Ongoing efforts on MPI-RMA at Atos

FORMA'22 – 15/06/2022

Marc Sergent,
BDS HPC R&D Software
marc.sergent@atos.net

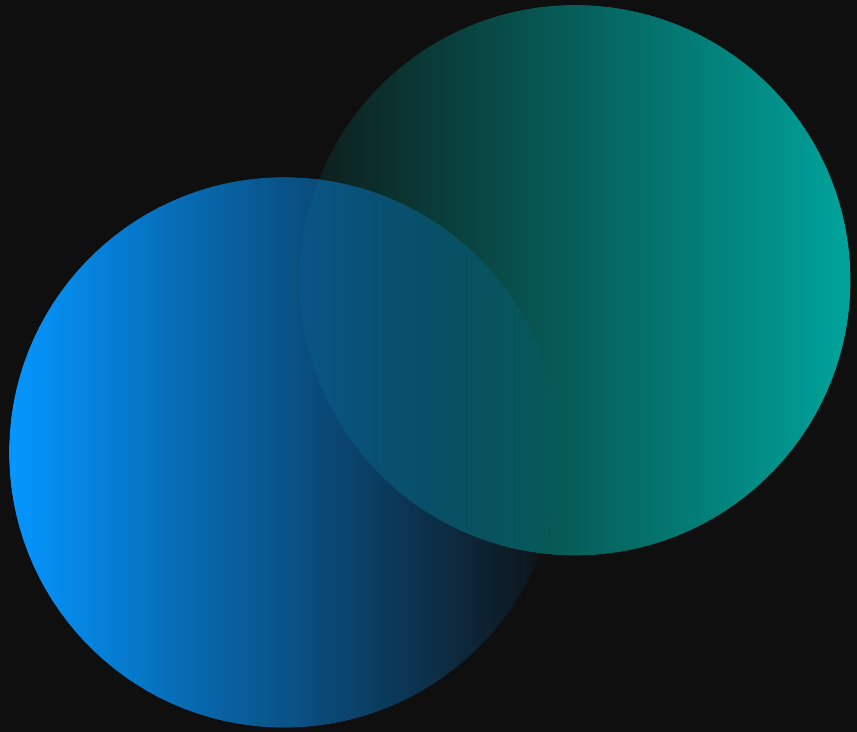
Atos

Outline

1. Lightweight synchronizations: Notified communications
2. Automatic data race detection: RMA-Analyzer
3. Q & A

Atos

1 Lightweight synchronizations: Notified communications

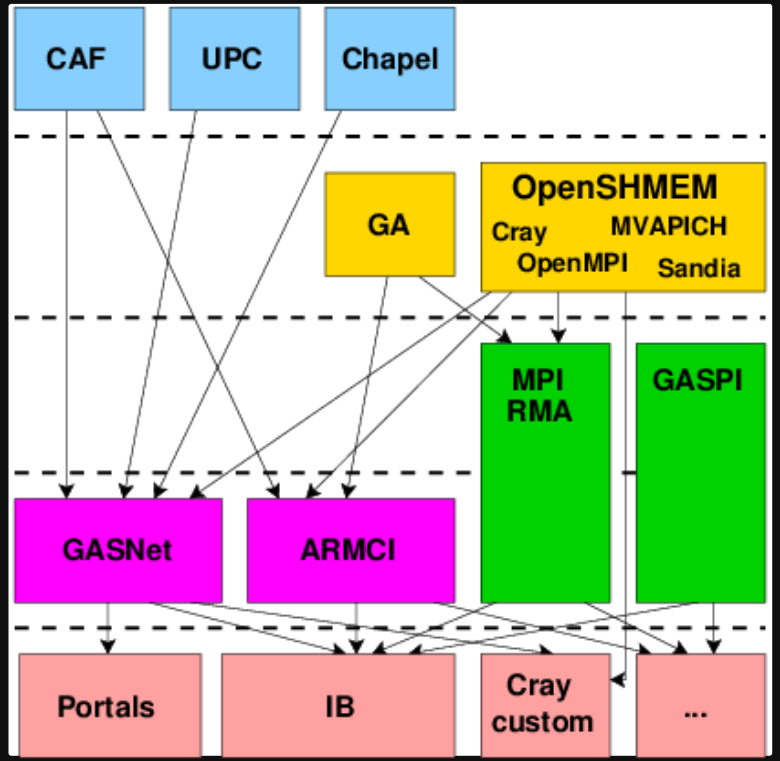


Atos

MPI-RMA as a PGAS runtime

Portability of PGAS runtimes

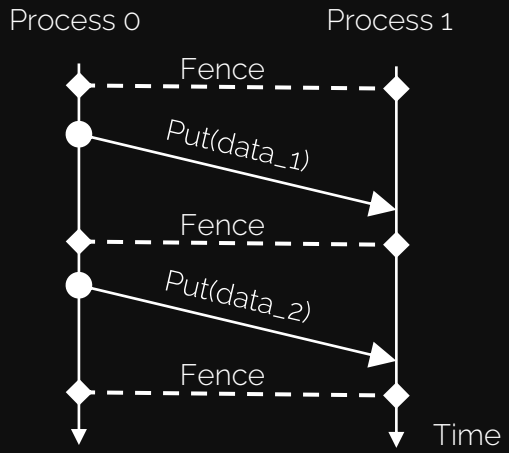
- PGAS languages and libraries uses several runtimes that implement the hardware support for communications
- Can one be used by all of them ?
  What is missing for one to be used by all ?
- MPI-RMA seems a good candidate for this

