Карпов Демид Константинович, студент 4 курса
Уфимский университет науки и технологий

ПЕРИФЕРИЙНЫЕ ВЫЧИСЛЕНИЯ В СИСТЕМАХ ИНТЕРНЕТА ВЕЩЕЙ: АРХИТЕКТУРА, ПРЕИМУЩЕСТВА И РИСКИ

Аннотация. В статье рассмотрена роль периферийных вычислений в системах Интернета вещей. Описана многоуровневая архитектура обработки данных, выделены преимущества локального анализа информации, основные ограничения и угрозы безопасности. Сформулированы практические принципы построения надежных edge-систем.

Ключевые слова: Периферийные вычисления, Интернет вещей, edge computing, облачные технологии, кибербезопасность, распределенные системы.

Введение

Развитие Интернета вещей приводит к постоянному увеличению числа датчиков, контроллеров, камер и исполнительных устройств. Традиционная схема предполагает передачу всей информации в удаленный центр обработки данных, однако такой подход не всегда соответствует требованиям реального времени. Задержка сети, нестабильность соединения и высокая стоимость передачи больших потоков данных ограничивают применение исключительно облачной модели. Поэтому часть вычислений переносится ближе к источнику информации – на шлюзы, локальные серверы и специализированные вычислительные узлы.

Периферийные вычисления, или edge computing, представляют собой распределенную модель, в которой обработка и хранение данных выполняются между конечным устройством и централизованным облаком. Согласно концептуальной модели NIST, туманные и периферийные уровни дополняют облачную инфраструктуру и позволяют размещать вычислительные ресурсы в непосредственной близости от устройств Интернета вещей [1]. Актуальность такого подхода связана с развитием промышленной автоматизации, интеллектуального транспорта, видеонаблюдения, медицины и систем «умного города».

Архитектура периферийной системы

Типовая архитектура включает три взаимосвязанных уровня. Первый уровень образуют конечные устройства: сенсоры, камеры, счетчики, производственные контроллеры и мобильные терминалы. Они формируют первичные данные и выполняют простейшие операции. Второй уровень представлен edge-шлюзами и локальными вычислительными узлами. Здесь выполняются фильтрация, агрегация, распознавание событий, временное хранение и управление устройствами. Третий уровень образуют облачные или корпоративные центры обработки данных, предназначенные для долгосрочного хранения, обучения моделей, объединения информации из разных объектов и централизованного администрирования.

Современная архитектура Multi-access Edge Computing предусматривает размещение вычислительных приложений на границе сети доступа. Это позволяет использовать сведения о состоянии сети и предоставлять сервисы с малой задержкой [2]. В практической системе границы между уровнями могут изменяться: часть функций переносится на более мощный шлюз, а ресурсоемкие задачи остаются в облаке. Поэтому распределение нагрузки должно определяться требованиями конкретного процесса, доступной производительностью и допустимым временем реакции.

Преимущества локальной обработки данных

Главное преимущество edge computing состоит в сокращении задержки. Решение принимается непосредственно на объекте, без ожидания передачи данных в удаленный центр. Это особенно важно для аварийной остановки оборудования, управления транспортом и анализа видеопотока. RFC 9556 отмечает, что временная чувствительность, объем данных и необходимость работы при нестабильном соединении являются ключевыми причинами перехода IoT-систем к периферийной обработке [3].

Второе преимущество связано со снижением сетевой нагрузки. Камера высокого разрешения может формировать непрерывный поток, но в облако часто требуется передавать только распознанные события, метаданные или короткие фрагменты. Локальная фильтрация уменьшает объем трафика и затраты на хранение. Третьим преимуществом является автономность: при временной потере внешней связи объект продолжает выполнять критические функции, а накопленные сведения синхронизируются после восстановления канала.

Ограничения и риски

Распределенная структура усложняет сопровождение. Организация получает множество удаленных узлов, различающихся производительностью, версиями программного обеспечения и качеством связи. Поэтому необходимы централизованные обновления, контроль конфигураций, мониторинг ресурсов и восстановление после сбоев.

