

УНИВЕРСИТЕТ ЗА НАЦИОНАЛНО И СВЕТОВНО СТОПАНСТВО

ФАКУЛТЕТ "Приложна информатика и статистика"

КАТЕДРА "Информационни технологии и комуникации"

Автореферат

на дисертационен труд за присъждане на образователна и научна степен "доктор" на
тема:

Методи за пренос и обработка на данни от Интернет на обектите
в платформа за големи данни

докторант: Николай Ангелов Къскатийски

научен ръководител: доц. д-р Любен Боянов

Научно направление:

"Приложение на изчислителната техника в икономиката"

София, 2021 г.

Дисертационният труд е обсъден и допуснат до публична защита на редовно заседание на катедрен съвет на катедра „Информационни технологии и комуникации“ на Университета за национално и световно стопанство, включващо членове на катедрата, състояло се на 12 март 2021 г. Авторът на дисертационния труд е задочен докторант в катедрата.

Публичната защита на дисертацията е насрочена за 01 юни 2021 г., от 10 часа в зала „Тържествена“ на УНСС, централно фоайе (№ 2032А), пред научно жури в състав:

Външни членове:

1. проф. д-р Красен Стефанов;
2. проф. д-р Красимир Шишманов;
3. доц. д-р Веселин Попов.

Вътрешни членове:

1. проф. д-р Камелия Стефанова;
2. доц. д-р Любен Боянов.

Материалите по защитата са публикувани на интернет-страницата на Университета за национално и световно стопанство и са на разположение на интересуващите се в сектор „Научни съвети и конкурси“ и на интернет страницата на УНСС – София

Съдържание

1	ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД.....	4
1.1	Актуалност на изследваната проблематика	4
1.2	Обект и предмет на изследването. Ограничения	5
1.3	Цели и задачи на изследването	5
1.4	Работни хипотези на дисертационния труд.....	6
1.5	Методология на работата	7
1.6	Приноси на труда	7
1.7	Практическо значение на труда.....	8
1.8	Обем и структура на дисертацията.....	9
2	КРАТКО ИЗЛОЖЕНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД	12
2.1	Въведение	12
2.2	Глава I.....	13
2.3	Глава II	21
2.4	Глава III.....	29
2.5	Глава IV.....	33
2.6	Заключения	45
3	СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД	47

1 ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

1.1 Актуалност на изследваната проблематика

През последните две десетилетия светът започна да генерира, пренася, обработва и съхранява огромни количества данни, като след всяка година тези данни стават все повече и повече. Хората и бизнеса може да съхраняват и анализират всяко парченце информация, което би им дало предимство в създаването на нови и по-ефективни подходи, модели и парадигми, които да подобрят както икономиката, така и условията за живот. Източниците на данни са на практика навсякъде около нас, независимо каква област на човешката дейност разглеждаме, потокът от данни се увеличава с всеки изминал момент. В близкото минало се събираха данни, тясно свързани с процесите, които хората пряко или чрез устройства наблюдават (като температура, влажност, количество изхарчени пари, количество произведени и продадени стоки и т.н.). Днес към тях се добавят и нови категории и видове данни - например от Интернет сайтове, от финансови транзакции, от дейности, свързани със сайтове, от мобилни телефони, както и всякаква друга разнородна информация от новият феномен - Интернет на Обектите (Ино - Internet of Things - IoT). Най-общо той представлява свързване на всеки обект или процес, който може да се дигитализира, към Интернет.

Ино като парадигма за предаване на данни между машини без намесата на човек, е тясно свързано с всички икономически области - от сферата на производството до сферата на услугите. Характерно за Ино е, че дори и количеството на данни генерирано от един обект да е сравнително малко, то значителното количество на обекти прави общия обем генерирани данни огромен. Тези специфични за сектора на информационните технологии (ИТ) дадености се появяват и налагат през последните години, поставяйки нови предизвикателства пред инженери и специалисти от областта на ИТ, а именно - да оптимизират процесите за пренос и обработка на големи количества данни. Специалисти и ИТ средства вече имат много важната задача да се адаптират към нарастващия поток от данни обменян между машините и да обръщат внимание на новите трудности, свързани с този поток. От тази гледна точка, развитието на протоколите за предаване на информация

от и в Ино, ефективността на тези протоколи, както и особеностите на системите за събиране, обработка и анализ на големи данни стават все по-важни. Настоящото развитие на ИТ сферата сочи, че това са теми, които няма да загубят своята актуалност в предстоящите десетилетия. Всичко това ще изисква разработчици и фирми да се запознаят, обучат и да могат да ползват тези системи, протоколи и платформи, като заедно с това извличат бизнес ползи от тях, непознати до момента.

1.2 Обект и предмет на изследването. Ограничения

Обект на настоящия труд са методите за комуникация на данни от Интернет на обектите към системите за обработка и съхранение на данни в частност на големи данни, при което възниква нужда от буфериране, мултиплексиране, интеграция и оперативна съвместимост.

Предмет на дисертационния труд са подходите за предаване и приемане на данни от Интернет на Обектите в системи за обработка на големи данни, тяхната обработка и интегриране, както и методологията за избор на комуникационни протоколи за данни от Ино.

Изследователският въпрос, който е поставен в настоящия труд, е „По какъв начин да се оценяват и използват различните протоколи за обмен на данни от Интернет на обектите и какви са методологиите за реализация на по-висока ефективност?“.

Ограничения на изследването, изследването на обекта се ограничава до преносът на данни генерирани от Интернет на обектите и тяхното съхранение в системи за работа с големи данни. Разглеждат се протоколи за пренос на данни използвани при Ино, като същите са сведени до най-използваните в света, според научната литература и мненията на експерти по този въпрос.

1.3 Цели и задачи на изследването

Основната цел на настоящата работа е да изследва приложимостта на протоколите в Ино от гледна точка на комуникацията между обектите и системата за обработка на големи данни, както и особеностите при приемане и интегриране на данни от Ино. Разглеждат се

подходи за осигуряване на ефективност в определена архитектура за обработка на данни от Ино от гл. т. на най-добро взаимодействие на нейните блокове по време на обработка на големи данни.

Задачите, формулирани при реализацията на целта на дисертацията:

- Изследване и анализ на съвременните протоколи за интегриране на данни от Ино за последваща обработка
 - Обзор и анализ на актуалните проблеми при събирането на данни от Ино породени от тяхната специфика.
- Изследване на софтуерните инструменти/платформи използвани за интегриране на данни в хранилища за големи данни, получени по различни комуникационни протоколи
 - Анализ на възможните предимства и недостатъци на различните протоколи при различни сценарии;
 - Разучаване работата и компонентите на функционираща система за големи данни.
- Дефиниране на критерии за избор на приложимостта на протоколи в Ино
 - Дефиниране на подходи за количествена оценка при избор на протокол за пренос на данни от Ино до етапа на тяхното съхранение;
 - Избиране на критерии за оценка на протокол за пренос на данни при използване на Ино;
 - Формулиране на метод за оценка на приложимост на определени протоколи за пренос на данни.

1.4 Работни хипотези на дисертационния труд

Основните работни хипотези, представени в настоящата работа са:

Хипотеза 1:

Възможно е да се посочат и класифицират комуникационни протоколи, които са подходящи и съответно - приложими за работа на Ино при различни условия.

Хипотеза 2:

Възможно е да се дефинира методология за използването на данни, приемани от Ино по различни протоколи, като при прилагането на тази методология се намират условията за по-добра ефективност при работата с големи данни.

Хипотеза 3:

Ефективната работа с данни в дадена референтна архитектура за Интернет на обектите изисква определени архитектурни и интеграционни решения.

1.5 Методология на работата

При работата върху дисертационния труд са използвани анализ и синтез, индукция и дедукция, моделиране, както и сравнителен подход. Използвани са способности от методът на аналогията и максималното правдоподобие и подходът на най-малките квадрати.

Проведени са практически експерименти с цел потвърждаване на следствията и закономерностите открити по време на разработката

1.6 Приноси на труда

Представеният дисертационен труд съдържа следните научни приноси:

- Разгледани, анализирани и са класифицирани основните характеристики на протоколи за Ино - MQTT, CoAP, AMQP, DDS и HTTP;
- Направен е анализ и е изведена характеристика с количествени показатели на комуникационни протоколи;
- Представени са критерии за избор на протоколи и на тази база е предложена количествена оценка за пригодност на всеки протокол с критериите.

Представени са следните научно-приложни приноси:

- Създадена е тестова среда за симулиране и оценка на реални сценарии за пренос на данни от комуникационни протоколи и е извършена верификация;

- Направено е изследване с пет от най-известните и използвани протоколи за пренос на данни – MQTT, CoAP, AMQP, DDS и HTTP, като при опитите за всеки от тях са използвани структурирани, полуструктурирани и неструктурирани данни;
- Протоколите са тествани за скорост на трафика на пренасяните данни и консумирана енергия от крайните ИО устройства;
- В системата за обработка на големи данни са предложени блокове в модула за приемане и разпределяне на данните - NiFi, които блокове извършват оптимизация на потока приемани данни;
- Инсталирани са и са конфигурирани редица пакети, които помагат за оптимизация на приеманите данни;
- Изследвани са и са представени резултатите от обмяната на съобщения и тяхната обработка в система за работа с големи данни;
- Намерена е зависимост между броят на получени съобщения и времето за запис в HDFS;
- Направени са изследвания в ефективността на работа на приеманите съобщения при работа с различни пакети за приемане на съобщенията (NiFi и Kafka).
- На базата на направените опити е посочено тясното място при приемането на данни от ИО в система за обработка за големи данни;
- Предложена е оптимизация на потоците на приемане и обработка на данни с буфер, който значително намалява времето за запис;

1.7 Практическо значение на труда

Днес съществуват множество протоколи за пренос на данни използвани в Интернет на обектите възникнали за решаването на един или друг проблем. Това многообразие може да се яви като пречка към изследователя, който не е запознат с отделните детайли и характеристики на протоколите за пренос на данни. Така е възможно да има затруднение в избора на протокол при проектирането на система, която включва обекти (сензори

актуатори) и системи за съхранение на големи данни. Възникват въпроси за критерии за избор на протокол пред други протоколи и с какво такъв избор ще допринесе на системата и какво евентуално ще се изгуби в конкретната реализация. В дисертационния труд е обърнато внимание на няколко широко разпространени Ино протокола - MQTT, CoAP, AMQP, DDS и за пълнота HTTP. С оглед релевантно сравнение между протоколите за пренос на данни е предложена методология базирана на точковите оценки с цел формулиране на количествена оценка на отделните протоколи спрямо предложени критерии.

Класическите архитектури използвани в платформите за големи данни не са оптимизирани да посрещнат нуждите за интеграция на данни от Ино, като възможен сценарий за подобряване на скоростта, с която данните се записват в хранилището е добавянето на буфер между брокера, който посреща данните в системата и последващото записване на данните в хранилището.

1.8 Обем и структура на дисертацията

Дисертационният труд се състои от въведение, изложение представено в четири глави, заключение, списък на използваните литературни източници и приложение. Той обхваща 165 страници, от които 146 страници изложение, 6 страници списък на информационните източници и 13 страници приложения. Изложението съдържа 22 таблици и 81 фигури. Информационните източници включват 102 литературни източника, от които 3 на кирилица и 99 на латиница.

