

Подходы к повышению эффективности Mesh-сети в вычислительной системе с распределенной памятью

Д. И. Хайдуков, А. А. Алексеев

Аннотация—В работе рассматриваются архитектурные подходы к повышению эффективности Mesh-сети в вычислительной системе с распределенной памятью. Основное внимание уделено решениям, применимым на ранних этапах проектирования и позволяющим систематически анализировать компромиссы между производительностью сети и затратами аппаратных ресурсов. Рассматриваются подходы, связанные с выбором алгоритма маршрутизации, изменением формы сети, масштабированием числа физических и виртуальных каналов, а также полным дублированием сетевой инфраструктуры. Эффективность сети оценивается с использованием совокупности метрик, включающей пропускную способность, среднюю задержку обработки запросов и производительность на ресурс. Оценка задержек выполняется на основе аналитической модели, ранее разработанной авторами и ориентированной на вычислительные системы с распределенной памятью, тогда как пропускная способность оценивается с применением потактового моделирования. Все архитектурные решения сопоставляются в рамках единого сценария, что позволяет последовательно оценить их влияние на распределение трафика, точки насыщения и структуру задержек. Полученные результаты показывают, что наибольшую эффективность демонстрируют решения, улучшающие распределение потоков при умеренном росте аппаратных ресурсов, тогда как простое наращивание емкости сети без структурных изменений приводит к убывающей отдаче.

Ключевые слова—аналитическое моделирование, вычислительная система, потактовое моделирование, распределенная память, сеть на кристалле.

I. ВВЕДЕНИЕ

Современные задачи компьютерного зрения, цифровой обработки сигналов и искусственного интеллекта предъявляют высокие требования к производительности и масштабируемости вычислительных систем, что обуславливает необходимость параллельной обработки больших объемов данных и развитие специализированных гетерогенных архитектур [1], объединяющих исполнительные устройства и аппаратные ускорители, ориентированные на конкретные классы нагрузок.

Ключевую роль в таких системах играет

распределенная память [2], определяющая эффективность обмена данными и степень использования вычислительных ресурсов. В условиях, когда доля площади кристалла, занимаемая памятью, может превышать 90% [3], особую значимость приобретает задача построения масштабируемых механизмов связи между вычислительными блоками и модулями памяти [4].

Обмен данными в вычислительных системах с распределенной памятью реализуется с использованием сетей на кристалле, обеспечивающих передачу данных и управляющих сообщений между исполнительными устройствами и памятью. Среди различных топологий наибольшее распространение получила двумерная Mesh-топология [5], отличающаяся регулярной структурой и хорошей масштабируемостью. Эффективность Mesh-сети в значительной степени определяется архитектурными решениями, алгоритмами маршрутизации и характеристиками маршрутизаторов, что делает выбор сетевой архитектуры одним из ключевых этапов проектирования таких систем.

В условиях высокой плотности компонентов и жестких ограничений по площади и энергопотреблению проектирование сетей на кристалле требует ранней оценки архитектурных решений [6], позволяющей на концептуальном уровне прогнозировать поведение системы и сравнивать альтернативные варианты до начала детальной реализации. В качестве основных метрик производительности при таком анализе используются пропускная способность и средняя задержка обработки запросов [7], тогда как энергоэффективность на ранних этапах оценивается косвенно через объем требуемых аппаратных ресурсов и тактовую частоту [8]. Это приводит к задаче многокритериальной оптимизации [9], в рамках которой необходимо найти компромисс между производительностью и стоимостью реализации [10, 11], для чего удобно использовать показатель производительности на ресурс [12], позволяющий наглядно сопоставлять эффективность различных архитектурных решений.

В данной работе рассматривается совокупность архитектурных подходов, направленных на повышение

эффективности Mesh-сети в вычислительной системе с распределенной памятью. Анализируются решения, связанные с выбором алгоритма маршрутизации, изменением топологии сети, масштабированием числа физических и виртуальных каналов, а также дублированием сетевой инфраструктуры. Оценка эффективности указанных подходов выполняется на основе метрик пропускной способности, средней задержки и производительности на ресурс с использованием методов раннего архитектурного анализа. Рассматриваемые решения и их сопоставление в рамках единого сценария применяются к вычислительной системе с распределенной памятью, что позволяет выявить их влияние на производительность и затраты аппаратных ресурсов в условиях, характерных для таких систем.

Дальнейшая структура статьи организована следующим образом. В разделе II представлен обзор литературы, включающий анализ архитектурных характеристик сетей на кристалле, обзор существующих подходов к повышению эффективности сети и рассмотрение инструментов для ранней оценки. В разделе III приведено описание исследуемого сценария. В разделе IV описаны рассматриваемые подходы к повышению эффективности Mesh-сети, а в разделе V представлены численные результаты проведенных экспериментов. Наконец, в разделе VI сформулированы основные выводы данной работы.

II. ОБЗОР ЛИТЕРАТУРЫ

A. Архитектурные характеристики сети

Среди ключевых архитектурных характеристик, определяющих эффективность Mesh-сети, выделяются алгоритм маршрутизации, архитектура маршрутизатора и политика управления виртуальными каналами.

1) Алгоритмы маршрутизации

В практических реализациях сетей на кристалле наибольшее распространение получили детерминированные алгоритмы маршрутизации [13], среди которых широко используется алгоритм DOR (Dimension Order Routing), выделяющийся простотой реализации и низкими накладными расходами.

Вместе с тем алгоритму DOR присуща выраженная несимметричность маршрутов (рис. 1А), приводящая к снижению пропускной способности вертикальных агентов при высоких интенсивностях трафика по сравнению с горизонтальными. Этот эффект особенно критичен для вычислительных систем с распределенной памятью, в которых значительная часть трафика сосредоточена в центральной области сети [14]. Для устранения данного недостатка был предложен более симметричный адаптивный алгоритм ADOR (рис. 1Б), обеспечивающий перераспределение нагрузки вертикальных агентов и более равномерное использование сетевых ресурсов.

2) Архитектура маршрутизатора

Архитектура маршрутизатора в современной сети на

кристалле должна обеспечивать корректность протокола, достаточную пропускную способность при характерных нагрузках и предсказуемые задержки при разумной стоимости аппаратных ресурсов [15].

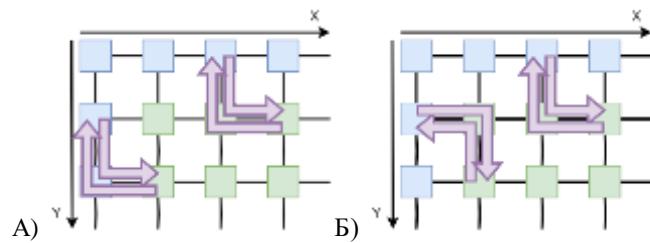


Рис. 1. Алгоритмы маршрутизации:
(А) DOR и (Б) ADOR

Одним из ключевых требований является отсутствие взаимоблокировок, что на практике часто достигается разделением коммуникаций на независимые физические каналы [16] для разных типов сообщений, например запросов и ответов, с такими правилами обмена, при которых исключаются циклические зависимости.

В промышленных реализациях доминируют буферные маршрутизаторы, способные поглощать кратковременные всплески трафика и эффективно разрешать локальную конкуренцию. Альтернативные решения без буферизации [17] или с минимальной буферизацией [18] исследуются в литературе как более энергоэффективные [19], однако их применение ограничено малыми масштабами и жесткими энергетическими ограничениями из-за отражений и обходов по неоптимальным маршрутам при занятости целевых направлений.

Реализация конвейера маршрутизатора также существенно влияет на производительность: однотактные решения обеспечивают минимальную задержку [20], но при росте масштабов системы и тактовых частот становятся трудно реализуемыми, что приводит к использованию многостадийных конвейеров, в том числе с вынесением отдельных стадий за пределы маршрутизатора [21] и с применением буферизации как на входах, так и на выходах [22, 23] для предотвращения узких мест при передаче между узлами.

3) Управление виртуальными каналами

Виртуальные каналы (ВК) являются ключевым механизмом повышения пропускной способности и снижения блокировок, поскольку позволяют логически разделять конкурирующие потоки и повышать эффективность использования буферов [24].

