f-HybridMem: A Fuzzy-based Approach for Decision Support in Hybrid Memory Management

1st Rodrigo Costa de Moura
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
rcmoura@inf.ufpel.edu.br

2nd Guilherme Bayer Schneider
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
gbschneider@inf.ufpel.edu.br

3rd Lizandro de Souza Oliveira
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
lsoliveira@inf.ufpel.edu.br

4th Mauricio Lima Pilla
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
pilla@inf.ufpel.edu.br

5th Adenauer Correa Yamin
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
adenauer@inf.ufpel.edu.br

6th Renata Hax Sander Reiser
Laboratory of Ubiquitous and Parallel Systems (LUPS)
Federal University of Pelotas
Pelotas, Brazil
reiser@inf.ufpel.edu.br

***Abstract*—In the Hybrid Memory research approach, the development of new storing data strategies combines the traditional volatile memories (DRAM) with the emerging non-volatile memory (NVM) to take advantage of their potential and best features. One question in the hybrid memory research is the decision of which memory technology to use in order to store data of distinct applications. These decision processes have to examine the behavior of memory operations, such as read/write frequency, and also their memory characteristics. Also, considering the large volume of parameters and the uncertainties inherent in the data migration, the decision to store data in hybrid memories is not a trivial task. This paper presents the f-HybridMem component, a fuzzy-based system to support the uncertainty in data management for hybrid memory architectures. Thus, this proposal contributes to determine a correct selection between memory modules by improving the data management, in a page level organization. The memory management takes into account the behavior of memory operations and the memory characteristics. Such architecture is conceived based on the following modules: (i) Access Updater, a hardware module to identify the pages access patterns, and (ii) f-HybridMem component, a fuzzy-based software module supporting the migration decision processes. In addition, tests are conducted attempt evaluating the accuracy of the fuzzy-based system. Moreover, the proposed evaluations aim to estimate the influence of the following parameters: Buffer Size, Counters Size, Frequency of Migration, Promote Value, Demote Value, providing a correct recommend for data migrations.**

***Index Terms*—Decision analysis, Decision Support, Hybrid memory, Memory Management, Fuzzy Logic Application.**

I. Introduction

The capacity of the main memory and its energy consumption have been a big challenge to the development of new computer systems. Advances in cloud computing and big data processing are examples of today's demands for high-speed, high-capacity and low-power memory architectures.

Some studies show that DRAM memory technology is achieving its limit of scalability [1], [2]. The main constraint of DRAM technology is related to energy consumption and cell sizes. As the memory size increases, its response time also increases. Consequently, in DRAM technology, the grown of main memory to meet large scale computing leads to an increase in energy consumption and also in power dissipation.

Considering this scenario, to achieve high memory capacity demand, new technologies have been developed as alternatives to the DRAM architecture. Thus, a strong trend in memory architecture is the Non-Volatile Memories (NVMs), providing lower static power and higher density when compared to DRAM structures. Besides that, in some cases, NVMs can achieve an overall low energy consumption. Because of these main characteristics, NVM technologies have the potential to overcome the scalability limits related to DRAM memories.

However, NVM application over traditional memory architectures leads to new challenges because of the low endurance of NVM and the asymmetry time executions between read and write operations. Moreover, in these scenarios, where NVM is still under development and the simple replacement of DRAM by NVM architecture would face high durability issues [3], the use of Hybrid Memories constitutes a new perspective to prospect solutions of such memory architecture problems. This approach integrates the use of both approaches, DRAM and NVM, exploiting the best features of each technology [4].

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

One basic problem in the hybrid memory research area is the following decision-question: which memory module need to be select for writing each data. This making decision problem has to consider the uncertainties inherent to multiple characteristics of hybrid memory architectures. See, e.g., the size of memories, speed of read/write operations, endurance, and energy consumption. Moreover, and more challenging, the decision has to examine and identify the patterns of memory accesses, also considering the uncertainty related to all previously described attributes.

This paper presents the f-HybridMem, a novel approach based on Fuzzy Systems to decision support in data management in Hybrid Main Memories. This approach takes into account uncertainties inherent to multiple characteristics of hybrid memory architectures and also variances between such memory technologies and their organization.

This paper is organized as follows. The background, which presents an overview of the memory technologies is presented in Section II. Section III discusses related works about data management in hybrid memories. In Section IV is presented the f-HybridMem proposal. Section V shows our experiments, evaluations and results. Finally, Section VI present our conclusions.

II. Background

This section presents an overview of memory technologies and features the motivation for using hybrid memories.

A. Memory Technologies

NVMs are classified according to their functional properties concerning to programming and erasing operations [5].

Various techniques use the NVM architecture because of their advantages over traditional DRAM and SRAM memories. Although such emerging memory market is still smaller than traditional ones, it is expected that this market will grow by 2021, reaching rates around 110% per year, when these new technologies will be used in many products [6].

Some examples of emerging memories are PCM/PCRAM (Phase Change Memory Random-Access Memory), MRAM (Magnetoresistive RAM), STT-MRAM/STT-RAM (Spin-Transfer Torque Magnetic RAM), RRAM/ReRAM (Resistive RAM), FRAM/FeRAM (Ferro-magnetic RAM) and DWM (Domain Wall Memory).

