ENTERPRISE INFORMATION SYSTEMS

Right-Sizing the Cloud: Predicting Compute and Storage Needs for SAP HANA Workloads to Cut Cloud Costs in the Kingdom

Muhammad Omar Muftakhar

Published November 2026

Abstract

When Saudi organisations move SAP systems to the cloud, the in-memory database usually becomes the largest single item on the infrastructure bill. Capacity is often bought for the size the system is expected to reach years later, so much of it sits idle for most of the contract. This paper asks how compute, memory and storage needs for SAP HANA workloads can be predicted well enough to provision closer to real demand without endangering performance. Using a design science approach, it reviews thirty journal articles (2020 to 2025, all refereed) on in-memory database behaviour, workload forecasting, autoscaling and cloud cost optimisation, and reads them against four documentary sources on vendor sizing practice and national cloud rules. The result is a five-layer right-sizing framework: a vendor-method baseline, continuous telemetry, a forecasting layer that reports uncertainty, a decision layer that converts forecasts into instance, commitment and storage-tier choices, and a governance loop. An invented three-year example indicates that stepping memory up as a forecast band requires, instead of provisioning for the final year from the start, would use about one sixth fewer terabyte-months. Evidence is supportive for machine-learning forecasts of processor and general resource use, mixed for cost savings outside laboratory traces and limited for memory-bound enterprise databases. All material came from published or official sources, and nothing was drawn from primary or proprietary data.

1 Introduction

SAP HANA keeps its working data in main memory. That design gives the speed that modern ERP and analytics depend on, but it also ties the cost of the database closely to the amount of memory provisioned. On cloud infrastructure, where memory-heavy instances carry premium prices and are often bought on multi-year commitments, a sizing error is expensive in both directions. Too little memory and the system slows, unloads data or fails; too much and the organisation pays for capacity it does not use.

The choice of size is usually made once, early, and with limited information. Vendor guidance requires a formal memory sizing before deployment, through a sizing tool for new implementations or sizing reports and notes for migrations, and calls for expert sizing in large or complex cases (Table 1, D3, D4). A common rule reserves as much memory for dynamic objects, such as query intermediates and loading, as for the compressed table data, which doubles the static estimate (D4). Teams then add headroom for growth over the contract and round up to the nearest instance size. The result is safe but rarely lean.

Saudi organisations have particular reasons to look more closely. National policy makes cloud the default for government IT investment and expects lower total cost of ownership from it (Table 1, D1), while localisation questions are referred to the national data office (D2), which can narrow the choice

of regions and instance families. Yet published work rarely asks how capacity for large in-memory business systems should be forecast and bought. Research on cloud workload prediction is abundant, but it mostly concerns processor use in web services, virtual machine fleets and container platforms.

Two studies anchor this paper. Myung and Choi (2021) built a machine-learning model that, given the workload type and input size, predicted the efficient memory allocation for data-parallel jobs in a big data engine; predicted and actual requirements were closely aligned, and memory efficiency improved by up to 1.89 times over default settings. Bach et al. (2022) described the testing of SAP HANA, a database with millions of lines of code, and argued that very large database systems need adapted testing approaches with challenges of their own. Together they show that memory needs can be learned from workload features, and that HANA is a complex, evolving system whose behaviour should not be assumed constant.

The aim is to design a prediction-led framework for right-sizing compute, memory and storage for SAP HANA workloads in the cloud, suitable for organisations in the Kingdom. The objectives are to identify the demand drivers and constraints specific to in-memory databases, to compare forecasting and provisioning methods in the literature, to assemble them into a decision framework and to illustrate its effect with an invented example. Three research questions follow. RQ1: what drives and constrains the compute, memory and storage needs of HANA workloads in the cloud? RQ2: which prediction and provisioning methods are suitable, and how strong is the evidence for them? RQ3: how can these methods be combined into a right-sizing process that lowers cost without risking service?

After the background on memory-bound databases and cloud pricing, the method is described. Findings for the first two questions come next, followed by the framework and its illustration, a discussion and a conclusion.

2 Background

2.1 Why in-memory databases are hard to size

In-memory systems behave differently from disk-based ones. Their performance depends on how efficiently processors reach data in memory, and studies of memory-resident databases show that access latency, cache misses and data layout shape run times in ways that general-purpose data structures do not always handle well (Qian et al., 2021). Processor choice matters in less obvious ways too. Benchmarking operators across processor families, Liu et al. (2023) found that cache architecture, rather than core count or cache size alone, often determined performance, and that operator speed on one processor was hard to predict from results on another. For capacity planning this means that compute needs cannot simply be scaled from the number of virtual cores.

Memory demand also has structure that sizing can exploit. Requests in in-memory stores tend to concentrate on a small fraction of objects, which allows hot data to be kept in the fastest memory and colder data moved to slower, cheaper tiers (Zhao et al., 2021). Enterprise systems show a similar pattern over time, since recent transactions are read far more often than records from previous years. If cold data can be placed outside main memory, the memory footprint grows more slowly than the database as a whole.