Настоящият труд е оформен по следния начин:

- Във въведението са представени обекта и предмета на дисертационния труд, както и работните хипотези и задачите поставени пред него;
- В глава 1 е направен обзор на характеристиките на големите данни, както и подходи за приемане на данни от различни източници, в частност от интернет на обектите, в системи за обработка на големи данни. Направен е обзор на някои от широко разпространените платформи за работа с големи данни, изведени са сравнителни

характеристики. Изведени са основни понятия и блок схеми на Ино, както и референтни архитектури;

- В глава 2 е направен обзор на често използвани модели за комуникация, характерни също и за Ино. Обзор на протоколи за пренос на данни както и мрежови протоколи, широко застъпени в Ино. Представени са платформи за работа с Ино, както и основни похвати използвани при статистически метод за точковите оценки;
- Глава 3 е посветена на методиката за дефиниране на критерии, значими за избор на протокол за данни при системи с участието на Ино. На база похвати заимствани от статистическите методи на точковите оценки се предлага оценка на критериите за приложимост към отделните протоколи за пренос на данни;
- Глава 4 е посветена на практическото изследване и верификация на избрани протоколи за пренос на данни и тяхната приложимост според избрани критерии. Акцентирано е над проблем, свързан със записването множество съобщения в хранилището на големи данни, открит по време на експеримента и начините за намаляване на влиянието му, чрез добавяне на буфер за агрегиране на съобщения;
- В последната част на дисертационния труд е направено обобщение на постигнатите резултати.

Структурата на изложението включва:

1 Въведение

1.1 Обект и предмет на дисертационния труд

1.2 Работни хипотези на дисертационния труд

1.3 Цел и задачи на дисертационния труд

1.4 Структура на дисертационния труд

Глава 1. Видове данни и големи данни

1.1 Големи данни – особености и характеристики

1.2 Структурирани, полуструктурирани и неструктурирани данни

1.3 Подходи за обработка на големи данни

1.4 Данни в реално време

1.5 Ламбда архитектура

1.6 Капа архитектура

1.7 Интернет на обектите

1.8 Приложения на Ино

1.9 Архитектура на Ино и референтни архитектури

1.10 Изводи

Глава 2 Протоколи и платформи за работа с Ино

2.1 Модели на обмен на информация при Ино

2.2 Типове Ино протоколи

2.3 Платформи за работа с Ино

2.4 Статистическо оценяване

2.5 Изводи

Глава 3 Приложимост на протоколи за пренос на данни към критерии

3.1 Критерии към протокол

3.2 Методика за избор на протокол

3.3 Изводи

Глава 4 Изследване на протоколи за комуникация на данни и интегрирането им в платформа за големи данни

4.1 Избор на протокол за пренос на данни

4.2 Пренос на данни

4.3 Запис в система за големи данни

4.4 Изводи

Заклучения

1. Научни приноси

2. Научно-приложни приноси

3. Публикации

4. Бъдеща работа

ЛИТЕРАТУРА:

Приложение А

Използван хардуер

Сумарен трафик през рутер

2 КРАТКО ИЗЛОЖЕНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

2.1 Въведение

Във въведението към дисертационния труд се съдържа изложение на предмета на изследването, неговото теоретично, практическо и социално значение. Посочени са целите и обхвата на анализа, някои ограничения, както и използваните научни методи. Идентифицирани са задачите на изследването и са изброени проблемните въпроси, които са предмет на изследване за доказване на поставената научна теза.

2.2 Глава I

Отделено е внимание на подходите за приемане на данните от Ино в системите за големи данни, като са описани основни похвати и процеси и са дадени примери как са подхождали някои от големите компании на пазара за такива системи.

Широкото определение на Интернет на Обектите обхваща почти всичко, което комуникира чрез световната мрежа. Представена е базова обобщена схема на основните компоненти на Ино, както и на някои от съществуващите индустриални стандарти използвани в Ино.

Отделните данни, генерирани от Интернет на Обектите не са големи по размер, но поради огромното количество от обекти, които генерират тези данни, както и честотата на генериране, Ино данните се отнасят към големи данни. Няма наложени ограничения над вида данни, които се генерират от Ино. В зависимост от величината, която се следи, данните могат да бъдат структурирани, полуструктурирани или неструктурирани. Разгледани са характеристиките на големите данни и е направен обзор на платформите за работа с големи данни.

Големи данни (Big Data) е термин, който е свързан с използването на големи набори (множества) от данни, които са с твърде голям обем/размер, бързо постъпващи и сложни, за да бъдат обработвани с традиционните и утвърдени системи за управление на бази данни (СУБД), както и с широко използваните програмни средства в областта на ИТ. Работата с големи данни включва набиране (придобиване, получаване), съхранение, търсене, споделяне, анализ и визуализация и всичко това не е подходящо да се прави със системи, непригодни за големи данни. Тенденцията към увеличаване на обема на данни произтича и от допълнителната информация, която може да се извлече от анализа на простите съвкупности от данни. Самата концепция и термин „големи данни“ може да се разглежда като разширена технология за бизнес-интелигентност с цел съхранение, обработка и анализ на данни, които преди това не е било възможно да се ползват и обработват поради ограниченията на действащите по това време информационни технологии. Някои от тези

данни може да са източник на непознати зависимости, а други може да доведат до откриване на непознати до момента грешки или новости/открития (вече има такива в медицината). Предизвикателствата пред големите данни включват тяхното регистриране, съхранение, анализ, търсене в тях, споделяне, трансфер, визуализация, правене на запитване/заявка в тях, обновяване, сигурност и видове източници за работа с тях.

Големите данни (а и данните по принцип), според начина на представяне могат да се класифицират в три категории - структурирани, полуструктурирани и неструктурирани.

Структурираните данни са тези, които са с определена дължина и формат. Примери за структурирани данни са числа, дати, групи думи и т.н. На този етап от развитието, това са данните, с които най-често хората работят, защото те са организирани в структури (като например СУБД), от където могат да се извличат, обработват и т.н. Структурираните данни се организират лесно в табличен вид. Източници на структурирани данни може да са както машини, така и хора, които работят с компютри. Машините могат да генерират структурирани данни от сензори, RFID, NFC, умни измервателни уреди, GPS, сървъри, компютърни приложения, апарати за продажба на стоки, финансови приложения и т.н. Хората генерират данни като имена, възраст, отговори на въпросници, кликове от уеб сайтове, игри и т.н.

Неструктурираните данни са онези, които нямат първоначално зададена или очаквана структура. Тяхната форма е нестандартна и данните са с различен формат и от различни източници. Примери за това са офис документи, ел. поща, блогове, цифрови изображения, видео, сателитни изображения, аудио/звукови файлове, някои сензорни данни, сканирани файлове и т.н. Както може да се види от списъка, много от данните, които се съхраняват на компютърните и други дигитални системи са неструктурирани. Към неструктурираните данни спадат и такива от социални мрежи като Twitter и Facebook.

Полуструктурираните данни са вид структурирани, но недефинирани данни, като например дефиниция на таблица в релационна СУБД. Примери за полуструктурирани данни са данни във формат XML или CSV файл, JSON, HTML и др.

В зависимост от вида на данните, начините за тяхната обработка и използване могат да бъдат различни и зависят от наличието на конкретни инструменти и техники за анализ.

Метаданните също могат да образуват големите данни. По същество те са данни за данните и могат да се добиват от анализа на обединени по подходящ начин множество несвързани източници. Тези данни могат да са сериозен източник на информация на по-високо ниво на абстракция.

Появата и разпространението на феномена големи данни доведе и до изисквания за нови технологии и архитектури за тяхната обработка и получаване на полза (добавена стойност) с прилагане на аналитични обработки. Поради много големия обем на данните, извършването на традиционна обработка и анализ става трудно или невъзможно със съществуващите до момента техники и средства. Досегашното структуриране на данните (най-често в бази данни с разполагане по редове и колони) е неподходящо за големи данни. Появява се необходимост от технологии, които позволяват мащабиране, обработка и анализ на огромни обеми от данни. Високопроизводителните изчисления (high-performance computing - HPC) имат инструменти, които не се справят добре с нарастващия обем и скорост, разнообразие на данните, като HPC се ползват предимно за специализирани задачи от областта на молекулярната динамика, ядрена физика, сложни симулации и други класове задачи, които са в класа високопаралелни задачи. Повечето от настоящите източници на големи данни не предполагат високопаралелни изчисления. Поради това се появяват нови паралелни и разпределени парадигми (като облачните изчисления), които са в състояние да обработват и съхраняват големи данни. Все по-често фирмите, а и отделни организации и хора съхраняват и обработват данните си в облак. Освен облаците се развиват и други подходи и инструменти - например Not Only Structured Query Language – NoSQL бази данни. Примери за инструменти и средства, за бази данни, които работят с големи данни са Cassandra, HBase, MongoDB и CouchDB.

В табличен вид са обобщени характеристиките на разгледаните пакети за работа с големи данни. Прави впечатление, че дори и пакетът да е с отворен код, то той не винаги е безплатен за комерсиална употреба.

Интеграцията на големи данни предоставя единна представа за данните за бизнес гъвкавост и вземане на решения и включва - откриване на данните, профилиране на данните, разбиране на данните, подобряване на данните, и трансформиране на данните, както е показано на Фигура 1. Процес на движение на данните от източника до системата за съхранение



Фигура 1. Процес на движение на данните от източника до системата за съхранение

Източник: собствено проучване

Интегрирането на данни е част от Архитектурния слой на големите данни, в който компонентите са отделени, за да могат да подпомогнат аналитичните възможности на системата. Този слой е свързан със съхранението на данни, което да позволи по-нататъшния им анализ, който може да се направи с помощта на различни инструменти, дизайнерски модели и други. Основните аспекти, по които се работи за подобряване качеството на данните могат да бъдат сведени до няколко основни парадигми, застанали зад интеграцията на данни а именно:

- Записите на данните трябва да са централизирани;
- Трябва да се осигури гъвкавост при селектирането на нужните данни;
- Намаляване на сложността на данните;
- Приоритизиране на данните;
- По-добро разбиране на информацията;
- Актуализиране на информацията.

Таблица 1 обобщава характеристиките на инструментите за интегриране на данни, и предоставя сравнение между горе изброените пакети за интеграция на данни.

Таблица 1. Сравнение на пакетите за интегриране на данни

Пакет	Скорост на данни	Размер на данните	Честота на данните	Формат на данни
Apache Flume	базирано на събития, не толкова на потоци от данни	в килобайти	буфериране когато данните не могат да се обработят в реално време	полуструктурирани данни предимно от дневници

Apache Nifi	може да обработва непрекъснати потоци от данни или случайни събития	максималния размер на съобщението е ограничен от наличната памет	буфериране когато данните не могат да се обработят в реално време	разнообразни, няма ограничение
Elastic Logstash	базирано на събития, не толкова на потоци от данни	максимален размер на съобщението стотици килобайти	буфериране когато данните не могат да се обработят в реално време	полуструктурирани данни предимно от дневници
Apache Gobblin	може да обработва непрекъснати потоци от данни или случайни събития	максималния размер на съобщението е ограничен от наличната памет	буфериране когато данните не могат да се обработят в реално време	разнообразни, няма ограничение

Преди няколко години беше много трудно да се анализират данни с размери от порядъка на петабайти или повече. Появата на Hadoop направи възможно изпълнението на аналитични запитвания върху огромно количество данни.

