

DOI: 10.18698/0236-3933-2015-6-46-62

УДК 351.814.331.3

## ОСОБЕННОСТИ ПЛАНИРОВАНИЯ ВОЗДУШНОГО ДВИЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ СИНОПТИЧЕСКИХ КАРТ, ПОСТРОЕННЫХ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ BIG DATA

**А.И. Власов, П.В. Новиков, А.М. Ривкин**

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация  
e-mail: dr-on@mail.ru; info@iu4.bmstu.ru

*Рассмотрены проблемы эффективного использования воздушного пространства нашей страны и способы их решения. Основное внимание уделено методам обеспечения качества планирования движения воздушных судов. Показано, что для эффективного использования имеющихся ресурсов необходимо максимально точно рассчитывать положение воздушного судна на всех этапах полета, т.е. 4D траекторию полета воздушного судна. Добиться этого можно, обрабатывая информацию о внешних факторах, влияющих на воздушное судно, которая содержится в метеопрогнозах. Однако объем метеоданных, необходимых для обработки, слишком велик чтобы обработать его с помощью стандартных средств в приемлемое время. Для решения этой проблемы используются такие технологии, как Big Data.*

**Ключевые слова:** 4D траектория, воздушное движение, метеопрогноз, формат GRIB, большие данные.

## FEATURES OF PLANNING AIR TRAFFIC USING WEATHER MAPS CONSTRUCTED WITH APPLICATION OF THE BIG DATA TECHNOLOGIES

**A.I. Vlasov, P.V. Novikov, A.M. Rivkin**

Bauman Moscow State Technical University, Moscow, Russian Federation  
e-mail: dr-on@mail.ru; info@iu4.bmstu.ru

*This article discusses both the problems of efficient use of the Russian airspace and the ways to solve the problems. The methods of quality assurance while planning the aircraft movements are described in detail. It is shown that the effective usage of the available resources requires determination of the aircraft's position during all phases of the flight, i.e. its 4D flight trajectory, with precise accuracy. This can be achieved by processing the information about the external factors affecting the aircraft, which is provided by weather forecast. However, the weather data required for processing is too big to be evaluated by conventional means within the reasonable time. To solve this problem, some modern technologies such as Big Data are used.*

**Keywords:** 4D path, air traffic, weather, GRIB, Big Data.

Для обеспечения безопасности полетов воздушных судов (ВС) в воздушном пространстве необходимы планирование и координация

их движения, а также непосредственное управление этим движением (передача информации, помощь в аварийных ситуациях, предотвращение таких ситуаций и т.д.). Эти функции в соответствии с действующим воздушным кодексом возложены на органы единой системы управления воздушным движением (УВД). Кроме того, часть функций по управлению движением ВС возложена на ведомственные органы управления. Они действуют в пределах установленных для них районов и зон. Конечная цель этих органов управления — обеспечение безопасности, регулярности и эффективности полетов. В действующей системе управления ведущая роль принадлежит единой системе УВД. Она имеет главный, зональные и региональные центры. Все воздушное пространство РФ поделено на секторы. Часть секторов зарезервирована для военных целей, часть — для гражданских целей. Военными секторами управляют ВВС.

Процесс оперативного управления включает в себя планирование, координирование и непосредственное УВД. Планирование воздушного движения выполняется с учетом пропускной способности воздушного пространства, аэродромов и возможностей органов УВД в обеспечении управления. Различают предварительное планирование (за несколько суток до дня полета) — для составления расписаний полетов, потоков движения ВС, графиков использования аэродромов и т.д. Суточное планирование — накануне дня полетов. Текущее — в процессе выполнения суточного плана полетов для корректировки условий полетов отдельных ВС.

Гражданские, военные и другие ведомства имеют собственные планы полетов. Их необходимо координировать между собой. Координирование заключается в согласовании планов полетов различных ведомств в соответствующих районах и зонах, включая перераспределение потоков движения ВС по воздушным трассам страны [1, 2].

В современных автоматизированных системах (АС) УВД большое внимание при планировании полетов уделяется расчету 4D-траектории ВС по данным плана полета, причем повышение достоверности такого расчета является актуальной задачей, поскольку данные расчета используются в задачах распределения плановой информации по диспетчерским рабочим местам, а в системах планирования — для определения загрузок секторов УВД и других элементов воздушного пространства.

Высокие требования к достоверности информации, на основе которой принимаются решения в системах планирования воздушного движения, требуют учета как можно большего числа факторов при расчете траектории, в противном случае эффективность принимаемых решений в процессах планирования будет снижаться. Для эффективного планирования и координирования полетов необходимо учитывать

метеоусловия на всех этапах полета. Несмотря на значительные достижения в области метеорологических наблюдений и прогноза опасных для полетов метеорологических явлений, более 30 % серьезных авиационных инцидентов и 40 % авиационных происшествий с человеческими жертвами связаны с влиянием метеорологических факторов на полеты ВС. Одной из основных причин авиационных происшествий и катастроф является быстрое устаревание метеорологической информации, а также недостаточная достоверность прогнозов погоды как по маршруту полетов самолетов, так и в районе аэродрома [3, 4].

Всемирной метеорологической организацией (ВМО) с 1980-х годов введена Всемирная система зональных прогнозов, чтобы поставлять странам-участникам единообразную прогностическую информацию о ветре, температуре, давлении и влажности на высотах, необходимую для использования в АС планирования полетов. Метеоинформация распространяется в электронном виде в специальном формате GRIB, который заменил Aeronautical Data Format (ADF) — аэронавигационный формат данных;

GRIB (GRIdded Binary) — формат предоставления и передачи данных, используемый в метеорологии для хранения исторических и прогнозируемых данных о погоде. Он был стандартизирован комиссией по основным системам ВМО, известен под номером GRIB FM 92-IX и описан в 306 номере руководства ВМО по кодам [5, 6].

