SFIO progress on Swiss-Tx

Swiss-Tx progress meeting September 25, 2000

Emin Gabrielyan

EPFL, Computer Science Dept.
Peripheral Systems Lab.
{Emin.Gabrielyan,RD.Hersch}@epfl.ch

- SFIO on Swiss-T1
- New optimizations of SFIO read/write operations
- SFIO on top of MPICH, performance on T1
- SFIO on top of FCI, performance on T1
- Conclusion
- Future work

SFIO is ported to Swiss-T1

- The SFIO library imports the application environment from CODINE in order to dynamically specify the set of I/O nodes

- Detection of a bug in MPICH on T1 (version 1.1 and 1.2.0): reception of data from the network into fragmented memory pointed by MPI derived datatype.

- SFIO is modified to avoid this bug when running under MPICH.

- Additional SFIO interface functions to dynamically access to the list of I/O nodes

Optimisation of SFIO read/write operations for consecusive single block requests

- Control information transfer optimisation for SFIO read and write operations. Control data transmitted in buffered asynchronous mode.

- Optimisation of transmission of data together with control information. Fragmented data together with controlling arrays are grouped into single datatype.

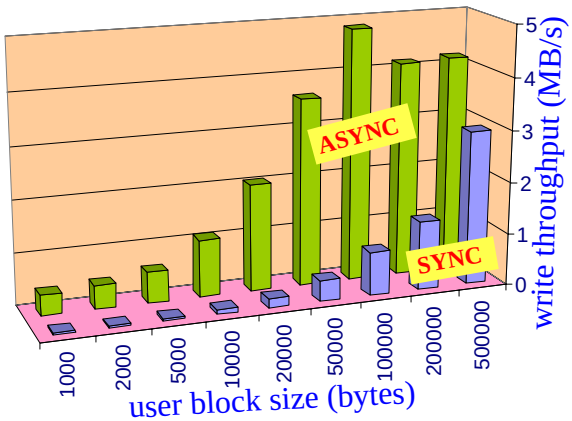
- Asynchronous optimisation of data reception at Compute Node for SFIO read operation.

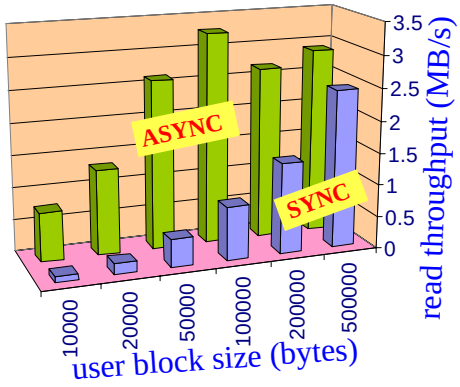
- Modifications of the SFIO library architecture.

- A graphical demonstration of data flow of optimized SFIO read operation.

Performance improvement by optimisation of control data transmission

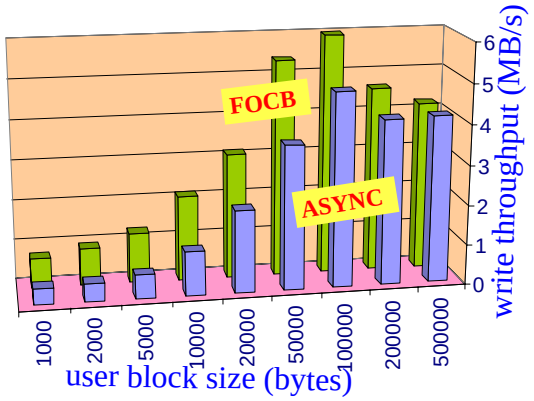
Consecutive accesses to a SFIO 10MB/500B/8IO file on Swiss-T0/Hub

