Q1: How do you define “Reproducibility”?  
A1: In terms of MPI, if “message receive orders” consistent across different executions, I define it as “Reproducible”

Q2: Where reproducibility has been or could be an issue? 
A2: Without MPI reproducibility, application developers will spend more time for debugging

Typical MPI non-reproducible application:

```c
MPI_Irecv(..., MPI_ANY_SOURCE, ...);
while(1) {
    MPI_Test(flag);
    if (flag) {
        <computation>
        MPI_Irecv(..., MPI_ANY_SOURCE, ...);
    }
}
```

Non-reproducible message receive:

1st execution: \((a+b)+c\)  
2nd execution: \(a+(b+c)\)
CASE STUDY: Monte Carlo Simulation (MCB)

- Non-reproducible control flow
  - Successful run, seg-fault or hang
- Non-reproducible numerical results
  - Floating-point arithmetic is “NOT” necessarily associative
    \[(a+b)+c \neq a+(b+c)\]

Developers need to do debug runs until the same bug manifests

Running as intended? Application bugs? Silent data corruption happened?

$\text{diff result\_run1.out result\_run2.out}$

```
result\_run1.out:< IMC E_RR_total
-3.3140234409e-05 -8.302693774e-08 2.9153322360e-08 -4.8198506756e-06 2.3113821822e-06
result\_run2.out:> IMC E_RR_total
-3.3140234410e-05 -8.302693776e-08 2.9153322360e-08 -4.8198506757e-06 2.3113821821e-06
```
Q3: What solutions do you envision?

A3: Record-and-replay

- Traces and records message receive orders in a run, and replays the orders in successive runs for debugging
  - Record-and-replay can reproduce a target control flow
  - Developers can focus on debugging a particular control flow

Although record-and-replay has several challenges, this technique can reproduce a particular control flows and numerical results, thereby can reduce cost for debugging non-reproducible MPI applications.