УДК 004.652

ПОСТРОЕНИЕ И АДАПТАЦИЯ NEWSQL СУБД В ЧАСТНОМ «ОБЛАКЕ»

А. С. Соболь

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: a1exander.so8ol@gmail.com

Анализируются вопросы построения NewSQL СУБД в частном «облаке» с учетом ограничений БД. Формализуются требования к будущей NewSQL СУБД, в которой планируется обслуживать систему электронного документооборота (СЭД) и систему бизнес-аналитики предприятия. Разрабатываются методики по использованию NewSQL систем в частном «облаке» для достижения наилучшей производительности при сохранении требований ACID (атомарность, согласованность, изолированность, надёжность). Приводятся результаты сравнения NewSQL СУБД в задачах обеспечения доступа и скорости обработки запросов на основании разработанных методик. Для выбранной СУБД определяется наилучший вариант её построения. Показано, что СУБД, отвечающей требованиям, является VoltDB. По сравнению с традиционными решениями, VoltDB превосходит их в производительности более чем в 2 раза. После применения разработанных методик производительность СУБД возросла на 14 %. Важно отметить, что представленное решение полностью удовлетворяет критериям, предъявляемым к частному «облаку».

Ключевые слова: СУБД, NewSQL, «облачные» вычисления, большие данные, СЭД.

DESIGN & ADAPTATION NEWSQL DBMS IN THE PRIVATE “CLOUD”

A. S. Sobol

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation.
E-mail: a1exander.so8ol@gmail.com

The paper dwells on the issues of building NewSQL database in a private “cloud”, subject to the limitations database. It reviews formalized requirements for a future NewSQL database, which will serve EDMS (electronic document management system) and a system of business intelligence of a company. There are developed methods for using NewSQL systems in a private “cloud” to achieve the best performance at preservation of the requirements of ACID (atomicity, consistency, isolation, durability). The results of comparing NewSQL database in problems of access and query processing based on the developed techniques are presented. The best version of the selected DBMS building is defined. It is shown that the database meets the requirements of a VoltDB. The VoltDB exceeds the performance more than 2 times compared with traditional solutions. The performance solution has increased by 14 % after application of the developed techniques. It is important to note that the presented solution fully meets the criteria for a private “cloud”.

Keywords: DBMS, NewSQL, Cloud computing, Big Data, EDMS.

Учитывая сохраняющуюся тенденцию роста объемов данных в OLTP-системах (Online Transaction Processing), требуется новое поколение решений, чтобы удовлетворить потребности потребителей. Существующие решения при этом, такие как реляционные БД, не имеют достаточного уровня производительности либо же обладают плохим горизонтальным масштабированием, а NoSQL-системы не удовлетворяют требованиям ACID, жизненно важным для СЭД и других систем, традиционно базирующихся на реляционных БД. При этом в отличие от NoSQL-систем, эта СУБД должна поддерживать приложения, уже написанные для предыдущих поколений СУБД. Таким образом, разговора о радикальных изменениях интерфейса, отказе от SQL или значительных изменениях в схеме данных и быть не может [1]. Дополнительным условием являются архитектурные особенности СЭД:

– B-Tree-индексирование;
– наличие в БД СЭД мигрирующих строк: строка за время своего существования в базе данных увеличивается и перемещается из исходной страницы в другую;
– «дерево» индексов, имеющее более 7 уровней;
– большая кардинальность таблиц: среднее значение ~26;
– фактор селективности таблиц, равный 0,038;
– присутствие таблиц маленького размера;
– интенсивные обновления части таблиц в пакетном режиме;
– наличие в колонках множества неопределённых значений, дающих значительную асимметрию распределения значений колонки;
– присутствие в БД BLOB-объектов.

Новым поколением, призванным справиться с данной проблемой и учитывающим эти ограничения,

являются NewSQL-системы, предназначенные, в первую очередь, для предприятий, которые планируют:

– миграцию существующих приложений для адаптации к новым тенденциям роста объема данных;

– разработку новых приложений с высокой масштабируемостью систем OLTP;

– опору на существующие знания использования OLTP.

