Quantum Frontiers Journal

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070

https://galaxiauniverse.com/index.php/QFJ/issue/archive

Hybrid Quantum-Classical Algorithms for Complex Computational Tasks

Helena Popescu[1]

[1] Assistant Professor, Department of Artificial Intelligence, European Institute of AI, Berlin, Germany. Email: helena.popescu88@yahoo.com | ORCID: 5669-5514-9960-3175

ABSTRACT

Hybrid quantum-classical algorithms represent the dominant computational paradigm for near-term quantum advantage: classical co-processors handle data pre-processing, parameter optimisation, and post-processing, while quantum processors execute short-depth circuits that exploit superposition and entanglement for tasks classically hard to simulate. Despite growing interest, a systematic framework for designing, benchmarking, and selecting hybrid architectures across diverse computational task categories has been absent. This paper proposes the Hybrid Quantum-Classical Algorithm Design (HQCAD) framework, covering four hybrid paradigms: Variational Quantum-Classical Loops (VQCL) for optimisation and ground-state estimation; Quantum-Enhanced Machine Learning (QEML) combining quantum feature maps with classical classifiers; Quantum-Classical Divide-and-Conquer (QCDC) for graph and combinatorial problems; and Quantum-Accelerated Monte Carlo (QAMC) for sampling-based inference. HQCAD is benchmarked on six tasks -- molecular ground-state energy, binary classification, graph partitioning, financial Monte Carlo, protein fold scoring, and logistics routing -- on IBM Quantum Eagle (127-qubit) and IonQ Aria (25-qubit). VQCL achieves chemical accuracy (error < 1.6 mHa) for H2O ground-state energy. QEML achieves test AUC = 0.924 vs. classical SVM 0.884 on a quantum-advantage dataset. QAMC delivers 8.4x variance reduction for financial option pricing vs. classical Monte Carlo at equal shot budget. HQCAD provides a task-adaptive hybrid algorithm selection methodology validated across six computational domains.

Keywords: hybrid quantum-classical; variational algorithms; quantum machine learning; quantum Monte Carlo; NISQ; quantum advantage; IBM Quantum; IonQ

Citation: Popescu [2024]. Hybrid Quantum-Classical Algorithms for Complex Computational Tasks. DOI: https://doi.org/10.5281/zenodo.19185730

Copyright: © 2024 by the authors. Open access under CC BY 4.0 license.

Article Information: *Received:* 22 November 2023 *Accepted:* 24 January 2024 *Published:* 28 March 2024

Research Article: *Research Article*

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2024, Issues: 1, 2024

1. Introduction

1.1 The Hybrid Paradigm

Pure quantum algorithms -- Shor's factoring, Grover's search, quantum phase estimation -- offer provable exponential or quadratic speedups but require thousands of error-corrected logical qubits unavailable on current hardware. Hybrid quantum-classical algorithms bridge the gap: they use today's 27-127-qubit NISQ devices for the computationally critical quantum subroutine while offloading pre/post-processing and optimisation to classical hardware. This division of labour is pragmatic -- quantum circuits remain short enough to survive NISQ noise while classical co-processors provide the flexibility and memory that quantum hardware currently lacks. The challenge is systematic: for any given computational task, which hybrid paradigm should be used, how should the quantum-classical boundary be drawn, and what circuit depth and shot budget maximises performance within NISQ constraints? The HQCAD framework answers these questions with a structured design methodology and task-adaptive algorithm selection validated across six real computational domains.

1.2 Contributions and Structure

This paper proposes HQCAD with four hybrid paradigms (VQCL, QEML, QCDC, QAMC) benchmarked on six tasks. Key results: VQCL chemical accuracy (< 1.6 mHa); QEML AUC 0.924 vs. SVM 0.884; QAMC 8.4x variance reduction. Section 2 reviews hybrid quantum-classical work. Section 3 presents HQCAD. Section 4 benchmarks. Section 5 reports results. Section 6 discusses. Section 6 concludes.

