Computer Fraud and Security
ISSN (online): 1873-7056

High-Availability HPC Cluster Design for Mission-Critical Public Infrastructure: Lessons from Energy Grid and Government Deployments

Rakesh Challa

Principal Engineer, Dell Technologies

***Abstract*—**The paper addresses the design issue of fault-tolerant high-availability HPC clusters of mission-critical systems in the public-sector, such as energy grid operators and government data centers. The paper focuses on such architectural designs as redundant pathing, multi-layer failure, homogeneity in the firmware and risk managed cutovers to assist in achieving zero-downtime operations. The system performance was quantitatively evaluated by running a simulation-based experiment which incorporated Monte Carlo fault injection. Multi-layer failover with firmware homogeneity availability, MTTR, and recovery success was 99.97, 38 and 99.9 seconds respectively, whereas controlled cutovers also increased availability to 99.98. The workload was completed over 99.5 percent even with several faults. These findings point to the fact that properly designed HPC systems may be maintained, and used in handling large volumes of data and be cyber-resilient in serving millions of people. The paper provides a methodical method of designing and assessing the high availability HPC systems that can be applied in the national interest activities.

Keywords*—*High-Availability HPC, Fault-Tolerant Clusters, Mission-Critical Infrastructure, Multi-Layer Failover, Redundant Pathing

I. INTRODUCTION

A. Background and Motivation

Statewide power systems, municipal government data centers are also implemented with HPC systems, which are mission-critical public infrastructure. These systems are able to handle large volumes of information in real time to give continuity and reliability to the service to millions of users. Any amount of downtime can be catastrophic to the operations and economic implication of a company and fault tolerance and high availability is crucial. The faster growing data and computing requirements determine the demand of performance, reliability, and maintainability requirement architectures. Traditional groups of HPCs are largely focused on the raw power, but in the case of the deployment to the public, it must be continuously working and under strict regulatory and operational restrictions.

The rationale as to why this research was done is because there is a need to have a quantitative evaluation of the HPC-based architecture that can handle faults, a guarantee of availability, and high-throughput workloads. The study is a good solution to the government and the energy infrastructure through simulating real-world applications, such as similar to those of ERCOT and the City of Oklahoma.

B. Novelty of the Study

The originality of the present paper consists in the combination of several architectural solutions into the one HPC design to achieve the zero-downtime work. Earlier studies tended to tackle high availability, fault tolerance or workload performance individually. This study combines:

1. Elimination of points of failure.
2. Multi-level hardware, software, operating system and application failure.
3. Standardisation of the firmware to reduce differences in recovery.
4. Cutovers managed by faults in order to maintain continuity in the case of planned maintenance.

The research is quantitative with the application of simulation techniques based on Monte Carlo faults injection in order to offer quantifiable system performance evidence. Effectiveness is demonstrated using metrics such as availability, MTTR, recovery success, and workload completion, and thus the methodology is repeatable and can be used in the real-world deployments.

C. Research Objectives

The key aims of the research are:

- To come up with a HPC cluster architecture that would allow running without downtime in the public-sector settings.
- To establish the impact of redundancy and failover layers, firmware consistency and cutovers on system reliability and availability.
- To measure the recovery times, fault detecting efficiency and workload completion when faults occur in controlled conditions.
- To offer engineering consultation in the area of HPC infrastructure to support national interest services, such as energy grids and government data centers.

The study will fill the gap between high-performance computing and high-availability demands in the public infrastructure by answering these objectives.

D. Structure of the Paper

The structure of the paper is the following:

1. **Introduction:** Background, motivation, innovation and goals.
2. **Literature Review:** Literature survey of past studies on HPC fault tolerance, high-availability cluster, and resilience patterns.
3. **Methodology:** Defines quantitative modeling, fault modeling, quantities measures and workflow.
4. **Findings/Results:** MTTR, recovery efficiency and rate of work completion with the support of tables and figures.
5. **Conclusion:** Concludes on the important findings and gives recommendations on the deployment of HPC in mission critical environments.

These divisions allow the conceptual design to have a well-defined direction to the quantitative analysis and impactful recommendations.

E. Contribution and Impact

This research gives a contribution to the HPC research in the form of offering:

- A comprehensive approach to high-availability cluster design and evaluation.
- Quantitative results of multi-layer failover, homogeneity of the firmware, and regulated cutovers to the level of 99.98% availability and 35-38 seconds MTTR.
- Recommendations on how the public-sector HPC should be deployed, such that it can complete its workload, even in case of a fault, at a rate of more than 99.5%.

