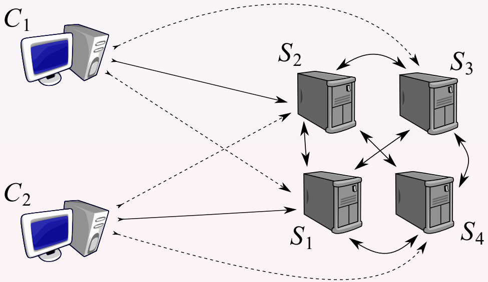
Combining client-centric and data-centric consistency models

May 24, 2017

Outline

Outline

Clients & Servers



Problem formulation

$$\text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}S$$
$$\text{c-c@}C \leftrightarrow S \quad \Leftarrow \quad \text{d-c@}S$$

$$\text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}C$$
$$\text{c-c@}C \leftrightarrow S \quad \Leftarrow \quad \text{d-c@}C$$

$$\text{d-c@}S \quad \overset{?}{\Rightarrow} \quad \text{c-c@}C \leftrightarrow S$$

$$\text{d-c@}S \wedge \text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}C$$

Problem formulation

$$\text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}S$$
$$\text{c-c@}C \leftrightarrow S \quad \Leftarrow \quad \text{d-c@}S$$

$$\text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}C$$
$$\text{c-c@}C \leftrightarrow S \quad \Leftarrow \quad \text{d-c@}C$$

$$\text{d-c@}S \quad \overset{?}{\Rightarrow} \quad \text{c-c@}C \leftrightarrow S$$

$$\text{d-c@}S \wedge \text{c-c@}C \leftrightarrow S \quad \overset{?}{\Rightarrow} \quad \text{d-c@}C$$

$\text{d-c} \in \{\text{seq, caus, PRAM, cache, proc}\}$
$\text{c-c} \subseteq \{\text{RYW, MW, MR, WFR}\}$

Notation

$\mathcal{X}$ the set of shared read-write objects (stored at servers, fully replicated),

$w_i(x_j)v$ write of a value v to the replica of an object $x \in \mathcal{X}$ at a server S_j issued by a client C_i

$r_i(x_j)v$ read from the replica of an object $x \in \mathcal{X}$ at a server S_j returning a value v to a client C_i

If some elements of the notation are not important or clear in the context they are used, they can be omitted. However, to emphasize that any replica of a given object, say x, is addressed, the notation x_* is used.

Order of operations

Definition 1. *An operation o1 precedes an operation o2 in **issue order** , iff they are requested by the same client, say C_i, and the request to execute o1 is sent before the request to execute o2 (which is denoted* $o1 \overset{C_i}{\rightharpoondown} o2$*).*

Definition 2. *An operation o1 precedes an operation o2 in **acceptance order**, iff the requests for these operations are accepted by the same server, say S_j, and o1 is executed by this server before o2 (which is denoted* $o1 \overset{S_j}{\rightharpoondown} o2$*).*

View

Definition 3. *Let $\mathcal{O}$ be a set of read and write operations, and $\rightarrowtail$ a total order relation in this set. View, $(\mathcal{O}, \rightarrowtail)$, is the set $\mathcal{O}$ ordered by the relation $\rightarrowtail$, provided that the following condition is satisfied:*

$$\forall_{x \in \mathcal{X}} \ \forall_{w(x)v, r(x)v \in \mathcal{O}} \ w(x)v \rightarrowtail r(x)v$$

Legality

Definition 4. *View $(\mathcal{O}, \rightarrowtail)$ on the server side is legal (s-legal for clarity) if it satisfies the following condition:*

$$\underset{x\in\mathcal{X}}{\forall} \; \underset{w(x)v, r(x)v\in\mathcal{O}}{\forall} \; \underset{w(x)u\in\mathcal{O}}{\nexists} (u \neq v \wedge w(x)v \rightarrowtail w(x)u \rightarrowtail r(x)v)$$

Definition 5. *View $(\mathcal{O}, \rightarrowtail)$ on the client side is legal (c-legal for clarity) if it satisfies the following condition:*

$$\underset{x\in\mathcal{X}}{\forall} \; \underset{r(x_j)v\in\mathcal{O}}{\forall} \; \underset{w(x_j)v\in\mathcal{O}}{\exists} \Big(w(x_j)v \rightarrowtail r(x_j)v \wedge \underset{w(x)u\in\mathcal{O}}{\nexists} (u \neq v \wedge w(x_j)v \rightarrowtail w(x)u \rightarrowtail r(x_j)v)\Big)$$