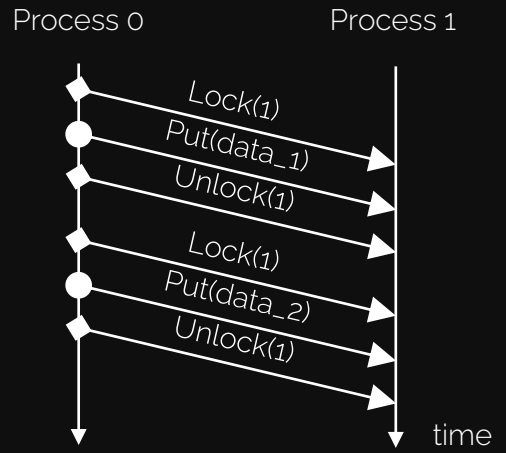


MPI RMA specificities

Ordering and synchronization in MPI-3

- Active Target mode
  - *Fence / Post-Start-Complete-Wait*
  - Receiver **is involved** in the synchronization
- Passive Target mode
  - *Lock(_all)/ unlock(_all)*
  - Receiver **is not involved** in the synchronization

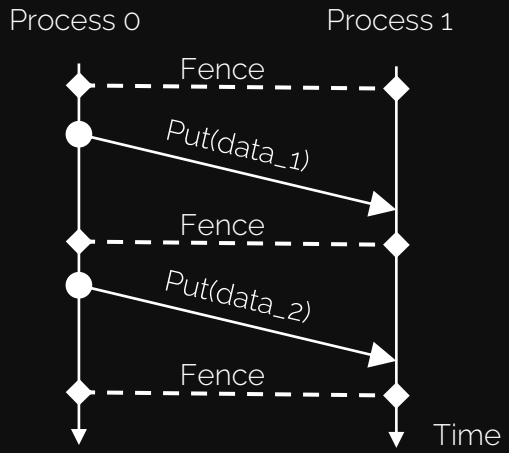


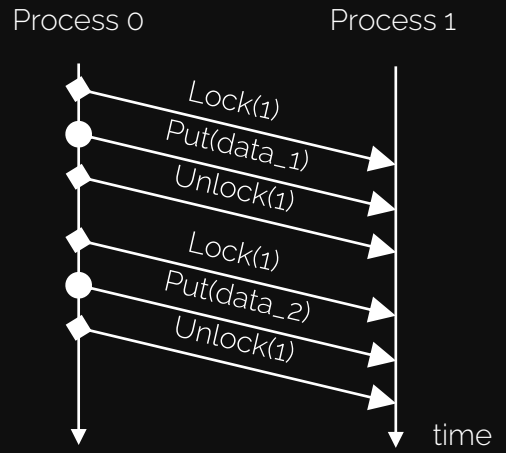


MPI RMA specificities

Ordering and synchronization in MPI-3

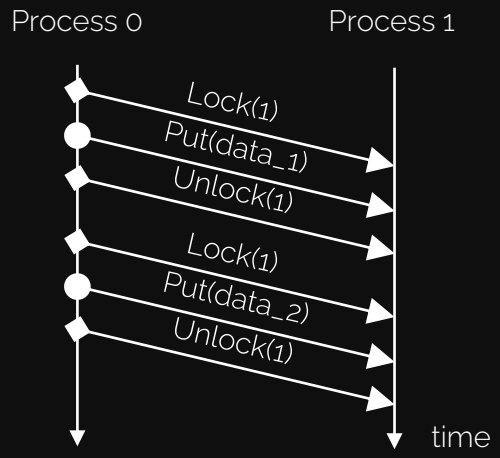
- Active Target mode
  - *Fence / Post-Start-Complete-Wait*
  - Receiver **is involved** in the synchronization
- **Passive Target mode**
  - *Lock(_all)/ unlock(_all)*
  - Receiver **is not involved** in the synchronization





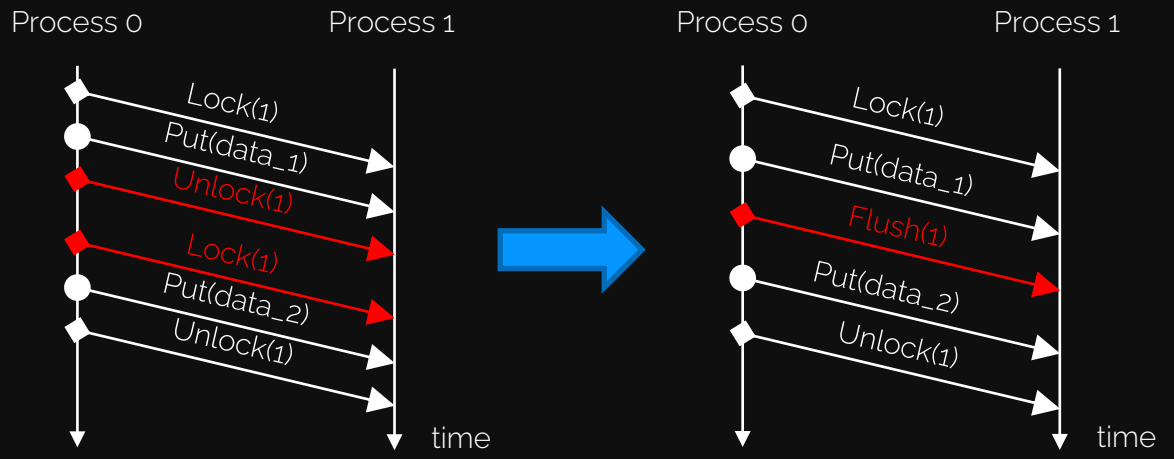
MPI RMA Passive Target mode
Advantages and drawbacks

