I/O Characteristics of Scientific Applications

Chen Wang[1], Adam Moody[2], Elsa Gonsiorowski[2], Kathryn Mohror[2], Marc Snir[1]

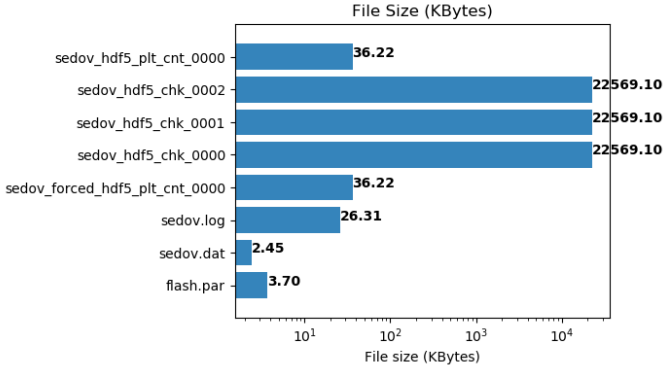
1. University of Illinois at Urbana-Champaign
2. Lawrence Livermore National Laboratory

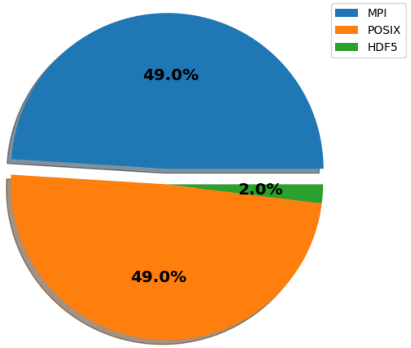
Some interesting questions

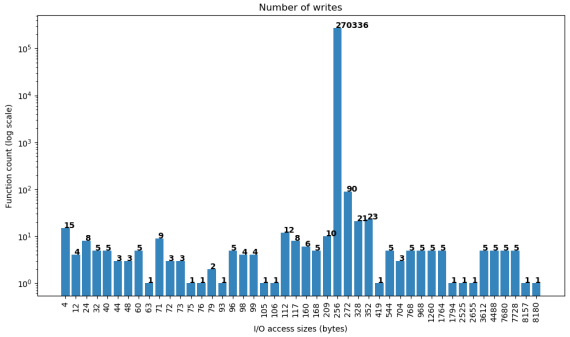
- To what extend can we relax POSIX semantics to improve I/O performance without breaking applications?
- What function calls (API) are needed to implement a file system?
- Do different processes read or write to the same part of a file?
- Do any applications actually write to the same bytes in a file twice?
- Do processes change files after they close the file? Or are the file contents permanent after closing the file?

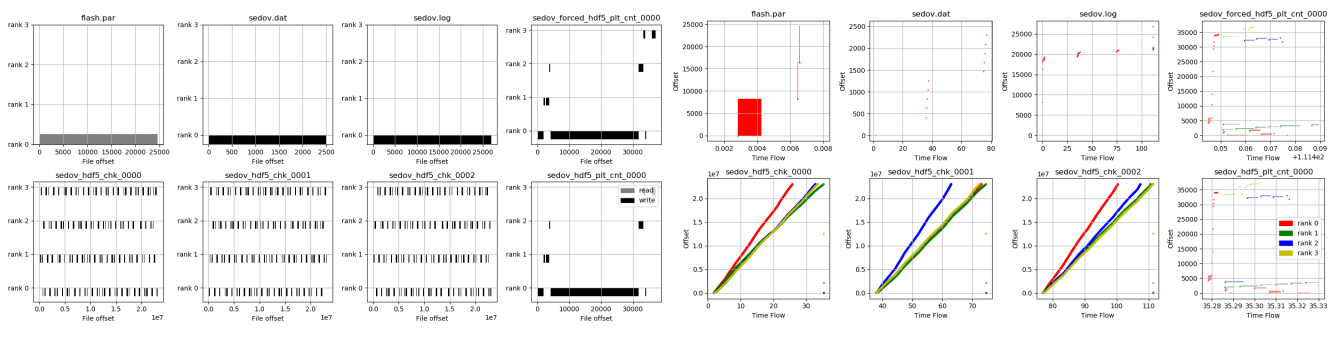
Recorder - a lightweight tracing library

- Able to trace HDF5, MPI-I/O, and POSIX I/O
- Captures all arguments with compressed encoding
  - Many analyses and visualization tasks and can be done without decompression
- Detailed visualization report
  - Function counts and function set
  - Access patterns of each rank
  - I/O granularities









