

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра _____

УТВЕРЖДАЮ

Заведующий кафедрой
прикладной информатики
и теории вероятностей

д.т.н., профессор

_____ К.Е. Самуйлов

«___» _____ 20__ г.

КУРСОВАЯ РАБОТА БАКАЛАВРА

на тему

«Принципы построения систем туманных вычислений»

010200 — Математика и компьютерные науки

Разработчик

Студент группы НК-201

Студенческий билет №: 1032172734

Папикян.Г.Т

«___» _____ 20__ г.

Руководитель

Доцент кафедры прикладной информатики и
теории вероятностей РУДН.

Москва 20__

Содержание

1. Введение.....	2
2. Применение системы.....	3
2.1. Умный транспорт.....	4
2.2. Умные здания.....	5
3. OFC - OpenFog Consortium.....	7
3.1. Ключевые идеи.....	8
3.2. Описание эталонной модели.....	14
3.2.1 Общие задачи (cross-cutting concerns).....	16
3.2.2 .Архитектурные части(Architecture views).....	18
4. Электрическая сеть на основе системы туманных вычислений.....	23
5. Источники.....	26

1. Введение

Согласно источнику [3], термин «Туманные Вычисления» (Fog Computing) был впервые введен в употребление сотрудником компании Cisco, Flavio Bonomi в 2012 году в работе «Fog Computing and Its Role in the Internet of Things». С ростом устройств интернета вещей увеличиваются и потоки данных, путешествующих по сети. Таким образом, увеличиваются и сетевые задержки, которые несовместимы с рядом систем, требующих обработки данных в реальном времени.

Концепция туманных вычислений подразумевает использование ресурсов устройств интернета вещей с целью перенесения вычислений, традиционно выполняющихся в облаке, ближе к грани сети. Таким образом обеспечивается ряд характеристик, ранее недоступных для систем облачных вычислений, одним из которых является низкая сетевая задержка. Важно отметить, что туманные вычисления призваны не заменять традиционные облачные, а дополнять их.

Далее мы рассмотрим, какие идеи лежат в основе туманных вычислений, также изучим некоторые системы, для которых свойства туманных вычислений являются критически важными.

Цель работы

Ознакомление с концепцией систем туманных вычислений и принципами их построения.

2. Применение системы.

Перед тем, как перейти к рассмотрению примеров использования системы туманных вычислений (далее СТВ) , необходимо составить определенные представления о ключевых ее свойствах. [1] Флавио Бономи дает следующее определение: «Система туманных вычислений — это платформа с высокой степенью виртуализации, предоставляющая услуги хранения, вычисления и сетевого взаимодействия между конечными устройствами и традиционными облачными вычислительными центрами, причем такая платформа чаще всего расположена на грани сети»

Он также подчеркивает некоторые особенности СТВ, возникающие по причине близкого расположения системы к грани сети:

- Осведомленность о местонахождении и низкие сетевые задержки
- Географическая распределенность (в отличие от централизованных облачных вычислительных центров)
- Огромное количество узлов
- Поддержка мобильности
- Взаимодействия в реальном времени
- Преимущественное использование беспроводного доступа
- Гетерогенность: узлы могут иметь разные характеристики, как физические, так и программные
- Поддержка аналитики данных в реальном времени и взаимодействия с облаком: СТВ играют большую роль в обработке данных близко к их источнику

Далее рассмотрим пару примеров применения СТВ, приведенных в источнике [4]

2.1. Умный транспорт

Согласно источнику [4], умные автомобили будут генерировать терабайты информации каждый день. Из этого делается вывод, что традиционный подход, основанный на централизованной системе управления в данном случае не подходит, так как могут возникнуть сетевые задержки (кроме того, в некоторых случаях обеспечение большой пропускной способности возможно, но экономически невыгодно относительно решений, основанных на СТВ)

В данном случае узлы СТВ разделены на несколько уровней, первый из которых составляют сами транспортные средства, способные локально обрабатывать информацию, собираемую с соответствующих датчиков, обеспечивая такие функции, как беспилотное движение, предотвращение столкновений, навигация, синхронизация скорости движения с потоком, и еще ряд других функций. (Отметим, что транспортные средства на этом уровне могут образовывать ad-hoc сети для осуществления описанных выше функций)

Следующий уровень — узлы, связанные с самой дорогой, это могут быть умные светофоры, или другого рода инфраструктурные узлы, которые тоже будут собирать информацию с ряда датчиков, например, с дорожных камер, для локальной обработки. Такие узлы более высокого уровня будут, по мере обнаружения транспортных средств, при необходимости, оповещать их о специфических дорожных условиях. Например, дорожный узел, обнаруживающий транспортное средство, движущееся с большой скоростью, может потребовать ее ограничения в зависимости от состояния данного участка дороги [2].