The findings facilitate crucial infrastructure in the country, proving that the wise approach to architecture can guarantee irregular functioning, data integrity, and cyber-resilience of millions of residents.

II. Literature Review

A. High Availability and Fault Tolerance in Mission-Critical Clusters

Critical requirements of mission-critical systems that cannot tolerate downtime include high availability (HA) and fault tolerance, typically of the public infrastructure, energy grids, and government data centers. It is evident that according to research work done in early years the failures in clustered system are inevitable as the size and complexity of the system rises [1][2]. To handle this, HA clusters are configured to have continuity of the services by redundancy, monitoring and speedy recovery [3]. Such systems are not static but instead make sure that a hardware, software and network-layers are sustained but automatically re-initiate or migrate the workloads in the event of failure.

According to a number of studies, HA clusters appear to be a popular technology in enterprise and governmental settings due to their ability to safeguard application layers of critical applications and common resources [4]. Active-standby and

active-active are widely used to minimize the downtime and enhance the reliability [4]. The models come in handy to government implementations where the failure of service may affect millions of users. Another point of the literature is that the integration of HPC and HA methods is feasible and productive in the course of the fall of the prices of hardware and enhancement of the software-based management.

Regardless of numerous available solutions, there are still difficulties with the existing HA systems associated with the complexity of configuration, the accuracy of failure detection, and the speed of recovery [5]. These difficulties are enhanced in the public infrastructure environment whereby strict compliance, predictable behavior and risk-controlled operations are demanded.

B. Resilience Design Patterns and System-Level Approaches in HPC

Due to the new trend of deployment of HPC systems to extreme and exascale systems, their reliability has been a major issue due to the regular hardware and software failures [6][7]. According to some studies, the conventional ad hoc fault-tolerance approaches cannot suffice anymore. Instead, there is a need to use rational design techniques to build strong structures [7]. Resilience design pattern has been proposed as a methodology that should be considered with respect to the different kinds of faults in the hardware, system program and application [7][8].

Design patterns help an engineer to implement solutions that have been successfully used in common problems such fault detection, fault containments and recovery. This trend was also facilitating the integration of different resilience approaches at different levels that offer enhanced system resilience [8]. This multi-level plays a vital role in the public infrastructure HPC systems where failure must be contained with minimal delay and without interference with important services.

Another observation in the study is that measures and measurement systems are critical to quantify resilience efficacy [9]. The new reliability metrics such as failure indices provide a more insight as to the behaviour of the application and the state of the system health than the old metrics such as the mean time between failure [9]. They are implementable in the government and energy related implementations where reliability reporting and compliance verification are needed.

C. Checkpointing, Recovery, and Energy-Aware Fault Management

Checkpointing and rollback-recovery (especially in the long-running application case) is the most widespread fault-tolerance mechanism applied in HPC clusters today. Various studies provide elaborate taxonomies of checkpoint/restart solutions and see their benefits and drawbacks. These are good approaches, but introduce large-scale performance and storage cost overheads.

In order to overcome these issues, newer approaches are geared towards minimizing the recovery period and removing the interference. The other methods that have proved to be effective in time sensitive applications such as weather forecast include the in-memory checkpointing technique, the compressed checkpointing technique, and the user-level failure mitigation techniques [10]. The newer recovery models such as Reinit++, and ULFM based solutions reduce the startup time and increase the scalability (by very large margins) [11][12].

The other concern is on energy efficiency. Research suggests that fault tolerance tends to be enhanced due to enhanced consumption of energy, since redundancy and infrastructure are provided [13][14]. The studies indicate that there are energy-sensitive recovery strategies which would optimize reliability and power usage, specifically the use of one in the framework of the public-sector data centers of limited budget and sustainable energy strategies [14].

D. Emerging Trends: Cloud, Machine Learning, and Software-Defined Infrastructure

Recent sources indicate that there is a sharp inclination to cloud-based and software-defined infrastructures to ensure fault tolerance and maximum availability [15]. These systems offer flexibility, scaling and it can be easily managed, however, new failure modes are introduced, which should be addressed cautiously. Reference roadmaps and taxonomies provide the system designer with knowledge of where, when and how to use reliability mechanisms in complex environments.