Серьезной проблемой является физическая доступность периферийного оборудования. Шлюз может находиться в цехе, автомобиле или общественном пространстве, поэтому злоумышленник способен получить доступ к накопителю, портам или сети. Кроме того, увеличение числа компонентов расширяет поверхность атаки. Рекомендации NIST для производителей IoT-продуктов предусматривают управление идентификацией устройств, защиту данных, безопасное обновление программного обеспечения и регистрацию событий [4].

Отдельный риск связан с ограниченными вычислительными ресурсами. На периферийном узле невозможно бесконечно увеличивать число приложений и моделей машинного обучения. Перегрузка процессора или памяти увеличивает задержку и может нарушить обработку критических событий. Следовательно, система должна иметь механизмы приоритизации, ограничения ресурсов и переноса второстепенных задач в облако.

Принципы построения надежной edge-инфраструктуры

Проектирование следует начинать с классификации данных и операций. Для каждой функции определяется максимально допустимая задержка, требуемая автономность и последствия отказа. Критические алгоритмы размещаются локально, аналитика длительных периодов – в центральной системе, а промежуточные задачи распределяются динамически. Архитектура ETSI MEC предполагает совместную работу периферийной и облачной инфраструктуры, а не их противопоставление [2].

Второй принцип – централизованное управление жизненным циклом. Каждый узел должен иметь уникальную идентичность, контролируемую конфигурацию и проверяемый источник обновлений. Практически это означает использование подписанных пакетов, поэтапное развертывание и возможность возврата к стабильной версии [4].

Третий принцип – наблюдаемость. Необходимо собирать метрики нагрузки, связи, времени обработки и ошибок. Журналы передаются в центральную систему, а критические уведомления формируются локально. Дополнительно применяются резервирование, очереди сообщений и временное хранение данных при разрыве связи.

Четвертый принцип – защита данных на всех этапах. Требуются шифрование каналов, разграничение прав, минимизация привилегий и регулярная ротация ключей. Защитные меры должны учитывать распределенное размещение ресурсов и возможность физического доступа к узлам [4].

Области практического применения

В промышленности периферийные узлы анализируют параметры оборудования и выявляют отклонения до аварии. В транспорте и видеонаблюдении локальная обработка применяется для распознавания объектов и передачи в центр только значимых событий. В сельском хозяйстве шлюзы объединяют данные датчиков и сохраняют работоспособность при нестабильной связи.

Экономический эффект формируется за счет снижения трафика, объема хранения и сохранения автономной работы. Однако избыточное число узлов повышает стоимость сопровождения, поэтому внедрение начинают с процессов, где задержка или автономность имеют измеримое значение.

Заключение

Периферийные вычисления являются важным этапом развития компьютерных технологий и Интернета вещей. Они позволяют выполнять обработку ближе к источнику данных, сокращать задержку, снижать сетевую нагрузку и сохранять критические функции при нестабильной связи. Наибольший эффект достигается в гибридной архитектуре, где edge-узлы отвечают за оперативные решения, а облачная инфраструктура – за долговременную аналитику и централизованное управление.

Одновременно распределенная модель повышает требования к безопасности, обновлению и мониторингу оборудования. Надежная система должна строиться на классификации задач, управляемом жизненном цикле узлов, защите данных и постоянном контроле состояния. Такой подход позволяет использовать преимущества периферийных вычислений без неконтролируемого роста технических и организационных рисков.

Список литературы:

1. Iorga M. et al. Fog Computing Conceptual Model. NIST SP 500-325. Gaithersburg, 2018. DOI: 10.6028/NIST.SP.500-325.

2. ETSI GS MEC 003 V4.1.1. Multi-access Edge Computing (MEC); Framework and Reference Architecture. 2024.

3. Hong J. et al. Internet of Things (IoT) Edge Challenges and Functions. RFC 9556. 2024. DOI: 10.17487/RFC9556.

4. Fagan M. et al. Foundational Cybersecurity Activities for IoT Product Manufacturers. NIST IR 8259 Rev. 1. 2026. DOI: 10.6028/NIST.IR.8259r1