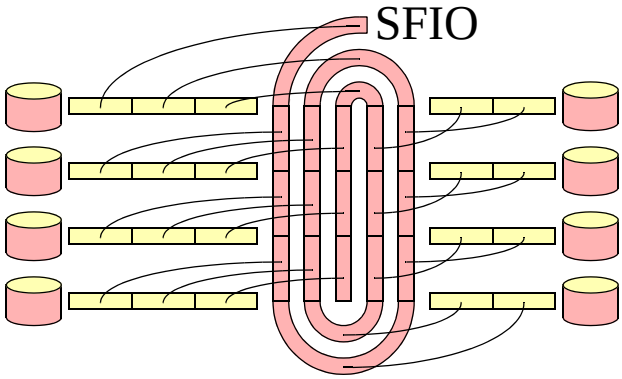
Eighteenth IEEE Symposium on Mass Storage Systems
held in cooperation with the Ninth NASA Goddard Conference on
Mass Storage Systems and Technologies
April 17-20, 2001
Hyatt Regency Islandia, San Diego, USA



Peripheral Systems Laboratory

SFIO a striped file I/O library for MPI

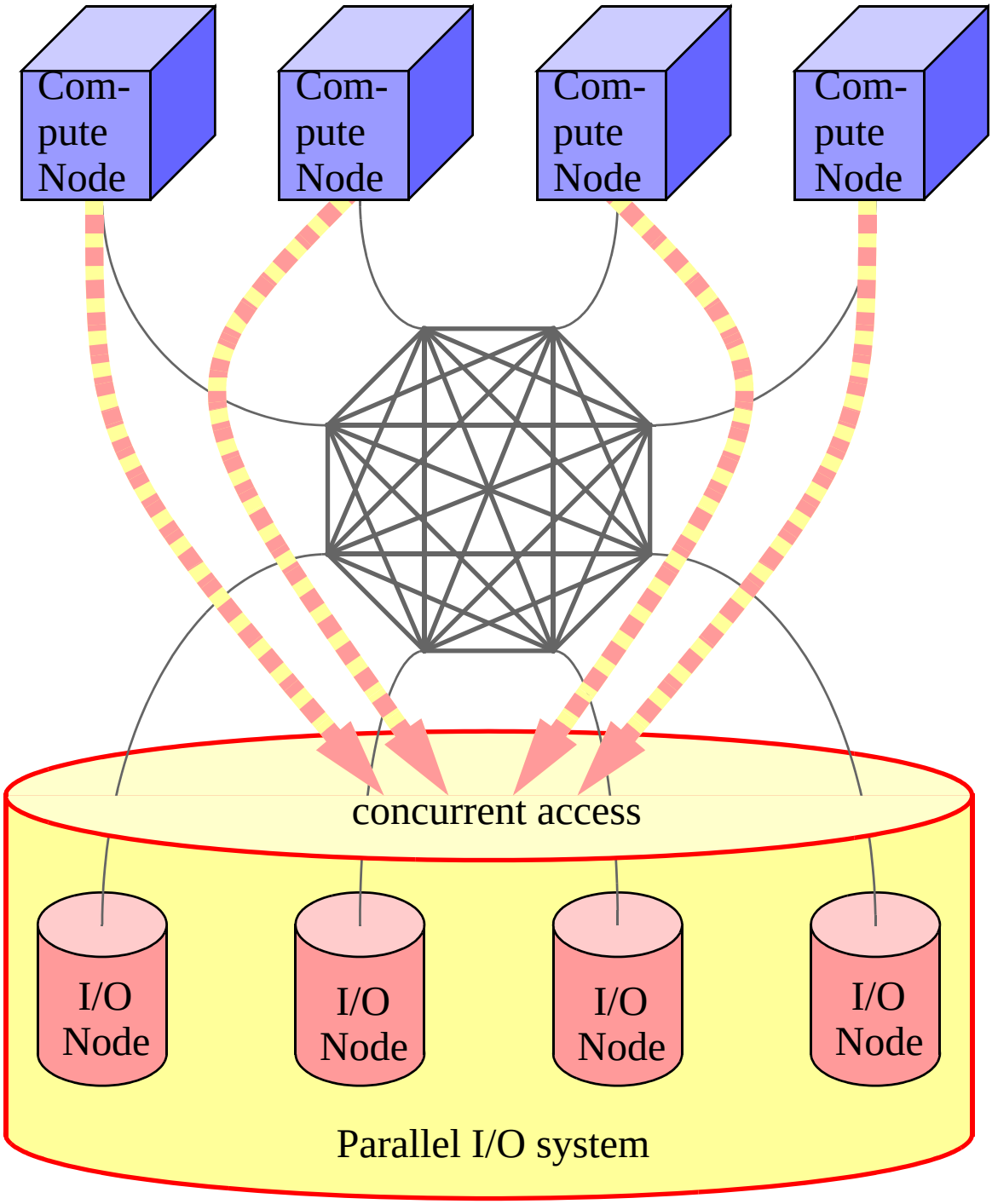
Emin Gabrielyan, Roger D. Hersch

École Polytechnique Fédérale de Lausanne, Switzerland

{Emin.Gabrielyan,RD.Hersch}@epfl.ch

.

