

mpipeline

A Parallelized Unix Pipe for Serial Filter Programs

Nathan Weeks <weeks@iastate.edu>

Pipes

- Task: align DNA sequences in two files to the soybean genome. Keep alignments >100 in length.
- Example without pipes:

```
$ ls
```

```
DNA1.fasta.gz DNA2.fasta.gz
```

```
$ gunzip DNA?.fasta.gz
```

```
$ blastn -infile DNA1.fasta -db Glycine_max > temp1.txt
```

```
$ blastn -infile DNA2.fasta -db Glycine_max > temp2.txt
```

```
$ awk '$5 - $4 > 100' temp1.txt temp2.txt > output.txt
```

- Note the use of temporary files

Pipes

- Pipe: a one-way communication channel between two processes on the same host
- Previous example with pipes

```
$ gzcat DNA?.fasta.gz |  
    blastn -db Glycine_max |  
        awk '$5 - $4 > 100' > finaloutput.txt
```

- Note the lack of temporary intermediate files

Problem

- Task: parallelize the previous example to run multiple simultaneous instances of blastn on 32 nodes
- Data distribution
 - Typical solution: split the two files into 32 files (and hope you don't exceed your quota or available disk space) and submit 32 blastn jobs
- Load Balancing
 - What if the time to process one of the 32 input files is much greater than the others (e.g., slow cluster node, longer sequences, or repetitive sequence?)

Solution

```
gzcat DNA?.fasta.gz |  
    mpirun -np 33 \  
    mpipipe -n 2000 'blastn -db Glycine_max' |  
        awk '$5 - $4 > 100' > output.txt
```

Solution

```
gzcat DNA?.fasta.gz |  
    mpirun -np 33 \  
    mpipipe -n 2000 'blastn -db Glycine_max' |  
        awk '$5 - $4 > 100' > output.txt
```

Uncompress & concatenate input files

Solution

```
gzcat DNA?.fasta.gz |  
    mpirun -np 33 \  
    mpipipe -n 2000 'blastn -db Glycine_max' |  
        awk '$5 - $4 > 100' > output.txt
```

(Parallelization) Run 32 worker processes, each of which runs the user-specified command, plus + 1 manager process that sends input to the workers & collects output from the workers

Solution

```
gzcat DNA?.fasta.gz |  
    mpirun -np 33 \  
    mpipe -n 2000 'blastn -db Glycine_max' |  
        awk '$5 - $4 > 100' > output.txt
```

(Data distribution & load balancing) Send 2000 “records” at a time to each worker process, collecting output from the worker process before sending it another 2000.

Solution

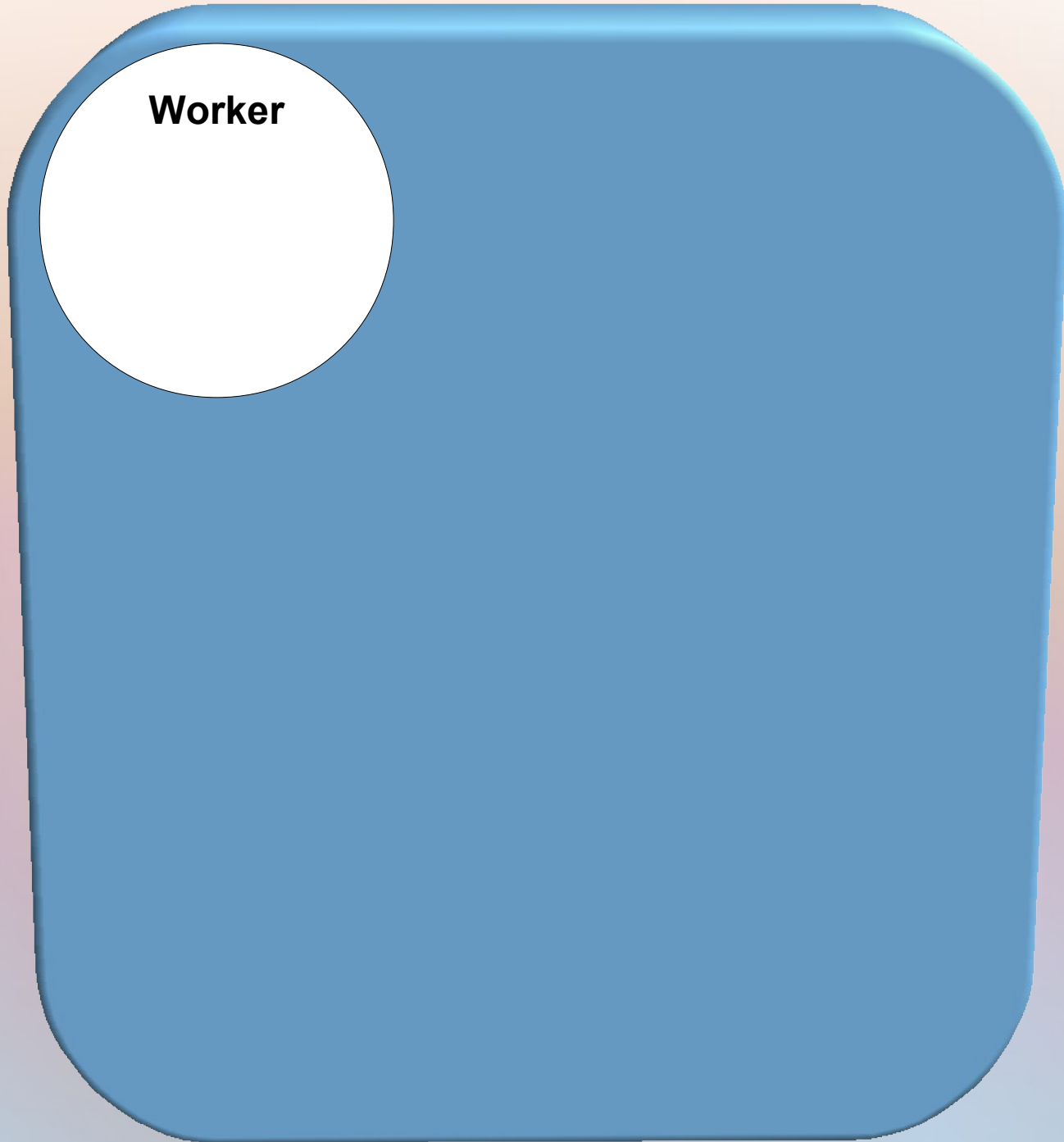
```
gzcat DNA?.fasta.gz |  
    mpirun -np 33 \  
    mpipipe -n 2000 'blastn -db Glycine_max' |  
        awk '$5 - $4 > 100' > output.txt
```

Output is printed (non-interleaved) to standard output; in this case, .

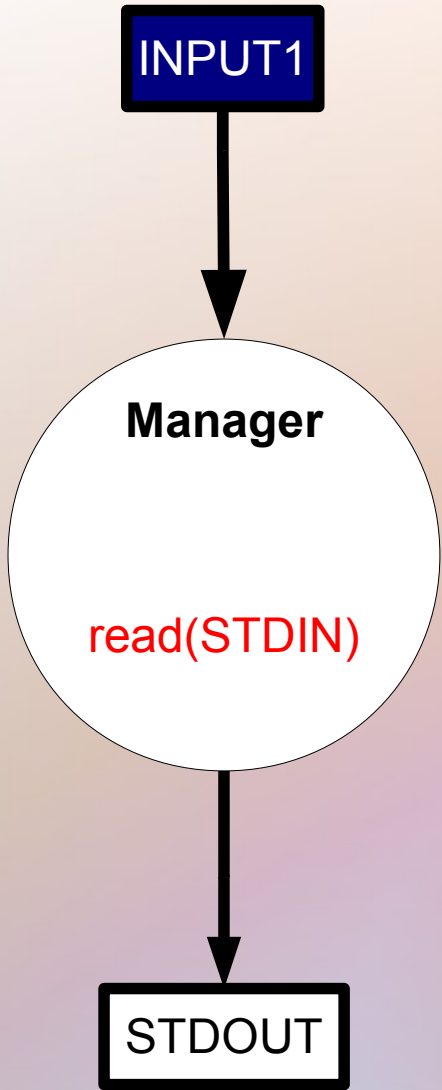
fork()

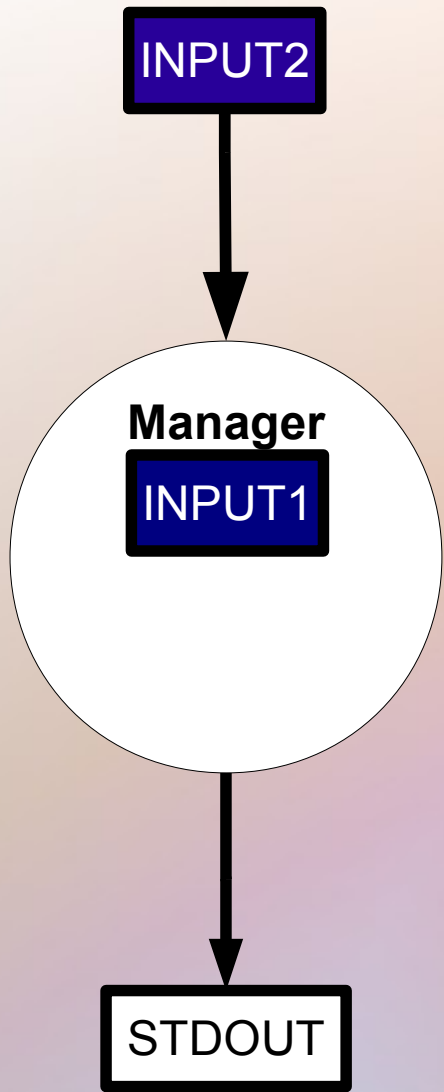
- Creates a new (“child”) process
 - clone of the parent (same program, same state)
 - has its own copy of all memory/variables in program
 - begins execution at the call to fork()

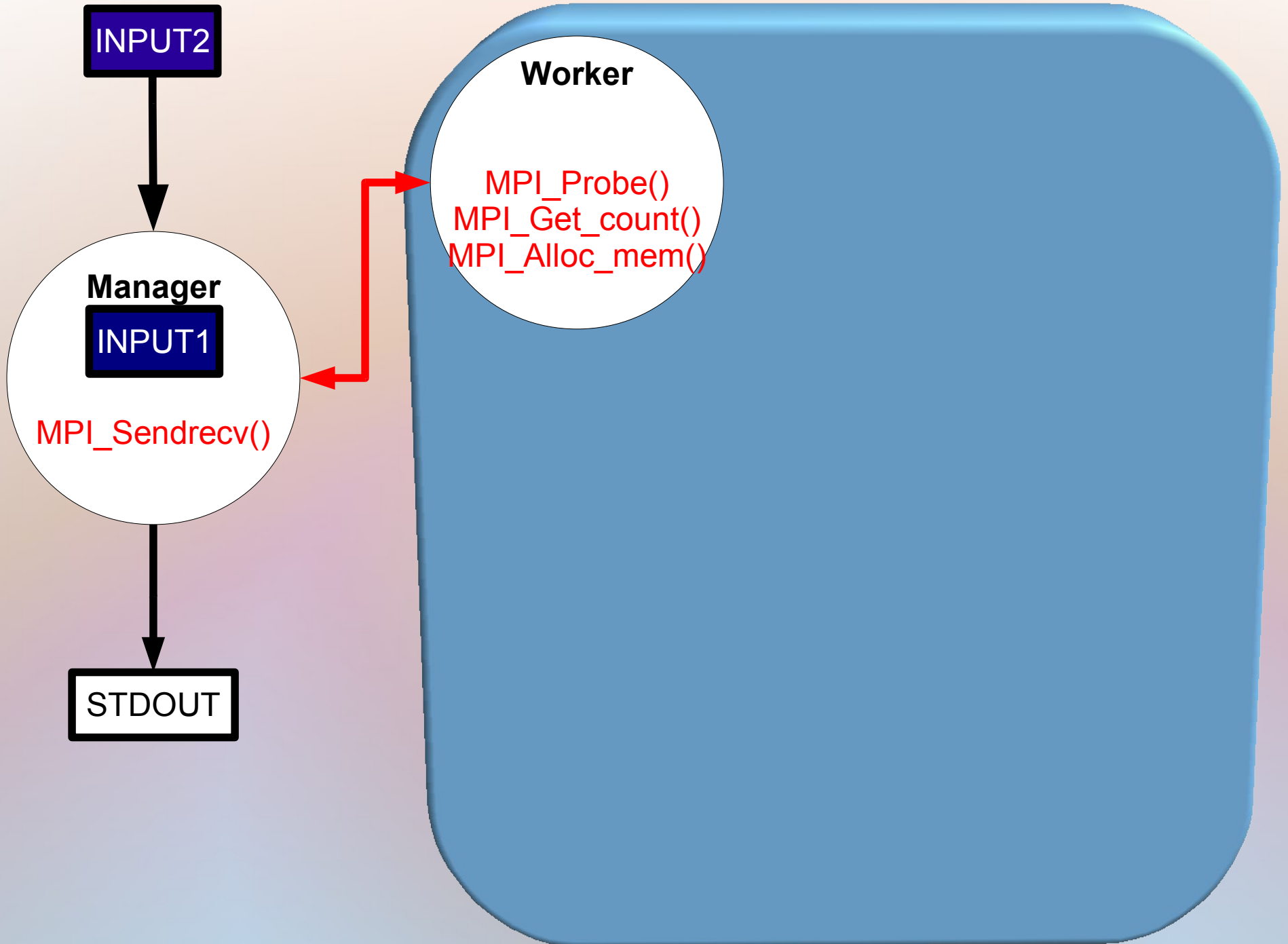
```
if (fork() == 0) {  
    // child process executes this  
} else {  
    // parent process executes this  
}
```