В днешно време има много платформи за обработка на данни. Някои поддържат поточно предаване на данни, а други поддържат реално поточно предаване на данни, което също се нарича данни в реално време.

Поточно предаване означава, че данните могат да се обработват в момента, в който те пристигнат, и след това да се обработват и анализират по време на интегрирането. При поточно предаване може да съществува известно забавяне на поточните данни от интегриращия слой.

Данните в реално време трябва да имат строги изисквания по отношение на времето. Така че обикновено се приема, че ако дадена платформа може да улови всяко събитие в рамките на 1 ms, тогава ние я наричаме данни в реално време или реално поточно предаване.

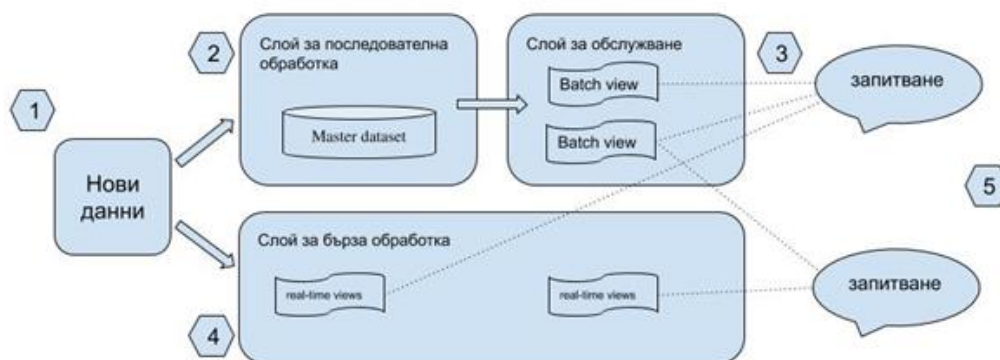
Когато се говори за вземане на бизнес решения, откриване на измами и анализ на дневници в реално време и предсказване на грешки в реално време, всички тези сценарии се отнасят за поточно предаване. Така че данните се получават незабавно, когато пристигат, наречени данни в реално време.

Apache Kafka може да интегрира данни от порядъка на милиони съобщения в секунда. Анализ на постоянни потоци от данни е възможен и от Apache Spark Streaming, Apache Flink, Apache Storm. Apache Spark Streaming е инструмент, който може да се разглежда като времеви прозорец (отрязък във времето), през който може да се разглежда опашка от

входящи съобщения и да се обработват като една партида, което може да се определи като обработка на реални потоци в микро бачове. За разлика от него, Apache Storm и Apache Flink могат да обработват данни в реално време.

Целта при поточната обработка и обработката в реално време е да се обработят всички постъпващи съобщения без загуба, но съществува вероятност, при която някои съобщения могат да бъдат пропуснати, което да доведе до грешки в системата. Разработени са няколко архитектури, които целят решение на този проблем.

Ламбда архитектурата (Lambda Architecture - LA) е термин за обща, разширяема и устойчива на грешки архитектура за обработка на данни. LA цели създаване на устойчива система, в която се избягват грешки (както хардуерни, така и от хора). Идеята е тази архитектура да обслужва големи натоварвания и различни случаи. С използването на тази архитектура се очаква да има ниско закъснение при четене и обновяване. Друг важен момент е, че системата трябва да е линейно скалируема. Потокът на данни в LA се разделя в две посоки – слой за пакетна обработка (batch layer) и слой за бърза обработка (speed layer) - т.е. обработка в реално време (Фигура 2. Ламбда архитектура)

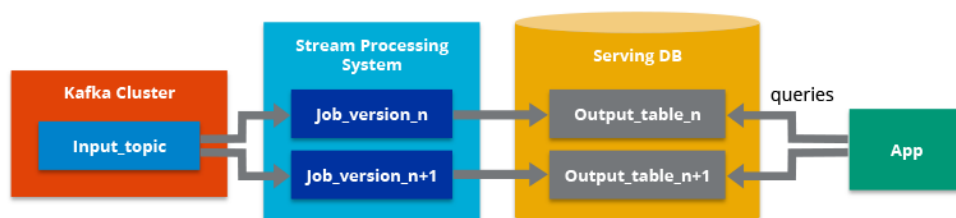


Фигура 2. Ламбда архитектура

Източник: собствено проучване

Капа е софтуерна архитектура, използвана за обработка на поточни данни. Основната предпоставка зад Капа архитектурата е, че може да извършва както обработка в реално

време, така и пакетна(групова) (batch) обработка, особено за анализи, с един технологичен стек. Тя се основава на стрийминг архитектура, при която входяща поредица от данни първо се съхранява в механизъм за съобщения като Apache Kafka. Оттам механизмът за обработка на потоци прочита данните и ги трансформира във формат, който може да се анализира, и след това ги съхранява в база данни за анализ, за да могат крайните потребители да правят заявки върху данните. Основната разлика с архитектурата Ламбда е, че всички данни се третираат така, сякаш са поток, така че механизмът за обработка на потоци действа като единствен механизъм за трансформация на данни - Фигура 3. Архитектурата Капа. Капа обикновено се изгражда около Apache Kafka заедно с високоскоростен механизъм за обработка на потоци

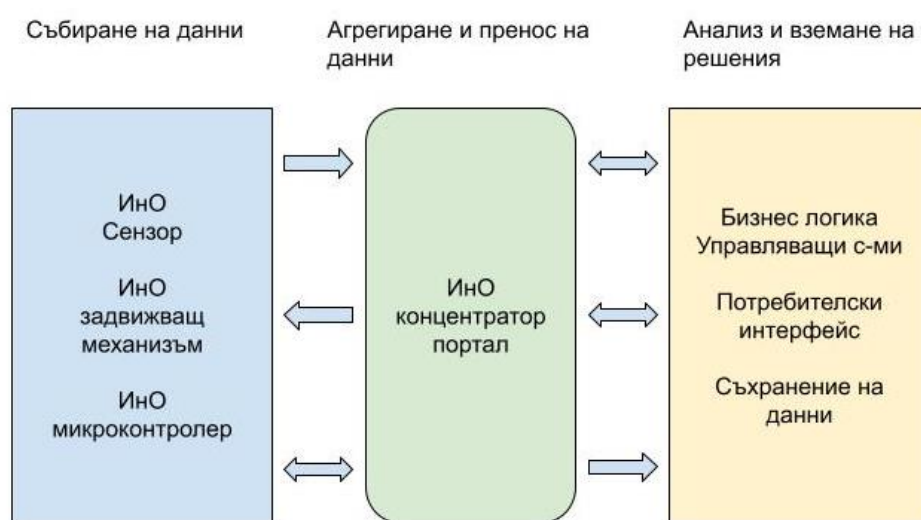


Фигура 3. Архитектурата Капа

Интернет на обектите (Internet of Things – IoT, или Интернет на нещата) е система от свързани електрически или механични обекти (включително машини, хора, животни с цифрови идентификатори), за които има възможност да се обменят данни по цифрови мрежи без участието на хора. Основната разлика между Ино и традиционния Интернет е, че докато при последния имаме човек от единия край на връзката (в началото или в края) то при Ино имаме друго устройство или машина. Когато се разглежда традиционния Интернет, в обмена на информация в крайните точки/терминали участва и човек (machine-to-human - M2H). При Ино обмена на данни и информация най-често става без човешки фактор (M2M или machine-to-machine). При традиционния Интернет обменът на информация се извършва с протоколи от стека TCP/IP. При Ино има свързване на разнородни физически обекти към мрежи и съответно – към Интернет, като може да се свързват и процеси с ограничения, наложени от тяхната специфика, поради което протоколите са много и различни. Свързаните обекти са уникално идентифицирани и могат както да предават данни за средата около тях

или за тяхното състояние, така и да приемат данни от отдалечени източници. Днес все повече обекти се свързват в цифрови жични или безжични мрежи, с което се предоставят възможност на хората (или устройства) да следят тяхното състояние. Примери за това са масово навлезлите умни часовници, устройствата в съвременните автомобили за следене на физически параметри по целия автомобил, както и наличието на хора или предмети около автомобила, идентификаторите по животни и т.н. и т.н. Възможностите за подобна идентификация на всеки обект и предаването на данни към произволно друго място обуславят огромните възможности на Ино, като средство за отчитане, следене, анализ и обратна връзка с всичко от реалната, заобикаляща ни среда.

Най-общо съдържанието на Ино е представено, заедно с обобщена на базата на редица други налични структури, на Фигура 4. Примерна блокова структура на Интернет на обектите



Фигура 4. Примерна блокова структура на Интернет на обектите

Източник: собствено проучване

Архитектурата на Ино не е еднозначно дефинирана, макар и по един или друг начин представените във Фигура 4 блокове и компоненти съществуват при повечето Ино конфигурации.

От гл.т. на функционалности, Ино архитектурите от край до край, може да се разгледат като:

- Свързаност и управление на обектите (устройствата);
- Интелигентна обработка и анализ на границата;
- Централизиран анализ и проява на интелект/възможности за самообучение;
- Приложна и бизнес интеграция;
- Сигурност и съответствие.

Съществуват различни видове Ино референтни архитектури, като най-общо може да се разгледат три основни вида – научни/стандартизационни, търговски и промишлени.

Редица големи компании за производство на софтуер и услуги също дефинират референтни архитектури за Ино, които да използват инструментите, които те поддържат. Това позволява на потребителите да намалят сроковете за внедряване, да избегнат наемането на ИТ персонал, като използват готови модулни решения. Големите компании осигурят техническата поддръжка и “ноухау” за изграждането на системи от свързани помежду си обекти, свързани в мрежа.

2.3 Глава II

Съществува многообразие от концепции за обмен на информация използвани от Ино. Широкото разпространение на някои от тях се дължи на характеристиките на Ино и протоколите за комуникация използвани при тях. Описани са някои от най-често използваните концепции за обмен на информация използвани при Интернет на Обектите. Направено е разграничение на протоколите използвани в Ино - най-общо на мрежови и протоколи за пренос на данни. Представени са най-често използваните и е направена обобщена сравнителна характеристика между тях в таблична вид.

Много от платформите за работа с големи данни могат да са част от система изградена за Ино. Разгледан е въпросът с мащабируемостта на платформите при работа с Ино, като критерий, който обуславя тяхната производителност. Направен е обзор на някои от най-често използваните платформи, в които участва или може да участва Apache Hadoop, както

са разгледани и техните основни характеристики. Причината за това е, че в опитите на настоящия труд е използван Apache Hadoop за съхранение на данните от Ино.

За да се направят статистически заключения и статистическо оценяване в дисертационния труд са заимствани похвати от точковите оценки представени в тази глава. Използвайки способности от методът на аналогията и максималното правдоподобие са оценени критериите за приложимост към отделните протоколи за пренос на данни. При намиране на минималния размер на буфер за съхранение на данни в системата за големи данни са използвани техники от подходът на най-малките квадрати.