Существенное влияние на характеристики сети оказывает политика распределения запросов по ВК. Динамическая аллокация [25] обеспечивает более гибкое использование буферного пространства и демонстрирует преимущества при переменной интенсивности трафика, тогда как статическая аллокация [26], закрепляющая ВК за типами сообщений, проще в реализации и естественным образом сочетается с доказуемой свободой от взаимоблокировок, особенно при детерминированной маршрутизации.

Дополнительным фактором является внутренняя микроархитектура коммутации маршрутизатора:

параллельная обработка нескольких ВК [27] позволяет при сопоставимых аппаратных ресурсах достигать более высокой пропускной способности по сравнению с одиночной последовательной обработкой большего числа ВК [28].

B. Подходы к повышению эффективности сети

В рамках вычислительной системы с распределенной памятью перспективно рассматривать несколько направлений увеличения эффективности Mesh-сети. Первое связано с увеличением числа ВК. Рост числа ВК приводит к разделению логических потоков на одном физическом канале и тем самым повышает пропускную способность всей сети. Несмотря на то, что при этом существенно увеличиваются затрачиваемые ресурсы, существуют работы, которые отмечают прирост показателя эффективности на ресурс [29].

Другое направление — оптимизация Mesh-сети за счет дополнительных физических связей между маршрутизаторами. В отличие от предыдущего направления пропускная способность повышается за счет более равномерного распределения физических потоков, а не логических. Как следствие, увеличение числа физических связей может быть более эффективным, чем увеличение числа ВК [30]. Хотя такой подход не просто увеличивает число ресурсов, а дополнительно усложняет физическую реализацию сети, существуют исследования [31], отмечающие его эффективность с точки зрения производительности на ресурс.

Еще одно направление связано с выбором геометрической формы Mesh-сети. При фиксированном числе узлов наиболее близкая к квадратной топология обладает большей полосой бисекции и меньшим диаметром, что снижает среднюю задержку. Этот вывод фигурирует как в практических работах по масштабированию многоядерных процессоров при проектировании новых поколений [32], так и в исследованиях, посвященных разработке аналитических моделей [33]. Однако исследователи отмечают, что использование сети квадратной формы не всегда физически реализуемо из-за особенностей производства вычислительных систем.

Наконец, возможен подход с дублированием Mesh-сети — использованием двух независимых межсоединений для объединения вычислительных устройств. Разделение трафика по независимым сетям позволяет оптимизировать производительность без чрезмерного усложнения одного маршрутизатора и дает гибкость в политике маршрутизации и обслуживания потоков. Существуют исследования [34], отмечающие лучшую эффективность такого подхода относительно использования альтернативных топологий по метрикам занимаемой площади и энергопотребления на полученную производительность. Другие работы [35] отмечают перспективность такого подхода с точки зрения гибкого управления энергопотреблением. Исследования по сопоставлению данного подхода с увеличением числа ВК [36] приходят к выводу, что дублирование сети эффективнее по производительности при примерно равных аппаратных ресурсах. Кроме того,

данний подход может использоваться для оптимизации отдельных приложений [37] в вычислительной системе, где одна сеть нацелена на максимальную пропускную способность, а дублирующая — на минимальную задержку.

В совокупности все эти направления образуют поле для честного сравнения конфигураций по показателю «производительность на ресурс».

C. Инструменты ранней оценки

В задачах ранней оценки эффективности сетей на кристалле применяются как симуляционные, так и аналитические методы моделирования [38]. Потактовые симуляторы [39] позволяют достоверно оценивать интегральные характеристики сети, прежде всего пропускную способность и поведение при насыщении, за счет детального воспроизведения взаимодействий между узлами и арбитража ресурсов. В то же время оценка средней задержки на основе потактового моделирования требует значительных вычислительных затрат и большого числа прогонов для получения статистически устойчивых результатов. В связи с этим на ранних этапах проектирования более эффективным подходом является использование аналитических моделей, позволяющих оценивать средние задержки с приемлемой точностью при существенно меньших затратах времени и с сохранением воспроизводимости выводов при стандартных допущениях о характере трафика и обслуживании.

В качестве потактовой модели в данной работе используется потактовый симулятор сетей на кристалле с открытым исходным кодом Noxim [40], ранее адаптированный авторами для оценки производительности Mesh-сети в вычислительной системе с распределенной памятью [14].

Для оценки средней задержки в Mesh-сети применяется аналитическая модель [41], разработанная авторами ранее и ориентированная на вычислительные системы с распределенной памятью. Модель учитывает ключевые архитектурные особенности маршрутизаторов сети, включая структуру портов с раздельными физическими каналами для данных и управляющих сообщений, а также использование ВК с FIFO-буферами для предотвращения взаимоблокировок и повышения емкости сети.

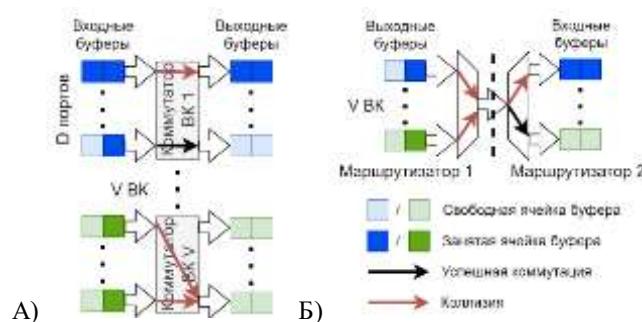


Рис. 2. Коллизии запросов (А) во входных и (Б) в выходных буферах

При моделировании входных буферов учитываются коллизии при маршрутизации и переполнение целевых выходных буферов, тогда как для выходных буферов

анализируются коллизии между ВК и блокировки, связанные с переполнением входных буферов соседних маршрутизаторов (рис. 2).

Средние времена обслуживания буферов позволяют представить маршрутизаторы в виде систем массового обслуживания типа $M/M/1/N$ [42] и вычислять средние задержки по аналитическим выражениям с расхождением не более 5% относительно потактовой модели.

III. ОПИСАНИЕ СЦЕНАРИЯ

В рамках данного исследования рассматривается Mesh-сеть размером $N \times M$ узлов, реализованная как часть архитектуры вычислительной системы с распределенной памятью (рис. 3). Агенты, формирующие запросы на чтение и запись, подключены к узлам по периметру сети, тогда как модули памяти размещены в центральной области (массив $(N - 2) \times (M - 2)$ внутренних узлов).

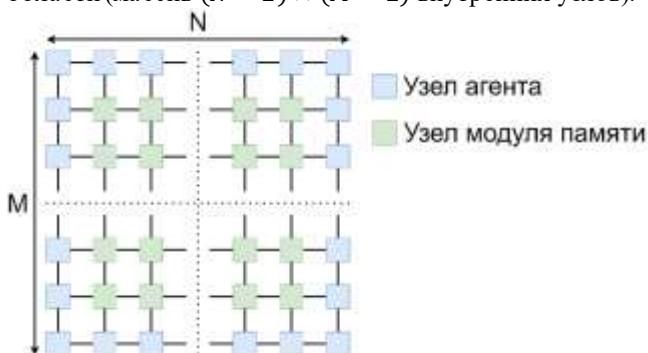


Рис. 3. Mesh-сеть $N \times M$ вычислительной системы с распределенной памятью

Каждый узел сети реализован в виде маршрутизатора с $D \geq 5$ портами (рис. 4), из которых не менее четырех используются для соединений между узлами, и не менее одного — для подключения агентов и модулей распределенной памяти.