Sets of operations

$\mathcal{O}_{C_i} = \mathcal{O}^R_{C_i} \cup \mathcal{O}^W_{C_i}$ — the set of operations issued by C_i

$\mathcal{O}_{S_j} = \mathcal{O}^R_{S_j} \cup \mathcal{O}^W_{S_j}$ — the set of operations executed by S_j

$\mathcal{O}^R_{S_j}$ — the set of all read operations executed by S_j as a result of client request

$\mathcal{O}^R_{S_j}/_{C_i} \subseteq \mathcal{O}^R_{S_j}$ — the set of all read operations executed by S_j as a result of a request by C_i

$\mathcal{O}^W_{S_j}$ — the set of all write operations executed by S_j as the result of both direct request from a client and cooperation with other servers, including a write operation defining an initial value (not issued by any client)

View (2)

Definition 6. *View on the server side for a server S_j is an s-legal view of the set $\mathcal{O}_{S_j}$ (denoted $(\mathcal{O}_{S_j}, \overset{S_j}{\rightarrowtail})$).*

Definition 7. *View on the client side for a client C_i is a c-legal view of the set $\mathcal{O}_{C_i}$ (denoted $(\mathcal{O}_{C_i}, \overset{C_i}{\rightarrowtail})$).*

Outline

Equivalence

Definition 8. *Two write operations — $w(x_*)v$, $w(y_*)u$ — are equivalent ($w(x_*)v \doteq w(y_*)u$) iff $x = y$ and $u = v$.*

Definition 9. *Two views of a set of operations, say $(\mathcal{O}_i, \stackrel{i}{\rightarrowtail})$ and $(\mathcal{O}_j, \stackrel{j}{\rightarrowtail})$, are equivalent if all the following conditions hold:*

$$(i) \quad \mathcal{O}_i^R = \mathcal{O}_j^R$$

$$(ii) \quad \mathcal{O}_i^W / \doteq \; = \mathcal{O}_j^W / \doteq$$

$$(iii) \quad \forall_{o1, o2 \in O_i^R} \left(o1 \stackrel{i}{\rightarrowtail} o2 \Leftrightarrow o1 \stackrel{j}{\rightarrowtail} o2 \right)$$

Local order

Definition 10. *Two operations, $o1 \in \mathcal{O}$ and $o2 \in \mathcal{O}$, are in local order on the client side iff they are in issue order, i.e.:*

$$o1 \xrightarrow{lo} o2 \Leftrightarrow \exists_{C_i} o1 \xrightarrow{C_i} o2$$

Definition 11. *Two operations, $o1 \in \mathcal{O}$ and $o2 \in \mathcal{O}$, are in local order on the server side iff they are in acceptance order, i.e.:*

$$o1 \xrightarrow{lo} o2 \Leftrightarrow \exists_{S_j} o1 \xrightarrow{S_j} o2$$

Causal order

Definition 12. *The operations, $o1 \in \mathcal{O}$ and $o2 \in \mathcal{O}$, are in causal order ($o1 \stackrel{co}{\rightarrow} o2$) iff one of the following conditions holds:*

$$
\begin{array}{ll}
(i) & o1 \stackrel{lo}{\rightarrow} o2 \\
(ii) & \underset{x \in \mathcal{X}}{\exists} (o1 = w(x)v \wedge o2 = r(x)v) \\
(iii) & \underset{o' \in \mathcal{O}}{\exists} \left(o1 \stackrel{co}{\rightarrow} o' \wedge o' \stackrel{co}{\rightarrow} o2 \right)
\end{array}
$$

Outline

Sequential consistency

Definition 13. *Sequential consistency is preserved if the views — one for each process — are equivalent to ones that are s-legal and satisfy the following conditions:*

$$(i) \quad \underset{P_i}{\forall} \ \underset{o1,o2 \in \mathcal{O}_{P_i} \cup \mathcal{O}^W}{\forall} \left(o1 \overset{lo}{\rightharpoondown} o2 \Rightarrow o1 \overset{P_i}{\rightarrowtail} o2 \right)$$

$$(ii) \quad \underset{w1,w2 \in \mathcal{O}^W}{\forall} \left(\underset{P_i}{\forall} w1 \overset{P_i}{\rightarrowtail} w2 \vee \underset{P_i}{\forall} w2 \overset{P_i}{\rightarrowtail} w1 \right)$$

