

А.В. Замятин

доктор техн. наук, доцент
E-mail: avzamyatin@inbox.ru

П.М. Острасть

канд. техн. наук
E-mail: op@sibmail.com

(Национальный исследовательский Томский
государственный университет)

Е.А. Телицын
(АО «ЭлеСи»)

В.Н. Тренькаев

канд. техн. наук, доцент
E-mail: tvnik@sibmail.com

В.Д. Яновский

E-mail: waldemar.yan@gmail.com

(Национальный исследовательский Томский
государственный университет)
г. Томск, Российская Федерация

Высокопроизводительный сервер истории системы диспетчерского управления и сбора данных

Представлены результаты исследований по построению высокопроизводительного сервера истории системы диспетчерского управления и сбора данных. Предложена и апробирована методика определения характеристик сервера истории, существенно влияющих на производительность. С использованием модели рабочей нагрузки, построенной на основе стандарта OPC UA, проведено тестирование экспериментального прототипа сервера истории. Результаты тестирования показывают, что сервер истории в предложенной архитектуре обеспечивает требуемые для современных систем промышленной автоматизации показатели.

Ключевые слова: SCADA-система; OPC UA; СУБД; БД; сервер истории; хранилище типа «ключ-значение»; временные ряды; производительность.

A.V. Zamyatin

Doctor of Techn. Sciences, Associate Professor
E-mail: avzamyatin@inbox.ru

P.M. Ostrast

Cand. of Techn. Sciences
E-mail: op@sibmail.com

(National Research Tomsk State University)

E.A. Telitsin
(JSC «Elesy»)

V.N. Trenkaev

Cand. of Techn. Sciences, Associate Professor
E-mail: tvnik@sibmail.com

V.D. Janowsky

E-mail: waldemar.yan@gmail.com

(National Research Tomsk State University)
Tomsk, Russian Federation

High-performance Data Historian of Supervisory Control and Data Acquisition

The article presents the results of research on high-performance data historian server design. Data historian server performance evaluation technique is proposed and tested. The experimental model of the data historian server using OPC UA model of workload performance is tested. It is shown that data historian with proposed architecture can provide the quality of service in terms of computational performance required for up-to-date SCADA systems.

Keywords: SCADA; OPC UA; DBMS; DB; data historian server; key-value store; time series; computational performance.

Введение

Неотъемлемой составляющей любой системы диспетчерского управления и сбора данных (SCADA-системы) является база данных (БД) технологических процессов. Обычно за ведение БД отвечает отдельная компонента SCADA-системы – сервер истории (СИ). Во многих современных SCADA-системах в качестве СИ используются решения на базе популярных коммерческих реляционных СУБД, например MS SQL или Oracle [1]. Однако такой подход часто дает недостаточную производительность, поскольку требуется сбор, обработка и хранения данных в режиме реального времени, но не учитывается специфика модели данных прикладной области. Более перспективным представляется использование специализированных СУБД – NoSQL СУБД или СУБД временных рядов [2...4]. Поэтому, данное исследование, преимущественно, посвящено исследованию элементов архитектуры специализированных СУБД, существенно влияющих на производительность СИ. Основными задачами СИ являются оперативный сбор и сохранение технологических данных, а также предоставление доступа различных приложений к архивным данным (системе генерации отчетов, приложению отображения в виде динамических графиков наблюдаемых параметров технологического процесса и др.). Сбор и предоставление данных осуществляется по технологии OPC UA, т. е. в соответствии со спецификацией OPC UA и на базе коммуникационного стека OPC UA [5]. Управление доступом и другие операции обработки данных выполняются с помощью СУБД, выполняющей при этом роль OPC UA клиента/сервера.

Как показывает практика, основным фактором, влияющим на быстродействие СИ, является выполняемая им роль СУБД. Получение/предоставление данных по OPC UA стандарту, а также некоторая предобработка данных, например фильтрация, в меньшей степени влияют на производительность СИ, чем выполнение операций над данными, которые реализуются посредством СУБД. В свою очередь, производительность СУБД существенным образом определяется моделью оперируемых данных. Распространенные реляционные СУБД позволяют задавать сложные структуры данных и использовать SQL язык для манипулирования данными, обеспечивают их целостность. Однако столь богатые функциональные возможности ведут к снижению производительности. Так, за счет отказа от полноценной поддержки языка SQL и/или ACID-транзакций (удовлетворяющих условиям атомарности, согласованности, изолированности и устойчивости), часто удается добиться увеличения производительности [6]. Кроме того, для БД в SCADA-системах типичная модель данных – иерархическая, а не реляционная. Технологические данные, как правило, представляются в виде временных рядов (например, собранные в разные моменты времени значения параметров технологических процессов). Поэтому не требуется обработка сложных структур данных, часто отсутствуют множественные

взаимосвязи между данными, необязательна поддержка транзакций.

Все это позволяет разработать оригинальную методику экспериментальной оценки характеристик СИ, определив основные элементы его архитектуры, существенно влияющие на производительность. Отдельного внимания требует ключевая часть СИ – хранилище данных, которое также требуется исследовать и выбрать базовую СУБД. Наконец, необходимо реализовать прототип СИ и оценить ключевые характеристики его производительности. Именно этим аспектам построения высокопроизводительного сервера истории системы диспетчерского управления и сбора данных посвящена данная работа.

