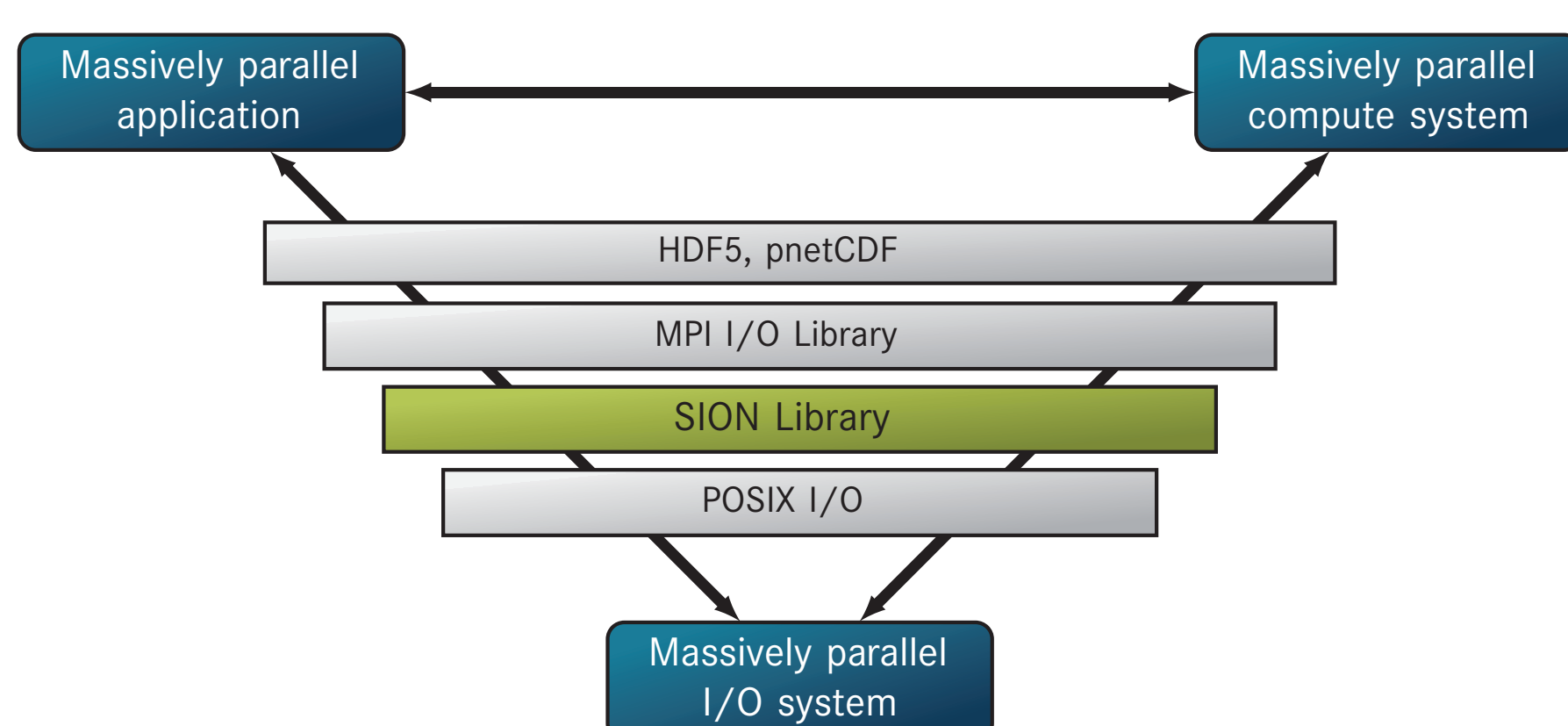


SIONlib: Scalable parallel I/O for task-local files

Contact: Wolfgang Frings

Motivation

- Parallelism of supercomputer hardware increases rapidly
- Simulation applications therefore need to be massively parallel
- File-I/O of applications and tools also becomes increasingly parallel
- Support for task-local binary-stream I/O from thousands of tasks is missing



