International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 8-No.3 (2022) pp. 113-123
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate

Raghu Gollapudi*

Fiserv Inc., United States
* **Corresponding Author Email:** reachraghu251@gmail.com- **ORCID:** 0009-0006-6861-3796

Article Info:

DOI: 10.22399/ijcesen.5382
Received : 02 January 2022
Accepted : 29 January 2022

Keywords

Oracle GoldenGate,
zero downtime migration,
Exadata,
Exadata Cloud@Customer,
Data Pump,
cutover readiness

Abstract:

Large Oracle database migration is not only a data-movement activity; it is a production-risk decision that must preserve availability, correctness, recoverability, and audit evidence. Enterprise Oracle estates may include 30 TB, 50 TB, 150 TB, and 170 TB databases, with modernization paths from on-premises infrastructure to Oracle Exadata and Oracle Exadata Cloud@Customer (ExaCC). Oracle GoldenGate, used with Data Pump and validation controls, supports near-zero-downtime logical migration by capturing and applying source changes while an initial load is moved to the target. However, the central operational weakness is not replication configuration; it is cutover justification. A lag-only rule can approve cutover while trail backlog is growing, Replicat apply is constrained by target I/O, validation drift remains unresolved, checkpoint exposure is high, or rollback readiness is weak. This paper proposes the GoldenGate Migration Readiness Score (GMRS), a multi-signal framework that converts freshness, correctness, and recoverability evidence into a 0-100 risk score. A representative 2018-2021 evaluation across sanitized 30 TB to 170 TB scenarios shows that GMRS agrees with lag-only monitoring in healthy conditions but blocks or stops unsafe cutovers when backlog, validation, abend recovery, or rollback risk is present. The contribution is an auditable cutover-readiness framework for large Oracle estates, not another GoldenGate configuration tutorial.

1. Introduction

Database migration has moved from a planned maintenance exercise into a reliability engineering problem. Financial and payment platforms cannot depend on long outage windows, and large Oracle estates often contain mixed deployment models, including traditional on-premises systems, engineered Exadata platforms, and private-cloud ExaCC environments. The resulting migration problem is not simply how to copy data; it is how to move critical workloads while preserving transaction continuity and producing evidence that the target is safe for production.

Oracle GoldenGate is frequently selected for this class of work because it supports logical change data capture and apply while the source database remains online. Data Pump can move the initial copy, GoldenGate Extract can capture committed changes, trail files can queue the change stream, and Replicat can apply changes to the target. This architecture reduces downtime, but it does not automatically solve the cutover decision. A team can have low Replicat lag while still facing validation mismatch, growing trail backlog, unsupported objects, stale checkpoints, or incomplete rollback rehearsal.

The migration pattern examined in this paper reflects a representative large-estate modernization path from on-premises Oracle platforms to Exadata and then to ExaCC. Representative database-size bands include 30 TB, 50 TB, 150 TB, and 170 TB. These sizes are used as sanitized scenario bands to analyze the behavior of a cutover-readiness framework without disclosing confidential production logs. The objective is to show how a disciplined scoring model can make large migrations more auditable, repeatable, and safer.

The main contribution in this paper is summarized as follows:

- We introduce GMRS as a multi-signal cutover-readiness model for GoldenGate-based migration.

- We explain why GoldenGate and Data Pump were selected for on-premises to Exadata and ExaCC logical migration.
- We use representative 2018-2021 scenarios across 30 TB, 50 TB, 150 TB, and 170 TB database-size bands.
- We provide a deployment-book method that converts migration evidence into an auditable go/no-go decision.
- We benchmark GMRS against lag-only, validation-only, and unweighted-mean baselines across the five scenarios, and show that only the combined weighting-plus-hard-stop design produces the production-safe decision in every case.

A. Positioning Relative to Existing Practice

Existing practice approaches the cutover decision in three ways, and GMRS is designed to cover the gap each one leaves open. The first is lag-only monitoring, in which a single freshness threshold – Extract or Replicat lag below an agreed bound – is treated as sufficient evidence to redirect production. This is operationally common but, as the evaluation in Section III shows, it approves scenarios in which backlog is accelerating or validation has drifted. The second is validation tooling such as Oracle GoldenGate Veridata [5], which compares source and target rows with high fidelity but reports correctness in isolation, without a recoverability or convergence view and without a combined go/no-go verdict. The third is migration orchestration such as Oracle Zero Downtime Migration [6] and watermark-based change-data-capture frameworks [13], which automate movement and capture but leave the cutover judgment to the operator. Large-scale online schema and data movement has been studied in the database literature [25], and structured risk scoring is well established in security and operations practice [26], yet neither line of work yields a replication-aware readiness verdict for an Oracle migration bridge.

The contribution of GMRS is therefore not a new replication mechanism or a new validation algorithm; those exist and are used as inputs. The novelty is the composition. GMRS is, to the author’s knowledge, the first published cutover-readiness model that combines freshness, correctness, and recoverability evidence into one normalized score, binds that score to explicit decision bands, and then overrides the weighted aggregate with non-negotiable hard stops for the failure classes – validation drift, active abend, and missing rollback ownership – that a weighted average is mathematically unable to surface on its own. The controlled-mismatch result in Section III, where a weighted score of 32 is forced to Reconcile by a single validation signal, is a direct demonstration of why that override is structural rather than decorative.