Traces from 22/25 applications

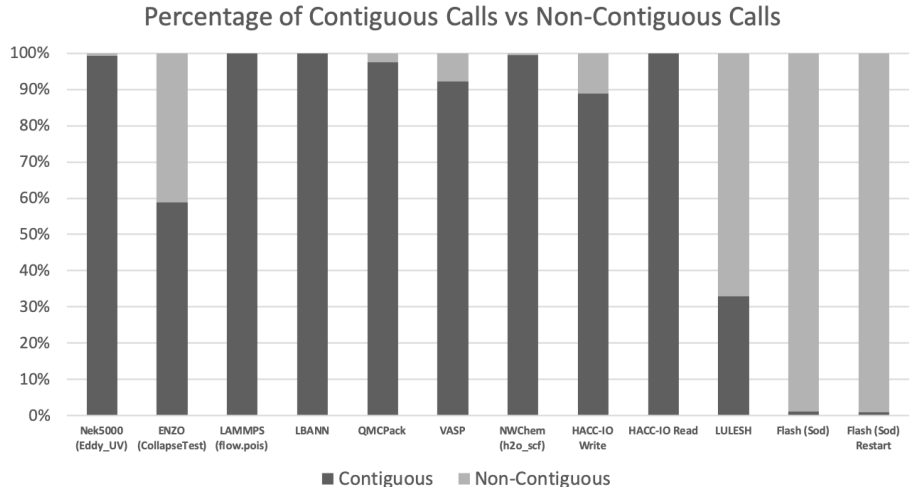
#	App	Version	Description
1	Flash [2]	4.4	
2	Nek5000	v19.0-rc1	High-order solver for computational fluid dynamics.
3	IOR	3.3.0+dev	Parallel filesystem I/O benchmark.
4	QMCPACK [10]	3.7.0	Electronic structure code that implements numerous Quantum Monte Carlo (QMC) algorithms.
5	VASP	5.4.4	Vienna Ab initio Simulation Package for atomic scale materials modelling.
6	LULESH [9]	2.0	Livermore unstructured Lagrangian explicit shock hydrodynamics.
7	ENZO	2.5	AMR simulation code for rich, multi-physics hydrodynamic astrophysical calculations.
8	LBANN [14]	1.0.0	Livermore big artificial neural network toolkit.
	ExaWind:		
9	- Nalu-Wind [6]	1.0	Wind energy focused variant of Nalu.
10	- OpenFast [1]	1.0.0	Open-source wind turbine simulation tool that was established with the FAST v8.
11	HACC-IO	1.0 beta	Hardware accelerated cosmology code simulation.
12	NWChem [13]	6.8.1	Open source high-performance computational chemistry.
13	ParaDis	2.5.1.1	Large scale dislocation dynamics simulation code to study the fundamental mechanisms of plasticity.
14	Keras [5]	2.2.4	A high-level neural networks API, written in Python.
15	Chombo [3]	3.2	Software for adaptive solutions of partial differential equations.
16	GTC [11]	0.92	Parallel, particle-in-cell code for turbulence simulation.
17	GAMESS [8]	0.92	General atomic and molecular electronic structure system.
18	Adcirc	53.04	A system for solving time dependent, free surface circulation and transport problems.
19	E3SM/CESM		
	Exaalt:		
20	- LAMMPS [12]	12Dec 18	Large-scale molecular dynamics code with a focus on materials modeling.
21	- LATTE [4]	1.2.1	Open source density functional tight binding molecular dynamics.
21	GTC-P		
22	MILC QCD	7.7.11	MILC collaboration code for lattice QCD calculations.
23	MSAProbs [7]	1.0.5	Parallel and accurate multiple sequence alignment.
24	mpiBLAST		Parallel implementation of NCBI BLAST
25	HavoqGT	0.1	HavoqGT is a framework for expressing asynchronous vertex-centric graph algorithms.

Some observations

- Rarely read back once written or closed.
- Many overlapping reads, almost no overlapping writes.
- Many attributes, e.g., last access time, last modification time, and last change time, are never used by applications directly.
- Most files are read-only or write-only.
- Many metadata related functions are called behind the scene.

Func	FLASH	Nek5000	LAMMPS	QMCPACK	ENZO	VASP	LBANN
lstat	✓			✓	✓		
lstat64	✓						
stat	✓				✓		
stat64		✓		✓		✓	
fstat	✓			✓	✓		
fstat64		✓				✓	
getcwd	✓			✓	✓	✓	
access	✓				✓		
faccessat							✓
umask	✓		✓	✓			
fileno		✓		✓		✓	✓
readlink				✓		✓	
unlink						✓	✓
mkdir		✓			✓		
readdir							✓
closedir							✓

App	R/W only	R→R	W→W	R→W	W→R
Flash	✓	×	×	×	×
Nek5000	✓	S;M	×	×	×
LAMMPS	✓	×	×	×	×
VASP	✓	S;M	S	×	×
QMCPack	✓	M	S;M	×	×
ENZO	×	×	×	S	×
LBANN	✓	M	×	×	×



More work to be done

- Collect traces for different configurations, e.g., different problem sizes, with/without OpenMP, with MPI-hints, etc.) and on larger scales.
- What is the minimum POSIX semantics requirements for applications?
- How much do reads skip around files?
- What is the read cache paging effectiveness for a given cache/page size?