Таким образом, в ходе первичных экспериментов было показано, что с использованием режима пониженного энергопотребления можно осуществлять извлечение уникальных характеристик ДОЗУ, что является альтернативным подходом, не требующим выключения и включения всей цифровой системы. Кроме того, использование режима пониженного энергопотребления представляет интерес для мобильных устройств. Однако первичные данные говорят о низкой случайности данных, полученных данным методом, что не дает использовать его для построения генераторов истинно случайных последовательностей.

Список использованных источников:

1. Tao, S. Reliable low-overhead arbiter-based physical unclonable functions for resource-constrained IoT devices / S. Tao, E. Dubrova // Proc. of the 4th Workshop on Cryptography and Security in Computing Systems. – Stockholm, Sweden, 2017. – P. 1–6.

2. Rosenblatt, S. A Self-Authenticating Chip Architecture Using an Intrinsic Fingerprint of Embedded DRAM / S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, S. S. Iyer // IEEE Journal of Solid-State Circuits. – 2013. – Vol. 48, No. 11. – P. 2934-2943.

KUBERNETES КЛАСТЕР ДЛЯ ПОЛНОТЕКСТОВОГО ПОИСКА

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

Сенькович Д.С.

Жвакина А. В. – к. т. н, доцент

На сегодняшний день в разработке программного обеспечения используется все больше независимых взаимодействующих между собой приложений. Особенно остро проблема проявляется при разработке приложений по обработке больших данных. Целью проводимых исследований является создание кластера Docker контейнеров с Apache Hadoop и Apache Solr под управлением Kubernetes. Решение полезно как для обучения технологии полнотекстового поиска, так и при развертывании кластера для тестирования и разработки.

Задача поиска по тексту предполагает работу со сложными индексами и большими объемами данных. Поэтому для ее решения образовалась целая группа отдельных инструментов.

Многие базы данных имеют встроенную возможность поиска по тексту, однако обычно это очень ограниченная реализация. В большинстве случаев следует использовать более подходящие технологии. Как следствие, получили широкое распространение технологии полнотекстового поиска, например, Apache Solr и Elasticsearch.

Solr − платформа полнотекстового поиска с открытым исходным кодом, основанная на проекте Apache Lucene. Ее основные возможности: полнотекстовый поиск, подсветка результатов, фасетный поиск, динамическая кластеризация, интеграция с базами данных, обработка документов со сложным форматом (например, Word, PDF). Так как в Solr есть возможность распределенного поиска и репликации, Solr хорошо масштабируем. По состоянию на май 2016 года Solr является вторым по популярности поисковым движком, уступая Elasticsearch.

Solr и Elasticsearch являются лидерами на рынке полнотекстового поиска. Более популярным решением является Elasticsearch ввиду простоты начала работы с ним. Solr в то же время имеет большую историю и является решением с открытым исходным кодов, также как и его экосистема. Обе системы богаты своими возможностями и не уступают друг другу по скорости, используя один движок для полнотекстового поиска Apache Lucene. Принимая во внимание, что и движок, и Apache Solr разрабатываются одной команды, можно предполагать лучшую интеграцию и использование возможностей движка. Из различий также можно отметить более долговременную поддержку существующих возможностей в Apache Solr и возможность использовать встроенное машинное обучение, что возможно только в платной версии Elasticsearch.

Solr использует Apache HDFS для хранения документов и индексов, что обеспечивает сохранность и быстрый доступ.

HDFS (Hadoop Distributed File System) -- файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера. Все блоки в HDFS (кроме последнего блока файла) имеют одинаковый размер, и каждый блок может быть размещён на нескольких узлах, размер блока и коэффициент репликации (количество узлов, на которых должен быть размещён каждый блок) определяются в настройках на уровне файла. Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов. Файлы в HDFS могут быть записаны лишь однажды (модификация не поддерживается), а запись в файл в одно время может вести только один процесс. Организация файлов в пространстве имён − традиционная иерархическая: есть корневой каталог, поддерживается вложение каталогов, в одном каталоге могут располагаться и файлы, и другие каталоги.

В связи с увеличением количества компонентов (программ) при разработке программного обеспечения установка и мониторинг, подготовка среды для предложения становятся сложнее. Поэтому активно развиваются технологии виртуализации, в частности, контейнеризации - виртуализации уровня ОС. Наиболее популярным решением является Docker.