В целом, предполагается, что каждый уровень будет отправлять данные на уровень выше. Таким образом, транспортные средства отправляют данные на дорожные узлы, а те, в свою очередь, производят локальный анализ, совмещая эти данные с полученными от своих датчиков, и отправляют их еще на уровень выше. На следующем уровне могут располагаться, например, районные или городские узлы. В действительности, количество уровней определяется конкретными задачами, но все данные в конце концов приходят на облачные вычислительные центры, где обеспечивается долгосрочный их анализ.

Следует заметить, что каждый уровень является самодостаточным, например, по той причине, что транспортные средства обладают собственными, пусть даже и ограниченными ресурсами, при невозможности установления соединения с дорожными узлами такие критически важные функции, как беспилотное вождение и предотвращение столкновений все еще будут обеспечены, а обрабатываемая информация будет храниться локально до тех пор, пока не появится возможность отправления ее на более высокий уровень. То же самое верно и для узлов на более высоком уровне. Таким образом, временная недоступность самого облака не станет причиной нарушения работы всего механизма.

2.2. Умные здания

Для функционирования такой системы необходимо большое количество разных датчиков и приводов, контролирующих и измеряющих состояние здания. Эти устройства контролируют противопожарные системы безопасности, доступ к помещениям, с помощью камер может производиться оценка количества людей в помещениях и т.д.

Некоторые из данных систем чувствительны к задержкам, например, система безопасности, поэтому предлагается использовать СТВ. Так, каждое помещение содержит собственный узел, занимающийся всеми необходимыми локальными процессами: регуляцией температуры и освещения, активацией систем безопасности и сбором прочих данных для анализа. Остальные узлы распределяются в соответствии с иерархической структурой здания: нижний уровень узлов — отдельные помещения, уровень выше — корпуса здания, еще выше — все здание, последним уровнем является централизованная система, основной задачей которой является сбор данных на долгосрочное хранение и их анализ. На этом примере хорошо видно, что чем выше уровень узлов, тем меньше непосредственным управлением конечных устройств и тем больше агрегацией и анализом данных они занимаются. Именно поэтому в ситуациях, когда верхние узлы временно недоступны, система продолжает оставаться работоспособной.

Рассмотренные выше примеры отражают характерные качества систем туманных вычислений: распределенность, низкие сетевые задержки, гетерогенность, использование беспроводного доступа, взаимодействие с облачными вычислительными центрами и.т.д

3. OFC - OpenFog Consortium

Согласно источнику [6], 17 ноября 2015, в результате объединения таких компаний, как ARM, Cisco, Dell, Intel, Microsoft и Принстонского университета, был основан консорциум OpenFog, с целью создания эталонной модели для систем туманных вычислений. Сегодня среди участников OFC находятся более 50 компаний, среди которых: Orange, Toshiba, Hitachi, Пизанский университет, и.т.д [7].

В феврале 2017 года была опубликована бумага под названием «OpenFog Reference Architecture for Fog Computing». OpenFog Reference Architecture определяет архитектуру СТВ лишь от среднего до высокого уровня абстракции. Обещается, что последующие публикации дополнят имеющуюся модель подробностями об устройстве более низкоуровневых частей системы.

OpenFog RA состоит из трех частей: первая часть описывает сценарии использования СТВ, вторая — ключевые идеи, на которых основана эталонная модель, а третья — саму эталонную модель.

Некоторые примеры из первой части мы уже рассмотрели, далее познакомимся с ключевыми идеями модели, изложенными во второй части, а затем рассмотрим эталонную модель.

Начать следует с самого важного. OpenFog RA дает следующее определение СТВ: «Туманные вычисления — это горизонтальная архитектура системного уровня, распределяющая функции вычисления, хранения, управления и сетевого взаимодействия ближе к пользователям и конечным устройствам пространства умных вещей»

3.1. Ключевые идеи

Рассмотрим 8 ключевых идей, с учетом которых построена архитектура:

1. Security (Безопасность)
2. Scalability (Масштабируемость)
3. Openness (Открытость)
4. Autonomy (Автономность)
5. Programmability (Программируемость)
6. Reliability, Availability, Serviceability
(Надежность, Доступность, Обслуживаемость)
7. Agility (Проворство)
8. Hierarchy (Иерархия)

Безопасность

В действительности, безопасность является самостоятельной, заслуживающей особого внимания, темой. OpenFog RA имеет дополнение с подробным описанием аспектов безопасности системы, которого мы касаться не будем. Вместо этого, попробуем описать глобальную идею обеспечения безопасности СТВ.

Дело в том, что безопасность системы всегда будет зависеть от множества особенностей конкретной области применения. По этой причине OpenFog RA определяет ряд методов обеспечения безопасной среды исполнения, абстрагируясь от возможных областей применения. В таком случае пользователи эталонной модели смогут сфокусироваться лишь на обеспечении безопасности своей реализации лишь с учетом тех особенностей, которые навязывает их область применения. Обеспечение безопасности исполняемой среды подразумевает обеспечение безопасности для узлов СТВ, механизмов управления ими и их сетевого взаимодействия.