След направените проучвания в съществуващата литература, най-общо може да се посочат четири модела на обмен на данни и информация между функционални компоненти в Ино. Първият е предаване/поставяне (Push), вторият е заявка - отговор (Request/Response), третият е абониране - информирание (Subscribe/Notify), а четвъртият – публикуване - абониране (Publish/Subscribe),

Широко разпространение при Ино (както и при работа с големи данни) през последните години има моделът Публикуване - абониране. С него обменът на информация може ефективно да редуцира излишните данни. Друга причина е, че значима част от крайните възли в Ино не притежават големи хардуерни възможности, оставайки брокера да намери получателя на съобщението, като разтоварва от подобно изискване другите Ино компоненти. Този модел е много удобен и за прекъснато подаване на данни, т.е. когато има значителни паузи във времето между отделните съобщения. Така може да се избегне често използваното синхронизиране при комуникационните протоколи - така нареченото ръкостискане (handshaking).

Съществуват класификации по различни признаци на протоколите използвани в Ино. Един от показателите може да е дали протокола е мрежови (network protocol) или за пренос на данни (data protocol).

- Мрежовите протоколи са проектирани и реализирани с цел да свързват (основно да извършват маршрутизацията) устройства в мрежи, като често това става в рамките на

Интернет. Примери за такива протоколи са LoRaWAN, Bluetooth, Zigbee и други. Някои от основните характеристики са обобщени в Таблица 2

Таблица 2 Характеристики на технологиите използвани за комуникация между Ино

Технология	Честота	Скорост	Обхват	Потребление на Енергия	Цена
2G/3G	Клетъчна мрежа	10 Mbps	Под 10 км	Високо	Висока
Bluetooth/BLE	2.4 ГХз	1,2,3 Mbps	90 метра	Ниско	Ниска
802.15.4	Под ГХз, 2.4 ГХз	40, 250 kbps	Повече от 150 км	Ниско	Ниска
LoRa	Под ГХз	<50 kbps	2-5 км	Ниско	Средна
LTE Cat 0/1	Клетъчна мрежа	1-10 Mbps	Под 10 км	Средно	Висока
NB-IoT	Клетъчна мрежа	0.1-1 Mbps	Под 10 км	Средно	Висока
SigFox	Под ГХз	<1 kbps	Под 10 км	Ниско	Средна
Weightless	Под ГХз	0.1-24 Mbps	Под 10 км	Ниско	Ниска
Wi-Fi	Под ГХз, 2.4 ГХз	0.1-54 Mbps	Под 90 метра	Средно	Ниска
WirelessHART	2.4 ГХз	250 kbps	Под 90 метра	Средно	Средна
ZigBee	2.4 ГХз	250 kbps	Под 90 метра	Ниско	Средна
Z-Wave	Под ГХз	40 kbps	Под 30 метра	Ниско	Средна

- Протоколите за пренос на данни, използвани в Ино, са предназначени за комуникация между устройства с ниска мощност и ограничени ресурси. Устройствата могат да

комуникират помежду си и без свързване чрез Интернет. Широко разпространени протоколи от този клас са MQTT, CoAP, AMQP, XMPP и др. От направените изследвания може да се направи обобщение показано в Таблица 3, като тези сравнения ще послужат за определяне на тежестите на отделни критерии в настоящото изследване.

Таблица 3. Сравнение на протоколи за пренос на данни

Протокол	Полезна информация	Потребление на енергия	Използвана честотна лента	Надеждност	Сигурност	Разпространение
MQTT	Високо	Средно	Високо	Много високо	Ниско	Много високо
AMQP	Средно	Високо	Средно	Високо	Много високо	Средно
CoAP	Много високо	Ниско	Много високо	Средно	Средно	Високо
HTTP	Ниско	Много високо	Ниско	Ниско	Високо	Ниско

Информацията, изпращана по мрежата, е във вид на пакети от данни. Ако два комуникационни възела (А и В) желаят да обменят информация, данните от предаващия възел А първо трябва да бъдат снабдени със служебна информация относно транспортирането им и съответно - пакетирани. Информацията се придвижва от А към В, като минава през различни системи и функционалности на отделните нива (слоеве). Приемащата част В премахва служебната информация, прибавена за целите на транспорта от изпращача.

По подобие на модела TCP/IP, за ИноО може да се изгради модел от използваните протоколи. Основните слоеве се дефинират по следния начин:

- Инфраструктурен (6LowPAN, IPv4/IPv6, RPL);
- Идентификационен (EPC, uCode, IPv6, URIs);
- Транспортен (WiFi, Bluetooth, LPWAN);
- Откриващ (Physical Web, mDNS, DNS-SD);
- пренос на данни (MQTT, CoAP, AMQP, Websocket, Node);
- Управление на устройства (TR-069, OMA-DM);
- Семантичен (JSON-LD, Web Thing Model).

В тялото на дисертационния труд са представени данни от изследване на едни от широко разпространените протоколи за пренос на данни за ИноО. Изследването е за избор на

ефективен протокол за ИО системи и разглежда MQTT, CoAP, AMQP и HTTP. Избрани са следните критерии за сравнение между отделните протоколи:

- Размер на полезната информация спрямо служебната в едно съобщение;
- Потребление на енергия спрямо необходими системни ресурси;
- Използвана честотна лента и закъснение;
- Надеждност спрямо гъвкавост;
- Сигурност срещу допълнителни услуги;
- Разпространение срещу стандартизация.

Днес съществуват множество платформи за работа с големи данни, които притежават различни характеристики. Изборът на подходяща платформа може да се обуславя от много фактори. Един важна функционалност на платформата е да се адаптира към количеството на данните, изискващи обработка и гъвкавостта на аналитичния апарат който се предлага. При изборът на подходяща платформа трябва да се вземат под внимание качества като: скорост на получаване на резултатите; големина на данните за обработка; как платформата отговаря при изменение на обема на обработваните данни; добавяне на допълнителни обработки на данните в бъдеще; колко критична е загубата на данни при възникване на проблеми до тяхното отстраняване.

Мащабируемостта на платформата е от голяма важност за справяне с нарастващите изисквания породени от увеличението на обема данни и повишените изисквания към методите за тяхната обработка. За да поддържат обработката на големи данни, различните платформи използват различни видове мащабиране. То може да се извършва в две посоки:

- Хоризонтално мащабиране;
- Вертикално мащабиране.

Вертикалното мащабиране позволява по-лесно добавяне на хардуер, но също така налага ограничения поради експоненциалното нарастване на цената на хардуера, свързано с пропорционалното нарастване на капацитета му.

Хоризонтално мащабираните платформи днес са придобили широко разпространение, не само заради ниските първоначални инвестиции, но и поради възможността по-лесно да се интегрират в облачните инфраструктури. Пионери в тази област са Google и Apache, следвани от редица големи компании. В разработката се използват и изследват софтуерни решения, които са хоризонтално мащабирани. В Таблица 4 са обобщени предимствата и недостатъците на разглежданите видове мащабиране.

Таблица 4. Сравнителна характеристика на хоризонталното и вертикално мащабиране.

Мащабиране	Предимства	Недостатъци
Хоризонтално мащабиране	Увеличение на производителността на малки стъпки когато е необходимо	Софтуерът трябва да се справи с разпределението на данните и комплексността породена от паралелната обработка
	Финансовите инвестиции са относително ниски	Ограничен брой софтуерни продукти се предлагат на пазара
	Системата може да се мащабира колкото е необходимо	
Вертикално мащабиране	Наличния софтуер може да се възползва от него без промяна	Значителни финансови инвестиции
	По-лесна поддръжка на хардуера в сравнение с хоризонталното	С оглед бъдещо развитие системата трябва да е по-мощна в началото, което води до нейното не пълна употреба
		Има технологичен лимит за мащабиране, който може да бъде достигнат по-бързо при вертикалното отколкото при хоризонталното

В Таблица 5 за изброени платформи подбрани на база на тяхното разпространение и са изложени на кратко техните предимства.

Таблица 5. Платформи за работа с ИнО

Платформа	Характеристики
Xplenty	Еластична и мащабируема облачна платформа; Осигурява незабавна свързаност с различни хранилища за данни и богат набор от готови компоненти за трансформация на данни; Богат език за сложни функции за подготовка на данни; Набор от API за допълнителни възможности за персонализиране на решенията.
Apache Hadoop	HDFS (Hadoop Distributed File System), която има способността да съхранява различни видове данни - видео, изображения, JSON, XML и обикновен текст в една и съща файлова система; Изключително полезен за научноизследователска и развойна дейност; Осигурява бърз достъп до данни; Силно мащабируем; Широкодостъпна услуга, базирана на клъстер от компютри.
Cloudera Distribution for Hadoop (CHD)	Широко разпространение; Cloudera Manager много добре администрира клъстера Hadoop; Лесно изпълнение; Опростено администриране; Висока сигурност и управление.
MongoDB	Лесна за научаване; Осигурява поддръжка за множество технологии и платформи; Опростена инсталация и поддръжка; Надеждна; Ниска цена.
Talend	Опростява ETL и ELT за големи данни; Висока скорост и добро мащабиране според нуждите; Доближава обработката на данни до почти реално време; Обработва множество източници на данни.
Tableau	Голяма гъвкавост за създаване на типа визуализации; Възможностите за смесване на данни от различни източници; Предлага готов набор от интелигентни функции; Висока скорост на обработка на данните; Готова поддръжка за връзка с повечето бази данни; Запитвания за данни без код; Готов за визуализация на мобилни устройства.
Microsoft Azure	Широк спектър от технологии с отворен код Apache, включително Hadoop и Spark; Собствен инструмент за анализ - HDInsight.
Google Cloud	BigQuery съхранява петабайта данни във формат върху, който лесно могат да се правят заявки; Cloud Dataflow анализира текущи потоци от данни и партии от исторически данни в паралел; Google Data Studio клиентите могат да превърнат различните данни в персонализирани графики.

За направа на статистически заключения и статистическо оценяване, дисертационния труд заимства похвати от точковите оценки.

Когато, на базата на характеристики на извадките, се правят оценки на неизвестни параметри на генералните съвкупности, резултатите са числа. Оценките може да бъдат средна аритметична, дисперсионна и др. Такива оценки се наричат точкови. Положението им на абсцисата ос при графично представяне се определя с точки.

Оценките имат вероятностен характер. Поради това, не може да се твърди, че дадена оценка напълно съвпада с оценявания параметър. Но винаги, когато са възможни различни оценки, съвършено логичен е стремежът да се намери най-добрата. Но това е само желание, тъй като няма еднозначен критерий и механизъм да се избере най-добрата оценка. Затова е необходимо да се знае на какви общи изисквания трябва да отговаря една точкова оценка, или какви свойства да притежава.

Намирането на точкови оценки, като се имат предвид изложените свойства, е възможно по различни методи.

Чрез методът на аналогията обикновено се намират оценки на параметри, които по характера си са моментни от някакъв порядък - средна аритметична, дисперсия и др. В основата на този метод стои принципът на аналогията: моментите в генералната съвкупност се оценяват чрез аналогичните моменти на извадката. Например за оценка на средната аритметична, дисперсията и др. служат средната аритметична, дисперсията и др. на извадката.

