

Improving Hybrid Multi-Level I/O Cache Lifetime Using Conservative Cache Management

Mostafa Hadizadeh¹, Reza Salkhordeh², and Hossein Asadi³

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.

¹mhadizadeh@ce.sharif.edu, ²reza.salkhordeh@sharif.edu, and ³asadi@sharif.edu

Corresponding author address: Hossein Asadi, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, Post Code: 11155-9516.

Abstract- Hybrid cache architectures are promising solutions for designing efficient I/O caching schemes. One of the major challenges of hybrid caches is the management of read miss data pages. As such pages are non-critical in terms of data reliability, they can be buffered in volatile memories such as DRAM as an *Endurance Friendly* (EF) solution. Low capacity of DRAM, however, provides a small hit ratio for these pages. A state-of-the-art solution has proposed a *State-Machine Based Insertion* (SMBI) scheme in order to improve cache performance by inserting read miss data pages in a *Solid-State Drive* (SSD), based on a static time periods. Nevertheless, the static approach of SMBI leads to considerable cache lifetime degradation compared to EF policy. In this paper, we propose *Conservative State-Machine based Insertion* (CSMI) scheme to improve I/O cache lifetime. CSMI dynamically adjusts the time that state-machine exploits EF policy based on workload behavior. To this end, CSMI employs a *Conservative Factor* (CF), which multiplies the number of periods that EF policy is exploited in case of low hit ratio. Our evaluations show that CSMI reduces the number of SSD writes up to 33.97%, compared to state-of-the-art solution with negligible performance overhead (average of 0.06%).

Keywords- Data Storage System, Solid-State Drive, Hybrid Multi-Level Cache, Lifetime.

ارائه یک روش محافظه کارانه به منظور بهبود طول عمر حافظه نهان ورودی/خروجی

مصطفی هادی زاده^۱، رضا سالخورده^۲، حسین اسدی^{۳*}

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران.

¹mhadizadeh@ce.sharif.edu, ²reza.salkhordeh@sharif.edu, and ³*asadi@sharif.edu

* نشانی نویسنده مسئول: حسین اسدی، تهران، خیابان آزادی، دانشگاه صنعتی شریف، دانشکده مهندسی کامپیوتر، کد پستی: ۱۱۱۵۵-۹۵۱۶

استفاده از معماری‌های ترکیبی حافظه نهان در سامانه‌های ذخیره‌سازی داده از جمله رویکردهایی است که می‌تواند منجر به بهبود دوام، کارایی و قابلیت اطمینان این سامانه‌ها شود. از جمله چالش‌های معماری‌های ترکیبی، چگونگی مدیریت فقدان خواندن است. مدیریت مبتنی بر ماشین حالت از جمله روش‌های مدیریت فقدان خواندن است. با این وجود، استفاده از رویکرد ایستا در طراحی ماشین حالت منجر به کاهش قابل توجه طول عمر حافظه نهان می‌شود. در این مقاله یک روش درج حافظه کارانه مبتنی بر ماشین حالت (CSMI) برای درج صفحات حاصل از فقدان خواندن در دیسک حالت جامد پیشنهاد می‌شود. CSMI به منظور افزایش طول عمر حافظه نهان، به صورت پویا تعداد بازه‌های استفاده از سیاست عدم درج فقدان‌های خواندن در ادوات با طول عمر محدود را افزایش می‌دهد. به این منظور، ضریب محافظه کاری (CF) به ساختار ماشین حالت اضافه می‌شود تا با افزایش تعداد بازه‌های عدم درج در هنگام مواجهه با نرخ برخورد پایین، تعداد دفعات نوشتن بر روی دیسک حالت جامد کاهش یافته و طول عمر حافظه نهان افزایش یابد. ارزیابی‌های انجام شده نشان می‌دهد روش پیشنهادی، با هزینه قابل چشم‌پوشی در حوزه کارایی (به طور میانگین ۰.۳۸٪ نرخ برخورد پایین‌تر و ۰.۰۶٪ زمان پاسخ بیشتر)، به طور میانگین برای ضرایب محافظه کاری با مقادیر ۲، ۳ و ۴ به ترتیب ۷.۴٪ (تا ۲۵.۴۸٪)، ۸.۶۳٪ (تا ۳۳.۹۷٪) و ۸.۹۴٪ (تا ۳۳.۹۷٪) تعداد دفعات نوشتن بر روی دیسک حالت جامد را نسبت به بهترین کار پیشین کاهش می‌دهد.

واژه‌های کلیدی: سامانه‌های ذخیره‌سازی داده، دیسک حالت جامد، معماری ترکیبی چندسطحی، طول عمر، کارایی.

۱- مقدمه

یافتن^۵ و چرخش^۶، این ادوات را به گلوگاه کارایی سامانه‌ها تبدیل کرده‌است [۳]. دیسک‌های حالت جامد^۷ (SSDs)، ادوات ذخیره‌سازی مبتنی بر حافظه‌های فلش^۸ هستند که بر خلاف دیسک سخت، ساختاری غیرمکانیکی دارند و کارایی بالاتری نسبت به دیسک‌های سخت ارائه می‌دهند [۴]. علی‌رغم کارایی بالاتر این ادوات، دوام^۹ محدود حافظه‌های فلش و قیمت بالاتر دیسک‌های حالت جامد نسبت به دیسک‌های سخت، مانع از جایگزینی کامل این دستگاه‌ها شده‌است.

استفاده از دیسک حالت جامد به عنوان بخشی از سامانه

افزایش سرعت تولید داده در سال‌های اخیر، ذخیره‌سازی داده را به یکی از اصلی‌ترین چالش‌های موجود در حوزه علوم و مهندسی رایانه تبدیل کرده‌است [۱][۲]. ذخیره‌سازی داده با چالش‌های متفاوتی مانند کیفیت خدمت^۱، قابلیت اطمینان^۲ و هزینه^۳ مواجه است [۳۹][۴۰]. افزایش فاصله بین کارایی پردازنده‌ها و دیسک‌های سخت^۴ (HDDs)، نیاز به فناوری‌ها و راه‌حل‌های جدید در حوزه ذخیره‌سازی داده را بیش از پیش کرده‌است [۳].

ساختار مکانیکی دیسک‌های سخت و تاخیرهایی مانند زمان

WED مورد ارزیابی قرار می‌گیرد و در صورتی که استفاده از این سیاست منجر به افزایش نرخ برخورد نشده باشد، تعداد پنجره‌های عدم استفاده از سیاست WED، به اندازه CF برابر افزایش خواهد یافت. به این ترتیب، CSMI با کاهش تعداد بازه‌های استفاده از WED، تعداد دفعات نوشتن بر روی دیسک حالت جامد را کاهش داده و طول عمر حافظه نهان را افزایش می‌دهد.

ارزیابی‌های انجام شده بر روی ۱۰ بارکاری پرکاربرد و با استفاده از ادوات واقعی نشان می‌دهد که روش CSMI نسبت به بهترین کار پیشین [۸]، به طور میانگین ۶.۲۷٪ (تا ۲۵.۷۴٪) تعداد نوشتن‌های ارسال شده به دیسک‌های حالت جامد را کاهش داده است. همچنین بررسی معیار NAND Writes [۴۱] کاهش تا ۳۳.۹۷٪ را نشان می‌دهد که نشانگر بهبود طول عمر حافظه نهان توسط روش CSMI است. این روش در حوزه کارایی نیز هزینه کمی به سامانه تحمیل می‌کند و به طور میانگین تنها ۰.۳۸٪ نرخ برخورد را کاهش و ۰.۰۶٪ زمان پاسخ را افزایش می‌دهد.

به طور خلاصه نوآوری‌های این مقاله به این شرح است:

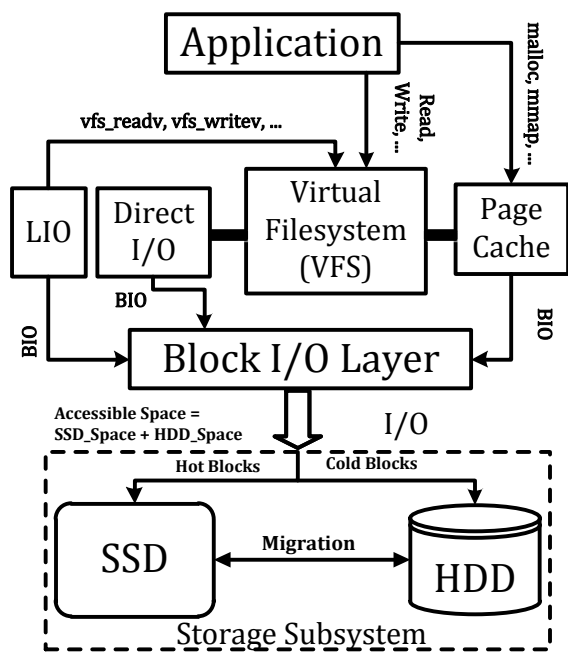
- در این مقاله یک روش درج محافظه کارانه مبتنی بر ماشین حالت (CSMI) به منظور مدیریت فقدان خواندن برای معماری‌های ترکیبی حافظه نهان ورودی/خروجی پیشنهاد می‌شود. روش پیشنهادی با تحلیل نرخ برخورد حاصل از درج صفحات فقدان خواندن در دیسک حالت جامد، به صورت پویا مدت زمان استفاده از سیاست عدم درج (EF) را افزایش می‌دهد. به این منظور CSMI از یک ضریب محافظه کاری (CF) بهره می‌برد و در هنگام مواجهه با نرخ برخورد پایین، تعداد بازه‌های استفاده از سیاست EF را CF برابر می‌کند.
- روش پیشنهادی بر روی معماری حافظه نهان سه سطحی ورودی/خروجی (TICA) [۸] اعمال شده و با هزینه قابل چشم‌پوشی در حوزه نرخ برخورد (به طور میانگین ۰.۳۸٪)، تا ۲۵.۷۴٪ تعداد دفعات نوشتن بر روی دیسک‌های حالت جامد را کاهش می‌دهد.
- تاثیر مقادیر مختلف CF مورد ارزیابی قرار گرفته و مشاهده می‌شود برای سه مقدار مختلف CF (۲، ۳ و ۴)، محافظه کارانه‌ترین حالت (مقدار ۴) با تاثیر قابل چشم‌پوشی بر نرخ برخورد، یک درصد تعداد نوشتن‌های ارسال شده به دیسک‌های حالت جامد را کاهش می‌دهد.