Однако учитывая относительную новизну данных решений и тот факт, что большинство из них всё ещё постоянно нуждаются в постоянной доработке и оптимизации, а также неопределённость выбора варианта развёртывания системы на предприятии, будь то DBaaS либо же экземпляр виртуальной машины, возникает необходимость проведения исследования в данном направлении, чтобы описать, когда требуется то или иное решение, а также формализовать методики по использованию данных систем при работе с конкретными БД.

Термин NewSQL – это сокращение для различных новых масштабируемых и высокопроизводительных SQL баз данных. NewSQL-поставщики имеют общую разработку новых продуктов для реляционных баз данных и услуг, призванных изменить реляционную модель для распределенных архитектур или для повышения производительности реляционных баз данных при условии, что горизонтальная масштабируемость больше не является необходимостью [2].

В статье ставится следующая задача: провести анализ имеющихся на рынке NewSQL СУБД, выбрать лучшую из доступных и адаптировать её с учётом архитектурных особенностей БД в частном «облаке» (модель обеспечения повсеместного и удобного сетевого доступа по требованию к общему пулу конфигурируемых вычислительных ресурсов, предоставляемому внутренней ИТ-службой предприятия). В решении будут использоваться БД СЭД DocsVision и БД системы бизнес-аналитики. Тестовым стендом будет являться кластер БД, построенный на платформе из четырёх серверов SuperMicro 2011 с процессорами Intel Xeon 2650 и 2620 и 48 GB DDR3 на каждом из них.

СУБД по умолчанию является MS SQL Server 2008R2 в редакции Standard и MySQL Cluster Carrier Grade 7.2. Задача будет разделена на четыре этапа:

– на первом этапе будут сформулированы технические требования к будущей системе, позволяющие выделить из всех NewSQL-систем наиболее подходящие;

– на втором этапе будет определён оптимальный способ размещения базы данных;

– на третьем – с использованием подготовленных запросов и тестов в отношении скорости их исполнения будут определены наиболее подходящие из выбранных на первом этапе СУБД;

– на четвертом этапе данные системы будут проанализированы и по возможности оптимизированы с учетом спецификации БД СЭД и БД системы бизнес-аналитики.

1. Ниже представлен набор минимальных требований к техническим характеристикам СУБД:

– SQL как основной механизм для взаимодействия;

– ACID-поддержка транзакций;

– механизм управления без применения блокировок;

– архитектура, обеспечивающая лучшую производительность узлов, чем любые из традиционных решений RDBMS;

– удобное масштабирование, способное управлять большим количеством узлов, не перенося «узкие» места;

– отказ от использования БД, опубликованных третьими лицами как «облачный» сервис (SQL Azure, Amazon RDS и др.);

– БД не должна использовать расширения традиционной SQL-архитектуры, наподобие Handler socket решений.

Ниже представлены классифицируемые на подгруппы NewSQL-системы (рис. 1), которые будут подвергнуты дальнейшему анализу [2].

Классификация NewSQL-систем основана на различных подходах, принятых сохранить SQL-интерфейс, а также решить масштабируемость и производительность, являющуюся проблемами традиционных решений OLTP:

