

Шурупов Илья Викторович

*магистр, информатика и вычислительная техника, Национальный
исследовательский университет «МИЭТ», г. Зеленоград*

РЕАЛИЗАЦИЯ ТРАНСПОРТНОГО МЕХАНИЗМА ОБМЕНА СООБЩЕНИЯМИ ДЛЯ РАСПРЕДЕЛЁННЫХ СЕРВИСОВ В ГЕТЕРОГЕННЫХ СИСТЕМАХ НА КРИСТАЛЛЕ

Аннотация. В статье рассматриваются вопросы проектирования и практической реализации транспортного механизма обмена сообщениями (middleware) для распределённых сервисов, исполняемых на гетерогенных системах на кристалле (СнК). В качестве целевой вычислительной платформы использована отечественная гетерогенная СнК «СКИФ», содержащая процессорные ядра различной архитектуры — управляющие ядра ARM и специализированные сигнальные ядра DSP Elcore50. В основу программного стека положена открытая реализация стандарта Data Distribution Service (DDS) от компании eProsima, объединяющая полнофункциональный агент Fast DDS и облегчённый клиент Micro XRCE-DDS для сред с ограниченными ресурсами. Предложен и реализован метод прямой одноранговой (P2P) маршрутизации сообщений между клиентами, устраняющий обязательное посредничество агента и сокращающий время доставки более чем в два раза. Разработана платформонезависимая библиотека межъядерного обмена mcom03IPC на основе разделяемой памяти и кольцевых буферов с механизмами явной синхронизации кэша. Приводятся результаты экспериментальной верификации на платформах x86-64 и СнК «СКИФ», подтверждающие эффективность предложенных решений.

Ключевые слова: гетерогенная СнК, middleware, DDS, Micro XRCE-DDS, межъядерное взаимодействие, разделяемая память, прямая P2P-маршрутизация, импортозамещение.

Annotation. The paper discusses the design and practical implementation of a message-passing transport mechanism (middleware) for distributed services running on heterogeneous systems-on-chip (SoC). The domestic heterogeneous SoC “SKIF”, comprising processor cores of different architectures — ARM control cores and specialised Elcore50 DSP signal cores — is used as the target computing platform. The software stack is based on the open-source Data Distribution Service (DDS) implementation by eProsima, combining the full-featured Fast DDS agent with the lightweight Micro XRCE-DDS client for resource-constrained environments. A method of direct peer-to-peer (P2P) message routing between clients is proposed and implemented, eliminating mandatory agent relaying and reducing delivery time by more than a factor of two. A platform-independent inter-core communication library, mcom03IPC, is developed on top of shared memory and ring buffers with explicit cache synchronisation. Experimental validation results on the x86-64 and SKIF SoC platforms are provided, confirming the effectiveness of the proposed solutions.

Keywords: heterogeneous SoC, middleware, DDS, Micro XRCE-DDS, inter-core communication, shared memory, direct P2P routing, import substitution.

Современные вычислительные системы всё чаще строятся как гетерогенные, объединяя в рамках одной системы на кристалле процессорные ядра различных типов и архитектур: универсальные ядра общего назначения, сигнальные процессоры DSP, программируемую логику FPGA и заказные аппаратные блоки. Такой подход обеспечивает высокую энергоэффективность, компактность конечного решения и позволяет закреплять вычислительно ёмкие задачи за специализированными ядрами. Однако разнородность архитектур, адресных пространств и программных сред порождает ключевую инженерную проблему — организацию надёжного и детерминированного обмена данными между ядрами.

Распределённые сервисы, характерным примером которых является потоковая обработка радиосигналов в реальном времени, редко ограничиваются одним вычислительным ядром или одной платой. Вычислительная нагрузка распределяется как между ядрами внутри одной СнК, так и между несколькими

СнК, объединёнными в единый комплекс. В этих условиях приложению требуется унифицированный транспортный уровень, скрывающий низкоуровневые детали конкретного канала передачи и предоставляющий высокоуровневые примитивы взаимодействия по модели «издатель — подписчик» (publish-subscribe). Отсутствие такой абстракции приводит к жёсткой привязке кода к топологии системы, усложняет масштабирование и повышает вероятность ошибок синхронизации.

Целью настоящей работы является реализация транспортного механизма обмена сообщениями для распределённых сервисов на гетерогенных СнК в виде программного модуля с платформонезависимым интерфейсом. Основным научно-инженерным вызовом выступает адаптация распределённого middleware, изначально ориентированного на сетевое взаимодействие, к архитектурным особенностям отечественной гетерогенной платформы, а также устранение накладных расходов, критичных для сервисов жёсткого реального времени.

