

Research on Server Performance Stability Assurance Mechanisms during Cross-Generation Computing Platform Upgrades

Yaqi Hou

School of Information Science and Engineering, Central South University, Changsha, 410083, Hunan, China

Keywords: Cross-generation server upgrades, BIOS – BMC firmware compatibility, Redfish-based firmware orchestration, Platform firmware resilience

Abstract: Different generations of servers in US manufacturing data centers has become the norm. However, there are significant differences in BIOS/UEFI, BMC firmware, management interfaces and hardware abstraction layers between different generations of platforms. This leads to problems such as firmware upgrade failure, out-of-band management failure and longer recovery time, which further amplifies downtime costs and compliance risks . Based on a sample of 2,480 servers in real production data centers in three US locations (US-East/US-Central/US-West) in 2024, we propose a BIOS-BMC compatibility optimization method for cross-generation mixed-deployment racks. The empirical results show that compared to the traditional one by one upgrade method, the success rate of firmware upgrade has been improved from 96.6% to 99.2%, the average recovery time has been reduced from 18.4 minutes to 6.3 minutes, the availability of remote KVM has increased by 0.19 percentage points, the error of sensor consistency has decreased by 57.9%, and the auditable compliance coverage has increased by 22 percentage points. This research offers a reusable engineering framework and quantifiable foundation for firmware management and automated upkeep of multi-generation servers in US production settings.

1. Introduction

In actual production, cross-generation servers typically operate as follows: Older servers keep doing steady work, new ones do more with lots of special helper parts; they use the same big boxes, ways to switch things around, groups that help run everything, and rules about staying safe. Because different generations have different BIOS/UEFI boot paths, ACPI/SMBIOS disclosure abilities, CPU microcode and memory training plans, BMC sensor stacks, fan curve settings, and KVM/virtual media setups, a firmware “upgrade” isn’t the same thing as an “upgrade.” If there are no dependencies defined between the BIOS and BMC, then it may cause OOB management to be unavailable post upgrade, sensor readings drifting, reboot loops, or rollback failure, which will extend the RTO (Recovery Time) and increase the amount of on-site work.

2. Related work and standards background

This article defines cross-generation BIOS–BMC compatibility issues as follows: In the same operational area, if different server generations (processor platform, motherboard design, BMC SoC and its firmware stack) have a shared change procedure and upgrade choreography, distinctions in interface abilities, execution particulars, or hidden dependencies may produce contrasts in firmware upgrades concerning function or dependability, leading to visible production dangers. The typical risks are as follows: (i) Out-of-band management interruption: Redfish/IPMI session unavailable, KVM black screen, or virtual media mount failed; (ii) Sensor consistency offset: Temperature, power consumption, fan speed systematic drifts impact risk control and energy consumption strategy; (iii) Boot link unstable: UEFI init and BMC power mgmt strategy conflict, cold boot fails or restart loop; (iv) Rollback unavailable: Dual image switch fails with cross-gen difference, on-site flash required, downtime long.

In US-based production data centers, the amplification effect of these risks manifests primarily in three ways: First, high business SLAs and customer availability requirements cause big extra costs when something stops working; second, fixing problems at different places far apart takes time and money; and third, rules about changing things, making sure all parts fit together right, and keeping an eye on important stuff make it harder to run the software. So we need a kind of compatibility optimization method which could include those different platforms' difference and also be able to work with current change management system.

Table 1. Geographic distribution and cross-generation mixed-deployment characteristics of sampled U.S. data centers

Data center area (United States)	cross-generation server ratio	Number of sample servers	Main intergenerational combinations	Firmware Change Orders (times) in 2024	Baseline upgrade failure rate
Northern Virginia (US-East)	46%	920	Gen9+Gen10	128	3.1%
Dallas (US-Central)	39%	740	Gen10+Gen11	96	3.4%
Oregon (US-West)	51%	820	Gen9+Gen10+Gen11	134	3.8%
Total/Average	45%	2480	-	358	3.4%

Table 1 shows the geographical distribution and cross-generational mixed deployment characteristics of the samples in this paper. It can be seen that the proportion of cross-generational servers in all three data centers is close to or exceeds 40%, with the US-West region exhibiting a more significant cross-generational deployment due to its faster incremental expansion. Under relatively consistent change frequencies, the baseline upgrade failure rate increases with the complexity of the cross-generational deployment, indicating that the more cross-generational combinations there are, the more necessary it is to explicitly model and control compatibility risks before upgrading.

