

А.А. Цветков

Мультиагентный подход к управлению операционной средой, содержащей сверхбольшие базы данных

АННОТАЦИЯ. Настоящая статья посвящена анализу проблем при эксплуатации операционных сред, использующих сверхбольшие базы данных.

Рассмотрена возможная архитектура среды на основе кооперации интеллектуальных/рациональных агентов, которые могут наблюдать за работой системы и, при необходимости, в значительной степени управлять ей в автоматическом режиме.

Ключевые слова и фразы: Сверхбольшая база данных, Интеллектуальный агент, Рациональный агент, Информационная безопасность

Введение

Возможность переводить в цифровую форму документы, книги, аудио и видео информацию, фотографии (в т.ч. с высоким разрешением) и др. привели к тому, что объем информации, хранимой в базах данных (далее БД), стал катастрофически увеличиваться. А это привело к появлению проблемы сверхбольших баз данных (далее, в соответствии с традицией принятой в публикациях, VLDB от английского термина Very Large Database): одно дело записать данные в БД, а другое – получить по запросу искомую информацию в разумное время.

Учитывая то, что объем информации постоянно увеличивается, определение VLDB не является строгим. Вот, например, одно из определений: VLDB – это БД, которая содержит экстремально большое количество кортежей или строк в БД, или, занимает экстремально большое физическое пространство системы хранения файловой системы (петабайты).

Как показано в [1], если в 2005 году VLDB считались те БД, которые имели объем 100 ТБ, то сейчас это БД с объемом порядка ПБ. Например, Национальный репозиторий США увеличивается на 28 ПБ ежегодно.

Кроме того, в [1] приведены основные причины проблемы VLDB – быстродействия:

- (1) Проблемы передачи больших объемов информации по сети, будь то Интернет или корпоративная сеть;
- (2) Использование компрессии данных - небольшой выигрыш в объеме никак не окупает проблемы, связанные с обратным преобразованием данных и организацией их хранения при получении;
- (3) Проблема индексов – нехватка памяти для размещения индексных таблиц целиком даже в кластерных архитектурах и, как следствие, необходимость использования индексов для индексных таблиц.

В данной работе предлагается подход к решению проблемы быстродействия на основе рациональных агентов (далее РА).

1. Архитектура VLDB с использованием рациональных агентов

Возможная обобщенная архитектура VLDB, в которой мы будем описывать использование РА, приведена на Рис. 1. Назначение компонентов и описание их взаимодействия приводится в Таблица 1. В такой архитектуре РА расположены на 3-х уровнях: коммуникационном, препроцессорном и кластерном. Такое решение позволяет обеспечить гибкость решения и распараллеливание процессов, выполняемых в границах системы. Далее рассмотрим более подробно работу системы на уровне РА.

