Thermal-Aware Optimization of Distributed Cryogenic Computing Systems

Nurzhan Zhuldassov, Daniel Štefankovič, and Eby G. Friedman

Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627
nzhuldas@ur.rochester.edu

***Abstract*—Cryogenic computing systems, such as superconductive electronics and quantum processors, consist of multiple, independent functional components. These components, functioning across a wide temperature range from room temperature to millikelvin temperatures, exhibit diverse power and performance characteristics. Efficient thermal management of these systems is important to minimize power consumption and heat load while satisfying performance constraints.**

A methodology for the thermal optimization of multi-temperature cryogenic computing systems is presented. Unlike previous approaches that rely on predetermined refrigeration stages and fixed computing assignments per stage, the proposed method determines the optimal number of temperature zones and component assignments per zone based on local power and performance profiles. The total power consumption is minimized while satisfying delay constraints. The methodology employs graph theory and optimization techniques to accelerate the search process. Two example cryogenic computing systems are optimized as case studies to demonstrate the methodology. A thirteen times and six times reduction in power is achieved in these case studies; concluding, for these cases, that the components should be grouped into three refrigeration chambers, each operating at different temperatures.

***Index Terms*—Thermal optimization, cryoCMOS, SFQ, cryogenic computing systems, graph theory**

I. Introduction

The computing industry is advancing along several distinct trajectories, many of which operate at cryogenic temperatures. Two examples of these trajectories are superconductive digital computing systems and quantum computing systems [1], [2]. Superconductive digital electronics is a highly promising beyond-CMOS technology, offering both ultra-low power consumption and ultra-high speed [1], [3]. These systems operate at cryogenic temperatures [4], as superconductivity is achieved in niobium at temperatures below 9.3 K and frequently cooled to liquid helium temperature, 4.2 K.

Despite the low energy efficiency of operating at cryogenic temperatures in small scale applications, a significant increase in efficiency is observed as the complexity of the system being cooled increases. For example, at 4 K, the energy efficiency can range from 1% of Carnot efficiency [5] for small systems to as high as 35% for large scale liquefaction plants [6]. This enhancement in energy efficiency allows large scale cloud computing centers to operate at greater performance and higher energy efficiency [7].

These systems can be partitioned into multiple computing components (or units), each of which can be placed within a different temperature zone. Quantum computers, for example, apart from the quantum processor operating at millikelvin temperatures, may consist of single flux quantum (SFQ) circuitry [1], CMOS circuitry, and various support blocks such as phase locked loop oscillators and low noise amplifiers [8]. Distributed computing systems are composed of thousands or even millions of processors distributed across multiple data centers.

Each unit exhibits different power consumption profiles, performance characteristics, and thermal load requirements. The available cooling power per kilowatt input differs across temperatures [9]. Consequently, certain circuits may benefit from operating at lower temperatures, whereas other circuitry would achieve satisfactory performance at higher temperatures. Increasing energy efficiency at cryogenic temperatures combined with the higher cooling efficiency of large scale refrigeration results in higher power efficiency and improved performance as compared to operating at room temperature [7],

This research is supported in part by the National Science Foundation under Grant No. 2124453 (DISCOVER Expeditions) and Grant No. 2308863, and the Department of Energy under FOA number DE-FOA-0002950.

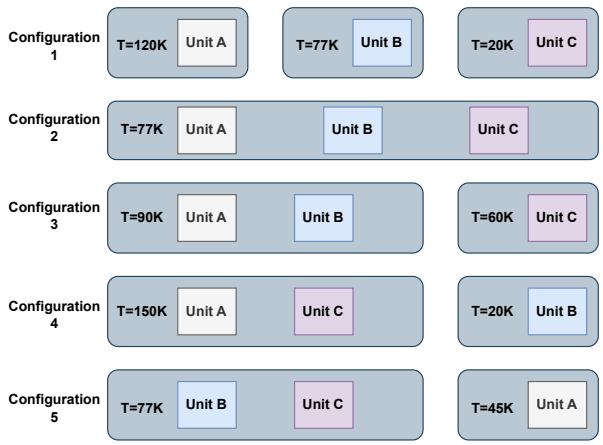

Fig. 1: Five different possible cooling configurations available for a three unit system

[10]. Previous approaches on managing this system have relied on a pre-determined number of cooling stages and unit configurations [9], [11].

