

УДК 621.311.001.57

СИСТЕМА РЕГИСТРАЦИИ ПАРАМЕТРОВ ИСПЫТАНИЙ СЛОЖНЫХ ИЗДЕЛИЙ НА ОСНОВЕ ДОКУМЕНТНО-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ

Васнев Н.В., Шмидт И.А.

*Пермский национальный исследовательский политехнический университет,
Пермь, e-mail: shmidt@msa.pstu.ru*

В статье рассмотрен принцип работы подсистемы регистрации параметров при проведении испытаний сложных технических изделий, в частности, газотурбинных установок. Выявлены недостатки существующего решения по сохранению информации в виде набора файлов данных. Была предложена подсистема регистрации параметров, базирующаяся на применении документно-ориентированной базы данных MongoDB, обоснован выбор данной БД. Была предложена структура записи для рационального хранения набора трендов данных испытаний. Для проверки технологии сохранения параметров авторами было разработано тестовое приложение, воспроизводящее процесс испытаний. Приложение реализовано в пакете LabVIEW. Описан подробный принцип работы приложения, рассмотрена связь документно-ориентированной базы данных MongoDB и пакета LabVIEW, приведены результаты работы приложения при различных исходных параметрах, показана зависимость скорости записи параметров от размера буфера.

Ключевые слова: хранение и сбор данных, MongoDB, LabVIEW

REGISTRATION SYSTEM PARAMETERS TESTING OF COMPLEX PRODUCTS, BASED ON DOCUMENT-ORIENTED DATABASE SYSTEMS

Vasnev N.V., Shmidt I.A.

Perm national research polytechnic university», Perm, e-mail: shmidt@msa.pstu.ru

The article discusses the method of organizing large data amounts storage during the testing of complex industrial products, in particular gas turbines. Found the drawbacks of the existing method for preservation of information in the form of a set of data files. Was proposed subsystem of register data based on the application of document-oriented database mongodb and proven it choice. Was offered a structure of record for efficient storage of a set data. To check the method of storage data the authors developed a test application, reproducing the process of testing. The application is implemented in the package LabVIEW. Was described the principle of the application, considered the relationship between the document-oriented database mongodb and the LabVIEW program and the results of using different initial parameters. The article shows the dependence of the rate parameters from the size of the buffer.

Keywords: storage and data collection, MongoDB, LabVIEW

Подсистемы регистрации параметров являются важной частью систем испытаний сложных изделий. Развитые системы испытаний могут регистрировать 100–500 параметров, при частоте их регистрации до 200 Гц [1]. Испытания при этом могут идти несколько часов. Качество таких подсистем определяется несколькими свойствами: отказоустойчивостью, долговечностью, скоростью записи, а также удобством извлечения и визуализации полученных данных.

Рассмотрим применяющийся в настоящее время способ сохранения информации при проведении испытаний газотурбинных установок газоперекачивающих агрегатов (ГТУ ГПА). Физические и «датчиковые» значения замеренных параметров (тренды) записываются во время испытаний в двоичный «плоский» файл. Для организации хранения данных такой системы используется структура (совокупность таблиц), описывающая каждый из регистрируемых параметров, которая называется паспорт испытания. Паспорт испытания хранится в РСУБД, он

содержит полное описание для каждого, измеряемого при испытаниях, параметра (см. рис. 1), в том числе: смещение для параметра относительно начала записи, характеристику канала измерения, включая тарифированную характеристику и т.д. Идентификация испытания проходит по имени двоичного файла, в котором закодированы дата и номер испытания. Такая структура хранения показывает довольно высокую скорость записи, но при этом низкий уровень надежности. В целом способ является слабо гибким, т.к. отсутствуют высокоуровневые способы поиска информации, а также нет возможности сопоставлять результаты различных испытаний (они хранятся в разных файлах).

Для повышения уровня надежности и гибкости системы регистрации параметров испытаний естественно вместо набора двоичных файлов использовать базу данных. Таким образом, встает задача выбора базы данных и разработки методики сохранения регистрируемых параметров в выбранную БД.

