

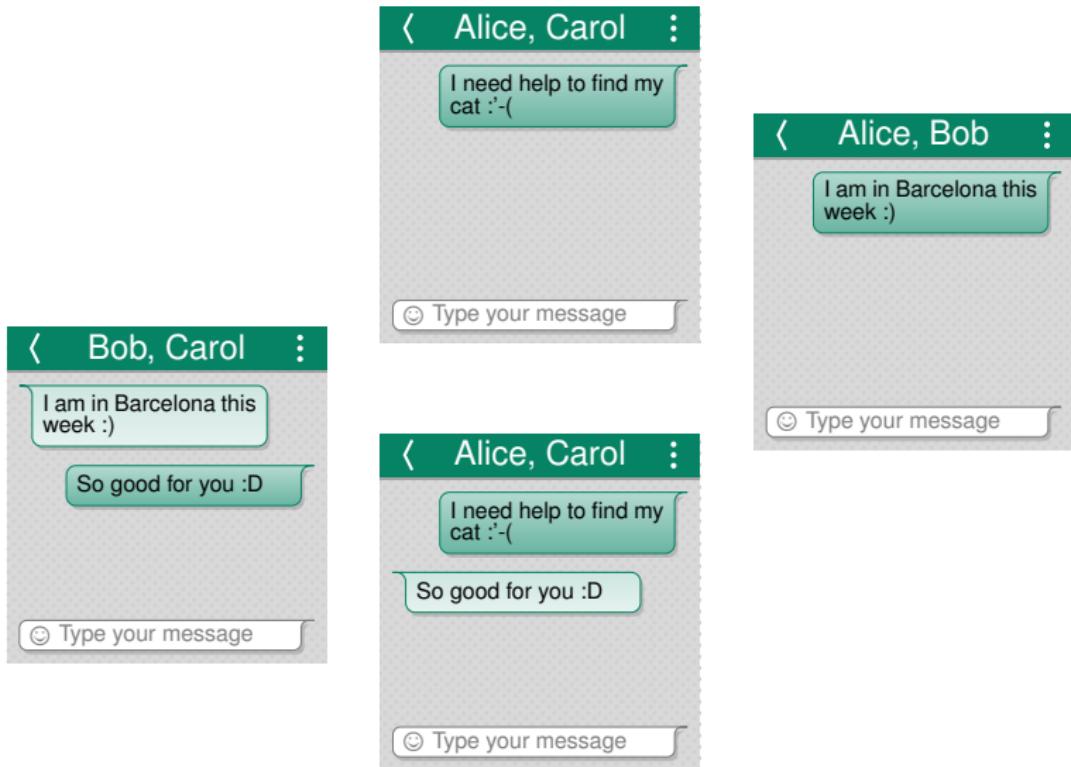
Causal Consistency : Beyond Memory

Matthieu Perrin Achour Mostefaoui Claude Jard

LINA, University of Nantes

March 16th, 2016

Motivation

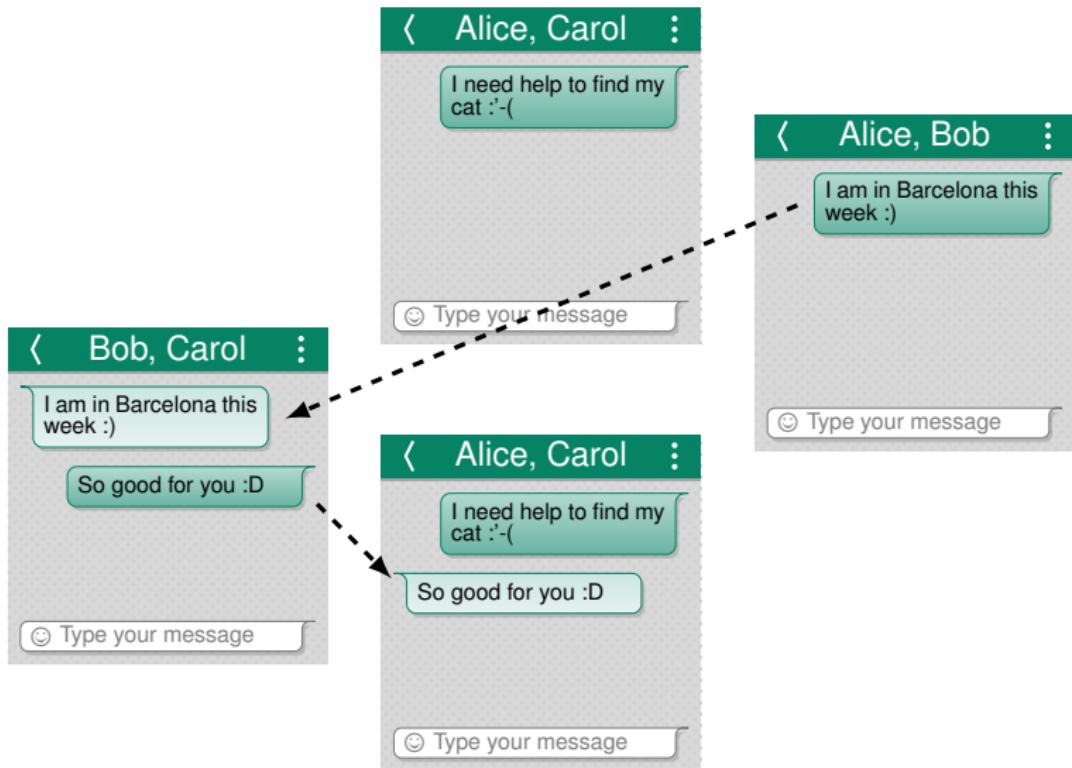


At Alice's

At Bob's

At Carol's

Motivation



At Alice's

At Bob's

At Carol's

What is causal consistency ?

Causal memory¹

- ▶ Build causal order
 - ▶ Writes-into order → semantic dependancies
 - ▶ Process order
- ▶ Each process respects causal order

¹ Ahamad et. al. *Causal Memory: Definition, Implementation and Programming.* DISQ 1995

What is causal consistency ?