Causal consistency

Definition 14. *Causal consistency is preserved if the views — one for each process — are equivalent to ones that are s-legal and satisfy the following condition:*

$$\underset{P_i}{\forall} \underset{o1,o2 \in \mathcal{O}_{P_i} \cup \mathcal{O}^W}{\forall} \left(o1 \stackrel{co}{\rightharpoondown} o2 \Rightarrow o1 \stackrel{P_i}{\rightarrowtail} o2 \right)$$

PRAM consistency

Definition 15. *PRAM consistency is preserved if the views — one for each process — are equivalent to ones that are s-legal and satisfy the following condition:*

$$\underset{P_i}{\forall} \underset{o1,o2 \in \mathcal{O}_{P_i} \cup \mathcal{O}^W}{\forall} \left(o1 \overset{lo}{\rightharpoondown} o2 \Rightarrow o1 \overset{P_i}{\rightarrowtail} o2 \right)$$

Cache consistency

Definition 16. *Cache consistency is preserved if the views — one for each process — are equivalent to ones that satisfy the following condition:*

$$\underset{x \in \mathcal{X}}{\forall} \; \underset{w1, w2 \in \mathcal{O}^W|\{x\}}{\forall} \left(\underset{P_i}{\forall} \, w1 \overset{P_i}{\rightarrowtail} w2 \vee \underset{P_i}{\forall} \, w2 \overset{P_i}{\rightarrowtail} w1 \right)$$

Processor consistency

Definition 17. *Processor consistency is preserved if the views — one for each process — are equivalent to ones that are s-legal and satisfy the following conditions:*

$$(i) \qquad \underset{P_i}{\forall} \; \underset{o1,o2 \in \mathcal{O}_{P_i} \cup \mathcal{O}^W}{\forall} \left(o1 \overset{lo}{\rightharpoondown} o2 \Rightarrow o1 \overset{P_i}{\rightarrowtail} o2 \right)$$

$$(ii) \qquad \underset{x \in \mathcal{X}}{\forall} \; \underset{w1,w2 \in \mathcal{O}^W|\{x\}}{\forall} \left(\underset{P_i}{\forall} w1 \overset{P_i}{\rightarrowtail} w2 \vee \underset{P_i}{\forall} w2 \overset{P_i}{\rightarrowtail} w1 \right)$$

Met vs. obtained data-centric model

Met data-centric model provided by servers independently of the interaction with clients

Obtained data-centric model resulting from session guarantees imposed on the interaction with clients

Met vs. obtained data-centric model

Met data-centric model provided by servers independently of the interaction with clients

Obtained data-centric model resulting from session guarantees imposed on the interaction with clients

Example for PRAM:

- met on the server side
$$\forall_{S_i}\ \forall_{o1,o2\in\mathcal{O}_{S_i}\cup\mathcal{O}^W}\left(\left(\exists_{S_j}\, o1 \overset{S_j}{\rightarrow} o2\right) \Rightarrow o1 \overset{S_i}{\rightarrowtail} o2\right)$$
- forced on the server side
$$\forall_{S_i}\ \forall_{o1,o2\in\mathcal{O}_{S_i}\cup\mathcal{O}^W}\left(\left(\exists_{C_j}\, o1 \overset{C_j}{\rightarrow} o2\right) \Rightarrow o1 \overset{S_i}{\rightarrowtail} o2\right)$$
- forced on the client side
$$\forall_{C_i}\ \forall_{o1,o2\in\mathcal{O}_{C_i}\cup\mathcal{O}^W}\left(\left(\exists_{C_j}\, o1 \overset{C_j}{\rightarrow} o2\right) \Rightarrow o1 \overset{C_i}{\rightarrowtail} o2\right)$$

Outline

Read-Your-Writes

Definition 18. *RYW guarantee is preserved if the views on the server side are s-legal and satisfy the following condition:*

$$\underset{C_i}{\forall}\ \underset{S_j}{\forall} \left(w(x)v \overset{C_i}{\rightarrow} r(y_j)u \Rightarrow w(x)v \overset{S_j}{\rightarrowtail} r_i(y)u \right)$$

