Active-Active for Oracle using GoldenGate

Janardh Bantupalli

About LinkedIn

LinkedIn is the world's largest professional network with more than 400 million members in over 200 countries and territories around the globe.

Mission

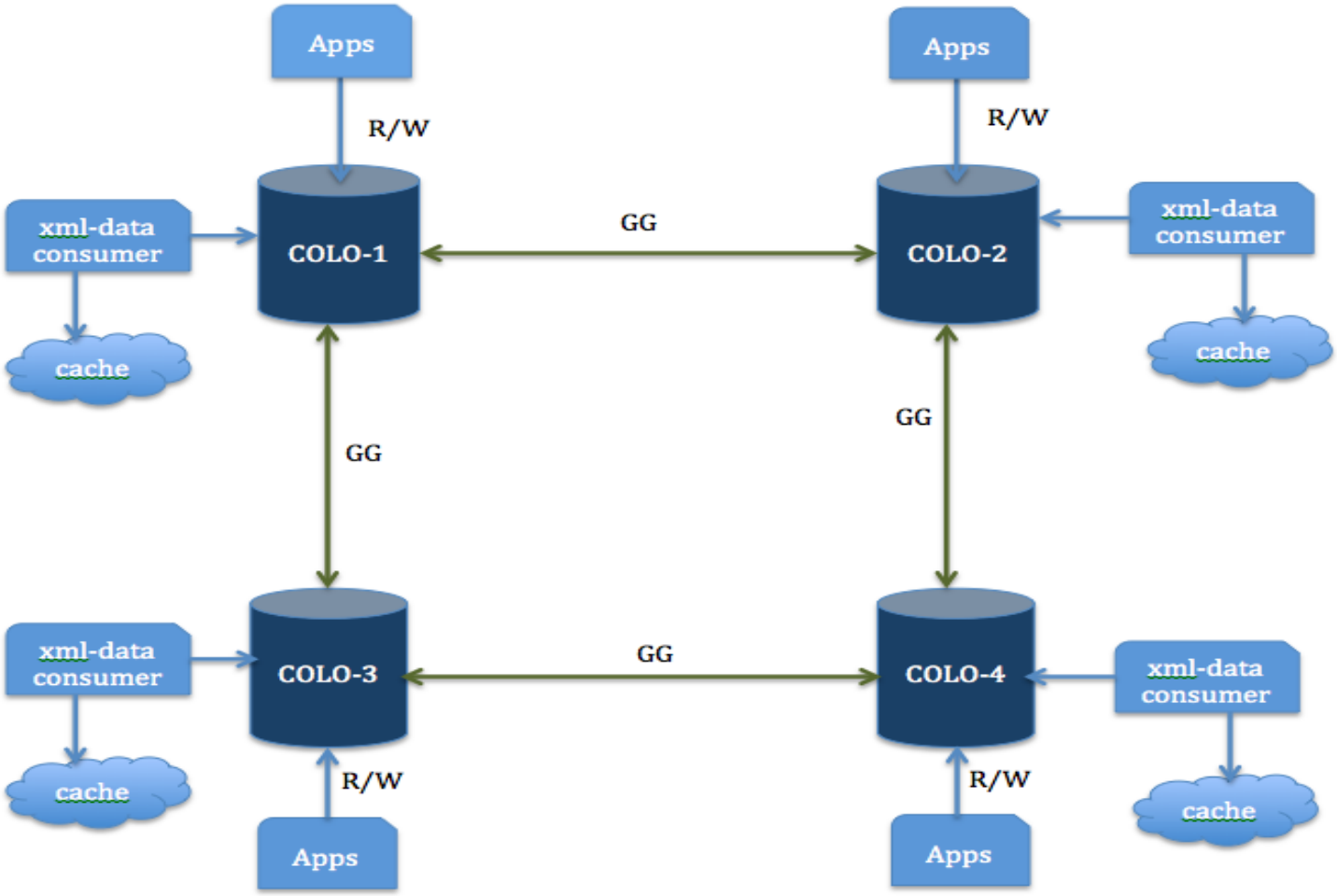
LinkedIn's mission is simple: connect the world's professionals to make them more productive and successful. When you join LinkedIn, you get access to people, jobs, news, updates, and insights that help you be great at what you do.

Active-Active: Design

- Two columns: gg_modi_ts, gg_status; gg_priority in some
  - Before triggers to populate values, except for gg_priority
  - Design isolation from application-db design

- Conflict resolution through gg_modi_ts, gg_priority
  - last writer wins (default)
  - priority resolution

Active-Active with GoldenGate (GG) @ LinkedIn



Oracle + Golden Gate: Active-Active

Configuration

- Active-Active with DDL Replication
- Full-row capture on source
- Conflict resolution through gg_modi_ts (timestamp), gg_priority
  - Last Writer Wins (LWW), by default
  - Priority resolution (beware of corner cases!)
- Data Encryption

Oracle + Golden Gate: Multi-Colo

Scaling

- Non-overlapping primary keys
- Unique (non-PK) column collision, across colos
- Parallelism for scaling, for DBs with high DML; multiple-thread-apply => eventual consistency
- Data Compression (for high cross-colo latencies)
- GG process off-loading, cascading

Multi(2+)-Way: Issues/Resolutions

- LWW & Update-to-Insert (updaterowmissing-overwrite)
- Hard Delete to Soft Delete Conversion
  - Row re-birth prevention
- Foreign Key Constraints
- No-validate constraints

Soft Deletes

- Facilitate row re-birth prevention
- New column deleted_ts and index (for incremental retrieval)
  - Data created by application
- Trigger at the source:
  - Identify new soft deletes and bump up gg_modi_ts by 30 days
  - Soft delete will win on all sites

LWW and Update-to-Insert Conversion

ela4	lva1	ltx1
:t1		
R1 Insert		
	:t2	
	R1 Insert (from ela4)	
	(GG Apply; no further replication of this)	
ela4-to-ltx1 replication delayed		
	:t3	
	R1 Update	
		:t4
		R1 Update, from lva1)
		(Update-to-insert**)**
		Can't skip update; latest version
ela4-to-ltx1 replication caught up		**:t5**
		R1 Insert (from ela4)
		LWW-Insert

FK Issue: Update-to-Insert Conversion

ela4	lva1	ltx1
:t1		
Rp (Parent) Insert		
Rc (Child) Insert		
	:t2	
	Rp Insert (from ela4)	
	Rc Insert (from ela4)	
ela4-to-ltx1 replication delayed/down	(GG Apply; no further replication of these)	
	:t3	
	Rc Update	
		:t4
		Rc Update, from lva1 (Update-to-insert)
		<<FK-Violation: No Rp (parent row)>>
		Replication blocked until Rp arrives

No-Validate Constraints

Table t1 has columns c1 to c3; c3 is currently NULL-able

ela4	lva1
Row r1: c1=1, c2=10, c3=NULL	
:t1	
NOT NULL novalidate constraint added on column c3; replicated to all colos (nulls not allowed, from this point)	
:t2	
Update t1 set c2=15 where c1=1 GG captures c1=1, c2=15, c3=NULL	
	:t4
	R1 applied (c1=1, c2=15, c3=NULL) **Apply fails because of NULL value for c3**

Solution:

Detect NULLs and pre-populate a default value (or) disable constraint

Multi (2+)-Way: Other Issues/Resolutions

- Multiple Unique Keys (More than 1, including PK)
- Clock-skew triggers (issue not specific to 3-way)
  - make gg-modified-ts to be always greater than the existing value
- Deadlock Mitigation
  - Optimal grouptransops value
- Design should work for any number of colos

Questions?