PCM employs a material called GeSbTe (GST), which is alloys of germanium, antimony, and tellurium. GST has two phases, known as an amorphous phase and a crystalline phase, representing the high and low electrical resistivity, respectively [7]. Compared to SRAM and eDRAM (embedded DRAM), these NVM technologies have higher density, lower standby power, better scalability, and are non-volatile [8]. However, PCM has some problems like write latency, write energy and endurance [7].

An MRAM is based on memory cells having two magnetic storage elements, one with a fixed magnetic polarity and the other one with a switchable polarity [5]. MRAM can be directly coupled with processors and used as both volatile and non-volatile storage. This characteristic differentiates MRAM from resistive RAM and PCM, limited by slow write speed and inadequate endurance, which are pursued primarily for storage applications [9].

An STT-RAM memory uses the magnetic tunnel junction (MTJ) for the storage element, reducing the energy required to write the cell. STT-RAM is a form of MRAM that uses *spin-transfer torque* to reorient the free layer by passing a large, directional write current through the MTJ [10].

FeRAM maintains the data without any external power supply by using a ferroelectric material in the place of a conventional dielectric material between the plates of the capacitor. One disadvantage of FeRAM is that the read cycle is destructive [5]. FeRAMs are expected to have many applications in small consumer devices such as personal digital assistants (PDAs), smartphones, power meters, smart cards, and in security systems. Even after FeRAM has achieved a level of commercial success, current FeRAM chips offer performance that is either comparable to or exceeding current Flash memories, but still slower than DRAMs [5].

A RM, also known as Domain Wall Memory (DWM), can achieve ultra-high storage density, fast access velocity and non-volatility. Former research has demonstrated that RM has potential to serve as on-chip cache or main memory. However, RM has more flexibility and difficulty in design space of main memory because it has more device level design parameters [11].

B. Hybrid Memories

There are several types of memory optimizations which are described in the literature. These approaches are focused on hardware or software optimizations, but also can be combined hardware-software optimizations.

Several works take advantage of new memory technologies to replace DRAM as main memory and improve life time [12], performance [13], energy and performance [14], [15], and other works turn their attention to energy optimization [16].

NVM technologies have been proposed as an alternative to mitigate the disadvantages of traditional DRAM and SRAM memories. In this scenario, hybrid architectures have been used to combine desirable characteristics of volatile and non-volatile memories. However, according to [17], there are significant performance issues caused by the interference due to data migration between DRAM and NVM and a lack of effective migration policies.

III. Related Works

Researchers have proposed using NVMs to address DRAM scalability. Then, it becomes relevant to use data management in hybrid memories to deal with multiple parameters in volatile and NVMs. In this case, it is meaningful to investigate data management strategies in hybrid memories.

DRAM-PCM hybrid memory architectures offer the merit of combining the advantages of DRAM and PCM and hiding their shortcomings. Khouzani et al. [18] emphasized that PCM is a promising candidate to be used as main memory in next

generation computer systems, given its non-volatility, potential high density, and low static power. Khouzani et al. [18] explore the interactions between DRAM and PCM to improve both the performance and the endurance of a DRAM-PCM hybrid main memory. The evaluation set includes selected benchmarks from the SPEC 2000 & 2006 suites. Experimental results have shown that with the proposed techniques are able to deliver 7% reduction in DRAM misses and 6% reduction in PCM writes. With this approach, they improve average memory hit time (AMHT) by 1.3% while energy consumption was reduced by 4% and it can prolong PCM lifetime by 72%. According to the authors, these results confirm that unlike traditional techniques which have to trade-off among performance, energy consumption, and lifetime, the proposed hybrid memory design can improve all the three factors at the same time.

Bock et al. [17] suggest a combination of DRAM+NVM to increase capacity and reliability, and to decrease energy consumption. They propose new analysis and simulation techniques to understand the behavior of software-managed hybrid memory. For evaluation, this work used a subset of the SPEC CPU2006 benchmark suite. These benchmarks were used with multiple data sets as a separate workload obtaining a total of 52 different benchmark/input combinations. This work has been identified factors that limit performance in hybrid main memory. Experimental results show that the highest L2 access latency reduction potential comes from developing better migration policies (27% average), followed by eliminating the overhead of the PCM bus queue (11%), L2 miss rate (6%), PCM bank queue (4%) and PCM row buffer miss rate (2%).

Other works approach page migration in hybrid memories. Bock et al. [19] describes Concurrent Migration of Multiple Pages (CMMP), a hardware-software mechanism for managing hybrid main memory (DRAM+PCM). According to the authors, this mechanism reduces contention at the memory system caused by migration and also improves performance by 14% and reduces energy consumption by 29% on average.

Cheng et al. [20] propose an adaptive page allocation and buffer management methodology for the hierarchical DRAM/PCM memory architecture. In this propose, a small DRAM is used as cache of PCM to reduce leakage power consumption. Energy consumption and access latency of PCM was improved by 25%.

Li et al. [21] propose a page-management mechanism called utility-based hybrid memory management (UH-MEM). This mechanism estimates the utility of migrating a page between different memory types and uses this information to guide data placement. This proposal improves performance by 14% on average compared to the state-of-art mechanisms.

In other extension, Liu et al. [22] introduced Memos, a memory management framework in the OS for horizontally integrated DRAM and NVM.

According to Huang et al. [23], most researches are based on migration to decrease the average memory access cost for the higher cost of NVMs read/write operation. However, the page migration itself is a high-cost operation. In this work, in order to decrease cost, they propose a virtual page behavior based page management policy (VBPM). They allocate virtual pages into DRAM or PCM physical pages correspondingly. Experimental results show that this approach decreases the average memory access time by 24% and improves real-time performance in critical path.

