УДК 004.056.3

П. Д. Осмоловский, С. А. Романенко
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Отказоустойчивость в компонентах системы видеонаблюдения

Представлена проблема снижения времени недоступности системы видеонаблюдения для охраны объектов повышенной опасности. Проведены анализ и формализация требований к решению, а также обзор инструментов, которые могут применяться в решении для высокой доступности сервера видеонаблюдения. В результате обзора проведено сравнение по ключевым критериям и определен оптимальный набор технологий для разработки. Предложено решение для повышения отказоустойчивости за счет высокой доступности сервера с использованием резервного узла. Определены специфичные требования, ограничения и допущения, в условиях которых необходимо обеспечить низкую частоту сбоев для системы высокой ответственности, производящей сбор, обработку и хранение видеоизображений с объектов высокой опасности. Разработана функциональная схема, которая показывает состав системы и взаимодействие ее элементов. На основе функционального взаимодействия было проведено проектирование структуры решения и поведения его компонентов при работе комплекса. В результате была разработана архитектура в высокой доступности сервера видеонаблюдения для охраны объектов повышенной опасности и определено поведение его функциональных компонентов.

Менеджер высокой доступности, отказоустойчивость, надежность работы, резервное копирование данных, восстановление данных

В области разработки современной программной инфраструктуры вопрос резервирования ресурсов и обеспечения высокой доступности стоит чрезвычайно остро. Повышение нагрузки и интенсивности работы требует реализации все более совершенных механизмов.

Система видеонаблюдения для охраны объектов повышенной опасности – это высоконагруженный комплекс, который должен обеспечивать бесперебойный сбор, обработку и отображение информации. Данный программный комплекс относится к числу систем повышенной ответственности и применяется на объектах с высоким уровнем опасности. Ключевым компонентом системы является сервер, который состоит из нескольких модулей и имеет единую точку входа. Для обеспечения надежности системы необходима ее работа в режиме высокой доступности, при этом специфика некоторых низкоуровневых датчиков не позволяет им взаимодействовать более чем с одним контроллером одновременно. Данные, с которыми работают пользователи, хранятся в базе данных (БД) под управлением PostgreSQL. Требования заказчика предписывают обеспечивать сохранность данных, но допускают потери изменений, сделанных при внештатных ситуациях. Инфраструктура системы должна быть организована таким образом, чтобы обеспечить заявленную надежность работы без помощи системных администраторов. Для обеспечения требований к надежности система будет использовать кластер из двух серверов: основного для штатных ситуаций и резервного для аварийных.

Актуальна комплексная задача по резервированию данных и поддержке высокой доступности системы с учетом указанных ограничений. Ключевыми показателями решения служат надежность и простота поддержки.

Обзор решений. Кластеры высокой доступности (КВД) можно квалифицировать как гарантирующие непрерывную или высокую доступность. Отличие заключается в том, что гарантия непре-

рывности работы требует огромных затрат, оправданных в системах реального времени, где потеря каждого кванта времени влечет за собой утрату контроля и потерю данных. Высокая доступность является наиболее удачным решением для большинства систем, где требуется кластеризация.

Задачу высокой доступности следует воспринимать как связку компонента внешней информационной доступности системы и компонента обеспечения целостности данных.

Для начальных решений важна простота развертывания и поддержки, которые могут быть обеспечены набором из технологий Heartbeat [1] и DRBD (Distributed Replicated Block Device – распределенное реплицируемое блочное устройство) [2]. Такая связка применима в задачах поддержки кластера из пары серверов. Алгоритм работы Active/Passive позволяет быть запущенным одному серверу, в то время как второй готов включиться в случае сбоя. Heartbeat – это служба, запущенная на отдельной от кластера машине, которая предоставляет внешним клиентам IP-адрес для взаимодействия с системой. В текстовый файл с базой доменных имен (/etc/hosts в случае Linux) обеих машин кластера помещаются адреса и имена серверов в следующем виде:

10.0.0.101 active.yourserver.com

10.0.0.102 passive.yourserver.com

Для корректной работы DRBD на каждой машине кластера конфигурируется раздел жесткого диска одинакового объема – это пространство будет «реплицируемым». Конфигурирование DRBD осуществляется с помощью встроенного инструмента drbdadm [3]. Схема работы DRBD отображена на рис. 1.

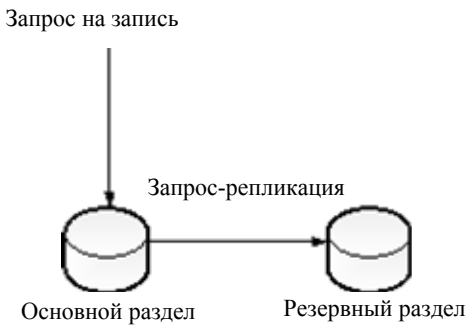

Рис. 1

На рис. 2 представлена схема работы данной связки технологий с указанием физических адресов машин.

Heartbeat – уже устаревшее решение, которое не поддерживается разработчиками. Его наследником с более гибкими и развитыми инструментами стала технология Corosync. Она входит в состав Pacemaker [4] и является эволюционным развитием стека Linux–HighAvailability. Эта связка достаточно хорошо себя зарекомендовала в качестве актуального решения с широкой сферой применения [5], [6]. Компонентом для связи служит Corosync, который решает следующие задачи:

– отслеживание статуса приложений;

– оповещение приложения о смене активного узла в кластере;

– отправка идентичных сообщений процессам на всех узлах;

– предоставление доступа к общей базе данных с конфигурацией и статистикой;

– отправка уведомлений об изменениях, произведенных в базе.