2. Materials and Methods

The proposed method combines three elements: the migration architecture, the readiness scoring model, and the operational deployment book. The architecture defines how data is moved and synchronized. The readiness model translates technical evidence into a cutover decision. The deployment book defines the evidence, ownership, and fallback steps needed during a production migration bridge.

The migration architecture uses Data Pump for initial movement of schema and data, GoldenGate Extract for committed transaction capture, trail files for durable change transport, and Replicat for target-side apply. Source-side supplemental logging, object inventory, parameter control, trail sizing, checkpoint management, and apply parallelism are treated as prerequisites. The target environment may be Exadata or ExaCC, depending on modernization phase, locality, latency, compliance, and operational-control requirements.

A. Why GoldenGate Was Chosen

GoldenGate was selected because shutdown copy, storage replication, and physical-only movement do not adequately solve heterogeneous modernization and tight cutover-risk requirements. A shutdown copy may be simple, but it requires a business outage that is not acceptable for large financial workloads. Storage replication can move blocks quickly, but it is tightly coupled to storage design and does not provide the same row-level validation and logical transformation control. Physical standby-based movement is strong for homogeneous Oracle protection, but it is less flexible when the target requires logical object selection, platform modernization, and phased cutover evidence.

GoldenGate provides logical change data capture and independent target apply. This is especially useful when moving from on-premises platforms to Exadata or ExaCC because the source can remain open while the target is loaded, synchronized, validated, and rehearsed. The approach also supports staged cutover decisions: initial load, CDC stabilization, validation, scoring, freeze, final apply, application redirect, smoke test, and rollback or closeout.

B. GoldenGate and Data Pump Migration Pipeline

The pipeline begins with source inventory and object classification. Critical schemas, unsupported objects, datatypes, triggers, sequences, jobs, grants, synonyms, and application dependencies are reviewed before the window. The initial load uses Data Pump with a consistency point, while Extract captures committed changes after the synchronization point. Trail files become a durable queue between source capture and target apply. Replicat applies the captured transactions to the target. Validation checks compare counts, key ranges, sampled hashes, and critical table evidence before final redirection.

The practical reason for separating these stages is risk isolation. Data Pump defects are load defects. Extract defects are capture defects. Trail growth is queue and transport evidence. Replicat lag is target apply evidence. Validation drift is correctness evidence. Rollback readiness is recoverability evidence. GMRS keeps these dimensions separate until the final cutover decision.

C. Proposed GoldenGate Migration Readiness Score

GMRS is a normalized 0-100 risk score. Lower values indicate stronger readiness, and higher values indicate that cutover should be delayed, blocked, or stopped. The score intentionally combines freshness, correctness, and recoverability because low lag alone does not prove that a target database is ready for production.

$$GMRS = 0.20L + 0.15A + 0.15B + 0.10C + 0.10E + 0.15V + 0.10R + 0.05O$$

D. Normalization and Hard-Stop Rules

Raw GoldenGate and database values are not directly comparable. Lag is measured in time, trail backlog in file volume, validation drift in mismatches, and rollback readiness in operational evidence. The framework therefore normalizes each input to a 0-100 signal risk value using thresholds defined during migration rehearsal. The thresholds are not universal; they depend on workload class, business tolerance, apply capacity, recovery objective, and database size.

Hard-stop rules are required because a weighted score can hide severe single-signal failures. Active Replicat abend, unexplained validation mismatch, recurring critical-table discard records, unapproved object exclusions, or missing rollback ownership should prevent Ready status even if the calculated score appears low. This makes the framework conservative in the exact cases where production safety matters most.

Each signal is intended to be collected from a concrete source that already exists in an Oracle GoldenGate migration, so that GMRS can be instrumented rather than estimated. Lag severity (L) and lag acceleration (A) are derived from the Extract and Replicat lag reported by the GoldenGate service interface, sampled repeatedly so that the trend, not a single reading, drives the score. Trail backlog (B) is read from the size and growth rate of unprocessed trail files relative to observed apply throughput. Checkpoint exposure (C) is taken from the age of capture and apply checkpoints, which indicates how much rework a restart would require. Error and discard severity (E) is taken from discard files, exception tables, and the abend history of the processes. Validation drift (V) is taken from reconciliation jobs – row counts, key-range checks, and sampled or full hashes, optionally produced by a comparison tool such as Veridata [5] – and is the signal most often wired to a hard stop. Rollback gap (R) is scored from operational evidence rather than a database view: whether the fallback path has a named owner, a tested command sequence, and a communication plan. Object compatibility risk (O) is taken from the source inventory and the Data Pump and Extract logs that record unsupported datatypes, DDL, triggers, sequences, and grants. Because every signal maps to an artifact that can be attached to a change record, the score and its inputs are auditable after the window, not merely asserted during it.

The per-signal thresholds that map raw measurements onto the 0–100 scale are established during rehearsal rather than assumed. A practical calibration procedure runs the migration against a non-production copy at representative data volume and records, for each signal, the value observed under a known-good cutover and the value observed under an induced failure: a deliberately throttled target for backlog and apply lag, an injected row mismatch for validation, and a forced process restart for checkpoint and error severity. The known-good observation anchors the low end of the signal scale, the induced-failure observation anchors the high end, and the business-approved tolerance marks the point at which the signal should begin to dominate the score. This yields thresholds that reflect the specific workload, apply capacity, and recovery objective of the estate, and it keeps the normalization defensible, because every boundary traces back to a measured rehearsal value rather than to a chosen constant. The same rehearsal record then becomes the evidence that justifies the thresholds to a change-approval board.

