TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

20.09.2024

Enable I/O Metadata Analysis in Score-P and OTF2

Sebastian Oeste (CIDS TU-Dresden), Radita Liem (RWTH Aachen University), Bert Wesarg (GWT TU-Dresden)

Parallel Tools Workshop
Dresden

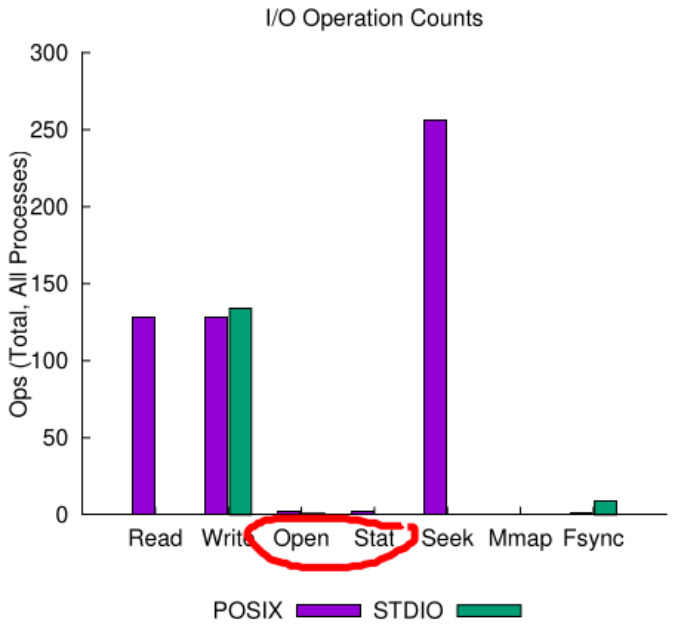
ZIH
Center for Information Services
and High Performance Computing

Motivation – What are I/O Metadata Operations?

- Operations that **create**, **read** or **modify** metadata records of a file.
- Metadata records of a file are represented as **inode** on Unix-like systems.
- An inode contain fields like:
  - Device ID‘s where the file resides
  - File type (regular, directory, socket, pipe, fifo, character / bock device)
  - Link count
  - User / Group ID
  - File size
  - Timestamps …
- **I/O Metatdata operations create, read or modify inode entries.**

Related Work - Darshan

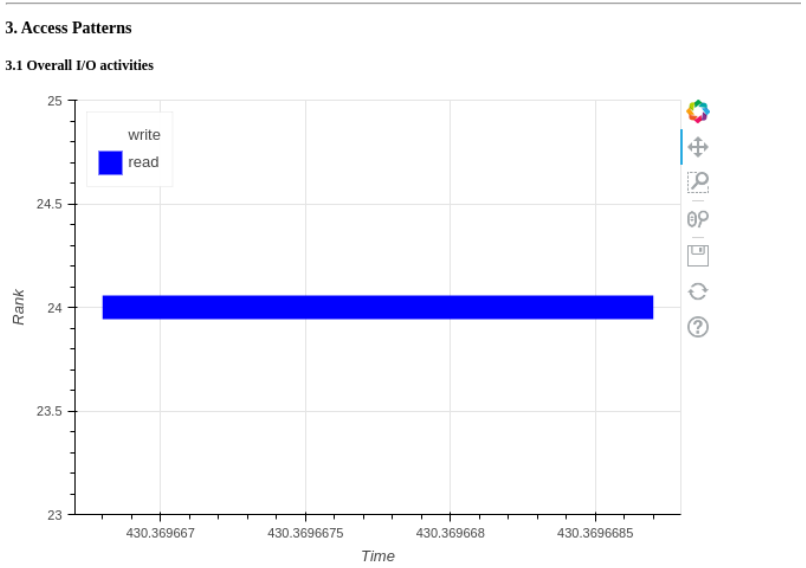
- Widely used for HPC I/O Analysis
- Aggregated profiles are default
- Focus on read and write pattern
- DXT Trace Records just for read and write

