

Hadoop — проект фонда [Apache Software Foundation](#) — проект фонда Apache Software Foundation, [свободно распространяемый](#) — проект фонда Apache Software Foundation, свободно распространяемый набор [утилит](#) — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, [библиотек](#) — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и [программный каркас](#) — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и программный каркас для разработки и выполнения распределённых программ, работающих на [кластерах](#) — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и программный каркас для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов.

Используется для реализации поисковых и контекстных механизмов многих высоконагруженных веб-сайтов: для [Yahoo!](#) — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и программный каркас для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Используется для реализации поисковых и

Считается одной из основополагающих технологий «[больших данных](#)». Вокруг Nadoor образовалась целая экосистема из связанных проектов и технологий, многие из которых развивались сначала в рамках проекта, а впоследствии стали самостоятельными.

Со второй половины 2000-х годов идёт процесс активной коммерциализации технологии, несколько компаний строят бизнес целиком на создании коммерческих дистрибутивов Nadoor и услуг по технической поддержке экосистемы, а практически все крупные поставщики информационных технологий для организаций в том или ином виде включают Nadoor в продуктовые стратегии и линейки решений. Разработка была инициирована в начале 2005 года [Дугом Каттингом](#)

Со второй половины 2000-х годов идёт процесс активной коммерциализации технологии, несколько компаний строят бизнес целиком на создании коммерческих дистрибутивов Nadoor и услуг по технической поддержке экосистемы, а практически все крупные поставщики информационных технологий для организаций в том или ином виде включают Nadoor в продуктовые стратегии и линейки решений.

Разработка была инициирована в начале 2005 года Дугом Каттингом

В [январе 2006 года](#) В январе 2006 года корпорация [Yahoo](#) выделила Hadoop в отдельный проект для разработки инфраструктуры распределённых вычислений. В [феврале 2008 года](#) разработки инфраструктуры распределённых вычислений. В феврале 2008 года Yahoo запустила кластерную поисковую машину на 10 тыс. [процессорных ядер](#), управляемую средствами Hadoop. В [апреле 2010 года](#) В апреле 2010 года корпорация [Google](#) в апреле 2010 года корпорация Google предоставила Apache Software Foundation права на использование технологии MapReduce, через три месяца после её защиты в [патентном бюро США](#).

Начиная с 2010 года Hadoop неоднократно характеризуется как ключевая технология «[больших данных](#)» Начиная с 2010 года Hadoop неоднократно характеризуется как ключевая технология «больших данных», прогнозируется его широкое распространение для [массово-параллельной](#) Начиная с 2010 года Hadoop неоднократно характеризуется как ключевая технология «больших данных», прогнозируется его широкое распространение для массово-параллельной обработки данных, и, наряду с Cloudera, появляется серия технологических стартапов, целиком

В **Hadoop Common** входят библиотеки управления файловыми системами, поддерживаемыми Hadoop и сценарии входят библиотеки управления файловыми системами, поддерживаемыми Hadoop и сценарии создания необходимой инфраструктуры и управления распределённой обработкой, для удобства выполнения которых создан специализированный упрощённый интерпретатор командной строки (*FS shell, filesystem shell*), запускаемый из оболочки операционной системы командой вида: `hdfs dfs -command URI`, где *command* — команда интерпретатора, а *URI* — список ресурсов с префиксами, указывающими тип поддерживаемой файловой системы, например `hdfs://example.com/file1` или `file:///tmp/local/file2`. Большая часть команд интерпретатора реализована по аналогии с соответствующими командами Unix (таковы, например, cat — список ресурсов с префиксами, указывающими тип поддерживаемой файловой системы, например `hdfs://example.com/file1` или `file:///tmp/local/file2`. Большая часть команд интерпретатора реализована по аналогии с соответствующими командами Unix (таковы, например, `cat`, chmod — список ресурсов с префиксами, указывающими тип

HDFS (*Hadoop Distributed File System*) — файловая система — 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.

Развёртывание экземпляра HDFS предусматривает наличие центрального узла имён ([англ. name node](#)), хранящего метаданные файловой системы и метаинформацию о распределении блоков, и серии узлов данных ([англ. data node](#)), непосредственно хранящих блоки файлов. Узел имён отвечает за обработку операций уровня файлов и каталогов — открытие и закрытие файлов, манипуляция с каталогами, узлы данных непосредственно отрабатывают операции по записи и чтению данных. Узел имён и узлы данных снабжаются [веб-серверами](#), отображающими текущий статус узлов и позволяющими просматривать содержимое файловой системы. Административные функции доступны из интерфейса командной строки.

Hadoop поддерживает работу и с другими распределёнными файловыми системами без использования HDFS, поддержка [Amazon S3](#) Hadoop поддерживает работу и с другими распределёнными файловыми системами без использования HDFS, поддержка Amazon S3 и [CloudStore](#) Hadoop поддерживает работу и с другими распределёнными файловыми системами без использования HDFS, поддержка Amazon S3 и CloudStore реализована в основном дистрибутиве. С другой стороны, HDFS может использоваться не только для запуска MapReduce-заданий, но

YARN ([англ.](#) *Yet Another Resource Negotiator* — «ещё один ресурсный посредник») — модуль, появившийся с версией 2.0 (2013), отвечающий за управление ресурсами кластеров и планирование заданий.

