

An MPI-IO Compliant Java based Parallel I/O Library

Ammar Ahmad Awan, Muhammad Bilal Amin, Shujaat Hussain, Aamir Shafi, Sungyoung Lee

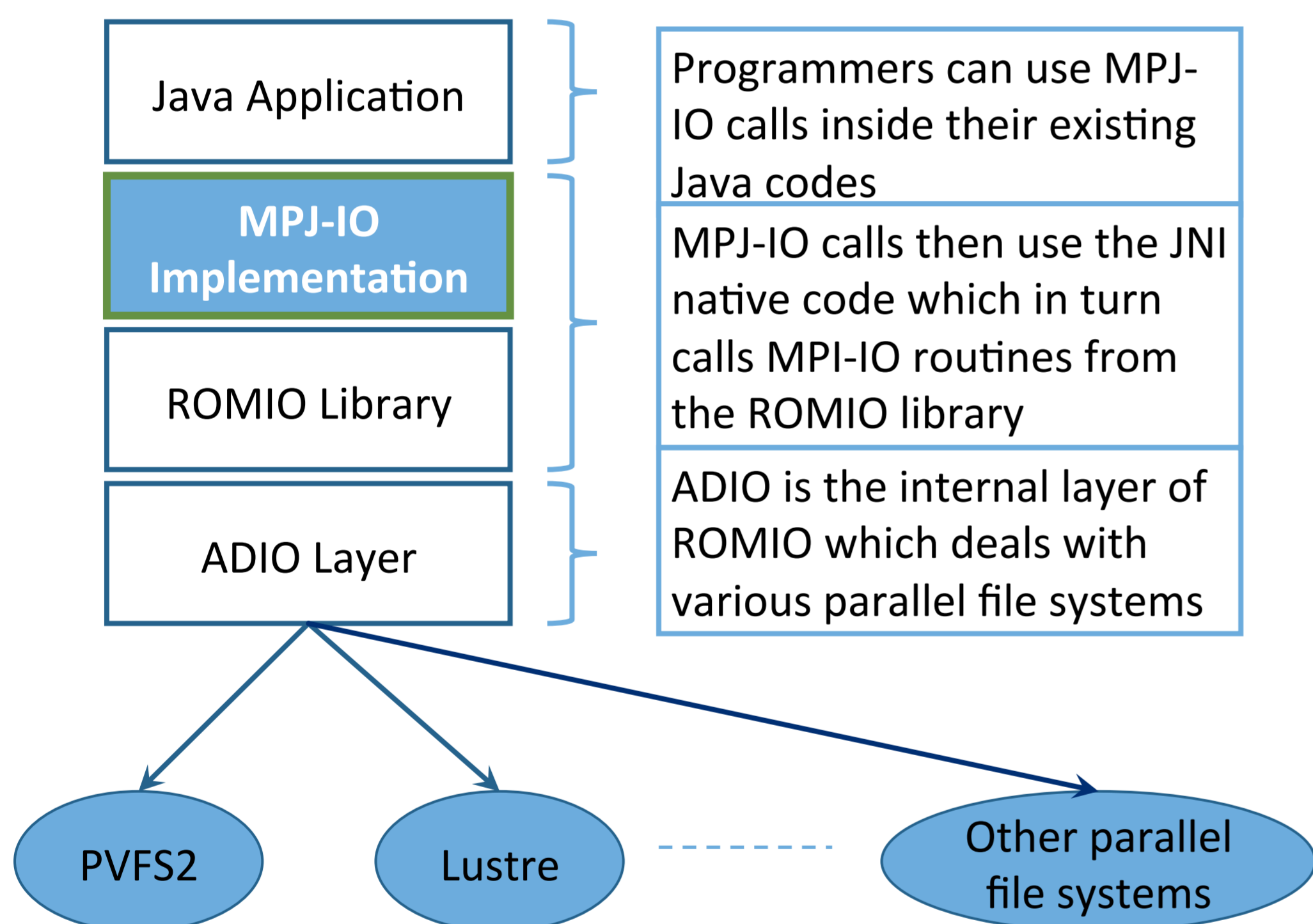
1. Motivation

- Lack of support for parallel file access in Java HPC libraries.
- No bindings for Java language in the MPI-I/O API Specifications.
- Existing Java I/O libraries are prototype versions and most of them are not available for download.

2. Goal

To enable Java HPC application programmers use the parallel file access interface in their existing as well as new applications without writing any explicit file handling code.