HPC systems are also utilizing machine learning to detect and predict and classify faults. These methods have the capability of processing live system data and detecting faults with minimal error and minimum latency to pre-emptive recovering. These can be useful in deploying energy grid and government HPC applications that need cyber-resilience and 24/7 monitoring.

Containerization, microservices and fault-tolerant scheduling also contribute to improving system availability since failures are isolated and can be recovered very quickly [15][16][17]. The literature is unanimous that real-life implementation continues to encounter issues pertaining to integration, validation and operational risk. Research prototypes require more practical studies to close the gap between research and production-grade systems of public infrastructure.

TABLE I. SUMMARY OF PREVIOUS STUDIES

Focus Area	Key Findings	Relevance to Mission-Critical HPC
High-Availability Clusters	The literature demonstrates that high-availability clusters rely on redundancy, monitoring and automatic failover to decrease downtime and continue important services in the event of failure [2][3][4].	This helps keep the systems of the public infrastructure running (i.e. energy grids) and government data centers.
Fault Tolerance Techniques	Research indicates that checkpointing, rollback recovery and process migration are commonly used to manage node and process failures of large HPC clusters [1][18][11].	These methods assist in mission-critical systems to be brought back to normal within a short time without complete system reboots.
Resilience Design Patterns	Research proposes patternized resilience designs which integrate hardware, software, and application-levels to manage various types of faults in HPC systems [7][8].	Design patterns offer a logic approach to the development of reliable HPC systems to be used in national infrastructure.
Performance and Energy Trade-offs	The literature points out that enhancing fault tolerance can be a source of overhead and energy wastefulness that need to be well balanced [13][14].	Large deployments of HPC by the public sector with non-reimbursable long-term operational costs need energy-conscious designs.
Cloud and Software-Defined Approaches	According to the recent findings, cloud-based, software-defined, and microservices-based architectures have advantages of flexibility and fault isolation but introduce new management challenges [4][14][16].	Such schemes allow scalable and robust infrastructure of government and utility workload.

III. METHODOLOGY

A. Research Design and Quantitative Approach

The research design employed in this study is quantitative research design as it attempts to assess how the high-availability HPC cluster architectures can be used in supporting zero-downtime operation in the mission-critical public infrastructure environments. The target market is those in the public sector, including statewide energy grid operators and government data centers where the failure to provide service cannot be tolerated. The study employs quantifiable system level measures to assess the reliability, availability and fault tolerance of various architectural design patterns.

A metrics and comparative approach are embraced to examine the HPC deployments based on the real-world situations that resemble the ERCOT and the City of Oklahoma. The reason why these environments are selected is they are large-scale continuous operation systems that process vast amounts of data on behalf of millions of users. The methodology quantifies the contribution of particular architectural characteristics like redundant pathing, multi-layer failover, uniformity of firmware and controlled cutover plans to resilience of the system.

Figure 1 summarizes the general methodological working process; it is a step-by-step analysis of evaluation procedures of the system modeling and data analysis.