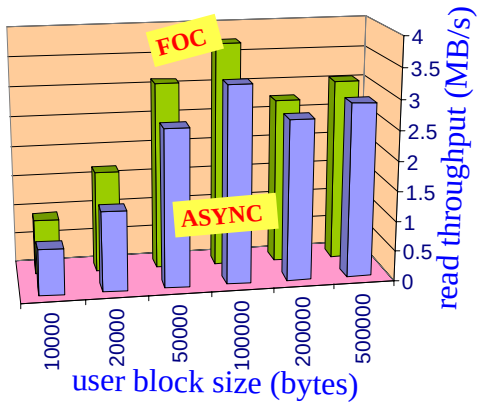




Performance improvement by grouping control data with access data

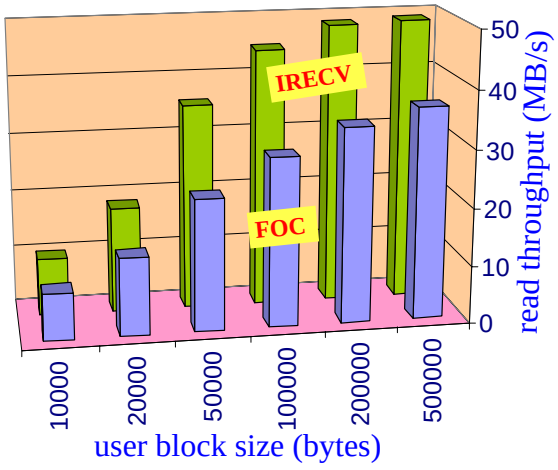
Consecutive accesses to a SFIO 10MB/500B/8IO file on Swiss-T0/Hub