Maddah et al. [24] propose a Cost Aware Flip Optimization for Asymmetric Memories (CAFO), a new cost aware flip reduction scheme. This scheme targets minimizing the cost incurred by the write operation. This model uses both PCM and STT-RAM through setting the costs of bit flips that would match the characteristics of the underlying technology. According to the authors, this proposal is capable of cutting down the write cost by up to 65% more than existing schemes.

Table I shows the summary of the related works.

TABLE I
SUMMARIZED RELATED WORKS

Reference	Hybrid	Volatile	NVM
[17], [18], [19], [21], [23]	Yes	DRAM	PCM
[20], [22]	Yes	DRAM	Generic
[24]	Yes	STT-RAM	PCM

Several other works take advantage of cache approach to deal with the high cost of read/write operation in NVMs and consequently to deal with this cost in hybrid memories that uses NVMs ([25], [26], [27], [28], [29], [30]).

Other works use Fuzzy Logic for decision support in cache memories ([31], [32], [33], [34], [35]).

IV. *f-HybridMem* PROPOSAL

The present work contributes to improving the data management, in a page level organization, in hybrid main memories by using a fuzzy-based approach.

This proposal considers a hybrid memory composed of two memory modules: a DRAM and an NVM module, shown in Fig.1. Because of the constraints of endurance and energy consumption on NVM, it is not recommended to execute many write operations in this module. Thus, it is desirable to store on NVM only pages with a high read rate, and keep all other pages on DRAM. So, the strategy of *f-HybridMem* for managing hybrid memories is by migrating pages between memory modules. Based on the access patterns of each page, the *f-HybridMem* can perform two types of page migration:

- Promotion - The migration of pages from NVM to DRAM, and
- Demotion - The migration of pages from DRAM to NVM.

At the beginning of memory management, all pages are stored in DRAM. Then, based on the access patterns of each page, the migrations start. To perform all this mechanism, the f-HybridMem architecture consists of two components, highlighted in gray in the Fig. 1:

(i) Access Updater, a hardware component responsible for storing the page accesses in the Access Buffer and periodically send the buffer data to f-HybridMem; and

(ii) f-HybridMem module, a software component responsible for evaluating the data received from the Access Updater, returning a migration recommendation value for each page in the buffer.

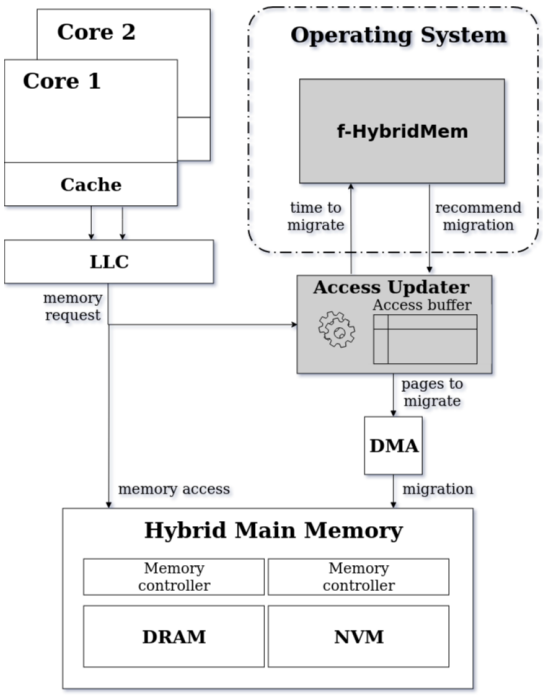

Fig. 1. Hybrid Main Memory architecture with f-HybridMem

A. Access Updater

The storing information about frequency access pages at the main memory is considered in the Access Updater module.

This hardware module is composed of the Access Buffer, the mechanism to update the buffer, and an interface to periodically call f-HybridMem module on the operating system. The Access Buffer holds the values of the volume of writings, the volume of readings and the recency of each page access.

The Access Updater module is placed between the last level cache and the main memory to intercept the accesses to the main memory. This interception is made without adding costs in the normal memory access.

Once memory access occurs, the Access Updates intercept it and updates the access buffer. The module stores the page address if it is not already stored, and increase its read or write counters, according to the type of memory access.

Periodically, the Access Updater requests to f-HybribMem, located in the operating system, to recommend page migrations. Thus, the Access Updater sends all the information stored in the buffer to the f-HybribMem and receives a list of migrations recommendations. Lastly, the Access Updater, based on the recommendations of fHybridMem, assigns to DMA the migration of the pages in the memory.

1) Access Buffer: The Access Buffer stores information about the frequency of read/write accesses to memory pages, as well as the recency and repetition of these accesses. The Access Buffer has three fields: page address, read counter and write counter.

Since the Access Buffer is an additional hardware, its size must be as small as possible [36]. Thus, the size of the buffer can be controlled based on two limits: the number of records in the buffer and the number of bits for the read/write counters. Also, it is necessary to evaluate the performance of the memory management mechanism and compare to the performance of using the same additional hardware to increase main memory.

In this work, it was considered strategies to reduce the hardware size. Due to this buffer limit size, the Least Recently Used Policy (LRU) is adopted, replacing pages in the buffer. Regarding the counter limits, when a record achieves the maximum representation the counters of all records are bit shifted by one to right. In this paper, the records in the Access Buffer are stored in the insertion order, where the position of the pages in the buffer divided by the buffer size represents the recency of access. See Session V presenting an evaluation of distinct buffer configurations.