1. Методика экспериментальной оценки характеристик сервера истории

Для определения характеристик и элементов архитектуры СИ, от которых существенно зависит производительность, а также для выявления оптимальных режимов работы СИ, предлагается методика, включающая следующие оценки:

- исследование зависимости производительности СИ от базовых характеристик ядра СУБД (метод описания данных, метод манипулирования данными, метод контроля параллельного доступа к данным и т. п.). Нахождение верхнего предела производительности ядра СУБД;

- исследование зависимости производительности СИ от базовых характеристик технологических данных – тип данных (теги, статусные сообщения, аварийные сообщения), число контролируемых параметров, число индексируемых атрибутов и т. п.;

- исследования зависимости производительности СИ от объемов и способов сжатия технологических данных;

- оценка ресурсоемкости базовых операций СИ таких, как сбор данных, предобработка данных, сжатие данных, сохранение данных, предоставление данных и т. п.;

- исследование зависимости производительности СИ от параметров дисковой подсистемы (зависимости объема записываемых на жесткий диск данных от объема поступающих технологических данных, скорости записи на жесткий диск от скорости поступления технологических данных и т. п.).

Следует отметить, что одно из требований, предъявляемых к СУБД для SCADA-систем, – высокая надежность сбора и сохранения технологических данных, что можно обеспечить, используя СУБД с ориентацией на хранение данных на жестких дисках, а не в оперативной памяти.

Исследования следует производить для разных режимов работы СИ:

- сбор оперативных технологических данных;
- сбор и одновременно анализ данных;
- исключительно анализ технологических данных.

Методику предлагается реализовать с помощью проведения экспериментов с рабочим прототипом СИ (экспериментальным образцом СИ). В качестве оценочных параметров производительности использованы следующие характеристики:

- пропускная способность ядра СУБД СИ/дисковой подсистемы;
- быстродействие базовых операций СИ;
- количество потребляемых ресурсов процессора/оперативной памяти.

При тестировании производительности СУБД в основном используются тесты, которые моделируют рабочую нагрузку, характерную для систем поддержки принятия решений, типовых транзакционных систем. В случае тестирования экспериментального образца СИ необходимо использовать модель рабочей нагрузки, которая характерна для систем промышленной автоматизации. Построение такой модели возможно на базе спецификации OPC UA.

Современный промышленный стандарт OPC UA унифицирует процесс передачи технологических данных в промышленных сетях. В качестве источников данных в данном стандарте выступают OPC UA серверы, которые имеют доступ как к оперативным, так и архивированным технологическим данным. Интерфейс между OPC UA сервером и клиентом определяется как набор сервисов. Вызов сервиса аналогичен вызову метода в языках объектно-ориентированного программирования.

Информационная модель технологии OPC UA основывается на понятии узла [7]. Узел имеет атрибуты, доступные для чтения, методы, которые могут быть вызваны, а также события, которые могут быть переданы. Узлы используются для представления реальных объектов предметной области. Адресное пространство OPC UA сервера представляет собой множество узлов, доступных клиенту с помощью сервисов. Каждый узел имеет уникальный идентификатор *nodeid*, состоящий из трех элементов: индекс пространства имен (*namespaceindex*), формат и тип данных идентификатора (*identifiertype*), идентификатор (*identifier*).

Рабочая нагрузка для СИ может быть представлена как набор значений параметров технологических процессов вместе с некоторыми их характеристиками [8]. Такие технологические данные будем далее называть сигналами (*тегами*), имеющими следующие характеристики:

- составной идентификатор узла (*nodeid*);
- значение контролируемого параметра технологического процесса (*value*);
- признак достоверности значения параметра (*statuscode*);
- время изменения параметра на источнике данных (*sourcetimestamp*);
- время получения значения параметра на сервере (*servertimestamp*).

В общем случае, технологические данные представляются как записи с полями, каждое из которых

соответствует некоторой характеристике технологического процесса, со скалярным либо строковым типом данных.

2. Нагрузочное тестирование хранилища данных

В качестве ядра СУБД (подсистемы хранения) для SCADA-системы предлагается использовать систему управления данными категории NoSQL, которая должна предоставлять низкоуровневый интерфейс доступа к данным. При этом высокоуровневые операции над технологическими данными, например операции над временными рядами или другие OPC UA операции, следует реализовать на базе такого ядра.

Существует несколько видов систем управления данными категории NoSQL: хранилища типа «ключ-значение», документные СУБД, BigTable-подобные СУБД [9]. Причем многие такие системы распространяются на условиях свободного лицензионного договора, что существенно облегчает их использование для исследовательских целей. Такие NoSQL СУБД различаются как методами описания и поддержки целостности данных, так и методами манипулирования данными. Для выявления NoSQL решения, которое наиболее пригодно для использования в SCADA-системах, в данной работе необходимо провести соответствующие экспериментальные исследования по оригинальной методике.

