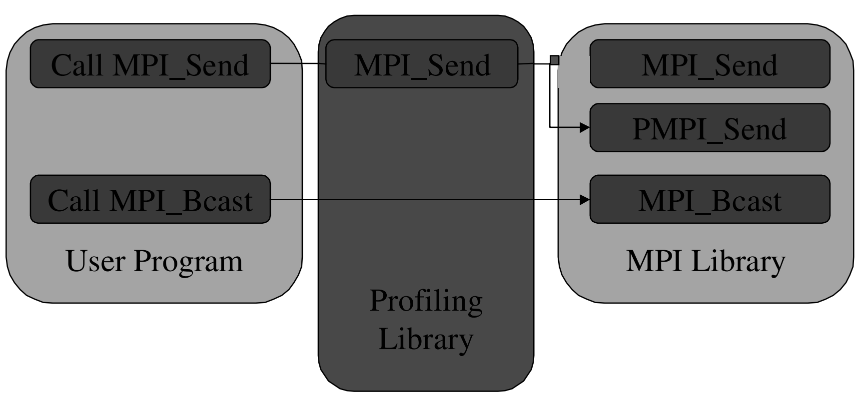
Using PMPI routines

- PMPI allows selective replacement of MPI routines at link time (no need to recompile)
- Some libraries already make use of PMPI
- Some MPI implementations have PMPI bugs
  - » IBM — PMPI_Wtime() returns 0
  - » Cray — PMPI in a separate library that some installations have not installed

1

Profiling Interface



2

Example Use of Profiling Interface

```
static int nsend = 0;

int MPI_Send( void *start, int count,
  MPI_Datatype datatype, int dest,
  int tag, MPI_Comm comm )
{
    nsend++;
    return PMPI_send(start, count, datatype,
                     dest, tag, comm);
}
```


Using the Profiling Interface from Fortran

```
Block data
common /mycounters/ nsend
data nsend/0/
end

subroutine MPI_Send( start, count, datatype, dest,&
                     tag, comm, ierr )
integer start(*), count, datatype, dest, tag, comm
common /mycounters/ nsend
save   /mycounters/
nsend = nsend + 1
call PMPI_send(start, count, datatype, &
                   dest, tag, comm, ierr )
end
```

Exercise: Find Unsafe Uses of MPI_Send

- Assume that you have a debugger that will tell you where a program is stopped (most will). How can you find unsafe uses of MPI_Send (calls that assume that data will be buffered) by running the program *without* making assumptions about the amount of buffering
  - » Hint: Use MPI_Ssend

Finding Unsafe uses of MPI_Send

```
subroutine MPI_Send( start, count, datatype, dest,
                     tag, comm, ierr )
integer start(*), count, datatype, dest, tag, comm
call PMPI_Ssend(start, count, datatype,
                     dest, tag, comm, ierr )
end
```

- MPI_Ssend will not complete until the matching receive starts
- MPI_Send can be implemented as MPI_Ssend
- At some value of *count*, MPI_Send will act like MPI_Ssend (or fail)

Finding Unsafe MPI_Send II

- Have the application generate a message about unsafe uses of MPI_Send
  - » Hint: use MPI_Issend
  - » C users can use __FILE__ and __LINE__
    - – sometimes possible in Fortran (.F files)

Reporting on Unsafe MPI_Send

```
subroutine MPI_Send( start, count, datatype, dest, tag, comm,
                     ierr )
integer start(*), count, datatype, dest, tag, comm
include 'mpif.h'
integer request,  status(MPI_STATUS_SIZE)
double precision t1, t2
logical flag

call PMPI_Issend(start, count, datatype, dest, tag, comm,
                 request, ierr )
flag = .false.
t1   = MPI_Wtime()
Do while (.not. flag .and. t1 + 10 .gt. MPI_Wtime())
    call PMPI_Test( request, flag, status, ierr )
Enddo
if (.not. Flag) then
    print *, 'MPI_Send appears to be hanging'
    call MPI_Abort( MPI_COMM_WORLD, 1, ierr )
endif
end
```

Logging and Visualization Tools

- Upshot and MPE tools
  http://www.mcs.anl.gov/mpi/mpich
- Pallas VAMPIR
  http://www.pallas.com/
- IBM VT
  http://www.rs6000/ibm.com/software/sp_products/pe.html
- Pablo http://www-pablo.cs.uiuc.edu/Projects/Pablo/pablo.html
- Paragraph
  http://www.ncsa.uiuc.edu/Apps/MCS/ParaGraph/ParaGraph.html
- Paradyn http://www.cs.wisc.edu/~paradyn
- Other vendor tools

5