Scalable Tools for Debugging Non-Deterministic MPI Applications

ReMPI: MPI Record-and-Replay tool

Scalable Tools Workshop

August 2nd, 2016

Kento Sato, Dong H. Ahn, Ignacio Laguna, Gregory L. Lee, Martin Schulz and Chris Chambreau

LLNL-PRES-698040

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Debugging large-scale applications is already challenging

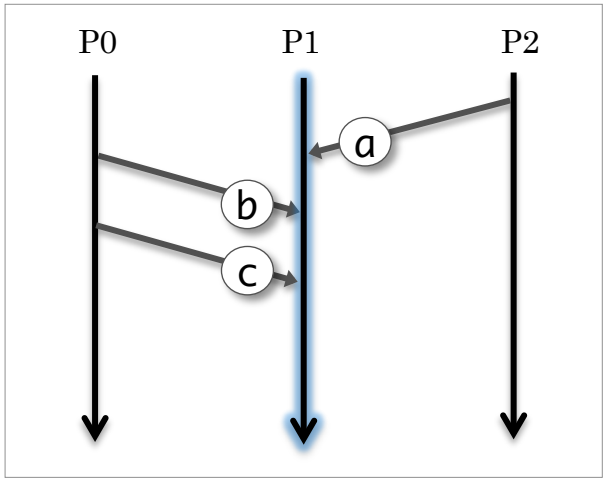
"On average, software developers spend 50% of their programming time finding and fixing bugs."[1]

[1] Source: http://www.prweb.com/releases/2013/1/prweb10298185.htm, CAMBRIDGE, UK (PRWEB) JANUARY 08, 2013

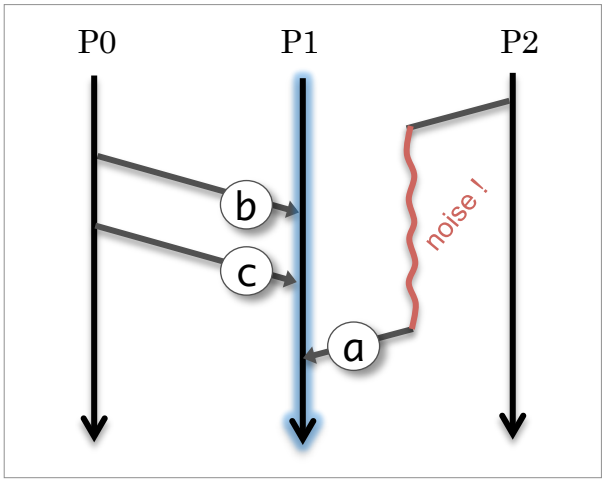
With trends towards asynchronous communication patterns in MPI applications, **MPI non-determinism** will significantly increase debugging cost

What is MPI non-determinism ?

- Message receive orders can be different across executions
  - Unpredictable system noise (e.g. network, system daemon & OS jitter)
- Floating point arithmetic orders can also change across executions



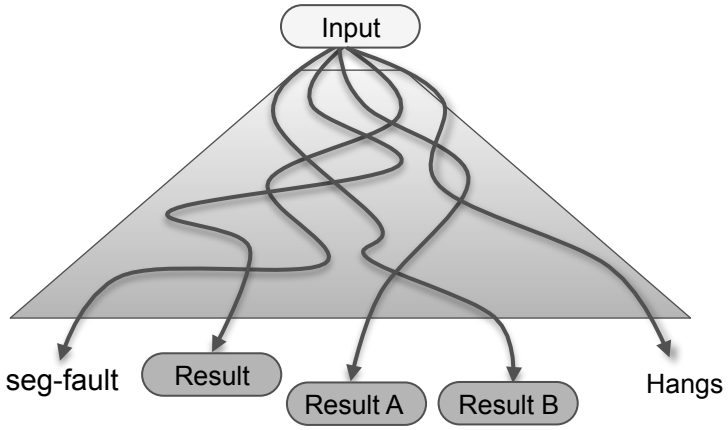
Execution A: (a+b)+c



Execution B: a+(b+c)

Non-determinism also increases debugging cost

- Control flows of an application can change across different runs



- Non-deterministic control flow
  - Successful run, seg-fault or hang
- Non-deterministic numerical results
  - Floating-point arithmetic is non-associative

$$(a+b)+c \neq a+(b+c)$$

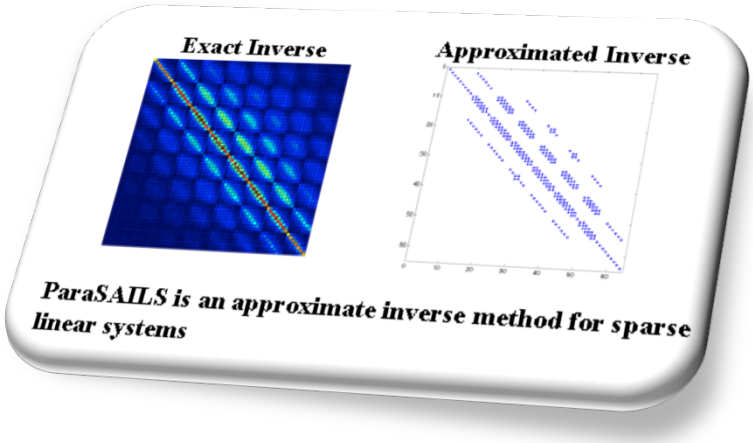
➔ Developers need to do debug runs until the target bug manifests