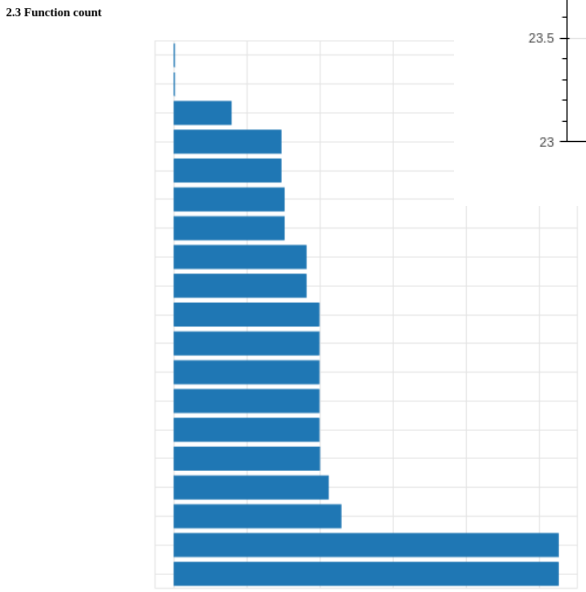


https://www.mcs.anl.gov/research/projects/darshan/data/

Related Work - Recorder

- Focus on I/O tracing
- Most I/O functions are recorded
- Hard to analyze real world applications
- Trace format not stable



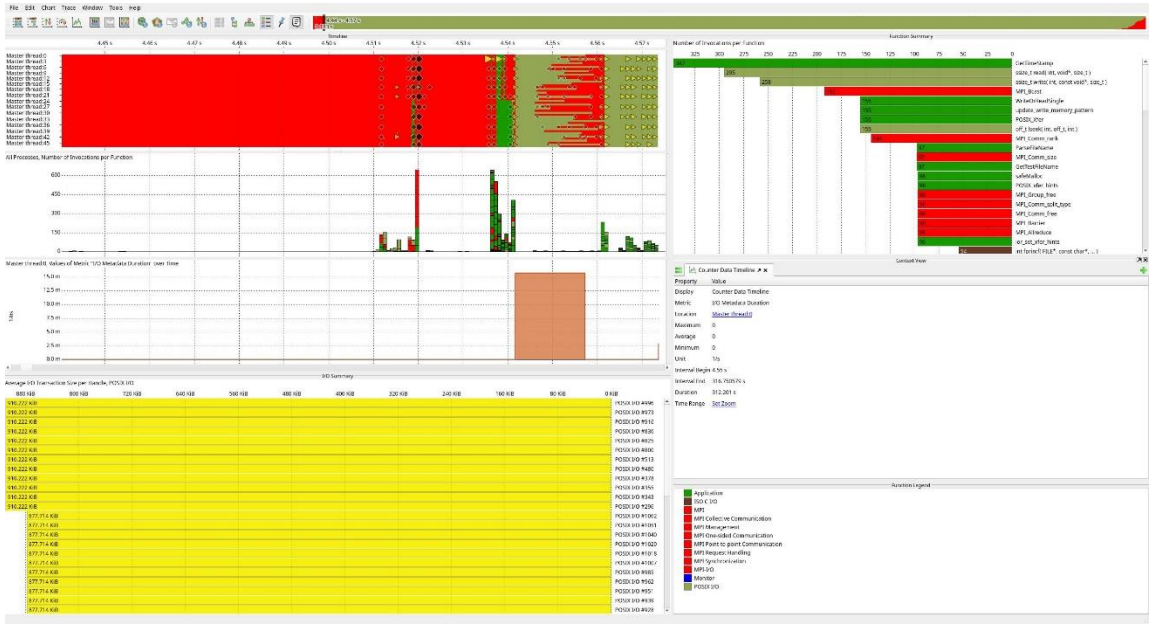


https://recorder.readthedocs.io/latest/index.html

Related Work – Score-P

- Score-P – partially supports I/O metadata operations
  - Open / create , close, sync, fsync, unlink
  - Stable trace format
  - Scalable analysis infratstructure (Vampir/VampirServer)
  - Harder to use