Целевая платформа и постановка задачи

В качестве целевой вычислительной платформы использована отечественная гетерогенная система на кристалле «СКИФ». Она включает управляющие ядра архитектуры ARM, работающие под управлением полноценной операционной системы Linux, и специализированные ядра цифровой обработки сигналов DSP Elcore50, функционирующие в условиях ограниченных ресурсов памяти и без стандартных системных сервисов. Ключевой особенностью платформы является асимметричная модель мультипроцессорности (AMP): различные ядра работают под управлением разных программных сред, обладают отличающимися ресурсными ограничениями и не поддерживают автоматический обмен данными на уровне операционной системы. Структурная организация СнК приведена на рисунке 1.

реализации модели публикации-подписки на уровне приложения. Второе направление предполагает применение высокоуровневых распределённых решений, таких как стандарт Data Distribution Service (DDS), изначально спроектированных для гетерогенных и распределённых систем: DDS реализует модель «издатель — подписчик» поверх протокола RTPS, поддерживает автоматическое обнаружение узлов и развитые политики QoS.

Из открытых реализаций DDS (OpenDDS, Eclipse Cyclone DDS, Fast DDS) для целевой платформы выбрана реализация компании eProsima. Определяющим фактором стала технология DDS-XRCE (eXtremely Resource-Constrained Environment): облегчённый клиент Micro XRCE-DDS выполняется на слабом ядре, включая DSP, а основная логика DDS сосредоточена на более мощном узле в составе агента Fast DDS. Такая гибридная клиент-серверная модель устраняет необходимость портирования полного DDS-стека на ресурсно-ограниченные ядра. Далее ядро, исполняющее код Micro XRCE-DDS, именуется клиентом, а ядро с полной реализацией Fast DDS — агентом.

Интеграция выбранного решения выявила два ограничения. Во-первых, программные интерфейсы клиента и агента различаются, что снижает переносимость компонентов; функционально при этом Micro XRCE-DDS является подмножеством Fast DDS, поэтому проблема решается введением унифицирующего интерфейса-обёртки («wrapper») над обоими протоколами. Во-вторых, в силу клиент-серверной природы XRCE весь трафик клиента проходит через агента: при передаче сообщения от одного клиента другому оно дважды пересекает уровень межъядерного взаимодействия — при отправке агенту и при перенаправлении получателю. Это снижает производительность как минимум вдвое без учёта дополнительной нагрузки на кэш и вычислительные операции, что неприемлемо для сервисов реального времени и требует оптимизации.

Библиотека межъядерного обмена mcom03IPC

Фундаментом транспортного уровня является разработанная библиотека межъядерного обмена mcom03IPC, оформленная в виде самостоятельного компонента и предоставляющая минимальный интерфейс двунаправленного канала между двумя ядрами. Передача данных организована через разделяемую память с использованием кольцевых буферов: для каждого направления выделяется отдельный буфер, что обеспечивает полнодуплексный обмен (рисунок 2). Каждый буфер функционирует по модели «один производитель — один потребитель» (single-producer, single-consumer), что позволяет отказаться от блокировок и упростить логику синхронизации.

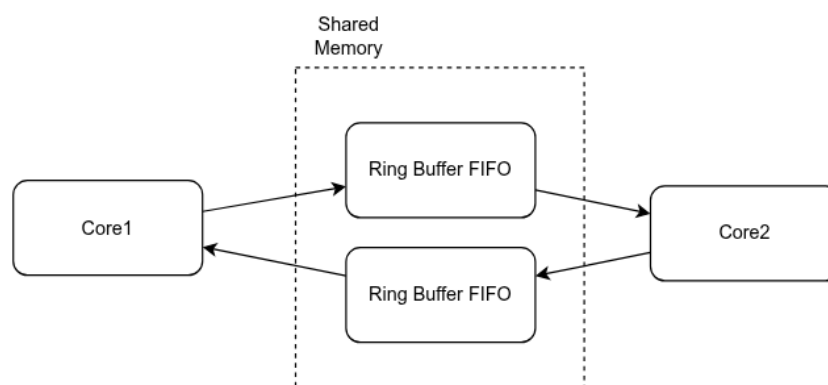


Рисунок 2 — Канал обмена между ядрами с двумя однонаправленными кольцевыми буферами

Архитектура библиотеки разделена на платформонезависимую и платформозависимую части. Платформонезависимое ядро реализует логику кольцевых буферов и протокол обмена; платформозависимая часть отвечает за низкоуровневое взаимодействие с аппаратурой — работу с разделяемой памятью, механизмами DMA и синхронизацией кэша. Для поддержки нового типа ядра достаточно реализовать фиксированный набор функций интерфейса, включая инициализацию общей памяти, уведомление противоположной стороны и синхронизацию кэша, что обеспечивает переносимость решения. Физически под общую память ядер ARM и DSP отведена область DDR, а для прямого канала между ядрами DSP используется внутренняя память MPSM.