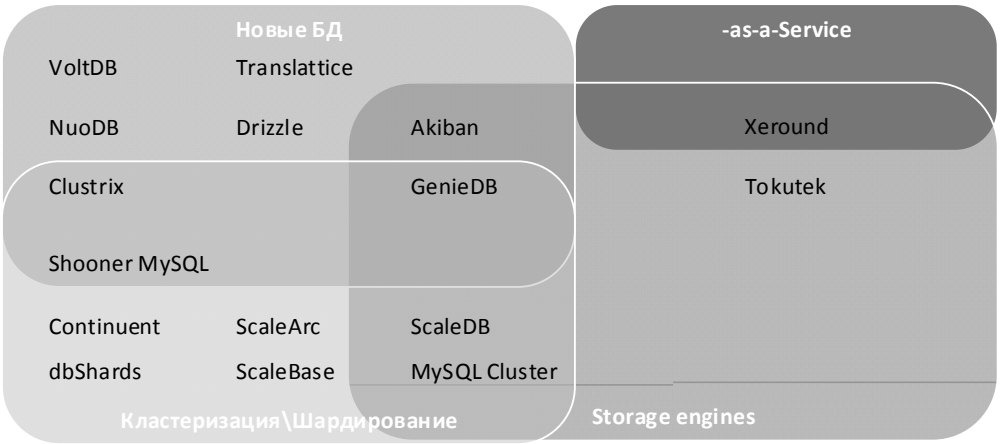

Рис. 1. NewSQL-системы

1) новые базы данных: NewSQL-система разрабатывается полностью с нуля с целью достижения масштабируемости и производительности;

2) новый механизм хранения MySQL (Storage engines): чтобы преодолеть проблемы масштабируемости MySQL, был создан ряд новых механизмов хранения, основанных на MySQL; положительной стороной данного решения является использование интерфейса MySQL; отрицательной – отсутствие миграции данных из других баз данных, включая старый MySQL;

3) прозрачное объединение в кластеры: эти решения сохраняют базы данных OLTP в своем оригинальном виде, но обеспечивают особенность расширения с прозрачной группировкой, гарантирующую масштабируемость [3].

Из описанных БД исключается группа Storage engines вследствие программной невозможности миграции данных из других БД. Группа «кластеризация\шардирование» не в полной мере отвечает необходимым требованиям, и поэтому на третьем этапе из этой группы будет рассмотрена лишь одна СУБД для обеспечения наглядности тестов производительности.

2. Немаловажным аспектом является вариант размещения БД, подразумевающий следующее:

– тип сервера, на котором будет развёрнута БД: виртуальный или физический;

– дисковая подсистема;

– архитектура решения: количество нод, способ их размещения, маршрутизация;

– наличие консоли управления ресурсами (DBaaS).

Для улучшения быстродействия кластер БД будет развёрнут на физических серверах. Так как основное предназначение данного решения – обработка данных внутри компании (частное «облако»), консоль управления ресурсами самой СУБД не имеет критической важности. Особенностью конфигурации данных серверов является использование SSD- (Solid State Drive) накопителей помимо использования RAID 10 массива с обычными SAS-винчестерами. В то время когда под TempDB и журналы транзакций выделяются SSD, расположенные непосредственно на серверах, сама БД находится в хранилище. Такое решение позволяет смоделировать распределённую среду компании, с головным офисом и филиалами, удалёнными территориально. Помимо этого, данная архитектура позволяет максимально эффективно использовать пул вычислительных ресурсов, удовлетворяя характеристикам, присущим частному «облаку». Общая схема размещения представлена на рис. 2.

3. Предъявленным требованиям, описанным на первом этапе, в полной мере удовлетворяют 2 СУБД: VoltDB и NuoDB. Clustrix и Continuent, основанные на MySQL, приведены для сравнения производительности. Данные СУБД опираются на идеологию «Shared Nothing»: в кластере, созданном в такой идеологии, узлы не разделяют ресурсы между собой. Каждый узел обрабатывает свой фрагмент базы. За счет этого существенно возрастает производительность и масштабируемость системы с такой организацией на нагрузках любого типа [1].

Сложность данной прикладной задачи определяется двумя параметрами:

– большим объемом данных;

– большим количеством одновременных транзакций.

Если в СЭД SSD-накопители, более близкие по скорости к оперативной памяти, чем к классическим винчестерам, могут эффективно решать задачу с обработкой BLOB (binary large object), то при работе с OLAP-кубами при интенсивной транзакционной нагрузке определяющим фактором производительности уже становится сама архитектура СУБД.