Pacemaker является менеджером ресурсов кластера [7]. Он:

1. Ищет и устраняет сбои на уровне узлов и служб.

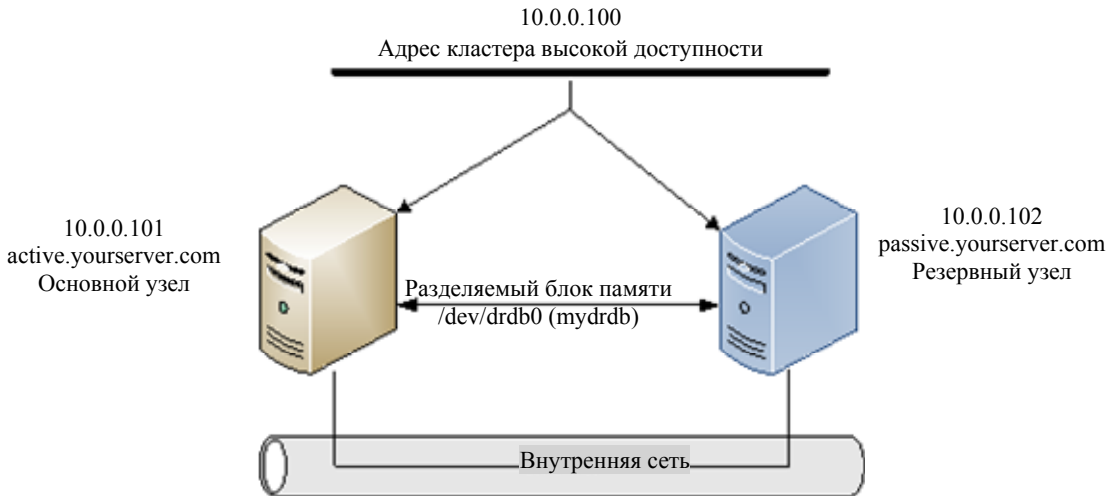

Рис. 2

2. Не зависит от подсистемы хранения.

3. Допускает использование практически любых источников ресурсов.

4. Поддерживает STONITH (Shoot-the-Other-Node-in-the-Head), т. е. недоступный узел не получает сообщения до тех пор, пока не обратится с обновлением своего статуса – опция доступна в случае подключения более двух машин в кластер.

5. Поддерживает ресурсозависимые кластеры любого размера.

6. Распространяет настройку узлов по заданной логике, что дает возможность управления из единой точки.

7. Позволяет контролировать порядок запуска ресурсов и задавать правила совместной работы.

8. Поддерживает расширенные типы ресурсов: клоны (когда ресурс запущен на множестве узлов) и дополнительные состояния (master/slave), что актуально для таких систем управления БД (СУБД), как MySQL, MariaDB, PostgreSQL, Oracle.

9. Имеет единую кластерную оболочку CRM с поддержкой запуска скриптов.

На рис. 3 отображена схема работы Active/Passive для связки Linux–HighAvailability.

Описанные решения распространяются под лицензиями «GNU General Public License version 2» и «BSD License 2.0», что позволяет использовать разработанное программное обеспечение в коммерческих целях.

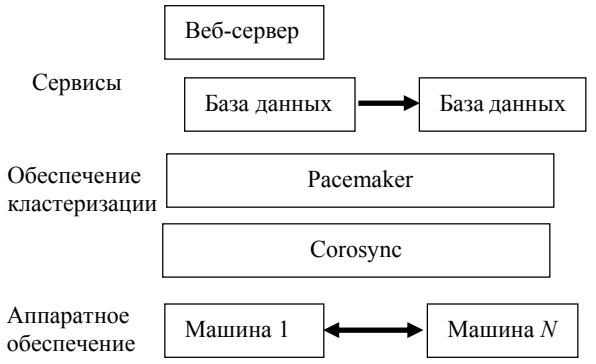

Рис. 3

Для задачи репликации данных следует отметить встроенные инструменты СУБД PostgreSQL. В них доступна потоковая физическая репликация данных, когда при приеме запросов на добавление, изменение и удаление данных основной сервер БД формирует и отправляет записи журнала (Write-Ahead Logs – WAL) на все подключенные для репликации серверы в виде конкретных изменений файлов с данными. Алгоритм работы описан на рис. 4.

Другой тип потоковой репликации – логический, при котором изменения передаются в более абстрактном формате SQL-запросов, выполняемых затем на стороне приема. Для данного решения требуется посредник – контроллер, который определяет правила и таблицы для синхронизации. Такое решение позволяет организовать сеть серверов БД для агрегации данных и использовать различные форматы их хранения. Такой вид репликации изображен на рис. 5.