Score-P
Scalable performance measurement infrastructure for parallel codes



OTF2
Open Trace Format 2

I/O Tools support

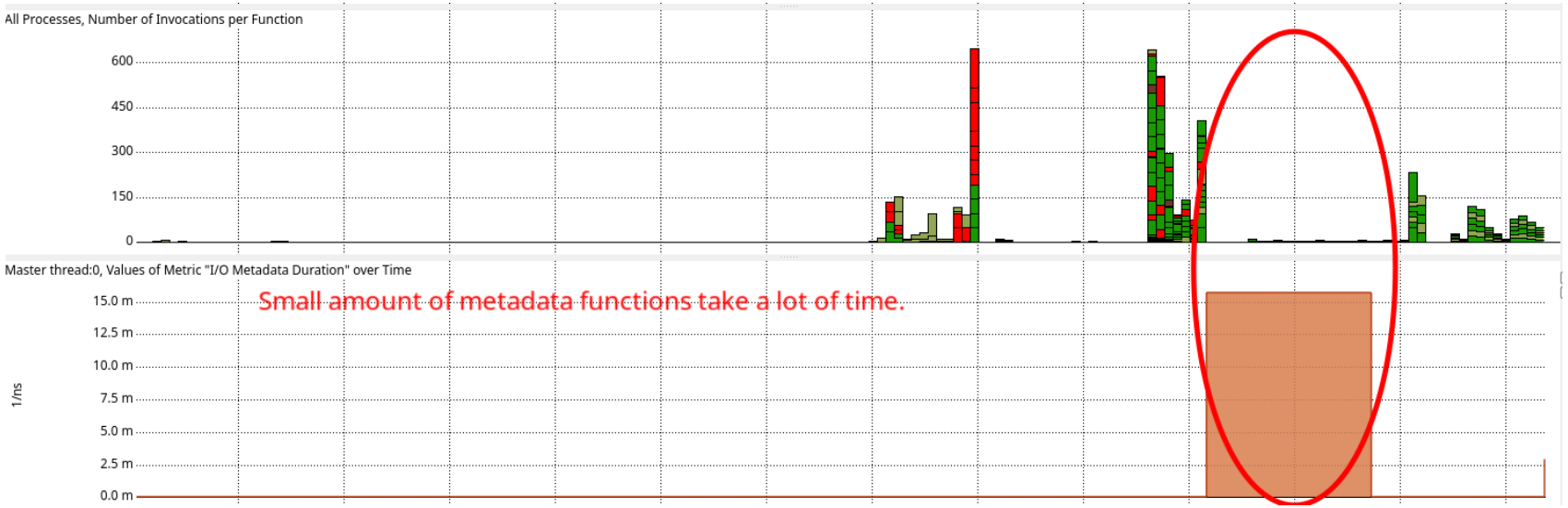
	Functions
All tools **(28)**	open, open64, creat, creat64, dup, dup2, read, write, pread, pwrite, pread64, pwrite64, readv, writev, lseek, lseek64, fsync, fdatasync, close, fopen, fopen64, fdopen, fclose, fwrite, fprintf, fread, fseek, fseeko
Darshan & Recorder only **(10)**	fileno, __xstat, __xstat64, __lxstat, __lxstat64, __fxstat, __fxstat64, mmap, mmap64, rename
Darshan & Score-P only **(29)**	openat, dup3, preadv, preadv64, preadv2, preadv64v2, pwritev, pwritev64, pwritev2, pwritev64v2, aio read, aio write, aio_return, lio_listio, freopen, fflush, fputc, fputs, printf, vfprintf, vprintf, fgetc, fscanf, vfscanf, fgets, fseeko64, fsetpos, fsetpos64, rewind
Recorder & Score-P only **(6)**	ftell, unlink, closedir, fcntl, remove, ftello
Darshan only **(16)**	__open__2, openat64, mkstemp, mkostemp, mkstemps, mkostemps, aio_read64, aio_write64, aio_return64, lio_listio64, freopen64, putw, getw, _IO_getc, _IO_putc, __isoc99_fscanf
Recorder only **(28)**	msync, getcwd, mkdir, rmdir, chdir, link, linkat, symlink, symlinkat, readlink, readlinkat, chmod, chown, lchown, utime, opendir, readdir, rewinddir, xmknod, xmknodat, pipe, mkfifo, umask, access, faccessat, tmpfile, truncate, ftruncate
Score-P only **(22)**	lockf, pselect, select, sync, syncfs, unlinkat, aio cancel, aio error, aio fsync, aio suspend, fgetpos, flockfile, ftrylockfile, funlockfile, getc, getchar, gets, putchar, puts, scanf, ungetc, vscan