- Ordering inside a Passive Target epoch with *Flush*
  - At sender's side



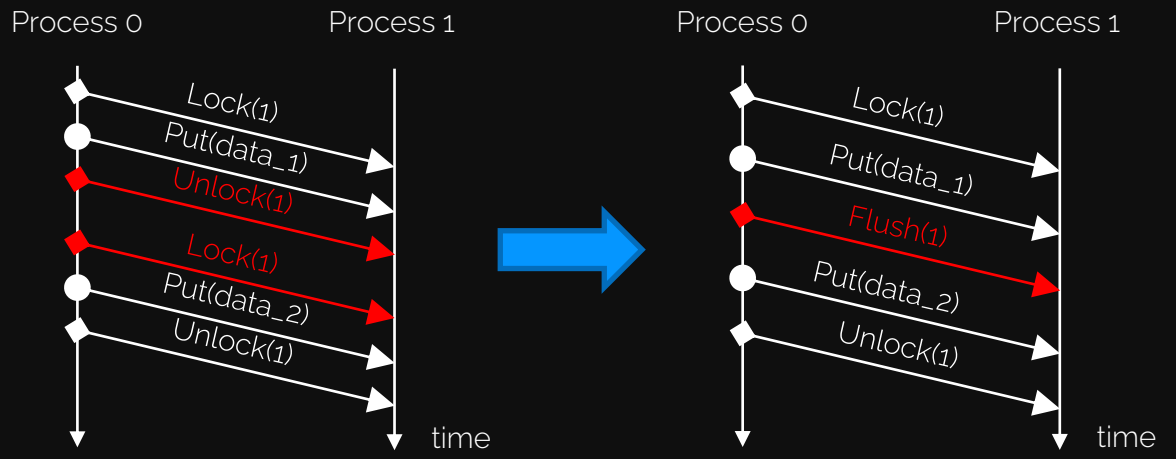
MPI RMA Passive Target mode
Advantages and drawbacks

- Ordering inside a Passive Target epoch with *Flush*
  - At sender's side



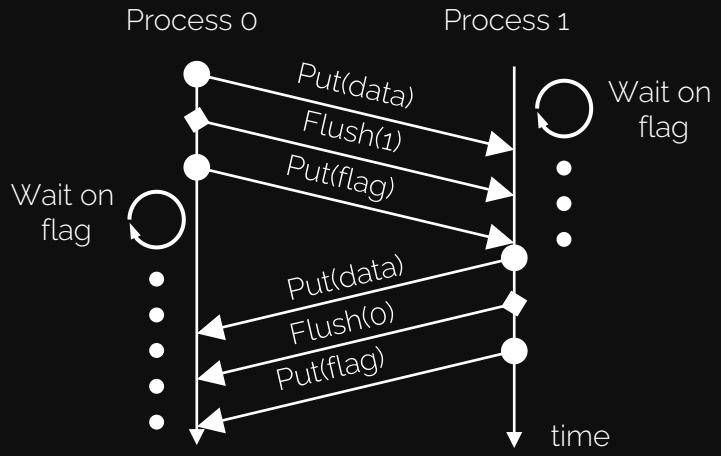
MPI RMA Passive Target mode
Advantages and drawbacks

- Ordering inside a Passive Target epoch with *Flush*
  - At sender's side
- Drawbacks
  - Receiver has no knowledge of when a piece of data has landed locally
    - **Synchronization is missing**



MPI RMA pingpong example
Why MPI-3 is not enough

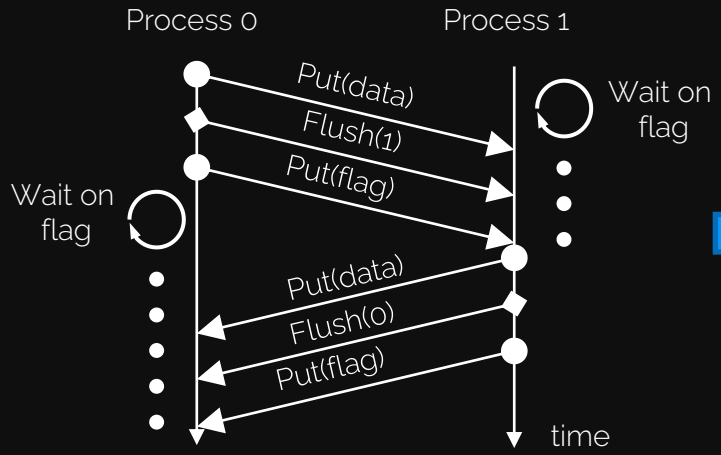
- Pattern for detection of data reception
  - *Put(data) / Flush / Put(flag)*
  - User-implemented waiting algorithm