Общий подход обеспечения безопасности выглядит как
обнаружение → подтверждение безопасности → установления доверия
Каждый узел системы должен иметь центр сертификации на самом низком уровне абстракции, который будет подтверждать безопасность программного или аппаратного обеспечения на один уровень выше, а те, в свою очередь, будут

подтверждать безопасность инфраструктуры на более высоком уровне. Все это будет происходить с помощью технологий электронной сертификации.

В результате будет создаваться цепь доверия от самых низких уровней системы до самого верхнего — облака.

Масштабируемость

Масштабируемость позволяет динамически регулировать вычислительные нагрузки, стоимость предоставляемых услуг, производительность и другие аспекты СТВ.

Эталонная модель OpenFog определяет следующие типы масштабируемости:

- Масштабируемость производительности: регуляция производительности системы в зависимости от потребностей конкретного приложения в разные моменты времени
- Масштабируемость объема: изменение размера сети вследствие добавления\удаления умных вещей, также подразумевается возможность добавления(удаления) дополнительных программных или аппаратных компонент непосредственно к узлам
- Масштабируемость надежности: увеличение надежности системы путем добавления резервных узлов
- Масштабируемость безопасности: добавление(удаление) дополнительных аппаратных(программных) частей к узлам для улучшения свойств безопасности в зависимости от области применения
- Масштабируемость аппаратного обеспечения: возможность изменения внутренних компонент узлов и количества самих узлов в сети
- Масштабируемость программного обеспечения: увеличение эффективности использования вычислительных ресурсов узлов, посредством применения различных технологий виртуализации.

Открытость

Открытость является важным условием широкого распространения СТВ. Закрытость системы может приводить к нежелательным последствиям вроде высокой стоимости СТВ, или отсутствия инноваций, кроме того, будут возможны ситуации, когда продукты разных производителей неспособны взаимодействовать друг с другом, что будет являться преградой для широкого распространения и развития СТВ.

Открытость в результате будет означать большое количество различных производителей на рынке, продукты которых все еще будут совместимы. Таким образом, например, узлы разных производителей смогут быть обнаружены и включены в сеть на ходу, как, например, происходило в примере с умным транспортом в предыдущем разделе.

Все это будет достигаться путем стандартизации, тестирования и открытых реализаций систем.

Автономность

Суть автономности заключается в том, что узлы системы смогут продолжить выполнение своих функций, даже в том случае, если облако будет недоступно. Это обеспечивается тем, что вычисления (и в целом, логика работы) будут распределены на всех уровнях узлов. В некоторых случаях работоспособность системы может сохраняться не только при недоступности облака, но и части самих узлов. Например, как мы видели в примере с умным транспортом, при недоступности дорожных узлов, транспортные средства имели возможность поддержания ряда критических механизмов, вроде избежания столкновений и автономного управления.

Автономность также касается обеспечения безопасности, управления узлами и стоимости. В случае с безопасностью, подразумевается, что узлы смогут автономно обнаруживать различные сетевые атаки. Автономность управления узлами предполагает, что система динамического распределения задач между узлами также продолжит функционирование при недоступности облака. А что касается стоимости, в начале раздела про автономные транспортные средства мы сказали, что в некоторых

случаях обеспечение большой пропускной способности может являться экономически невыгодным решением; возможность осуществления автономных вычислений позволяет отправлять на облако намного меньше данных, так как данные для функционирования узлов обрабатываются локально, а на облако передаются только данные, необходимые для долговременного анализа поведения систем.

Программируемость

Подразумевается что узлы смогут быть перепрограммированы динамически и полностью автономно. Будет существовать система, распределяющая задачи между узлами. Также подразумевается возможность перепрограммирования более низкоуровневых частей системы; это касается, например, всевозможных обновлений операционных систем или драйверов узлов.

Программируемость опирается на различные технологии виртуализации для обеспечения эффективного использования ресурсов узлов. Таким образом, в результате применения технологий виртуализации увеличивается переносимость программных компонент между узлами.

Как результат, программируемость имеет положительное влияние на такие аспекты системы, как безопасность и стоимость: увеличение безопасности происходит за счет того, что появляется возможность автоматического применения патчей при обнаружении уязвимости, а стоимость снижается за счет эффективного распределения ресурсов.

Надежность, Доступность, Обслуживаемость

От английского Reliability, Availability, Serviceability (RAS), эти три свойства используются для оценивания всевозможных компьютерных систем, как программных, так и аппаратных. Под надежностью подразумевается способность системы корректно функционировать, даже если некоторые ее составные части вышли из строя. Доступность оценивает время активности системы за определенный период. Обслуживаемость оценивает простоту ремонта\обслуживания системы.

Далее приведем некоторые способы обеспечения данных свойств в СТВ.

Надежность: ведение системных журналов; обеспечение инструментов автоматической диагностики, и исправления неисправностей в системе; тестирование системных компонент, таких как драйверы.