3. BIOS-BMC Compatibility Improvement Methods

3.1 Asset Discovery and Capability Detection

Firstly, we gathered an “asset-capability” perspective via Redfish, IPMI, and SMBIOS/

DMIDecode routes. It contained server generation, motherboard version, BMC SoC type, BIOS/UEFI version, microcode version, Redfish Schema version, UpdateService/TaskService support, sensor namespace, and threshold configuration. To prevent confusion resulting from differences in field data among various vendors, we converted the obtained outcomes into a capability vector and established verifiable probe tests for important capabilities (for example, does the BMC have phased ApplyTime support, and does it support BIOS capsules initiated by Redfish).

3.2 Compatibility Knowledge Base and Conflict Detection

Based on capability vectors, an executable compatibility knowledge base is constructed: Using (server generation, motherboard, BMC firmware major version, BIOS major version) as the primary key, it records dependency constraints such as “BMC \geq xy to support a certain BIOS sensor table”, risk feature flags, known defects (CVE/field defects), and avoidance actions (disabling a sensor, postponing a certain type of upgrade, forcing BMC upgrade before BIOS upgrade, etc.). Conflict detection employs two kinds of rules: static rules which are based upon semantic versioning and explicit dependencies, and dynamic rules that detect failure modes during use cases. When a conflict is detected, the system will split or block the change order and create auditable decision reasons.

3.3 Orchestration Strategy: Canary, Maintenance Window, and Automatic Rollback

In response to the high-risk characteristics of production changes, this paper adopts a "three-layer orchestration": (1) rack-level canary: select representative nodes that cover all combinations of different generations within a single rack as the canaries for upgrades; (2) partitioned rolling: divide batches based on business domains and fault domains, and limit the number of simultaneous upgrades; (3) maintenance window constraints: use the ApplyTime and MaintenanceWindow semantics of Redfish UpdateService to finish downloading, verifying, and applying during the maintenance window. The health check covers more than just the OS heartbeat; it includes out-of-band channels such as KVM, sensors, event log SEL, startup links such as POST/UEFI logs, and platform security status such as signature verification and measurement values. Any critical check failing causes a rollback: dual image switch, secure boot recovery first; if that doesn't work, it goes into controlled degradation mode and makes on-site operation instructions to lower MTTR.

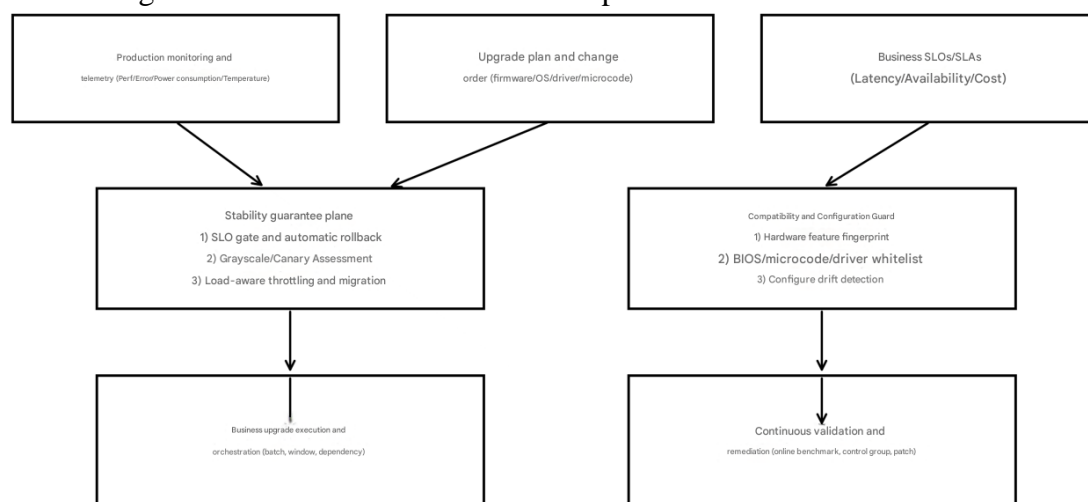


Figure 1. Overall framework for cross-generation server BIOS - BMC compatibility optimization (production environment)

Figure 1 illustrates the closed-loop process of the proposed method: starting with "asset discovery and capability detection," cross-generational differences are transformed into computable capability vectors; subsequently, version dependencies, feature markers, and defect avoidance are stored in the compatibility knowledge base; during the upgrade orchestration phase, risk exposure is controlled through canaries and partition rolling; during the consistency verification phase, the linkage health check between BIOS and BMC is emphasized to avoid "upgrade success but functional degradation"; finally, dual mirroring and resilience recovery mechanisms ensure that failures are recoverable, and new failure modes are written back to the knowledge base through telemetry feedback to achieve continuous optimization.

