

УДК 021.8 + 025.1

ББК 78.34

СЕТЕЦЕНТРИЧЕСКИЙ ПОДХОД К СОЗДАНИЮ РАСПРЕДЕЛЕННЫХ СИСТЕМ ОБРАБОТКИ БОЛЬШИХ МАССИВОВ ДАННЫХ В НАУЧНЫХ РАСЧЁТАХ

Душкин Д. Н., Мошников А. С.

*(Учреждение Российской академии наук Институт
проблем управления РАН, Москва)*

Пухова А. Г.

(École polytechnique fédérale de Lausanne, Lausanne)

Статья посвящена исследованию применения сетецентрического подхода для проведения научных работ, оперирующих большими массивами данных; созданию концептуальной модели интероперабельного программного комплекса, совмещающего преимущества использования распределенной базы данных и математического пакета Matlab.

Ключевые слова: сетецентрический подход, научные вычисления, обработка данных, применения Matlab.

1. Введение

При решении ряда научно-исследовательских и практических задач часто возникает проблема обработки больших массивов данных. Обычно результаты измерений хранятся в неструктурированном виде, что вызывает необходимость применения специализированных программных средств анализа существующего набора данных и выемки необходимых. В частности, данные могут быть слабо структурированными, в силу большой длительности эксперимента и зашумления естественного не совершенства лабораторной базы.

Основная особенность этих задач заключается в том, что обрабатываемые данные из-за большого объема не могут быть целиком помещены в оперативную память вычислительной

машины. Также нередко может возникать проблема, связанная с ограниченностью памяти компьютера, предназначенного для размещения данных.

Цели данной статьи: исследование применения сетецентрического подхода для проведения научных работ, оперирующих большими массивами данных; создание концептуальной модели программного комплекса, решающего поставленные задачи.

В описываемой модели исходными данными является массив числовых значений, хранящихся в текстовом файле. Данные импортируются в распределенную базу данных (РБД), проводятся необходимые вычисления с использованием этих данных, результаты которых также сохраняются в РБД и отображаются на экране.

Исходя из сетецентрической концепции построения [5], основными характеристиками системы являются: работа в сетевой среде (в т.ч. возможность доступа посредством сети Интернет), высокая скорость обработки данных и полная интероперабельность в рамках действующих стандартов. Полнота реализации перечисленных выше характеристик и ряда других [5] позволяет судить об уровне сетецентричности системы.

Аппаратные средства ограничены 2-3 ПК среднего уровня производительности, однако в перспективе описываемый подход может использоваться в кластерных системах и системах «облачных» вычислений.

2. Постановка задачи

1.1. ПРИМЕНЕНИЕ MATLAB

В практической деятельности научные работники различных направлений используют математические программы для своих целей. Широкое применение нашёл пакет *Matlab*.

Matlab обладает возможностью для создания специальных наборов инструментов, расширяющих его функциональность. Наборы инструментов представляют собой коллекции функций, написанных на языке *Matlab* для решения определённого класса задач. Наиболее широко используются следующие инструменты:

1. Цифровая обработка сигналов, изображений и данных.
2. Системы управления.
3. Финансовый анализ.
4. Анализ и синтез географических карт.
5. Визуализация и представление данных.
6. Взаимодействие с внешними программными продуктами.
7. Базы данных.
8. Научные и математические пакеты.
9. Нейронные сети.
10. Нечёткая логика.
11. Символьные вычисления.

1.2. ПРИМЕНЕНИЕ ДОПОЛНИТЕЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ

В основе сетецентрического подхода [4] лежит концепция создания новых информационных систем, обладающих свойствами интероперабельности, автономности, отказоустойчивости и ряда других, с использованием существующих аппаратных и программных средств. Достижение этой цели возможно за счет использования программной архитектуры, реализующей стандартный интерфейс доступа к ресурсам системы.

С 2005 г. этот подход получил широкое распространение в корпоративных системах, поскольку позволил унифицировать разработку различных элементов системы, при этом сохраняя обратную совместимость с наследственным ПО за счет написания для них стандартизированных интерфейсов.

В научной среде этот подход оправдан в свете того, что существует большое количество различных инструментов, используемых при проведении исследований, но каждый из этих инструментов предоставляет свой протокол для внешнего взаимодействия, т.н. *API* (англ. *Application programming interface* — интерфейс программирования приложений).

Основным техническим способом реализации сетецентрического подхода является сервис-ориентированная архитектура, использующая в своей основе стандарты *WSDL*, *SOAP*, *XML* и ряд спецификаций для построения программных доменов [5].

В работе будет рассматриваться использование языка программирования *Erlang* и РБД *Riak* [6], написанной на *Erlang*.