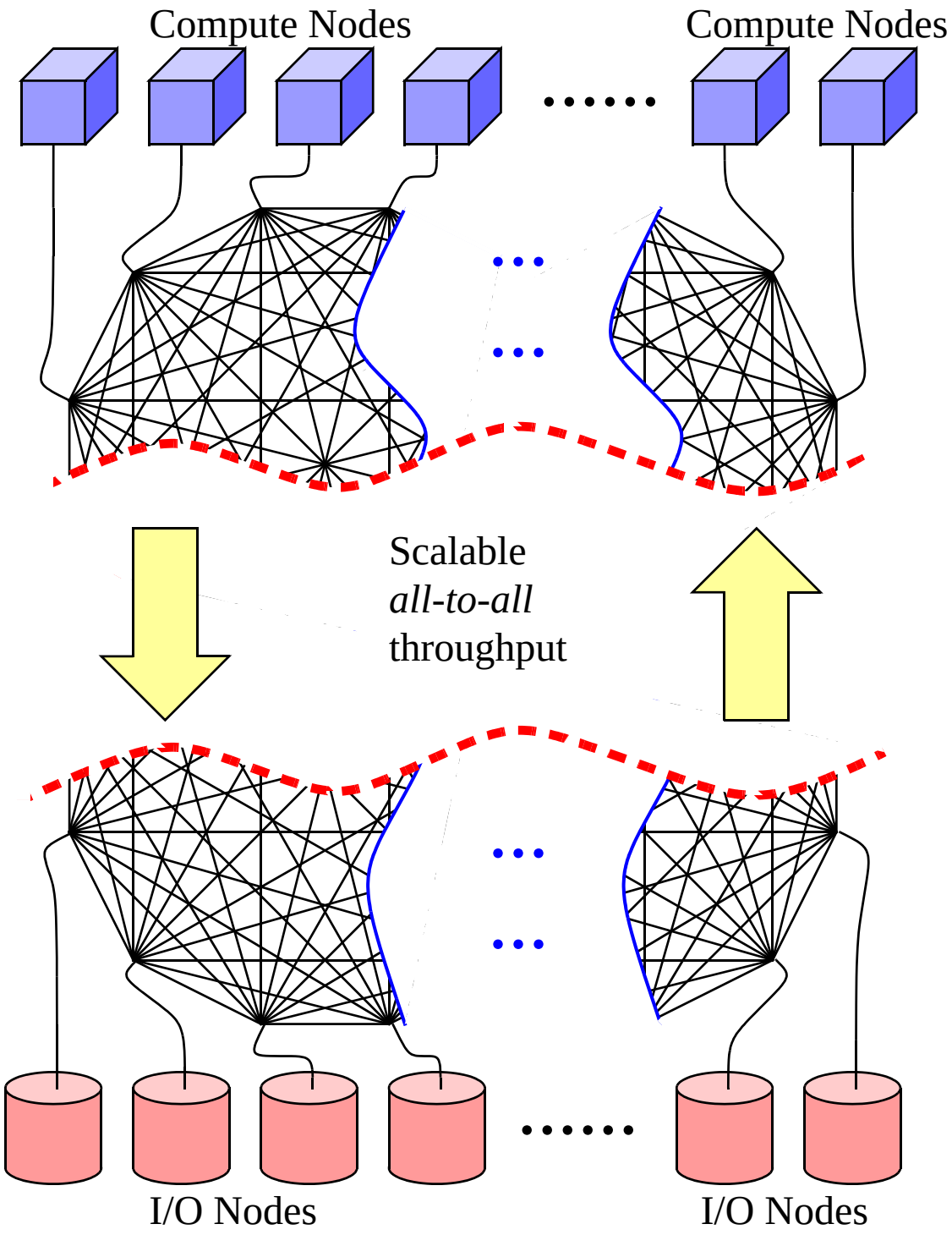
Concurrent Access



Parallel I/O systems should offer highly concurrent access capabilities to the common data files by all parallel application processes.

.