B. Fuzzy Modeling of f-HybridMem System

According with [37], a fuzzy set X is characterized by its membership function (MF) $\mu_A : \chi \to [0,1]$, with $\mu_A(u)$ interpreting the membership degree of an element u in the universe $\chi \neq \emptyset$, related to a fuzzy set A. Thus, a fuzzy set A can be described by pairs, associating $u \in \chi$ to its membership value $\mu_\chi(u)$ as given in the following:

$$A = \{(u, \mu_A(u)) \colon u \in \chi, \mu_\chi(u) \in [0,1]\}. \tag{1}$$

When $\#\chi = n$, we indicate a fuzzy value as $\mu_X(u_i) = x_i \in A$, $i \in \mathbb{N}_n = \{1, 2, \ldots, n\}$. Moreover, by considering the natural order $\leq$ on $[0,1]$, we can define the lattice $L([0,1]) = ([0,1], \leq, \vee, \wedge, 1, 0)$ of all fuzzy values, taking the supremum and infimum operators as follows:

$$x \vee y = \max(x, y) \quad \text{and} \quad x \wedge y = \min(x, y). \tag{2}$$

In this study, graphic expressions of MF is given by trapezoidal functions and the main functions which qualify operators on $L([0,1])$ are reported below [38]:

- A triangular norm (t-norm) $T : [0,1]^2 \to [0,1]$ satisfies the commutative, associative and monotonic properties and it has the $1 \in [0,1]$ as neutral element. The operator $\min : [0,1]^2 \to [0,1]$ interprets intersections on $L([0,1])$.
- A triangular conorm (t-conorm) $S : U^2 \to U$ that satisfies the commutative, associative and monotonic properties and it has the $0 \in U$ as neutral element. The operator $\max : [0,1]^2 \to [0,1]$ interprets the union on $L([0,1])$.

In addition, a fuzzy negation $N : [0,1] \to [0,1]$ is a decreasing function such that $N(0) = 1$ and $N(1) = 0$. Function $N(x) = 1 - x$ provide interpretation to the complement of a FS on $L([0,1])$. And, a fuzzy implication $I : U^2 \to U$ preserves the values of classical implication: $I(1,1) = I(0,0) = I(0,1) = 1$, $I(1,0) = 0$ and also verifies the isotonicity in the first argument and anti-tonicity in the second one.

The *f-HybridMem* verifies the priority of each page be switched from one to other memory, taking into account a Rule Base acting on the three steps: Fuzzification, Inference, and Defuzzification. So, it returns as output the priority of each page. The modeling of the fuzzy system was performed using the Juzzy [39] module.

1) f-HybridMem Data Base - Membership Functions: During the study of variables considering the expertise opinions, each one of linguistic variables (LV) was associated to four distinct FS, using the trapezoidal graphical representation to corresponding membership functions.

A setting reading related to the simulated cloud computing environment is performed to measure attributes as Recency of Access (RA), Read Frequency (RF) and Write Frequency (WF). The reading values are applied to the standard scale, considering the interval $[0, 10]$, for RA Eq.(3), RF Eq.(4) and WF Eq.(5) in order to obtain their membership degrees:

$$RA = p_i(LA)/MaxDistance \quad (3)$$

$$RF = h_i(RC)/MaxR \quad (4)$$

$$WF = p_i(WC)/MaxW \quad (5)$$

The following parameters are considered:

- p_i denoting the $page(i)$ of the memory environment;
- LA indicating the executed memory instructions count after the last access on the same page;
- RC denoting the read operation count on the page;
- WC denoting the write operation count on the page;
- $MaxDistance$ representing the instruction count executed after least recently accessed page
- $MaxW$ representing the highest number of write operations among pages;
- $MaxR$ indicating the highest number of read operations among pages;

The linguistic terms (LT) defining FS related to variable RA are stated as follows: "Low" (LRA), "Medium" (MRA) and "High" (HRA - best case), in $[0, 10]$, as shown in Figure 2(a).

In the case of FS associated with variable RF, we have that: "Low" (LRF), "Medium" (MRF) and "High" (HRF - best case), as graphically represented in Figure 2(b).

In the design of the FS for WF, were considered " Low " (LWF), " Medium " (MWF) and " High " (HWF - best case) as LT, see Figure 2(c).

The output variable (Promotion) is also adapted to a standard scale, and the LT corresponding FS used are as follows: "Low" (LP), "Medium" (MU) and "High" (HP - best case). These Membership Functions are presented in Figure 2(d).

2) Fuzzification: At this stage, the input values (already set for an observed scale in the section IV-B1) is mapped to the fuzzy domain.