Features

- Easy-to-use library to read and write binary task-local data from massively parallel applications
- Maps large number of logical task-local files onto one or a small number of physical files
- Meta-data handling
- Parallel open and close
- POSIX-I/O for read and write
- Minimal source-code changes
- Support for MPI, C, C++, Fortran, (OpenMP in development)
- Serial tools for meta-data dumping, defragmentation and splitting of files
- Optimized I/O: Alignment at file system blocks to avoid contention during write

Supported platforms

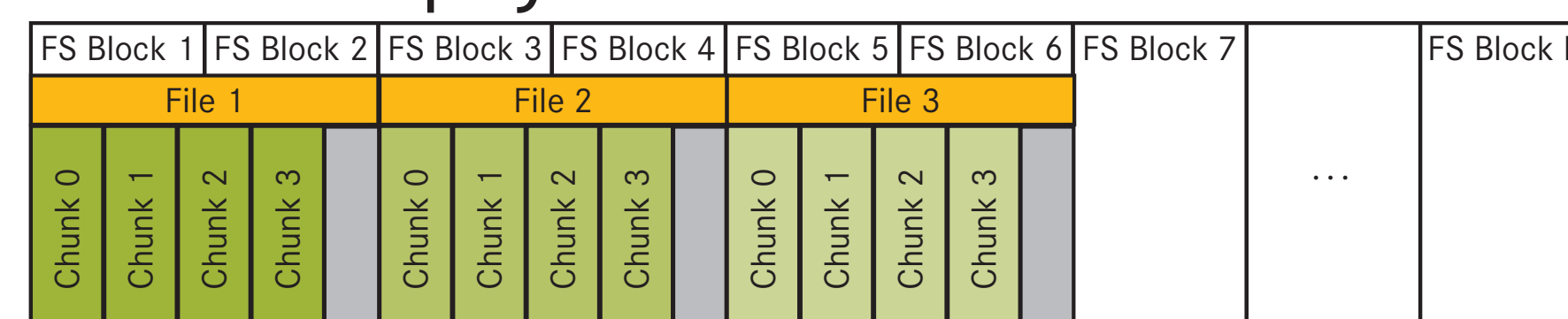
- Cray XT4
- IBM Blue Gene/L and Blue Gene/P
- Linux-based PC clusters

Application area

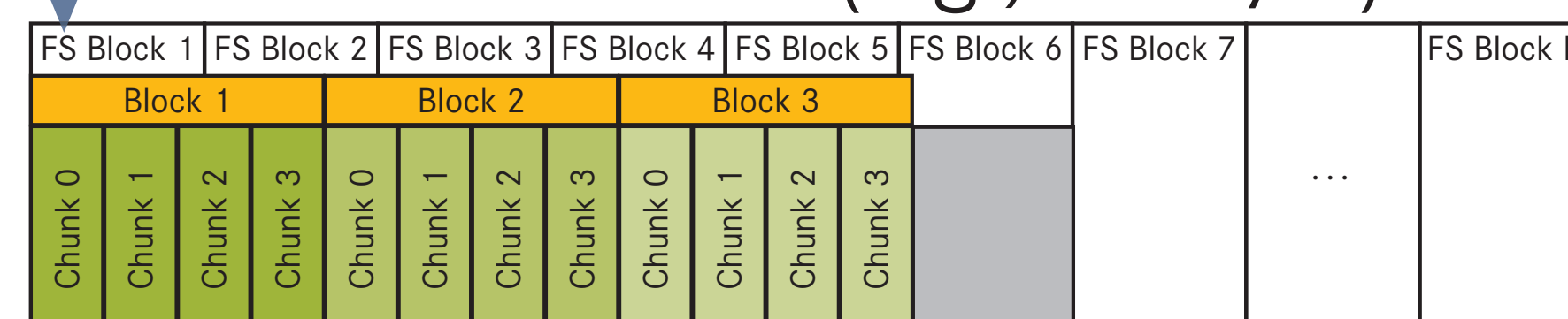
- Parallel applications and tools with binary I/O to task-local files:
 - Performance measurement tools writing/reading trace files
 - Simulation applications writing scratch/checkpoint files