The methodology proposed here addresses the limitations of earlier approaches by optimizing the number of refrigeration stages based on the specific requirements of the system. Rather than relying on fixed unit assignments per refrigeration stage, the proposed method introduces flexibility by enabling units to be grouped based on the local performance and power profiles, ensuring that each group operates at the most energy efficient temperature. Both the number of stages and the configuration of the units within those stages are determined based on performance constraints, while maximizing energy efficiency and minimizing power consumption. As an example, consider an idealized system composed of three units with five different configurations, as shown in Fig. 1. Each of these configurations can operate at a different set of temperatures, producing different performance and power consumption characteristics. The proposed methodology would determine the optimal configuration and set of temperatures.

The paper is organized as follows: A proposed solution to the temperature optimization problem is reviewed in Section II. A discussion of the algorithmic complexity is also presented, followed by a method to accelerate the algorithm. Two case studies, a hybrid superconductive/semiconductor distributed computing system and a quantum computing system, are optimized using the proposed methodology, as discussed in Section III. Some conclusions are offered in Section IV.

II. Methodology

Optimization of multi-temperature zone systems is achieved here through a graph theoretic approach [12]. The proposed methodology is divided into two primary phases: the construction of the graph of a system and determining the optimal path within the graph. In the first phase, the system, composed of multi-temperature zones, is divided into multiple units or groups of units, and the corresponding graph is constructed. A unit here represents an independent functional component of the system, characterized by a power and delay which varies according to the operating temperature. As depicted in Fig. 2, each node (A, B, and C) in the graph represents a group of units, while the edge weights (T_{ij}, p_{ij}, d_{ij}) represent an operating temperature T_{ij} for a specific group. Each edge is assigned two weights at a given temperature: power consumption p_{ij} and delay d_{ij}. The optimal path through the graph corresponds to the path that minimizes the total power consumption while satisfying a target delay constraint. The total power consumption and delay are, respectively, the sum of the power weights along a path, and the sum of the delay weights along a path. The power weight is a cost function which depends upon the temperature of the current and previous chambers and the power consumption of the unit, as described in the next subsection. In the second phase, the optimal path is determined using several optimization techniques, including graph pruning and dynamic programming (ϵ-dominance based partitioning [13]), to accelerate the search process. After determining the optimal path for all of the constructed graphs, the results are compared to determine the optimal number of temperature zones.

An algorithm is proposed to solve the problem of temperature optimization, as described in Section II-A. An assessment of the complexity of the algorithm is discussed in Section II-B, and a method to accelerate the algorithm is reviewed in Section II-C.

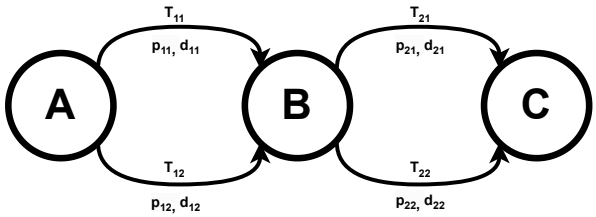

Fig. 2: An example of a constructed graph of a partitioned multi-temperature computing system. The nodes represent the partitioned group of units, and the edges represent the temperature of operation and corresponding power consumption and delay of the unit operating at that temperature.

A. Proposed algorithm

The proposed algorithm systematically explores all possible groupings and sequences of units, applying edge dependent shortest path computation to determine the optimal configuration. The methodology consists of the following steps:

1. Generate all possible groupings of units.
2. For each group configuration, all possible combinations (permutations) are generated. Since the temperature of each chamber within a refrigerator must be in descending order from room temperature to cryogenic temperatures, the order of the groups is important.
3. For each configuration of groupings and permutations, determine the optimal path using an edge dependent shortest path algorithm. The algorithm is applied to each configuration to determine the most power efficient set of temperatures while satisfying the delay constraint.
4. Select the most optimal solution: The most energy efficient paths for each configuration are compared, and the lowest power consuming configuration (and set of temperatures) are selected.

The algorithm generates all possible groups of units and all possible permutations in which a set of groups can be ordered or arranged.

