28. Chen C., Sun J., Liu Y., Dong J., Zheng M. Formal modeling and validation of Stateflow diagrams // International Journal on Software Tools for Technology Transfer. 2012. V. 14. N 6. P. 653–671.
29. Малаховски Я.М., Корнеев Г.А. Применение зависимых систем типов со структурной индукцией для верификации реактивных программ // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 6 (82). С. 63–67.
30. Катериненко Р.С., Бессмертный И.А. Верификация данных в системах отслеживания задач с помощью продукционных правил // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 1 (83). С. 86–90.
31. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб: Наука, 1998. 617 с.
32. Cardei I., Jha R., Cardei M., Pavan A. Hierarchical architecture for real-time adaptive resource management // Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms. Secaucus, NJ, USA: Springer-Verlag, 2000. P. 415–434.
33. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. СПб: Питер, 2010. 176 с.
34. Dijkstra E.W. Guarded commands, non-determinacy and formal derivation of programs // Communications of the ACM. 1975. V. 18. N 8. P. 453–457 [Электронный ресурс]. Режим доступа: http://www.cs.virginia.edu/~weimer/615/reading/DijkstraGC.pdf, свободный. Яз. англ. (дата обращения 25.11.2013).

Лукин Михаил Андреевич – программист, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия, lukinma@gmail.com

Michael Lukin – programmer, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia, lukinma@gmail.com

УДК 004.65

АНАЛИЗ ДАННЫХ НА ОСНОВЕ ПЛАТФОРМЫ SQL-MAPREDUCE

А.А. Дергачев[a]

[a] Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия, dam600@mail.ru

Рассмотрены проблемы, связанные с применением реляционных СУБД в области анализа больших объемов данных, в том числе данных, предоставляемых для аналитики посредством веб-сервисов в Интернет.
Возможность их решения может быть представлена веб-ориентированной распределенной системой анализа данных, исполнительным ядром которой является процессор сервисных запросов. Функции такой системы аналогичны функциям реляционных СУБД, только применительно к веб-сервисам. Процессор сервисных запросов необходим для формирования и исполнения плана вызова веб-сервисов анализа данных. Эффективность такой веб-ориентированной системы зависит от эффективности плана вызова веб-сервисов и программной реализации веб-сервисов, основным элементом которых являются средства хранения анализируемых данных – реляционные СУБД. Развитию возможностей реляционных СУБД для анализа больших объемов данных и уделено основное внимание в данной работе, а именно – оценке перспективности реализации веб-сервисов анализа данных на основе платформы SQL/MapReduce. Для достижения поставленной цели в качестве прикладной была выбрана аналитическая задача, характерная для различных социальных сетей и веб-порталов, связанная с анализом данных об их посещаемости различными пользователями. В рамках практической части исследования был реализован алгоритм формирования плана вызова веб-сервисов для решения прикладной аналитической задачи и выполнен эксперимент, подтверждающий эффективность технологии SQL/MapReduce и перспективность применения ее при реализации веб-сервисов анализа данных.
Ключевые слова: анализ данных, веб-сервисы, SQL, MapReduce, СУБД.

DATA ANALYSIS BY SQL-MAPREDUCE PLATFORM

A. Dergachev[b]

[b] Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia, dam600@mail.ru

The paper deals with the problems related to the usage of relational database management system (RDBMS), mainly in the analysis of large data content, including data analysis based on web services in the Internet. A solution of these problems can be represented as a web-oriented distributed system of the data analysis with the processor of service requests as an executive kernel. The functions of such system are similar to the functions of relational DBMS, only with the usage of web services. The processor of service requests is responsible for planning of data analysis web services calls and their execution. The efficiency of such web-oriented system depends on the efficiency of web services calls plan and their program implementation where the basic element is the facilities of analyzed data storage – relational DBMS. The main attention is given to extension of functionality of relational DBMS for the analysis of large data content, in particular, the perspective estimation of web services data analysis implementation on the basis of SQL/MapReduce platform. With a view of obtaining this result, analytical task was chosen as an application-oriented part, typical for data analysis in various social networks and web portals, based on data analysis of users' attendance. In the practical part of this research the algorithm for planning of web services

calls was implemented for application-oriented task solution. SQL/MapReduce platform efficiency is confirmed by experimental results that show the opportunity of effective application for data analysis web services.
Keywords: data analysis, web services, SQL, MapReduce, DBMS.

