

APPLYING NEW VIRTUALIZATION TECHNOLOGIES TO THE LEARNING PROCESS

Zurab Gasitashvili, Zurab Modebadze, David Satsradze^a

Georgian Technical University, Georgia, Tbilisi, 0160, Kostava str. 77

E-mail: ^ad_satsradze@gtu.ge

The active adoption of virtualization methods in existing IT infrastructures and the transition to the use of GRID and cloud computing requires new personnel who own these technologies. The article proposes an approach for creating a training ground using new methods of virtualization of computing resources. Using the proposed approach on the basis of existing computer technology, it is possible to create a failover cluster, inside which already configured virtual environments are launched in the corresponding containers for various computing tasks, as well as for organizing the training process.

Keywords: GRID, cloud computing, virtualization, cluster, Mesos, containers, Docker, virtual learning environments.

Zurab Gasitashvili, Zurab Modebadze, David Satsradze

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ПРИМЕНЕНИЕ НОВЫХ ТЕХНОЛОГИЙ ВИРТУАЛИЗАЦИИ В УЧЕБНОМ ПРОЦЕССЕ

З. Гаситашвили¹, З. Модебадзе¹, Д. Сацерадзе^{1,a},

¹ *Грузинский Технический Университет, Грузия, Тбилиси, 0160, ул. Костава 77*

E-mail: ^a d_satseradze@gtu.ge

Активное внедрение методов виртуализации в действующие IT-инфраструктуры и переход на использование GRID и облачных вычислений требует новых кадров, владеющих этими технологиями. В статье предложен подход для создания учебного полигона с применением новых методов виртуализации вычислительных ресурсов. С помощью предложенного подхода на базе существующей компьютерной техники возможно создание отказоустойчивого кластера, внутри которого в соответствующих контейнерах запускаются уже настроенные виртуальные среды как для разных вычислительных задач, так и для организации процесса обучения.

Ключевые слова: GRID, облачные вычисления, виртуализация, кластер, Mesos, контейнеры, Docker, виртуальные среды обучения.

Zurab Gasitashvili, Zurab Modebadze, David Satseradze

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Введение

Как известно существует дефицит специалистов способных обслуживать как уже существующие распределенные вычислительные системы (РВС), так и разрабатывать новые.

Основная проблема, с которой сталкиваются при освоении технологий распределённых вычислений — это проведение практических занятий. Учитывая ограниченные возможности бюджета ВУЗов, сравнительно высокую стоимость оборудования для создания соответствующей вычислительной инфраструктуры и/или доступные мощности электропитания-кондиционирования и их ежемесячной оплаты, становится понятным почему у ВУЗов не всегда есть возможность организовать соответствующий учебный процесс, во время которого каждому учащемуся можно будет выделить инфраструктуру, состоящую из нескольких вычислительных машин.

С целью подготовки соответствующих специалистов в рамках образовательной программы «Компьютерные Науки» в Грузинском Техническом Университете на базе «Лаборатории Искусственного Интеллекта и обработки Больших Данных» планируется создание многофункционального полигона с использованием методологии виртуализации. Предполагается его взаимодействие с Вычислительным Облаком Университета, созданного при активном участии Объединённого Института Ядерных Исследований для проведения совместных научных исследований. Кроме этого, в реализации полигона и в процессе обучения по программе «Компьютерные науки» будет учтен богатый опыт сотрудников ЛИТ ОИЯИ.

2. Учебный полигон

Полигон представляет собой шесть одинаковых рабочих станций с операционной системой CentOS 7, которые после соответствующего «апгрейда» и закупки сетевого оборудования будут объединены в отказоустойчивый кластер с помощью Apache Mesos.

Суть работы Apache Mesos сводится к объединению физических ресурсов нескольких компьютеров (процессор, память, жесткие диски) в единый виртуальный пул, после чего указанные ресурсы будут выделяться для выполнения конкретных задач используя методы контейнеризации [4].