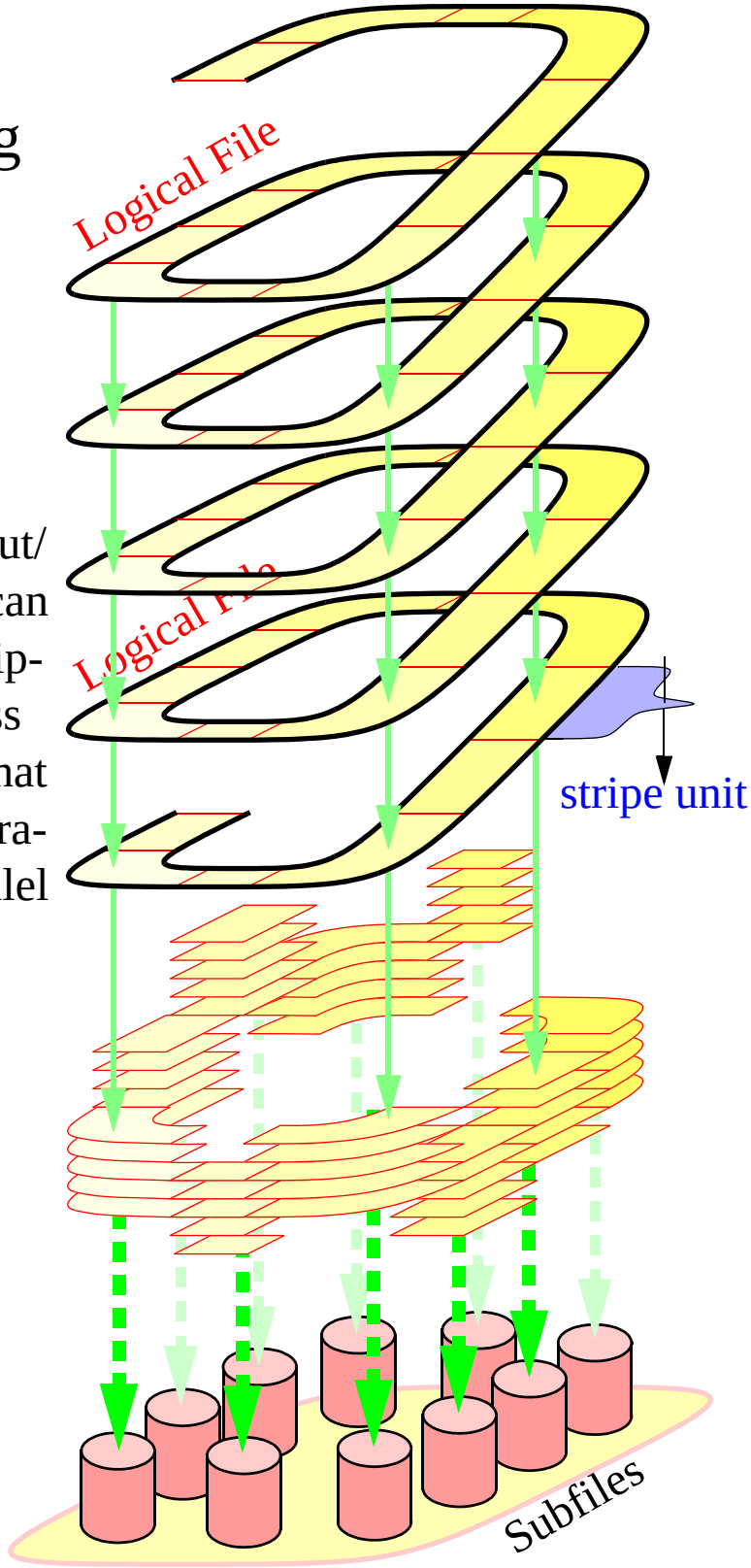
Scalability



Parallel I/O systems should exhibit linear increase in performance when increasing both the number of I/O nodes and the number of application's processing nodes.

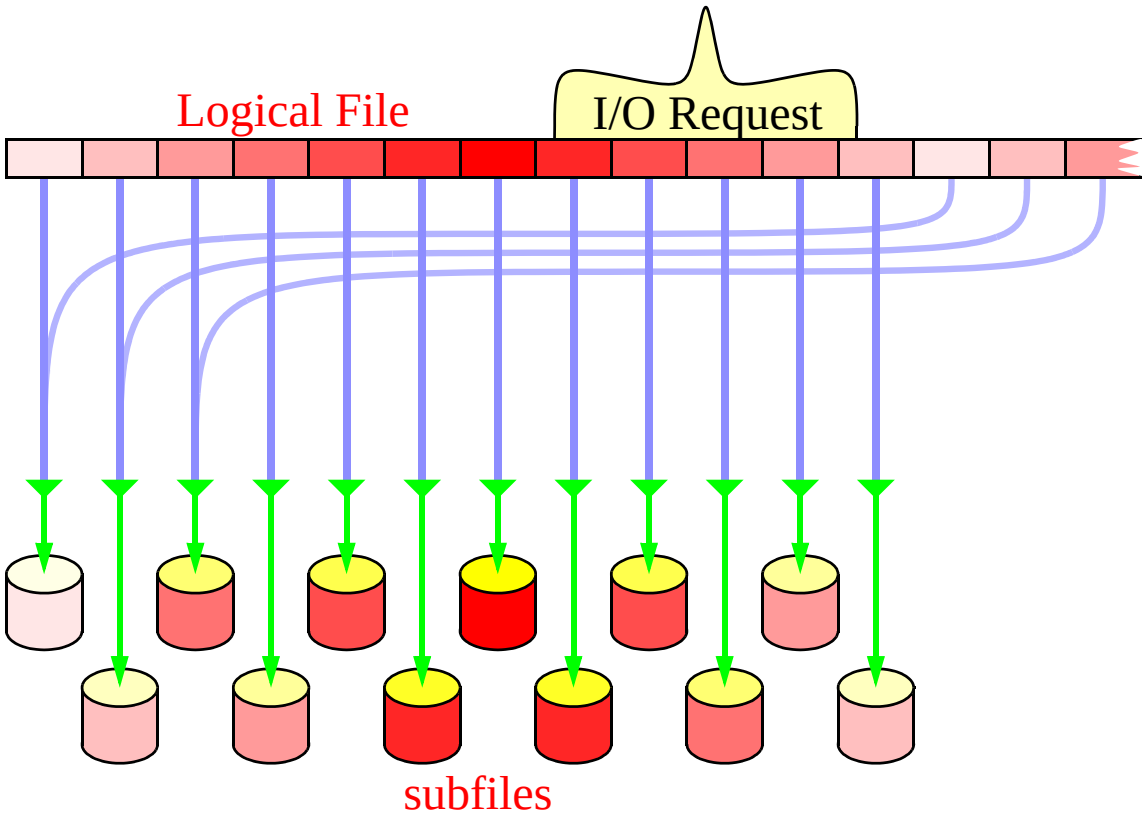
Parallel File Striping Paradigm

Parallelism for input/ output operations can be achieved by striping the data across multiple disks so that read and write operations occur in parallel



.