Для построения наиболее точной четырехмерной траектории ВС, необходимо иметь информацию о всех внешних факторах, которые могут влиять на ВС. В современных системах обычно используют метеоинформацию о зональной и меридиональной составляющей скорости ветра, передаваемую в GRIB формате. Реже используют информацию о температуре, давлении и относительной влажности. Метеоданные, передаваемые в GRIB формате, обычно привязаны к узлам сетки с шагом 1250 м, что не позволяет обеспечить высокую точность планирования полетов [6, 7].

Основной причиной низкого качества планирования является большой шаг сетки и отсутствие необходимых вычислительных мощностей для обработки файлов с меньшим шагом сетки. При уменьшении шага сетки в 10 раз, с 1250 до 125 м, размер данных, которые нужно передать, увеличится в 10 раз. Средний объем метеоинформации о скорости ветра, покрывающей территорию РФ, за день составляет ~ 250 КБ. Данный прогноз содержит информацию о зональной и меридиональной скоростях ветра в узлах сетки с шагом 1250 м вдоль меридиана и данные о 11 слоях с 99... 1000 гПа.

При увеличении плотности узлов сетки в 10 раз получается ~ 2,5 МБ без увеличения плотности по высоте. При уменьшении шага по высоте в 2 раза объем данных составит 5 МБ. Это объем

чистых данных о скорости ветра в сжатой форме. Добавим к этому данные о температуре, давлении и относительной влажности и получим  $\sim 12,5$  МБ чистой сжатой информации. Автоматизированные системы УВД необходимы не только для актуальной информации о текущей метеобстановке, а также для прогноза на срок от 6 ч до 10 суток.

При шаге прогноза в 6 ч для прогноза на 10 суток получим  $10 \times 24/6 = 40$  порций данных каждые 6 ч. Таким образом, АС УВД должна получить и обработать 500 МБ информации за 6 ч или 2 ГБ в день. Если уменьшить шаг прогноза до часа, что даст значительно более точные данные для планирования, объем данных, получаемых системой, станет 3 ГБ, а получать их система станет каждый час. В день системе придется обработать 72 ГБ. Однако в РФ существует 133 метеоцентра и для получения наиболее точного прогноза следует получать информацию из максимального числа источников, а далее аппроксимировать полученную информацию для дальнейшего использования при расчете 4D-траектории ВС [6–9].

Уровень сложности системы определяется большим числом синоптических эффектов; обилием взаимных связей между этими эффектами; одновременным протеканием и взаимосвязями между разными явлениями физико-химической природы в локальных объемах; нелинейными зависимостями между переменными параметрами и т.д.

Реализация перечисленных особенностей по улучшению качества планирования требует значительных вычислительных мощностей. В настоящей работе предложен новый подход к обработке метеоданных, основанный на технологии “Big Data” (большие данные), который позволяет эффективно обрабатывать значительные объемы структурированных и неструктурированных данных.

**1. Использование технологии Big Data при анализе метеобстановки.** Введение термина “большие данные” приписывают Клиффорду Линчу, редактору журнала Nature, подготовившему к 3 сентября 2008 года специальный номер журнала с темой “*Как могут повлиять на будущее науки технологии, открывающие возможности работы с большими объемами данных?*”, в котором были собраны материалы о феномене взрывного роста объемов данных, многообразии обрабатываемых данных и технологических перспективах в парадигме вероятного скачка “от количества к качеству”, термин был предложен по аналогии с расхожими в деловой англоязычной среде метафорами “*большая нефть*”, “*большая руда*” [10–12].

Большие данные (Big Data) — в информационных технологиях — это серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объемов [10].

В качестве определяющих характеристик для больших данных отмечают “три V”: объем (volume, в смысле физического объема), скорость (velocity — в смысле как скорости прироста, так и необходимости высокоскоростной обработки и получения результатов), многообразие (variety — в смысле возможности одновременной обработки различных типов структурированных и полуструктурированных данных) [13].

Большие данные имеют очень большой объем, поступают с большой скоростью (объем постоянно растет) и часто имеют очень сложную структуру, либо вообще не структурированы. В связи с перечисленными характеристиками больших данных (объем, неструктурированность) традиционные способы хранения и обработки данных на основе систем управления базами данных (СУБД) не подходят. Нужно использовать архитектуры распределенной параллельной обработки и хранения данных. Сегодня самой распространенной моделью обработки больших объемов данных является концепция MapReduce. Наиболее известная и часто используемая программная реализация MapReduce выполнена компанией Apache и называется Hadoop.

Наиболее часто в качестве базового принципа обработки больших данных используют архитектуру Shared Nothing (*ничего не разделяется*) (рис. 1), обеспечивающую массивно-параллельную обработку, масштабируемую без деградации производительности на сотни и тысячи узлов [9–12].

“Ничего не разделяется” — это распределенная вычислительная архитектура, в которой каждый вычислительный узел работает независимо, отсутствует единая для всей системы точка подключения. Данные разбиваются на множество независимых порций и каждый вычислительный узел обрабатывает свою порцию данных. Ни один узел не делится памятью или собственным дисковым пространством с другими узлами кластера [11, 12, 14, 15].