Существенную сложность представляет обеспечение когерентности кэша в системах без аппаратной поддержки согласования. Поскольку DMA-контроллер обращается к оперативной памяти напрямую, минуя кэш процессора, изменения, выполненные одной стороной, могут оставаться невидимыми другой. Для устранения рассогласования применяется явная программная синхронизация: при передаче от процессора выполняется сброс (write-back) модифицированных строк кэша до начала транзакции, а перед чтением — инвалидация устаревших копий; корректный порядок операций гарантируется барьерами памяти. Часть служебных полей — указатели буферов и флаги уведомления — размещается в некэшируемой области, что исключает необходимость их синхронизации. Дополнительно рассмотрен приём двойного отображения одной физической области в адресное пространство посредством mmap: он «склеивает» конец и начало кольцевого буфера, позволяя выполнять непрерывную линейную запись без проверки границ при каждой операции.

Оптимизация одноранговой маршрутизации в стеке DDS

Суть оптимизации состоит в том, чтобы сообщения, конечными получателями которых являются другие клиенты, направлялись им напрямую, минуя «крюк» через агента (рисунок 3). Для реализации прямого канала клиент абстрагируется набором логических объектов — портов, каждый из которых соответствует объекту чтения или записи данных (DataReader / DataWriter), связанному с определённой темой обмена (Topic). Задача сводится к отслеживанию этих объектов и построению таблиц зависимостей между издателями и подписчиками как на стороне агента, так и на стороне клиента.

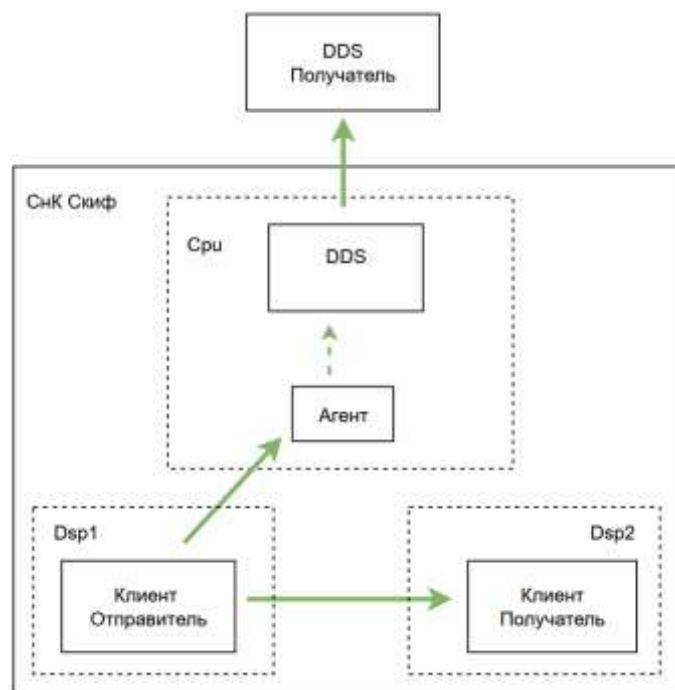


Рисунок 3 — Путь сообщения после оптимизации: прямой канал между клиентами

Агент собирает сведения о подключениях при обработке сообщений типа CREATE: через интерфейс полной реализации Fast DDS он извлекает описание темы (имя и тип данных) для создаваемых объектов `DataReader` и `DataWriter` и заносит их во внутренние структуры данных. Обнаружив зависимость между подписчиком на одном клиенте и издателем на другом, агент направляет отправителю новое служебное сообщение типа CLIENT-P2P, сообщающее о наличии прямого получателя. Для поддержки этого механизма на стороне клиента реализовано хранение нескольких транспортов вместо единственного канала к агенту: транспорты организованы в виде односвязного списка, а выбор транспорта выполняется динамически на этапе отправки по таблице соответствия «порт — транспорт».

