Sciences du Nord Nature Science and Technology
Vol. 1, No. 02, December 2024: pp. 162-176
E-ISSN: 3046-6504
P-ISSN: 3047-4922
DOI: https://doi.org/10.71238/snnst.v1i02.253

Predictive Migration Intelligence for Risk-Aware Transformation of Mission-Critical Cloud Platforms

Pranayachandanreddy Gottimukkala [1], Sangeeth Kumar Vanam [2], Gnana Sunny Antony [3], Spencer Bharath Amirthaiyan [4]
[1] Independent Researcher and pranaychandanreddy@gmail.com
[2] Independent Researcher and sangeethvanam@gmail.com
[3] Independent Researcher and Sunnyantony91@gmail.com
[4] Independent Researcher and spencerbharath@gmail.com
Coresponding Author: pranaychandanreddy@gmail.com

ARTICLE HISTORY
Received Dec, 2024 **Revised Dec, 2024** **Accepted Dec, 2024**

ABSTRACT
The move of mission critical workloads to cloud platforms brings with it higher risk of outages, latency problems, architectural lock-in, and cost risk, with the decision to move often based on a static checklist and not predictive evidence. In this systematic review 26 peer-reviewed articles and standards are synthesized to build a holistic picture of predictive migration intelligence, defined as workload forecasting, failure prediction, and quantified risk models, that can be used to plan, sequence, and supervise cloud transformation. The data from sources were segmented into six thematic clusters and quantitized, normalized and compared across the thematic clusters. The synthesis found that very large datacenters had reported cost advantages ranging from 5.7 to 7.1 times compared to medium sized datacenters, that a 1% per-server probability of slow response led to 63% of requests being delayed when fanned out across 100 servers, and that production machine learning models achieved accuracy rates of 79% to 90% in predicting the behavior of virtual machines. Hedged requests were reported to be able to decrease the 99.9 percentile latency from 1,800ms to 74ms for only 2% more requests, and an analysis of 597 outages over 32 services showed that both migration changes and misconfiguration were among the most common triggers for outages. Based on this, a layered approach is suggested that includes a shared knowledge layer comprising a Monitor-Analyze-Plan-Execute loop, where risk exposure scored, tail-aware performance predicted and failure forecasted by this layer serve as gators to each migration wave. Ethical, regulatory and operational challenges to the use of such predictive components are also explored.

Keywords: *Autonomic Computing; Cloud Migration; Failure Prediction; Microservices; Mission-Critical Systems; Predictive Analytics; Risk Exposure; Tail Latency; Workload Forecasting*

INTRODUCTION

Cloud computing is now accepted as a model of services comprising a pool of configurable resources provided on demand via a network and shared between multiple service consumers, with minimal or no interaction between the service provider and the service consumer to provision and release services [1]. Migration is a high consequence engineering action when the model is used for mission critical platforms, defined as platforms whose service disruptions result in serious financial, safety, and societal damage. Migration, in this case, refers to moving, and often rearchitecting, applications, data, and operations from legacy environments to cloud environments.

As illustrated in Figure 1 [2] the economic argument for migration was laid out with the observation that very large datacenters get significant economies of scale over medium sized

datacenters, namely around 7.1, 5.7 and 7.1 on network, storage and administration costs respectively. Elasticity was also defined by cost associativity, which means that 1,000 servers for 1 hour and 1 server for 1,000 hours will cost the same. However, challenges such as service availability, data lock-in, data transfer delays and performance variability were also found and further exacerbated by moving mission-critical workloads.

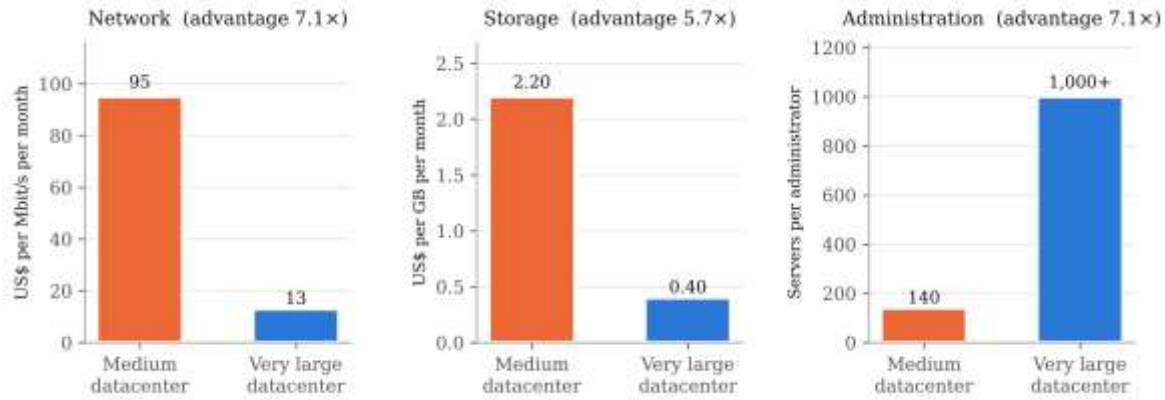

Figure 1. Economies of Scale of Very Large Versus Medium-Sized Datacenters Across Network, Storage, and Administration Costs

Source: Adapted from the cost comparison reported in [2].

A complementary survey defined cloud as a multi-layered system that encompasses hardware, infrastructure, platform and application layers, and listed open research issues such as automated provisioning of services, virtual machine migration, server consolidation, energy management, traffic management, and data security [3]. Other challenges are similar to migration activities as they involve moving resources, changing network connections, and changing security boundaries at the same time. The survey also highlighted the need for foresight in provisioning decisions, and the present review has taken a predictive approach to provisioning.

