Создание виртуальных кластеров Apache Spark в облачных средах с использованием систем оркестрации[1]

[1]О. Д. Борисенко <al@somestuff.ru>
[1]Р. К. Пастухов <pastukhov@ispras.ru>
[1,2,3]С. Д. Кузнецов <kuzloc@ispras.ru>
[1]Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, дом 25
[2]Московский государственный университет имени М.В. Ломоносова,
119991 ГСП-1 Москва, Ленинские горы, МГУ имени М.В. Ломоносова, 2-й учебный корпус, факультет ВМК
[3]Московский физико-технический институт,
141700, Московская область, г. Долгопрудный, Институтский пер., 9

Аннотация. Apache Spark является одним из наиболее производительных распределенных фреймворков для обработки больших данных в парадигме Map-Reduce. С распространением облачных технологий и предоставления ресурсов по запросу все более актуальной становится задача построения виртуальных вычислительных кластеров для конкретной задачи. В работе представлен краткий обзор разработанного решения для создания виртуальных кластеров Apache Spark в облачной среде Openstack и подведение итогов исследования о способах создания виртуальных кластеров Apache Spark в открытых облачных средах. Решение построено с использованием системы оркестрации Ansible. В работе будет проведено качественное сравнение разработанных в ИСП РАН подходов к решению задачи.

Ключевые слова: Apache Spark; Openstack; Amazon EC2; Map-Reduce; HDFS; виртуальные кластеры; облачные вычисления; Big Data; Apache Ignite.

DOI: 10.15514/ISPRAS-2016-28(6)-8

[1] Работа выполнена при поддержке гранта РФФИ No14-07-00602 А «Исследование и разработка методов автоматизации масштабирования и разворачивания виртуальных кластеров для обработки сверхбольших объёмов данных в облачной среде Openstack».

Для цитирования: Борисенко О.Д., Пастухов Р.К., Кузнецов С.Д. Создание виртуальных кластеров Apache Spark в облачных средах с использованием систем оркестрации. Труды ИСП РАН, том 28, вып. 6, 2016 г., стр. 111-120. DOI: 10.15514/ISPRAS-2016-28(6)-8

1. Введение

Проект Apache Spark [1] является одной из наиболее развитых и производительных [2] реализаций подхода Map-Reduce [3]. В то же время данный проект является частью инфраструктуры Apache Software Foundation для обработки больших данных и обладает возможностями совместной работы с другими проектами этой инфраструктуры (такими как Apache Hadoop [4], YARN [5], Mesos [6], Ignite [7] и другими). Каждое из перечисленных программных решений масштабируется с использованием распределенного режима работы.

Однако настройка каждого из решений в распределенном окружении является очень трудоемкой задачей, требующей глубокого понимания принципов работы и точек взаимодействия каждой из систем. Кроме того, настройка взаимодействия инструментов анализа больших данных в облачной среде в ручном режиме является экономически неоправданной, поскольку облачные среды предоставляют ресурсы по запросу с оплатой за время использования ресурса. При таком подходе ручная настройка виртуального кластера обладает сразу несколькими недостатками: во-первых, пользователь системы должен оплачивать фактический простой вычислительных ресурсов; причем чем больше вычислительный кластер, тем дольше происходит настройка без средств автоматизации процесса. Во-вторых, пользователь вынужден уничтожать виртуальный кластер после расчетов, поскольку оплата берется не за фактически использованные, а за зарезервированные ресурсы. В этом случае при появлении новых задач кластер необходимо настраивать заново.

Предлагаемые в работе подходы позволяют избежать процесса настройки виртуальных кластеров в открытой облачной среде и избежать траты временных и физических ресурсов множества исследователей. В качестве облачной среды выбран проект Openstack [8] как наиболее динамически развивающийся и предоставляющий наиболее широкий спектр возможностей среди аналогов.