Традиционно оценка производительности СУБД производится с помощью тестов, которые моделируют рабочую нагрузку, характерную для некоторой предметной области [10]. Существует достаточно много признанных специалистами тестов производительности такого типа, например семейство тестов TPC (*Transaction Processing Performance Council*) или AS3AP (*ANSI SQL Standard Scalable and Portable*). Однако большинство таких тестов предназначены для оценки реляционных СУБД, причем с использованием модели рабочей нагрузки, которая неадекватна для систем промышленной автоматизации. В тестовых испытаниях экспериментального образца СИ используется генератор тестовых данных, построенных на основе спецификации OPC UA, что позволяет симулировать различные виды технологической нагрузки.

Для проведения тестовых испытаний в качестве ядра СУБД экспериментального образца СИ предлагается использовать хранилище типа «ключ-значение». Последнее в простейшем случае можно рассматривать как ассоциативный массив, когда каждому значению сопоставляется уникальный ключ. В хранилищах типа «ключ-значение» работа с данными осуществляется с помощью простых операций вставки, удаления и поиска по ключу, вторичные ключи и индексы не поддерживаются. На основе такого NoSQL-хранилища возможно реализовывать OPC UA операции над сигналами.

При сохранении и обработке сигналов с помощью хранилища типа «ключ-значение» в качестве ключа предлагается рассматривать пару (идентификатор узла,

время изменения), а в качестве значения – совокупность других характеристик сигнала. При записи в хранилище сигналы сначала упорядочиваются по идентификатору, а потом по времени. В качестве базовых операций над сигналами выступают операции вставки, чтения, удаления значений контролируемых параметров за заданный промежуток времени.

Одно из главных требований, предъявляемых к СУБД для SCADA-систем, – высокая скорость операции вставки данных, т. к. технологические процессы генерируют данные с высокой скоростью и в режиме реального времени. Кроме того, в рамках рассматриваемой архитектуры возможно использование только встраиваемой СУБД с ориентацией на хранение данных на жестких дисках.

Исходя из данных требований, проведен сравнительный анализ существующих хранилищ типа «ключ-значение» [10]. В итоге, с учетом результатов тестирования [12...16], выбраны следующие кандидаты на роль ядра СУБД для экспериментального образца СИ: LevelDB [17], HyperLevelDB [18], Sophia [19], BangDB [20]. Кроме выбранных NoSQL-хранилищ оценивалась также реляционная СУБД SQLite [21].

Чтобы сделать окончательный выбор хранилища типа «ключ-значение», для каждого кандидата проведено тестирование скорости вставки данных с использованием модели рабочей нагрузки, построенной в соответствии со спецификацией OPC UA, и описанной выше. Разработан генератор тестовых данных и тестовый стенд с возможностью подключения хранилища из заданного списка (*LevelDB, HyperLevelDB, Sophia, BangDB, SQLite*). При проведении исследований использована рабочая станция со следующими характеристиками: процессор – Intel® Core™ i5-6500 CPU @ 3.20GHz × 4, объем оперативной памяти – 8 ГБ, жесткий диск – 1TB 7200 RPM 32MB Cache, операционная система – Ubuntu 16.04 LTS, файловая система – ext4.

В ходе тестирования в течении заданного промежутка времени на хранилище подается симулируемый поток сигналов и замеряется скорость вставки данных. Запись сигналов производится при упорядоченном или случайном ключе, по одному сигналу или пакетами. Требуется выявить хранилище типа «ключ-значение» с наибольшей скоростью вставки.

В таблице приведены результаты нагрузочного тестирования кандидатов. Из данной таблицы видно,

что наибольшую скорость вставки показало хранилище LevelDB с результатом 440 тыс. сигналов/секунду при записи одиночными сигналами и 862 тыс. сигналов/секунду при записи большими группами сигналов.

3. Методика оценки производительности сервера истории

Для выполнения оценки производительности реализован прототип сервера истории, обобщенная архитектура которого представлена на рисунке 1. Этот экспериментальный прототип имеет в своем составе две типовые компоненты СИ: Коллектор и Сервер исторических данных (СИД).

Коллектор располагается вблизи Источника технологических данных и выполняет функцию «сохрани и передай». Коллектор в качестве OPC UA клиента подключается к Источнику технологических данных, собирает и передает данные СИД, где они записываются в БД и предоставляются Потребителю технологических данных. Основная задача Коллектора – буферизация данных в случае нарушения связи с СИД, для чего используется файловый кольцевой буфер, накапливающий технологические данные до восстановления соединения. СИД содержит ядро СУБД (подсистема хранения), которое предоставляет низкоуровневый интерфейс доступа к данным. В модуле OPC UA операций на базе возможностей ядра реализуются операции над технологическими данными в соответствии со спецификацией OPC UA.



