

УДК 004.896

МЕТОД СБОРА И ОБРАБОТКИ ДАННЫХ В СЕТЯХ ПРОМЫШЛЕННОГО ИНТЕРНЕТА ВЕЩЕЙ

Массеров Д.Д., Массеров Д.А.

ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет имени Н.П. Огарёва», Саранск, e-mail: resurs2003@bk.ru

Основной целью данной работы является повышение эффективности сбора и обработки информации в промышленном интернете вещей за счет использования модифицированного метода сбора и обработки данных, называемого Protobuf. В статье рассматривается сравнение различных методов сбора и обработки информации для установления совершенствования по сравнению с ними. Внедрение IIoT (промышленного интернета вещей) позволяет пользователям использовать данные для предиктивного анализа, сокращения времени простоя оборудования, централизованного хранения и удаленного мониторинга активности. С внедрением умного производства информация, полученная от устройств, должна быть обработана, что требует значительного времени и памяти для хранения накопленной информации. В связи с этим возникает задача минимизации времени обработки и размера файла для передачи по сети. В работе предложен метод сериализации и десериализации данных на основе метода Protobuf, который позволяет повысить эффективность обработки информации по временным показателям и уменьшить объем передаваемой информации. В результате работы показано увеличение скорости обработки информации, получаемой от IIoT оборудования за счет использования механизма сериализации и уменьшения размера файла информации, передаваемой по сети.

Ключевые слова: IIoT, интернет вещей, десериализация, сериализация, обработка информации, умный концентратор

DATA COLLECTION AND PROCESSING METHOD IN INDUSTRIAL INTERNET OF THINGS NETWORKS

Masserov D.D., Masserov D.A.

Ogarev Mordovia State University, Saransk, e-mail: resurs2003@bk.ru

The main purpose of this paper is to improve the efficiency of information collection and processing in the Industrial Internet of Things through the use of a modified method of data collection and processing, called Protobuf. The paper discusses the comparison of different methods of collecting and processing information to establish improvements over them. The implementation of the IIoT (Industrial Internet of Things) allows users to use data for predictive analysis, reduced equipment downtime, centralized storage and remote activity monitoring. With the introduction of smart manufacturing, information received from devices must be processed, which requires significant time and memory to store the accumulated information. This raises the challenge of minimizing the processing time and file size for transmission over the network. In this paper, we propose a method for serialization and deserialization of data based on the Protobuf method, which can improve the efficiency of information processing in terms of time and reduce the volume of transmitted information. As a result of the work it is shown to increase the speed of processing the information received from IIoT equipment by using the mechanism of serialization and reducing the file size of the information transmitted over the network.

Keywords: IIoT, Internet of things, deserialization, serialization, information processing, smart hub

Существует три основных метода [1] сбора и обработки данных, которые были проанализированы: метод сбора и обработки информации с помощью шлюза, метод сбора и обработки информации с помощью умного концентратора, метод сбора и обработки информации на базе промышленных шин. Метод сбора и обработки информации с использованием умного концентратора имеет преимущество перед другими методами, так как помогает модернизировать и достичь высокого уровня производства в обрабатывающей промышленности.

Устройства и датчики генерируют и собирают огромное количество данных, общий объем собранных данных может быть настолько велик, что передать их по сети для дальнейшего анализа может быть невозможно. Датчики могут передавать эту информацию каждые 30 с, и таких устройств

может быть несколько сотен. Это может быть лишь один из десятков типов датчиков, имеющих различные типы данных, которые вместе представляют собой огромный неоднородный набор информации из многих источников. Это приводит к высоким затратам на сбор данных и интеграцию оборудования с системой обработки данных. Также большой проблемой является недостаток вычислительной мощности и ресурсов хранения для выполнения сложных задач анализа и машинного обучения [2].

Кроме того, большие объемы данных трудно обрабатывать, поскольку требуется больше времени для ответа. Таким образом, передача важных данных через интернет для проведения важного анализа часто становится проблемой [3].

Метод сбора и обработки информации с помощью умного концентратора имеет

преимущество перед другими методами, так как помогает в модернизации и достижении высокого уровня производства в обрабатывающей промышленности. Это достигается благодаря тому, что основным элементом данного метода является умный концентратор, который предоставляет удобные решения для масштабирования системы с помощью новых устройств, осуществляет сбор и форматирование данных с использованием процессов сериализации, что уменьшает объем информации, которую необходимо передавать на облачную платформу для дальнейшего анализа.

Материалы и методы исследования

1. Метод сбора и обработки информации с помощью шлюза

Этот метод содержит три модуля: модуль умного объекта, модуль шлюза и модуль центра управления (сервера). Каждый модуль является многоуровневым (включая сенсорный, сетевой и прикладной уровни) и выполняет определенные функции для поддержки мониторинга взаимозависимой среды [4].

Модуль шлюза является мостом между умными объектами и центром управления. С другой стороны, в случае когда умные объекты не имеют доступного прямого соединения для прямой связи с центром управления посредством телекоммуникационных технологий, шлюзы будут соединять эти умные объекты с центром управления. Также модуль шлюза будет выполнять обязанности прикладного уровня при отсутствии прикладного уровня в модуле умного объекта.