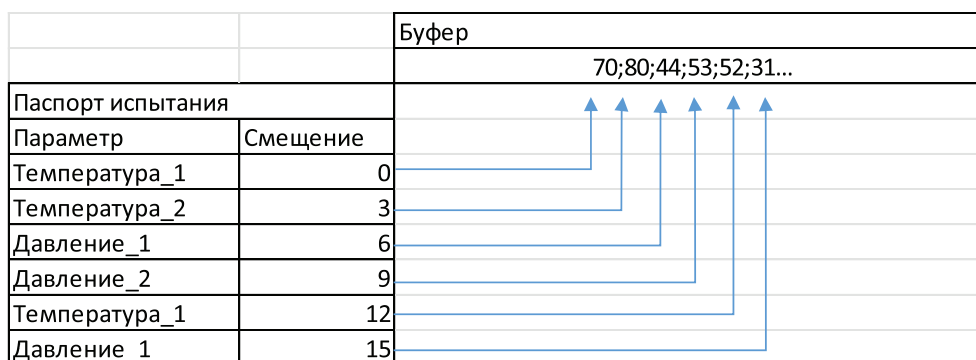


Рис. 1. Организация данных при сохранении в двоичный файл

В рамках данной задачи рассмотрим варианты применимых типов баз данных: РСУБД и документно-ориентированные БД. Применение РСУБД было бы целесообразно при хранении неизменного набора параметров, однако реальные испытания характеризуются различным набором параметров при проведении различных испытаний. Кроме этого механизмы транзакций и контроля целостности данных избыточны для хранения технологических параметров [3]. Применение документно-ориентированных баз данных, которые позиционируются как системы для хранения больших объемов данных без жесткой структуры, является разумной альтернативой. Из существующего многообразия документно-ориентированных баз данных была выбрана MongoDB [2]. Приведем ее характеристики.

- Гибкость и скорость. Хранилища ключей и значений благодаря своей простоте работают быстро и относительно легко масштабируются.

- Простота вывода информации из базы. MongoDB предоставляет развитые механизмы запросов.

- Надежность. БД предоставляет надежная система восстановления данных при включенном механизме дублирования.

- Доступность. MongoDB полностью бесплатна.

- Данные хранятся в BSON формате, что позволяет производить интеграцию с другими языками программирования.

Определим структуру данных обеспечивающих рациональное сохранение трендов параметров. Для отработки вариантов хранения разработано тестовое приложение, которое подключается к MongoDB, затем генерирует числовые значения, имитирующие реальный поток данных испытаний, и заносит их в БД.

Опишем кратко структуру подсистемы регистрации параметров на примере системы для испытаний ГТУ ГПА [4]. Все данные перед тем как попасть в БД, проходят три уровня обработки. Первый, нижний уровень подсистемы обеспечивает опрос модулей связи с объектом, получение данных и их транспортировку на второй уровень. На втором уровне проходит первичная обработка поступивших данных и их графическая визуализация для управления процессом испытаний. В качестве основы для программного обеспечения второго и первого уровня использована многофункциональная платформа LabVIEW.

Третий верхний уровень подсистемы получает информацию от среднего и должен обеспечивать хранение всей информации по ходу испытаний с возможностью выполнения гибких запросов по любым параметрам, включая поиск результатов испытаний и их сравнение. [5]

При получении информации с первого уровня на верхних уровнях запускаются обработчики данных и выделяются сегменты памяти, называемые буферами опроса. Буфер опроса – это область в памяти содержащая коды измеряемых параметров. Каждый буфер опроса характеризуется своим собственным набором содержащихся в нем регистрируемых параметров и частотой обновления. Информация с первого на второй уровень передается кадрами. Кадр – это набор значений параметров, относящихся к одной временной отметке. Для достижения быстрой скорости записи буфер большого размера может принимать несколько полученных кадров. Когда этот буфер заполняется, он передается на верхний уровень для записи в БД [6].

Разработанное авторами тестовое приложение, которое относится к третьему уровню, позволяет решить следующие задачи:

- Проверка возможности записи данных в БД MongoDB средствами LabVIEW.

- Определение и изменение структуры документа, пригодной для сохранения текущих параметров процесса.

- Определение времени сохранения данных в зависимости от количества регистрируемых параметров и записей в БД.

Для работы с базой данных MongoDB из пакета LabVIEW необходимо добавить в проект библиотеки MongoDB.Bson.dll и MongoDB.Driver.dll. После этого в LabVIEW будут доступны компоненты .NET, которые позволяют создавать BSON-классы и получать доступ к их функциям. С помощью .NET-компонент происходит подключение к серверу, на котором хранится БД, далее подключение к БД, затем подключение к необходимым коллекциям, из которых можно считывать или записывать данные.