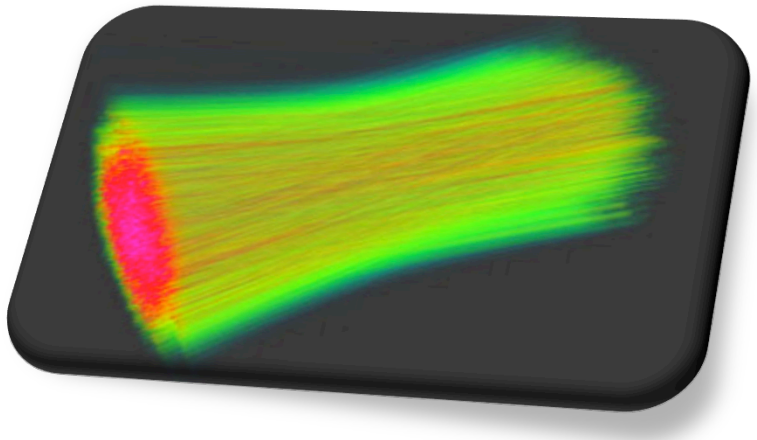
In non-deterministic applications, it's hard to reproduce bugs and incorrect results. It costs excessive amounts of time for "**reproducing**" target bugs

Non-deterministic bugs

--- Case study: Pf3d and Diablo/Hypre 2.10.1

- Debugging non-deterministic hangs often cost computational scientists substantial time and efforts





- **Diablo** - hung only once every 30 runs after a few hours
- The scientists spent 2 months in the period of 18 months and gave up debugging it

  Hypre is an MPI-based library for solving large, sparse linear systems of equations on massively parallel computers

- **Pf3d** – hung only when scaling to half a million MPI processes
- The scientists refused to debug for 6 months …

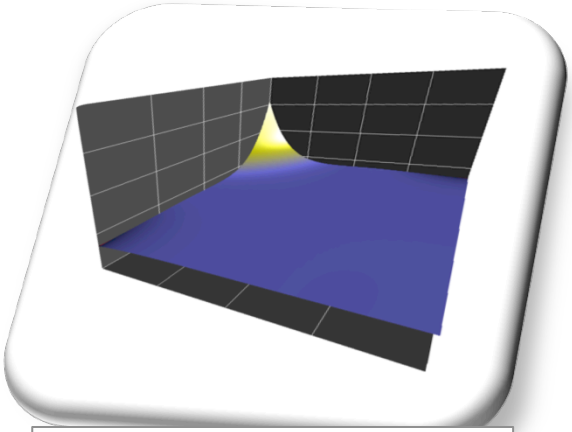
Non-deterministic numerical result

--- Case study: "Monte Carlo Simulation" (MCB)

- CORAL proxy application
- MPI non-determinism

Table 1: Catalyst Specification

Nodes	304 batch nodes
CPU	2.4 GHz Intel Xeon E5-2695 v2 (24 cores in total)
Memory	128 GB
Interconnect	InfiniBand QDR (QLogic)
Local Storage	Intel SSD 910 Series (PCIe 2.0, MLC)



MCB: Monte Carlo Benchmark

Final numerical results are different between 1st and 2nd run

```
$ diff result_run1.out result_run2.out
result_run1.out:< IMC E_RR_total  -3.3140234 09e-05  -8.3026937 74e-08  2.9153322360e-08  -4.8198506 56e-06  2.3113821 22e-06
result_run2.out:> IMC E_RR_total  -3.3140234 10e-05  -8.3026937 76e-08  2.9153322360e-08  -4.8198506 57e-06  2.3113821 21e-06
```

09e-05 10e-05 | 74e-08 76e-08 | 56e-06 57e-06 | 22e-06 21e-06

* The source was modified by the scientist to demonstrate the issue in the field

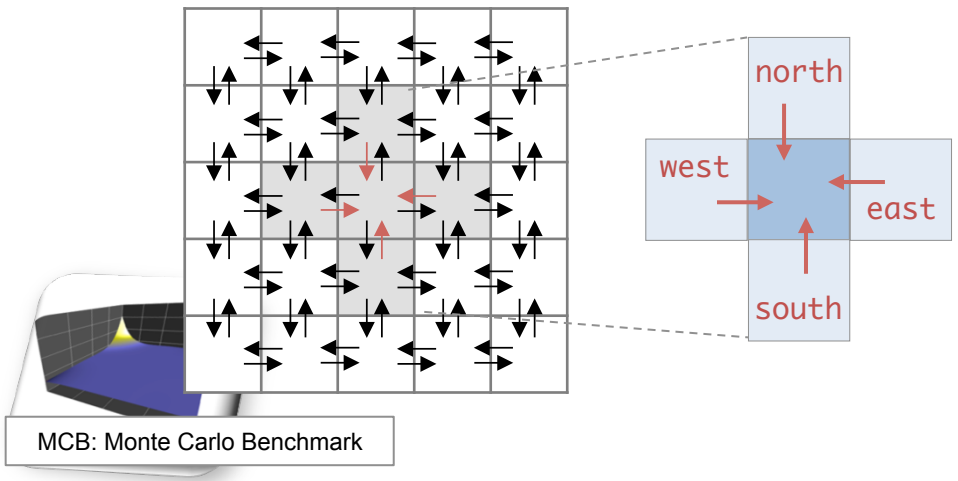
Why MPI non-determinism occurs ?