Инструменты СУБД содержат еще одно решение – потоковую параллельную репликацию. Оно реализовано инструментом Pgpool и является удаленным узлом балансировки [9]. Для приложений это – наиболее прозрачный метод с одним прозрачным подключением к пулу серверов БД. Контроллер Pgpool активируется на удаленной машине, которая соединяется с машинами кластера через сетевое соединение. Допускается запуск Pgpool на машине с основным узлом кластера в случаях, когда разработчик уверен в аппаратной отказоустойчивости, – таким образом осуществляется защита от сбоев программного обеспечения. В конфигурации прописываются роли машин по их адресу. Pgpool берет на себя перенаправление запросов подключенным серверам.

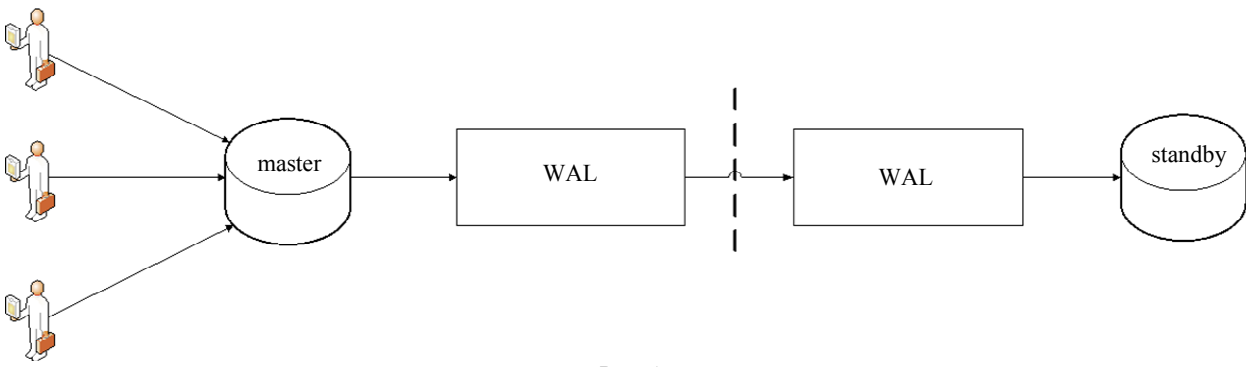

Рис. 4

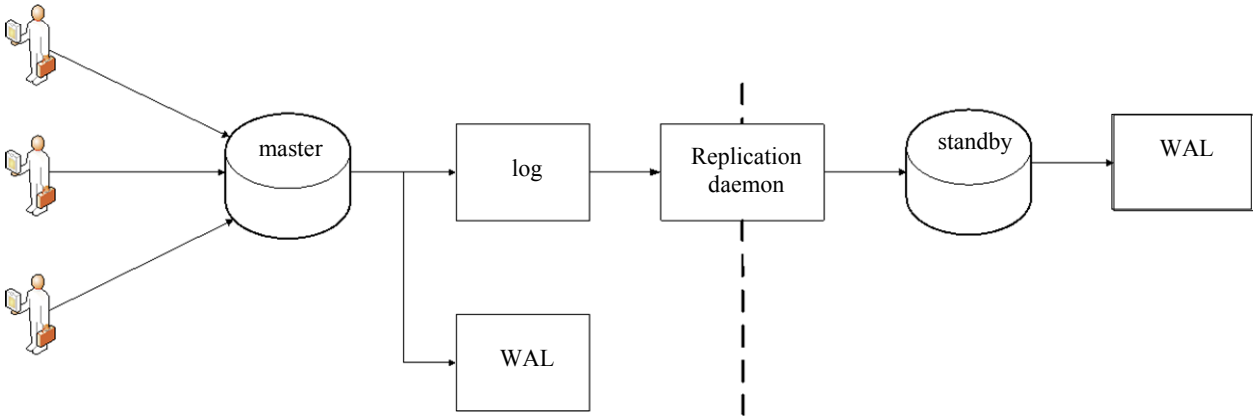

Рис. 5

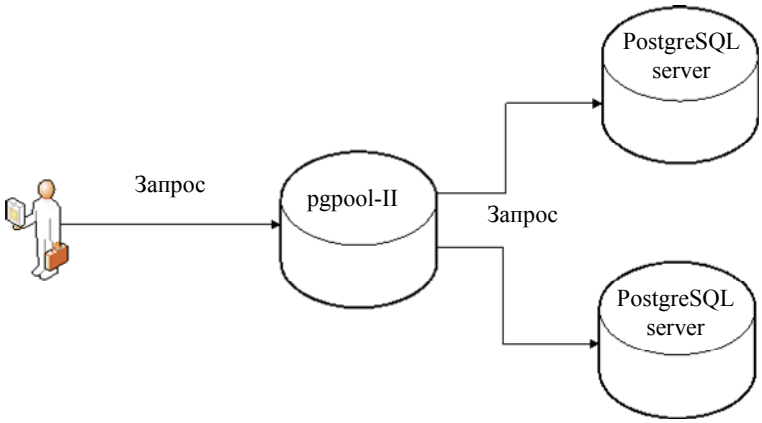

Рис. 6

Решение содержит множество вариантов использования и в вопросе репликации служит аналогом логической репликации. Устройство Pgpool описано на рис. 6.

СУБД PostgreSQL содержит инструменты для копирования данных в бинарный вид и их восстановление. Связка данных команд может быть использована в специфичном решении, где требуется периодичная физическая репликация данных.