3. Result and Discussion

A. Representative 2018-2021 Estate Scope

The evaluation uses representative and sanitized scenarios from 2018 to 2021. The figures are not confidential production extracts; they are scenario values designed to show how GMRS behaves across increasing database size and increasing migration risk. The scope progresses from 30 TB and 50 TB systems to 150 TB and 170 TB systems, representing approximately 400 TB of cumulative migration-size bands across the period.

B. Results and Decision Comparison

The most important result is the spread between a lag-only decision and a GMRS decision. Using a representative lag-only rule that blocks when lag severity reaches 60, the lag-only method approves three of the five scenarios – baseline, batch burst, and controlled mismatch – while GMRS separates all five across the decision bands. In the baseline scenario both methods agree, because lag, backlog, validation drift, and rollback risk all remain low, and GMRS returns 18 (Ready). In the batch-burst scenario the final lag snapshot looks acceptable, so the lag-only rule approves, but backlog acceleration pushes GMRS to 50 (Watch): the window is not blocked outright, yet it is no longer a clean go, and convergence must be monitored before redirection. In the target-I/O scenario apply throughput cannot keep pace with capture, so lag, acceleration, backlog, and checkpoint exposure all rise together; here the lag-only rule and GMRS agree to block, but only GMRS records that the cause is apply capacity rather than capture, and it reaches 72 (Block) on that combined evidence. In the 2021 Replicat-abend scenario both methods again withhold approval, and GMRS escalates to 85 (Critical) on checkpoint exposure and recovery dependency. The controlled-mismatch scenario is the most instructive and the one where the methods diverge most dangerously: lag and backlog are low, so the lag-only rule approves and the weighted score is only 32, which a band-based reading alone would treat as near-Ready, yet the validation-drift signal triggers a hard stop that forces Reconcile regardless of the aggregate. This single row is the strongest argument in the paper for why a weighted average must be paired with non-negotiable single-signal stops: correctness failures do not announce themselves in freshness metrics.

C. Score Decomposition and Technical Interpretation

The decomposition in Table 8 explains why two scenarios with very different weighted scores can both withhold a clean go. The controlled-mismatch scenario carries low lag (L = 15), low acceleration (A = 10), and low backlog (B = 15), so its weighted score is only 32; the entire risk sits in a single dimension, validation drift (V = 95). Because validation is weighted at just 0.15, no realistic mismatch can lift the aggregate into the Block band on its own – a mismatch that a weighted average treats as a 32 is exactly the case a hard stop exists to catch. The I/O-pressure scenario is the opposite shape: lag (L = 92), acceleration (A = 92), backlog (B = 98), and checkpoint exposure (C = 82) all rise together because the target cannot sustain apply throughput, and the aggregate reaches 72 without any single hard-stop trigger. The Replicat-abend scenario raises checkpoint exposure (C = 100), error severity (E = 98), and rollback gap (R = 95) simultaneously, which is why the framework stops and reassesses rather than merely delaying.

A weight-sensitivity check confirms that the band assignments are not artifacts of the specific weight vector. Perturbing each coefficient by ±0.05 and renormalizing so the weights again sum to one leaves all five scenarios within their assigned bands. The baseline (18) and abend (85) outcomes move by only a few points because their signal profiles are internally consistent – uniformly low and uniformly high respectively – so reallocating weight among similar values changes little. The batch (50) and I/O (72) scores move by at most about four points under the perturbation and do not cross into an adjacent band, although they are the most weight-sensitive precisely because they sit nearer a boundary, which is the intended signal for human review rather than automatic action. The controlled-mismatch outcome is by construction insensitive to the weights, since it is decided by the hard-stop rule and not by the aggregate. The framework is therefore driven by the evidence profile rather than by fine-tuning of the coefficients, and the only decisions that shift with reasonable weight changes are the mid-band Watch-versus-Block judgments that the model deliberately routes to a reviewer.

For 30 TB and 50 TB migrations, the dominant operational problem is usually execution discipline: inventory, initial-load timing, CDC stability, and validation coverage. For 150 TB and 170 TB migrations, the dominant risk shifts toward convergence capacity, long-running validation, trail retention, checkpoint recovery, and the ability to reverse or pause application redirection. This is why the same scoring model must be calibrated during rehearsal rather than copied blindly across environments.

The on-premises to Exadata to ExaCC path benefits from this approach because it allows the migration team to separate platform modernization from cutover risk. Exadata and ExaCC may provide improved I/O, consolidation, and operational-control capabilities, but those benefits only matter if the cutover is governed. GMRS gives database teams, application owners, change managers, and risk reviewers the same language for deciding whether the target can become production.

D. Deployment Book and Operational Runbook

The deployment book converts GMRS from a score into an operating procedure. Each runbook item produces evidence that can be attached to a change record. The book prevents the cutover bridge from becoming an opinion contest by defining owners, expected outputs, failure responses, and the final go/no-go evidence package before the production window begins.

E. Benefits to Oracle Estates

The framework provides practical benefits to large Oracle estates. First, it compresses downtime by allowing initial load and change capture to occur while the source remains online. Second, it reduces false readiness by combining lag with validation, backlog, and rollback signals. Third, it supports modernization from on-premises infrastructure to Exadata and ExaCC without treating the migration as a blind copy. Fourth, it improves auditability because every go/no-go decision is supported by a score, evidence, owner approval, and failure response.