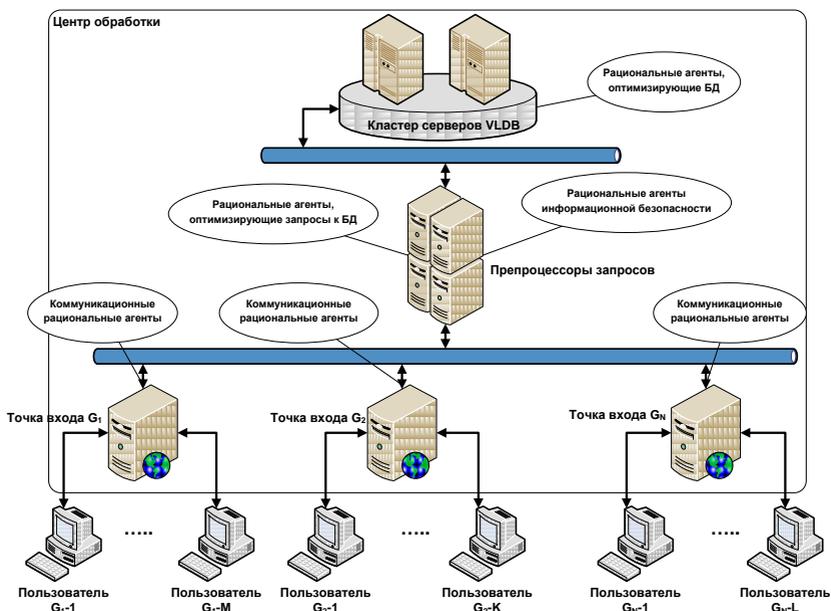


Рис. 1. ОБОБЩЕННАЯ АРХИТЕКТУРА VLDB, ИСПОЛЬЗУЮЩАЯ РАЦИОНАЛЬНЫЕ АГЕНТЫ

ТАБЛИЦА 1. КОМПОНЕНТЫ АРХИТЕКТУРЫ VLDB И ИХ ВЗАИМОДЕЙСТВИЕ

Наименование компонента	Описание компонента
Кластер серверов VLDB	Компонент содержит операционную систему, обеспечивающую работу кластера, систему управления БД (далее СУБД), непосредственно БД, РА, оптимизирующие БД
Препроцессоры запросов	Компонент обеспечивает прием запросов с компонентов «Точка входа G_i », содержит РА, оптимизирующие запросы к БД и получающие результаты запроса от БД, содержит РА, обеспечивающие

Наименование компонента	Описание компонента
	аутентификацию пользователя в процессе его взаимодействия с БД после прохождения идентификации на входе
Точка входа G_i	Компонент обеспечивает первичную идентификацию и аутентификацию пользователей, принимает запросы от пользователей и передает их препроцессорам запросов, принимает результаты запросов от препроцессоров запросов и передает их пользователям, обеспечивает работу коммуникационных агентов, которые выполняют функции оптимизации и маршрутизации запросов
Пользователь G_{i-j}	Рабочая станция субъекта взаимодействующего с VLDB
Рациональные агенты, оптимизирующие БД	Программное обеспечение, реализующее функциональность рациональных агентов, которые оптимизируют работу СУБД и структуру БД (см. ниже)
Рациональные агенты, оптимизирующие запросы к БД	Программное обеспечение, реализующее функциональность рациональных агентов, которые оптимизируют запросы к БД (см. ниже)
Рациональные агенты информационной безопасности	Программное обеспечение, реализующее функциональность рациональных агентов, которые обеспечивают информационную безопасность (см. ниже)
Коммуникационные рациональные агенты	Программное обеспечение, реализующее функциональность рациональных агентов, которые оптимизируют работу системы на уровне коммутаций

2. Многоагентная архитектура

В [2] приводится описание теоретическое описание работы ансамбля РА, которые решают некоторую общую задачу, но при этом взаимодействуют между собой. В [3] предлагается вариант использования многоагентной архитектуры (МнА) на базе стохастических автоматов.

На Рис. 2 показано использование МнА для системы, использующей VLDB.