Введение

Рынок аналитических инструментов на сегодняшний день широко представлен готовыми программными решениями. Однако программные продукты такого рода обычно сложно поддаются настройке под конкретные требования организации [1] и не имеют технической возможности доступа к данным, предоставляемым в Интернет посредством веб-сервисов. В связи с этим в последнее время возрос интерес к построению систем анализа данных на основе готовых аналитических платформ [2], позволяющих с использованием интегрированных в них средств создавать новые аналитические веб-ориентированные инструменты [3]. Такие аналитические системы представляют собой сложные программно-аппаратные комплексы, основным элементом которых являются средства распределенного хранения анализируемых данных [4–6]. Согласно статистике, в качестве таких средств на сегодняшний день наибольшее распространение получили систем управления базами данных (СУБД), среди которых лидирующие позиции, порядка 80–90%, занимают реляционные СУБД [7, 8]. Исходя из этого, в данной работе вопрос построения аналитической платформы рассматривается с позиций развития концепции реляционных СУБД как перспективной применительно к веб-сервисам [9].

Целью настоящей работы является подтверждение эффективности подхода к расширению парадигмы, заложенной в основе реляционных СУБД, моделью распределенных вычислений MapReduce, и его перспективности при реализации веб-сервисов, входящих в состав веб-ориентированной системы анализа данных.

Расширение языка SQL моделью распределенных вычислений MapReduce

Структурированный язык запросов SQL (Structured Query Language) и параллельные архитектуры СУБД на сегодняшний день уже не удовлетворяют требуемым показателям производительности по анализу данных [10]. Это связано с несколькими причинами.

Во-первых, одним из важнейших свойств языка SQL является свойство декларативности – в запросе указывается, какие данные необходимо извлечь или модифицировать, но не указывается, каким образом этот запрос должен быть обработан. Это значительно упрощает процесс формулировки SQL-запросов, но усложняет задачу оптимизации при формировании плана выполнения запроса, который позволил бы минимизировать время, необходимое для его выполнения. Одним из перспективных вариантов решения проблемы является расширение языка SQL, а именно, его интеграция с моделью распределенных вычислений MapReduce, которая используется для организации параллельной обработки больших объемов данных в компьютерных кластерах. Модель MapReduce была представлена компанией Google в 2004 г. [11]. Данная модель реализована в виде фреймворка, работающего поверх распределенной файловой системы GFS (Google File System), и широко применяется в программных продуктах самой компании Google. Однако, являясь сугубо проприетарной, она недоступна для сторонних разработчиков. Альтернативной свободно доступной реализацией стал проект сообщества Apache Software Foundation под названием Apache Hadoop. Фреймворк для реализации MapReduce-вычислений, который называется Hadoop MapReduce, работает поверх распределенной файловой системы HDFS (Hadoop Distributed File System), предназначенной для хранения файлов большого размера, поблочно распределенных между узлами вычислительного кластера. Все файловые блоки, кроме последнего, имеют одинаковый размер, и при этом каждый блок может быть размещен в нескольких узлах. Благодаря использованию механизмов репликации обеспечивается устойчивость распределенной системы к отказам отдельных узлов. Файлы в HDFS могут записываться лишь однажды, при этом не поддерживаются механизмы их модификации, и запись в файл в одно и то же время может вести только один процесс. Таким образом, реализована модель однократной записи в файл с последующим его многократным чтением, способствующая упрощению механизмов обеспечения целостности данных. Работа всей файловой системы находится под централизованным управлением узла имен, хранящего все метаданные о файлах системы, в том числе информацию об их размерах, размещении блоков и их реплик и т.д. Узел имен отвечает за обработку операций уровня файлов и каталогов, таких как открытие и закрытие файлов, манипуляцию каталогами. Сами блоки данных хранятся в серии узлов данных. Узлы данных отвечают за обработку операций по чтению и записи данных. Широкую известность в области информационных технологий модель MapReduce получила именно благодаря открытости и доступности реализации Hadoop, которая применяется в различных научных и исследовательских проектах, стимулируя тем самым разработчиков данной модели к постоянному ее совершенствованию. Однако важно заметить, что реализация Hadoop MapReduce полностью основана на спецификациях компании Google.

