Semantics of Remote Direct Memory Access

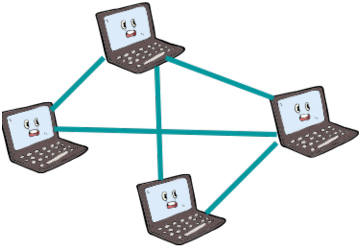
OOPSLA 24

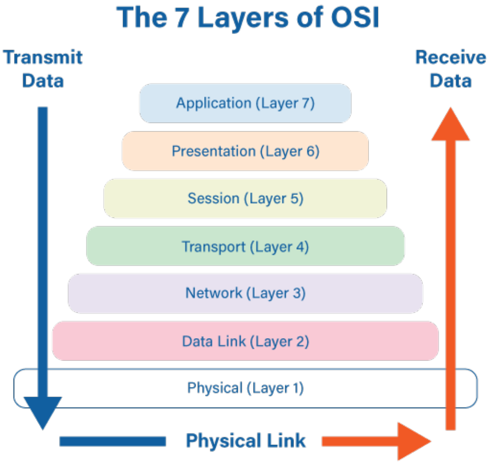
Guillaume Ambal[1], Brijesh Dongol[2], Haggai Eran[3], Vasileios Klimis[4], Ori Lahav[5], Azalea Raad[1]

[1]Imperial College London, [2]University of Surrey, [3]NVIDIA, [4]Queen Mary University of London, [5]Tel Aviv University

October 25, 2024

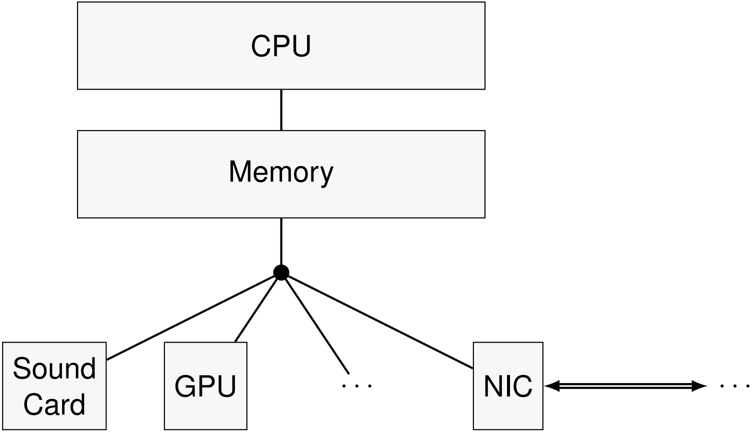
Problem: slow communication



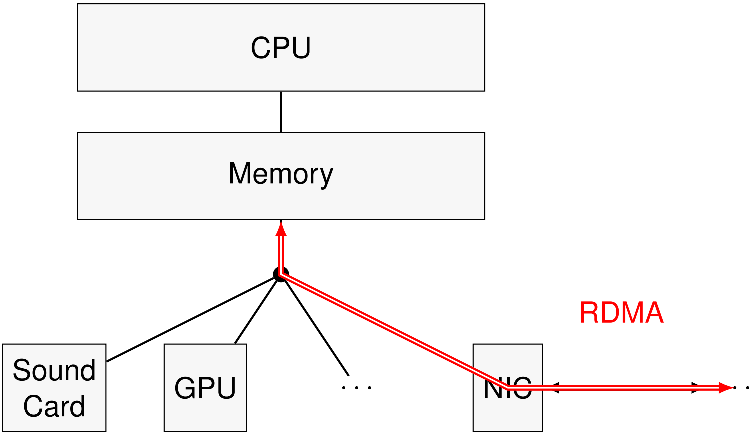


Latency : $\sim$ ms

Direct Memory Access



Direct Memory Access



Remote Direct Memory Access

RDMA

- Data-transfer protocol for High-Performance Computing
- Latency : $\sim \mu$s
- Recently becoming widespread

Remote Direct Memory Access

RDMA

- Data-transfer protocol for High-Performance Computing
- Latency : $\sim \mu$s
- Recently becoming widespread

Our work: formalizing RDMA semantics (assuming TSO for the CPU)

- Operational Semantics
- Declarative Semantics
- Proved Equivalent
- Empirical Validation

Model and Syntax

Model:

- network with several nodes (n_1, n_2, ...)
- each nodes can have several threads
- each thread has a Queue-Pair (communication channel) towards each other node

Syntax

$$
\begin{array}{rlr}
\text{Mem Locs} & x, y, z, \ldots & \\
\text{Commands} & c ::= \quad x := e \mid \texttt{mfence} & \leftarrow \text{usual TSO} \\
& \qquad \mid x^n := y \mid x := y^n \mid \texttt{poll}(n) & \leftarrow \text{RDMA}
\end{array}
$$

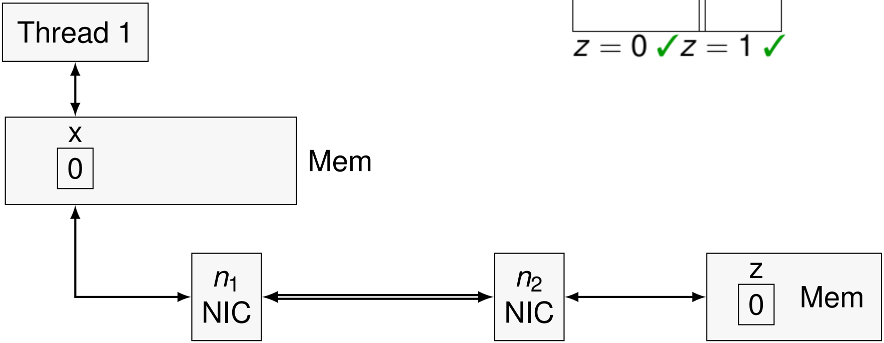
RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	
$z = 0$ ✓	$z = 1$ ✓

RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	

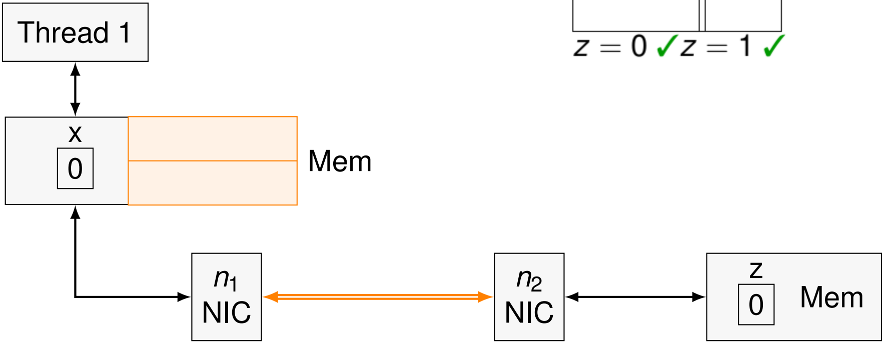
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	

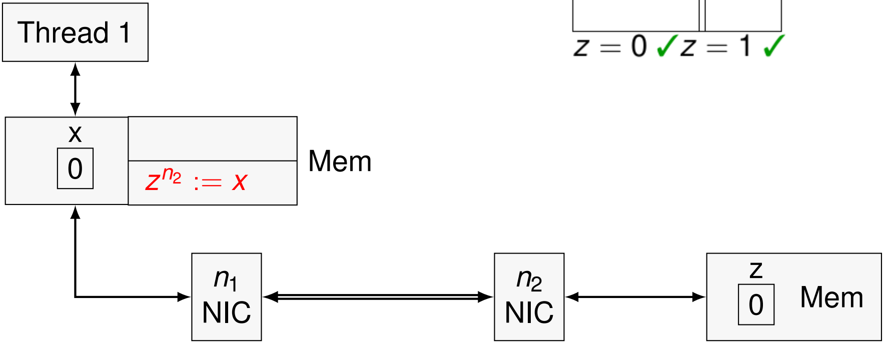
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $x := 1$	

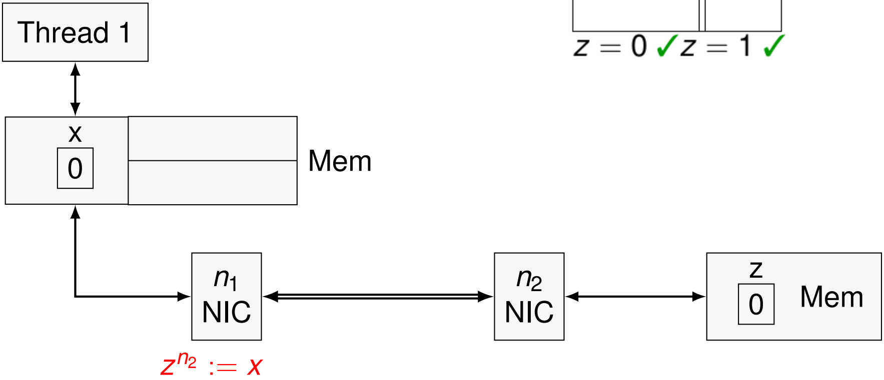
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $x := 1$	

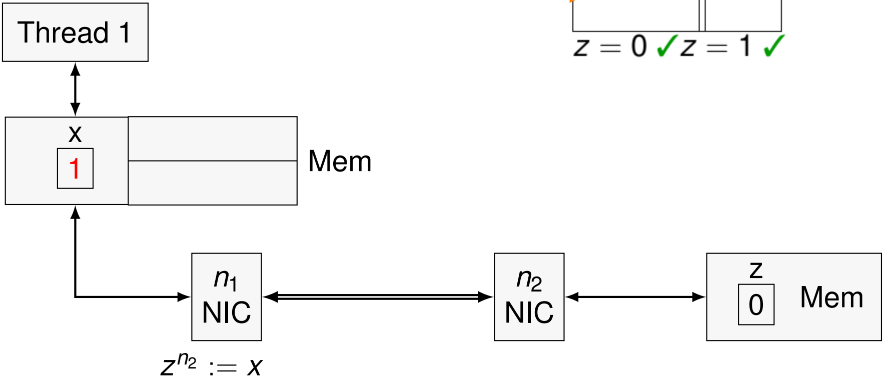
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$	
$x := 1$	

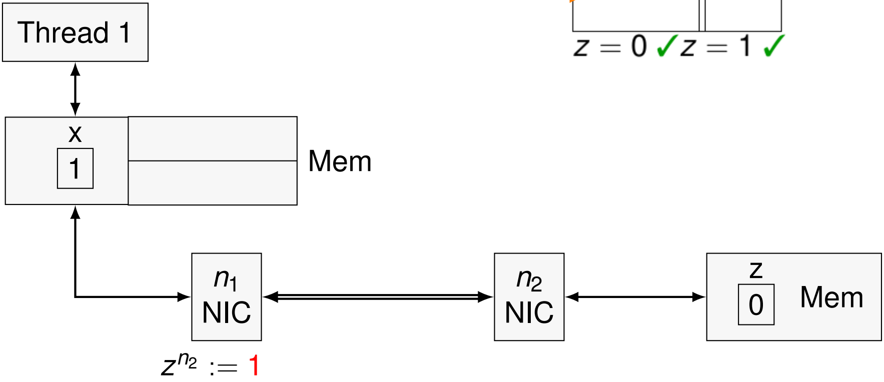
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $x := 1$	

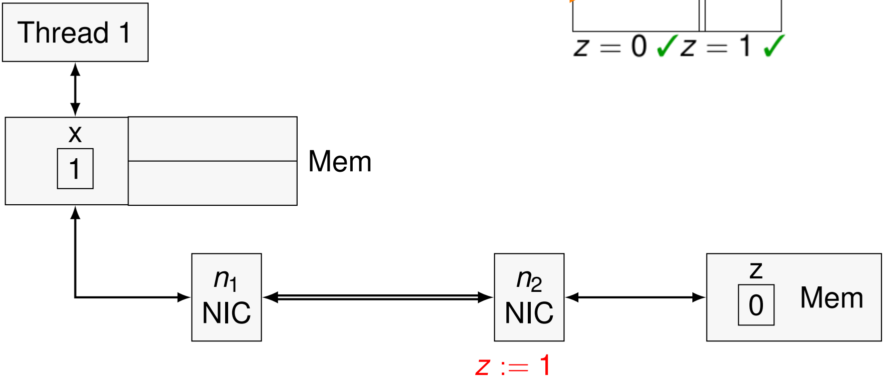
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	