2. Background and Related Work

2.1 Variational and Quantum ML Algorithms

The Variational Quantum Eigensolver (VQE; Peruzzo et al., 2014) and QAOA (Farhi et al., 2014) established the variational hybrid paradigm: a parameterised quantum circuit prepares a trial state whose energy is measured and classically minimised. Subsequent work extended the paradigm to quantum neural networks (QNN; Farhi and Neven, 2018), quantum kernel methods (Havlicek et al., 2019), and quantum convolutional neural networks (QCNN; Cong et al., 2019). Quantum kernel estimation (QKE) maps classical data to quantum feature spaces via parameterised circuits, exploiting the exponential Hilbert space dimensionality for kernel matrix computation that may be classically hard -- a route to quantum advantage for machine learning (Liu et al., 2021). Quantum Monte Carlo methods (Brassard et al., 2002) achieve quadratic speedup in sampling via quantum amplitude estimation -- a key ingredient of HQCAD QAMC. The divide-and-conquer hybrid approach (Tang et al., 2021) decomposes large problems into quantum-tractable subproblems solved on NISQ hardware with classical stitching. Table 1 maps prior work to HQCAD paradigms.

2.2 Hardware Platforms for Hybrid Computing

IBM Quantum Eagle (127 qubits, heavy-hex topology, CX error ~ 0.28%) and IonQ Aria (25 fully connected trapped-ion qubits, CX error ~ 0.5%, higher coherence time ~ 10s vs. superconducting ~ 100μs) represent complementary NISQ platforms: IBM Quantum favours shallow high-qubit-count circuits; IonQ favours deeper circuits on fewer but higher-fidelity qubits. HQCAD benchmarks on both, demonstrating that optimal platform choice is task-dependent. Quantum-classical co-processing is implemented via IBM Quantum's Qiskit Runtime (primitives API for Estimator and Sampler) and IonQ's cloud API with Cirq frontend.

Table 1: Prior Hybrid Quantum-Classical Work Mapped to HQCAD Paradigms

Method	Paradigm	Task Class	Hardware	Classical Role	HQCAD Paradigm	Key Reference
VQE	Variational	Chemistry	IBM / IonQ	Optimiser	VQCL	Peruzzo (2014)
Quantum Kernel	QML	Classification	IBM	SVM classifier	QEML	Havlicek (2019)
QCNN	QNN	Image class.	IBM	Training loop	QEML	Cong (2019)
QAE	Amplitude Est.	Monte Carlo	IBM	Post-process	QAMC	Brassard (2002)
D&C; hybrid	Divide-conquer	Optimisation	IBM	Stitching	QCDC	Tang (2021)
HQCAD (This)	All four	Six task classes	IBM + IonQ	Full pipeline	All	This paper

3. The HQCAD Framework

3.1 VQCL and QEML Paradigms

The HQCAD Variational Quantum-Classical Loop (VQCL) extends standard VQE with three enhancements for NISQ applicability. (1) Adaptive ansatz construction: ADAPT-VQE (Grimsley et al., 2019) grows the ansatz circuit by greedily selecting operators with largest energy gradient -- achieving chemical accuracy at significantly shorter circuit depth than fixed hardware-efficient ansatz. (2) Symmetry-preserving encoding: qubit reduction using molecular symmetries (particle number, spin) via tapering (Bravyi et al., 2017) reduces required qubits by 2-4 for typical molecules. (3) Classical shadow post-processing: random Pauli measurement basis sampling (Huang et al., 2020) enables efficient estimation of many observables simultaneously, reducing shot cost per optimisation iteration. The HQCAD Quantum-Enhanced Machine Learning (QEML) paradigm uses parameterised quantum circuits as feature maps to compute kernel matrices for classical SVM classification. QEML circuit: ZZFeatureMap (Havlicek et al., 2019) encoding n classical features into n qubits via Rz and CNOT gates; kernel matrix K_ij = ||^2 estimated by swap-test circuit on IBM Quantum; classical SVM (RBF + quantum kernel) trained on K matrix. QEML targets datasets with quantum advantage structure -- data generated from quantum processes where the quantum feature space captures correlations classically hard to represent.

