

УСТРОЙСТВО РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ ДЛЯ ВЫПОЛНЕНИЯ ЗАДАЧ DATA SCIENCE

Алексей Юрьевич Тимонин

Пензенский государственный университет, Пенза, Россия

c013s017b301f018@mail.ru

Аннотация. Рассматриваются вопросы организации аппаратно-программных сред для распределенного процесса решения задач в областях науки о данных, таких как машинное обучение, большие данные, массово-параллельная обработка данных. Предложенные подходы могут использоваться при организации соответствующих лабораторных и практических занятий на базе компьютерных аудиторий ФВТ ПГУ.

Ключевые слова: архитектуры параллельной обработки данных, наука о данных, кластеры, Hadoop, большие данные

Наукой о данных (англ. Data Science) называют академическую дисциплину, находящуюся на стыке математики и информатики, которая изучает вопросы компьютеризированного сбора, анализа, хранения, передачи и визуализации данных вне зависимости от их структурированности и объема. К данной дисциплине помимо классических методов и технологий математической статистики, баз данных, компьютерного моделирования относят перспективные направления информатики: большие данные (англ. Big Data), машинное обучение (англ. Machine Learning) и нейронные сети, интеллектуальный анализ данных (англ. Data Mining), обработку естественного языка, визуальную аналитику, массово-параллельную обработку данных и облачные технологии. Основными технологическими и программными средствами Data Science принято считать [1]:

- Apache Hadoop – свободно распространяемый набор программного обеспечения для организации массово-параллельной обработки данных и распределения ресурсов в высоконагруженных системах;
- Python – интерпретируемый объектно-ориентированный язык программирования с простым синтаксисом и поддержкой внешних библиотек, реализующих множество функций, таких как научные вычисления (NumPY, SciPY), аналитика (Pandas, StatsModels), визуализация данных (Matplotlib, Seaborn), машинное обучение и нейронные сети (Keras, PyTorch), параллельные и распределенные вычисления (Ray);
- язык программирования R – используется в качестве языка научных и инженерных вычислений, является свободной альтернативой пакетам математического ПО MATLAB и MathCAD;
- SQL и NoSQL базы данных – используются для хранения и обработки гетерогенных данных разной степени структурированности;

- графовые базы данных – используются для хранения и визуализации данных, представляющих собой сеть.

Ввиду того, что большинство прикладных задач науки о данных связано с аналитической обработкой больших объемов разнородной информации, будь то тренировка нейронной сети, либо статистическое исследование, либо автоматизированное управление и мониторинг сложной технической системой, к организации вычислительной среды предъявляются требования по обеспечению высокого уровня производительности, отказоустойчивости и совместного доступа конечных пользователей с целями оперативной корректировки работы окружения, а также получения результатов выполнения задач. Эти требования можно обеспечить использованием интегральных схем специального назначения (англ. ASIC), однако у них имеются существенные недостатки: сложность проектирования, возможность использования только в конкретной задаче, стоимость производства. Поэтому альтернативным выходом является использование классических персональных ЭВМ, объединяемых в кластер с целью масштабирования вычислительных возможностей.

Организация распределенных вычислений проводится объединением компьютеров, находящихся в одной локальной сети, в кластер Hadoop. Apache Hadoop поддерживает ЭВМ на базе центральных процессоров с архитектурами набора команд x86-x64 и AArch64 (ARM64). Работа Hadoop на классических персональных компьютерах (x86-x64) возможна под управлением операционных систем Linux или Microsoft Windows с установленной подсистемой для запуска Linux-приложений WSL2 (англ. Windows Subsystem for Linux version 2). Работа на ЭВМ с архитектурой ARM64, которые в большинстве представлены мобильными и встраиваемыми устройствами, обеспечивается только в адаптированных операционных системах семейства Linux. Основными трудностями при развертывании Linux на устройствах с архитектурой ARM являются фрагментированность и закрытость аппаратных компонентов, а также отсутствие единого стандарта проектирования подобных устройств. В настоящее время предпринимаются успешные попытки стандартизации и переноса аппаратно-зависимого кода в ядро Linux как со стороны компаний-производителей, так и со стороны сообщества Linux разработчиков [2, 3].

Фреймворк Hadoop предполагает разделение всех ЭВМ в кластере на 2 категории: главные и подчиненные [4]. На главных располагаются узлы имен (англ. Name Nodes) распределенной файловой системы HDFS, главный сервер СУБД HBase и основные сервисы ресурсного посредника YARN ResourceManager. Подчиненные машины содержат узлы данных (англ. Data Nodes) HDFS, службы узлов YARN NodeManagers, «региональные» серверы HBase и прикладные аналитические приложения. Количество подчиненных узлов может динамически изменяться в запущенном кластере. Это обеспечивается выполнением следующих требований:

- рабочая нагрузка на подчиненных узлах изолирована от задач, запущенных на главных узлах;
- данные, хранящиеся на подчиненных узлах, частично дублируются на них для достижения отказоустойчивости на случай отключения, зависания или выхода из строя части узлов.

