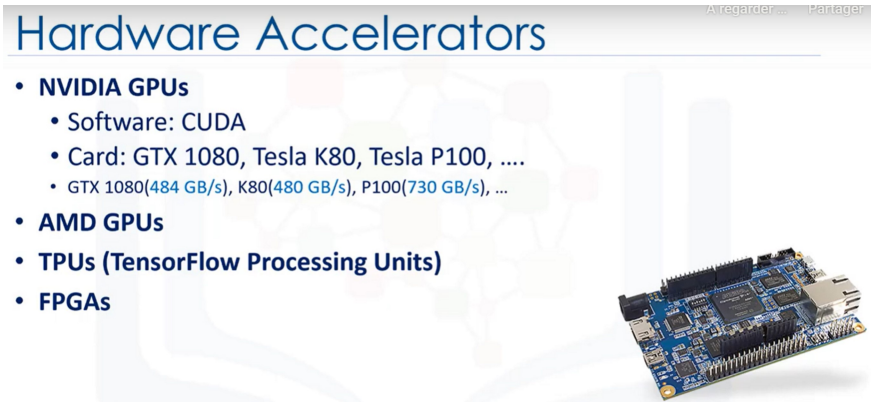
Mod 3: Deep Learning in the Cloud

Friday, May 1, 2020 8:44 AM

HARDWARE ACCELERATORS



- NVIDIA GPUs (CUDA)
  CUDA es un lenguaje de alto nivel para programar a los GPU.
  El desempeno y velocidad de las GPU depende del ancho de banda de la memoria. Esta característica (que sirve para recuperar data de varias dimensiones) es lo que las hace adecuada para DL.

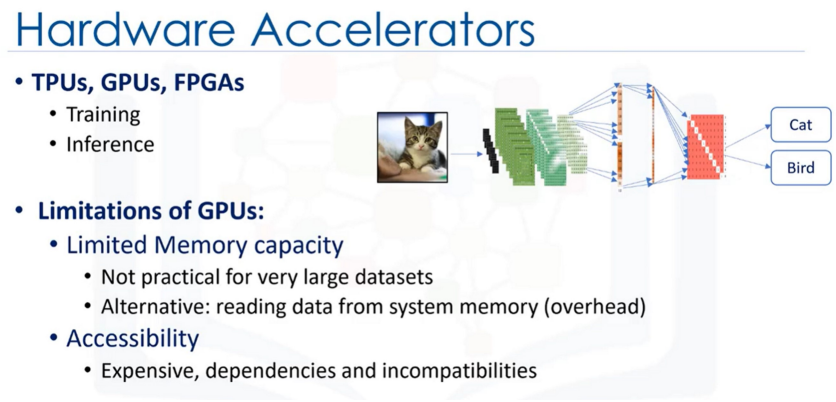
  Ejemplos:
  - GTX1080 (484 GB/s)
  - K80 (480 GB/s)
  - P100 (730 GB/s)

- AMD GPUs (OpenCL)
  OpenCL no es popular.

- Google TPUs (TensorFlow)
  Procesadores disenados especialmente para usar TensorFlow.

- FPGAs (VHDL)
  Field Programmable Gate Arrays - Matriz de puertas lógicas programables en campo.
  Es un dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada en el momento, mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

  Un procesador tiene un número fijo de recursos. Pero con un FPGA es posible construir un circuito personalizado y explotar el paralelismo con un lenguaje de programación.
  Intel trabaja en hacerlos más rápidos.



TPUs, GPUs y FPGAs pueden usarse para el entrenamiento y la inferencia.

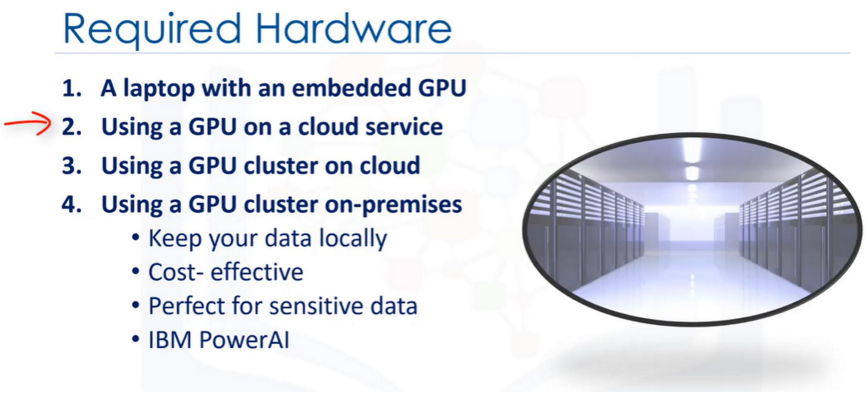
Limitaciones de GPUs

- Memoria limitada: Actualmente tienen hasta 16GB de memoria. Esto no es lo óptimo para bases de datos grandes, ya que es necesario guardar la data en la memoria del GPU para utilizarla.
  Lo que se hace actualmente es usar la memoria del sistema para leer los datos. Y esto ocasiona mucho (overhead) sobrecoste.
  Entonces es necesario una plataforma que pueda manejar un acceso rapido a la memoria del sistema y un rapido intercambio de datos entre los GPUs.