Рис. 1. Архитектура экспериментального образца Сервера истории

Таблица

Результаты тестирования кандидатов

		Скорость вставки данных × 10 ³ , сигналов/секунду				
вид записи	тип ключа	LevelDB	HyperLevelDB	BangDB	Sophia	SQLite
одиночная	упорядоченный	372	388	300	65	51
	случайный	70	218	191	47	52
групповая	упорядоченный	862	676	нет опции	92	129
	случайный	440	323	нет опции	48	138

В рамках рассмотренной архитектуры экспериментального образца СИ возможно оценить ресурсоемкость следующих операций:

- Операции на Коллекторе:
 - получение данных по OPC UA технологии;
 - обработка данных (фильтрация, сериализация, запись в буфер);
 - передача данных СИД по сети (на базе TCP/IP протокола);
- Операции на СИД:
 - обработка данных (десериализация, подготовка к записи в хранилище);
 - низкоуровневые операции ядра СУБД;
 - высокоуровневые OPC UA операции;
 - предоставление данных по OPC UA технологии.

Одно из главных требований, предъявляемых к СИ, – обеспечение сбора данных в режиме реального времени. Поэтому при апробировании вышеописанной методики по выявлению характеристик СИ, существенно влияющих на его производительность, рассматривается только режим «чистого» сбора данных (ситуация одновременного выполнения операций вставки и чтения данных исключена из рассмотрения). Кроме того не учитывается задержка при получении/предоставлении данных по OPC UA технологии, а также при передаче данных по компьютерной сети.

В ходе тестирования экспериментального образца СИ на Коллектор в течении заданного промежутка времени подается симулируемый поток сигналов и измеряется быстродействие ресурсоемких операций, которые задействованы при сборе технологических данных:

- сбор и обработка данных на Коллекторе (операция 1);
- предобработка данных на СИД (операция 2);
- запись данных в БД с помощью LevelDB (операция 3).

Рассматривается два способа представления ключа хранилища LevelDB: ключ – это пара (идентификатор узла, время измерения), либо ключ – это пара (время измерения, идентификатор узла). В первом случае сигналы при записи упорядочиваются сначала по идентификатору, а потом по времени, а во втором случае – наоборот. Запись сигналов производится порциями с предварительным накоплением сигналов в буфере.

Параметры генератора тестовых данных: способ генерации значения контролируемого параметра – псевдослучайный, частота изменения значения – 10^2 мс, количество сигналов – 10^5 , время изменения контролируемого параметра – текущее системное время, продолжительность генерации – $12 \cdot 10^4$ мс. Тем самым моделируется ситуация продолжительной пиковой нагрузки, при которой на SCADA-систему поступает большой поток технологических данных.

На рисунке 2 приведены результаты тестирования для ключа (идентификатор узла, время измерения) и ключа (время измерения, идентификатор узла). Показана зависимость быстродействия рассмотренных операций СИ от размера буфера ядра СУБД.

Как видно из рисунка 2, самая медленная операция СИ – операция записи сигналов в БД (операция 3). Данная операция примерно в 2,5...3 раза медленнее, чем операции сбора данных (операция 1), то есть производительность СИ может быть увеличена в 2,5...3 раза, если использовать более быструю подсистему хранения, чем LevelDB. При ключе (идентификатор узла, время измерения) наблюдается максимум для скорости операции 3 при размере буфера 6,5 Мбайт. При ключе (время измерения, идентификатор узла) скорость операции 3 почти не зависит от размера буфера ядра СУБД. Максимальная скорость записи сигналов в БД, которую демонстрирует экспериментальный образец СИ – 958 тыс. сигналов/секунду или 49,82 Мбайт/с (размер сигнала равен 52 байта), что для соответствует требованиям, предъявляемым к современным SCADA-системам.

Для СУБД с ориентацией на хранение данных на жестких дисках возможности дисковой подсистемы в конечном счете задают пределы быстродействия СУБД. С целью определения максимально возможной производительности СИ, измеряется производительность дисковой подсистемы рабочей станции, на которой установлен СИД.

Рассмотрим случай пиковой нагрузки на СИ в режиме сбора данных. При такой нагрузке объем технологических данных может существенно превышать размер кэша дисковой подсистемы. Поэтому размер тестовых данных в дисковой подсистеме берем больше размера кэша жесткого диска. В соответствии с работой [22], используем паттерн рабочей нагрузки, характерный для БД, используемый для логирования или сбора статистики. В таком случае считается, что на уровне жесткого диска задействована только операция записи, причем в режиме последовательного доступа. В качестве инструмента тестирования производительности дисковой подсистемы используется штатная для ОС Linux утилита fio [23].

На рисунке 3 приведены результаты тестирования дисковой подсистемы в случае, когда имеется один (рис. 3, а) и два (рис. 3, б) независимых потока, работающих с жестким диском одновременно. Показана зависимость скорости записи от размера файла при разных размерах блока. Рассмотрены малые размеры файла, сравнимые с размером кэша жесткого диска, а также большие размеры файла, соответствующие модели пиковой нагрузки. Как видно из рисунка 3, скорость записи при фиксированном размере блока мало изменяется при разных размерах файла, в тоже время зависимость скорости записи от размера блока при фиксированном размере файла может быть существенной.

