Using Gaming GPUs in Deep Learning

Gangwon Jo†, Jungho Park†, and Jaejin Lee‡

† ManyCoreSoft Co., Ltd. ‡ Seoul National University

Introduction

Deep learning on gaming GPUs?

(+) Good performance

· Deep learning does not use DP operations

(+) Cost effectiveness

(-) Cooling problem

· Gaming GPUs are not designed for high-density systems

· Causes erroneous computations: no ECC memory in gaming GPUs

· Shortens the lifetime of GPUs

· Requires fast but very noisy cooling fans

(-) Limited memory capacity

· Training a DNN requires large memory capacity

e.g., VGG-16 network, batch size 64 → 10 GB of GPU memory

Solution: **DEEP Gadget**

A gaming-GPU-based HPC system for deep learning

Two techniques to overcome the problems of gaming GPUs:

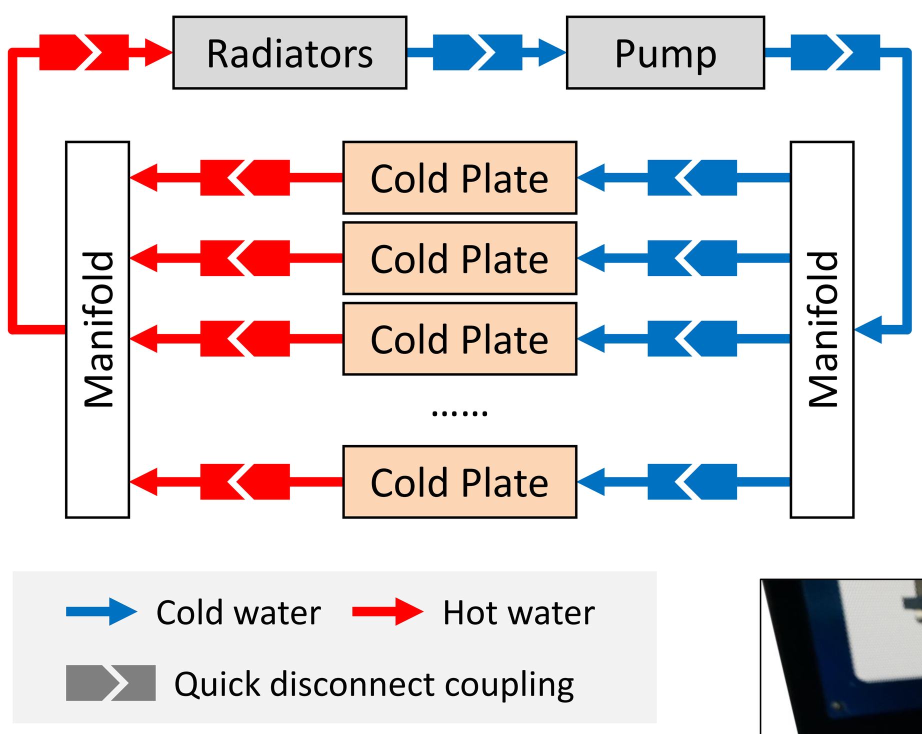
A closed-circuit water cooling system

+

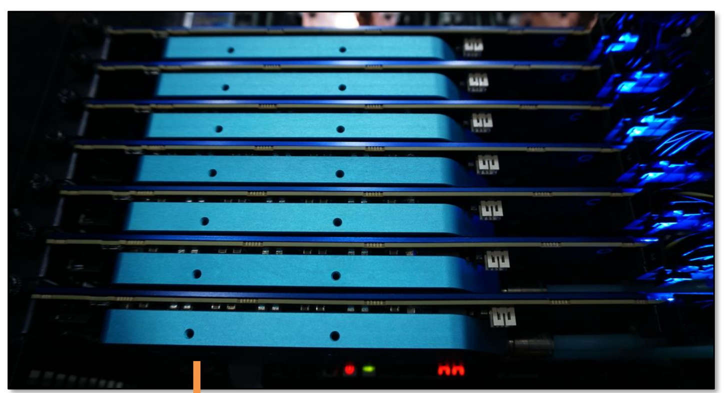
A VMDNN library (virtual GPU memory for deep neural networks)