Доступность: для увеличения времени доступности системы возможно использование избыточных узлов, использование ресурсов облака в критических ситуациях, чтобы система продолжала быть доступной, а также применение технологий машинного обучения для оптимизации времени устранения неполадок.

Обслуживаемость: простоту обслуживания можно достичь улучшением дизайна систем, автоматизацией процессов установки программного обеспечения; также, простота обслуживания пересекается с требованием об открытости СТВ: если системы будут открытыми и хорошо документированными, множество производителей сможет предоставлять услуги по обслуживанию, что положительно скажется на данном свойстве.

Проворство

Здесь речь идет о превращении данных в полезную информацию.

По мере того, как данные поднимаются вверх по иерархии узлов, начиная от датчиков и заканчивая облаком, все больше становится их общий объем и, следовательно, все больше полезной информации из них можно извлечь в результате анализа. С этой точки зрения, лучше всего — анализировать данные на облаке, так как там их всегда больше, чем на любом другом уровне. Но с другой стороны, чем больше данных пропускается через сеть, тем менее экономически выгодной становится вся система.

Функционирование каждого узла полагается на извлечение информации из данных, собранных датчиками. Если объем данных, необходимых для работы определенного узла становится доступным уже на каком-то уровне, не доходя до облака, то предлагается там же и извлекать информацию, сокращая, таким образом, массив данных, передаваемых на облако. То есть идея в том, чтобы анализ данных происходил так скоро, при подъеме по иерархии, как скоро это становится

возможным. Например, в случае с умным транспортом, узлы светофоров в результате анализа данных дорожных камер могут определить транспортные средства, которым нужно отправить предупреждение, и поэтому нет необходимости отправлять эти данные на облако, чтобы только после анализа на нем светофор узнал, что дальше делать. Таким образом, извлечение информации происходит так скоро, как скоро это возможно.

Иерархия

Системы туманных вычислений предполагают иерархию. И в частности, по этой причине говорится, что СТВ дополняют облачные системы, а не заменяют их.

Количество уровней иерархии определяется конкретными применениями и потребностями, например, как в случае с умными зданиями, чем больше помещений в зданиях и чем больше самих зданий в системе, тем больше уровней иерархии возникает. Необходимость новых уровней в системе узлов обусловлена эффективностью анализа поступающих данных. Например, если в какой-то стране, в которой все города реализуют систему умных зданий, возникает необходимость получения статистики по каждому городу, то необходимо иметь уровень узлов, который бы представлял отдельные города, создавая таким образом отдельный уровень, занимающийся созданием и обработкой таких статистических данных.

С увеличением уровня узлов в иерархии, также возрастает «интеллект системы». На высших уровнях обрабатываются большие массивы данных, собранных всеми теми узлами, которые расположены ниже в иерархии. А на низших уровнях, хоть и происходит некий локальный анализ, но данных намного меньше, и при этом больше датчиков, с которыми узел непосредственно взаимодействует.

Таким образом, складывается картина, согласно которой, узлы, расположенные ниже в иерархии, должны обладать меньшими вычислительными ресурсами, но большей разновидностью интерфейсов ввода-вывода; а узлы, расположенные выше, меньшей разновидностью интерфейсов ввода-вывода, но большими вычислительными ресурсами.

3.2. Описание эталонной модели

Описание эталонной модели глобально разделено на две части: описание различных архитектурных частей (architecture views) системы и определение ряда задач (cross-cutting concerns), которые являются общими для всех архитектурных уровней и частей системы, и таким образом, заслуживающие особого внимания.

(Архитектурные части в данном случае уподобляются уровням абстракции)

Разделение на описание различных архитектурных частей обусловлено тем, что вся эталонная модель изначально предназначается для определенных групп заинтересованных лиц (stakeholders), и подобное раздельное описание частей позволяет достаточно четко разграничить задачи, относящиеся к этим самым группам заинтересованных лиц. Такими заинтересованными лицами могут быть производители низкоуровневой аппаратуры, разработчики программного обеспечения, лица, занимающиеся интеграцией различных частей системы, техническая поддержка, и.т.д

Архитектура системы представляется следующим изображением:

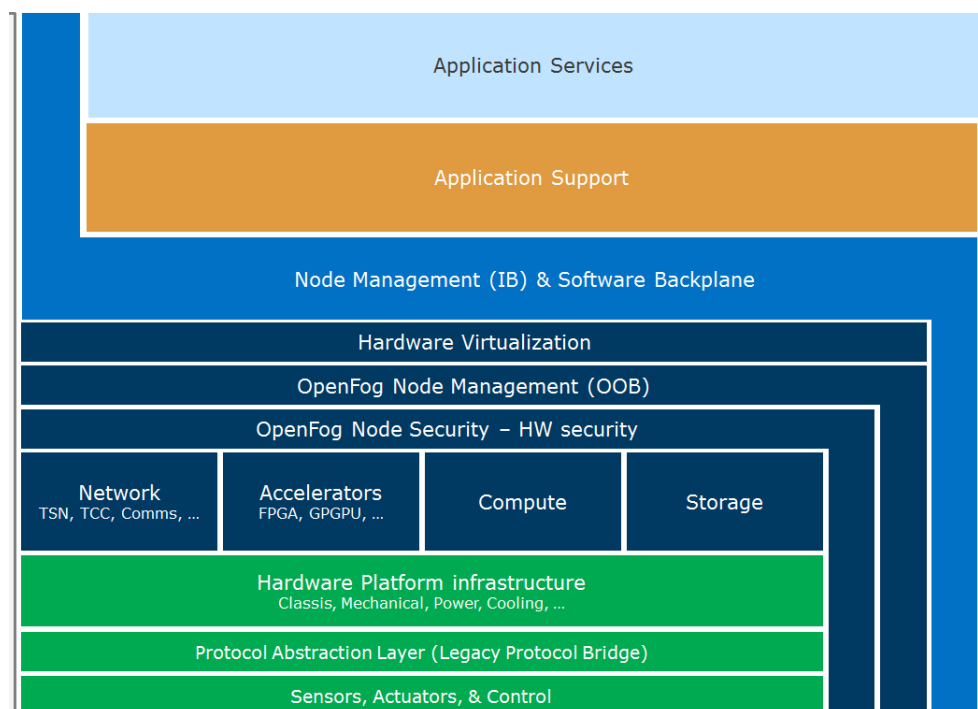


Рис.1 Архитектура системы туманных вычислений [4].

Архитектурными частями являются:

- Программное обеспечение: представляется первыми тремя уровнями на изображении, то есть здесь мы имеем дело с прикладными программными обеспечением, реализующим определенную бизнес-логику, а также с системой управления и распределения задач между узлами.
- Система: на изображении представляется уровнями от hardware virtualisation до hardware platform infrastructure
- Узел: представляется последними двумя уровнями на изображении.

Общими для всех уровней задачами являются:

- Производительность: Для обеспечения производительности всей системы необходима вовлеченность всех групп заинтересованных лиц.
- Управляемость: подразумевается управление множеством аспектов системы, что также связано с обеспечением Надежности, Доступности, Обслуживаемости
- Анализ данных и управление: анализ данных и управление происходят на разных уровнях, а управление не всегда относится к физическим устройствам (в зависимости от сценария использования).
- Безопасность: Ранее мы говорили о принципе обеспечения безопасности на всех уровнях, предполагающей создание цепочки доверия, начинающейся от самого нижнего уровня, и доходящего до самого верхнего — облака, очевидно, подобное распределение данной задачи на все уровни, делает безопасность общей задачей всех заинтересованных групп.
- Многоуровневые приложения: должно быть возможным эксплуатирование приложений на разных уровнях узлов.

3.2.1 Общие задачи (cross-cutting concerns)

Производительность.

По причине наличия «интеллекта» в узлах, вся сеть должна быть способна адаптироваться к изменениям трафика. В следствие инспекции траффика возможно выделение бОльших\мЕньших ресурсов для определенных частей системы, в данным случае будут применяться технологии виртуализации. Кроме того, если какой-то узел занимается инспекцией трафика, то, вероятно, такой узел будет нуждаться больше в процессорном времени и меньше в долгосрочной памяти, следовательно, и с этой точки зрения можно оптимизировать потребление ресурсов.

В целом, очевидно, нельзя допускать, чтобы увеличение производительности одной части системы повлекло за собой ее уменьшение в другой.

Безопасность.

Безопасность Системы туманных вычислений должна обеспечиваться политикой безопасности, определяющей, какие части системы, при каких обстоятельствах, к каким частям системы могут получать доступ. Разные части такой политики далее будут реализованы разными частями системы: в одних случаях они будут интегрированы в аппаратное обеспечение, а в других — применяться динамически.

Обеспечение безопасности можно разбить на три более специфические подзадачи:

Обеспечение конфиденциальности, то есть предотвращение распространения приватных данных; целостности, то есть предотвращения внесения изменений в защищенные файлы системы или в сам исходный код неавторизованным лицам; доступности, то есть способности системы предоставлять услуги пользователям с учетом возможных критических нагрузок;

Управляемость.

Управляемость обеспечивается реализацией специальных интерфейсов, которые разделяются на два вида:

внутренние — являющиеся частью операционной системы узла; такие интерфейсы обычно с определенной регулярностью отправляют специальные

сигналы(с определенной частотой) системе управления более высокого уровня для подтверждения работоспособности;

внешние — являющиеся отдельными самостоятельными системами, связанные с помощью проводных или беспроводных соединений с узлом. Также внешние интерфейсы управления могут быть частями узла, но при этом полностью автономными от программной среды самого узла. Такая автономия от систем узла предоставляет более широкие возможности по управлению.