Causal memory¹

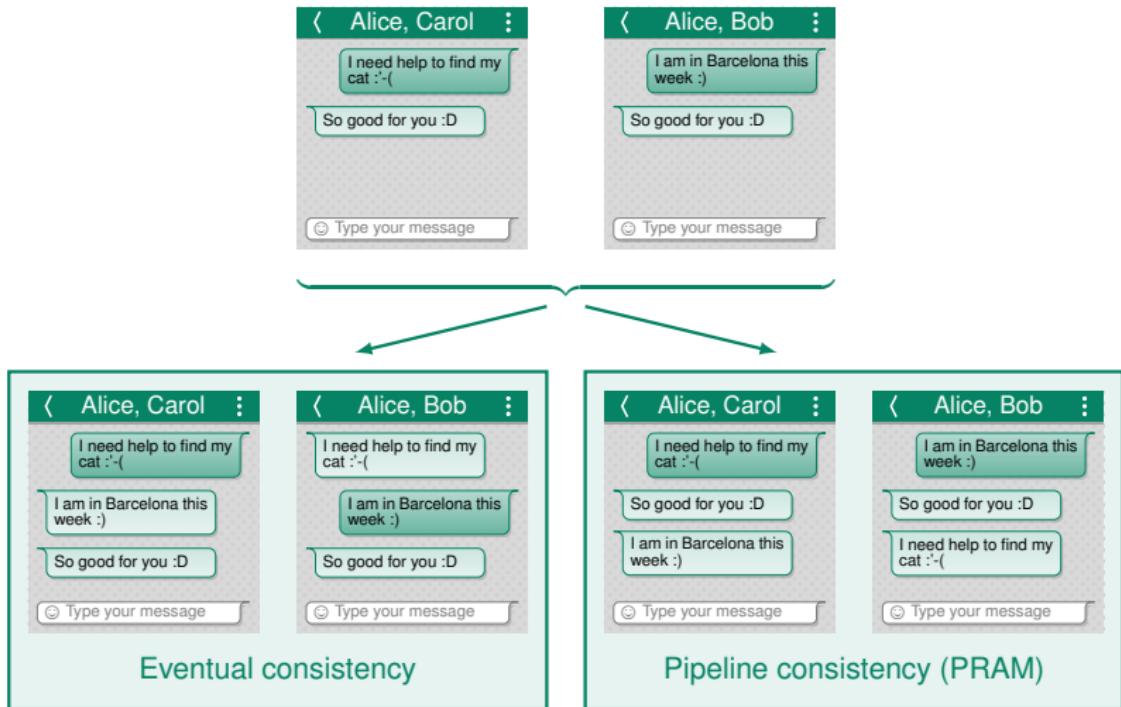
- ▶ Build causal order
 - ▶ Writes-into order → semantic dependancies
 - ▶ Process order
- ▶ Each process respects causal order

Approach

- ▶ Specify data type sequentially
- ▶ Suppose existence of a partial order
- ▶ Add links partial order/sequential specification
- ▶ ⇒ Semantic dependancies are respected

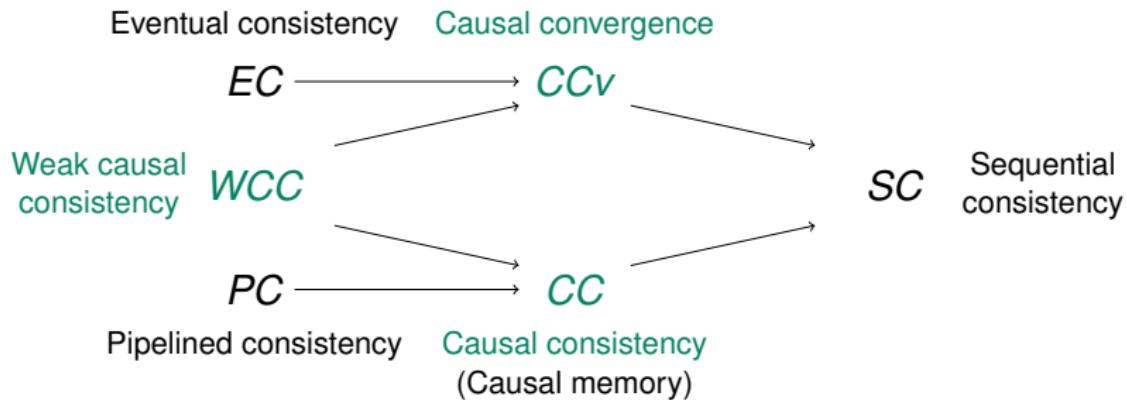
¹ Ahamad et. al. *Causal Memory: Definition, Implementation and Programming.* DISQ 1995

Pipeline Consistency versus Eventual Consistency²



²P. M. J. Update Consistency in Partitionable Systems. IPDPS 2015

Contributions



Outline

1. Specifying shared objects
2. Weak causal consistency and causal convergence
3. Causal consistency

Specifying shared objects

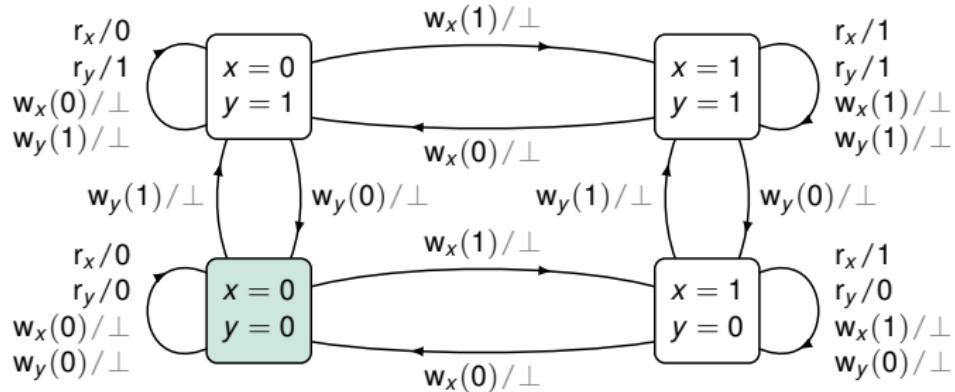
Sequential specification

- ▶ How the object behaves in a sequential environment
 - ▶ Memory
 - ▶ Data base
 - ▶ Instant messaging service

Consistency criterion

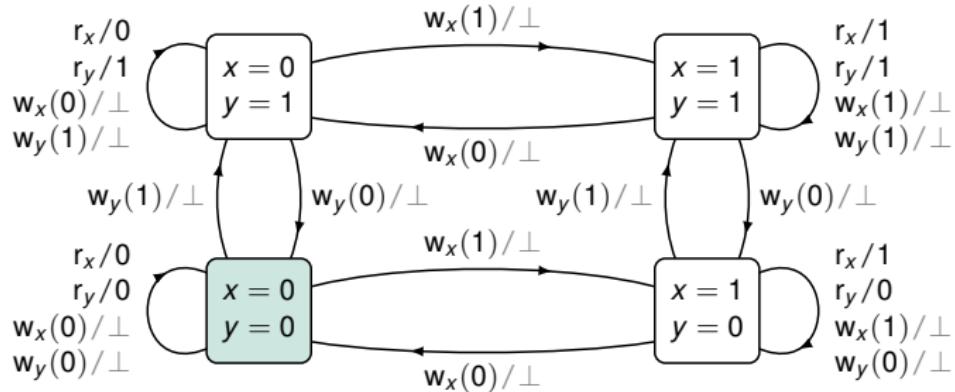
- ▶ How distribution impacts the sequential behaviour
 - ▶ Causal consistency
 - ▶ Eventual consistency
 - ▶ Sequential consistency