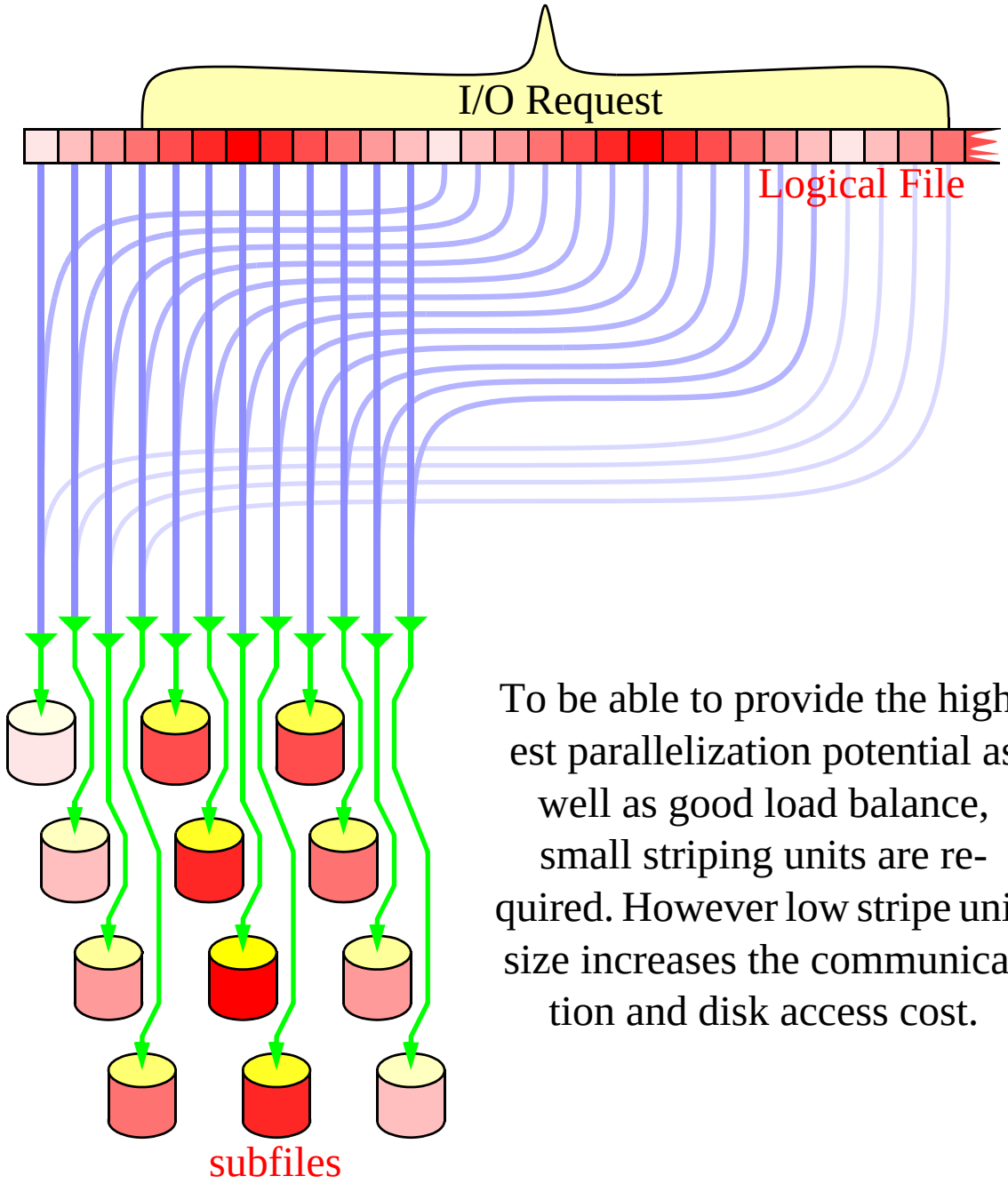
Stripe Unit Size



Large stripe unit size does not offer a high parallelization potential

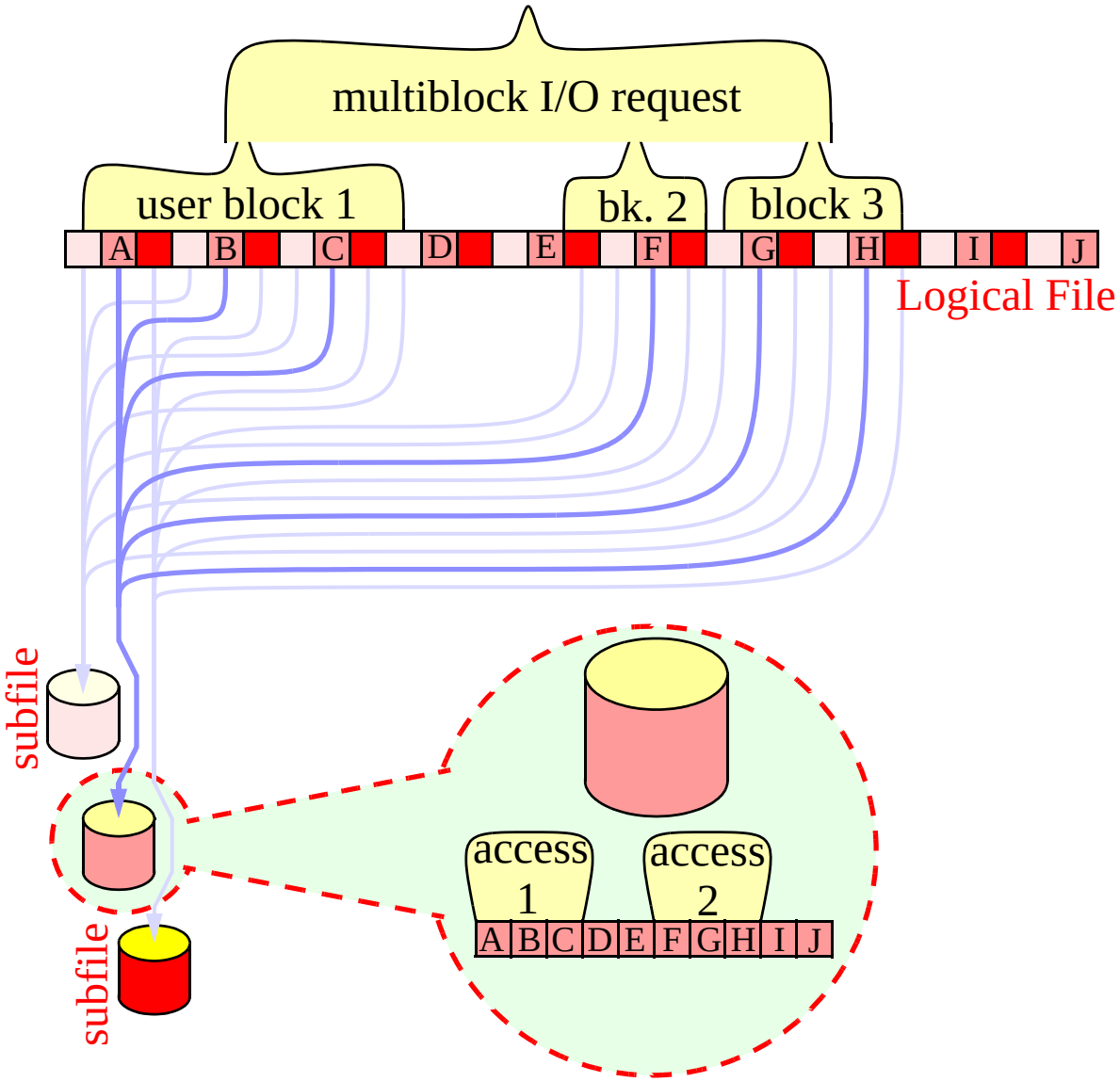
.