Docker − это инструмент, предназначенный для упрощения создания, развертывания и запуска приложений с помощью контейнеров. Контейнеры позволяют разработчику упаковать приложение со всеми необходимыми ему компонентами, такими как библиотеки и другие зависимости, и поставлять все это как один пакет. Таким образом, благодаря Docker, разработчик может быть уверен, что приложение будет

работать на любой другой машине Linux, независимо от каких-либо пользовательских настроек, которые могут быть у машины, которые могут отличаться от машины, используемой для написания и тестирования кода.

Docker помогает в решение унификации использования программного обеспечения, но остается необходимость мониторинга и поддержки самих контейнеров Docker, в чем помогает Kubernetes.

Kubernetes, или k8s, или ”kube"− открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями. Оригинальная версия разработана компанией Google. Впоследствии Kubernetes был передан под управление Cloud Native Computing Foundation. Предназначение Kubernetes − предоставить «платформу для автоматического развёртывания, масштабирования, управления приложениями на кластерах или отдельных хостах». Kubernetes группирует контейнеры, составляющие приложение, в логические блоки для упрощения управления и обнаружения. Kubernetes устраняет многие ручные процессы, связанные с развертыванием и масштабированием контейнерных приложений. Другими словами, Kubernetes позволяет объединять группы хостов, работающих под управлением контейнеров Linux, и легко и эффективно управлять этими кластерами. Такие кластеры могут охватывать узлы в общедоступных, частных или гибридных облаках.

Kubernetes поддерживает различные технологии контейнеризации, включая Docker, VMWare и ряд других. Kubernetes является наиболее популярным решением по сравнению с Docker Swarm и Nomad ввиду своего функционала и расширяемости, несмотря на сложности и нужный уровень технической подготовки для работы с ним. Целью исследования является построение масштабируемого отказоустойчивого кластера контейнеров Docker, поэтому использоваться будет Kubernetes.

В работе Apache Solr и Apache Hadoop (в частности, используемая его часть HDFS) помещены в Docker контейнеры для унификации и упрощения работы с ними. Docker контейнеры сгруппированы в Kubernetes кластер для упрощения управления и мониторинга. Таким образом, решение представляет собой конфигурируемый кластер полнотекстового поиска, удобный для обучения и разработки.

Список использованных источников:

1. Хайлоад. Полнотекстовый Поиск. [Электронный ресурс]. – Режим доступа:https://ruhighload.com/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9+%D0%BF%D0%BE%D0%B8%D1%81%D0%BA - Дата доступа: 24.03.2018.

2. Apache Solr. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Apache_Solr. – Дата доступа: 25.03.2017.

3. Docker Cookbook – O'Reilly Media, 2015. [Электронный ресурс]. – Режим доступа: http://shop.oreilly.com/product/0636920036791.do – Дата доступа: 25.03.2017.

4. Scaling Big Data with Hadoop and Solr - Second Edition – Packt, 2015. [Электронный ресурс]. – Режим доступа: http://shop.oreilly.com/product/0636920036791.do – Дата доступа: 25.03.2017.

УЛУЧШЕНИЕ КАЧЕСТВА ОЦЕНКИ АРХИТЕКТУРЫ ПО С ПОМОЩЬЮ ИСПОЛЬЗОВАНИЯ SAAM МЕТОДОЛОГИИ

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

Сорокина А.Г.

Волорова Н.А. – к.т.н., доцент

В настоящее время архитектура программного обеспечения является объектом повышенного внимания, как в академических кругах, так и в промышленной разработке. Но, несмотря на популярность этой темы, достаточно мало внимания уделяется методам анализа разработанной архитектуры, которые могут показать, что она удовлетворяет определенным свойствам.

На большинстве крупных проектов особенно из сферы финансов, обороны и медицины проводят рецензирование разработанного дизайна, которое является важным компонентом процесса создания программного обеспечения. Тем не менее, такие проверки часто проводятся не постоянно и многие участники процесса считают их напрасной тратой времени, так как они не выявляют всех проблем и дефектов, которые впоследствии выявятся только в процессе разработки, что уже слишком поздно.

Эта проблема появляется вследствие того, что в большинстве случаев сложно объективно оценить заявления разработчика о качествах, присущих созданной программной архитектуре. Примерами таких заявлений являются фразы на подобие [1]:

- Мы разработали компоненты, которые можно перенастроить с минимальными усилиями.
- Этот метод поощряет такое разделение приложения на части, которое уменьшит стоимостьпоследующих модификаций системы.
- Этот подход лучше, чем традиционные подходы с точки зрения модульности и повторного использования кода.

Трудность в оценки этих требований возникает по двум причинам. Во-первых, различные архитектурные решения не используют общий словарь. А когда разрабатываются новые архитектуры, то они обычно разрабатывают новые термины для описания или используют старые термины по-новому. Во-вторых,