- It's typically due to communication with MPI_ANY_SOURCE
- In non-deterministic applications, each process doesn't know which rank will send message
- Messages can arrive in any order from neighbors ➔ inconsistent message arrivals

MPI_ANY_SOURCE communication

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

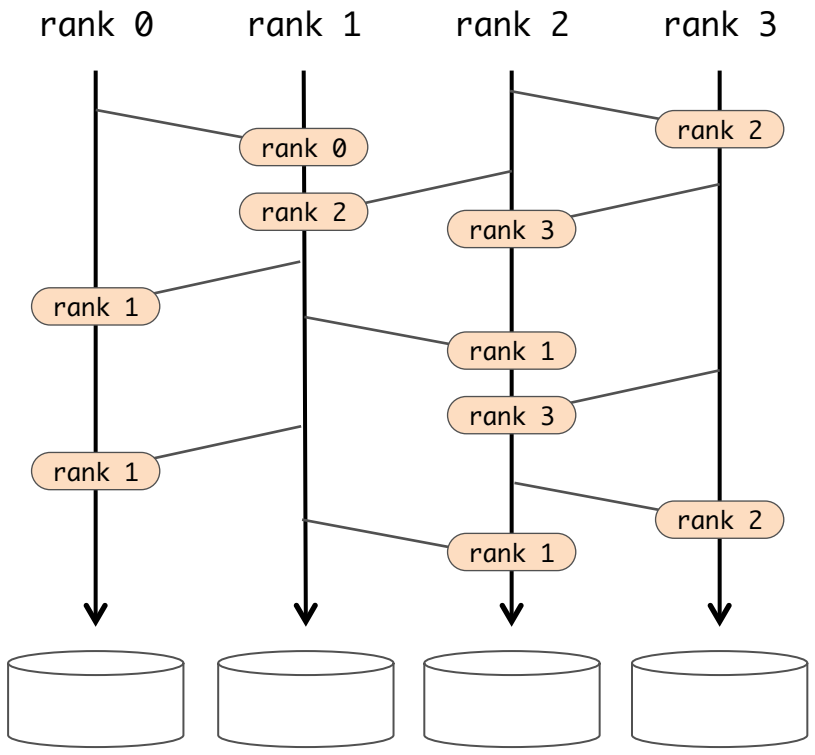
Communications with neighbors



ReMPI can reproduce message matching

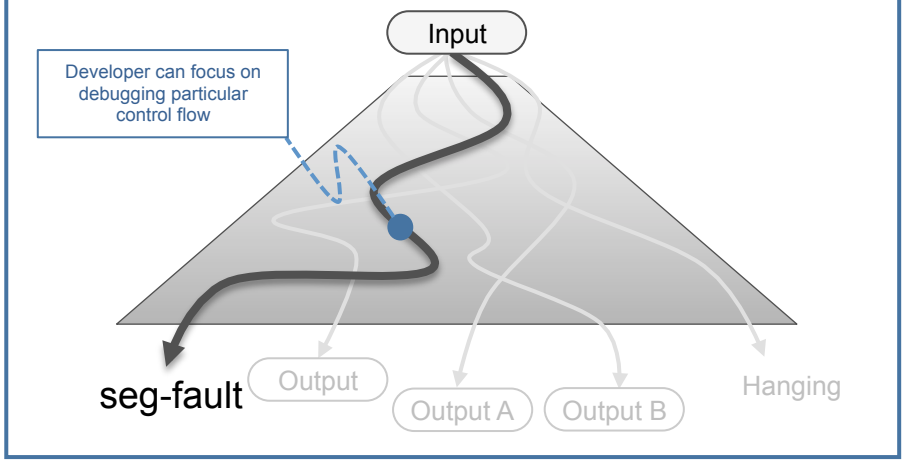
- ReMPI can reproduce message matching by using record-and-replay technique

Record-and-replay



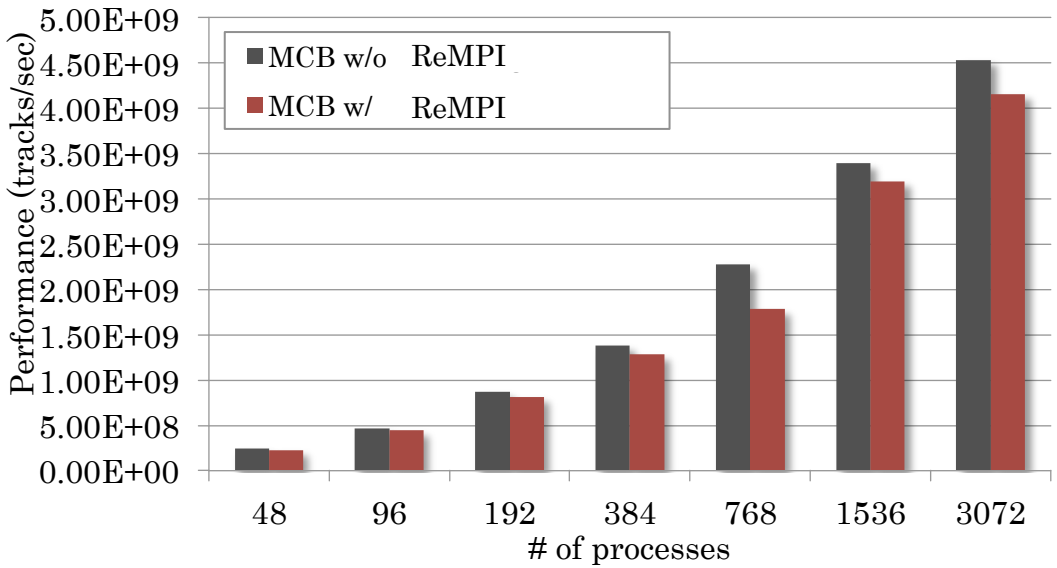
- Traces, records message receive orders in a run, and replays the orders in successive runs for debugging
  - Record-and-replay can reproduce a target control flow
  - Developers can focus on debugging a particular control flow