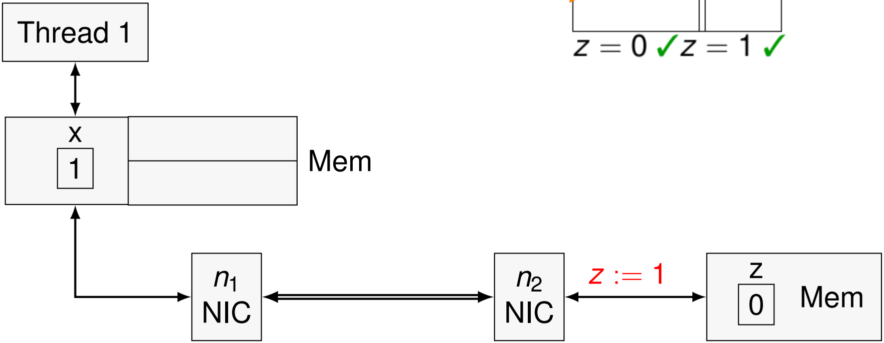
$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

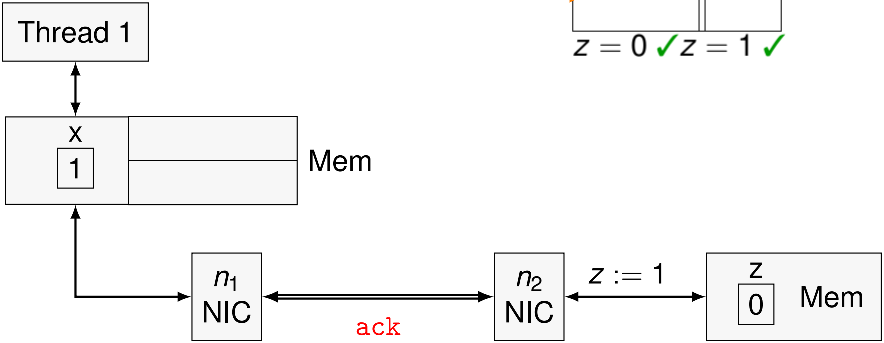
$x = 0$	$z = 0$
$z^{n_2} := x$ $x := 1$	

$z = 0$ ✓ $z = 1$ ✓



RDMA example 1/4

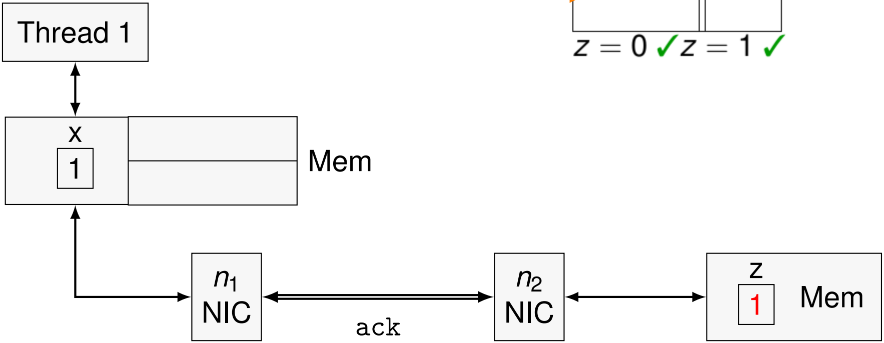
$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	
$z = 0$ ✓	$z = 1$ ✓



RDMA example 1/4

$x=0$	$z=0$
$z^{n_2} := x$	
$x := 1$	

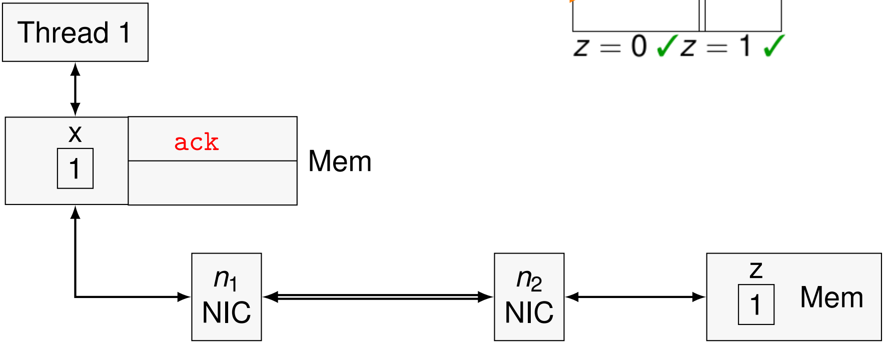
$z=0$ ✓ $z=1$ ✓



RDMA example 1/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $x := 1$	

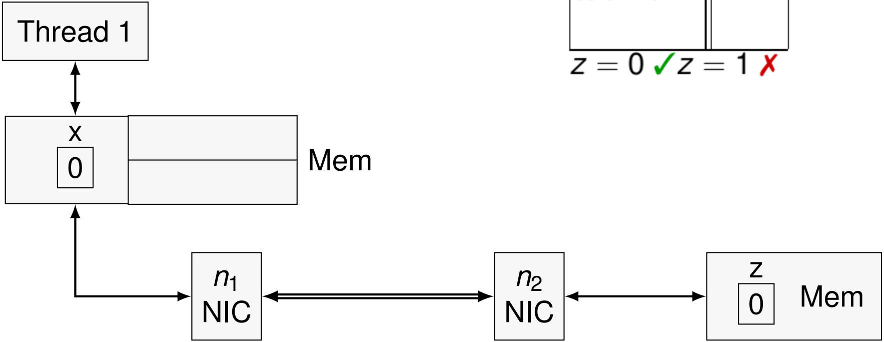
$z = 0$ ✓ $z = 1$ ✓



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

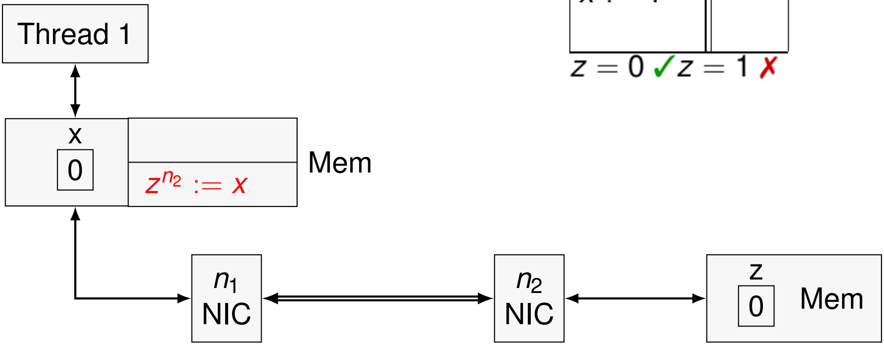
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ `poll`(n_2) $x := 1$	

