

2. Система контроля функционирования;
 3. Система сбора и анализа статистических данных.
 4. Универсальный интерфейс программирования приложений (API).
- Примеры использования языка программирования Java при реализации внешних систем управления:
1. При реализации диалогового интерфейса язык программирования Java используется во всех вышеперечисленных системах управления. Таки образом реализуется мультиплатформенный интерфейс и минимизируются затраты на адаптацию к различным операционным системам.
 2. При разработке подсистем контроля функционирования и сбора и анализа статистических данных компании HP и Hitachi использовали JDBC для организации связи Java-приложений с базой данных. Именно эта библиотека входит в комплект JavaBeans и поставляется в комплекте средств разработки JDK. Эта библиотека обеспечивает Java-программам универсальное средство для работы с СУБД.
 3. Для работы с своим API компания Hitachi использует язык JavaScript, а компания HP в качестве базового языка для своего API выбрала язык программирования Java.

Таким образом мы видим, что язык программирования Java нашел очень широкое применение при реализации систем управления центрами обработки данных и полноценного конкурента в этом сегменте при создании мультиплатформенных приложений у него пока нет.

Список использованных источников:

1. HP Integrated Lights-Out. User Guide. - Part Number 238882-004 – 2003 – 249 с.
2. IBM System i and System p. Managing the Hardware Management Console (HMC) – 1998 – 132с.
3. Integrated Dell Remote Access Controller 7 (iDRAC7) Version 1.50.50 User's Guide – 2014 – 258 с.
4. BROCADE Fabric OS. Administrator's Guide - 53-1002920-02 – 2013 – 694 с.
5. Cisco Fabric Manager Fundamentals. Configuration Guide - OL-21502-02 – 2010 – 338 с.
6. HP Systems Insight Manager 7.5. User Guide - Part Number: 601823-402a – 2016 – 257 с.
7. IBM Tivoli Monitoring Administrator's Guide - SC22-5446-01
8. Hitachi Command Suite. Tuning Manager. Server Administration Guide - MK-92HC021-38 – 2015 – 280 с.

РАСПРЕДЕЛЕННЫЙ АНАЛИЗ КРИПТОВАЛЮТНОГО BLOKCHAIN

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ващук Д.В.

Самаль Д. И. – к.т.н, доцент

В последнее время криптовалюты приобретают все большую ценность, и все больше финансовых организаций, продавцов и потребителей обращают внимание на них. В первую очередь это связано с преимуществами, которые дают криптовалюты перед фиатной валютой: отсутствующие или мизерные комиссии, анонимность, возможность быстро и беспрепятственно отправлять «криптоденьги» в любой уголок мира. Разработчики и финансисты нуждаются в системах, которые бы позволили быстро находить определенную информацию в Blockchain. Но разработать такую систему не так просто, так как приходится постоянно работать с большими объемами данных в Blockchain. Задача ускорения анализа криптовалютного Blockchain на примере криптовалюты Bitcoin была решена в рамках данной работы.

Bitcoin — полностью распределенная платежная система, которая использует одноименную расчетную единицу — биткойн. Система предоставляет возможность осуществления полностью необратимых денежных транзакций, когда денежный перевод происходит без доверенной стороны-посредника, и ни одна из сторон не могла бы заблокировать, отменить или принудительно совершить транзакцию. Для обеспечения функционирования и защиты системы используются криптографические алгоритмы, но вся информация о транзакциях всегда доступна в открытом виде, так как не шифруется [1]. Запущенные на множестве узлов в сети Internet программы-клиенты соединяются между собой в одноранговую сеть. Заранее известны объем и время эмиссии новых биткойнов. Но распределяются они случайно среди тех, кто использует свое оборудование для расчета хеш-кодов блоков, которые являются специфичным механизмом регулирования и подтверждения легитимности операций в системе Bitcoin — метод Proof-of-work [2].

Распределенную базу блоков транзакций, впервые реализованную в криптовалюте Bitcoin, называют Blockchain. Блок транзакций — специальная структура для записи группы транзакций в системе Bitcoin и аналогичных ей. Чтобы транзакция считалась подтвержденной, ее формат и криптографическую подпись должны проверить и затем группу транзакций записать в блок. Блоки одновременно формируются множеством агентов системы — так называемых, майнеров. Удовлетворяющие критериям блоки отправляются в сеть, включаясь в распределенную базу блоков. Копия Blockchain одновременно хранится на множестве компьютеров и синхронизируется согласно правилам построения цепочки блоков [2, 3].