Отдельного рассмотрения потребовал случай, когда сообщение должно поступить одновременно и агенту, и другому клиенту. Простое отключение оптимизации для таких сообщений на практике обесценивало бы её, поэтому для каждого получателя выбирается наиболее эффективный маршрут, а сообщение при необходимости отправляется в несколько транспортов. При этом агент проверяет, требуется ли дальнейшая пересылка сообщения клиентам или его

следует передать только в полную реализацию Fast DDS, что исключает повторное получение (дублирование) сообщения. Проверка необходима даже в полностью оптимизированном режиме, поскольку часть служебных сообщений и потоков данных с особыми параметрами QoS не использует прямой канал.

Экспериментальные результаты

Экспериментальное исследование проводилось по единой методике: для сообщений различного объёма измерялось полное время доставки по одному топику в сценариях «клиент — агент» и «клиент — клиент». На платформе x86-64 в качестве транспорта был реализован плагин на основе очередей сообщений POSIX (mqqueue). Сравнение исходной и оптимизированной реализаций (рисунок 4) подтвердило ожидаемый результат: время доставки сообщения между клиентами сокращается более чем в два раза. Кривые для обеих реализаций имеют сходный характер и различаются примерно на постоянный коэффициент, что классифицирует оптимизацию как улучшение константного множителя, не изменяющее асимптотическую сложность. Характерный перелом графика в области около 8 КБайт соответствует активации механизма фрагментации сообщений.

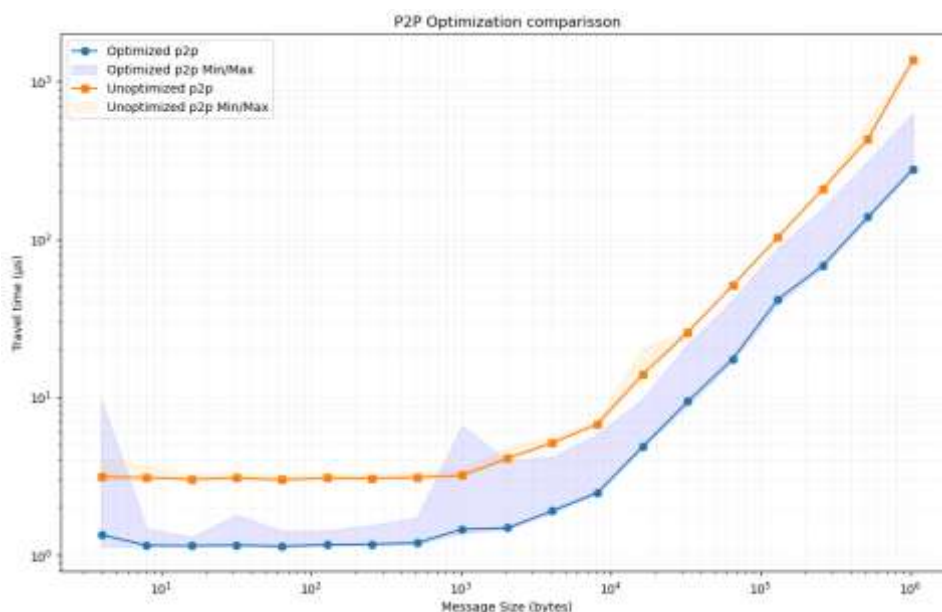


Рисунок 4 — Время доставки сообщения между клиентами: исходная и оптимизированная реализации (x86-64)

После стабилизации решения на x86-64 программный стек был перенесён на СнК «СКИФ». Благодаря тому, что Fast DDS реализован на C++11, а компоненты XRCE — на C, портирование не потребовало значительных усилий: поддержка ARM обеспечивается «из коробки», а для архитектуры Elcore50 потребовалась лишь адаптация функций измерения времени. Для целевой платформы разработаны два транспортных плагина — для взаимодействия ARM ↔ DSP (через область DDR) и DSP ↔ DSP (через память MPSM), — использующие библиотеку mcom03IPC. Результаты на «СКИФ» согласуются с полученными на x86-64: отношение времени доставки неоптимизированной и оптимизированной версий превышает два. При этом на графиках наблюдаются характерные пороговые области, обусловленные ограничениями кэш-памяти отправителя и получателя, а достигнутая пропускная способность межъядерного канала при передаче крупных блоков приближается к 4,7 ГБайт/с.

Единый интерфейс взаимодействия для приложения обеспечивается разработанной обёрткой над протоколами XRCE и DDS, инкапсулирующей создание темы обмена, объектов чтения и записи, а также два режима получения данных — опрос (polling) и обратный вызов (callback). Тем самым прикладной код полностью абстрагирован от того, на каком именно ядре и посредством какого транспорта выполняется обмен.