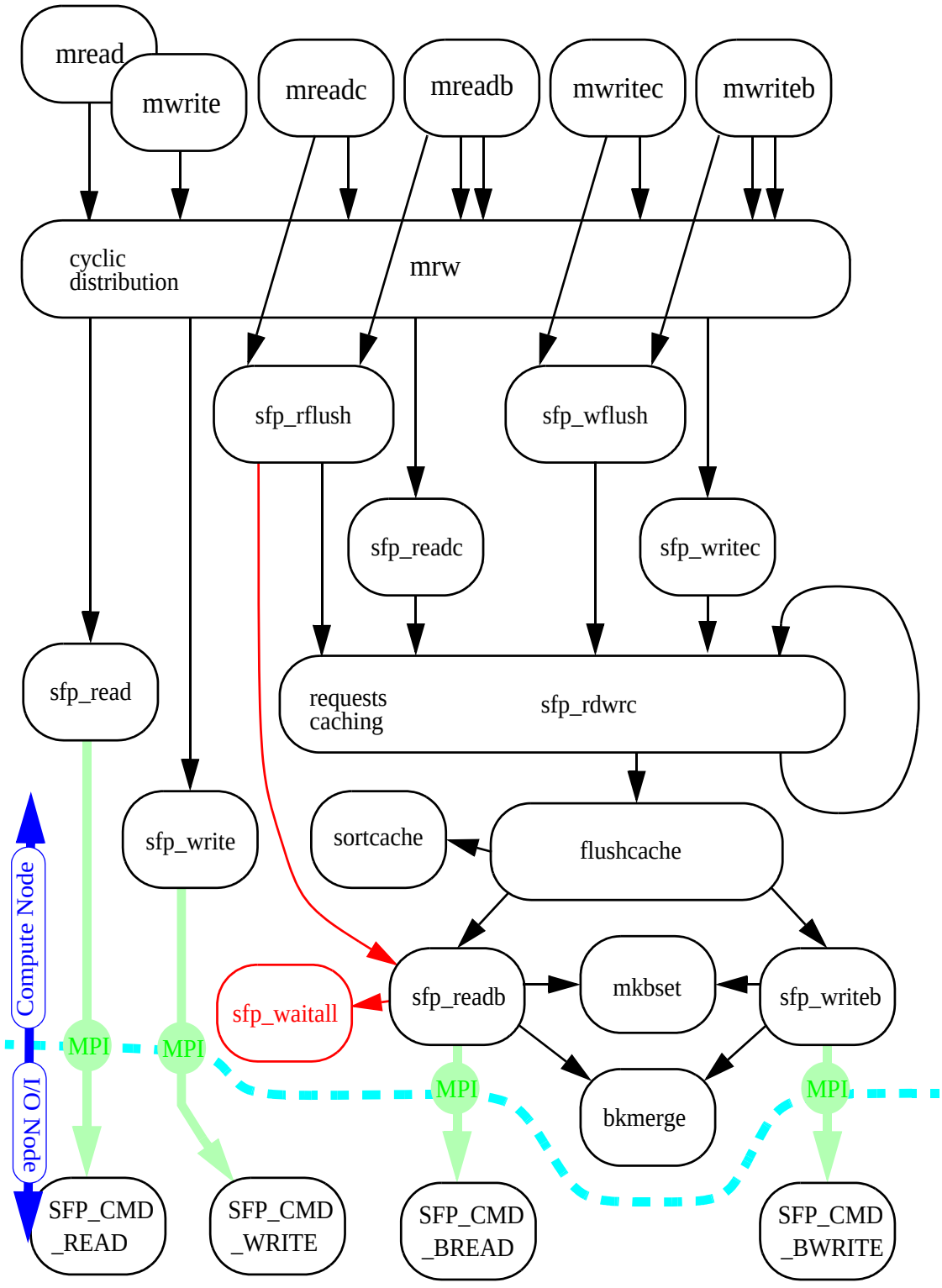


Performance improvement by asynchronous data reception optimization

Consecutive accesses to a SFIO 10MB/500B/6IO file on Swiss-T1Baby/TNet

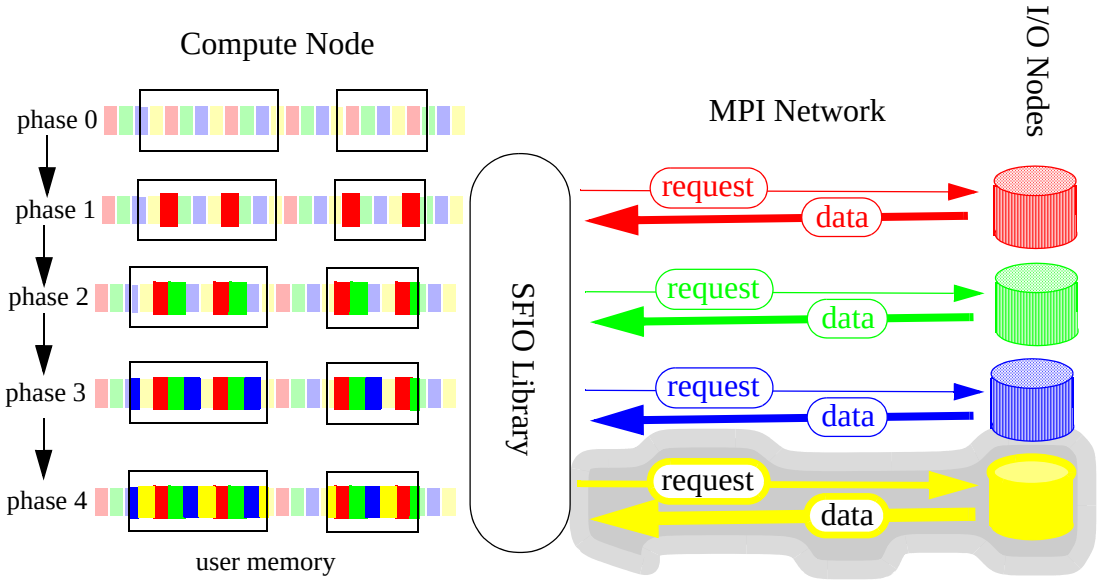


-

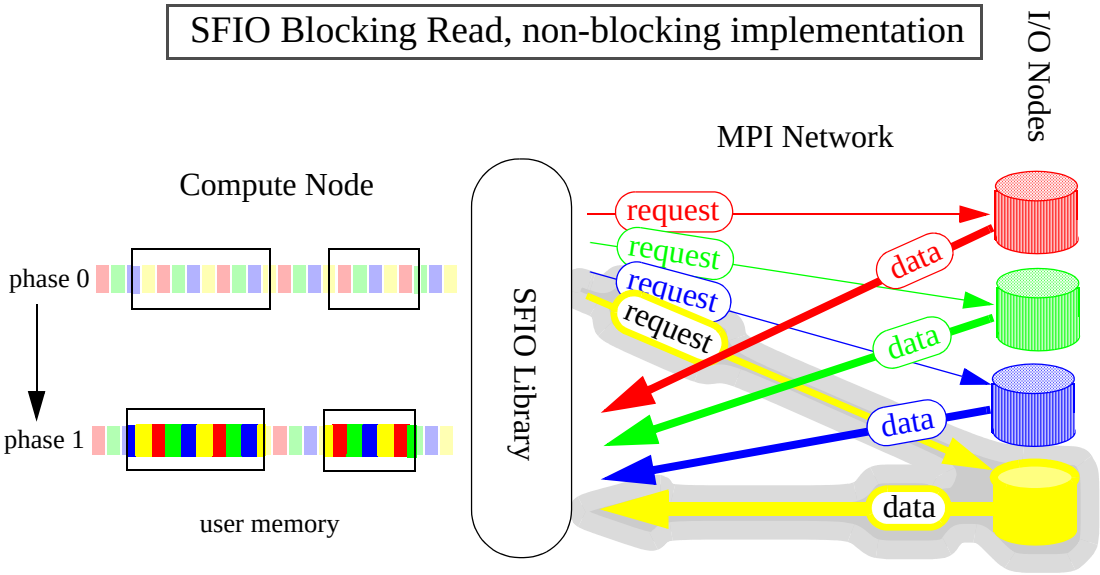

mread
mwrite
mreadc
mreadb
mwritec
mwriteb
cyclic
distribution
mrw
sfp_rflush
sfp_wflush
sfp_readc
sfp_writec
sfp_read
requests
caching
sfp_rdwrc
sfp_write
sortcache
flushcache
Compute Node
I/O Node
sfp_waitall
sfp_readb
mkbset
sfp_writeb
MPI
MPI
MPI
MPI
bkmerge
SFP_CMD
_READ
SFP_CMD
_WRITE
SFP_CMD
_BREAD
SFP_CMD
_BWRITE