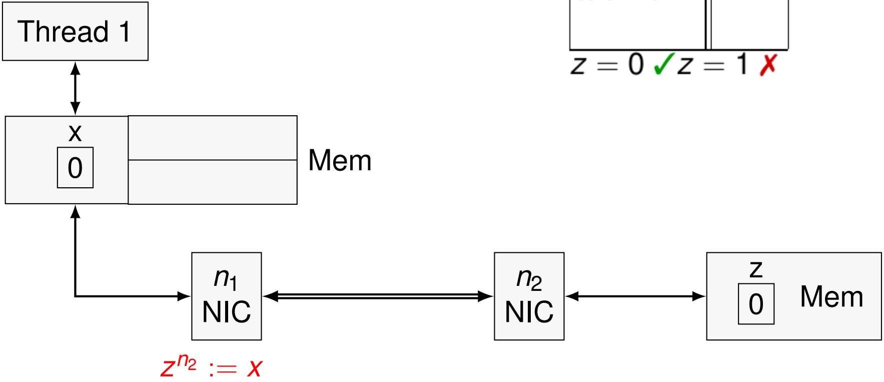
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

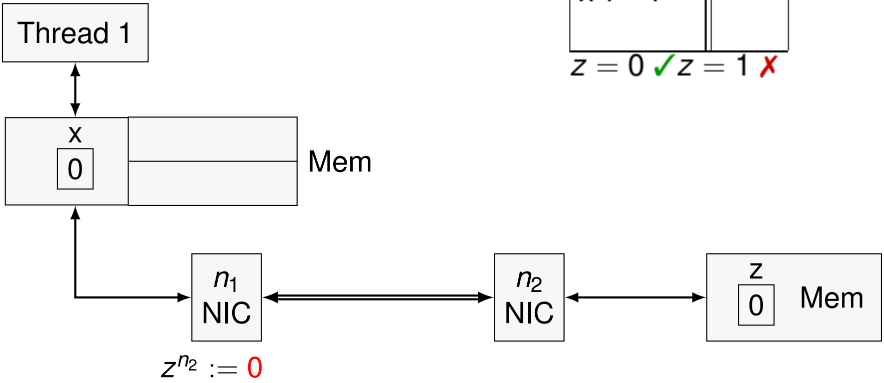
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

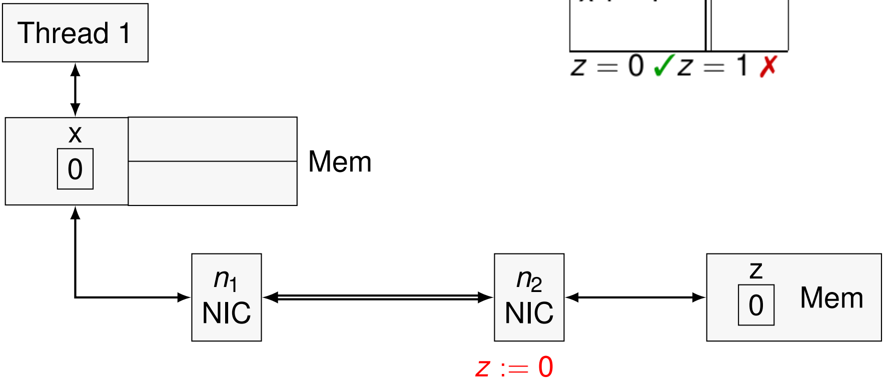
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ `poll`(n_2) $x := 1$	

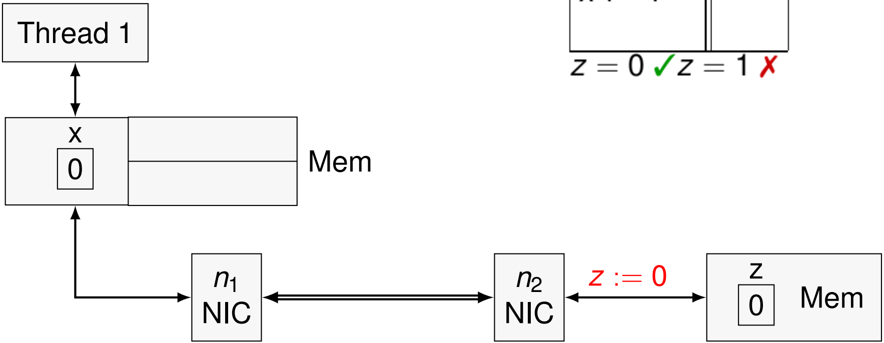
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x=0$	$z=0$
$z^{n_2} := x$ poll(n_2) $x := 1$	

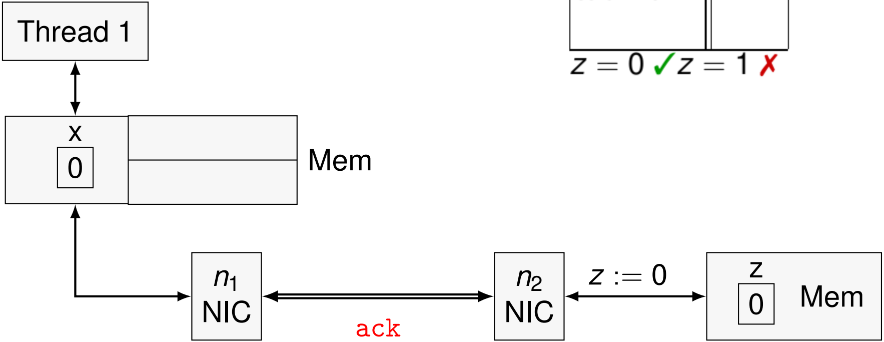
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ `poll`(n_2) $x := 1$	