Corollary 1. *If RYW is met, the view on the client side satisfies the following condition:*

$$\underset{C_i}{\forall} \left(w(x)v \overset{C_i}{\rightarrow} r(y)u \Rightarrow \underset{S_j}{\forall}\, w(x_j)v \overset{C_i}{\rightarrowtail} r(y_j)u \right)$$

Monotonic-Writes

Definition 19. *MW guarantee is preserved if the views on the server side are s-legal and satisfy the following condition:*

$$\underset{C_i}{\exists}\, w(x)v \overset{C_i}{\rightharpoonup} w(y)u \Rightarrow \underset{S_j}{\forall}\, w_i(x)v \overset{S_j}{\rightarrowtail} w_i(y)u$$

Corollary 2. *If MW is met, the view on the client side satisfies the following condition:*

$$\underset{C_i}{\forall} \left(w(x)v \overset{C_i}{\rightharpoonup} w(y)u \Rightarrow \underset{S_j}{\forall}\, w(x_j)v \overset{C_i}{\rightarrowtail} w(y_j)u \right)$$

Monotonic-Reads

Definition 20. *MR guarantee is preserved if the views on the server side are s-legal and satisfy the following condition:*

$$\underset{C_i}{\forall}\ \underset{S_j}{\forall}\left(r(x_*)v \overset{C_i}{\rightharpoondown} r(y_j)u \Rightarrow w(x)v \overset{S_j}{\rightarrowtail} r_i(y)u\right)$$

Corollary 3. *If MR is met, the view on the client side satisfies the following condition:*

$$\underset{C_i}{\forall}\left(r(x_*)v \overset{C_i}{\rightharpoondown} r(y_j)u \Rightarrow w(x_j)v \overset{C_i}{\rightarrowtail} r(y_j)u\right)$$

Writes-Follow-Reads

Definition 21. *WFR guarantee is preserved if the views on the server side are s-legal and satisfy the following condition:*

$$\underset{C_i}{\exists}\, r(x_*)v \overset{C_i}{\rightharpoonup} w(y)u \Rightarrow \underset{S_j}{\forall}\, w(x)v \overset{S_j}{\rightarrowtail} w_i(y)u$$

Corollary 4. *If WFR is met, the view on the client side satisfies the following condition:*

$$\underset{C_i}{\forall} \left(r(x_*)v \overset{C_i}{\rightharpoonup} w(y)u \Rightarrow \underset{S_j}{\forall}\, w(x_j)v \overset{C_i}{\rightarrowtail} w(y_j)u \right)$$

Outline

The impact of sequential consistency

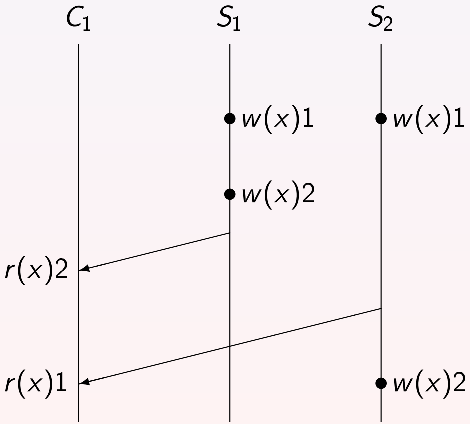
Theorem 1. *If sequential consistency is preserved on the server side, writes-follow-reads guarantee is satisfied in the client-server interaction (Seq $\Rightarrow$ WFR).*

The impact of sequential consistency

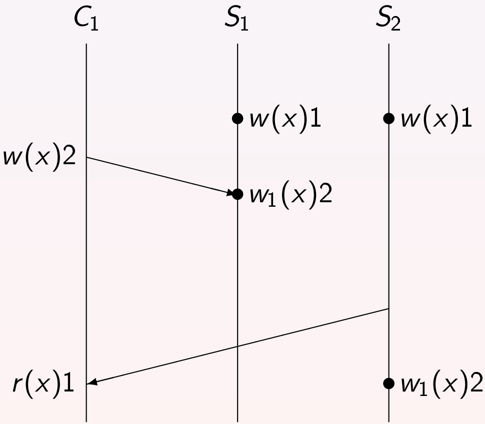
Theorem 2. *If sequential consistency is preserved on the server side, writes-follow-reads guarantee is satisfied in the client-server interaction (Seq ⇒ WFR).*