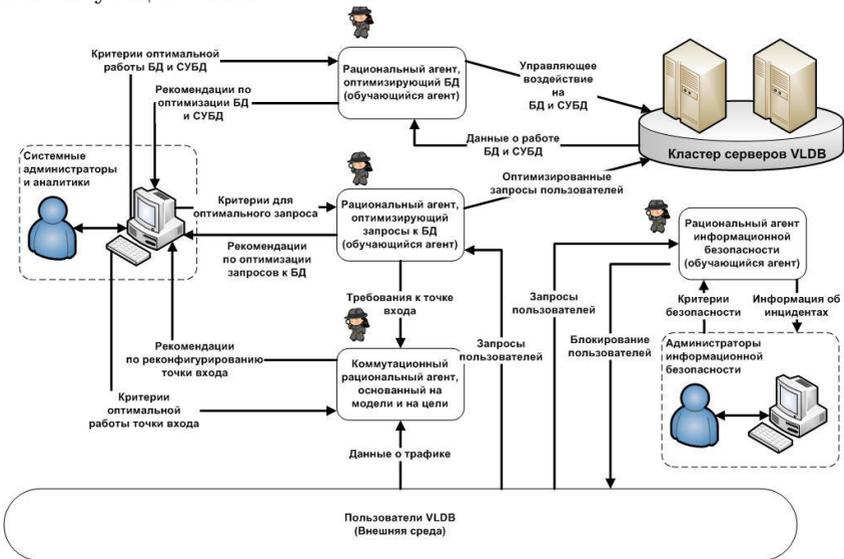


Рис. 2. Модель многоагентного формирования операционной среды для системы с VLDB

В данной модели введены дополнительные, относительно предыдущей модели, компоненты: «Системные администраторы и аналитики» и «Администраторы информационной безопасности». Это связано с тем, что, при запуске системы в эксплуатацию, а затем в процессе эксплуатации, необходимо создать некоторую стартовую конфигурацию РА, а затем контролировать работу РА и, при необходимости, вмешиваться в работу, например, если РА

не может найти решения, слишком долго его ищет, принимает неверное решение.

В дальнейшем изложении мы будем предполагать, что для каждого из РА все элементы, которые с ним взаимодействуют, являются в его точки зрения внешней средой. Кроме того, будем рассматривать два типа агентов (см. Таблица 2).

Таблица 2. Типы рациональных агентов

Тип рационального агента	Описание рационального агента
Рациональный агент, основанный на модели и на цели	РА данного типа следит за состоянием внешней среды, а также за множеством целей, которых он пытается достичь, и выбирает действие, позволяющее достичь этих целей.
Обучающийся рациональный агент	РА данного типа похож на РА, основанный на модели и на цели, но содержит компонент, позволяющий обучаться.

2.1. Коммутационный рациональный агент

Коммутационный РА (далее КРА) построен на базе концепции агента, основанного на модели и на цели. Модель взаимодействия КРА с окружающим миром представлена на Рис. 3.

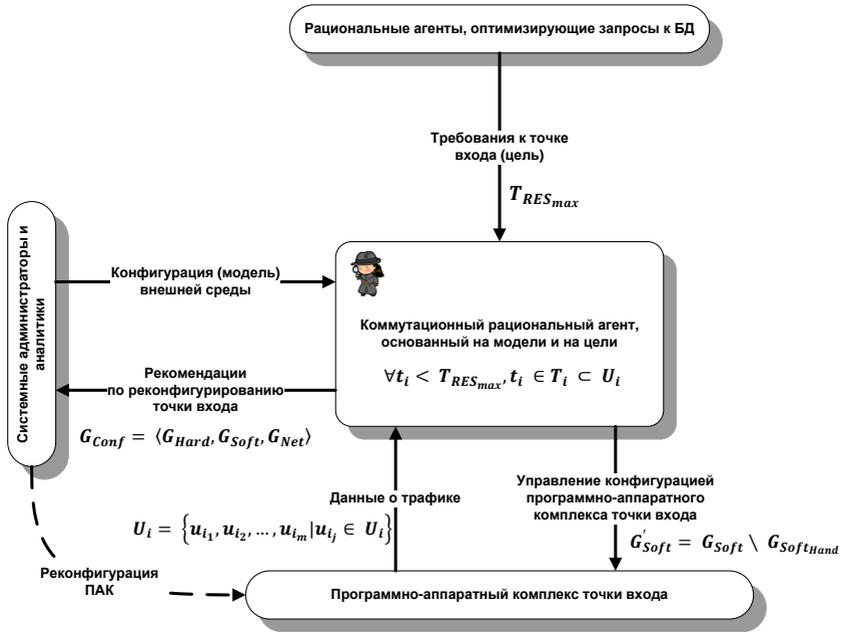


Рис. 3. Модель взаимодействия КРА с окружающим миром

КРА получает информацию о конфигурации внешней среды в потоке данных от объекта «Системные администраторы и аналитики» в виде набора параметров, характеризующих аппаратные средства (далее АС), программные средства (далее ПС), вычислительную сеть (далее ВС), которые представлены в виде кортежа (медленные изменения):

$$G_{Conf} = \langle G_{Hard}, G_{Soft}, G_{Net} \rangle \tag{1}$$

где G_{Hard} - множество параметров, характеризующих АС;
 G_{Soft} - множество параметров, характеризующих ПС;
 G_{Net} - множество параметров, характеризующих ВС.