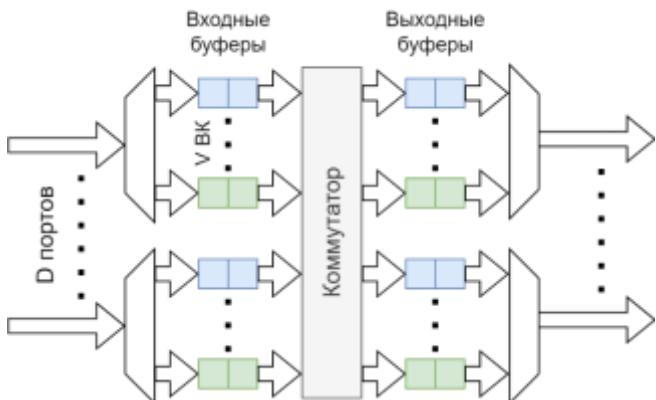


Рис. 4. Структура узла сети (для 1 набора физических каналов)

Каждый порт маршрутизатора реализует два двунаправленных физических канала — канал передачи данных и канал передачи управляющих сообщений (запросов на чтение и подтверждений записи). Протокол, используемый в физических каналах, допускает передачу посылки каждый тик по каждому из направлений.

Каждый физический канал разделяется на V ВК с параллельной независимой коммутацией, статически

закрепляемых за запросами на всем пути следования. Входы и выходы ВК оснащены FIFO-буферами глубины 2, что определяет минимальную задержку в маршрутизаторе, равную 2 тактам. Доступ ВК к общему физическому каналу и выходным портам разрешается по принципу Round Robin.

Передача в сети осуществляется посылками: запросы на чтение и запись инициируются агентами, а модули памяти возвращают либо подтверждение записи, либо пакет данных в ответ на запрос чтения.

При маршрутизации посылок используется детерминированный подход. В качестве механизма избегания взаимоблокировок используются ВК и статическое распределение трафика по ним. Для трафика запросов и ответов используются различные ВК. Для каждого ВК зафиксирован используемый порядок обхода направлений сети: YX или XY.

Генерация трафика в сети моделируется как равномерное случайное воздействие. Агенты с заданной интенсивностью формируют запросы на чтение и запись в произвольные узлы памяти. После обработки запроса модуль памяти отправляет обратно пакет с данными, либо подтверждение записи. Отношение между чтением и записью в потоке трафика фиксировано как 1:1. Интенсивность генерации запросов варьируется в зависимости от сценария эксперимента.

Для оценки эффективности сети используются следующие основные метрики:

- пропускная способность, т.е. фактическая интенсивность обработки запросов в зависимости от заданной интенсивности, оцениваемая с помощью потактовой модели;
- средняя задержка обработки запроса, измеряемая как суммарное время от момента формирования запроса до получения ответа (посылки данных или подтверждения), оцениваемая с помощью аналитической модели;
- производительность на ресурс, оцениваемая как отношение вышеперечисленных метрик производительности к числу затраченных ресурсов, оцениваемых качественно, таких как число триггеров, соединений между узлами и портов модулей распределенной памяти.

IV. ПОДХОДЫ К ПОВЫШЕНИЮ ЭФФЕКТИВНОСТИ СЕТИ НА КРИСТАЛЛЕ

Для анализа эффективности различных направлений оптимизации были сформированы 8 конфигураций сети на кристалле, различающихся по архитектуре сети, архитектуре маршрутизатора и, как следствие, по требуемым аппаратным ресурсам. Рассматриваемая вычислительная система объединяет восемь агентов, подключенных к вертикальным узлам сети, и шестнадцать агентов, подключенных к горизонтальным узлам. Сводная характеристика конфигураций приведена в таблице 1.

В качестве базовой конфигурации, относительно которой выполняется сравнение, используется Mesh-сеть размером 10×6 узлов, построенная по алгоритму DOR. Каждый маршрутизатор имеет $D = 5$ двунаправленных

физических портов (4 сетевых и 1 локальный) и $V = 2$ ВК.

Таблица 1. Рассматриваемые конфигурации

№	Сеть		Маршрутизатор		Оценка ресурсов			
	2 Mesh-сети	Размер	Порты модуля памяти, шт	Алгоритм	Число сетевых портов, шт	Триггеры	Соединения узлов	Порты памяти
1	Нет	10x6	1	DOR	4	1x	1x	1x
2	Нет	10x6	1	ADOR	4	2x	1x	1x
3	Нет	10x10	1	DOR	4	2x	2x	2x
4	Нет	10x10	1	ADOR	4	4x	2x	2x
5	Нет	10x6	2	DOR	6	1.5x	1.5x	2x
6	Нет	10x6	2	ADOR	6	3x	1.5x	2x
7	Да	10x6	2	DOR + IDOR	4	2x	2x	2x
8	Да	10x6	2	ADOR + ADOR	4	4x	2x	2x

A. Изменение алгоритма маршрутизации

Первое направление оптимизации связано с заменой алгоритма маршрутизации. В конфигурации №2 вместо алгоритма DOR используется ADOR, обеспечивающий симметричное распределение маршрутов относительно центра сети. Для предотвращения взаимоблокировок, возникающих вследствие пересечения потоков при изменении порядка обхода координатных осей, число ВК в каждом маршрутизаторе удваивается. Это приводит к двукратному росту числа триггеров при неизменной топологии сети, числе соединений между узлами и числе портов модулей памяти.

B. Изменение формы сети

Вторая группа конфигураций направлена на анализ влияния геометрической формы сети на ее характеристики. В базовой конфигурации используется прямоугольная топология 10×6 , тогда как в конфигурации №3 форма сети изменена на квадратную 10×10 , что приводит к удвоению числа маршрутизаторов и модулей памяти. Подключение горизонтальных агентов в конфигурации №3 сохраняет структуру базовой конфигурации, а вертикальные агенты подключаются к центральным узлам каждой стороны, что обеспечивает симметричное и компактное взаимодействие с областью распределенной памяти. Для сохранения общего объема внутренней памяти объем памяти, приходящийся на каждый модуль, уменьшается вдвое.

В результате для конфигурации №3 удваиваются все основные аппаратные ресурсы по сравнению с базовой конфигурацией, включая число триггеров, соединений между узлами и портов памяти. Конфигурация №4 отличается от конфигурации №3 использованием алгоритма маршрутизации ADOR вместо DOR. Для предотвращения взаимоблокировок в этом случае удваивается число ВК, что приводит к дополнительному росту числа триггеров при неизменных значениях остальных ресурсных параметров.

C. Увеличение числа физических каналов

Третье направление оптимизации связано с изменением

архитектуры маршрутизатора при сохранении исходной топологии сети. В конфигурации №5 увеличено число физических портов маршрутизатора, что позволяет более равномерно распределять трафик и имитировать эффекты, характерные для более квадратной формы сети, без изменения геометрии расположения узлов. Для этого число сетевых портов увеличено с четырех до шести, а число локальных портов подключения модулей памяти — с одного до двух. Вертикальные агенты в данной конфигурации рассматриваются как двухпортовые с равномерным распределением нагрузки, тогда как горизонтальные агенты сохраняют подключение по одному порту.

С точки зрения аппаратных затрат такая архитектура приводит к увеличению числа триггеров и соединений между узлами в 1.5 раза относительно базовой конфигурации, а число портов памяти удваивается.

Конфигурация №6 отличается от конфигурации №5 использованием алгоритма ADOR вместо DOR. Для предотвращения взаимоблокировок в этом случае удваивается число ВК, что приводит к дополнительному увеличению числа триггеров при неизменных значениях числа соединений и портов памяти.

D. Дублирование сети

Четвертое направление оптимизации направлено на исследование эффекта от полного дублирования сети на кристалле. Основная идея заключается в разделении потоков, формируемым горизонтальными и вертикальными агентами, по двум независимым сетям, что устраняет взаимное влияние трафика и конфликты маршрутов в центральной области.

В конфигурации №7 реализуются две независимые Mesh-сети, предназначенные соответственно для обслуживания горизонтальных и вертикальных агентов. Каждый модуль памяти подключается к обеим сетям через отдельные порты. Для сети горизонтальных агентов используется алгоритм маршрутизации DOR, тогда как для сети вертикальных агентов применяется инвертированный порядок обхода координатных осей (IDOR): XY для запросов и YX для ответов. Независимость сетей и взаимно инвертированные

маршруты исключают пересечения потоков, что обеспечивает свободу от взаимоблокировок и позволяет сохранить число ВК в каждой сети на уровне базовой конфигурации.