Классический кластер Hadoop включает в свой состав 1 главный узел и 2 или более подчиненных узла. Примерная структура кластера показана на Рисунке 1. Для достижения приемлемого уровня производительности и надежности вычислений при большом количестве подчиненных узлов рекомендуется создать дублирующий главный узел, либо перенести часть его функционала на другую ЭВМ. Для взаимодействия с подчиненными узлами мастер-узел должен быть обеспечен сетевыми портами с высокой пропускной способностью (от 1 Гбит/с). Другим требованием к главному узлу является использование высоконадежного хранилища для пространства имен и системного журнала событий, что обеспечивается с помощью технологий RAID, снимков диска (англ. snapshot) и резервного копирования на удаленное хранилище по расписанию.

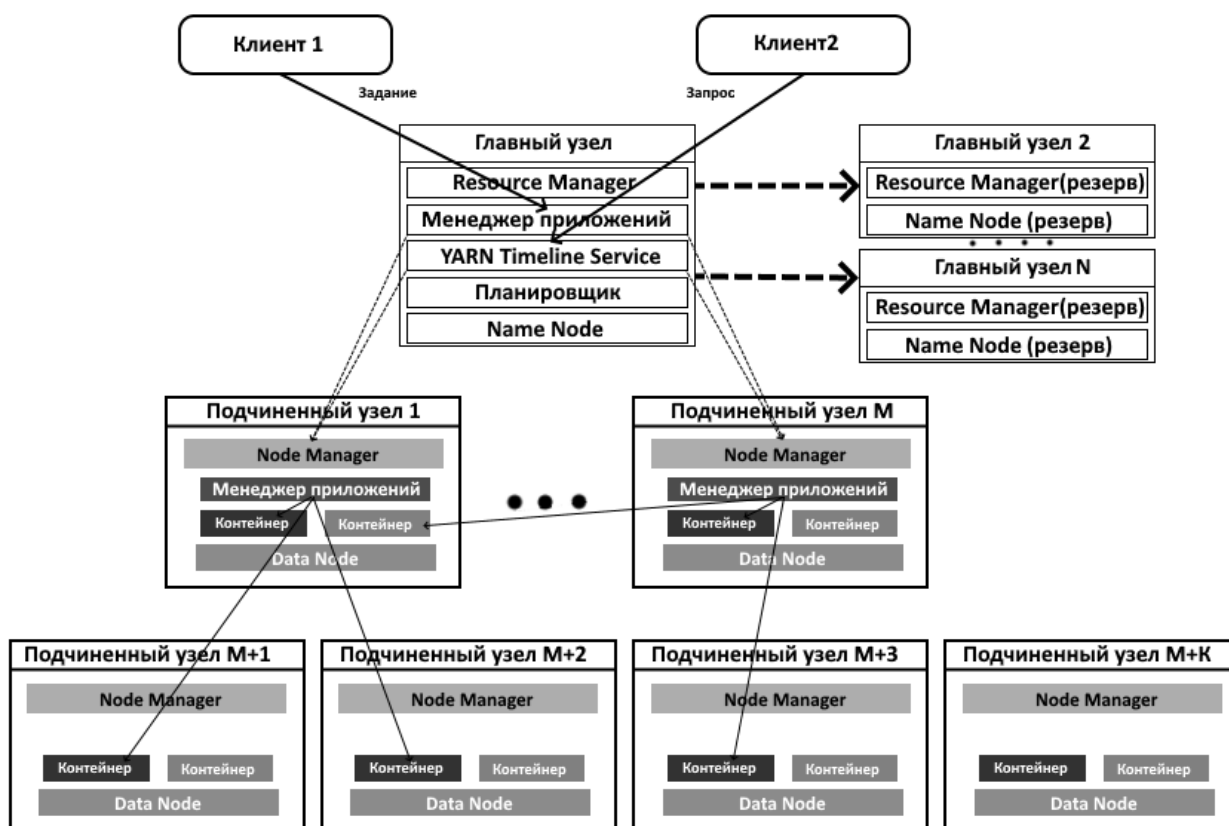


Рисунок 1 – Структурная схема кластера Hadoop с указанием пользователей.

Требования к подчиненным узлам в первую очередь выражаются в объеме имеющейся оперативной памяти и емкости дискового хранилища для хранения и обработки получаемых исходных данных. Параметры производительности центрального процессора, скорости копирования данных между оперативной памятью и диском являются вторичными. Приемлемые значения пропускной способности сетевых интерфейсов

подчиненных узлов составляют от 100Мбит/с. Совместно с описанными подчиненными узлами можно применять специализированные кэширующие узлы для хранения и управления промежуточными результатами вычислений с целью разгрузки ресурсов распределенной вычислительной среды. К таким узлам предъявляются требования по обеспечению высокой пропускной способности сети и наличия памяти с высокой скоростью операций чтения. От таких ЭВМ не требуется высокой производительности при обработке данных. Поэтому в качестве кэширующих узлов можно использовать устройства, спроектированных на энергоэффективных ARM платформах.