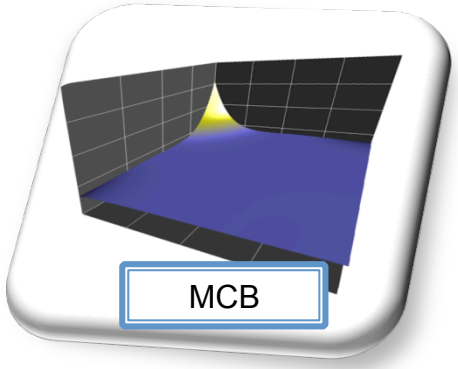
Debugging a particular control flow in replay



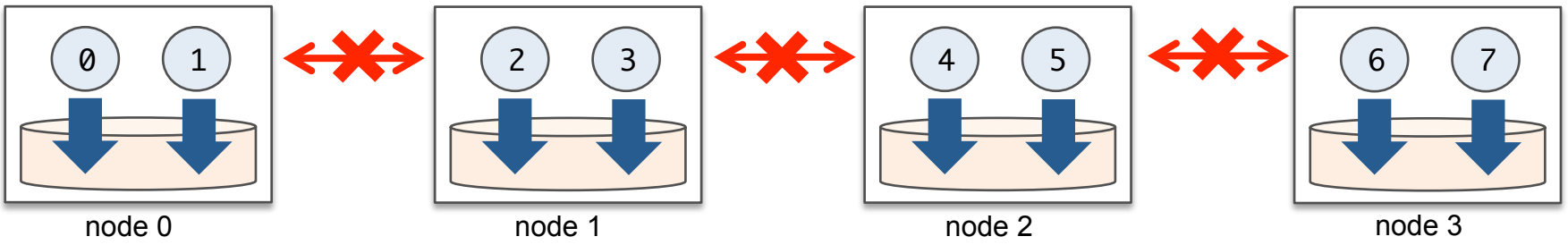
Record overhead to performance

- Performance metric: how many particles are tracked per second



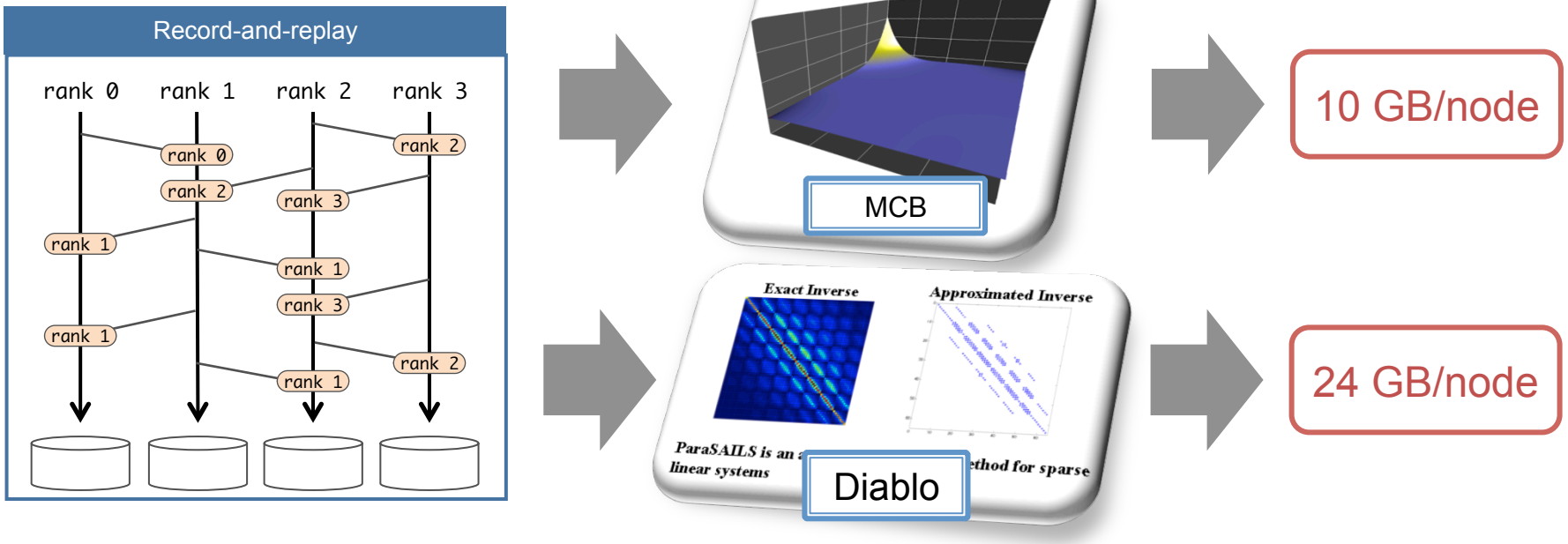


- ReMPI becomes scalable by recording to local memory/storage
  - Each rank independently writes record → No communication across MPI ranks