Во-вторых, концепция «мастер–работник», заложенная в архитектуре параллельных СУБД, также накладывает ряд ограничений [12]. Основной недостаток данных систем состоит в том, что процесс-мастер

является «узким местом» всей системы. Это связано с тем, что он осуществляет всю финальную обработку данных в последовательной форме, поскольку не все операции могут обрабатываться параллельно на стороне процессов-работников. Соответственно, чем больше данных возвращается от процессов-работников, тем дольше процесс-мастер будет осуществлять их финальную обработку, что может негативно сказаться на времени выполнения запроса в целом. При этом использование различных форм распараллеливания запросов между процессами-работниками приводит к усложнению задачи их оптимизации.

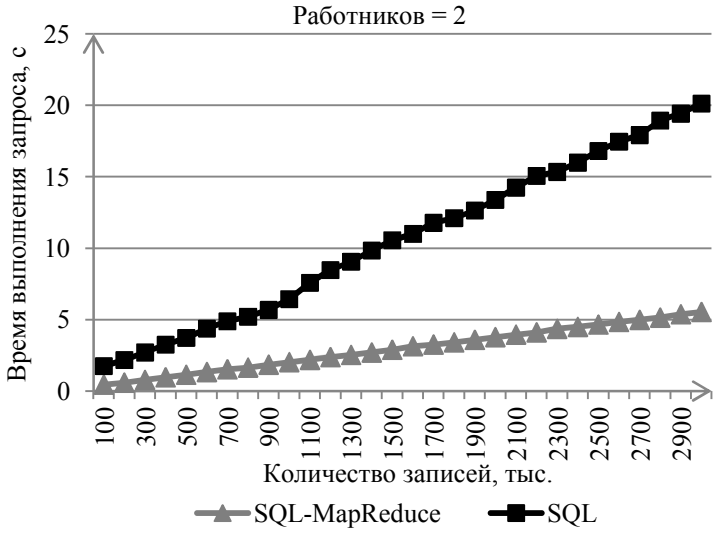
Все эти факторы указывают на необходимость поиска решений, которые позволили бы обойти ряд сложившихся ограничений по применению данных технологий в области анализа данных. Одним из таких решений является подход, предложенный компанией Aster Data Systems, приобретенной позже компанией Teradata. Он заключается в расширении возможностей языка SQL посредством хорошо распараллеливаемых табличных функций, которые можно вызывать прямо из операторов выборки. Для обеспечения работы данного механизма компанией была разработана технология под названием SQL/MapReduce, реализованная в SQL-ориентированной массивно-параллельной СУБД nCluster, которая поставляется в составе одноименной аналитической платформы SQL/MapReduce [13]. Согласно модели MapReduce, обработка данных осуществляется в две фазы. При выполнении фазы Map из набора входных данных происходит формирование промежуточных пар ключ–значение. Затем все пары с одинаковыми значениями промежуточных ключей предаются в фазу Reduce для их финальной обработки. Обработка данных организуется на основе использования двух типов функций – Map-функций и Reduce-функций. Функции могут быть вызваны из SQL-запросов в любом порядке произвольное количество раз с возможностью вложенных вызовов, в отличие от классической модели MapReduce, где исполнение этих фаз носит последовательный характер. Основным достоинством такого решения является реализация логики по обработке, сортировке и группировке данных внутри SQL/MapReduce-функций, которые исполняются на стороне работников. За счет этого значительно упрощаются сами SQL-запросы и, как следствие, задача их оптимизации. Также происходит уменьшение роли мастера, поскольку в данном случае он отвечает лишь за распределение задач между работниками, сбор от них всех готовых результатов и формирование итогового результата без какой-либо его дополнительной обработки на своей стороне. Все это позволяет значительно сократить время выполнения аналитического запроса, что наглядно демонстрируют результаты эксперимента, представленные ниже. Помимо этого, благодаря заимствованию принципов модели MapReduce, появилась возможность анализа полуструктурированных данных, таких как файлы-журналы и XML-файлы, а также неструктурированных данных, таких как простой текст, для анализа которых язык SQL и реляционные СУБД являются малопригодными в силу того, что они основываются на фиксированной схеме хранения данных.