Оба вида репликации и бинарное копирование-восстановление данных требуют связки с инструментами обнаружения сбоев. При этом Pgpool допускает отдельное использование, так как включает в себя механизмы мониторинга состояния кластера БД.

Для сопоставления решения были выделены следующие критерии:

1) расширяемость – поддержка большего числа машин в кластере;

2) гибкость – возможность настройки поведения в случае отключения одной из машин;

3) поддержка – наличие активного сообщества, обновлений от разработчиков;

4) ресурсы – отсутствие необходимости использования дополнительных машин для конфигурирования и корректной работы.

Оценка решений предусматривает выявление хорошо проработанной функциональной возможности из критерия. Сравнение представлено в табл. 1.

Выбор решения. По требованию заказчика все использованные инструменты должны пройти сертификацию и быть лицензированы для использования в конечном решении. При этом

Таблица 1

Решения	Критерии оценки			
	Расширяемость	Гибкость	Поддержка	Ресурсы
Corosync и Pacemaker	+	+	+	–
Heartbeat и DRBD	+	+	–	+
Pgpool	–	–	+	+
Удаленное копирование, загрузка дампов и репликация	+	–	+	+

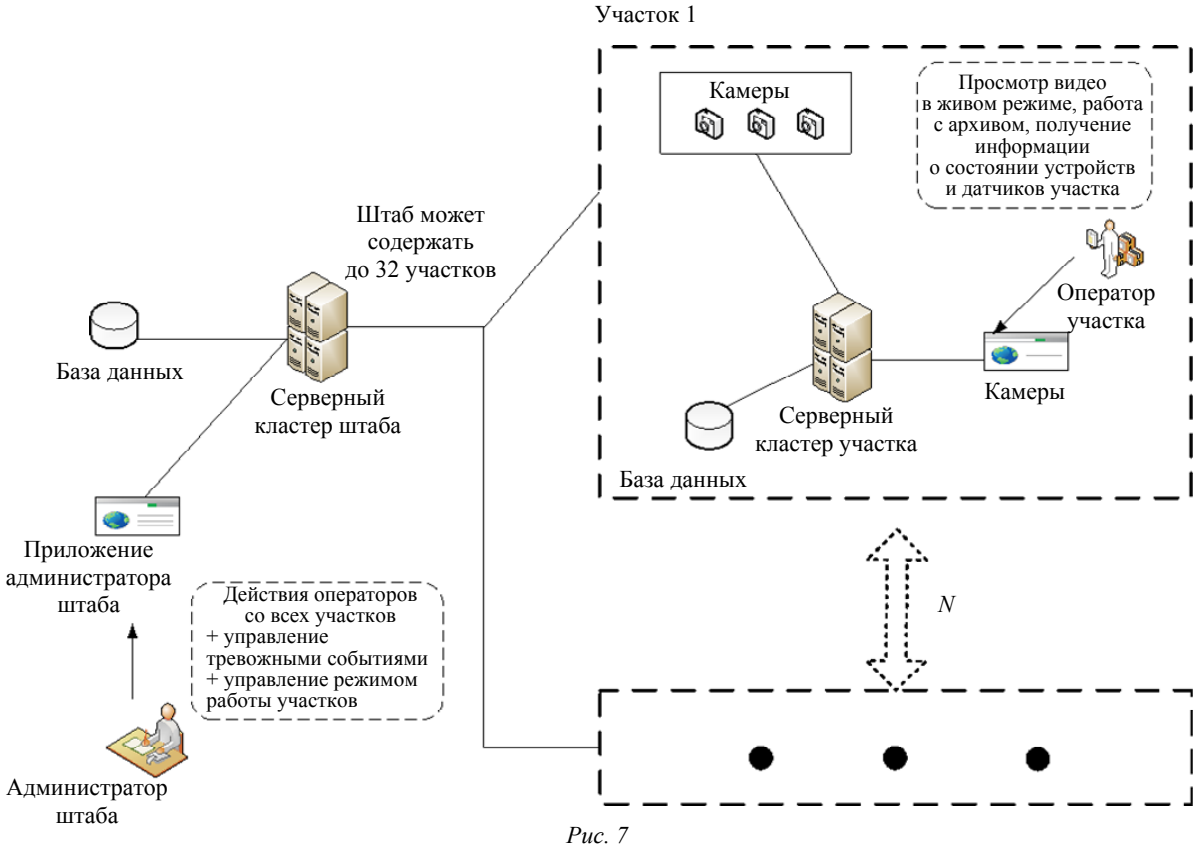

Рис. 7

СУБД для PostgreSQL необходимые процедуры уже пройдены. Также существуют специальные требования к документированию решений и исходных кодов программ, что накладывает серьезные ограничения на внедрение достаточно удачных связок технологий, описанных в обзоре решений. С учетом того что задача весьма специфична и поведение при сбоях заранее определено и не будет изменяться, было принято решение о разработке собственного инструмента. Ключевой задачей является обеспечение режима высокого доступа при работе с кластером серверов, создание резервных копий и их загрузка на резервном узле в случае сбоя, гибкая настройка детектора сбоя и возможность отключения части компонентов. Решение должно быть настраиваемым с помощью файлов конфигурации и иметь возможность автоматизированного восстановления без вмешательства сотрудников.