Knowing about the risks involves having the right set of words for reliance. Dependability is broken down into availability, reliability, safety, confidentiality, integrity and maintainability, and is achieved by preventing faults, tolerating faults, removing faults, and forecasting faults [4]. In this review, we define predictive migration intelligence (PMI) as the systematic application of forecasting models, failure predictors and quantified risk estimates to plan, sequence and supervise cloud transformation efforts.Predictive migration intelligence (PMI) in this review is defined as the systematic application of forecasting models, failure predictors and quantified risk estimates to plan, sequence and supervise cloud transformation efforts. The gap being bridged is the lack of a unified synthesis of the knowledge of the migration process and predictive operational analytics.

METHODS

The review protocol was adapted from a previous systematic review of cloud migration studies has reduced 23 primary studies to a Cloud Reference Migration Model (Cloud-RMM) that includes planning, execution, and evaluation processes, as well as crosscutting concerns [5]. The evidence corpus consisted of 26 sources published from 1991 to 2022 in the present review. All sources were evaluated as to their relevance to migration planning, predictive analysis, and

reliability and data items pertaining to methods, quantitative results and limitations were extracted into a structured format. The classification produced is presented in Table 1, and in graphically illustrated in the following table illustrating the temporal distribution of the corpus:

Table 1. Classification of the Evidence Corpus by Source Type, Thematic Cluster, and Contribution

Ref.	Source	Type	Cluster	Principal contribution to PMI
[1]	Mell and Grance (2011)	Standard	Cloud foundations	Reference definition of characteristics, service models, and deployment models
[2]	Armbrust et al. (2010)	Analysis	Cloud foundations	Economies of scale, elasticity, and principal obstacles
[3]	Zhang et al. (2010)	Survey	Cloud foundations	Layered architecture and open provisioning and migration challenges
[4]	Avizienis et al. (2004)	Taxonomy	Dependability and risk	Attributes, threats, and means of dependability
[5]	Jamshidi et al. (2013)	Systematic review	Migration process models	Cloud-RMM and migration types
[6]	Kratzke and Quint (2017)	Systematic mapping	Cloud-native targets	Definition of cloud-native applications
[7]	Balalaie et al. (2016)	Experience report	Migration process models	Incremental microservice migration enabling DevOps
[8]	Jamshidi et al. (2018)	Retrospective	Cloud-native targets	Evolution and open challenges of microservices
[9]	Di Francesco et al. (2019)	Systematic mapping	Cloud-native targets	Mapping of 103 studies on architecting with microservices
[10]	Fritzsch et al. (2019)	Classification	Migration process models	Four categories of decomposition approaches
[11]	Taibi et al. (2017)	Empirical	Migration process models	Motivations and issues reported by 21 practitioners
[12]	Gholami et al. (2017)	Empirical	Migration process models	Challenges of legacy-to-cloud migration
[13]	Opara-Martins et al. (2016)	Empirical	Dependability and risk	Business impact of vendor lock-in
[14]	Shahin et al. (2017)	Systematic review	Delivery pipelines	69 papers and 30 CI/CD approaches
[15]	Hassan et al. (2021)	Survey	Cloud-native targets	Benefits and limits of serverless computing

[16]	Cortez et al. (2017)	Production system	Predictive and autonomic	Workload prediction at 79%–90% accuracy
[17]	Lorido-Botran et al. (2014)	Review	Predictive and autonomic	Taxonomy of auto-scaling techniques
[18]	Beloglazov and Buyya (2012)	Algorithmic and simulation	Predictive and autonomic	Competitive analysis and consolidation heuristics
[19]	Duc et al. (2019)	Survey	Predictive and autonomic	Machine learning for reliable edge-cloud provisioning
[20]	Salfner et al. (2010)	Survey	Dependability and risk	Online failure prediction taxonomy and metrics
[21]	Boehm (1991)	Methodological	Dependability and risk	Risk exposure and risk reduction leverage
[22]	Gunawi et al. (2016)	Empirical	Dependability and risk	Analysis of 597 outages across 32 services
[23]	Dean and Barroso (2013)	Industrial analysis	Dependability and risk	Tail latency amplification and hedged requests
[24]	Kephart and Chess (2003)	Vision	Predictive and autonomic	MAPE-K self-management loop
[25]	Salehie and Tahvildari (2009)	Survey	Predictive and autonomic	Landscape of self-adaptive software
[26]	Paleyes et al. (2022)	Survey of case studies	Predictive and autonomic	Challenges of deploying machine learning

Note: CI/CD = continuous integration and continuous delivery or deployment; MAPE-K = Monitor-Analyze-Plan-Execute over shared Knowledge.

Thematic coding was based on the concept of the systematic mapping study on cloud-native applications, where publications were classified together according to the common themes, and a cloud-native application was defined as an elastic, distributed system that consists of independently deployable units with minimal shared state and is operated on a self-service elastic platform [6]. Accordingly, six clusters were identified: cloud foundations, migration process models, cloud-native and microservice targets, delivery pipelines, dependability and risk and predictive and autonomic management. The quantitative evidence was standardized to common units if possible and the qualitative data was cross-source compared in order to avoid the bias of giving too much weight to any one study.