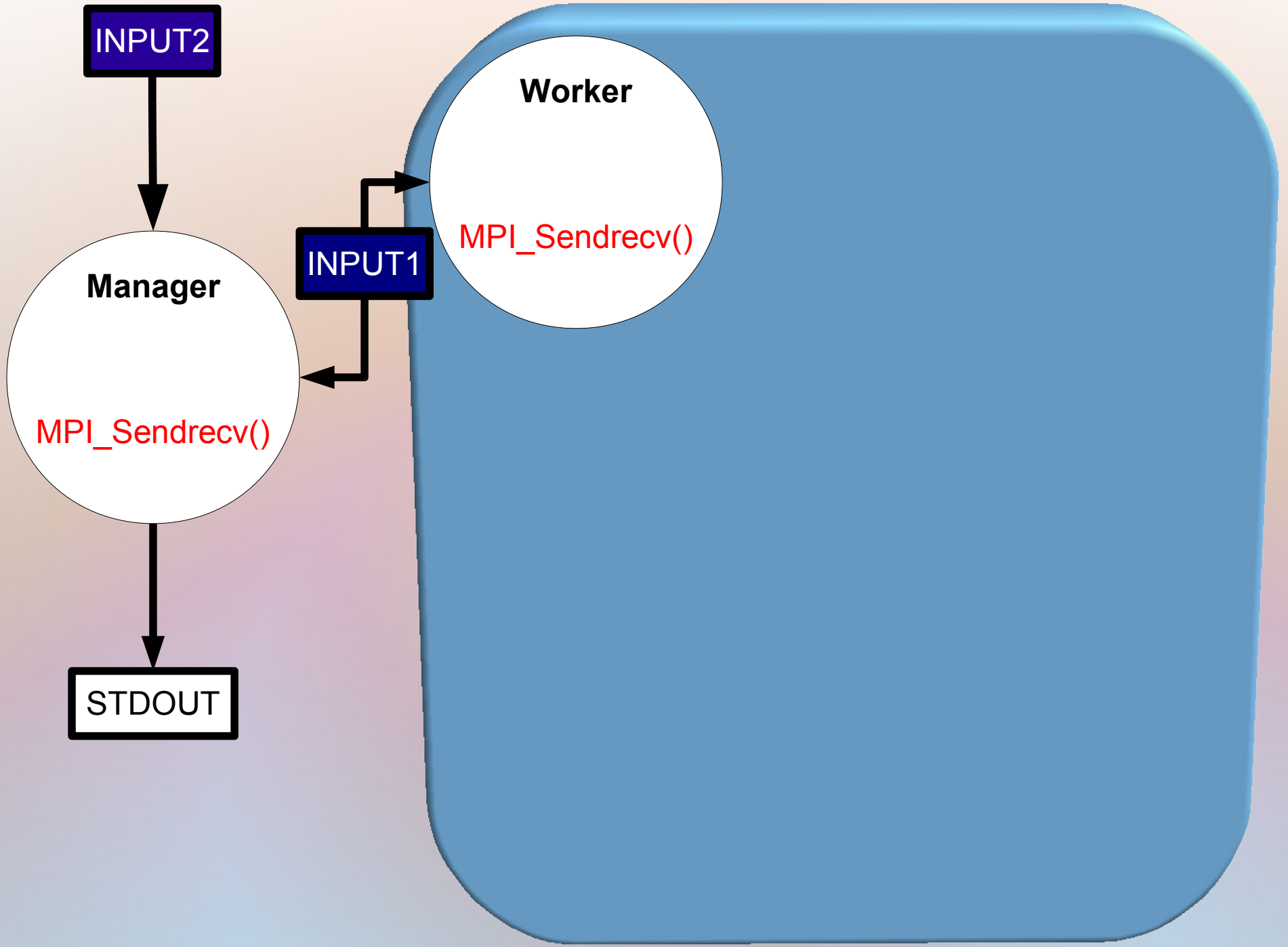




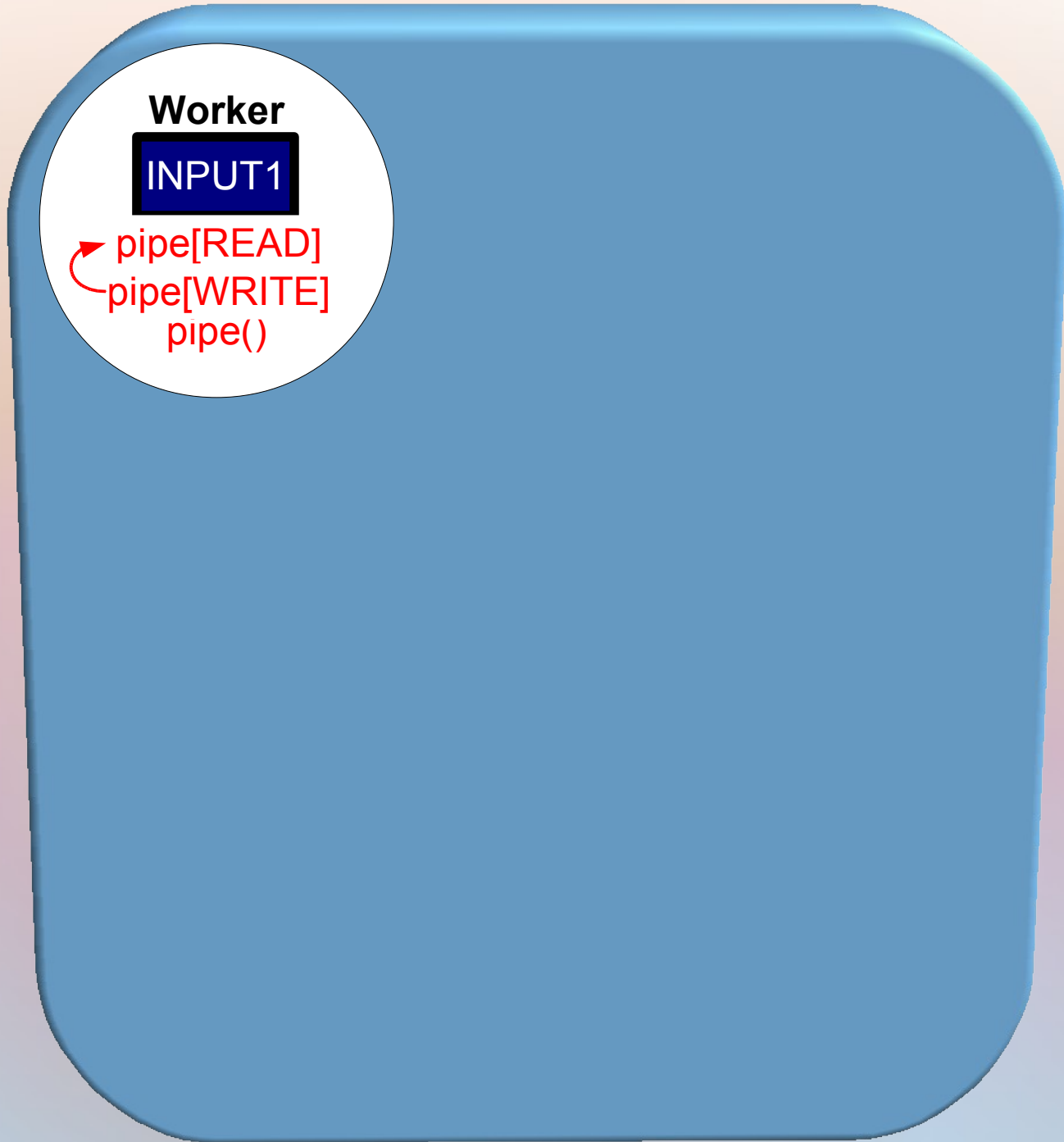
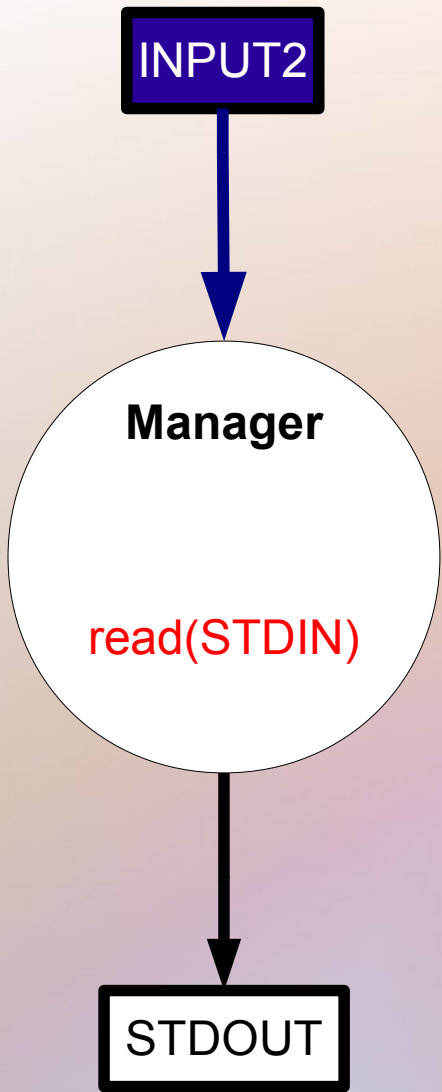


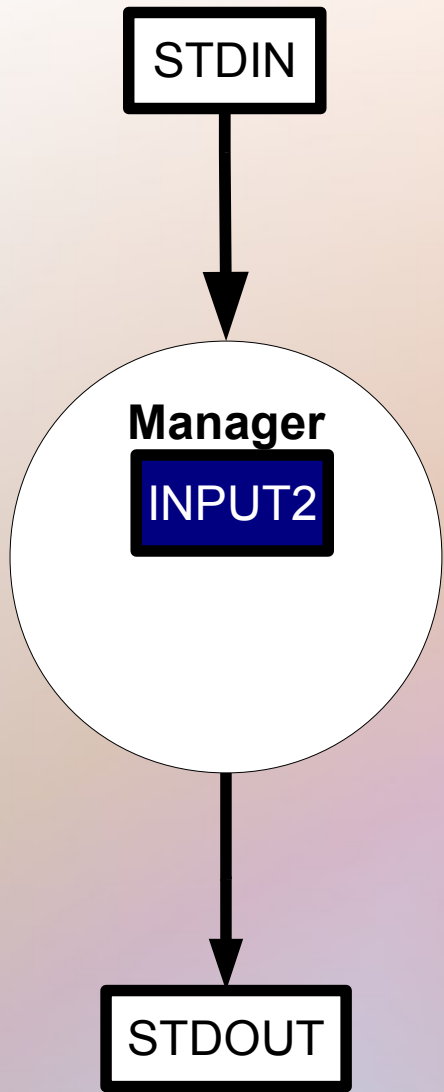


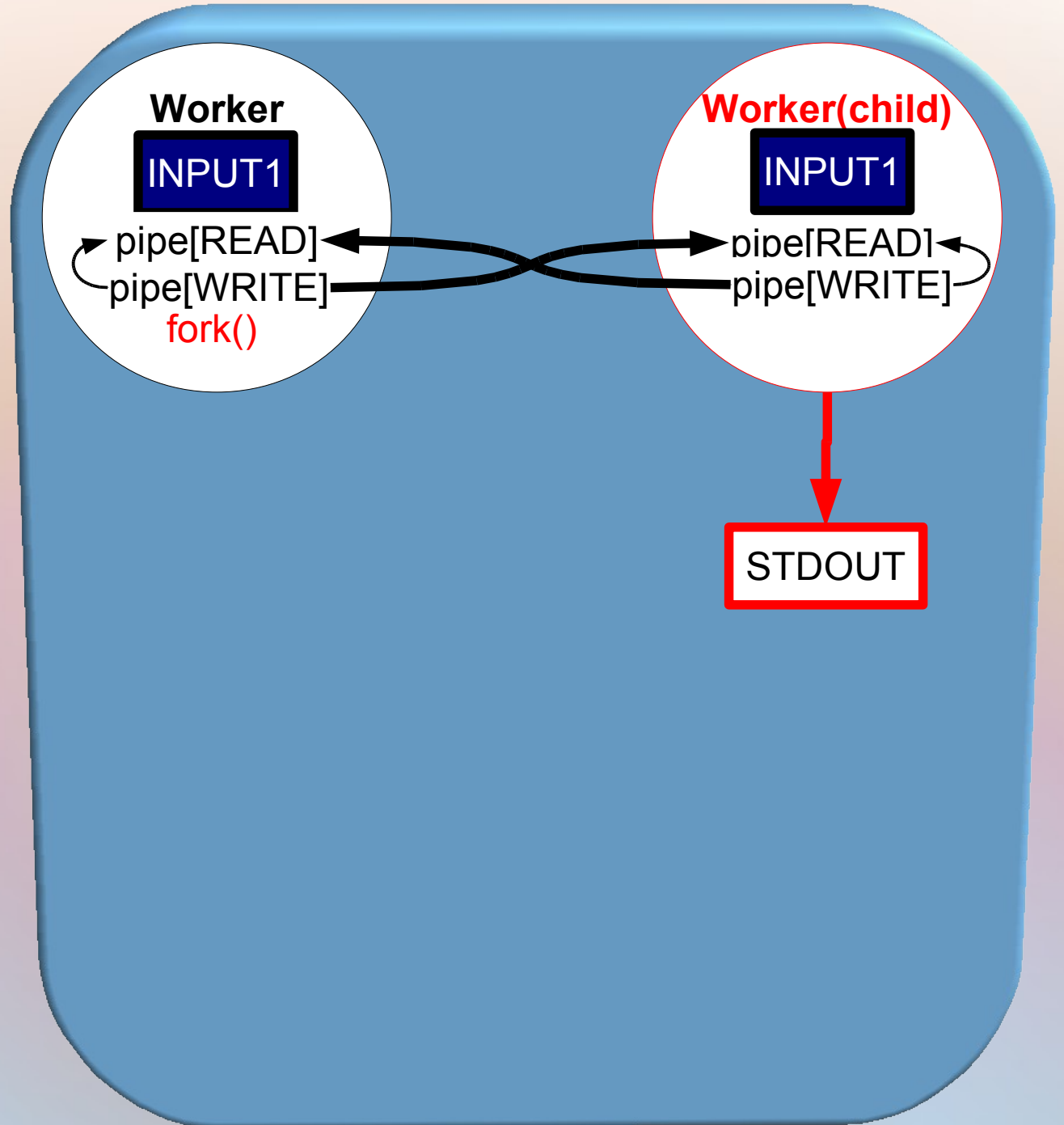
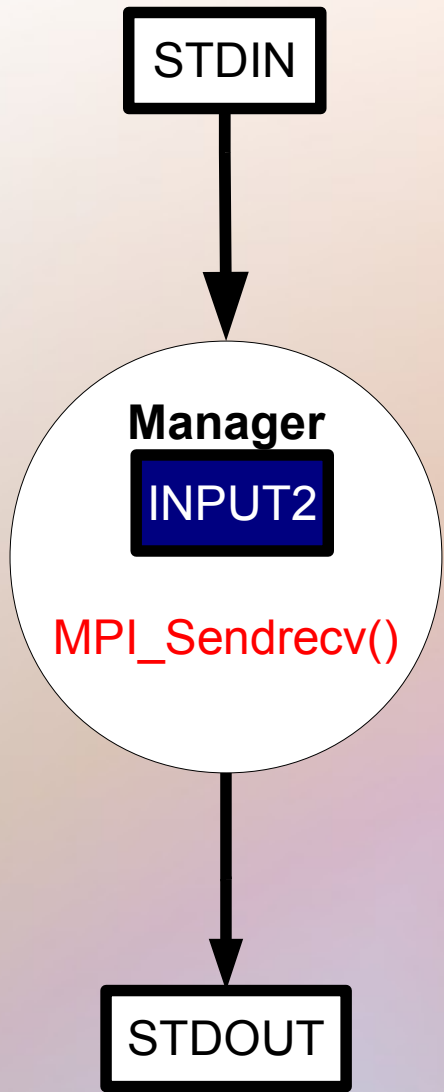