2. Построение решения

На момент начала исследований в 2014 году существовало ровно одно решение для автоматизации создания виртуальных кластеров Apache Spark. Оно было предложено разработчиками Apache Spark и было совместимо исключительно со средой Amazon EC2 [9]. Настройка окружения подразумевала использование заранее настроенного образа виртуальной

машины (AMI) на базе Red Hat Enterprise Linux 5.3. Предлагаемый разработчиками Apache Spark способ обладал следующими недостатками:

- решение использовало API закрытой облачной платформы Amazon EC2 и было непереносимым на другие облачные среды;
- использование заранее настроенного образа системы существенно снижало выбор компонентов пользователем;
- базовый образ на основе RHEL 5.3 ограничивал использование образа пользователем с лицензионной точки зрения и мог использоваться только в среде Amazon;
- среда Amazon EC2 не позволяла выгружать образы виртуальных машин из собственной среды.

В ходе исследований были изучены основные подходы к созданию виртуальных кластеров в Openstack.

Первый подход [10] заключался в переносе функциональности решения от разработчиков Apache Spark для Amazon EC2 на использование API Openstack, исследовании взаимосвязей компонентов и построении аналогичного решения, но с использованием образов операционных систем без каких-либо предустановленных компонентов (с целью воспроизводимости). Решение было основано на клиентских библиотеках Openstack, библиотеки paramiko для обеспечения защищенных соединений с управляемыми узлами и реализовано на языке Python.

Второй подход [11] заключался в прямой интеграции с внутренней экосистемой облака Openstack. В октябре 2014 года произошел первый публичный релиз подсистемы Openstack Sahara [12]. Система поддерживала возможность создания виртуальных кластеров с дистрибутивами Apache Hadoop от вендоров Cloudera, Hortonworks, MapR (т.е. в версиях, отличных от версий Apache Software Foundation). В состав подобных кластеров входили некоторые дополнительные инструменты, например, Apache Hive, но интеграции с Apache Spark проект не предоставлял. В ходе исследований мы разработали набор компонентов для прямой интеграции Apache Spark с компонентами виртуальных кластеров, создаваемых Openstack Sahara, и кроме того, позволяющих использовать Apache Spark с использованием подхода PaaS. Это означает, что от пользователя скрывается внутренняя структура кластера и предоставлялся специальный слой управления, который позволяет исполнять программы Apache Spark через веб-интерфейс облачной среды или через REST API без прямого подключения к вычислительному кластеру. Также был интегрирован слой, позволяющий прозрачно использовать объектное хранилище Openstack Swift в качестве прозрачной замены распределенной файловой системы HDFS. Наше решение было включено в официальный релиз Openstack Liberty и присутствует в основной кодовой базе проекта Openstack с сентября 2015 года. Данный подход, тем не менее,

обладает рядом недостатков, которые будут рассмотрены в следующем разделе.

Третий подход заключается в использовании внешних средств оркестрации для создания виртуальных кластеров. Данный подход был реализован с использованием системы оркестрации Ansible и служебной программы, предоставляющей командный интерфейс для задания параметров кластера. В ходе разработки данного решения удалось разделить слои управления облачной средой, настройки компонентов операционной системы, настройки компонентов Apache Spark, настройки дополнительных инструментов и использовать необходимые комбинации этих. Среди дополнительных компонентов представлены Apache Hadoop, NFS, Apache Ignite, слой интеграции с Openstack Swift, Ganglia, Jupyter. Важно отметить, что явной привязки к дистрибутиву и версии операционной системы нет (на текущий момент поддерживаются Ubuntu 12.04, 14.04, 16.04; добавление RHEL-подобных дистрибутивов является сравнительно простой задачей). Решение попало в список одобренных [13] разработчиками Apache Spark в отличие от предыдущих подходов. Также стоит отметить, что за счет разделения сценариев, это решение сравнительно легко можно дополнить поддержкой любой другой облачной среды.

3. Сравнение решений

Каждый из описанных подходов (и их реализаций) обладает рядом достоинств и недостатков. Рассмотрим их по порядку.