После развертывания кластера в роли его ядра будут выступать три управляющих компьютера (узла) т. н. Mesos Masters, а на оставшихся трех узлах (Mesos Slaves) будут выполняться различные задачи в соответствии с программами обучения. Согласно архитектуре Apache Mesos, узлы Masters являются ведущими, и они отвечают за предоставление ресурсов, распределения задач между действующими ведомыми узлами (Mesos Slaves). Взаимодействие между Ведущими (Masters) узлами осуществляет сервис (демон) ZooKeeper, который проводит выборы лидера. Количество трёх управляющих компьютеров (их может быть и больше, но оно должно быть нечетным) обусловлено необходимостью наличия кворума для голосования и обеспечением требования отказоустойчивости кластера. В случае выхода из строя действующего лидера, ZooKeeper проводит выборы и избирает нового лидера из двух оставшихся узлов. Общая архитектура и схема работы кластера показано на *Функционирование кластера* осуществляется по следующему алгоритму:

1. Служба ZooKeeper проводит выборы лидера.
2. Ведомые устройства (Mesos Slaves) публикуют доступные ресурсы для ведущих устройств (Mesos Master-ов).
3. Ведущие устройства (Mesos Masters) отправляет предложения о наличии ресурсов специальным приложениям (Фреймворкам).
4. Планировщик (Scheduler) выделяет соответствующие ресурсы, необходимые для каждой задачи(Task).
5. Ведущие узлы (Mesos Master-ы) отправляют задания ведомым устройствам.

3. Механизм контейнеров

Каждая задача будет запускаться в отдельном изолированном контейнере, что обеспечит целостность обрабатываемых данных, так как программы из разных контейнеров не могут воздействовать друг на друга. Контейнеры будут реализованы с помощью механизма Docker, который будет запущен на трех ведомых узлах (Mesos Slave) кластера.

Основанная на открытом исходном коде контейнерная технология Docker была опубликована в 2013 году под лицензией Apache 2.0 и представляет собой новый метод виртуализации. В отличие от аппаратной виртуализации с применением гипервизора, контейнеризация представляет собой виртуализацию на уровне ядра операционной системы и не зависит от оборудования, что позволяет существенно экономить ресурсы и увеличивать эффективность работы с приложениями. Этот тип виртуализации позволяет упаковывать программное обеспечение по изолированным средам — контейнерам. Каждый из этих виртуальных блоков содержит все нужные элементы для работы приложения. Это дает возможность одновременного запуска большого количества контейнеров на одном хосте. Кроме минимального потребления ресурсов, Docker позволяет:

- увеличить скорость развертывания — вспомогательные компоненты можно не устанавливать, а использовать уже готовые docker-образы (шаблоны);
- скрывать процессы и работать с небезопасным кодом;
- осуществлять простое масштабирование, оптимизировать файловую систему.

Главным компонентом, Docker-а является Docker-демон (Docker-daemon) — сервер контейнеров. Демон управляет Docker-объектами (сети, хранилища, образы и контейнеры), а также может связываться с другими демонами для управления сервисами Docker.

Другими компонентами являются:

- Docker-клиент (Docker-client / CLI) — интерфейс взаимодействия пользователя с Docker-демоном;
- Docker-образ (Docker-image) — файл, включающий зависимости, сведения, конфигурацию для дальнейшего развертывания и инициализации контейнера, которые будут формироваться на основе учебных программ;
- Docker-файл (Docker-file) — описание правил по сборке образа, в котором первая строка указывает на базовый образ;
- Docker-контейнер (Docker-container) — это легкий, автономный исполняемый пакет программного обеспечения, который и будет передаваться учащимся для проведения лабораторных и практических работ;
- Том (Volume) — эмуляция файловой системы для осуществления операций чтения и записи.
- Реестр (Docker-registry) — зарезервированный компьютер, используемый для хранения docker-образов.

Прежде чем запустить контейнер, пользователь с помощью клиентского интерфейса отдает команду Docker-демону (Docker-daemon) на создание Docker-образа (Docker-image) используя Docker-файл (Docker-file). Демон создает образ или скачивает его из публичного (Docker Hub) или из частного реестра т.е. с внешних репозиторий.

Docker-образ (Docker-image) — шаблон только для чтения (read-only) с набором некоторых инструкций, предназначенных для создания контейнера. Он состоит из слоев, которые Docker комбинирует в один образ при помощи вспомогательной файловой системы UnionFS. Так решается проблема нерационального использования дисковой памяти. Параметры образа определяются в Docker-file. Если требуется файл, то скачиваться будут только нужные слои. Например, когда нужно доработать программное обеспечение или модифицировать образ, изменив несколько файлов. После загрузки на сервер будут отправлены слои, содержащие только модифицированные данные.