3) Rule Base (RB): In *f-HybridMem* component, the development of the RB was based on the expertise of specialists. The RB should be easily understandable and editable since there is no difficulty in adding new rules whether other input variables are desired to be manipulated. Three factors are considered in its construction:

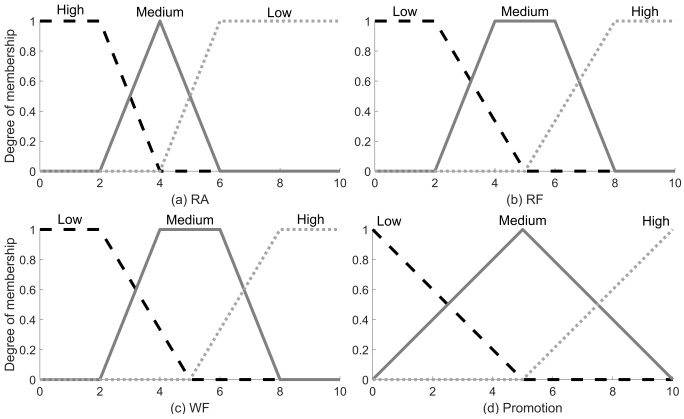

Fig. 2. RA, RF, WF and Promotion in the default scale

- LV as name FS, turning the modeling closer to the real world system;
- Type "AND" connections are taken into account to create the relationship among the input variables;
- Type fuzzy implications performing the affirmative modus in inference scheme, related to the generalized modus ponens (GMP): "if X is A, then Y is B".

4) Inference: In inference process, the composition operators is performed over FS relating the antecedents of rules to implications using the generalized modus ponens operator.

(i) It performs the application of fuzzy operators when input data consist of three values resulting from fuzzification, applying the fuzzy implication $I(x, y) = MAX(1-x, y)$. The "AND" fuzzy operator aggregates the main rules and, the method MIN (minimum) returning related fuzzification values;

(ii) Implication Fuzzy Method Application, performs a combination of the value obtained in the fuzzy operator applied and the values of FS output rule, using the method MIN (minimum) on these combinations;

(iii) Aggregation Fuzzy Method Application, resulting composition of the fuzzy output of each rule by using the method MAX (maximum), thus creating a single fuzzy region to be analyzed by the next Fuzzy process module.

5) Defuzzification: In defuzzification step of the *f-HybridMem* system, the region transformation results on a discrete value (related to promotion) applying the center of the area. This method calculates the centroid (u) of the area consisting of the output of the fuzzy inference system (connecting of all contribution rules stated in sections IV-B3 and IV-B2) according with the following equation:

$$u = \frac{\sum_{i=1}^{N} u_i \mu_{OUT}(u_i)}{\sum_{i=1}^{N} \mu_{OUT}(u_i)} \quad (6)$$

V. *f-HybridMem* EVALUATION

First, traces of memory access were collected by running a subset of benchmarks from Mibench [40], over GEM5 [41] and NVMain simulators [42]. We selected the following

benchmarks: *basicmath*, *FFT*, *qsort* and *typeset*. The basic math test performs simple mathematical calculations. On the other hand, *FFT* performs a Fast Fourier Transform and its inverse transform on an array of data. The *qsort* test sorts a large array of strings into ascending order using the quick sort algorithm. *Typeset* is a general typesetting tool, that has a front-end processor for HTML.

These memory traces were used as input for all tests performed in this paper. We used an in-house simulator to model the hybrid main memory and the Access Updater. The f-HybridMem module was implemented using the Juzzy tool.

The tests performed aims to evaluate the influence of five parameters in the architecture performance:

- Buffer Size: number of pages in the buffer;
- Size of Counters: assuming the counter maximum value;
- Frequency of Migration: representing a periodicity, in number of instruction, that the f-HybridMem will be called to recommend migrations;
- Promote Value: The threshold value to accept the recommendation for migrating pages from NVM to DRAM;
- Demote Value: The threshold value to accept the recommendation for migrating pages from DRAM to NVM.

Table II presents the values adopted for each parameter. The tests combined all of the values of the parameters, totalizing 2700 architecture configurations.

The five figures from Fig. 3 to Fig. 7 show the evaluations of each parameter for the four benchmarks. The values presented in each chart corresponds to the best result among all the tests for the evaluated parameter. As the fuzzy algorithm seeks to group the maximum of read operations in the NVM memory and avoid write operations in this memory, the information used to measure the efficiency of the fuzzy algorithm was the relation between reads and writes operations on NVM. In this context, two metrics were adopted:

- Reads/Writes ratio - considers only the effective read and write operations obtained from the traces;
- Reads/Writes ratio + Migration - considers the effective read/write operations plus the migration operation costs.

Fig. 3 shows the best results for the five buffer sizes. It can be seen that when migration costs are not considered, the best results are given for the smaller buffers for the basicmath and typeset benchmarks. FFT benchmark, on the other hand, does not seem to be affected by the buffer size. QuickSort had its best result with an intermediate buffer. These behaviors demonstrate that, as opposed to expectative, large buffers do not improve the accuracy of the algorithm. The size of the buffer is directly related to the recency parameter, dividing the position of the page on buffer by the buffer size.

Thus, in large buffers, pages with different recency values are evaluated by the fuzzy mechanism with very close values, reducing the weight of this parameter in decision making. Besides, the results including the migration cost, it is possible to see that larger buffers decrease performance. This indicates a high volume of migrations in the larger buffers, confirming that the variation in the access recency value is not very significant, leading to a huge volume of pages being migrated.