Small Striping Units



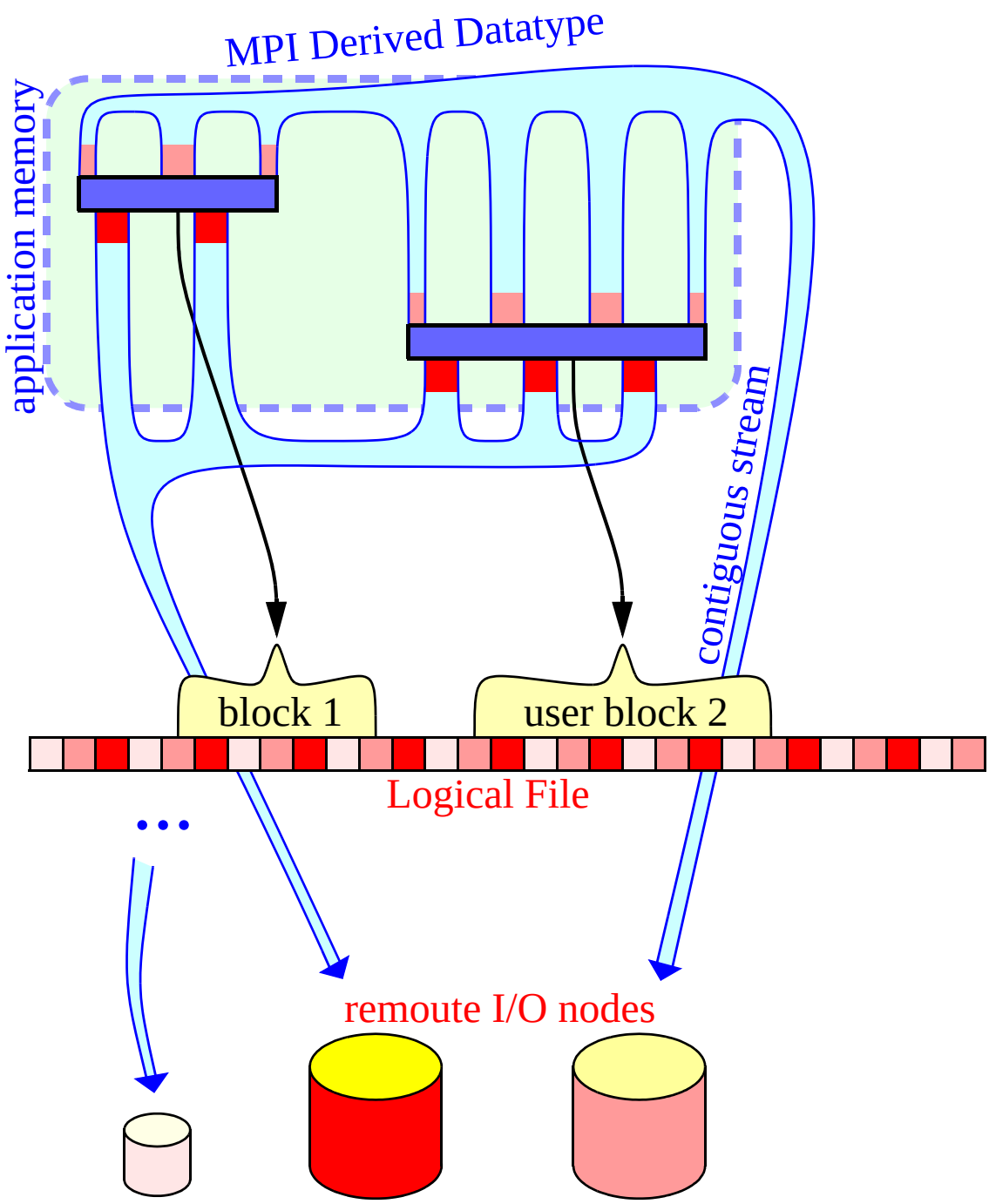
To be able to provide the highest parallelization potential as well as good load balance, small striping units are required. However low stripe unit size increases the communication and disk access cost.

Disk access optimization



SFIO merges several disk accesses into a single disk access request. The algorithm implemented on the compute node tries to combine all overlapping or consecutive I/O requests. The 6 original data segments to be accessed are grouped into 2 remote subfile requests.