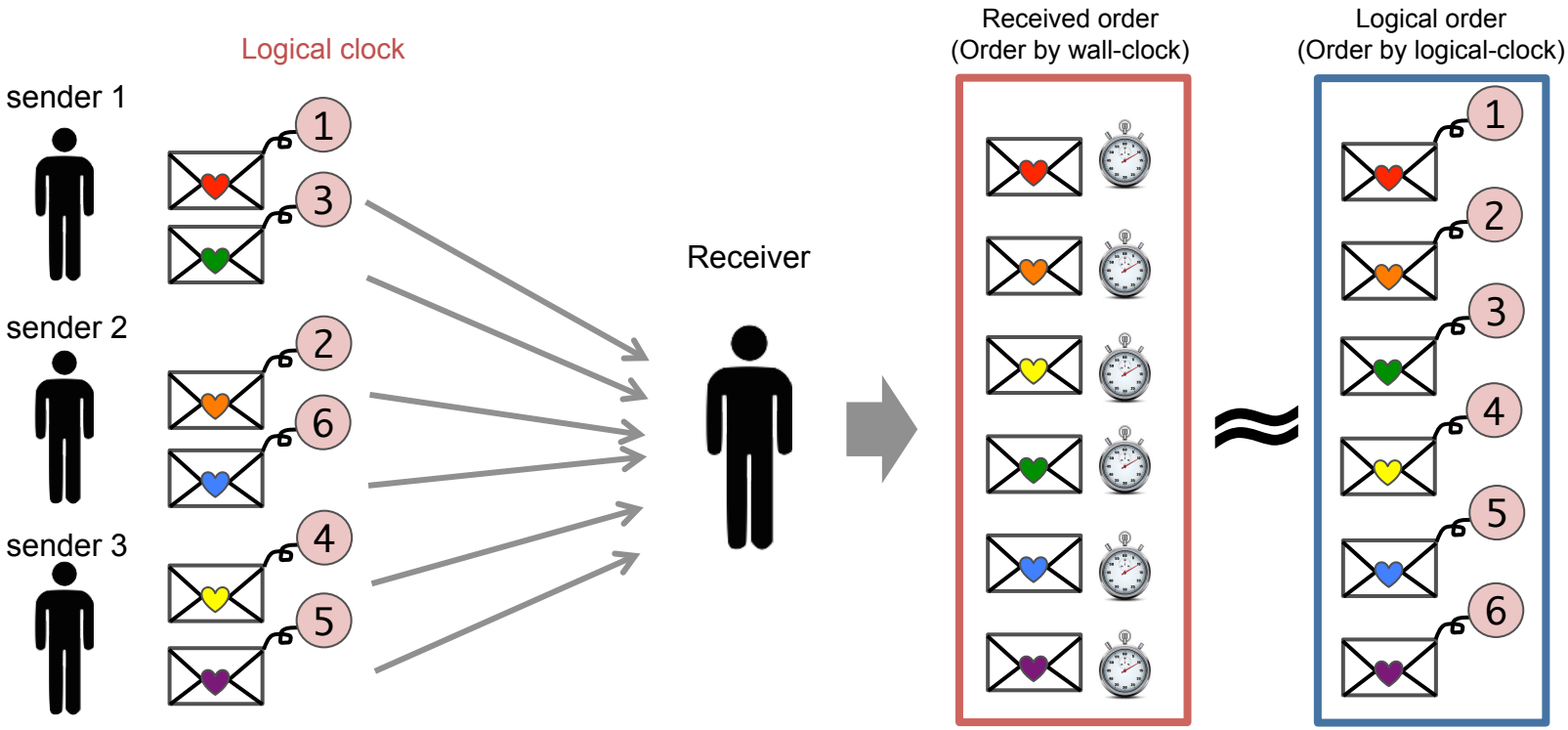
Record-and-replay won't work at scale

- Record-and-replay produces large amount of recording data
  - Over "10 GB/node" per day in MCB
  - Over "24 GB/node" per day in Diablo
- For scalable record-replay with low overhead, the record data must fit into local memory, but capacity is limited
  - Storing in shared/parallel file system is not scalable approach
  - Some systems may not have fast local storage



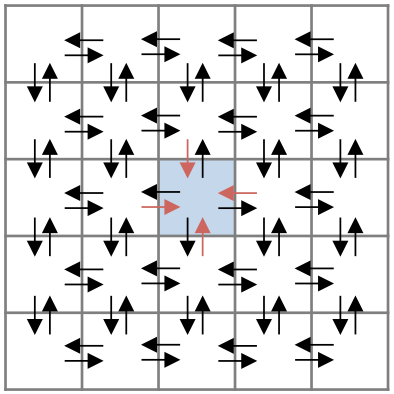
Challenges Record size reduction for scalable record-replay

Clock Delta Compression (CDC)