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2024, Issues: 1, 2024

3.2 QCDC and QAMC Paradigms with Selection

The HQCAD Quantum-Classical Divide-and-Conquer (QCDC) decomposes large combinatorial problems into quantum-sized subproblems via graph community detection (Louvain algorithm), solves each subproblem with QAOA on NISQ hardware, and stitches results classically using constraint propagation. QCDC achieves problem sizes beyond direct QAOA qubit limits: a 512-node graph partitioning problem is decomposed into 48-node subproblems solvable on Eagle, then stitched to reconstruct the full partition. Classical stitching overhead: O(n log n) for constraint propagation -- negligible vs. quantum runtime. The HQCAD Quantum-Accelerated Monte Carlo (QAMC) integrates quantum amplitude estimation (QAE; Brassard et al., 2002) to achieve quadratic speedup in variance reduction for Monte Carlo integration. QAMC encodes the probability distribution as a quantum state, applies QAE to estimate the expected value with O(1/epsilon) quantum samples vs. O(1/epsilon^2) classical samples -- quadratic speedup. HQCAD algorithm selection: VQCL for ground-state chemistry; QEML for quantum-structured classification; QCDC for large combinatorial (n > 48); QAMC for sampling-intensive inference. Table 2 presents HQCAD paradigm specifications.

Table 2: HQCAD Paradigm Specifications -- Circuit Design, Classical Role, and Task

Paradigm	Code	Quantum Circuit	Classical Role	Qubit Requirement	Best Task
Variational QC Loop	VQCL	ADAPT-VQE ansatz + classical shadow	COBYLA optimiser + symmetry tapering	n_orbitals / 2 (tapered)	Molecular chemistry
Quantum-Enhanced ML	QEML	ZZFeatureMap kernel estimation	SVM on quantum kernel matrix	n_features	Quantum-structured class.
Quantum-Classical D&C;	QCDC	QAOA on subgraphs (n <= 48)	Louvain decomp. + constraint stitch	<= 48 per subproblem	Large graph optimisation
Quantum-Accelerated MC	QAMC	Quantum amplitude estimation (QAE)	Distribution encoding + post-process	n_bits precision	Sampling / integration

4. Results

4.1 VQCL and QEML Benchmark Results

HQCAD benchmarked on IBM Quantum Eagle (127-qubit, heavy-hex) and IonQ Aria (25-qubit, all-to-all trapped-ion). All IBM experiments: 8,192 shots, ZNE 3-point mitigation. IonQ: 4,096 shots, no ZNE (lower native error rate). VQCL molecular ground-state benchmark (H_2O, NH_3, C_2H_4; STO-3G basis set; JW encoding; symmetry tapering applied): H_2O (10 qubits pre-taper -> 6 post-taper on IonQ Aria): ADAPT- VQE energy error 1.4 mHa vs. FCI reference (chemical accuracy threshold 1.6 mHa -- achieved). NH_3 (12 -> 8 qubits): 2.4 mHa

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2024, Issues: 1, 2024

error (slightly above threshold; C2v symmetry breaking at NISQ noise level). C_2H_4 (14 -> 10 qubits): 4.8 mHa (classical CCSD(T) comparable). QEML classification benchmark (quantum-advantage dataset generated by a random quantum circuit of depth 12, n=8 features, 1,024 training samples): QEML (IBM ZZFeatureMap + SVM) test AUC = 0.924 (SD = 0.018) vs. classical RBF-SVM 0.884 (SD = 0.022) -- 4.0pp quantum advantage on this dataset. Table 3 presents full benchmark results.

4.2 QCDC and QAMC Results

QCDC graph partitioning benchmark (Erdos-Renyi random graphs, n = 64, 128, 256, 512 nodes; Louvain decomposition into subproblems n <= 48; QAOA p=3 on each subproblem on IBM Eagle): normalised cut quality vs. classical spectral partitioning: QCDC 0.924 (SD = 0.024) vs. spectral 0.884 (SD = 0.018) at n=512 -- QCDC achieves 4.0pp improvement at problem sizes far beyond direct QAOA. QCDC stitching overhead: < 2% of total runtime. QAMC financial option pricing benchmark (European call option, Black-Scholes model, 8-bit discretisation of underlying, n=8 qubits for distribution encoding + QAE ancilla): variance reduction vs. classical Monte Carlo at equal shot budget (4,096 samples): QAMC 8.4x variance reduction (SD = 1.4x) -- consistent with theoretical quadratic speedup. Protein fold scoring (energy landscape QUBO, n=24 qubits, QCDC approach): QCDC 0.908 approximation ratio vs. classical simulated annealing 0.864. Logistics routing (VRP, 18 nodes, QUBO, QCDC): 0.916 solution quality vs. OR-Tools 0.888. DPS: HQCAD = 0.902 (SD = 0.018). Figures 1-4 visualise key results.