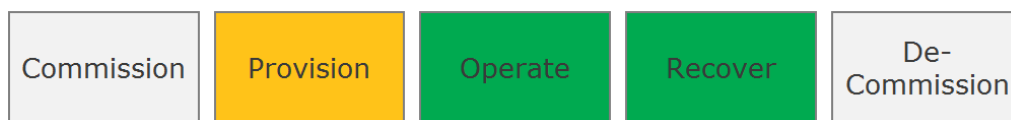


Рис.2 Жизненный цикл узла, с точки зрения управления [4].

Жизненный цикл управления иллюстрируется изображением выше.

Этап «*Commission*»: подразумевает начальную инициализацию узла которая включает в себя такие рутинные операции, как сертификация, настройка времени и.т.д.

«*Provision*»: подразумевает регистрацию узла, при которой системе управления сообщается специфическая информация об узле и начинается использование ресурсов данного узла.

«*Operate*»: на данном этапе полагается работа узла в штатном режиме с учетом всех требований Надежности,Доступности и Обслуживаемости

«*Recovery*»: на этом этапе полагается восстановления узла после определенных сбоев при помощи интерфейсов, описанных выше. При этом полагается, что это будет автономный процесс, в котором могут также принимать участие другие(работоспособные) узлы системы.

«*De-Commission*»: на этом этапе узел подготавливается для переиспользования, такая подготовка подразумевает форматирование носителей информации, во избежание передачи персональных данных от одного приложения другому.

3.2.2. Архитектурные части (Architecture views)

Узел.

В описании эталонной модели узел является самой низкоуровневой составляющей. Круг заинтересованных лиц в случае с «узлом», включает разработчиков микросхем, производителей кремниевых деталей, разработчиков микропрограмм и.т.д

Основными аспектами данного уровня являются:

безопасность узла: так как узлы системы являются физическими устройствами, со всеми вытекающими из этого угрозами физического доступа к аппаратному обеспечению, необходимо: во первых, конструировать узлы таким образом, чтобы получение несанкционированного физического доступа было наиболее трудной задачей, и во-вторых, необходимо оснащать узлы датчиками обнаружения физического взлома системы. При этом, узлы также должны оснащаться механизмами защиты на случай, если физический взлом произошел. В случае физического взлома узел должен отправить соответствующий сигнал специальным системам мониторинга безопасности, и далее осуществить все или некоторые из следующих действий: уничтожение частных данных, уничтожение всех данных, в том числе системных записей, необратимое выведение устройства из строя.

Другой аспект безопасности — программный. Загрузка системы должна начинаться из постоянного запоминающего устройства, которое потенциально считается безопасным и не подверженным атакам (следовательно, эта часть системы должна быть глубоко интегрирована в аппаратное обеспечение). Далее, такой доверенный загрузчик системы должен следить за тем, чтобы другое загружаемое ПО и сам образ загружаемой операционной системы были сертифицируемы. Кроме всего прочего, процесс сертификации также должен быть возможен между разными узлами, дистанционно.

управление узлом: узел должен обладать изолированным внутренним устройством, выполняющим мониторинг за состоянием системы. Такое устройство должно заниматься конфигурированием и управлением системой; также быть частью цепочки доверия узла для обеспечения обновлений системы.

сеть: сеть узлов должна предоставлять QoS для обеспечения достаточного качества обслуживания, должен осуществляться мониторинг потока данных на

предмет чувствительных к задержкам пакетов, то есть должна происходить приоритезация, и все это должно происходить именно на уровне узла.

Связь между узлами может осуществляться как с помощью проводного, так и беспроводного соединений. Выбор конкретного вида соединения зависит от условий применения и сценариев использования системы. Например, узлы в производственных помещениях могут быть соединены при помощи проводов, но например, система умного транспорта подразумевает беспроводные соединения.

И в случае с проводными, и в случае с беспроводными соединениями существует множество стандартов, которые могут быть использованы. К проводным соединениям можно отнести, например, Ethernet, к беспроводным — Wireless WAN (WWAN), Wireless LAN (WLAN), and Wireless Personal Area Networks (WPAN).

ускорители: в случае с узлами, основным предназначением которых является сбор и анализ данных, могут потребоваться дополнительные ускорители, такие как GPU(графический процессор) или FPGA(Программируемая пользователем вентильная матрица). Графический процессор, например, обладает большим количеством простых ядер, что позволяет пользоваться возможностью параллелизации вычислений.

Вычисления и память: в случае с вычислениями необходимо учитывать, что их осуществление должно быть возможно в достаточно суровых условиях, предполагающих, например, высокие температуры ($\sim 100^{\circ}\text{C}$). Что касается памяти, в большинстве случаев будут применяться твердотельные накопители по причине их надежности и малого энергопотребления.

Система

Следующий уровень абстракции — системный. Заинтересованными лицами на данном уровне являются системные архитекторы, производители аппаратных платформ... На данном уровне рассматривается система из нескольких узлов, образующих платформу для конкретного сценария использования.

