Aggregation $\neq$ Replication

Carlos Baquero
Universidade do Minho & INESC TEC

Dagstuhl Seminar 19442, October 2019

Dagstuhl Seminar 19442

The seminar aims to focus on answering the following major questions in addition to those raised by participants:

- Which abstractions are required in emergent fields of distributed systems, such as mixed cloud/edge computing and IoT?
- How can language abstractions be designed in a way that they provide a high-level interface to programmers and still allow fine-grained tuning of low-level properties when needed, possibly in a gradual way?
- Which compilation pipeline (e.g., which intermediate representation) is needed to address the (e.g., optimization) issues of distributed systems?
- Which research issues must be solved to provide tools (e.g., debuggers, profilers) that are needed to support languages that target distributed systems?
- Which security and privacy issues come up in the context of programming languages for distributed systems and how can they be addressed?
- What benchmarks can be defined to compare language implementations for distributed systems?

Internet of things (and users)

Carlos Baquero
Universidade do Minho & INESC TEC

Context / System model

Constraints

- Geo-distribution: Availability Zones, Edge, Things & Users
- Asynchrony, Independent failure, Crashes (and possibly recovery)

Aspirations

- Scalability in numbers and distances
- Partition tolerance, local availability and autonomy

Toolbox

Replication and Aggregation

Replication for high availability and low latency

Shared state among replicas. Local uncoordinated updates. Integration of received remote updates. Distributed logs of operations. Causal consistency. Conflict free replicated data types.

Distributed data aggregation and summarization

Local sources of data: storage space, load, temperature, ... Source location and time (points or streams). Global aggregates: counts, maximum, average, top-k, CDF. Global status and prediction.

Example Quiz: Temperature control

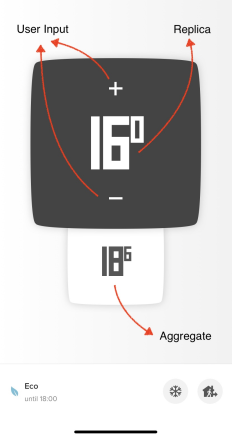
Carlos Baquero
Universidade do Minho & INESC TEC

Can you find: (1) replica state, (2) user input, (3) data aggregate?

Example Quiz: Temperature control

Carlos Baquero Universidade do Minho & INESC TEC

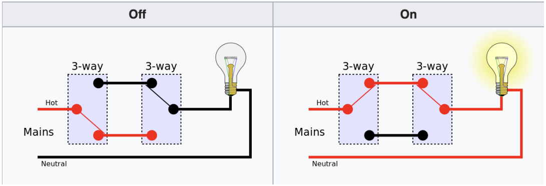
Can you find: (1) replica state, (2) user input, (3) data aggregate?



Replication

LightKone blog post: *Aggregation is not Replication*

> *"... in replication there is an abstraction of a single replicated state that can be updated in the multiple locations where a replica is present."*



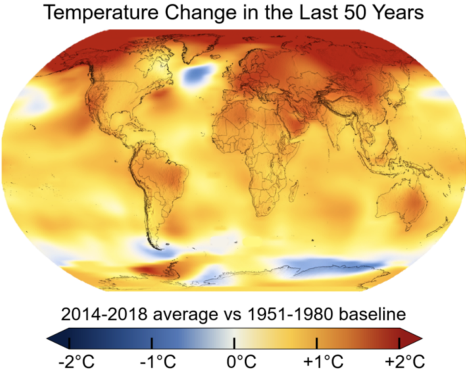
(Creative Commons: `https://en.wikipedia.org/wiki/Multiway_switching`)

Aggregation

Carlos Baquero Universidade do Minho & INESC TEC

LightKone blog post: *Aggregation is not Replication*

> *"...data to be aggregated is often not directly controlled by users, it usually results from an external physical process or the result of complex system evolutions."*



(Public Domain: https://en.wikipedia.org/wiki/Global_warming)

Replication

Conflict Free Replicated Data Types

Operation Based (reliable causal delivery - exactly once)

- **Classic**: Operations are translated into effects, broadcast effects
  Delivery log is sequential and causal consistent
- **Pure**: Operations are broadcast as is, adapted on delivery
  Delivery log holds causal partial order

State Based (commutative, associative, idempotent)

- **Classic**: Operations mutate local state, ships full state
- δ **State**: Operations are translated into state deltas, ships deltas

Only some (sequential) data types have truly commutative operations

Commutative: $f(g(x)) = g(f(x))$

- $\text{inc}(\text{dec}(x)) = \text{dec}(\text{inc}(x))$
- $\text{add}(a, \text{add}(b, x)) = \text{add}(b, \text{add}(a, x))$
- Operations can be shipped as is

Non commutative: $f(g(x)) \neq g(f(x))$

- $\text{add}(v, \text{rmv}(v, x)) \neq \text{rmv}(v, \text{add}(v, x))$
    - **Classic**: Translate to embed rmv $\rightarrow$ add or add $\rightarrow$ rmv
    - **Pure**: Ship as is. Use order info in delivery