Помимо архитектуры Shared Nothing существуют архитектуры Shared Disk (*общий диск*) и Shared Memory (*общая память*). Клас-

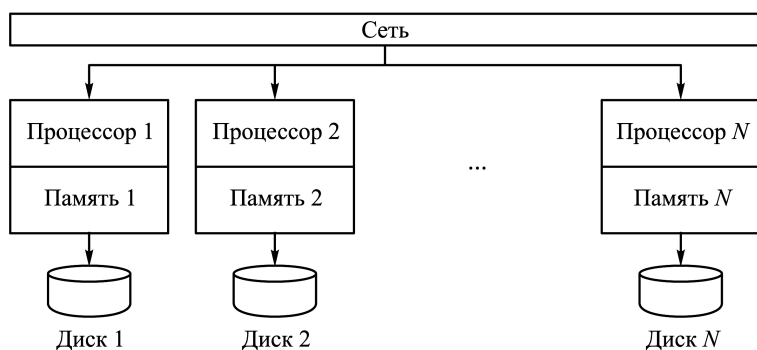
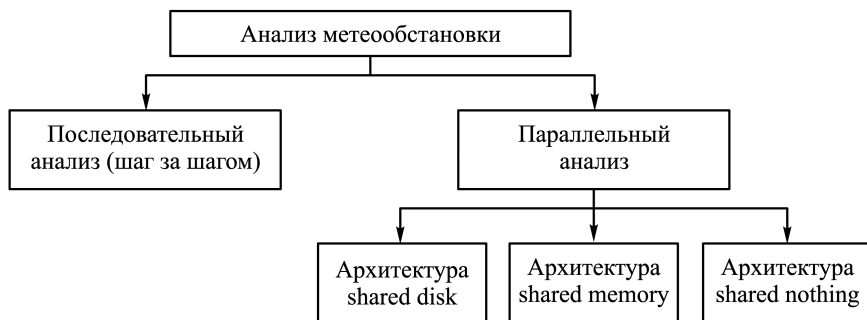


Рис. 1. Структура Shared Nothing архитектуры



**Рис. 2. Классификация подходов анализа метеобстановки**

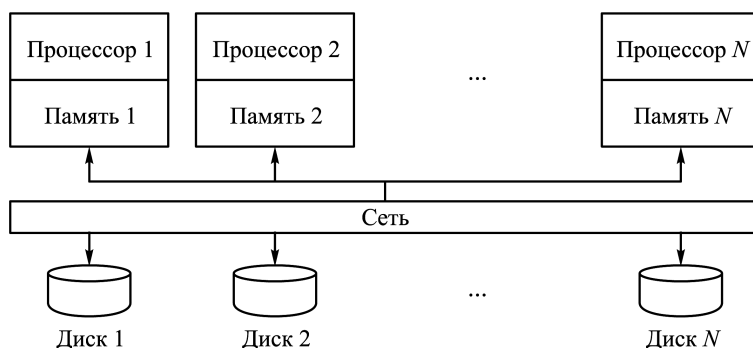
сификация подходов для анализа метеобстановки представлена на рис. 2.

Вариант с последовательным анализом дорогостоящий и не имеет возможности бесконечного наращивания ресурсов при необходимости. Данный подход хорош при небольших объемах информации или когда нет временных рамок. Также данный вариант применим, когда задачу невозможно распараллелить, т.е. для вычисления шага  $n$  необходимы результаты шага  $n - 1$ .

В случае с анализом метеобстановки всю задачу можно легко распараллелить, тем самым ускорив выполнение. Архитектура Shared Disk подразумевает использование общего сетевого хранилища. Структура архитектуры Shared Disk представлена на рис. 3.

Данная архитектура имеет слабое место — систему долговременно-го хранения информации. При необходимости считывания или записи большого объема информации одновременно диск становится узким местом. Архитектура Shared Nothing лишена этого недостатка, однако в ней присутствует избыточность информации.

Архитектура Shared Memory имеет общую оперативную память, доступ к которой осуществляется по сети. В данном случае узким местом является оперативная память. К тому же скорость сети значительно ниже, чем скорость общения процессора с оперативной памятью



**Рис. 3. Структура Shared Disk архитектуры**

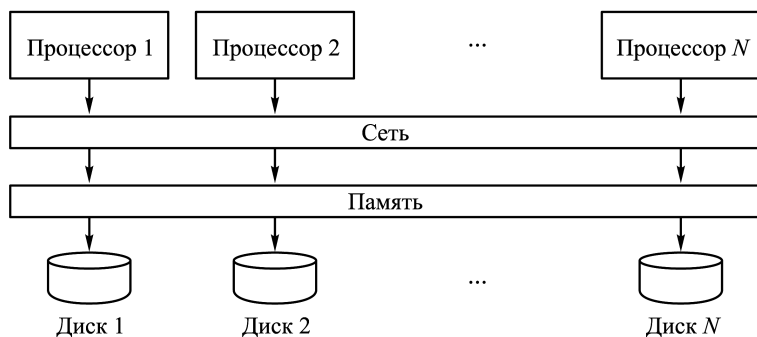


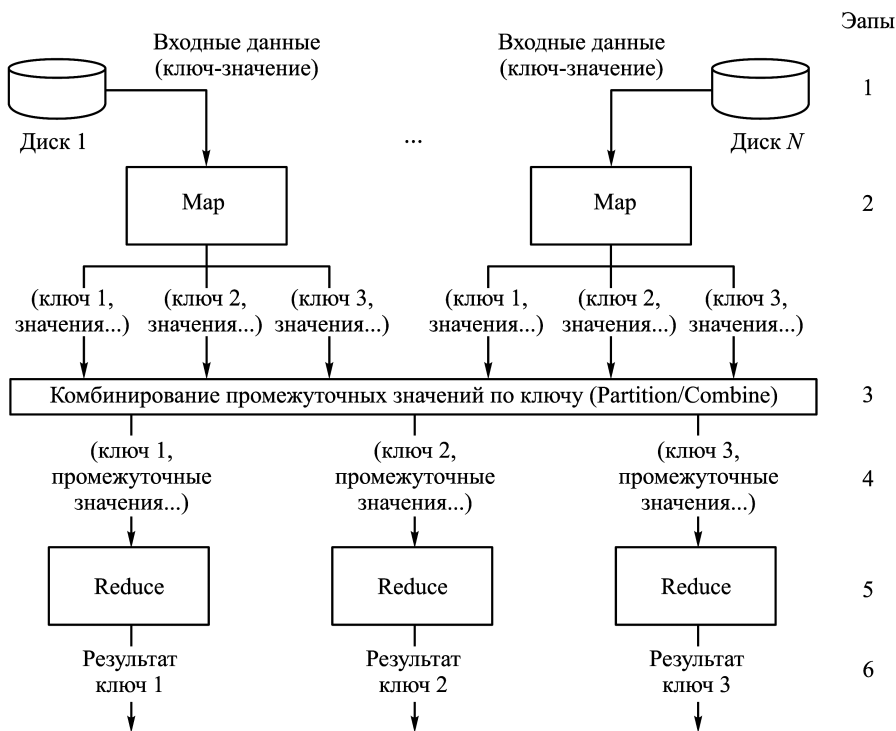
Рис. 4. Структура Shared Memory архитектуры