		Debit	Credit
Transaction e14768c1d648b98a52cb796af30af186140c5209a2fb53f1c8097db579f01cc0			
INPUTS			
Previous Output	Signature		
6120ceaab25cfee257... : 0	3046022100aaf227f9...	0.0145	
6eb36c1d347f8fdb6e... : 1	3046022100810c9d7a...	0.0923	
OUTPUTS			
Address	Spent		
1NqUaJrFeStshjad1bhrEFFzWSQw6JHbqv	<input checked="" type="checkbox"/>		0.0122
1FrtRypBwstUQ4X9KQdQByx6fWXLGGuPNT	<input checked="" type="checkbox"/>		0.0945
Transaction b6f4ec453a021ac561b01039f78e7168a653af176353c86d607343cc77e779b9			
INPUTS			
Previous Output	Signature		
e14768c1d648b98a52... : 0	30450221008a396b69...	0.0122	
OUTPUTS			
Address	Spent		
1HiMoMgBaAikFHgAt3M4YJt4p4HrnsiXu	<input type="checkbox"/>		0.001
1Q3Jw1wRxZyJ767mVrtEpBVTH49HNBA83V	<input checked="" type="checkbox"/>		0.0111

Рис. 1. Структура списка транзакций

На данный момент свободно распространяется не так уж много инструментов, с помощью которых можно проанализировать Blockchain и найти необходимую информацию. К задачам анализа Blockchain можно отнести: поиск списка переводов средств с определенного кошелька; поиск адресов кошельков, которые осуществляли транзакции на исследуемый адрес кошелька; сбор различного рода статистики по осуществлению денежных переводов в реальном времени. Как правило, ускорение анализа в найденных инструментах происходит благодаря предварительному индексированию Blockchain. Еще большего ускорения анализа можно достичь при распределении вычислений на несколько вычислительных ядер, а еще лучше – на несколько вычислительных машин. В контексте данной работы использовался функциональный язык программирования Elixir [4] и несколько одноплатных компьютеров с процессором ARMv7 Cortex-A и операционной системой Debian GNU/Linux. Общая структура кластера выглядит следующим образом:

- Несколько вычислительных узлов с большим объемом дискового пространства, предпочтительно SSD, используются для запуска *bitcoind* сервиса, который взаимодействует с остальными узлами Bitcoin сети, загружает и верифицирует блоки и транзакции, т.е. поддерживает Blockchain в актуальном состоянии. Помимо этого, данный сервис предоставляет RPC-JSON API для других приложений [1].
- Остальные узлы не нуждаются в больших объемах дискового пространства и используются для производства задач анализа блоков и транзакций, которые были прочитаны из Blockchain [3].
- Необходим еще один узел, который будет выступать в качестве супервизора. В задачи супервизора входит: формирование задач для каждого вычислительного узла и последующий сбор и объединение результатов.

Для полноценной работы с *bitcoind* JSON-RPC на языке Elixir необходимо было доработать Open source модуль Gold [5].