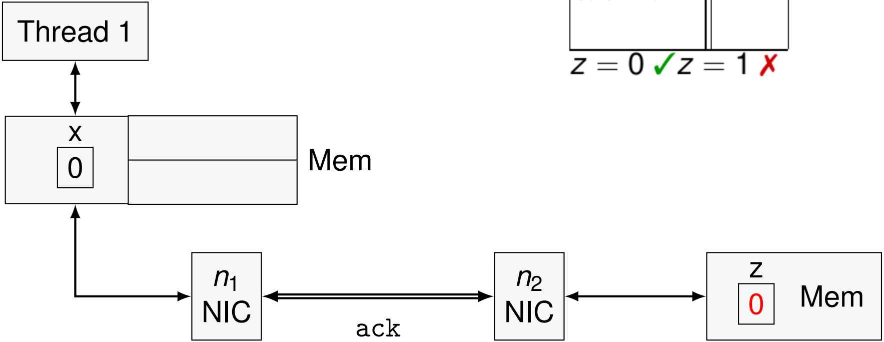
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ poll(n_2) $x := 1$	

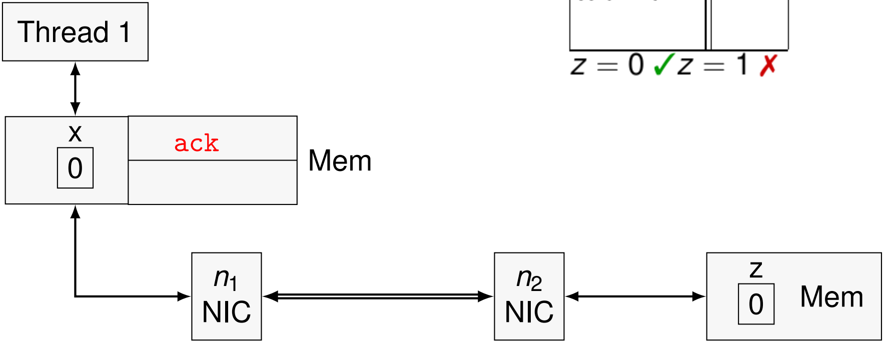
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

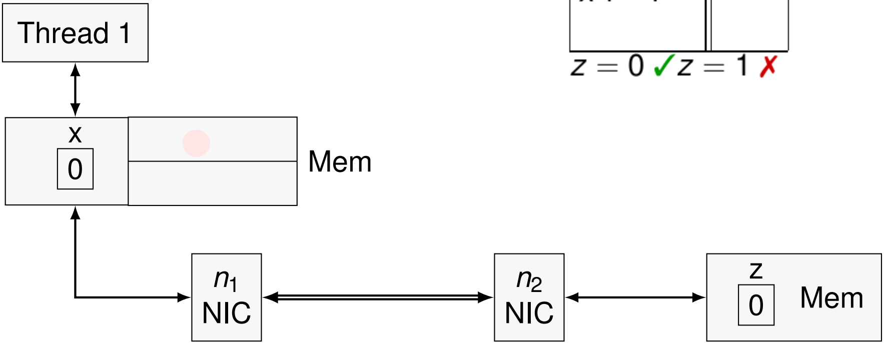
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x=0$	$z=0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

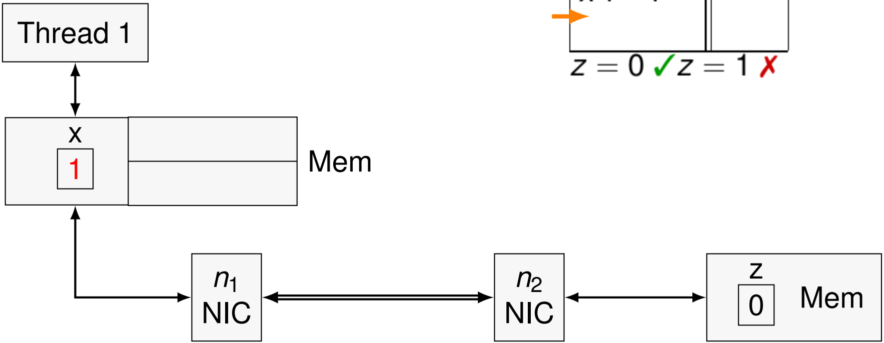
$z = 0$ ✓ $z = 1$ ✗



RDMA example 2/4

$x = 0$	$z = 0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

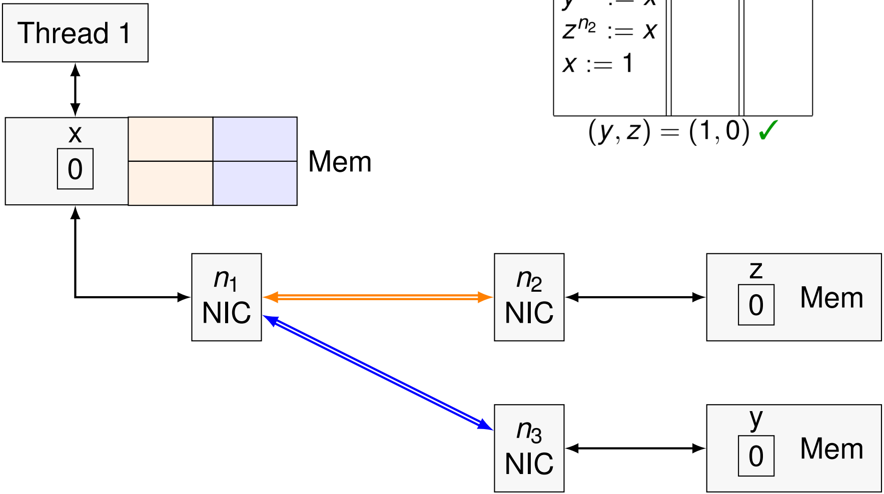
$z = 0$ ✓ $z = 1$ ✗



RDMA example 3/4

$x = 0$	$z = 0$	$y = 0$
$y^{n_3} := x$ $z^{n_2} := x$ $x := 1$		

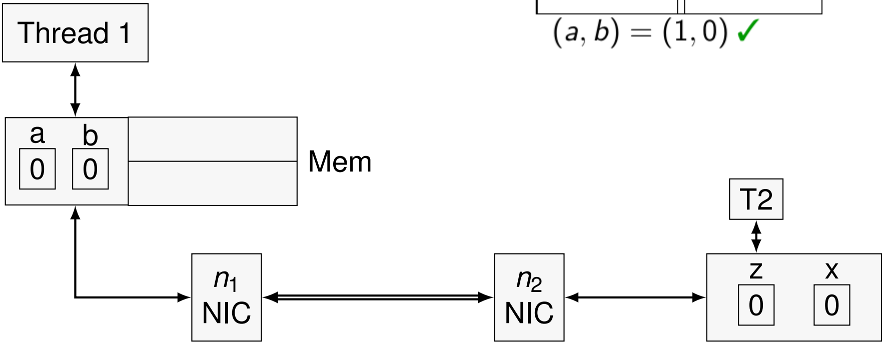
$(y, z) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$	$x := 1$
$b := x^{n_2}$	$z := 1$

