

## ИСПОЛЬЗОВАНИЕ ЯЗЫКА JAVA ДЛЯ ПОСТРОЕНИЯ МУЛЬТИПЛАТФОРМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ЦОД

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

Казак Н. П.

Искра Н. А. - магистр техн. наук, старший преподаватель

В современных ЦОД одной из главных задач является построение мультиплатформенной системы управления ресурсами. Требование мультиплатформенности для систем управления возникает потому, что серверный пул, часть ресурсов которого используется для функционирования систем управления, может быть реализован с использованием различных серверных платформ, функционирующих под управлением различных операционных систем. При решении данного класса задач одним из наиболее приоритетных инструментов является язык программирования Java.

Системы управления ресурсами подразделяются на два класса: внешние системы управления и интегрированные системы управления.

Можно выделить базовые требования к типовой интегрированной системе управления ресурсами:

1. Код должен выполняться на всех основных аппаратных платформах и работать в среде всех основных операционных систем.
2. Программное обеспечение системы управления должно запускаться на любом сервере либо компьютере без процедуры установки.
3. Программное обеспечение должно быть максимально изолировано, работа данного программного обеспечения не должна влиять на отказоустойчивость серверной платформы.
4. Программное обеспечение должно требовать для своей работы минимальное количество дополнительных программных продуктов.
5. Программное обеспечение системы управления должно поддерживать многопоточность для оптимального использования ресурсов мультипроцессорных многоядерных серверных систем.
6. Система управления должна взаимодействовать со всеми основными web-браузерами.

Рассмотрим преимущества языка программирования Java при реализации данного класса систем:

1. Язык программирования Java является мультиплатформенным, поэтому программное обеспечение написанное на нем выполняется на всех основных аппаратных платформах и может работать в среде всех основных операционных систем.
2. Программное обеспечение системы управления может выполняться на любом сервере. Единственным требованием является установка на данном сервере исполняющей системы Java, так называемой Java Runtime Environment (JRE).
3. Программное обеспечение реализованное на языке Java выполняется на обособленной виртуальной машине JVM и не взаимодействует напрямую с аппаратными ресурсами сервера и программными ресурсами операционной системы.
4. Для работы программного обеспечения написанного на языке Java требуется только установка Java Runtime Environment.
5. Одной из сильных сторон виртуальной машины Java является её способность работы с несколькими потоками. JVM оптимизирована для больших многоядерных машин, и она без проблем может управлять большим количеством потоков. Наиболее очевидная область применения многопоточности – это программирование интерфейсов систем управления.
6. Все основные web-браузеры имеют поддержку для работы с приложениями написанными на языке Java.

Перечислим основные задачи, решаемые с использованием языка программирования Java в данных системах:

1. Консоли управления серверами.
2. Модули доступа к внешним носителям информации.
3. Диалоговые модули управления подсистемами сервера.
4. Диалоговые модули управления инфраструктурой.

К данному классу систем относятся: HP Integrated Lights-Out, IBM Hardware Management Console (HMC), integrated Dell Remote Access Controller, Brocade SAN Fabric Manager, Cisco SAN switch manager и другие.

При реализации внешних систем управления есть определённые задачи, для решения которых использование языка программирования Java является наиболее оптимальным:

1. Построение универсального диалогового интерфейса, который функционирует под управлением различных операционных систем.
2. Реализация WEB-интерфейса приложений.
3. Реализация универсального интерфейса программирования приложений (API).
4. Реализация мультиплатформенных модулей сбора и анализа статистической информации.

Рассмотрим, как используется язык программирования Java при реализации таких систем на примере HP System Insight Manager, IBM Tivoli, Hitachi Tuning Manager.

Внешние системы управления можно разделить на несколько основных подсистем:

1. Диалоговая система управления;

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].