NGI Initiative
Toward a bridge to the semantic gap

Jean-Thomas Acquaviva
jtacquaviva@ddn.com

ESiWACE has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 675191
www.esiwace.eu
Breakdown of the I/O Critical Path

HDD dominates I/O latency
Software layer is the bottleneck
Object is an appealing and generic concept

Adding semantic over object layer
→ Burden shifted to the application dev.
→ or add a component to provide a link

\textit{obj-uuid} ↔ \textit{my-semantic}
obj-uuid ↔ my-semantic

→ Add some extra software layers
→ Hey, my-semantic is critical for data interpretation

→ Let‘s add High Availability and Fault Tolerant to the mechanism
→ HA database is complicated let‘s use standard software component
→ HDFS comes into play
→ Performance get killed by the Software
TensorFlow already includes many filesystem implementations, such as:

→ A standard POSIX filesystem (NFS)
→ HDFS - the Hadoop File System
→ GCS - Google Cloud Storage filesystem
→ S3 - Amazon Simple Storage Service filesystem
→ A "memory-mapped-file" filesystem

It is possible to implement a custom filesystem
SUMMIT (IBM) EFFICIENCY
#1 on IO500 at SC18

ior_easy_read 1788.320 GB/s
ior_hard_read 27.403 GB/s
→ **1.53% efficiency**!

ior_easy_write 2158.700 GB/s
ior_hard_write 0.572 GB/s
→ **hey that’s 0.025% efficiency**!

[Summary] Data files in
/gpfs/alpine/stf007/scratch/gmarkoma/io-500-dev/datafiles/io500.2018.11.09-03.12.50

Don’t worry that much, ADIOS lib will shield the FS from such pattern.
SEG-Y is a file format heavily used in the O&G community → Originally to store Seismic data on Tape → Large legacy (and new) software base → Huge data legacy

A segy file contains a header and a sequence of data which describes the trace coordinates etc and the trace data itself:
Brings the known advantages of specialized lib.

- Easy to use, geophysicist-friendly C++ and C APIs
- Reduces maintenance → Reduces codebase sizes substantially
- Multi-Layer solution → separate file-format processing, layout and
- Scalable / Performance

- Computational geophysicists / software engineers writing seismic processing software on HPC clusters
  → without needing to HPC hardware experts (but actually they still are)

References


"ExSeisPIOL: Extreme-Scale Parallel I/O for Seismic Workflows", RICE O&G HPC, 2017 https://www.youtube.com/watch?v=Y00Js6uPWI0
ExaSeisDat: Results

- Transparent insertion of a new storage layer in the workflow

- Comparing GPFS against an IME setup.
- Sort of a 400 GiB SEG-Y file (400 GiB read, 400 GiB write), 4 nodes.
- Total Time: 63% GPFS I/O, only 5% IME I/O for sort

Courtesy of Cathal O’Broin
Next Generation Interface

Goals:
→ Maintain / increase SW developer productivity and code reliability
→ Deliver Robust performance
  → Protect file system from deviant I/O patterns
  → Internal support of hardware diversity
→ Log structured has proven to be fairly good
Next Generation Interface

→ Keep the software stack under control
→ Keep semantic close to end-user
  → Specialized layout have a strong track record
  → Object are too generic (slightly controversial!)

→ Semantic gap
  → Storage stores 0 and 1
  → Not possible to bring computation to 0 & 1
  → Parallel File System with data slicing are making things worse

→ Bringing compute closer to storage semantic
  → Semantic Storage Layer tools
  → Earth System Data Middle Ware
Layout has to be though jointly with workflow
ESiWACE has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 675191

www.esiwace.eu