File format schemes

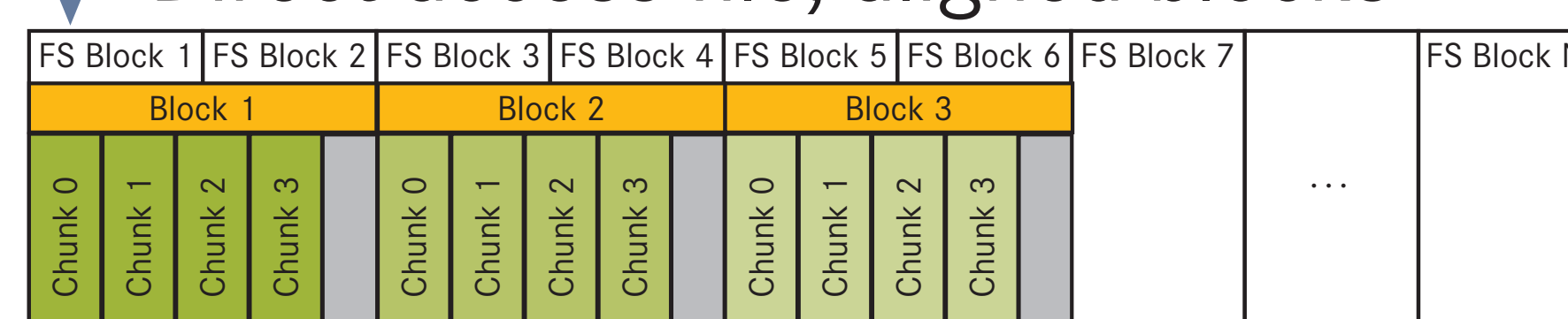
Task-local physical files



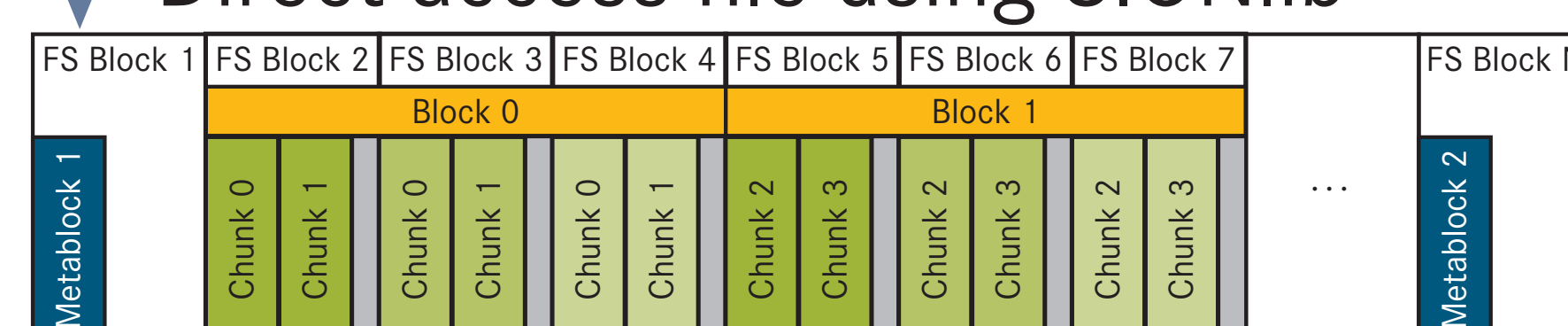
Direct access file (e.g., MPI-I/O)