  *Sobrecoste: es el exceso de tiempo de computación, memoria, ancho de bando u otros recursos, que son necesarios para realizar una tarea específica.

- No es simple comprarlos y conectarlos en la PC personal: son muy caros y tienen dependencias e incompatibilidades (como es usual con el HW)
  Además se necesita más de 1 GPU para trabajar con bases de datos grandes.

HOW DOES ONE USE A GPU?



El aprendizeje profundo necesita de mucha potencia de cálculo, pero ésta depende de la tarea que se quiere hacer.

Opciones para trabajar:

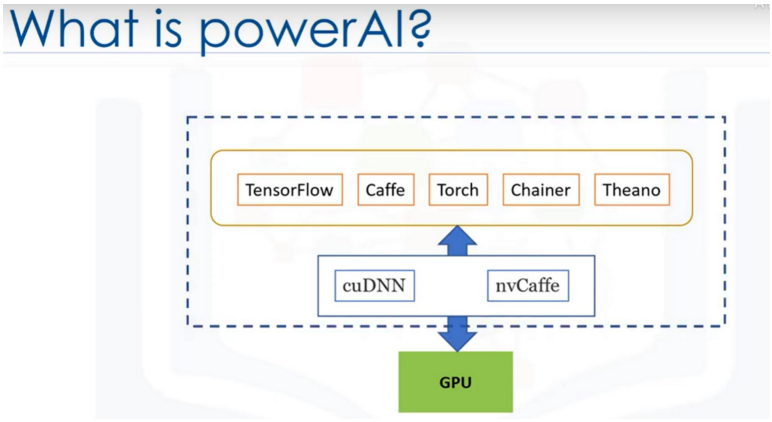
- Laptop con un GPU integrado
  Es útil para aprender CUDA y entrenar algunos modelos. Aunque no es suficiente para problemas particulares. Entonces, lo que se puede hacer es reducir el tamano del dataset o del modelo de acuerdo a la laptop. Pero usualmente los resultados no son buenos.

- Usar GPUs de un servicio cloud (uno o varios)
  Provedores como IBM Cloud, Amazon AWS o Google cloud ofrecen máquinas virtuales con GPUs. Hay muchas opciones de HW (tipo de procesador, memoria y GPUs) y precios a elegir.
  La desventaja es que hay que subir la base de datos y hacer el entrenamiento en la nube; lo que puede implicar problemas de privacidad además de un elevado costo según el número de horas que tome el trabajo.

  **Usar 1 GPU para hacer experimentos con muestras de la base de datos. Y según eso ampliar a una instancia de 8 a 16 GPUs o sino usar un cluster de GPUs para distribuir la carga de cálculo.

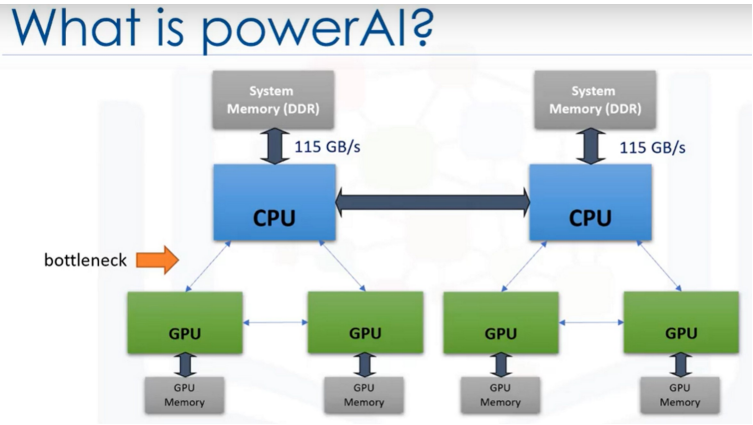
- Usar GPUs en las instalaciones de la empresa
  La base de datos se mantiene privada y el costo económico del cálculo es bajo.