Proof. Let us assume that a write operation — $w(x)v$ — violates WFR, while the servers preserve sequential consistency. This means that the client issuing the write operation has read a value — u ($r(y)u$) — from another server, and the currently used server is not aware of the operation that defined the value ($w(y)u$). Consequently, the operations $w(x)v$ and $w(y)u$ are observed by the servers in the reverse order, which violates sequential consistency. □

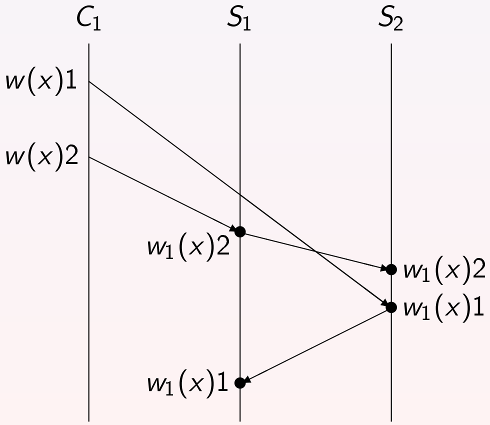
Violation of MR in sequentially consistent execution



Violation of RYW in sequentially consistent execution



Violation of MW in sequentially consistent execution



Sequential consistency under additional assumptions

Corollary 5. *In the case of synchronous call, sequential consistency ensures MW guarantee.*

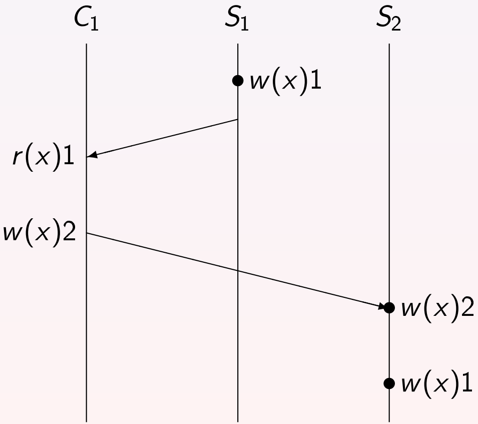
Corollary 6. *Although sequential consistency does not ensure MR guarantee, MR is satisfied, if a client issues a (perhaps dummy) write before a read operation, provided that RYW is fulfilled.*

The impact of weaker data-centric models

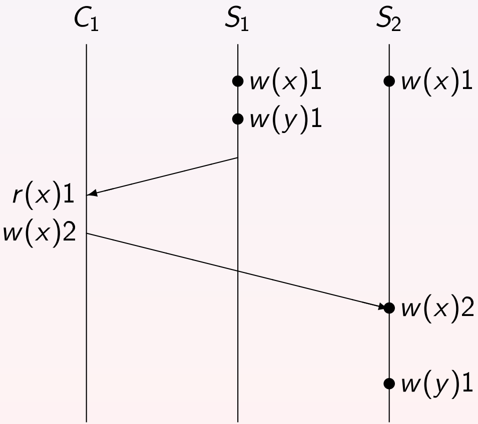
As for weaker data-centric consistency models, they cannot offer more than sequential consistency. Hence, there is the question of writes-follow-reads guarantee.

Cache consistency (thereby processor consistency) ensure WFR with respect to a given variable.

Violation of WFR in causally consistent execution

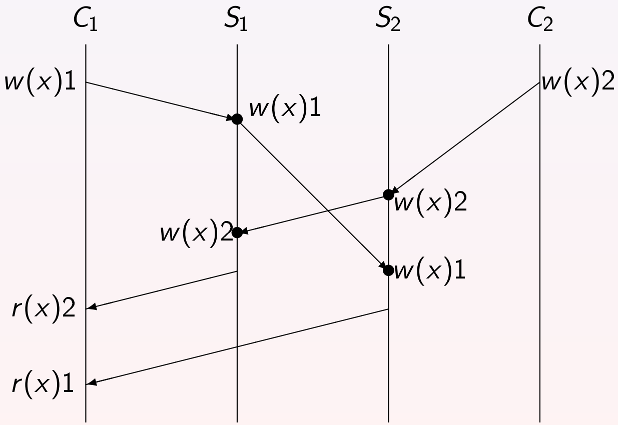


Violation of WFR with respect to y in processor consistent execution



Outline

Violation of s-legality on the client side



Which session guarantees are violated?