Finally, the database is not static. Data volumes grow with transactions, new modules and new users; archiving shrinks them; upgrades can change memory consumption. The testing burden described by

Bach et al. (2022) for a system of this scale underlines that behaviour may differ between releases. Sizing therefore cannot be a one-off exercise if the aim is to track real demand.

Availability design multiplies every sizing decision. Production HANA systems are commonly paired with a standby copy, in the same region or another, and the standby must be large enough to take over the full workload. Each step in production memory is then paid for at least twice, and often three times when a further copy is kept for disaster recovery. Non-production systems add more: development, quality assurance and sandbox copies are frequently sized like production, even though they run only during working hours or for limited projects.

2.2 Cloud pricing and the cost of headroom

Cloud providers sell compute in fixed instance shapes and under several contract types: on-demand, reserved or committed capacity and revocable transient capacity. Cost depends on the mix. Mandal et al. (2023) developed reservation algorithms that bound the cost of combining reserved and on-demand instances against the optimum, even when future spot prices are unknown. Chen et al. (2024) showed that a base contract sized for stationary demand, supplemented by shorter or cancellable reservations for surges, can be optimal, with the best policy depending on surge duration and cancellation fees. Prices also move: price-prediction work for reserved instances found that machine learning captured the evolution of prices, though older data did not improve accuracy (Fragiadakis et al., 2023).

Storage adds its own pricing complexity. A survey of cloud storage costs set out a taxonomy that includes not only capacity charges but also network, operation and retrieval costs, and showed how trade-offs between storage, computation, caching and networking complicate selection (Khan et al., 2024). Fragiadakis et al. (2024) proposed a decision support framework that brings hedonic, time-series, clustering and machine-learning methods together to compare pricing policies across services and providers. For a HANA estate, which needs persistence volumes, log volumes, backups and often replicas, storage choices can add materially to the bill.

Haubenschild and Leis (2025) offer the closest analytical treatment of database cost in the cloud. Their model framework specifies workloads by data size, performance, latency, availability and durability, evaluates designs by their processor, memory, storage and network needs and identifies the cost-optimal architecture and hardware configuration. Although calibrated on a different storage engine and not on HANA, it shows that cost-optimal provisioning can be derived from explicit workload parameters rather than from rules of thumb.

2.3 Policy and vendor guidance

Table 1 lists the four documents consulted. The Cloud First Policy asks civilian entities to choose cloud for new IT investment, expects savings in total cost of ownership and defines the order in which service models are preferred (D1). In their 2024 edition, the cloud cybersecurity controls apply to government tenants and their providers, and its 2025 update moved localisation requirements to the national data office, which entities must consult before acting (D2). Where data must stay in the Kingdom, the set of available regions, and therefore instance families and prices, may be narrower than in global catalogues. The two vendor documents describe the sizing methods that any cloud deployment of HANA starts from (D3, D4).

Table 1. Documentary sources consulted (not counted among the thirty reviewed articles)

Code	Document	Issuer and year	Provisions relevant to this study
D1	Cloud First Policy	Ministry of Communications and Information Technology, 2020	Cloud is the default for new IT investment by civilian entities; savings of about 30% in total cost of ownership are described as typical for infrastructure migration; software services preferred over platform and infrastructure services
D2	Cloud Cybersecurity Controls (CCC-2:2024)	National Cybersecurity Authority, 2024 (updated 2025)	Applies to government tenants and their providers; subcontrols requiring in-Kingdom provision of storage, processing, recovery, monitoring and support were deleted and localisation referred to the National Data Management Office
D3	SAP HANA Master Guide, document version 1.0	SAP SE, 2024	Memory sizing is required for every deployment: a sizing tool for new implementations, sizing reports or notes for migrations, and expert sizing for large or complex systems
D4	Performing SAP HANA sizing, training unit	SAP SE (accessed November 2026)	Sizing revolves around main memory; compression depends on scenario; static memory for table data plus an equal reserve for dynamic objects, so total memory is twice the static requirement; expert sizing for consolidation and high-volume migrations

3 Method

3.1 Design science framing

The study seeks a useful artefact, a right-sizing framework, rather than a test of theory, so it follows a design science logic. The problem was defined from vendor sizing practice and the cost pressures described above; objectives for a solution were set as accuracy with stated uncertainty, protection of service levels, cost transparency and suitability for memory-bound databases; the framework was designed from the evidence; and its behaviour was demonstrated with an invented example. Evaluation at this stage is analytical: it checks whether each layer is grounded in evidence, whether the framework covers compute, memory and storage, and whether it respects the constraints of a stateful database.

3.2 Sources and selection

Literature was gathered during November 2026 from a freely accessible bibliographic database, with identifiers and publication details then confirmed in registration records supplied by publishers. Search strings paired in-memory database, SAP HANA, memory prediction, workload forecasting, resource usage prediction, autoscaling, reserved instances, cloud pricing, storage cost and cost optimisation with terms for cloud and enterprise systems. To be included, a study had to be a refereed journal article published from 2020 to 2025 that addressed prediction, provisioning or cost of computing resources in a way transferable to databases. Conference papers, preprints and records whose identifier would not resolve were left out. Thirty articles were retained, with the two anchor studies (Myung & Choi, 2021; Bach et al., 2022) chosen first for their relevance to memory prediction and to HANA itself.