Решение на основе Python и использования HTTP REST API Openstack

Достоинства:

- Первое подобное решение для открытых облачных сред.
- За счет использования REST API Openstack имеет минимальное количество зависимостей и работает быстрее остальных.

Недостатки:

- Из-за быстрой смены кодовой базы Openstack и изменения API работает только до версии Openstack Kilo и только с плоской организацией сетей на базе Nova Networking (т.е. нет поддержки Openstack Neutron: на момент реализации, Openstack Neutron назывался Openstack Quantum и не обладал стабилизированным API).
- Очень трудоемок процесс интеграции дополнительных инструментов для анализа данных (таких как Apache Hive, Ignite и т.п).

Решение, интегрированное в Openstack

Достоинства:

- Поддерживается большим сообществом и с момента релиза Openstack Liberty существует и развивается самостоятельно.
- Интегрировано с большим количеством инструментов для работы с большими данными.
- Предоставляет уровень абстракции “обработка данных как услуга” и не требует особенных знаний от исследователя при должной настройке облачной среды.
- Обеспечивается возможность при необходимости увеличивать размер вычислительного кластера.
- Хорошо подходит для промышленных окружений с налаженными сценариями вычислений.

Недостатки:

- Существенное отставание от актуальных версий Apache Spark и невозможность получить более новую версию без вмешательства в кодовую базу Openstack. Более того, одновременно предоставляется только две последних минорных версии Apache Spark на момент выпуска версии Openstack. Openstack выходит 2 раза в год, Apache Spark — 3-4 в зависимости от года.
- Использование дистрибутивов инструментов анализа данных от вендоров (отличаются как между собой, так и от версии сообщества)
- Процедура первичного создания шаблона вычислительного кластера требует вмешательства инженера, понимающего специфику вычислительной задачи и устройство Apache Spark.
- Существенно меньший контроль над ходом исполнения задач: подразумевается, что запущенная программа никогда не содержит ошибок; процесс нахождения ошибок чрезвычайно сложен.
- Требует дополнительных ресурсов для облачной среды: подразумевается, что этот сервис запущен постоянно на одном из контроллеров облака. Кроме того, у этого сервиса есть явные зависимости от других необязательных сервисов Openstack (например, Openstack Heat), из-за чего этот сервис сравнительно редко встречается в действующих облачных средах на основе Openstack.
- Процесс создания кластера является длительным процессом по сравнению с остальными решениями, как за счет устройства самой Openstack Sahara, так и за счет большого количества промежуточных абстракций, обеспечивающих интеграцию между инструментами обработки данных.

Решение на основе технологий оркестрации

Достоинства:

- Очень легко расширяется по сравнению с первыми двумя решениями.

- Поддерживает все существующие версии Apache Spark, начиная с 1.0 в любых поддерживаемых комбинациях с Apache Hadoop. Процесс добавления новых (или собственных) версий требует минимальных усилий.
- Сравнительно легко добавлять другие облачные среды в качестве инфраструктурной основы.
- Содержит множество дополнительных инструментов (Apache Hadoop, Apache Ignite, Jupyter, Ganglia, NFS, интеграция с Swift) и позволяет легко добавлять новые за счет слабой связности кодовой базы.
- Работает с любой версией Openstack и, наиболее вероятно, не устареет, поскольку основные библиотеки поддерживаются в том числе Openstack Foundation.
- Одобрено разработчиками Apache Spark.

Недостатки:

- Обладает меньшей гибкостью интеграции с дополнительными инструментами обработки больших данных за счет меньшей связности компонентов.
- Не предоставляет уровень “вычислений как сервиса”.
- Для более тонкой настройки системы, человек должен вмешиваться в шаблоны конфигурационных файлов или добавлять новые параметры к решению (например, для изменения степени репликации HDFS).

Каждый из походов предоставляет собственные преимущества, и их совокупность полностью покрывает возможные потребности исследователей.