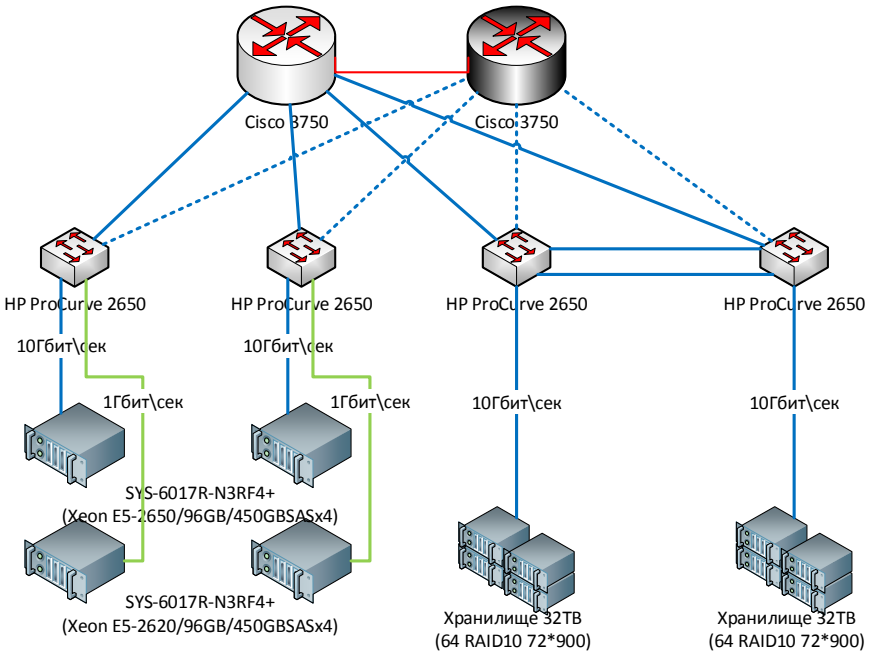

Рис. 2. Общая схема размещения кластера БД

Методика тестирования представлена двумя видами тестов: «формальными» и функциональными. Под формальными тестами понимается выполнение неких стандартных операций при заданных условиях на конкретном наборе данных. Такими тестами являются создание множества таблиц, вставка значений в таблицы, объединение таблиц, выборка значений. Функциональные тесты являются ориентированными на исследование характеристик СУБД при решении определенной прикладной задачи, исходные данные берутся непосредственно из СЭД и системы бизнес-аналитики.

Для измерения результатов производительности были созданы 50 таблиц с приведённой ниже схемой:

```
CREATE TABLE new_table;

ALTER TABLE new_table ADD COLUMN
    id character varying(20) NOT NULL,
    seq integer NOT NULL,
    col3 character varying(16) NOT NULL,
    col4 character varying(5) NOT NULL,
    col5 character varying(50) NOT NULL,
    col6 character varying(1000),
    col7 character varying(300) NOT NULL,
    col8 character varying(150),
    col9 timestamp NOT NULL,
    col10 smallint DEFAULT 0 NOT NULL,
    col11 timestamp NOT NULL,
    col12 character varying(15) NOT NULL,
    col13 character(1) NOT NULL,
    col14 character(1) NOT NULL,
    col15 timestamp DEFAULT timestamp NOT
NULL;

ALTER TABLE "new_table" ADD PRIMARY
KEY("id","seq");
CREATE UNIQUE INDEX "iuk_table" ON
"new_table"("id","col3","col4","col5");
CREATE INDEX "ink1_table" ON
"new_table"("id","col9" DESC,"col14").
```

Далее система проходила 3 этапа формальных тестов:

1) после создания базы данных для вставки 31 миллиона записей данных в 50 таблиц производительность измеряется при нагрузке INSERT FULL (ПОЛНАЯ ВСТАВКА) в течение 30 мин;

2) после создания базы данных для вставки 80 миллионов записей данных в 50 таблиц производительность измеряется при нагрузке SELECT (ВЫБОР), ограниченной возможностями центрального процессора (CPU Bound);

3) после создания базы данных для вставки 80 миллионов записей данных в 50 таблиц производительность измеряется при нагрузке SELECT, ограниченной скоростью ввода-вывода (I/O Bound).