Цель, которую должен достигать КРА: управление программно-аппаратным комплексом (далее ПАК) точки входа такое, чтобы прохождение каждого из пользовательских запросов не превышало времени $T_{RES_{max}}$, которое определяет РА, оптимизирующий запросы к БД. Цель КРА может быть представлена в виде следующего выражения:

$$\forall t_i < T_{RES_{max}}, t_i \in T_i \subset U_i \quad (2)$$

т.е. все временные интервалы t_i для запросов из множества всех запросов T_i , отправляемых и принимаемых пользователями, не должны превышать $T_{RES_{max}}$. При этом временная задержка определяется каждым из компонентов из множества U_i , описывающих трафик.

Следует учесть, что часть управляющих воздействий КРА может осуществлять самостоятельно, например, изменение конфигурационных файлов, установка обновлений и т.д.; а часть воздействий может осуществляться через компонент «Системные администраторы и аналитики» путем отправки им рекомендаций по конфигурированию ПАК точки входа.

На Рис. 4 представлена модель потоков между компонентами модели.

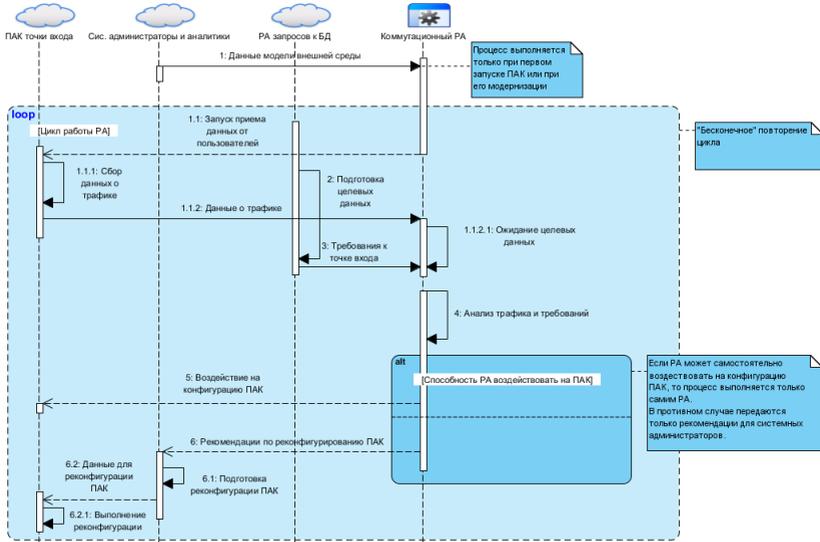


Рис. 4. Модель потоков в нотации UML для КРА

2.2. Рациональный агент информационной безопасности

РА информационной безопасности (далее ИБ) построен на базе концепции агента, основанного на модели обучающегося агента. Модель взаимодействия РА ИБ с окружающим миром представлена на Рис. 5.

Основной задачей РА ИБ является слежение за поведением каждого из пользователей, работающих через данную точку входа, на уровне поведения при формировании запросов к VLDB, например, время когда пользователь обычно взаимодействует с БД, сколько времени занимает один сеанс взаимодействия, длина поисковых запросов, содержание поисковых запросов, особенности записи команд поиска/добавления/редактирования записей, предметная область (далее ПрО), которая интересует пользователя и др. При отклонении в текущем сеансе от одного из параметров, это является основанием для дополнительной идентификации пользователя или, при подозрении на атаку, блокирование учетной записи с отправкой соответствующего уведомления

администраторам ИБ (далее АИБ), которые могут внести коррективы в работу РА ИБ.