-

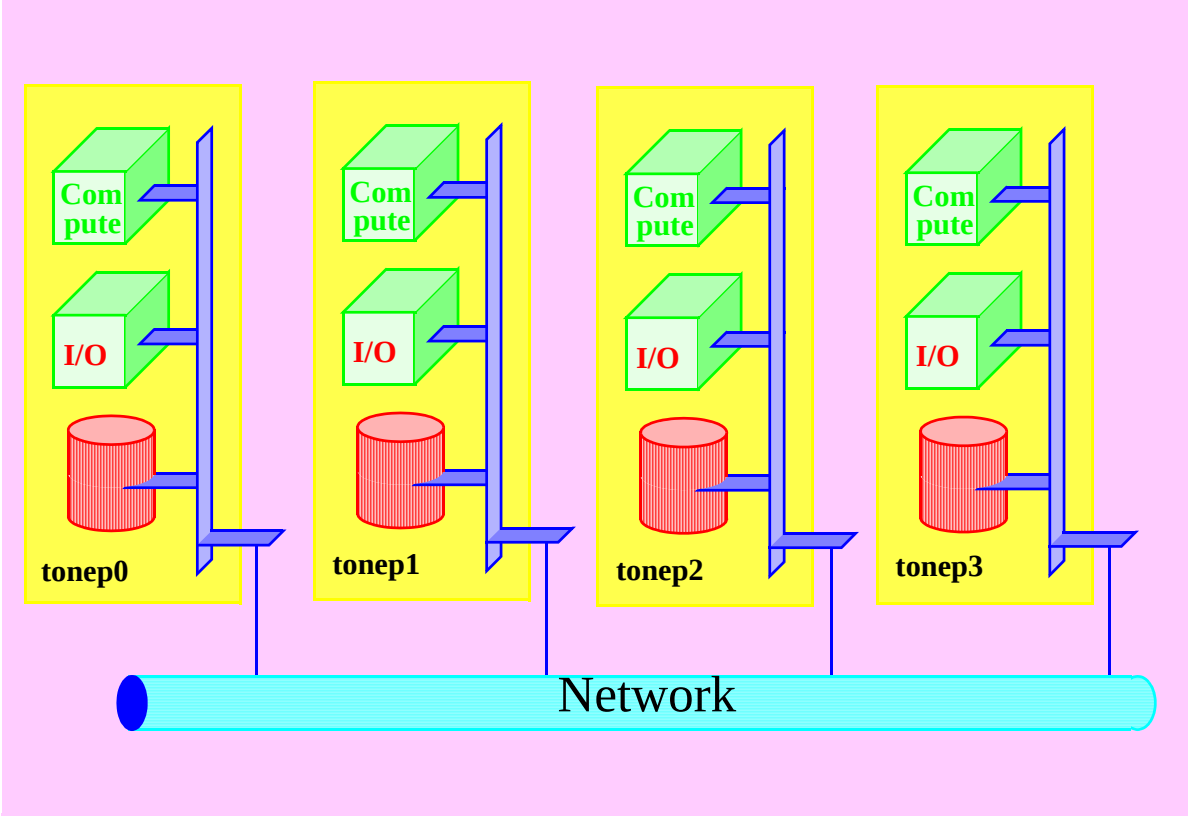
SFIO Blocking Read, blocking implementation