ذخیره‌سازی داده، از جمله رویکردهای طراحی سامانه‌های رایانه‌ای با قیمت مناسب است. از جمله ساختارهای ارائه شده با این رویکرد می‌توان به: (۱) رده‌بندی^{۱۰} [۵]، (۲) سامانه‌های مجتمع حافظه-دیسک^{۱۱} (MDIS) [۶] و (۳) استفاده از دیسک حالت جامد به عنوان حافظه نهان [۷] اشاره کرد. استفاده از دیسک حالت جامد به عنوان حافظه نهان این امکان را فراهم می‌کند تا با پرداخت هزینه کمتر، از سرعت بالاتر این دستگاه استفاده شود و فاصله بین کارایی حافظه اصلی^{۱۲} و سامانه ذخیره‌سازی داده کاهش یابد. از جمله اصلی‌ترین چالش‌های موجود در این ساختار، کاهش طول عمر حافظه نهان به دلیل افزایش فشار نوشتن^{۱۳} بر روی این ادوات است [۸][۹]. معماری‌های ترکیبی حافظه نهان از جمله راهکارهایی هستند که با استفاده از حافظه‌های بادوام‌تر نسبت به دیسک حالت جامد (مانند DRAM) در ساختار حافظه نهان، طول عمر دیسک حالت جامد را افزایش می‌دهند [۸][۱۰].

یکی از مهم‌ترین چالش‌های موجود در معماری‌های ترکیبی، چگونگی مدیریت فقدان خواندن^{۱۴} است. درج صفحات حاصل از فقدان خواندن در حافظه DRAM، بدون کاهش قابلیت اطمینان، طول عمر حافظه نهان را افزایش می‌دهد و یک سیاست مدیریتی سازگار با دوام^{۱۵} (EF) به شمار می‌آید [۸]. با این وجود، فضای کم‌تر DRAM در مقایسه با دیسک حالت جامد در معماری‌های ترکیبی منجر به کاهش فرصت صفحات حاصل از فقدان خواندن جهت برخورد^{۱۶} می‌شود.

روش درج مبتنی بر ماشین حالت^{۱۷} (SMBI) از جمله روش‌های ارائه شده به منظور مدیریت کارایی^{۱۸} فقدان خواندن است. این روش با در نظر گرفتن یک معماری سه سطحی شامل DRAM و دیسک‌های حالت جامد با دوام‌های مختلف، از دو سیاست سازگار با دوام (EF) و درج در دستگاه بادوام‌تر^{۱۹} (WED) به منظور مدیریت فقدان خواندن استفاده کرده تا به مصالحه‌ای بین کارایی و دوام دست یابد. با این وجود، رویکرد استفاده از تعداد بازه‌های ثابت استفاده از سیاست EF در SMBI، منجر به کاهش قابل توجه طول عمر دیسک حالت جامد (تا ۹۰٪) نسبت به سیاست EF می‌شود [۸].

در این مقاله، یک رویکرد محافظه کارانه^{۲۰} به منظور استفاده از سیاست WED به منظور بهبود طول عمر حافظه نهان ارائه خواهد شد. به این منظور، روش درج محافظه کارانه مبتنی بر ماشین حالت^{۲۱} (CSMI) ارائه می‌شود. در این روش، ضریب محافظه کاری^{۲۲} (CF) به ساختار ماشین حالت اضافه می‌شود تا به این ترتیب مدت زمانی که ماشین حالت از سیاست WED استفاده می‌کند کاهش یابد. در CSMI، نرخ برخورد^{۲۳} حاصل از سیاست



شکل ۱- ساختار رده‌بندی [۳۸][۵]

دونسخه‌نویسی^{۲۶} داده است. از این رو ساختاری یک لایه‌ای که فضای ذخیره‌سازی و حافظه را به صورت عرضی^{۲۷} در کنار یکدیگر مورد استفاده قرار می‌دهد، پیشنهاد شده‌است [۲۴][۶] [۲۵].

در روش‌های مبتنی بر رده‌بندی، همه درخواست‌ها این امکان را ندارند که از سرعت بالاتر دیسک حالت جامد استفاده کنند. رده‌بندی برای بارهای کاری با الگوهای ثابت و بدون تغییرات ناگهانی دسترسی، مناسب است. در هنگام رخ دادن تغییر در رفتار بارکاری، باید در چینش داده‌ها تغییر ایجاد شود و قطعه‌های داغ جدید مشخص بشوند؛ فرایندی که سربار محاسباتی زیادی دارد و می‌تواند بر کارایی سامانه ذخیره‌سازی اثر گذارد [۳۷]. روش‌های مبتنی بر MDIS، به دلیل تفاوت کارایی قابل توجه بین DRAM و دیسک حالت جامد، نیاز به حافظه‌های غیرفرار سریع‌تر دارند که این حافظه‌ها هنوز به صورت تجاری تولید نشده‌اند. همچنین این رویکرد نیاز به تغییر ساختار سیستم عامل‌های رایج فعلی دارد.

رویکرد سوم، استفاده از دیسک حالت جامد به عنوان حافظه نهان است. در این رویکرد لایه بالاتر به فضای آدرس دیسک حالت جامد دسترسی ندارد اما این امکان برای درخواست‌ها فراهم شده تا از سرعت بالاتر دیسک حالت جامد استفاده کنند. در پژوهش‌هایی با رویکرد استفاده از دیسک حالت جامد به عنوان حافظه نهان، معمولاً یک یا دو عامل از عواملی مانند قابلیت اطمینان، دوام یا کارایی مدنظر قرار گرفته و یکی از دو سیاست نهان‌سازی^{۲۸} درون‌نویسی^{۲۹} و پس‌نویسی^{۳۰} مورد استفاده قرار گرفته شده‌است. در سیاست درون‌نویسی، همواره یک نسخه از

در ادامه این مقاله، بخش ۲ به پیش‌زمینه و پژوهش‌های پیشین می‌پردازد. در بخش ۳، به طرح روش پیشنهادی خواهیم پرداخت. در بخش ۴ راهکار ارائه شده مورد ارزیابی قرار خواهد گرفت و سرانجام در بخش ۵ به نتیجه‌گیری و کارهای آتی خواهیم پرداخت.

۲- پیش‌زمینه و پژوهش‌های پیشین

دیسک‌های حالت جامد مبتنی بر حافظه‌های فلش با مصرف توان کم‌تر و سرعت بالاتر نسبت به دیسک‌های سخت [۳]، فرصت‌های متعددی در حوزه‌های پژوهشی و صنعتی ایجاد کرده‌اند. قیمت بالا و طول عمر محدود دیسک‌های حالت جامد از جمله عواملی هستند که منجر به استفاده از دیسک حالت جامد به عنوان بخشی از سامانه‌های ذخیره‌سازی داده شده‌است. در این بخش ابتدا به بررسی رویکردهای مختلف استفاده از دیسک حالت جامد می‌پردازیم و سپس به بررسی پژوهش‌های پیشین مرتبط خواهیم پرداخت.

۲-۱- رویکردهای استفاده از دیسک حالت جامد در سامانه‌های

ذخیره‌سازی داده

استفاده از دیسک حالت جامد به عنوان بخشی از سامانه ذخیره‌سازی داده این امکان را فراهم می‌کند که با صرف هزینه کم، از مزایای این دستگاه استفاده کنیم. یکی از رویکردهای استفاده از دیسک حالت جامد، رده‌بندی است [۵]. در روش‌های مبتنی بر رده‌بندی، از دیسک حالت جامد و دیسک سخت در کنار هم و به صورت جفت‌شده^{۲۴} استفاده می‌شود. به این ترتیب هم فضای دیسک سخت و هم فضای دیسک حالت جامد در اختیار کاربر قرار می‌گیرد و کاربر به فضای آدرس هر دو دستگاه دسترسی خواهد داشت. در روش‌های مبتنی بر این رویکرد، می‌توان با استفاده از دیسک حالت جامد با حجم کم در کنار دیسک سخت با حجم زیاد، با صرف هزینه کم، از مزایای دیسک حالت جامد بهره برد. در رویکردهای مبتنی بر رده‌بندی، تلاش می‌شود تا داده‌های پرارجاع در دیسک حالت جامد و داده‌های کم‌ارجاع‌تر در دیسک سخت نگه داشته شوند. شکل ۱ ساختار کلی رده‌بندی را نشان می‌دهد [۲۲][۲۳].

رویکرد دیگر، استفاده از دیسک‌های حالت جامد در سامانه‌های مجتمع حافظه-دیسک است. شکل ۲، ساختار MDIS را نشان می‌دهد. ایده اصلی سامانه‌های مجتمع حافظه-دیسک، تغییر ساختار رایج سلسله‌مراتبی^{۲۵} به منظور کاهش کاستی‌های سلسله مراتب حافظه فعلی مانند جایجایی مکرر داده‌ها و

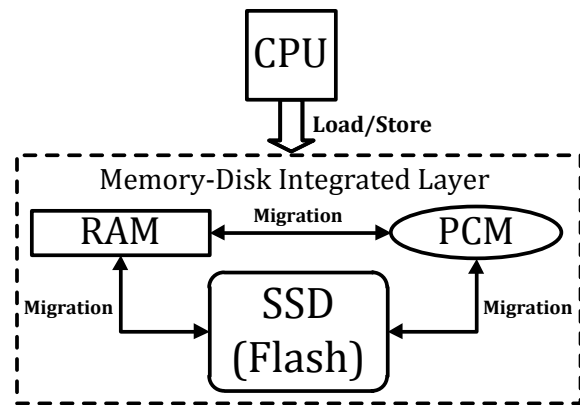
راهکارهایی برای مقابله با این چالش بود.

در S-RAC [۱۱]، تلاش شده تا به بارهای کاری با هزینه نهان سازی متوسط و فایده زیاد، فرصت بیشتری جهت حضور در حافظه نهان داده شود. به این منظور ابتدا ۳۲ بارکاری مراکز داده مورد بررسی و ارزیابی قرار گرفته و به ۶ دسته تقسیم شدند. سپس با استفاده از فاصله استفاده مجدد^{۳۸}، صفحه‌هایی که باید به آن‌ها فرصت مجدد داد، انتخاب می‌شوند. هنگامی که این صفحه‌ها به انتهای صف می‌رسند، به جای اخراج از حافظه نهان، مجدداً در حافظه نهان درج می‌شوند. از جمله روش‌هایی که سعی در بهبود نهان سازی با توجه به خصوصیت به روزرسانی خارج از محل^{۳۹} دارند، می‌توان به [۴۲] و [۴۳] اشاره کرد.