Все вышеперечисленные нагрузки создавались в 56 потоках. Один INSERT состоит из одного запроса INSERT, тогда как SELECT состоит из трех запросов SELECT с первичным ключом, уникальным индексом и неуникальным индексом в каждом [4].

Функциональным тестом являлся запрос к СЭД, очищающий БД от сообщений неактивных бизнес-процессов. Названия таблиц были изменены:

```
DELETE [table_1]
FROM [table_1]mes
JOIN [table_2]main ON
mes.InstanceID=main.InstanceID
      WHERE main.state !=1
```

Ниже в таблице представлены результаты тестирования производительности.

Рассмотрим результаты детально: производительность NewSQL-систем выше производительности стандартных OLTP решений в среднем в 1,36 раза. Системы, основанные на методах кластеризации и шардирования, показывают средний результат.

После создания базы данных с 50 таблицами производилось тестирование производительности (рис. 3) с нагрузкой, ограниченной возможностями ЦП (SELECT 5 PRO FULL) и скоростью ввода-вывода (SELECT 100 PRO FULL), в течение 10 минут. В нагрузке с SELECT 5 запросами, область поиска запроса была сужена для того, чтобы полностью разместить необходимую страницу в буфере памяти и поддерживать желаемое 100%-ное значение результативности буфера.

Результаты тестирования производительности СУБД

		SQL		NewSQL			
				Кластеризация\Шардирование		Новые БД	
		MS SQL Server	MySQL CCG	Continuent	Clustrix	NuoDB	VoltDB
Транзакций в секунду	INSERT FULL	5890	5196	7105	6887	7193	7980
	SELECT 5 PRO FULL	4201	4206	4533	4421	2646	4059
	SELECT 100 PRO FULL	1389	1416	1680	1512	2150	2318
Время (минуты)	Очистка сообщений в СЭД	242	-	-	-	142	103
	Очистка сообщений в СЭД после оптимизации	-	-	-	-	129	91
INSERT FULL	IO Write	7534	6899	10952	10860	12096	13779
	IO Read	7110	6297	10341	10250	10841	12154
% использования CPU	INSERT FULL	80%	60%	95%	100%	100%	100%
	SELECT 5 PRO FULL	0%	0%	0%	0%	0%	0%
	SELECT 100 PRO FULL	100%	100%	100%	100%	100%	100%
	QUERY	100%	-	-	-	100%	100%

В SELECT 100, чтобы не размещать необходимую страницу в буфере памяти полностью и предотвратить частую замену страниц, область поиска запроса SELECT, наоборот, была расширена. Количество операций ввода-вывода увеличилось, так как рабочая нагрузка весьма интенсивна.

В первом случае при отсутствии операций ввода-вывода производительность NewSQL решений падает по сравнению с SQL-системами. При этом если производительность VoltDB остается в среднем на том же уровне, то потеря производительности NuoDB очевидна. Во втором случае NewSQL-системы уверенно опережают по производительности MS SQL Server и MySQL CCG.

Следующий тест – подготовленный запрос (рис. 4) в БД СЭД.

Здесь отчетливо видно преимущество VoltDB. NuoDB, как представитель NewSQL, так же показывает результаты, превосходящие SQL решение по скорости обработки почти в 2 раза.

В целом настоящий эксперимент подтверждает тот факт, что при увеличении нагрузки NewSQL-решения показывают лучшие результаты по сравнению с классическими OLTP SQL-системами. При этом из рассмотренных систем для использования в компании выбор падает на VoltDB – именно эта СУБД показала наилучшие результаты в ходе тестирования.