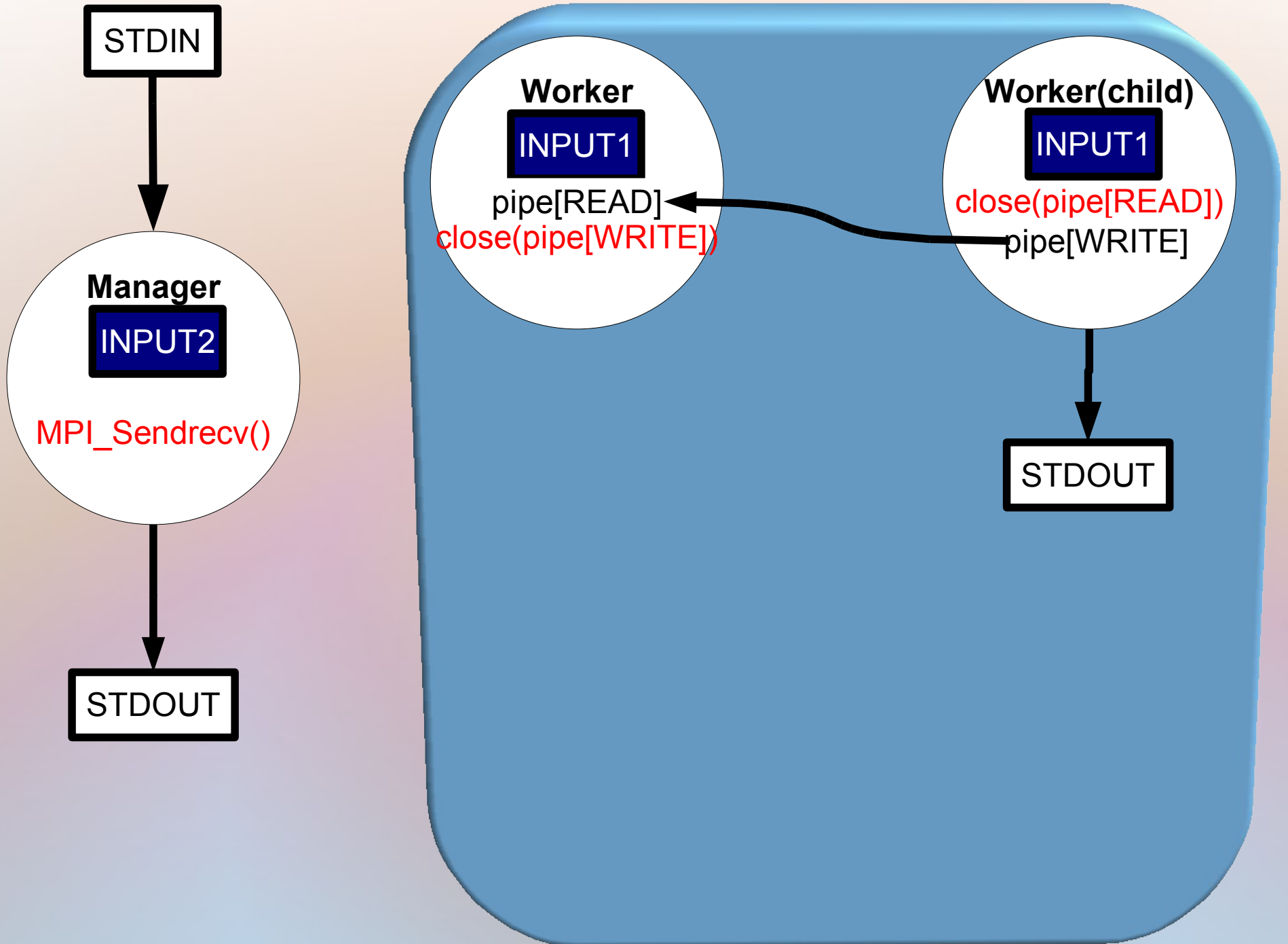


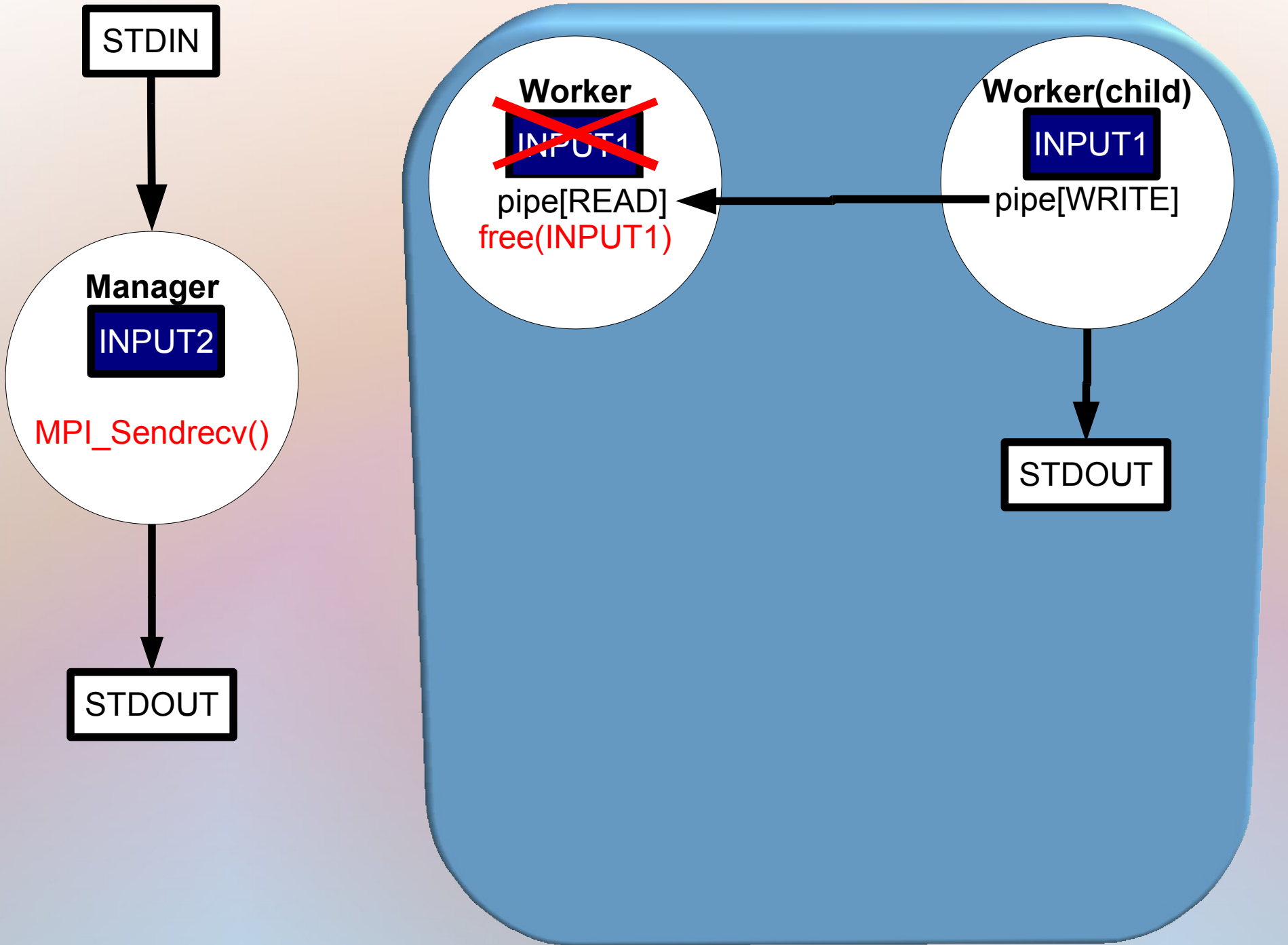


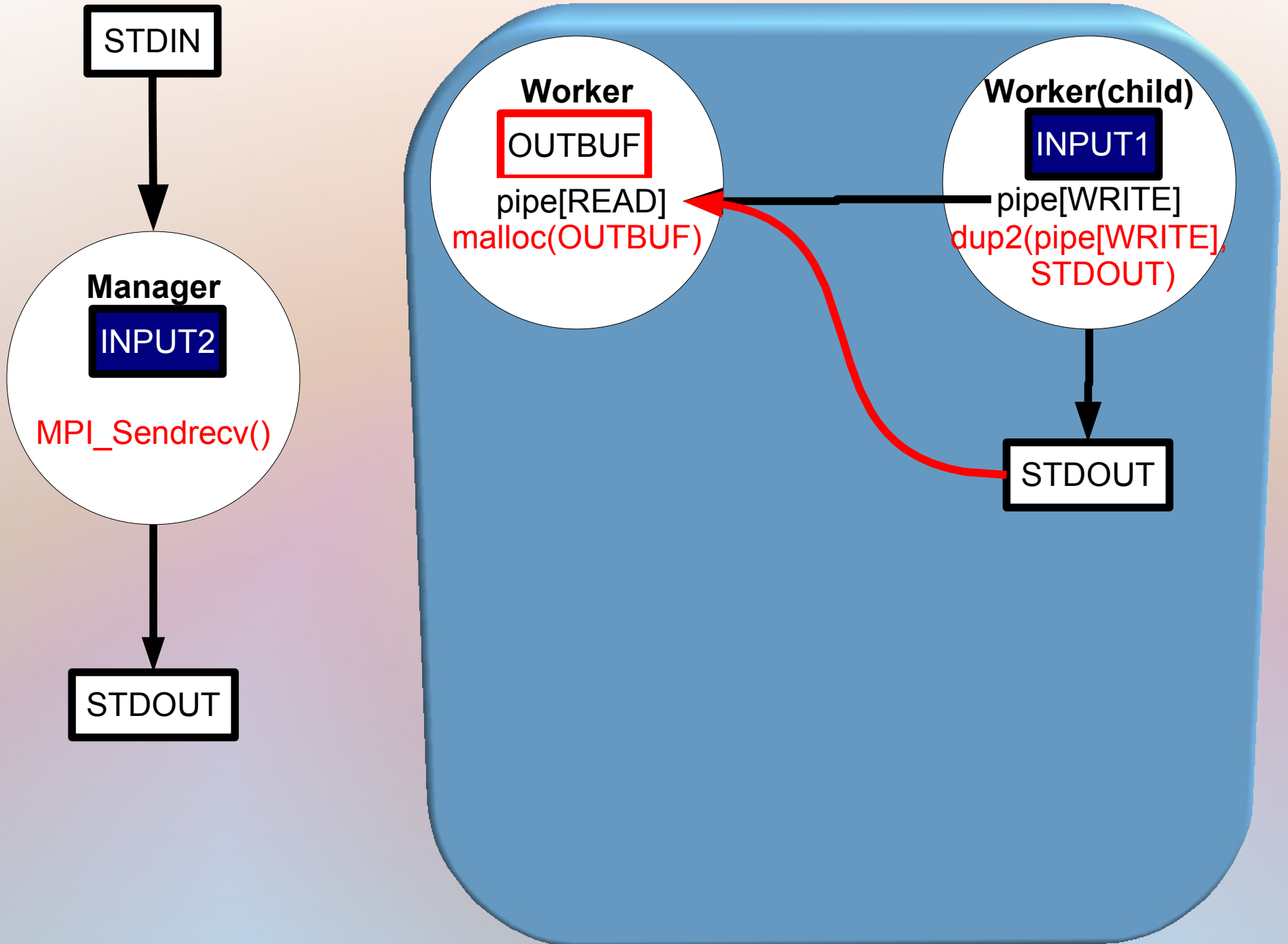