3.3 Analysis

Coding was done by the author in a single pass with a second reading of every article that informed a framework layer. Studies were coded by resource predicted, method family, horizon, data source, evaluation metric and reported effect on cost or service. Methods were grouped into families, and each family was assessed for what it contributes to sizing a stateful in-memory database. Evidence was rated supportive where several studies with real traces agreed, mixed where results depended on setting or were shown only in simulation, and limited where the topic had little direct coverage. The illustrative example uses invented values throughout, and the study involved neither primary data collection nor access to any organisation's systems.

4 Findings

4.1 Drivers and constraints of HANA capacity (RQ1)

Three drivers stand out. Memory is the first and dominant one. Vendor sizing centres on main memory, derives it from the source database or from business volumes and doubles static needs to cover dynamic use (Table 1, D4). Growth in transaction data, new functions and analytics loads then push memory up over time, while archiving and moving cold data out of memory push it down. Memory-specific prediction can work: Myung and Choi (2021) learned efficient memory from workload type and input size, and Wang et al. (2020) used a stacked ensemble with a probabilistic second layer to forecast memory demand for complex simulations in the cloud, outperforming single models and quantifying uncertainty.

Compute is the second driver, but it is harder to transfer between instance types. Processor architecture affects operator performance in ways that core counts do not capture (Liu et al., 2023), and memory access patterns limit how far added cores help (Qian et al., 2021). Performance prediction for black-box virtual machines is possible by identifying the application and its most informative runtime metrics (Huang et al., 2024), which suggests that measured, workload-specific benchmarks are safer than vendor-neutral conversions when changing instance families.

Storage is the third. Persistence, logs, backups and replicas each have different performance and cost profiles, and the full storage bill includes network and operation charges (Khan et al., 2024). Hot-cold separation reduces both memory and fast storage needs (Zhao et al., 2021). Storage is, however, the easiest of the three to forecast. Data and log volumes grow broadly with business activity, backup volumes follow retention policy, and none of them needs to be bought in large fixed steps, since most cloud storage can be extended online. Where storage errors arise, they usually come from retention rules that keep too many copies in expensive tiers, not from poor prediction.

The main constraint is statefulness. Most autoscaling research assumes that capacity can be added or removed quickly by starting or stopping instances. Quattrocchi et al. (2024) compared the native autoscalers of three major providers with control-theoretic and queueing-based alternatives and found the latter better at balancing service violations and resource use, but all were designed for replicated, mainly stateless services. Chouliaras and Sotiriadis (2023) did scale a relational database, classifying predicted demand and adjusting containers in real time, and reported a 48% reduction in deployment cost when one tenant ran alone. A large HANA instance cannot be resized as freely: moving to a different memory size normally means a planned change of instance with downtime or a failover. Right-sizing must therefore work on a horizon of weeks or months, not minutes.

Non-production systems are the exception, and a significant one. Development, test and training systems often run on schedules that leave them idle at night and at weekends, and they tolerate restarts. For these systems, scheduled shutdown, smaller shapes and on-demand pricing are practical, and the performance-cost search methods described below apply more directly. Separating the estate into a stateful production core and a flexible periphery is therefore the first step in any right-sizing effort.

4.2 Prediction and provisioning methods (RQ2)

Five method families are relevant (Table 2). Statistical and filtering methods provide transparent baselines. A Kalman filter predicted cloud server indicators with high accuracy and was proposed for anticipating resource exhaustion before failures occur (Gyeera et al., 2023). Deep sequence models dominate recent work on processor and workload forecasting. Hybrid recurrent networks combining bidirectional, stacked and gated units outperformed statistical models for virtual machine processor use (Karim et al., 2021); a multivariate bidirectional model reduced prediction error compared with univariate and other deep models and made autoscaling more efficient (Dang-Quang & Yoo, 2022); and a gated recurrent design with sliding-window preparation lowered errors on production traces and cut the number of active hosts needed (Xu et al., 2022). Ensembles and generative models go further. Particle swarm optimisation weighted three recurrent learners into a meta-learner (Leka et al., 2023), deep belief networks with swarm-tuned parameters improved processor forecasts for physical hosts (Wen et al., 2020), and hybrid generative adversarial models predicted both workload values and their direction (Maiyza et al., 2023).