II. Метод сбора и обработки информации с помощью умного концентратора

Метод сбора и обработки информации, позволяющий использовать информационную и операционную инфраструктуру для улучшения управления, мониторинга и контроля существующих и новых устройств. Он состоит из трех основных компонентов: слой устройств, умный концентратор, облачная платформа микросервисов. Представленный метод также может быть реализован на существующих умных предприятиях для оптимизации управления умным оборудованием и повышения эффективности производства [5].

Умный концентратор может выступать в качестве шлюза для существующих устройств IoT. Функционально концентратор предлагает простой и удобный способ предварительной обработки данных, обмена данными и связи между существующими устройствами IoT и серверной платформой, что означает, что использование умного концентратора может обновить или улучшить каналы обмена данными и связи.

III. Метод сбора и обработки информации на основе промышленных шин

Этот метод осуществляется на четырех уровнях: уровень устройств, датчиков и механизмов, уровень поставщика данных, уровень кроссплатформенного программного обеспечения и уровень приложений [6].

Уровень поставщика данных предназначен для приема данных от промышленных шин и передачи их на уровень кроссплатформенного программного обеспечения, которое содержит программный коммуникационный модуль для каждой промышленной шины.

Назначение этого модуля – реализация цикла сбора информации. Интерфейс для приема и передачи данных используется поставщиком данных, отдельным программным модулем, для получения данных из промышленных сетей и хранения их в буферном кэше. Эта память используется для более быстрого реагирования на запросы с уровня промежуточного ПО.

Результаты исследования и их обсуждение

Исходя из выбранного прототипа, модифицированный метод сбора и обработки информации выглядит следующим образом (рис. 1). Основой сбора данных являются датчики, механизмы, которые разбросаны в определенной географической зоне. Они обмениваются данными по сети для автономного сбора данных и передачи их на специальный узел, который считается промежуточной точкой для сбора информации.

Для хранения и передачи информации, полученной от устройств, используется механизм сериализации, который позволяет уменьшить размер передаваемых файлов, что в свою очередь снижает нагрузку на сеть и сокращает время, необходимое для передачи информации для дальнейшего анализа.

Информация, полученная от датчиков, механизмов, устройств, передается в умный концентратор, используя протокол интернета вещей CoAP или MQTT.

Проанализированная на микросервисной платформе IoT, информация может быть передана в другие информационные или операционные системы или сохранена в базе данных.

1. Сбор данных

В зависимости от требований к сбору информации передача собранных данных может осуществляться периодически или по событию. Умный концентратор – это узел с двумя и более сетевыми интерфейсами, он собирает и выполняет промежуточную обработку данных, полученных от устройств сбора информации.



Рис. 1. Модифицированный метод



Рис. 2. Общий процесс сбора информации

Датчики и механизмы – это небольшие электронные устройства, способные измерять физические величины (например, температуру, свет, давление) и передавать ее на блок обработки информации. Учитывая достижения в области микроэлектроники, технологии и программное обеспечение для беспроводной передачи

данных позволяют производить микро-сенсоры объемом в несколько кубических миллиметров [6].

Общий процесс сбора информации и передачи ее на умный концентратор показан на рис. 2. Устройства могут иметь свои собственные методы связи, такие как Wi-Fi, BLE, ZigBee и Z-Wave.

2. Данные о передаче технологий в HUB

Сенсорные сети – это пространственно распределенные датчики, которые отслеживают физические или экологические условия, такие как температура, звук, давление и т.д., и совместно передают свои данные по сети. Такие сети могут использоваться в большом количестве различных приложений: промышленная автоматизация, системы контроля микроклимата, системы безопасности и сигнализации, учет и оптимизация энергопотребления и т.д. [7]. Зона покрытия таких сетей может составлять от нескольких метров до нескольких километров. Одними из основных стандартов для реализации таких сетей являются Wi-Fi, BLE, ZigBee, Z-Wave.

3. Протоколы для информационных данных

В модифицированном способе протокол CoAP предназначен для передачи между устройствами в одной ограниченной сети, например, с низким энергопотреблением, в сетях с потерями, а также между устройствами в разных сетях, подключенных к интернету.

CoAP может быть интегрирован с такими форматами данных, как XML, JSON, Protobuf, для эффективного взаимодействия с другими платформами.

В модифицированном методе протокол MQTT обеспечивает минимальные требова-

ния к ресурсам и используется для передачи информации. Протокол не накладывает ограничений на формат данных.

4. Сериализация

На рис. 3 показан общий процесс сериализации-десериализации.

В предлагаемом методе создания распределенного сервиса, части которого должны обмениваться информацией со сложной структурой, в данном случае для передаваемых данных создается код, выполняющий процессы сериализации и десериализации.

Для объекта, заполненного необходимыми данными, вызывается созданный код сериализации, в результате на выходе получаем, например, XML file.

Полученная последовательность битов записывается в базу данных, память или file, который затем отправляется принимающей стороне. Для десериализации служба-получатель создает объект того же типа и вызывает необходимый код, в результате чего получается объект с теми же данными, что и объект службы-отправителя.