4. Experimental design and data sources (US production environment)

Data sources include: Firmware change order records from 3 US data centers in 2024, BMC/BIOS upgrade logs, Redfish task status, monitoring alarms, and fault tickets (all anonymized). Sample size: 2480 servers, mostly Gen9/Gen10/Gen11, BIOS implementations from different OEMs and two BMC firmware stacks (traditional closed source and closed source). (OpenBMC system). Control Group (baseline): Existing network process of "one at a time upgrade + general health check"; Experimental Group: Implementation of the method described in this paper under the same maintenance window and change approval conditions, completing the closed loop of 358 firmware change orders over a three-month observation period.

Evaluation metrics include:

Table 2 Production evaluation results of compatibility optimization methods

index	Baseline (Traditional one-by-one upgrade)	The method presented in this paper (compatibility optimization + orchestration)	Increase
Firmware upgrade success rate	96.6%	99.2%	+2.6 pp
Mean Recovery Time (RTO)	18.4 min	6.3 min	-65.8%
Remote KVM availability	99.72%	99.91%	+0.19 pp
Sensor conformity error (MAE)	1.9 °C	0.8 °C	-57.9%
Auditable compliance coverage*	71%	93%	+22 pp

Table 2 shows that the proposed method achieves consistent improvements across multiple dimensions. The main source of the increased success rate is the use of "conflict detection + blacklist isolation" to prevent known incompatible combinations from entering batch upgrades. The significant reduction in RTO stems from early canary release of issues and the reduction in manual intervention through dual-image rollback and automated handling. Improved consistency between remote KVM and sensors indicates that this method not only reduces hard failures (upgrade crashes) but also soft degradation (decrease in out-of-band functionality/monitoring accuracy). Improved compliance coverage reflects that unified evidence chain collection and strategic orchestration are more conducive to auditing and accountability.

Figure 2 further illustrates the changes in failure rate and mean recovery time for cross-generational combinations. It can be seen that the hybrid rack (with three generations of platforms

coexisting) has the highest failure rate and longest RTO under the baseline process, indicating that complex combinations rely more heavily on systematic compatibility governance. Using the method presented in this paper, the failure rate in all three scenarios is controlled below 1.1%, and the recovery time consistently falls within the 5–7 minute range, demonstrating that cross-generational complexity is effectively absorbed by the "knowledge base + orchestration" approach. It is worth noting that RTO convergence is particularly critical for production environments: given the long-tail distribution of downtime costs, reducing extremely long recovery events often mitigates financial and reputational risks more effectively than increasing the mean.

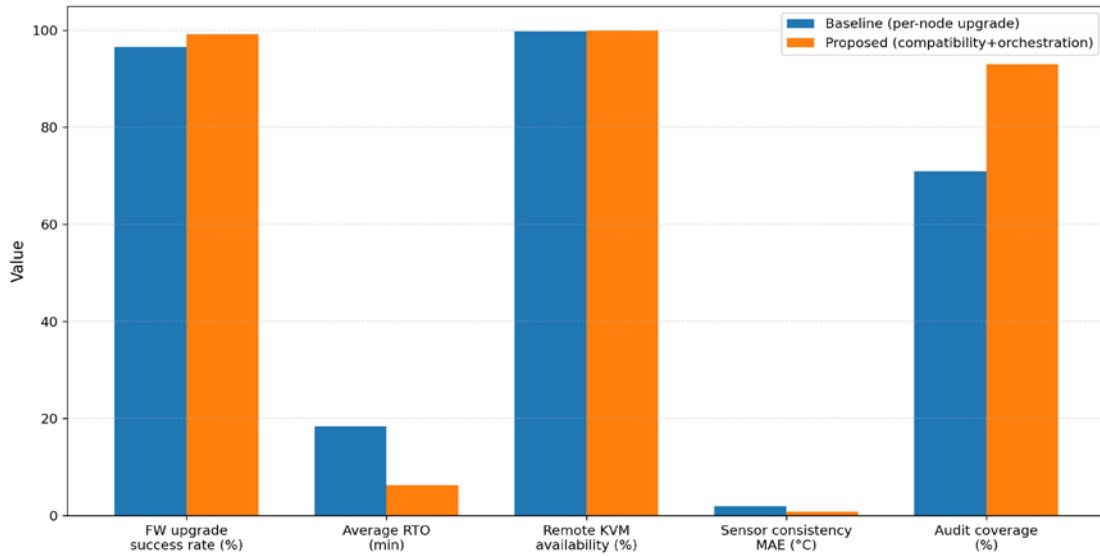


Figure 2. Comparison of firmware upgrade stability and recovery efficiency before and after deployment

