

## **Висновки**

На базі досліджень можна зробити такі висновки:

- визначено технологічну послідовність процесу виконання очисних робіт 4-ї південної лави ВП «Шахта «Капітальна», в якій видобування вугілля в лаві здійснюється механізованим комплексом 1МКД-90;
- розроблено образ очисного вибію та сформовано дескриптивну модель процесу;
- розроблено модель процесу "видобування вугілля" в міжнародному стандарті IDEF0 за допомогою CASE-засобу AllFusion Process Modeler з метою керування сучасними процесно-орієнтованими підприємствами вугільної галузі.

Дана методика дозволяє стандартизувати оцінки діяльності вугільної шахти з погляду процесного підходу. Все це дозволить підвищити рівень керування гірничим підприємством, особливо в умовах конкурентоспроможної економіки й використання інформаційних систем керування підприємством.

Наступним етапом дослідження буде імітаційне моделювання, щоб мати можливість регулювати можливий розвиток подій і впливати на результати виконання процесу, приймати різні рішення відносно організації процесу.

## **Список літератури**

1. Слесарев В.В. Процесний підхід до управління вугледобувними підприємствами. /В.В. Слесарев ,С.В. Козир.; Сборник научных трудов международной конференции «Современные информационные технологии подготовки инженерных кадров для горной промышленности и транспорта-2017» – Д.:НГУ, 2017.– с.475-480.
2. Елиферов В. Г. Бизнес-процессы. Регламентация и управление / В. Г. Елиферов, В. В. Репин. – М. : ИД "ИНФРА-М", 2009. – 320 с.
3. Новицкий И.В. Алгоритм работы имитационной модели системы принятия решений диспетчером угольной шахты. / И.В. Новицкий., А.В. Малиенко; Національний гірничий університет. Науково-технічний збірник. Гірнична електромеханіка та автоматика. Випуск № 97 . Дніпропетровськ, 2016, с.28 – 35.
4. Пономаренко В. С. Теорія та практика моделювання бізнес-процесів : монографія / В. С. Пономаренко, С. В. Мінухін, С. В. Знахур. – Х. : Вид. ХНЕУ, 2013. – 244 с.
5. ДСТУ ISO 9001:2001. Системи управління якістю. Вимоги. [Текст]. – Офіційний вісник України. – 2006 – N 20. – 1435 с. – (Національні стандарти України).
6. Ули Вали. Управление бизнес-процессами от моделирования до мониторинга с использованием продуктов WebSphere V6 [Текст]: Поддержка IBM Supporttie / Ули Вали, Л. Лейбович, Э. Превоств, и др.; пер. с англ. А. Закис. 1-е изд. IBM® Redbook, 2007. – 423 с.

*Рекомендовано до друку: д-ром техн. наук, проф. Ткачовим В.В.*

УДК 004.45; 004.77

***И.Е. Таланин, д-р физ.-мат. наук, В.И. Таланин, канд. физ.-мат. наук, С.А. Сабанов***  
*(Украина, г. Запорожье, Запорожский институт экономики и информационных технологий)*

## **ОСОБЕННОСТИ ПРИМЕНЕНИЯ ДОКУМЕНТНЫХ БАЗ ДАННЫХ В НАГРУЖЕННЫХ СЕТЯХ**

**Аннотация** Рассмотрены особенности баз данных. Разработан тестовый стенд для выполнения тестирования и анализа полученной статистики. Предложена методика тестирования и сравнения баз данных различных архитектур с использованием клиент-серверной архитектуры тестового стенда с множественным количеством клиентов, каждый из которых в несколько потоков атакует сервер баз данных запросами и собирает статистику по определенным параметрам, на основе которых впоследствии и выполняется сравнение различных баз данных.

**Ключевые слова:** анализ, база данных, тестирование

**Анотация** Розглянуто особливості баз даних. Розроблено тестовий стенд для виконання тестування і аналізу отриманої статистики. Пропонується методика тестування і порівняння баз даних різних архітектур з використанням клієнт-серверної архітектури тестового стенду з багатьма клієнтами, кожен з яких кількома потоками атакує сервер баз даних запитами та збирає статистику за певними параметрами, на базі яких після цього виконується порівняння різних баз даних.

**Ключові слова:** аналіз, база даних, тестування

**Abstract** Features of databases are considered. A test stand was developed to perform testing and analysis of the statistics obtained. A technique for testing and comparing databases of different architectures using a client-server architecture of a test stand with multiple clients, each of which in several threads attacks the database server with requests, and collects statistics on certain parameters, on the basis of which subsequently a comparison of different databases is performed.

**Key words:** analysis, database, testing

**Актуальность работы.** В современном мире информация является одним из важнейших видов товара. При этом часто более значимыми являются не те данные, что хранятся где-нибудь, а результаты обработки этих данных. Поскольку данных много, а, соответственно, и пользовательских запросов к ним, то чрезвычайно важным условием становится повышение скорости обработки информации. Сегодня люди стараются максимально автоматизировать любой вид своей деятельности, насколько это возможно, поэтому данные в основном обрабатываются специальным программным обеспечением.