Дублирование сетевой структуры приводит к удвоению всех аппаратных ресурсов по сравнению с базовой конфигурацией, включая число триггеров, соединений между узлами и портов памяти, поскольку каждая из сетей содержит собственный набор маршрутизаторов и линий связи.

Конфигурация №8 сохраняет общую организацию конфигурации №7, однако в обеих сетях используется алгоритм маршрутизации ADOR. Каждый агент и каждый модуль памяти подключаются к двум независимым сетям с равномерным распределением нагрузки. В результате все аппаратные ресурсы увеличиваются вдвое относительно конфигурации №2, то есть варианта с алгоритмом ADOR без дублирования сети.

V. ЧИСЛЕННЫЕ РЕЗУЛЬТАТЫ

A. Пропускная способность

Результаты анализа пропускной способности (рис. 5) показывают, что все рассмотренные конфигурации превосходят базовую по средней пропускной способности сети и, в частности, обеспечивают улучшение пропускной способности вертикальных агентов.

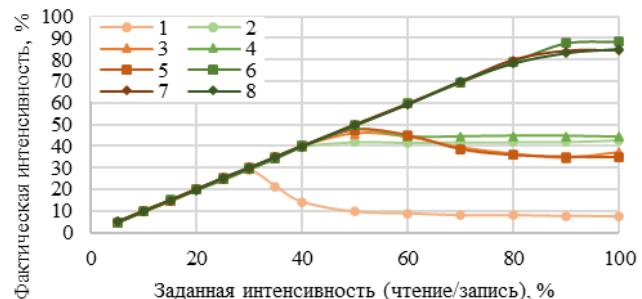
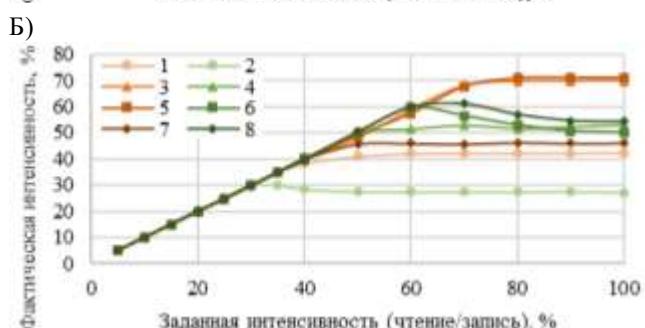
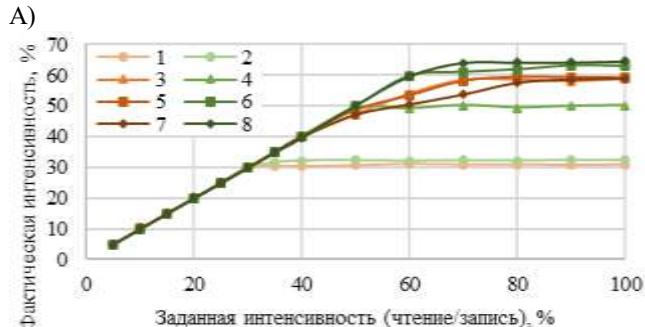


Рис. 5. Полученная средняя пропускная способность, в зависимости от заданной интенсивности запросов для агентов (А) во всех узлах; (Б) в горизонтальных узлах; (В) в вертикальных узлах

При малых нагрузках до интенсивности порядка 30% различия между конфигурациями отсутствуют, что соответствует режиму без выраженной конкуренции за сетевые ресурсы.

Базовая конфигурация 1 характеризуется резким снижением пропускной способности вертикальных агентов при увеличении нагрузки: после 30% заданной интенсивности их фактическая интенсивность монотонно падает и при 100% составляет лишь 7–8%. Это обусловлено коллизиями при маршрутизации вертикальных потоков и доминированием горизонтальных агентов, число которых вдвое больше.

Применение алгоритма ADOR в конфигурации 2 приводит преимущественно к перераспределению пропускной способности между агентами без существенного роста суммарной пропускной способности. Пропускная способность вертикальных агентов возрастает за счет снижения пропускной способности горизонтальных, что связано с инверсией порядка обхода координат для вертикальных потоков.

Конфигурации 3 и 5 демонстрируют близкие характеристики. Использование алгоритма DOR приводит к снижению пропускной способности вертикальных агентов после интенсивности около 50% и выходу на плато порядка 35%, аналогично базовой конфигурации. При этом данные конфигурации обеспечивают наивысшую пропускную способность горизонтальных агентов. Улучшение суммарных показателей по сравнению с базовой конфигурацией достигается за счет увеличенной емкости сети и большего числа портов подключения модулей памяти, что обеспечивает более равномерное распределение трафика.

В конфигурации 4 применение алгоритма ADOR обеспечивает более симметричное распределение пропускной способности между горизонтальными и вертикальными агентами по сравнению с конфигурацией 3. Однако из-за преобладания горизонтальных агентов средняя пропускная способность оказывается ниже, чем в конфигурации 3.

Конфигурация 6 демонстрирует наилучшие показатели по пропускной способности вертикальных агентов, что обусловлено сочетанием алгоритма ADOR и дополнительными портами подключения, улучшающими распределение вертикального трафика. При этом для горизонтальных агентов наблюдается

снижение пропускной способности при интенсивностях выше 60%, однако в среднем по всем агентам данная конфигурация относится к числу лидирующих.

Конфигурации 7 и 8 обеспечивают сопоставимо высокую, близкую к максимальной, пропускную способность вертикальных агентов на уровне порядка 85%. Различия между ними проявляются для горизонтальных агентов: конфигурация 7 уступает конфигурации 8 из-за меньшего числа доступных портов памяти. В результате конфигурация 8 демонстрирует наивысшую среднюю пропускную способность, тогда как конфигурация 7 по этому показателю сопоставима с конфигурацией 5. Для конфигурации 8 при этом отмечается снижение пропускной способности горизонтальных агентов при интенсивностях выше 70%.

Анализ средней интенсивности запросов, приходящихся на один порт распределенной памяти (рис. 6), показывает, что для всех рассмотренных конфигураций данная величина не превышает 25%. Это означает, что каждый порт модулей распределенной памяти используется в среднем не чаще одного раза в четыре такта как для операций чтения, так и для операций записи. Таким образом, при наблюдаемой нагрузке на порты распределенной памяти становится возможной их работа на более низкой тактовой частоте по сравнению с Mesh-сетью без заметного влияния на ее пропускную способность. Потенциальное снижение тактовой частоты модулей памяти в 2–4 раза может привести к существенному сокращению энергопотребления вычислительной системы.

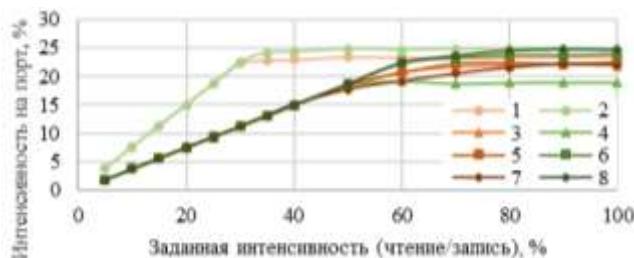


Рис. 6. Средняя интенсивность (чтение/запись), приходящаяся на 1 порт распределенной памяти, в зависимости от заданной интенсивности запросов

Кроме того, результаты показывают, что конфигурации с одним портом памяти (1 и 2) достигают точки насыщения при интенсивности порядка 30%, тогда как конфигурации с двумя портами памяти (3–8) начинают насыщаться лишь при интенсивностях около 50%. Это демонстрирует ограниченность пропускной способности сети в архитектурах с модулями распределенной памяти, подключенными по 1 порту.

Для наглядной иллюстрации распределения нагрузки при заданной интенсивности запросов 100% на рис. 7 приведены тепловые карты фактической интенсивности агентов для всех рассматриваемых конфигураций. Числа по периметру соответствуют интенсивности отдельных агентов, а значение в центре отражает среднюю интенсивность запросов, приходящуюся на 1 порт распределенной памяти.