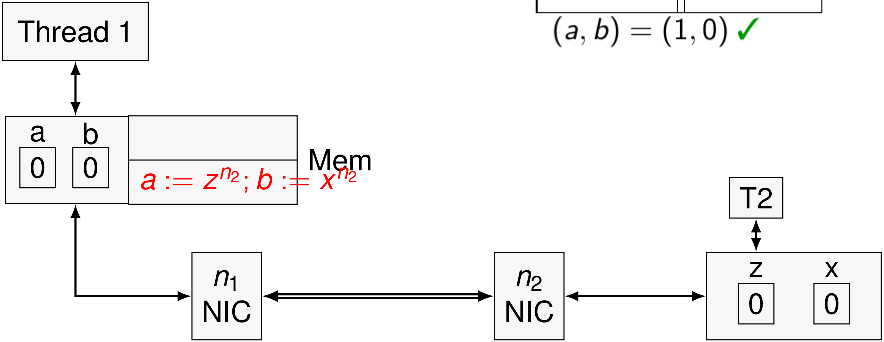
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

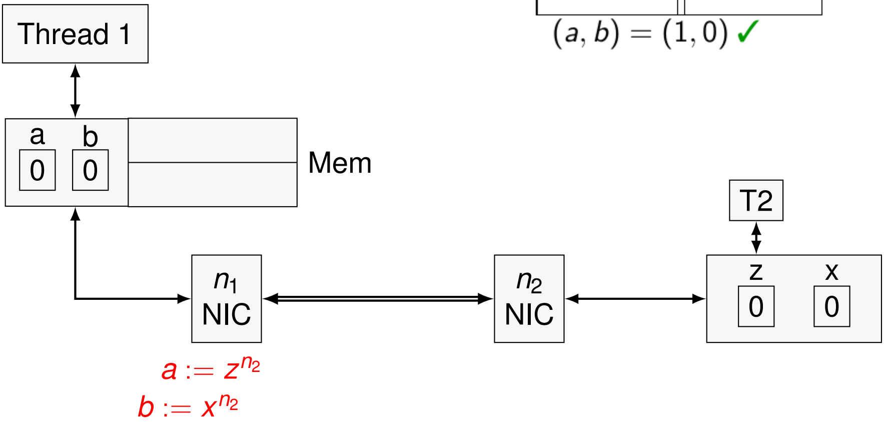
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

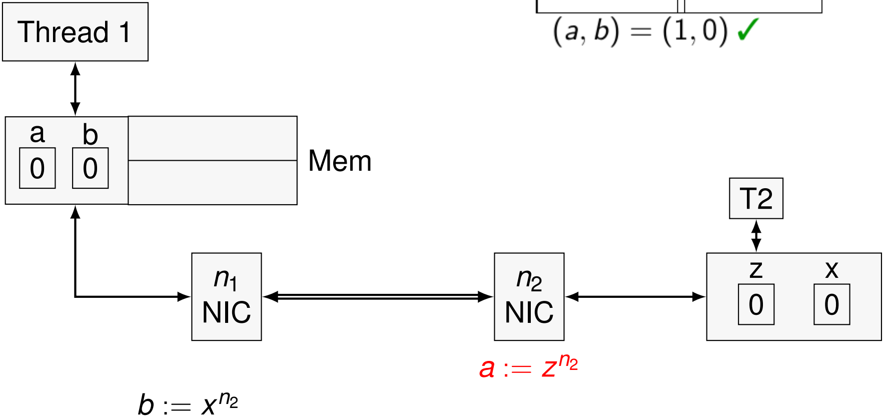
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$	$x := 1$
$b := x^{n_2}$	$z := 1$

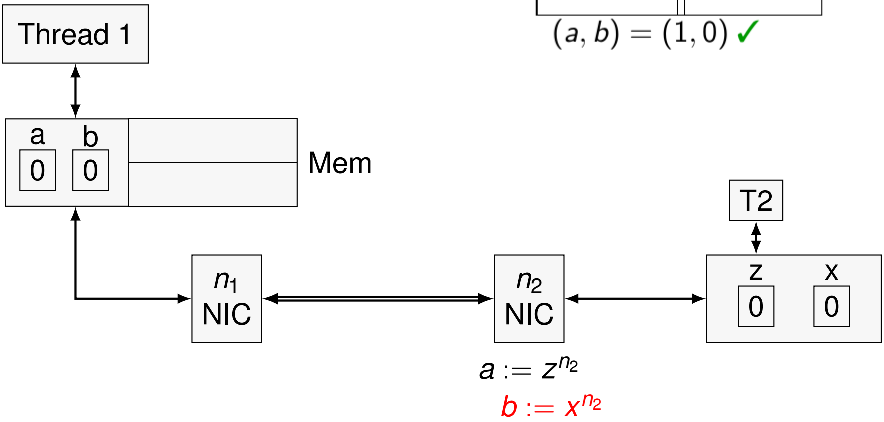
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

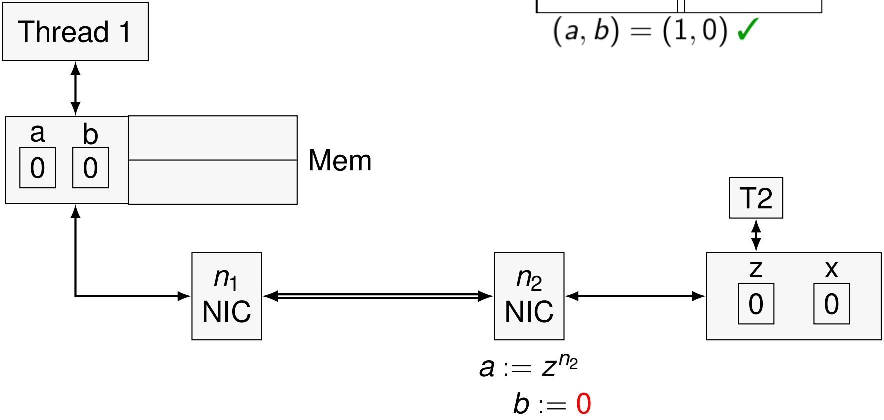
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