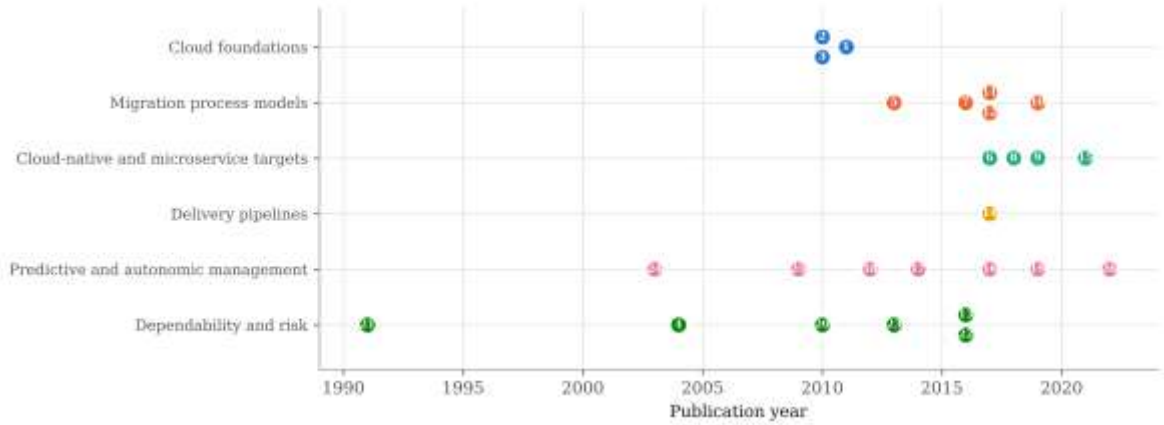

Figure 2. Temporal Distribution of the Evidence Corpus by Thematic Cluster

Note: Marker numbers correspond to reference numbers; vertically offset markers share a publication year.

RESULTS: SYNTHESIS OF EVIDENCE

3.1 Migration Process Models and Architectural Targets

An industrial experience report described the migrations of a mobile backend-as-a-service product and showed that the decomposition into microservices helped to enable DevOps practices: the services were independently deployable, this allowed for continuous delivery, per-service scaling, and autonomy for the teams [7]. It was a gradual migration, and the configuration services, service discovery, load balancing and circuit breakers were added one by one, not in one giant leap. This incremental sequencing is important for risk-aware transformation, as each step represents a limited change with which results can be measured before authorization of the next step in the transformation.

A retrospect of microservices highlighted the service paradigm's evolution from SOA, and pointed to some key considerations that persist, such as finding the right boundaries for services, managing distributed data consistency, and operating a large population of services with sufficient observability [8]. The retro perspective indicates that the paradigm had moved significantly further than the academic consolidation, suggesting that a lot of migration guidance has come from practitioner stories. The evidence of migration effort and the post-migration behavior is still quite limited, further highlighting the need for evidence-based decision support.

A systematic mapping of 103 primary studies on architecting with microservices revealed that research has been active since the paradigm became prominent in 2014 and that the field is at an early stage of maturity, with migration and quality attributes like performance and maintainability being amongst the most discussed ones [9]. The mapping then showed that architectural knowledge is being built up at a faster pace than validated decision models for migration. A summary of the identified process models and decomposition strategies for the entire corpus is compiled and compared in Table 2.

In order to investigate the decomposition of a monolith, existing techniques were classified in four categories: static code analysis aided, metadata aided, workload-data aided, and dynamic microservice composition [10]. Workload-data aided approaches are especially important to PMI since they rely on the measured runtime behavior of the systems call frequencies, execution traces, etc. – instead of only the source structure. The classification also identified that tool support and

validation on industrial scale systems is still limited, limiting confidence in direct use on mission-critical estates.

Table 2. Comparison of Migration Process Models and Decomposition Strategies

Model or strategy	Source	Stages or categories	Primary migration risk addressed
Cloud Reference Migration Model (Cloud-RMM)	[5]	Planning; execution; evaluation; crosscutting concerns	Unstructured or ad hoc migration decisions
Migration types I to IV	[5]	Component replacement; partial migration; whole-stack migration; cloudification	Mismatch between migration scope and risk appetite
Incremental microservice migration	[7]	Staged extraction with discovery, configuration, circuit breaking, and containers	Failure of single, large-scale cut-over
Refactoring approach classification	[10]	Static code analysis; metadata; workload data; dynamic composition	Incorrectly drawn service boundaries
Cloud-native application definition	[6]	Elastic, distributed units with isolated state on a self-service platform	Relocation without realizing elasticity benefits

Source: Synthesized from [5], [6], [7], and [10].

The opinions of practitioners were solicited through an empirical investigation, involving 21 practitioners who had adopted microservices, with maintainability and scalability cited as the two greatest motivations, with the greatest challenges being the decoupling of monolithic data, the communication between services, and estimating migration efforts [11]. The study also revealed that migration was viewed as expensive in the short term, and that the returns would be seen only after operating for some time. This discovery emphasizes the importance of modelling the benefits of migration over a longer time horizon rather than just at the cut-over horizon.

Finally, an empirical study on legacy-to-cloud migration showed that the analysis of legacy dependencies, the compatibility of legacy components with cloud services, the preservation of quality of service and gaps in staff skills are all challenges that arise in the migration process [12]. These challenges were demonstrated to interact: a poor understanding of legacy architecture will make it more difficult to re-engineer and will make it more likely that performance will regress. These interactions support a composite (not single factor) migration risk model, with technical and organizational indicators being weighted together.

Business perspective analysis was done on Vendor lock-in using survey of practitioners, revealing that vendor lock-in would be a constraint for organizations to change vendors after migration, due to lack of interoperability and portability standards and proprietary interfaces and data formats [13]. Lock-in thus turns a seemingly reversible infrastructure choice into a quasi-strategic commitment. For mission critical platforms, the expected exit cost should therefore be added to the migration risk scores used in conjunction with performance and availability as well as the availability of portable abstraction.

