

УДК: 004.4:004.27:004.382

Бакайкина Виктория Геннадиевна,

Ассистент кафедры Программной инженерии

ФГБОУ ВО «Поволжский государственный университет телекоммуникаций и

информатики»

Ким Сергей Владимирович,

Студент

ФГБОУ ВО «Поволжский государственный университет телекоммуникаций и

информатики»

ОПЕРАЦИОННЫЕ СИСТЕМЫ ДЛЯ ПАМЯТИ-ОРИЕНТИРОВАННЫХ ВЫЧИСЛЕНИЙ: ПРЕОДОЛЕНИЕ ПАРАДИГМЫ ФОН НЕЙМАНА И ПРОБЛЕМЫ «СТЕНЫ ПАМЯТИ»

Аннотация: Современная компьютерная архитектура, основанная на классической парадигме фон Неймана, столкнулась с фундаментальным ограничением, известным как «стена памяти». Несоответствие между скоростью процессора и пропускной способностью подсистемы памяти приводит к значительным потерям производительности и энергии на перемещение данных. Развитие новых классов энергонезависимой памяти, интерконнектов типа Compute Express Link и концепции вычислений в памяти требует переосмысления роли операционной системы. Цель данной работы - системный анализ вызовов, которые гетерогенная память с вычислительными возможностями создает для традиционных подсистем управления памятью, планирования задач и организации файловых систем. В статье обосновывается тезис о необходимости разработки новых абстракций ядра для управления неоднородными вычислительными ресурсами, распределенными по иерархии памяти, что знаменует переход от процессо-центричной к памяти-ориентированной модели операционной системы.

Ключевые слова: память-ориентированные вычисления, операционная система, управление памятью, Compute Express Link, вычисления в памяти, гетерогенная архитектура.

OPERATING SYSTEMS FOR MEMORY-ORIENTED COMPUTING: OVERCOMING THE VON NEUMANN PARADIGM AND THE "MEMORY WALL" PROBLEM

Bakaikina Victoria Gennadievna,

Assistant Professor of Software Engineering

Volga Region State University of Telecommunications and Informatics

Kim Sergey Vladimirovich,

Student

Volga Region State University of Telecommunications and Informatics

Abstract: Modern computer architecture based on the classical von Neumann paradigm has encountered a fundamental limitation known as the "memory wall". The discrepancy between the processor speed and the bandwidth of the memory subsystem leads to significant losses in performance and energy spent on data movement. The development of new classes of non-volatile memory, Compute Express Link interconnects, and the concept of in-memory computing requires rethinking the role of the operating system. The purpose of this work is a system analysis of the challenges that heterogeneous memory with computing capabilities creates for traditional memory management, task planning, and file system organization subsystems. The article substantiates the thesis about the need to develop new kernel abstractions for managing heterogeneous computing resources distributed over a memory hierarchy, which marks the transition from a processor-centric to a memory-oriented operating system model.

Keywords: memory-oriented computing, operating system, memory management, Compute Express Link, in-memory computing, heterogeneous architecture.

Введение

Операционные системы исторически развивались как уровень абстракции, скрывающий аппаратную неоднородность от приложений. Успешными примерами таких абстракций являются файл для дискового пространства, сокет для сетевого взаимодействия и виртуальная память для физической памяти. Однако текущая волна инноваций в полупроводниковых технологиях ставит под сомнение эффективность этих устоявшихся моделей. Появление энергонезависимой

оперативной памяти, стандарта Compute Express Link для создания пулов разделяемой памяти и архитектур Process-in-Memory ведет к радикальному усложнению подсистемы памяти. Она перестает быть пассивным, однородным и иерархическим ресурсом, превращаясь в активную, гетерогенную и географически распределенную среду выполнения. Это создает беспрецедентные вызовы для архитекторов ОС, требуя пересмотра принципов управления ресурсами на фундаментальном уровне. Данная статья посвящена анализу этих вызовов и формированию видения архитектуры ОС, способной эффективно управлять памятью-ориентированными вычислительными платформами [1][2].

Эволюция аппаратной парадигмы и новые вызовы для ОС

Краеугольным камнем современных вычислительных систем остается архитектура фон Неймана с ее разделением процессора и памяти. Для смягчения последствий «стены памяти» десятилетиями использовалась сложная иерархия кэшей, что, однако, лишь отсрочивало кризис. Современные инновации, такие как Intel Optane Persistent Memory, ломают границу между памятью и хранилищем, требуя от ОС гарантий согласованности данных при прямом байтовом доступе [4]. Стандарт CXL, в свою очередь, позволяет создавать разделяемые пулы памяти за пределами сокета процессора, превращая память в сетевой ресурс [2]. Наиболее радикальный прорыв — технологии вычислений в памяти, где в модуль DRAM или на контроллер NAND встраивается вычислительная логика, способная выполнять определенные операции без передачи данных в CPU [3]. Эта гетерогенность, представленная в Таблице 1, делает традиционную модель однородного адресного пространства устаревшей.