The specific group configuration and permutation sequence are passed to the next algorithm, which determines the most optimal path for a specific configuration and sequence. Each new path adds and/or removes a path with a power and delay worse than the newly explored path. When a new edge is explored, a cost function for refrigerating a group, $C\left(P_G\left(T\right), T, T_{amb}\right)$, is determined. This cost function depends upon the temperature of the current chamber T and the ambient temperature T_{amb}, which is the temperature of the previous refrigerator chamber,

$$C\left(P_G\left(T\right), T, T_{amb}\right) = P_G(T)\left(\frac{T_{amb}/T - 1}{\eta_{II}/100} + 1\right), \quad (1)$$

where η_{II} is the second law efficiency of the refrigerator [14]. After the set of optimal temperatures is obtained for each configuration and sequence, the results are compared to obtain the most optimal group and configuration.

B. Complexity

The complexity of the proposed algorithm depends upon three factors: the number of groups, the possible permutations of the established groups, and the efficiency of the path finding algorithm. The number of possible combinations of groups is determined by the Bell number [15], which increases exponentially with the number of units. Determining the optimal path for a large number of units therefore becomes computationally expensive. Furthermore, the number of refrigerator stages is limited, since more than a certain number of chambers may reduce overall power efficiency [16]. For example, a refrigerator with seven stages is considered in [16]. In the proposed methodology, the number of refrigerator stages is limited to ten to consider systems which may require additional refrigerator stages.

A second factor contributing to the algorithmic complexity is the number of permutations which is $n!$, where n is the number of groups. A third factor is the complexity of the edge dependent shortest path finding algorithm. Without graph pruning, the complexity is $O(n^k)$, where k is the number of available temperatures for each refrigerator stage. After employing graph pruning, the complexity is reduced and depends upon the power and delay of each unit and the target delay. From experiments, the complexity is close to $O(n^{k/1.5})$.

The overall complexity of the algorithm is therefore $\text{Bell}(n) \times n! \times O(n^{k/1.5})$. For a large number of available temperatures per chamber, estimating the optimal path becomes computationally expensive due to the $n^{k/1.5}$ term. Methods for accelerating the algorithm are therefore considered.

C. ϵ-Dominance based partitioning

All non-dominated, Pareto optimal solutions to the problem are computationally expensive. To improve the convergence in determining the optimal path and to increase the diversity of the considered paths, the dynamic programming concept of epsilon (ϵ)-dominance is employed [13]. With ϵ-dominance, the objective space is partitioned into boxes (hypercubes) of size ϵ to maintain the diversity of the solutions. A single solution within each box is selected to speed up the convergence process [13], [17] and can be considered as an archiving technique.

In the proposed algorithm, the ϵ-dominance technique reduces the number of delay values under consideration while maintaining the diversity of the potential solutions. Before proceeding to the next node, the existing paths are archived based on the minimum delay and ϵ. The epsilon dominance approach guarantees that the retained solutions are within ϵ of the true Pareto front in the delay dimension, where the Pareto front represents the set of non-dominated solutions which offer the best tradeoff between power and delay. Epsilon dominance

reduces the number of paths that needs to be stored and evaluated. To produce results within 10% of the desired values, a hypercube size of 10% is used ($\delta = 0.1$). Each hypercube contains solutions with objective values differing from one another by no more than 10%.

After employing dynamic programming, the complexity is further reduced by considering fewer paths. The complexity of the edge dependent shortest path finding algorithm becomes $O(b^2 \times k^2 \times n)$, where b is the number of boxes,

$$b \approx \frac{1}{\delta} log \frac{d^{max}}{d^{min}}. \quad (2)$$

Here, d^{max} and d^{min} are, respectively, the maximum and minimum power consumption objective. The overall complexity of the algorithm with ϵ-dominance is therefore Bell$(n) \times n! \times O(b^2 \times k^2 \times n)$.

III. Case Study

Two different computing systems have been considered as a case study. These computing systems are the support circuitry for a quantum computing system (readout and control) and a cloud computing system. Each system consists of multiple units. Each unit operates at a different temperature ranging from 300 K (RT) to 3 K. The power consumption and delay of each unit are randomly assigned based on the range of reported values [18]–[22]. The power and delay vary at different temperatures in a exponential manner and are modeled as

$$y = y_{min} + (y_{max} - y_{min}) \frac{1 - e^{-kT}}{1 - e^{-kT_{max}}}, \quad (3)$$

where y is either the delay or power, T is the temperature, and k is a coefficient modeling the rate of increase or decrease of the power or delay with temperature. The expression is a simplified model which captures the exponential reduction in power and delay with temperature. The power dissipation at room temperature is randomly chosen between 5 watts and 100 watts. The delay at room temperature is randomly chosen between 50 nanoseconds and 1,000 nanoseconds. The variation in power dissipation between the minimum and maximum temperature is randomly chosen to be between five to ten times, and the delay variation between the minimum and maximum temperature is randomly assigned as five to fifteen times. The coefficient k in (3) is randomly assigned between 0.006 to 0.015 for each unit. These ranges are based on trends in power reduction and performance improvements observed in cryogenic circuits [18]–[22].

