Active / Active Configurations with Oracle Active Data Guard

Aris Prassinos
Distinguished Member of Technical Staff
MorphoTrak, SAFRAN Group
MorphoTrak
SAFRAN Group

- US subsidiary of Sagem Sécurité, SAFRAN Group
- Leading innovators in multi-modal Biometric Identification and Verification
  - Fingerprint, palmprint, iris, facial
- Government and Commercial customers
  - Law enforcement, border management, civil identification
  - Secure travel documents, e-passports, drivers’ licenses, smart cards
  - Facility / IT access control
- Chosen as Biometric Provider for FBI Next Generation Identification Program
  http://www.sagem-securite.com/eng/site.php?spage=04010847
Printrak BIS

- Printrak Biometrics Identification Solution
- Over 100 turnkey production installations worldwide
- Java-based application using Service Oriented Architecture
- Oracle Database 11g
  - Active Data Guard, RAC, XML DB, SecureFiles, ASM
Printrak BIS Database

- Homegrown repository
  - Biometrics and scanned documents stored as LOBs (OOW 2008 S298756)
  - Descriptive data stored as XML (OOW 2009 S311519)

- Homegrown workflow manager

- JMS backing store

- Auditing logs

- Read intensive mixed OLTP workload
Disaster Recovery objectives

- Goal is to minimize overall system cost of a Disaster Recovery architecture by achieving maximum utilization of the DR site
  - Cost includes: hardware, licensing, development, maintenance, support

- Constraints
  - WAN with up to 10ms latency between Primary and DR datacenters
  - Clients experience similar latency connecting to either datacenter
  - Well defined throughput and response time requirements
  - Strong data consistency required
  - Data cannot be logically partitioned to allow update-anywhere without conflicts
  - Minimal data loss RPO
  - RTO measured in minutes
DR architecture

- Oracle Active Data Guard in Maximum Availability (SYNC) mode
- Routing all application Writes to Primary
  Load balancing application Reads to both Primary and Standby
- Hardware traffic managers allow clients to transparently connect to either datacenter
- Relying on application server multi-pool capabilities for client failover
  (e.g. JBoss HA Datasources / Weblogic Multi Data Sources)
- Using FSFO with Observer on a third site to avoid split brain
Role-based Services

- For each application define two services: *_RW and *_RO
  - *_RW service running on Primary
    - *_RO service running on both Primary and Standby
- Using startup trigger to start services that run on all RAC instances on 11gR1
- Using FAN callouts to start singleton RAC services on 11gR1
  - Startup trigger is role-aware
    - but cannot relocate services when their instance fails
  - 11gR1 srvctl is not role-aware
- Role-based services can be used with 11gR2 srvctl for all types of services
Application modifications

• Latency tolerance not globally applicable to application queries
  • Mix of zero and low latency tolerance application queries

• All transactions need to be able to read their own writes immediately

• Application modifications necessary to use role-based services
  • Using database links and synonyms not feasible for our application
  • Stopping and restarting services based on Standby lag not practical either
  • Using connection pool checker would cause frequent invalidations / reconnections

• Application wrapper layer implemented using a Decorator design pattern
  • Wrapper layer consists of mostly standardized code
  • Low marginal cost when new APIs added to application
Runtime service selection

• For each application method determine which service to use based on latency tolerance and transactional affinity
  • For Writes: use *_RW service
  • For zero latency or short Reads: use *_RW service
  • For latency tolerant long Reads:
    • Use *_RW service if already inside a transaction
      • application server transaction APIs used to determine this
    • Use *_RO service if within acceptable staleness
      • In 11gR1 use query_scn rather than v$dataguard_stats to calculate lag
      • In 11gR2 the STANDBY_MAX_DATA_DELAY feature can be used instead of explicitly calculating lag
Load balancing effectiveness

- Query load balancing not perfect due to unnecessary redirects to Primary
  - Overall lag may be large but tables queried not affected
    - checking ora_rowscn not a practical solution
  - Apply Lag measurement precision
    - 3sec in 11gR1 / 1sec in 11gR2
  - Short reads not load balanced
    - to avoid lag calculation overhead

- When Standby down need to stop load balancing queries to avoid stalling due to TCP timeout
  - Cannot use ONS to switch datasource definition in this scenario
  - Setting SQLnetDef.TCP_CONNTIMEOUT_STR low is not adequate
  - A hardware traffic manager can be used to virtualize the location of the *_RO service
  - Application server multi-pools best solution if available
Example DR System

- 10ms redundant network between Primary and DR datacenters
- 50ms – 80ms network between clients and either Primary or DR
- Redo rate up to 3 MB / sec
- Oracle 11g two-node RAC used for both Primary and DR

- Impact on Primary 5% - 10% depending on latency and redo rate
- Standby Apply Lag < 3sec depending on redo rate
- Primary stalling when connectivity to Standby first lost < 10sec
  - NET_TIMEOUT=10
- Total downtime until failover approx. 2 minutes
  - FSFO threshold = 1 minute
    - allows for RAC node eviction and other transitory outages
    - Database failover takes 1 minute to complete once threshold expires
System cost

- Deploy two medium sized systems (in terms of #CPUs) used in tandem instead of two large ones having the second as a passive standby

- Significantly lower overall Oracle licensing costs due to better CPU utilization (even after taking into account the additional Oracle Active Data Guard licenses)

- Lower administration cost than Multi-Master Replication
  - Administrator does not need intimate knowledge of application
  - No effort required to detect and resolve data conflicts

- Not necessary to do backups on both Primary and Standby
  - But alternate backup plans needed when chosen backup site is offline
Conclusion

- Oracle Active Data Guard
  - Can be used to process OLTP query workload – not just reports - with proper application modifications

- No excessive trial-and-error tuning necessary if best practices followed

- Tuning effort is required to minimize impact to Primary when Standby down

- Simple administration but not a lights-out solution

- Overall results depend on the trade-offs you are willing to make
Q & A

Primary

Active Standby

Observer

Data Guard Maximum Availability - SYNC
continuous redo shipping, validation and apply
(up to 10ms network latency - approx 60 miles)