Выбор *Erlang* в качестве основной вычислительной платформы обусловлен следующими её основными особенностями:

1. Язык разрабатывается компанией *Ericsson* для работы в сетевой среде в рамках распределенных систем.
2. В стандартную поставку включен фреймворк *OTP (open telecom platform)* – набор библиотек для разработки сетевых приложений. Надежность *Erlang* и *OTP* подтверждена многими телекоммуникационными компаниями мира, такими как *T-Mobile*, *Nortel* и др.
3. Сопоставимая с *MPI* скорость работы в сетевой среде в рамках распределенных приложений [3].
4. Использование динамической типизации и простого синтаксиса языка разработка систем на *Erlang* занимает меньшее время, чем на *MPI*.
5. В настоящее время *Erlang* является *open-source* продуктом.

РБД *Riak* является *open-source* реализацией РБД *Amazon Dynamo*, описанной в соответствующей статье [2]. *Riak* позволяет за короткое время реализовывать кластерные системы хранения и обработки данных.

2. Пример задачи

2.1. ИСХОДНЫЕ ДАННЫЕ И ОСНОВНЫЕ ФУНКЦИИ

В зависимости от конкретного приложения исходные данные могут быть поданы в различных форматах. С другой стороны, для выполнения преобразования наиболее удобным является формат представления данных, при котором отсчеты записаны последовательно в виде чисел с плавающей запятой в двоичном формате. В таком случае становится возможным выполнять вычисления непосредственно после чтения фрагмента файла. Будем считать, что эти данные находятся в текстовом файле (.txt) и представляют с собой двумерный массив размерностью $N \times M$, где N – число строк, M – число столбцов. Требуется преобразовать исходные данные из файла в рабочее пространство *Matlab*, после этого появляется возможность использования широкого набора функций, которыми обладает «матричная лаборатория».

В первую очередь необходимо провести интерполяцию данных, после этого требуется отфильтровать данные и сгладить выбросы. На финальном этапе требуется провести Фурье преобразование и представить исходные данные в виде спектральной плотности распределения. Для решения этой задачи используется быстрое Фурье-преобразование, операция, сопоставляющая функции вещественной переменной другую функцию вещественной переменной.

Средства *Matlab* полностью позволяют решать поставленную задачу, но возможности для экспорта данных из файлов формата *.txt* ограничены размером не более 60 Мб.

Применение, описанной выше нереляционной РБД, позволяет решить проблему хранения данных. Использование РБД так же обеспечит предоставление сетевого доступа к результатам вычислений и обеспечение сохранности данных.

2.2. ПРИМЕНЕНИЕ ДОПОЛНИТЕЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ

Использование системы, в состав которой входит РБД, позволяет хранить большое количество данных, при этом, не беспокоясь об их сохранности и возможностях дальнейшего наращивания ресурсов системы, благодаря встроенной возможности горизонтального масштабирования.

Современные РБД являются нереляционными (в литературе также популярен термин «*NoSQL*»), то есть в отличие от традиционных СУБД данные могут храниться в следующих структурах:

- Документах: *CouchDB*, *IBM Lotus Notes*, *MongoDB*,
- Графах: *Neo4j*,
- В парах «ключ-значение» (существует множество подвидов): *Riak*, *memcached*, *Google BigTable*,
- Объекты: *bd4o*.

Несмотря на различные типы хранения и представления данных у *NoSQL* БД есть общие черты, такие как: отсутствие предопределенной схемы данных (*schemeless*) и ориентация на возможность легкой горизонтальной масштабируемости.

Каждый тип *NoSQL* БД, согласно CAP теореме Брюера [1], может реализовывать только два из трех основных свойств:

- согласованность данных (англ. *consistency*) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;
- доступность (англ. *availability*) — любой запрос к распределённой системе завершается корректным откликом;
- устойчивость к разделению (англ. *partition tolerance*) — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

Выбранная РБД *Riak* реализует доступность и устойчивость к разделению, за счет механизмов репликации и распределенного хранилища данных типа «ключ-значение». Согласованность данных реализуется как «согласованность в конечном счете» (англ. *eventually consistent*) [7], т.е. после обновления данных на одном сервере в конечном счете данные будут обновлены и на других серверах через некоторый промежуток времени, т.н. окно несогласованности.

Такой подход позволяет хранить большие массивы данных, некритичных к несогласованности различных версий на различных серверах, т.е. таких данных, для которых не критична характеристика актуальности. Результаты исследований являются такими данными, поскольку внесенные однажды в БД они редко будут обновляться.

С экономической точки зрения такой подход оправдан тем, что РБД *Riak* является *open-source* решением, в отличие от расширения программного пакета *Matlab* для проведения распределенных вычислений, стоимость которого составляет от 3000\$.