A system in the first case study, a cloud computer, consists of six units where 50 different operating temperatures are considered for each unit. A system in the second case study, support circuitry for a quantum computer, consists of seven units and 20 different operating temperatures considered for each unit. The target delay constraint is 600 nanoseconds. Optimization results for both case studies are listed in Table I where the most optimal grouping, most optimal number and sequence of chambers, and corresponding temperature for each chamber are listed. Both six and seven unit systems are shown to operate most efficiently within a three chamber refrigerator. In non-optimized cloud computing and quantum computing systems, where each unit is placed at the lowest available temperature, the total power consumption is, respectively, 17.6 kilowatts and 20.9 kilowatts. After optimizing the temperatures based on the proposed methodology, these systems consume 1.4 kilowatts and 3.4 kilowatts, achieving, respectively, an almost thirteen and six times reduction in power. Two additional case studies, systems with eight and nine units, are listed in Table I. The most optimal number of chambers for the eight and nine unit systems are, respectively, two and six.

IV. Conclusions

Advanced computing systems increasingly operate at cryogenic temperatures; particularly, cloud computing systems and quantum computing systems. These systems often include specialized circuitry that operate across a broad spectrum of temperatures—from room temperature to a few kelvin—necessitating multi-temperature zones to minimize power consumption by managing the local operating temperature of each chamber. The proposed methodology focuses on the thermal optimization of cryogenic computing systems operating across multiple temperature zones.

Unlike previous approaches that rely on a predetermined number of refrigeration stages and fixed unit configurations, the proposed algorithm determines both the optimal number of temperature zones and groups of functional units based on the performance and power profile of each subsystem. By employing a graph theoretic approach, each unit or group of units is represented as a node in a graph with the edges denoting different operating temperatures with assigned power and delay weights. The optimal path through this graph describes a system that minimizes the total power consumption while satisfying performance constraints, thereby maximizing energy efficiency and reducing overall power usage and local heat loads.

Two practical case studies are evaluated to demonstrate the methodology, one system with six units (a cloud computer) and another system with seven units

Table I: Optimization results for four case studies. Set of the most optimal group, most optimal number and sequence of chambers, and corresponding temperatures for six, seven, eight and nine unit systems. The considered systems are cryogenic cloud computing (CC) and quantum computing (QC) systems. Each unit has 20 to 50 different operating temperatures, and the target delay constraint is 600 to 800 nanoseconds for each case study. The most optimal number of chambers for the first two systems is three. For the eight and nine unit systems, the most optimal number of chambers is, respectively, two and six.

Type	**Delay limit** D_{max}**, [ns]**	**n units**	**n temp-s**	**n chambers**	**Unit sequence**	**Temperatures** T, [K]	**Delay** D_i, [ns]	**Power** P_i, [W]
CC1	600	6	50	3	[1], [3], [2, 4, 5, 6]	[28.6, 12.3, 6.4]	599.1	1,379.0
QC1	600	7	20	3	[1], [4], [2, 3, 5, 6, 7]	[12.8, 4.9, 3.0]	595.6	3,404.4
CC2	800	8	50	2	[8], [1, 2, 3, 4, 5, 6, 7]	[3.6, 3.0]	787.7	3,895.5
QC2	750	9	30	6	[8], [1], [9], [7], [6], [2, 3, 4, 5]	[23.6, 9.1, 7.8, 6.6, 3.5, 3.0]	748.0	1,493.5

(readout and control circuitry for a quantum computer)—each operating at temperatures ranging between 300 K to 3 K. The power consumption of the cloud computing system is reduced from 17.6 kilowatts for a non-optimized system to 1.4 kilowatts, achieving a nearly thirteen fold reduction in power. For the quantum computing system, a six fold reduction in power is demonstrated, from 20.9 kilowatts to 3.4 kilowatts. In both scenarios, the optimal number of refrigerators is three.