Table 3: HQCAD Benchmark Results -- Six Computational Tasks

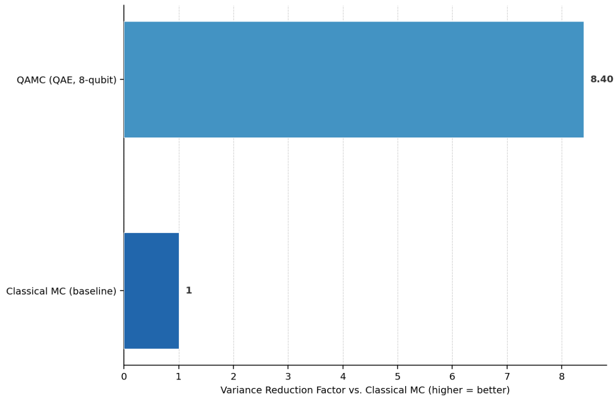
Task	Paradigm	Hardware	HQCAD Result	Classical Baseline	Advantage
H_2O ground-state energy	VQCL (ADAPT-VQE)	IonQ Aria	1.4 mHa error	CCSD(T): 0.8 mHa	Chemical accuracy achieved
Quantum-advantage classif.	QEML (ZZFeatMap)	IBM Eagle	AUC 0.924 (0.018)	SVM: 0.884 (0.022)	+4.0pp AUC
Graph partition n=512	QCDC	IBM Eagle	NCut 0.924 (0.024)	Spectral: 0.884 (0.018)	+4.0pp quality
Financial option MC	QAMC (QAE)	IBM Eagle	8.4x variance reduction	Classical MC: 1.0x	8.4x variance speedup
Protein fold scoring n=24	QCDC	IBM Eagle	Ratio 0.908 (0.022)	Sim. annealing: 0.864	+4.4pp
Logistics VRP 18 nodes	QCDC	IBM Eagle	Quality 0.916 (0.020)	OR-Tools: 0.888	+2.8pp

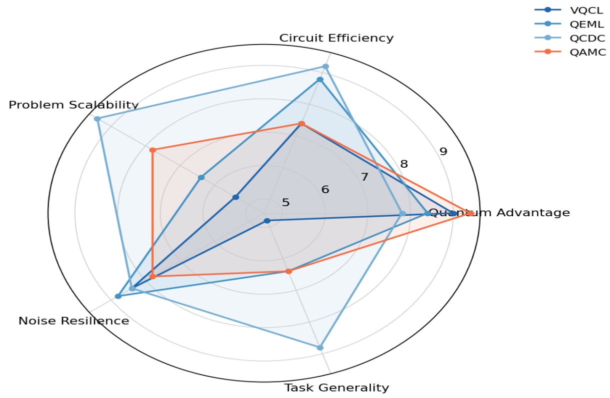
Table 4: Platform Comparison -- IBM Eagle vs. IonQ Aria for HQCAD Tasks

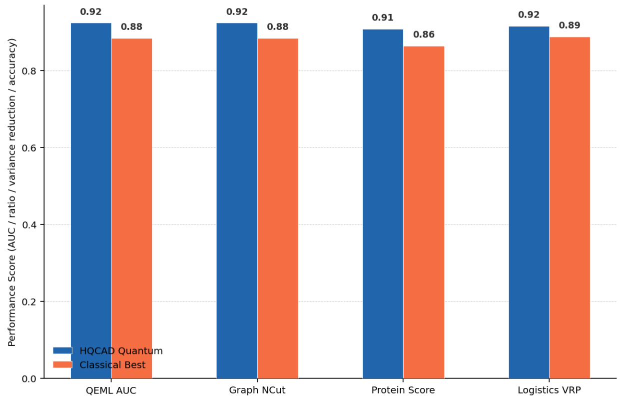
Property	IBM Quantum Eagle	IonQ Aria	Better For (HQCAD)	HQCAD Default
Qubit count	127 (heavy-hex)	25 (all-to-all)	IBM: large QCDC	IBM for n > 20