Preservation of s-legality

Cache consistency on the server side is the necessary consistency condition to ensure any data-centric model on the client side!

Lemma 1. *If all session guarantees (i.e. RYW, MW, MR, WFR) are satisfied in the client-server interaction and the views on the server side are s-legal, the view on the client side is equivalent to an s-legal one.*

Theorems for pure cc model — PRAM consistency

Theorem 3. *On the server side, PRAM consistency is obtained if and only if RYW and MW guarantees are preserved.*

Theorems for pure cc model — PRAM consistency

Theorem 5. *On the server side, PRAM consistency is obtained if and only if RYW and MW guarantees are preserved.*

Lemma 4. *If RYW, MW, and MR are preserved, client's view obtains local order.*

Lemma 5. *If RYW, MW, and MR are preserved, client's view is legal.*

Theorem 6. *On the client side, PRAM consistency is obtained if RYW, MW, and MR are preserved.*

Theorems for pure cc model — causal consistency

Theorem 7. *On the server side, causal consistency is obtained if and only if RYW, MW, WFR and MR guarantees are preserved.*

Theorems for pure cc model — causal consistency

Theorem 9. *On the server side, causal consistency is obtained if and only if RYW, MW, WFR and MR guarantees are preserved.*

Lemma 7. *If RYW, MW, WFR and MR are preserved, client's view obtains causal order.*

Theorem 10. *On the client side, causal consistency is obtained if RYW, MW, WFR and MR are preserved.*

Theorems for pure cc model — conclusions

c-c@$C \leftrightarrow S$	$\substack{\Rightarrow \\ \Leftarrow}$	d-c@$\substack{S \\ C}$
{RYW, MW}	$\Leftrightarrow$	PRAM@S
{RYW, MW, MR}	$\Rightarrow$	PRAM@C
{RYW, MW, MR, WFR}	$\Leftrightarrow$	causal@S
{RYW, MW, MR, WFR}	$\Rightarrow$	causal@C

Theorems for pure cc model — conclusions

c-c@$C \leftrightarrow S$	$\substack{\Rightarrow \\ \Leftarrow}$	d-c@$\substack{S \\ C}$
{RYW, MW}	$\Leftrightarrow$	PRAM@S
{RYW, MW, MR}	$\Rightarrow$	PRAM@C
{RYW, MW, MR, WFR}	$\Leftrightarrow$	causal@S
{RYW, MW, MR, WFR}	$\Rightarrow$	causal@C

Corollary 8. *In fact, because of cache consistency on the server side, processor consistency (instead of PRAM) and appropriately stronger consistency then causal is actually obtained.*

Theorems for combined cc and dc models — sequential consistency

Theorem 11. *If sequential consistency is met on the server side, and RYW, MW and MR guarantees are imposed on the client-server interaction, sequential consistency is obtained on the client side.*

Theorems for combined cc and dc models — causal consistency

Theorem 12. *If causal consistency is met on the server side, and RYW, MW, MR and WFR guarantees are imposed on the client-server interaction, causal consistency is obtained on the client side.*

Theorems for combined cc and dc models — PRAM consistency

Theorem 13. *If PRAM consistency is met on the server side, and RYW, MW and MR guarantees are imposed on the client-server interaction, PRAM consistency is fulfilled on the client side.*

Theorems for combined cc and dc models — cache consistency

Theorem 14. *If cache consistency is met on the server side, and RYW and MR guarantees are imposed on the client-server interaction, cache consistency is fulfilled on the client side.*

Theorems for combined cc and dc models — conclusions

$\mathrm{d-c@}S$ (met)	$\wedge$	$\mathrm{c-c@}C \leftrightarrow S$	$\Rightarrow$	$\mathrm{d-c@}C$ (obtained)
sequential		{RYW, MW, MR}		sequential
causal		{RYW, MW, MR, WFR}		causal
PRAM		{RYW, MW, MR}		PRAM
cache		{RYW, MR}		cache
model x		?		weaker than x