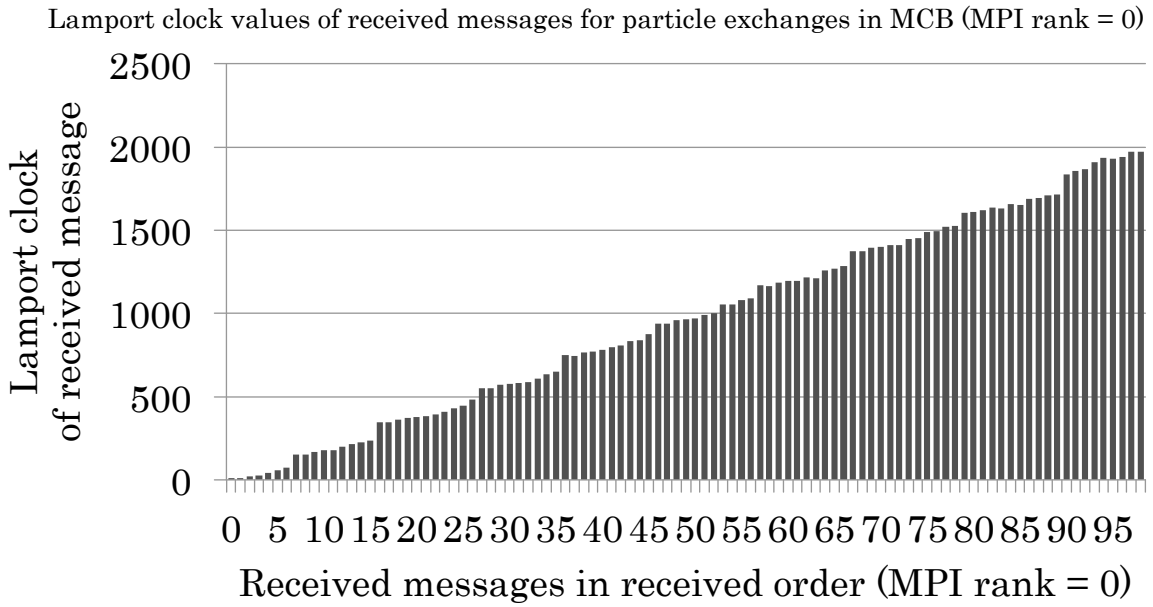


Logical clock vs. wall clock

" The global order of messages exchanged among MPI processes are very similar to a logical-clock order (e.g., Lamport clock) "

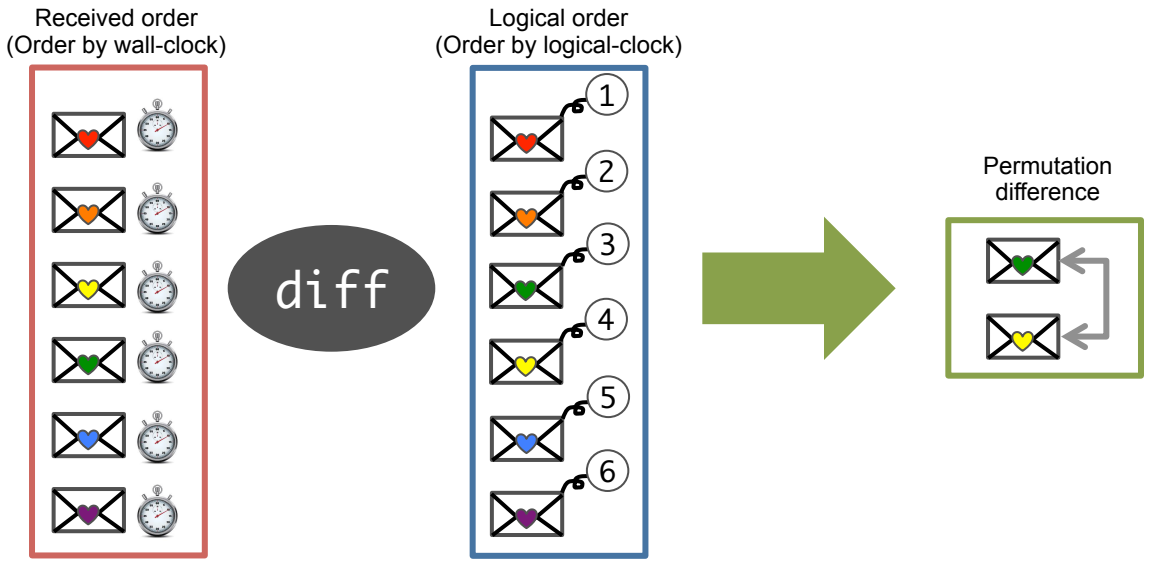


Each process frequently exchanges messages with neighbors



Clock Delta Compression (CDC)

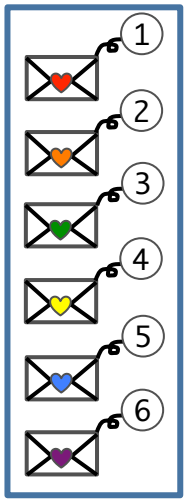
- Our approach, clock delta compression, only records the difference between received order and logical order instead of recording entire received order

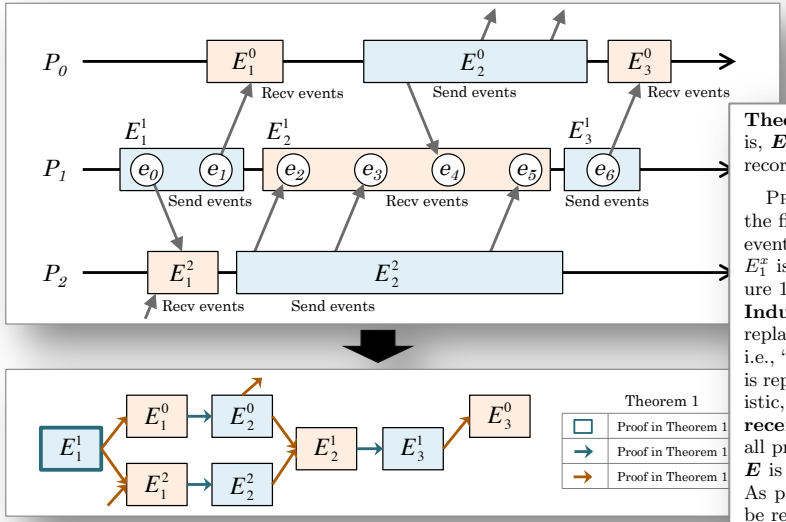


Logical clock order is reproducible [1]

- Logical-clock order is always reproducible, so CDC only records the permutation difference

Logical order
(Order by logical-clock)





Theorem 1. CDC can correctly replay message events, that is, $\boldsymbol{E} = \hat{\boldsymbol{E}}$ where $\boldsymbol{E}$ and $\hat{\boldsymbol{E}}$ are ordered sets of events for a record and a replay mode.