3.2 Delivery Pipelines and Target Platforms

They reviewed 69 papers about continuous integration, continuous delivery, and continuous deployment, and identified 30 approaches and tools, and recurrent issues around testing, system design, coordination and tooling [14]. Automated deployment is a key component of risk-aware migration that reduces the time lag between a change and the actual result of that change, and allows for the rollback of the change quickly when the measured behavior differs from the predicted behavior. Figure 3 compares the size of the evidence base for the principal secondary and empirical studies in the corpus, with the outage and mapping studies having the greatest number of studies in support of their respective studies.

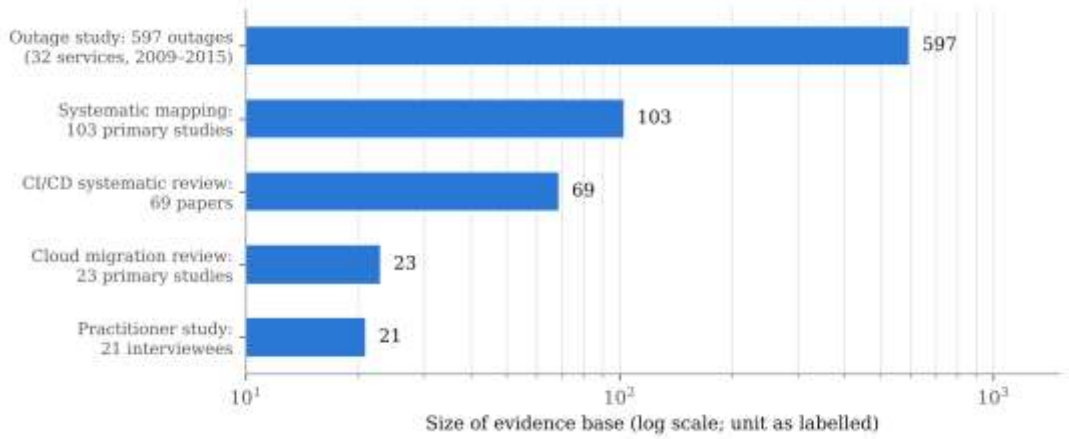

Figure 3. Comparative Size of the Evidence Bases Underpinning Principal Empirical and Secondary Studies

Source: Compiled from [5], [9], [11], [14], and [22]; units differ by study as labelled.

A serverless architecture has been explored as a next-generation paradigm that runs functions on demand, scales automatically, is billed on a fine granular level and has concerns such as cold-start latency, execution time limits, state management, and vendor dependence [15]. The serverless model removes the operational risk of having to plan ahead for capacity in favor of varying latencies and platform coupling for mission-critical migration. As a result, a predictive migration plan is needed to assess not only if the workload can be migrated but also which target abstraction (virtual machines, containers, or functions) has the lowest overall risk.

3.3 Predictive Resource and Failure Management

Production-scale evidence for workload prediction was given by Resource Central, a system deployed on a large public cloud that was trained using past VM telemetry to predict VM attributes for workloads including average and 95th percentile CPU utilization, lifetime, deployment size and workload class, with accuracy varying between 79% and 90% [16]. The predictions were fed to the cluster scheduler via an client library, which enabled utilization-aware oversubscription decisions. This finding demonstrates that the workloads hosted on the cloud are regular enough to be predicted, an essential condition for predicting the demand for resources after migration.

Elastic applications' auto-scaling techniques were categorized under threshold-based rules, reinforcement learning, queuing theory, control theory and time-series analysis, and were presented in the context of a Monitor-Analyze-Plan-Execute process [17]. Time-series and queuing models can be used to make proactive provisioning decisions, but require some calibration work, while

threshold rules are simple and reactive. Table 3 shows a comparison of these families with production workload prediction and online failure prediction, and suggests that proactive techniques are more suitable for the migration windows where demand patterns can change rapidly.

Table 3. Comparative Assessment of Predictive Techniques Relevant to Migration Planning

Technique family	Mode	Typical inputs	Strength	Limitation	Source
Threshold rules	Reactive	CPU load, request rate	Simple and transparent	Lag and oscillation under abrupt shifts	[17]
Queuing theory	Proactive	Arrival and service rates	Analytical latency estimates	Assumes stationary distributions	[17]
Control theory	Reactive or proactive	Deviation from set-point	Formal stability properties	Controller tuning effort	[17]
Reinforcement learning	Adaptive	State, action, reward	Learns without a prior model	Long training and unsafe exploration	[17]
Time-series analysis	Proactive	Historical demand	Anticipates periodic load	Sensitive to regime change	[17]
Tree-ensemble workload prediction	Proactive	VM and subscription history	79%–90% production accuracy	Requires extensive history	[16]
Online failure prediction	Proactive	Logs, symptoms, error reports	Lead time for mitigation	False alarms erode trust	[20]

Source: Synthesized from [16], [17], and [20].