REFERENCES

[1] G. Krylov, T. Jabbari, and E. G. Friedman, *Single Flux Quantum Integrated Circuit Design, Second Edition*, Springer, 2024.

[2] M. H. Devoret, A. Wallraff, and J. M. Martinis, "Superconducting Qubits: A Short Review," *arXiv preprint cond-mat/0411174*, November 2004.

[3] T. Jabbari *et al.*, "Repeater Insertion in SFQ Interconnect," *IEEE Transactions on Applied Superconductivity*, Vol. 30, No. 8, Article 5400508, December 2020.

[4] A. Mitrovic and E. G. Friedman, "Thermal Exploration of RSFQ Integrated Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 32, No. 4, pp. 728–738, April 2024.

[5] S. Carnot, "Reflections on the Motive Power of Fire, and on Machines Fitted to Develop that Power," *Paris: Bachelier*, Vol. 108, 1824.

[6] D. A. Cardwell, D. C. Larbalestier, and A. Braginski, *Handbook of Superconductivity: Processing and Cryogenics, Volume Two*, CRC Press, 2022.

[7] D. S. Holmes, A. L. Ripple, and M. A. Manheimer, "Energy-Efficient Superconducting Computing—Power Budgets and Requirements," *IEEE Transactions on Applied Superconductivity*, Vol. 23, No. 3, pp. 1,701,610, June 2013.

[8] O. Mukhanov *et al.*, "Scalable Quantum Computing Infrastructure Based on Superconducting Electronics," *Proceedings of the IEEE International Electron Devices Meeting*, pp. 31.2.1–31.2.4, December 2019.

[9] N. Zhuldassov, R. Bairamkulov, and E. G. Friedman, "Thermal Optimization of Hybrid Cryogenic Computing Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 31, No. 9, pp. 1339–1346, September 2023.

[10] N. Zhuldassov and E. G. Friedman, "Temperature–Frequency Boundary of Cryogenic Dynamic Logic," *Microelectronics Journal*, Vol. 135, p. 105763, March 2023.

[11] N. Zhuldassov, R. Bairamkulov, and E. G. Friedman, "Heat Load Efficiency in Multi-Temperature Cryogenic Computing Systems," *Cryogenics*, 2024 (in review).

[12] R. Bairamkulov and E. G. Friedman, *Graphs in VLSI*, Springer, 2023.

[13] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," *Evolutionary Computation*, Vol. 10, No. 3, pp. 263–282, September 2002.

[14] R. G. Ross Jr, "Refrigeration Systems for Achieving Cryogenic Temperatures," *Proceedings of the Low Temperature Materials and Mechanisms*, pp. 127–200, CRC Press, August 2016.

[15] M. Aigner, "A Characterization of the Bell Numbers," *Discrete Mathematics*, Vol. 205, No. 1-3, pp. 207–210, July 1999.

[16] R. J. Thomas, P. Ghosh, and K. Chowdhury, "Optimum Number of Stages and Intermediate Pressure Level for Highest Exergy Efficiency in Large Helium Liquefiers," *International Journal of Refrigeration*, Vol. 36, No. 8, pp. 2438–2457, June 2013.

[17] A. Menchaca-Méndez *et al.*, "A Co-Evolutionary Scheme for Multi-Objective Evolutionary Algorithms Based on ϵ - Dominance," *IEEE Access*, Vol. 7, pp. 18 267–18 283, February 2019.

[18] I. Byun *et al.*, "A Next-Generation Cryogenic Processor Architecture," *IEEE Micro*, Vol. 41, No. 3, pp. 80–86, March 2021.

[19] I. Byun *et al.*, "CryoCore: A Fast and Dense Processor Architecture for Cryogenic Computing," *Proceedings of the IEEE Annual International Symposium on Computer Architecture*, pp. 335–348, May 2020.

[20] G. Lee, D. Min, I. Byun, and J. Kim, "Cryogenic Computer Architecture Modeling With Memory-Side Case Studies," *Proceedings of the International Symposium on Computer Architecture*, pp. 774–787, June 2019.

[21] E. Garzón, A. Teman, and M. Lanuzza, "Embedded Memories for Cryogenic Applications," *Electronics*, Vol. 11, No. 1, p. 61, December 2021.

[22] S. Resch, H. Cilasun, and U. R. Karpuzcu, "Cryogenic PIM: Challenges & Opportunities," *IEEE Computer Architecture Letters*, Vol. 20, No. 1, pp. 74–77, May 2021.