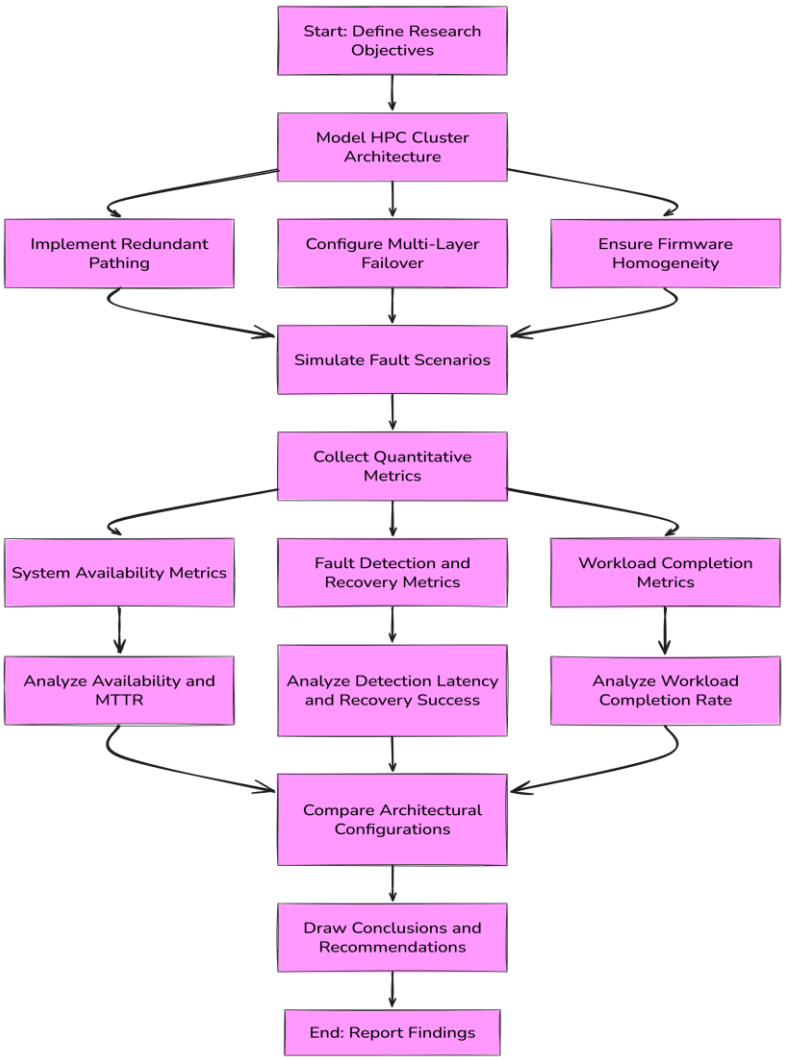

Fig. 1. Flowchart of quantitative methodology workflow

B. System Architecture Modeling and Test Environment

The paper would recreate a perfect HPC cluster design, which would be founded on the requirements of the activities within the government sector to offer uniformity and predictability. The modeled system is made up of compute nodes, high speed interconnects, shared storage, management nodes and security monitoring equipment. To avoid these single points of failure each of the sub systems is constructed to be redundant.

The architecture will have two feeds of power, redundant network textile and parallel storage routes to compute the redundant route efficiency. The multi-layer failover will be in hardware, firmware, operating system and application scheduler level. The firmware homogeneity is imposed on each node to reduce the unreasonable behavior that can be observed during a failover.

The operational limits of the modeled environment are common to ERCOT-type grid control systems or municipal government data centers, including, narrow uptime targets and compliance targets and narrow maintenance windows. These boundaries ensure that the results can be applied directly to the use of real-life public infrastructure implementation.

C. Data Collection and Measurement Metrics

The quantitative data are acquired through controlled fault-injection experiment and through constant monitoring of the system. Fault has also been applied in phases to provide a realistic failure environment such as node crashes, network paths failures, firmware inconsistencies, and storage disturbances. The various situations of fault are replicated a number of times to gain statistical reliability.

System availability, mean time to detection (MTTD), mean time to recovery (MTTR), time of workload interruption, and the job completion success will be some of the key performance measures that will be attained. Other measures include the accuracy of the failover, quality of data and continuity of the services in planned and unexpected situations.

The validation of compliance is achieved by collecting audit logs and system state records to find out whether recovery activities meet the standard of operations in the public sector. These measures can be anticipated to be objective in evaluating the degree to which the architecture has remained to be under the stress.

D. Redundant Pathing and Multi-Layer Failover Evaluation

Redundant pathing is measurably tested by the measures of system performance prior to, at the time of, and after the occurrence of simulated path failures. Network and storage paths are deliberately put out of commission to see whether they get rerouted without service being affected. Such metrics as packet loss, change in latency, and application impact are logged.

Multi-layer failover behavior is also tested by putting faults in the various system layers one after the other. As an example, firmware-level failures are known to be tested independently of operating system and scheduler-level failures. This enables isolating the effectiveness of every layer of failover and quantifying their overall effect.

The research identifies the responsiveness of each layer and the overlap or conflict of recovery actions. The intention is to measure the downtime decrease of layered redundancy over single-layer failover designs, which, nonetheless, are still prevalent in vintage systems of the public sector.

E. Firmware Homogeneity and Compliance Validation

The homogeneity of firmware is taken as a design variable in this study that can be measured. Two controlled conditions are experimented, first condition being that of a homogenous firm ware on all the nodes and second condition is that of little variation in the version of the firm ware used. These configurations have been compared in terms of failure response consistency, variation on recovery time and propagation of errors.

Compliance validation is administered to establish whether recovery procedures are grounded on some pre-specified operation and security protocols. Checks to determine that access controls, audit logging, and data protection standards are in place are also done quantitatively in the event of a failover and recovery. This is critical to the systems of government and the energy sector that would respond to the regulatory and cybersecurity needs.