A comparative study implemented the main families on a shared platform with three benchmark traces and showed that no family wins everywhere, with trade-offs between accuracy, training cost and robustness (Saxena et al., 2023). That finding supports the third family, adaptive selection. Nawrocki et al. (2023) let the system choose among forecasting algorithms according to load characteristics, improving prediction quality by 9.28% to 80.68% over a fixed choice on production servers. Earlier work combined anomaly detection with machine learning to configure resources within quality-of-service limits, reporting cost reductions of 51% to 85% in tests (Nawrocki & Osypanka, 2021), and the same group showed that filtering anomalies improved long-term reservation planning for over 1,700 virtual machines (Nawrocki & Sus, 2022) and that a six-stage process with self-adaptation produced savings of 31% to 89% depending on scenario (Osypanka & Nawrocki, 2023).

The fourth family quantifies uncertainty. Bayesian deep learning models that predict several resources jointly and report confidence improved results on service-level metrics, and transfer learning helped them adapt to new data centres (Rossi et al., 2025). For a database that cannot be resized at short notice, a forecast band is more useful than a point forecast, because the provisioning decision turns on the upper range of plausible demand.

The fifth family converts forecasts into purchase decisions. Reservation algorithms (Mandal et al., 2023), base-plus-surge contract policies (Chen et al., 2024) and risk-aware portfolios that mix transient and standard capacity (Chhetri et al., 2023) show how predicted demand can lower cost. Transient capacity, however, suits stateless or restartable tiers, not the primary database. Service quality must be kept in view throughout. Nawrocki and Osypanka (2021) found that cost optimisation often degrades responsiveness unless quality-of-service limits are built into the configuration step, and their approach, tested in a commercial cloud environment, respected such limits while still lowering cost. For HANA, the equivalent limits are response times for critical transactions and the completion of batch windows. Performance-cost trade-offs can also be searched systematically: modelling

autoscaling with stochastic Petri nets and a metaheuristic exposed configurations that met response-time targets at lower cost (Fé et al., 2022).

Table 2. Method families for predicting and provisioning HANA capacity

Family	Role in the framework	Limits for HANA	Key sources	Evidence
Statistical and filtering baselines	Transparent first forecast; early warning of exhaustion	Weak with abrupt changes	Gyeera et al. (2023)	Mixed
Deep sequence and ensemble models	Accurate forecasts once history exists	Mostly processor traces of virtual machine fleets	Karim et al. (2021); Xu et al. (2022); Leka et al. (2023)	Supportive
Adaptive selection with anomaly filtering	Chooses method per load pattern; cleaner long-term plans	Few database-specific tests	Nawrocki et al. (2023); Nawrocki & Sus (2022)	Mixed
Probabilistic and memory-specific models	Forecast bands for resize decisions	Little evidence for enterprise databases	Rossi et al. (2025); Wang et al. (2020); Myung & Choi (2021)	Limited
Commitment and pricing models	Convert forecasts into reservations and tiers	Transient capacity unsuitable for primary database	Chen et al. (2024); Mandal et al. (2023); Chhetri et al. (2023)	Mixed

4.3 Strength of evidence

The evidence is supportive for machine-learning forecasts of processor and general resource use, which are tested on several public and production traces. It is mixed for cost savings. Large reported reductions come from particular systems, simulations or test periods and depend on how wasteful the baseline was. It is limited where the target is an enterprise database held in memory: only a few studies forecast memory, none uses HANA workloads and none addresses the planned downtime that resizing a large in-memory instance involves. Peer-reviewed studies have not yet evaluated the vendor rules themselves.

5 A right-sizing framework for HANA workloads

5.1 Five layers

The framework has five layers (Figure 1). The baseline layer establishes the starting size with the vendor method: a sizing tool or sizing report, a doubling of static memory for dynamic use and expert review where systems are consolidated or very large (Table 1, D3, D4). The baseline is a starting point, not a three-year commitment. The telemetry layer then collects, from the first day of operation, used and peak memory by component, table growth by module, processor utilisation and wait times, storage growth for data, log and backup volumes, and business drivers such as document counts and active users.

The forecasting layer produces monthly forecasts of memory, compute and storage with uncertainty bands over a horizon at least as long as the lead time for a resize. It starts with a transparent baseline such as a growth or filtering model, adds learned models once enough history exists and uses adaptive selection to choose among them (Nawrocki et al., 2023). Anomalies such as one-off data

loads are filtered before training (Nawrocki & Sus, 2022). Forecasts use business drivers as inputs where possible, since memory follows data volume more closely than time alone.

The decision layer turns forecasts into three kinds of choice. For instance size, the rule is to move up when the upper band for the coming lead-time window approaches a set fraction of current capacity, and to consider moving down after archiving or tiering frees memory. For commitments, the base level that forecasts show will persist is covered by reserved or committed capacity, while non-production systems and short peaks use on-demand capacity (Chen et al., 2024; Mandal et al., 2023). For storage, cold data and older backups move to cheaper tiers, with retrieval costs counted (Khan et al., 2024). Each layer has a measure. For forecasts, the share of months in which actual demand stayed inside the band matters more than average error, because a tight band that is often breached is unsafe. For decisions, the measures are average utilisation of provisioned memory, the number of emergency resizes and cost per business transaction. The governance layer reviews forecast accuracy, cost and service levels each quarter, adjusts thresholds and records every resize decision with its evidence.

