Simba: Tunable End-to-End Data Consistency for Mobile Apps

Dorian Perkins†, Nitin Agrawal†, Akshat Aranya†, Curtis Yu*, Younghwan Go*, Harsha V. Madhyastha†, and Cristian Ungureanu†
NEC Labs America†, UC Riverside*, KAIST*, U Michigan†

Motivation
- Wide-spread use of data-centric mobile apps
- Data consistency is a primary requirement
- App devs tasked with ensuring consistency of user’s data
- Current solutions are inflexible
- Rolling own service is difficult
  - Failures, Conflicts, Connectivity, Consistency, etc.

Consistency Study
We studied 23 popular mobile apps and found that half of them exhibit undesirable behavior!

Simba Design
- Key Features
  - Simple, high-level programming abstractions
  - Transparent handling of data sync & failures
  - Atomicity across tabular & object data
  - End-to-end tunable consistency
  - Scalable architecture

Simba Table
- Logical Abstraction
  - Name, Quality, Photo, Thumbnail
- Physical Layout
  - Tabular, Object

Data Sync Abstraction
- Unified tabular + object rows
- Row-level atomicity
- Per-table consistency scheme
  - Strong, Causal, or Eventual
- Offline support
- Conflict detection/resolution

Performance
Efficient Syncing via Change-sets
Change cache enables sync of only modified row data (e.g., update of a single 64 KiB chunk in a 1 MiB object)

Scalability
16 Gateways and 16 Stores, 500 ops/sec
Simba Cloud scales well with increasing tables and clients

Simba Source Code: https://github.com/SimbaService/Simba
Project Homepage: http://tinyurl.com/SimbaService

Study Findings
- Diverse consistency requirements
- Sync semantics often oblivious to consistency
- Limited offline support
- Inadequate error propagation
- Atomicity violations of inter-dependent data

European Conference on Computer Systems (EuroSys)
April 21–24, 2015, Bordeaux, France