Data-Centric Consistency - Disposition

Lukas Peter Jørgensen, 201206057, DA4

26. marts 2014

1 Terms

Consistency model	A contract between processes and the data store, if processes agree to obey certain rules, the store promises to work correctly.

2 Replicating data

Two main reasons for replicating data: **Reliability** and **performance**. It becomes reliable because, if one replica crashes, you can just switch to another one, and you can perform on n replicas, and assume that the most common answer is the correct one (in case some doesn't reply the same answer as the other because of corruption).
Performance comes from being able to divide work over multiple servers, or by placing a replica closer to the client (geographically) so the latency is smaller. (Like the Netflix vs Verizon issue).

Cons:
The cons of replicating data is the trouble with keeping the data consistent.

3 Continuous consistency

Distinguish 3 independent axes for defining inconsistencies:

1. Deviation in numerical values between replicas.
2. Deviation in staleness between replicas.
3. Deviation with respect to the ordering of update operations.

Numerical deviation

Useful for when the data have numerical semantics e.g. stock market prices.
Distinguishing between absolute and relative numerical deviation

Numerical deviation could as well be understood as the number of updates applied to a given replica, which haven't been seen by others. In this case the deviation in the value is also referred to as its weight.

Staleness deviation

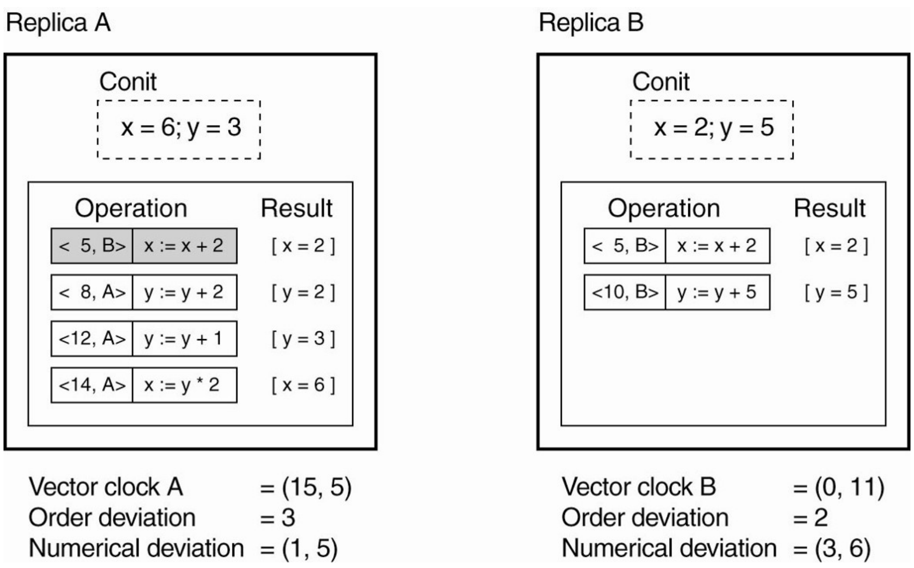
Staleness deviations relate to the last time a replica was updated. For some applications it can be tolerated that a replica provides old data, as long as it is not *too* old.

Order deviation

Finally there are classes of applications in which the ordering of updates are allowed to be different at the various replicas, as long as the differences remain bounded.

Conit

A consistency unit, a conit may be a record representing a single stock, or a weather report.



Replica A recieved the operation $< 5, B > x := x + 2$ from replica B and has committed it.
Replica A also has three tentative update operations: $< 8, A >, < 12, A >$ and $< 14, A >$, which brings it's ordering deviation to 3, A's vector clock becomes $< 15, 5 >$ because the last operation $< 14, A >$ is followed by an incrementation.
The only operation from B that A has not yet seen is $< 10, B >$, therefore it's numerical deviation with respect to operation is 1, the weight of the deviation can be expressed as the maximum difference between the committed values of x and y at A, and the result from operations at B not seen by A. The committed value at A is $(x, y) = (2, 0)$ whereas the for A unseen-operation at B yields a difference of $y = 5$, therefore the Numerical deviation for A is $(1, 5)$

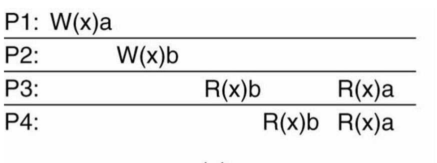
Similarly for B. It has two tentative update operations: $< 5, B >$ and $<$

$10, B >$, which means it has an order deviation of 2. Because B has not yet seen a single operation from A, it's vector clock is $(0, 11)$. The numerical deviation is 3, with a total weight of 6, because B's committed value is: $(x, y) = (0, 0)$, whereas the tentative operations at A will bring x to 6.

4 Sequential consistency

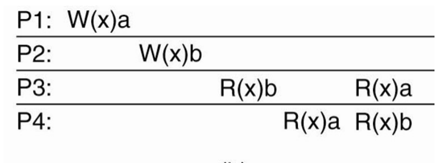
A data store is said to be sequentially consistent when it satisfies the following condition:

When processes run concurrently on (possibly) different machines, any valid interleaving of read and write operations is acceptable behaviour, but *all processes must see the same interleaving of operations.*



P1:	W(x)a				
P2:		W(x)b			
P3:			R(x)b		R(x)a
P4:				R(x)b	R(x)a

(a)



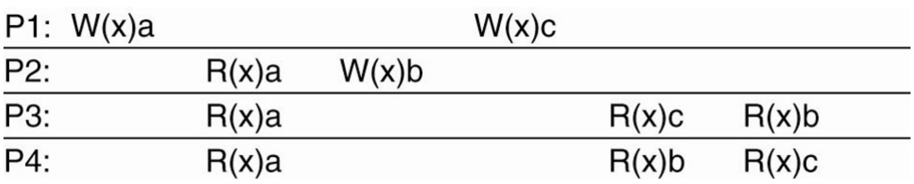
P1:	W(x)a				
P2:		W(x)b			
P3:			R(x)b		R(x)a
P4:				R(x)a	R(x)b

(b)

(a) is valid because all processes see the same interleaving, while (b) is not because $P3$ and $P4$ see different interleavings, ending up with different values for x.

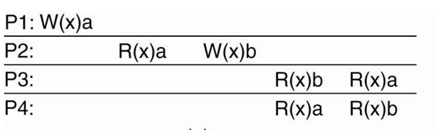
5 Causal consistency

Much like sequential consistency, but only for potentially causally related events.



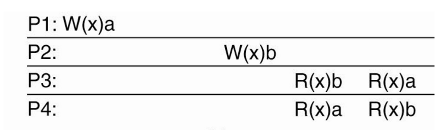
P1:	W(x)a			W(x)c		
P2:		R(x)a	W(x)b			
P3:		R(x)a			R(x)c	R(x)b
P4:		R(x)a			R(x)b	R(x)c

These events are causally consistent but not sequentially consistent, because $W(x)c$ and $W(x)b$ are concurrent (i.e. not causally related).



P1:	W(x)a				
P2:		R(x)a	W(x)b		
P3:				R(x)b	R(x)a
P4:				R(x)a	R(x)b

(a)



P1:	W(x)a			
P2:		W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

(b)

Here (a) is not causally consistent because