В базовой конфигурации 1 наблюдается резкая деградация пропускной способности вертикальных

агентов до 5–10%, обусловленная коллизиями при маршрутизации их запросов. Пропускная способность горизонтальных агентов существенно неоднородна и изменяется от 17% до 68% по мере удаления от центра, что связано с прямоугольной формой сети.

Применение алгоритма ADOR в конфигурации 2 приводит к выравниванию пропускной способности вертикальных агентов на уровне 41–45%, однако не устраняет неравномерность горизонтальных агентов, значения которых варьируются в диапазоне 13–47%. Такая картина является следствием перераспределения маршрутов без изменения геометрии сети и числа портов памяти.

Конфигурации 3 и 5 демонстрируют сходное поведение. Для вертикальных агентов сохраняется заметная неравномерность (27–43% и 26–47% соответственно), обусловленная коллизиями при маршрутизации с использованием DOR. При этом пропускная способность горизонтальных агентов становится существенно более равномерной (51–81% и 58–82%), что объясняется увеличенным числом портов подключения модулей распределенной памяти.

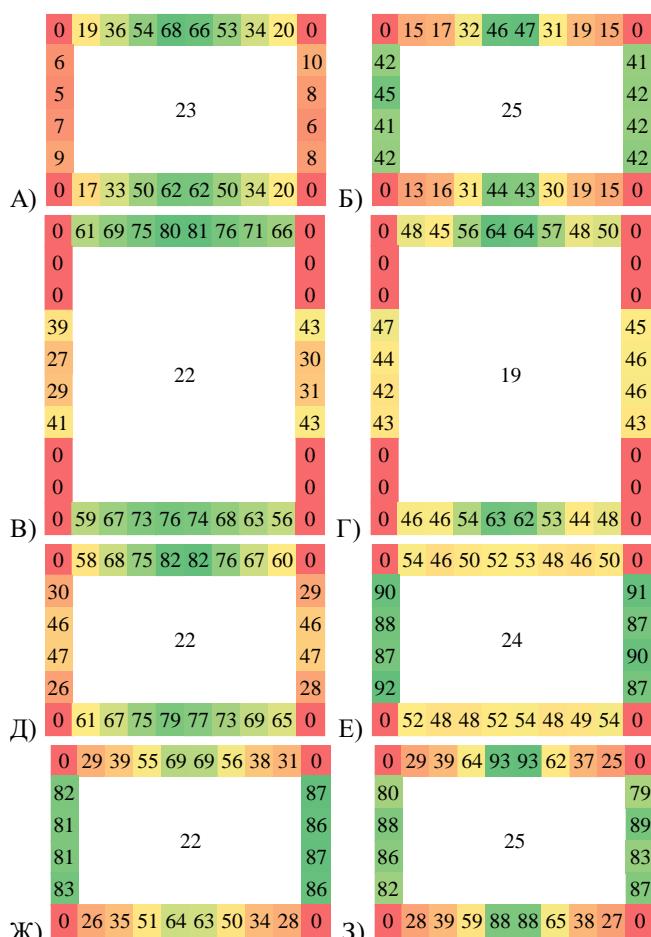


Рис. 7. Тепловая карта: пропускная способность (%) агентов при заданной интенсивности 100% для конфигурации (А) 1; (Б) 2; (В) 3; (Г) 4; (Д) 5; (Е) 6; (Ж) 7; (З) 8; число в центре соответствует средней интенсивности (%) запросов на порт памяти

В конфигурации 4 использование алгоритма ADOR, как и в конфигурации 2, выравнивает пропускную способность вертикальных агентов до 42–47%.

Одновременно за счет большего числа портов памяти по сравнению с базовой конфигурацией уменьшается неравномерность распределения нагрузки между горизонтальными агентами (44–64%).

Конфигурация 6 обеспечивает наивысшую и наиболее равномерную пропускную способность вертикальных агентов (87–92%) благодаря сочетанию алгоритма ADOR и двухпортового подключения, обеспечивающего симметричное распределение вертикального трафика. Пропускная способность горизонтальных агентов при этом остается стабильной в диапазоне 46–54%, чему способствует увеличенное число портов памяти.

Для конфигураций 7 и 8 характерна высокая и выровненная пропускная способность вертикальных агентов (81–87% и 80–89% соответственно). В конфигурации 7 это достигается за счет разделения сетей и использования инвертированного порядка маршрутизации (DOR + IDOR), что устраняет коллизии между горизонтальными и вертикальными потоками. В конфигурации 8 аналогичный эффект обеспечивается применением алгоритма ADOR. Для обеих конфигураций сохраняется неравномерность пропускной способности горизонтальных агентов, обусловленная прямоугольной формой сети и, в случае конфигурации 7, ограниченным числом доступных портов памяти.

B. Средняя задержка

Анализ средней задержки обработки запросов (рис. 8) показывает, что для каждой из рассматриваемых конфигураций существует характерная точка насыщения, после которой задержка резко возрастает. Все конфигурации демонстрируют смещение точки насыщения вправо по сравнению с базовой.

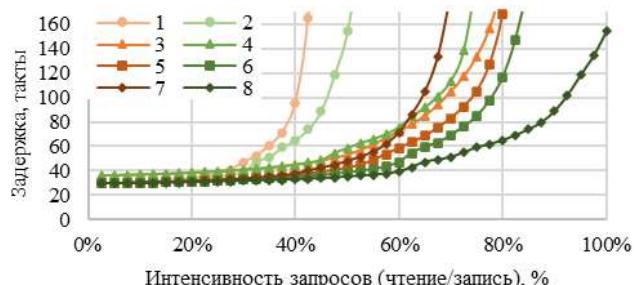


Рис. 8. Средняя задержка обработки запросов для конфигураций 1-8

Все конфигурации сдвигают точку насыщения вправо относительно базовой. Базовая конфигурация достигает насыщения при интенсивности 42.5%. Для конфигураций 2–8 получены следующие значения: 2 – 50%, 3 – 77.5%, 4 – 73.5%, 5 – 80%, 6 – 83%, 7 – 70%, 8 – 100%.

При малых нагрузках до интенсивности порядка 30% средние задержки всех конфигураций практически совпадают и лежат в узком интервале 29–36 тактов, что

соответствует режиму отсутствия выраженной конкуренции за сетевые ресурсы.

В диапазоне средних интенсивностей порядка 45–60% конфигурация 7 демонстрирует меньшую среднюю задержку по сравнению с конфигурациями 3 и 4. Однако при дальнейшем росте нагрузки ее задержка превышает значения для конфигураций 3 и 4, которые сохраняют более низкие задержки вплоть до собственных точек насыщения. В остальном пересечений между кривыми при приближении к насыщению не наблюдается: относительный порядок конфигураций по средней задержке сохраняется во всем соответствующем диапазоне интенсивностей.

Более детальная структура задержек в точках насыщения представлена на рис. 9 в виде распределения средней задержки между входными и выходными буферами, а также между трафиком запросов и ответов. Анализ по типу буферов позволяет выделить две характерные группы конфигураций.

Для конфигураций 1, 3, 5 и 7, использующих алгоритм DOR, основная доля задержки приходится на входные буферы и составляет порядка 51–54%. Это указывает на доминирование коллизий при маршрутизации запросов. Баланс между входными и выходными буферами в этих конфигурациях близок к симметричному, при этом наиболее равномерное распределение наблюдается в конфигурации 3. Для конфигураций 2, 4, 6 и 8 с алгоритмом ADOR характерна противоположная картина: преобладают задержки в выходных буферах на уровне 57–58%, что свидетельствует о более равномерном распределении трафика по сети и смещении узких мест в область конкуренции между ВК.

Анализ распределения задержек по типу трафика выявляет устойчивые пары конфигураций с близкой структурой задержек: 1–7, 2–8, 3–5 и 4–6. В конфигурациях 1, 2, 7 и 8 основная доля задержки приходится на запросы, что связано с высокой плотностью сходящихся потоков в центральной области сети. Это согласуется с результатами анализа пропускной способности (рис. 7), где наблюдается выраженная неравномерность нагрузки горизонтальных агентов. Напротив, в конфигурациях 3 и 4, а также 5 и 6, более равномерное распределение трафика в центральной зоне приводит к снижению относительной доли задержек по запросам и росту вклада задержек по ответам.