در [۱۳]، یک افزونه^{۴۰} برای حافظه‌های نهان پیشنهاد شده است. در این افزونه، مستقل از الگوریتم مورد استفاده در حافظه نهان، داده‌های محبوب مشخص شده و وارد حافظه نهان می‌شوند. همچنین این افزونه سعی دارد مانع از ورود داده‌های نامحبوب شود که تنها موجب افزایش نوشتن‌ها بر روی دیسک حالت جامد می‌شوند. به این منظور فرصتی بیشتر از اندازه حافظه نهان به صفحه‌های محبوب داده می‌شود؛ برای هر صفحه حاضر در حافظه نهان، یک مرز تعیین می‌شود و صفحه تا زمانی که به مرز تعیین شده نرسیده، از حافظه نهان خارج نمی‌شود. دسته دیگری از پژوهش‌ها روش‌هایی را به منظور استفاده از دیسک حالت جامد به عنوان حافظه نهان در سامانه‌های مجازی^{۴۱} پیشنهاد داده‌اند [۳۰][۳۱].

در [۱۴]، یک الگوریتم آگاه از دیسک حالت جامد بر اساس الگوریتم ARC [۱۵] ارائه شده است. در این الگوریتم، شرط درج داده در دیسک حالت جامد، دسترسی چندباره به آن صفحه است. در ReCA [۷]، ابتدا بارهای کاری به ۵ دسته تقسیم و پیکربندی بهینه حافظه نهان برای هر دسته ارائه می‌شود. سپس در حین اجرای هر بارکاری، پیکربندی بهینه متناسب با آن بارکاری به حافظه نهان اعمال می‌شود. در [۳۲][۳۳][۳۴]، روش‌های نهان سازی مبتنی بر دیسک حالت جامد برای سامانه‌های کلید-مقدار^{۴۲} پیشنهاد شده است.

استفاده از معماری‌های ترکیبی رویکرد دیگری است که به منظور افزایش طول عمر و کارایی حافظه نهان سامانه‌های ذخیره سازی داده پیشنهاد شده است. در این معماری‌ها، از حافظه‌هایی با دوام بالاتر و کارایی بهتر (مانند DRAM، STT-MRAM^{۴۳} یا PCM^{۴۴}) در کنار دیسک حالت جامد استفاده می‌شود. به این ترتیب تلاش می‌شود تا فشار نوشتن بر روی دیسک‌های حالت



شکل ۲: ساختار MDIS [۶]

داده در دیسک سخت وجود دارد و از این رو رخ دادن خرابی در دیسک حالت جامد منجر به از دست دادن داده نمی‌شود. در سیاست پس‌نویسی، درخواست‌های نوشتن ابتدا در دیسک حالت جامد نوشته می‌شوند و بعداً به صورت ناهمگام به دیسک سخت ارسال می‌شوند.

۲-۲- پژوهش‌های پیشین

از جمله پژوهش‌هایی که از سیاست درون‌نویسی استفاده کرده‌اند می‌توان به DuraCache [۱۷] و Azor [۱۸] اشاره کرد. در [۱۷]، دو روش تبدیل خطا^{۴۱} و DR-SSD^{۴۲} به منظور افزایش طول عمر دیسک‌های حالت جامد در صورت بروز خطا یا فرسودگی^{۴۳} ارائه شده است. در روش تبدیل خطا، در هنگام رخ دادن یک خطای غیر قابل اصلاح، این خطا تبدیل به یک فقدان^{۴۴} در حافظه نهان شده و داده از دیسک سخت خوانده می‌شود. روش DR-SSD از افزونگی^{۴۵} بیشتر برای افزایش طول عمر دیسک حالت جامد استفاده می‌کند. در این روش وقتی دیسک حالت جامد به آستانه فرسودگی خود نزدیک می‌شود، بخشی از هر قطعه^{۴۶} در دیسک حالت جامد برای نگهداری کد اصلاح خطای^{۴۷} سایر صفحات موجود در قطعه اختصاص داده می‌شود.

در پژوهش Azor [۱۸]، یک حافظه نهان بر مبنای اولویت‌بندی قطعه‌های مختلف داده پیشنهاد شده است که در فرایند اولویت‌بندی، بالاترین اولویت به فراداده اختصاص می‌یابد و اولویت‌های بعدی بر اساس تعداد دفعات دسترسی به یک قطعه مشخص می‌شود. اولویت بالاتر فراداده می‌تواند منجر به اخراج داده‌های با ارزش به جای فراداده‌های کم‌ارزش شود.

دسته دیگر پژوهش‌ها از سیاست پس‌نویسی به منظور مدیریت حافظه نهان استفاده می‌کنند. یکی از چالش‌های اصلی این رویکرد، از دست رفتن داده در هنگام رخ دادن خرابی‌های غیرقابل بازیابی در دیسک حالت جامد است. از این رو باید به دنبال

جامد کاهش یابد. همچنین کارایی بهتر این حافظه‌ها منجر به افزایش کارایی حافظه نهان می‌شود [۸][۱۲][۱۳][۲۶]. در [۱۶]، حافظه‌های نهان ترکیبی چندسطحی مورد ارزیابی قرار گرفته‌است. همچنین معایب حافظه‌های غیراختصاصی^{۴۵} بیان شده و رویکردهای مختلف طراحی حافظه‌های نهان اختصاصی^{۴۶} مورد تحلیل و ارزیابی قرار گرفته‌است.

در [۱۲]، یک حافظه نهان ترکیبی متشکل از حافظه‌های PCM و فلش برای زیرسامانه ذخیره‌سازی پیشنهاد شده‌است. در این معماری، از PCM و فلش در یک سطح و در کنار هم استفاده شده و سعی می‌گردد تا درخواست‌ها متناسب با خصوصیات هر کدام از این حافظه‌ها به آن‌ها فرستاده شود. به سبب دوام بالاتر حافظه PCM نسبت به فلش و قابلیت به‌روزرسانی در محل این حافظه، درخواست‌های نوشتن بر روی PCM نوشته می‌شود. اگر درخواست نوشتن منجر به برخورد در فلش شود، صفحه مربوطه نامعتبر شده و در PCM نوشته می‌شود.

در [۱۰]، یک حافظه نهان دو سطحی شامل DRAM و دیسک حالت جامد ارائه شده است. در uCache، درخواست‌ها ابتدا در DRAM (که توسط یک صف دو سطحی مدیریت می‌گردد) ذخیره می‌شوند. داده‌های موجود در سطح دوم DRAM، پس از خروج از سطح اول حافظه نهان در دیسک حالت جامد درج می‌شوند. این روش به دلیل درج داده‌های دست‌خورده در حافظه فرار DRAM، قابلیت اطمینان پایینی دارد.

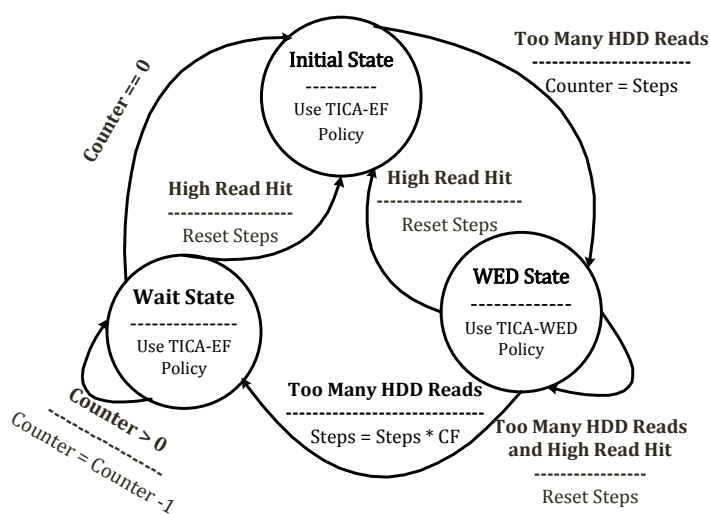
در [۸]، یک معماری با توجه به هر سه حوزه قابلیت اطمینان، دوام و کارایی ارائه شده‌است. شکل ۳، ساختار TICA را نشان می‌دهد. در این معماری، از دو دیسک حالت جامد ناهمگون^{۴۷} و DRAM استفاده شده‌است. TICA با استفاده از

۳- روش پیشنهادی

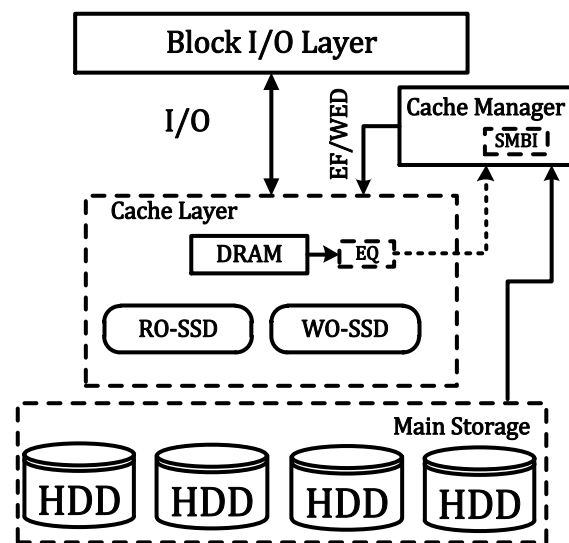
روش درج مبتنی بر ماشین حالت ارائه شده در [۸]، به منظور مدل‌سازی رفتار بارکاری توسط ماشین حالت، از بازه‌های نمونه‌برداری^{۵۱} برای اندازه‌گیری نرخ برخورد و دسترسی به دیسک سخت بهره می‌برد (گذارهای ماشین حالت در پایان هر بازه اندازه‌گیری صورت خواهد گرفت). ماشین حالت دارای سه حالت است: حالت اولیه^{۵۲}، حالت WED و حالت انتظار^{۵۳}. در SMBI، حالت انتظار و حالت اولیه از سیاست EF و حالت WED از سیاست WED استفاده می‌کنند. ماشین حالت زمانی به حالت انتظار می‌رود که استفاده از حالت WED منجر به نرخ برخورد خوبی نشود. به این ترتیب، در حالت انتظار به مدت چند بازه از سیاست EF استفاده می‌شود تا از درج بی‌رویه صفحات حاصل از

در [۸]، یک معماری با توجه به هر سه حوزه قابلیت اطمینان، دوام و کارایی ارائه شده‌است. شکل ۳، ساختار TICA را نشان می‌دهد. در این معماری، از دو دیسک حالت جامد ناهمگون^{۴۷} و DRAM استفاده شده‌است. TICA با استفاده از

در [۸]، یک معماری با توجه به هر سه حوزه قابلیت اطمینان، دوام و کارایی ارائه شده‌است. شکل ۳، ساختار TICA را نشان می‌دهد. در این معماری، از دو دیسک حالت جامد ناهمگون^{۴۷} و DRAM استفاده شده‌است. TICA با استفاده از



شکل ۴- ساختار روش درج محافظه‌کارانه مبتنی بر ماشین حالت



شکل ۳- ساختار TICA [۸]

فقدان خواندن در WO-SSD جلوگیری شود.