4. Сформулированы правила, разработанные в ходе конфигурирования и тестирования, представляющие собой методики оптимизации БД СЭД и БД системы бизнес-аналитики и оптимизации обращений к ним для NewSQL-систем в целом и для VoltDB в частности:

– по возможности разделять транзакции для таблиц, чтобы максимизировать частоту одноразделных транзакций и минимизировать частоту многораздельных транзакций;

– оценивать объемы данных и частоту каждой транзакции для приложения, чтобы определить, какие столбцы использовать для разметки;

– использовать несколько SQL-запросов в хранимых процедурах. Десять SQL-запросов в одной одноразделной процедуре может быть в 10 раз быстрее, чем десять процедур с одним запросом в каждой;

– устанавливать флаг IsFinal истинным в последнем вызове VoltExecuteSQL () в каждой хранимой процедуре СУБД VoltDB. Это повышает производительность, особенно для многораздельных транзакций;

– так как различные приложения и аппаратные средства имеют разные характеристики, лучше использовать бенчмаркинг разметки для определения оптимального количества разделов для приложения;

– по возможности использовать асинхронные вызовы процедур и создания клиентских подключений ко всем узлам в кластере, чтобы максимизировать пропускную способность;

– нежелательно создавать таблицы с очень большим количеством строк (т. е., много столбцов или большие столбцы типа VARCHAR). Лучше создать несколько меньших таблиц с общим ключом разделения;

– нежелательно создавать запросы, которые возвращают большие объемы данных (например, SELECT * FROM TABLE без каких-либо ограничений), особенно для многораздельных транзакций;