TCP (Transmission Control Protocol) – один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных. При разработке систем повышенной ответственности наиболее надежным механизмом мониторинга является периодический опрос с помощью TCP-сообщений.

СУБД – это комплекс программно-языковых средств, позволяющих создавать базы данных и управлять данными. Для резервирования данных наиболее надежны разовые запросы копирования и загрузки инструментами СУБД PostgreSQL.

Описание решения. При проектировании решения была разработана функциональная схема системы (рис. 7).

При интеграции инструментов кластеризации каждый из серверов расширяется с помощью утилиты, взаимодействующей с отдельным приложением для управления режимом работы. Данная структура представлена на рис. 8.

Для реализации решения используется два компонента сервера – принятия решений и обеспечения высокой доступности – и отдельный модуль высокой доступности. В составе модуля 3 компонента: мониторинга основного узла, мониторинга резервного узла и управления состоянием сервера. Модуль определяет режим работы сервера на основе данных, получаемых от компонента обеспечения высокой доступности. Решение имеет два сценария работы (управление режимом работы сервера осуществляется через вызов системных команд в терминале операционной системы). В случае запуска на основном узле задача заключается в обеспечении его запущенности. Для модуля резервного узла необходима также согласованность работы основного и резервного серверов – их одновременную работу нельзя до-