Считается, что паттерн рабочей нагрузки для дисковой подсистемы, соответствующий базам данных, имеет размер блока 8 или 64 Кбайт [22]. Исходя из этого, верхнюю границу скорости вставки можно оценить как 150 Мбайт/с – это скорость записи жесткого диска при одном пишущем потоке, размере файла 1 Гбайт и размере блока 64 Кбайт. Максимальная скорость записи

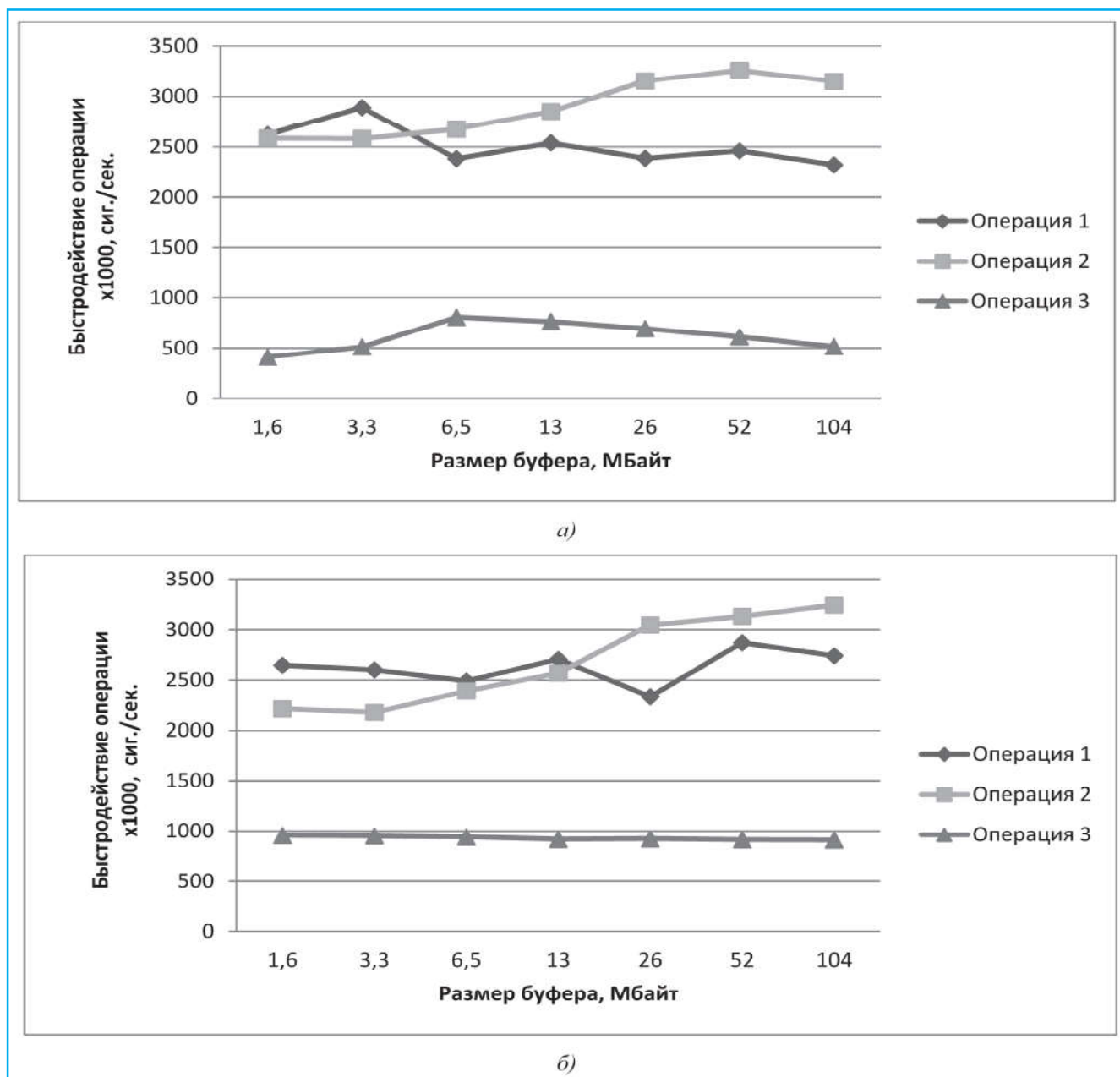


Рис. 2. Результаты тестирования быстродействия операций СИ: а) ключ – идентификатор узла, время измерения; б) ключ – время измерения, идентификатор узла

сигналов, полученная при тестировании экспериментального образца СИ – 49,82 МБайт/с. Таким образом, скорость вставки технологических данных может быть увеличена не более чем в 3 раза.

Заключение

В связи с ростом количества и скорости сбора данных в современных системах промышленной автоматизации повышаются требования к производительности SCADA-систем, в частности к производительности сервера истории как важной составляющей таких систем. В данной работе предложена и апробирована методика экспериментальной оценки различных характеристик

СИ, существенно влияющих на его производительность. На тестовом стенде с помощью модели технологических данных, построенной на основе OPC UA стандарта, оценивается быстродействие ресурсоемких операций СИ.