Во всех парах 1–2, 3–4, 5–6 и 7–8 применение алгоритма ADOR приводит к уменьшению доли задержек, приходящихся на запросы, за счет дополнительного выравнивания распределения трафика. При этом в конфигурациях 2 и 8 баланс между запросами и ответами улучшается, тогда как в конфигурациях 4 и 6 наблюдается дополнительное смещение баланса в сторону задержек по ответам.

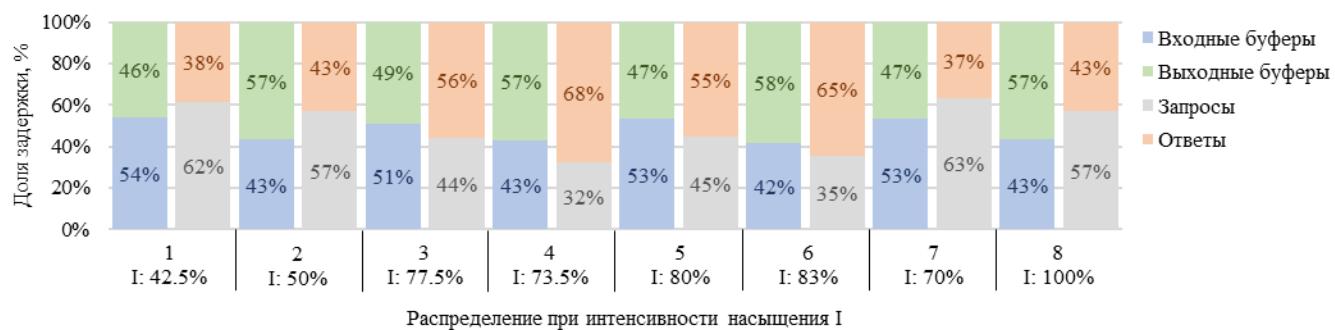


Рис. 9. Распределение средней задержки (~160 тактов) при насыщении сети

C. Эффективность конфигураций

Для сопоставления производительности на ресурс используется агрегированный ресурсный показатель R , определяемый как произведение относительных множителей по числу триггеров, соединений между узлами и портов модулей распределенной памяти по отношению к базовой конфигурации. На рис. 10 представлены зависимости двух ключевых метрик от R : (A) средней фактической интенсивности при заданной интенсивности 100% и (Б) интенсивности насыщения.

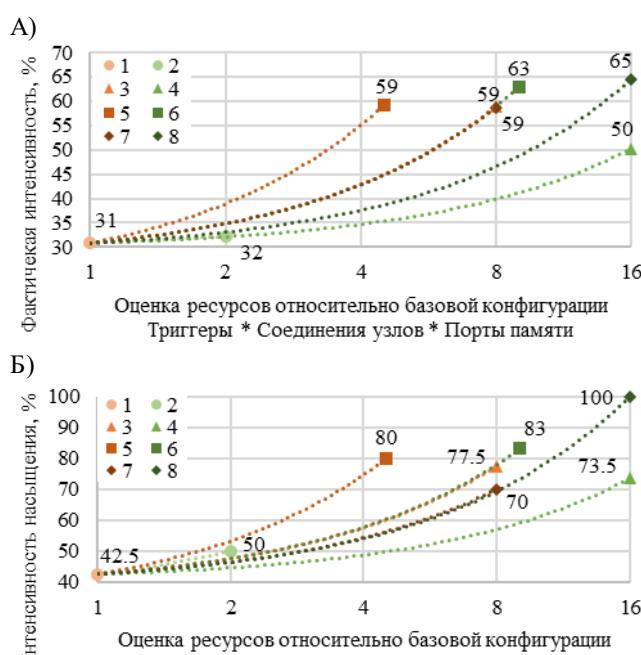


Рис. 10. Эффективность конфигураций по ключевым характеристикам (А) средняя фактическая интенсивность при заданной интенсивности 100%; (Б) интенсивность насыщения

Анализ полученных зависимостей показывает, что конфигурация 5 демонстрирует наивысшую эффективность по критерию «производительность на ресурс» одновременно по обеим метрикам. В противоположность этому конфигурация 4 оказывается наименее эффективной, несмотря на существенное увеличение аппаратных ресурсов.

Конфигурация 8 обеспечивает наилучшие абсолютные значения пропускной способности и интенсивности насыщения, однако требует максимальных затрат ресурсов. При этом по эффективности на ресурс она

превосходит конфигурацию 4, несмотря на одинаковое значение агрегированного показателя R , что указывает на принципиальную роль архитектурных решений, а не только объема ресурсов.

Конфигурации 3 и 6 демонстрируют одинаковую эффективность на ресурс по обеим метрикам, формируя общую кривую эффективности, что свидетельствует о сопоставимой отдаче от различных архитектурных подходов при равных затратах ресурсов. Аналогично, конфигурации 3 и 7 обеспечивают одинаковую фактическую интенсивность при заданной интенсивности 100% при равных значениях R , однако по интенсивности насыщения на ресурс конфигурация 7 уступает конфигурации 3 и располагается на одной кривой с конфигурацией 8.

Конфигурация 2 характеризуется минимальным выигрышем по обеим метрикам при наименьшем увеличении ресурсов. По эффективности на ресурс при интенсивности 100% она сопоставима с конфигурацией 4, а по интенсивности насыщения уступает лишь конфигурации 5.

VI. ЗАКЛЮЧЕНИЕ

В данной работе выполнено комплексное исследование архитектурных подходов к повышению эффективности Mesh-сети в вычислительной системе с распределенной памятью на основе методов ранней архитектурной оценки. Рассмотрены решения, затрагивающие как алгоритмы маршрутизации, так и структуру сети и маршрутизаторов, включая изменение формы сети, масштабирование числа физических и ВК, а также полное дублирование сетевой инфраструктуры. Для сопоставления эффективности конфигураций использованы метрики пропускной способности, средней задержки и производительности на ресурс, что позволило рассматривать архитектурные решения в контексте многоокритериальной оптимизации.

Оценка задержек выполнена с использованием аналитической модели, ранее разработанной авторами и ориентированной на вычислительные системы с распределенной памятью, а пропускная способность оценивалась на основе потактового моделирования. Такое сочетание позволило получить интерпретируемые результаты при существенно меньших затратах времени по сравнению с использованием исключительно детальных симуляционных моделей. Сопоставление аналитической и потактовой оценок подтверждает применимость выбранного подхода для раннего анализа

архитектурных решений.

Сравнение конфигураций по показателю производительности на ресурс показало, что наибольшую эффективность демонстрируют решения, сочетающие умеренное увеличение аппаратных ресурсов с улучшением распределения трафика, в то время как простое наращивание емкости без изменения структуры потоков приводит к убывающей отдаче. Это подчеркивает важность комплексного анализа архитектурных решений и невозможность выбора оптимальной конфигурации на основе одной метрики.

Рассмотренные в работе архитектурные подходы и их сопоставление в рамках единого сценария впервые применены к вычислительной системе с распределенной памятью, что позволяет использовать полученные выводы в качестве практических рекомендаций при проектировании таких систем. Предложенный подход к анализу может быть использован на ранних этапах разработки для обоснованного выбора архитектуры сети на кристалле с учетом компромиссов между пропускной способностью, задержками и затратами аппаратных ресурсов.