Network Communication Optimization

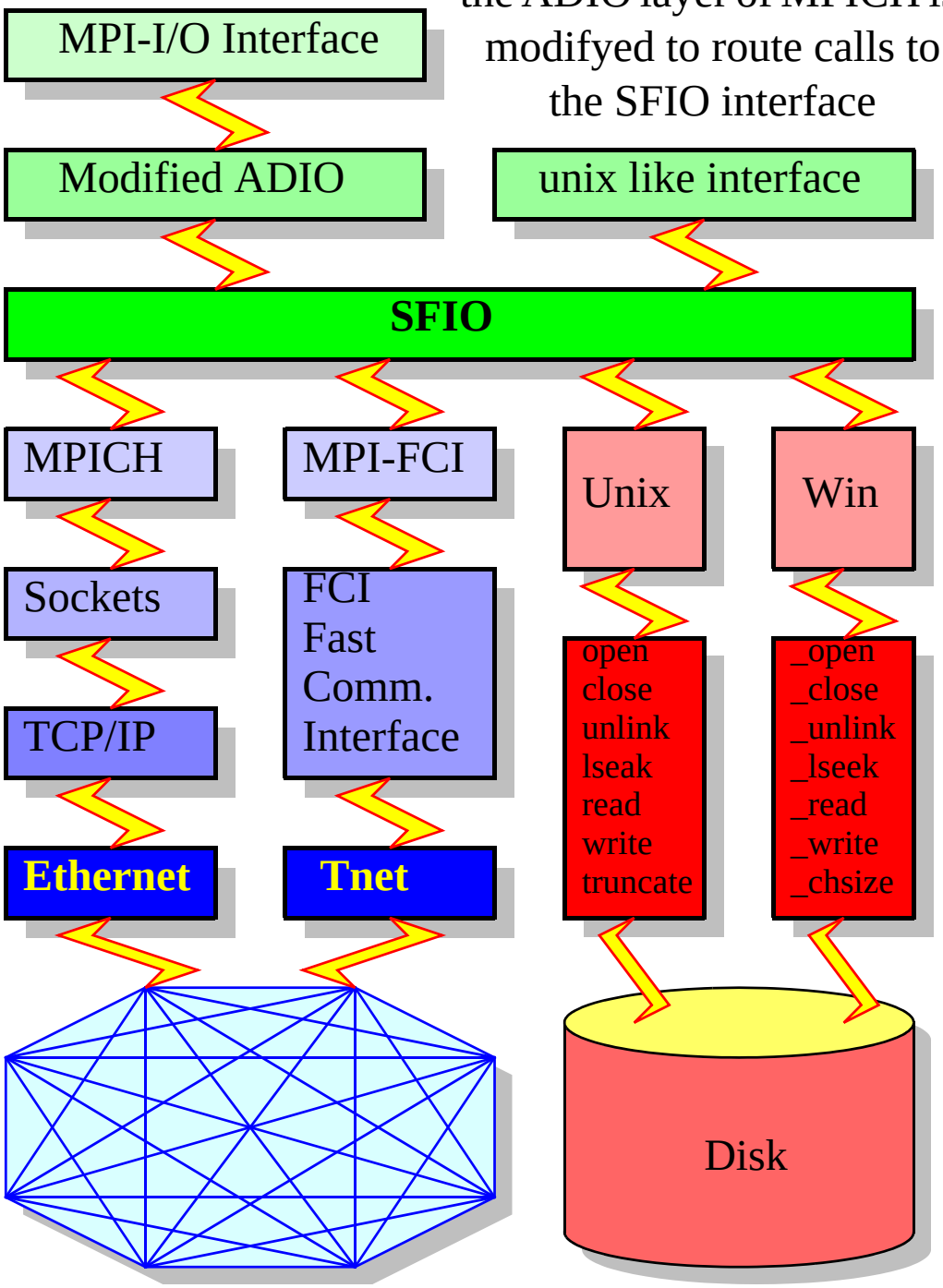


Low stripe unit size increases communication cost. SFIO integrates the relevant optimizations. It creates dynamically a derived datatype and transmits highly fragmented data as a single stream without additional copy.

.

MPI-I/O on top of SFIO

For few MPI-I/O operations, the ADIO layer of MPICH is modifyed to route calls to the SFIO interface

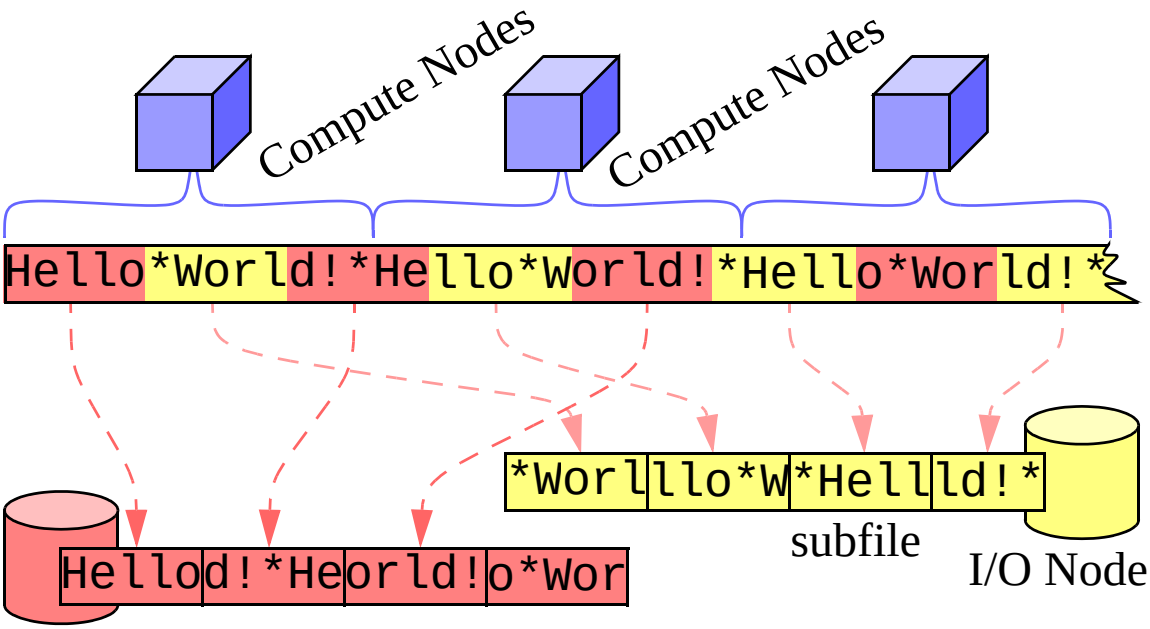


The SFIO library is implemented using MPI-1.2.
It is therefore as portable as MPI-1.2.

.

