Large Scale Distributed Systems

Sistemas Distribuídos em Larga/Grande Escala

Carlos Baquero
Universidade do Minho

MIEI SDLE 2021

Online


REGRAS DE ETIQUETA NAS AULAS ONLINE
Eis as principais regras que deves respeitar durante as aulas por videoconferência nas várias disciplinas.
USA ROUPA ADEQUADA
Deverás usar roupa adequada; não apareças de tronco nu, de pijama ou com roupa que não usarias na escola. Estás numa aula; veste-te de acordo com a ocasião.
AÇÃO!
CÂMARA E MICROFONE
A câmara deverá estar sempre ligada, a não ser que o professor não o exija. O microfone deverá estar desligado, exceto quando em discussão em grupo ou quando o professor indicar.
DEVES SER PONTUAL
Tal como na escola, deves entrar na sala de aula virtual atempadamente (5 minutos antes) para garantir que consegues aceder à aula.
MATERIAL
Antes da aula começar, deves ter o material necessário junto a ti: caderno, material de escrita, manual e caderno de atividades.
COMIDA E BEBIDA
Durante as aulas, não poderás comer (tal como na escola). Podes ter uma garrafa de água, de sumo ou chá, para não teres que te ausentar durante a aula.
EVITA DISTRAÇÕES
Desliga o telemóvel, desativa as notificações, encerra a sessão no Facebook, Twitter, Instagram, TikTok e outras Redes Sociais, de forma a minimizar as distrações. Se tens irmãos mais novos, pede, se possível, que tomem conta deles, para que possas estar a 100% na aula.
SÊ EDUCADO!
Ser bem educado significa aguardar a sua vez de falar (coloca a mão no ar), respeitar os colegas, respeitar os professores, usar linguagem adequada e estar no seu melhor comportamento.
COLOCA QUESTÕES E ESCLARECE DÚVIDAS
Deves aproveitar as aulas síncronas para colocar questões e esclarecer as tuas dúvidas. Em grupo, torna-se tudo mais fácil; vais ver que algum colega até vai ter uma boa ideia que te ajudará.
O(A) professor(a) também estará disponível para te ajudar. Aproveita estes momentos!
Filomena Mourinho | mrsmourinhosclass.wordpress.com

Plan

- Models and limitations in distributed computing
- Scalable distributed topologies
- Design of large scale systems
- Clocks and scalable logical time
- Data synchronization and eventual consistency

Grading

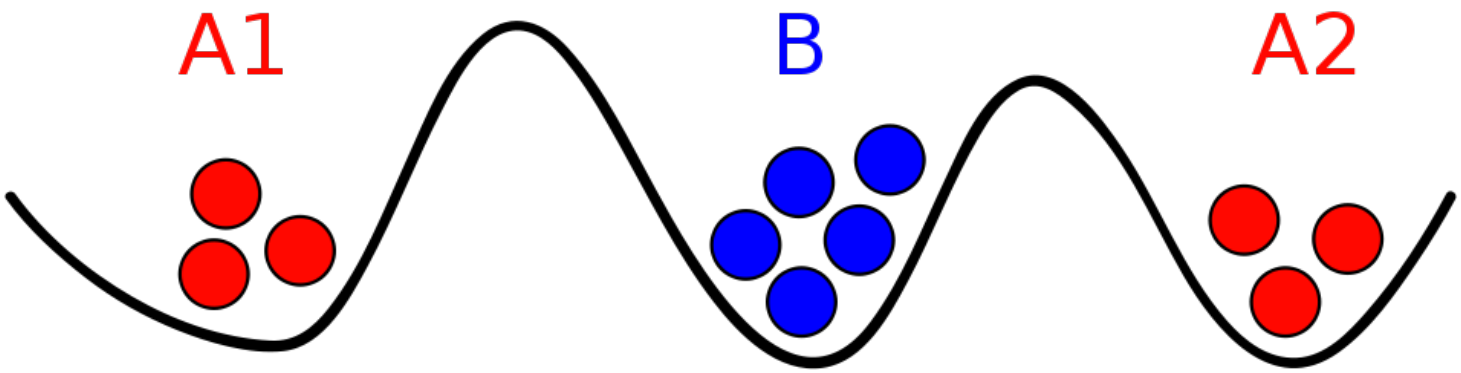
- A: Class activities and particiation 20%
- G: Group project assignment 50%
- E: Individual test/exam 30%
- Both G and E have minimum grade of 10 in 20

Two (gangster) Generals Paradox

A group of gangsters are about to pull off a big job. The plan of action is prepared down to the last detail: Some of the men are holed up in a warehouse across town, awaiting precise instructions. It is absolutely essential that the two groups act with complete reliance on each other in executing the plan.

