TOWARDS AFFORDABLE EXTERNALLY CONSISTENT GUARANTEES FOR GEO-REPLICATED SYSTEMS

Manuel Bravo
Luís Rodrigues
Geo-replication
Geo-replication
Geo-replication
Tension between performance and semantics

strong consistency

eventual consistency
Tension between performance and semantics

strong consistency

natural semantics

high latency

e.g., Google Spanner [OSDI’12]
eventual consistency
Tension between performance and semantics

- **strong consistency**
  - natural semantics
    - high latency
      - e.g., Google Spanner [OSDI’12]
  - low latency
    - error-prone
      - e.g., Dynamo [SOSP’07]
- **eventual consistency**
Tension between performance and semantics

- **strong consistency**
  - natural semantics
  - high latency
- **eventual consistency**
  - low latency
  - error-prone
Tension between performance and semantics

strong consistency  

natural semantics  

high latency

low latency  

error-prone

eventual consistency
Tension between performance and semantics

strong consistency  |  eventual consistency

natural semantics  |  low latency

high latency       |  error-prone
Tension between performance and semantics

- **strong consistency**
- **eventual consistency**

linearizability

- **natural semantics**
  - high latency

- **low latency**
  - error-prone
Tension between performance and semantics

strong consistency  eventual consistency

linearizability  sequential

R1  R1  R1
R2  R3  R2
R3  R2

natural semantics  low latency
high latency  error-prone
Tension between performance and semantics

strong consistency  
eventual consistency

linearizability  sequential  causal

natural semantics  low latency  high latency  error-prone
Tension between performance and semantics

- **strong consistency**
  - linearizability
  - sequential

- **eventual consistency**
  - causal + convergence (CRDTs)
  - causal

**natural semantics**

- **low latency**
- **high latency**

**error-prone**
Alternative

mixing semantics, consistent \textbf{when necessary}

\textbf{RedBlue Consistency} \hfill \textbf{Parallel Snapshot Isolation}
\hfill \textbf{[OSDI’12]} \hfill \textbf{[SOSP’11]}

\textbf{Explicit Consistency} \hfill \textbf{Pileus}
\hfill \textbf{[EuroSys’15]} \hfill \textbf{[SOSP’13]}

\textbf{Session guarantees}
Interestingly, most previous solutions do not consider external—or global—guarantees. Interestingly, most previous solutions do not consider external—or global—guarantees. Interestingly, most previous solutions do not consider external—or global—guarantees.

Mixing semantics, consistent when necessary.

Alternative

RedBlue Consistency
[OSDI’12]

Parallel Snapshot Isolation
[SOSP’11]

Explicit Consistency
[EuroSys’15]

Pileus
[SOSP’13]

Session guarantees
[SOSP’97]
External consistency

clients are served with a view of the system **consistent to how an external observer** would witness the succession of events.
Externally consistent guarantees

very **powerful semantics**
Externally consistent guarantees

very powerful semantics

an operation observes all operations that were completed as when the operation began
Externally consistent guarantees

very **powerful semantics**

an operation observes all operations that were completed as when the operation began

**can cope with back-channeling!**
Externally consistent guarantees

very **powerful semantics**

an operation observes all operations that were completed as when the operation began

**can cope with back-channeling!**

but, requires a lot of coordination then, **very expensive**
Use case: an auction service

two roles: auctioneers and buyers

operations (among many): start an auction, place a bid, close an auction, report the winner
Use case: an auction service
Use case: an auction service

start auction!
Use case: an auction service

start auction!
Use case: an auction service

start auction!
Use case: an auction service

100$
Use case: an auction service

start auction!

bid!

100$
Use case: an auction service

start auction!

bid! bid! bid!
Use case: an auction service

start auction!

bid! bid! bid!
Use case: an auction service

start auction!

bid! bid! bid! bid!
Use case: an auction service

start auction!

bid! bid! bid!

bid!

bid!
Use case: an auction service

100$
100$
120$
115$
120$
115$
130$
125$
start auction!

bid! bid! bid!

bid!
bid!
bid!
Use case: an auction service

start auction!

bid! bid! bid!

bid!

bid!

bid!
Use case: an auction service

start auction!

bid! bid! bid!

close auction!
Use case: an auction service

start auction!

bid! bid! bid!

close auction!
Use case: an auction service

start auction!
bid! bid! bid!
close auction!
Use case: an auction service

start auction!

bid! bid! bid!

close auction!

report winner!