TABLE II
SIMULATION PARAMETERS VALUES

Parameter	Values
Buffer size	[16, 32, 64, 128, 256]
Size of counters	[4, 8, 16, 32, 64]
Frequency of migration	[2048, 8192, 32768]
Promote value	[4, 5, 6, 7, 8, 9]
Demote value	[1, 2, 3, 4, 5, 6]

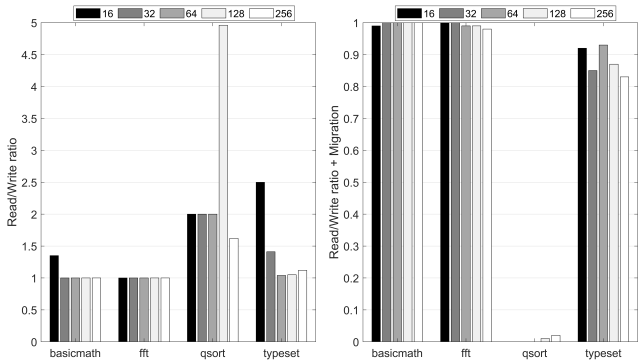

Fig. 3. Best results of read/write ratio for Buffer Size parameter

Fig. 4 shows results for the five size counters. The Basicmath and Typeset benchmarks show that the intermediate values are responsible for better results when excluding the migration cost. In the FFT and QuickSort benchmarks, there was no difference between the counter sizes. When considering the migration cost, the variation between the parameters was insignificant for all benchmarks.

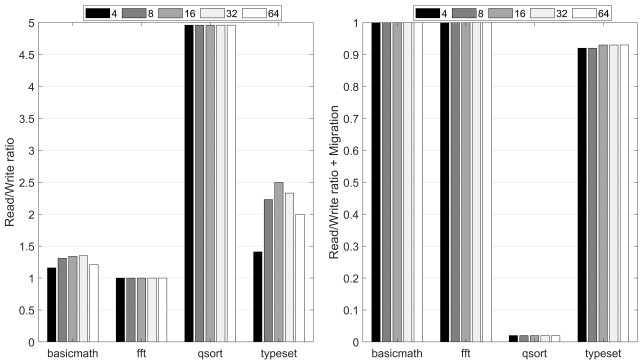

Fig. 4. Best results of read/write ratio for Counter Size parameter

Regarding the Promote value parameter, Fig. 5, the basicmath benchmark shows the behavior as expected, where the highest Promote values reach the best result when the migration cost is not considered. The other benchmarks are not significantly affected by this parameter.

The Demote Value parameter, Fig. 6, shows that when the Demote value is 1, the performance is the minimum, which means that there were no demotions and, consequently, no data on NVM. Moreover, the best results are achieved

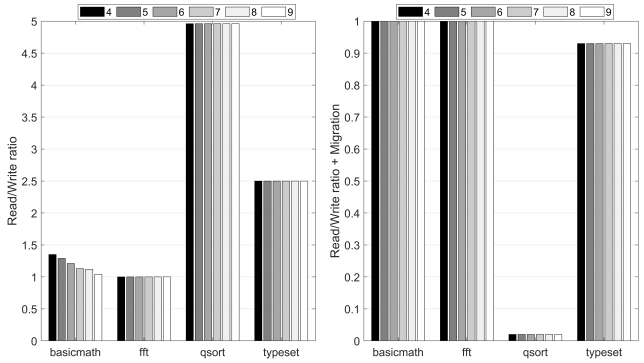

Fig. 5. Best results of read/write ratio for Promote Value parameter

when Demote is between 2 and 4. This result is as expected, however, this great difference between parameters 1 and 2 leads to a need for new tests with intermediate values. In the Fig. 7 Basicmath and Quicksort show the best results with shorter frequency of migration, while the typeset shows the opposite. This indicates that the benchmark characteristics lead to different results. Testes with higher and lower values are needed for better analysis.

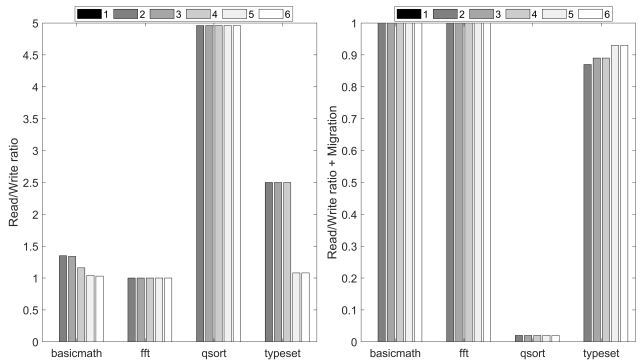

Fig. 6. Best results of read/write ratio for Demote Value parameter

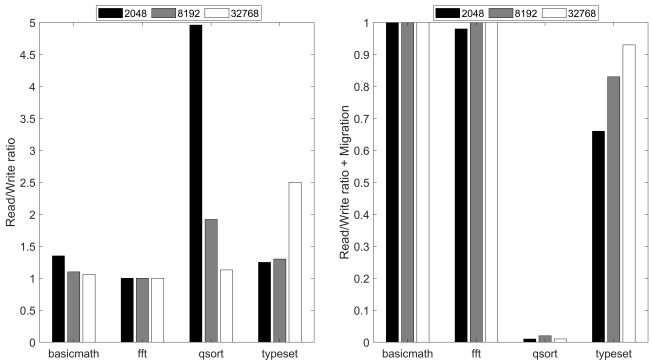

Fig. 7. Best results of read/write ratio for Frequency of Migration parameter

VI. Conclusions and Future Trends

In this paper, we present an architectural proposal based on Fuzzy Systems for decision support in Hybrid Memories data management. For this purpose, we model and implement a hybrid memory architecture with two modules: (i) Access Updater, a hardware module to identify the pages access patterns, and (ii) f-HybridMem, a fuzzy-based software module to support migration decision. The tests conducted with the proposed architecture attempt to evaluate the accuracy of the fuzzy system decision making. Moreover, the evaluations aim to estimate the influence of the five parameters related to architecture: Buffer Size, Counters Size, Frequency of Migration, Promote and Demote Value.