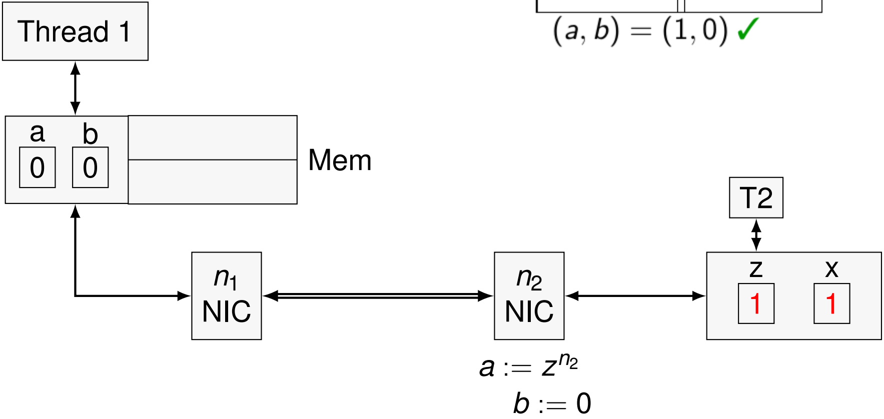
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$	$x := 1$
$b := x^{n_2}$	$z := 1$

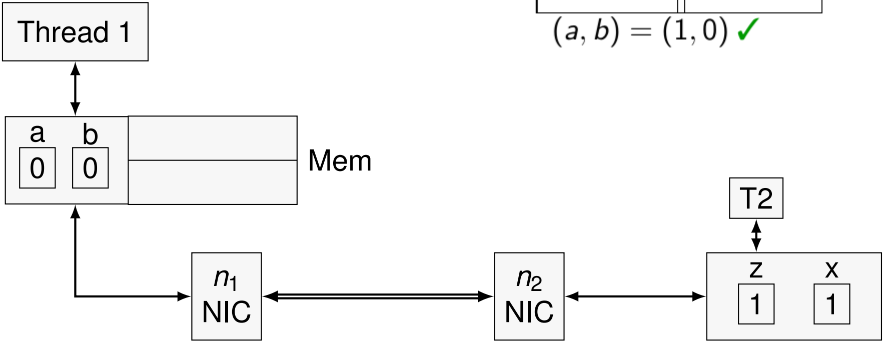
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

$(a, b) = (1, 0)$ ✓



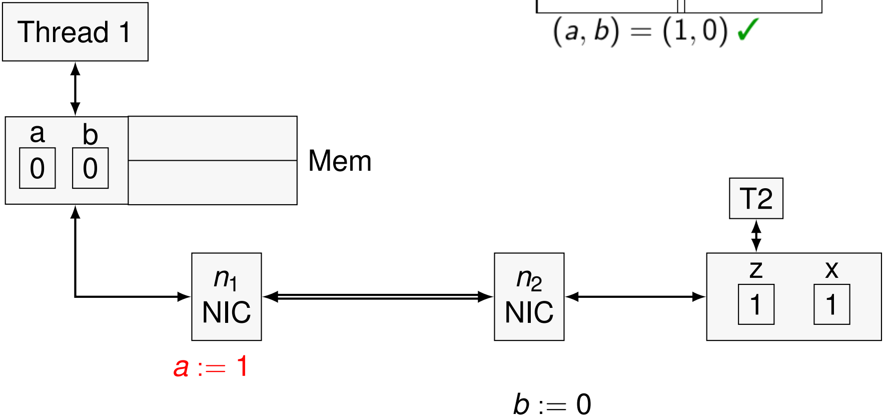
$a := 1$

$b := 0$

RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$	$x := 1$
$b := x^{n_2}$	$z := 1$

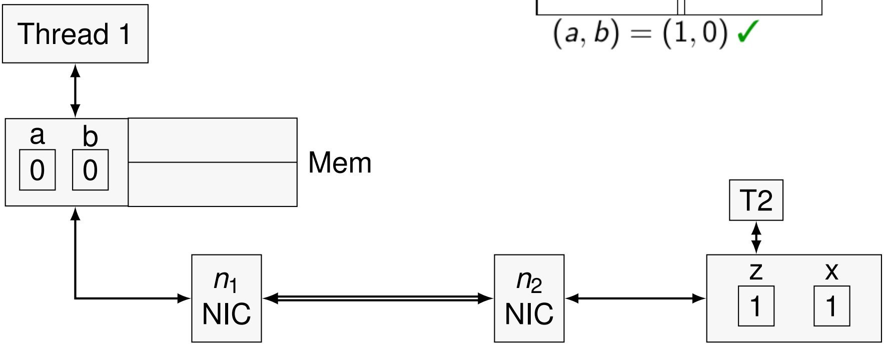
$(a, b) = (1, 0)$ ✓



RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

$(a, b) = (1, 0)$ ✓



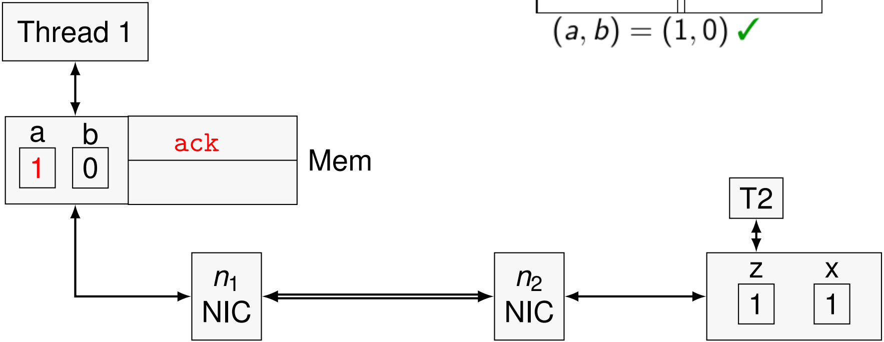
$a := 1$

$b := 0$

RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

$(a, b) = (1, 0)$ ✓

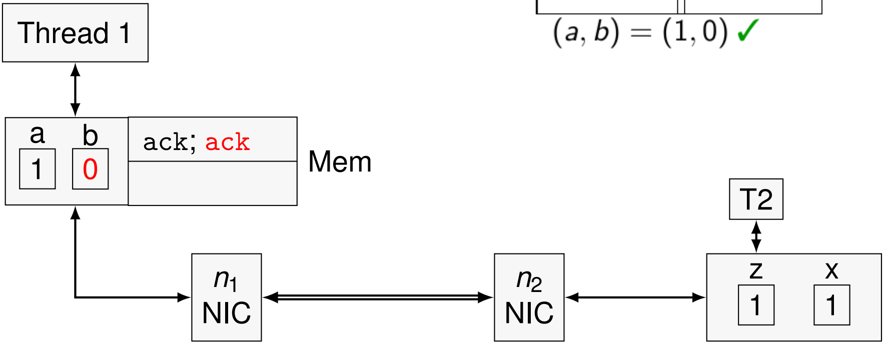


$b := 0$

RDMA example 4/4

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$	$x := 1$
$b := x^{n_2}$	$z := 1$