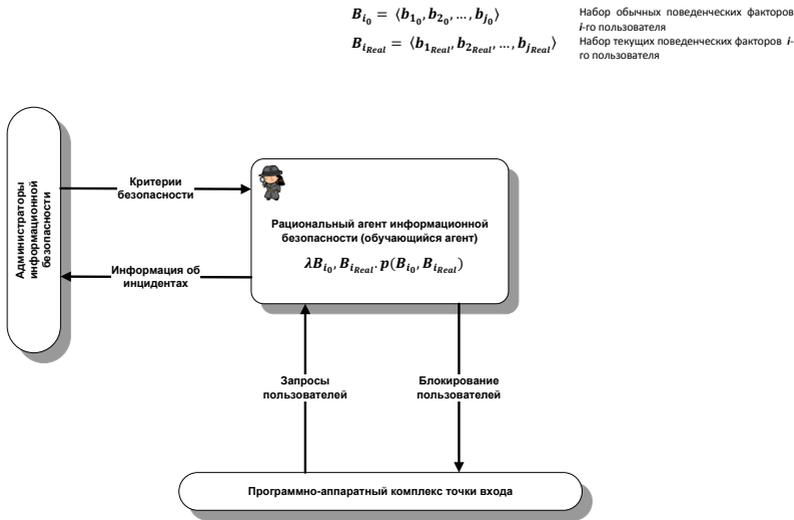


Рис. 5. Модель взаимодействия РА ИБ с окружающим миром

Формально функцией РА ИБ является вычисление вероятности, с которой может наступить то или иное событие, связанное с ИБ, как показано в следующем выражении:

$$\lambda B_{i_0}, B_{i_{Real}} \cdot p(B_{i_0}, B_{i_{Real}}) \quad (3)$$

где p - вероятность реализации угрозы «подмена пользователя»;

$\mathbf{B}_{i_0} = \langle b_{1_0}, b_{2_0}, \dots, b_{j_0} \rangle$ - набор обычных поведенческих факторов i -го пользователя;

$\mathbf{B}_{i_{Real}} = \langle b_{1_{Real}}, b_{2_{Real}}, \dots, b_{j_{Real}} \rangle$ - набор текущих поведенческих факторов i -го пользователя.

Интервал p , при котором РА ИБ воспринимает текущее значение, как показатель угрозы, определяется АД.

На Рис. 6 представлена модель потоков между компонентами модели.

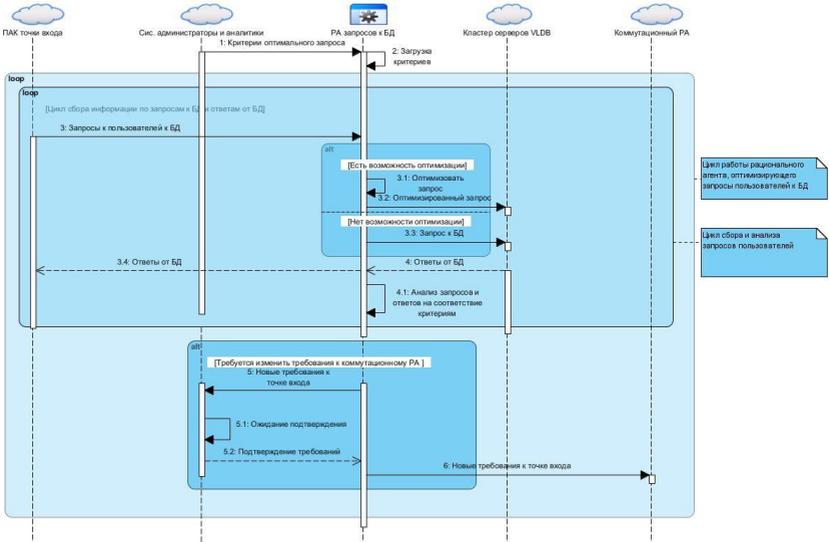


Рис. 6. Модель потоков в нотации UML для РА ИБ

2.3. Рациональный агент, оптимизирующий запросы к БД

РА, оптимизирующий запросы к БД (далее РА ЗБД) построен на базе концепции агента, основанного на модели обучающегося агента. Модель взаимодействия РА ЗБД с окружающим миром представлена на Рис. 7.

Основной задачей РА ЗБД является оптимизация запросов, поступающих от пользователей к VLDB: как непосредственно написанных пользователем, так и сформированных ПС, которые использует пользователь. Предполагается, что в соответствие каждому пользователю будет сопоставлен «персональный» РА ЗБД для ускорения обработки запросов.