Тъй като много параметри, за които се търсят оценки, имат формата на моменти на разпределението, методът намира широко приложение.

Метод на максималното правдоподобие се основава на принципа, оценката на неизвестния параметър на генералната съвкупност да се намери по начин, при който е максимално правдоподобно предположението, че възпроизвежда параметъра на генералната съвкупност. За целта се търси максимум на специална функция, която се нарича функция на правдоподобие. Приложението на метода е свързано с конкретния вид на разпределението

на генералната съвкупност, т.е. трябва да е известна функцията на разпределението. Това ограничава в голяма степен възможностите за практическото му използване.

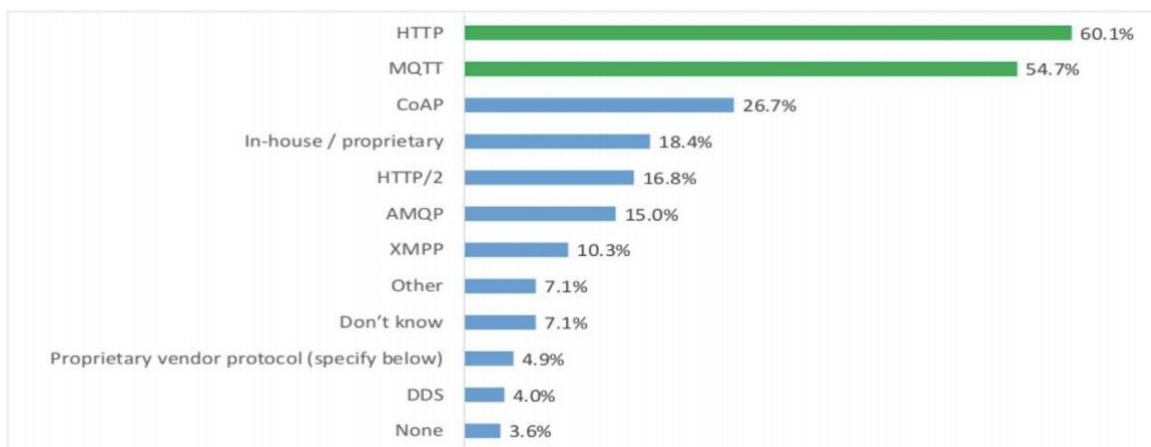
Методът на най-малките квадрати е един от най-широко прилаганите методи. Може успешно да се използва за получаване на неизместени и ефективни оценки, когато те са линейни функции на наблюдаваните стойности на случайната величина. Методът на най-малките квадрати също не е универсален, когато обаче са осигурени посочените условия (линейност на функцията и някои други), той е предпочитан, защото е много удобен. При нормално разпределени съвкупности методът на най-малките квадрати води до резултати, които се получават и по метода на максималното правдоподобие

При избора на метода имат значение и такива съображения, които не се включват в разглежданите свойства на оценките и в конструкцията на методите. Те са чисто практически и се отнасят до преценка на разходите на труд, средства и време. Съвсем не е без значение на каква "цена" може да се получи желаната оценка и какви потребности тя ще задоволява.

2.4 Глава III

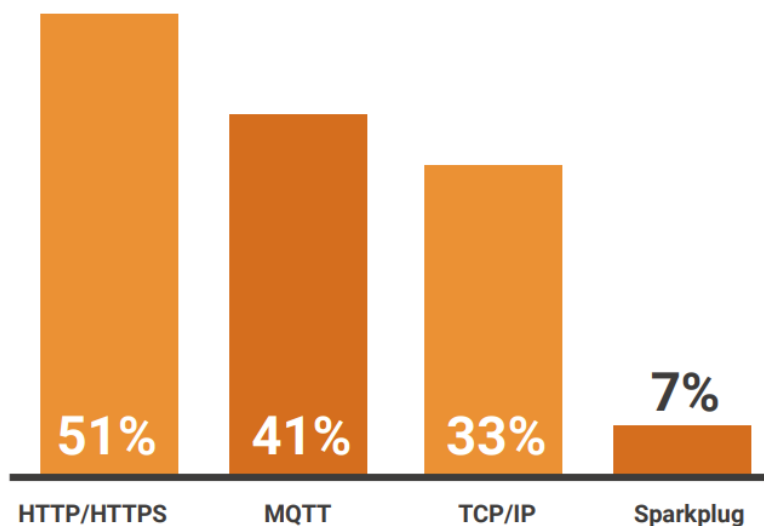
В настоящият труд се разглеждат протоколи за обмен на данни между устройства в Ино без намесата на човек. Данните могат да се обменят по много и различни съществуващи протоколи. Този факт може да затрудни избора на даден протокол или да внесе съмнения защо даден протокол е предпочетен пред друг. В литературата съществуват сравнения между отделни протоколи, но сравненията са направени откъм техническа гледна точка, при това без количествена характеристика. Това налага анализ на всеки протокол по отделно, с оглед неговото проучване и не подпомага вземането на обосновано решение как всеки отделен протокол би се вписал с система, където източници на данни са от Интернет на Обектите, а получател – система за обработка на големи данни. С цел улесняването на бъдещи разработчици и даже системни архитекти при избор на протокол за данни, тук се предлага метод с набор от критерии за оценка на протоколи за пренос на данни.

Eclipse Foundation прави анкети всяка година всред най-големите компании с разработки в тази област. Разликата в употребата на отделните протоколи в анкетата проведена през 2017 Фигура 5 за употребата на най-често срещаните протоколи е много сходна с тази за 2020 на Фигура 6.



Фигура 5. Процентно разпределение на протоколите през 2017

Източник: Eclipse Foundation



Фигура 6. Процентно разпределение на протоколите през 2020

Източник: Eclipse Foundation

От анкетата става ясно, че най-широко разпространени са HTTP и MQTT. Двата протокола са диаметрално противоположни както модели и характеристики. MQTT е базиран на модела публикувай-абонирай, а HTTP - на заявка-отговор. HTTP е широко разпространен в мрежата и работи с протоколи за маршрутизиране на по-ниски нива, докато MQTT има много малък овърхед и изисква по-малко системни ресурси

След запознаване с редица публикации и анкети във връзка комуникационни протоколи в Ино, се откриха следните критерии, които бяха определени като значими за това какъв протокол да се използва. Списъкът критерии не е изчерпателен и не включва всичките съществуващи такива, но е база, на която може да се стъпи при избор, какъвто може да се направи при изграждането на конкретна система. Тези критерии биха могли да бъдат ревизирани от бъдещи проектантите и допълнени спрямо техните конкретни нужди:

- Тип на Ино;
- Свързаност;
- Енергийни ограничения;
- Посока на предаване на данни;
- Мобилност;
- Тип на данните;
- Икономическа област;
- Количество свързани обекти;
- Мащабируемост;
- Среда;
- Скорост;
- Сигурност;
- Честота на включване;
- Изискване към предаваните данни.

При изборът на протокол за пренос на данни от Ино към платформа за обработка на големи данни се приемат следните допускания:

- за нуждите на приложението не е изгодно да се създава нов протокол, поради необходимостта за това време, верификация и затруднена поддръжка, както и поради наличието на многобройни съществуващи и работещи такива;
- за нуждите на приложението може да се подбере вече съществуващ протокол от най-широко разпространените, а именно: MQTT, CoAP, AMQP, DDS или HTTP.

При създаването на методика за оценяване приложимостта на даден протокол към конкретна реализация е необходимо да се използва подходящ метод за оценка. Избран е „методът на точковите оценки“, който ще позволи количествено измерване на дадена величина. Това ще даде възможност да се направи сравнение между отделните критерии и ще улесни сравнителния анализ. При тази методология се дефинират интервали от допустими стойности на оценката на дадена характеристика за съответния протокол и на база получената сума се съставя тегло (стойност) за приложимостта на протокола. За присвояване на оценки на протоколите към критерии са използвани:

- Данни от анкетата на Eclipse Foundation;
- Сравнения в научни публикации;
- Разговори с експерти в областта.

На база отделните критерии, идентифицирани по-горе, на всеки протокол се дава обобщена количествена оценка на базата на това до колко е пригоден за да посрещне избраните критерии. Приемаме следните теглови коефициенти:

- 9 - най-добро съвпадение;
- 7 - много добро съвпадение;
- 5 - добро съвпадение;
- 3 - има съвпадение;
- 1 - няма или много малко съвпадение.

Резултатът е представен в Таблица 6. Приложимост към критерий. При изборът на значими критерии и сумиране на теглата на отделните протоколи става ясно кой от тях би бил приложим в дадена ситуация.

Таблица 6. Приложимост към критерий

Критерий	MQTT	CoAP	AMQP	DDS	HTTP
Тип на Ино (от потребителско към индустриално)	1	5	3	7	9
Свързаност (от безжично към кабел)	7	9	5	3	1
Енергийни ограничения	7	9	3	5	1

Посока на предаване на данни	3	1	5	7	9
Мобилност	7	5	1	3	9
Тип на данните (от неструктурирани полу и структурирани)	7	9	3	1	5
Икономическа област	7	9	3	1	5
Количество свързани обекти	1	3	5	7	9
Мащабируемост	9	7	5	3	1
Среда	9	3	1	5	7
Скорост	7	9	5	3	1
Сигурност	1	3	7	5	9
Честота на включване	3	1	5	7	9
Изискване към предаваните данни	7	9	5	3	1

2.5 Глава IV

Бяха показани множество методи и протоколи за пренос на данни от обектите към системите за големи данни. В настоящата разработка е избран подход за сравнение на няколко протоколи за пренос на данни с участието на критерии, подходящи за системи с участието обекти, предаващи данни и системи за приемане, съхранение и обработка на големи данни. Целта е да се предложи метод за оценка за приложимост на определен протокол.

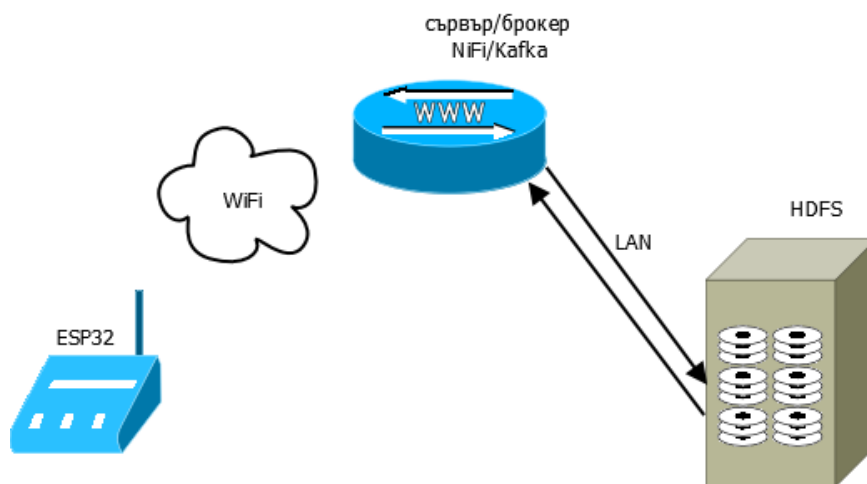
От направените експерименти се вижда, че тясното място за интеграция на данни, постъпващи от ИНО е записът в хранилището на системите за големи данни. От друга страна си струва да се отбележи, че хранилищата за данни са оптимизирани за съхранение на големи обеми от данни и не е рентабилно отделните съобщения да се съхраняват в отделни файлове, поради размера на клъстера на файловата система, който е далеч по-голям от размерът на едно съобщение, което води до неефективно използване на мястото за съхранение. Това налага обединението на повече съобщения в един файл. Многократното отваряне на файл и позициониране на курсорът води до увеличаване на времето за извършване на операциите по запис на информацията в файла. С цел по-добра ефективност е важно да се търсят подходи за преодоляване на този проблем. Това може да стане чрез използване на подходящи протоколи, както и с оптимизиране на системата за съхранение и

обработка на данните. Последното може да се направи чрез добавяне на буфер за агрегиране на множество съобщения преди те да бъдат записани в хранилището.