Разработчикам приложений, направленных на работу с базами данных в свою очередь важно уметь правильно определять требования, выставляемые их приложением к хранилищу и иметь возможность подбора оптимального варианта из всех представленных на рынке. От этого будет зависеть то, насколько надежным будет их продукт и станет ли он успешным. И помогают им в этом специальные системы, которые позволяют в автоматическом режиме быстро выполнять сравнение различных вариантов и, таким образом, помогают принять наиболее верное решение на основе поставленных условий.

Поскольку большинство программ сегодня работают с базами данных, то разработчикам необходимо иметь удобный и гибкий инструмент для тестирования и сравнения различных решений с целью выбора подходящего варианта. Это будет ускорять процесс разработки, упростит процесс поддержки приложения в будущем, поскольку от изначально правильно подобранных технологий, при возможном увеличении нагрузки на приложение в дальнейшем будет зависеть общая надежность и уровень доверия к системе.

**Целью работы** является исследование поведения нереляционных баз данных в нагруженных сетях и их сравнение с реляционной моделью для определения выбора наиболее оптимальной в конкретных условиях.

**Общие вопросы организации хранения данных.** Структурирование – это процесс расположения в определенном порядке, или по определенной схеме [1]. Такой порядок может быть определен по-разному. Самый простой вариант – это расположение информации в хронологическом порядке. Так, например, информацию о некотором событии, взятую из разных источников, располагают последовательно от более раннего к более позднему или наоборот. Таким образом, получаем структуру с данными, расположенными в соответствии со временем, которое описывается данным блоком информации. Другим способом структурирования информации является расположение каждого блока информации в разные разделы в зависимости от того элемента, который описывает данный информационный блок. Существует несколько способов структурирования информации. В основном отличия состоят в том, по какому принципу происходит структурирование [1].

**Модель данных.** В классической теории баз данных, модель данных есть формальная теория представления и обработки данных в системе управления базами данных (СУБД), которая включает, по меньшей мере, следующие аспекты:

- аспект структуры: методы описания типов и логических структур данных в базе данных;
- аспект манипуляции: методы манипулирования данными;
- аспект целостности: методы описания и поддержки целостности базы данных.

Аспект структуры определяет, что из себя логически представляет база данных, аспект манипуляции определяет способы перехода между состояниями базы данных (то есть способы модификации данных) и способы извлечения данных из базы данных, аспект целостности определяет средства описаний корректных состояний базы данных.

Модель данных – это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы – поведение данных [2].

Каждая база данных и СУБД строится на основе некоторой явной или неявной модели данных. Все СУБД, построенные на одной и той же модели данных, относят к одному типу. Например, основой реляционных СУБД является реляционная модель данных, сетевых СУБД – сетевая модель данных, иерархических СУБД – иерархическая модель данных и т.д.

Иногда встречается использование термина "модель данных" в смысле "схема базы данных" ("модель базы данных"). Такое использование является некорректным. Модель данных есть теория, или инструмент моделирования, в то время как модель базы данных (схема базы данных) есть результат моде-

лирования. По выражению К. Дейта соотношение между этими понятиями аналогично соотношению между языком программирования и конкретной программой на этом языке [2].

М. Р. Когаловский поясняет эволюцию смысла термина следующим образом [3]. Первоначально понятие модели данных употреблялось как синоним структуры данных в конкретной базе данных. В процессе развития теории систем баз данных термин "модель данных" приобрел новое содержание. Возникла потребность в термине, который обозначал бы инструмент, а не результат моделирования, и воплощал бы, таким образом, множество всевозможных баз данных некоторого класса. Во второй половине 1970-х годов во многих публикациях, посвященных указанным проблемам, для этих целей стал использоваться все тот же термин "модель данных". В настоящее время в научной литературе термин "модель данных" трактуется в подавляющем большинстве случаев в инструментальном смысле (как инструмент моделирования) [4].

Модель данных определяется как комбинация трех компонентов [5]:

- Коллекции типов объектов данных, образующих базовые строительные блоки для любой базы данных, соответствующей модели.
- Коллекции общих правил целостности, ограничивающих набор экземпляров тех типов объектов, которые законным образом могут появиться в любой такой базе данных.
- Коллекции операций, применимых к таким экземплярам объектов для выборки и других целей.

**Основные параметры при оценке эффективности работы базы данных.** Для примера возьмем приложение, построенное на какой-либо базе данных (на данном этапе его архитектура не имеет значения). Предположим, что изначально программой (сайтом) пользуется небольшое количество людей и там хранятся незначительные наборы данных. При таком раскладе приложение работает корректно, а сервер баз данных пока еще хорошо справляется с нагрузкой. Обобщим и скажем, что, практически любая разработка должна подразумевать дальнейшее развитие и популяризацию, а если так, то наше приложение уже скоро потребует больше ресурсов на качественную обработку запросов от вдруг увеличившегося числа пользователей.

Таким образом, мы приходим к тому, что эффективность сервера баз данных напрямую зависит от возможности спокойно реагировать на резкое увеличение нагрузки, без потери производительности. Тут следует также обратить внимание на тот факт, что мы не учитываем "личностные" характеристики отдельно взятой базы данных, то есть степень ее технологического развития. Такое обобщение позволяет полностью оценивать различные базы данных в пределах определенных ограничений среды и конфигурации. Ограничения могут быть, например, в мощности физических компонентов сервера, основными из которых являются центральный процессор, объем оперативного запоминающего устройства, скорость работы подсистемы ввода/вывода. Также при оценке базы данных учитывается скорость обработки пользовательских запросов в зависимости от динамики изменения нагрузки. При этом должно учитываться качество сетевого соединения, если тестирование проводится, например, в пределах одной организации.