Specification of the memory



operation : input/output

Specification of the memory



operation : input/output

Sequential specification

Path in the transition system, starting in q_0

- ▶ finite or infinite
- ▶ hidden operations: $\sigma_i/\sigma_o \Rightarrow \sigma_i$

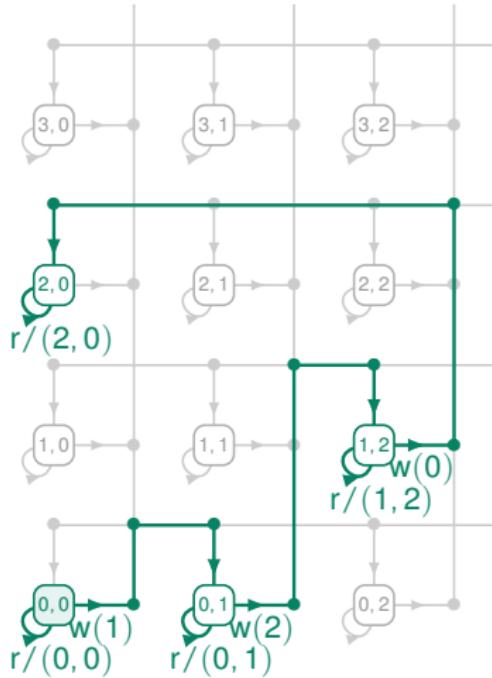
Window stream of size k

Operation $w(n)/\perp$

adds n in the stream

Operation $r/(n_1, \dots, n_k)$

return the k last values



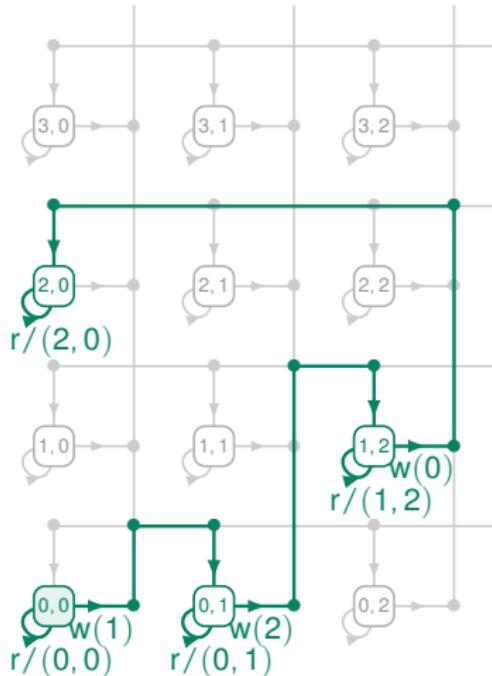
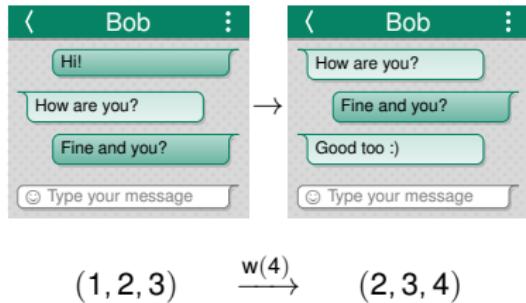
Window stream of size k

Operation $w(n) / \perp$

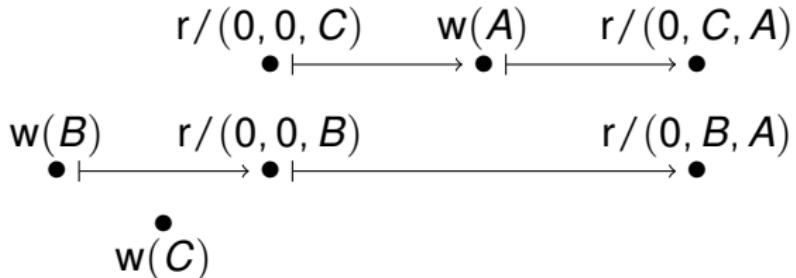
adds n in the stream

Operation $r / (n_1, \dots, n_k)$

return the k last values



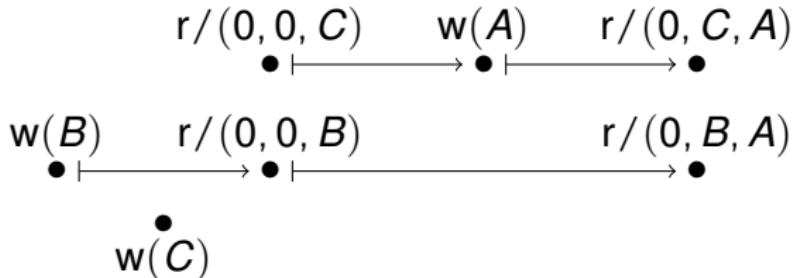
Distributed history



Formally

- ▶ Partially ordered set of events, labelled with operations
- ▶ Each event has a finite past

Distributed history



In systems composed of processes

Events: call of ADT operations

Labels: operations of the ADT

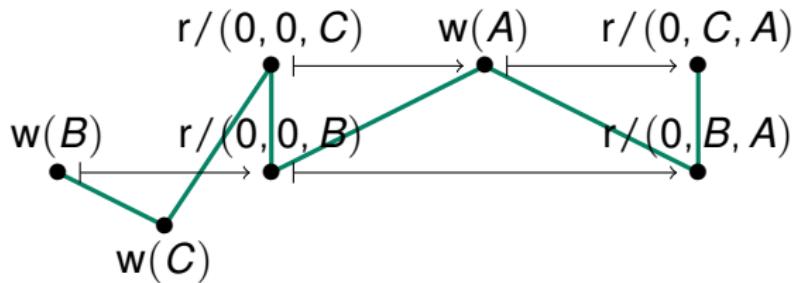
Order: same process, same order

Linearisation

$$l \in \text{lin}(H)$$

- ▶ Sequence of operations
- ▶ Respect of process order

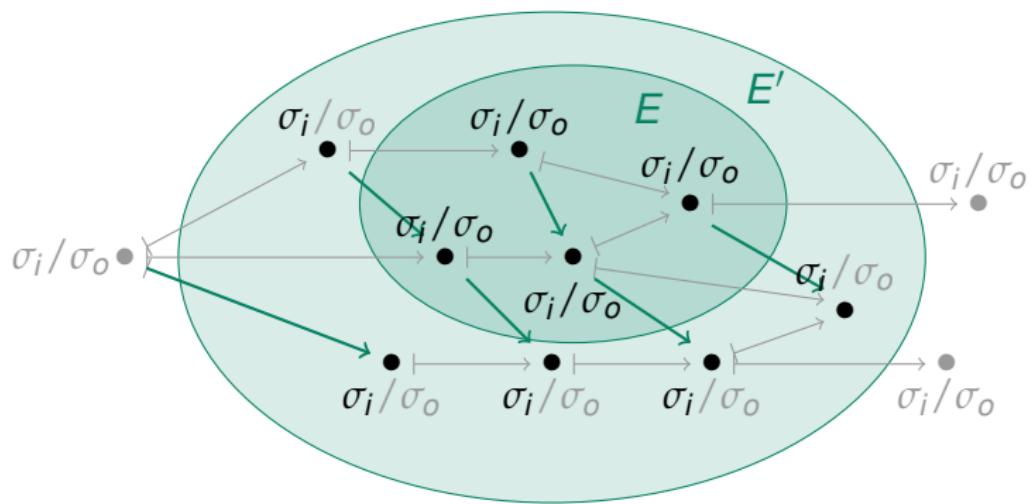
$$l = w(B) \cdot w(C) \cdot r/(0, 0, C) \cdot r/(0, 0, B) \cdot w(A) \cdot r/(0, B, A) \cdot r/(0, C, A)$$