در TICA-A، ماشین حالت حداکثر به مدت ۴ بازه نمونه برداری در حالت انتظار به سر خواهد برد (در صورتی که ماشین حالت در حالت انتظار باشد و رفتار بارکاری تغییر کند، ماشین حالت از حالت انتظار خارج می شود). علی رغم عملکرد بهتر TICA-A نسبت به روش TICA-WED در حوزه دوام، استفاده از تعداد بازه های ثابت حضور در حالت انتظار می تواند تعداد دفعات نوشتن بر روی دیسک های حالت جامد را نسبت به TICA-EF تا ۹۰٪ افزایش می دهد [۸].

استفاده از تعداد بازه های ثابت و محدود حضور در حالت انتظار، می تواند منجر به چرخه های از درج صفحات حاصل از فقدان خواندن، بدون افزایش نرخ برخورد شود. به عنوان مثال، اگر در یک بارکاری اکثر درخواست ها الگوی دسترسی یک باره داشته باشند، استفاده از تعداد بازه های ثابت تنها منجر به کاهش طول عمر خواهد شد. مثلاً حالتی را در نظر بگیرید که در آن تعداد بازه های حضور در حالت انتظار برابر با ۴ باشد. با استفاده از روش TICA-A در یک بارکاری با نرخ برخورد پایین و فقدان خواندن بالا، به ازای هر ۶ بازه نمونه برداری، در یک بازه از سیاست WED استفاده می شود. فرض کنید این بارکاری شامل ۱۰۰۰ بازه نمونه برداری باشد. استفاده از TICA-A منجر به استفاده از سیاست WED در ۱۶۷ بازه می شود. از طرفی دیگر اگر تعداد بازه های حضور در حالت انتظار برابر با مقداری بزرگ و ثابت باشد، شانس پر شدن دیسک حالت جامد از داده های داغ تر، کاهش می یابد.

به منظور غلبه بر این مشکل، روش درج محافظه کارانه مبتنی بر ماشین حالت (CSMI) پیشنهاد می شود. در CSMI، تلاش می شود تا تعداد بازه های حضور در حالت انتظار به صورت پویا و بر اساس رفتار بارکاری تغییر کند. به این ترتیب، با افزایش تعداد بازه های استفاده از سیاست EF، طول عمر حافظه نهان افزایش می یابد. شکل ۴، ماشین حالت تغییر یافته مربوط به CSMI را نشان می دهد. به منظور تغییر پویای تعداد بازه های حضور در حالت انتظار، در CSMI از عاملی به نام ضریب محافظه کاری^{۵۴} (CF) استفاده می شود. وظیفه CF، افزایش تعداد بازه های انتظار در هنگام رخ دادن نرخ برخورد پایین در بازه های استفاده از سیاست WED است.

در هر بار گذار ماشین حالت به حالت انتظار، این امکان وجود دارد که بارکاری رفتاری مشابه الگوی دسترسی یک باره داشته باشد. ضریب محافظه کاری موظف است تعداد بازه های حضور در حالت

انتظار را برای دفعه بعدی حضور در این حالت، افزایش دهد. به این منظور پیش از ورود به حالت انتظار، تعداد بازه های حضور در حالت انتظار (Steps) در ضریب محافظه کاری ضرب می شود تا به این ترتیب، در دفعه بعدی که ماشین حالت در حالت انتظار قرار گیرد، بازه های استفاده از سیاست EF افزایش یابد. به منظور تطبیق بهتر ماشین حالت با تغییر رفتار بارکاری، در بازه هایی که بارکاری نرخ برخورد بالایی ارائه می دهد، مقدار Steps مجدداً بازنشانی^{۵۵} می شود.

به این ترتیب، CSMI می تواند با افزایش تعداد بازه های حضور در حالت انتظار، تعداد بازه های استفاده از سیاست WED که منجر به بهبود کارایی نمی شود را کاهش دهد. مثلاً در مثال بالا (بارکاری با نرخ برخورد پایین و فقدان خواندن بالا شامل ۱۰۰۰ بازه نمونه برداری) به ازای ضریب محافظه کاری برابر با ۲، تنها در ۸ بازه از سیاست WED استفاده می شود. به این ترتیب، روش پیشنهادی نسبت به روش TICA-A، ۹۵.۲٪ تعداد بازه های استفاده از سیاست WED را کاهش داده است.

به صورت کلی، در حالت خاصی^{۵۶} که یک بارکاری با نرخ بسیار بالای فقدان خواندن و الگوی دسترسی با محلیت بسیار کم داشته باشیم، نرخ استفاده از سیاست WED در روش TICA-A رابطه ۱ به دست می آید:

$$\text{Ratio of WED} = \frac{N}{N(2+Steps)} \quad (1)$$

که در این رابطه N تعداد بازه های استفاده از سیاست WED و Steps تعداد بازه های حضور در حالت انتظار است. با توجه به این که در حالت فوق، در هر چرخه^{۵۷} ماشین حالت، در یک بازه از سیاست WED استفاده می شود، N برابر تعداد چرخه های ماشین حالت نیز هست. نرخ استفاده از سیاست WED در روش CSMI از رابطه ۲ به دست می آید:

$$\text{Ratio of WED} = \frac{N}{2N + \frac{Steps(1-CFN+1)}{(1-CF)}} \quad (2)$$

که در این رابطه N تعداد بازه های استفاده از سیاست WED، CF ضریب محافظه کاری و Steps تعداد بازه های پیش فرض حضور در حالت انتظار است. در این روش، تعداد بازه های حضور در حالت انتظار از یک تصاعد هندسی با قدر نسبت CF پیروی می کند. به این ترتیب، برای مقادیر بزرگ N، روش پیشنهادی می تواند تعداد بازه های استفاده از سیاست WED را به طور قابل توجهی کاهش دهد و تعداد بازه های استفاده از سیاست WED نسبت به روش TICA-A، به سمت صفر میل می کند.

الگوریتم ۱: درج محافظه کارانه مبتنی بر ماشین حالت

Procedure CSMI :

```

1 Initial SP: Sampling_Period
2 Initial Current_State = Initial_State
3 Initial at Beginning of each SP HDD_Read=0, Read_Hit=0,
  Request_Counter=0
4 Counter=Steps
5 For Each Request
6   Request_Counter++
7   IF Requested Page was Read
8     IF Requested Page was in Cache
9       Read_Hit++
10    ELSE
11      HDD_Read++
12  IF Request_Counter == SP
13    IF Current_State == Initial_State
14      IF (HDD_Read/SP) > THigh_HDD_Access
15        USE WED
16        Counter = Steps
17        Current_State = WED_State
18    ELSE IF Current_State == WED_State
19      IF (HDD_Read/SP) > TStill_High_HDD_Access
20        IF (Read_Hit/SP) > TGood_Read_Hit
21          USE WED Policy
22          Current_State = WED_State
23          Reset Steps
24        ELSE
25          USE EF
26          Current_State = Wait_State
27          Steps = Steps * CF
28      ELSE IF (Read_Hit/SP) > TGood_Read_Hit
29        USE EF Policy
30        Reset Steps
31        Current_State = Initial_State
32    ELSE IF Current_State == Wait_State
33      IF Counter == 0
34        USE EF
35        Counter = Steps
36        Current_State = Initial_State
37      ELSE IF (Read_Hit/SP) > TGood_Read_Hit
38        USE EF
39        Reset Steps
40        Current_State = Initial_State
41      ELSE
42        Counter --

```

آن توجه داشت، تغییر احتمالی رفتار بارکاری در هنگام حضور ماشین حالت در حالت انتظار است. این تغییر رفتار می تواند به این شکل باشد که بارکاری تعدادی درخواست نوشتن ارسال کند و سپس داده های نوشته شده را بخواند. در نتیجه دیسک های حالت جامد از داده های داغ تر پر می شوند و نرخ برخورد بالا خواهد رفت. همچنین ممکن است بارکاری ابتدا الگوی دسترسی یک باره داشته باشد و سپس الگوی دسترسی تغییر کند. اگر مدت زمان حضور در حالت انتظار طولانی باشد و روشی جهت خروج از آن وجود نداشته باشد، اثر تغییر رفتار بارکاری دیرتر تشخیص داده می شود. به منظور رفع این مشکل، در حالت انتظار، نرخ برخورد نیز بررسی می شود. اگر نرخ برخورد از آستانه ای ($T_{Good_Read_Hit}$) بالاتر رود، ماشین حالت بی توجه به مدت زمانی که از حضور در حالت انتظار

الگوریتم ۱ چگونگی کارکرد CSMI را نشان می دهد. با توجه به عملکرد خوب سیاست EF در حوزه دوام، در حالت اولیه و حالت انتظار از این سیاست استفاده می شود. در طول هر بازه نمونه برداری، تعداد برخوردهای خواندن و تعداد دفعات فقدان خواندن (خواندن از دیسک سخت) محاسبه می شود. در پایان بازه، اگر میزان دسترسی ها به دیسک سخت از یک مقدار آستانه ($T_{High_HDD_Access}$) بیشتر بود، مقدار Counter برابر Steps تعیین شده و سپس ماشین حالت به حالت WED رفته و شروع به درج صفحات حاصل از فقدان خواندن در WO-SSD می کند (خطوط ۱۴ تا ۱۷ الگوریتم ۱).