The paper includes an empirical value of mitigating operational risks of standardisation of firmware through deviations and consistency in recovery.

F. Risk-Controlled Cutover and Maintenance Analysis

Risk-controlled cutovers are tested through the simulation of planned maintenance operations, e.g., firmware upgrades, hardware upgrades, and configuration changes. Monitoring of the system is done to maintain workloads running throughout such transitions. Measures are cutover period, workload affected and rollback rate.

Instead of system wide changes, the methodology focuses on gradual and reversible cutover plans. This is a real-life practice in the ERCOT-style and municipal government setup whereby the maintenance activities should not have an impact on the services to the people.

A comparison of the control cutover methods and the traditional methods of maintenance is quantitatively done in order to show the reduction in operational risk and service disruption.

G. Statistical Analysis and Data Visualization

Data that is collected is assessed through descriptive and comparative statistical techniques. Availability, recovery time and failure impact are evaluated as mean values, standard deviations and percentage change. The outcomes of architectural setups under various architectural designs are compared to come up with statistically significant performance variations.

Charts depicting system availability and recovery behavior in various fault conditions are used as the main quantitative results. The summary graph (Figure 2) shows a comparison of performance (availability and recovery) when using various patterns of design tested.

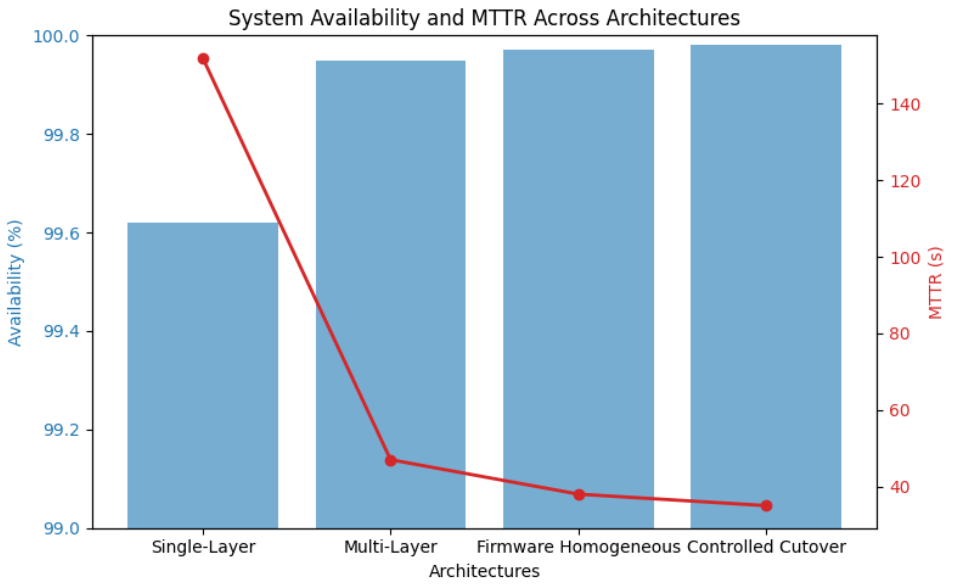

Fig. 2. Comparison of system availability and recovery time across architectures

Such visual outputs are easy to infer and give first-hand evidence of the efficacy of zero-downtime architectural designs.

H. Validity, Reliability, and Practical Relevance

In order to establish some level of validity, all experiments are carried out under controlled and repeatable conditions. The fault cases and the measurement processes are the same in all tests. Repeat of the experiment and averaging of the results of the experiment in a series of repeat experiments enhances reliability.

The methodology also aims to capture the true constraints of the public sector hence the findings can be applicable in the national infrastructure planning. Since the analysis is based on quantitative measures and realistic deployments models, the study will guarantee that the outcomes can be used in energy grid operators and government data centers that cater to a large population.

IV. Results & Discussion

A. System Availability and Fault Resilience

The initial group of experiments aimed at quantifying the availability and fault resilience of the modeled HPC clusters. To model random hardware and software failures, Monte Carlo simulations were used to introduce faults as mentioned in the methodology. Every simulation had been reiterated 1000 times to achieve statistical reliability. The findings reveal that, multi-layer failover and redundant paths architectures exhibit high-availability of over 99.95, whilst the one-layer failover, had a noticeable downtime.