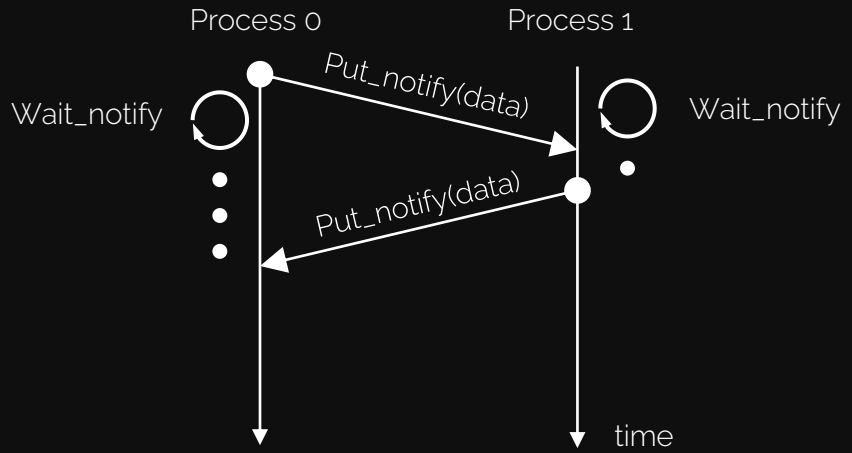
Source: "Remote Memory Access Programming in MPI-3", Hoefler et al.

MPI RMA pingpong example
Why MPI-3 is not enough

- Pattern for detection of data reception
  - *Put(data) / Flush / Put(flag)*
  - User-implemented waiting algorithm



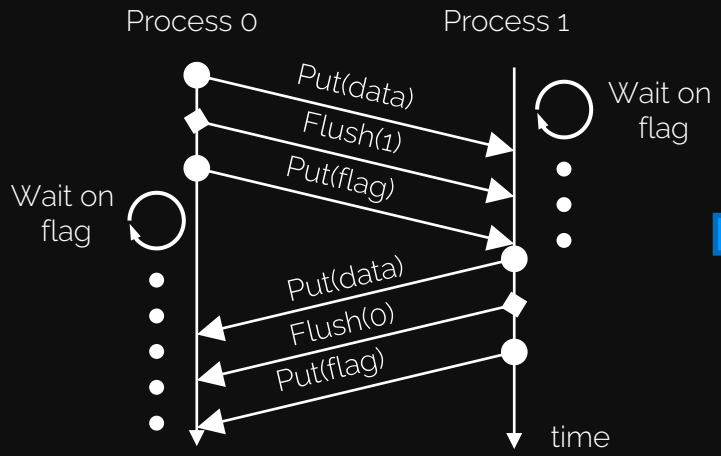
- What users want
  - Embedding flag with the communication
  - Simple waiting semantics



Source: "Remote Memory Access Programming in MPI-3", Hoefler et al.

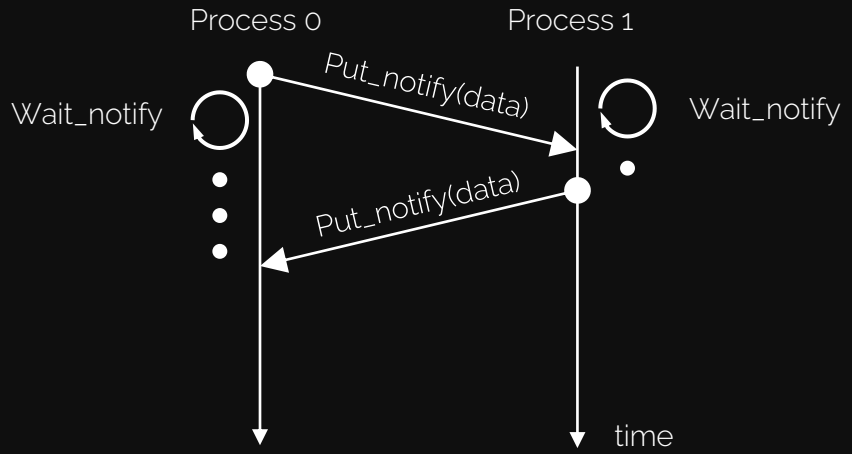
MPI RMA pingpong example
Why MPI-3 is not enough

- Pattern for detection of data reception
  - *Put(data) / Flush / Put(flag)*
  - User-implemented waiting algorithm



- What users want
  - Embedding flag with the communication
  - Simple waiting semantics

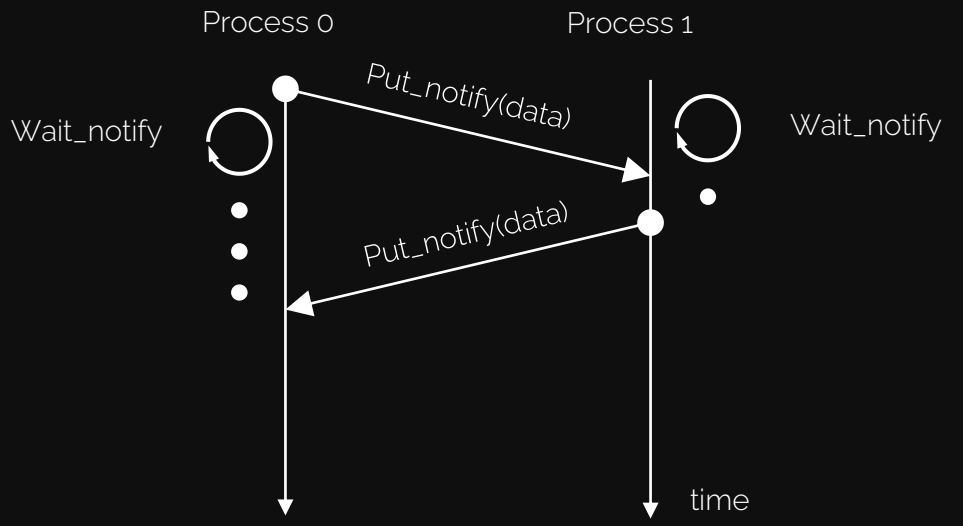
=> **Notified communications**



Source: "Remote Memory Access Programming in MPI-3", Hoefler et al.