PROOF (MATHEMATICAL INDUCTION). **(i) Basis**: Show the first send events are replayable, i.e., $^{\forall}x$ s.t. "E_1^x is send events" $\Rightarrow$ "E_1^x is replayable". As defined in Definition 7.(i) E_1^x is deterministic, that is , E_1^x is always replayed. In Figure 12, E_1^1 is deterministic, that is, is always replayed. **(ii) Inductive step for send events**: Show send events are replayable if the all previous message events are replayed, i.e., "$^{\forall}\boldsymbol{E} \rightarrow E$ s.t. $\boldsymbol{E}$ is replayed, E is send event set" $\Rightarrow$ "E is replayable". As defined in Definition 7.(ii), E is deterministic, that is, E is always replayed. **(iii) Inductive step for receive events**: Show receive events are replayable if the all previous message events are replayed, i.e., "$^{\forall}\boldsymbol{E} \rightarrow E$ s.t. $\boldsymbol{E}$ is replayed, E is receive event set" $\Rightarrow$ "E is replayable". As proofed in Proposition 1, all message receives in E can be replayed by CDC. Therefore, all of the events can be replayed, i.e., $\boldsymbol{E} = \hat{\boldsymbol{E}}$. (Mathematical induction processes are graphically shown in Figure 12.) ■

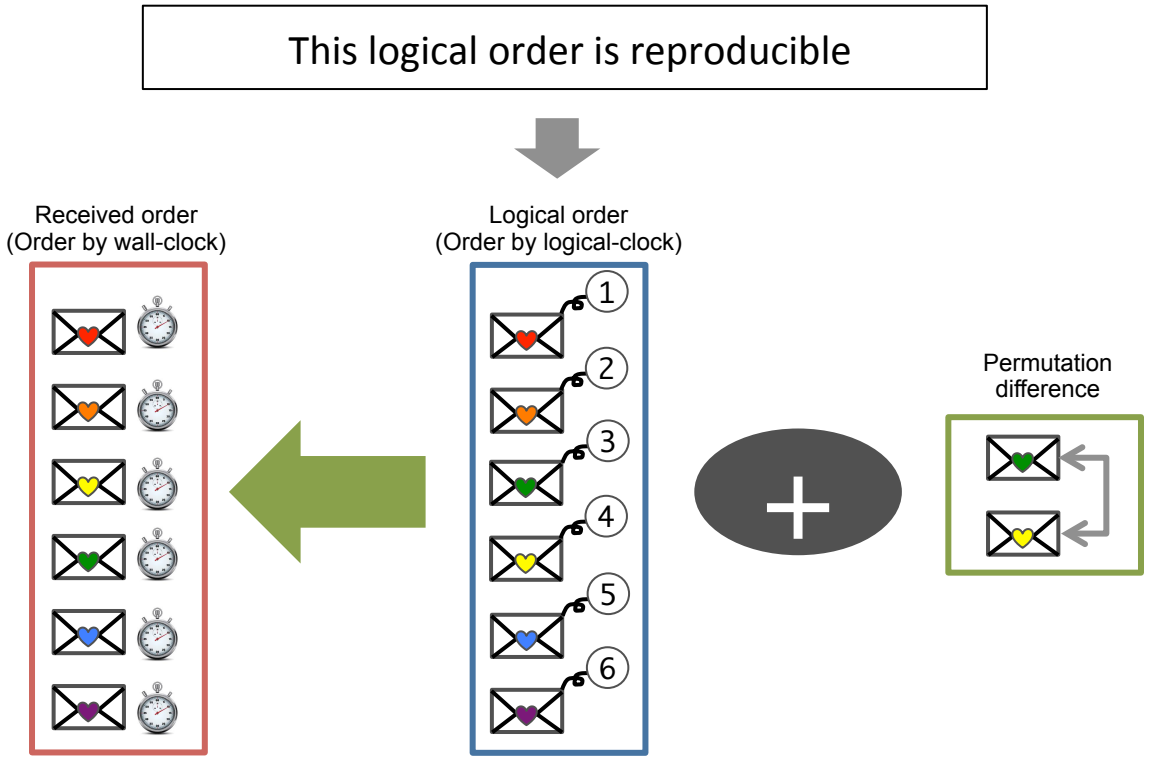
Theorem 2. CDC can replay piggyback clocks.

PROOF. As proved in Theorem 1, since CDC can replay all message events, send events and clock ticking are replayed. Thus, CDC can replay piggyback clock sends. ■

[1] Kento Sato, Dong H. Ahn, Ignacio Laguna, Gregory L. Lee and Martin Schulz, "Clock Delta Compression for Scalable Order-Replay of Non-Deterministic Parallel Applications", In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis 2015 (SC15), Austin, USA, Nov, 2015.