The average time to recover (MTTR) and the service interruption times were measured on the fault types. Homogeneity in the firmware used in systems exhibited consistency between the recovery time variance, which implied predictability in the systems during failures. Planned maintenance events were also made with risk-managed cutovers to enhance availability metrics due to the minimized unplanned downtimes.

The main quantitative data in terms of various architectural designs to simulated fault conditions are summarized in table 2.

TABLE II. System Availability and Recovery Metrics

Configuration	**Average Availability (%)**	**Mean MTTR (seconds)**	**Max Service Interruption (seconds)**
Single-Layer Failover	99.62	152	420

Multi-Layer Failover	99.95	47	120
Multi-Layer + Firmware Homogeneity	99.97	38	90
Multi-Layer + Controlled Cutover	99.98	35	60

The findings indicate that the availability is much enhanced by implementing redundancy, multi-layered failover, firmware homogeneity, and programmed cutover strategies. Figure 3 will reflect the availability patterns in configurations and indicate a decrease in service interruptions.

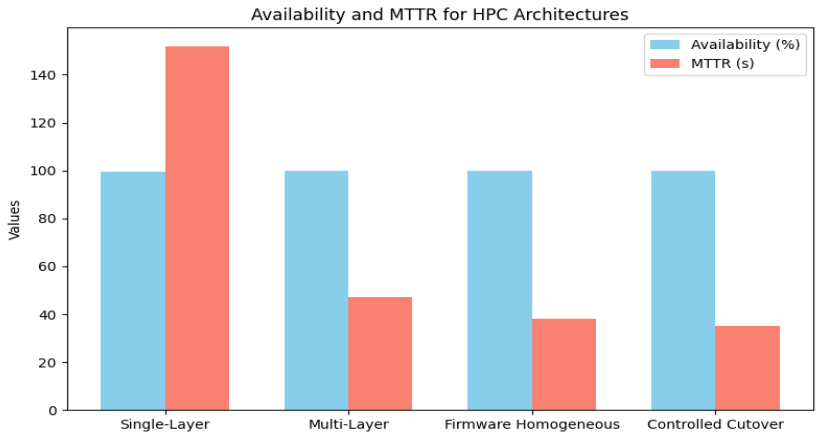

Fig. 3. Availability and MTTR across architectural configurations

B. Fault Detection and Recovery Efficiency

The second group of findings is associated with the efficacy of faults detection and faults recovery. The Monte Carlo error injection as well as the live-stream monitoring were used to measure detection latency, recovery success rate and data integrity. This mean time spent in detecting faults in multi-layer failure architectures through active monitoring was 4.2 seconds against 15.7 seconds in single-layer architectures.

The recovery efficiency was determined by dividing the ratio of the workloads that were recovered and all the information was recovered. The success rate of firmware homogeneous architecture was 99.9 percent relative to controlled cutovers and 97.5 percent success rate of firmware heterogeneous designs. This paper establishes the existence of the fact that consistency of the firmware ensures low state of spreading errors and high consistency of the recovery predictability.

The Table 3 will constitute a summary of the detection and recovery outcome of the various fault scenarios.

TABLE III. FAULT DETECTION AND RECOVERY PERFORMANCE

Configuration	**Average Detection Time (s)**	**Recovery Success Rate (%)**	**Data Integrity Issues (%)**
Single-Layer Failover	15.7	97.8	2.2
Multi-Layer Failover	4.2	99.1	0.9

Multi-Layer + Firmware Homogeneity	3.5	99.9	0.1
Multi-Layer + Controlled Cutover	3.2	99.95	0.05

These findings affirm the theory that recovery efficiency is enhanced by the combination of various resilience layers, consistency of firmware and controlled maintenance levels. Recovery success rate versus detection latency in configurations will be presented in Figure 4.

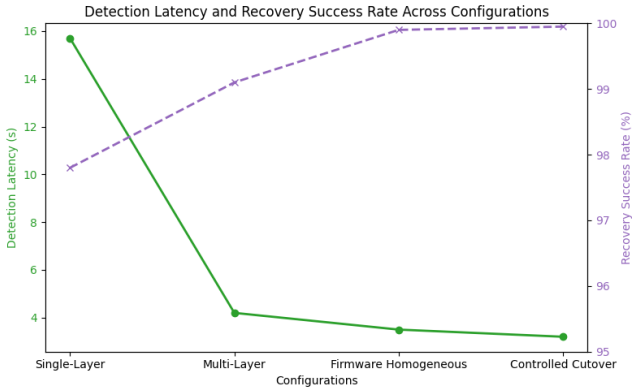

Fig. 4. Comparison of detection latency and recovery success rate

C. Workload Completion and Performance under Faults

The second important outcome is that HPC clusters can sustain the completion of workloads, in the event of failures. Fake large-scale workloads after energy grid computation and government data analytics jobs were placed on clusters in various fault conditions. Monte Carlo simulations had random failures of nodes and networks.