CRDTs
State Based

Carlos Baquero
Universidade do Minho & INESC TEC

State evolution defined by a semi-lattice. Join $\sqcup$ and partial order $\sqsubseteq$

State - operations induce state mutations

$$X \sqsubseteq m(X)$$

E.g. add_a over $\{b, e\}$ mutates it into $\{a, b, e\}$

δ State - single out the mutation

$$m(X) \equiv X \sqcup m^\delta(X)$$

E.g. add_a^δ over $\{b, e\}$ derives $\{a\}$

Do all commutative types lead to trivial state translations?
No, and the cause is not ordering but derives from idempotency

CRDTs

δ state based grow-only set

$$
\begin{aligned}
\mathsf{GSet}\langle E\rangle &= \mathcal{P}(E) \\
\bot &= \{\} \\
\mathsf{insert}^{\delta}_{i}(e,s) &= \{e\} \\
\mathsf{elements}(s) &= s \\
s \sqcup s' &= s \cup s'
\end{aligned}
$$

Sets are "naturally" idempotent

CRDTs
δ state based grow-only counter

Carlos Baquero
Universidade do Minho & INESC TEC

$$
\begin{aligned}
\mathrm{GCounter} &= \mathbb{I} \hookrightarrow \mathbb{N} \\
\bot &= \{\} \\
\mathrm{inc}_i^{\delta}(m) &= \{i \mapsto m(i) + 1\} \\
\mathrm{value}(m) &= \sum_{j \in \mathbb{I}} m(j) \\
m \sqcup m' &= \{j \mapsto \max(m(j), m'(j)) \mid j \in \mathrm{dom}\, m \cup \mathrm{dom}\, m'\}
\end{aligned}
$$

Counter state must be made idempotent

Replication

Recap

Support for high availability leads to analysis of data type operations and their possible adaptations for adequate dissemination under the chosen network system model

Adaptation for dissemination will also occur in Aggregation

Aggregation

Carlos Baquero
Universidade do Minho & INESC TEC

Aggregation can be simply defined as:

“the ability to summarize information”

(Renesse, Birman, Vogels. Astrolabe. ACM TOCS, 2013)

Aggregation
Problem Definition

Carlos Baquero
Universidade do Minho & INESC TEC

An aggregation function f takes a multiset of elements from a domain I and produces an output of a domain O.

$$f : \mathbb{N}^I \to O$$

Resulting that:

- Order is not relevant
- Elements can occur multiple times

E.g. multiset $M = \{10, 32, 10, 7\}$, aggregation func: $\max(M) = 32$

Aggregation
Typical functions

Carlos Baquero
Universidade do Minho & INESC TEC

- sum
- count
- max or min
- average
- mode

Aggregation

Decomposable aggregation functions

Self-decomposable function f

For some merge operator $\oplus$ and all non-empty multisets X and Y:

$$f(X \uplus Y) = f(X) \oplus f(Y)$$

E.g.

$\text{count}(\{x\}) = 1$
$\text{count}(\{X \uplus Y\}) = \text{count}(X) + \text{count}(Y)$

Some functions are not self-decomposable but can be transformed

Decomposable function f

For some function g and a self-decomposable aggregation function h, it can be expressed as:

$$f = g \circ h$$

E.g.

$$\begin{aligned}
\text{average}(X) &= g(h(X)) \\
h(\{x\}) &= (x, 1) \\
h(X \uplus Y) &= h(X) + h(Y) \\
g((s, c)) &= s/c
\end{aligned}$$

Aggregation
Duplicates / Idempotency

Carlos Baquero
Universidade do Minho & INESC TEC

Duplicate insensitive

Result only depends on *support set* of the multiset
E.g.

$$\min(\{1,3,1,2,4,5,4,5\}) = \min(\{1,3,2,4,5\}) = 1$$

Duplicate sensitive

Multiplicity is relevant to result
E.g.

$$8 = \text{count}(\{1,3,1,2,4,5,4,5\}) \neq \text{count}(\{1,3,2,4,5\}) = 5$$

Aggregation
Adding idempotency

Carlos Baquero Universidade do Minho & INESC TEC

Functions like count and sum can be made duplicate insensitive by a stochastic transformations

E.g.

Extrema propagation derives a vector of k exponential random variables that can be aggregated by max and used to approximate the aggregated sum. Thus relaxing (exactly-once) network assumptions.

Aggregate sums by Extrema Propagation

- Node i holds value v_i
- Initially $V_i = \text{rexp}(k, v_i)$
- On gossip message V_m from anther node do $V_i = \max(V_i, V_m)$
- Aggregate sum estimation value is $\frac{k-1}{\sum V_i}$

Further reference

- Search for blog post "Aggregation is not Replication"
- Nuno Preguiça, Carlos Baquero, Marc Shapiro: Conflict-Free Replicated Data Types CRDTs. Encyclopedia of Big Data Technologies 2019
- Paulo Jesus, Carlos Baquero, Paulo Sérgio Almeida: A Survey of Distributed Data Aggregation Algorithms. IEEE Communications Surveys and Tutorials 17(1): 381-404 (2015)