Одним из самых основных аспектов данного уровня является виртуализация аппаратных ресурсов, предоставляемых узлами.

Программное обеспечение

Здесь рассматриваются аспекты, касающиеся программного обеспечения, работающего на системе из нескольких узлов. Основные заинтересованные лица: интеграторы систем, архитекторы ПО, разработчики приложений. Данный уровень, в свою очередь, подразделяется на несколько подуровней:

предоставления услуг — реализует специфические услуги\функции определенного сценария применения. Это, по сути, является уровнем, где расположено прикладное программное обеспечение.

поддержки — не имеет отношения к прикладным приложениям, основной целью является поддержка приложений, запущенных на подуровне предоставления услуг.

управления узлом и объединительного программного обеспечения — предоставляет возможность запуска программного обеспечения на узле и обеспечивает сетевое взаимодействие между разными узлами.

Объединительное программное обеспечение включает в себя ядра операционных систем, запускающихся поверх виртуализированных ресурсов узлов, интерфейсы для сетевого взаимодействия между узлами; также предоставляет функцию программной виртуализации и контейнеризации.

Контейнеризация подразумевает, что несколько приложений работают в одной операционной системе, но являются изолированными с точки зрения памяти и процессорных ресурсов.

На данном подуровне также происходит взаимодействие между узлами и умными вещами, следовательно, именно здесь должна обеспечиваться конфиденциальность передаваемых данных. Здесь должны решаться и другие проблемы, связанные с безопасностью, например, когда разные узлы обнаруживают друг-друга и устанавливают соединение в первый раз, должен произойти процесс сертификации.

Управление узлом подразумевает поддержку узлов, а следовательно и всей системы, как с точки зрения ПО, так и с точки зрения аппаратного обеспечения, в

желаемом состоянии. Это в большой степени относится к требованиям о Надежности, Доступности и Обслуживаемости.

Подуровень поддержки предоставляет общие услуги всем приложениям, находящимся на уровне выше. Такими услугами могут быть среды выполнения, такие как JVM или Node.js; базы данных; утилиты для кодирования и декодирования данных; различные разделяемые библиотеки; Данные услуги могут также быть контейнеризированы, и при этом предоставлять услуги при помощи интерфейсов, определяемых контейнерами.

И наконец, подуровень предоставления услуг. Данные подуровень определяется конкретными сценариями использования, и не смотря на это, этот уровень тоже можно разделить на составные части.

Первая такая часть — сервис взаимодействия с периферийными устройствами. Данный сервис предоставляет интерфейсы для общения с сенсорами и другими подобными «низкоуровневыми» устройствами.

Вторая — основной сервис(core service). Функцией такого сервиса является с одной стороны, предоставление удобного доступа более высокоуровневых систем к собираемым от сенсоров данным, с другой — предоставление высокоуровневым системам более простого способа управления низкоуровневыми устройствами. т.е по сути, этот сервис является неким «переводчиком» между физическими устройствами и всей остальной частью системы.

Третья часть — сервис поддержки, он реализует рутинные функции, такие как ведение журнальных записей, регистрация сервисов, и.т.д

Следующая часть — сервис аналитики. Именно здесь происходит анализ собранных данных на узлах. Аналитика может быть разной, в зависимости от ресурсов конкретного узла. В одних случаях это простой анализ для выявления отклонений значений от их норм, а в других — предсказательная анализ, с привлечением технологий машинного обучения.

Предпоследняя часть — интеграционный сервис. Данный сервис определяет способы передачи данных между узлами, т.е, с помощью данного сервиса два узла, желающие обмениваться данными, определяют как, куда, когда и в каком формате эти данные будут отправляться.

Последняя часть — сервис пользовательского интерфейса, призванный отображать необходимые данные в удобном для пользователя виде.

Очевидно, что описанные части расположены друг относительно друга в иерархической структуре. Другое замечание заключается в том, что наличие всех этих частей в узле не обязательно, например, сервис пользовательского интерфейса необходим не во всех случаях.

4. Электрическая сеть на основе системы туманных вычислений

В заключение, рассмотрим модель электрической сети, предложенной в источнике [5]. Такая электрическая сеть, согласно авторам, будет обладать некоторыми преимуществами по сравнению с обычными сетями: у пользователей появится возможность лучшего отслеживания и контроля количества используемой энергии, также будет возможно распространения электроэнергии от источников возобновляемой энергии в более широком масштабе (утверждается, что используемые сейчас электрические сети обычно распределяют электроэнергию от нескольких станций, в число которых не входят источники возобновляемой энергии); ожидается, что подобный подход, обеспечивающий эффективное распределение и использование электроэнергии также будет иметь положительное воздействие на проблему глобального потепления.