در حالت WED، اگر نرخ برخورد خواندن از آستانه ای ($T_{Good_Read_Hit}$) بیشتر شود و دسترسی ها به دیسک سخت کاهش یابد، این احتمال داده می شود که WO-SSD از داده های داغ پر شده باشد. در نتیجه ماشین حالت مجدداً به حالت اولیه می رود و مقدار Steps نیز بازنشانی شده و برابر با مقدار اولیه قرار داده می شود (خطوط ۲۸ تا ۳۱ الگوریتم ۱). همچنین اگر نرخ برخورد افزایش یابد ولی کماکان دسترسی ها به دیسک سخت از آستانه ای ($T_{Still_High_HDD_Access}$) بیشتر باشد، ماشین حالت کماکان در حالت WED می ماند چرا که WO-SSD هنوز از داده های داغ پر نشده است. با روی دادن این وضعیت نیز مقدار Steps بازنشانی می شود. دلیل بازنشانی مقدار Steps این است که با روی دادن این وضعیت ها، استفاده از سیاست WED منجر به افزایش کارایی شده است. به این ترتیب نیازی به تعداد بازه های انتظار زیاد نیست (خطوط ۱۹ تا ۲۳ الگوریتم ۱).

در صورتی که استفاده از WED منجر به افزایش نرخ برخورد نشود، این امکان وجود دارد که تداوم استفاده از این سیاست تنها منجر به کاهش طول عمر حافظه نهان شود. به همین منظور ماشین حالت به حالت انتظار می رود. همچنین با توجه به این که استفاده از WED منجر به بهبود کارایی نشده، تعداد بازه های انتظار (Steps) در ضریب محافظه کاری (CF) ضرب می شود تا در دفعه بعدی که ماشین حالت به حالت انتظار می رود، زمان بیشتری را در این حالت بماند. به این ترتیب، از دیسک حالت جامد در برابر درج بیهوده صفحات حاصل از فقدان خواندن، محافظت می شود (خطوط ۲۵ تا ۲۷ الگوریتم ۱).

در حالت انتظار، ماشین حالت از سیاست EF استفاده می کند. تعداد بازه هایی که ماشین حالت در حالت انتظار به سر می برد برابر با Counter است و با صفر شدن Counter، ماشین حالت مجدداً به حالت اولیه می رود تا دوباره با دادن شانس به سیاست WED، اثر استفاده از این سیاست را بررسی کند. از جمله عواملی که باید به

شکل، CSMI-TICA به ازای ضریب محافظه کاری‌های مختلف، نرخ برخورد تقریباً یکسانی نسبت به TICA-A ارائه می‌دهد. CSMI-TICA-CF4 که محافظه کارانه ترین ضریب را در بین پیکربندی‌های مختلف CSMI داراست، به طور میانگین، تنها ۰.۳۸٪ نرخ برخورد را نسبت به TICA-A کاهش می‌دهد.

دلیل عملکرد تقریباً مشابه روش CSMI-TICA نسبت به TICA-A، عکس‌العمل سریع این روش به تغییرات است. در CSMI با رخ دادن هر گونه تغییر در رفتار بارکاری (نرخ برخورد بازه‌ها)، ماشین حالت، فارغ از تعداد بازه‌هایی که باید در حالت انتظار سپری کند، مجدداً به حالت اولیه بر می‌گردد و تغییراتی که بر روی Steps اعمال کرده را بی‌اثر می‌کند. به این ترتیب، ماشین حالت این امکان را دارد که با رخ دادن تغییر در رفتار بارکاری، با ادامه درخواست‌ها به مانند یک بارکاری جدید رفتار کند و با همان سرعتی که TICA-A به تغییرات واکنش نشان می‌دهد، تغییر حالت بدهد. از طرف دیگر CSMI موفق شده تا در بازه‌هایی که سیاست WED بهره زیادی در حوزه کارایی ندارد، استفاده از این سیاست را متوقف سازد و به این ترتیب، علی‌رغم استفاده بیشتر از سیاست EF، نرخ برخورد تقریباً یکسانی نسبت به TICA-A ارائه بدهد.

از جمله عوامل مهم دیگر، تاثیر ضریب محافظه کاری‌های مختلف بر روی نرخ برخورد CSMI است. مطابق شکل ۵، ضریب محافظه کاری‌های مختلف تاثیر بسیار کمی در نرخ برخورد دارد. دلیل این اتفاق، تاثیر کم استفاده از سیاست WED در این بارهای کاری است. در واقع حالت WED نمی‌تواند به اندازه کافی WO-SSD را از داده‌های داغ پر کند و تاثیر آن در افزایش نرخ برخورد کم‌تر از تغییر رفتار بارکاری و اثر EQ است. به این ترتیب، کاهش CF، که منجر به افزایش درج در WO-SSD می‌شود، نمی‌تواند تاثیر زیادی بر نرخ برخورد گذارد و صرفاً طول عمر حافظه نهان را کاهش می‌دهد.

تاثیر اندک حاصل از تغییر مقدار CF بر نرخ برخورد به سبب فرصت بیشتری است که مقادیر کم‌تر CF به صفحات حاصل از فقدان خواندن می‌دهند. به عنوان مثال در بارکاری Hm_1، ضریب محافظه کاری برابر با ۲ توانسته نرخ برخورد بالاتری نسبت به دو مقدار دیگر فراهم کند. استفاده از این مقدار منجر می‌شود ماشین حالت تعداد بازه‌های کم‌تری را در حالت انتظار سپری کند. در نتیجه از سیاست WED بیشتر استفاده می‌شود و با درج صفحات حاصل از این سیاست در WO-SSD، این امکان فراهم

طی شده، به حالت اولیه خواهد رفت و به سبب این تغییر، Steps نیز بازنشانی می‌شود. دلیل بازنشانی این است که حدس می‌زنیم رفتار بارکاری تغییر کرده و دیگر نیازی به صرف زمان زیاد جهت حضور در حالت انتظار نیست (خطوط ۳۳ تا ۴۲ الگوریتم ۱).

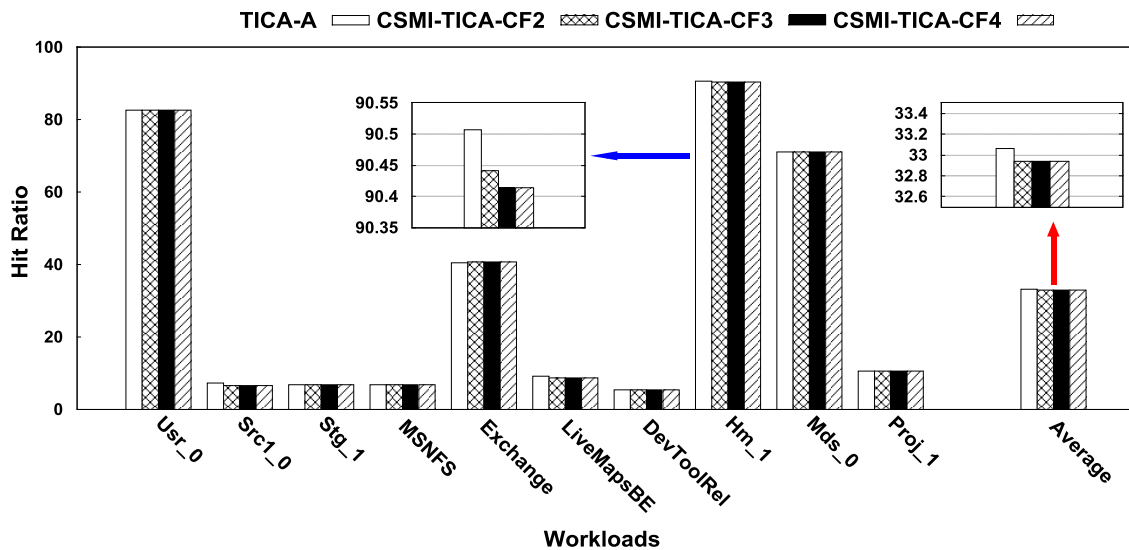
۴- ارزیابی

در این فصل به بررسی عملکرد روش پیشنهادی در حوزه‌های طول عمر و کارایی می‌پردازیم. به این منظور، یک شبیه‌ساز مبتنی بر ردگیری^{۵۸} توسعه داده می‌شود که نرخ برخورد و تعداد دفعات نوشتن بر روی دیسک‌های حالت جامد را برای هر بارکاری استخراج می‌کند. برای ارزیابی زمان پاسخ، از روش Record & Replay و ادوات واقعی استفاده می‌شود. همچنین از ابزار Samsung Magician DC [۴۱] برای بررسی پارامتر Writes (تعداد بخش‌های^{۵۹} فلش نوشته شده) به منظور ارزیابی دقیق‌تر طول عمر استفاده می‌شود. به منظور ارزیابی CSMI، این روش بر روی معماری سه سطحی (TICA) اعمال می‌شود. برای ارزیابی CSMI-TICA، ۱۰ بارکاری از مجموعه‌های MSRC [۱۹] و SNIA [۲۰][۲۱] مورد استفاده قرار می‌گیرد.