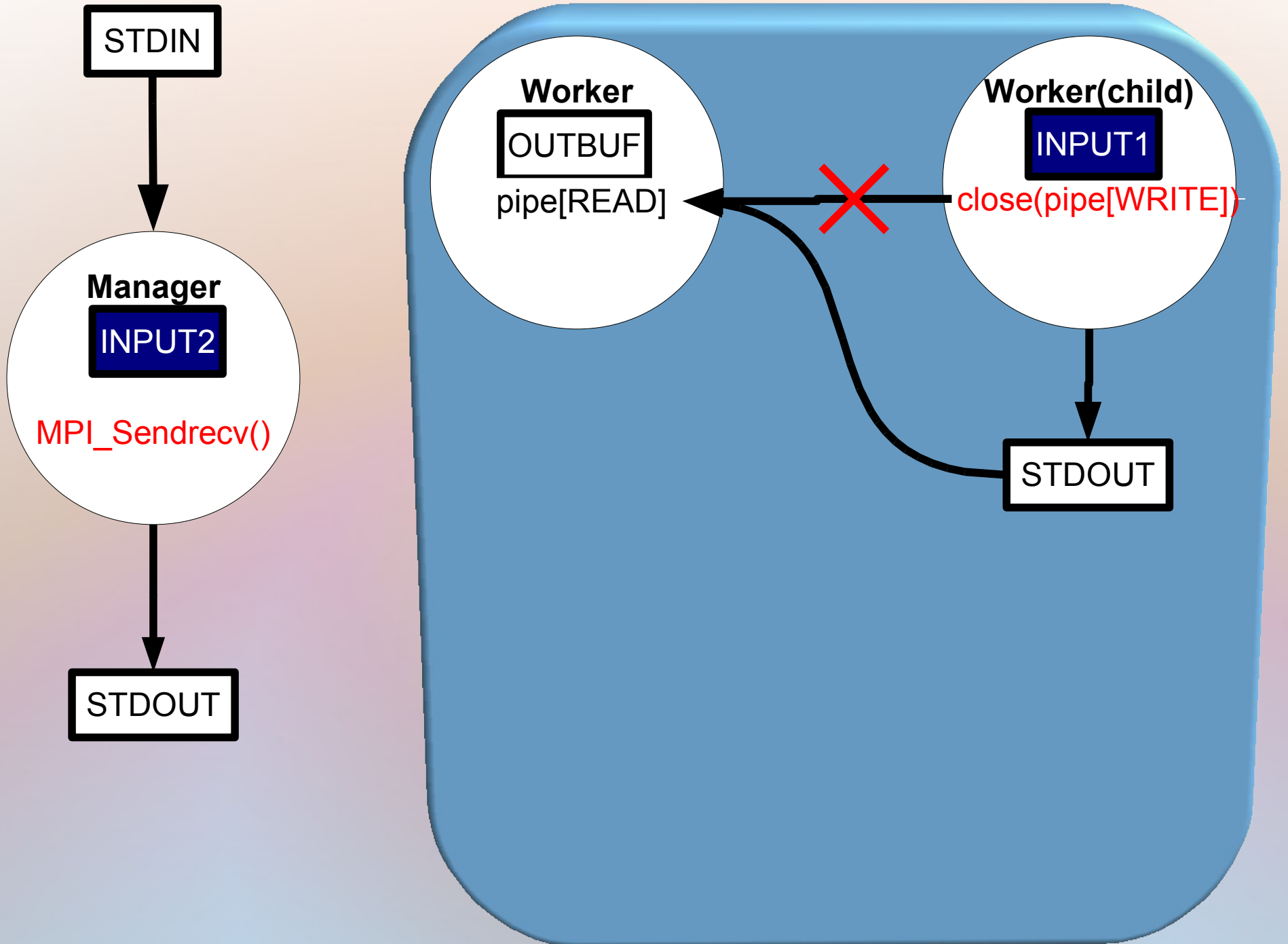


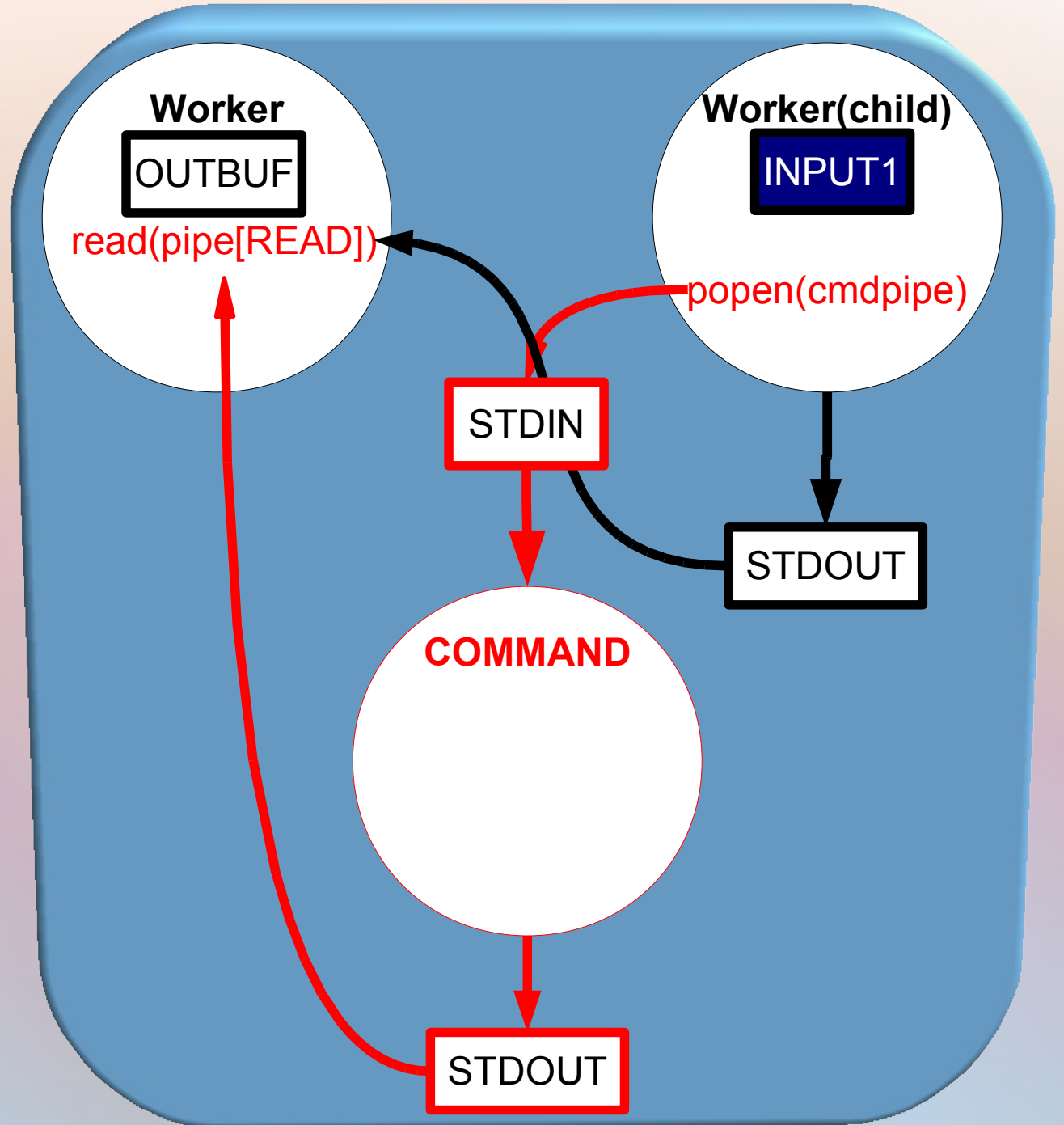
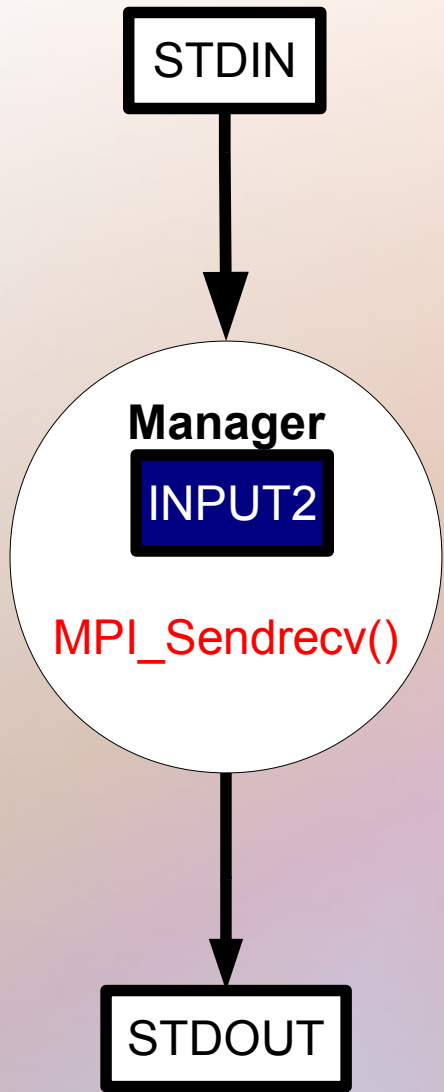