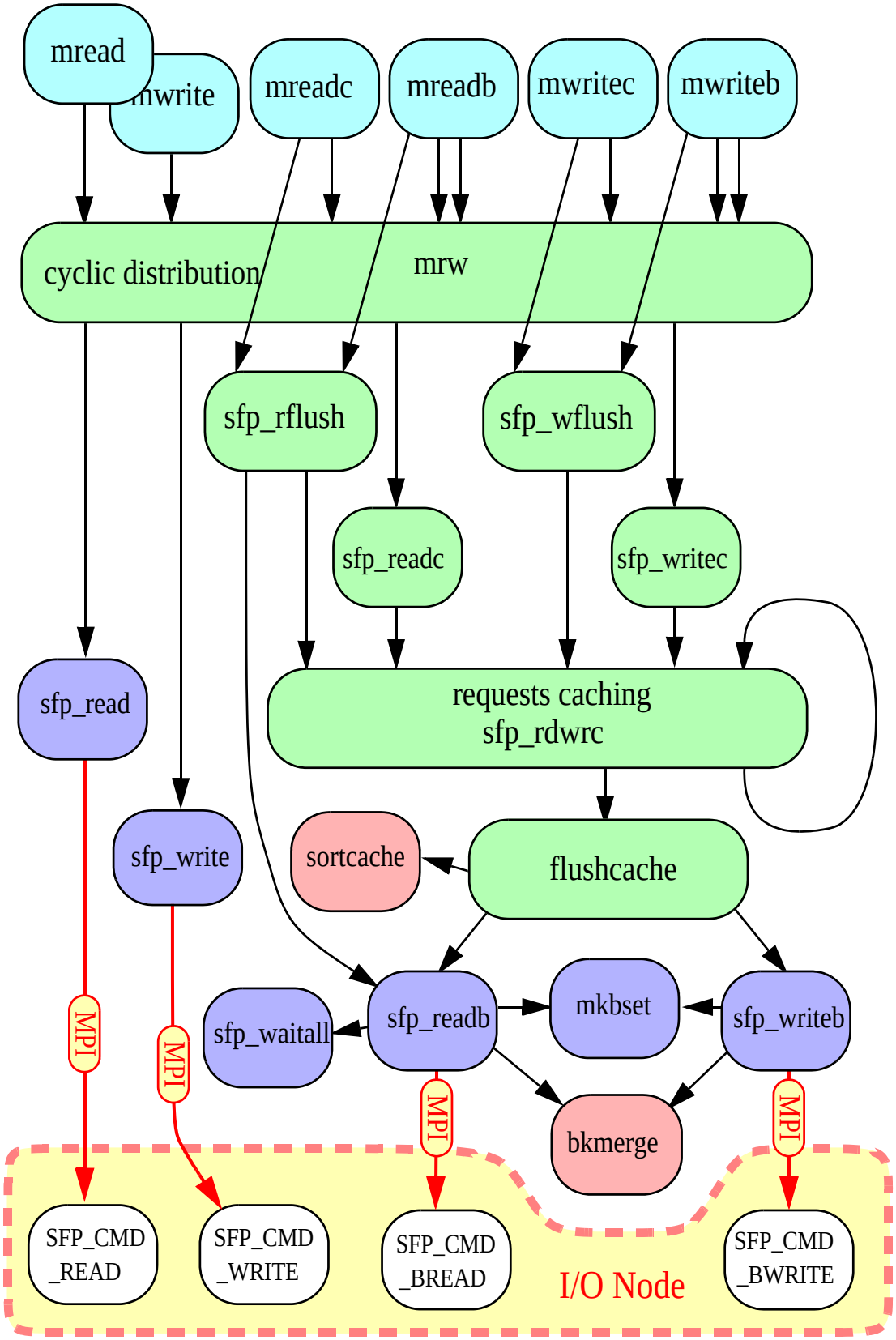
SFIO interface

```
#include <mpi.h>
#include "/usr/local/sfio/mio.h"
int _main(int argc, char *argv[])
{
 MFILE *f;
 char bu[]="Hello*World!*";
 int r=rank();
 f=mopen("p1,/tmp/a;p2,/tmp/a;",5);
 mwritec(f,13*r,bu,13);
 mclose(f);
}
```



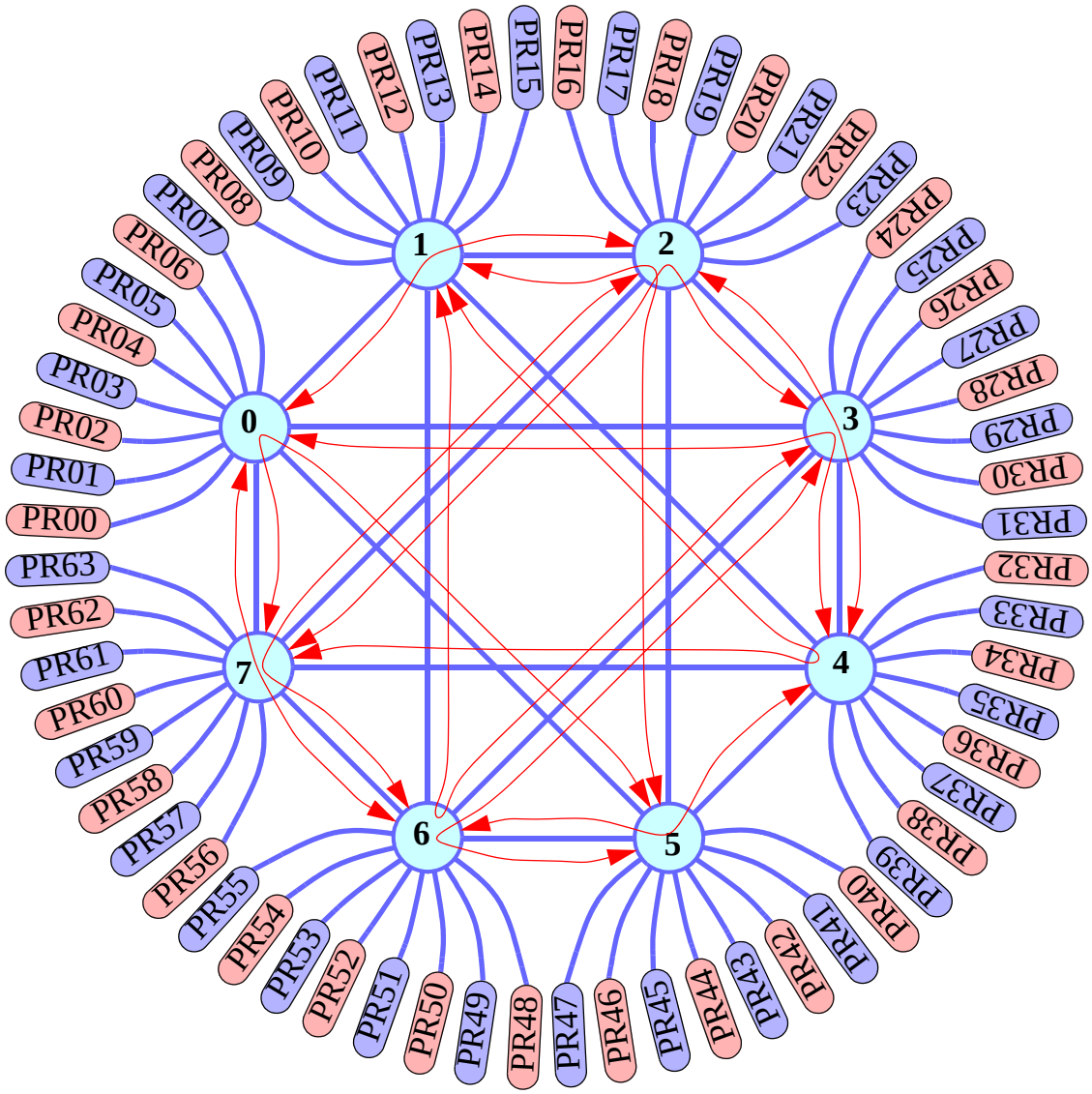
Multiple compute nodes accessing a striped file. The striped file with a stripe unit size of 5 bytes consists of two subfiles.