Notified communications
Rationale

- Notification flag embedded with the data communication
  - Ensures data + flag ordering
- Optimized waiting mechanisms
  - Implemented inside the runtime
- Existing notifications-like mechanisms
  - GASPI/GPI-2
  - Cray SHMEM / OpenSHMEM
  - UPC
  - ARMCI



Notified communications for MPI
Existing proposals made to the MPI Forum

- **Sync-and-Notify**
  - *Flush_notify / unlock_notify*
  - *Set/Wait_notify(count)*
  - Notification separate from communication operations
    - **Will not improve performance**

Atos

Notified communications for MPI
Existing proposals made to the MPI Forum

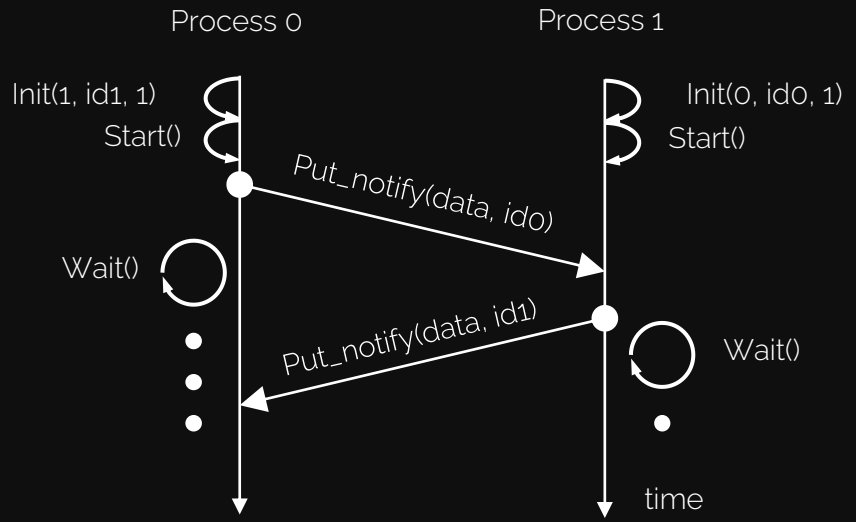
- **Sync-and-Notify**
  - *Flush_notify / unlock_notify*
  - *Set/Wait_notify(count)*
  - Notification separate from communication operations
    - **Will not improve performance**

- **Op-and-Notify**
  - ***Put/Get_notify***
  - *Set/Wait_notify(count)*
  - Only one counter per window
    - **Cannot identify a specific communication**

Notified communications for MPI
Existing proposals made to the MPI Forum

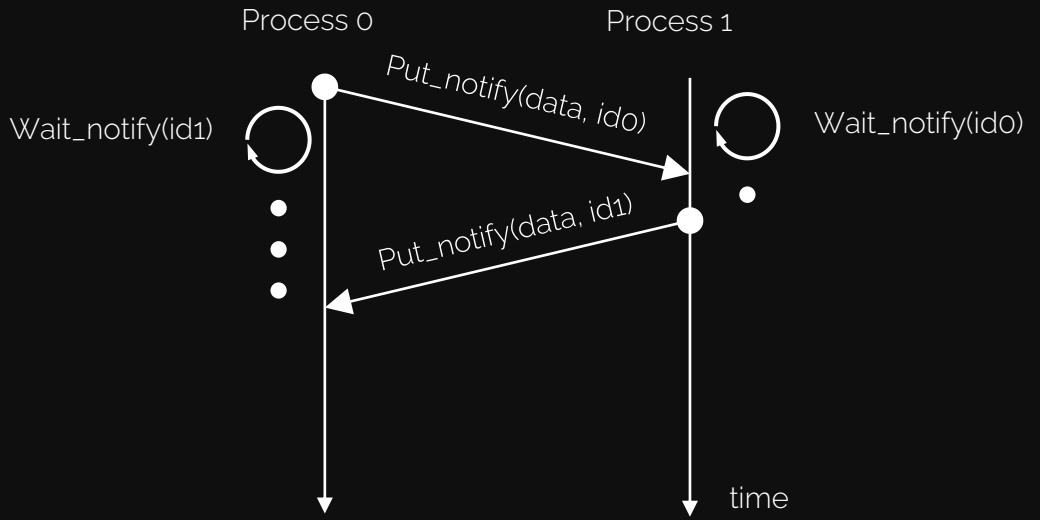
- **Matched Notifications**
  - *Put/Get_notify(id)*
  - **Allow fine-grained synchronizations**
  - Synchronization model
    - *Init/Start/Wait*
    - **Reuse of existing routines**
    - **Requires an implicit synchronization at *Start* or *Wait***

"Notified access: Extending remote memory access programming models for producer-consumer synchronization", Hoefler et al., IPDPS'15



