Title: Sentinel: Efficient tensor migration and allocation on heterogeneous memory systems for deep learning

Conference: HPCA 2021

Institution: University of California, Merced

Paper: <http://www.pasalabs.org/papers/2021/hpca21_sentinel.pdf>

|  |  |
| --- | --- |
| Key Point | Automatically optimizes tensor management on HM.  1, Dynamic profiling  2, coordinates OS and runtime-level profiling. |
| Try To Solve | 1, Memory Capacity is a major bottleneck for training DNNs. => Heterogeneous(异构) memory(HM): combine fast and slow memory can provide bigger capacity.  3, Using HM imposes new challenges on Tensor migration and allocation for DNN training.  4, Prior works relies on DNN domain knowledge and cause unnecessary tensor immigration. |
| Prior works compare and motivations | 1, dynamic profiling: catch the effect of inter-operation parallelism, and can be adapted by various input data size.  2, graph agnostic: not relay on specific model architectures.  4, page level false: Avoiding page-level false sharing is necessary to improve page migration efficiency and achieve additional savings of fast memory usage |
| Dynamic Profiling and Analysis | Dynamic profiling:  1, Collects information: (1) main memory accesses time per tensor, (2) tensor size and (3) lifetime  method: each memory page only have on tensor(a tensor can be keep in multi pages if it’s bigger) + use poisoning PTE  2, Observations:  1, a large number of small tensors (smaller than page size) with short lifetime (within one layer) in DNN training workloads. – In ResNet-32; 92% short lifetime tensor and 98% small tensor.  Transformer模型详解（图解最完整版）  2, more than half of the tensor was cold, while only a little tensor is hot.  3, some cold tensor was stored in hot pages – page-level false. |
| How to Do | Step1, Dynamic Profiling, use 1 training step to collect information.  Step2, Allocate a continuous memory space in fast memory to storage these short lived tensor and not involve tensor movement – only takes a small capacity and only keep for this layer’s execution;  Step3, For long-lived tensors: because it will be used by another layer(sparsely and periodically). We migrate them based on their access frequency. — split each training step (1 forward + 1 backward) into many migration intervals based on DNN topology. And for each interval, migrates tensor needed for the next interval. – overlap with execution.  Two new Problems for tensor migration from slow to fast  1, tensor migration not finished because of lack of space.  1, based on the profiling result, if the tensor was not needed by remaining operation in this interval, start to offload data.  2, set a proper interval length.  2, tensor migration not finished because of lack of time.  1, let the operation wait for the tensor migration or keep the tensor in slow memory. It need a balance, so in the framework, when this case happened, use another two training step to test both of these two methods and select the better one. – this is not need for GPU – CPU set.  2, set a proper interval length  3, for ResNet-32, the author set a series evaluation for different interval length. |