Данные, получаемые в ходе испытания, должны быть структурированы. Для этого требуется разработать сложную структуру BSON-документа, которая должна показать быструю скорость записи и высокую гибкость, при этом позволяющую легко извлекать полученные данные для визуализации. Вставка одной записи по разработанной структуре в MongoDB выглядит следующим образом:

```
db.collection_name.insert({sensor_name: [{time:0.1, value:0.21},{ time:0.2, value:0.25}]})
```

Входными параметрами такой структуры являются время записи и значение физической величины, которые необходимо перевести в числовой BSON-формат. Так как MongoDB работает с JSON-

структурой, которая состоит из пары ключ – значение, то необходимо привязать значение параметра и времени к нужным ключам. Предварительно создается объект класса BsonDocument. Полученные с помощью .NET-компонент пары ключ-значения для времени и величины параметра объединяются в список для записи в BSON-документ. Чтобы занести объединенные параметры в документ, вызывается функция Add(), которая входным значением принимает список элементов. На рис. 2 представлен SubVI для формирования BSON-документа.

Выходом SubVI будет ссылка на BSON-документ. При испытании исследовательских систем производится сбор параметров с сотен датчиков. Каждый параметр имеет свою частоту записи. Частота записи некоторых параметров может совпадать. Поэтому для достижения требуемой скорости сохранения данных в БД параметры с одинаковыми частотами можно записывать за один кадр. Кадр – набор значений параметров, относящихся к одной временной отметке. Кроме этого будет использован буфер, который будет записывать все параметры (как набор кадров) в оперативную память, затем сохранять массив данных в БД.

Тестовое приложение разбито на фреймы структуры «Последовательность» (Sequence Structure). В первых фреймах происходит инициализация базы данных и подключение к необходимой коллекции. В последних проходит закрытие БД и вывод возникших ошибок. Между этими фреймами происходит генерация сигналов и формирование массива документов. Структура приложения показана на рис. 3.

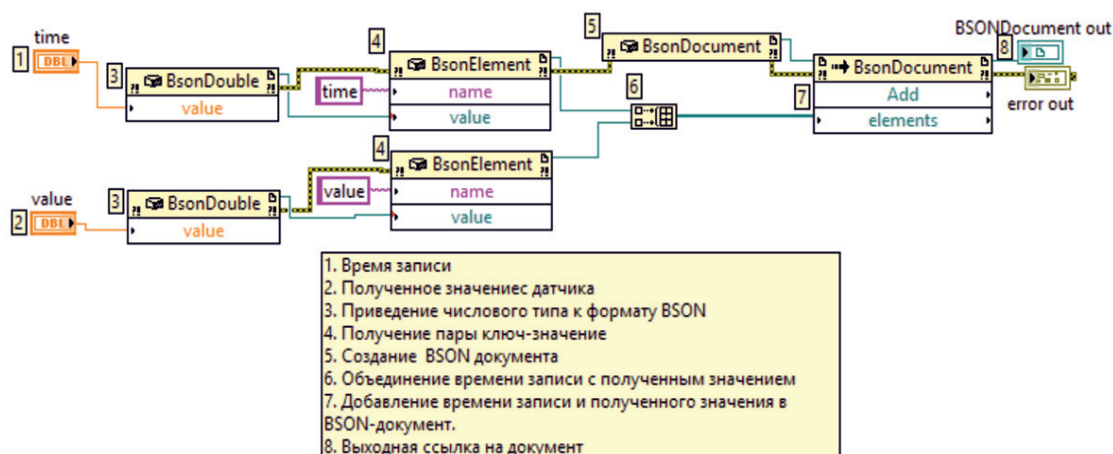
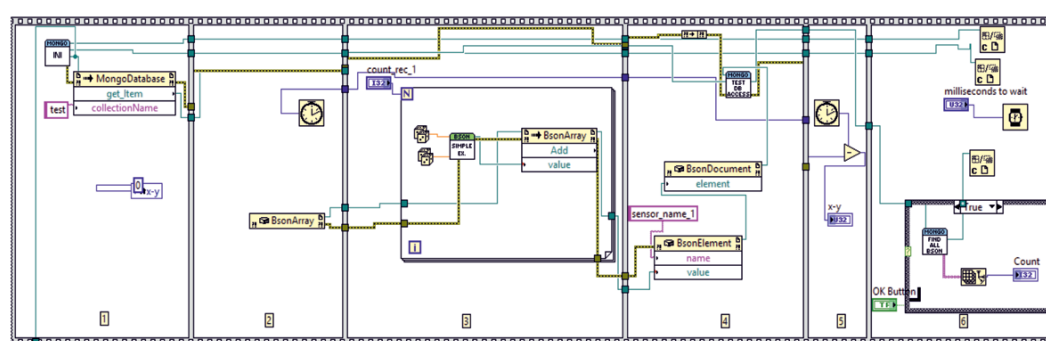