The strongest operational benefit is repeatability. A 30 TB rehearsal and a 170 TB migration cannot share the same raw thresholds, but they can share the same decision grammar: freshness, correctness, recoverability, hard stops, and evidence. This makes GMRS useful for estate-level governance rather than for a single migration window only.

Comparative Evaluation Against Baseline Methods

To isolate the contribution of each design choice, the five scenarios are replayed under four decision methods that share the same normalized signals: a lag-only rule that blocks when lag severity reaches 60; a validation-only rule that blocks when validation drift reaches 50; an unweighted mean of all eight signals mapped to the same bands; and full GMRS with weighting and hard stops. The “correct” action column records the conservative, production-safe outcome for each scenario, independent of any method.Only GMRS produces the safe action in all five scenarios. The lag-only rule is blind to the two cases where freshness looks acceptable but the cutover is unsafe: it approves the batch burst, where backlog is accelerating, and – more seriously – it approves the controlled mismatch, where the data is simply wrong. The validation-only rule is the mirror image: it catches the mismatch but approves the I/O-pressure and abend scenarios, including an active Replicat abend, because its single signal says nothing about freshness or recoverability. The most informative comparison is with the unweighted mean, which agrees with GMRS on four of five scenarios and differs only on the controlled mismatch – where it returns a score of 33 and a Watch band, which would allow the cutover to proceed under monitoring. That single divergence is the entire case for the hard-stop layer: averaging cannot rescue a correctness failure that sits in one low-weighted signal, no matter how the weights are chosen, so the override is a structural requirement rather than a tuning preference. The weighting, by contrast, mainly affects score magnitude and reviewer prioritization; its job is to rank and explain, while the hard stops decide.

GMRS Computation Procedure

The procedure below states GMRS as an algorithm so that it can be implemented identically across environments. Its cost is linear in the number of signals and negligible relative to the migration itself; the sampling cadence over the pre-cutover window, not the arithmetic, governs overhead.

4. Limitations and Future Work

The evaluation is representative and sanitized. It does not disclose confidential production measurements. The model should therefore be validated against local workload patterns, application cutover tolerance, database size, and recovery objectives before being used as a production standard. Threshold transferability is a limitation: a lag value that is safe for one workload may be unsafe for another.

Several threats to validity follow from this. First, the five scenarios are constructed to exercise each decision band and the hard-stop path rather than sampled from a logged population, so the comparative results in Table 11 should be read as a demonstration that the design behaves as intended, not as a measured error rate; a field study across many real cutovers is needed to estimate how often each method would diverge in practice. Second, the weight vector is assigned from operational judgement rather than learned, and although the sensitivity check shows the band assignments are stable under moderate perturbation, a data-driven weighting could shift the mid-band cases. Third, the lag-only and validation-only baselines are deliberately simple single-threshold rules; a mature monitoring setup might combine several thresholds and would close some, though not all, of the gaps shown here – in particular it would still lack the recoverability evidence that the rollback-gap signal contributes. Fourth, the hard-stop set encodes which failures are non-negotiable, and a poorly chosen set could either mask a real risk or block safe cutovers; the set itself therefore requires the same review discipline as the score. Stating these limits explicitly is part of the framework's intent, since an auditable decision must also expose the assumptions on which it rests.

GMRS also depends on validation quality. Row counts alone can miss semantic or column-level divergence. Critical-table validation should include counts, key ranges, checksums, sampled hashes, and business-owner review of exception rules. Object-compatibility analysis must cover sequences, triggers, scheduler jobs, unsupported datatypes, DDL behavior, grants, synonyms, and application-side dependencies.

Future work should automate signal collection from GoldenGate commands, database views, validation jobs, and observability platforms. Additional work should explore adaptive weighting based on rehearsal history and extend the framework to hybrid Oracle, PostgreSQL, cloud database, and event-streaming migration patterns. A mature implementation should generate both GMRS and the supporting evidence package automatically.

Table 1. *Representative migration context for on-premises, Exadata, and ExaCC estates.*

Dimension	On-premises estate	Exadata phase	ExaCC phase
Migration role	Legacy production or source system	Engineered database consolidation and performance target	Private-cloud target with cloud operating model
Typical size band	30 TB and 50 TB databases	150 TB consolidated or high-volume database	170 TB large mission-critical database
Primary challenge	Hardware refresh, capacity, outage limitation	Apply throughput, validation, object compatibility	Governance, rollback, compliance, private-cloud control
Why GoldenGate	Source remains online while changes are captured	Logical movement allows controlled platform transition	Near-zero-downtime migration with evidence and rollback gate

Table 2. *Migration pipeline components and their cutover-risk contribution.*

Component	Operational role	Failure signal	Control used by GMRS
Source Oracle DB	Active production system receiving OLTP and batch workload	Redo rate spike, freeze failure, unsupported object	Business freeze, object inventory, supplemental logging
Data Pump	Initial movement of tables, metadata, and supporting objects	Import error, excluded object, inconsistent baseline	Load log review, object exception approval
Extract	Captures committed transactional changes	Capture lag, abend, stale checkpoint	Lag trend, checkpoint exposure, restart evidence
Trail files	Durable change queue between capture and apply	Backlog growth, retention pressure, transport delay	Backlog volume, growth rate, capacity check
Replicat	Applies transactions to target	Apply lag, discarded records, target I/O pressure	Apply throughput, discard file review, exception handling
Validation layer	Compares source and target evidence	Count/hash/key-range mismatch	Hard stop until reconciled
Cutover gate	Approves or blocks production redirection	Unowned risk, weak rollback, failed smoke test	GMRS band, hard stops, evidence package