Заключение и перспективы развития

В работе решена научно-практическая задача создания транспортного механизма обмена сообщениями для распределённых сервисов на отечественной гетерогенной СнК «СКИФ». Разработанный модуль предоставляет платформонезависимый интерфейс на базе стандарта DDS, а предложенная схема прямой P2P-маршрутизации в связке Fast DDS / Micro XRCE-DDS обеспечивает устойчивое сокращение задержки межклиентского обмена более чем в два раза. Основой межъядерного взаимодействия служит библиотека mcom03IPC, реализующая эффективный полнодуплексный обмен через разделяемую память с корректной обработкой когерентности кэша.

На следующих этапах развития проекта планируется расширение набора поддерживаемых политик QoS для прямого канала, интеграция дополнительных типов вычислительных ядер и отечественных СнК, а также исследование механизмов фрагментации крупных сообщений с учётом ограничений кэш-подсистемы целевых платформ. Достигнутые показатели производительности свидетельствуют о применимости предложенного решения в задачах, предъявляющих жёсткие требования к задержкам и пропускной способности межпроцессорного взаимодействия.

Литература

1. Object Management Group. Data Distribution Service (DDS) : Version 1.4 [Electronic resource]. — OMG, 2015. — 180 p. — URL: omg.org/spec/DDS (дата обращения: 08.08.2026).
2. Object Management Group. DDS for Extremely Resource Constrained Environments (DDS-XRCE) : Version 1.0 [Electronic resource]. — OMG, 2020. — 116 p. — URL: omg.org/spec/DDS-XRCE (дата обращения: 08.08.2026).
3. eProsima Fast DDS Documentation [Electronic resource] / eProsima. — 2024. — URL: fast-dds.docs.eprosima.com (дата обращения: 05.08.2026).
4. eProsima Micro XRCE-DDS Documentation [Electronic resource] / eProsima. — 2024. — URL: micro-xrce-dds.docs.eprosima.com (дата обращения: 05.08.2026).
5. OpenAMP Project. Remote Processor Messaging (RPMsg) Framework [Electronic resource]. — 2023. — URL: openampproject.org (дата обращения: 01.08.2026).
6. Микросхема интегральная 1892BA018 («СКИФ») : руководство пользователя [Электронный ресурс] / АО НПЦ «ЭЛВИС». — Зеленоград, 2023. — URL: elvees.ru (дата обращения: 07.08.2026).

7. Таненбаум, Э. Современные операционные системы / Э. Таненбаум, Х. Бос. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — ISBN 978-5-496-01395-6.
8. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. — 3-е изд. — М. : Вильямс, 2013. — 1328 с. — ISBN 978-5-8459-1794-2.
9. Страуструп, Б. Язык программирования C++ / Б. Страуструп. — 4-е изд. — М. : Бином, 2015. — 1136 с. — ISBN 978-5-9518-0596-3.

References

1. Object Management Group. Data Distribution Service (DDS): Version 1.4 [Electronic resource]. — OMG, 2015. — 180 p. — URL: omg.org/spec/DDS (accessed: 08.08.2026).
2. Object Management Group. DDS for Extremely Resource Constrained Environments (DDS-XRCE): Version 1.0 [Electronic resource]. — OMG, 2020. — 116 p. — URL: omg.org/spec/DDS-XRCE (accessed: 08.08.2026).
3. eProsima Fast DDS Documentation [Electronic resource] / eProsima. — 2024. — URL: fast-dds.docs.eprosima.com (accessed: 05.08.2026).
4. eProsima Micro XRCE-DDS Documentation [Electronic resource] / eProsima. — 2024. — URL: micro-xrce-dds.docs.eprosima.com (accessed: 05.08.2026).
5. OpenAMP Project. Remote Processor Messaging (RPMsg) Framework [Electronic resource]. — 2023. — URL: openampproject.org (accessed: 01.08.2026).
6. 1892VA018 (“SKIF”) Integrated Circuit: User Manual [Electronic resource] / ELVEES RnD Center. — Zelenograd, 2023. — URL: elvees.ru (accessed: 07.08.2026).
7. Tanenbaum, A. S. Modern Operating Systems / A. S. Tanenbaum, H. Bos. — 4th ed. — Pearson, 2014. — 1136 p. — ISBN 978-0-13-359162-0.

8. Cormen, T. H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. — 3rd ed. — MIT Press, 2009. — 1312 p. — ISBN 978-0-262-03384-8.
9. Stroustrup, B. The C++ Programming Language / B. Stroustrup. — 4th ed. — Addison-Wesley, 2013. — 1368 p. — ISBN 978-0-321-56384-2.