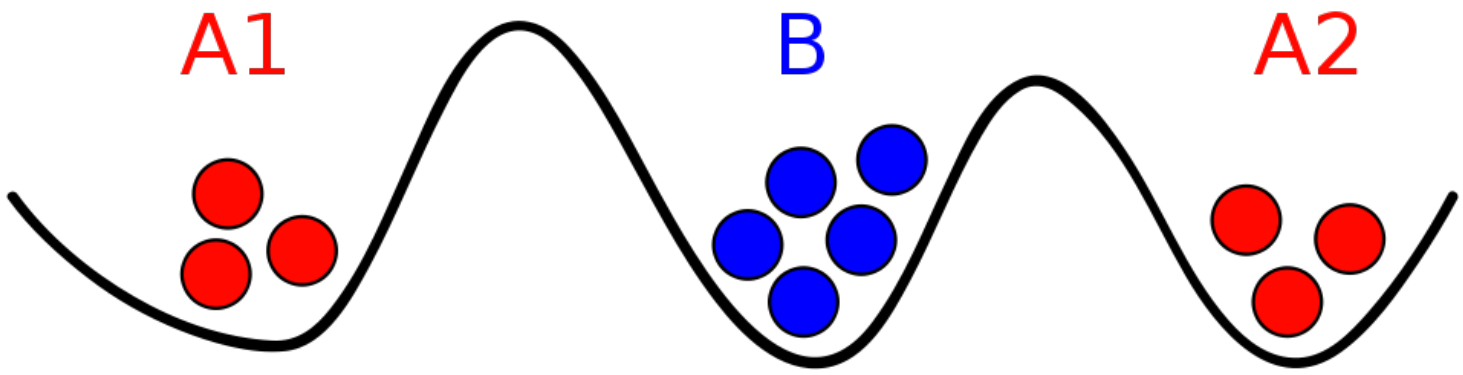
in "Some Constraints and Trade-offs in the Design of Network Communications". Akkoyunlu, Ekanadham and Huber. 1975.

Two (gangster) Generals Paradox



- Red gangsters have more mobsters if together, but need to attack at the same time.
- Messengers are unreliable.
- How to coordinate an attack?

Two (gangster) Generals Paradox



- Red gangsters have more mobsters if together, but need to attack at the same time.
- Messengers are unreliable.
- How to coordinate an attack?

`http://en.wikipedia.org/wiki/Two_Generals'_Problem`

Synchronous model

Expected in small scale and cost dominated. Ex: Cars subsystems

- Processing delays have a known bound
- Message delivery delays have a known bound
- Rate of drift of local clocks has a known bound
- Difference between local clocks has a known bound

in “Distributed Systems for System Architects”. Veríssimo and Rodrigues. 2001.

Asynchronous model

Adapted to large scale and can spread costs. Ex: Internet

- Processing delays are unbounded or unknown
- Message delivery delays are unbounded or unknown
- Rate of drift of local clocks is unbounded or unknown
- Difference between local clocks is unbounded or unknown

in “Distributed Systems for System Architects”. Veríssimo and Rodrigues. 2001.

EC and CAP for Geo-Replication

Eventually Consistent. CACM 2009, Werner Vogels

- In an ideal world there would be only one consistency model: when an update is made all observers would see that update.
- Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.

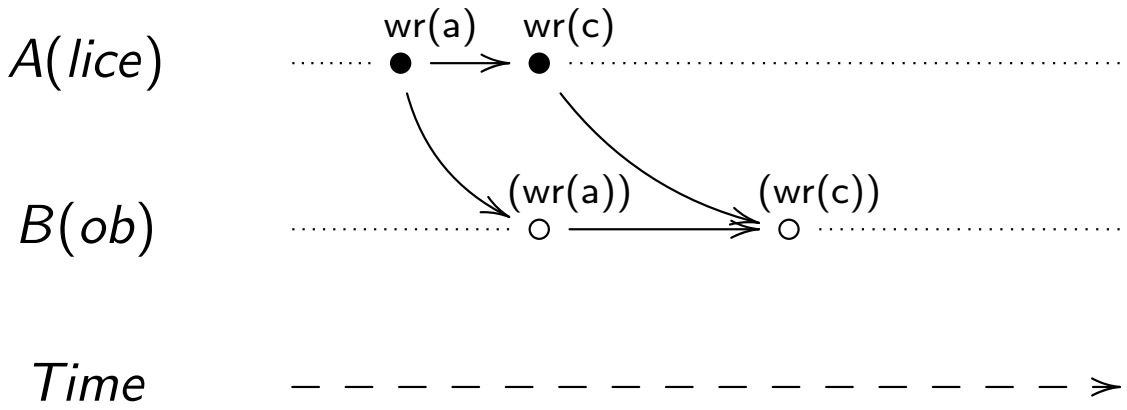
CAP theorem. PODC 2000, Eric Brewer

Of three properties of shared-data systems – data consistency, system availability, and tolerance to network partition – only two can be achieved at any given time.