Таблица 1.**Сравнительная характеристика современных типов памяти и их влияние на подсистемы ОС**

Тип памяти	Ключевые свойства	Основной вызов для ОС	Пример реализации
Традиционная DRAM	Высокая скорость, энергозависимость	Оптимизация задержки, NUMA-балансировка	DDR4/DDR5
Persistent Memory (PMEM)	Энергонезависимость, байтовая адресуемость	Обеспечение согласованности данных, новые модели программирования	Intel Optane PMEM
CXL-память (тип 3)	Удаленное подключение, разделение между узлами	Обнаружение, аттестация, управление когерентностью	CXL 2.0/3.0, пулы памяти
Память с PIM	Наличие встроенной вычислительной логики	Распознавание вычислительных функций, планирование задач на память	Samsung HBM-PIM, UPMEM

Операционная система теперь не просто выделяет страницы, а выбирает оптимальный тип памяти для каждой задачи на основе его задержки, пропускной способности, энергонезависимости и вычислительного потенциала.

Трансформация ключевых подсистем операционной системы

Новая аппаратная реальность требует глубокой перестройки ядра ОС. Во-первых, подсистема управления памятью перестает быть монолитной. Ей необходима карта топологии, описывающая не только задержки доступа (как в NUMA), но и функциональные атрибуты каждого региона памяти: является ли он энергонезависимым, поддерживает ли атомарные операции, доступен ли для

прямого выполнения кода. Это требует расширения структур данных ядра, таких как `struct zone`, и создания нового механизма политик размещения данных, учитывающего семантику приложения [5][6]. Во-вторых, планировщик задач должен эволюционировать в планировщик вычислительных ресурсов. Если задача содержит операции, которые могут быть эффективно исполнены на вычислительном модуле внутри памяти PIM, планировщик должен учитывать эту возможность. Возникает концепция гетерогенного планирования, где выбор места выполнения (CPU, PIM, GPU) определяется не только загрузкой ядер, но и локальностью данных и энергоэффективностью операции. Это ставит сложные вопросы балансировки нагрузки и синхронизации между разнородными вычислительными устройствами. В-третьих, файловые системы для энергонезависимой памяти, такие как NOVA или BPFS, демонстрируют отход от блочной модели. Они работают непосредственно со структурами данных в PMEM, используя механизм прямого доступа (DAX), что исключает копирование в страничный кэш. Это ведет к слиянию подсистем управления памятью и хранением данных, требуя новых механизмов обеспечения отказоустойчивости и безопасности на уровне байтовых адресов, а не блоков [7].

Перспективные модели программирования и абстракции ядра

Для раскрытия потенциала памяти-ориентированных вычислений недостаточно лишь адаптировать ядро; необходимы новые программные абстракции. Один из подходов - расширение модели виртуальной памяти. Регионам памяти с особыми свойствами могут назначаться новые флаги защиты или типы, что позволит ядру и приложениям явно управлять размещением данных. Например, анонсирование памяти с флагом `MAP_PMEM` могло бы гарантировать ее выделение из пула энергонезависимой памяти [8]. Другой путь - создание специализированных драйверов и подсистем, предоставляющих доступ к вычислительным функциям PIM как к особым устройствам (`/dev/pim`) [1]. Однако это возвращает нас к модели ввода-вывода, которую PIM призван преодолеть. Более элегантным решением представляется интеграция компиляторов и сред выполнения, которые могли бы автоматически идентифицировать паттерны

доступа к данным (например, операции над плотными массивами) и транслировать их в инструкции для PIM-устройств, с которыми ОС взаимодействует через единый интерфейс планирования. График на Рисунке 1 схематично иллюстрирует эту эволюцию архитектуры взаимодействия.