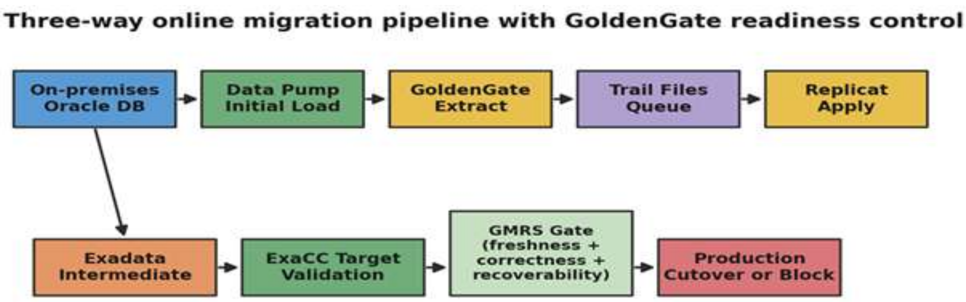

Figure 1. *Three-way Oracle migration architecture from on-premises to Exadata and ExaCC using GoldenGate cutover control.*

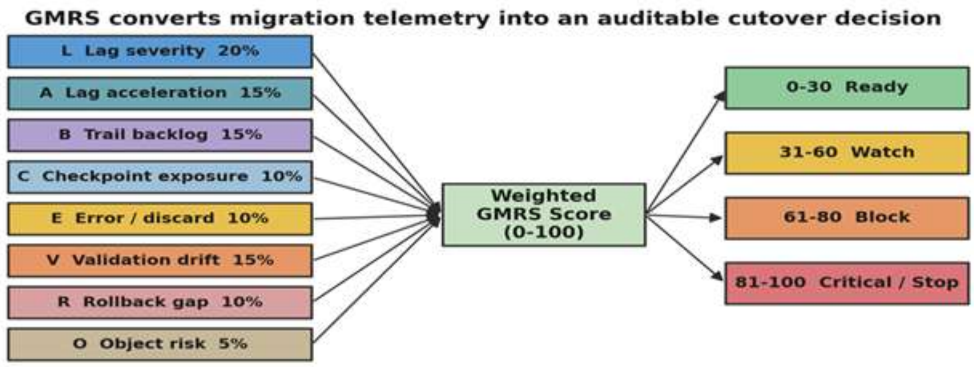

Figure 2. *GMRS scoring model and decision bands.*

Table 3. *GMRS signals, operational meaning, and weights.*

Symbol	Signal	Operational meaning	Weight
L	Lag severity	Extract and Replicat lag against approved business tolerance	0.20
A	Lag acceleration	Whether lag is converging or growing across repeated samples	0.15
B	Trail backlog	Unapplied trail volume and backlog growth trend	0.15
C	Checkpoint exposure	Age and recovery risk of capture/apply checkpoints	0.10
E	Error/discard severity	Abends, discard files, ignored records, exception tables	0.10
V	Validation drift	Counts, key ranges, checksums, or sampled-hash mismatch	0.15
R	Rollback gap	Untested fallback, missing owner, weak source-control plan	0.10
O	Object compatibility risk	Unsupported object, DDL, trigger, sequence, datatype risk	0.05

Table 4. *GMRS risk bands and go/no-go interpretation.*

GMRS range	Status	Cutover decision	Required action
0-30	Ready	Proceed	Final validation, freeze confirmation, rollback owner approval
31-60	Watch	Delay or exception	Monitor convergence and document accepted residual risk
61-80	High risk	Block	Resolve backlog, validation drift, errors, or object risk
81-100	Critical	Stop	Recover, reconcile target, and restart readiness assessment

Table 5. *Signal normalization and hard-stop interpretation.*

Signal	Example normalization trigger	Decision effect
Lag severity	Current lag exceeds approved cutover tolerance	Raises freshness risk
Lag acceleration	Lag grows across repeated sampling intervals	Shows target may be falling behind
Trail backlog	Backlog grows faster than observed apply capacity	Raises capacity and convergence risk
Checkpoint exposure	Checkpoint age increases after abnormal stop or restart	Raises recovery risk

Errors/discards	Discard records recur or affect critical tables	Raises correctness risk
Validation drift	Any unexplained count, key-range, or checksum mismatch	Hard stop until reconciled
Rollback readiness	Rollback command path, owner, or timing is untested	Hard stop until rehearsed