Projection

$(H^\rightarrow).\pi(E, E')$

- ▶ Replaces process order by \rightarrow
- ▶ Keeps reads of E
- ▶ Keeps writes of E'



Consistency criterion

Sequential specification

- ▶ $L : \text{ADT} \rightarrow \text{set of sequential histories}$

Consistency criterion

- ▶ $C : \text{ADT} \rightarrow \text{set of distributed histories}$

E.g. Sequential Consistency

$$SC : T \mapsto \{H \in \mathbb{H} : \text{lin}(H) \cap L(T) \neq \emptyset\}$$

Outline

1. Specifying shared objects
2. Weak causal consistency and causal convergence
3. Causal consistency

Causal order and time zones

Causal order →

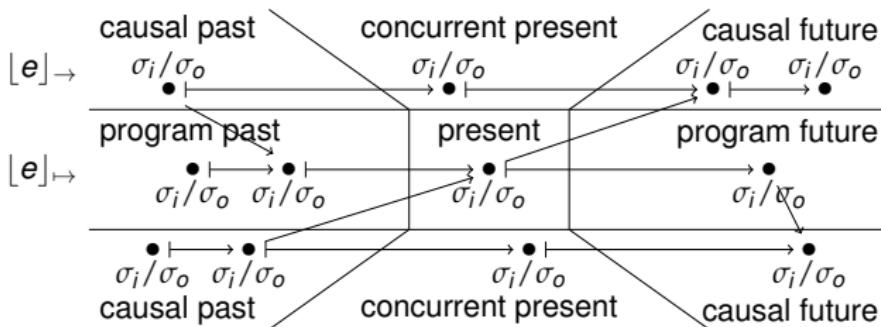
- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