$(a, b) = (1, 0)$ ✓



Other examples

$x=0$	$z=0$
$z^{n_2} := x$ $x := 1$	

$z=0$ ✓ $z=1$ ✓

$x=0$	$z=0$
$z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

$z=0$ ✓ $z=1$ ✗

$x=0$	$z=0$
$z^{n_2} := x$ $z^{n_2} := x$ $\texttt{poll}(n_2)$ $x := 1$	

$z=0$ ✓ $z=1$ ✓

$y=0$	$x=0$
$x^{n_2} := 1$ $\texttt{poll}(n_2)$ $a := y$	$y^{n_1} := 1$ $\texttt{poll}(n_1)$ $b := x$

$a=b=0$ ✓

$y=w=0$	$x=z=0$
$x^{n_2} := 1$ $c := z^{n_2}$ $\texttt{poll}(n_2)$ $\texttt{poll}(n_2)$ $a := y$	$y^{n_1} := 1$ $d := w^{n_1}$ $\texttt{poll}(n_1)$ $\texttt{poll}(n_1)$ $b := x$

$a=b=0$ ✗

Operational Semantics

Extension of TSO operational semantics

Actually, 2 operational semantics

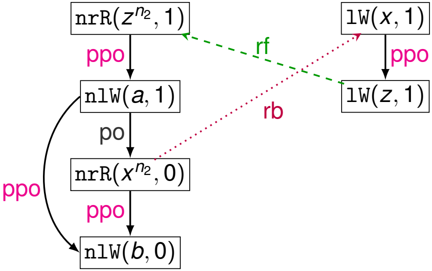
- “concrete” close to hardware
- “simplified” with abstracted Queue-Pairs
- proved equivalent

Declarative Semantics

Declarative Semantics: representing an execution with a graph.

$a, b = 0$	$x, z = 0$
$a := z^{n_2}$ $b := x^{n_2}$	$x := 1$ $z := 1$

$(a, b) = (1, 0)$ ✓



Declarative Semantics

Extension of the usual TSO declarative semantics

- Usual relations rf, rb, mo
- New relations pf (polls-from) and nfo (NIC flush order)
- Semantics defined with standard acyclicity conditions

Proved equivalent to the operational semantics!

Empirical Validation

Tested $\sim$40 examples on 2 setups (*Infiniband* and *RoCE*), with the help of NVIDIA engineers.

- Did *not* observe any forbidden beviours (the semantics is not too strong)
- Did observe *most* allowed beviours (with enough stress) (the semantics is not too weak)

Conclusion

Several semantics for RDMA, proved equivalent.

Can observe most weak behaviours in practice.

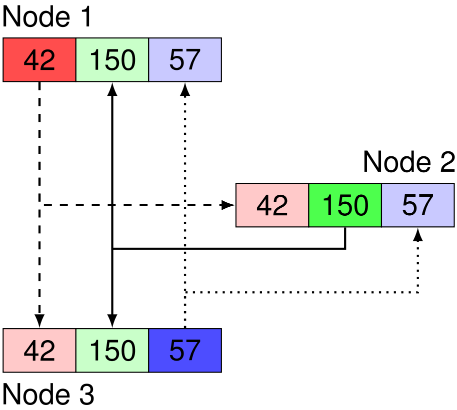
Future Work in Progress

- Use semantics to formalise abstract data-structures and libraries
- Restricting programs for better guarantees (Robustness / DRF-like property)

Shared data-structures

Example:
Shared State Table (SST)[1]

Each node has a value and a copy of other nodes values

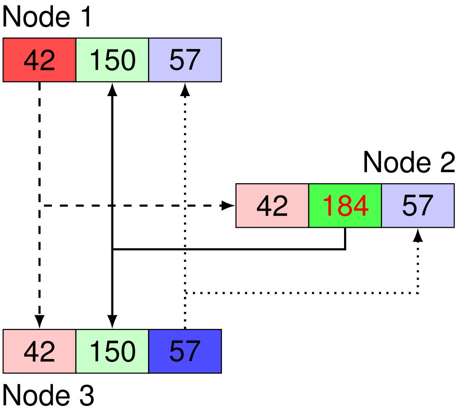


[1] Jha et al. 2018, "Derecho: Fast State Machine Replication for Cloud Services".

Shared data-structures

Example:
Shared State Table (SST)[1]

Each node has a value and a copy of other nodes values

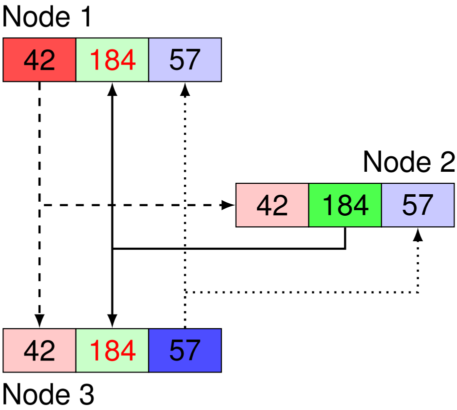


[1] Jha et al. 2018, "Derecho: Fast State Machine Replication for Cloud Services".

Shared data-structures

Example:
Shared State Table (SST)[1]

Each node has a value and a copy of other nodes values



[1] Jha et al. 2018, "Derecho: Fast State Machine Replication for Cloud Services".

Store Buffer Example

$y = 0$	$x = 0$
$x^{n_2} := 1$	$y^{n_1} := 1$
$\mathtt{poll}(n_2)$	$\mathtt{poll}(n_1)$
$a := y$	$b := x$

$(a, b) = (0, 0)$ ✓