**Разработка тестового стенда.** Целью разработки собственного тестового стенда является исследование возможности сравнивать различные базы данных по определенным статистическим данным, полученным в результате выполнения отдельных тестовых случаев. Общая схема стенда представлена на рис. 1.

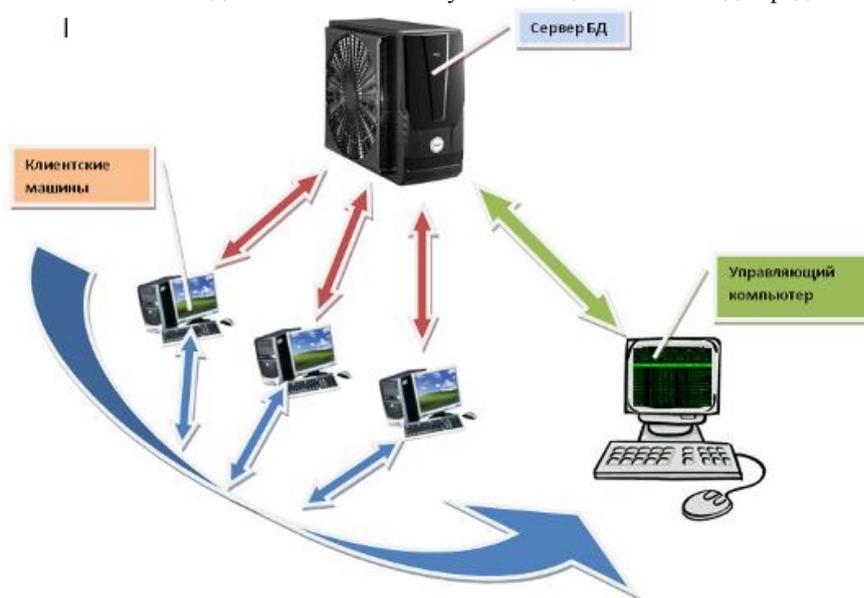


Рис. 1. Общая структура тестового стенда

Вся система тестирования состоит из 3 ключевых частей (система управления, эмуляторы клиентов, монитор ресурсов на сервере баз данных).

Каждый тестовый случай имеет следующие уникальные в своем случае данные:

- 1) параметры, определяющие уникальность теста (рис. 2): тип базы данных; имя; описание;
- 2) параметры для подключения к базе данных (рис. 3): имя пользователя; пароль; адрес сервера баз данных; порт; имя базы данных;
- 3) изменяемый набор тестовых запросов (рис. 4): наборы для категорий запросов INSERT, SELECT, UPDATE, DELET;
- 4) параметры теста: количество потоков(подключений) на клиенте (рис. 5); количество запросов в одном потоке; категории запросов участвующие в текущем тесте; результаты тестирования (рис. 6).

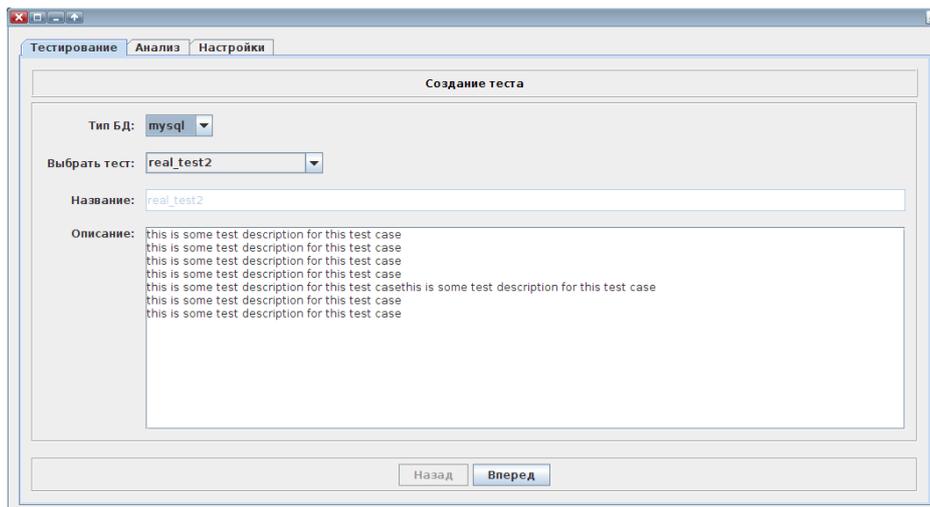


Рис. 2. Первый экран процесса тестирования

Выбирать можно пока только между двумя типами баз данных (mongo и MySQL). При выборе ранее созданного теста загружаются все данные выбранного теста и затем их можно редактировать, но следует учесть, что все предыдущие результаты и настройки данного теста будут затерты. Подключение осуществляется через JDBC драйвер к MySQL и через библиотеку mongo.jar к mongoDB. База данных для тестирования должна быть создана заранее через какую-нибудь СУБД. При добавлении нового теста проводится его валидация путем попытки выполнения данного запроса сразу на сервере баз данных, и если запрос выполняется без ошибок, то он попадает в соответствующий список запросов, иначе выводится сообщение об ошибке. После завершения тестирования можно заняться сравнением результатов. Для этого необходимо выбрать тесты для анализа на вкладке “Анализ” (рис. 6), после чего появится окно с графиками (рис. 7). Если необходимо более детально рассмотреть результаты в какой-нибудь конкретной категории запросов, то можно выбрать соответствующую вкладку, например, SELECT, как показано на рис. 8.