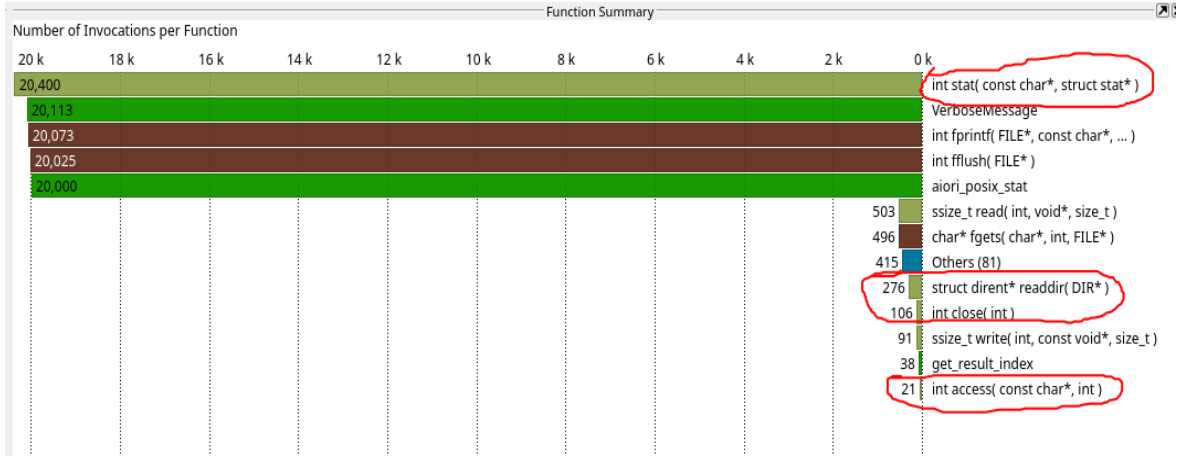
Mango-IO: I/O Metrics Consistency Analysis
Radita Liem; Sebastian Oeste; Jay Lofstead; Julian Kunkel

What we did so far…

- Added missing metadata operation to Score-P
- First draft just add enter / leave events
  - Sufficient to count calls and measure latencies
  - Not enough for a holisitic I/O Analysis or I/O pattern detection
- What we want: use of OTF2 IO records
  - Need additions in OTF2
  - Add more semantic e.g. affected file, type of operation
  - Useful to group access on a per-file basis
  - Allows for consistency analysis, metadata semantics

Enter / Leave for I/O Metatdata Operations





New I/O Metadata functions in Score-P

Operation Group	Functions
Create	mkdir, mkdirat, link, linkat, symlink, symlinkat, pipe, mkfifo, mkfifoat, opendir, fdopendir, mkstemp
Read	stat, statx, fstat, lstat, fstatat, access, faccessat, readdir, readdir_r,
Write	rename, renameat, chmod, fchmodat, chown, fchownat, lchown, utime, utimensat, utimes, umask, truncate, ftruncate
Delete	rmdir, closedir

- The green highlighted functions work on handles (struct DIR* or file descriptors)
- The others work on file names.

I/O Definitions in OTF2 – State of the Art

IoFile:
- Polymorph definition of an **IoFile**
- Definitions for **IoRegularFile** and **IoDirectory**
- **IoFile** can be seen as an inode for metadata operations