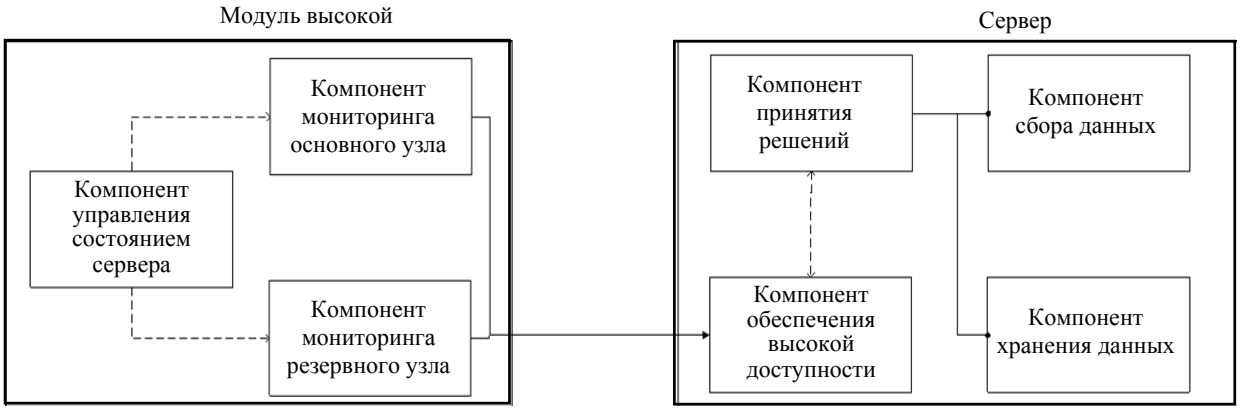

Рис. 8

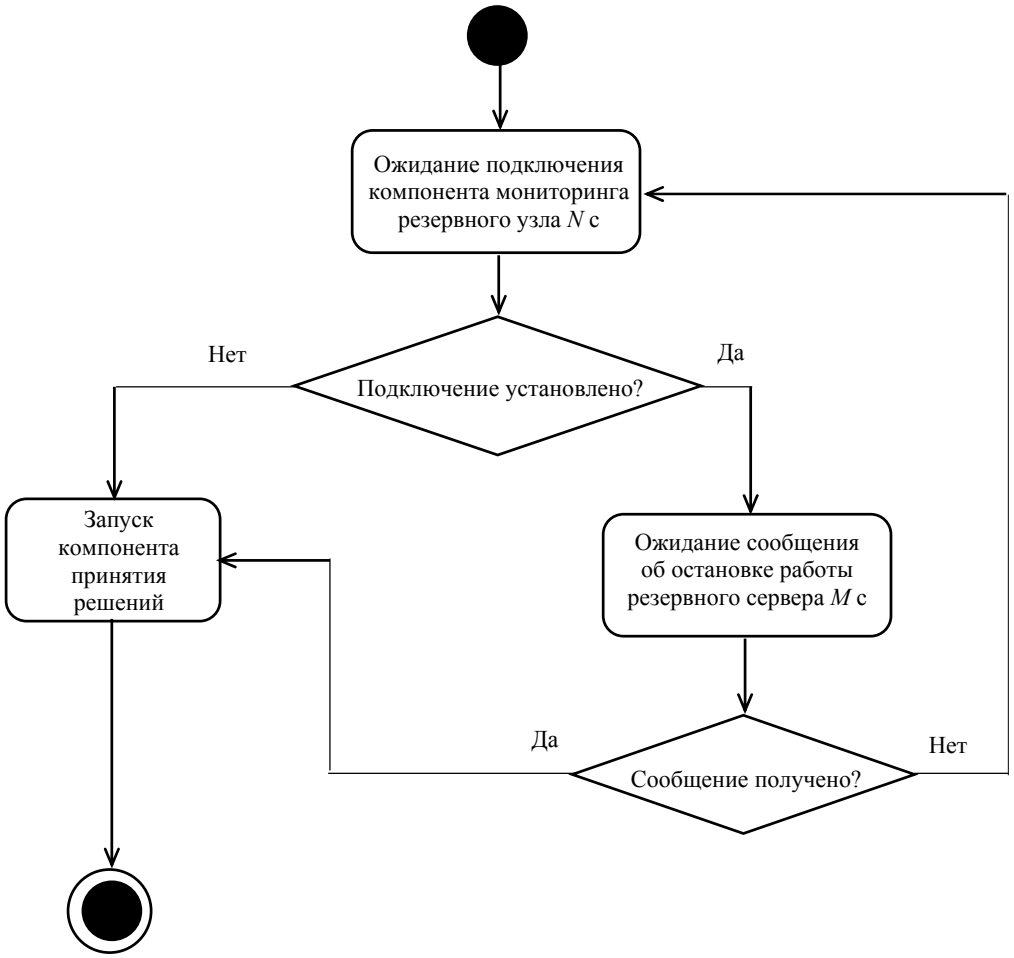

Рис. 9

пускать, так как это повлечет искажение данных, получаемых с датчиков охраны. Алгоритм показан на блок-схеме активности (рис. 9).

При переключении активного узла на резервный компонент управления состоянием сервера производит незамедлительный запуск сервера для обеспечения доступности. На рис. 10 представлена блок-схема активности этого компонента при запуске на резервном узле.

Взаимодействие с компонентом управления состоянием сервера осуществляется при помощи компонентов мониторинга. Полученная информация обрабатывается, и компонент управления состоянием сервера принимает решения для поддержки работы активного узла. Механизм отслеживания работы узла представлен на блок-схеме активности (рис. 11).

Доступ к базе данных основного узла осуществляется при помощи конфигурации механизма дозволенного удаленного доступа по сетевому адресу.