Если в БД малого и среднего размера критичность к эффективности запроса не имеет столь большого значения, то для

VLDB это становится проблемой, т.к. неэффективный запрос может привести, как отмечено в [4], к:

- Увеличению стоимости коммуникаций – неэффективный запрос может привести к огромному массиву данных, отправляемых в ответ;
- Увеличению стоимости доступа к вторичной памяти – на эту стоимость влияет длина выбираемых данных (главным образом, размер промежуточных результатов), кластеризация данных на физических страницах, размер доступного буферного пространства и скорость используемых устройств;
- Увеличение стоимости хранения – т.е. возрастает потребность в устройствах памяти для хранения основных и промежуточных результатов;
- Увеличение стоимости вычислений – т.е. стоимость (время) загрузки центрального процессора (далее ЦП).

Для вычислений в VLDB эти проблемы возрастают на порядки: любой неэффективный запрос, написанный пользователем с низкой квалификацией или намеренно злоумышленником – может привести к атаке типа «отказ в обслуживании» (DoS-атака).

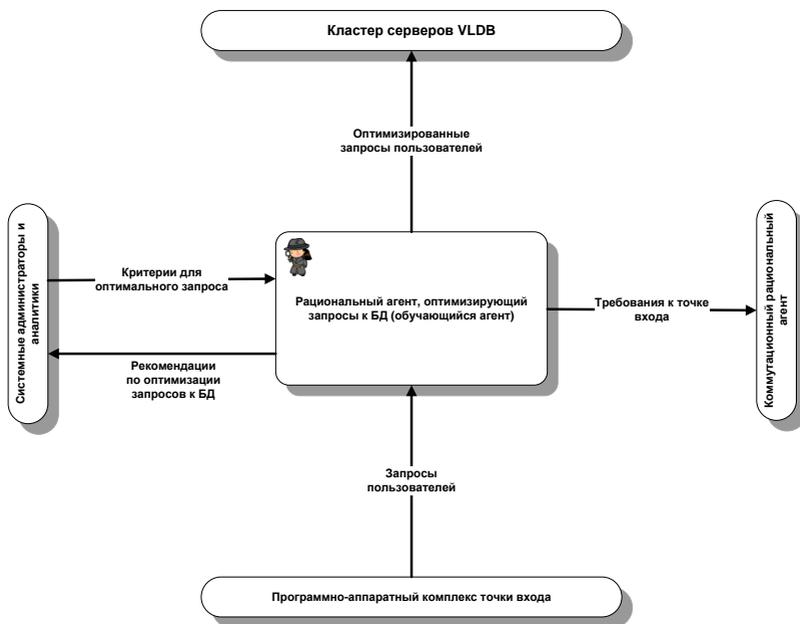


Рис. 7. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ РА ЗБД С ОКРУЖАЮЩИМ МИРОМ

Т.е. процессы, выполняемые внутри РА, должны при поступлении пользовательского запроса:

- (1) Выполнять «быструю» проверку – оценить надежность пользователя, как автора запросов, количество удачных использований данного запроса;
- (2) Если шаг 1 не выполним – выполнять анализ запроса, используя общую для всех РА ЗБД базу знаний удачных запросов;
- (3) Если шаги 1 и 2 не дали результатов – провести внутреннее моделирование использования запроса и, если результат будет положительным, запросить подтверждение вывода РА ЗБД у системных администраторов и аналитиков, а после получения подтверждения и успешного применения в реальной VLDB – внести данный запрос в «персональный» РА ЗБД и общую базу знаний для всех РА ЗБД;

- (4) Если шаг 3 не привел к успеху – отправить сообщение об ошибке пользователю и понизить его рейтинг, как автора запросов.

Кроме того, РА ЗБД должен оценивать временные максимальный интервалы времени, которые затрачиваются запросами и результатами запросов в целях оценки ширины коммуникационного канала для каждой из точек доступа. По результатам направлять требования к точкам доступа.

На Рис. 8 представлена упрощенная модель некоторых потоков между компонентами модели.