Экспериментально показано, что элементом архитектуры СИ, который приводит к существенному снижению производительности, является ядро СУБД (подсистема хранения), предоставляющее низкоуровневый интерфейс доступа к данным. Использование хранилища типа «ключ-значение» в качестве ядра СУБД обеспечивает требуемые для современных SCADA-систем показатели – обслуживание технологических процессов с сотнями тысяч контролируемых

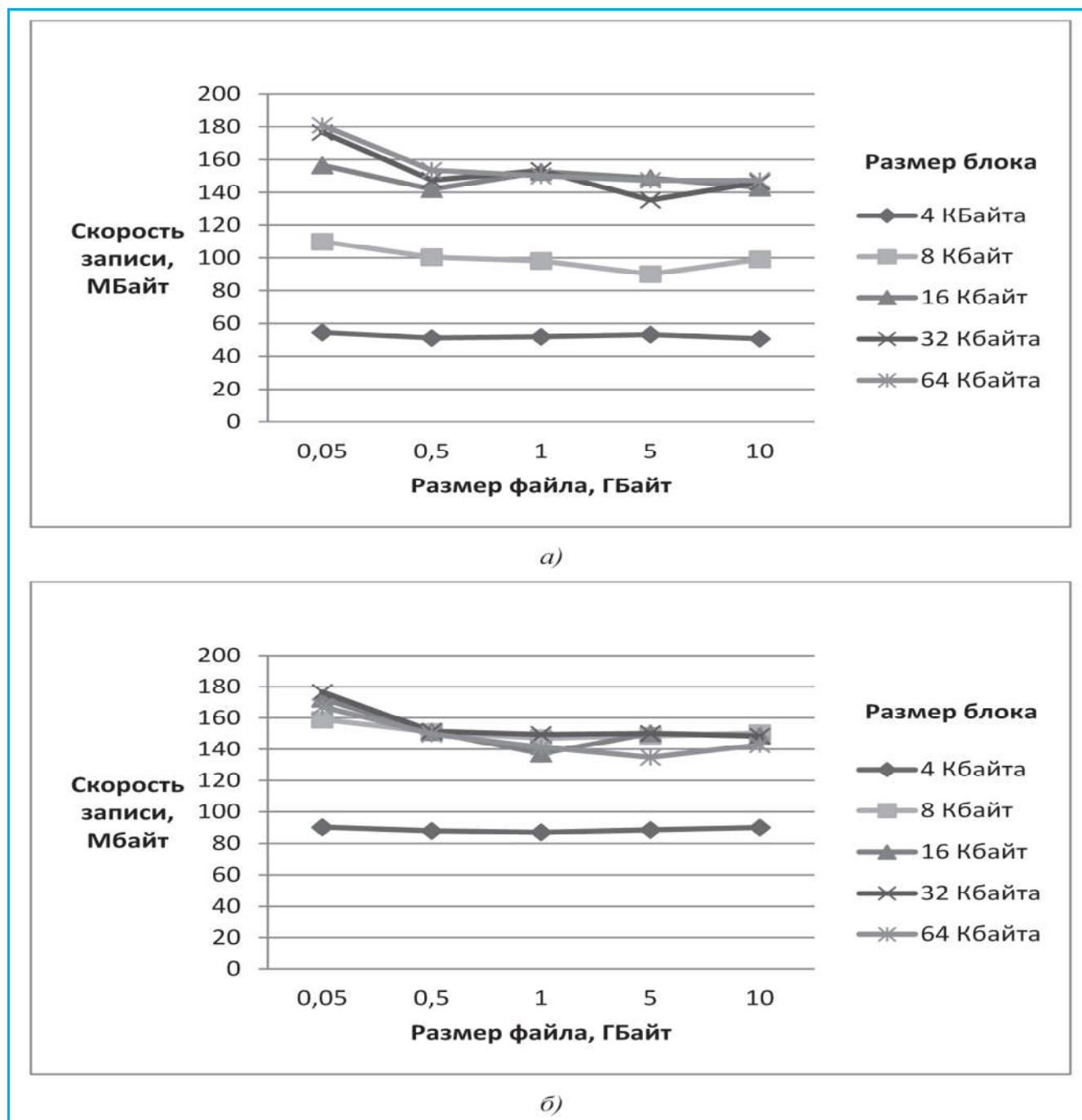


Рис. 3. Скорость записи дисковой подсистемы: а) один пишущий поток; б) два пишущих потока параметров при частоте обновления значений в сотни миллисекунд.

В выполненном исследовании рассмотрены не все возможные рабочие режимы СИ и способы увеличения производительности. Например, на этом этапе исключен из рассмотрения случай обработки аварийных сообщений, не исследованы зависимости рабочих характеристик СИ от способов сжатия технологических данных и пр. Все эти варианты также в различной степени влияют на производительность сервера истории системы диспетчерского управления и сбора данных и будут исследованы в перспективе.

Список литературы

1. Швецов Д.П. Интеллектуальные системы хранения данных в АСУТП // *Современные технологии автоматизации*. 2011, № 4. С. 42–46.
2. Москвитин Д.В. InfinityHistoryServer 3.0. История в деталях // *Промышленные АСУ и контроллеры*. 2008, № 10. С. 31–36.
3. Шевчук Е.В., Шевчук Ю.В. Современные тенденции в области хранения и обработки сенсорных данных // *Программные системы: теория и приложения*. 2015, Т. 6, № 4(27). С. 157–176.