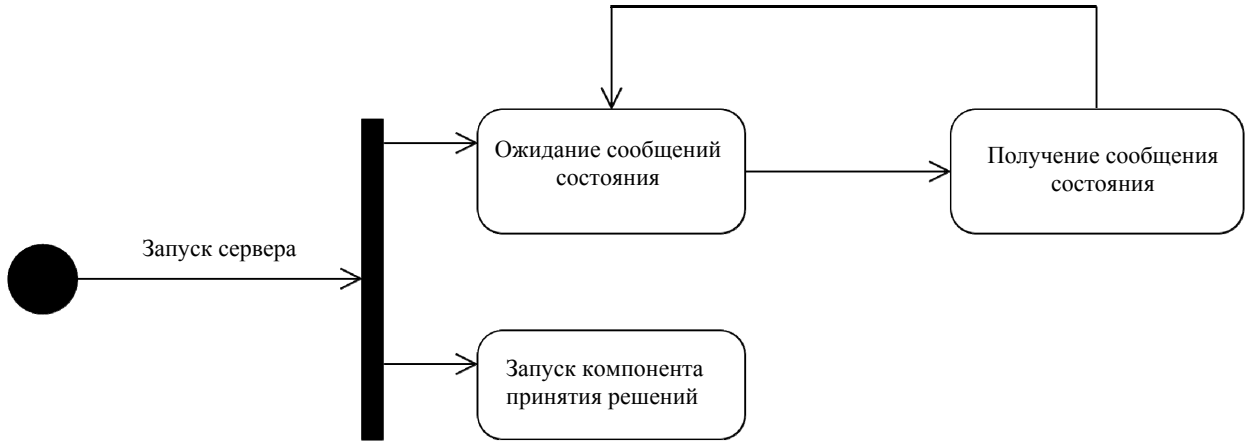

Рис. 10

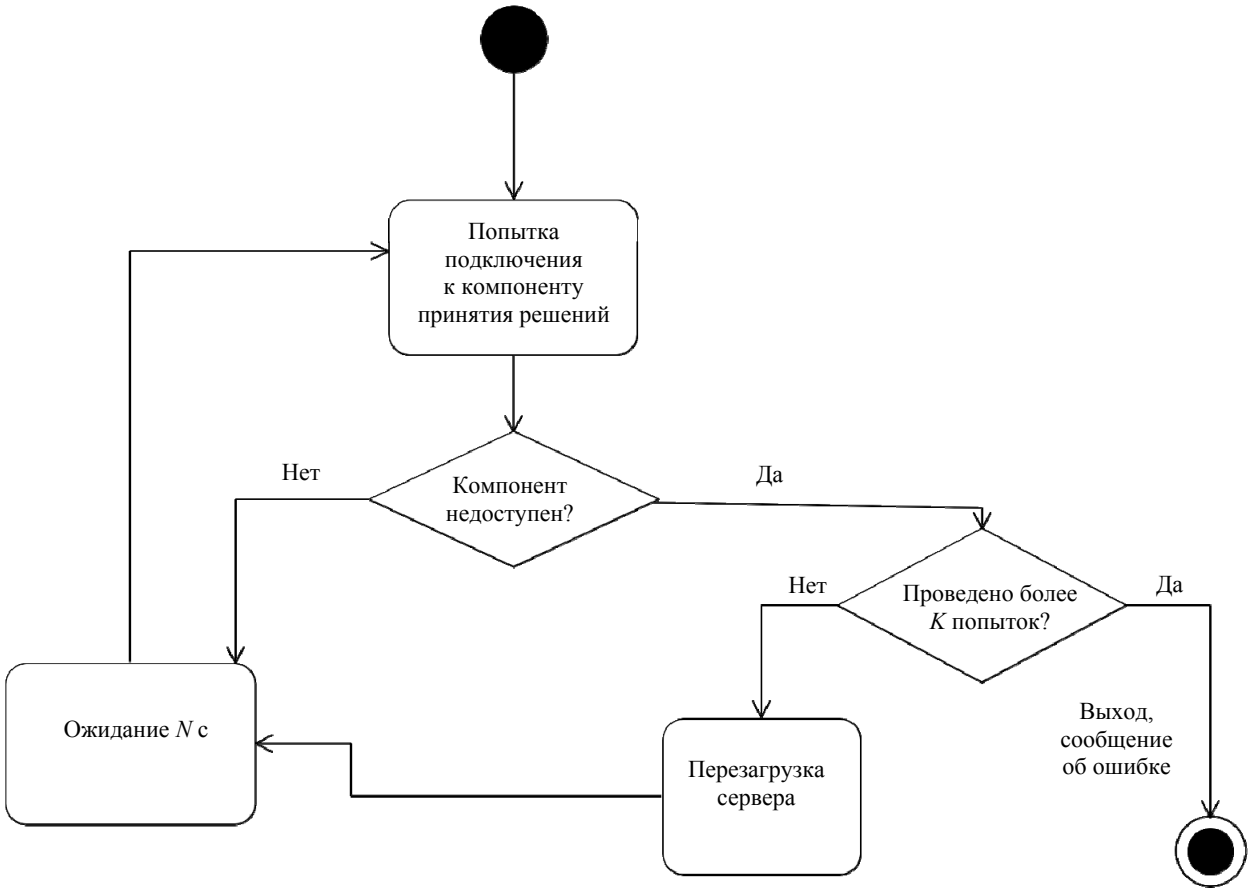

Рис. 11

Таблица 2

Содержимое	Значение
Ping	Сообщение о корректном режиме работы компонента принятия решений основного узла и его запрос
Slave-stopped	Сообщения от компонента управления состоянием сервера резервного узла. Уведомляет о текущем режиме работы компонента принятия решений резервного узла и запрашивает состояние основного сервера
Slave-running	

Для мониторинга передаются сообщения в формате, описанном в табл. 2.

На резервном узле компонент управления состоянием сервера проводит глобальный мониторинг обоих компонентов принятия решений, в зависимости от результата которого принимает решение о текущем режиме работы. Также в его задачи входит резервирование данных во время штатной работы основного сервера, загрузка актуального слепка в БД перед запуском резервного

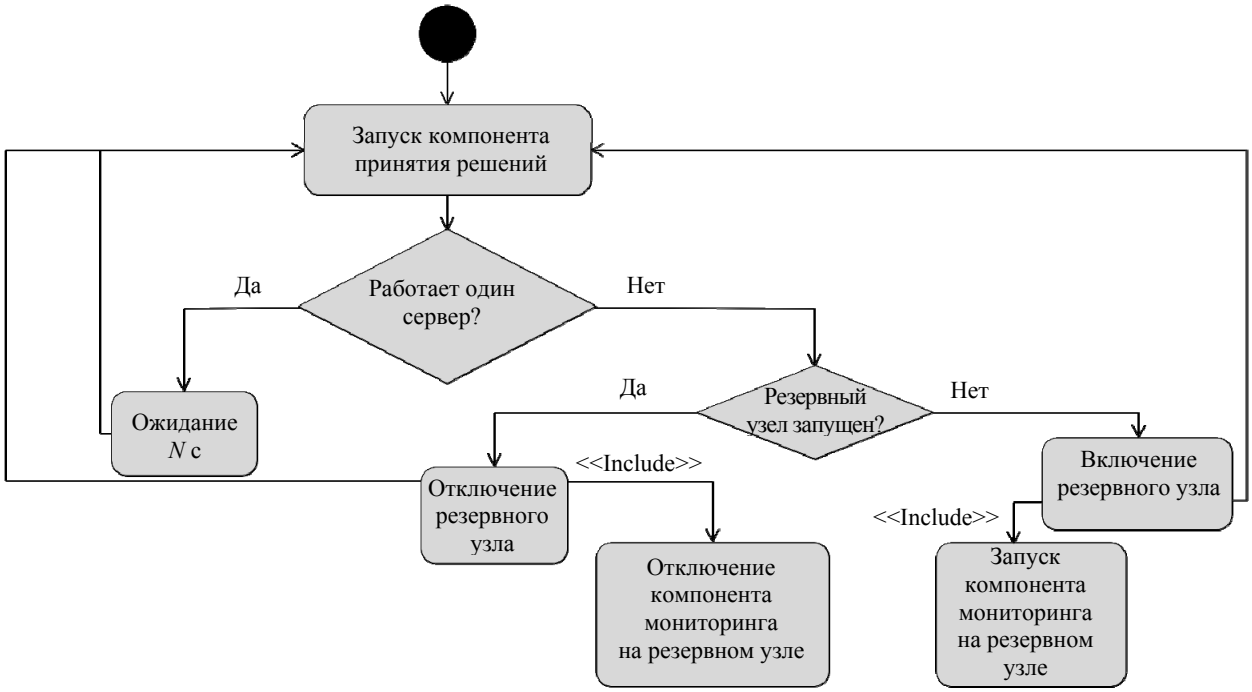

Рис. 12

и основного компонентов принятия решений. Данные возможности реализованы с помощью вызовов команд СУБД PostgreSQL. Поведение программы описано на рис. 12.