Если в предыдущих выпусках эта функция была интегрирована в модуль [MapReduce](#), где была реализована единым компонентом (*JobTracker*), то в YARN функционирует логически самостоятельный [демон](#) — планировщик ресурсов (*ResourceManager*), абстрагирующий все вычислительные ресурсы кластера и управляющий их предоставлением приложениям распределённой обработки.

Работать под управлением YARN могут как MapReduce-программы, так и любые другие распределённые приложения, поддерживающие соответствующие программные интерфейсы; YARN обеспечивает возможность параллельного выполнения нескольких различных задач в рамках кластера и их изоляцию (по принципам [мультиарендности](#)).

Разработчику распределённого приложения необходимо реализовать специальный класс управления приложением (*ApplicationMaster*), который отвечает за координацию заданий в рамках тех ресурсов, которые предоставит планировщик ресурсов; планировщик ресурсов же отвечает за создание экземпляров класса управления приложением и взаимодействия с ним через соответствующий сетевой протокол.

YARN может быть рассмотрен как кластерная операционная система в том смысле, что ведает интерфейсом между аппаратными ресурсами кластера и широким классом приложений, использующих его мощности для выполнения вычислительной обработки.

Hadoop MapReduce — [программный каркас](#) — программный каркас для программирования распределённых вычислений в рамках парадигмы [MapReduce](#) — программный каркас для программирования распределённых вычислений в рамках парадигмы MapReduce. Разработчику приложения необходимо реализовать базовый обработчик, который на каждом вычислительном узле кластера обеспечит преобразование исходных пар «[ключ — значение](#)» в промежуточный набор пар «ключ — значение», и обработчик, сводящий промежуточный набор пар в окончательный, сокращённый набор (*свёртку*, класс, реализующий интерфейс Reducer). Каркас передаёт на вход свёртки отсортированные выводы от базовых обработчиков, сведение состоит из трёх фаз — *shuffle* (*масовка*, выделение нужной секции вывода), *sort* (*сортировка*, группировка по ключам выводов от распределителей — досортировка, требующаяся, когда разные атомарные обработчики возвращают наборы с одинаковыми ключами, при этом, правила сортировки на этой фазе могут быть заданы программно и использовать какие-либо особенности внутренней структуры ключей) и собственно *reduce* (*усечение*) —

Hadoop MapReduce позволяет создавать задания как с базовыми обработчиками, так и со свёртками, написанными без использования Java: утилиты *Hadoop streaming* позволяют использовать в качестве базовых обработчиков и свёрток любой исполняемый файл позволяют использовать в качестве базовых обработчиков и свёрток любой исполняемый файл, работающий со стандартным вводом-выводом операционной системы (например, утилиты командной оболочки UNIX позволяют использовать в качестве базовых обработчиков и свёрток любой исполняемый файл, работающий со стандартным вводом-выводом операционной системы (например, утилиты командной оболочки UNIX), есть также SWIG позволяют использовать в качестве базовых обработчиков и свёрток любой исполняемый файл, работающий со стандартным вводом-выводом операционной системы (например, утилиты командной оболочки UNIX), есть также SWIG-совместимый прикладной интерфейс программирования *Hadoop pipes* на C++.

В состав дистрибутивов Hadoop входят реализации различных конкретных базовых обработчиков и свёрток, наиболее типично используемых в распределённой обработке. В первых версиях

Одной из основных целей Hadoop изначально было обеспечение горизонтальной масштабируемости. Одной из основных целей Hadoop изначально было обеспечение горизонтальной масштабируемости кластера посредством добавления недорогих узлов (оборудования массового класса, англ. commodity hardware), без прибегания к мощным серверам и дорогим сетям хранения данных), без прибегания к мощным серверам и дорогим сетям хранения данных. Функционирующие кластеры размером в тысячи узлов подтверждают осуществимость и экономическую эффективность таких систем, так, по состоянию на 2011 год известно о крупных кластерах Hadoop в Yahoo (более 4 тыс. узлов с суммарной ёмкостью хранения 15 Пбайт каждый), Facebook (около 2 тыс. узлов на 21 Пбайт) и Ebay (700 узлов на 16 Пбайт). Тем не менее, считается, что горизонтальная масштабируемость в Hadoop-системах ограничена, для Hadoop до версии 2.0 максимально возможно оценивалась в 4 тыс. узлов при использовании 10 MapReduce-заданий на узел. Этому ограничению способствовала концентрация в модуле MapReduce функций контроля за жизненным циклом заданий с выносом её в модуль YARN в Hadoop 2.0 и