Экспериментальная реализация SQL/MapReduce-функций

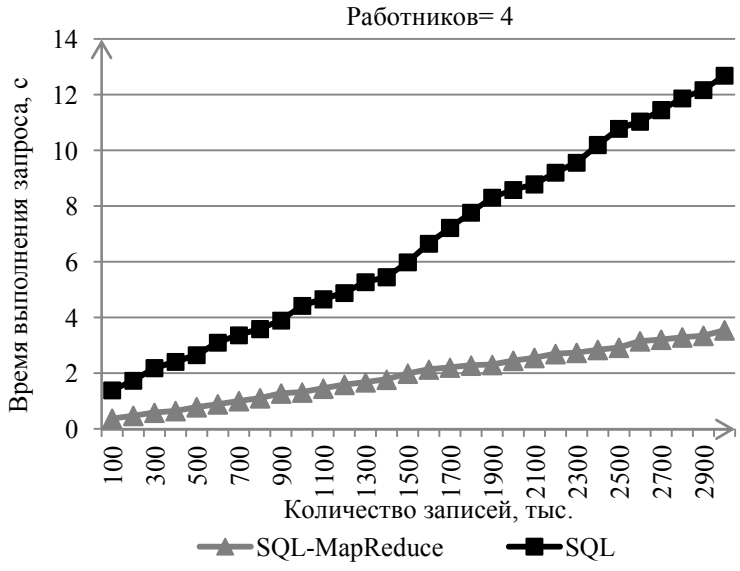
Для выполнения экспериментальной части исследования был создан программно-аппаратный комплекс на базе компьютера, оснащенного процессором AMD Phenom II Six-Core 1075T с тактовой чистотой 3 ГГц, 16 ГБ оперативной памяти и жестким диском емкостью 1 ТБ, под управлением 64-разрядной операционной системы Windows 7. Программная реализация системы организации доступа к веб-сервисам анализа данных развертывалась на сервере приложений Java EE GlassFish 3.1 и пробной версии СУБД nCluster компании Aster Data Systems. При создании модели были развернуты две виртуальные машины с использованием платформы VMware Workstation 9.0, которые затем были объединены в единую сеть. В качестве операционных систем на виртуальных машинах использовался 64-разрядный дистрибутив SUSE Linux Enterprise 11. Один из образов использовался для моделирования узла-распорядителя (мастера), а другой узла-исполнителя (рабочего). Такая довольно упрощенная конфигурация модели была обусловлена, во-первых, ограничением на количество моделируемых узлов кластера, накладываемым пробной версией реляционной СУБД nCluster, во-вторых, вычислительными ресурсами, располагаемыми в ходе выполнения данного эксперимента. В качестве прикладной была выбрана аналитическая задача, характерная для различных социальных сетей и веб-порталов, связанная с анализом данных об их посещаемости различными пользователями. Для проведения эксперимента был создан файл-журнал, содержащий информацию о времени посещения, идентификаторе пользователя и имени посещаемой страницы. Всего файл содержал информацию приблизительно о тридцати тысячах различных пользователей, которые посещали одиннадцать различных страниц на протяжении трех месяцев. Весь файл состоял из трех миллионов уникальных записей. На основе этих данных осуществлялся подсчет количества посещений пользователями различных страниц по часам для каждого дня каждого месяца. Для решения данной аналитической задачи было разработано два типа SQL-запросов. Один из них выполнял ее стандартными средствами языка SQL:

```
SELECT Год, Месяц, Число, Час_посещения, Страница,
COUNT(Пользователь) AS Количество_посещений
FROM (
SELECT TO_CHAR(DATESTAMP, 'YYYY') AS Год,
                TO_CHAR(DATESTAMP, 'MM') AS Месяц,
```

```
        TO_CHAR(DATESTAMP, 'DD') AS Число,
        TO_CHAR(DATESTAMP, 'HH24":00"') AS Час_посещения,
        PAGE AS Страница,
        CUSTOMER_ID Пользователь
    FROM test_table) AS sub_select
GROUP BY Год, Месяц, Число, Час_посещения, Страница
ORDER BY Страница, Год, Месяц, Число, Час_посещения
```



а



б

Рисунок. Графики времени выполнения простого SQL-запроса и запроса, содержащего вызов SQL/MapReduce-функции, при использовании двух (а) и четырех (б) работников

Обработка приведенного выше SQL-запроса осуществляется следующим образом. Сначала происходит выполнение вложенного подзапроса. В нем для разбиения входного значения, содержащего временную отметку вида «2013-03-17 16:35:59», используется встроенная функция TO_CHAR, которая на основе указанного формата (YYYY, MM, DD и т.д.) осуществляет выделение в качестве отдельных полей года, месяца, числа и времени в 24-часовом формате. Для большей наглядности запроса в нем были использованы псевдонимы, задаваемые с помощью оператора AS, такие, как Год, Месяц, Число и т.д. В результате подзапрос возвращает таблицу, содержащую все полученные значения в качестве отдельных столбцов, с добавлением к ним столбцов с именем страницы и идентификатором пользователя. Далее эта таблица передается в основной запрос. В нем осуществляется подсчет количества посещений с помощью оператора COUNT() по полю Пользователь, хранящего информацию об идентификаторе пользователя, по строкам, в которых содержатся одинаковые значения в полях Год, Месяц, Число, Час_посещения, Стра-