в современном компьютере. Структура архитектуры Shared Memory представлена на рис. 4.

Shared Nothing архитектура также имеет слабое место — сеть, однако в вычислительной модели MapReduce, рассмотренной далее, количество данных, передающихся по сети минимизировано. В случае с метеообстановкой, присутствует необходимость обработки большого количества данных, частое обращение к диску и постоянная работа с памятью. Объемы информации растут, а значит необходимо иметь возможность линейно масштабировать систему. Исходя из этих данных, отметим, что архитектура Shared Nothing подходит лучшим образом.

**2. Особенности реализации обработки метеоданных с использованием вычислительной модели MapReduce.** Модель MapReduce была разработана компанией Google, известной не только своими технологическими достижениями, но и способностью их скрывать. Именно в таком духе выдержана статья Джеффри Дина и Санжая Чемавата “MapReduce: упрощенная обработка данных на больших кластерах” (MapReduce: Simplified Data Processing on Large Clusters), недосказанности которой компенсирует работа Ральфа Ламмеля из исследовательского центра Microsoft в Редмонде “Модель программирования MapReduce компании Google” (Google’s MapReduce Programming Model) [16].

MapReduce — модель распределенных вычислений, используемая для параллельных вычислений над очень большими, несколько Петабайт, наборами данных в компьютерных кластерах. В модели MapReduce обработка данных состоит из двух шагов: Map и Reduce. На Map-шаге происходит предварительная обработка входных данных. Для этого один из компьютеров, в данном случае главный узел, получает входные данные задачи, разделяет их на части и передает другим компьютерам (рабочим узлам) для предварительной обработки. На Reduce-шаге происходит консолидация предварительно обработанных на Map-шаге данных. После консолидации главный узел получает ответы от рабочих узлов и на их основе формирует результат — реше-



**Рис. 5. Основные этапы реализации модели MapReduce**

ние задачи, которая изначально формулировалась [16, 17]. Основные этапы реализации модели MapReduce представлены на рис. 5.

Маршрут реализации процедур MapReduce можно представить в следующем виде.

### **Маршрут обработки данных по модели MapReduce**

#### **Обработка входных данных.**

*Реализуемая функция:* входные данные делятся на блоки предопределенного размера (от 16 до 128 МБ) — *сплиты* (от англ. split).

*Действие:* MapReduceFramework закрепляет за каждой функцией Map определенный сплит.

#### **Выполнение Map-функций.**

*Реализуемая функция:* каждая функция Map получает на вход список пар “ключ-значение”  $\langle k, v \rangle$ , обрабатывает их и на выходе получает ноль или более пар  $\langle k', v' \rangle$ , являющихся промежуточным результатом:  $\text{map}(k, v) \rightarrow [(k', v')]$ , где  $k'$  — в общем случае произвольный ключ, несовпадающий с  $k$ .

*Действие:* все операции  $\text{map}()$  выполняются параллельно и не зависят от результатов работы друг друга. Каждая функция  $\text{map}()$  получает на вход свой уникальный набор данных, не повторяющийся ни для какой другой функции  $\text{map}()$ .

#### **Комбинирование промежуточных значений (Partition/Combine).**

*Реализуемая функция:* распределение промежуточных результатов, по-



лученных на этапе map, по reduce-заданиям:  $(k', \text{reducers\_count}) \rightarrow \text{reducer\_id}$ , где  $\text{reducers\_count}$  — число узлов, на которых запускается операция свертки;  $\text{reducer\_id}$  — идентификатор целевого узла  $[(k', v')] \rightarrow (k', [v'])$ .

*Действие:* в простейшем случае,  $\text{reducer\_id} = \text{hash}(k') \bmod \text{reducers\_count}$ , где  $\text{hash}(k')$  — хеш-функция от ключа  $k'$ ,  $A \bmod B$  — вычисление остатка от деления  $A$  на  $B$ .

Основная цель этапа partition — это балансировка нагрузки. Некорректно реализованная функция может привести к неравномерному распределению данных между reduce-узлами.

Данная формула — это частный пример, в котором для распределения данных по узлам свертки используется хеш-функция. Она позволяет равномерно распределять нагрузку между узлами, которые будут выполнять свертку. Если, например, вместо использования хеш-функции поставить статическое значение, то все данные попадут на один узел свертки. Таким образом, распределение данных будет неравномерным. Работать будет только один узел, остальные будут “простаивать”. В некоторых случаях имеет смысл разработать свою функцию распределения нагрузки. Например, для распределения данных по месяцам. Если в качестве ключа  $k'$  будет использоваться порядковый номер месяца в году (1...12), а число узлов для операции будет выставлено на 12 (чтобы каждый узел обрабатывал свой месяц), то чтобы выполнить свертку по каждому месяцу на одном узле, функция вычисления номера узла может быть следующей:  $\text{reducer\_id} = k' \bmod \text{reducers\_count}$ . Некорректно реализованная функция может привести к неравномерному распределению данных между reduce-узлами. Функция *combine* запускается после map-фазы. В ней происходит промежуточная свертка локальных по отношению к функции map значений. Основное значение функции *combine* — комбинирование промежуточных данных, что, в свою очередь, ведет к уменьшению объема передаваемой между узлами информации.