3. Implementation Stack



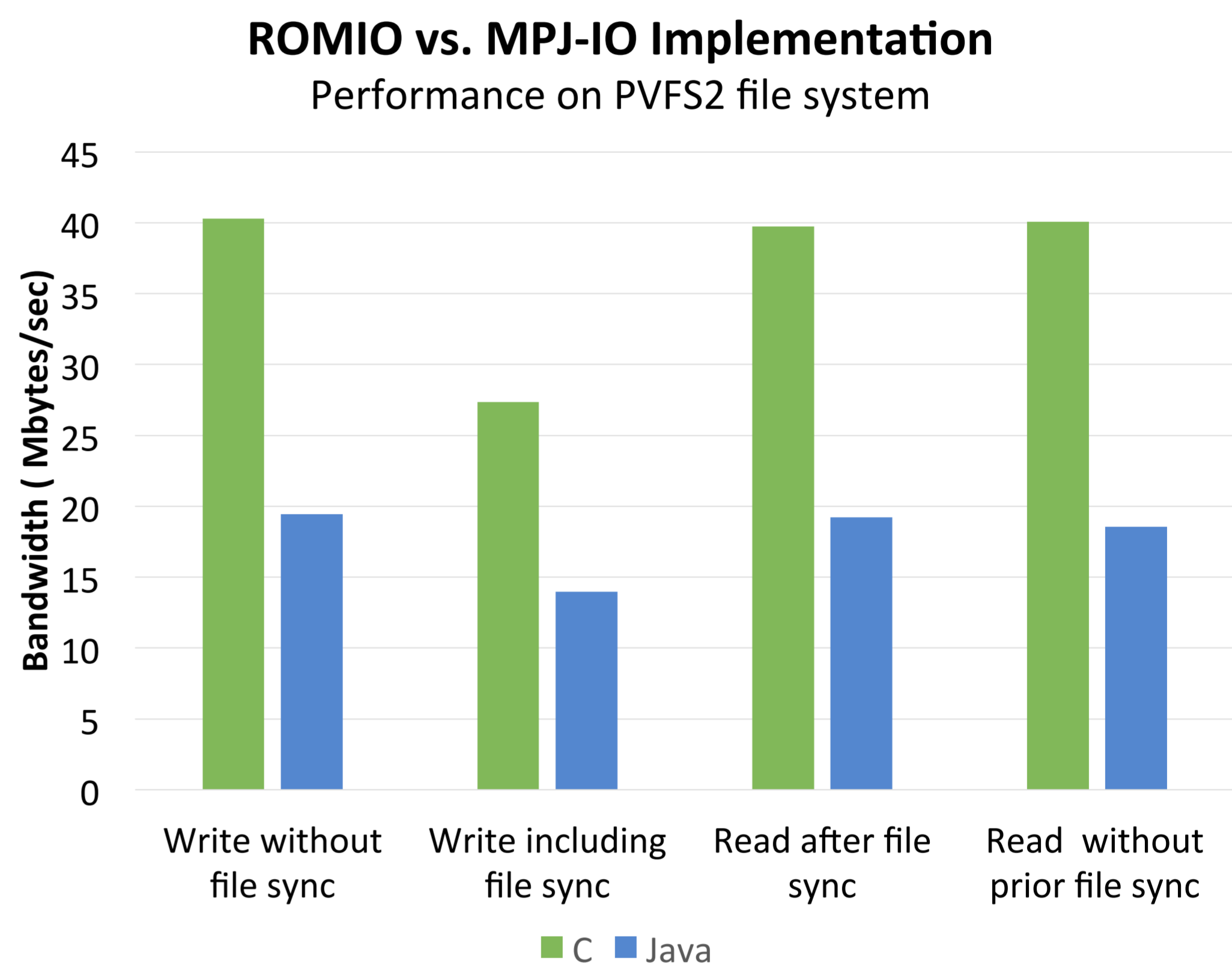
4. Uniqueness

- MPI-IO compliant API.
- Builds on top of widely accepted mpiJava software.
- Modular design to allow integration with existing MPI-like Java libraries.
- Support for Native networking hardware in the JNI wrapper version.
- Compatible with Parallel File Systems supported by ROMIO library.

5. Methodology

- Adapted the MPI-IO API for Java language and modeled it inside the mpiJava 1.2 API - (MPJ-IO API)
- Implemented the proposed MPJ-IO API
 1. Pure Java approach
 2. Java Native Interface (JNI) wrapper approach

6. Performance Evaluation



7. Proposed MPJ-IO API

```
File Intracomm.fileOpen(String filename, int amode,
                        Info info)
void Intracomm.fileClose(File file)

void File.delete(string filename, Info info)

void File.seek(Offset offset, int whence)

void File.sync()

void File.setAtomicity(boolean flag)

boolean File.getAtomicity()

Status File.readAt(Offset offset, Object buf,
                  int bufOffset, int count, Datatype datatype)

Status File.writeAt(Offset offset, Object buf,
                   int bufOffset, int count, Datatype datatype)
```

The MPJ-IO API is available from:

<http://hpc.seecs.nust.edu.pk/~ammar/images/draft.pdf>