За проверка на ефективността и приложимостта на методологията от предишната глава, спрямо редица критерии на протокол в Ино, в тази глава се представят серия то експерименти, целящи да докажат (или променят) приложимостта на отделните критерии към избраните протоколи. Основните експерименти са разделени на 2 групи:

- как един обект предава данни по различни протоколи, при което се измерват характеристики като:
 - изпращане на данни от устройството (в частност рутер свързан между устройството и брокер/сървър);
 - енергия използвана при предаването.
- какво е влиянието на броя съобщения от брокера/сървъра към системата за записване на големи данни, при което се измерва:
 - закъснението между генерирането и получаването на едно съобщение, породено от преносът на съобщението през брокера/сървъра;
 - времето за запис на всички получени съобщения.

Експериментите от **първа група** са организирани около система, която приема данни по капа архитектура, избрана поради факта, че е опростена версия на ламбда и се използва широко от Ино. Ще се използват софтуерни инструменти, които са описани във втора глава. Написана е програма на microPython за всеки отделен протокол използваща готови библиотеки. Програмата се пуска за 600 секунди, през които непрекъснато изпраща предварително зададено съобщение организирано в цикъл.



Източник: собствено проучване

За Обект (Ино) е избрано устройство на фирмата WeMos с SoC (System on chip) ESP32 - WROOM32.

Програмен език е избран microPython подари широкото разпространение на Python и наличието на библиотеки за разгледаните протоколи.

За симулиране на данни от различни икономически области ще се използват готови данни то хранилище в интернет на адрес <https://www.kaggle.com/>. Подборът освен по области е направен и по типове данни от структурирани през полуструктурирани, както и неструктурирани.

Първото множество от данни [<https://www.kaggle.com/atulanandjha/temperature-readings-iot-devices>] е типичен за Ино - данни за температурата в помещение. Те са получени от проект за устройства, които следят температурата в заобикалящата ги среда. Данните представляват: уникален адрес на устройството, идентификатор на заобикалящата го среда, час в който е направено измерването и стойността на измерената температура в градуси С.

Вторият екстракт от данни е от финансовия сектор и е извлечен от сайт [<https://it.investing.com>] и представлява финансова информация за курса на американския долар през 2019 за всеки работен ден.

Третото множество от данни е избрано от [<https://www.kaggle.com/mmmarchetti/tweets-dataset>]. Данните са събран чрез обхождане на REST API на Twitter с помощта на библиотеката на Python tweepy3. То съдържа туйтовете на 20-те най-популярни потребители на Twitter (т.е. абонати с най-много последователи). Наборът от данни съдържа комбинация от относително структурирани туйтове, както и напълно неструктурирани туйтове, написани в разговорен стил.

Като обобщение на извършените експерименти и получените резултати е оформена Таблица 7, в която те са систематизирани и показани:

Таблица 7. Систематизирани резултати от измерванията

		MQTT	CoAP	AMQP	DDS	HTTP	
Структурирани данни	Консумирана енергия	14	5	10	13	12	mAh
	Трафик към Ино	3483	198	170	182	177	kB
	Трафик от Ино	4228	412	360	247	149	kB
Полуструктурирани данни	Консумирана енергия	14	5	11	12	13	mAh
	Трафик към Ино	4532	208	204	189	184	kB
	Трафик от Ино	6307	552	445	266	176	kB
Неструктурирани данни	Консумирана енергия	13	5	10	12	12	mAh
	Трафик към Ино	1494	174	195	163	161	kB
	Трафик от Ино	3343	907	422	230	204	kB

От експеримента се вижда, че MQTT предава данни на порядък по-бързо от останалите протоколи, но за сметка на това, този протокол е с по-висока консумация на енергия. HTTP консумира близко количество енергия, но за много по-малък обем данни (почти 20 пъти при структурираните данни) в сравнение с MQTT.

От представеният код при опитите се вижда, че програмния код при HTTP е много по-малко в сравнение с MQTT, което прави HTTP по-лесен за използване.

При CoAP количеството обменени данни е сходно с това на HTTP, но за сметка на това изразходваната енергия е почти 50% по малко.

AMQP е сравним с CoAP като количество обменени данни за едно и също време, но изразходваната енергия при AMQP е по-голяма.

AMQP е сравним с DDS като количество обменени данни и изразходвана енергия, но кодът при DDS е доста усложнен поради необходимостта от създаване на обекти (което в някои случаи е преимущество на този протокол).

От направения експеримент се вижда, че тегловните коефициенти за критериите:

- Енергийни ограничения;
- Тип на данните (от неструктурирани полу и структурирани);
- Скорост;
- Честота на включване;

са добре подбрани и съответстват на получените резултати.

Типът на данните не оказва значително влияние над консумираната енергия и количеството на обменената информация. В същото време е видно, че видът на протокола е от значение. Прави впечатление, че скоростта на обменената информация при полуструктурираните данни е най-голяма, следователно те успяват да използват най-пълноценно честотната лента на канала.

Както е отбелязано в първа глава, при работата с големи данни съществуват няколко основни фази за преминаване на данните от обект от ИНО към хранилище за големи данни:

- фаза на получаване;
- фаза на записване на данните в някакво хранилище, и
- последващата обработка или извършване на заявки върху тези данни.

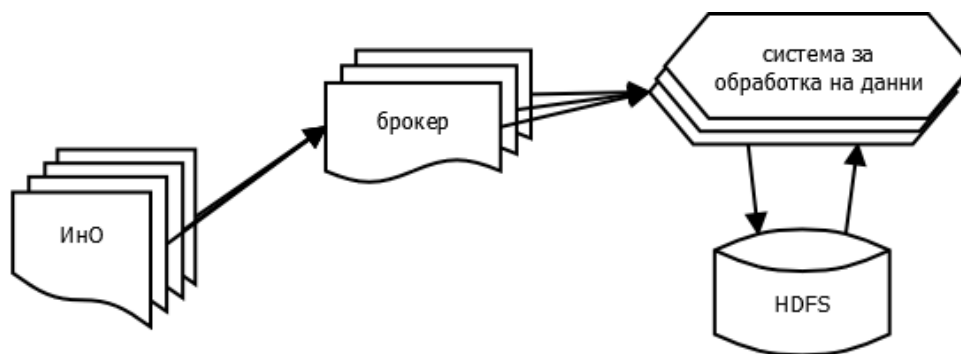
Много често тези фази са интегрирани. Например по време на получаване се извършва някаква обработка на данните, след което те се записват. Платформите за работа с големи данни са проектирани да обработват големи обеми от данни от сравнително малко на брой източници. Това е проучено и доказано с редица тестове, които изследват къде се намират тесните места на една система за приемане и обработка на големи данни.

Изследванията сочат, че системата е ограничена от броя потребители и скоростта на ефективна работа на HDFS деградира с увеличаване на броя файлове, особено ако са с малък размер. Както беше отбелязано по-горе, характерно за ИНО, в области, където се използват сензори и идентификатори е, че данните са малки по обем, но многобройни. Хоризонталното мащабиране ще се справи с увеличението на броя на свързаните обекти и подобно увеличение не би следвало да е от голямо значение за производителността на системата. От друга страна сравнително малкия обем на данни предавани от един обект за една сесия може да се окаже от решаващо значение за производителността. Стига се до извода, че платформите за работа с големи данни не са оптимизирани да работят в такъв режим.

За целите на експериментите от **втора група** се използва персонален компютър с ОС Ubuntu, с 32GB RAM, Intel i5 процесор и SSD. За максимална коректност на резултатите, по време на опитите не е имало други потребители, които да използват компютъра.

Опитната постановка разглежда основно процесите около съхранението/записа данни, а не тяхното генериране. Приема се, че има един обект (краен елемент/източник от Ино), който ще генерира данни, един брокер или приемник на данни и един компонент на системата за обработка на данни, като компонентите на Hadoop са разположени на едно място.

На Фигура 8 е показана диаграма как отделните системи комуникират помежду си. Данните от Ино се приемат от брокер, който разпределя натоварването и предотвратява струпването на данни на входа. Системата за обработка на данни взаимодейства с брокера, като извършва (при необходимост) трансформация на данните и ги съхранява в системата за съхранение на данни.



Фигура 8. Данни от Ино

Източник: собствено проучване

В експеримента се използват две **системи за обработка на данни** - Apache NiFi и Apache Kafka.

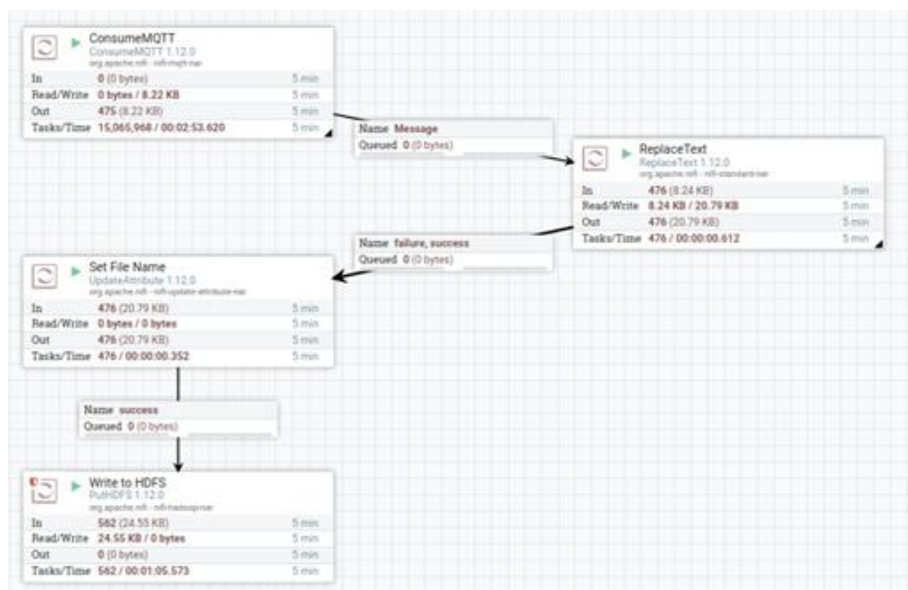
Инсталацията на Apache NiFi е версия 1.12.0, като се ползва конфигурацията по подразбиране. Инструментът има уеб-базиран графичен интерфейс за проектиране, контрол, обратна връзка и наблюдение на потоците от данни. Apache NiFi предлага широк набор от

така наречени „процесори“, които са по същество Java програми, които имат стандартизиран вход и изход и параметризуеми настройки, чрез собствен скриптов език. Всеки процесор има нула или повече връзки, дефинирани за него. След като процесорът завърши обработката на определено съобщение, той ще насочи (или „прехвърли“) резултатът към някоя от връзките с други компоненти (процесори) и така се определя пътя на потокът от данни.