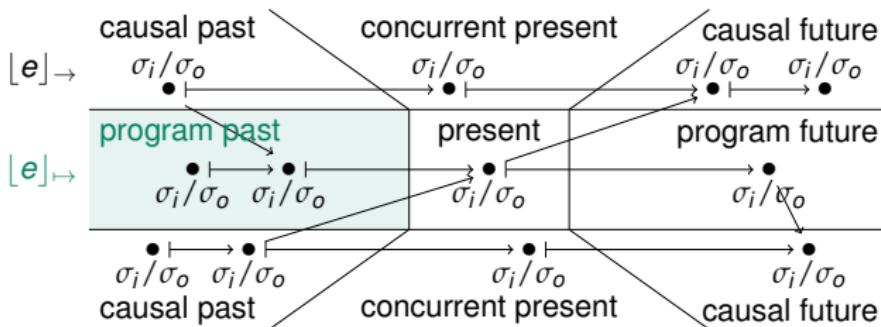


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

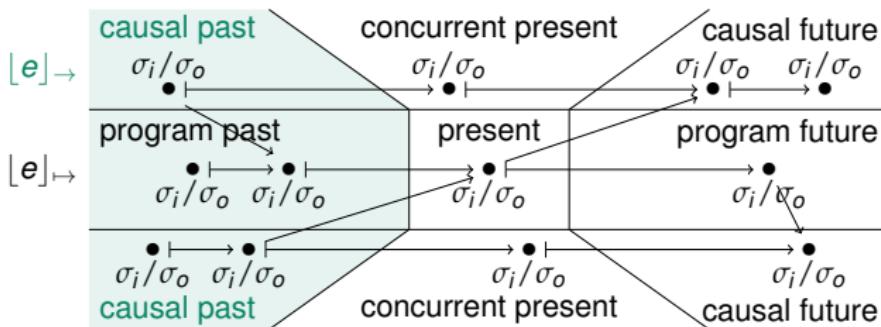


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

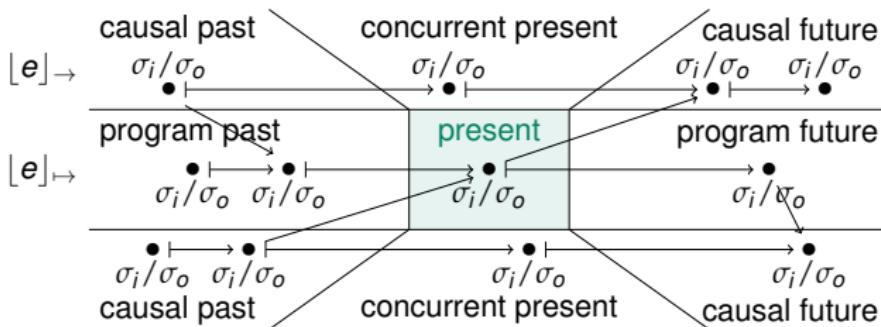


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

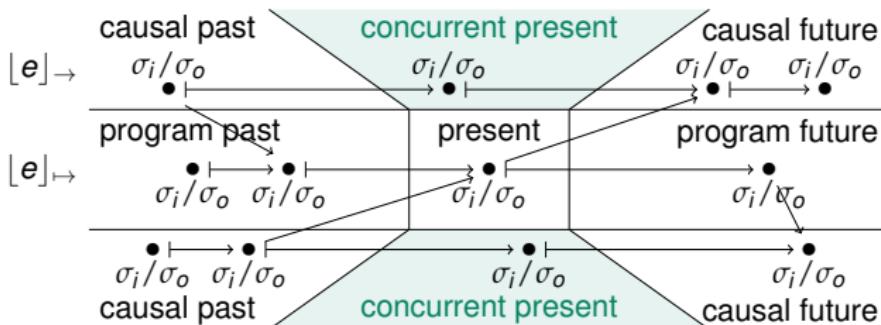


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

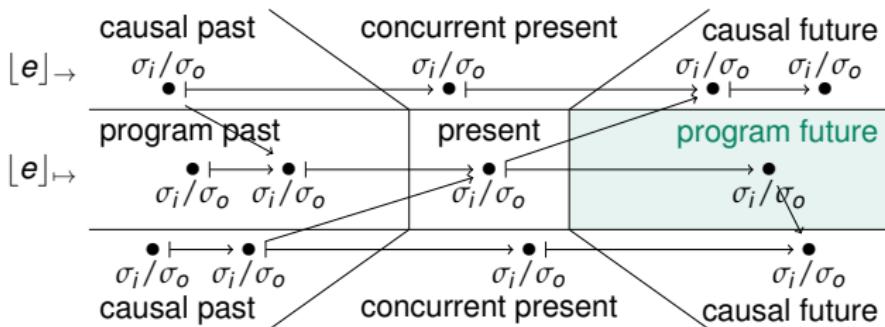


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

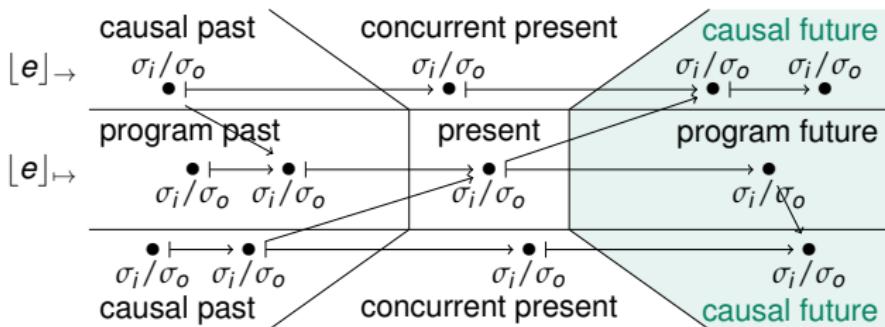


Causal order and time zones

Causal order →

- ▶ Partial order on all events
- ▶ Contains process order
- ▶ For each event : **cofinite** causal future
 - ▶ necessary to make it not trivial
 - ▶ necessary for pipelined consistency $PC \leq CC$
 - ▶ necessary for eventual consistency $EC \leq CCv$

6 time zones per event

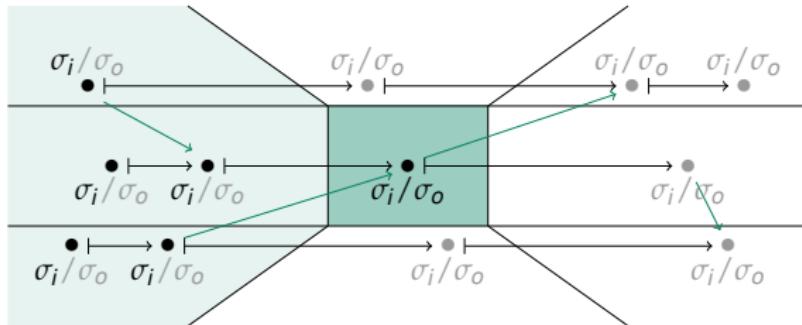


Weak causal consistency

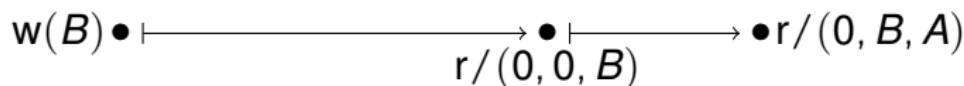
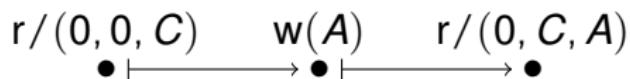
$H \in WCC(T)$ if

- ▶ there exists a causal order \rightarrow , for each event e

$$\text{lin}((H^\rightarrow). \pi(\{e\}, [e]_\rightarrow)) \cap L(T) \neq \emptyset$$



