Parallel I/O Middleware

Vendel Szeremi
STFC Daresbury Laboratory, UK

ARCHER / PRACE Training 2-3 Sept 14

Middleware

- ADIOS - Adaptable IO System

  Oak Ridge National Laboratory

  https://www.olcf.ornl.gov/center-projects/adios/

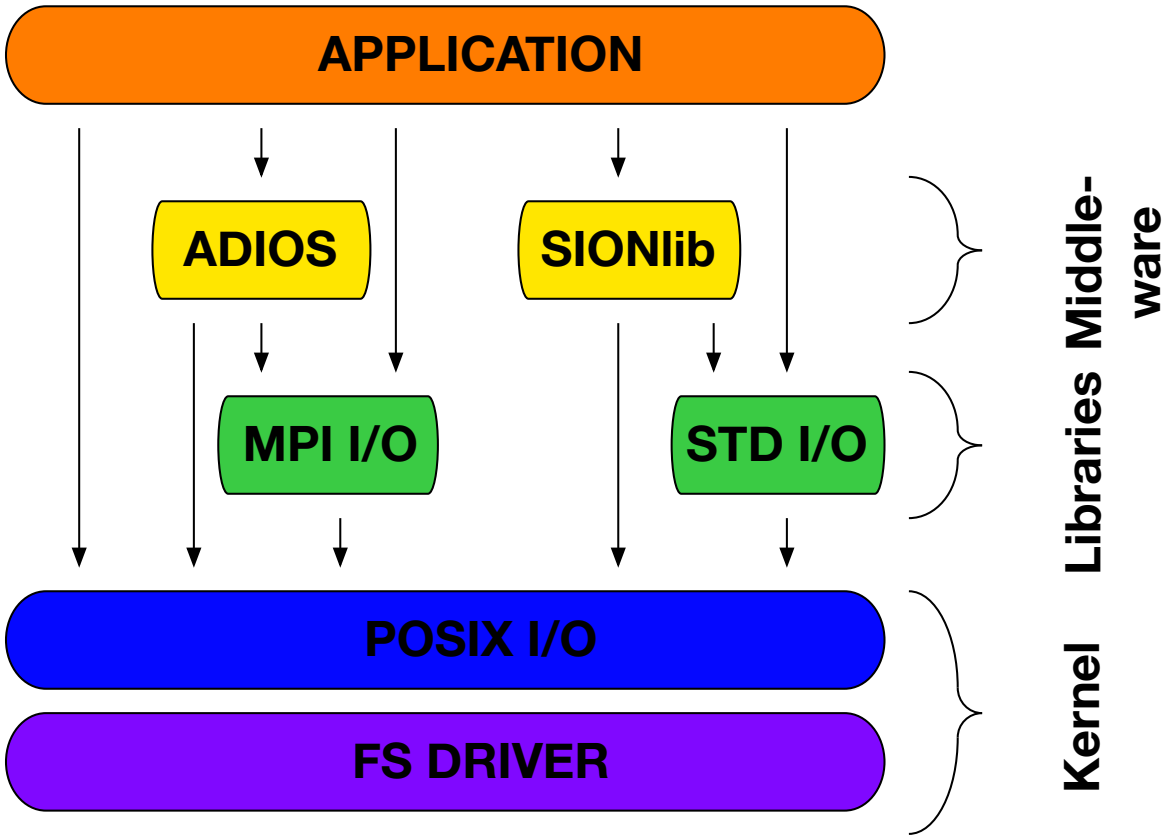
- SIONLib

  Forschungszentrum Jülich

  http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/SIONlib/_node.html

Science & Technology Facilities Council

I/O Software Stack



ADIOS

- C and Fortran APIs
- XML configuration files
- range of transport methods
- data transformations, e.g. compression
- BP file format, utilities
- takes care of endian conversion for reading

ADIOS

- XML API
- No-XML Write API
- Group concept: set of variables
- Each group can have a different transport method which is best suited for I/O pattern

ADIOS

Mainline Transport Methods:

- NULL
- POSIX
- MPI / MPI_LUSTRE / MPI_AGGREGATE
- VAR_MERGE
- Dataspaces / DIMES
- Flexpath
- PHDF5 / NC4 (NetCFD 4)

ADIOS XML CONFIG FILE

```
<adios-config>
  <adios-group>
      <var ... />
      <attribute .../>
  </adios-group>
  <method ... />
  <buffer ... />
</adios-config>
```

ADIOS XML CONFIG FILE

```
<?xml version="1.0"?>
<adios-config host-language="C”>
    <adios-group name="u_array">
        <var name="NX" type="integer"/> <var name="NY" type="integer"/>
        <var name="NZ" type="integer"/>
        <var name="u" gwrite="u" type="double" dimensions="NX,NY,NZ"/>
    </adios-group>
    <method group="u_array" method="POSIX"/>
<!--' <method group="u_array" method="MPI"/> '-->
<!--' <method group="u_array" method="PHDF5"/> ’→
    <buffer size-MB="20" allocate-time="now"/>
</adios-config>
```

ADIOS

```
adios_init("homb_c.xml", comm);
adios_open(&adios_handle, "u_array",
        fname_shared, "w", comm);

#include "gwrite_u_array.ch”

adios_close(adios_handle);
adios_finalize(rank);
```

SIONlib

- single shared file or multiple shared files
- intended for “temporary” I/O
- alignment to filesystem block boundaries
- padding to avoid contention

SIONlib

```
/* Open */
fh = fopen(fname, "bw", ...);

/* Write */
 fwrite(bindata, 1, nbytes, fh);

/* Close */
 fclose(fh);
```

SIONlib

```
/* Collective Open */
 sion_handle = sion_paropen_mpi(fname_shared, "w",
    &sion_numFiles, sion_gComm, &sion_lComm,
    &sion_chunksize, &sion_fsblksize,
    &sion_globalrank, &fh, &sion_new_fname);
/* Write */
fwrite(bindata, 1, nbytes, fh);
 /* Collective Close */
 sion_parclose_mpi(sid);
```

SIONlib

```
int sion_paropen_mpi(
    char *fname, const char *file_mode,
    int *numFiles,
    MPI_Comm gComm, MPI_Comm *lComm,
    ion_int64 *chunksize, sion_int32 *fsblksize,
    int *globalrank,
    FILE **fileptr, char **newfname
);
```