

TOWARDS A CONSISTENCY-AWARE DATA PLACEMENT MECHANISM

RainbowFS Final Workshop

2022-03-28

CREDITS

- Work done during Etienne Mauffret's Ph.D. thesis

Univ. Savoie Mont Blanc – RainbowFS

- Currently ATER at ENS Lyon
- Ph.D. defense planned in a couple of months

DATA REPLICATION IN LARGE-SCALE DISTRIBUTED SYSTEMS (I)

- Why
 - Fault tolerance
 - Multiple copies for each piece of data
 - => Enhance durability and availability
 - Performance
 - Copies close to end users => low latency
 - Local copy (local accesses / “off-line” operations)
 - Many copies => load balance (e.g. Content Delivery Networks)

DATA REPLICATION IN LARGE-SCALE DISTRIBUTED SYSTEMS (2)

- Issues / questions to answer
 - How to add redundancy ?
 - Erasure coding
 - Replication
 - How many copies ?
 - How to handle “healing” (when/how maintenance/optimization should be performed) ?
 - Which consistency model among copies ?
 - **Where copies should be placed ?**

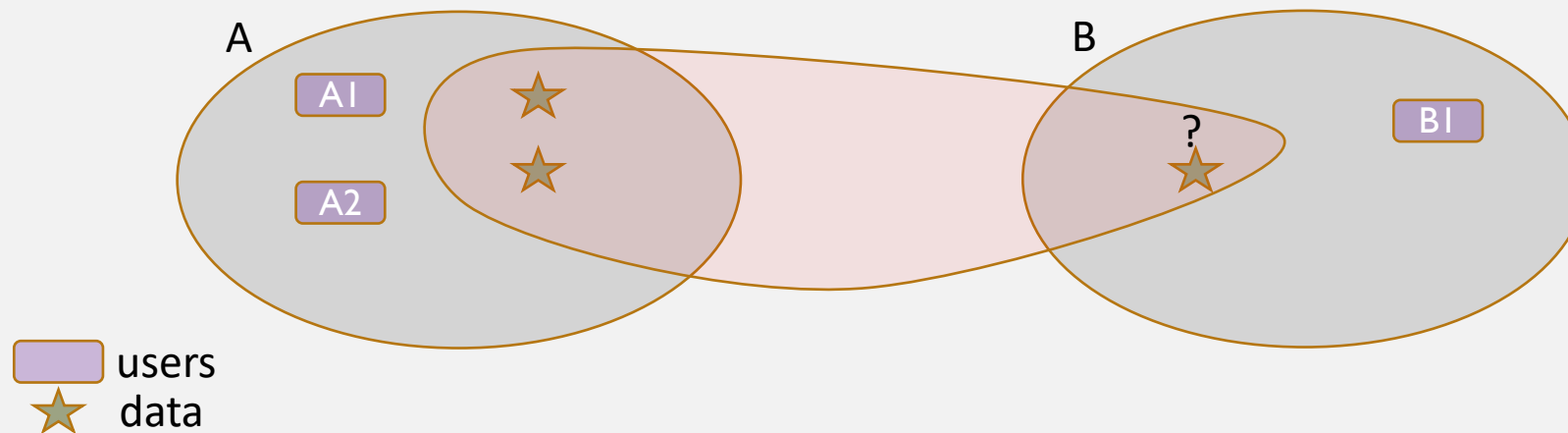
DATA PLACEMENT IS KEY

- Data placement impacts
 - Data access performance (data availability)
 - Close copies (distance users/data)
 - Synchronization cost among copies
 - Healing process performance => resilience
 - RelaxDHT

DATA PLACEMENT IS HARD

- Many (dynamic) criteria to take into consideration
 - Fault tolerance (correlated failures, healing process performance)
 - Data access (a popular video / a backup)
 - Localization
 - Kind (reads/writes)
 - Frequency
 - **Consistency protocol**
- NP-hard problem

DATA PLACEMENT SHOULD TAKE CONSISTENCY IN CONSIDERATION



CONSISTENCY AWARE DYNAMIC DATA REPLICATION CONTEXT AND GOAL

- Context: tradeoff performance vs consistency (CAP theorem)
 - Good performance => adapted data consistency
 - Just-right consistency : synchronize only if necessary

=> Data management systems already offer multiple consistency “levels”
- Goal: data replication should take into consideration
 - Access patterns
 - Location / type / frequency
 - Consistency protocol (strong or relaxed consistency, kind of synchronization, ...)

CONSISTENCY AWARE DYNAMIC DATA REPLICATION APPROACH (I)

- The developer sets weights and priorities
 - Are synchronizations more critical than user accesses (consistency protocol into consideration) ?
 - Are some users more important than others ?
- Maintain metrics for each piece of data (monitoring)
 - Access statistics (read / write frequencies for each user location)
 - Locally (on each node storing a copy)

CONSISTENCY AWARE DYNAMIC DATA REPLICATION APPROACH (2)

- Periodically compute a new replica-set for each data
 - On a per-data basis
 - Analytically compute a new replica-set taking into consideration
 - The static weights - **the consistency protocol**
 - The “recent” (dynamic) access statistics – **the access patterns**
- => computationally intensive (test all possibilities)

CANDORSIM – A DATA PLACEMENT TOOL

- A discrete event simulator based on PeerSim
 - Simulates storage nodes/user nodes, data accesses, synchronization
 - Nodes monitor data accesses using our approach
 - Periodically, nodes
 - compute the right placement taking into account
 - Weights
 - Data access statistics
 - Moves pieces of data to adapt placement if necessary

PERSPECTIVES

- Short term enhancement
 - Reduce the problem size (topology-aware group of nodes)
 - Adapt “when necessary” (vs periodically)
- Dynamic adaptation of the replication factor
 - Create/remove copies according to access patterns/consistency needs

=> from consistency-aware data placement to consistency-aware data replication...
- Auto-tune => use traces and the simulator to learn which weights for which consistency protocol
- SkyData => toward intelligent and autonomous data !