Проверка доступности осуществляется с помощью отправки сообщения и проверки, доставлено ли оно. В случае если попытка окончилась неудачей, сервер считается недоступным. После проверки обоих серверов результат каждой из проверок становится аргументом логического оператора «Исключающий ИЛИ».

Результатом проведенной работы стала оценка инструментов кластеризации и позиционирование их применения. Наиболее универсальные механизмы формируют связки Corosync–Pacemaker и DRBD–Heartbeat. Они дают возможность организовать быстрое решение без учета специфики сценариев использования и архитектуры приложения. Эффективность таких решений определяется конфигурацией механизмов обнаружения сбоя и восстановления данных, а также сценариями использования кластера. В условиях разработки систем повышенной ответственности наиболее важен аспект контроля работы встроенных инструментов и их доступной поддержки. В результате исследования инструментов для выбора систем решения был разработан собственный инструмент, адаптированный к специфике проекта и выполняющий требования заказчика.

СПИСОК ЛИТЕРАТУРЫ

1. Руководство по использованию Heartbeat. URL: http://www.linux-ha.org/doc/users-guide/users-guide.html (дата обращения 11.01.2020).

2. Distributed Replicated Block Device. URL: https://en.wikipedia.org/wiki/Distributed_Replicated_Block_Device (дата обращения 11.01.2020).

3. DRBD. Описание и пример использования. URL: https://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/кластеризация/drbd (дата обращения 14.03.2020).

4. Описание Pacemaker. URL: https://wiki.clusterlabs.org/wiki/Pacemaker (дата обращения 14.03.2020).

5. Пример использования связки Corosync-Pacemaker. URL: https://xakep.ru/2019/01/24/corosync-pacemaker/#toc04 (дата обращения 14.03.2020).

6. Описание механизма работы Corosync-Pacemaker. URL: https://habr.com/ru/post/107837/ (дата обращения 14.03.2020).

7. Руководство по использованию Pacemaker. URL: https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/pdf/Clusters_from_Scratch/Pacemaker-1.1-Clusters_from_Scratch-en-US.pdf (дата обращения 14.03.2020).

8. PgPool-II. Описание. URL: https://www.pgpool.net (дата обращения 14.03.2020).

P. D. Osmolovsky, S. A. Romanenko
Saint Petersburg Electrotechnical University

FAULT-TOLERANCE IN CLOSED CIRCUIT TV PROGRAM COMPONENTS

Presents the problem of reducing the time of inaccessibility of a video surveillance system for the protection of high-risk facilities. The analysis and formalization of requirements for the solution. A review of the tools to be used in the solution for high availability of the video surveillance server was conducted. As a result of the review, a comparison was made according to key criteria and the optimal set of technologies for development was determined. A solution to improve fault tolerance through the development of a server high availability solution using a backup node is proposed. For the solution, specific requirements, restrictions and assumptions are identified, under the conditions of which it is necessary to ensure a low failure rate for a high responsibility system that collects, processes and stores video images from high-risk objects. A functional diagram has been developed that shows the composition of the system and the interaction of its elements. Based on the functional interaction, the design of the solution structure and the behavior of its components during the operation of the complex was carried out. As a result, the architecture of a high-availability solution for a video surveillance server to protect high-risk objects was developed and the behavior of its functional components was determined.

High availability manager, fault-tolerance, reliability of operation, database back-up, data restore

УДК 681.5; 622.691.4.053

Д. Х. Имаев, С. В. Квашнин, М. Ю. Шестопалов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Моделирование технологического процесса компримирования природного газа как объекта управления. Установившиеся режимы

Обсуждаются вопросы моделирования управляемого технологического процесса компримирования природного газа в установившихся режимах. Моделирование представляется как последовательное раскрытие неопределенности причинно-следственной топологии, структуры и параметров объекта, среды его функционирования. Характеристики центробежных нагнетателей предложено аппроксимировать искусственными нейронными сетями. На примере параметрической идентификации компьютерной модели компрессорной линии по архивным данным делается вывод о целесообразности сочетания графических и численных методов анализа установившихся режимов компримирования. Близость результатов модельного анализа к архивным данным реального компрессорного цеха магистрального газопровода позволяет заключить, что предлагаемая методология моделирования, идентификации и анализа установившихся режимов компримирования природного газа имеет перспективы дальнейшего развития и практического применения.

Природный газ, компримирование, установившийся режим, центробежный нагнетатель, управление, модель, газодинамическая характеристика, нейронная сеть, компьютерная имитация, идентификация

Автоматическое управление газоперекачивающими агрегатами и компрессорными цехами магистральных газопроводов способно ослаблять нежелательное влияние среды на параметры перекачиваемого газа без необходимости изменения самой технологии компримирования и конструкции агрегатов. Это достигается внешними по отношению к объекту информационно-алгоритмическими средствами, которые сводятся к получению информации о состоянии процесса, принятию решений по оказанию управляющих воздействий и их исполнению относительно малыми по