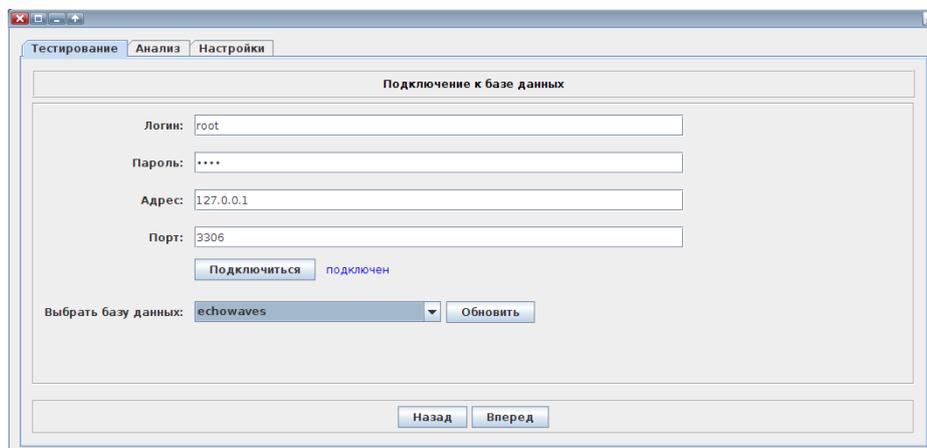


Рис. 3. Настройка подключения к БД

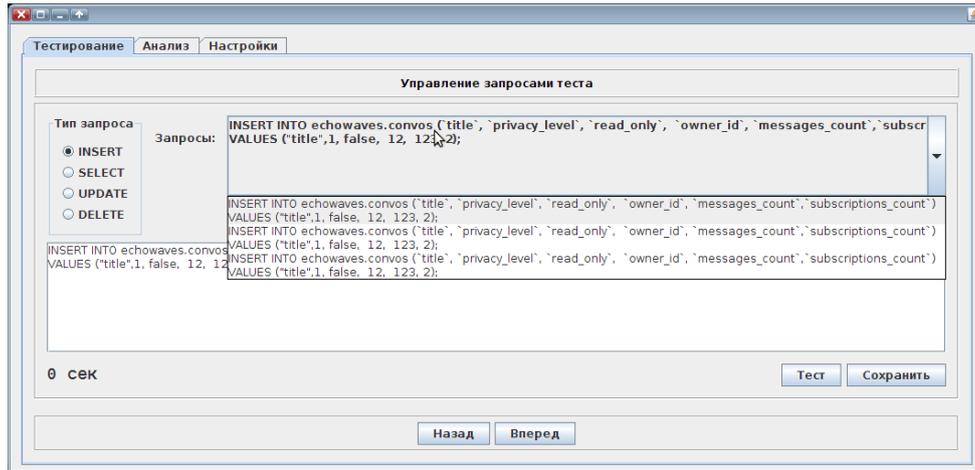


Рис. 4. Управление запросами теста

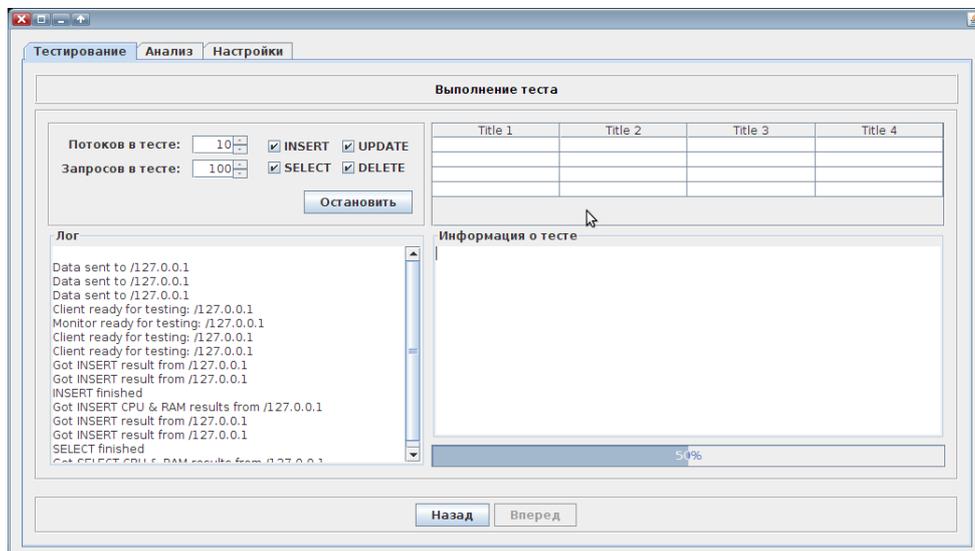


Рис. 5. Окно процесса тестирования

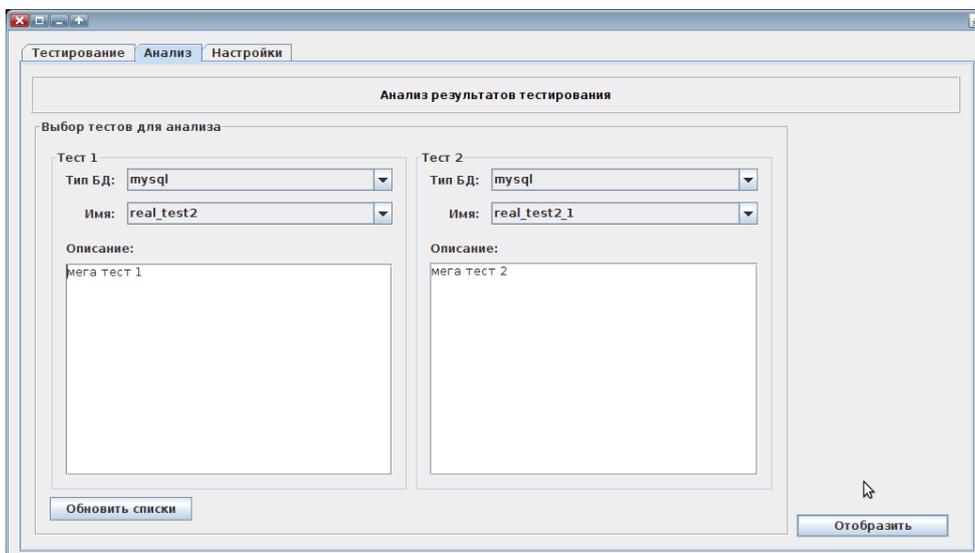


Рис. 6. Выбор результатов тестирования для сравнения

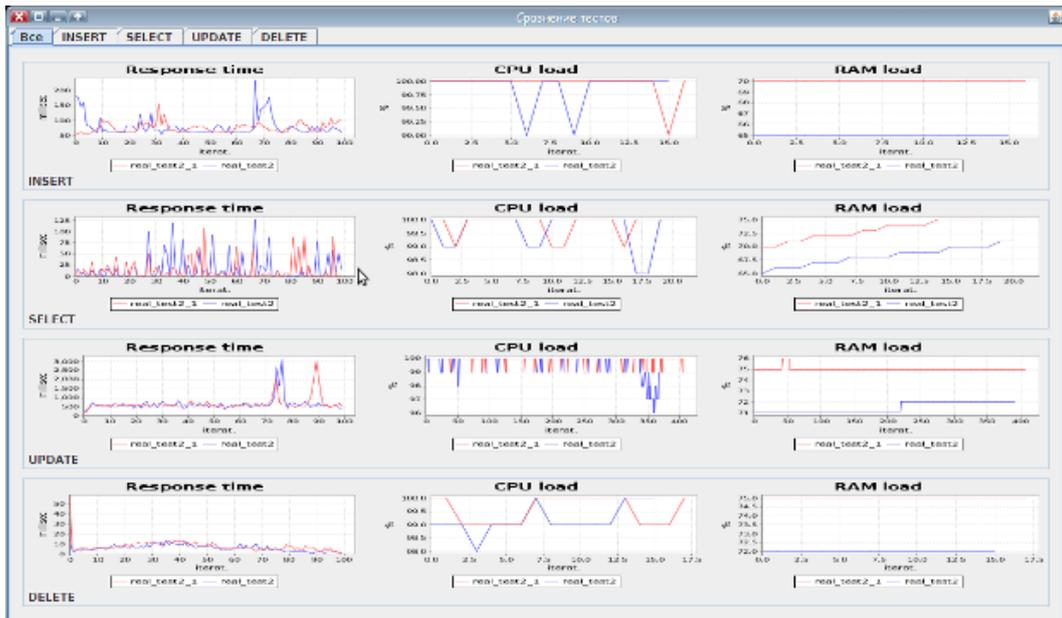


Рис. 7. Сводные результаты сравнения

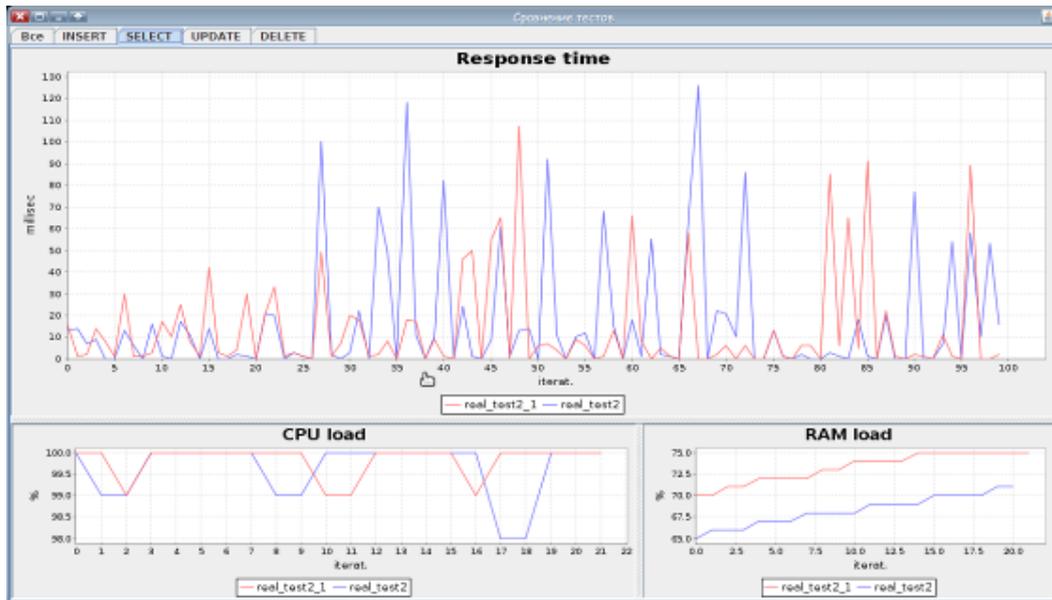


Рис. 8. Результаты сравнения по определенной категории

Каждый тестовый случай может быть сохранен в файл и продолжен или выполнен повторно с новыми изменениями в любое удобное время. После успешного прохождения тестирования результаты всех тестов становятся доступными для сравнения на вкладке “Анализ”.

В задачи клиентской части системы входит выполнение по команде отправки множества запросов на сервер баз данных в соответствии с конфигурацией, полученной с управляющего компьютера и вычисление среднего времени ответов сервера, после чего производится отправка результатов на управляющий компьютер. Программа-монитор работающая на сервере баз данных на протяжении выполнения каждого этапа тестирования фиксирует уровень загрузки центрального процессора и оперативного запоминающего устройства сервера, и затем по запросу отправляет информацию на главный компьютер, где уже и формируется конечный результат. Как и клиентская часть, программа-монитор реализована в виде консольного приложения. На главном компьютере, сразу после получения всех статистических данных со всех клиентских машин и монитора производится объединение, сохранение результатов и вывод графиков проведенного теста. Вся система реализована с использованием языка Java, благодаря чему может работать на любой платформе, где установлен Java версии не ниже 1.6.

Детальный алгоритм работы тестового стенда показан на рис. 9. Взаимодействие сервера с клиентом и монитором показано на рис. 10. Модель данных показана на рис. 11.