We will focus on AP.

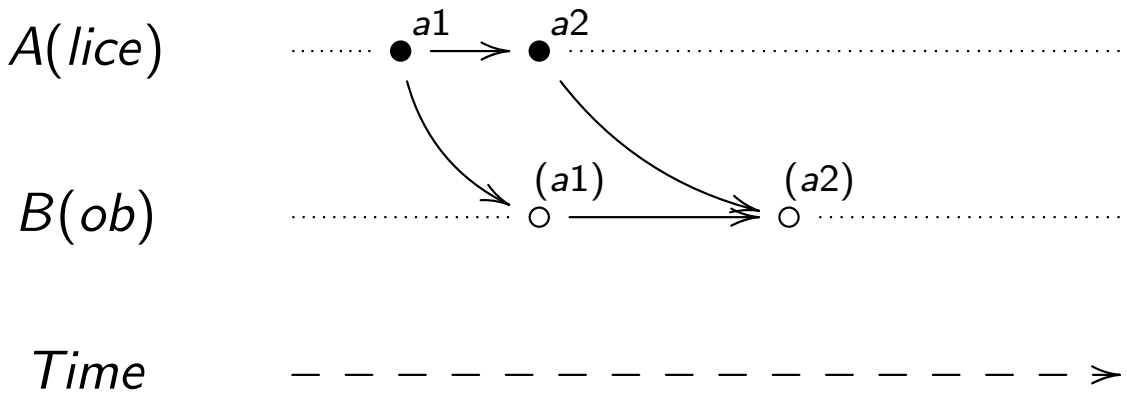
Reliable source FIFO delivery

Register with write operations



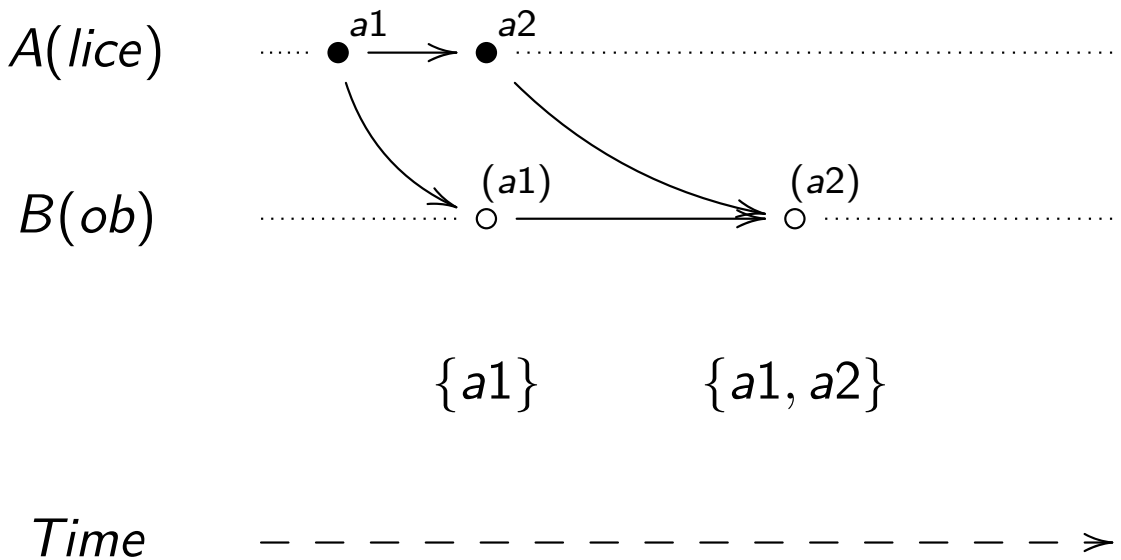
Reliable source FIFO delivery