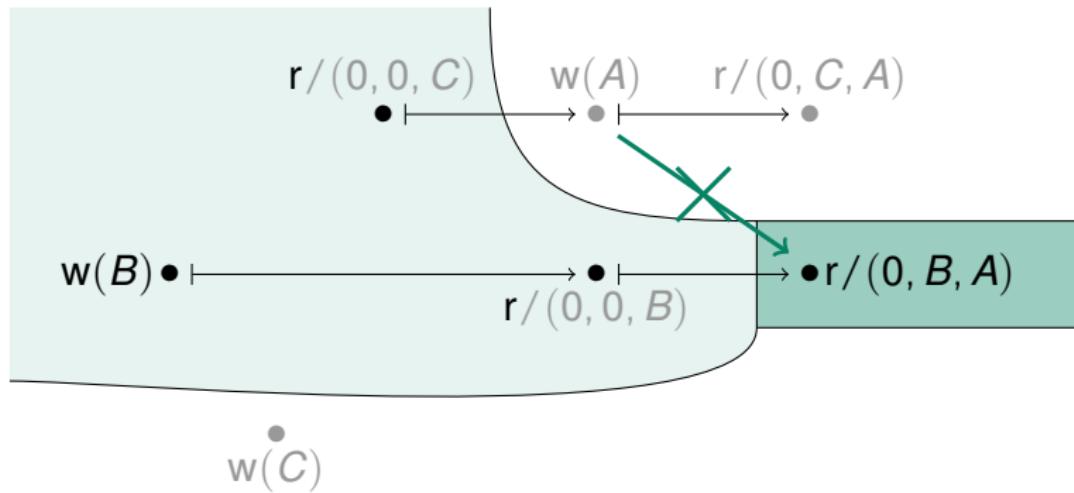
Alice, Bob and Carol



w(C)
•

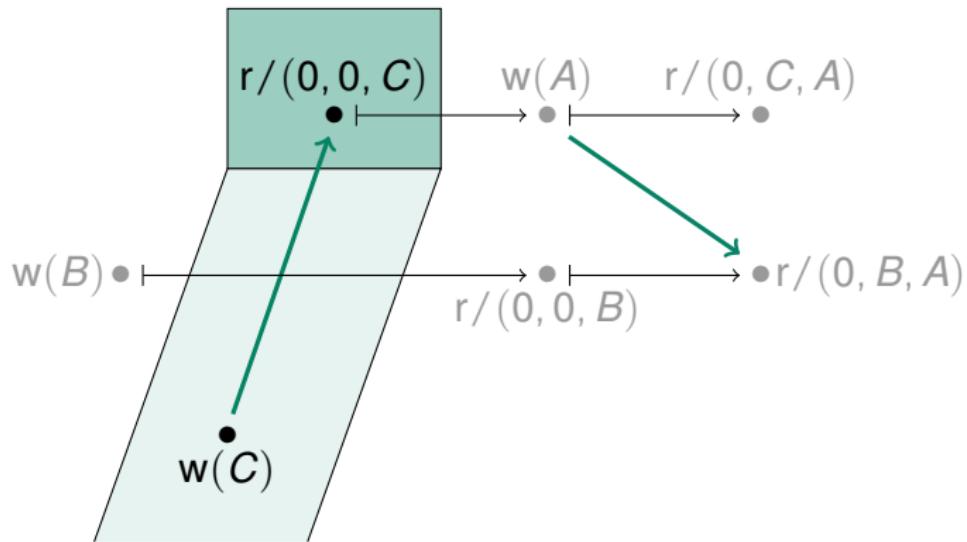
$H \in WCC(\mathcal{W}_3)?$

Alice, Bob and Carol



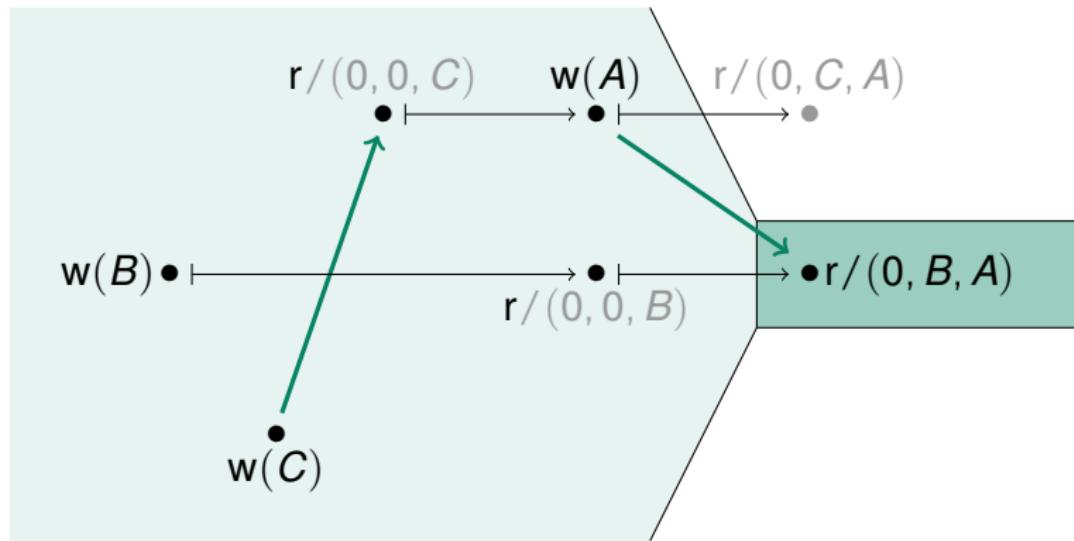
$H \in WCC(\mathcal{W}_3)?$

Alice, Bob and Carol



$H \in WCC(\mathcal{W}_3)?$

Alice, Bob and Carol



$H \in WCC(\mathcal{W}_3)?$

Causal convergence

$H \in CCv(T)$ if

- ▶ there exists
 - ▶ a causal order \rightarrow
 - ▶ a total order \leq that contains \rightarrow
- ▶ for each event e

$$\text{lin}((H^{\leq}).\pi(\{e\}, \lfloor e \rfloor_{\rightarrow})) \cap L(T) \neq \emptyset$$

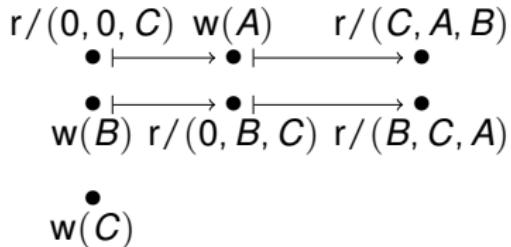
Causal convergence

$H \in CCv(T)$ if

- ▶ there exists
 - ▶ a causal order \rightarrow
 - ▶ a total order \leq that contains \rightarrow
- ▶ for each event e

$$\text{lin}((H^{\leq}).\pi(\{e\}, \lfloor e \rfloor_{\rightarrow})) \cap L(T) \neq \emptyset$$

Examples



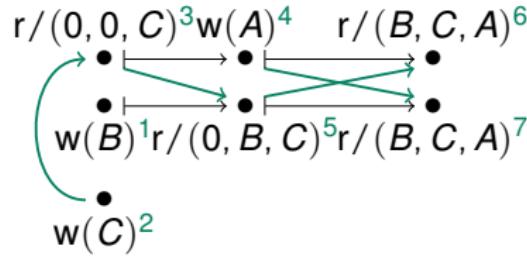
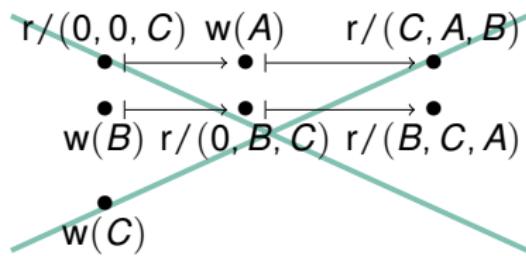
Causal convergence

$H \in CCv(T)$ if

- ▶ there exists
 - ▶ a causal order \rightarrow
 - ▶ a total order \leq that contains \rightarrow
- ▶ for each event e

$$\text{lin}((H^{\leq}).\pi(\{e\}, \lfloor e \rfloor_{\rightarrow})) \cap L(T) \neq \emptyset$$

Examples



Properties

- ▶ H : distributed history
- ▶ T : abstract data type
- ▶ write : not a loop
- ▶ read : output depends of the state

Weak causal consistency

$$\begin{aligned} & H \in WCC(T) \\ \wedge & \text{ no concurrent writes} \\ \implies & H \in SC(T) \end{aligned}$$

Causal convergence

$$\begin{aligned} & H \in CCv(T) \\ \wedge & \text{ no read concurrent with a write} \\ \implies & H \in SC(T) \end{aligned}$$

Outline

1. Specifying shared objects
2. Weak causal consistency and causal convergence
3. Causal consistency

Causal memory

Writes-into order

- ▶ Relation $w_x(n) \multimap r_x/n$
- ▶ At most one predecessor per read
- ▶ Read 0 if no predecessor

Causal memory

There exists \multimap such that

- ▶ there exists a partial order \rightarrow that contains \multimap and \mapsto
- ▶ for each process p , $\text{lin}((H^\rightarrow).(\pi(p, E_H))) \cap L(\mathcal{M}) \neq \emptyset$