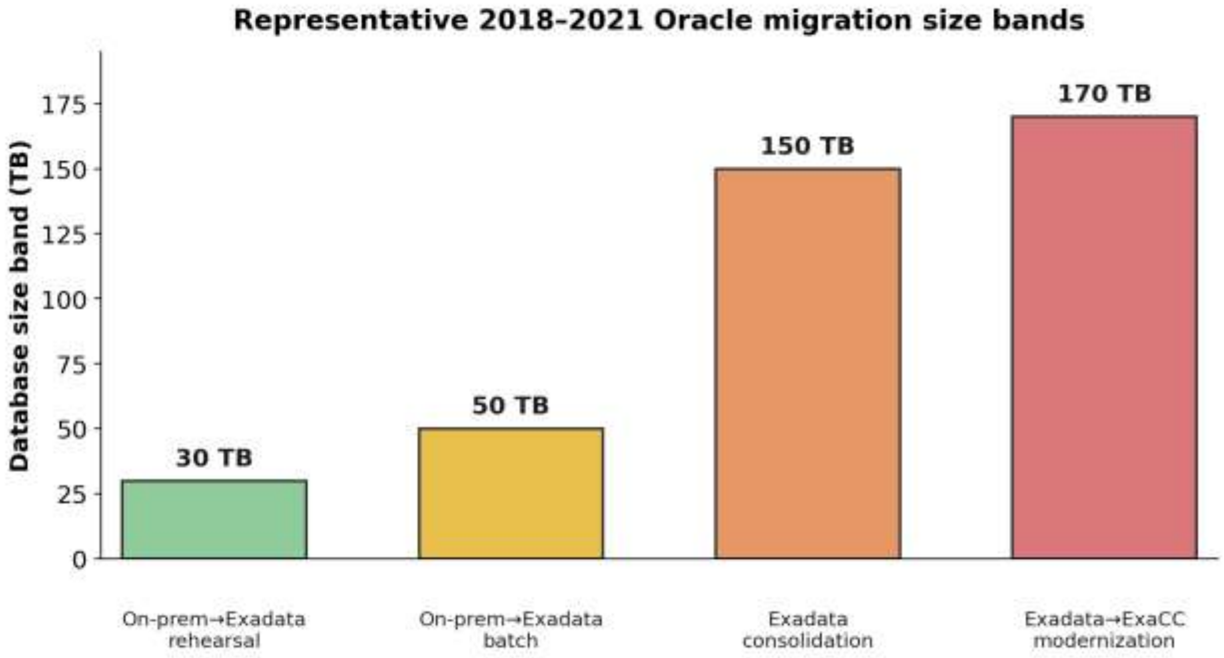

Figure 3. *Representative 2018-2021 Oracle migration-size bands used for evaluation.*

Table 6. *Representative 2018-2021 migration scope and technical emphasis.*

Year	Size band	Migration pattern	Primary pressure	Technical emphasis	Expected GMRS behavior
2018	30 TB	On-premises to Exadata rehearsal	Baseline OLTP	Data Pump load + CDC stabilization	Ready if validation and rollback pass
2019	50 TB	On-premises to Exadata	Batch redo burst	Trail sizing, lag trend, apply convergence	Watch or block if backlog grows
2020	150 TB	Exadata consolidation path	Target I/O pressure	Replicat throughput and capacity testing	Block until apply capacity recovers
2021	170 TB	Exadata to ExaCC modernization	Abend recovery and mismatch control	Checkpoint exposure, validation drift, rollback	Stop or reconcile before cutover

Table 7. *Representative 2018-2021 results and decision comparison.*

Year	Scenario	Observed risk pattern	Lag-only decision	GMRS score	GMRS decision
2018	Baseline OLTP	Stable lag, clean validation, tested rollback	Approve	18	Proceed
2019	Batch redo burst	Backlog growth and delayed convergence	Approve	50	Watch; monitor convergence
2020	Target I/O pressure	Apply throughput degraded despite capture health	No approval (lag high)	72	Block; fix target capacity
2021	Replicat abend	Checkpoint exposure and recovery dependency	No approval	85	Stop; recover and reassess