Unique tags (dots) support retries and FIFO ordering



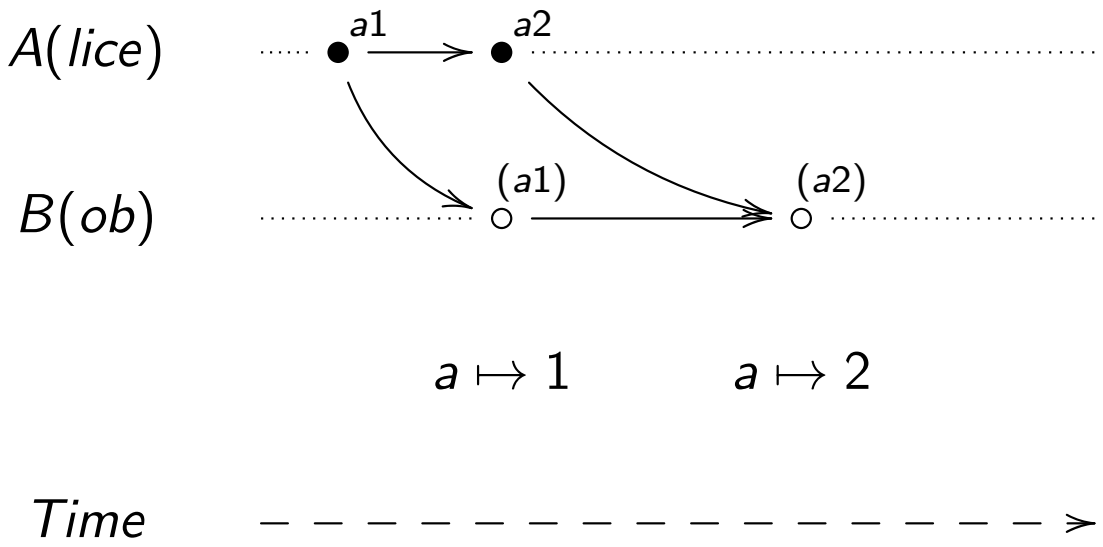
Reliable source FIFO delivery

Unique tags (dots) support retries and FIFO ordering



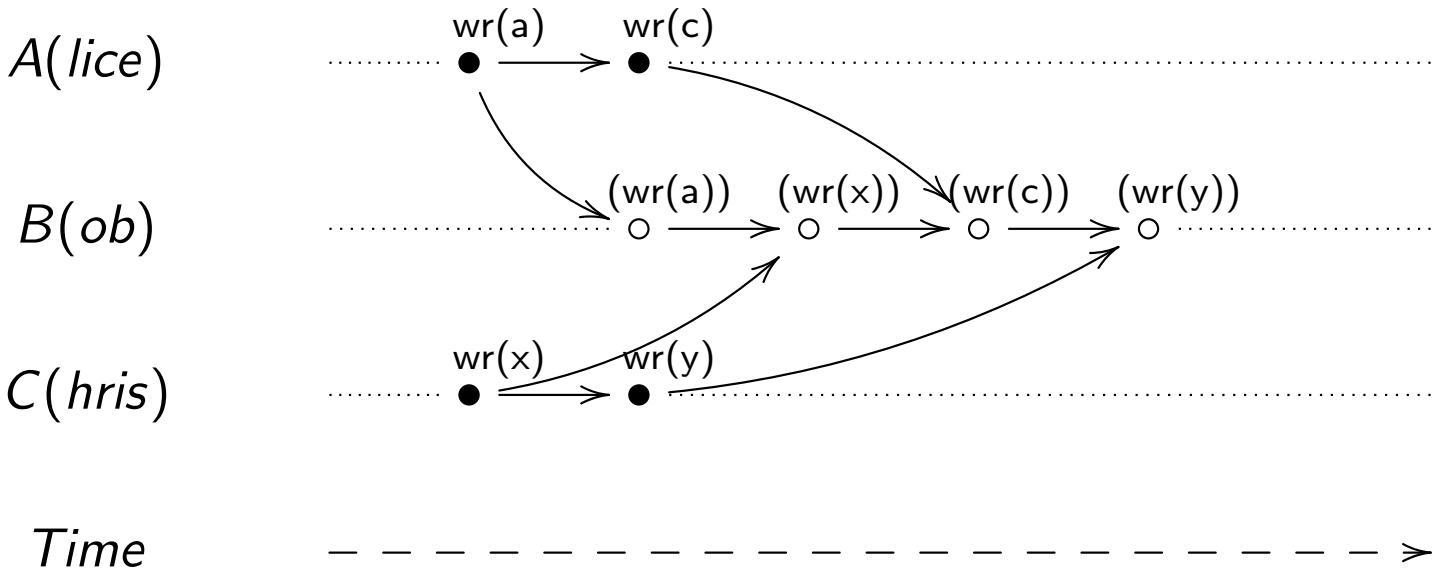