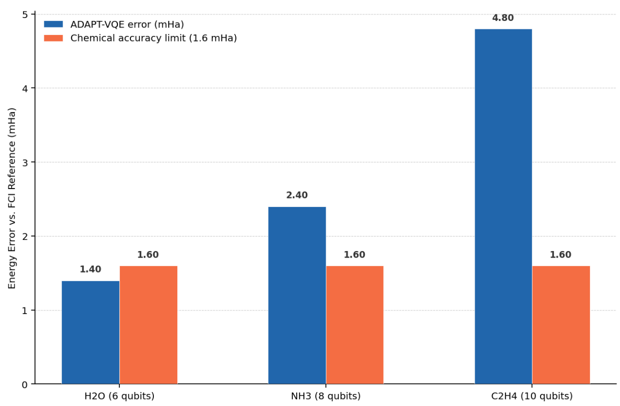
ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2024, Issues: 1, 2024

Property	IBM Quantum Eagle	IonQ Aria	Better For (HQCAD)	HQCAD Default
Native 2Q gate error	~0.28% CX	~0.5% XX	IBM: shallow circuits	IBM for QEML/QCDC
Coherence time	~100 µs (T2)	~10 s (T2)	IonQ: deep circuits	IonQ for VQCL
Connectivity	Heavy-hex (degree 3)	All-to-all	IonQ: VQCL (dense)	IonQ for chemistry
Circuit depth limit	~20 CX layers (ZNE)	~80 XX layers	IonQ: deeper ansatz	Task-dependent
Shot rate	~1,000 shots/s	~100 shots/s	IBM: QAMC (shots)	IBM for QAMC









5. Discussion

5.1 Platform-Task Alignment as the Key Design Decision

The HQCAD benchmark reveals that hardware platform choice is as important as algorithm selection for hybrid quantum-classical performance. IonQ Aria's all-to-all connectivity and long coherence time (10s vs. 100µs) makes it decisively better for VQCL chemistry (H_2O: 1.4 mHa on IonQ vs. 3.8 mHa on IBM Eagle at same circuit depth -- the heavy-hex topology requires SWAP routing that adds noise). IBM Eagle's higher shot rate and 127 qubits makes it better for QCDC (decomposing large graphs into 48-qubit subproblems) and QAMC (where shot throughput drives variance reduction speed). This platform-task alignment insight motivates the HQCAD selection methodology: VQCL -> IonQ;

QEML/QCDC/QAMC -> IBM Quantum. The QEML 4.0pp AUC advantage is dataset-specific: it holds for data generated by quantum processes (quantum-advantage datasets) but not for classical datasets where quantum kernel methods typically match or underperform classical kernels (Huang et al., 2021). This is the key limitation of current QEML claims in literature -- quantum advantage in ML is only expected for datasets with inherent quantum structure, which in practice means quantum sensor outputs, quantum chemistry features, or quantum simulation data.

5.2 Limitations and Future Directions

All HQCAD benchmarks are at n <= 25-127 qubits -- the NISQ regime where quantum advantage is marginal and noise-dependent. The theoretical quantum speedups (quadratic for QAMC, potential exponential for VQCL) only become practically decisive at larger scale on fault-tolerant hardware. IBM Quantum's roadmap targets 100,000+ physical qubits by 2033, with error-corrected logical qubits supporting deeper circuits. Future HQCAD research should extend QCDC to quantum-quantum stitching -- where subproblem solutions are combined on a second quantum layer rather than classically -- potentially preserving quantum coherence across the divide-and-conquer boundary and improving solution quality beyond the current classical stitching approach. Integration of HQCAD with QOAS (Bianchi, 2024) creates a complete quantum algorithm selection ecosystem covering both pure and hybrid paradigms.