100$
115$
120$
125$

130$
100$
120$
115$
125$
130$
115$
130$

100$
120$
130$
Use case: an auction service

start auction!
bid! bid! bid!
bid! bid!
close auction!

report winner!

100$
115$
120$
125$

130$

100$
120$
115$
125$

130$

100$
120$
115$
130$
125$

125$
Use case: an auction service

- start auction!
- bid! bid! bid! bid! bid!
- close auction!
- report winner!
Use case: an auction service

start auction!

bid! bid! bid!

bid! bid!

close auction!

report winner!
Use case: an auction service

start auction!

bid! bid! bid!

bid! bid!

close auction!

report winner!
Use case: an auction service

start auction!
bid! bid! bid!
close auction!

report winner!
How can we provide developers with mechanisms to overcome this type of problems without slowing down the system significantly?
Use case: an auction service

How can we provide externally consistent guarantees on demand?
Our work addresses this inherent tension between performance and meaningful semantics.

We present a new consistency model: external causality.

takes causal consistency (strongest available) and spice it up with external consistency guarantees.
External causality

**internal operations:** read from a causally consistent snapshot

**external operations:** read from a casual snapshot that includes latest updates as of the time when the operation began
Hypothesis

internal operations are **highly predominant** and **cheap** to implement (and we know how to)

external operations—although **expensive**—are **rarely required**
What about concurrency?

External causality allows concurrency, only guaranteeing that operations (both internal and externals) are executed in causal order at each site.

This is not ideal from the semantics point of view, but it has significant benefits in practice.
External causality: example execution

Site a  Site b

Real Time

Internal external causality
External causality: example execution

Site a  Site b

Real Time

internal external causality
External causality: example execution

Site a

Site b

Real Time

a.1

b.1

internal external causality
External causality: example execution

Site a

- a.1
- a.2

Site b

- b.1

Real Time

- internal
- external causality
External causality: example execution

Site a

- a.1
- a.2

Site b

- b.1
- b.2

Real Time

Internal external causality
External causality: example execution

Site a

- a.1
- a.2

Site b

- b.1
- b.2
- b.3

Real Time

Internal causality

External causality
External causality: example execution

Site a

- a.1
- a.2

Site b

- b.1
- b.2
- b.3

Real Time

Internal external causality

Anything committed locally and elsewhere
External causality: example execution

Site a

- a.1
- a.2

Site b

- b.1
- b.2
- b.3

Real Time

Anything committed locally and elsewhere

Internal
External
Causality
External causality: example execution

Site a

Site b

Real Time
External causality: example execution

Site a
- a.1
- a.2
- a.3

Site b
- b.1
- b.2
- b.3

Real Time

internal
external
causality
External causality: example execution

Site a
- a.1
- a.2
- a.3

Site b
- b.1
- b.2
- b.3

Real Time

internal
external causality
External causality: example execution

Site a

- a.1
- a.2
- a.3

Site b

- b.1
- b.2
- b.3

Real Time

Internal causality

External causality
External causality: example execution

Site a
- a.1
- a.2
- a.3

Site b
- b.1
- b.2
- b.3

Real Time

- Internal
- External causality
External causality: example execution

Site a

- a.1
- a.2
- a.3

Site b

- b.1
- b.2
- b.3
- b.4

Real Time

Internal

External causality
External causality: causal serialisations of
External causality: causal serialisations of

Site a

- a.1
- b.1
- a.2
- b.2
- a.3
- b.3
- b.4

Site b

- a.1
- a.2
- a.3
- b.1
- b.2
- b.3
- b.4
External causality: causal serialisations of
Practical issues

Internal operations can read from the local site, without prior communication.

External operations require prior communication—**prepare phase**—to compute the external snapshot from which they are gonna read from.

To ensure external guarantees, the write quorum of both internal and external operations **must overlap** with the quorum used by external operations in its prepare phase.
Let’s recap: an auction service

two roles: auctioneers and buyers

operations (among many): start an auction, place a bid, close an auction, report the winner
Let’s recap: an auction service

two roles: auctioneers and buyers

operations (among many): start an auction, place a bid, close an auction, report the winner
Let’s recap: an auction service
Let’s recap: an auction service

start auction!
Let’s recap: an auction service

start auction!
Let’s recap: an auction service

start auction!
Let’s recap: an auction service

100$

start auction!

bid!
Let’s recap: an auction service

start auction!

bid!
Let’s recap: an auction service

start auction!

bid! bid! bid!
Let’s recap: an auction service

start auction!

bid! bid! bid!
Let’s recap: an auction service

start auction!
bid! bid! bid!