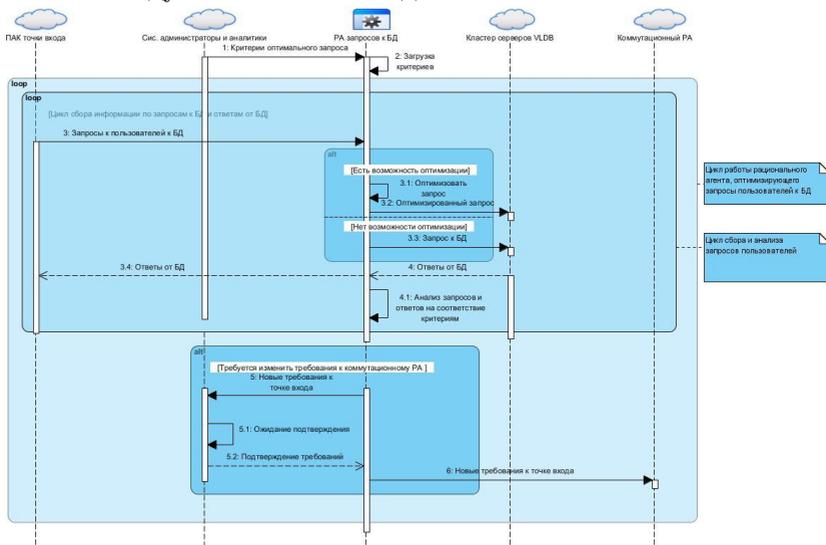


Рис. 8. Модель потоков в нотации UML для РА ЗБД

2.4. Рациональный агент, оптимизирующий БД

РА, оптимизирующий БД (далее РА ОБД) построен на базе концепции агента, основанного на модели обучающегося агента. Модель взаимодействия РА ОБД с окружающим миром представлена на Рис. 9.

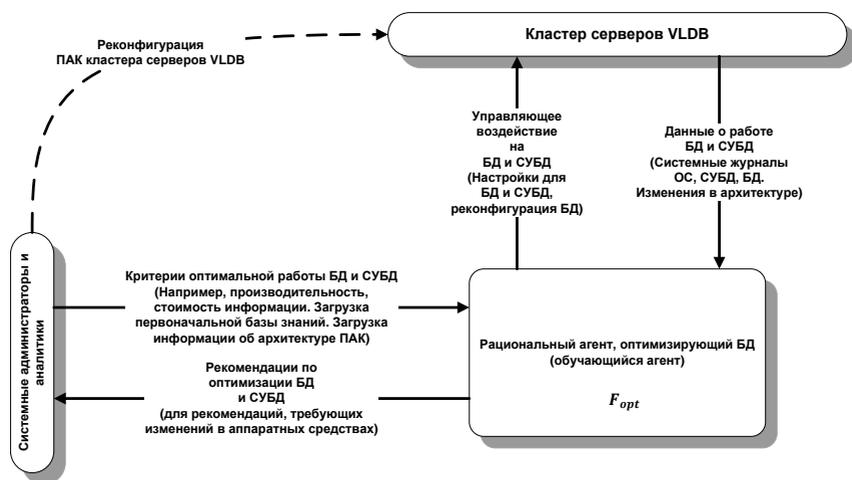


Рис. 9. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ РА ОБД С ОКРУЖАЮЩИМ МИРОМ

Основной задачей РА ОБД является оптимизация работы СУБД в целом и отдельных БД, управляемых ею. Анализ работы VLDB строится на основе анализа системных журналов отдельных компонентов, образующих СУБД для VLDB: АС серверного кластера, ОС, самой СУБД, отдельных БД, расположенных в кластере и др. По многим вопросам улучшения работы VLDB в настоящий момент РА ОБД может давать исключительно рекомендации, которые нужно выполнять вручную системными администраторами и аналитиками.

Т.е. основными процессами, которые должен выполнять РА ОБД являются:

- (1) Анализ системных журналов и превентивный поиск проблем;
- (2) Моделирование внесения изменений в архитектуру;
- (3) Уведомление системных администраторов и аналитиков о возможных проблемах и, если найдено решение, о возможном решении;

- (4) При подтверждении администраторами успешности предложенного решения, внесение решения в базу знаний РА ОБД.

На Рис. 10 представлена упрощенная модель отдельных потоков между компонентами модели.