IoHandle:
- Reflects a file descriptor based on a **IoFile** defintion

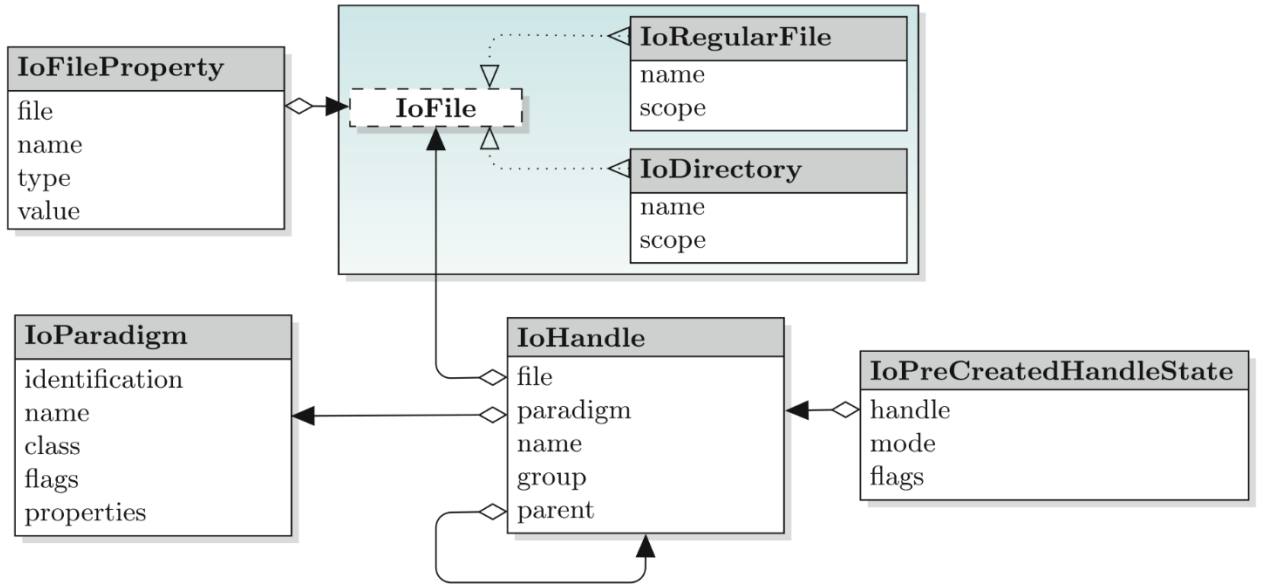
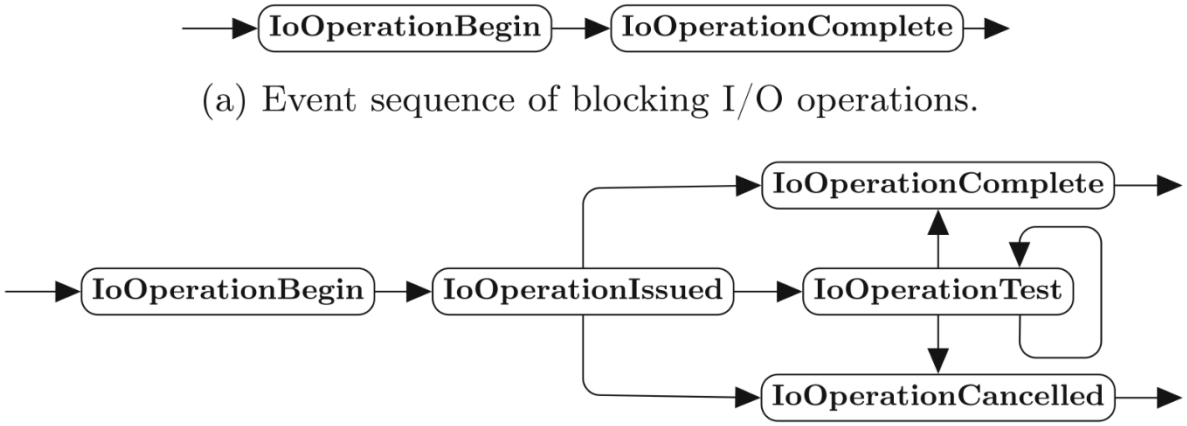


Fig. 2. Overview of definitions to reflect I/O resources and their relationships.

A Methodology for Performance Analysis of Applications Using Multi-layer I/O
Ronny Tschüter(B) , Christian Herold, Bert Wesarg, and Matthias Weber

I/O Events in OTF2 – State of the Art

- **IoCreateHandle**
- **IoDestroyHandle**
- IoDuplicateHandle
- IoSeek
- IoChangeStatusFlags
- **IoDeleteFile**
- IoAcquireLock.
- IoReleaseLock
- IoTryLock



(a) Event sequence of blocking I/O operations.

(b) Event sequence of non-blocking I/O operations.

Fig. 5. Sequence of generated events for different I/O operation types.

A Methodology for Performance Analysis of Applications Using Multi-layer I/O
Ronny Tschüter(B) , Christian Herold, Bert Wesarg, and Matthias Weber

Discussion: Proposal for I/O Metadata in OTF2

- Addition of an **IoCreateFile** event as counterpart to **IoDeleteFile**
  - Just creation of a file no handle will be produced
  - Might change existing implementations of calls like`*open* ` **IoCreateFile** + **IoCreateHandle**
- **IoLink, IoPipe, IoFifo** definition from **IoFile** to support link and symlink, mkfifo, pipe, etc...
- **IoMetadataOperationType** enum to indicate type of metadata operation {create, read, write, delete}
- What kind of events should metadata operations use?
  - IoBeginOperation / IoCompleteOperation?
  - A new generic one?
  - A bunch of new operation defined events?