Reliable source FIFO delivery

Initial sequences of dots can be compacted into a vector entry



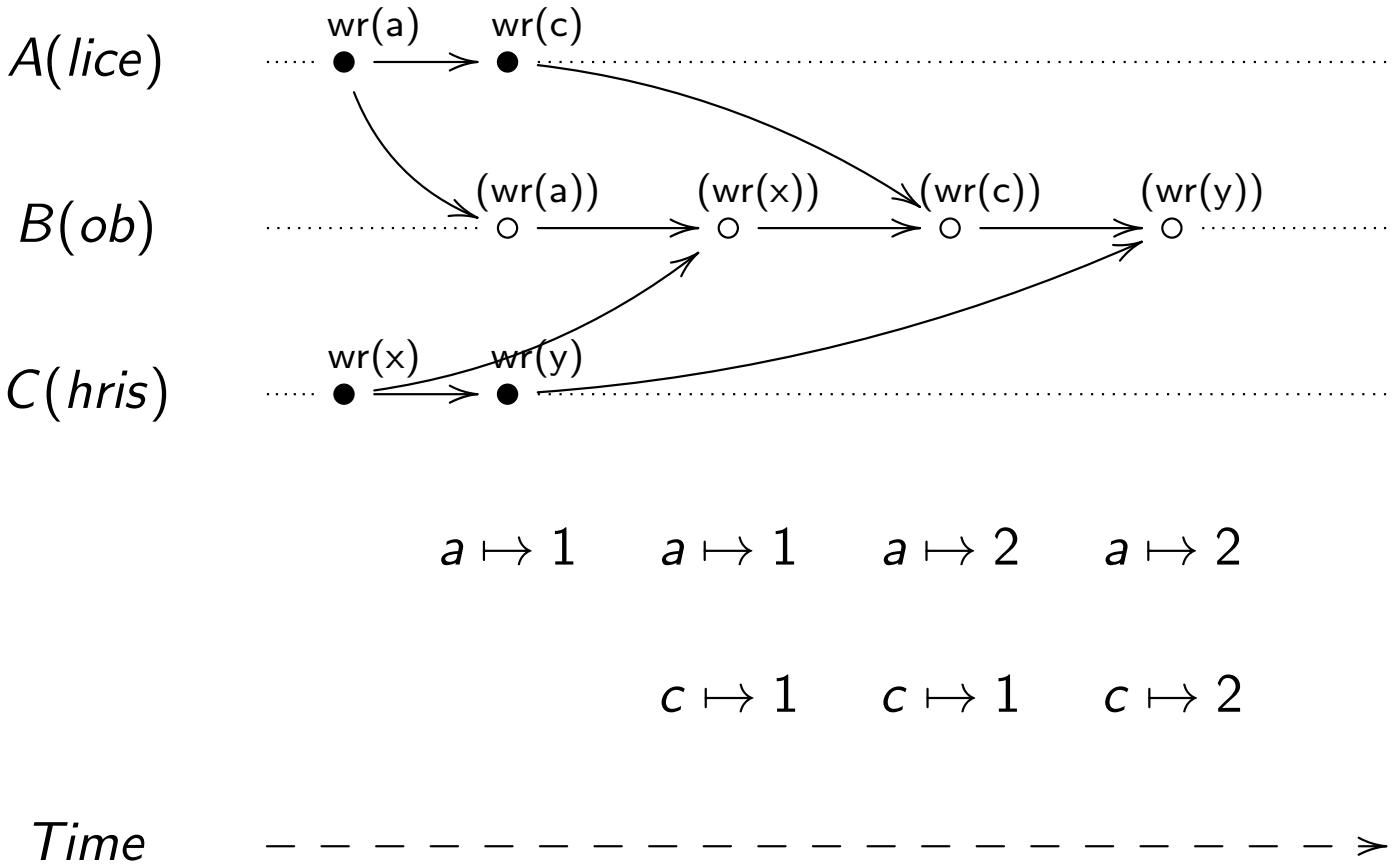
Reliable source FIFO delivery

FIFO allows arbitrary interleaving of sources