Ещё одним ограничением Hadoop-систем является размер оперативной памяти на узле имён (*NameNode*), хранящем всё пространство имён кластера для распределения обработки, притом общее количество файлов, которое способен обрабатывать узел имён — 100 млн. Для преодоления этого ограничения ведутся работы по распределению узла имён, единого в текущей архитектуре на весь кластер, на несколько независимых узлов. Другим вариантом преодоления этого ограничения является использование распределённых [СУБД](#)), хранящем всё пространство имён кластера для распределения обработки, притом общее количество файлов, которое способен обрабатывать узел имён — 100 млн. Для преодоления этого ограничения ведутся работы по распределению узла имён, единого в текущей архитектуре на весь кластер, на несколько независимых узлов. Другим вариантом преодоления этого ограничения является использование распределённых СУБД поверх HDFS, таких как [HBase](#)), хранящем всё пространство имён кластера для распределения обработки, притом общее количество файлов, которое способен обрабатывать узел имён — 100 млн. Для преодоления этого ограничения ведутся

Существуют тенденции как к снижению вычислительной мощности узлов и использованию процессоров с низким энергопотреблением ([ARM](#) Существуют тенденции как к снижению вычислительной мощности узлов и использованию процессоров с низким энергопотреблением (ARM, [Intel Atom](#) Существуют тенденции как к снижению вычислительной мощности узлов и использованию процессоров с низким энергопотреблением (ARM, Intel Atom), так и применения высокопроизводительных вычислительных узлов одновременно с сетевыми решениями с высокой пропускной способностью ([InfiniBand](#) Существуют тенденции как к снижению вычислительной мощности узлов и использованию процессоров с низким энергопотреблением (ARM, Intel Atom), так и применения высокопроизводительных вычислительных узлов одновременно с сетевыми решениями с высокой пропускной способностью (InfiniBand в [Oracle Big Data Appliance](#) Существуют тенденции как к снижению вычислительной мощности узлов и использованию процессоров с низким энергопотреблением (ARM, Intel Atom), так и применения высокопроизводительных вычислительных узлов одновременно с сетевыми решениями с высокой пропускной

В руководстве Hadoop (первых версий, ранее 2.0) указывалось, что приемлемым уровнем параллелизма является использование 10—100 экземпляров базовых обработчиков на узел кластера, а для задач, не требующих значительных затрат процессорного времени — до 300; для свёрток считалось оптимальным использование их по количеству узлов, умноженному на коэффициент из диапазона от 0,95 до 1,75 и константу `mapred.tasktracker.reduce.tasks.maximum`. С большим значением коэффициента наиболее быстрые узлы, закончив первый раунд сведения, раньше получают вторую порцию промежуточных пар для обработки, таким образом, увеличение коэффициента избыточно загружает кластер, но при этом обеспечивает более эффективную балансировку нагрузки. В YARN вместо этого используются конфигурационные константы, определяющие значения доступной оперативной памяти и виртуальных процессорных ядер, доступных для планировщика ресурсов, на основании которых и определяется уровень параллелизма.

На фоне популяризации Hadoop в 2008 году и сообщениях о построении Hadoop-кластеров в Yahoo и Facebook, в октябре 2008 года была создана компания Cloudera во главе с Майклом Ольсоном, бывшим генеральным директором Sleepycat (фирмы-создателя Berkeley DB), целиком нацеленная на коммерциализацию Hadoop-технологий. В сентябре 2009 в Cloudera из Yahoo перешёл основной разработчик Hadoop Дуг Каттинг, и благодаря такому переходу комментаторы охарактеризовали Cloudera как «нового знаменосца Hadoop», несмотря на то, что основная часть проекта была создана всё-таки сотрудниками Facebook и Yahoo. В 2009 году основана компания MapR, поставившая целью создать высокопроизводительный вариант дистрибутива Hadoop, и поставлять его как собственническое программное обеспечение.

В апреле 2009 года Amazon запустил облачный сервис Elastic MapReduce, предоставляющий подписчикам возможность создавать кластеры Hadoop и выполнять на них задания с повременной оплатой. Позднее, в качестве альтернативы, подписчики Amazon Elastic MapReduce получили выбор между классическим дистрибутивом от Apache и дистрибутивами от MapR.

В 2011 году Yahoo выделила подразделение, занимавшееся разработкой и использованием Hadoop, в самостоятельную компанию — Hortonworks, вскоре новой компании удалось заключить соглашение с Microsoft о совместной разработке дистрибутива Hadoop для Windows Azure и Windows Server. В том же году все крупные производители технологического программного обеспечения для организаций в том или ином виде включили Hadoop-технологии в стратегии и продуктовые линейки. Так, Oracle выпустила аппаратно-программный комплекс Big Data appliance (заранее собранный в телекоммуникационном шкафу и предконфигурированный Hadoop-кластер с дистрибутивом от Cloudera), IBM на основе дистрибутива Apache создала продукт BigInsights, EMC лицензировала у MapR их высокопроизводительный Hadoop для интеграции в продукты незадолго до этого поглощённой Greenplum (позднее это бизнес-подразделение было выделено в самостоятельную компанию Pivotal, и она перешла на полностью самостоятельный дистрибутив Hadoop на базе кода Apache), Teradata заключила соглашение с Hortonworks по интеграции Hadoop в аппаратно-программный комплекс массово-параллельной обработки Aster MapReduce appliance. В 2013 году собственный дистрибутив Hadoop создала Intel.