Dynamic consolidation of virtual machines has been analysed using competitive analysis, which resulted in the identification of an optimal online deterministic algorithm with competitive ratio of two for the single migration problem; and adaptive heuristics that were evaluated under the assumption of utilization traces generated by over 1,000 virtual machines in simulation on 800 heterogeneous hosts [18]. The length of a live migration was estimated using Equation (1), which takes into account the number of bytes currently being used by the VM and the bandwidth available in the network, and a loss of roughly 10% of CPU usage was assumed in each migration. Best performance was obtained with the use of local regression for the overload detection and minimal migration time for the selection of virtual machines.

$$T_{\mathrm{m}} = \frac{M_{\mathrm{RAM}}}{B} \tag{1}$$

An overview of machine learning techniques for reliable resource provisioning in edge-cloud systems organized the literature into three main categories: supervised regression, neural time-series models, and reinforcement learning for workload prediction and placement, which are major

families of machine learning techniques for provisioning reliable resources in edge-cloud systems [19]. The survey revealed a need for reliable provisioning, which is tied closely with the quality and representativeness of the training data, and that models trained on one infrastructure may suffer on another. This is directly relevant to migration, as the target environment usually does not have historical telemetry data used for training the predictors.

The taxonomy of online failure prediction surveys the failure tracking, symptom monitoring, detected error reporting, and undetected error auditing, as well as precision, recall, and false positive rate of the prediction, with the lead time, which is the time elapsed between a failure prediction and a real failure, as the assessment metric [20]. These are formalized in Equations (2) and (3) and enable a migration controller to determine if a predictor can provide warning in time for mitigation, such as delaying a migration wave or creating extra replicas. In Table 4, representative quantitative benchmarks for predictive mechanisms are compiled.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

Table 4. Quantitative Benchmarks for Predictive and Performance Mechanisms Extracted from the Corpus

Metric	Reported value	Context	Source
Workload prediction accuracy	79%–90%	Production VM utilization, lifetime, and class prediction	[16]
Consolidation simulation scale	800 hosts	Heterogeneous physical hosts in simulation	[18]
Workload trace scale	More than 1,000 VMs	Real utilization traces at 5-minute intervals over 10 days	[18]
Competitive ratio	2	Optimal online deterministic single-migration algorithm	[18]
Migration-induced degradation	About 10% of CPU	Assumed overhead during live migration	[18]
Fan-out slow-request probability	63%	p = 1% per server, N = 100 servers	[23]
99th-percentile latency	10 ms vs. 140 ms	Single leaf request vs. completion of all leaf requests	[23]
Hedged 99.9th-percentile latency	1,800 ms to 74 ms	1,000-key retrieval with 2% additional requests	[23]
Outage corpus	597 outages, 32 services	Public outage reports, 2009 to 2015	[22]

Source: Values extracted from [16], [18], [22], and [23].

3.4 Risk Quantification and Dependability Evidence

Software risk management has been formalized using software risk exposure (Equation (4)) and software risk reduction leverage (Equation (5)) [21]. Also proposed were 10 of the most important software risk items, such as unrealistic schedules and budgets, shortfalls in externally furnished components, and real-time performance shortfalls. Table 5 is an example of how these

constructs may be applied in a migration risk register, and Figure 4 shows the exposure surface before and after mitigation.

$$\mathrm{RE} = P(\mathrm{UO}) \times L(\mathrm{UO}) \quad (4)$$

$$\mathrm{RRL} = \frac{\mathrm{RE}_{\mathrm{before}} - \mathrm{RE}_{\mathrm{after}}}{C_{\mathrm{RR}}} \quad (5)$$

Table 5. Illustrative Migration Risk Register Evaluated with Risk Exposure and Risk Reduction Leverage

ID	Migration risk	P	L	RE	Mitigation	Cost	RE after	RRL
R1	Data inconsistency during cut-over	0.30	9	2.70	Dual-write with reconciliation	1.0	0.90	1.80
R2	Tail-latency regression after decomposition	0.40	6	2.40	Hedged requests and fan-out limits	0.5	0.90	3.00
R3	Upgrade-induced outage	0.25	8	2.00	Canary waves with automated rollback	0.8	0.64	1.70
R4	Provider lock-in exit cost	0.50	5	2.50	Portable abstractions and data formats	1.5	1.50	0.67
R5	Predictor drift in target environment	0.35	4	1.40	Drift monitoring and retraining	0.4	0.60	2.00

Note: Illustrative values constructed to demonstrate Equations (4) and (5); P, L, and cost are expressed in normalized units and are not empirical measurements. Risk categories are informed by [11], [13], [22], and [23].

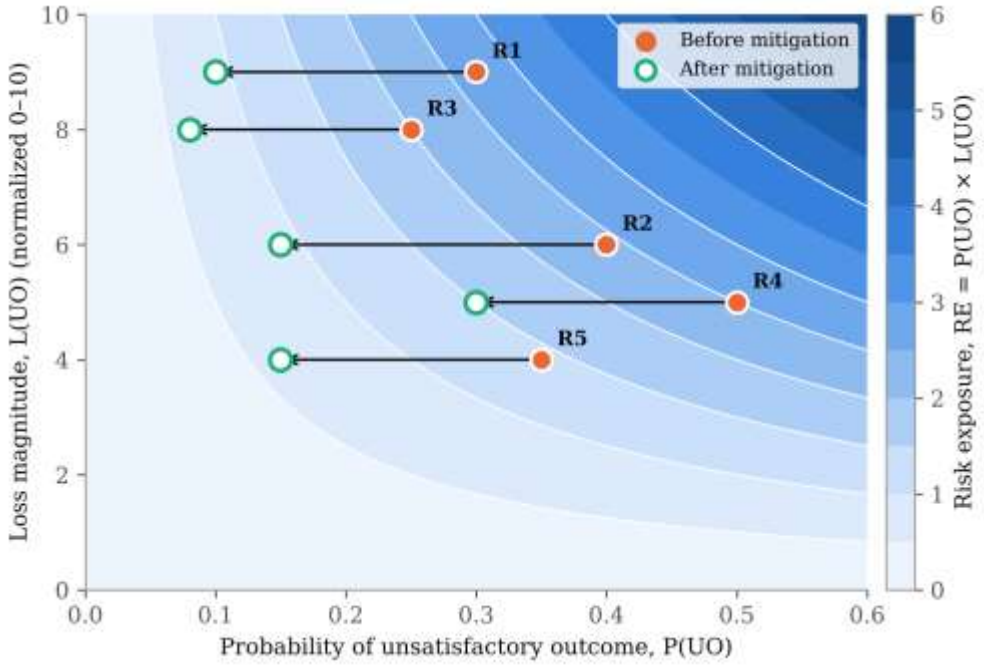

Figure 4. Risk Exposure Surface with Illustrative Migration Risks Before and After Mitigation