ница, с их последующей группировкой. В конце производится сортировка результирующих строк в алфавитном порядке по значению поля Страница и в порядке естественного возрастания значений полей Год, Месяц, Число, Час_посещения.

Другой запрос сформулирован с применением SQL/MapReduce-функции, которая была разработана с использованием языка программирования Java:

```
SELECT *
FROM LogAnalyzer(
        ON test_table
        PARTITION BY PAGE)
```

Вызываемая в запросе SQL/MapReduce-функция LogAnalyzer реализована как функция над разделами или же, если говорить в рамках терминологии технологии MapReduce, как Reduce-функция. При выполнении функции над разделами каждая группа строк, образованная на основе спецификации раздела PARTITION BY вызова функции, обрабатывается ровно одним экземпляром данной функции. При этом способе обработки данных экземпляр получает всю группу строк целиком. В данном случае разбиение на разделы осуществляется по имени страницы (PAGE). Таким образом, каждый экземпляр функции занимается обработкой всей информации о посещении пользователями только для одной конкретной страницы. Теперь вся логика по обработке, группировке и сортировке входных данных, а также формат результирующих данных определяются внутри функции LogAnalyzer. После своего исполнения функция возвращает на место ее вызова в основном запросе результирующую таблицу.

При проведении эксперимента осуществлялась оценка времени выполнения простого SQL-запроса и запроса, содержащего вызов функции при изменении количества входных строк данных, что отражено на рисунке. При этом модель была сконфигурирована таким образом, чтобы можно было поставить эксперимент с имитацией работы одного мастера и двух или четырех работников.

На рисунке представлены результаты эксперимента в виде графиков, отражающих время выполнения простого SQL-запроса и запроса, содержащего вызов SQL/MapReduce-функции. Графики строились на основе усредненных значений четырех измерений времени обработки для обоих запросов с точностью до микросекунд, которые выполнялись на каждом шаге эксперимента. Из них хорошо видно, что использование SQL/MapReduce-функции позволяет существенно уменьшить время выполнения запроса по обработке исходных данных в соответствии с поставленной задачей. Так, в данном случае время, которое проходит с момента запуска пользователем запроса на исполнение до того, когда ему будет возвращен результат, удалось уменьшить в среднем в 3,5 раза.

За счет увеличения же количества работников с двух до четырех удалось сократить время выполнения запроса в среднем лишь на 30%. Здесь сказываются накладные расходы, связанные с обработкой вызова функции на стороне мастера, а также координацией и назначением на выполнение ее экземпляров работниками.

Заключение

Полученные в ходе эксперимента положительные результаты подтверждают преимущества изменения логики обработки и принципов построения аналитического запроса, которые были реализованы при расширении концепции реляционной СУБД и языка SQL моделью распределенных вычислений MapReduce. Исходя из этого, можно сделать вывод, что применение технологии SQL/MapReduce для реализации веб-сервисов, входящих в состав веб-ориентированной системы анализа данных, является перспективным.

Следующим шагом развития концепции реляционных СУБД применительно к веб-сервисам может стать разработка перспективной архитектуры веб-ориентированной системы анализа данных, исполнительным ядром которой должен стать процессор сервисных запросов, нацеленный на формирование эффективного плана вызова веб-сервисов анализа данных.

Литература

1. Курочкин Д.Э., Бураков П.В. Задачи развития IT-инфраструктуры предприятия // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 2 (78). С. 74–77.
2. Марьин С.В., Ковальчук С.В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде // Изв. вузов. Приборостроение. 2011. Т. 54. № 10. С. 21–28.
3. Алексеев С.А. Формирование общего информационного ресурса в корпоративной сети социальной организационно-технической системе // Изв. вузов. Приборостроение. 2009. Т. 52. № 12. С. 8–11.
4. Кириллов В.В., Лукьянов Н.М. Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранилища данных // Научно-технический вестник СПбГУ ИТМО. 2008. № 11 (56). С. 9–16.