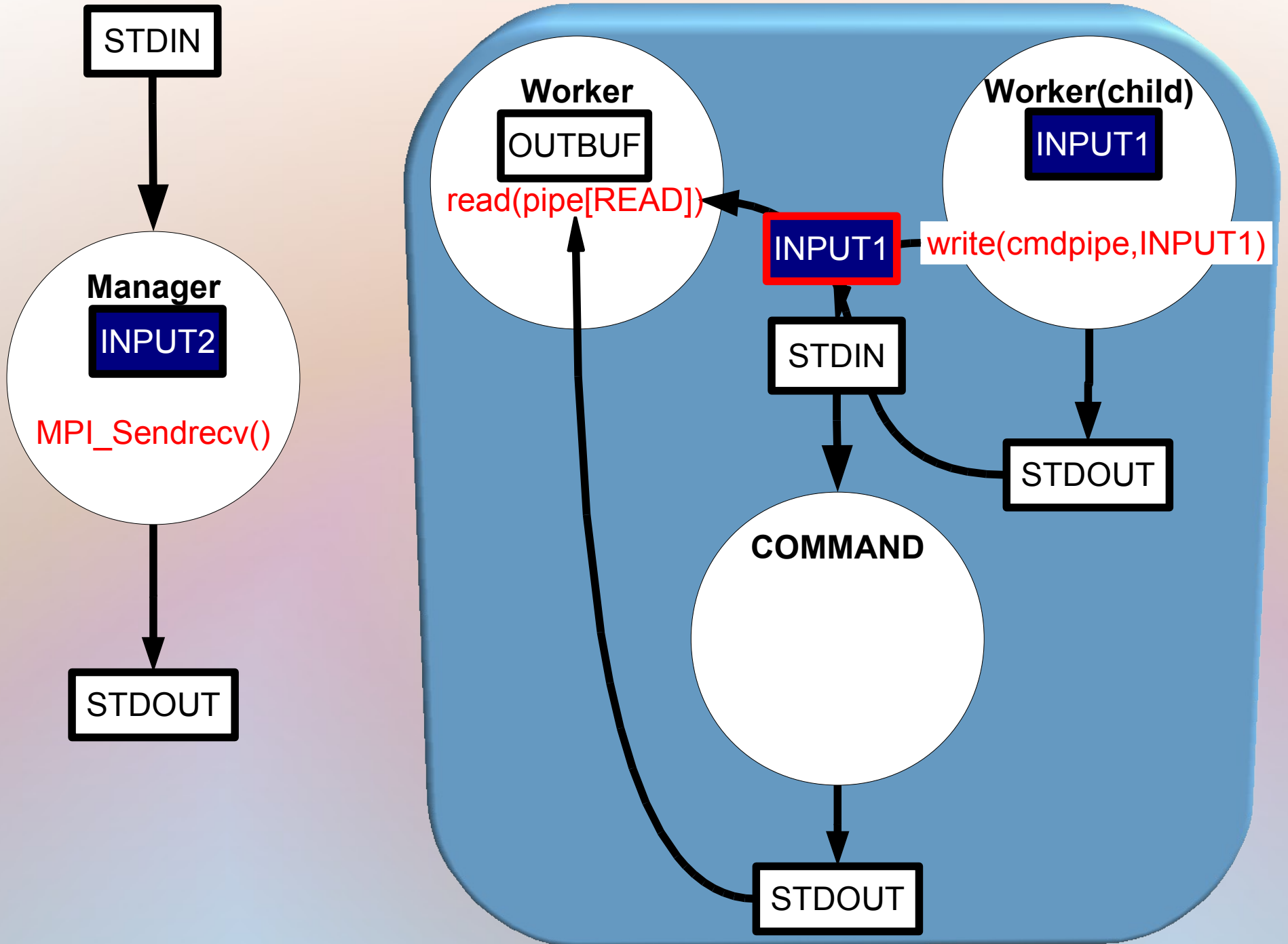


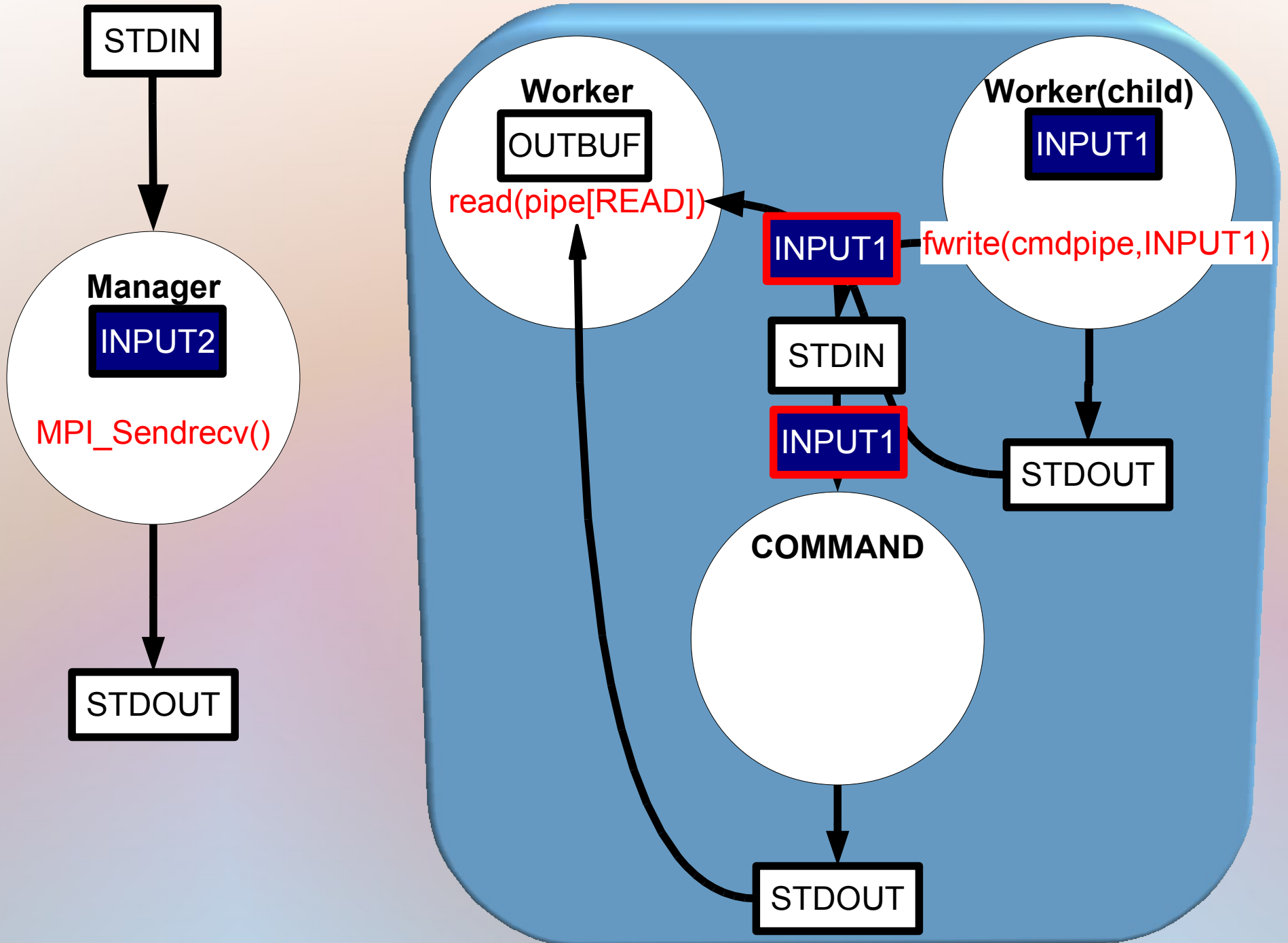


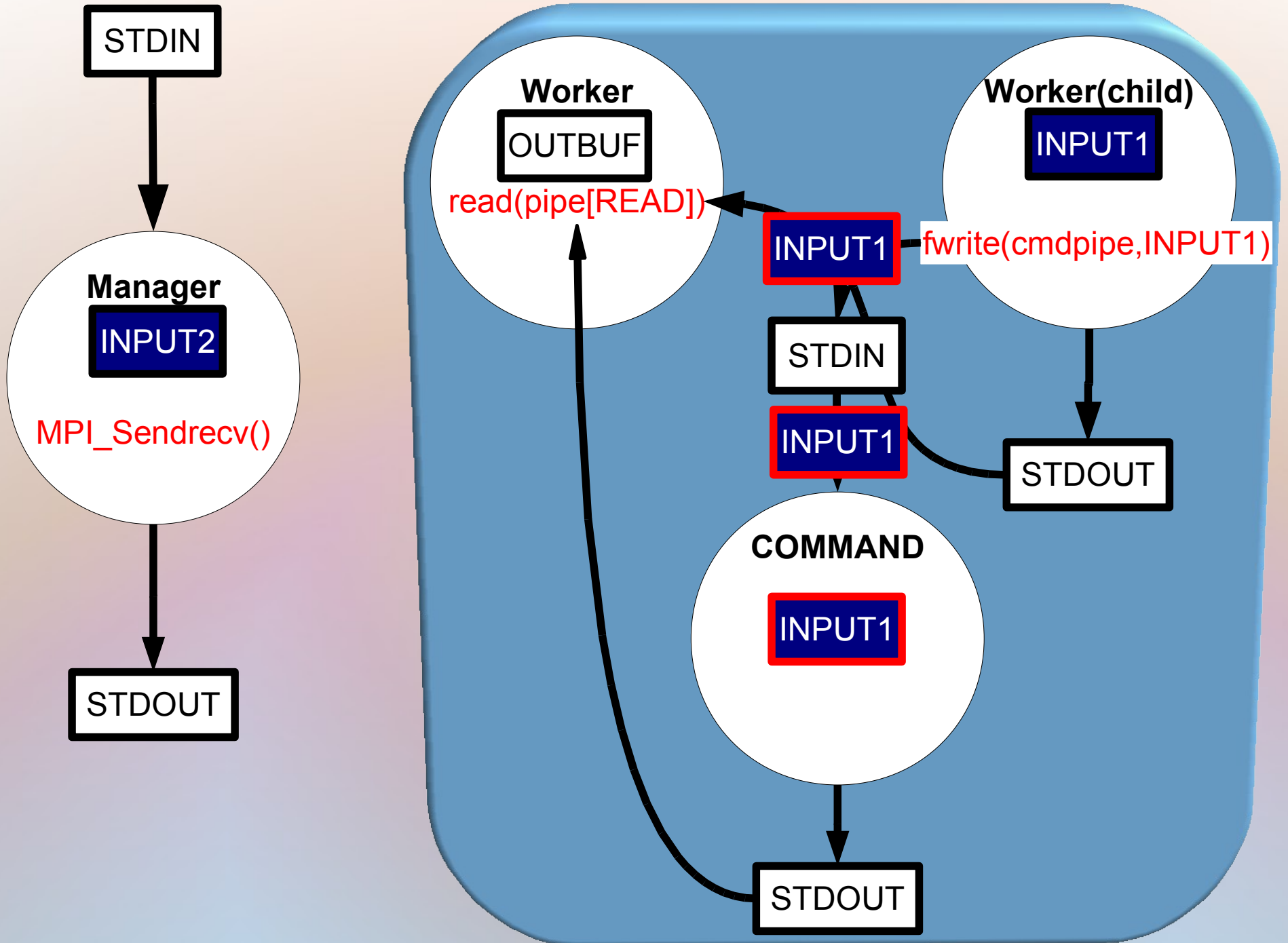


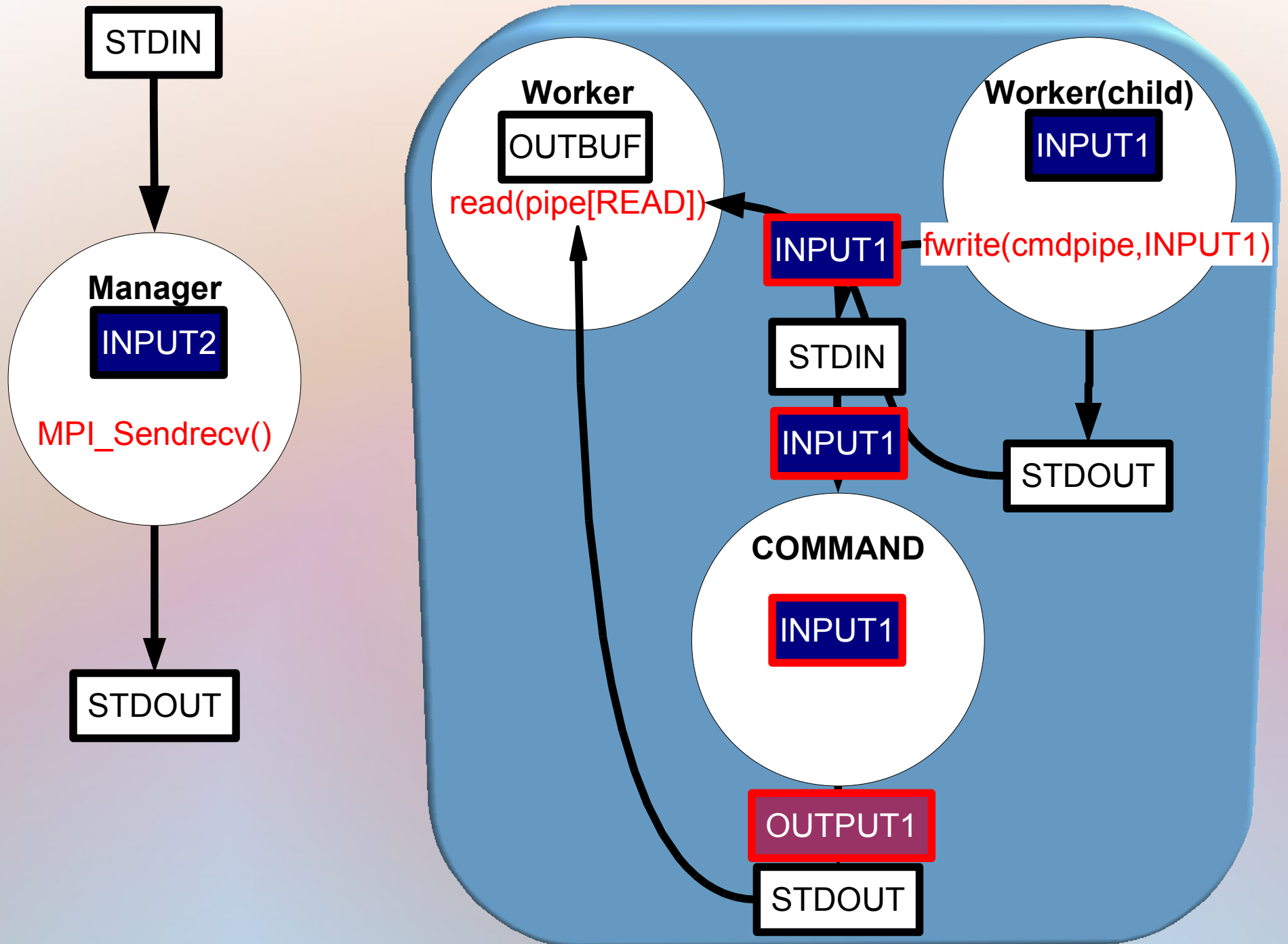


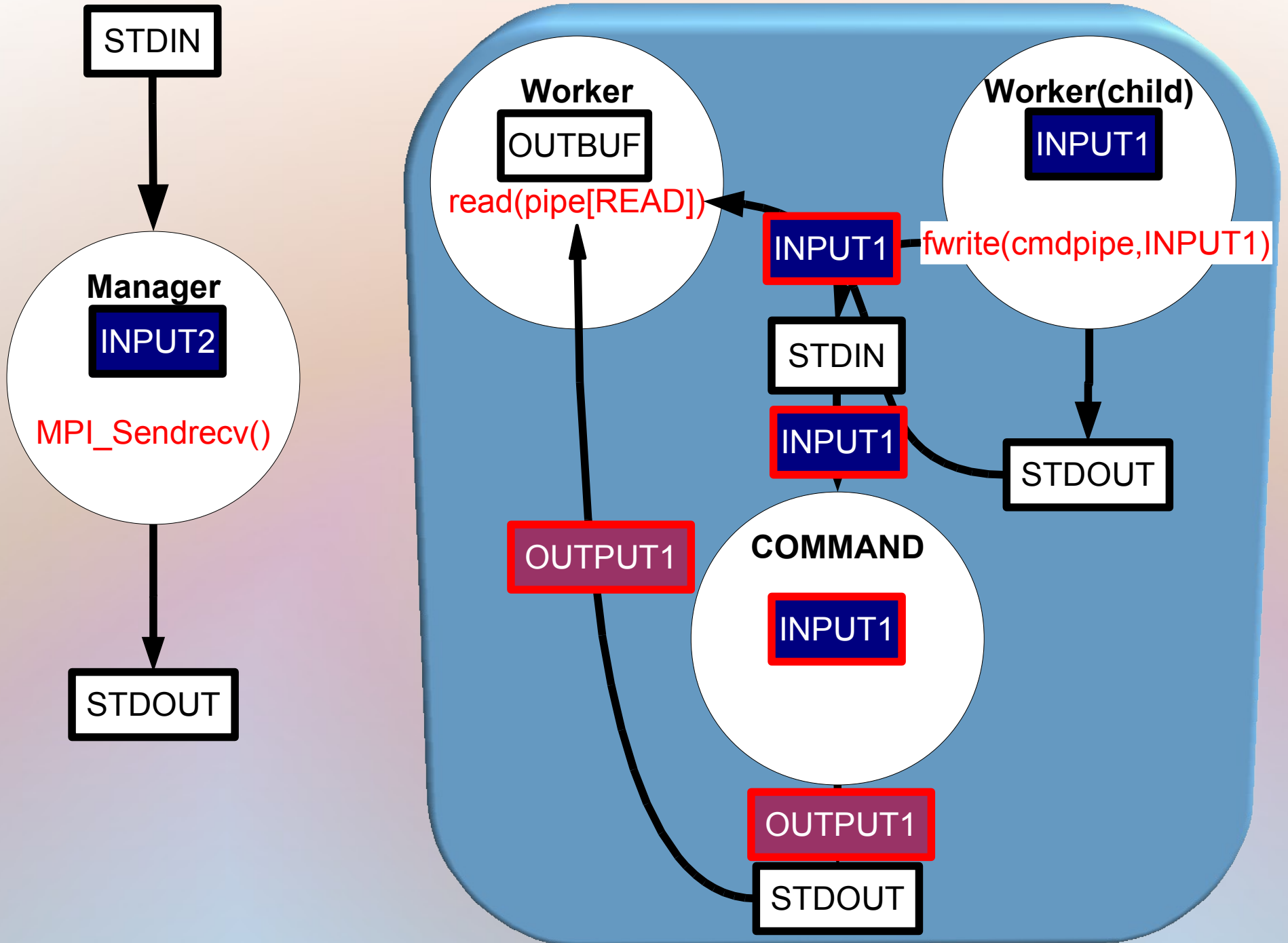


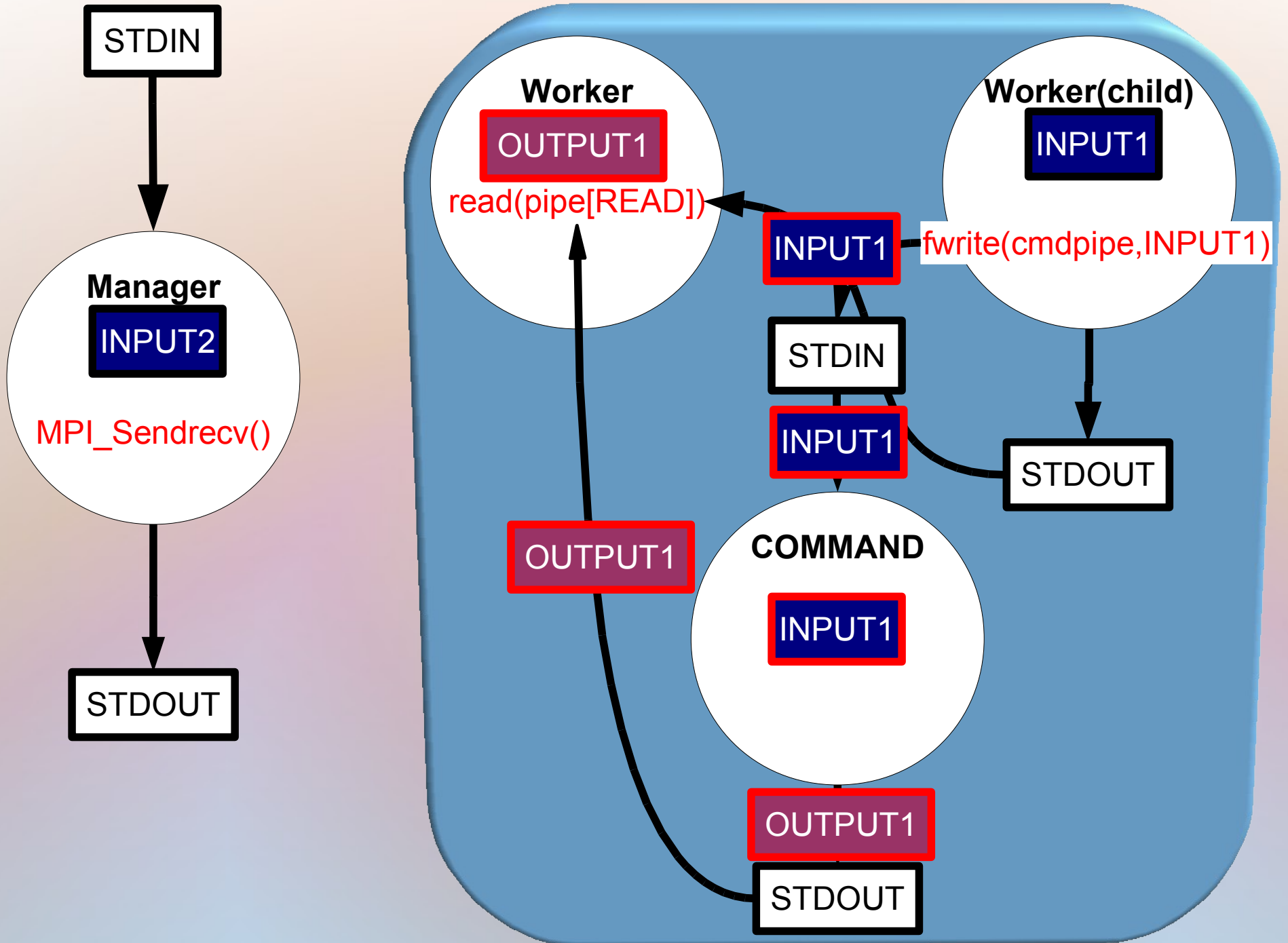


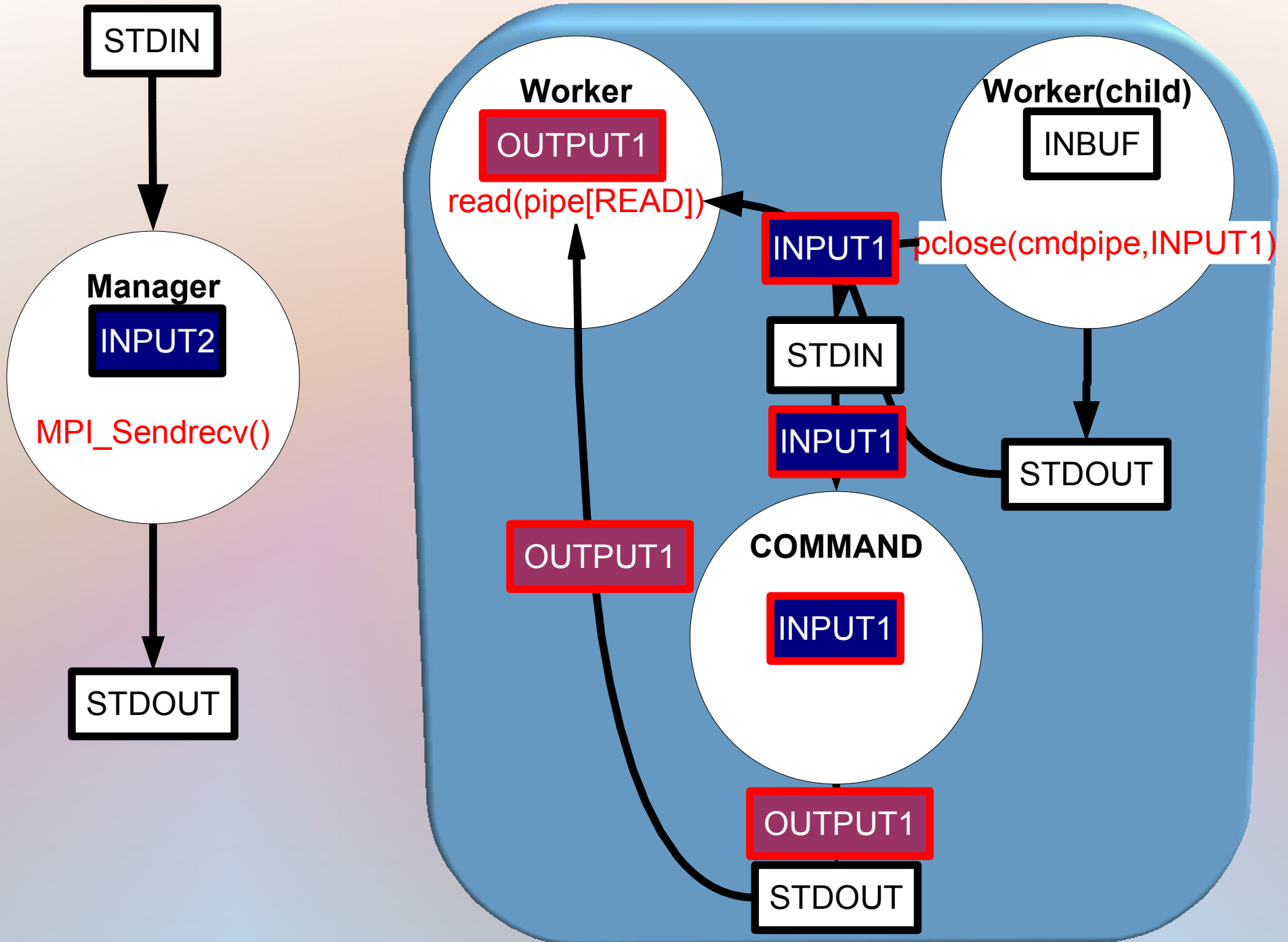


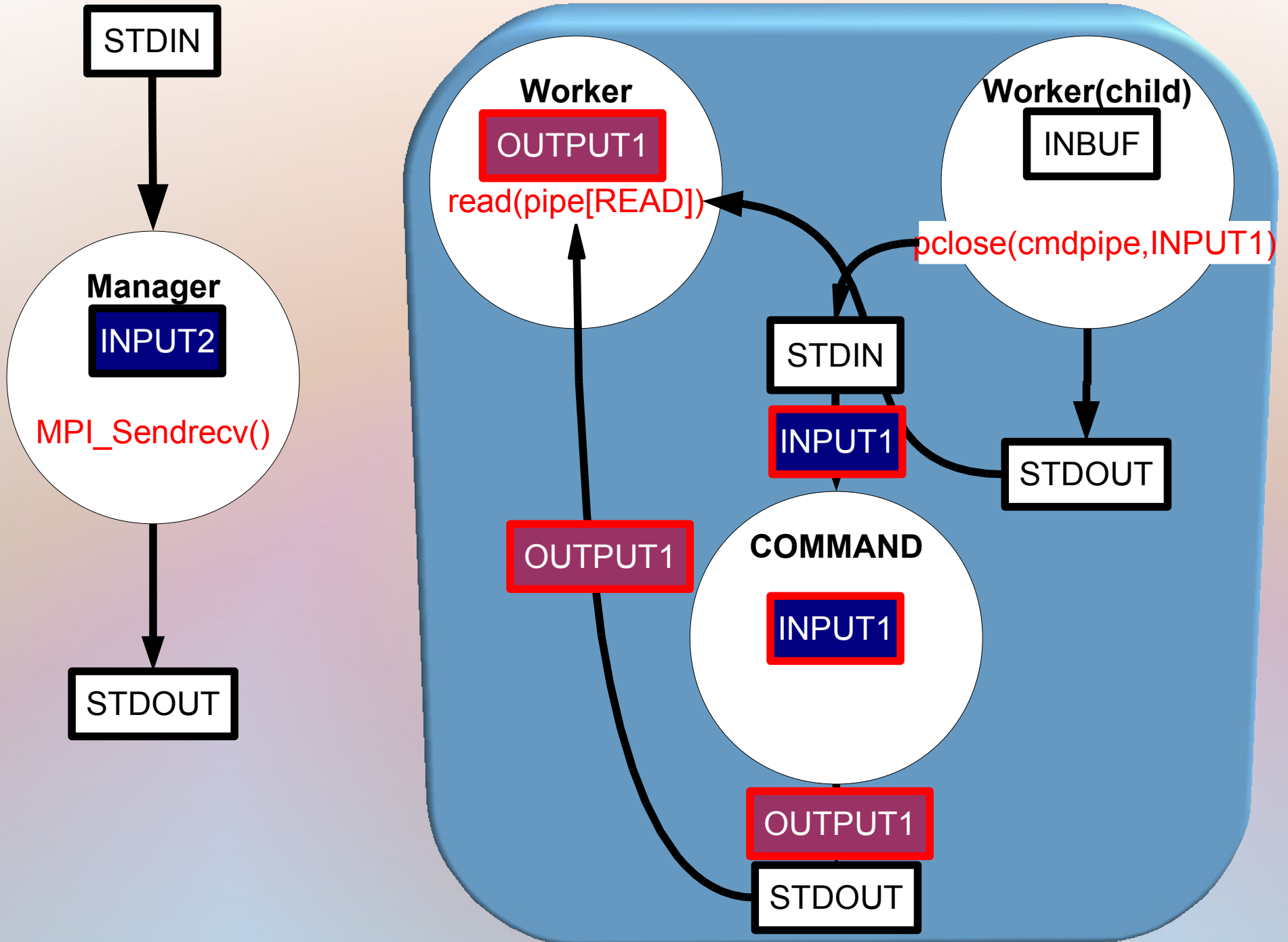


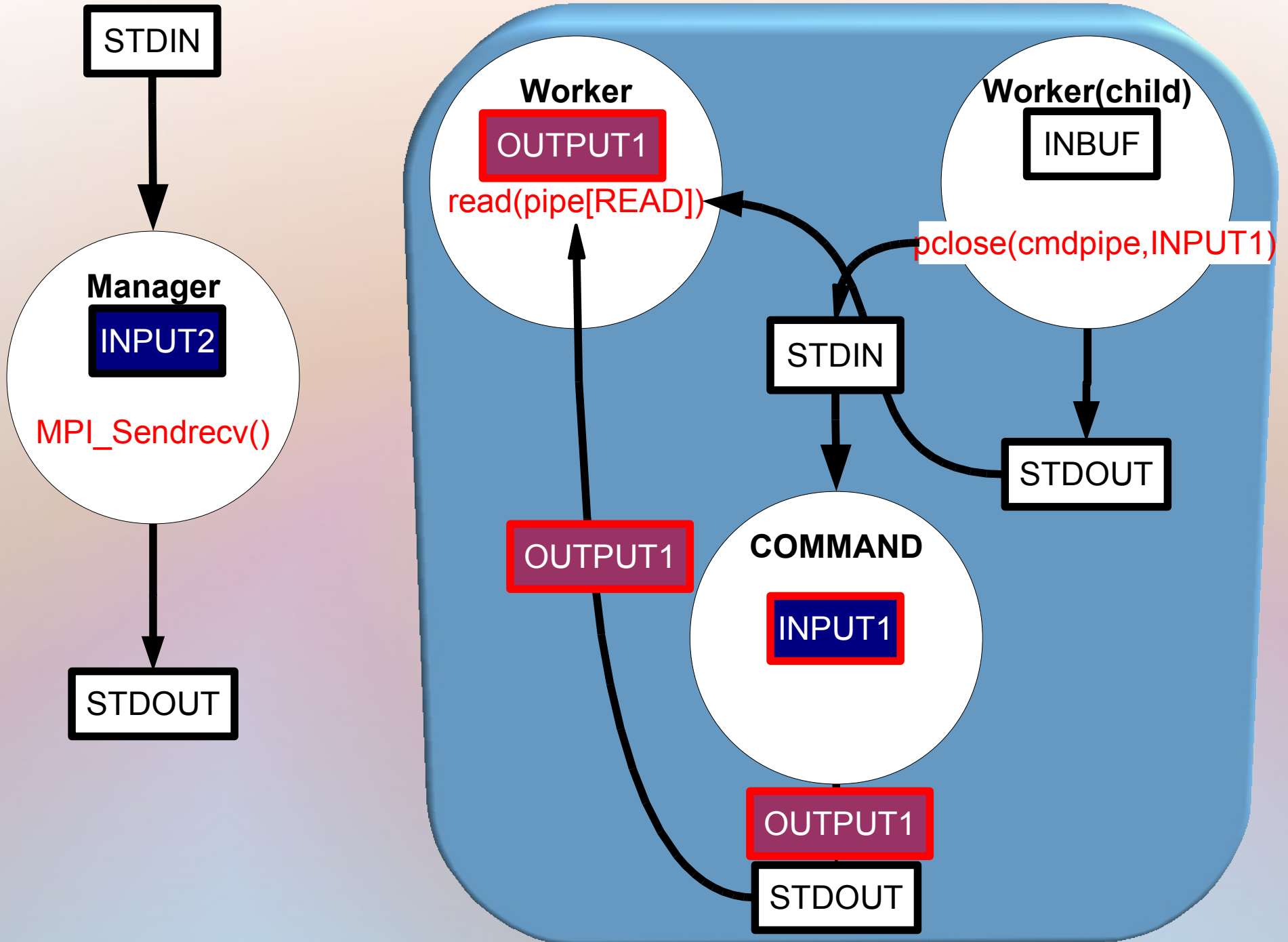


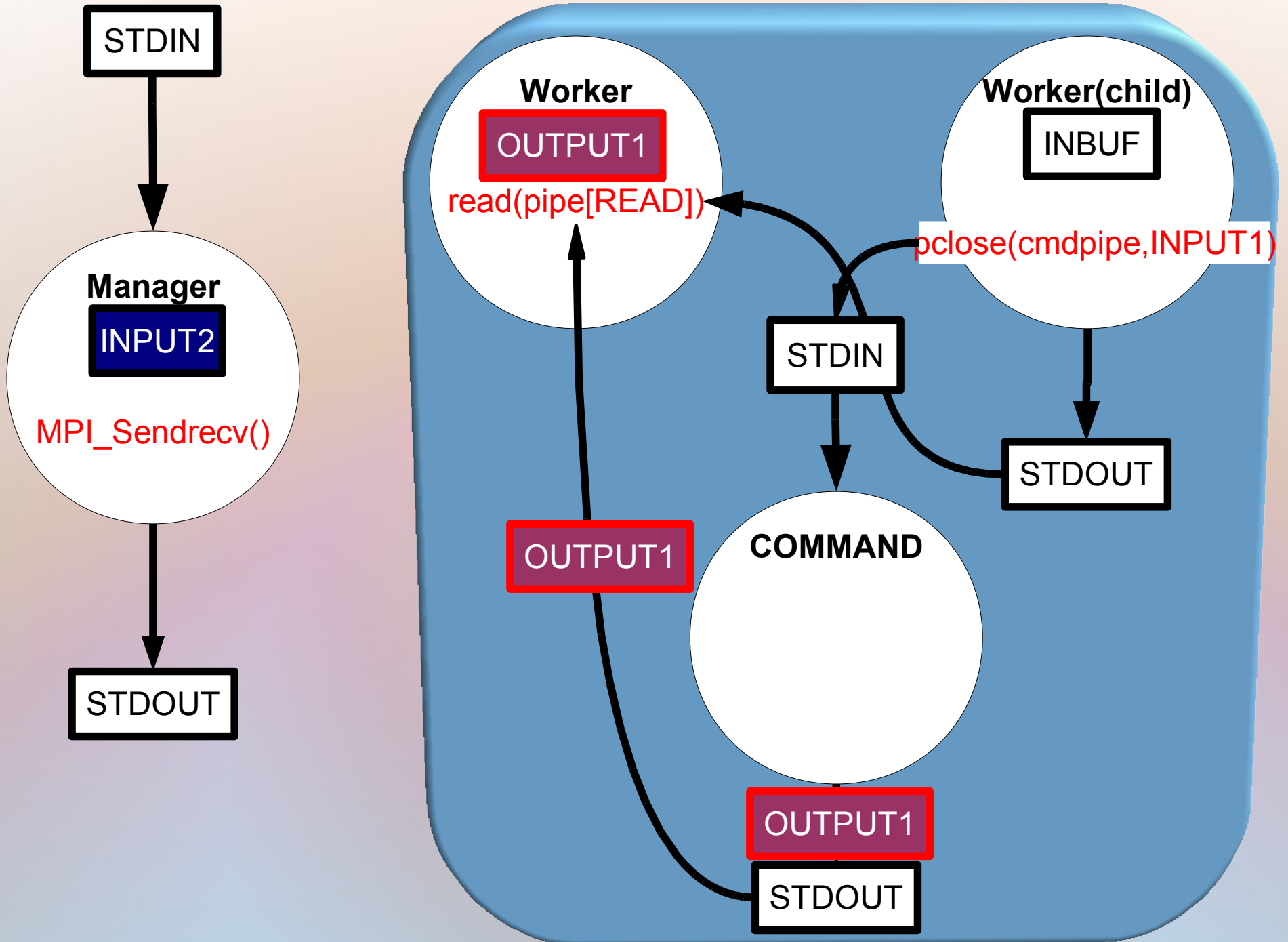