4. Brice Chardin, Jean-Marc Lacombe, Jean-Marc Petit Data historians in the data management landscape // Technology Conference on Performance Evaluation & Benchmarking, in conjunction with VLDB (TPCTC 2012), LNCS, edited by Springer, 2012, pp. 124–139. ISBN 978-3-642-36726-7.
5. Богданов Н., Киселева О. OPC Unified Architecture: изменения в популярной технологии информационных обменов с точки зрения инженера // *Современные технологии автоматизации*. 2010, № 3. С. 82–87.
6. Bernstein P.A., Goodman N., Hadzilacos V. Concurrency Control and Recovery in Database Systems. Addison-Wesley, ISBN 0-201-10715-5, 1986.
7. OPC Foundation: OPC Unified Architecture Specification Part 5: Information model – Release 1.03. July 19. 2015, 107 p.
8. OPC Foundation: OPC Unified Architecture Specification Part 4: Services – Release 1.03. July 19. 2015, 171 p.
9. Кузнецов С.Д., Посконин А. В. Системы управления данными категории NoSQL // *Программирование*. 2014, № 6. С. 34–47.
10. The benchmark handbook for database and transaction processing systems, Jim Gray (Ed.), Morgan Kaufmann, 2nd Ed. 1993. P. 592, ISBN 1-55860-292-5.
11. Best key-value stores. *Электронный ресурс*: <https://www.g2crowd.com/categories/key-value-stores> (дата обращения: 20.07.2017).
12. LevelDB benchmarks. *Электронный ресурс*: <http://www.lmdb.tech/bench/microbench/benchmark.html> (дата обращения: 20.07.2017).
13. Benchmarking LevelDB vs. RocksDB vs. HyperLevelDB vs. LMDB performance for InfluxDB. *Электронный ресурс*: <https://www.influxdata.com/benchmarking-leveldb-vs-rocksdb-vs-hyperleveldb-vs-lmdb-performance-for-influxdb/> (дата обращения: 20.07.2017).
14. Performance data for LevelDB, Berkley DB and BangDB for random operations. *Электронный ресурс*: <http://highscalability.com/blog/2012/11/29/performance-data-for-leveldb-berkeley-db-and-bangdb-for-rando.html> (дата обращения: 20.07.2017).
15. HyperLevelDB performance benchmarks. *Электронный ресурс*: <http://hyperdex.org/performance/leveldb/> (дата обращения: 20.07.2017).
16. BangDB vs LevelDB – performance comparison. *Электронный ресурс*: <http://www.iqlect.com/blog/2016/07/12/bangdb-vs-leveldb-performance-comparison/> (дата обращения: 20.07.2017).
17. LevelDB. *Электронный ресурс*: <http://leveldb.org/> (дата обращения: 20.07.2017).
18. HyperLevelDB. *Электронный ресурс*: <https://github.com/rescrv/HyperLevelDB> (дата обращения: 20.07.2017).
19. Sophia. *Электронный ресурс*: <http://sophia.systems/> (дата обращения: 20.07.2017).
20. BangDB. *Электронный ресурс*: <http://bangdb.com/> (дата обращения: 20.07.2017).
21. SQLite. *Электронный ресурс*: <https://www.sqlite.org/> (дата обращения: 20.07.2017).
22. White paper: Fujitsu PRIMERGY server. Basics of disk i/o performance. Fujitsu. 2011. 15 p.
23. Inspecting disk IO performance with fio. *Электронный ресурс*: <https://www.linux.com/learn/inspecting-disk-io-performance-fio> (дата обращения: 20.07.2017).

References

1. Shvetsov D.P. Intellectualnye sistemy khraneniya dannykh v ASUTP [Intelligent storage systems in the automated control system of technological processes]. *Sovremennye tekhnologii avtomatizatsii* [Modern automation technologies]. 2011, no. 4, pp. 42–46.
2. Moskvitin D.V. InfinityHistoryServer 3.0. Istoriya v detalyakh [InfinityHistoryServer 3.0. History in Detail]. *Promyshlennyye ASU i kontrolyery* [Industrial Automatic Control Systems and Controllers]. 2008, no. 10, pp. 31–36.
3. Shevchuk Ye.V., Shevchuk Yu.V. Sovremennyye tendentsii v oblasti khraneniya i obrabotki sensornykh dannykh [Current trends in the storage and processing of sensory data]. *Programmnyye sistemy: teoriya i prilozheniya* [Program systems: theory and applications]. 2015. Vol. 6, no. 4(27), pp. 157–176.
4. Brice Chardin, Jean-Marc Lacombe, Jean-Marc Petit Data historians in the data management landscape // Technology Conference on Performance Evaluation & Benchmarking, in conjunction with VLDB (TPCTC 2012), LNCS, edited by Springer, 2012, pp. 124–139. ISBN 978-3-642-36726-7.
5. Bogdanov N., Kiseleva O. OPC Unified Architecture: izmeneniya v populyarnoy tekhnologii informatsionnykh obmenov s tochki zreniya inzhenera [OPC Unified Architecture: changes in the popular technology of information exchange from the engineer's point of view]. *Sovremennyye tekhnologii avtomatizatsii* [Modern automation technologies]. 2010, no. 3, pp. 82–87.
6. Bernstein P.A., Goodman N., Hadzilacos V. Concurrency Control and Recovery in Database Systems. Addison-Wesley, ISBN 0-201-10715-5, 1986.
7. OPC Foundation: OPC Unified Architecture Specification Part 5: Information model – Release 1.03. July 19. 2015, 107 p.
8. OPC Foundation: OPC Unified Architecture Specification Part 4: Services – Release 1.03. July 19. 2015, 171 p.
9. Kuznetsov S.D., Poskonin A. V. Sistemy upravleniya dannymi kategorii NoSQL [Data management systems of the NoSQL category]. *Programmirovaniye* [Programming]. 2014, no. 6, pp. 34–47.
10. The benchmark handbook for database and transaction processing systems, Jim Gray (Ed.), Morgan Kaufmann, 2nd Ed. 1993. P. 592, ISBN 1-55860-292-5.