Идея заключается в том, чтобы использовать двусторонние линии передач электроэнергии, способные также обеспечивать передачу информации между пользователями и провайдерами электроэнергии. Так, появится возможность обеспечения пользователей информацией о потреблении электроэнергии почти в реальном времени. Этим будет заниматься Advanced Metering Infrastructure (AMI) — одна из архитектурных частей системы. Другими частями системы являются Home Area Networks (HANS), Business Area Networks (BANs), Neighborhood Area Networks (NANs) и облачные вычислительные центры. Кроме того, данные, собираемы AMI, могут также включать приватную информацию, например, типы устройств-потребителей и время потребления.

Чем больше будет становиться такая сеть, тем больше данных будет по ней передаваться. В добавок к этому, есть необходимость обеспечения конфиденциальности использования собираемых данных, по этим причинам предлагается использовать СТВ.

Предлагаемая модель состоит из трех уровней:

Первый уровень включает в себя модули АМІ, различные умные устройства. На этом уровне модули АМІ собирают информацию о энергопотреблении и отправляют ее на следующий уровень. На втором уровне находятся специализированные узлы, собирающие данные от модулей АМІ и передающие их дальше, на облако. А третьим уровнем именно облако и является.

Первый уровень иерархии может обеспечивать общение между умными устройствами и модулями АМІ. Таким образом, если, например, электромобиль заряжается от одной из зарядных станций, не принадлежащих владельцу этого автомобиля, то модуль АМІ данной станции может отправить информацию о потреблении тому модулю АМІ, который принадлежит владельцу автомобиля. (Подобные возможности, пожалуй, являлись бы самыми заволашеванными, с точки зрения пользователей)

На втором уровне каждый узел имеет определенную область покрытия и взаимодействует лишь с определенной частью модулей АМІ. Самой важной функцией узлов на данном уровне является то, что они разделяют всю собираемую информацию на конфиденциальную и публичную. Первый тип информации хранится локально, причем временно. Второй — отправляется на облако. В результате такого разделения обеспечивается конфиденциальность хранения частных данных, также сокращаются задержки при попытке получения этих данных пользователями. Таким образом, здесь хранится подробная статистика об использовании электроэнергии, с информацией о типах устройств, времени, когда те осуществляли потребление и.т.д, а на облако будет отправляться информация, наподобие общего количества потребленной электроэнергии, без дополнительных деталей.

Третий уровень будет использоваться провайдерами электроэнергии для выставления счетов.

Такие свойства СТБ, как географическая распределенность, осведомленность о местонахождении, низкие сетевые задержки... используются в данной системе. Использование осведомленности о местонахождении видно, например, на примере с зарядкой электромобиля, а географическая распределенность — естественное свойство электрических сетей, сочетающееся со свойствами СТБ.

Рассмотрим теперь некоторые из «ключевых идей» предыдущей части в свете данного примера.

Безопасность: в предыдущем разделе мы сказали, что пользователи эталонной модели получают возможность сфокусироваться на обеспечении безопасности лишь специфических частей своих систем. В данном пример, такой специфической частью является приватная информация об использовании электроэнергии, а безопасность обеспечивается вторым уровнем узлов системы.

Масштабируемость: а в данной системе мы видим масштабируемость объема и масштабируемость производительности — со временем будут появляться новые части сети, при расширении городов; информация о потреблении электроэнергии, попадая в реальном времени к провайдерам, позволяет осуществлять эффективное распределение ресурса между частями сети.

Открытость: мы видели, как AMI узлы могут общаться друг с другом на примере с зарядкой электромобиля. Кроме всего прочего, можно предположить, что такое же взаимодействие придется осуществлять и на грани электрических сетей разных провайдеров, то есть, даже AMI модули разных провайдеров должны уметь взаимодействовать друг с другом.

Автономность: очевидно, что при недоступности облака, второй и первый уровни данной модели смогут продолжить функционирование.

5. Источники

- [1] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli, «Fog Computing and Its Role in the Internet of Things», 1st ACM SIGCOMM Workshop on Mobile Cloud Computing, pages 13-15, 2012
- [2] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. «Fog Computing: A Platform for Internet of Things and Analytics», volume 546 of Studies in Computational Intelligence, Springer International Publishing, Cham, Switzerland, pages 169-186, 2014.
- [3] Kevin Bachmann «Design and Implementation of a Fog Computing Framework», pages 7-22, 2017
- [4] OpenFog Consortium Architecture Working Group, «OpenFog Reference Architecture for Fog Computing», pages 13-88, 2017
- [5] Feyza Yildirim Okay, Suat Ozdemir «A fog computing based smart grid model» International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, pages 1-5, 2016
- [6] “Internet of Things Leaders Create OpenFog Consortium to Help Enable End to End Technology Scenarios for the Internet of Things” November 17, 2015 <https://www.openfogconsortium.org/news/internet-of-things-leaders-create-open-fog-consortium-to-help-enable-end-to-end-technology-scenarios-for-the-internet-of-things/>
[Accessed May 12, 2019]
- [7] OpenFog Consortium membership information page <https://www.openfogconsortium.org/membership-information/>
[Accessed May 12, 2019]