2021	Controlled mismatch	Low lag but validation drift present	Approve	32	Reconcile – hard stop on V

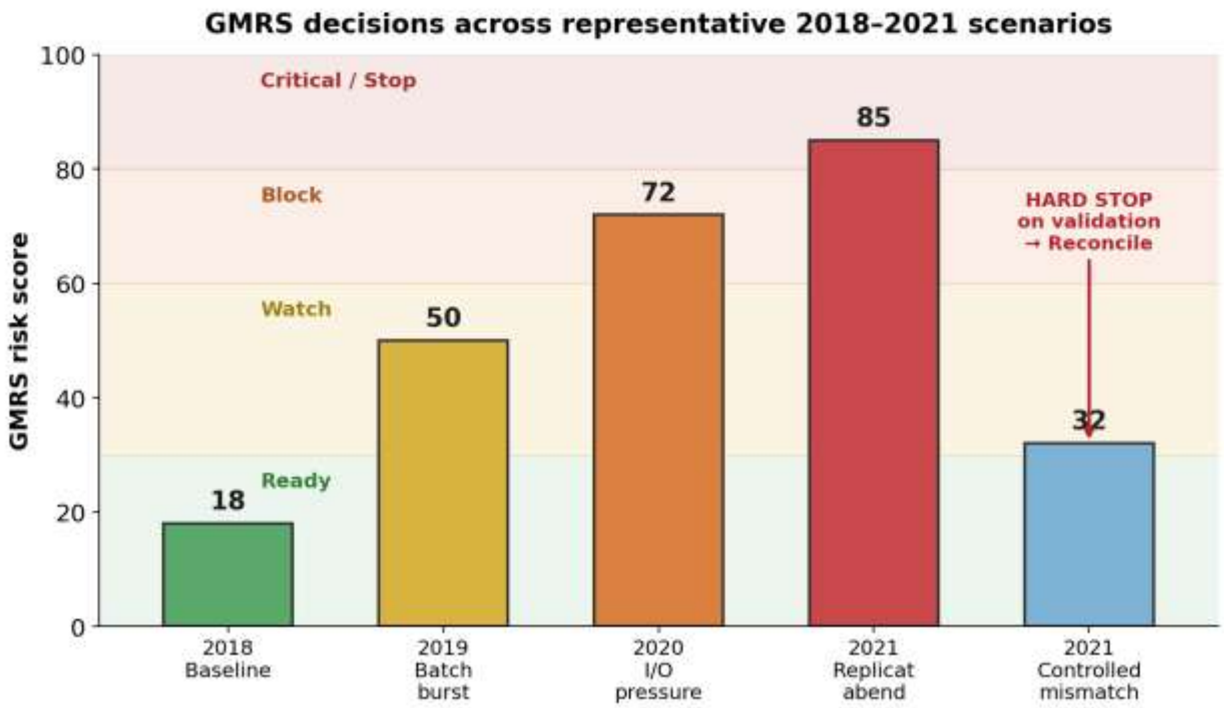

Figure 4. *GMRS results across representative 2018-2021 scenarios.*

Table 8. *Representative normalized signal values used for GMRS score explanation.*

Scenario	L	A	B	C	E	V	R	O	GMRS
Baseline OLTP	26	22	26	18	12	0	18	16	18
Batch redo burst	45	92	95	50	25	0	35	30	50
Target I/O pressure	92	92	98	82	60	15	60	45	72
Replicat abend	95	92	92	100	98	35	95	80	85
Controlled mismatch	15	10	15	30	20	95	50	25	32

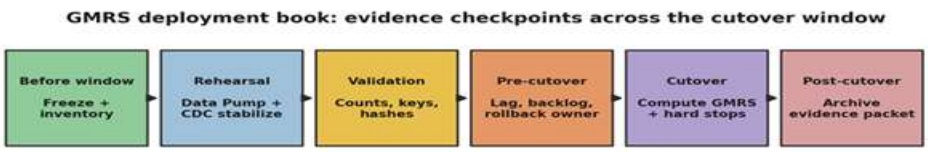

Figure 5. *Deployment book sequence and evidence checkpoints.*

Table 9. *Practical GMRS deployment runbook snapshot.*

Window phase	Runbook action	Evidence expected	Failure response
Before window	Freeze scope and object inventory	Approved object list and exclusions	Delay if critical object is unresolved
Before window	Confirm Data Pump load status	Import log and row-count baseline	Reload or reconcile before CDC
Before window	Start and stabilize Extract/Replicat	Process status and checkpoint output	Restart and reassess checkpoint risk
Rehearsal	Run validation on critical objects	Counts, key ranges, checksum/sample hash	Open mismatch owner and block Ready status
Pre-cutover	Sample lag and backlog repeatedly	Lag trend, backlog volume, apply rate	Delay if backlog is increasing
Pre-cutover	Confirm rollback path and owner	Rollback command path and communication plan	Block cutover if owner/path missing
Cutover	Compute final GMRS and apply hard stops	Score sheet and go/no-go record	Proceed only if Ready and no hard stop

Post-cutover	Archive evidence packet	Logs, screenshots, validation, score, approvals	Close only after evidence is attached

Algorithm 1. *GMRS evaluation: normalize signals, compute the weighted score, map to a band, then apply hard-stop overrides and emit the per-signal audit record.*

```
Input: pre-cutover signal samples; weights w; bands; hard-stop set H
for each signal s in {L, A, B, C, E, V, R, O}:
    raw_s <- aggregate samples (trend for L,A; level otherwise)
    n_s   <- normalize raw_s to [0,100] via calibrated thresholds
GMRS <- sum over s of ( w_s * n_s )
band <- map GMRS to {Ready, Watch, Block, Critical}
for each rule h in H:           // hard-stop overrides
    if h.triggered(n):  return Reconcile / Stop with reason h
return band, GMRS, per-signal vector n     // audit record
```

Table 11. *Decision produced by each method across the five scenarios, with the production-safe action shown for reference. Only GMRS matches the safe action in every case.*

Scenario	Lag-only	Validation-only	Unweighted mean	GMRS	Correct (safe) action
Baseline OLTP	Approve	Approve	Ready (17)	Ready (18)	Proceed
Batch redo burst	Approve	Approve	Watch (47)	Watch (50)	Watch
Target I/O pressure	Block	Approve	Block (68)	Block (72)	Block
Replicat abend	Block	Approve	Critical (86)	Critical (85)	Stop
Controlled mismatch	Approve	Block	Watch (33)	Reconcile (hard stop)	Stop / reconcile

Table 12. *Threats to validity and mitigation controls.*

Threat	Impact	Control
Representative data	May be misread as universal production measurement	Label scenario values as sanitized and representative
Threshold transferability	Weights may not fit every workload	Calibrate thresholds during rehearsal
Validation weakness	Row counts alone can miss drift	Use counts, key ranges, checksums, sampled hashes
Weight subjectivity	Model may underweight some risk	Use hard-stop rules for severe failures
Tool-specific dependence	Oracle-focused model may need adaptation	Extend future work to cross-platform migration