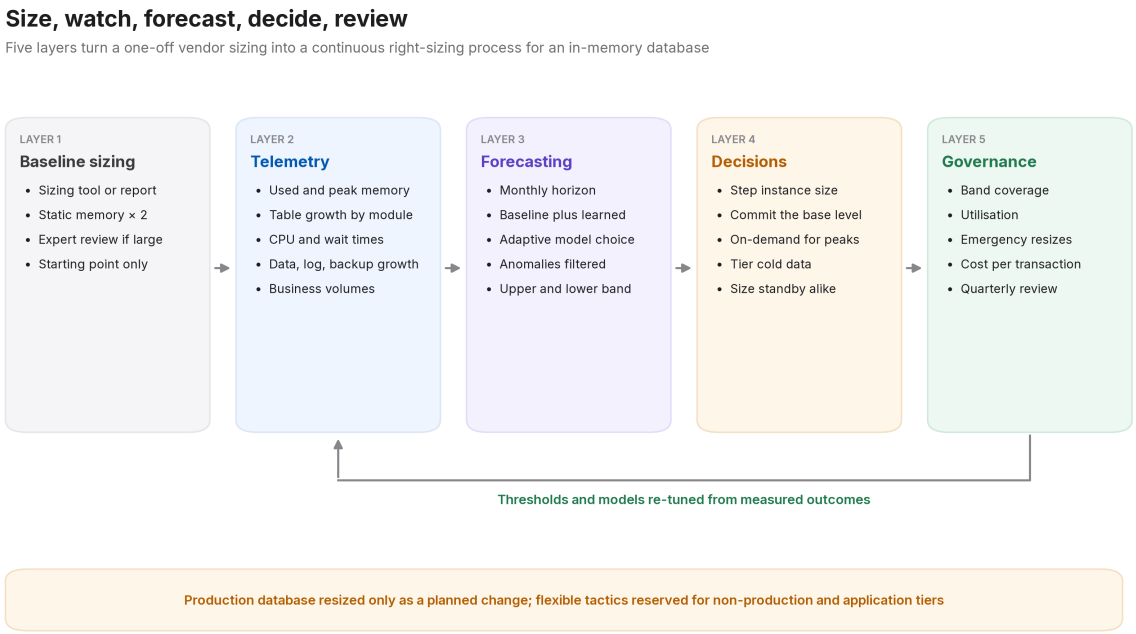

Figure 1. Five-layer right-sizing framework for SAP HANA workloads in the cloud

5.2 Design rules

Four rules keep the framework safe. Protect the primary database first: aggressive techniques such as transient capacity or rapid autoscaling belong in application servers, test systems and analytics sandboxes, not in the production database. Plan resizes as change events, scheduled in maintenance windows and rehearsed on a copy, because a memory change in a large in-memory system is not instantaneous. Prefer bands to points, since the cost of running short is higher than the cost of modest headroom. And benchmark before changing instance families, because processor architecture can alter performance unexpectedly (Liu et al., 2023; Huang et al., 2024).

Two Saudi-specific practices complement these rules. Confirm with the national data office which regions are acceptable for each data class before modelling prices, since a narrower set of regions changes both the instance catalogue and the price basis (Table 1, D2). And link the framework to fin-

ancial reporting so that the savings expected under national cloud policy (D1) can be shown, not assumed.

A further practice concerns timing. Commitments are usually signed at go-live, which is exactly when telemetry is thinnest. Where contracts permit, a short initial term or a smaller commitment for the first six to twelve months, followed by a larger one once forecasts have been checked against real growth, reduces the risk of locking in an oversized base. The same logic applies to the standby and recovery copies, whose size should follow production through the same decision rule rather than being fixed separately.

5.3 Illustrative example

An invented case shows the effect. Assume a production HANA database whose required memory, including the dynamic allowance, starts at 1.5 TB and grows by 2% a month, reaching about 3.0 TB after three years. Assume also, hypothetically, that instances are available at 2, 3, 4 and 6 TB, priced in proportion to memory. These values are illustrative and do not describe any real system or provider.

A conventional plan provisions 4 TB from the start to cover year-three growth, consuming 144 terabyte-months over 36 months. Under the framework, the decision layer moves up when the forecast for three months ahead, inflated by a 10% uncertainty band, exceeds 90% of current capacity. In the example this triggers a move to 3 TB in month 3 and to 4 TB in month 23, for a total of 120 terabyte-months, about 17% less (Figure 2). The saving is modest by the standards of the laboratory studies, which is the point: it comes without reducing safety margins, because capacity always stays above the upper forecast band. Larger savings would follow if archiving or data tiering slowed memory growth, or if the initial sizing had been more generous than the doubling rule.

The size of the effect depends on growth. Repeating the invented calculation with monthly growth of 1%, the final requirement is about 2.1 TB, the conventional plan provisions 3 TB throughout and the stepped plan saves about 6%, with a single step in month 7. At 3% a month, the requirement reaches about 4.2 TB, the conventional plan provisions 6 TB from the start and the stepped plan, with three steps, saves about 29%. In none of the three variants did provisioned capacity fall below required memory in any month. Fast-growing systems therefore gain most from forecasting, while slow-growing ones gain mainly from a smaller initial commitment.