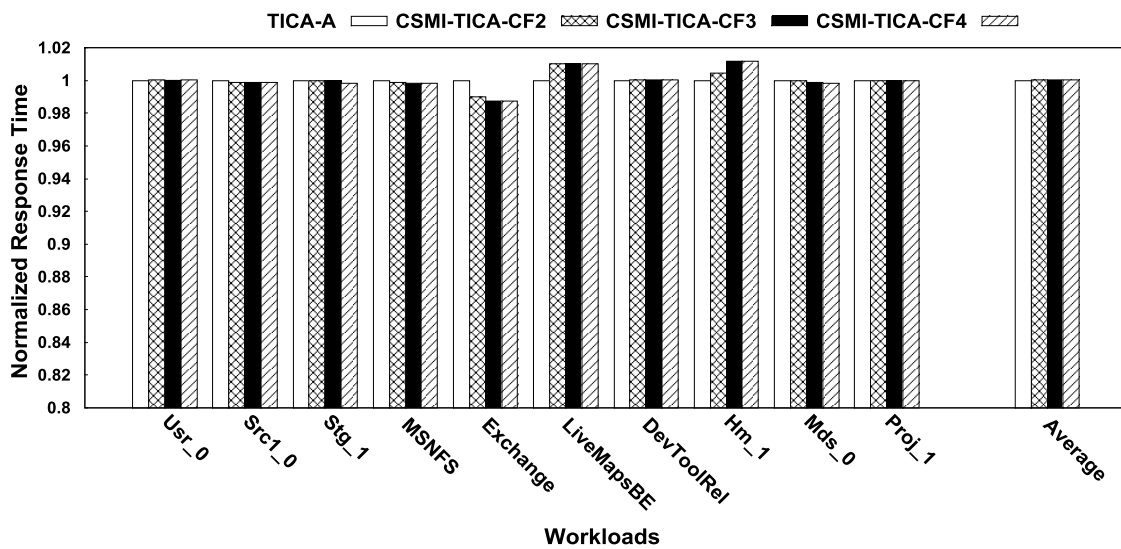
به منظور مقایسه، روش تطبیقی پیشنهاد شده در [۸] (TICA-A) پیاده‌سازی می‌شود و اندازه DRAM و دیسک‌های حالت جامد مطابق مقادیر فرض شده در [۸] قرار داده می‌شود. همچنین مقدار آستانه‌های مورد استفاده نیز برابر با ۰.۲ برای THigh_HDD_Access، ۰.۱ برای TStill_High_HDD_Access و ۰.۲ برای TGood_Read_Hit قرار داده می‌شود. مقدار اولیه Steps نیز مانند [۸] برابر با ۴ در نظر گرفته می‌شود. مقادیر اولیه برای آستانه‌ها و Steps مورد استفاده عیناً برابر با مقادیر مطرح شده در [۸] قرار داده شده‌است. دلیل انتخاب این مقادیر، یکسان بودن شرایط آزمایش‌ها بوده‌است تا صرفاً اثر روش CSMI به عنوان یک روش پویا مورد ارزیابی قرار گیرد. عامل دیگری که باید در نظر گرفته شود، تاثیر CF بر روی کارکرد روش CSMI است. به این منظور، سه مقدار ۲ (CSMI-TICA-CF2)، ۳ (CSMI-TICA-CF3) و ۴ (CSMI-TICA-CF4) برای CF در نظر گرفته می‌شود تا تاثیر این ضریب بر روی کارایی و طول عمر ارزیابی شود.

۴-۱- کارایی

به منظور ارزیابی چگونگی عملکرد CSMI-TICA در حوزه کارایی، به بررسی نرخ برخورد این روش در بارهای کاری مختلف می‌پردازیم. شکل ۵، نرخ برخورد TICA-A و CSMI-TICA به ازای ضریب محافظه کاری‌های مختلف را نشان می‌دهد. مطابق این



شکل ۵- نرخ برخورد



شکل ۶- زمان پاسخ

کارایی نسبت به TICA-A دارد. قابل توجه‌ترین تغییرات مربوط به بارهای کاری Exchange و MSNFS، Hm_1، LiveMapsBE است که در این بارهای کاری نیز تفاوت ناچیز است (بیشترین افت کارایی در Hm_1 با ۱.۱٪ کاهش و بیشترین بهبود در Exchange با ۰.۵٪ افزایش کارایی). تفاوت‌های موجود به سبب تفاوت بین کارایی اجزای حافظه نهان و تفاوت در نرخ برخورد است که منجر به تفاوت جزئی شده‌است.

۴-۲- طول عمر

هدف اصلی روش CSMI، بهبود طول عمر دیسک‌های حالت جامد است. به منظور ارزیابی تاثیر TICA-A و CSMI، تعداد نوشتن‌های انجام شده توسط هر کدام از این روش‌ها بر روی دیسک‌های حالت جامد را بررسی می‌کنیم. شکل ۷ تعداد نوشتن‌های انجام شده بر روی دیسک‌های حالت جامد توسط

می‌شود تا دسترسی‌های بعدی از WO-SSD پاسخ داده شود. با این وجود تاثیر این صفحات در نرخ برخورد کلی حافظه نهان ناچیز است.

نرخ برخورد پارامتری است که شهودی کلی از چگونگی عملکرد روش پیشنهادی در حوزه کارایی ارائه می‌دهد. با این وجود، با توجه به تفاوت موجود در کارایی اجزای مختلف حافظه نهان، نرخ برخورد نمی‌تواند لزوماً پارامتری جامع و دقیق برای ارزیابی کارایی باشد. به منظور بررسی دقیق‌تر کارایی روش پیشنهادی، زمان پاسخ روش پیشنهادی با استفاده از روش Record & Replay [۸] و با استفاده از ادوات واقعی مورد ارزیابی قرار می‌گیرد. شکل ۶، زمان پاسخ نرمال شده روش پیشنهادی نسبت به روش TICA-A را نشان می‌دهد.

مطابق شکل ۶، CSMI-TICA تغییرات بسیار اندکی در حوزه

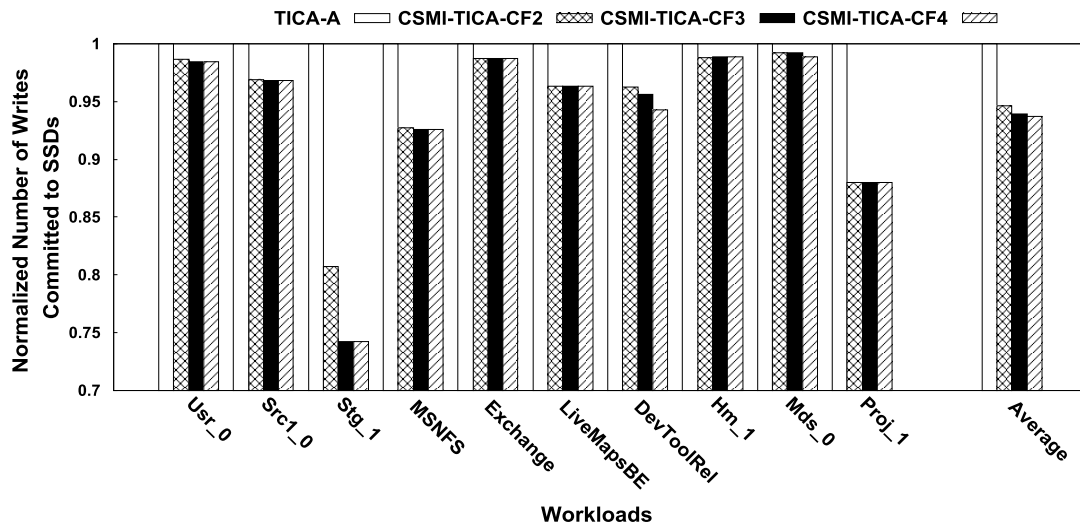
می‌شود در بارهای کاری Stg_1, Usrc_0, MSNFS و Src1_0 افزایش ضریب محافظه کاری از دو به سه منجر به کاهش بیشتر نوشتن‌ها می‌گردد. دلیل این کاهش، افزایش سریع‌تر تعداد بازه‌های حضور در حالت انتظار است. با این حال کاهش ضریب محافظه کاری از سه به چهار منجر به تفاوت قابل توجهی در میزان نوشتن‌ها نمی‌شود. دلیل این عدم کاهش، افزایش زیاد تعداد بازه‌های حضور در حالت انتظار است که اثر تعداد بازه‌های استفاده از سیاست WED را کم می‌کند. با این وجود کاهش ضریب محافظه کاری در بارهای کاری Mds_0 و DevToolRel از سه به چهار منجر به کاهش میزان نوشتن شده‌است.

باید به این نکته توجه داشت که افزایش مقادیر CF الزاما منجر به بهبود بیشتر دوام نمی‌شود. ساختار ماشین حالت به گونه‌ای است که تلاش می‌کند به مصالحه‌ای بین کارایی و دوام دست یابد. به این ترتیب، روش پیشنهادی تلاش می‌کند تا در بازه‌هایی با

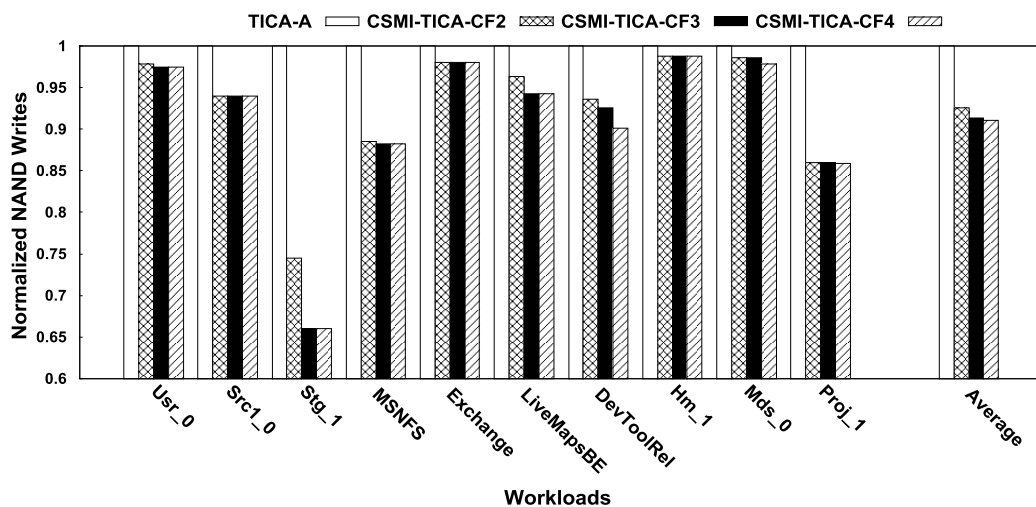
TICA-A و CSMI-TICA با ضریب محافظه کاری‌های مختلف را نشان می‌دهد. نتایج هر ستون به تعداد نوشتن‌های انجام شده توسط TICA-A نرمال شده‌است.