Рис. 2. SubVI для создания BSON-документа



1. Подключение к базе данных MongoDB.
2. Инициализация массива. Начало отчета времени.
3. Генерация сигналов. Запись в буфер.
4. Запись в массив. Создание BSON-документа. Запись в БД.
5. Конец отчета времени.
6. Подсчет количества записей. Закрытие БД.

Рис. 3. Структура тестового приложения

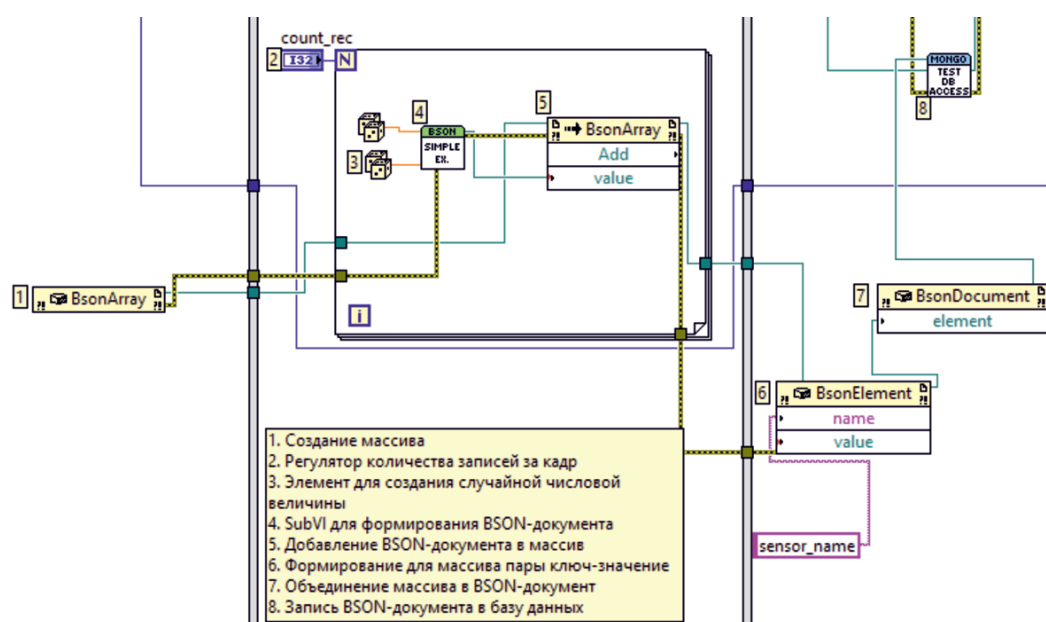


Рис. 4. Формирование массива документов

Зависимость скорости записи от размера буфера

N	to, мс	t1, мс	Примечания	Min to, мс	Max to, мс
1	11	11	Стабильная скорость записи	11	12
10	32,5	3,25	Небольшой разброс во времени	25	40
100	202	2,02	Редкие скачки Max to >= 380	330	360
500	1125	2,25	Очень редко скачки Max to <= 1160	1112	1138
1000	2724	2,724	После 25000 записей увеличивается частота скачков Max to >= 2800. С увеличением количества записей величина при скачке увеличивается. При 70000 записей скачки Max to >= 2900	2688	2760

Перед тем как сформировать выходной BSON-документ, необходимо создать объект типа `BsonArray`, который будет хранить документы с полученными значениями времени и физической величины. Далее происходит формирование случайных сигналов и их передача в `SubVI`, рассмотренный выше. `SubVI` возвращает ссылку на сформированный документ, который с помощью функции `Add ()` у класса `BsonArray` добавляется в созданный ранее массив. Работа всех компонентов проходит в цикле. Регулируя количество итераций, изменяют количество величин параметров для внесения в БД. Функция `Add ()` класса `BsonArray` возвращает ссылку на массив. Для того чтобы можно было обращаться к массиву, ему так же присваивается ключ. Полученная пара добавляется в BSON-документ, который заносится в БД. В кадрах модели были расположены компоненты, отсчитывающие время от начала работы каждого кадра, благодаря чему можно было рассчитать время записи одного сформированного документа в БД. На рис. 4 представлены компоненты, отвечающие за формирование массива для одного датчика.