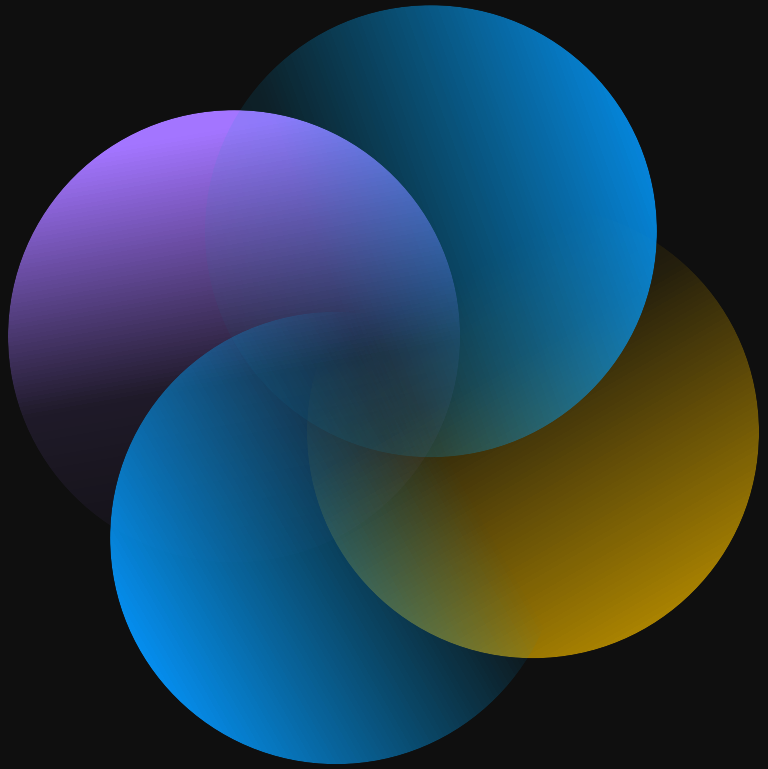
Notified communications for MPI
Our notified communications proposal

- Matched Notifications
  - **Remove *Init/Start/Wait* pattern**
  - **Add *Test/Wait_notify(id)***
  - **Remove implicit synchronization**
  - **Simplified user interface**
  - **Cannot count batch of notifications**
  - **More unexpected behaviors to handle for the programmer**



"Efficient Notifications for MPI One-Sided Applications", Sergent et al., EuroMPI'19

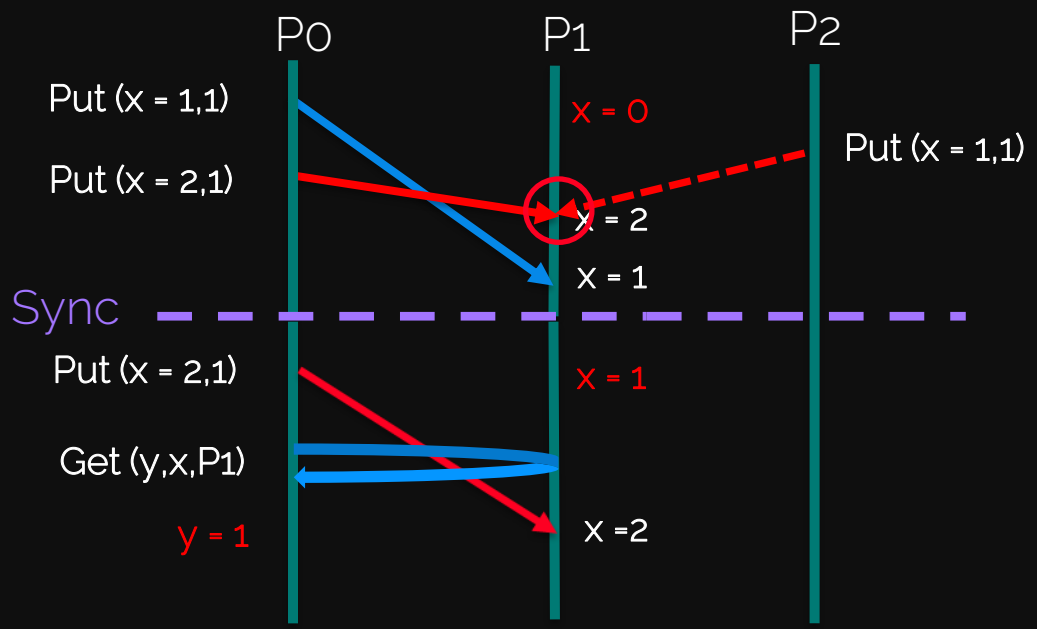
2. Automatic data race detection: RMA-Analyzer



MPI-RMA Programming
Major challenges

Due to the asynchronous nature of RMA operations:

- **Completion:** The completion of the RMA communication operations is not known
- **Ordering**: MPI provides no ordering guarantees for RMA operation
- **Atomicity**: Regular MPI_Put and MPI_Get operations are non-atomic