### **Выполнение операций Copy / Compare / Merge.**

*Реализуемая функция:*  $\text{compare}(k'n, k'n + 1) \rightarrow \{-1, 0, +1\}$ .

*Действие:* *copy* — копирование данных, полученных в результате работы функций map и *combine* (если такая была определена), с map-узлов на reduce-узлы;

*compare (или sort)* — сортировка, группировка по ключу  $k$  полученных в результате операции *copy* промежуточных значений на reduce-узле.

*merge* — “слияние” данных, полученных от разных узлов, для операции свертки.

### **Выполнение функции Reduce.**

*Реализуемая функция:* *framework* вызывает функцию *reduce* для каждого уникального ключа  $k'$  в отсортированном списке значений.

$\text{Reduce}(k', [v']) \rightarrow [v'']$ .

*Действие:* все операции `reduce()` выполняются параллельно и не зависят от результатов работы друг друга. Таким образом, результаты работы каждой функции `reduce()` пишутся в отдельный выходной поток.

### **Формирование выходного потока — Output write.**

*Реализуемая функция:* результаты, полученные на этапе `reduce`, записываются в выходной поток.

*Действие:* каждый `reduce`-узел пишет в собственный выходной поток.

Преимущество модели MapReduce заключается в том, что она позволяет распределено проводить операции предварительной обработки и консолидации. Операции предварительной обработки работают независимо друг от друга и могут выполняться параллельно. Аналогично, множество рабочих узлов могут осуществлять консолидацию. Так, MapReduce может быть использован для сортировки петабайта данных, что займет всего лишь несколько часов, в то время как при традиционном подходе к сортировке это займет несколько дней или недель. Параллелизм также дает некоторые возможности восстановления серверов после частичных сбоев: если в рабочем узле, выполняющем операцию предварительной обработки или консолидации, возникает сбой, то его работа может быть передана другому рабочему узлу [17].

**3. Большие данные применительно к обработке метеоинформации.** В случае с метеоданными, которые поступают лавинообразно, задача получения и анализа метеоинформации идеально подходит для решения средствами “больших данных”. Как уже было отмечено, необходимо обрабатывать около 72 ГБ сырых данных в день. Если для задач анализа необходимо накапливать данные, то за год успеет набежать 25 ТБ.

Метеоинформация приходит из разных источников. Даже если это будут файлы формата GRIB версии 2 их внутренний формат может быть различным (например, разный тип сетки). Кроме того, информацию о погоде могут поставлять частные компании, в таком случае формат данных может быть абсолютно непредсказуем. Технологии “больших данных” позволяют проверять и согласовывать поступающие данные с уже имеющимися данными. Причем они могут храниться как в реляционной базе данных, так и файлах файловой системы.

Одно из важнейших преимуществ такого подхода заключается в стоимости. Модульная организация вычислительного узла в случае нехватки места для хранения данных либо процессорных мощностей позволяет добавить в кластер новое оборудование без простоя системы и необходимости выкидывать старое оборудование.

**4. Особенности применения фреймворк с открытым исходным кодом Apache Hadoop** [18], в котором реализована вычислительная

модель MapReduce. Hadoop является проектом фонда Apache Software Foundation, и набором утилит, библиотек, программным каркасом для разработки и выполнения распределенных программ, работающих на кластерах из сотен и тысяч узлов.

Hadoop начинался с проекта Nutch [18]. Группа разработчиков попыталась построить систему веб-поиска с открытым кодом, однако проблемы с обработкой данных стали возникать даже на относительно небольшом множестве компьютеров. После того как компания Google опубликовала статьи о Google File System (GFS) и MapReduce, направление работы прояснилось. Системы Google разрабатывались для решения именно тех проблем, которые возникали с Nutch. И тогда два разработчика начали в свободное время работать над воссозданием этих систем как составной части Nutch.

В результате удалось заставить Nutch работать на 20 машинах, но вскоре стало ясно, что в огромных масштабах Web речь идет о выполнении на тысячах компьютеров, и что еще важнее — объем работы был слишком большим для двух программистов с неполным рабочим днем. Примерно в это же время компания Yahoo! заинтересовалась проектом и быстро собрала группу для продолжения разработки. Группа разработчиков выделила часть Nutch, относящуюся к распределенным вычислениям, в отдельный подпроект, который получил название Hadoop (рис. 6). При поддержке Yahoo! проект Hadoop вскоре превратился в технологию, способную работать в масштабах Web [18].

Поскольку Big Data подразумевает быстрое поступление большого объема данных, традиционные системы хранения на основе СУБД не успевают сохранить этот поток. Нужна простая и быстрая система, которая позволит с минимальными накладными расходами сохранять поток поступающих слабоструктурированных данных. В Hadoop для этого используется отказоустойчивая распределенная файловая система HDFS (Hadoop Distributed File System). Она разбивает поступающие данные на блоки и каждый блок попадает на отведенные ему место в пуле серверов, причем обычно хранится не менее трех копий блоков, что обеспечивает высокую надежность хранения. Если HDFS обнаруживает исчезновение блока, то она автоматически восстанавливает резервную копию, снова доводя их число до трех. Преимущество HDFS по сравнению с традиционной технологией RAID состоит в том, что не требуется специальной аппаратуры, а достаточно простого набора серверов, что особенно привлекательно при использовании облачных инфраструктур. Однако за все приходится платить: HDFS менее эффективно использует дисковое пространство, но в условиях постоянного снижения цен на диски это не является серьезным ограничением [19–21].