SFIO Blocking Read, non-blocking implementation



-

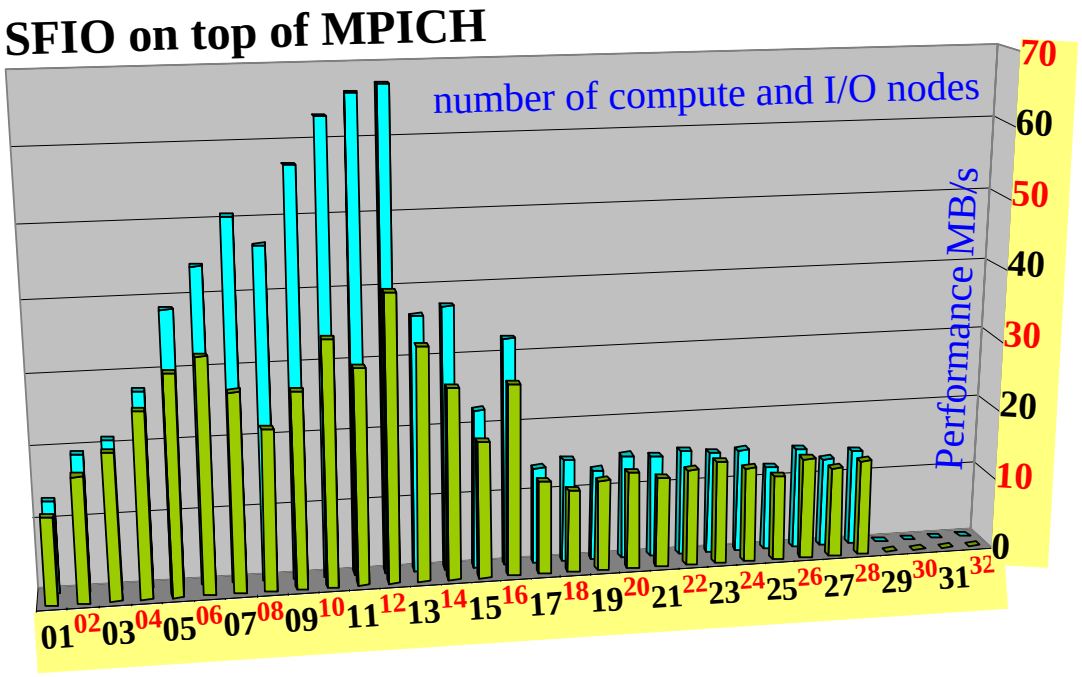


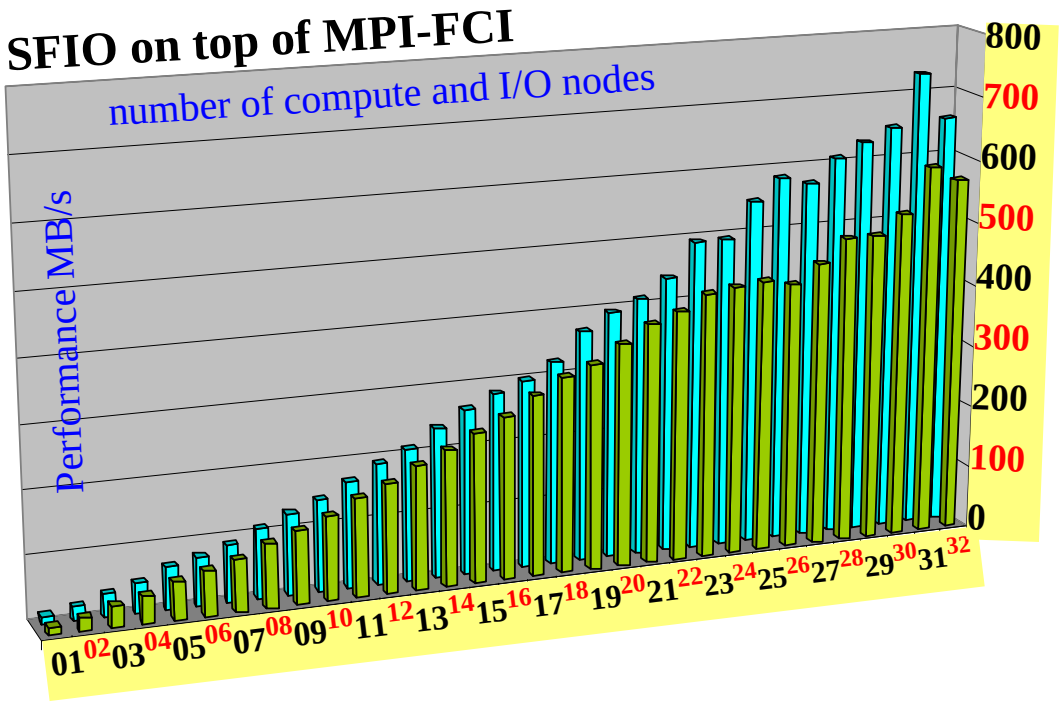
- SFIO All-to-All concurent write access from all compute nodes to all I/O nodes

- Global File size is 2000MByte

- Stripe unit size is 200Byte only

SFIO all-to-all I/O performance on Swiss-T1's Fast Ethernet and Tnet





- Superlinear speedup of SFIO/FCI due to augmentation of cache effect when increasing the number of I/O nodes.

Conclusion

- SFIO is portable, highly scalable, and ready for the distribution.

Future work

- SFIO performance benchmarking on the large supercomputer of Sandia National Laboratory.
- Adapt from T0 to T1 the modifications of MPICH/ADIO which provide a routing of a sub-set of MPI-I/O operations to the SFIO.
- Performance measurements of MPI-I/O inter-faced to SFIO through MPICH/ADIO.
- Possibly, creation of a portable MPI-I/O interface library to SFIO.
- Asynchronous implementation of blocking write operation. Pipelining on the I/O node.