5. Новосельский В.Б., Павловская Т.А. Выбор и обоснование критерия эффективности при проектировании распределенных баз данных // Научно-технический вестник СПбГУ ИТМО. 2009. № 2 (60). С. 76–82.
6. Лукьянов Н.М., Дергачев А.М. Организация сетевого взаимодействия узлов распределенной системы хранения данных // Научно-технический вестник СПбГУ ИТМО. 2011. № 2 (72). С. 137–140.
7. DB-Engines. Ranking the popularity of database management systems. 2012 [Электронный ресурс]. Режим доступа: http://db-engines.com/en/blog_post/1, свободный. Яз. англ. (дата обращения 09.06.2013).
8. Зализняк Е. Рынок СУБД. 2009 [Электронный ресурс]. Режим доступа: http://www.cnews.ru/reviews/index.shtml?2005/08/15/184770_1, свободный. Яз. рус. (дата обращения 09.06.2013).
9. Дергачев А.М. Проблемы эффективного использования сетевых сервисов // Научно-технический вестник СПбГУ ИТМО. 2011. № 1 (71). С. 83–86.
10. Agrawal R., Ailamaki A., Bernstein P.A., Brewer E.A., Carey M.J., Chaudhuri S., Doan A., Florescu D., Franklin M.J., Garcia-Molina H., Gehrke J., Gruenwald L., Haas L.M., Halevy A.Y., Hellerstein J.M., Ioannidis Y.E., Korth H.F., Kossmann D., Madden S., Magoulas R., Ooi B.C., O'Reilly T., Ramakrishnan R., Sarawagi S., Stonebraker M., Szalay A.S., Weikum G. The Claremont Report on Database Research // Sigmod Record. 2008. V. 37. N 3. P. 9–19.
11. Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters // Proc. of the Sixth Symposium on Operating System Design and Implementation. San Francisco, CA, 2004. P. 137–150.
12. van der Lans R.F. Using SQL-MapReduce® for Advanced Analytical Queries [Электронный ресурс]. Режим доступа: http://www.asterdata.com/resources/ assets/ar_SQL-MapReduce_for_Advanced_Analytics.pdf, свободный. Яз. англ. (дата обращения 06.06.2013).
13. Friedman E., Pawlowski P., Cieslewicz J. SQL/MapReduce: A practical approach to self-describing, polymorphic, and parallelizable userdefined functions // Proc. of the 35th VLDB Conference. Lyon, France, 2009. P. 1402–1413.

Дергачев Александр Андреевич – аспирант, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия, dam600@mail.ru

Alexander Dergachev – postgraduate, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia, dam600@mail.ru

УДК 004.043, 004.5, 37.04

О ВЛИЯНИИ АДАПТИВНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ НА НАДЕЖНОСТЬ И ЭФФЕКТИВНОСТЬ ФУНКЦИОНИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Ю.О. Фуртат[a]

[a] Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, Киев, Украина, saodhar@ukr.net

В современных автоматизированных системах пользователи часто сталкиваются с проблемой информационной перегрузки из-за постоянно возрастающих объемов информации, требующей обработки за короткое время. Работа в таких условиях отрицательно сказывается на качестве работы операторов систем и на надежности самих систем.

Одним из подходов к решению задачи информационной перегрузки является создание для автоматизированных систем персонализированных интерфейсов, учитывающих особенности работы пользователей с информацией. Характеристики оператора системы, определяющие предпочитаемые им форму и темп представления информации, формируют когнитивный портрет пользователя.

Для диагностирования характеристик применяется или профессиональное тестирование с привлечением специалистов-психологов, или оперативное тестирование на рабочем месте пользователя. Второй вариант представляется более предпочтительным для использования в автоматизированных системах, поскольку не возникает проблемы нехватки специалистов–психологов. Составление когнитивного портрета при этом проводится в результате взаимодействия пользователя с программными средствами диагностирования, основанными на методиках когнитивной психологии.

Эффект от применения в автоматизированной системе персонализированного пользовательского интерфейса можно оценить, установив, как уменьшение времени реакции пользователя на критические события влияет на уровень надежности и эффективности функционирования системы. Для этого используются формулы теории надежности сложных автоматизированных систем, показывающие зависимость надежности системы от времени реагирования пользователя на критическое событие.

Ключевые слова: автоматизированная система, пользовательский интерфейс, персонализация, адаптация интерфейса, когнитивный профиль.