4. Достигнутые результаты

В ходе проекта был полностью реализован весь спектр подходов по созданию виртуальных кластеров Apache Spark в среде Openstack, и два из трех решений получили поддержку сообщества. Все результаты опубликованы под открытой и свободной лицензией Apache 2.0 и доступны любому желающему как для использования, так и для модификации под свои нужды.

Список литературы

[1]. Shanahan J. and Dai L. Large Scale Distributed Data Science using Apache Spark. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). ACM, New York USA, pp. 2323-2324.

[2]. Li M., Tan J., Wang Y., Zhang L., Salapura V. SparkBench: a comprehensive benchmarking suite for in memory data analytic platform Spark. In Proceedings of the 12th ACM International Conference on Computing Frontiers (CF '15). ACM, New York USA, Article 53.

[3]. Jeffrey D., Sanjay G. MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.M. Bhandarkar, "MapReduce programming with

apache Hadoop," Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, Atlanta, GA, 2010, pp. 1-1.

[4]. Vavilapalli V., Murthy A., Douglas C., Agarwal S., Konar M., Evans R., Graves T., Lowe J., Shah H., Seth S., Saha B., Curino C., O'Malley O., Radia S., Reed B., Baldeschwieler E. Apache Hadoop YARN: yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (SOCC '13). ACM, New York USA, 2013, Article 5.

[5]. Страница проекта Apache Mesos: http://mesos.apache.org

[6]. Guller, Mohammed. Cluster Managers. Big Data Analytics with Spark. Apress, 2015. 231-242.

[7]. Dinsmore, Thomas W. In-Memory Analytics. Disruptive Analytics. Apress, 2016, pp. 97-116.

[8]. Sefraoui, Aissaoui O, Eleuldj M. OpenStack: toward an open-source solution for cloud computing. International Journal of Computer Applications 55.3, 2012.

[9]. Hazelhurst, Scott. Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud. Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology. ACM, 2008.

[10]. Борисенко О.Д., Лагута А.В., Турдаков Д.Ю., Кузнецов С.Д., Автоматическое создание виртуальных кластеров Apache Spark в облачной среде Openstack, Труды ИСП РАН, том 26, вып. 4, 2014 г., стр. 33-44. DOI: 10.15514/ISPRAS-2014-26(4)-4

[11]. Aleksiyants A., Borisenko O., Turdakov D., Sher A., Kuznetsov S. Implementing Apache Spark Jobs Execution and Apache Spark Cluster Creation for Openstack Sahara. Trudy ISP RAN/Proc. ISP RAS, vol. 27, issue 5, 2015, pp. 35-48. DOI: 10.15514/ISPRAS-2015-27(5)-3.

[12]. Ibrahim, Asmaa, Nawawy. A study of adopting big data to cloud computing. Technology Innovation and Entrepreneurship Center, Egypt Technology Innovation and Entrepreneurship Center, Egypt, 2015, pp. 1-7.

[13]. Список одобренных проектов, связанных с Apache Spark. https://cwiki.apache.org/confluence/display/SPARK/Third+Party+Projects

Deploying Apache Spark virtual clusters in cloud environments using orchestration technologies

[1]*Borisenko O. <al@somestuff.ru>*
[1]*Pastukhov R. <pastukhov@ispras.ru>*
[1,2,3]*Kuznetsov S. <kuzloc@ispras.ru>*
[1]*Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia
[2]*Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russia.
[3]*Moscow Institute of Physics and Technology (State University)*
9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia

Abstract. Apache Spark is a framework providing fast computations on Big Data using MapReduce model. With cloud environments Big Data processing becomes more flexible since they allow to create virtual clusters on-demand. One of the most powerful open-source cloud environments is Openstack. The main goal of this project is to provide an ability to create virtual clusters with Apache Spark and other Big Data tools in Openstack. There exist three approaches to do it. The first one is to use Openstack REST APIs to create instances and then deploy the environment. This approach is used by Apache Spark core team to create clusters in propriatary Amazon EC2 cloud. Almost the same method has been implemented for Openstack environments. Although since Openstack API changes frequently this solution is deprecated since Kilo release. The second approach is to integrate virtual clusters creation as a built-in service for Openstack. ISP RAS has provided several patches implementing universal Spark Job engine for Openstack Sahara and Openstack Swift integration with Apache Spark as a drop-in replacement for Apache Hadoop. This approach allows to use Spark clusters as a service in PaaS service model. Since Openstack releases are less frequent than Apache Spark this approach may be not convenient for developers using the latest releases. The third solution implemented uses Ansible for orchestration purposes. We implement the solution in loosely coupled way and provide an ability to add any auxiliary tool or even to use another cloud environment. Also, it provides an ability to choose any Apache Spark and Apache Hadoop versions to deploy in virtual clusters. All the listed approaches are available under Apache 2.0 license.

Keywords: Apache Spark, Openstack, Amazon EC2, Map-Reduce, HDFS, virtual cluster, cloud computing, Big Data, Apache Ignite.

DOI: 10.15514/ISPRAS-2016-28(6)-8

For citation: Borisenko O., Pastukhov R., Kuznetsov S. Deploying Apache Spark virtual clusters in cloud environments using orchestration technologies. *Trudy ISP RAN/Proc.ISP RAS*, vol. 28, issue 6, 2016. pp. 111-120 (in Russian). DOI: 10.15514/ISPRAS-2016-28(6)-8

References

[1]. Shanahan J. and Dai L. Large Scale Distributed Data Science using Apache Spark. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). ACM, New York USA, pp. 2323-2324.

[2]. Li M., Tan J., Wang Y., Zhang L., Salapura V. SparkBench: a comprehensive benchmarking suite for in memory data analytic platform Spark. In Proceedings of the 12th ACM International Conference on Computing Frontiers (CF '15). ACM, New York USA, Article 53.

[3]. Jeffrey D., Sanjay G. MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.M. Bhandarkar, "MapReduce programming with apache Hadoop," Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, Atlanta, GA, 2010, pp. 1-1.

[4]. Vavilapalli V., Murthy A., Douglas C., Agarwal S., Konar M., Evans R., Graves T., Lowe J., Shah H., Seth S., Saha B., Curino C., O'Malley O., Radia S., Reed B., Baldeschwieler E. Apache Hadoop YARN: yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (SOCC '13). ACM, New York USA, 2013, Article 5.

[5]. Apache Mesos project home page: http://mesos.apache.org

[6]. Guller, Mohammed. Cluster Managers. Big Data Analytics with Spark. Apress, 2015. 231-242.

[7]. Dinsmore, Thomas W. In-Memory Analytics. Disruptive Analytics. Apress, 2016, pp. 97-116.

[8]. Sefraoui, Aissaoui O, Eleuldj M. OpenStack: toward an open-source solution for cloud computing. International Journal of Computer Applications 55.3, 2012.

[9]. Hazelhurst, Scott. Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud. Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology. ACM, 2008.

[10]. Borisenko O., Laguta A., Turdakov D., Kuznetsov S, Automating cluster creation and management for Apache Spark in Openstack cloud, Trudy ISP RAN/Proc. ISP RAS, vol 26, issue 4, 2014, pp. 33-44 (in Russian). DOI: 10.15514/ISPRAS-2014-26(4)-4

[11]. Aleksiyants A., Borisenko O., Turdakov D., Sher A., Kuznetsov S. Implementing Apache Spark Jobs Execution and Apache Spark Cluster Creation for Openstack Sahara. Trudy ISP RAN/Proc. ISP RAS, vol. 27, issue 5, 2015, pp. 35-48. DOI: 10.15514/ISPRAS-2015-27(5)-3.

[12]. Ibrahim, Asmaa, Nawawy. A study of adopting big data to cloud computing. Technology Innovation and Entrepreneurship Center, Egypt Technology Innovation and Entrepreneurship Center, Egypt, 2015, pp. 1-7.

[13]. List of approved third-party project for Apache Spark. https://cwiki.apache.org/confluence/display/SPARK/Third+Party+Projects