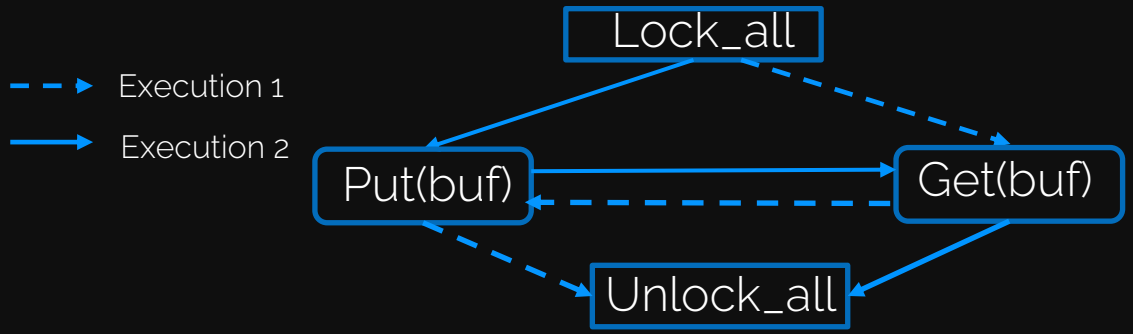
Data Race in MPI-RMA

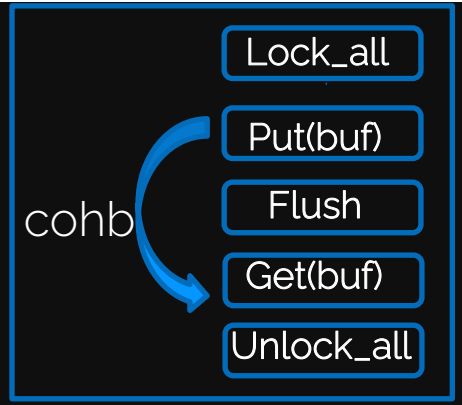
What is a Data Race ?

- Data-race = anomaly of concurrent accesses by two or more processes to a shared variable and at least one is a WRITE
- Two events *a* and *b* are concurrent within an epoch when they are not ordered by Consistency Order Happens-before (||cohb)

source: "Remote Memory Access Programming in MPI-3", Hoefler et al

$$a \parallel_{cohb} b \iff a \not\xrightarrow{cohb} b \wedge b \not\xrightarrow{cohb} a \qquad a \xrightarrow{cohb} b \iff a \xrightarrow{hb} b \wedge a \xrightarrow{co} b$$





Data Race in MPI-RMA Programs

Examples of Data Race errors

P0 (Origin)	P1 (Target)
	Window location X
Win_lock_all	Win_lock_all
Put(buf, 1, X)	
buf = ..	
Win_unlock_all	Win_unlock_all

Consistency error in a process

P0 (Origin)	P1 (Target)
	Window location X
Win_lock_all	Win_lock_all
Put(_, 1, X)	**Get(X, 0, _)**
Win_unlock_all	Win_unlock_all

Consistency error between two processes

P0 (Origin)	P1 (Target)	P2 (Origin)
	Window location X	
Win_lock_all	Win_lock_all	Win_lock_all
Put(_, 1, X)		**Put(_, 1, X)**
Win_unlock_all	Win_unlock_all	Win_unlock_all

Consistency error among several processes

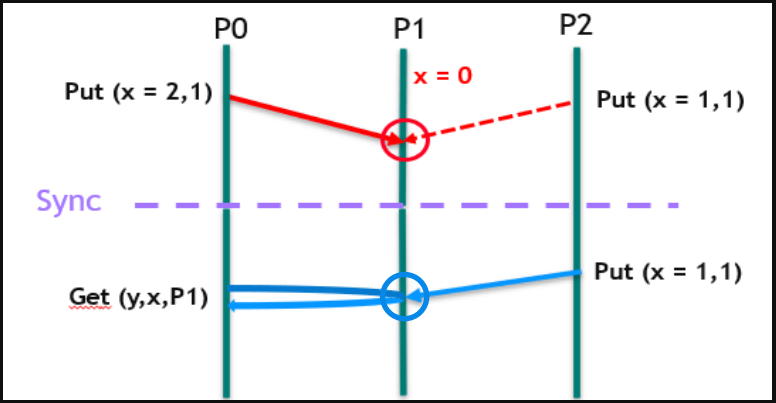
Atos

Dynamic Data race Detection for MPI-RMA Programs

RMA Operations Compatibility

		ORIGIN		TARGET		LOAD	STORE
		GET	PUT	GET	PUT		
O	GET	x	x	x	x	x	x
	PUT	x	✓	✓	x	✓	x
T	GET	x	✓	✓	x	✓	x
	PUT	x	x	x	x	x	x
LOAD		x	✓	✓	x	-	-
STORE		x	x	x	x	-	-

Compatibility of RMA operations and local load/store accesses



How to read the table :