Water Cooling

Closed-circuit direct water cooling system



- Attaches a cold plate to every GPU & CPU
- Brings water into the plates
- Every component can be easily disconnected
- Note: different from immersion cooling



Single-slot GPU cold plates

· 7 water-cooled GPUs can be installed on a commodity motherboard

· The existing products requires a dual-slot space

Product: DEEP Gadget

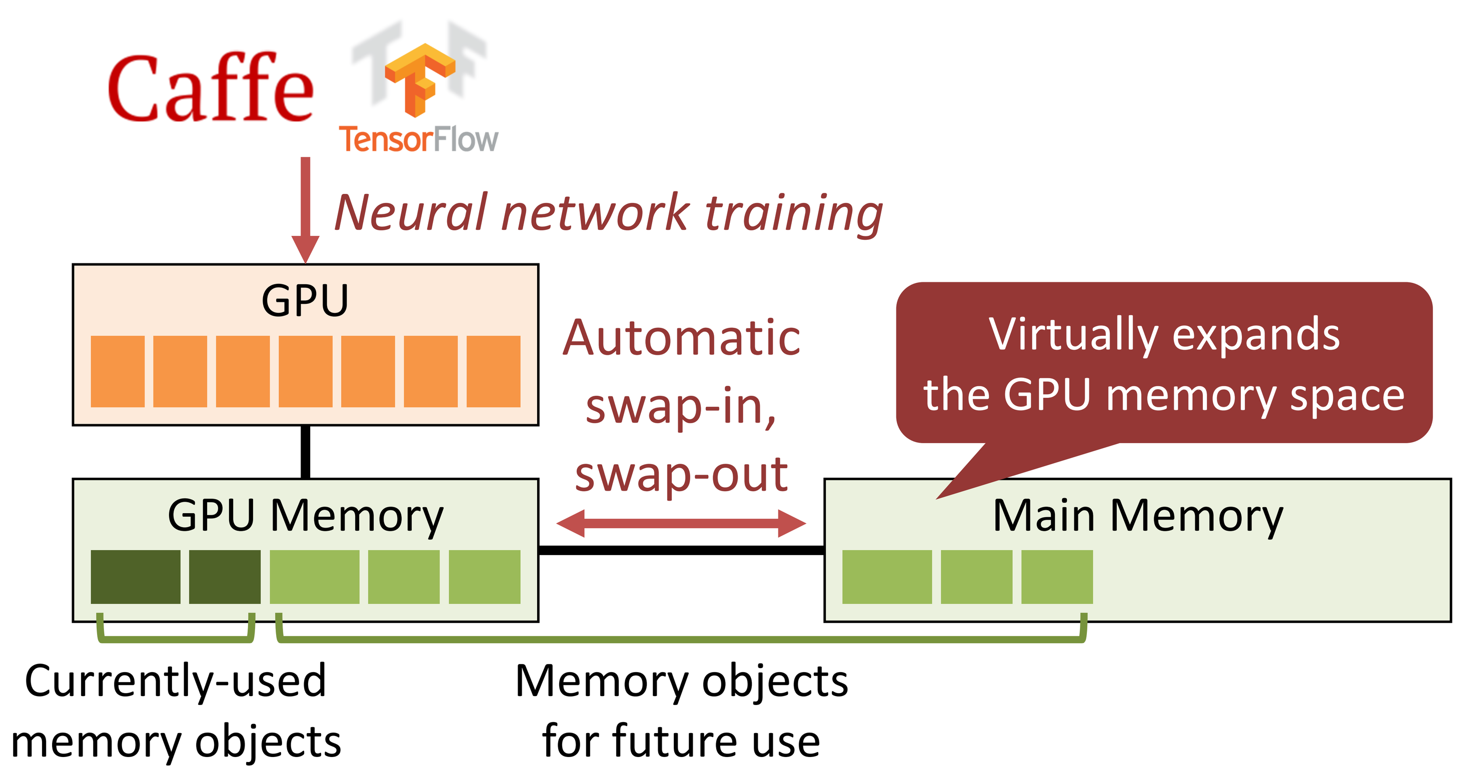
Gaming-GPU-based deep learning appliance