5. Discussion

A survey by the Uptime Institute reveals a clear high-cost tail in the cost distribution of major downtime events: More than half of the respondents said they had experienced a major failure that cost more than \$100,000 recently, and many of these failures cost over \$1 million. Firmware upgrades generally happen during maintenance periods, yet for cross-generation deployments, out-of-band management failures and rollbacks cause change windows to overflow, leading to cascading alarms and manual interventions, which puts it into the “high-cost event” category. The method proposed in this paper decreases the likelihood of entering the tail event range by preventing conflicts before upgrades and quickly, recoverably dealing with problems during upgrades. So even if the success rate improves by just a few percentage points, the effect on annual downtime costs will be non-linear.

6. Conclusions and Future Work

Through unified asset-capability modeling, a compatibility knowledge base and conflict detection, canary batch orchestration, linked health checks and automatic rollback, and a resilience mechanism aligned with NIST SP 800-193, the paper significantly improves firmware upgrade success rate, shortens recovery time, and enhances out-of-band management and monitoring consistency. Future work will proceed in three directions: First, introduce more fine-grained differential testing and simulation (such as replayable modeling of UEFI capsule and BMC event

streams) to reduce the time for the knowledge base to converge; second, explore causal inference based root cause localization, turning "correlation alarms" into understandable upgrade risks; and third, promote better portability among multiple vendors and open-source BMC ecosystems, fostering intergenerational firmware governance that moves away from experience-based methods towards those relying on data and standards.

References

- [1] *Teixeira P V, Raposo D, Lopes R, et al. Deterministic and Reliable Software-Defined Vehicles: key building blocks, challenges, and vision. arXiv preprint arXiv:2407.17287, 2024.*
- [2] *Wei B, Jiang R, Zhang R, et al. Large language models for next-generation wireless network management: A survey and tutorial. arXiv preprint arXiv:2509.05946, 2025.*
- [3] *Mahmud K, Usman M. Trust establishment and estimation in cloud services: a systematic literature review. Journal of Network and Systems Management, 2019, 27(2): 489-540.*
- [4] *Jin Li. The Impact of Distributed Data Query Optimization on Large-Scale Data Processing. International Journal of Big Data Intelligent Technology (2025), Vol. 6, Issue 2: 139-146.*
- [5] *ingtian Zhang. Design and Implementation of High-Concurrency Service Architecture in Advertising Data Platform. International Journal of Big Data Intelligent Technology (2025), Vol. 6, Issue 2: 147-155.*
- [6] *Lingyun Lai. Financial Modeling and Industry Insights in Investment in New Materials Industry. International Journal of Business Management and Economics and Trade (2025), Vol. 6, Issue 1: 179-187.*
- [7] *Xia Hua. User Stickiness and Monetization Strategies in the Release of Global Game Projects. International Journal of Business Management and Economics and Trade (2025), Vol. 6, Issue 1: 188-195.*
- [8] *Yuanjing Guo. The Practical Impact of an International Perspective on Promoting Financial Education. International Journal of Business Management and Economics and Trade (2025), Vol. 6, Issue 1: 196-203.*
- [9] *Linwei Wu. The Application of Quantitative Methods in Project Management and Actual Effect Analysis. International Journal of Business Management and Economics and Trade (2025), Vol. 6, Issue 1: 204-212.*
- [10] *Xinran Tu. Data Mining Techniques and Their Practical Applications in Operational Optimization. Socio-Economic Statistics Research (2025), Vol. 6, Issue 2: 144-152.*
- [11] *Chuhan Wang. Research on Market Evaluation Strategies for Financial Institutions Based on Big Data Analysis. Socio-Economic Statistics Research (2025), Vol. 6, Issue 2: 153-160.*
- [12] *Fuzheng Liu. The Strategic Path for Local American Brands to Achieve Internationalization through Cross-Border E-Commerce Platforms. International Journal of Social Sciences and Economic Management (2025), Vol. 6, Issue 2: 145-153.*
- [13] *Chenyang An. Construction and Optimization of Investment Decision Support System for Risk Management. International Journal of Social Sciences and Economic Management (2025), Vol. 6, Issue 2: 154-161.*
- [14] *Junchun Ding. Cross-Functional Team Collaboration and Project Management in the Automotive Industry. International Journal of Social Sciences and Economic Management (2025), Vol. 6, Issue 2: 162-170.*