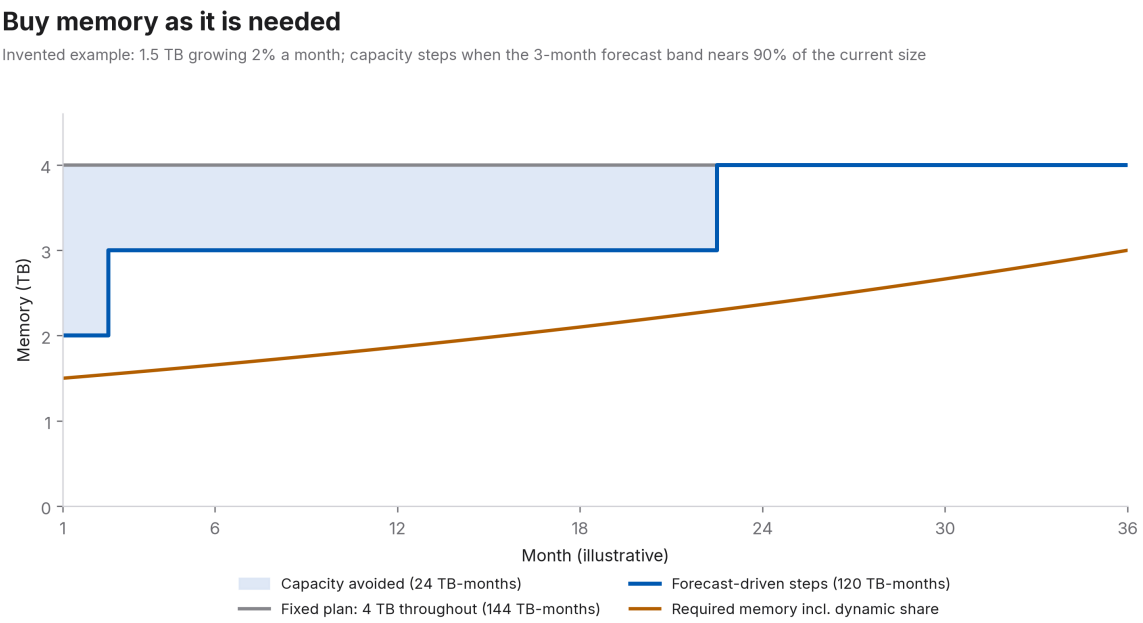

Figure 2. Illustrative memory provisioning over 36 months: fixed final-year sizing compared with forecast-driven steps (invented values, not data from any real system)

6 Discussion

6.1 Contribution to knowledge

Both anchor studies point in a useful direction. Myung and Choi (2021) showed that memory needs can be learned from workload characteristics, and Bach et al. (2022) showed how complex and fast-moving HANA itself is. This paper connects those findings with the much larger literature on cloud workload prediction and cost optimisation and identifies what does not transfer. Most forecasting studies predict processor use for fleets of interchangeable machines over short horizons; HANA sizing needs memory forecasts for a single stateful system over horizons of months, with explicit uncertainty. The framework fills that gap by pairing vendor sizing with a forecast band and a change-managed decision rule.

The review also qualifies the large savings reported in the literature. Reductions of half or more usually reflect elastic, stateless workloads or poorly sized baselines. For a memory-bound database the realistic gains come from avoiding front-loaded capacity, choosing commitments carefully and moving cold data out of memory, and they should be estimated case by case.

A third contribution is to place prediction inside an operating discipline. Pricing research already frames cloud cost as a joint concern of engineering, finance and business teams, and decision support for comparing pricing policies across providers is available (Fragiadakis et al., 2024). The framework gives such teams a concrete object to manage: a forecast band, a threshold and a documented decision for each resize. That makes savings auditable and errors visible, which matters for public bodies whose spending is scrutinised.

6.2 Implications for organisations in the Kingdom

For Saudi organisations moving SAP to the cloud, the first implication is to stop treating the initial sizing as final. Contracts should allow capacity to step up during the term without penalty, and

commitments should cover the level that forecasts show will persist rather than the final-year peak. The second is to build telemetry from day one, since forecasting needs history. The third is to develop skills that sit between database administration and finance, the role usually described as cloud financial operations, so that forecasts lead to purchasing decisions. Public bodies could share anonymised growth profiles, which would allow better forecasts for new entrants than any single organisation's data allows.

For policymakers and the agencies that oversee government cloud adoption, the findings suggest that guidance on cost should cover how capacity is bought, not only where systems are hosted. Model contract terms that allow capacity to step up and down, shared benchmarks for in-Kingdom instance families and a common template for reporting utilisation would help organisations realise the savings that national policy anticipates. Providers serving the Saudi market would also benefit, since transparent right-sizing makes long-term commitments easier for customers to justify.