Рисунок 1. Эволюция архитектуры взаимодействия процессора и памяти

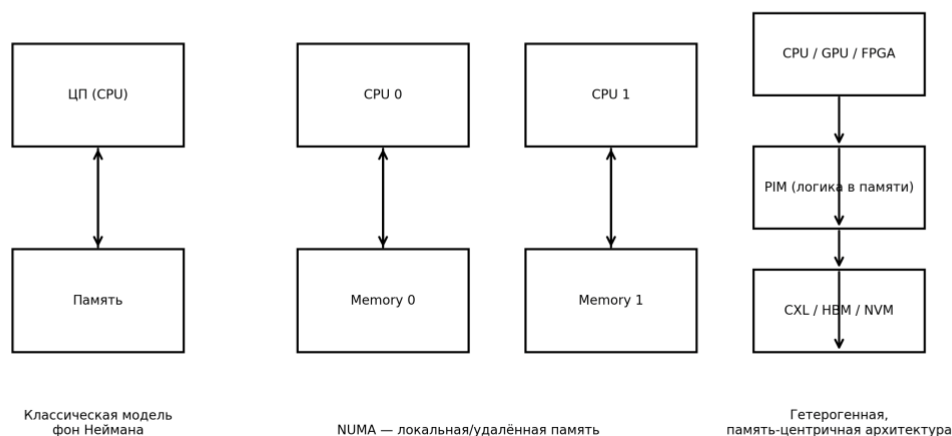


Рисунок 1. Эволюция архитектуры взаимодействия процессора и памяти: от фон Неймана к гетерогенной памяти-ориентированной модели.

Подобные изменения требуют тесной кооперации между разработчиками ОС, компиляторов и аппаратного обеспечения, формируя новую исследовательскую повестку [8].

Заключение

Анализ современных тенденций развития аппаратного обеспечения подтверждает неизбежность перехода от процессо-центричной к памяти-ориентированной вычислительной парадигме. Новые классы памяти, сочетающие свойства хранения, оперативного доступа и вычислений, делают традиционную архитектуру операционных систем, построенную вокруг центрального процессора и однородной иерархии памяти, принципиально неадекватной. Управление гетерогенной памятью становится центральной и наиболее сложной задачей ОС, что требует трансформации ее ключевых подсистем - от менеджера физической памяти и планировщика до файловых систем. Успешная адаптация операционных

систем к этой новой реальности лежит на пути разработки новых абстракций ядра, способных единообразно представлять разнородные вычислительные ресурсы, распределенные в пространстве памяти. Это включает создание сложных политик размещения данных, основанных на их семантике и требуемых гарантиях, а также механизмов планирования, которые рассматривают вычислительные модули в памяти как полноправные участники выполнения задач. Реализация этих идей откроет путь к созданию высокопроизводительных, энергоэффективных и масштабируемых систем, преодолевающих ограничения уходящей эпохи фон Неймана.

Список использованных источников

1. Boroumand A., Ghose S., Kim Y. и др. A Modern Primer on Processing in Memory [Электронный ресурс]. – 2023. – Режим доступа: <https://arxiv.org/abs/2306.12512> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

2. Compute Express Link Consortium. Compute Express Link (CXL) Specification 3.0 [Электронный ресурс]. – 2022. – Режим доступа: <https://www.computeexpresslink.org/download-the-specification> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

3. Mutlu O., Ghose S., Gómez-Luna J. и др. Processing-in-Memory: A Workload-Driven Perspective [Электронный ресурс] // IBM Journal of Research and Development. – 2021. – Режим доступа: <https://arxiv.org/abs/2103.06324> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

4. Intel Corporation. Intel Optane Persistent Memory [Электронный ресурс]. – 2023. – Режим доступа: <https://www.intel.com/content/www/us/en/architecture-and-technology/optane-persistent-memory.html> (свободный). – Заглавие с экрана. – Дата обращения: 11.12.2025.

5. Gouk D., Lee S., Kwon M. и др. Direct Access, High-Performance Memory Disaggregation with

DirectCXL [Электронный ресурс] // Proceedings of the USENIX Annual Technical Conference (ATC). – 2022. – Режим доступа: <https://www.usenix.org/conference/atc22/presentation/gouk> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

6. The Linux Kernel Documentation. Heterogeneous Memory Management (HMM) [Электронный ресурс]. – 2024. – Режим доступа: <https://docs.kernel.org/mm/hmm.html> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

7. Xu J., Swanson S. NOVA: A Log-structured File System for Hybrid Volatile/Non-volatile Main Memories [Электронный ресурс] // Proceedings of the USENIX Conference on File and Storage Technologies (FAST). – 2016. – Режим доступа: <https://www.usenix.org/conference/fast16/technical-sessions/presentation/xu> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

8. Young V., Nair P. J., Qureshi M. K. DeACT: Architecture-Aware Virtual Memory Support for Fabric Attached Memory Systems [Электронный ресурс] // Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA). – 2023. – Режим доступа: <https://ieeexplore.ieee.org/document/10070956> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.