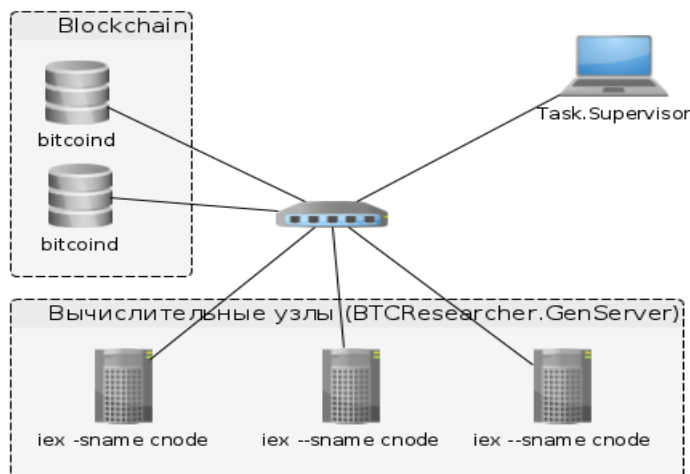


Рис. 2. Предлагаемая инфраструктура вычислительного кластера

Важной особенностью является то, что по умолчанию *bitcoind* индексирует транзакции только с *неиспользованными выводами* и транзакции, принадлежащие к текущему кошельку (если сервис собран с поддержкой функции кошелька, что не требуется в нашем случае). Неполная индексация приводит к тому, что некоторые JSON-RPC запросы *getrawtransaction* с корректным набором параметров могут завершиться с ошибкой. Индексация всех транзакций необходима для быстрого доступа к информации о любой транзакции по ее TXID. Для создания полного индекса транзакций необходимо запускать *bitcoind* с опцией *txindex* [1].

На каждом вычислительном узле запускается процесс виртуальной машины Beom VM командой *ix*. Для аутентификации узлов используется механизм *cookies*. Список узлов и *cookies* указываются в настройках супервизора.

Таким образом, описанное архитектурное решение дает существенный прирост в скорости анализа Blockchain и поиска необходимой информации в большом объеме данных. Ускорение происходит за счет распределения нагрузки на несколько вычислительных узлов и наличия в сети нескольких Blockchain, поддерживаемых в актуальном состоянии сервисом *bitcoind*.

Список использованных источников:

1. Bitcoin Developer Reference [Электронный ресурс]. – Режим доступа – <https://bitcoin.org/en/developer-reference>
2. Nakamoto, С. Bitcoin: A Peer-to-Peer Electronic Cash System [Электронный ресурс]. – Режим доступа – <https://bitcoin.org/bitcoin.pdf>
3. Так что же такое биткоин? [Электронный ресурс]. – Режим доступа – <http://bitnovosti.com/2014/07/17/tak-chto-zhe-takoe-bitcoin/>
4. Elixir: Getting started [Электронный ресурс]. – Режим доступа – <http://elixir-lang.org/getting-started>
5. Gold GIT repository [Электронный ресурс]. – Режим доступа – <https://github.com/denis4net/gold>

СИСТЕМА ПОЗИЦИОНИРОВАНИЯ И УПРАВЛЕНИЯ ПОЛОЖЕНИЕМ КВАДРОКОПТЕРА НА ПЛОСКОСТИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баранов Д. А.

Самаль Д. И. – канд. техн. наук, доцент

В современном мире наблюдается тенденция по все большей автоматизации рабочего процесса. В этом трудящимся помогают роботы, компьютеры и другие приборы, которые решают определенную задачу в автоматическом режиме почти без вмешательства людей.

Квадрокоптеры — это летательные аппараты, принцип передвижения в воздухе которых основан на вращающихся в противоположных направлениях парах винтов. Они уже нашли применение во многих сферах деятельности человека, таких как, например, охранная или транспортная. Цель данного исследования — расширить сферу использования квадрокоптеров за счет предоставления им возможности передвижения по плоскости.

Для решения поставленной задачи больше всего подходят квадрокоптеры X- и H-образного типа благодаря симметричному размещению винтов на корпусе. Математические модели движения таких квадрокоптеров уже хорошо изучены и описаны. Для того, чтобы такой дрон передвигался по плоскости, достаточно домножить всю систему уравнений, описывающую движение квадрокоптера в вертикальной и горизонтальной плоскости, на -1 . Тем самым, создается необходимая тяга, направленная в обратную сторону, нежели во время обычного взлета, что обеспечит прилипание дрона к поверхности, а сила трения между поверхностью и квадрокоптером не даст ему сползть вниз. Таким образом он может зафиксироваться на любой поверхности и продолжать находиться там на протяжении всего времени автономной работы (см. рисунок 1 (слева)).

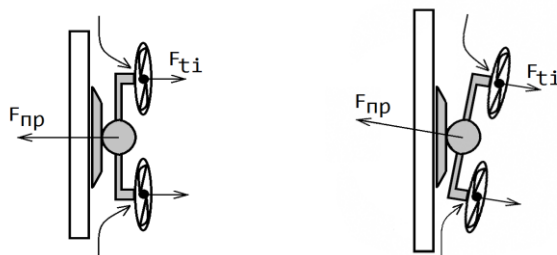


Рис.1 Направление сил, действующих на квадрокоптер в неподвижном режиме(слева) и при движении вверх(справа)