В настоящият експеримент са използвани няколко „процесора“, които съществуват в пакета Apache NiFi:

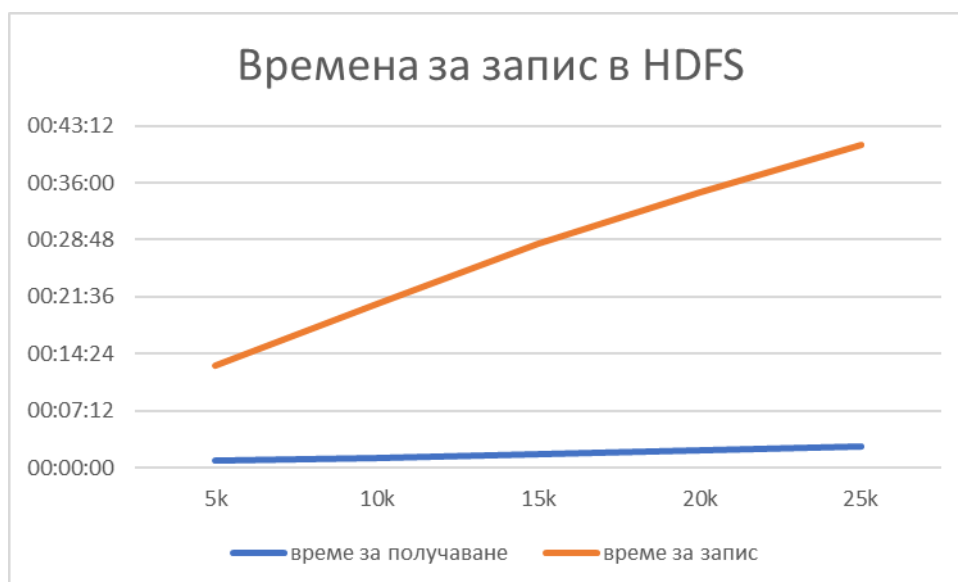
- ConsumeMQTT е името на процесора за връзка с MQTT брокер;
- ReplaceText е името на процесор на NiFi, който се ползва за обработката на данните при постъпване;
- “Set file Name” е името на процесор от тип “UpdateAttribute”, този процесор актуализира атрибутите на входния поток, като използва свойства или правила, добавени от потребителя;
- „MergeContent” е процесор, който обединява група съобщения заедно;
- „Write to HDFS” е името на процесор от тип „PutHDFS”, който се грижи за свързване с инстанция на Hadoop.

С готовите процесори, които NiFi поддържа е построен граф за приемане на данни от MQTT брокер. След като данните се приемат, следва обработка, и записване в HDFS. С помощта на графичното представяне на NiFi пътя на данните изглежда както е показан на Фигура 10



Фигура 9. Обработка на данни в NiFi.

Времената за записване в HDFS се изчисляват въз основа на разликата във времевите маркери за създаване на файл и времето на последния достъп в атрибута на файла в HDFS. Връзката между времето за завършване на запис без буфер и времето за получаване на съобщения е линейна - Фигура 13



Фигура 10. Времена за записване в HDFS

Източник: собствено проучване

Резонно възникват съмнения дали това забавяне не е характерно само за Apache NiFi, затова експериментът се повтаря с друга система за управление на съобщенията - Apache Kafka. Диаграмата на опитната постановка е показана на Фигура 11. като тя наподобява модела, при който се ползва NiFi.



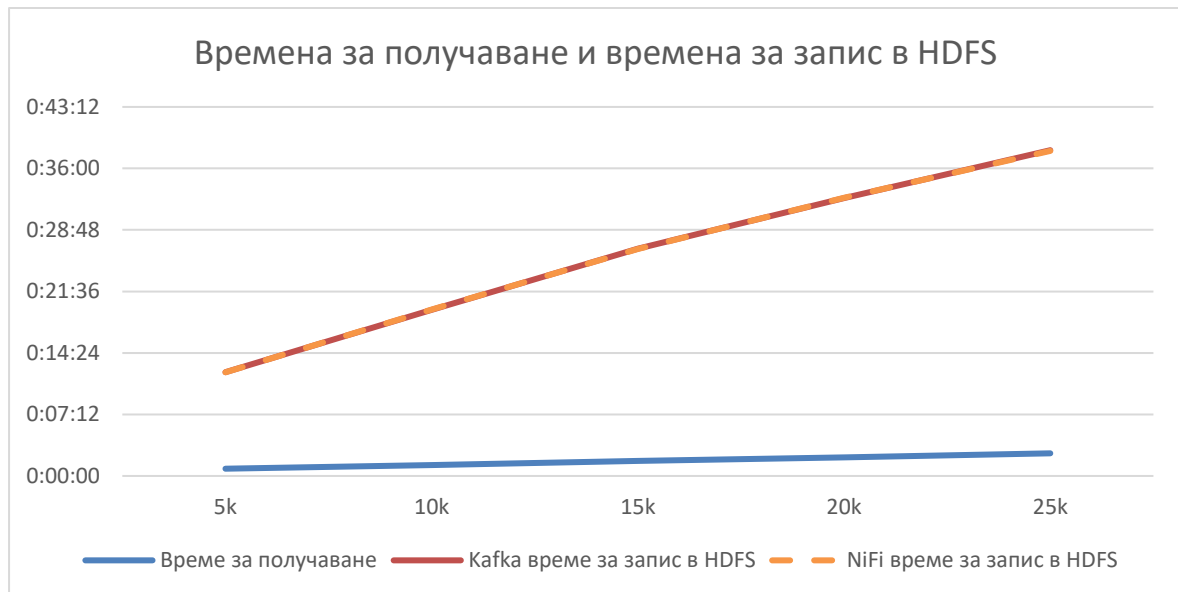
Фигура 11. Кафка настройка.

Постановка е, че Kafka ще превежда съобщения получени от MQTT брокера към HDFS. Инсталацията на Kafka е направена на същия компютър, който е описан по горе и работи в режим с един активен клиент. Kafka е конфигурирана за абонамент към същата тема от същия брокер както и при NiFi. За запис в HDFS бе избрана Camel библиотека.

Apache Kafka. Тя съчетава три ключови възможности за стрийминг на събития, да публикува (пише) и да се абонира за (четене) на потоци от събития; да съхранява потоци от събития, както и да обработва потоци от събития. Kafka е разпределена система, състояща се от сървъри и клиенти, които комуникират чрез TCP мрежовия протокол. Клиентите позволяват да се разработват разпределени приложения и микроуслуги, които четат, пишат и обработват потоци от събития успоредно, в мащаб и по начин, устойчив на грешки.

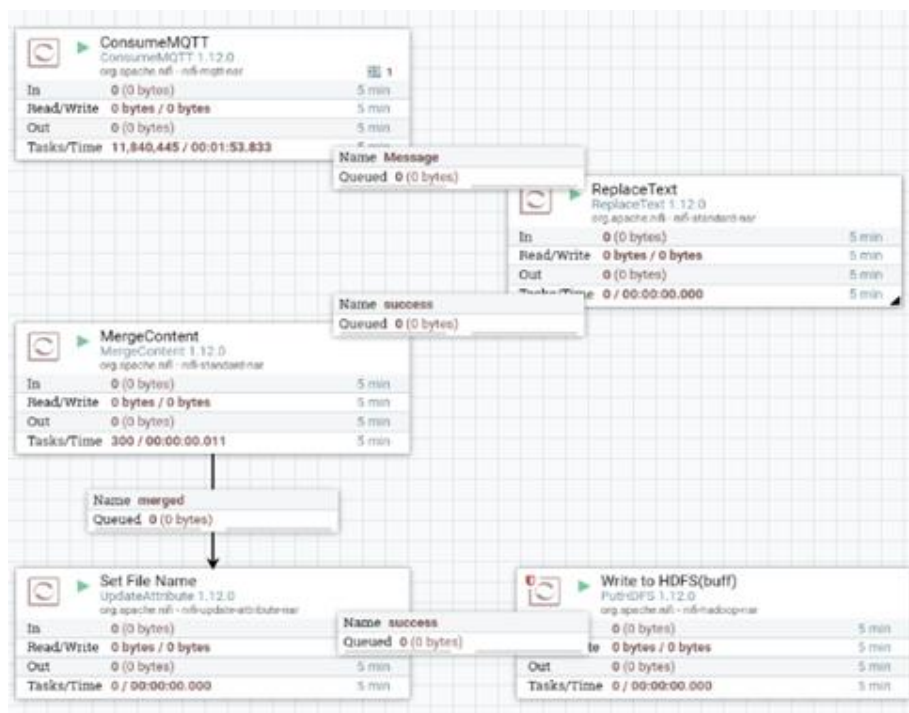
Използва се стандартна инсталация на версия 2.6.0. Конфигурацията на тази система е базирана на редица конфигурационни файлове с разширение .properties. Kafka разполага с готов конектор за брокер на MQTT, но няма налични за CoAP, както и такъв за писане в HDFS. За това тук се ползват налични библиотеки от трета страна - Camel на адрес <https://camel.apache.org/components/3.4.x/kafka-component.html> .

Резултатите от няколко опита са показани на Фигура 12. Както се вижда, няма големи разлики във времената между NiFi и Kafka.



Фигура 12. Времена за получаване и времена за запис в HDFS.

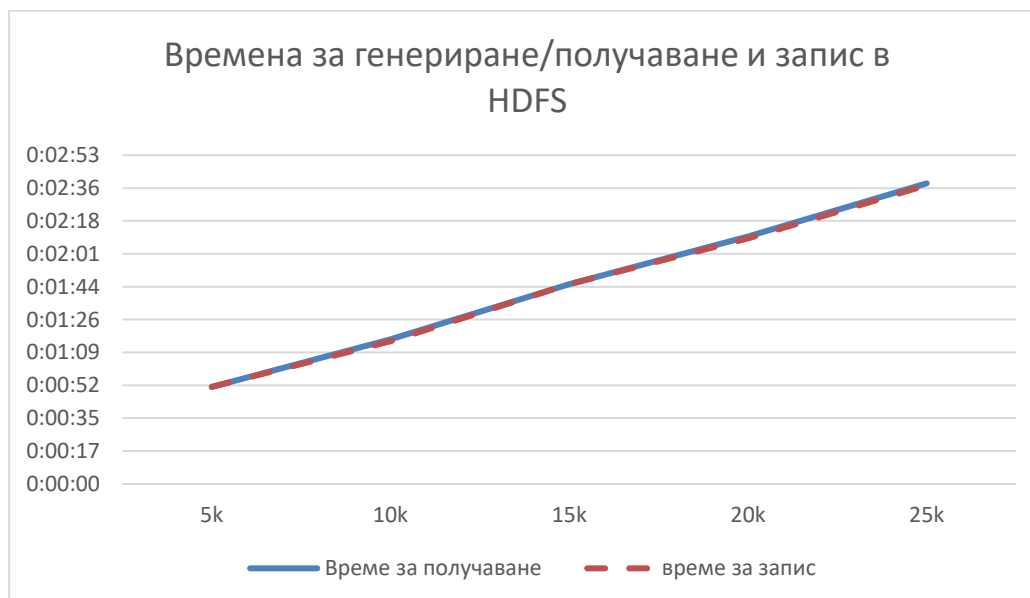
Добавянето на буфер, който да поема потока от данни и да го групира преди да го запише, би могло да подобри времената за запис. За целта в Apache NiFi, към съществуващата структура (граф) добавяме нов процесор, MergeContent, който изпълнява ролята на буфер. Това е показано на Фигура 13. Буферът е добавен преди процесора за име на файла



Фигура 13. Буферът е добавен преди процесора за име на файла

Източник: собствено проучване

След повторение на експериментите проведени с предишната конфигурация, резултатите са показани графично на Фигура 14.