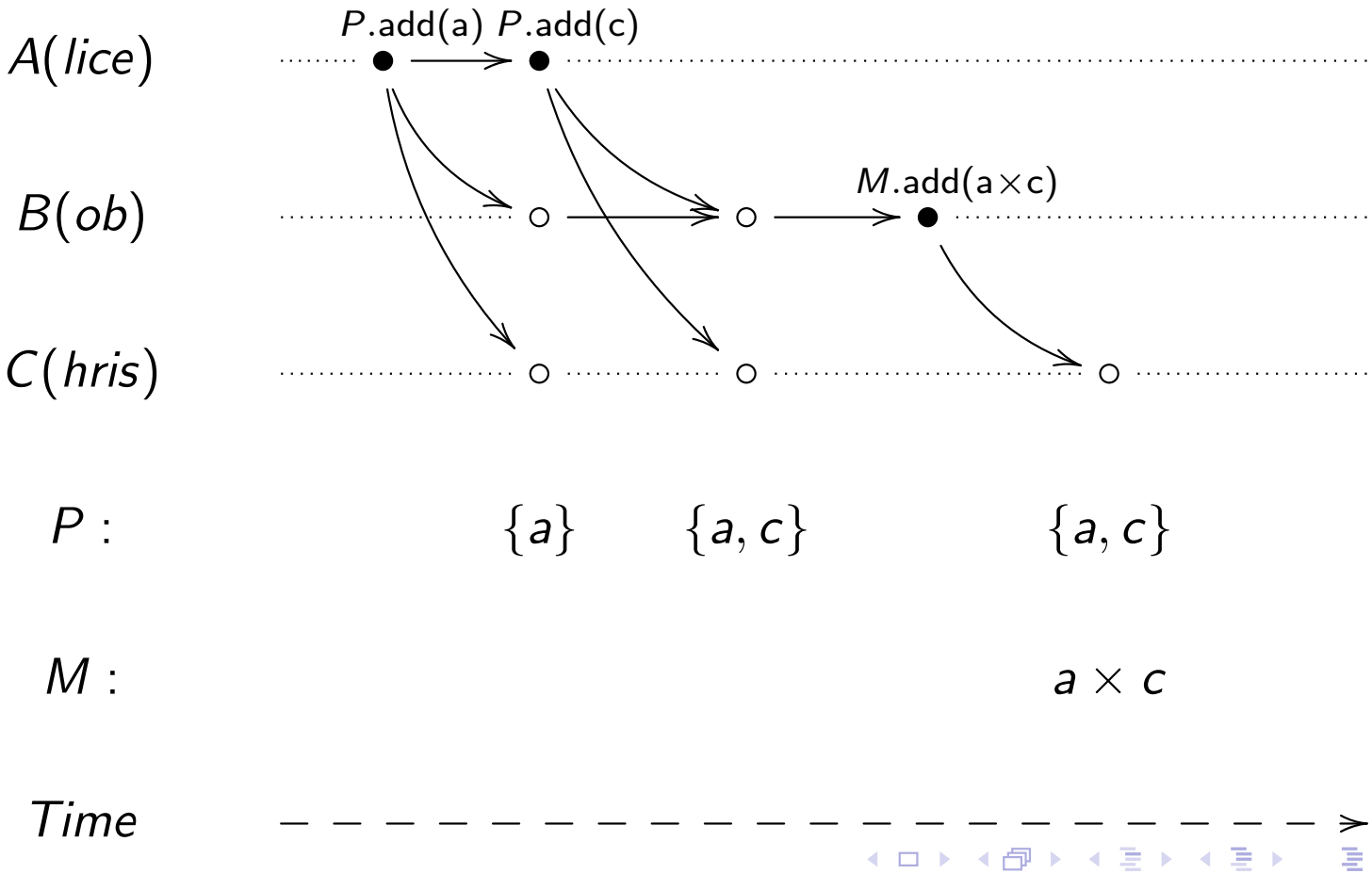
Reliable source FIFO delivery

To ensure FIFO each nodes needs one entry per direct peer



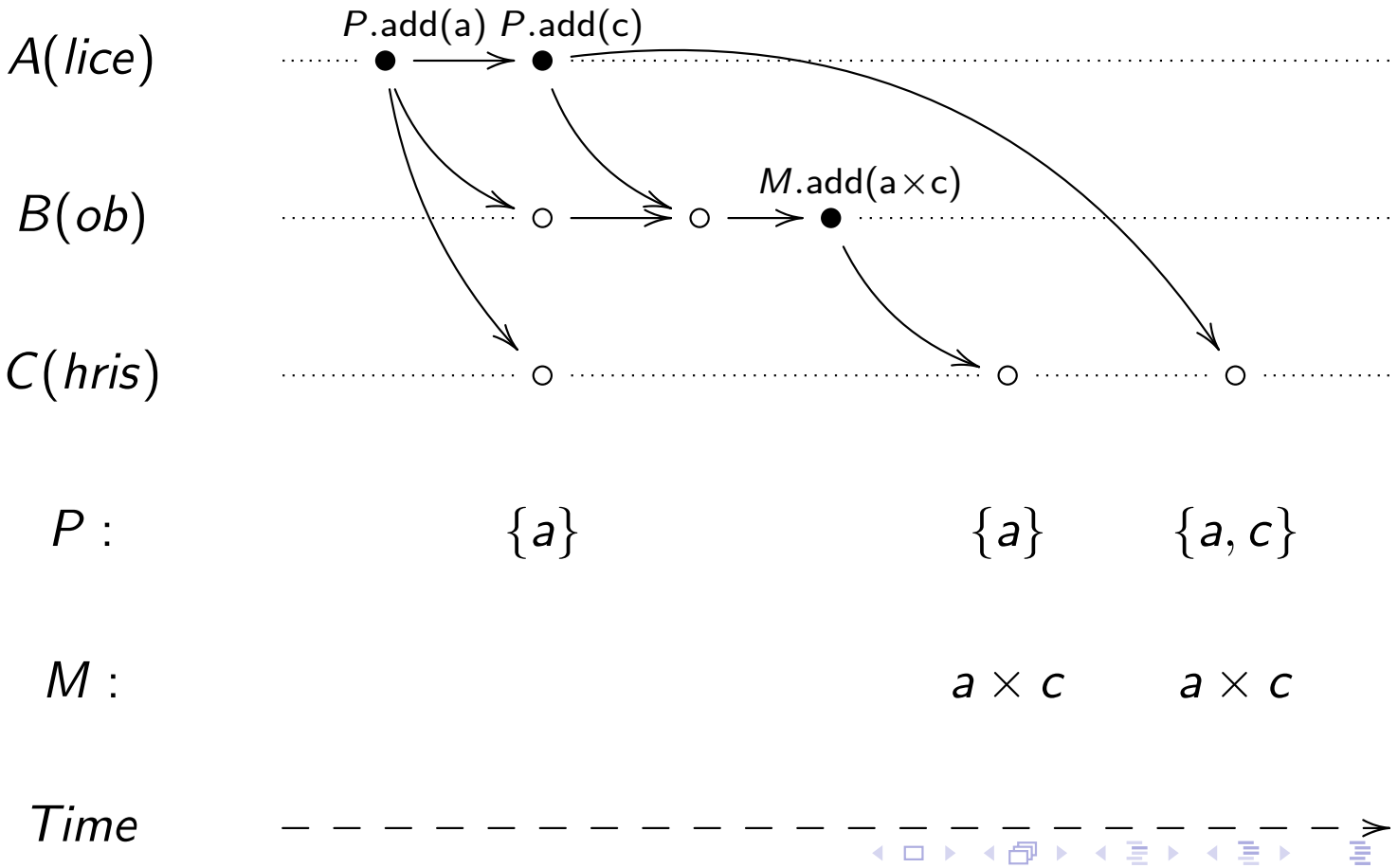
Reliable causal delivery