http://deepgadget.com

A possible configuration

Component	Specification
CPU	2x Intel Xeon E5-2630 v4 (water-cooled)
GPU	7x NVIDIA GeForce GTX 1080 Ti (water-cooled)
Main memory	128 GB DDR4 2,400 MHz
Motherboard	ASUS Z10PE-D8
(PCIe slots)	2x PCIe 3.0 @ x16 5x PCIe 3.0 @ x8
Storage	250 GB M.2 NVMe SSD + 32 TB RAID 6 HDD storage (6x 8 TB SATA3 HDD)
Power supply	2x 1,300 W
OS	Ubuntu 16.04 LTS
Software	CUDA Toolkit 9.0, cuBLAS 9.0, cuDNN 6.0, NCCL, VMDNN library

Virtual GPU Memory for DNNs



Based on two observations:

- Each CUDA kernel of deep learning frameworks accesses only the memory objects related to a single layer of the neural network
  ⇒ Most of the memory objects can be swapped out
- A deep learning framework repeats a set of CUDA kernel calls millions of times in a fixed order
  ⇒ We can expect the memory-object access pattern of future kernel calls

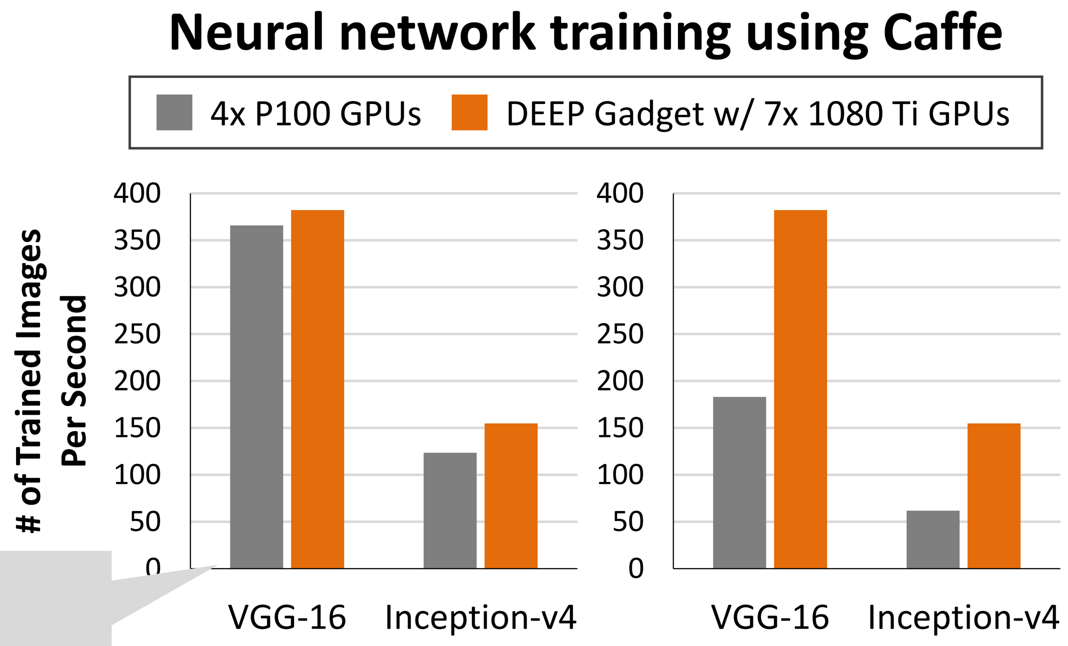
VMDNN library

- Automatically swap in & swap out GPU memory objects
- Generates an optimal swapping schedule to maximally hide memory transfers with GPU computations
  · Better than CUDA Unified Memory
- Transparent to the target deep learning framework
  · Implemented as a shared library & linked with the framework by LD_PRELOAD
  · Intercepts all CUDA kernel calls

Evaluation

- 2—2.5x cost effectiveness
- GPU temperature: 70˚C at full load
- Training a DNN requiring 60 GB of GPU memory

Neural network training using Caffe



4x P100 GPU system
- CPU: 2x Intel Xeon E5-2683 v4
- GPU: 4x NVIDIA Tesla P100 SXM2
- Main memory: 512 GB DDR4 2,133 MHz

(a) Performance **(b) Cost Effectiveness** (Performance Per $25,000)