Results show that the fuzzy decision support is able to correctly recommend migrations based on 3 parameters: frequency of reads, frequency of writes and recency of access. However, a large volume of migrations was observed, which limits the efficiency of the use of NVM memory. So, as a continuation of this work, it is necessary to adjust the fuzzy rules to reduce the volume of migrations.

Further work intends to evaluate the performance of the memory management mechanism compared to the performance of using the same additional hardware to increase main memory. Also, it is possible to expand the analysis of f-HybridMem component in different applications. For this, we are considering the use a wider range of benchmarks, considering the tuning and selection of inputs in database. In order to improve the f-HybridMem decision making, we are promoting new developments in the RB not only basing on the expertise and knowledge of specialists but also associating possible automate learning strategies and considering other families of fuzzy operators to achieve new comparisons. And, we are investigating the Multi-Value Fuzzy Logic approach, in particular, the Interval-valued Fuzzy Logic to model not only the uncertainty of input data (Buffer Size, Counters Size, Frequency of Migration, Promote Value and Demote Value), but also the imprecision of the computational calculations. See, for instance, [43]–[45]. In addition, this extension based on interval approach can be applied in an integrate way to admissible linear orders, enabling the result comparisons and also preserving the imprecision of final data.

Acknowledgment

This work was supported by CNPq, PQ(309160/2019-7) and PqG/FAPERGS 02/2017(17/2551-0001207-0).

References

[1] B. C. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, and D. Burger, "Phase-change technology and the future of main memory," *IEEE Micro*, 2010.

[2] E. Kultursay, M. Kandemir, A. Sivasubramaniam, and O. Mutlu, "Evaluating stt-ram as an energy-efficient main memory alternative," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, Apr 2013. [Online]. Available: http://dx.doi.org/10.1109/ISPASS.2013.6557176

[3] A. Eisenman, D. Gardner, I. AbdelRahman, J. Axboe, S. Dong, K. Hazelwood, C. Petersen, A. Cidon, and S. Katti, "Reducing dram footprint with nvm in facebook," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 42:1–42:13. [Online]. Available: http://doi.acm.org/10.1145/3190508.3190524

[4] X. Wu, J. Li, L. Zhang, E. Speight, and Y. Xie, "Power and performance of read-write aware hybrid caches with non-volatile memories," in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, apr 2009. [Online]. Available: https://doi.org/10.1109/date.2009.5090762

[5] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale research letters*, vol. 9, no. 1, p. 526, 2014.

[6] Y. Développement. (2016, 28 July) Storage-class memory will be the clear go-to market for emerging non-volatile memory in 2021. [Online]. Available: http://www.yole.fr/Emerging_NVM_Market.aspx#.WYEepojyvDf.

[7] Y. Joo, D. Niu, X. Dong, G. Sun, N. Chang, and Y. Xie, "Energy-and endurance-aware design of phase change memory caches," in *Design, Autom. & Test in Eur. Conf. & Exhibition, 2010*. IEEE, 2010, pp. 136–141.

[8] J. Wang, X. Dong, Y. Xie, and N. P. Jouppi, "i2wap: Improving non-volatile cache lifetime by reducing inter-and intra-set write variations," in *IEEE 19th Intl. Symp. on High Perf. Comp. Arch.* IEEE, 2013, pp. 234–245.

[9] S. H. Kang and C. Park, "Mram: Enabling a sustainable device for pervasive system architectures and applications," in *IEEE Intl. Electron Devices Meeting*, Dec 2017, pp. 38.2.1–38.2.4.

[10] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient stt-ram caches," in *Intl. Symp. on High Perf. Comput. Arch.* IEEE, 2011, pp. 50–61.

[11] H. Zhang, C. Zhang, Q. Hu, C. Yang, and J. Shu, "Perf. anal. on structure of racetrack memory," in *23rd Asia and South Pacific Design Autom. Conf.*, Jan 2018, pp. 367–374.

[12] A. Sampson, J. Nelson, K. Strauss, and L. Ceze, "Approximate storage in solid-state memories," *ACM Trans. on Comput. Syst.*, vol. 32, no. 3, p. 9, 2014.

[13] B. Li, S. Shan, Y. Hu, and X. Li, "Partial-set: write speedup of pcm main memory," in *Design, Autom. Test Eur. Conf. and Exhibition, 2014*. IEEE, 2014, pp. 1–4.

[14] Z. Zhang, Z. Jia, P. Liu, and L. Ju, "Energy efficient real-time task scheduling for embedded systems with hybrid main memory," *Journal of Signal Processing Syst.*, vol. 84, no. 1, pp. 69–89, 2016.

[15] Q. Hu, G. Sun, J. Shu, and C. Zhang, "Exploring main memory design based on racetrack memory technology," in *Intl. Great Lakes Symp. on VLSI (GLSVLSI)*, May 2016, pp. 397–402.

[16] G. Wang, Y. Guan, Y. Wang, and Z. Shao, "Energy-aware assignment and scheduling for hybrid main memory in embedded systems," *Computing*, vol. 98, no. 3, pp. 279–301, 2016.

[17] S. Bock, B. R. Childers, R. Melhem, and D. Mossé, "Characterizing the overhead of software-managed hybrid main memory," in *2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Oct 2015, pp. 33–42.