Note: Contours denote iso-exposure levels; positions correspond to the illustrative values in Table 5.

Data on failure causes was gathered from a study of 597 unplanned outages over a six-year period (2009-2015) that examined the root causes, impacts, durations and recovery procedures for failures from public post-mortem reports and news coverage of failures of 32 popular Internet services [22]. The most common issues identified were upgrades, network problems, software bugs, misconfiguration, and traffic load, with cross-service dependencies being demonstrated that could cause localized issues to propagate. These categories can be used as leading indicators in a predictive migration risk model, as migration is an inherent process of upgrading, reconfiguration and new dependency chains.

This "performance risk at scale" was observed: If a request is sent to N servers, each of which has a slow-response probability of p, then the overall slow-request probability is given by Equation (6), meaning that a 1% slow-response probability per server corresponds to a 63% slow-response probability for the request at 100 servers [23]. The measurements performed on a production service yielded 99th percentile latencies of 10ms for a single leaf request, while it took 140ms to complete all the leaf requests. As seen in Figure 5 and Figure 6, hedged requests reduced the 99.9th percentile latency for retrieval of 1000 keys from 1800ms to 74ms, while only increasing the number of requests by 2%.

$$P_{\text{slow}} = 1 - (1 - p)^N \tag{6}$$

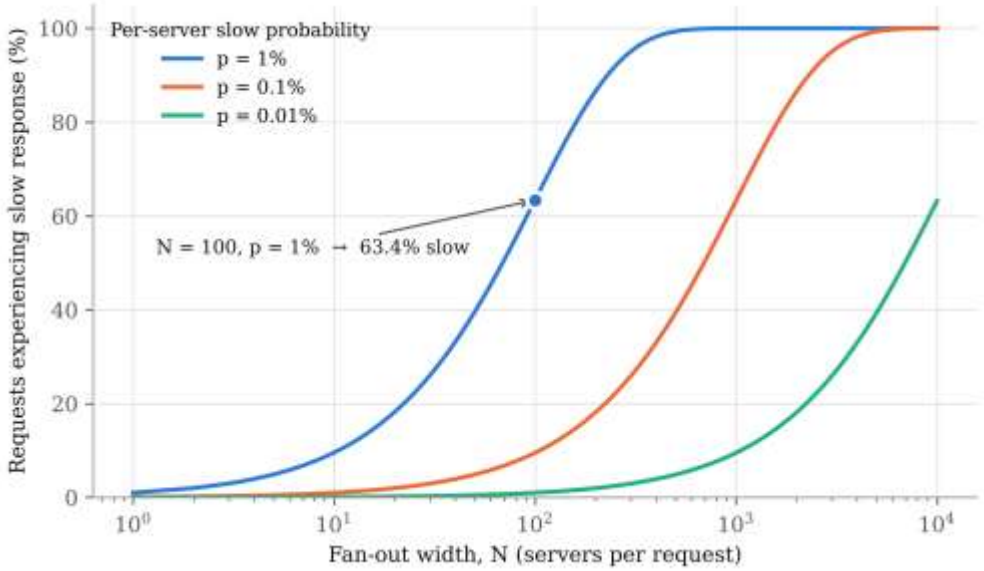

Figure 5. Amplification of Slow Responses with Fan-Out Width for Three Per-Server Slow-Response Probabilities

Note: Curves computed from Equation (6); the highlighted point reproduces the case reported in [23].

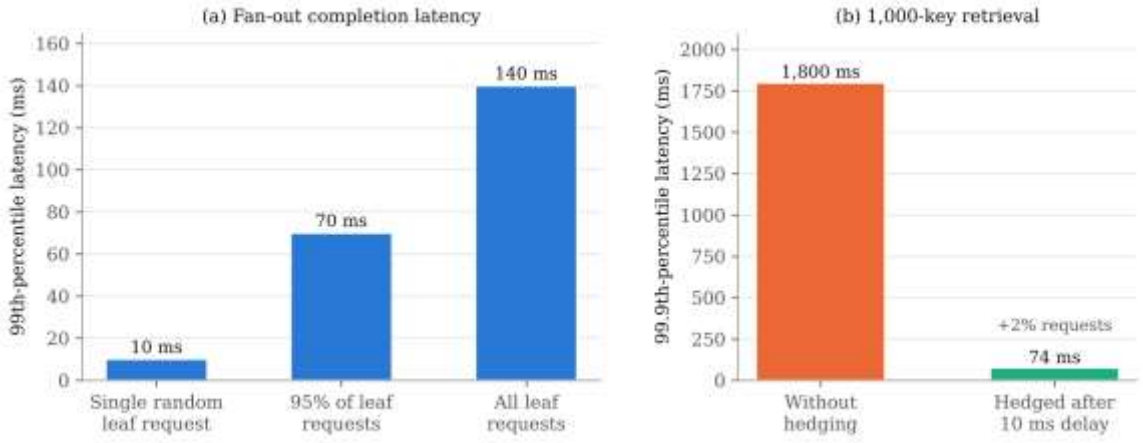

Figure 6. Measured Fan-Out Completion Latency and the Effect of Hedged Requests on Extreme Tail Latency

Source: Adapted from measurements reported in [23].