Разработчику приложения для Hadoop MapReduce необходимо реализовать базовый обработчик, который на каждом вычислительном

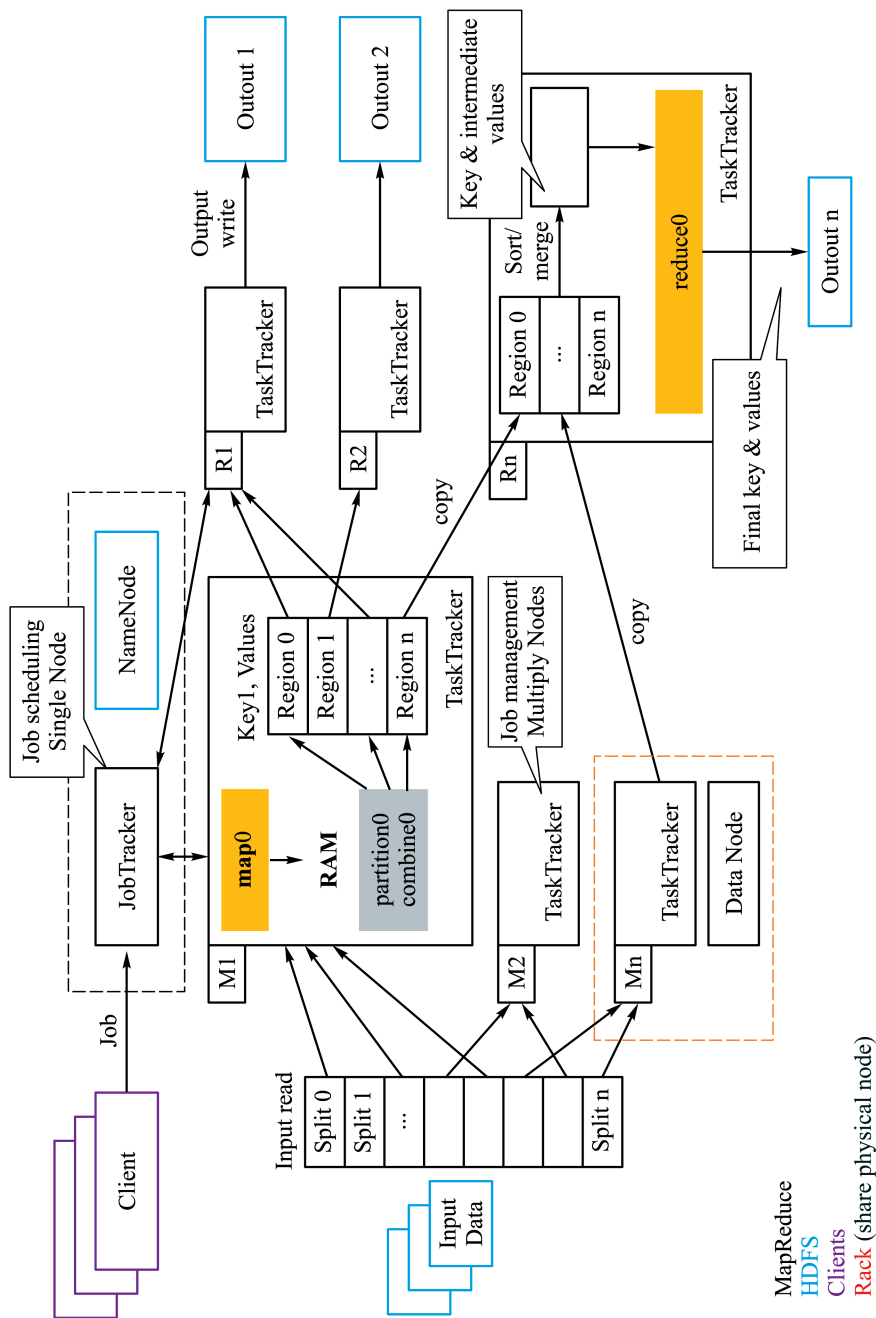


Рис. 6. Архитектура Apache Hadoop

узле кластера обеспечит преобразование исходных пар ключ/значение в промежуточный набор пар ключ/значение (класс, реализующий интерфейс Mapper), и обработчик, сводящий промежуточный набор пар в окончательный, сокращенный набор (класс, реализующий интерфейс Reducer) [18].

Все остальные фазы выполняются Hadoop без дополнительного кодирования со стороны разработчика. Кроме того, среда выполнения Hadoop MapReduce выполняет следующие функции: планирование и распараллеливание заданий; перенос заданий к данным; синхронизация выполнения заданий; перехват “проваленных” заданий; обработка отказов выполнения заданий и перезапуск проваленных заданий; оптимизация сетевых взаимодействий.

Hadoop MapReduce использует архитектуру “master-worker”, где master — единственный экземпляр управляющего процесса (JobTracker), как правило, запущенный на отдельной машине (вычислительном узле). Worker-процессы — это произвольное множество процессов TaskTracker, исполняющихся на DataNode. JobTracker и TaskTracker формируются над уровнем хранения HDFS. TaskTracker исполняются в соответствии с принципом “данные близко”, т.е. процесс TaskTracker располагается максимально близко с узлом DataNode, данные которого обрабатываются.

Приведенные принципы расположения JobTracker- и TaskTracker-процессов позволяют существенно сократить объемы передаваемых по сети данных и сетевые задержки, связанные с передачей этих данных, — основные “узкие места” производительности в современных распределенных системах.