Key value store, mapping keys (Players and Matches) to sets



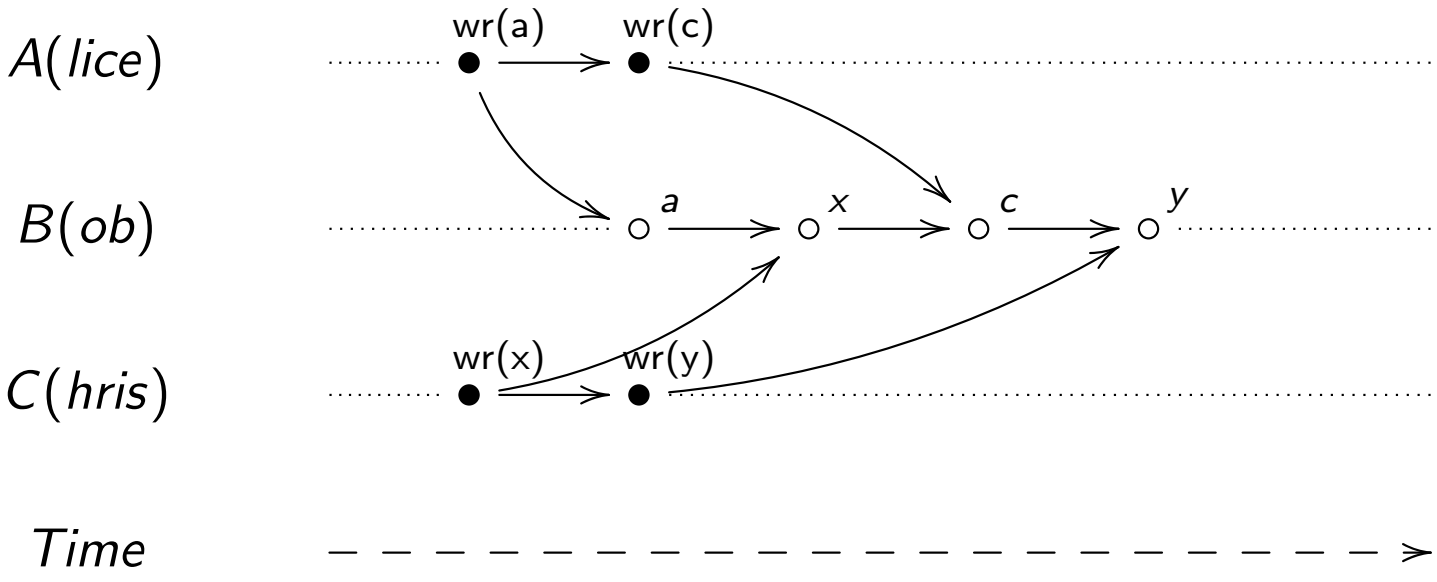
FIFO versus causal delivery

Causal delivery is required for causal consistency
(this execution is FIFO but not Causal)