Pipeline Consistency³

$H \in PC(T)$ if

- ▶ For each process p

$$\text{lin}((H^\rightarrow). \pi(p, E)) \cap L(T) \neq \emptyset$$

³Lipton, Sandberg. *PRAM: A Scalable Shared Memory*. 1988

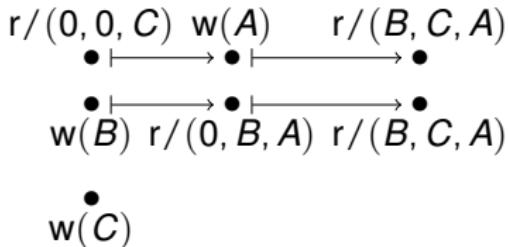
Pipeline Consistency³

$H \in PC(T)$ if

- ▶ For each process p

$$\text{lin}((H^\rightarrow). \pi(p, E)) \cap L(T) \neq \emptyset$$

Examples



³Lipton, Sandberg. *PRAM: A Scalable Shared Memory*. 1988

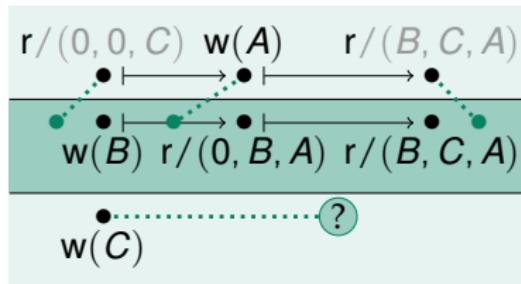
Pipeline Consistency³

$H \in PC(T)$ if

- ▶ For each process p

$$\text{lin}((H^\rightarrow). \pi(p, E)) \cap L(T) \neq \emptyset$$

Examples



³Lipton, Sandberg. *PRAM: A Scalable Shared Memory*. 1988

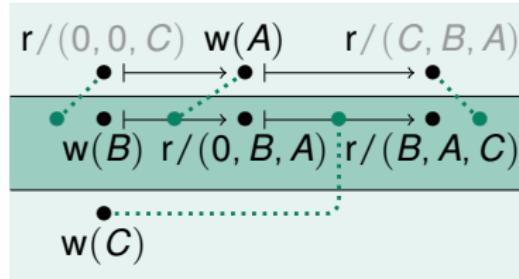
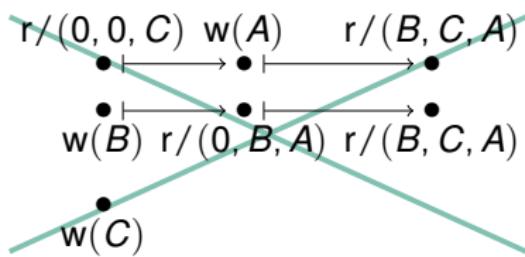
Pipeline Consistency³

$H \in PC(T)$ if

- ▶ For each process p

$$\text{lin}((H^\rightarrow). \pi(p, E)) \cap L(T) \neq \emptyset$$

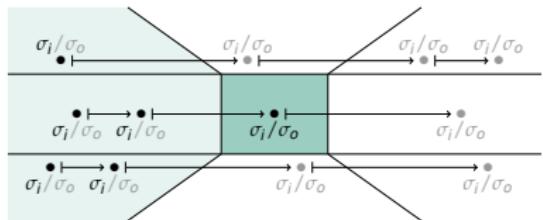
Examples



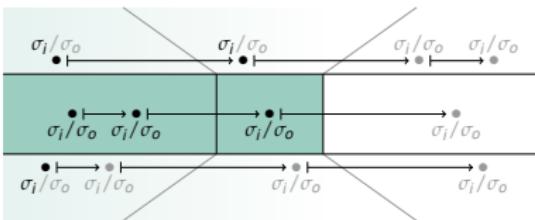
³Lipton, Sandberg. PRAM: A Scalable Shared Memory. 1988

Causal consistency

Weak causal consistency

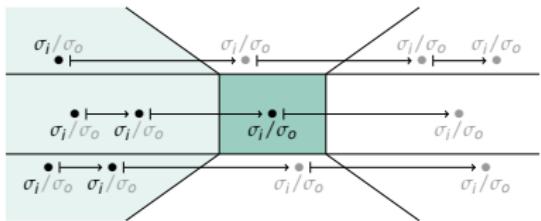


Pipeline consistency

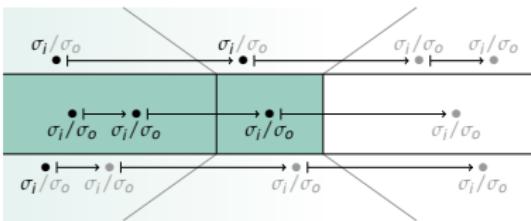


Causal consistency

Weak causal consistency



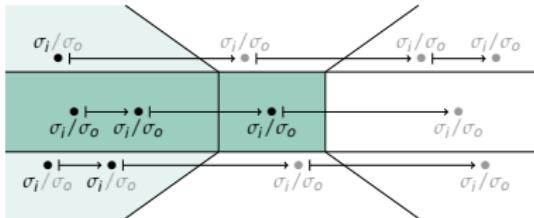
Pipeline consistency



Causal consistency: $H \in CC(T)$ if

- There exists a causal order \rightarrow , for each event e

$$\text{lin}((H^\rightarrow). \pi([e]_\rightarrow, [e]_\rightarrow)) \cap L(T) \neq \emptyset$$



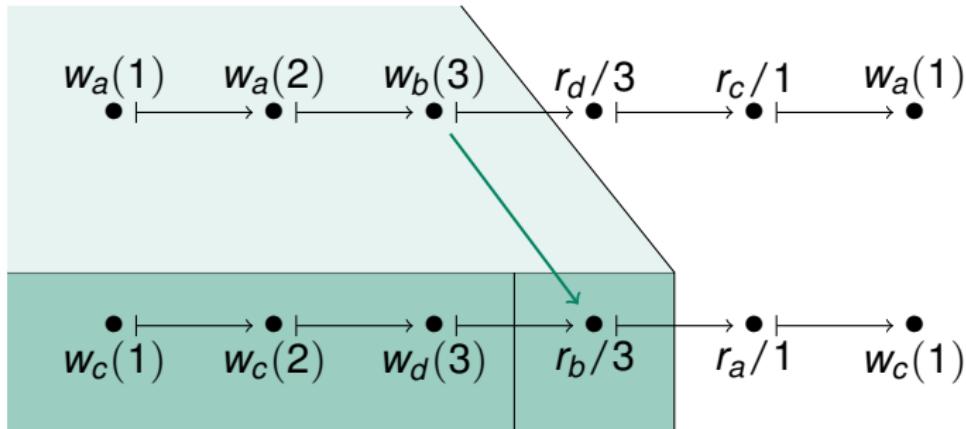
Limit of causal memory



Causal consistency?

Causal memory?

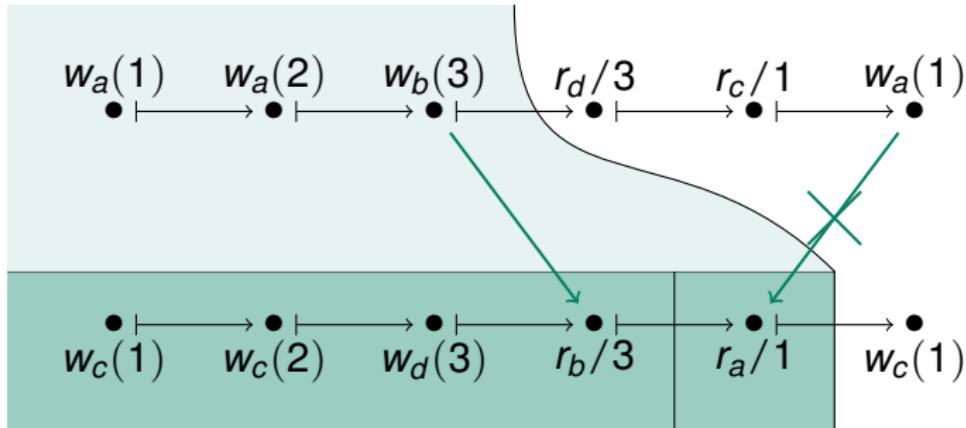
Limit of causal memory



Causal consistency?