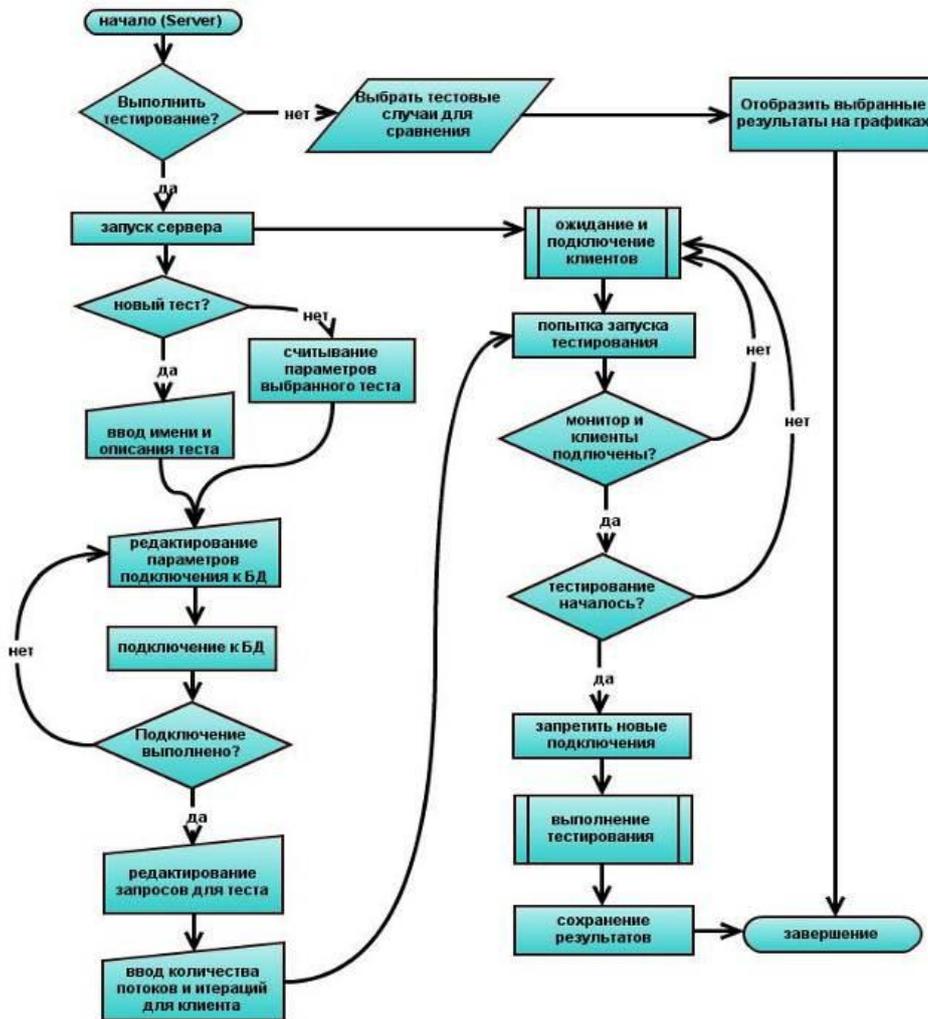


Рис. 9. Схема работы тестового стенда

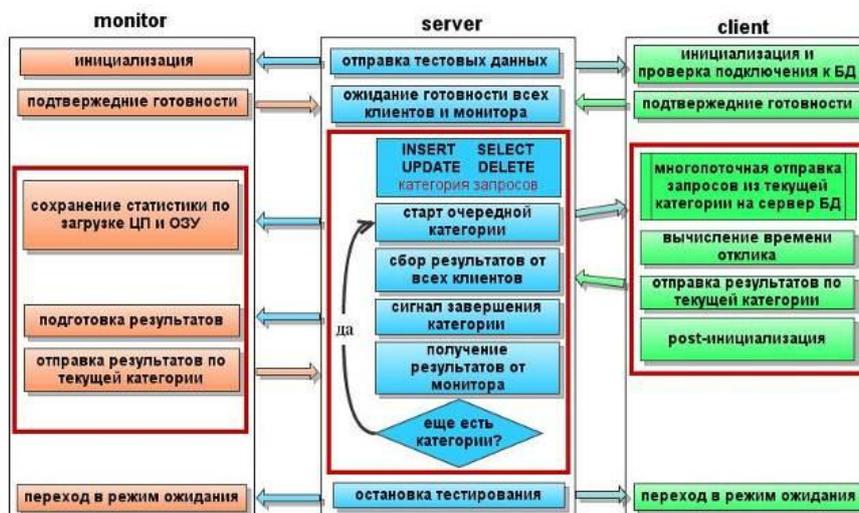


Рис. 10. Взаимодействие сервера с клиентами и монитором

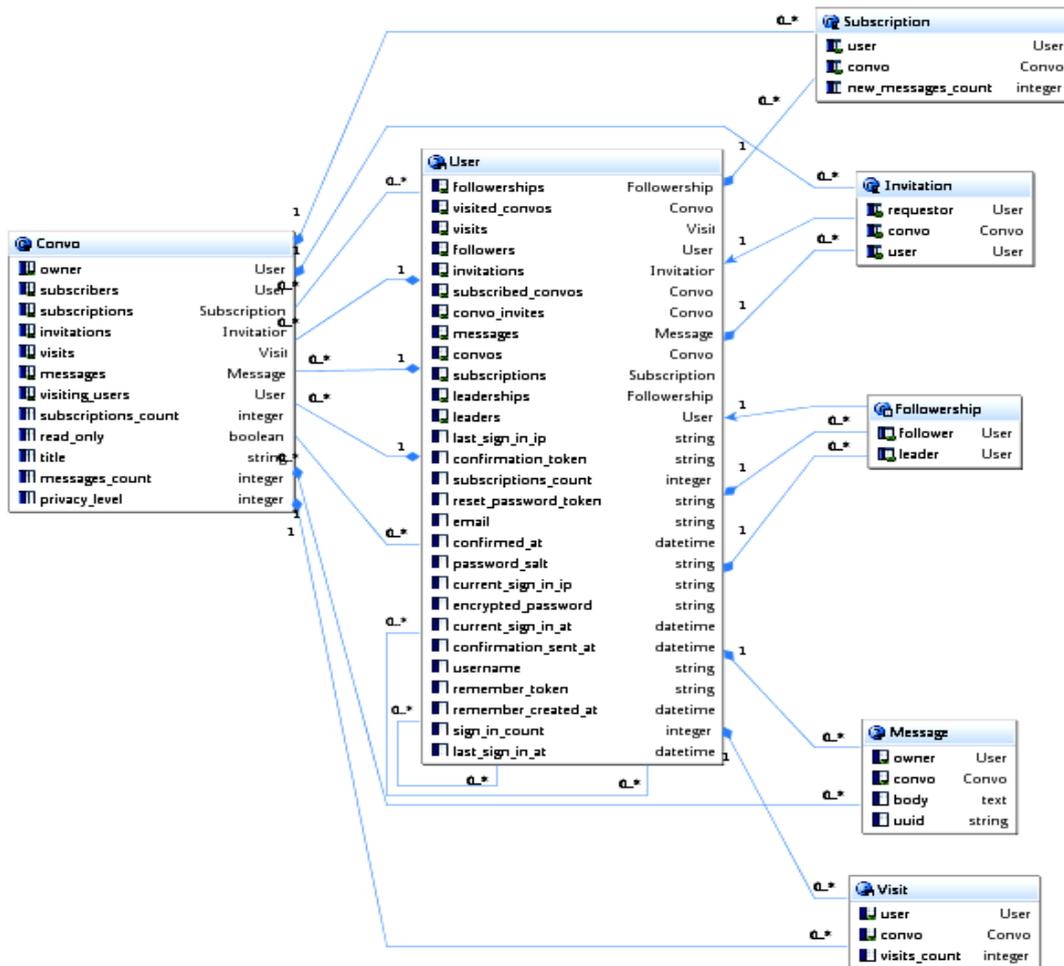


Рис. 11. Модель данных БД для тестирования

**Выводы.** Предложена методика тестирования и сравнения баз данных различных архитектур с использованием клиент-серверной архитектуры тестового стенда с множеством клиентов, каждый из которых в несколько потоков атакует сервер баз данных запросами и собирает статистику по определенным параметрам, на основе которых впоследствии и выполняется сравнение различных баз данных.