**Заключение.** Планирование и координация воздушного движения — один из важнейших элементов в сфере авиации, помогающий эффективно использовать имеющийся ресурс воздушного пространства и наземных служб, при этом сохраняя безопасность перемещения ВС. Для извлечения максимальной пользы из имеющихся ресурсов без ущерба для авиации, необходимо иметь точные данные о внешних факторах, которые влияют на ВС. Чем более полной будет картина текущей метеобстановки, тем точнее аналитики смогут сделать прогнозы на длительный срок, которые в последствии будут использоваться при планировании использования воздушного пространства. Однако в настоящий момент современные системы используют устаревшие механизмы обмена метеоданными, в то время как сбор метеоданных осуществляется с помощью современного оборудования. Это несоответствие систем сбора, передачи и обработки информации приводит к тому, что данные, получаемые системами планирования, имеют низкое качество, что влияет на точность прогнозов, эффективность использования воздушных ресурсов и повышает вероятность возникновения авиакатастроф.

Все перечисленное возможно при использовании Hadoop кластера для получения, обработки и хранения метеоинформации. В апреле 2008 года технология Hadoop побила мировой рекорд по скорости сортировки терабайта данных. На 910-узловом кластере один терабайт был отсортирован за 209 с (чуть менее 3,5 мин), тогда как рекорд предыдущего года составлял 297 с. В ноябре того же года компания Google сообщила, что ее реализация MapReduce отсортировала один терабайт за 68 с. В мае 2009 года было объявлено, что рабочая группа в Yahoo! использовала Hadoop для сортировки одного терабайта за 62 с [18].

Современные технологии, позволяют минимизировать простой текущего оборудования. Это позволяет получать, обрабатывать и хранить значительно большие объемы данных, чем 10 лет назад. В случае возникновения нехватки вычислительных мощностей или места на дисках не потребуется полностью обновлять парк оборудования, необходимо добавить нужное количество ресурсов к уже имеющимся, без простоя системы. Рассматриваемые технологии не требуют замены существующего оборудования, что позволяет эффективнее использовать финансовые ресурсы. Использование технологий “большие данные” позволило увеличить точность расчета четырехмерной траектории полета ВС и, как следствие, оптимизировать использование воздушного пространства, а также повысить безопасность российской авиации.

*Работа выполнена при частичной поддержке Гранта РФФИ № 12-07-31151 в рамках программы “Мой первый грант”.*

## ЛИТЕРАТУРА

1. *Воздушный кодекс Российской Федерации*. М.: Омега-Л, 2005. 64 с.
2. *Сборник аэронавигационной информации Российской Федерации*. М.: ЦАИ ГА, 2008.
3. *Новиков П.В.* Алгоритм высокоточных вычислений метеодобавки скорости ветра для участка полета воздушного средства // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 11. С. 5–7. URL: <http://elibrary.ru/item.asp?id=17105440>.
4. *Всемирная Метеорологическая Организация. Технический регламент: Сб. основных документов. № 2. Т. 2: Метеорологическое обслуживание международной аэронавигации / Всемирная Метеорологическая Организация. Женева : Секретариат ВМО, 2007. 180 с.*
5. *Производство полетов воздушных судов // Международная организация гражданской авиации – ИКАО. 2006. 386 с.*
6. *Наставление по кодам. Т. 1.2: Международные коды / Всемирная Метеорологическая Организация. Женева: Секретариат ВМО, 2008. С. 25–192.*
7. *Ривкин А.М.* Модель полета воздушных судов на эшелоне // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2011. № 11. С. 15.
8. *Ривкин А.М.* Кроссплатформенный конвертор GRIB формата метеоданных для системы управления полетами по эшелонам. М.: Изд-во МГТУ им. Н.Э. Баумана, 2012. 97 с.

9. Власов А.И. Пространственная модель оценки эволюции методов визуального проектирования сложных систем // Датчики и системы. 2013. № 9 (172). С. 10–28.
10. Черняк Л. Большие Данные — новая теория и практика // Открытые системы. СУБД. М.: Открытые системы, 2011. № 10. Электронный ресурс. Режим доступа: <http://www.osp.ru/os/2011/10/13010990/> (дата обращения 01.12.2013).
11. Dana Blankenhorn. Shared nothing coming to open source. ZDNet (27 February 2006). Электронный ресурс. Режим доступа: <http://www.zdnet.com/blog/open-source/shared-nothing-coming-to-open-source/580> (дата обращения 01.12.2013).
12. *The Case for Shared Nothing Architecture* by Michael Stonebraker [Originally published in Database Engineering. Vol. 9. No. 1 (1986)]. С. 1–5.
13. Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. Электронный ресурс. Режим доступа: <http://www.gartner.com/newsroom/id/1731916> (дата обращения 01.12.2013).
14. Черняк Л. Смутное время СУБД // Открытые системы. 2012. № 2. Электронный ресурс. Режим доступа: <http://www.osp.ru/os/2012/02/13014107/> (дата обращения 01.12.2013).
15. Оленин О. NoSQL: назад в будущее // Открытые системы. 2012. № 2. Электронный ресурс. Режим доступа: <http://www.osp.ru/os/2012/02/13012856/> (дата обращения 01.12.2013).
16. Черняк Л. MapReduce — будущее баз данных // Открытые системы. СУБД. 2009. № 02. Электронный ресурс. Режим доступа: <http://www.osp.ru/os/2009/02/7322603/> (дата обращения 01.12.2013).
17. Dean J. and Ghemawat S. MapReduce: Simplified data processing on large clusters. In Proceedings of the Sixth Conference on Operating System Design and Implementation (Berkeley, CA, 2004). С. 137–149.
18. Уайт Том. Hadoop. Подробное руководство = Hadoop: The Definitive Guide. СПб.: Питер, 2013. С. 19–435.
19. Черняк Л. MapReduce — Hadoop против Big Data // Открытые системы. СУБД. 2010. № 07. Электронный ресурс. Режим доступа: <http://www.osp.ru/os/2010/07/13004186/> (дата обращения 01.12.2013).
20. Власов А.И., Цыганов И.Г. Адаптивная фильтрация информационных потоков в корпоративных системах на основе механизма голосования пользователей // Информационные технологии. 2004. № 9. С. 12.
21. Preimesberger, Chris Hadoop, Yahoo, 'Big Data' Brighten BI Future. EWeek (15 August 2011). С. 1–6.