БИБЛИОГРАФИЯ

- [1] Peccerillo, Biagio, et al. "A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives." *Journal of Systems Architecture* 129 (2022): 102561.
- [2] McKee, Sally A. "Reflections on the memory wall." *Proceedings of the 1st conference on Computing frontiers*. 2004.
- [3] Chen, Xiaowen. Efficient Memory Access and Synchronization in NoC-based Many-core Processors. Diss. KTH Royal Institute of Technology, 2019.
- [4] Gholami, Amir, et al. "Ai and memory wall." *IEEE Micro* 44.3 (2024): 33-39.
- [5] Milton, Jonathan, and Payman Zarkesh-Ha. "Impacts of topology and bandwidth on distributed shared memory systems." *Computers* 12.4 (2023): 86.
- [6] Gries, Matthias. "Methods for evaluating and covering the design space during early design development." *Integration* 38.2 (2004): 131-183.
- [7] Jerger, Natalie Enright, Tushar Krishna, and Li-Shiuan Peh. *On-chip networks*. Springer Nature, 2022.
- [8] Chen, Wai Kai. *The electrical engineering handbook*. Elsevier, 2004.
- [9] Palermo, Gianluca, et al. "A system-level methodology for fast multi-objective design space exploration." *Proceedings of the 13th ACM Great Lakes symposium on VLSI*. 2003.
- [10] Deshwal, Aryan, et al. "MOOS: A multi-objective design space exploration and optimization framework for NoC enabled manycore systems." *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-23.
- [11] Xue, Runzhen, et al. "Multi-objective optimization in cpu design space exploration: Attention is all you need." *arXiv preprint arXiv:2410.18368* (2024).
- [12] Bakhoda, Ali, John Kim, and Tor M. Aamodt. "Throughput-effective on-chip networks for manycore accelerators." *2010 43rd Annual IEEE/ACM international symposium on microarchitecture*. IEEE, 2010.
- [13] Lallas, Efthymios N. "An Evaluation of Routing Algorithms in Traffic Engineering and Quality of Service Provision of Network on Chips." *Engineering* 13.1 (2021): 1-17.
- [14] Khaidukov, Danila, and Alexandr Alekseev. "Routing Collision Mitigation Approaches in a Distributed Memory System on Chip". *Nanoindustiya*[Nanoindustry] 11s 18.135 (2025): 1298-1304.
- [15] Dimitrakopoulos, Giorgos, Anastasios Psarras, and Ioannis Seitanidis. *Microarchitecture of Network-on-chip Routers*. Vol. 1025. Berlin, Germany: Springer, 2015.
- [16] Hansson, Andreas, Kees Goossens, and Andrei Rădulescu. "Avoiding message-dependent deadlock in network-based systems on chip." *VLSI design* 2007.1 (2007): 095859.
- [17] Moscibroda, Thomas, and Onur Mutlu. "A case for bufferless routing in on-chip networks." *Proceedings of the 36th annual international symposium on Computer architecture*. 2009.
- [18] Fallin, Chris, et al. "MinBD: Minimally-buffered deflection routing for energy-efficient interconnect." *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*. IEEE, 2012.
- [19] Ausavarungnirun, Rachata, and Onur Mutlu. "Energy-Efficient Deflection-based On-chip Networks: Topology, Routing, Flow Control." *arXiv preprint arXiv:2112.02516* (2021).
- [20] Fernandes, Ramon, et al. "OcNoC: Efficient one-cycle router implementation for 3d mesh network-on-chip." *2015 28th International Conference on VLSI Design*. IEEE, 2015.
- [21] Concer, Nicola, Michele Petracca, and Luca P. Carloni. "Distributed flit-buffer flow control for networks-on-chip." *Proceedings of the 6th IEEE/ACM/IFIP international conference on Hardware/Software codesign and system synthesis*. 2008.
- [22] Chuang, Shang-Tse, et al. "Matching output queueing with a combined input/output-queued switch." *IEEE Journal on Selected Areas in Communications* 17.6 (1999): 1030-1039.
- [23] Papaphilippou, Philippos, and Thiem Van Chu. "Efficient deadlock avoidance for 2-D mesh NoCs that use OQ or VOQ routers." *IEEE Transactions on Computers* 73.5 (2024): 1414-1426.
- [24] Dally, William J. "Virtual-channel flow control." *ACM SIGARCH Computer Architecture News* 18.2SI (1990): 60-68.
- [25] Nicopoulos, Chrysostomos A., et al. "ViChaR: A dynamic virtual channel regulator for network-on-chip routers." *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*. IEEE, 2006.
- [26] Mallappa, Uday, Chung-Kuan Cheng, and Bill Lin. "JARVA: Joint Application-Aware Oblivious Routing and Static Virtual Channel Allocation." *IEEE Des. Test* 39.6 (2022): 16-27.
- [27] Rao, Supriya, et al. "Vix: Virtual input crossbar for efficient switch allocation." *Proceedings of the 51st Annual Design Automation Conference*. 2014.
- [28] Noh, Seongmin, et al. "Multiplane virtual channel router for network-on-chip design." *2006 First International Conference on Communications and Electronics*. IEEE, 2006.
- [29] Nadi, M., M. H. Ghadiry, and M. K. Dermany. "The effect of number of virtual channel on NOC EDP." *Journal of applied mathematics & informatics* 2010 (2010): 539-551.
- [30] Carara, Everton, Fernando Moraes, and Ney Calazans. "Router architecture for high-performance NoCs." *Proceedings of the 20th annual conference on Integrated circuits and systems design*. 2007.
- [31] Gulzari, Usman Ali, et al. "Comparative analysis of 2D mesh topologies with additional communication links for on-chip networks." *Computer Networks* 241 (2024): 110193.
- [32] Кожин, А., Е. Кожин, and Д. Шпагитев. "Исследование топологий сетей на кристалле многоядерных процессоров с архитектурой «Эльбрус»." *Электроника: Наука, технология, бизнес* 7 (2020): 132-136.
- [33] Moadeli, Mahmoud, Alireza Shahabi, and Wim Vanderbauwheide. "Analytical modelling of communication in the rectangular mesh noc." *2007 International Conference on Parallel and Distributed Systems*. IEEE, 2007.
- [34] Balfour, James, and William J. Dally. "Design tradeoffs for tiled CMP on-chip networks." *ACM International conference on supercomputing 25th anniversary volume*. 2006.
- [35] Das, Reetuparna, et al. "Catnap: Energy proportional multiple network-on-chip." *Proceedings of the 40th annual international symposium on Computer architecture*. 2013.
- [36] Yoon, Young Jin, et al. "Virtual channels vs. multiple physical networks: A comparative analysis." *Proceedings of the 47th Design Automation Conference*. 2010.
- [37] Mishra, Asit K., Onur Mutlu, and Chita R. Das. "A heterogeneous multiple network-on-chip design: an application-aware approach." *Proceedings of the 50th annual design automation conference*. 2013.
- [38] Bjerrregaard, Tobias, and Shankar Mahadevan. "A survey of research and practices of network-on-chip." *ACM Computing Surveys (CSUR)* 38.1 (2006): 1-es.
- [39] Khan, Sarzamin, et al. "Comparative analysis of network-on-chip simulation tools." *IET Computers & Digital Techniques* 12.1 (2018): 30-38.
- [40] Catania, Vincenzo, et al. "Noxim: An open, extensible and cycle-accurate network on chip simulator." *2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP)*. IEEE, 2015.

- [41] Khaidukov, Danila, and Alexandr Alekseev. "An Analytical Latency Estimation Approach in a Mesh Network in a Distributed Memory System on Chip ". International Journal of Open Information Technologies 14.1 (2026): 66-80.
- [42] Thomas, Marlin U. "Queueing systems. volume 1: Theory (leonard kleinrock)." SIAM Review 18.3 (1976): 512-514.

Efficiency Improvement Approaches for a Mesh Network in a Distributed Memory System on Chip

Danila Khaidukov, Aleksandr Alekseev

Abstract—This paper studies architectural approaches to improving the efficiency of a mesh network in a distributed memory system on chip. The focus is on solutions applicable at early design stages, enabling analysis of trade-offs between network performance and hardware resource costs. The approaches include routing algorithm selection, network shape modification, scaling of physical and virtual channels, and duplication of the network infrastructure. Network efficiency is evaluated using throughput, average request latency, and performance per resource as the main metrics. Latency is estimated using an analytical model previously developed by the authors and tailored to distributed memory systems, while throughput is evaluated via cycle-accurate simulation. All approaches are compared within a unified experimental scenario, ensuring consistent assessment of their impact on traffic distribution, saturation behavior, and latency structure. The results show that the most efficient configurations improve traffic distribution with a moderate increase in hardware resources, whereas simple capacity scaling without structural changes leads to diminishing returns. The presented analysis provides practical guidance for selecting mesh network architectures in distributed memory systems at early design stages.