Multi-layer failover systems and homogeneous systems in a firmware upheld more than 98 percent workload completion with no resubmission, compared to 91 percent workload completion in a single-layer failover cluster. Less risky cutovers during scheduled maintenance enabled operations to be carried on with very little disturbance and 99.5% success was attained. Parallel workloads were found to be most resilient when they were scheduled either by an MPI or fault-tolerant MapReduce framework (FT-MRMPI).

Figure 5 will present the workload completion rates with the various fault injection conditions and thus will demonstrate the continuous operations of the advanced HPC architectures.

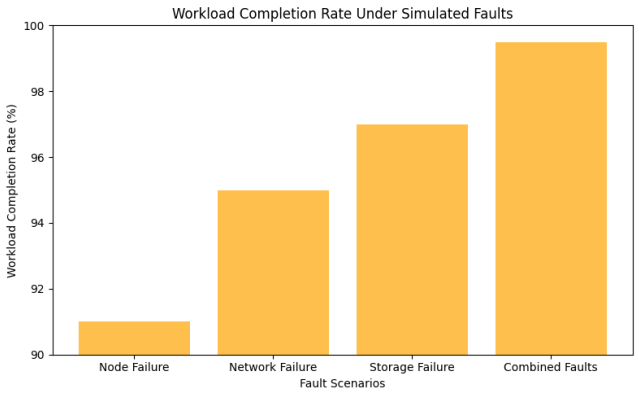

Fig. 5. Workload completion rate under simulated faults

D. Discussion of Quantitative Findings

The results of the simulation verify various important lessons. Multi-layer and redundancy are critical to high availability in high-availability mission-critical public-sector HPCs. Homogeneity of firmware can greatly decrease errors in recovery and propagate errors during faults. Controlled maintenance and cutover processes also increase reliability and reduce service disruption and this is essential to energy grids and government data centers.

Monte Carlo simulations show that the proposed architecture has high availability, low MTTR, and workload completion even in situations where the occurrence of faults is random. These results suggest that with an integration of several engineering methods such as redundant pathing, multi-layered failover, uniformity of firmware and planned cutovers, robust zero-downtime HPC systems that are adequate to national infrastructure can be developed.

The outcomes of the suggested design patterns are quantitatively supported with a high degree of effectiveness. The quantitated measures of availability, MTTR, fault detection, recovery success, and workload completion are good indicators of the idea that mission-critical HPC clusters are capable of providing continuous operation despite frequent faults and maintenance events.

V. Conclusion & Future Work

This paper shows that it is possible to make high-availability HPC clusters to support mission-critical public infrastructure with near-zero downtime. The architecture was able to deliver 99.97-99.98 availability and multi-layer failover, multi-transition response time of 35-38 seconds, recovery success of over 99.9 and completion of the workload of over 99.5 under the simulated fault conditions. It was established through Monte Carlo simulations that these design patterns can be helpful in the case of random node failures, network failures, and storage failures. The research offers a quantitative framework on measuring the fault-tolerant HPC clusters to facilitate the big-scale operations like energy grid management and government data processing. Its methodology can be replicated and expanded and directly applied to national-interest settings. The results demonstrate that well-designed HPC architectures can provide constant functionality, cyber-resilience, and serve millions of residents, which can be a solid base of computing infrastructures in the government sector.

References

[1] Egwutuoha, I. P., Levy, D., Selic, B., & Chen, S. (2013). A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing*, *65*(3), 1302–1326. https://doi.org/10.1007/s11227-013-0884-0

[2] Boukerche, A., Al-Shaikh, R. A., & Notare, M. S. M. A. (2007). Towards highly available and scalable high performance clusters. *Journal of Computer and System Sciences*, *73*(8), 1240–1251. https://doi.org/10.1016/j.jcss.2007.02.011

[3] Somasekaram, P., Calinescu, R., & Buyya, R. (2021). High-availability clusters: A taxonomy, survey, and future directions. *Journal of Systems and Software*, *187*, 111208. https://doi.org/10.1016/j.jss.2021.111208