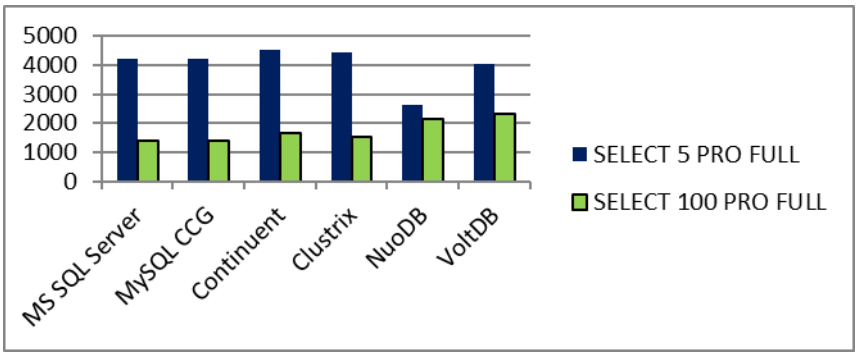

Рис. 3. Количество транзакций в секунду при выполнении операции SELECT

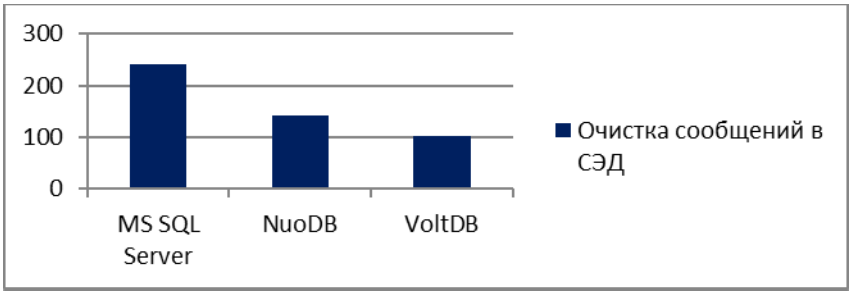

Рис. 4. Время исполнения запроса (в минутах) на очистку сообщений из БД СЭД

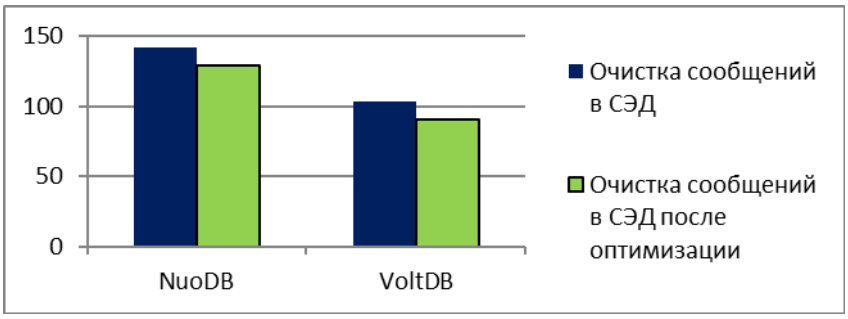

Рис. 5. Время исполнения запроса (в минутах) на очистку сообщений из БД СЭД до и после оптимизации

– даже небольшие таблицы должны быть разделены и не должны дублироваться, если они часто обновляются;

– не рекомендуется делать большую обработку в асинхронных обратных вызовах. За один раз обрабатывается только один обратный вызов, так что лучше делать эффективные процедуры обратного вызова, чтобы избежать задержек;

– не стоит тестировать приложения на standalone-серверах;

– не желательно вызывать ClientFactory. createClient () более одного раза в каждом клиентском приложении, иными словами, должен быть только один экземпляр клиента в пределах каждого клиентского приложения.

Данные правила позволили поднять производительность СУБД в среднем ещё на 9–17 % (рис. 5) в зависимости от конкретной задачи. Данные правила не являются универсальными, но их можно рассматривать не только применительно к БД компании, но и в целом к БД, хранящим в себе BLOB и имеющим более 10000 транзакций в секунду.В целом, использование правильно сконфигурированного NewSQL-решения позволило достичь необходимого уровня производительности без тотального изменения структуры БД, а также добиться прозрачного масштабирования на будущее. Самое главное в таком решении в отличие от standalone-конфигурации сервера баз данных, что оно полностью соответствует критериям, предъявляемым к частному «облаку»: объединение ресурсов, унифицированность и эластичность.

Библиографические ссылки

1. Кластерные СУБД [Электронный ресурс] // Информационный бюллетень компании «Инфосистемы Джет» : [сайт]. URL: http://www.jetinfo.ru/stati/klasternye#gl_2 (дата обращения: 07.07.2013).

2. MySQL vs. NoSQL and NewSQL – survey results [Электронный ресурс] // The 451 group : [сайт]. URL: http://blogs.the451group.com/opensource/2012/05/25/mysql-vs-nosql-and-newsql-survey-results/ (дата обращения: 07.05.2013).

3. NewSQL – The New Way to Handle Big Data [Электронный ресурс] // Linux for you : [сайт]. URL: http://www.linuxforu.com/2012/01/newsql-handle-big-data/ (дата обращения: 10.06.2013).

4. CUBRID vs. MySQL performance test results before and after the SSD usage [Электронный ресурс] // CUBRID : [сайт]. URL: http://www.cubrid.org/ssd_performance_test (дата обращения: 08.07.2013).

References

1. *Klasternye SUBD. Informatsionnyy byulleten' kompanii "Infosistemy Dzhet"*. Available at: http://www.jetinfo.ru/stati/klasternye#gl_2 (date of view: 07.07.2013).

2. MySQL vs. NoSQL and NewSQL – survey results. The 451 group. Available at: http://blogs.the451group.com/opensource/2012/05/25/mysql-vs-nosql-and-newsql-survey-results/ (accessed: 07.05.2013).

3. NewSQL – The New Way to Handle Big Data. Linux for you. Available at: http://www.linuxforu.com/2012/01/newsql-handle-big-data/ (accessed: 10.06.2013).

4. CUBRID vs. MySQL performance test results before and after the SSD usage. CUBRID. Available at: http://www.cubrid.org/ssd_performance_test (accessed: 08.07.2013).