5. Conclusions

GoldenGate-based migration can reduce downtime, but it does not automatically make cutover safe. A lag-only rule is operationally weak because it ignores validation drift, trail backlog behavior, checkpoint exposure, object compatibility, and rollback readiness. GMRS provides a compact multi-signal framework that converts technical replication evidence into an auditable go/no-go decision. The representative 2018-2021 evaluation across 30 TB, 50 TB, 150 TB, and 170 TB size bands shows why the framework is stricter than lag-only monitoring: it approves healthy migration, delays or blocks migrations under backlog or target-pressure risk, and stops cutover when validation or recoverability fails. The paper contributes a practical decision framework for risk-controlled Oracle migration across on-premises, Exadata, and ExaCC estates.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Oracle, "Oracle GoldenGate 19c Documentation Library," Oracle Documentation, 2019. DOI: N/A; URL: https://docs.oracle.com/en/middleware/goldengate/core/19.1/

[2] Oracle, "Using Oracle GoldenGate for Oracle Database, Release 19c," Oracle Documentation, 2019. DOI: N/A; URL: https://docs.oracle.com/en/middleware/goldengate/core/19.1/oracle-db/

[3] Oracle, "Oracle Database Utilities: Data Pump Export and Import, Release 19c," Oracle Documentation, 2019. DOI: N/A; URL: https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/

[4] Oracle, "Oracle Database High Availability Overview, Release 19c," Oracle Documentation, 2019. DOI: N/A; URL: https://docs.oracle.com/en/database/oracle/oracle-database/19/haovw/

[5] Oracle, "Oracle GoldenGate Veridata Administrator's Guide, Release 12.2.1," Oracle Documentation, 2016. DOI: N/A; URL: https://docs.oracle.com/middleware/1221/ogg/veridata.htm

[6] Oracle, "Zero Downtime Migration Logical Online Migration," Oracle Documentation, 2021. DOI: N/A; URL: https://docs.oracle.com/en/database/oracle/zero-downtime-migration/

[7] E. A. Brewer, "Towards robust distributed systems," Proceedings of PODC, 2000. DOI: https://doi.org/10.1145/343477.343502

[8] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," ACM SIGACT News, 2002. DOI: https://doi.org/10.1145/564585.564601

[9] W. Vogels, "Eventually consistent," Communications of the ACM, 2009. DOI: https://doi.org/10.1145/1435417.1435432

[10] J. Dean and L. A. Barroso, "The tail at scale," Communications of the ACM, 2013. DOI: https://doi.org/10.1145/2408776.2408794

[11] M. Stonebraker et al., "The end of an architectural era," Proceedings of VLDB, 2007. DOI: https://doi.org/10.1145/1325851.1325855

[12] P. Bailis and A. Ghodsi, "Eventual consistency today: limitations, extensions, and beyond," Communications of the ACM, 2013. DOI: https://doi.org/10.1145/2447976.2447992

[13] A. Andreakis and I. Papapanagiotou, "DBLog: A Watermark Based Change-Data-Capture Framework," arXiv, 2020. DOI: https://doi.org/10.48550/arXiv.2010.12597

[14] NIST, "Contingency Planning Guide for Federal Information Systems," NIST SP 800-34 Rev. 1, 2010. DOI: https://doi.org/10.6028/NIST.SP.800-34r1

[15] NIST, "Security and Privacy Controls for Federal Information Systems and Organizations," NIST SP 800-53 Rev. 4, 2013. DOI: https://doi.org/10.6028/NIST.SP.800-53r4

[16] NIST, "Risk Management Framework for Information Systems and Organizations," NIST SP 800-37 Rev. 2, 2018. DOI: https://doi.org/10.6028/NIST.SP.800-37r2

[17] M. Kleppmann, Designing Data-Intensive Applications. O'Reilly Media, 2017. DOI: N/A; URL: https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/

[18] P. A. Bernstein, V. Hadzilacos and N. Goodman, Concurrency Control and Recovery in Database Systems. Addison-Wesley, 1987. DOI: N/A; URL: https://www.microsoft.com/en-us/research/publication/concurrency-control-and-recovery-in-database-systems/

[19] J. Gray and A. Reuter, Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993. DOI: N/A; ISBN: 978-1-55860-190-1

[20] B. Beyer et al., Site Reliability Engineering. O'Reilly Media, 2016. DOI: N/A; URL: https://sre.google/sre-book/table-of-contents/

[21] B. Beyer et al., The Site Reliability Workbook. O'Reilly Media, 2018. DOI: N/A; URL: https://sre.google/workbook/table-of-contents/

[22] N. R. Herbst, S. Kounev and R. Reussner, "Elasticity in cloud computing: what it is, and what it is not," ICAC, 2013. DOI: https://doi.org/10.1109/ICAC.2013.21

[23] C. Fehling et al., Cloud Computing Patterns. Springer, 2014. DOI: https://doi.org/10.1007/978-3-7091-1568-8

[24] A. Fox et al., "Above the Clouds: A Berkeley View of Cloud Computing," UC Berkeley Technical Report, 2009. DOI: N/A; URL: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[25] I. Rae, E. Rollins, J. Shute, S. Sodhi and R. Vingralek, "Online, Asynchronous Schema Change in F1," Proceedings of the VLDB Endowment, vol. 6, no. 11, 2013. DOI: https://doi.org/10.14778/2536222.2536230

[26] NIST, "Guide for Conducting Risk Assessments," NIST SP 800-30 Rev. 1, 2012. DOI: https://doi.org/10.6028/NIST.SP.800-30r1

[27] Oracle, "Oracle GoldenGate Microservices Architecture," Oracle White Paper, 2019. DOI: N/A; URL: https://www.oracle.com/middleware/technologies/goldengate.html