DISCUSSION: AN INTEGRATED PREDICTIVE MIGRATION INTELLIGENCE FRAMEWORK

4.1 Framework Architecture

The synthesized evidence is incorporated in autonomic computing vision in which autonomic systems achieve self-configuration, self-optimization, self-healing and self-protection via a Monitor-Analyze-Plan-Execute loop on shared Knowledge (MAPE-K) [24]. As illustrated in Figure 7, the proposed framework consists of four stages: (1) Monitor, (2) Analyze, (3) Plan, and (4) Execute.The proposed framework comprises four stages: (1) Monitor, (2) Analyze, (3) Plan, and (4) Execute, as depicted in Figure 7. The relationship between the layers of the framework and the supporting evidence is shown in Table 6.

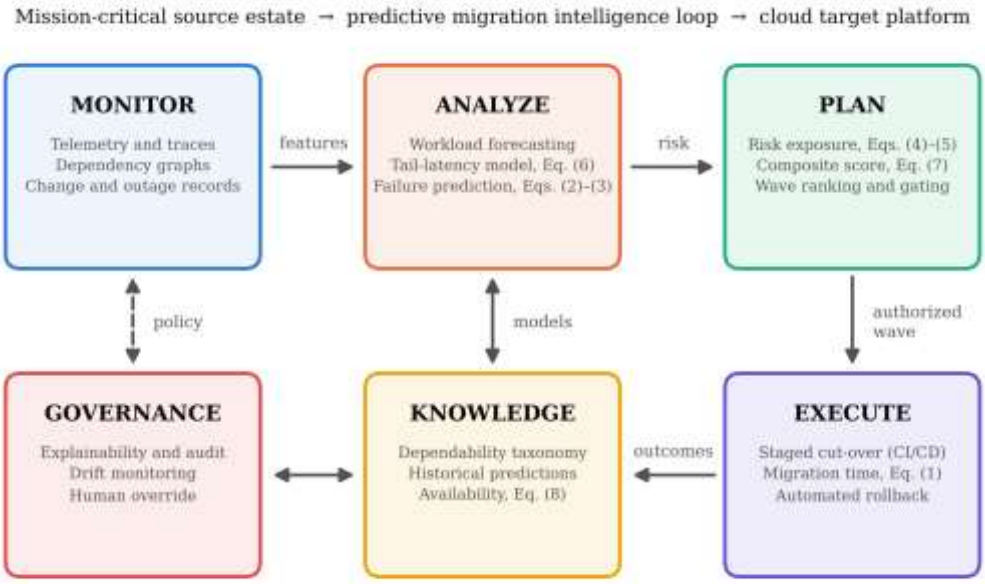

Figure 7. Layered Architecture of the Proposed Predictive Migration Intelligence Framework Based on the MAPE-K Loop

Note: Equation numbers indicate where each quantitative model is applied within the loop.

Table 6. Mapping of Framework Layers to Techniques, Metrics, and Supporting Evidence

Layer	Function	Techniques	Indicative metric	Sources
Monitor	Capture telemetry, dependencies, and changes	Symptom monitoring; outage root-cause taxonomy	Instrumentation coverage	[20], [22]
Analyze	Forecast demand, latency, and failure	Tree-ensemble prediction; time-series models; tail model	Accuracy; precision; recall; slow-request probability	[16], [17], [23]
Plan	Rank and sequence migration waves	Risk exposure; risk reduction leverage; composite score	RE; RRL; composite risk score	[5], [21]
Execute	Staged cut-over with rollback	CI/CD; incremental extraction; consolidation heuristics	Migration time; rollback time	[7], [14], [18]
Knowledge	Retain models, policies, and history	Dependability taxonomy; self-adaptive models	Availability	[4], [24], [25]
Governance	Assure predictive components	Explainability; drift monitoring; audit trails	Drift alarms; audit completeness	[13], [26]

Source: Synthesized by mapping the corpus onto the MAPE-K loop described in [24].

Regarding explicit models of goals and uncertainty, as well as assuring the adaptation loop, self-adaptive software research has focused on these themes [25]. To that end, a composite migration risk score for wave i is defined in Equation (7) and availability is monitored in Equation (8): a weighted sum of the normalized risk dimensions, such as availability, tail latency, data integrity, lock-in, and organizational readiness. Progress is not allowed based on when the schedule will be met, but only when the composite score is below a threshold which is determined by the criticality of the platform.

$$R_i = \sum_{k=1}^{K} w_k r_{ik}\,,\ \sum_{k=1}^{K} w_k = 1 \tag{7}$$

$$A = \frac{\text{MTTF}}{\text{MTTF+MTTR}} \tag{8}$$

4.2 Barriers, Ethical Considerations, and Regulatory Implications

The implementation of the predictive elements of this framework was faced with the difficulties reported in a survey of machine learning deployment case studies covering data management, model learning and verification, and deployment, as well as crosscutting issues of ethics, law, end user trust and security [26]. Predictions for regulated mission-critical platforms: gate migrations need to be explainable and auditable, training data need to meet privacy and residency requirements, and after cut-over, monitoring for model drift needs to be implemented. Table 7 is a summary of these barriers and mitigations, and shows governance as an inherent component of PMI.