Фигура 14. Времена за генериране/получаване и запис в HDFS

Източник: собствено проучване

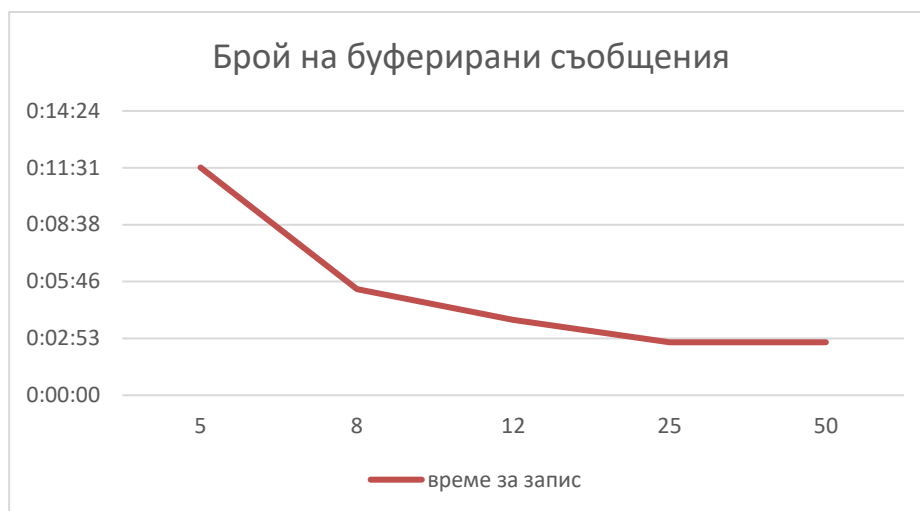
От представените резултати става ясно, че добавянето на буфер преди запис в HDFS намалява значително времето за запис. Времето за запис в HDFS е по-малко от нужното за генериране на всички съобщения. Това е така, защото наличието на буфер отлага във времето началото на запис във файла. Отлагането на записа в HDFS довежда до ситуация, в която някакво събитие е настъпило, но не е отразено в хранилището за данни. Това оказва негативно влияние над изискването, формулирано при дефиниране на опитната постановка, данните да се намират в HDFS в реално време. Буферът има няколко параметъра, които могат да влияят над моментът, в който започва даден запис в HDFS. Единият параметър е размер, а другият - време. При Apache NiFi, процесорът който натрупва данни има таймер, служещ за управление предаването на данните към следващия процесор. Другият параметър е максималния брой съобщения или размер на буфера.

Наличието на буфер подобрява работата на системата, като в същото време въвежда закъснение, което може да е проблем в някои ситуации. С промяна размера на буфера може да се влияе на това закъснение. Ако размерът на буфера се намали, намалява се и закъснението, но съществува риск да се увеличи многократно общото време за запис.

Опитите започват с минимален размер на буфера и фиксиран брой съобщения, като в последствие се увеличава размера на буфера. Получените резултати се сравняват. Подобно на метода на най-малките квадрати, този подход може да ни даде насока как да продължим изменението на размерът на буфера. Итерациите и увеличаване на размерът на буфера трябва да спрат преди времето за запис да стане по-малко от времето за генериране на данните. Това ще гарантира компромис между закъснението породено от наличието на буфер и закъснението породено от големия брой съобщения, които трябва да се запишат в HDFS.

За целите на експеримента броят на съобщенията се фиксират на 25000, а периодът на изпращане на съобщение на 100 ms. Времето за генериране на тези съобщения е 00:02:42. С цел намиране на оптимален размер на буфера започва промяна на броя съобщения, които се буферират. Резултатите са показани на Фигура 15. От резултатите можем да се съди, че времето за запис намалява експоненциално спрямо броя на буферирани съобщения. Вижда

се, че закъснението при записът на данни в HDFS нараства с нарастване броя на съобщенията записвани в буфера.



Фигура 15. Брой на буферирани съобщения спрямо времето за запис

Източник: собствено проучване

2.6 Заключение

Класифицирането на разгледаните протоколи за пренос на данни от Ино и анализът на техните характеристики сочи, че не може да се изведе и открие един протокол, който превъзхожда останалите. Освен HTTP, всички споменати протоколи са позиционирани като Ино протоколи за публикуване/абонамент в реално време с възможност за поддръжка на голямо количество устройства. Тук е важно да се отбележи, че при разглеждането на протоколите има значение и по какъв начин за различните приложения се дефинират категории като „реално време“ (секунди, милисекунди или микросекунди) и „неща/обекти“ (WSN възел, мултимедийно устройство, лично преносимо устройство, медицински скенер, управление на двигателя и т.н.), защото те се отразяват и на избора на протокол за съответния случай.

Днес Интернет на Обектите работи с множество различни комуникационни протоколи. Заедно с това се появяват нови приложения и подходи на използване на Ино както в промишлени, така и в битови и други приложения, което изисква тяхната работа да е с

подходящи протоколи за пренасяне на данни. Заедно с това трябва да се следи за безопасното и ефективно управление на устройствата, което е ключовият момент за устойчивото развитие на Ино в световен мащаб. Това е основателна причина, поради която работата и ефективността на различните Ино протоколи е от голямо значение. При избора на подобен протокол е важно и познаването на собствените бизнес нужди и изисквания, на предимствата и недостатъците на наличните протоколи на базата на което да се избере този, който най-добре отговаря на нуждите в конкретен случай.

В литературата съществуват множество източници, които разглеждат характеристиките на отделните протоколи от техническа гледна точка. В същото време не бяха открити източници, които да анализират приложимостта на отделните протоколи на базата на различни критерии.

В дисертационния труд е обърнато внимание на няколко от най-популярните протоколи, използвани в Ино - MQTT, CoAP, AMQP, DDS и HTTP. Всички те се ползват от много приложения и области като методите за пренос на данни. Разгледани са парадигмите за приемане и обработка на приети от Ино данни в системи за големи данни, както и някои от най-популярните продукти в платформи за работа с големи данни. Направено е сравнение между протоколите за пренос на данни, като е предложена методология базирана на точковите оценки с цел формулиране на количествена оценка на отделните протоколи спрямо предложени критерии. Този подход е избран, за да бъде улеснен изследовател или разработчик, който не е запознат в детайли с многообразието от протоколи, за първоначална оценка на приложимост на даден протокол към конкретна разработка на система.

Класическите архитектури използвани в платформите за големи данни не са оптимизирани да посрещат нуждите за приемане и обработка на данни от Ино, където се търси подобряване на скоростта, с която данните се записват в хранилището. Показано е, че с добавяне на буфер между брокера, който поема за разпределение данните в системата и последващото записване на данните в хранилището може да се получи повишаване на ефективността на работата на системата. При буферирането се натрупват известен брой съобщения, с което се редуцират операциите на работа с файлове в хранилището. Наличието

на такъв интегриращ елемент в системата води до забавяне на записа на данните в хранилището. Това може да не е подходящо за случаите, когато трябва да се вземат решения в реално време. Моментът, в който информацията става налична в хранилището и достъпна за анализ, може да се контролира с размера на буфера - броя на съобщенията и/или времето, за което данните от буфера се записват в хранилището.

3 СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД

Nikolai Kaskatiiski, Models and approaches of Internet of Things in education, 7th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2017), November 3-4th, 2017, UNWE, Sofia, Bulgaria, ISSN 2367-7635 (PRINT), ISSN 2367-7643 (ONLINE) достъпна на: <http://icaictsee.unwe.bg/past-conferences/ICAICTSEE-2017.pdf> pp. 554-561.

Николай Къскатийски, Сравнителна характеристика на данните в застрахователния сектор, Новите информационни технологии и големите данни: Възможности и перспективи при анализите и управленските решения в бизнеса и икономиката и социалната сфера Сборник с доклади от международна научна конференция организирана от катедра „Статистика и иконометрия - със съдействието на факултет „Приложна информатика и статистика - и в партньорство с Националния статистически институт, 6 – 7 юни 2019, издателски комплекс УНСС, ISBN 978-619-232-292-2 стр. 231-242.

Николай Къскатийски, Size and speed of data generated by Fleet Management Software, VIII Международна научна конференция техника. Технологии. Образование. Сигурност 2020 SCIENTIFIC TECHNICAL UNION OF MECHANICAL ENGINEERING INDUSTRY-4.0, 108, Rakovski Str., 1000 Sofia, Bulgaria, ISSN PRINT 2367-8399, ISSN WEB 2534-8493 достъпна на: <https://stumejournals.com/journals/tm/2020/1/pdf> pp. 13-16.

N Kaskatiiski and L Boyanov, Acquiring and storing Internet of Things data into Hadoop, IOP Conference Series: Materials Science and Engineering, Volume 1031, International Conference on Technics, Technologies and Education 2020 (ICTTE 2020) 4th-6th November 2020, Yambol, Bulgaria, IOP Conf. Ser.: Mater. Sci. Eng. 1031 012060, достъпна на: <https://iopscience.iop.org/article/10.1088/1757-899X/1031/1/012060> .

[https://www.scopus.com/results/results.uri?src=s&st1=&st2=&sot=b&sdt=b&origin=searchbasic&rr=&sl=91&s=\(AUTHOR-NAME\(Boyanov\)%20AND%20TITLE\(Acquiring%20and%20storing%20Internet%20of%20Things%20data%20into%20Hadoop\)\)&searchterm1=Boyanov&connector=AND&searchterm2=Acquiring%20and%20storing%20Internet%20of%20Things%20data%20into%20Hadoop&searchTerms=&connectors=](https://www.scopus.com/results/results.uri?src=s&st1=&st2=&sot=b&sdt=b&origin=searchbasic&rr=&sl=91&s=(AUTHOR-NAME(Boyanov)%20AND%20TITLE(Acquiring%20and%20storing%20Internet%20of%20Things%20data%20into%20Hadoop))&searchterm1=Boyanov&connector=AND&searchterm2=Acquiring%20and%20storing%20Internet%20of%20Things%20data%20into%20Hadoop&searchTerms=&connectors=)