Causal memory?

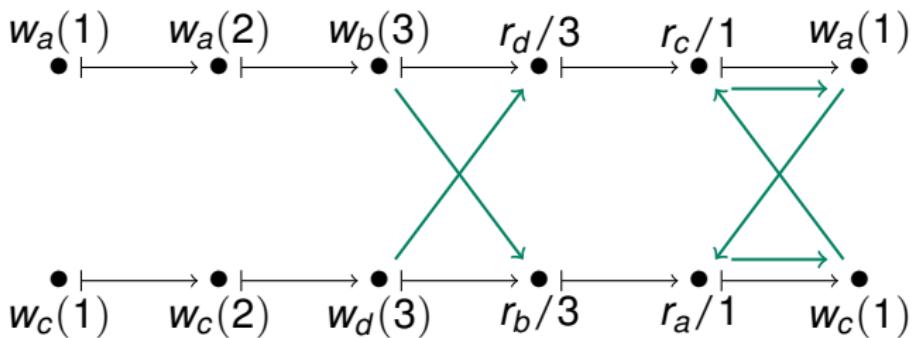
Limit of causal memory



Causal consistency?

Causal memory?

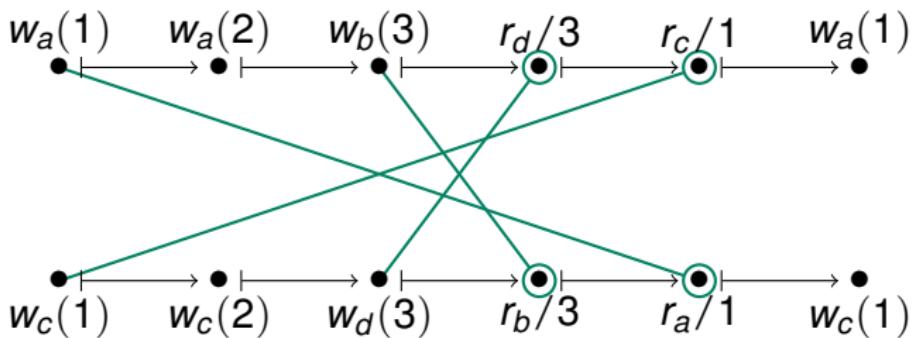
Limit of causal memory



Causal consistency

Causal memory?

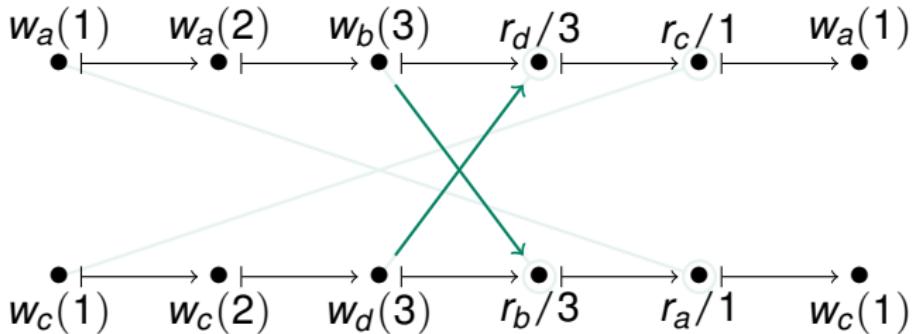
Limit of causal memory



~~Causal consistency~~

Causal memory?

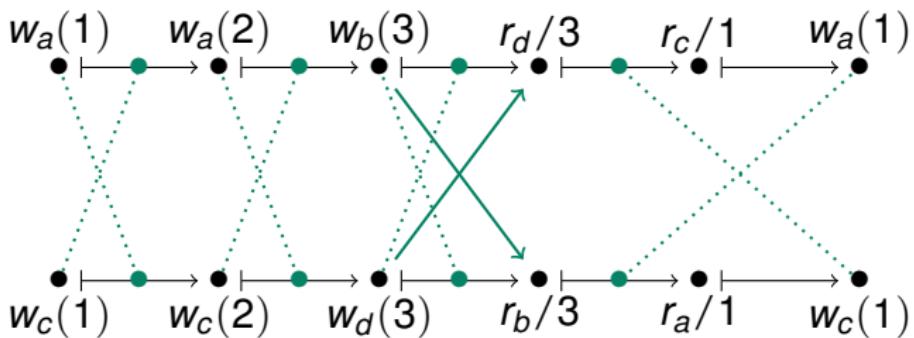
Limit of causal memory



~~Causal consistency~~

Causal memory?

Limit of causal memory



Causal consistency

Causal memory?

Link with causal memory

Property 1

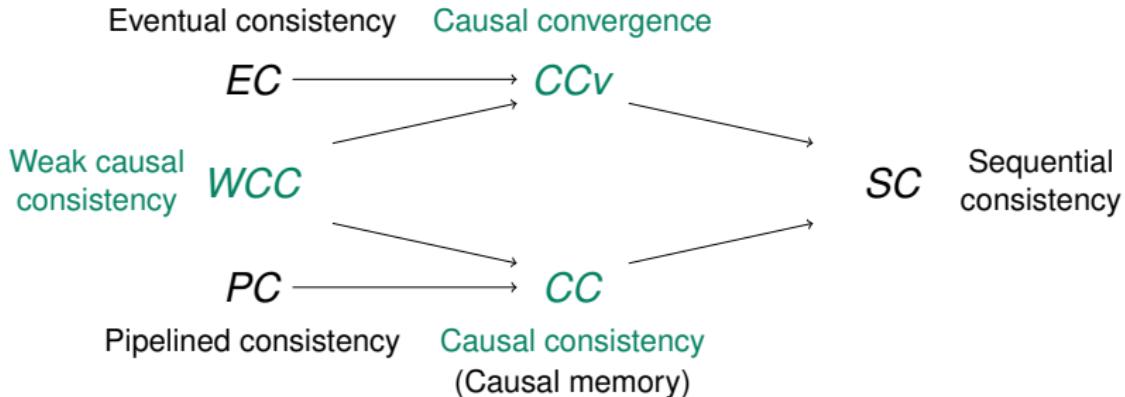
H causally consistent for memory
 $\implies H$ consistent for causal memory

Property 2

Suppose H does not contain twice $w_x(n)$.

H consistent for causal memory
 $\implies H$ causally consistent for memory

Conclusion



When to use?

Causal convergence: for distributed systems

Causal consistency: for parallel algorithms