tar2h5: Small Files Packer for Machine Learning Tasks

Gerd Heber (The HDF Group), Dawei Mu (NCSA), Volodymyr Kindratenko (NCSA)
HAL System Overview

- NSF-funded IBM cluster for Deep Learning applications
  - 16x compute nodes,
  - 640x physical CPU cores,
  - 64x Nvidia V100 GPUs
  - 224 TB of All-Flash Storage

- The Origin of Machine Name
  - 2001: a space odyssey
  - Early concept of an artificial intelligence system
  - Didn’t end well and we decided to give “him” a second chance
HAL System Overview

- IBM POWER9 CPUs
  - 14nm finFET semiconductor
  - Stronger Thread Performance – SMT
  - POWER ISA 3.0 Architecture
  - Enhanced Cache Hierarchy
  - NVIDIA NVLink 2.0
  - I/O System – PCIe Gen4
- 2x 20 Cores with SMT4
  - Map to OS as 160 CPUs per node
HAL System Overview

- **NVIDIA V100 GPUs**
  - Peak 7.8 TFLOP/s (double-precision).
  - Peak 15 TFLOP/s (single-precision).
  - SM / Cores : 80 / 5120.
  - HBM2 Memory 16 GB : 900 GB/s.
  - Config up to 128 KB L1 Cache per SM.
  - Config up to 96 KB Shared Memory per SM.
  - Constant memory 64 KB.
  - 65536 32-bit Registers per SM.
  - Clock Frequency : 1530 MHz
HAL System Overview

• DDN GS400NVE Flash Arrays Server
  • 224 TB usable GPFS
  • 8x EDR Infiniband 100 GB/s bandwidth
  • The mean time to list all home directories has been 7ms with a standard deviation of just 17.8%
HAL Software Overview

• HAL Software
  • OS : CentOS Linux 7.7
  • Compilers :
    • GNU 4.8.5
    • Advance Toolchain 12.0
    • IBM XL 16.1.1
    • CUDA 10.2.89
    • PGI 2019.10

• Tools :
  • PowerAI 1.7.0 (Watson Machine Learning Community Edition)
  • OpenMPI 4.0.3
  • CMake 3.14.0
  • Singularity 3.5.3
The IO Challenge

- The catalyst of the rise of machine learning - Datasets
  - dataset composed of millions of small files
  - dominant random-access pattern

- Researchers new to this area
  - have expertise in domain science
  - don't have a lot of HPC experience
  - use small files as dataset
  - produce overwhelming workload to shared storage
  - => we used to reduce compute nodes to ease the I/O pressure
Tar2h5: Small Files Packer

- Convert Tape ARchives to HDF5 files
  - easy to use
  - solutions for different scenarios
- Functions
  - archive checker
  - h5compactor
  - h5shredder
archive checker

- archive_checker
  - check how many files can be extracted from the input tar file.
- archive_checker_64k
  - check if any files within input tar files larger than 64 KB.
h5compactor

- h5compactor
  - converts input tar file to HDF5 file, all files within tar file should smaller than 64KB, using the file names as dataset names.
- **h5compactor-sha1**
  - converts input tar file to HDF5 file, all files within tar file should smaller than 64KB, using files' SHA-1 digest as dataset names.
• h5shredder
  • converts input tar file to HDF5 file, *no file size limitation*, concatenate data, names, and offsets into 4 separate arrays for random access.
User Case

• Project Title
  • Efficient Large-Scale Video Generation with GANs

• HAL User
  • CS Research Assistant: Daniel B. McKee

• Data Info
  • large Kinetics-400 video dataset composed of around 240k videos
  • the number of files is around 25 million
  • total size of the JPG dataset around 125GB as a tar file

• Performance Comparison
  • Loading from the compact HDF5 file made training about 5x faster
The tar2h5 tools are available on GitHub

https://github.com/HDFGroup/tar2h5
Visualization with HDFCompass

• https://support.hdfgroup.org/projects/compass/
Future Work

• Mixed Precision Support in HDF5
  • IEEE FP16
  • Google BFloat16
  • NVIDIA TensorFloat (TF32)
  • AMD FP24
THANK YOU FOR YOUR TIME!