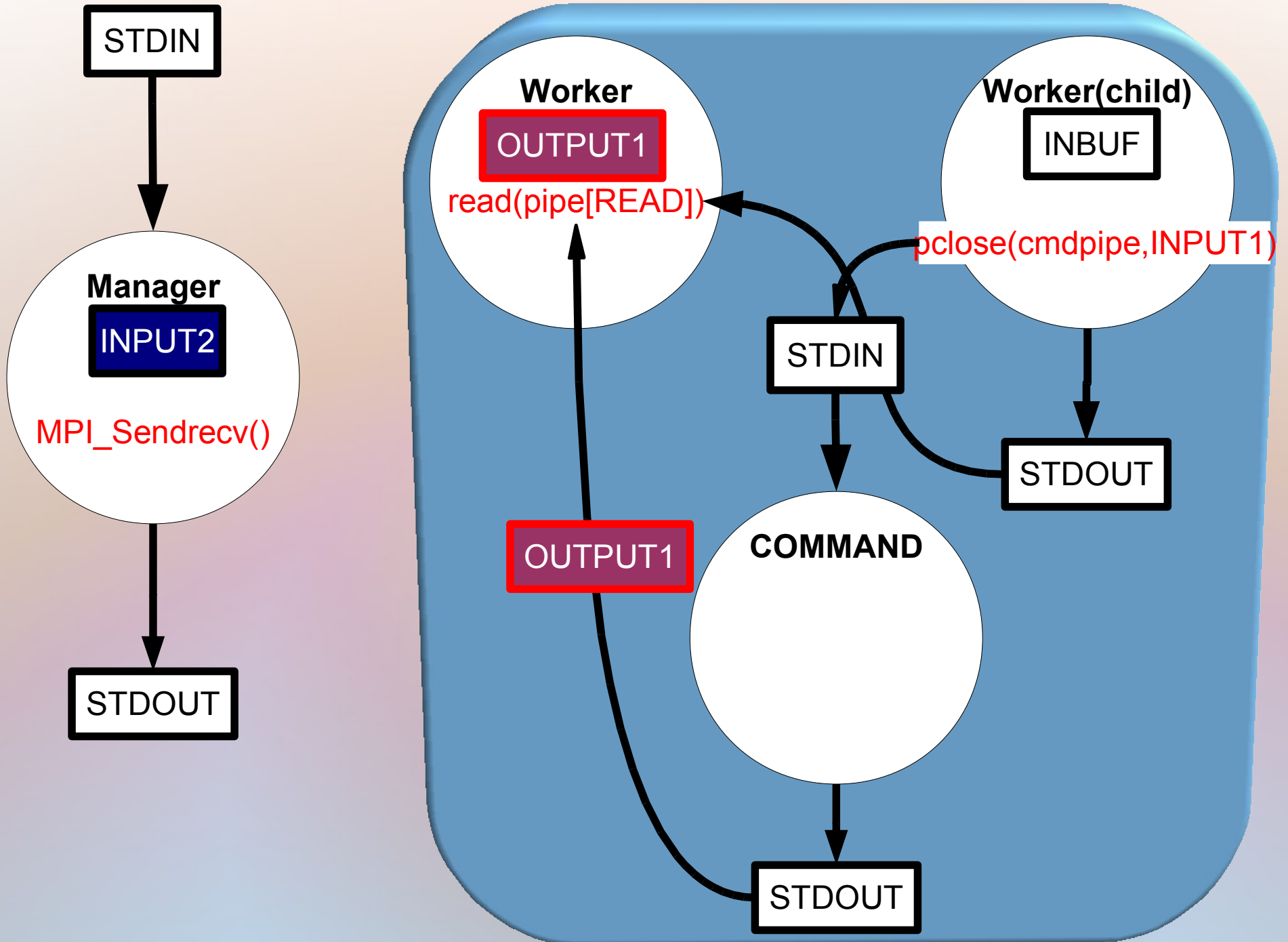


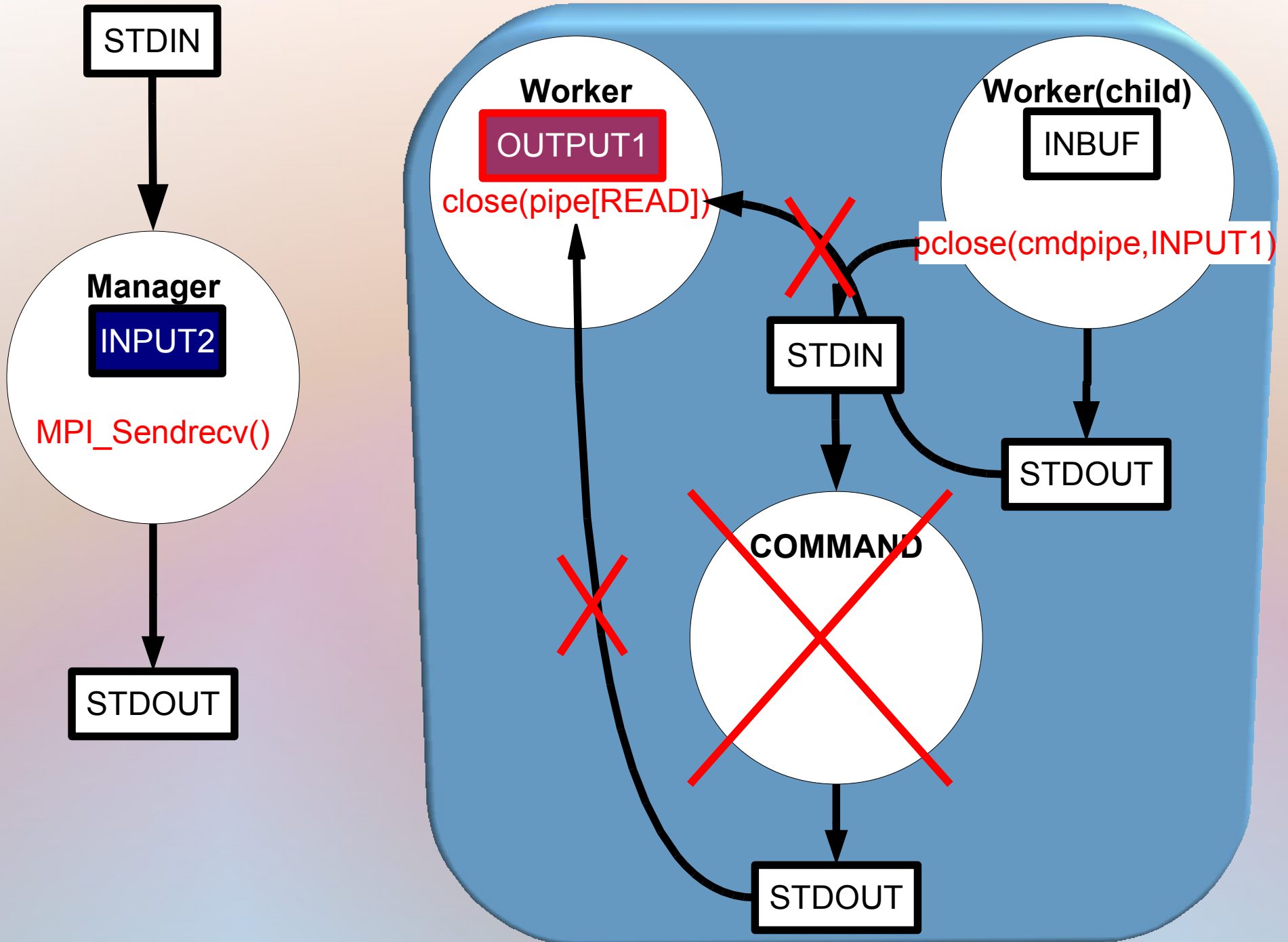


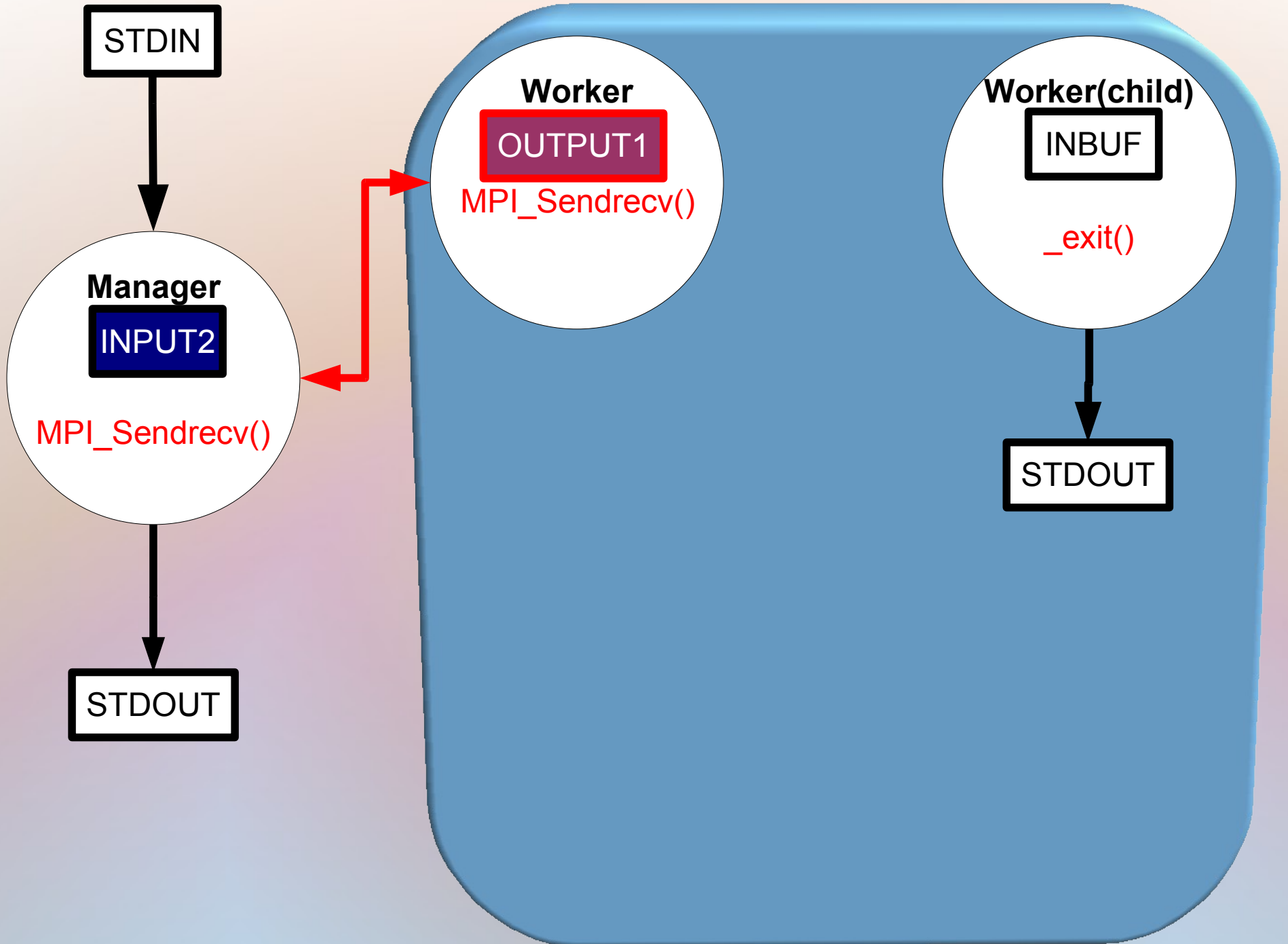


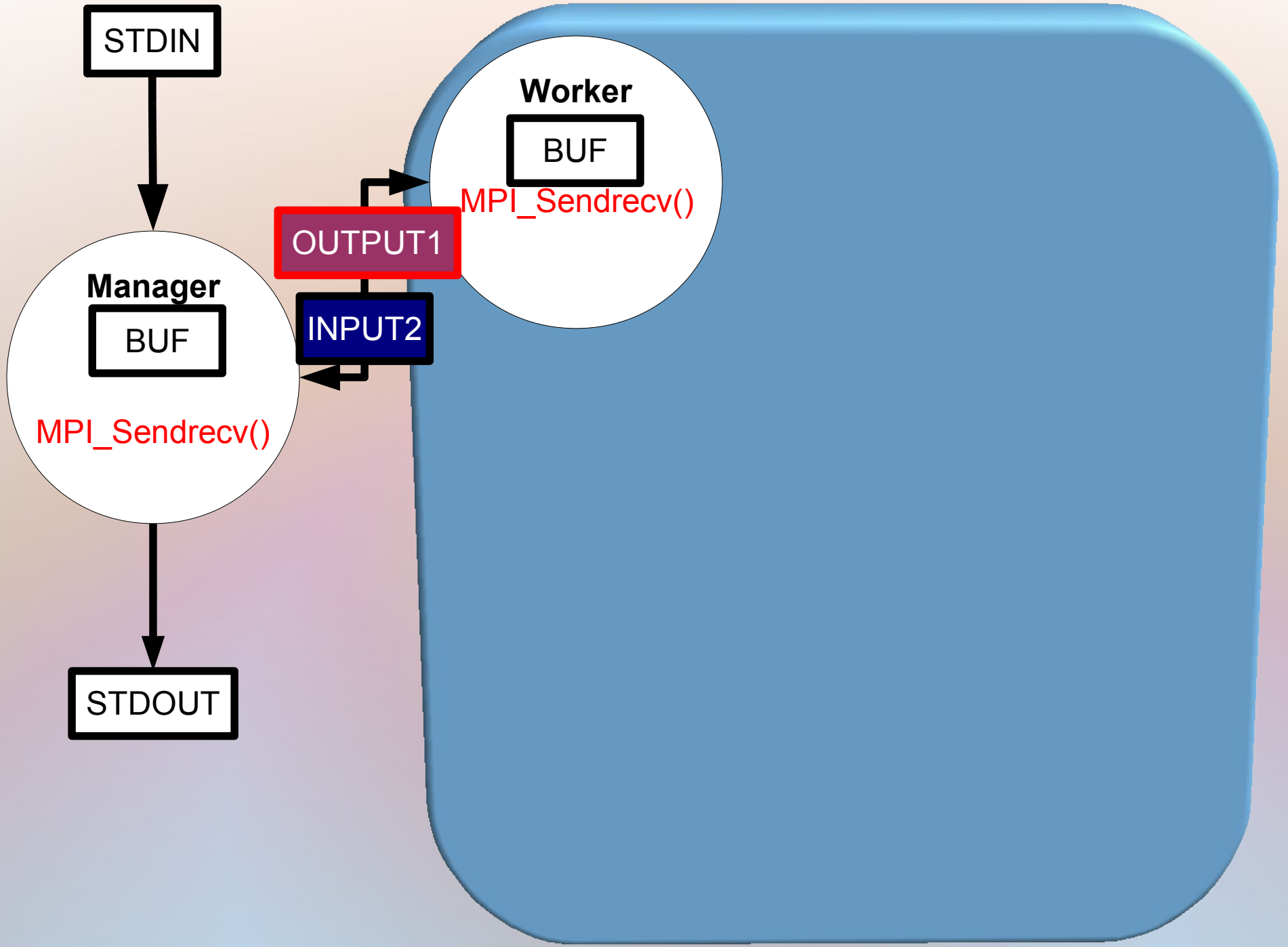


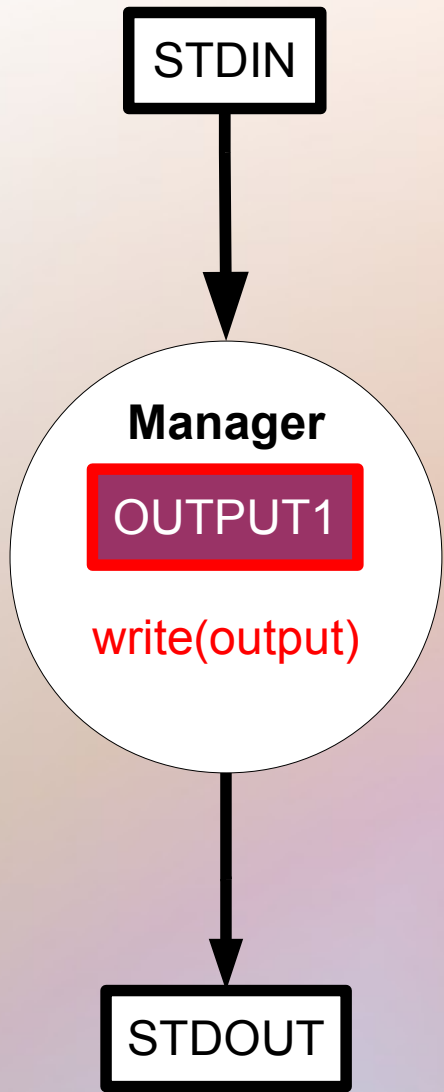


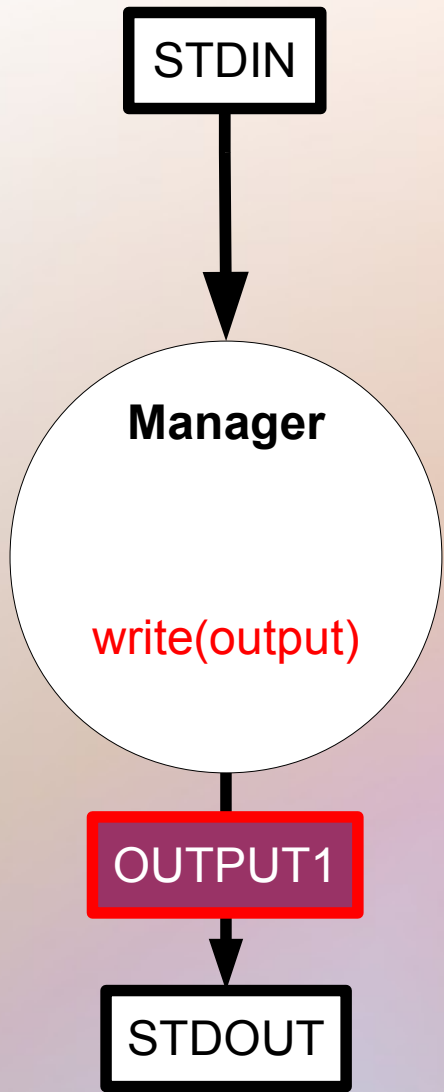








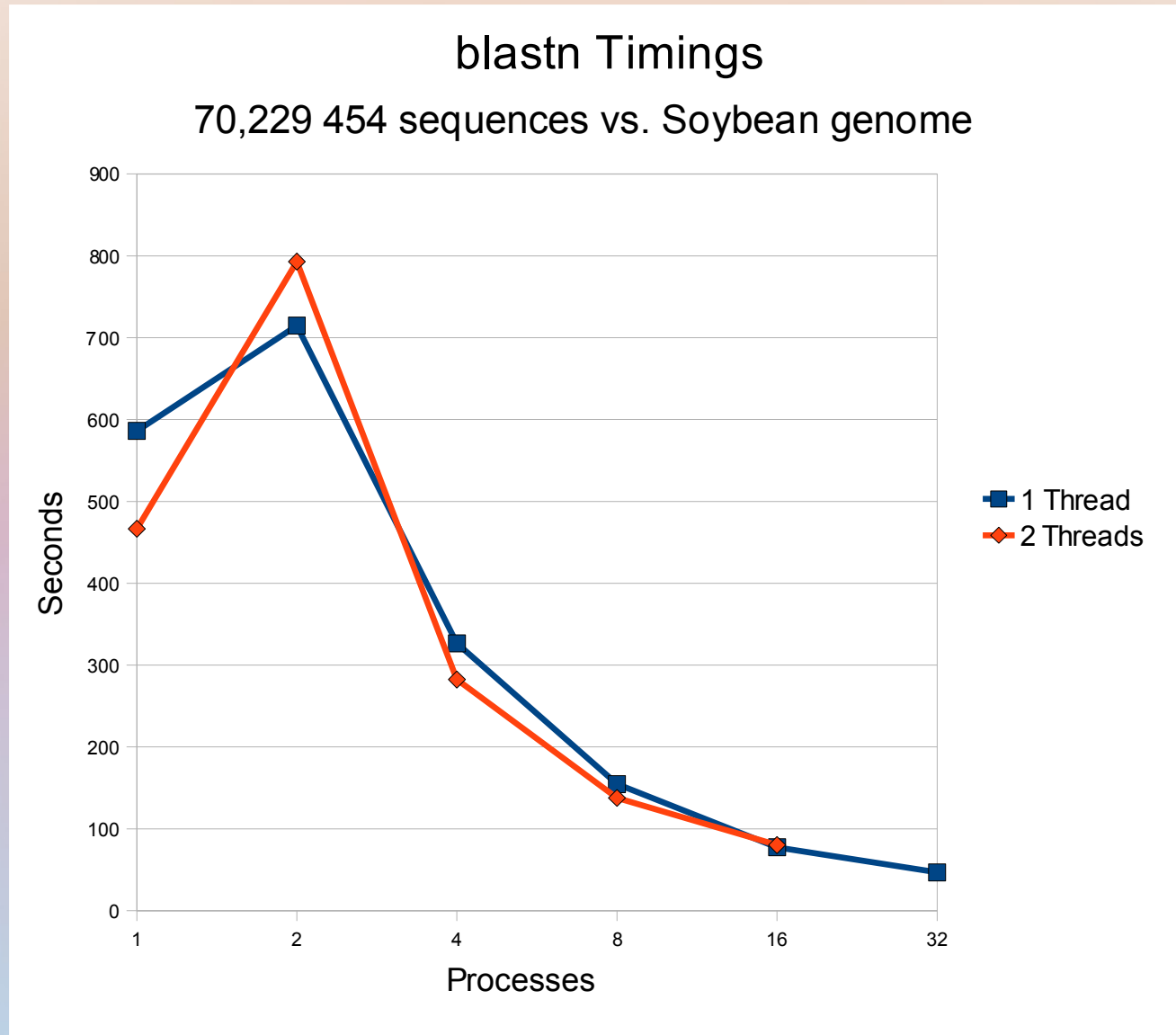




Benchmark

- System: hpc-class
- Program: blastn (NCBI blast-2.2.19+)
- Input Data
 - Query
 - 70,229 “454” EST sequences (average length approx. 233 nucleotides)
 - Target
 - Glycine max (soybean) genome
955,054,837 (known) nucleotides

Benchmark



Problems

- Open MPI uses polling for blocking MPI calls
 - Excessive CPU usage while waiting for a message
 - Possible workaround: replace “MPI_Recv(…)” with

```
received = false;
```

```
MPI_Irecv(..., request, ...);
```

```
do {
```

```
    MPI_Test(request, received, ...);
```

```
    if (received == true) break;
```

```
    else sleep(seconds);
```

```
} while (true)
```

Tested Configurations

- Mac OS X 10.5.x (Intel iMac)
 - Open MPI 1.2.3 (included in default installation)
- Solaris 10 (Sun Blade 1000 workstation)
 - Sun HPC Cluster Tools 8.1 (Open MPI 1.3)
- Linux (hpc-class cluster)
 - Open MPI 1.3.1

MPIPIPE

Questions?