Когда необходимо централизованно управлять несколькими контейнерами, которые задействованы в общем проекте, то применяют Docker Compose. Это пакетный менеджер, который разработчики рекомендуют использовать во время проектирования сложных программных продуктов состоящих из множества процессов и сервисов. Для объединения

множества Docker хостов в один виртуальный хост служит Docker Swarm – средство развертывания кластера Docker, который можно использовать для быстрого развертывания кластера хостов как на локальном компьютере, так и на поддерживаемых облачных платформах. Для этого Docker Swarm предоставляет REST API интерфейс, совместимый с Docker API.

Метод создания информационных систем из множества контейнеров, каждый из которых отвечает только за одну определенную задачу, а «общение» осуществляется через сетевые порты и общие каталоги называется оркестрацией.

Для учебного полигона выбор Apache Mesos в качестве оркестратора вместо Docker Swarm-а обусловлено необходимостью запускать большее количество контейнерных кластеров в процессе обучения.

4. Возможности полигона

Наличие полигона с использованием кластера и технологии контейнеризации с ее потенциалом позволят в разы повысить эффективность решения широкого спектра учебно-методических задач, и будет хорошей опорой в осуществлении научно-исследовательской деятельности.

Для проведения лабораторных и практических занятий к учебному полигону студенты будут подключаться по сети из компьютерных классов общего назначения согласно составленному расписанию и в соответствии со своей учебной программой и политикой доступа. Во время обучения студенты последовательно учатся:

- процессу создания и настройке кластера на базе Apache Mesos;
- организации Docker-хостов на которых будут создаваться, запускаться и настраиваться соответствующие Docker-контейнеры;
- работе с различными приложениями, которые запускаются в контейнерах;
- технологиям создания и настройки систем визуального мониторинга и контроля ресурсов полигона.

Полигон можно будет использовать как облачную платформу во время проведения лабораторных работ и практических занятий по предметам SCADA системы, Интернет Вещей, а также других IoT-решений таких как мобильные приложения, «умные» системы, «умный дом», «умное предприятие», «умный город» и т. д.

С помощью запущенных контейнерах можно изучать вопросы, связанные с компьютерным моделированием, экологией, изменением климата, медицины, физики. Кроме этого можно исследовать искусственные нейронные сети и проводить машинное обучение и т.д. Одним словом, круг изучаемых вопросов и спектр возможностей полигона довольно обширен.

Интерес вызывает возможность взаимодействия полигона с Вычислительным Облаком университета, работающего на платформе Open Nebula. Для этого будет создан отдельный контейнер, который предоставит определенную часть свободного дискового пространства полигона для хранения резервных копий виртуальных машин.

5. Заключение

Предложенный в статье подход построения учебного полигона соответствует как современным тенденциям развития IT индустрии, так и учебным планам. В связи с неизбежным ростом доли дистанционного образования в системе обучения студентов, наличие такого полигона повышает удобство и качество образования, подымает престиж университета и является неплохой базой для подготовки/переподготовки действующих IT-специалистов начального звена.

Литература

- [1]. Кирсанова А.А., Радченко Г.И., Черных А.Н. – «Обзор технологий организации туманных вычислений» // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 9, № 3. С. 35–63. DOI: 10.14529/cmse200303.
- [2]. Филимонов А. Ю., Аксёнова К. А., Климова А. С., Колодов С. Д. – «Построение лабораторного стенда для исследования программно конфигурируемых сетевых инфраструктур» // Известия ВУЗОВ. Приборостроение. 2020. Т. 63, №2, стр. 178-186, ISSN:00213454.
- [3]. Лисецкий Ю.М., Саблий Ю.Ю – «Гиперконвергентная Технология С Открытым Кодом» // ISSN 1028-9763. Математичні машини і системи, 2019, № 1, стр.49-55.
- [4]. Грушин Д.А, Кузюрин Н. Н. – «Задачи оптимизации размещения контейнеров MPI-приложений на вычислительных кластерах» // Труды ИСП РАН, том 29, вып. 6, 2017, стр. 229-244.
- [5]. Государев И. Б. – «Развертывание и интеграция инновационных сред: бордакстинг, облачные хостинги и EDX» // Компьютерные инструменты в образовании. №1, 2014, стр.26-36.