Functional Architecture



.

Swiss-T1 Topology



0 Switch

PR00 IO Processor

PR01 Compute Processor

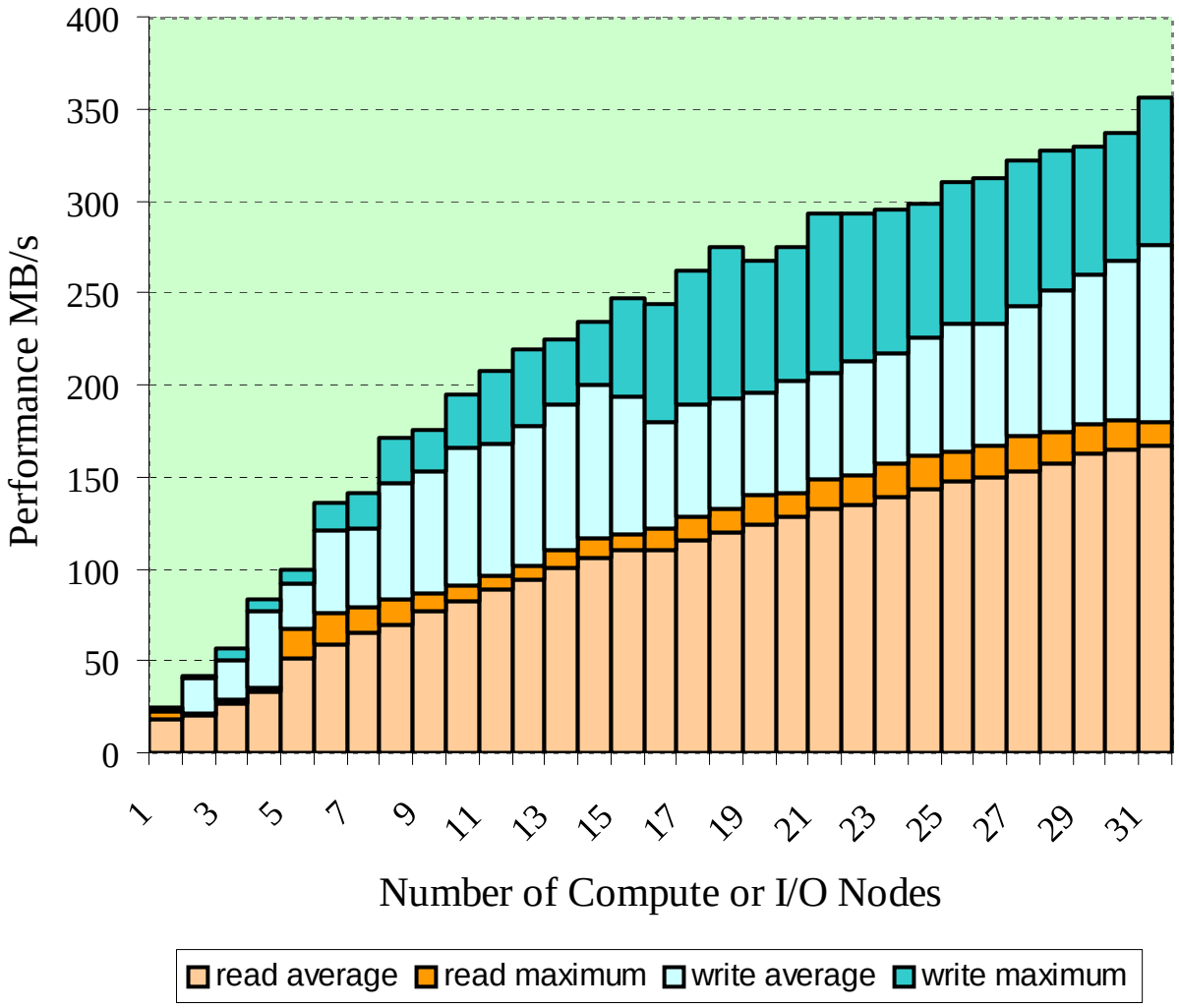
TNET connection

Routing

Performance results have been measured on the Swiss-T1 machine. The Swiss-T1's TNET Network consists of eight 12-port full crossbar switches. Routing between switches is static. Throughput of TNET link is ~86MB/s

.

SFIO on the Swiss-Tx machine



The performance of SFIO over MPI-FCI is measured for concurrent access from all compute nodes to all I/O nodes. In order to limit operating system caching effects, the total size of the striped file linearly increases with the number of I/O nodes up to 32GB. The stripe unit size is 200 bytes. The MPI-FCI application's I/O performance is measured as a function of the number of Compute and I/O nodes. For each configuration, 53 measurements are carried out.

.