11. Best key-value stores. URL: <https://www.g2crowd.com/categories/key-value-stores> (date of the application: 20.07.2017).
12. LevelDB benchmarks. URL: <http://www.lmdb.tech/bench/microbench/benchmark.html> (date of the application: 20.07.2017).
13. Benchmarking LevelDB vs. RocksDB vs. HyperLevelDB vs. LMDB performance for InfluxDB. URL: <https://www.influxdata.com/benchmarking-leveldb-vs-rocksdb-vs-hyperleveldb-vs-lmdb-performance-for-influxdb/> (date of the application: 20.07.2017).
14. Performance data for LevelDB, Berkley DB and BangDB for random operations. URL: <http://highscalability.com/blog/2012/11/29/performance-data-for-leveldb-berkeley-db-and-bangdb-for-rando.html> (date of the application: 20.07.2017).
15. HyperLevelDB performance benchmarks. URL: <http://hyperdex.org/performance/leveldb/> (date of the application: 20.07.2017).
16. BangDB vs LevelDB – performance comparison. URL: <http://www.iqlect.com/blog/2016/07/12/bangdb-vs-leveldb-performance-comparison/> (date of the application: 20.07.2017).
17. LevelDB. URL: <http://leveldb.org/> (date of the application: 20.07.2017).
18. HyperLevelDB. URL: <https://github.com/rescrv/HyperLevelDB> (date of the application: 20.07.2017).
19. Sophia. URL: <http://sophia.systems/> (date of the application: 20.07.2017).
20. BangDB. URL: <http://bangdb.com/> (date of the application: 20.07.2017).
21. SQLite. URL: <https://www.sqlite.org/> (date of the application: 20.07.2017).
22. White paper: Fujitsu *PRIMERGY* server. Basics of disk i/o performance. Fujitsu. 2011. 15 p.
23. Inspecting disk IO performance with fio. URL: <https://www.linux.com/learn/inspecting-disk-io-performance-fio> (date of the application: 20.07.2017).

Информация об авторах

Замятин Александр Владимирович, доктор технических наук, доцент

E-mail: avzamyatin@inbox.ru

Острасть Павел Михайлович, кандидат технических наук

E-mail: op@sibmail.com

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет»

634050, Российская Федерация, г. Томск, пр. Ленина, 36

Телицын Евгений Александрович, инженер, начальник отдела

Акционерное общество «ЭлеСи»

634021, Российская Федерация, г. Томск, ул. Алтайская, 161А

Тренькаев Вадим Николаевич, кандидат технических наук, доцент, ведущий инженер учебно-научной лаборатории программно-аппаратных комплексов автоматизации Научно-образовательного центра компьютерных наук и технологий

E-mail: tvnik@sibmail.com

Яновский Владимир Дмитриевич, программист

E-mail: waldemar.yan@gmail.com

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет»

634050, Российская Федерация, г. Томск, пр. Ленина, 36

Information about the authors

Zamyatin Aleksandr Vladimirovich, Doctor of Technical Sciences, Associate Professor

E-mail: avzamyatin@inbox.ru

Ostrast Pavel Mikhaylovich, Candidate of Technical Sciences

E-mail: op@sibmail.com

Federal State Autonomous Educational Institution of Higher Education «National Research Tomsk State University»

634050, Russian Federation, Tomsk, Lenin Avenue, 36

Telitsin Yevgeny Aleksandrovich, Engineer, Head of Department

Joint Stock Company «EleCy»

634021, Russian Federation, Tomsk, str. Altay, 161A

Trenkaev Vadim Nikolaevich, Candidate of Technical Sciences, Associate Professor, Leading engineer of the educational and scientific laboratory of software and hardware automation complexes of the Scientific and Educational Center of Computer Science and Technology

E-mail: tvnik@sibmail.com

Janowsky Vladimir Dmitrievich, Programmer

E-mail: waldemar.yan@gmail.com

Federal State Autonomous Educational Institution of Higher Education «National Research Tomsk State University»

634050, Russian Federation, Tomsk, Lenin Avenue, 36