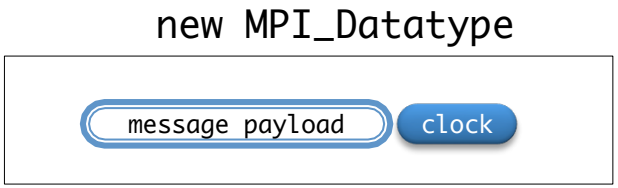
Clock Delta Compression (CDC)

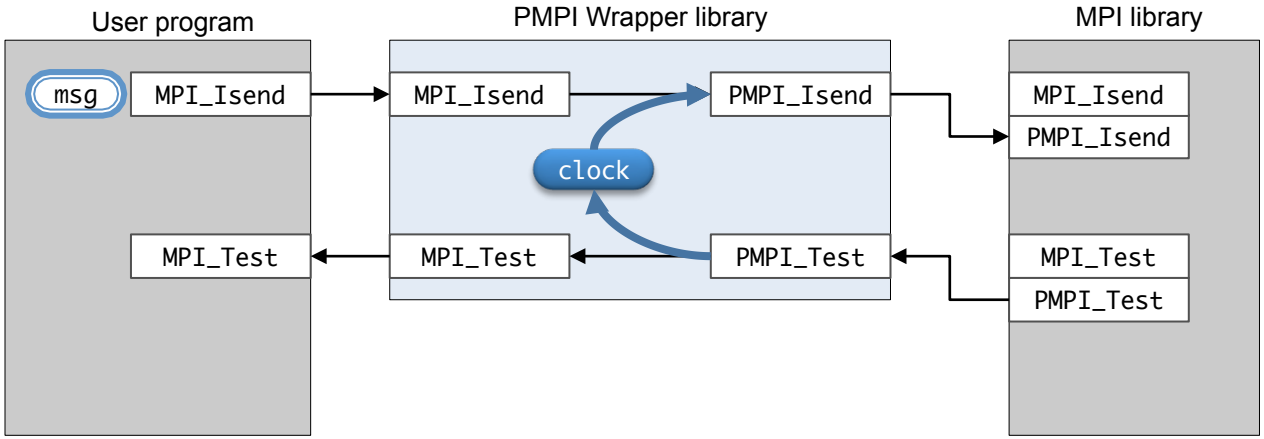
- Our approach, clock delta compression, only records the difference between received order and logical order instead of recording entire received order



Implementation

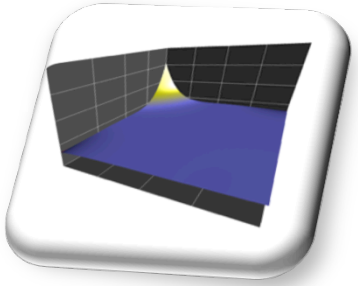
- We use PMPI wrapper
  - Tracing message receive order
  - Clock piggybacking
- Clock piggybacking [1]
  - When sending an MPI message, the PMPI wrapper defines a new MPI_Datatype that combines message payload & clock

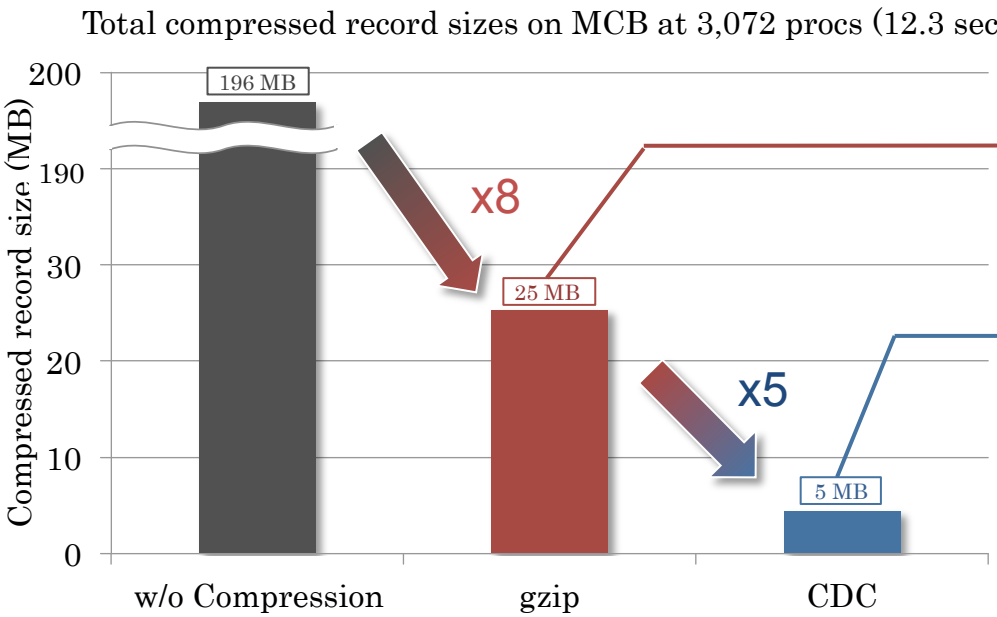




[1] M. Schulz, G. Bronevetsky, and B. R. Supinski. On the Performance of Transparent MPI Piggyback Messages. In Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, pages 194–201, Berlin, Heidelberg, 2008. Springer-Verlag.

Compression improvement in MCB





gzip itself can reduce the original format by 8x

5x more reduction

- For example, if 1GB of memory per node for record-and-replay …
  - w/o compression: 2 hours
  - gzip: 19 hours
  - CDC: 4 days

High compression Compressed size becomes 40x smaller than original size

Summary

- Non-determinism is a common issue in debugging MPI applications
- ReMPI can help to reproduce buggy MPI behaviors with minimum record size