Table 7. Deployment Barriers for Predictive Components and Corresponding Mitigations

Concern	Barrier	Mitigation within PMI	Ethical or regulatory implication
Data management	Sparse telemetry in the target environment	Transfer of source-side models with conservative priors	Residency and privacy of operational telemetry
Model learning	Non-stationary workloads after migration	Drift detection and scheduled retraining	Documented model lineage
Model verification	Unverified predictions gating critical waves	Shadow-mode validation before gating authority	Accountability for automated decisions
Model deployment	Integration of predictors into pipelines	Predictions exposed as pipeline quality gates	Change-management audit trails
Ethics and law	Opaque risk scores	Interpretable weighted scores, Equation (7)	Explainability obligations in regulated sectors
End-user trust	False alarms delaying migration waves	Precision thresholds and human override	Human oversight of critical decisions
Security	Telemetry exposure and model tampering	Access control and integrity verification	Confidentiality and integrity duties

Source: Synthesized from the challenge categories in [26], with dependability attributes from [4] and prediction metrics from [20].

CONCLUSION

This systemic review brought together evidence from 26 sources and formed a unified view of migration intelligence, the effective tool for the risk-aware transformation of mission critical cloud platforms. The synthesis illustrated that the constituent capabilities have evolved significantly independently: We were able to predict workload behavior at production scale with precision between 79% and 90%, characterized failure predictors using precision, recall, and lead time, modelled the tail latency analytically, and hedged it at a 2% request overhead, and formalised risk exposure as the product of probability and loss. The main contribution is in embedding these capabilities in a MAPE-K framework where each migration wave is gated by a composite risk score, which is based on explicit equations, comparative tables, and an evidence-to-layer mapping. There were also persistent limitations identified such as the lack of industrial validation of decomposition techniques, lack of historical telemetry in the target environments, and the governance requirements of predictive models. Empirical validation of composite risk scoring on live migrations, transfer learning for telemetry-sparse targets, and auditable pipelines for predicting, for regulated sectors, should be a priority for future work.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, 2011. doi: 10.6028/NIST.SP.800-145.

[2] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010, doi:

10.1145/1721654.1721672.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, 2010, doi: 10.1007/s13174-010-0007-6.

[4] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 11–33, 2004, doi: 10.1109/TDSC.2004.2.

[5] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 142–157, 2013, doi: 10.1109/TCC.2013.10.

[6] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing: A systematic mapping study," *J. Syst. Softw.*, vol. 126, pp. 1–16, 2017, doi: 10.1016/j.jss.2017.01.001.

[7] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables DevOps: migration to a Cloud-Native architecture," *IEEE Softw.*, vol. 33, no. 3, pp. 42–52, 2016, doi: 10.1109/ms.2016.64.

[8] P. Jamshidi, C. Pahl, N. C. Mendonca, J. Lewis, and S. Tilkov, "Microservices: the journey so far and challenges ahead," *IEEE Softw.*, vol. 35, no. 3, pp. 24–35, 2018, doi: 10.1109/ms.2018.2141039.

[9] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *J. Syst. Softw.*, vol. 150, pp. 77–97, 2019, doi: 10.1016/j.jss.2019.01.001.

[10] J. Fritzsch, J. Bogner, A. Zimmermann, and S. Wagner, "From monolith to microservices: A classification of refactoring approaches," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, 2019, pp. 128–141. doi: 10.1007/978-3-030-06019-0_10.

[11] D. Taibi, V. Lenarduzzi, and C. Pahl, "Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 22–32, 2017, doi: 10.1109/MCC.2017.4250931.

[12] M. F. Gholami, F. Daneshgar, G. Beydoun, and F. Rabhi, "Challenges in migrating legacy software systems to the cloud: An empirical study," *Inf. Syst.*, vol. 67, pp. 100–113, 2017, doi: 10.1016/j.is.2017.03.008.

[13] J. Opara-Martins, R. Sahandi, and F. Tian, "Critical analysis of vendor lock-in and its impact on cloud computing migration: A business perspective," *J. Cloud Comput.*, vol. 5, no. 4, 2016, doi: 10.1186/s13677-016-0054-z.

[14] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017, doi: 10.1109/ACCESS.2017.2685629.

[15] H. B. Hassan, S. A. Barakat, and Q. I. Sarhan, "Survey on serverless computing," *J. Cloud Comput.*, vol. 10, no. 39, 2021, doi: 10.1186/s13677-021-00253-7.

[16] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource Central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 153–167. doi: 10.1145/3132747.3132772.

[17] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, 2014, doi: 10.1007/s10723-014-9314-7.

[18] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012, doi: 10.1002/cpe.1867.

[19] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, 2019, doi: 10.1145/3341145.

[20] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surv.*, vol. 42, no. 3, 2010, doi: 10.1145/1670679.1670680.

[21] B. W. Boehm, "Software risk management: Principles and practices," *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, 1991, doi: 10.1109/52.62930.

[22] H. S. Gunawi, "Why does the cloud stop computing? Lessons from hundreds of service outages," in *Proc. 7th ACM Symp. Cloud Comput. (SoCC)*, 2016. doi: 10.1145/2987550.2987583.

[23] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, 2013, doi: 10.1145/2408776.2408794.

[24] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer (Long. Beach. Calif).*, vol. 36, no. 1, pp. 41–50, 2003, doi: 10.1109/MC.2003.1160055.

[25] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 2, 2009, doi: 10.1145/1516533.1516538.

[26] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," *ACM Comput. Surv.*, vol. 55, no. 6, 2022, doi: 10.1145/3533378.