В таблице представлено описание основных характеристик методов сериализации с использованием XML, JSON, Protobuf.

Для модифицированного метода сбора и обработки информации предлагается использовать метод сериализации Protobuf.



Рис. 3. Общий процесс сериализации-десериализации

Описание основных параметров различных методов сериализации

Способ	JSON	XML	ProtoBuf
Стандартизация	+	+	+
Спецификация	STD 90/RFC 8259	Рекомендуется W3C: 1.0 and 1.1	Спецификация языка буферов протокола версии 3
Двоичный	-	Частично	+
Доступен для понимания человеком	+	+	-
Поддержка ссылок	+	+	-
Стандартные API	JSONQuery, JSON-Path, JSON-LD	DOM, SAX, XQuery, XPath	Java, C++, C#, Python

По сравнению с Protobuf, JSON и XML передают детали метаданных, что добавляет полезную нагрузку к полезной нагрузке. Использование Protobuf для сериализации и десериализации будет потреблять меньше процессорного времени и памяти, поэтому время обработки будет быстрее по сравнению с JSON и XML. Protobuf сжимает данные и генерирует плотные данные. Если сравнивать с XML, то он составляет почти 1/3 размера, а если сравнивать с JSON, то 1/2 [8].

JSON и XML читаемы человеком и небезопасны для передачи данных по сети. Если необходимо, чтобы ответ не был прочитан пользователем, то необходимо использовать Protobuf. Пользователю также необходим proto file для десериализации потока объектов.

5. Микросервисная платформа

В модифицированном методе микросервисная платформа структурирует приложение как набор сервисов. Микросервисы обычно основаны на бизнес-функциях [9].

Развертывание с использованием контейнеров обеспечивает переносимость. Контейнеризация сокращает время развертывания, поскольку требует только включения нужного контейнера, не затрагивая другие контейнеры, запущенные на том же хосте [10].

Платформа микросервиса может передавать полезную статистику или информацию другим информационным или операционным системам, таким как система управления производством, система автоматического хранения и поиска, система управления активами предприятия с помощью Protobuf по протоколу CoAP или MQTT.

Заключение

В данной работе проведен анализ существующих методов сбора и обработки информации в сети IoT, на основе которого был выбран прототип метода сбора и обработки информации.

Методика сбора и обработки информации была модифицирована, отличием ее является использование сериализации на основе метода Protobuf, что позволяет повысить эффективность обработки информации во времени и уменьшить объем передаваемой информации.

Список литературы

1. Ven. R. Three Industrial IoT Implementation Challenges. 2018. URL: <https://dzone.com/articles/3-iiot-industrial-internet-of-things-implementation> (дата обращения: 08.02.2023).
2. Массеров Д.А., Массеров Д.Д. Анализ, проблемы и возможности использования больших данных в городском хозяйстве // Научное обозрение. Технические науки. 2022. № 6. С. 27–31.
3. Касперская Н.И. Анализ больших данных в ИБ предприятий. Перспективы развития // Защита информации. Инсайт. 2019. № 3 (87). С. 34–43.
4. Khan W.Z., Aalsalem M.Y., Khan M.K., Hossain M.S., Atiqzaman M. A reliable Internet of Things based architecture for oil and gas industry. In: 2017 19th International Conference on Advanced Communication Technology (ICACT). 2017. P. 705–710. DOI: 10.23919/ICACT.2017.7890184.
5. Lee K.M., Zhang S.Z., Ng K.H. Development of an industrial Internet of things suite for smart factory towards reindustrialization. Adv. Manuf. 2017. No. 5 (4). P. 335–343. DOI: 10.1007/S40436-017-0197-2.
6. Ungurean I., Gaitan N., Gaitan V. A middleware based architecture for the industrial internet of things. KSII Trans. Internet Inf. Syst. 2016. No. 10 (7). P. 2874–2891. DOI: 10.3837/TIS.2016.07.001.
7. Al Hadidi M., Al-Azzeh J.S., Tkalich O.P., Odarchenko R.S., Gnatyuk S.O., Khokhlova Y.Y. Zigbee, bluetooth and Wi-Fi complex wireless networks performance increasing. Int. J. Commun. Antenna Propag. 2017. No. 1 (48). P. 1–48. DOI: 10.15866/IJRECAP.V7I1.10911
8. Shinde Y. Protobuf Performance Comparison. 2016. [Электронный ресурс]. URL: <https://dzone.com/articles/protobuf-performance-comparison-and-points-to-make> (дата обращения: 08.02.2023).
9. Жукова Н.А., Тристанов А.Б., Тин Т., Аунг М. О проблеме сбора данных в сетях интернета вещей с динамической структурой // Известия КГТУ. 2021. № 61. С. 105–118.
10. Babaria U. IoT Development Needs Microservices and Containerization. 2018. [Электронный ресурс]. URL: <https://www.einfochips.com/blog/why-iiot-development-needs-microservices-and-containerization> (дата обращения: 08.02.2023).