- Read/Write on the same memory location

✕ = Consistency order between the two operations is not guaranteed

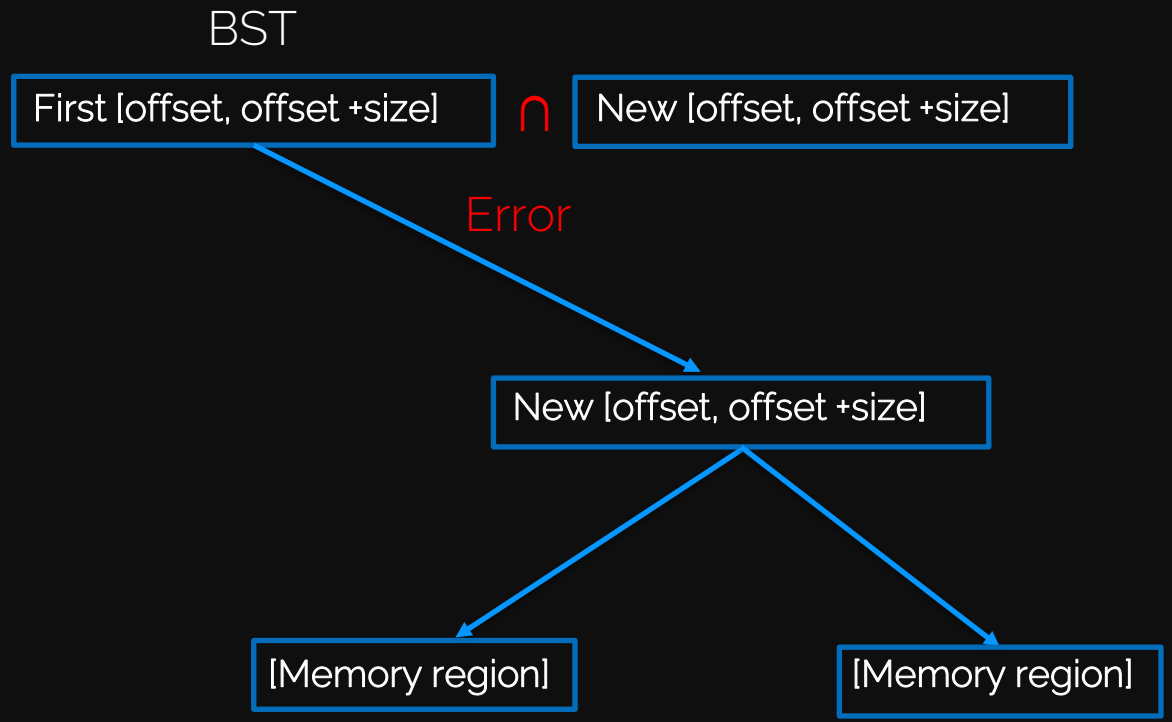
✔= Consistency order ok

- We separate operations from/to the origin and target processes
  - O Get = local write
  - O Put = local read
  - T Get = Remote read (RMA Read)
  - T Put = Remote write (RMA Write)

Dynamic Data Race Detection for MPI-RMA Programs

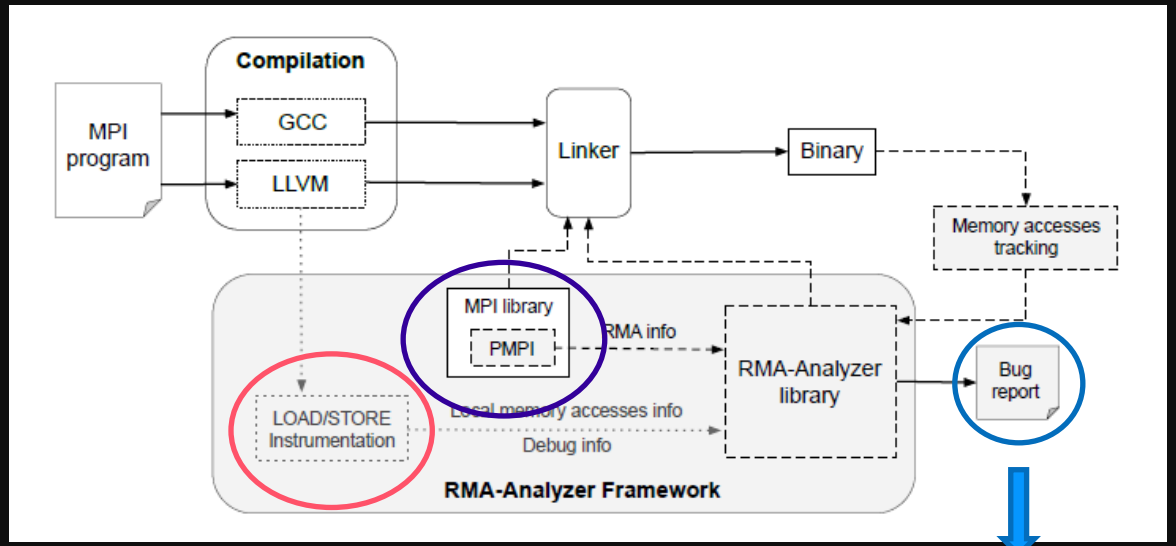
Data Race Detection Algorithm

- Each process creates a BST
- Store the memory region as an interval of [offset, offset+size] + Access Rights
- If the memory access overlaps a stored memory access
  - At least one of the accesses is a Write
    - Error STOP the program
- Reset the BST between epochs



Dynamic Data Race Detection for MPI-RMA Programs

RMA-Analyzer Framework Overview



- PMPI
  To Collect RMA Information

- LLVM Pass
  To instrument local accesses

"Static and Dynamic Data Race Detection for MPI-RMA Programs", Ait Kaci et al., EuroMPI'21

P0 (Origin)	P1 (Target)	P2 (Origin)
	Window location X	
Win_lock_all	Win_lock_all	Win_lock_all
Put(_, 1, X)		Put(_, 1, X)
Win_unlock_all	Win_unlock_all	Win_unlock_all

```
$ mpirun -np 3 ./rr_put_put
[RMA-ANALYZER Process 1] Error when inserting memory
access of type RMA_WRITE from file
remote_remote/rr_put_put.c at line 35 with already
inserted access of type RMA_WRITE from file
remote_remote/rr_put_put.c at line 35.
The program will be exiting now with MPI_Abort.
```


Atos

Thank you for attending

Questions ?

Thank you

For more information please contact:
marc.sergent@atos.net
dl-rd-drim@atos.net