6.3 Limitations and future research

This review has limits. Its thirty articles include none that forecasts HANA memory directly, so the framework transfers methods from neighbouring settings. Vendor sizing guidance, though public, has no independent validation. The example uses invented values and a simplified price model with no discounts, minimum terms or downtime costs. The search was limited to English-language journal articles, so practitioner reports and conference papers, where much engineering experience with HANA operations appears, were excluded, and studies reporting large savings may be more likely to be published than those reporting small ones. Future work should test the framework on real HANA telemetry, compare learned forecasts of memory growth with vendor sizing over full contract periods, and measure the cost and service effect of forecast-driven resizing in regions available to Saudi organisations.

7 Conclusion

Memory dominates the cost of running SAP HANA in the cloud, and the usual practice of buying for the final year from the start leaves capacity idle for much of the contract. This paper has drawn on thirty studies and four documentary sources to design a five-layer right-sizing framework that begins with vendor sizing, learns from telemetry, forecasts demand with uncertainty, converts forecasts into instance, commitment and storage choices and closes with governance. An invented case shows that stepping capacity up with demand can cut provisioned memory by about a sixth without lowering safety margins. The evidence base is strong for forecasting in general but thin for in-memory ERP databases, and testing on real workloads is the necessary next step.

References

[1] Bach, T., Andrzejak, A., Seo, C., Bierstedt, C., Lemke, C., Ritter, D., Hwang, D. W., Sheshi, E., Schabernack, F., Renkes, F., Gaumnitz, G., Martens, J., Hoemke, L., Felderer, M., Rudolf, M., Jambigi, N., May, N., Joy, R., Scheja, R., . . . Lehner, W. (2022). Testing very large database management systems: The case of SAP HANA. Datenbank-Spektrum, 22(3), 195–215. https://doi.org/10.1007/s13222-022-00426-x

[2] Chen, S., Lei, J., & Moinzadeh, K. (2024). Cost optimization in cloud computing: Capacity reservation for intermittent random demand surges. *Production and Operations Management*, 33(6), 1265–1284. https://doi.org/10.1177/10591478241251614

[3] Chhetri, M. B., Forkan, A. R. M., Vo, Q. B., Nepal, S., & Kowalczyk, R. (2023). Towards proactive risk-aware cloud cost optimization leveraging transient resources. *IEEE Transactions on Services Computing*, 16(4), 3014–3026. https://doi.org/10.1109/tsc.2023.3253473

[4] Chouliaras, S., & Sotiriadis, S. (2023). An adaptive auto-scaling framework for cloud resource provisioning. *Future Generation Computer Systems*, 148, 173–183. https://doi.org/10.1016/j.future.2023.05.017

[5] Dang-Quang, N.-M., & Yoo, M. (2022). An efficient multivariate autoscaling framework using Bi-LSTM for cloud computing. *Applied Sciences*, 12(7), Article 3523. https://doi.org/10.3390/app12073523

[6] Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T. A., Min, D., Choi, E., Silva, F. A., & Maciel, P. R. M. (2022). Performance-cost trade-off in auto-scaling mechanisms for cloud computing. *Sensors*, 22(3), Article 1221. https://doi.org/10.3390/s22031221

[7] Fragiadakis, G., Filiopoulou, E., Michalakelis, C., Kamalakis, T., & Nikolaidou, M. (2023). Applying machine learning in cloud service price prediction: The case of amazon IaaS. *Future Internet*, 15(8), Article 277. https://doi.org/10.3390/fi15080277

[8] Fragiadakis, G., Tsadimas, A., Filiopoulou, E., Kousiouris, G., Michalakelis, C., & Nikolaidou, M. (2024). Cloud PricingOps: A decision support framework to explore pricing policies of cloud services. *Applied Sciences*, 14(24), Article 11946. https://doi.org/10.3390/app142411946

[9] Gyeera, T. W., Simons, A. J. H., & Stannett, M. (2023). Kalman filter based prediction and forecasting of cloud server kpis. *IEEE Transactions on Services Computing*, 16(4), 2742–2754. https://doi.org/10.1109/tsc.2022.3217148

[10] Haubenschild, M., & Leis, V. (2025). Oltp in the cloud: Architectures, tradeoffs, and cost. *The VLDB Journal*, 34(4), Article 42. https://doi.org/10.1007/s00778-025-00913-z

[11] Huang, D., Costero, L., Pahlevan, A., Zapater, M., & Atienza, D. (2024). CloudProphet: A machine learning-based performance prediction for public clouds. *IEEE Transactions on Sustainable Computing*, 9(4), 661–676. https://doi.org/10.1109/tsusc.2024.3359325

[12] Karim, M. E., Maswood, M. M. S., Das, S., & Alharbi, A. G. (2021). BHyPreC: A novel Bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine. *IEEE Access*, 9, 131476–131495. https://doi.org/10.1109/access.2021.3113714