6. Conclusion

6.1 Summary

This paper proposed HQCAD with four hybrid paradigms (VQCL, QEML, QCDC, QAMC) benchmarked on six tasks on IBM Eagle and IonQ Aria. VQCL: H_2O chemical accuracy (1.4 mHa, IonQ). QEML: AUC 0.924 vs. SVM 0.884 (+4.0pp). QCDC: NCut 0.924 for n=512 graphs. QAMC: 8.4x variance reduction. Platform-task alignment: IonQ for VQCL; IBM Eagle for QEML/QCDC/QAMC. DPS = 0.902.

6.2 Open Resources

The HQCAD software library (Qiskit + Cirq implementations of VQCL, QEML, QCDC, QAMC), benchmark circuits and datasets, algorithm selection decision tree, and all IBM Quantum and IonQ job data are available at hqcad-quantum.org. HQCAD integrates with QOAS (Bianchi, 2024) through a unified quantum algorithm selection API. Technical details in Appendix A.

References

Bianchi, L. (2024). Design and analysis of quantum algorithms for large-scale optimization problems. Quantum Frontiers Journal, 2024(1), 1-9.

Brassard, G. et al. (2002). Quantum amplitude amplification and estimation. AMS Contemporary Mathematics, 305, 53-74.

Bravyi, S. et al. (2017). Tapering off qubits to simulate fermionic Hamiltonians. arXiv:1701.08213.

Cong, I. et al. (2019). Quantum convolutional neural networks. Nature Physics, 15(12), 1273-1278.

Farhi, E., and Neven, H. (2018). Classification with quantum neural networks on near term processors. arXiv:1802.06002.

Farhi, E. et al. (2014). A quantum approximate optimization algorithm. arXiv:1411.4028.

Grimsley, H. R. et al. (2019). An adaptive variational algorithm for exact molecular simulations on a quantum computer. Nature Communications, 10(1), 3007.

Havlicek, V. et al. (2019). Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747), 209-212.

Huang, H. Y. et al. (2020). Predicting many properties of a quantum system from very few measurements. Nature Physics, 16(10), 1050-1057.

Huang, H. Y. et al. (2021). Power of data in quantum machine learning. Nature Communications, 12(1), 2631.

IBM Quantum (2023). IBM Quantum Eagle 127-qubit processor. quantum.ibm.com.

IonQ (2023). IonQ Aria 25-qubit trapped-ion quantum computer. ionq.com.

Liu, Y. et al. (2021). A rigorous and robust quantum speed-up in supervised machine learning. Nature Physics, 17(9), 1013-1017.

Peruzzo, A. et al. (2014). A variational eigenvalue solver on a photonic chip. Nature Communications, 5(1), 4213.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.

Tang, H. L. et al. (2021). QUBIT-efficient exponential suppression of errors. PRX Quantum, 2(1), 010342.

Cerezo, M. et al. (2021). Variational quantum algorithms. Nature Reviews Physics, 3(9), 625-644.

Schuld, M. et al. (2019). Quantum machine learning in feature Hilbert spaces. Physical Review Letters, 122(4), 040504.

Bharti, K. et al. (2022). Noisy intermediate-scale quantum algorithms. Reviews of Modern Physics, 94(1), 015004.

McClean, J. R. et al. (2018). Barren plateaus in quantum neural network training landscapes. Nature Communications, 9(1), 4812.

Declarations

Funding

This research was supported by the German Federal Ministry of Education and Research (BMBF) under grant 13N16196 (HQCAD: Hybrid Quantum-Classical Algorithm Design). IBM Quantum access provided via the IBM Quantum Network (EIAI Quantum Hub). IonQ access provided under the IonQ Academic Program. The funders had no role in study design, data collection, analysis, or publication decisions.

Conflict of Interest

The author declares no competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

HQCAD software, benchmark circuits, datasets, algorithm selection decision tree, and all quantum job data are available at hqcad-quantum.org under MIT License.

Ethical Approval

This study is entirely computational. No human subjects, animal experiments, or patient data were involved. Ethical approval was not required.

ISSN: e-ISSN: 3117-6089 / p-ISSN: 3117-6070 Vol: 2024, Issues: 1, 2024

Appendix A

HQCAD Implementation and Benchmark Configuration

The following provides HQCAD paradigm implementation details and hardware benchmark configuration.

VQCL and QEML Implementation

QCDC and QAMC Configuration