Causal Consistency

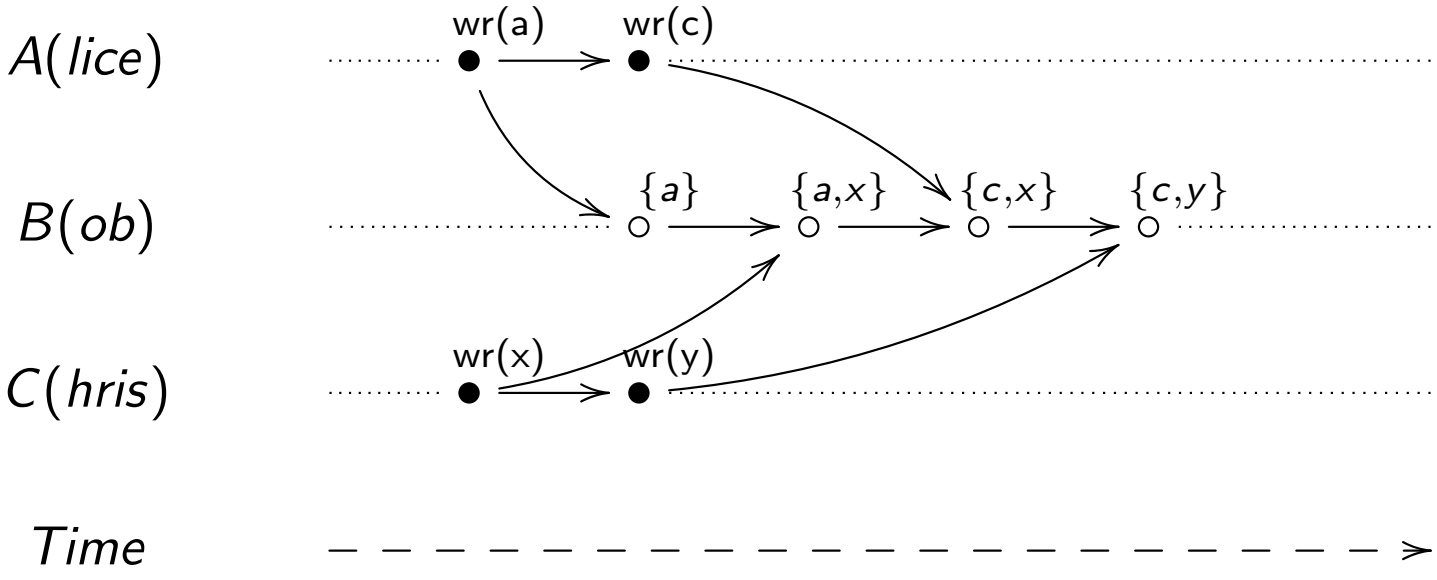
Register with write operations



Tagged causal delivery conveys concurrency information

"Observable" Causal Consistency

Multi-value register with write operations



Tagged causal delivery conveys concurrency information

(C)AP and Observable Causal Consistency

- An eventually consistent data store implementing MVRs cannot satisfy a consistency model strictly stronger than *observable causal consistency* ($\mathcal{OCC}$). $\mathcal{OCC}$ is a model somewhat stronger than causal consistency, which captures executions in which client observations can use causality to infer concurrency of operations. This result holds under certain assumptions about the data store.

in "Limitations of Highly-Available Eventually-Consistent Data Stores" Attiya, Ellen and Morrison. 2015. Evolved from "Consistency, Availability, and Convergence". Mahajan, Alvisi and Dahlin. 2011.

(C)AP and Observable Causal Consistency

- An eventually consistent data store implementing MVRs cannot satisfy a consistency model strictly stronger than *observable causal consistency* ($\mathcal{OCC}$). $\mathcal{OCC}$ is a model somewhat stronger than causal consistency, which captures executions in which client observations can use causality to infer concurrency of operations. This result holds under certain assumptions about the data store.

in "Limitations of Highly-Available Eventually-Consistent Data Stores" Attiya, Ellen and Morrison. 2015. Evolved from "Consistency, Availability, and Convergence". Mahajan, Alvisi and Dahlin. 2011.

CAP Barrier

Stronger consistency guarantees move across the CAP barrier