به طور میانگین، روش CSMI-TICA با ضریب محافظه کاری‌های ۲، ۳ و ۴ به ترتیب تعداد دفعات نوشتن بر روی دیسک‌های حالت جامد را ۵.۳۸٪، ۶.۱۲٪ و ۶.۲۷٪ کاهش داده‌است. دلیل این کاهش، استفاده کم‌تر این روش از سیاست WED است که منجر به افزایش طول عمر حافظه نهان شده‌است. همان‌طور که مشاهده شد، افزایش تعداد بازه‌های حالت انتظار نه تنها تاثیر قابل توجهی بر روی نرخ برخورد نداشته بلکه می‌تواند تا ۲۵.۷۴٪ (Stg_1) تعداد دفعات نوشتن بر روی دیسک‌های حالت جامد را کاهش دهد.

هر چه رفتار بارکاری به الگوی دسترسی یک‌باره نزدیک‌تر باشد، بهبود حاصل از روش CSMI بیشتر خواهد شد. همچنین مشاهده



شکل ۷- تعداد نوشتن‌های انجام شده بر روی دیسک‌های حالت جامد



شکل ۸- معیار NAND Writes

Writes در تمام بارهای کاری بهبود بیشتری فراهم می‌کند. دلیل این بهبود، مکانیزم‌های داخلی دیسک حالت جامد نظیر زیاله‌روبی است که در بررسی‌های سطح سامانه در نظر گرفته نمی‌شود. شایان ذکر است هر چه تفاوت تعداد نوشتن‌های ارسال شده بین دو روش بیشتر باشد، بهبود در معیار NAND Writes نیز بیشتر می‌شود. به عنوان مثال می‌توان به بارهای کاری Stg_1، Proj_1، DevToolRel، MSNFS و LiveMapsBE اشاره کرد. به صورت میانگین، روش CSMI-TICA برای ضرایب محافظه‌کاری با مقادیر ۲، ۳ و ۴ به ترتیب ۷.۴٪، ۸.۶۳٪ و ۸.۹۴٪ میزان NAND Writes را نسبت به TICA-A کاهش داده‌است (با تغییر ادوات مورد استفاده، امکان تغییر نتایج فوق وجود دارد).

۵- جمع‌بندی و کارهای آتی

استفاده از معماری‌های ترکیبی حافظه نهان، رویکردی کارا به منظور بهبود نهان‌سازی ورودی/خروجی است. از جمله مهم‌ترین عوامل در مدیریت معماری‌های ترکیبی، مدیریت درج صفحات حاصل از فقدان خواندن در لایه‌های مختلف حافظه نهان است. در این مقاله، یک روش محافظه‌کارانه به منظور مدیریت فقدان خواندن در ادوات با طول عمر محدود ارائه گردید. در این روش با تغییر ماشین حالت پیشنهاد شده در [۸]، مدت زمان عدم درج صفحات در دیسک حالت جامد افزایش می‌یابد. ارزیابی‌های انجام شده نشان داد این روش تا ۳۳.۹۷٪، تعداد دفعات نوشتن بر روی دیسک‌های حالت جامد را نسبت به بهترین پژوهش مشابه، بهبود داده‌است.

استفاده از دیسک‌های حالت جامد ناهمگون و معماری ترکیبی چندسطحی حافظه نهان، رویکرد دیگری برای سامانه‌های ذخیره‌سازی با کارایی بالاست که هزینه بیشتری نسبت به سامانه‌های سنتی ذخیره‌سازی در پی دارد. استفاده از این معماری مناسب کاربردهایی است که به دنبال سرعت بالای ذخیره‌سازی و قابلیت اطمینان بالا هستند و به منظور دستیابی به این خصوصیات، حاضر به پرداخت هزینه بالاتر استفاده از دیسک‌های حالت جامد مختلف و همین‌طور DRAM می‌باشند. در کارهای آتی می‌توان با تغییر بازه‌های نمونه‌برداری به صورت پویا، حساسیت ماشین حالت پیشنهاد شده به تغییرات بارکاری را افزایش داد. استفاده از دیسک‌های حالت جامد با اندازه‌های متفاوت و بهینه‌سازی روش پیشنهادی بر مبنای این تغییر و بهینه‌سازی روش پیشنهادی بر پایه استفاده از حافظه‌های غیرفرار نوظهور [۲۷][۲۸][۲۹] نیز می‌تواند از جمله کارهای آتی باشد.

محلیت کم درخواست‌ها، رویکردی با تمرکز بر بهبود دوام در پیش گیرد و این امکان وجود دارد که با مقادیر کم CF نیز به حداکثر کاهش قابل دست‌یابی برسیم. در نتیجه افزایش CF الزام نمی‌تواند منجر به بهبود بیشتر طول عمر شود. به منظور افزایش بیشتر طول عمر، می‌توان از روش ایستای TICA-EF [۸] استفاده کرد. شایان ذکر است در [۸]، روش TICA-A به منظور بهبود کارایی روش TICA-EF ارائه شده و این روش در حوزه دوام، بدتر از روش TICA-EF عمل می‌کند.

عامل مهم دیگر در حوزه دوام، تاثیر مقادیر مختلف ضریب محافظه‌کاری بر روی دوام حافظه نهان است. به جز بارکاری DevToolRel، در سایر بارهای کاری ضرایب استفاده شده منجر به پایداری عملکرد CSMI-TICA در حوزه طول عمر شده‌است. در حقیقت، یا تاثیر تغییر رفتار بارکاری بیشتر از تاثیر درج مبتنی بر ماشین حالت است، و یا تعداد بازه‌های انتظار به حدی افزایش یافته (و استفاده از WED کاهش یافته) که اثر کمی بر طول عمر می‌گذارد. با توجه به این‌که ضریب محافظه‌کاری ۴ نسبت به ضریب محافظه‌کاری ۲، به طور میانگین، ۱٪ تعداد دفعات نوشتن بر روی دیسک حالت جامد را کاهش داده‌است، استفاده از این مقدار می‌تواند منجر به بهبود بیشتر طول عمر حافظه نهان شود.

بررسی طول عمر با استفاده از ادوات واقعی: به منظور بررسی دقیق‌تر حوزه طول عمر، از ابزار Samsung Magician DC [۴۱] برای ارزیابی تاثیر روش پیشنهادی بر طول عمر ادوات واقعی استفاده می‌شود. این ابزار با استفاده از قابلیت SMART⁶⁰، پارامترهای مختص هر دستگاه در حوزه طول عمر، خرابی و قابلیت اطمینان را ارائه می‌دهد. با توجه به عدم دسترسی به مکانیزم‌های داخلی دیسک‌های حالت جامد و چگونگی کارکرد لایه ترجمه فلش^{۶۱} (FTL)، امکان ارائه تعداد دفعات نوشته شدن یک صفحه فلش وجود ندارد. در نتیجه، معیار NAND Writes که توسط ابزار فوق گزارش می‌شود به عنوان معیار دقیق‌تر ارزیابی طول عمر در نظر گرفته می‌شود.

شکل ۸، عملکرد روش پیشنهادی نسبت به TICA-A را در معیار NAND Writes نشان می‌دهد. نتایج نسبت به TICA-A نرمال شده‌است. مطابق این شکل، در معیار NAND Writes نیز روش پیشنهادی عملکرد بهتری نسبت به TICA-A ارائه می‌دهد. نکته مهم در این نتایج، افزایش بهبود حاصل شده توسط روش CSMI-TICA است. با مقایسه شکل ۷ و شکل ۸ که تعداد نوشتن‌های ارسال شده به دیسک‌های حالت جامد را نشان می‌دهد، مشاهده می‌شود روش CSMI-TICA در معیار NAND

on *Mass Storage Systems and Technologies (MSST)*, Long Beach, CA, pp. 1-14, May 2013.