Результаты работы представлены в таблице, где N – количество записанных величин параметра, t_0 – общее время записи, $\text{Min } t_0$ – минимальное наблюдаемое время записи, $\text{Max } t_0$ – максимальное наблюдаемое время записи.

Таблица отражает зависимость скорости записи параметров от размера буфера. Изменяя размер буфера, можно добиться требуемой скорости записи.

Заключение

Была рассмотрена связь документо-ориентированной базы данных MongoDB и пакета LabVIEW, была разработана сложная структура хранения данных, которая является быстродействующей по скорости записи и удобной по выводу данных из БД. Для большого количества датчиков достаточно объявить каждый датчик в модели, не меняя при этом структуры хранилища. В статье подробно описано, какие компоненты пакета LabVIEW необходимо задей-

ствовать для того, чтобы получать значения физических величин с датчиков и записывать их в БД. Изменяя размер буфера, можно изменять скорость записи данных.

Список литературы

1. Кавалеров Б.В., Казанцев В.П., Шмидт И.А. Компьютерные и полунатурные испытания средств управления энергетических газотурбинных установок // Информационно-управляющие системы. – 2011. – № 4. – С. 34–41.
2. Кайл Бэнкер. MongoDB в действии – М.: ДМК Пресс, 2012. – 395 с.
3. Попов Д.А., Шмидт И.А. Разработка системы управления архивными данными испытаний газотурбинных установок большой мощности // Современные проблемы науки и образования. – 2014. – № 2.; URL: <http://science-education.ru/ru/article/view?id=12035>.
4. Попов Д.А., Шмидт И.А. Оперативный сбор параметров при испытаниях различных типов ГТУ на базе оборудования NATIONAL INSTRUMENTS // Научно-технический вестник Поволжья. – 2014. – № 2. – С. 192–196.
5. Попов Д.А., Шмидт И.А. Выбор методов реализации верхнего уровня проведения испытаний газотурбинных установок на базе аппаратных средств NATIONAL INSTRUMENTS // Научные исследования и инновации. – 2012. – Т. 6, № 1–4. – С. 249–254.
6. Попов Д.А., Шмидт И.А. Разработка функциональной структуры программного комплекса испытаний газотурбинных установок мощностью до 40 мвт // Научные исследования и инновации. – 2012. – Т. 6, № 1–4. – С. 264–270.

References

1. Kavalero B.V., Kazancev V.P., Shmidt I.A. Komp'yuternye i polunaturalnye ispytaniya sredstv upravleniya jenergeticheskikh gazoturbinnykh ustanovok // Informacionno-upravljajushhie sistemy. 2011. no. 4. pp. 34–41.
2. Kaji Bjenker. MongoDB v dejstvii M.: DMC Press, 2012. 395 p.
3. Popov D.A., Shmidt I.A. Razrabotka sistemy upravlenija arhivnymi dannymi ispytaniy gazoturbinnykh ustanovok bolshoj moshhnosti // Sovremennye problemy nauki i obrazovaniya. 2014. no. 2.; URL: <http://science-education.ru/ru/article/view?id=12035>.
4. Popov D.A., Shmidt I.A. Operativnyj sbor parametrov pri ispytaniyah razlichnykh tipov GTU na baze oborudovaniya NATIONAL INSTRUMENTS // Nauchno-tehnicheskij vestnik Povolzhja. 2014. no. 2. pp. 192–196.
5. Popov D.A., Shmidt I.A. Vybore metodov realizacii verhnego urovnja provedeniya ispytaniy gazoturbinnykh ustanovok na baze apparatnykh sredstv NATIONAL INSTRUMENTS // Nauchnye issledovaniya i innovacii. 2012. T. 6, no. 1–4. pp. 249–254.
6. Popov D.A., Shmidt I.A. Razrabotka funkcionalnoj struktury programmnogo kompleksa ispytaniy gazoturbinnykh ustanovok moshhnostju do 40 mvt // Nauchnye issledovaniya i innovacii. 2012. T. 6, no. 1–4. pp. 264–270.