Общие рекомендации по настройке узлов кластера Hadoop [4]:

1. Наличие отдельных дисковых разделов под операционную систему, файлов Hadoop и журналов системных событий.
2. Размер области подкачки (англ. Swap) должен быть в 2 раза больше объема имеющейся на узле оперативной памяти.
3. Для хранения потоковых и других часто перезаписываемых данных не рекомендуется использовать SSD и NAND накопители, ввиду опасности быстрого износа таких устройств.

После установки и первичной настройки фреймворка Hadoop на ЭВМ локальной сети, нужно назначить роль каждому узлу будущего кластера:

1. На каждой ЭВМ в кластере редактируется файл /etc/hosts, в котором компьютерам присваиваются имена хостов по IP-адресам.
2. На узлах настраивается защищенное соединение ssh для обращения друг к другу без ввода пароля. Созданный ssh-ключ сохраняется на каждом из компьютеров.
3. Назначается главный узел кластера посредством ввода имени определенного хоста в файл etc/hadoop/masters.
4. Список подчиненных узлов указывается в файле etc/hadoop/slaves на главном узле аналогичным образом.
5. На мастер-узле форматируется Name Node с помощью команды `$HADOOP_HOME/bin/hadoop namenode -format`
6. Завершающим шагом является запуск системных служб Hadoop командой `$HADOOP_HOME/sbin/start-all.sh`

После завершения настройки кластера можно приступить к настройке приложений на подчиненных узлах и добавлению их в планировщик на главном узле.

Для надежного хранения данных в Apache Hadoop используются специализированная файловая система HDFS [4], обладающая возможностью проводить асинхронную репликацию данных на уровне кластера, а не на уровне отдельных узлов. Другой особенностью HDFS является реализация принципа WORM (англ. Write-once and read-many, единожды записать – много раз прочитать), что выражается в отсутствии конфликтов

множественной записи. Также в данной файловой системе предусмотрено сжатие данных и журналирование.

Однако одним важным недостатком HDFS является отсутствие встроенных механизмов проверки ссылочной целостности данных. Также нужно учитывать, что хранение и обработка некоторого множества данных может осуществляться целиком внутри отдельных узлов. Как следствие, возникает задача выбора локальной файловой системы для отдельных узлов распределенной вычислительной среды, которая должна обладать свойствами отказоустойчивости, масштабируемости, поддержки очень большого объема записываемых данных. Одним из возможных решений будет использование ZFS [5] – файловой системы, рассчитанной на хранение до 256 ЗиБ в томе и позволяющей организовывать пулы данных в рамках одной ЭВМ. Данная файловая система поддерживает механизмы создания снимков томов, копирования при записи и сжатия данных. Обеспечение целостности данных заложено в архитектуре ZFS, применяющей хэш-дерево для всех операций с данными. Другой значимой особенностью является возможность создания гетерогенного хранилища в рамках одной ЭВМ: например, можно объединить несколько устройств хранения в единый пул данных и настроить зеркалирование с помощью механизма виртуальных устройств (аналог RAID), выделить высокоскоростные накопители в качестве кэша записи для медленных устройств и снижения нагрузки на ОЗУ, развернуть другую файловую систему поверх виртуального устройства ZFS.

В настоящее время появляется все больше возможностей по созданию высокопроизводительных систем без дополнительных затрат, связанных с использованием специализированного оборудования. Область задач науки о данных основывается на обработке большого количества разнородных данных. Установка и настройка программных продуктов для работы с большими данными или нейронными сетями на отдельную ЭВМ не представляет серьезных трудностей. Однако производительности одного персонального компьютера не всегда бывает достаточно для решения прикладных задач Data Science даже в рамках соответствующих учебных курсов. Создание распределенной вычислительной среды на базе локальной сети из персональных ЭВМ сопряжено с рядом особенностей, часть которых были раскрыты в данной работе.

Список литературы

1. Грас, Д. Data Science. Наука о данных с нуля / Д. Грас. – СПб.: БХВ-Петербург, 2021. – 416 с.
2. Ghuloum, A. Fresher OS with Projects Treble and Mainline / A. Ghuloum, M. Ben Ari // Android Developers Blog [Электронный ресурс]. – Режим доступа: <https://android-developers.googleblog.com/2019/05/fresher-os-with-projects-treble-and-mainline.html> (дата обращения: 28.08.2021).
3. postmarketOS // real Linux distribution for phones [Электронный ресурс]. – Режим доступа: <https://wiki.postmarketos.org/wiki/Devices> (дата обращения: 28.08.2021).

4. Karambelkar, H. Apache Hadoop 3 Quick Start Guide / H. Karambelkar. – Packt, 2018. – 226 p.

5. Руководство по администрированию файловых систем ZFS Solaris [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/cd/E19253-01/820-0836/index.html> (дата обращения: 28.08.2021).

Информация об авторах

Тимонин Алексей Юрьевич, к.т.н., старший преподаватель кафедры «Системы автоматизированного проектирования», Пензенский государственный университет.