100$
115$
120$

100$
120$
115$

100$
120$
115$
130$
Let’s recap: an auction service

start auction!
bid! bid! bid! bid! bid!
Let’s recap: an auction service

start auction!

bid! bid! bid!

bid!

bid!

bid!

bid!

100$
120$
115$
125$

100$
120$
115$
130$

100$
115$
120$
125$

100$
115$
120$
130$
Let’s recap: an auction service

start auction!

bid! bid! bid! bid!
Let’s recap: an auction service

start auction!
bid! bid! bid!
close auction!
Let’s recap: an auction service

start auction!
bid! bid! bid! bid! bid!
close auction!
Let’s recap: an auction service

start auction!

bid! bid! bid!

close auction!
Let's recap: an auction service

start auction!

bid! bid! bid! bid! bid!

close auction!
Let’s recap: an auction service

start auction! bid! bid! bid! bid! bid!
close auction!

bid!

100$
115$
120$
125$

130$

130$

100$
115$
120$
125$

100$
115$
120$
125$

130$

100$
115$
120$
125$

130$

100$
115$
120$
125$

130$

100$
115$
120$
125$

130$

100$
115$
120$
125$

130$

100$
115$
120$
125$

130$
Let’s recap: an auction service

start auction!

bid! bid! bid!

close auction!
Let's recap: an auction service

start auction!

bid! bid! bid!

bid! bid!

report winner!

100$
120$
130$
115$
125$
130$

start auction!

bid! bid! bid!

bid! bid!

report winner!

100$
120$
130$
115$
125$
130$

Let’s recap: an auction service

start auction!

bid! bid! bid! bid! bid!

close auction!

report winner!
Let’s recap: an auction service

start auction!
bid! bid! bid!
close auction!

report winner!
Let's recap: an auction service

- Start auction!
  - Bid! Bid! Bid!
- Close auction!
Let’s recap: an auction service

100$
115$
120$
125$
130$

start auction!

bid! bid! bid!
bid! bid!

close auction!

report winner!
Let's recap: an auction service

start auction!

bid! bid! bid!

bid! bid!

close auction!

report winner!

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$

100$
115$
120$
125$

X

130$
Other use cases: a cache
Other use cases: a cache

When users access the website, this is loaded from the cache.
Other use cases: a cache

When users access the **website**, this is **loaded from the cache**

This **cache** could be **updated** periodically, by means of an **internal operation**
Other use cases: a cache

When users access the website, this is loaded from the cache.

This cache could be updated periodically, by means of an internal operation.

There could be an alternative option to refresh the cache with the latest information by means of an external operation.
Other use cases: a cache

When users access the website, this is loaded from the cache.

This cache could be updated periodically, by means of an internal operation.

There could be an alternative option to refresh the cache with the latest information by means of an external operation.
Other use cases: a cache

When users access the website, this is loaded from the cache.

This cache could be updated periodically, by means of an internal operation.

There could be an alternative option to refresh the cache with the latest information by means of an external operation.
Other use cases: a cache

When users access the website, this is loaded from the cache.

This cache could be updated periodically, by means of an internal operation.

There could be an alternative option to refresh the cache with the latest information by means of an external operation.

Useful to read time-sensitive information.
Other use cases: a cache
Other use cases: a cache

You should check what have just happen Lisbon!
Other use cases: a cache

You should check what have just happen Lisbon!

Go to publico.pt!
Other use cases: a cache

- You should check what have just happen Lisbon!
- Go to publico.pt!
- Opening Safari!
Other use cases: a cache
Other use cases: a cache
Other use cases: a cache
Other use cases: a cache

Awesome news in Lisbon!