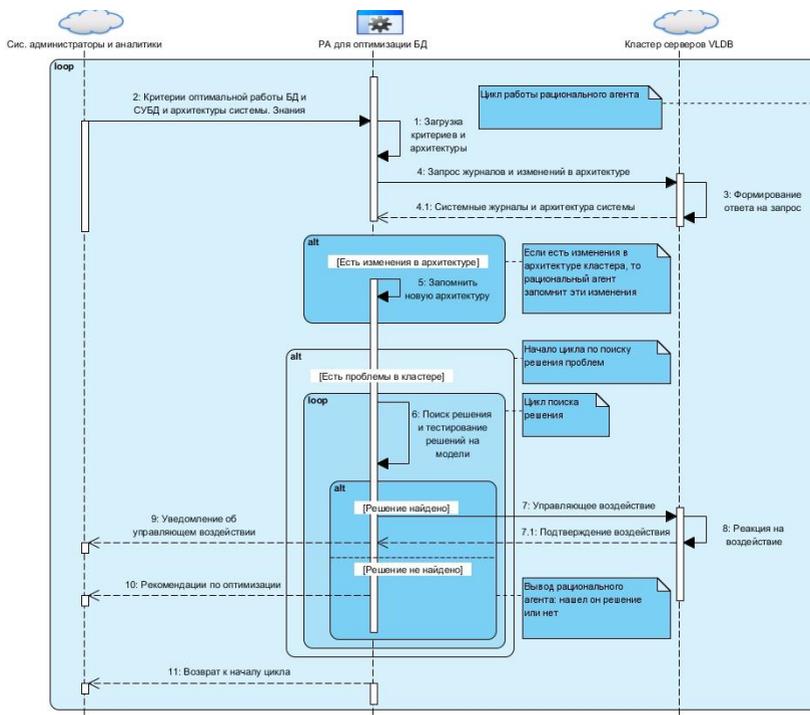


Рис. 10. Модель потоков в нотации UML для РА ОБД

Заключение

В настоящей работе показано, что значительную часть проблем, связанных с работой VLDB, можно решить путем использования ассоциаций рациональных агентов, обучаемых в процессе работы в операционной среде VLDB.

Насколько известно автору, подобный подход до сих пор не применялся именно для VLDB, но первые теоретические результаты показывают эффективность такого подхода.

Благодарности. Автор выражает свою благодарность своему научному руководителю проф. Н.Н. Непейводе за терпеливое обсуждение идей и ряд ценных замечаний, внесенных при редактировании материала.

Список литературы

- [1] Арендт Эрик. «Экстремальные» базы данных: Самые большие и самые быстрые. Чему учит нас опыт работы с «экстремальными» базами данных // developerWorks Россия. - :IBM, 2010г. http://www.ibm.com/developerworks/ru/library/DBM_ag_Issue109_Extreme/
- [2] Стюарт Рассел, Питер Норвиг. Искусственный интеллект. Современный подход. // Москва : Издательский дом "Вильямс", 2006. - 2 : Т. 1 : стр. 1408.
- [3] Nikolai N. Nepejvoda, Alexey Tsvetkov and Margarita Frolova. Knowledge Structurization by Multi-agent Approach in a Non-formalizable Subject Area of Health and Medicine. // Proceedings of the International Conference of Control, Dynamic Systems, and Robotics. Ottawa, Ontario, Canada, May 15-16 2014. Paper No. 102. ISBN: 978-1-927877-02-9
- [4] Matthias Jarke, Jurgen Koch. Query Optimization in Database Systems. // Computing Surveys, Vol. 16, No. 2, June 1984

Об авторе:



Алексей Анатольевич Цветков

Научный сотрудник, Институт программных систем
РАН

e-mail: alexey_tsvetkov@hotmail.com

A.A. Tsvetkov. Multiagent approach to the controlling of the operational environment containing extremely large databases.

ABSTRACT. This article focuses on the analysis of problems in the system maintenance of the operational environment using extremely large databases. Proposes a possible architecture environment based on the cooperation of intelligent / rational agents, that can monitor the operation of the system and, if necessary, to a large extent control it automatically.

Key Words and Phrases: extremely large databases, intelligent agent, rational agent, information security, VLDB