[18] H. Aghaei Khouzani, F. S. Hosseini, and C. Yang, "Segment and conflict aware page allocation and migration in dram-pcm hybrid main memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 9, pp. 1458–1470, Sep. 2017.

[19] S. Bock, B. R. Childers, R. Melhem, and D. Mossé, "Concurrent migration of multiple pages in software-managed hybrid main memory," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, Oct 2016, pp. 420–423.

[20] W. Cheng, P. Cheng, and X. Li, "Adaptive page allocation of dram/pcram hybrid memory architecture," in *2016 5th International Symposium on Next-Generation Electronics (ISNE)*, May 2016, pp. 1–2.

[21] Y. Li, S. Ghose, J. Choi, J. Sun, H. Wang, and O. Mutlu, "Utility-based hybrid memory management," in *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2017, pp. 152–165.

[22] L. Liu, S. Yang, L. Peng, and X. Li, "Hierarchical hybrid memory management in os for tiered memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2223–2236, Oct 2019.

[23] J. Huang, X. Zhang, G. Han, G. Jia, H. Liyou, and J. Wan, "Virtual page behavior based page management policy for hybrid main memory in cloud computing," in *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Dec 2016, pp. 120–124.

[24] R. Maddah, S. M. Seyedzadeh, and R. Melhem, "Cafo: Cost aware flip optimization for asymmetric memories," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2015, pp. 320–330.

[25] S. Mittal and J. S. Vetter, "Ayush: A technique for extending lifetime of sram-nvm hybrid caches," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 115–118, July 2015.

[26] X. Cai, L. Ju, M. Zhao, Z. Sun, and Z. Jia, "A novel page caching policy for pcm and dram of hybrid memory architecture," in *2016 13th International Conference on Embedded Software and Systems (ICESS)*, Aug 2016, pp. 67–73.

[27] G. Ju, Y. Li, Y. Xu, J. Chen, and J. C. S. Lui, "Stochastic modeling of hybrid cache systems," in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Sep. 2016, pp. 69–78.

[28] J. Choi and G. Park, "Nvm way allocation scheme to reduce nvm writes for hybrid cache architecture in chip-multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2896–2910, Oct 2017.

[29] D. Chen, H. Jin, X. Liao, H. Liu, R. Guo, and D. Liu, "Malru: Miss-penalty aware lru-based cache replacement for hybrid memory systems," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 1086–1091.

[30] Y. Guo, W. Xiao, Q. Liu, and X. He, "A cost-effective and energy-efficient architecture for die-stacked dram/nvm memory systems," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, Nov 2018, pp. 1–2.

[31] S. Khajoueinejad, M. Sabeghi, and A. Sadeghzadeh, "A fuzzy cache replacement policy and its experimental performance assessment," in *2006 Innovations in Information Technology*, Nov 2006, pp. 1–5.

[32] H. Diab, A. Kashani, and A. Nasri, "Cache replacement engine: A fuzzy logic approach," in *2009 International Conference on the Current Trends in Information Technology (CTIT)*, Dec 2009, pp. 1–7.

[33] M. Y. Qadri and K. D. McDonald-Maier, "A fuzzy logic based dynamic reconfiguration scheme for optimal energy and throughput in symmetric chip multiprocessors," in *2010 NASA/ESA Conference on Adaptive Hardware and Systems*, June 2010, pp. 333–339.

[34] Y. M. Chung and Z. A. Halim, "An improved adaptive neuro-fuzzy inference system as cache memory replacement policy," in *2016 IEEE Industrial Electronics and Applications Conference (IEACon)*, Nov 2016, pp. 330–335.

[35] J. Niu, J. Xu, and L. Xie, "Online fuzzy logic control with decision tree for improving hybrid cache performance symposia," in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, June 2016, pp. 511–516.

[36] S. Bock, "Collaborative hardware-software management of hybrid main memory," January 2018. [Online]. Available: http://d-scholarship.pitt.edu/33455/

[37] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 6 1965. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S001999586590241X

[38] E. P. Klement, R. Mesiar, and E. Pap, *Triangular norms*. Springer Science & Business Media, 2013, vol. 8.

[39] C. Wagner, "Juzzy - a java based toolkit for type-2 fuzzy logic," in *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, April 2013, pp. 45–52.

[40] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *IEEE Intl. Workshop on Workload Characterization*. IEEE, 2001, pp. 3–14.

[41] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, "The gem5 simulator," *ACM SIGARCH Comput. Arch. News*, vol. 39, no. 2, pp. 1–7, 2011.

[42] M. Poremba, T. Zhang, and Y. Xie, "Nvmain 2.0: A user-friendly memory simulator to model (non-) volatile memory systems," *IEEE Comput. Arch. Letters*, vol. vol. 14, no. inssue: 2, pp. 140–143, 2015.

[43] B. M. Moura, G. B. Schneider, A. C. Yamin, M. L. Pilla, and R. H. Reiser, "Allocating virtual machines exploring type-2 fuzzy logic and admissible orders," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[44] A. Argou, R. Dilli, R. Reiser, and A. Yamin, "Exploring type-2 fuzzy logic with dynamic rules in iot resources classification," in *2019 IEEE Intl Conf on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[45] R. Dilli, A. Argou, M. Pilla, A. M. Pernas, R. Reiser, and A. Yamin, "Fuzzy logic and mcda in iot resources classification," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 761–766.