## REFERENCES

- [1] Air Code of the Russian Federation. Moscow, Omega-L Publ., 2005. 64 p.
- [2] Aeronautical Information Publication of the Russian Federation. Moscow, CAICA 2008.
- [3] Novikov P.V. Algorithm precision calculations meteorobavki wind speed for the flight portion of the air assets. *Nauka i obrazovanie. MGTU im. N.E. Bauman.* [Science and Education of the Bauman MSTU. Electronic Journal], 2012, no. 11, pp. 5–7. Available at: <http://elibrary.ru/item.asp?id=17105440>
- [4] World Meteorological Organization. Technical Regulations: Sat key documents. No. 2, vol. 2. Meteorological Service for International Air Navigation. *World Meteorological Organization.* Geneva, WMO Secretariat, 2007. 180 p.
- [5] Aircraft Operations. *International Civil Aviation Organization – ICAO*, 2006. 386 p.
- [6] Manual on Codes. Vol. 1.2. International Codes. *World Meteorological Organization.* Geneva, WMO Secretariat, 2008, pp. 25–192.

- [7] Rivkin A.M. Model aircraft flight at flight level. *Nauka i obrazovanie. MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU. Electronic Journal], 2011, no. 11, pp. 15–15.
- [8] Rivkin A.M. Krossplatformennyy konvertor GRIB formata metodannykh dlya sistemy upravleniya poletami po eshelonam [Cross-platform format converter GRIB weather data for flight control system by tiers]. Moscow, MGTU im. N.E. Baumana Publ., 2012. 97 p.
- [9] Vlasov A.I. Spatial model of an assessment of evolution of methods of visual design of difficult systems. *Datchiki i sistemy* [Sensors and systems], 2013, no. 9 (172), pp. 10–28 (in Russ.).
- [10] Chernyak L. Big Data — a new theory and practice. *Otkrytye sistemy. SUBD* [Open Systems. DBMS], 2011, no. 10. Available at: <http://www.osp.ru/os/2011/10/13010990/> (accessed 01.12.2013).
- [11] Dana Blankenhorn Shared nothing coming to open source. ZDNet (27 February 2006). Available at: <http://www.zdnet.com/blog/open-source/shared-nothing-coming-to-open-source/580> (accessed 01.12.2013).
- [12] The Case for Shared Nothing Architecture by Michael Stonebraker [Originally published in Database Engineering, vol. 9, no. 1 (1986)], pp. 1–5.
- [13] Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. Available at: <http://www.gartner.com/newsroom/id/1731916> (accessed 01.12.2013).
- [14] Chernyak L. Troubles DBMS. *Otkrytye sistemy* [Open Systems], 2012, no. 2. Available at: <http://www.osp.ru/os/2012/02/13014107/> (accessed 01.12.2013).
- [15] Olenin O. NoSQL: back to the future. *Otkrytye sistemy* [Open Systems], 2012, no. 2. Available at: <http://www.osp.ru/os/2012/02/13012856/> (accessed 01.12.2013).
- [16] Chernyak L. MapReduce — future database. *Otkrytye sistemy. SUBD* [Open Systems. DBMS], 2009, no. 02. Available at: <http://www.osp.ru/os/2009/02/7322603/> (accessed 01.12.2013).
- [17] Dean J., Ghemawat S. MapReduce: Simplified data processing on large clusters. *Proceedings of the Sixth Conference on Operating System Design and Implementation*. Berkeley, CA, 2004, pp. 137–149.
- [18] White T. Hadoop. Detailed guidance = Hadoop: The Definitive Guide. St. Petersburg, Peter, 2013, pp. 19–435.
- [19] Chernyak L. MapReduce — Hadoop against Big Data. *Otkrytye sistemy. SUBD* [Open Systems. DBMS], 2010, no. 07. Available at: <http://www.osp.ru/os/2010/07/13004186/> (accessed 01.12.2013).
- [20] Vlasov A.I. Tsyganov I.G. Adaptive filtration of information streams in corporate systems on the basis of the mechanism of vote of users. *Informatsionnye tekhnologii* [Information technologies], 2004, no. 9, p. 12 (in Russ.).
- [21] Preimesberger Ch. Hadoop, Yahoo, 'Big Data' Brighten BI Future. EWeek (15 August 2011), pp. 1–6.

Статья поступила в редакцию 30.06.2015

Власов Андрей Игоревич — канд. техн. наук, доцент кафедры “Проектирование и технология производства электронной аппаратуры” МГТУ им. Н.Э. Баумана. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Vlasov A.I. — Ph.D. (Eng.), Associate Professor, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University, Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.



Новиков Павел Васильевич — магистрант кафедры “Проектирование и технология производства электронной аппаратуры” МГТУ им. Н.Э. Баумана.

МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Novikov P.V. — Master’s Degree Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University.

Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

Ривкин Андрей Маркович — магистрант кафедры “Проектирование и технология производства электронной аппаратуры” МГТУ им. Н.Э. Баумана.

МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

Rivkin A.M. — Master’s Degree Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University.

Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

**Просьба ссылаться на эту статью следующим образом:**

Власов А.И., Новиков П.В., Ривкин А.М. Особенности планирования воздушного движения с использованием синоптических карт, построенных с применением технологий Big Data // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2015. № 6. С. 46–62.

**Please cite this article in English as:**

Vlasov A.I., Novikov P.V., Rivkin A.M. Features of planning air traffic using weather maps constructed with application of the Big Data technologies. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Bauman, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2015, no. 6, pp. 46–62.