Useful to cope, at least partially, with back-channelling
A word on fault tolerance

to ensure external guarantees, the write quorum of both internal and external operations must overlap with the quorum use by external operations in its prepare phase
A word on fault tolerance

to ensure external guarantees, the write quorum of both internal and external operations must overlap with the quorum use by external operations in its prepare phase

we plan to only ack write operations when these have been installed in two sites, in order to tolerate the failure of one site
A word on fault tolerance

to ensure external guarantees, the write quorum of both internal and external operations must overlap with the quorum use by external operations in its prepare phase.

we plan to only ack write operations when these have been installed in two sites, in order to tolerate the failure of one site.

thus, the prepare phase of external operations does not require contacting all sites.
A word on fault tolerance

**bad things:** we are making writes slightly more expensive
A word on fault tolerance

bad things: we are making writes slightly more expensive

good things: reads of internal operations still require a single site
A word on fault tolerance

**bad things:** we are making writes slightly more expensive

**good things:** reads of internal operations still require a single site

**more good things:** we enhance availability when compared to causal consistency, which otherwise is **sticky available**
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]

causal
CBCAST

forced
ABCAST

immediate
GBCAST
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]

RedBlue consistency [OSDI’ 12]
Related work

closest to our work:
lazy replication [TOCS’ 92] and the ISIS work [TOCS’ 87]

RedBlue consistency [OSDI’ 12]
Related work

immediate
GBCAST
Related work

Performed at all replicas in the same order relative to all operations
Related work

Performed at all replicas in the same order relative to all operations

Ordered consistently with external events
Related work

- **immediate GBCAST**

  Performed at all replicas in the same order relative to all operations

  Ordered consistently with external events

  The operation becomes all-present after completion
Related work

Performed at all replicas in the same order relative to all operations

Ordered consistently with external events

The operation becomes all-present after completion
Related work

Performed at all replicas in the same order relative to all operations

Ordered consistently with external events

The operation becomes all-present after completion
Related work

Performed at all replicas in the same order relative to all operations

Ordered consistently with external events

The operation becomes all present after completion
Related work

- Performed at all replicas in the same order relative to all operations
- Ordered consistently with external events
- The operation becomes all present after completion

**external**
Related work

- Performed at all replicas in causal order
- Ordered consistently with external events
- The operation becomes all-present after completion
Related work

Performed at all replicas in causal order

Ordered consistently with external events

The operation becomes all present after completion
Open questions

shall we add a new operation that becomes \textbf{all-present after completion} but it is not order consistently with external events?
Open questions

shall we add a new operation that becomes **all-present after completion** but it is not order consistently with external events?
Open questions

shall we add a new operation that becomes all-present after completion but it is not order consistently with external events?
Open questions

shall we add a new operation that becomes all-present after completion but it is not order consistently with external events?

Performed at all replicas in causal order

Ordered consistently with external events
Open questions

shall we add a new operation that becomes *all-present after completion* but it is not order consistently with external events?

Performed at all replicas in causal order

Ordered consistently with external events
Open questions

shall we add a new operation that becomes all-present after completion but it is not order consistently with external events?

Performed at all replicas in causal order

Ordered consistently with external events

The operation becomes all-present after completion
Open questions

shall we add a new operation that becomes **all-present after completion** but it is not order consistently with external events?

Performed at all replicas in causal order

Ordered consistently with external events

The operation becomes all-present after completion

this could further help to **beat back-channeling**
Open questions

shall we **add** mechanisms to ensure **convergence** (CRDTs) and to preserve invariants (forced/ **red operations**) in order to build a complete system?
Open questions

shall we add mechanisms to ensure convergence (CRDTs) and to preserve invariants (forced/red operations) in order to build a complete system?
Open questions

shall we add mechanisms to ensure convergence (CRDTs) and to preserve invariants (forced/red operations) in order to build a complete system?
Open questions

shall we **add** mechanisms to ensure **convergence** (CRDTs) and to preserve invariants (forced/ **red operations**) in order to build a complete system?
Open questions

shall we **add** mechanisms to ensure **convergence** (CRDTs) and to preserve invariants (forced/ **red operations**) in order to build a complete system?
Open questions

shall we **add** mechanisms to ensure **convergence** (CRDTs) and to preserve invariants (forced/ **red operations** ) in order to build a complete system?
Open questions

how can we **efficiently implement** external causality in a geo-replicated, distributed system?
we present a new consistency model: **external causality**

addresses the inherent tension between **performance and meaningful semantics**

combines causal consistency (strongest available) and external consistency (strongest semantically)