Direct access file, aligned blocks

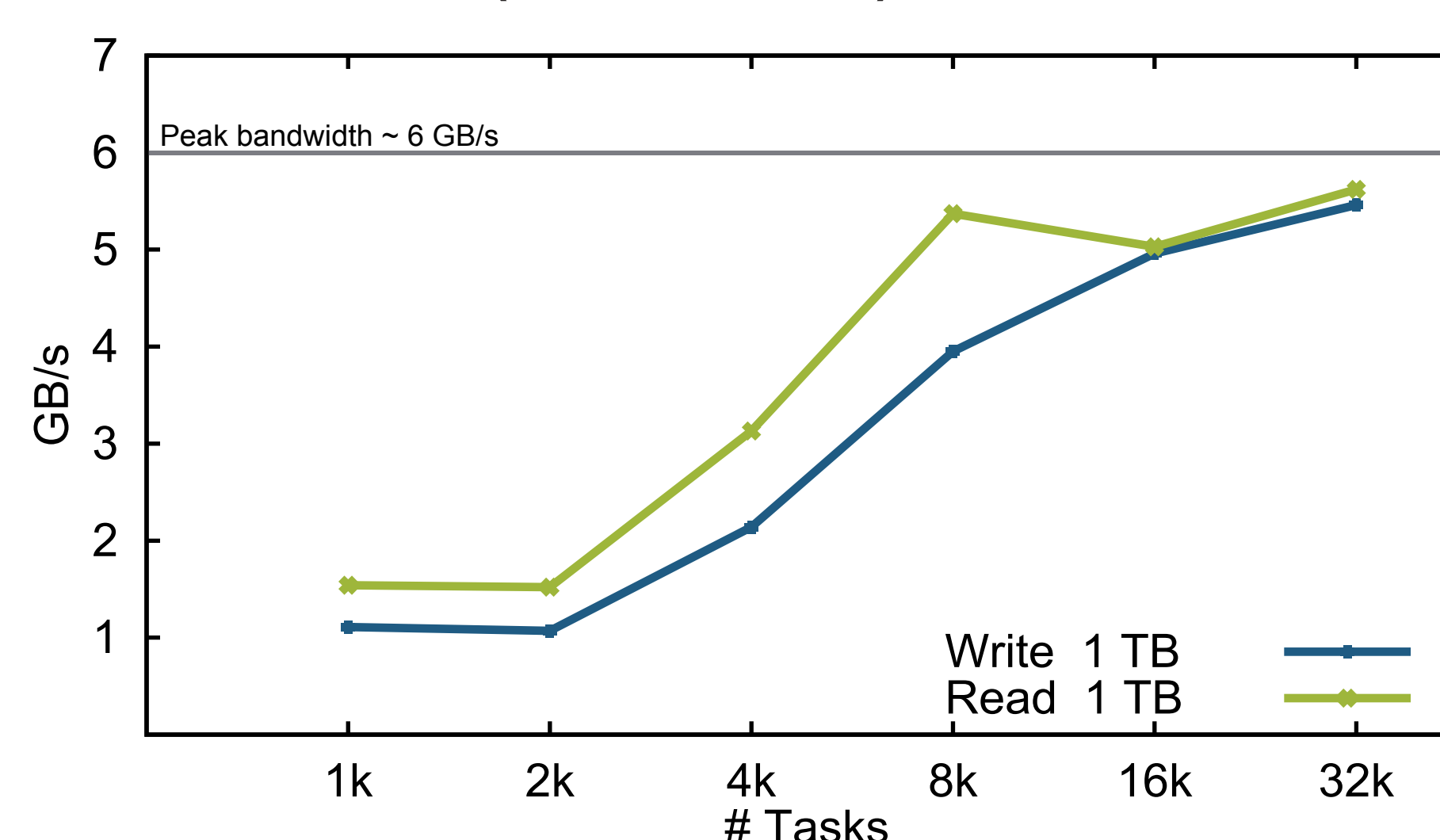


Direct access file using SIONlib

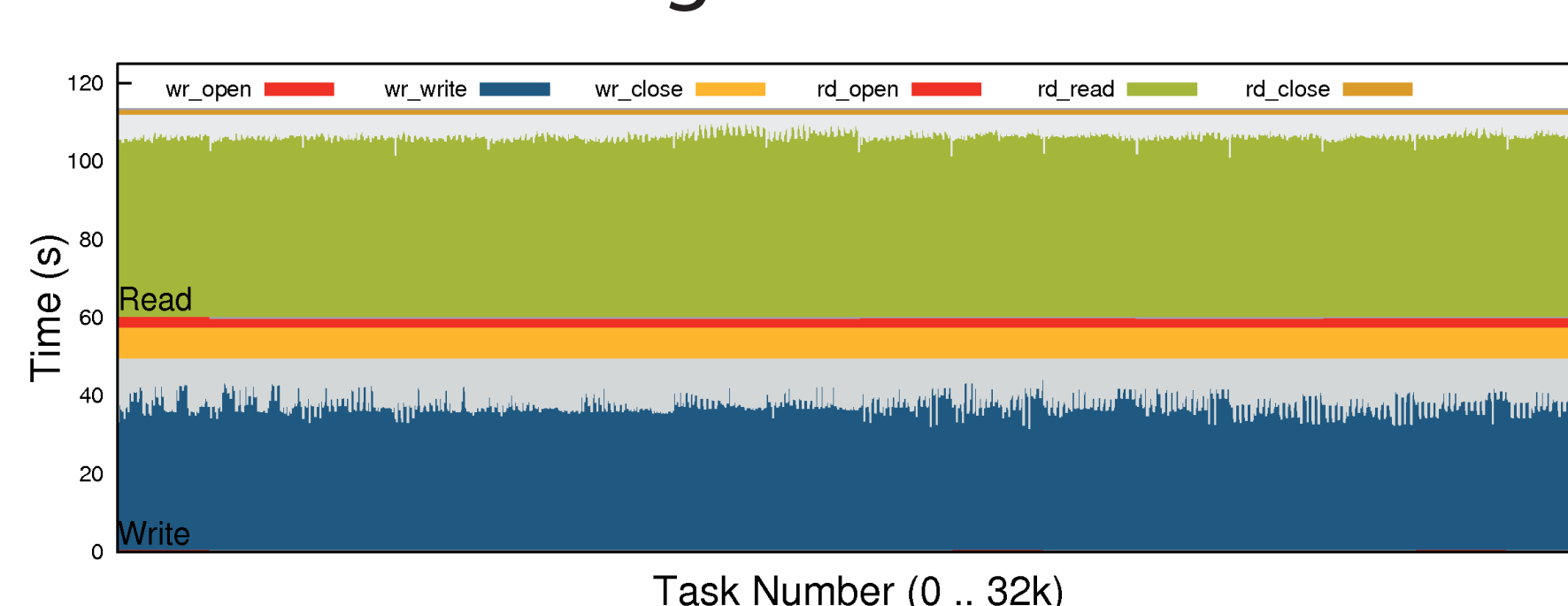


Results

Bandwidth (write/read)



Task-level timings



References

[1] SIONlib: Scalable parallel I/O for task-local files, <http://www.fz-juelich.de/jsc/sionlib>

Advantages

- Significantly reduced time to create and open files in parallel
- Simplified file handling (e.g. listing of directory)
- No performance penalty during read/write

Usage

C-Code: writing with SIONlib

```

int sid;

sid=sion_paropen_mpi(fname,"bw",numFiles,
                    mpicomm,...,
                    &fileptr); // collective

while(loop){
    sion_ensure_free_space(sid,nbytes);
    fwrite(data,1,nbytes,fileptr);
}

sion_parclose_mpi(sid); // collective
  
```

C-Code: reading with SIONlib

```

int sid;
sid=sion_paropen_mpi(fname,"br",numFiles,
                    mpicomm,...,
                    &fileptr); // collective

while(!sion_feof(sid)) {
    btoread=sion_bytes_avail_in_block(sid);
    bread=fread(localbuffer,1,btoread,fileptr);
}

sion_parclose_mpi(sid); // collective
  
```

File create and open

#tasks	single files	MPI-I/O	SIONlib
1024	8.28 s	0.90 s	0.63 s
2048	19.59 s	0.33 s	0.53 s
4096	39.67 s	0.40 s	0.91 s
8192	58.16 s	0.39 s	0.45 s
16384	246.20 s	0.80 s	0.86 s

JUGENE: Time to create and open files, inclusive writing SION metablock

Contention

bandwidth	write	read
SIONlib (aligned)	5.4 GB/s	5.1 GB/s
MPI I/O (not aligned)	2.8 GB/s	3.0 GB/s

JUGENE: 16k tasks, 16 files, 16.5 MB/task