OpenPOWER

Indrajit Poddar, STSM, IBM Systems

Seetharami Seelam, RSM, IBM Research
High performance with Big Data

Easy scale-out with:

But….

Deep Learning and Machine Learning training is hard to distribute.
Training can take hours, days or weeks.

Input data and model sizes are becoming larger than ever (e.g. video input, billions of features etc.)

Moore’s law is dying

Resulting in….

Unprecedented demand for offloaded computation, accelerators, and higher memory bandwidth systems.

© 2016 IBM Corporation
OpenPOWER: Open Hardware for High Performance

Traditional
Intel x86

OpenPOWER

Upto:
12 cores per cpu
96 hardware threads per cpu
1 TB RAM
7.6Tb/s combined I/O Bandwidth
GPUs and FPGAs coming…

Systems designed for big data analytics and superior cloud economics

http://www.softlayer.com/bare-metal-search?processorModel[]=8
Docker Containers running on Power have Superior Density

- Wider, Faster Memory Interface, Faster Cores with More Threads
- Split-Core Mode supports Interactive Web style Apps better
- 2x Greater Density of Containers per systems lowers Cost
- >40% better Throughput and 4x better Latency
- OpenPower ecosystem offers wide range of Open HW Platforms

Total: **10,011** Containers on One System: Ubuntu(8028), Node.js(991), Wordpress(992)
A Consistent Developer Experience

Multi-platform Docker images

docker pull ubuntu:latest

will get you the POWER/LinuxOne/X86 specific ubuntu image!!!

Thursday teaser: `docker run estesp/busybox` on any of `{s390x,ppc64le,amd64,aarch64,arm}` and

OpenPOWER: GPU support

IBM Spectrum Conductor includes enhanced support for fine grained GPU and CPU scheduling with Apache Spark and Docker

Mesos support for GPUs coming!!

Credit: Kevin Klaues, Mesosphere
Machine Learning and Deep Learning analytics on OpenPOWER

No code changes needed!!
Try out Machine Learning and the Developer Challenge
Enter to win exciting prizes!!  openpower.devpost.com

Huge speed-ups with GPUs and OpenPOWER!

```
In [7]:
# CPU mode
net.forward()  # call once for allocation
timeit net.forward()
1 loop, best of 10: 7.22 s per loop

That's a while, even for a batch size of 30 images. Let's switch to GPU mode.

In [8]:
# GPU mode
caffe.set_device(0)
caffe.set_mode_gpu()
net.forward()  # call once for allocation
timeit net.forward()
10 loops, best of 10: 56.4 ms per loop
```
Learn More and Get Started...

Register for a SuperVessel Account and take deep learning notebooks running in docker containers a spin!

https://ny1.ptopenlab.com/bigdata_cluster
Which Celebrity Do You Look Like?

Press "Take Photo" to select a photo:

Live Demo at the Booth!!
Thank you!

@ipoddar
@sseelam