**Existential Consistency:**
Measuring and Understanding Consistency at Facebook

Haoran Lu*^, Kaushik Veeraraghavan^, Philippe Ajoux^, Jim Hunt^, Yee June Song^, Wendy Tobagus^, Sanjeev Kumar^, Wyatt Lloyd *^  
*University of Southern California, ^Facebook

**Problem**
- Benefits of stronger consistency unclear because the anomalies they prevent were not quantified
  - Anomalies are strange behavior that defies user expectations
- No prior work for a large production Web service

**Motivation**
- Helps answer the question whether adopted consistency model is “sufficient”
- Helps with system debugging
- Better understands the tradeoff between stronger consistency and higher performance

**Methods**
- Measure TAO, the write-through cache for the social graph
  - Eventually consistent
  - Single-master
- Two types of analysis
  - Principled: a log based offline checker; checks against linearizability, per-object sequential, and read-after-write
  - Practical: a real-time monitoring system; checks how well updates converge at replicas

**Principled Analysis**
- Log requests from web servers
  - Sample based on objects (rate: 1/1M objects)
  - Spreads over 12 days (8/20/2015-8/31/2015)
  - 2.76 B requests
  - 17 M data objects
- Trace pre-processing
  - 75% of objects only have reads in the trace
  - 20% of objects only have writes in the trace
  - Neither can exhibit anomalies
  - Remaining 5% of objects and 24% of requests

**Takeaways**
- First study of consistency in large-scale production system
- Facebook TAO is highly consistent
  - greater than 99.99%
- Stronger consistency has benefits
- Future research should include transactions to maximize benefits
- Φ-consistency is a good metric for real-time consistency monitoring

**Practical Analysis**
- Offline Checker (linearizability)
  - A directed graph captures state transition of the system
  - Vertices: write requests; Edges: real-time order
  - Merge: pair a read with write to capture the transition order seen by the user
  - Cycle after merge -> user seen order != state transition -> anomaly

**Results (anomaly rate)**
- Linearizability: 5 per million
- Per-Object Sequential: 1 per million
- Read-After-Write
  - Global: 4 per million
  - Per-region: 2 per million
  - Per-cluster: 1 per million

**Practical Analysis**
- Φ(P)-consistency
  - Measures the convergence of replica set P
  - Φ(P) = 100% if all replicas always return same results
- Real-time monitor
  - Simultaneously reads at each replica in P
  - Monitor Φ(P)-inconsistency
  - when responses of replicas are not identical

**Results**
- R_i : G -> region to global
- Spike -> increase of inconsistency

---

**Diagram:**
- Graph illustrating linearizability, per-object sequential, and read-after-write.
- bridges between web servers and data objects showing real-time order and state transition.

---

**Figure:**
- Graph showing Φ(Φ(Φ(P)-inconsistency)) over time.
- Comparing failure and non-failure regions.