[13] Khan, A. Q., Matskin, M., Prodan, R., Bussler, C., Roman, D., & Soylu, A. (2024). Cloud storage cost: A taxonomy and survey. *World Wide Web*, 27(4), Article 36. https://doi.org/10.1007/s11280-024-01273-4

[14] Leka, H. L., Fengli, Z., Kenea, A. T., Hundera, N. W., Tohye, T. G., & Tegene, A. T. (2023). Pso-based ensemble meta-learning approach for cloud virtual machine resource usage prediction. *Symmetry*, 15(3), Article 613. https://doi.org/10.3390/sym15030613

[15] Liu, Z., Han, R., Zhang, Y., Zhang, Y., Tang, X., Deng, G., Zhong, T., Dementiev, R., Lu, Y., & Que, M. (2023). Exploring fine-grained in-memory database performance for modern CPUs. *IEEE Transactions on Parallel and Distributed Systems*. Advance online publication. https://doi.org/10.1109/tpds.2023.3262782

[16] Maiyza, A. I., Korany, N. O., Banawan, K., Hassan, H. A., & Sheta, W. M. (2023). VTGAN: Hybrid generative adversarial networks for cloud workload prediction. *Journal of Cloud Computing*, 12(1), Article 97. https://doi.org/10.1186/s13677-023-00473-z

[17] Mandal, S., Maji, G., Khatua, S., & Das, R. K. (2023). Cost minimizing reservation and scheduling algorithms for public clouds. *IEEE Transactions on Cloud Computing*, 11(2), 1365–1380. https://doi.org/10.1109/tcc.2021.3133464

[18] Myung, R., & Choi, S. (2021). Machine-learning based memory prediction model for data parallel workloads in Apache Spark. *Symmetry*, 13(4), Article 697. https://doi.org/10.3390/sym13040697

[19] Nawrocki, P., & Osypanka, P. (2021). Cloud resource demand prediction using machine learning in the context of QoS parameters. *Journal of Grid Computing*, 19(2), Article 20. https://doi.org/10.1007/s10723-021-09561-3

[20] Nawrocki, P., & Sus, W. (2022). Anomaly detection in the context of long-term cloud resource usage planning. *Knowledge and Information Systems*, 64(10), 2689–2711. https://doi.org/10.1007/s10115-022-01721-5

[21] Nawrocki, P., Osypanka, P., & Posluszny, B. (2023). Data-driven adaptive prediction of cloud resource usage. *Journal of Grid Computing*, 21(1), Article 6. https://doi.org/10.1007/s10723-022-09641-y

[22] Osypanka, P., & Nawrocki, P. (2023). QoS-Aware cloud resource prediction for computing services. *IEEE Transactions on Services Computing*, 16(2), 1346–1357. https://doi.org/10.1109/tsc.2022.3164256

[23] Qian, Z., Wei, J., Xiang, Y., & Xiao, C. (2021). A performance evaluation of DRAM access for in-memory databases. *IEEE Access*, 9, 146454–146470. https://doi.org/10.1109/access.2021.3123379

[24] Quattrocchi, G., Incerto, E., Pinciroli, R., Trubiani, C., & Baresi, L. (2024). Autoscaling solutions for cloud applications under dynamic workloads. *IEEE Transactions on Services Computing*, 17(3), 804–820. https://doi.org/10.1109/tsc.2024.3354062

[25] Rossi, A., Visentin, A., Carraro, D., Prestwich, S., & Brown, K. N. (2025). Forecasting workload in cloud computing: Towards uncertainty-aware predictions and transfer learning. *Cluster Computing*, 28(4), Article 258. https://doi.org/10.1007/s10586-024-04933-2

[26] Saxena, D., Kumar, J., Singh, A. K., & Schmid, S. (2023). Performance analysis of machine learning centered workload prediction models for cloud. *IEEE Transactions on Parallel and Distributed Systems*, 34(4), 1313–1330. https://doi.org/10.1109/tpds.2023.3240567

[27] Wang, S., Yao, Y., Zhu, F., Tang, W., & Xiao, Y. (2020). A probabilistic prediction approach for memory resource of complex system simulation in cloud computing environment. *Symmetry*, 12(11), Article 1826. https://doi.org/10.3390/sym12111826

[28] Wen, Y., Wang, Y., Liu, J., Cao, B., & Fu, Q. (2020). CPU usage prediction for cloud resource provisioning based on deep belief network and particle swarm optimization. *Concurrency and Computation: Practice and Experience*, 32(14), Article e5730. https://doi.org/10.1002/cpe.5730

[29] Xu, M., Song, C., Wu, H., Gill, S. S., Ye, K., & Xu, C. (2022). EsDNN: Deep neural network based multivariate workload prediction in cloud computing environments. *ACM Transactions on Internet Technology*, 22(3), 1–24. https://doi.org/10.1145/3524114

[30] Zhao, W., Du, Y., Zhang, M., Liu, M., Jin, K., & Ausavarungnirun, R. (2021). Application-oriented data migration to accelerate in-memory database on hybrid memory. *Micromachines*, 13(1), Article 52. https://doi.org/10.3390/mi13010052