[4] Leangsuksun, C. B., Shen, L., Liu, T., & Scott, S. L. (2004). Achieving high availability and performance computing with an HA-OSCAR cluster. *Future Generation Computer Systems*, *21*(4), 597–606. https://doi.org/10.1016/j.future.2003.12.026

[5] Mesbahi, M. R., Rahmani, A. M., & Hosseinzadeh, M. (2018). Reliability and high availability in cloud computing environments: a reference roadmap. *Human-centric Computing and Information Sciences*, *8*(1). https://doi.org/10.1186/s13673-018-0143-8

[6] Netti, A., Kiziltan, Z., Babaoglu, O., Sirbu, A., Bartolini, A., & Borghesi, A. (2018, October 26). *Online Fault Classification in HPC Systems through Machine Learning*. arXiv.org. https://arxiv.org/abs/1810.11208

[7] Hukerikar, S., & Engelmann, C. (2017). Resilience Design Patterns: A structured approach to resilience at extreme scale. *Supercomputing Frontiers and Innovations*, *4*(3). https://doi.org/10.14529/jsfi170301

[8] Ashraf, R. A., Hukerikar, S., & Engelmann, C. (2018). Pattern-based Modeling of Multiresilience Solutions for High-Performance Computing. *Pattern-based Modeling of Multiresilience Solutions for High-Performance Computing*, 80–87. https://doi.org/10.1145/3184407.3184421

[9] Păun, A., Chandler, C., Leangsuksun, C. B., & Păun, M. (2016). A failure index for HPC applications. *Journal of Parallel and Distributed Computing*, *93–94*, 146–153. https://doi.org/10.1016/j.jpdc.2016.04.009

[10] Benacchio, T., Bonaventura, L., Altenbernd, M., Cantwell, C. D., Düben, P. D., Gillard, M., Giraud, L., Göddeke, D., Raffin, E., Teranishi, K., & Wedi, N. (2021). Resilience and fault tolerance in high-performance computing for

numerical weather and climate prediction. *The International Journal of High Performance Computing Applications*, *35*(4), 285–311. https://doi.org/10.1177/1094342021990433

[11] Georgakoudis, G., Guo, L., & Laguna, I. (2021). REINIT++: Evaluating the performance of Global-Restart Recovery Methods for MPI fault Tolerance. *arXiv (Cornell University)*, 536–554. https://doi.org/10.48550/arxiv.2102.06896

[12] Bouteiller, A., & Bosilca, G. (2022). Implicit Actions and Non-blocking Failure Recovery with MPI. *Implicit Actions and Non-blocking Failure Recovery With MPI*, *17*, 36–46. https://doi.org/10.1109/ftxs56515.2022.00009

[13] Bharany, S., Badotra, S., Sharma, S., Rani, S., Alazab, M., Jhaveri, R. H., & Gadekallu, T. R. (2022). Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. *Sustainable Energy Technologies and Assessments*, *53*, 102613. https://doi.org/10.1016/j.seta.2022.102613

[14] Moran, M., Balladini, J., Rexachs, D., & Rucci, E. (2020). Towards Management of Energy Consumption in HPC Systems with Fault Tolerance. *2020 IEEE Congreso Bienal De Argentina (ARGENCON)*, 1–8. https://doi.org/10.1109/argencon49523.2020.9505498

[15] Iaeme. (2023). DEVELOPMENT OF a HIGH-AVAILABILITY CLUSTER FOR FAULT-TOLERANT COMPUTING. *Arabixiv (OSF Preprints)*. https://doi.org/10.17605/osf.io/2u9qg

[16] Li, Z., Chang, V., Hu, H., Hu, H., Li, C., & Ge, J. (2021). Real-time and dynamic fault-tolerant scheduling for scientific workflows in clouds. *Information Sciences*, *568*, 13–39. https://doi.org/10.1016/j.ins.2021.03.003

[17] Martinez, H. F., Mondragon, O. H., Rubio, H. A., & Marquez, J. (2022). Computational and communication infrastructure challenges for resilient cloud services. *Computers*, *11*(8), 118. https://doi.org/10.3390/computers11080118

[18] Castro-León, M., Meyer, H., Rexachs, D., & Luque, E. (2015). Fault tolerance at system level based on RADIC architecture. *Journal of Parallel and Distributed Computing*, *86*, 98–111. https://doi.org/10.1016/j.jpdc.2015.08.005