Keywords—analytical modeling, computing system, cycle-accurate modeling, distributed memory, network on chip.

REFERENCES

- [1] Peccerillo, Biagio, et al. "A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives." *Journal of Systems Architecture* 129 (2022): 102561.
- [2] McKee, Sally A. "Reflections on the memory wall." *Proceedings of the 1st conference on Computing frontiers*. 2004.
- [3] Chen, Xiaowen. Efficient Memory Access and Synchronization in NoC-based Many-core Processors. Diss. KTH Royal Institute of Technology, 2019.
- [4] Gholami, Amir, et al. "Ai and memory wall." *IEEE Micro* 44.3 (2024): 33-39.
- [5] Milton, Jonathan, and Payman Zarkesh-Ha. "Impacts of topology and bandwidth on distributed shared memory systems." *Computers* 12.4 (2023): 86.
- [6] Gries, Matthias. "Methods for evaluating and covering the design space during early design development." *Integration* 38.2 (2004): 131-183.
- [7] Jerger, Natalie Enright, Tushar Krishna, and Li-Shiuan Peh. On-chip networks. Springer Nature, 2022.
- [8] Chen, Wai Kai. The electrical engineering handbook. Elsevier, 2004.
- [9] Palermo, Gianluca, et al. "A system-level methodology for fast multi-objective design space exploration." *Proceedings of the 13th ACM Great Lakes symposium on VLSI*. 2003.
- [10] Deshwal, Aryan, et al. "MOOS: A multi-objective design space exploration and optimization framework for NoC enabled manycore systems." *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-23.
- [11] Xue, Runzhen, et al. "Multi-objective optimization in cpu design space exploration: Attention is all you need." *arXiv preprint arXiv:2410.18368* (2024).
- [12] Bakhoda, Ali, John Kim, and Tor M. Aamodt. "Throughput-effective on-chip networks for manycore accelerators." *2010 43rd Annual IEEE/ACM international symposium on microarchitecture*. IEEE, 2010.
- [13] Lallas, Efthymios N. "An Evaluation of Routing Algorithms in Traffic Engineering and Quality of Service Provision of Network on Chips." *Engineering* 13.1 (2021): 1-17.
- [14] Khaidukov, Danila, and Aleksandr Alekseev. "Routing Collision Mitigation Approaches in a Distributed Memory System on Chip". *Nanoindustiya*[Nanoindustry] 11s 18.135 (2025): 1298-1304.
- [15] Dimitrakopoulos, Giorgos, Anastasios Psarras, and Ioannis Seitanidis. Microarchitecture of Network-on-chip Routers. Vol. 1025. Berlin, Germany: Springer, 2015.
- [16] Hansson, Andreas, Kees Goossens, and Andrei Rădulescu. "Avoiding message-dependent deadlock in network-based systems on chip." *VLSI design* 2007.1 (2007): 095859.
- [17] Moscibroda, Thomas, and Onur Mutlu. "A case for bufferless routing in on-chip networks." *Proceedings of the 36th annual international symposium on Computer architecture*. 2009.
- [18] Fallin, Chris, et al. "MinBD: Minimally-buffered deflection routing for energy-efficient interconnect." *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*. IEEE, 2012.
- [19] Ausavarungnirun, Rachata, and Onur Mutlu. "Energy-Efficient Deflection-based On-chip Networks: Topology, Routing, Flow Control." *arXiv preprint arXiv:2112.02516* (2021).
- [20] Fernandes, Ramon, et al. "OcNoC: Efficient one-cycle router implementation for 3d mesh network-on-chip." *2015 28th International Conference on VLSI Design*. IEEE, 2015.
- [21] Concer, Nicola, Michele Petracca, and Luca P. Carloni. "Distributed flit-buffer flow control for networks-on-chip." *Proceedings of the 6th IEEE/ACM/IFIP international conference on Hardware/Software codesign and system synthesis*. 2008.
- [22] Chuang, Shang-Tse, et al. "Matching output queueing with a combined input/output-queued switch." *IEEE Journal on Selected Areas in Communications* 17.6 (1999): 1030-1039.
- [23] Papaphilippou, Philippos, and Thiem Van Chu. "Efficient deadlock avoidance for 2-D mesh NoCs that use OQ or VOQ routers." *IEEE Transactions on Computers* 73.5 (2024): 1414-1426.
- [24] Dally, William J. "Virtual-channel flow control." *ACM SIGARCH Computer Architecture News* 18.2SI(1990): 60-68.
- [25] Nicopoulos, Chrysostomos A., et al. "ViChaR: A dynamic virtual channel regulator for network-on-chip routers." *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*. IEEE, 2006.
- [26] Mallappa, Uday, Chung-Kuan Cheng, and Bill Lin. "JARVA: Joint Application-Aware Oblivious Routing and Static Virtual Channel Allocation." *IEEE Des. Test* 39.6 (2022): 16-27.
- [27] Rao, Supriya, et al. "Vix: Virtual input crossbar for efficient switch allocation." *Proceedings of the 51st Annual Design Automation Conference*. 2014.
- [28] Noh, Seongmin, et al. "Multiplane virtual channel router for network-on-chip design." *2006 First International Conference on Communications and Electronics*. IEEE, 2006.
- [29] Nadi, M., M. H. Ghadiry, and M. K. Dermany. "The effect of number of virtual channel on NOC EDP." *Journal of applied mathematics & informatics* 2010 (2010): 539-551.
- [30] Carara, Everton, Fernando Moraes, and Ney Calazans. "Router architecture for high-performance NoCs." *Proceedings of the 20th annual conference on Integrated circuits and systems design*. 2007.
- [31] Gulzari, Usman Ali, et al. "Comparative analysis of 2D mesh topologies with additional communication links for on-chip

- networks." Computer Networks 241 (2024): 110193.
- [32] Kozhin, A., E. Kozhin, and D. Shpagilev. "Investigation of Network-on-Chip Topologies for Multicore Processors with the Elbrus Architecture." Electronics Science Technology Business 198.7 (2020): 132-136.
- [33] Moadeli, Mahmoud, Alireza Shahabi, and Wim Vanderbauwhede. "Analytical modelling of communication in the rectangular mesh noc." 2007 International Conference on Parallel and Distributed Systems. IEEE, 2007.
- [34] Balfour, James, and William J. Dally. "Design tradeoffs for tiled CMP on-chip networks." ACM International conference on supercomputing 25th anniversary volume. 2006.
- [35] Das, Reetuparna, et al. "Catnap: Energy proportional multiple network-on-chip." Proceedings of the 40th annual international symposium on Computer architecture. 2013.
- [36] Yoon, Young Jin, et al. "Virtual channels vs. multiple physical networks: A comparative analysis." Proceedings of the 47th Design Automation Conference. 2010.
- [37] Mishra, Asit K., Onur Mutlu, and Chita R. Das. "A heterogeneous multiple network-on-chip design: an application-aware approach." Proceedings of the 50th annual design automation conference. 2013.
- [38] Bjerrregaard, Tobias, and Shankar Mahadevan. "A survey of research and practices of network-on-chip." ACM Computing Surveys (CSUR) 38.1 (2006): 1-es.
- [39] Khan, Sarzamin, et al. "Comparative analysis of network-on-chip simulation tools." IET Computers & Digital Techniques 12.1 (2018): 30-38.
- [40] Catania, Vincenzo, et al. "Noxim: An open, extensible and cycle-accurate network on chip simulator." 2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP). IEEE, 2015.
- [41] Khaidukov, Daniil, and Alexandr Alekseev. "An Analytical Latency Estimation Approach in a Mesh Network in a Distributed Memory System on Chip ". International Journal of Open Information Technologies 14.1 (2026): 66-80.
- [42] Thomas, Marlin U. "Queueing systems. volume 1: Theory (leonard kleinrock)." SIAM Review 18.3 (1976): 512-514.