Для демонстрации данной методики был разработан тестовый стенд для сравнения реляционной (MySQL) и noSQL (mongo) баз данных, состоящий из управляющего приложения (сервера), множества клиентских программ и программы монитора, которая работает на сервере баз данных и фиксирует динамику изменения загрузки аппаратной части сервера в процессе тестирования. Система позволяет выводить сравнительные графики разных тестов для дальнейшего анализа.

Разработанная система подтверждает актуальность использования распределенной методики нагрузочного тестирования баз данных, поскольку она позволяет моделировать и прогнозировать работу реального многопользовательского приложения на какой-либо базе данных. Это в свою очередь позволяет правильно координировать процесс его разработки, помогает в выборе хранилища и его оптимальной конфигурации.

Проведенные испытания показали, что тестовый стенд позволяет выполнять нагрузочное тестирование, получает достаточно точные и наглядные результаты, которые помогают выполнять сравнение баз данных в ограниченных условиях будущего использования.

Практическая ценность предложенной методики тестирования в виде разработанного программного комплекса заключается в следующем:

- тестирование происходит в среде максимально приближенной к той, в которой будет работать приложение;
- одновременно тестируется влияние сети на взаимодействие с базой данных;
- гибкость модели – можно создать любое количество клиентов, использовать определенные запросы, применять различные конфигурации сервера баз данных;
- при небольшой доработке может быть использована для тестирования других любых реляционных/noSQL.

В качестве перспективы развития приложения можно добавить параметризацию запросов; более гибкое управление параметрами тестирования и клиентами; систему критериального выбора для результатов тестирования, которая бы позволяла выбирать оптимальную базу данных по заданным критериям отбора (например, специфические базы в различных отраслях промышленности, в том числе в горной промышленности). Такие доработки сделают инструмент более удобным и универсальным.

### Список использованных источников

1. Мирошниченко Е.А. К формальному определению понятия “база данных” / Е.А. Мирошниченко // Журн. Проблемы информатики – 2011 – №2. – С. 83-87.
2. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – М.: Вильямс, 2005. – 1328 с.
3. Когаловский М. Р. Энциклопедия технологий баз данных / М.Р. Когаловский. – М.: Финансы и статистика, 2002. – 800 с.
4. Кузнецов С. Д. Основы баз данных / С.Д. Кузнецов. – М.: БИНОМ, 2007. – 484 с.
5. Кригель А. SQL. Библия пользователя. Язык запросов SQL / А. Кригель, Б.Труханов. – М.: Диалектика, 2009. – 752 с.

*Рекомендовано до друку: д-ром техн. наук, проф. Переверзевим А.В.*

УДК 621.396.4

*Ю.А. Паламарчук, Є.М. Сніжко, канд. техн. наук, Н.П. Боцьва, канд. біол. наук  
(Україна, Дніпро, Дніпровський національний університет імені Олеся Гончара)*

### ОПТИМІЗАЦІЯ ЧАСУ РОБОТИ АВТОНОМНИХ ВУЗЛІВ СЕНСОРНОЇ МЕРЕЖІ ДЛЯ АВТОМАТИЗАЦІЇ КОНТРОЛЮ ДОВКІЛЛЯ

***Анотація.** Вирішення актуальної для гірничої сфери задачі підтримки безпечних умов праці персоналу можливо на базі бездротових сенсорних мереж автоматизованого контролю фізичних параметрів середовища в підземних виробках. Розглянуто побудову моделі та моделювання бездротової мережі з кластерною організацією у складі сенсорних вузлів, координаторів кластерів та базової станції. Розраховані параметри роботи мережі залежно від кількості вузлів та розташування базової станції. На основі проведених досліджень однорангових структур мереж з багатокроковими алгоритмами передачі даних показано, що кластерна організація мережі є більш ефективною за енергозбереженням та тривалістю роботи завдяки додатковому стисненню інформації при об'єднанні інформаційних потоків від елементів одного кластера.*

***Ключові слова:** бездротова сенсорна мережа, кластер, автоматизований контроль.*

***Аннотация.** Решение актуальной задачи поддержки безопасных условий труда персонала шахт возможно на базе беспроводных сенсорных сетей автоматизированного контроля физических параметров в подземных выработках. Рассмотрено построение модели и моделирование беспроводной сети с кластерной организацией в составе сенсорных узлов, координаторов кластеров и базовой станции. Рассчитаны параметры работы сети в зависимости от количества узлов и расположения базовой станции. На основе проведенных исследований одноранговых структур сетей с многошаговыми алгоритмами передачи данных показано, что кластерная организация сети является более эффективной по энергосбережению и продолжительности работы благодаря дополнительному сжатию информации при объединении информационных потоков от элементов одного кластера.*

***Ключевые слова:** беспроводная сенсорная сеть, кластер, автоматизированный контроль.*

***Abstract.** Solving the urgent task of the mining industry to support safe working conditions of personnel may be based on wireless sensory networks for automated control of physical parameters in underground workings. The construction of the model and wireless network modeling with cluster organization consisting of sensor nodes, cluster coordinators and base station are considered. The network parameters are calculated based on the number of nodes and the location of the base station. Based on peer-reviewed peer-to-peer network structures, it has been shown that network cluster organization is more effective in terms of energy saving and the duration of work due to the additional compression of information when combining information flows from elements of the same cluster.*

***Keywords:** wireless sensory network, cluster, automated control.*