DEEP LEARNING IN THE CLOUD



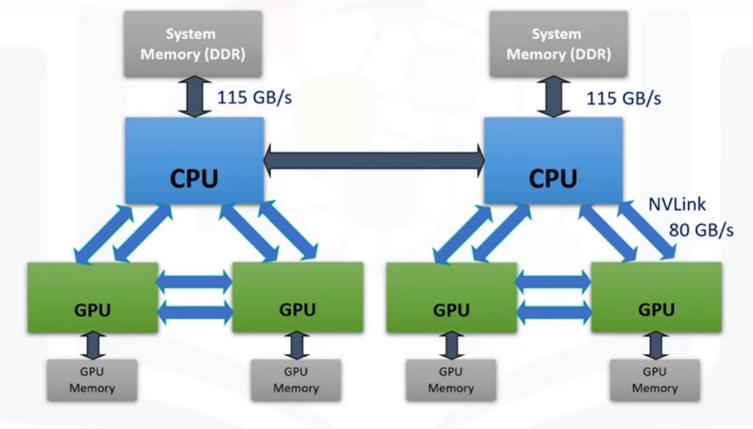
Existen muchos frameworks como TensorFlow, Caffe, Torch, etc para disenar y entrenar modelos de Aprendizaje Profundo. Para acelerar el entrenamiento y la inferencia, haciendo uso de GPUs, éstos modelos necesitan diferentes tipos de librerías como cuDNN y nvCaffe.

PowerAI es un paquete de distribución de software que contiene este tipo de librerías.



De la imagen, consideremos que la base de datos a analizar está en la memoria del sistema de la PC (DDR)
Cuando se analiza estos datos, haciendo uso del GPU, es necesario mover pedazos de esta data hacia el GPU a través del CPU. Esto es un cuello de botella porque la data tiene que pasar por el estrecho PCIe.

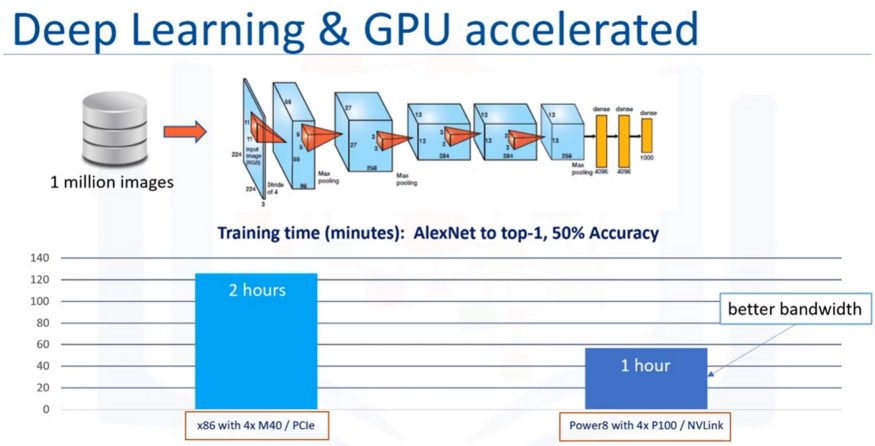
En sí es un problema de ancho de banda porque es el flujo de data lo que va a determinar el desempeno final de la carga de trabajo.



NVLink, es un protocolo de comunicación que permite un alto ancho de banda y por lo tanto rapidez en conexiones punto a punto múltiples.
PowerAI utiliza NVLink para incrementar el ancho de banda de un sistema. Permite que los GPUs ejecuten más ciclos de entrenamiento en menos tiempo.
*IBM usa el Tesla P100 (GPU) y Power8 (CPU)

Tipos de conexiones NVLink

- Entre GPUs: Permite reducir el tiempo de espera del GPU **(disminución del tiempo de copia de memoria caché)**
- Entre GPU y CPU: Permite que haya un acceso rapido a la memoria del sistema, la cual contiene la enorme base de datos. Entonces la recarga de datos desde la memoria del sistema a la memoria del GPU es más rápida



En la imagen se muestra un benchmark del tiempo de entrenamiento entre sistemas que usan PCIes vs NVLink.
Se usa:

- AlexNet, una enorme base de datos (CNN)
- 4 GPUs

Se observa que al haber un mejor ancho de banda, el tiempo de entrenamiento se reduce a la mitad.

Lab
Notebook
https://github.com/IBM/skillsnetwork/blob/master/gpu/nn_scaling.ipynb

Importar el Notebook a Watson Studio
https://github.com/IBM/skillsnetwork/wiki/Watson-Studio-Notebook-Import