- [17] R. S. Liu, C. L. Yang, C. H. Li, and G. Y. Chen, "DuraCache: A Durable SSD Cache Using MLC NAND Flash," *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, USA, pp. 1-6, Jun. 2013..
- [18] Y. Klonatos, T. Makatos, M. Marazakis, M. Flouris, and A. Bilas, "Azor: Using Two-Level Block Selection to Improve SSD-based I/O Caches," *6th IEEE International Conference on Networking, Architecture and Storage (NAS)*, Dalian, China, pp.309-318, Jul. 2011.
- [19] D. Narayanan, A. Donnelly, and A. Rowstron, "Write Off-Loading: Practical Power Management for Enterprise Storage," *ACM Transactions on Storage (TOS)*, vol. 4, no. 3, pp. 1–23, Nov. 2008.
- [20] Storage Networking Industry Association IOTTA Repository, "Microsoft Enterprise Traces," [Online]. Available: <http://iotta.snia.org/traces/130>. Accessed: 01/01/2018.
- [21] Storage Networking Industry Association IOTTA Repository, "Microsoft Production Server Traces," [Online]. Available: <http://iotta.snia.org/traces/158>. Accessed: 01/01/2018.
- [22] S. He, Y. Wang, Z. Li, X. H. Sun, and C. Xu, "Cost-Aware Region-Level Data Placement in Multi-Tiered Parallel I/O Systems," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 28, no. 7, pp. 1853-1865, Jul. 2017.
- [23] R. Salkhordeh, H. Asadi, and S. Ebrahimi, "Operating System Level Data Tiering Using Online Workload Characterization," *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1534-1562, Apr. 2015.
- [24] D. Lee, S. Yoon, J. Kim, C. C. Weems, and S. D. Kim, "A New Memory-Disk Integrated System with HW Optimizer," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 12, no. 2, pp. 11:1-11:23, Apr. 2015.
- [25] S. K. Yoon, M. Y. Bian, and S. D. Kim, "An Integrated Memory-Disk System with Buffering Adapter and Non-Volatile Memory," *Springer Design Automation for Embedded System*, vol. 17, no. 3, pp. 609-626, Sep. 2013.
- [26] Y. Cheng, W. Chen, Z. Wang, X. Yu, and Y. Xiang, "AMC: an adaptive multi-level cache algorithm in hybrid storage systems," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 16, pp. 4230-4246, Nov. 2015.
- [27] R. Salkhordeh and H. Asadi, "An Operating System level data migration scheme in hybrid DRAM-NVM memory architecture," *51st IEEE/ACM Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, pp. 936-941, Mar. 2016.
- [28] E. Cheshmikhani, H. Farbeh, S.G. Miremadi, and H. Asadi, "TA-LRW: A Replacement Policy for Error Rate Reduction in STT-MRAM Caches," *IEEE Transactions on Computers (TC)*, Vol. 68, No. 3, March 2019.
- [29] R. Salkhordeh, O. Mutlu, and H. Asadi, "An Analytical Model for Performance and Lifetime Estimation of Hybrid DRAM-NVM Main Memories," *IEEE Transactions on Computers (TC)*, In Press, 2019.
- [30] R. Koller, A. J. Mashtizadeh and R. Rangaswami, "Centaur: Host-Side SSD Caching for Storage Performance Control," *2015 IEEE International Conference on Autonomic Computing (ICAC)*, Grenoble, France, pp. 51-60, Jul. 2015.
- [31] S. Ahmadian, O. Mutlu, and H. Asadi, "ECI-Cache: A High-Endurance and Cost-Efficient I/O Caching Scheme for Virtualized Platforms," *ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*, Irvine, CA, USA, pp. 9:1-9:34, Jun. 2018.
- [32] Z. Shen, F. Chen, Y. Jia, and Z. Shao, "DIDACache: A Deep Integration of Device and Application for Flash Based Key-Value Caching," *15th USENIX Conference on File and Storage Technologies (FAST)*, Santa Clara, CA, USA, pp. 391-405, Feb. 2017.
- [33] S. Xu, S. Lee, S. W. Jun, M. Liu, and J. Hicks, "Bluecache: A Scalable Distributed Flash-Based Key-Value Store," *42nd international conference on very large data bases (VLDB)*, New Delhi, India, pp. 301-312, Mar. 2016.
- [1] M. Khoso, "How Much Data Is Produced Every Day?," [Online]. Available: <http://www.northeastern.edu/levelblog/2016/05/13/how-much-data-produced-every-day/>. Accessed: 01/01/2018.
- [2] Violin Memory, "The Importance of How Data is Stored," [Online]. Available: <https://www.violinsystems.com/all-flash-storage/importance-of-data-storage/>. Accessed: 27/02/2019.
- [3] Intel, "Game-changing Fast Memory Also a Big Deal for Big Data," [Online]. Available: <https://newsroom.intel.com/editorials/3d-xpoint-memory-storage>. Accessed: 23/06/2018.
- [4] F. Chen, D. A. Koufaty, and X. Zhang, "Understanding Intrinsic Characteristics and System Implications of Flash Memory Based Solid State Drives," *32th International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, USA, pp. 181-192, Aug. 2009.
- [5] F. Chen, D. A. Koufaty, and X. Zhang, "Hystor: Making the Best Use of Solid State Drives in High Performance Storage Systems," *25th International Conference on Supercomputing (ICS)*, Tucson, AR., USA, pp. 22–32, Jun. 2011.
- [6] S. K. Yoon, Y. S. Youn, S. J. Nam, M. H. Son, and S. D. Kim, "Optimized Memory-Disk Integrated System with DRAM and Nonvolatile Memory," *IEEE Transactions on Multi-Scale Computing Systems (TMSCS)*, vol. 2, no. 2, pp. 83-93, Apr. 2016.
- [7] R. Salkhordeh, S. Ebrahimi, and H. Asadi, "ReCA: an Efficient Reconfigurable Cache Architecture for Storage Systems with Online Workload Characterization," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 29, no.7, pp.1605-1620, Jul. 2018.
- [8] R. Salkhordeh, M. Hadizadeh, and H. Asadi, "An Efficient Hybrid I/O Caching Architecture Using Heterogeneous SSDs," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Vol. 30, Issue 6, pp. 1238-1250, Jun. 2019.
- [9] S. Huang, Q. Wei, J. Chen, C. Chen, and D. Feng, "Improving Flash-Based Disk Cache with Lazy Adaptive Replacement," *29th IEEE Symposium on Mass Storage Systems and Technologies (MSST)*, Long Beach, CA, USA, pp. 1-10, May 2013.
- [10] D. Jiang, Y. Che, J. Xiong, and X. Ma, "uCache: A Utility-Aware Multilevel SSD Cache Management Policy," *10th IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, Zhangjiajie, China, pp. 391-398, Nov. 2013.
- [11] Y. Ni, J. Jiang, D. Jiang, X. Ma, J. Xiong, and Y. Wang, "S-RAC: SSD Friendly Caching for Data Center Workloads," *9th ACM International System Storage Conference (SYSTOR)*, pp. 8:1-8:12, Jun. 2016.
- [12] L. Shi, J. Li, C. J. Xue, and X. Zhou, "Hybrid Nonvolatile Disk Cache for Energy-efficient and High-performance Systems," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 1, pp. 8:1-8:23, Jan. 2013.
- [13] Y. Liang, Y. Chai, N. Bao, H. Chen, and Y. Liu, "Elastic Queue: A Universal SSD Lifetime Extension Plug-in for Cache Replacement Algorithms," *9th ACM International Systems & Storage Conference (SYSTOR)*, pp. 5:1-5:11, Jun. 2016.
- [14] S. Huang, Q. Wei, D. Feng, J. Chen, and C. Chen, "Improving Flash-Based Disk Cache with Lazy Adaptive Replacement," *ACM Transactions on Storage (TOS)*, vol. 12, no. 2, pp. 8:1-8:24, Feb. 2016.
- [15] N. Megiddo and D. S. Modha, "Outperforming LRU with an Adaptive Replacement Cache Algorithm," *Computer*, vol. 37, no. 4, pp. 58-65, Apr. 2004.
- [16] R. Appuswamy, D. C. van Moolenbroek, and A. S. Tanenbaum, "Cache, Cache Everywhere, Flushing All Hits Down The Sink: On Exclusivity in Multilevel, Hybrid Caches," *IEEE 29th Symposium*

- [39] M. Kishani, M. B. Tahoori, and H. Asadi, "Dependability Analysis of Data Storage Systems in Presence of Soft Errors", *IEEE Transactions on Reliability (TR)*, Vol. 68, Issue 1, Mar. 2019.
- [40] S. Ahmadian, R. Salkhordeh, and H. Asadi, "LBICA: A Load Balancer for I/O Cache Architectures", *54th IEEE/ACM Design, Automation and Test in Europe Conference (DATE)*, Florence, Italy, pp. 1196-1201, Mar. 2019.
- [41] Samsung, "Samsung SSD Magician DC 2.0," [Online.] Available: http://downloadcenter.samsung.com/content/SW/201711/201711115131146153/Samsung_Magician_DC_Brand_v2_User_Guide.pdf. Accessed: 26/10/19.
- [42] Q. Xia and W. Xiao, "High-Performance and Endurable Cache Management for Flash-Based Read Caching," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 27, no. 12, pp. 3518-3531, Dec. 2016.
- [43] J. Wan, W. Wu, L. Zhan, Q. Yang, X. Qu, and C. Xie, "DEFT-Cache: A Cost- Effective and Highly Reliable SSD Cache for RAID Storage," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Orlando, FL, USA, pp. 102-111, May 2017.
- [34] Z. Shen, F. Chen, Y. Jia, Z. and Shao, "Optimizing Flash-based Key-value Cache Systems," *8th USENIX Conference on Hot Topics in Storage and File Systems (HotStorage)*, Denver, CO, USA, pp. 1-5, Jun. 2016.
- [35] J. J. Kan, C. Park, C. Ching, J. Ahn, Y. Xie, M. Pakala, and S. H. Kang, "A Study on Practically Unlimited Endurance of STT-MRAM," *IEEE Transactions on Electron Devices (TED)*, vol. 64, no. 9, pp. 3639-3646, Sep. 2017.
- [36] M. Tarihi, H. Asadi, A. Haghdoost, M. Arjomand, and H. Sarbazi-Azad, "A Hybrid Non-Volatile Cache Design for Solid-State Drives Using Comprehensive I/O Characterization," *IEEE Transactions on Computers (TC)*, vol. 65, no. 6, pp. 1678-1691, Jun. 2016.
- [37] J. Niu, J. Xu and L. Xie, "Hybrid Storage Systems: A Survey of Architectures and Algorithms," *IEEE Access*, vol. 6, pp. 13385-3406, Feb. 2018.
- [38] W. Fischer, "Linux Storage Stack Diagram," [Online]. Available: https://www.thomas-krenn.com/en/wiki/Linux_Storage_Stack_Diagram. Accessed: 08/04/2019.

زیرنویس ها:

- ³⁸ Reuse Distance
- ³⁹ Out-of-Place Update
- ⁴⁰ Plug-in
- ⁴¹ Virtual Systems
- ⁴² Key-Value
- ⁴³ Spin-Torque Transfer Magnetic RAM
- ⁴⁴ Phase-Change Memory
- ⁴⁵ Inclusive
- ⁴⁶ Exclusive
- ⁴⁷ Heterogeneous
- ⁴⁸ Write-Optimized
- ⁴⁹ Read-Optimized
- ⁵⁰ Evicted Queue
- ⁵¹ Sampling Periods
- ⁵² Initial State
- ⁵³ Wait State
- ⁵⁴ Conservative Factor
- ⁵⁵ Reset
- ⁵⁶ Corner Case
- ⁵⁷ Iteration
- ⁵⁸ Trace
- ⁵⁹ Sector
- ⁶⁰ Self-Monitoring, Analysis and Reporting Technology
- ⁶¹ Flash Translation Layer
- ¹ Quality of Service
- ² Reliability
- ³ Cost
- ⁴ Hard Disk Drives
- ⁵ Seek Time
- ⁶ Rotation
- ⁷ Solid-State Drives
- ⁸ Flash
- ⁹ Endurance
- ¹⁰ Tiering
- ¹¹ Memory-Disk Integrated Systems
- ¹² Main Memory
- ¹³ Write Pressure
- ¹⁴ Read Miss
- ¹⁵ Endurance-Friendly (EF)
- ¹⁶ Hit
- ¹⁷ State Machine Based Insertion (SMBI)
- ¹⁸ Efficient
- ¹⁹ Write to Endurable Device (WED)
- ²⁰ Conservative
- ²¹ Conservative State Machine Based Insertion
- ²² Conservatism Factor
- ²³ Hit Ratio
- ²⁴ Coupled
- ²⁵ Hierarchical
- ²⁶ Duplication
- ²⁷ Horizontal
- ²⁸ Caching
- ²⁹ Write-Through
- ³⁰ Write-Back
- ³¹ Error Transformation
- ³² Dynamic-Rate SSD
- ³³ Wear-out
- ³⁴ Miss
- ³⁵ Redundancy
- ³⁶ Block
- ³⁷ Error Correction Code