3. Применение параллельных вычислений

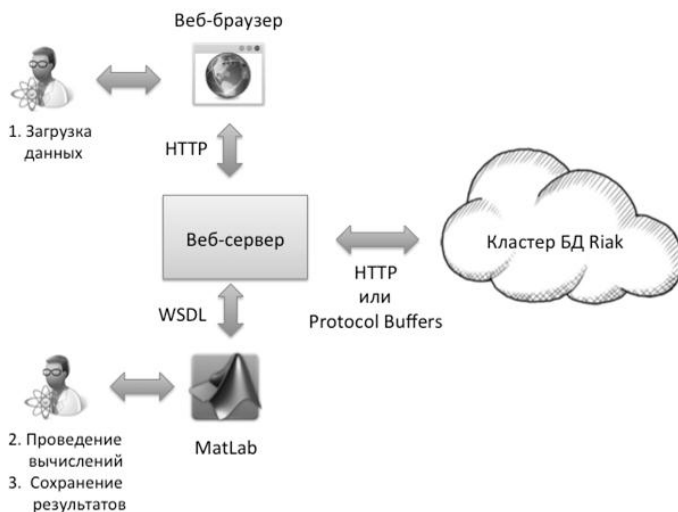


Рисунок 1. Концептуальная схема описываемого метода

На рис. 1 приведена концептуальная схема описываемого метода. Схему можно описать в контексте обработки запросов на выполнение различных задач.

Загрузка массива данных происходит с помощью обычного веб-браузера, установленного в любой современной операционной системе:

- По протоколу *HTTP* браузер получает с веб-сервера страницу с интерфейсом загрузки данных. Пользователь выбирает файл с данными на своем компьютере и веб-браузер отправляет их на веб-сервер.
- Веб-сервер обрабатывает данные, приводит их в форму для записи в БД и связывается с одним из серверов в кластере БД по протоколу *HTTP* или *Google Protocol Buffers* для отправки данных.
- Внутри кластера данные распределяются по узлам и создаются необходимые служебные данные (например, вторичные индексы).

Проведение вычислений происходит с помощью программы *MatLab*:

- Пользователь в программном пакете *MatLab* создает функцию для проведения необходимых вычислений с входными данными, полученными с веб-сервиса по протоколу *WSDL*.
- При запуске функции *Matlab* связывается с веб-сервером по протоколу *WSDL* для получения определенных данных.
- Веб-сервер обрабатывает входной запрос и связывается с одним из серверов в кластере БД для получения данных. Количество узлов, с которыми необходимо согласовать чтение данных, указывается в запросе. Обычно это 2 узла.
- Внутри кластера происходит формирование и отправка ответного массива данных с учетом параметров согласования.
- Веб-сервер возвращает данные *Matlab*.
- *Matlab* проводит необходимые вычисления, и результаты вычислений опционально сохраняет в кластере БД с помощью веб-сервера.

При большом количестве пользователей схему можно дополнить простыми балансировщиками нагрузки на участках связи клиентов с веб-сервером и веб-сервера с кластером БД.

4. Заключение

В процессе разработки концептуальной модели системы были исследованы, с одной стороны, различные способы обработки больших массивов данных, стратегии предварительной выборки и алгоритмы сжатия; с другой — возможности применения на практике метода кратно-масштабного анализа.

Соответственно, результаты работы можно разделить на две части.

Во-первых, создан математический модуль использующий базовые блоки *Matlab* для обработки данных и преобразования результатов вычислений.

Во-вторых, разработана концептуальная модель системы, позволяющая организовать работу с большими массивами данных при проведении научных исследований.

Литература

1. BREWER, ERIC A. *Towards robust distributed systems*. – Portland, OR: ACM. – 2000. – Т. 19.
2. GIUSEPPE DECANDIA, DENIZ HASTORUN, MADAN JAMPANI, GUNAVARDHAN KAKULAPATI, AVINASH LAKSHMAN, ALEX PILCHIN, SWAMINATHAN SIVASUBRAMANIAN, PETER VOSSHALL AND WERNER VOGELS. *Dynamo: Amazon's Highly Available Key-value Store*. – Washington, USA: ACM. – 2007.
3. JOHN DAVID ERIKSEN. *Erlang Versus MPI – Final Results and Source Code*. – 2009. – URL: <http://jkndrkn.livejournal.com/205249.html>.
4. NCOIC *Lexicon*. – NCOIC.org. – 2008. – URL: <http://wiki.ncoic.net/index.php/LexiconV2.0>.
5. OSVALDS G., BOWLER M., JONES A. *NCOIC Interoperability Framework*. – NCOIC. – 2010. – P. 10
6. *Riak overview*. Basho technologies, inc. – URL: <http://www.basho.com/products/riak/overview.php>.
7. W. VOGELS. *Eventually Consistent* // ACM Queue. – 2008. – Т. 6. – № 6.

NETWORK-CENTRIC APPROACH TO CREATION DISTRIBUTED SYSTEMS OF DATA PROCESSING IN SCIENTIFIC CALCULATIONS

Dimitry Dushkin, Institute of Control Sciences of RAS, Moscow, postgraduate student (ddushkin@asmon.ru).

Alexander Moshnikov, Institute of Control Sciences of RAS, Moscow, postgraduate student (moshnikov.alex@gmail.com).

Anna Puhova, Écolepolytechniquefédérale de Lausanne, Lausanne, magister (puhbot@mail.ru).

Abstract: Article is devoted application research network-centric approach for carrying out of the scientific works operating with the data; to creation of conceptual interoperable model of the program complex combining advantages of use of the distributed database and mathematical package MatLab.

Keywords: network-centric, network-centric approach, scientific calculations, data processing, Matlab applications.