

И.В. ГАЛОВ, Д.Ж. КОРЗУН
**ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ К СБОЯМ SMART-M3
ПРИЛОЖЕНИЯ НА УРОВНЕ ПРОГРАММНОЙ
ИНФРАСТРУКТУРЫ**

Галов И.В., Корзун Д.Ж. Обеспечение устойчивости к сбоям Smart-M3 приложения на уровне программной инфраструктуры.

Аннотация. Платформа Smart-M3 позволяет создавать программные приложения как интеллектуальное пространство, в котором агенты, выполняемые на разнообразных устройствах вычислительной среды, взаимодействуют через совместное накопление и использование информации. Актуальной задачей является поддержка работоспособности приложения в условиях возникновения сбоев за счет делегирования функций хранения выделенному элементу инфраструктуры приложения. Второе решение состоит из механизмов восстановления сетевых соединений. Для экспериментального исследования используется существующее Smart-M3 приложение — система интеллектуального зала SmartRoom. На ее примере показана эффективность применения предлагаемых решений.
Ключевые слова: интеллектуальные пространства, платформа Smart-M3, Интернет физических устройств, работоспособность приложения, программная инфраструктура, восстановление от сбоев.

Galov I.V., Korzun D.G. Fault Tolerance Support for a Smart-M3 Application on the Software Infrastructure Level.

Abstract. The Smart-M3 platform allows constructing software applications consisting of agents that interact by sharing information in a smart space. An important problem is dependability of the application in case of failures, which is a common place for existing networked environments. In this paper, we consider a generic software infrastructure for Smart-M3 applications and propose two solutions to support the application fault tolerance. Our first solution is introduction of a content service, which provides safety of volumetric data and their integrity due to delegation of storage functions to a separate element of the application infrastructure. The second solution is mechanisms for network connections recovery. For experimental case study, we use an existing Smart-M3 application — the SmartRoom system. Based on this case we show effectiveness of the proposed solutions.

Keywords: Smart Spaces, Smart-M3 platform, Internet of Things, fault tolerance, software infrastructure, fault recovery.

1. Введение. Рассматривается класс вычислительный сред, локализованных в ограниченном физическом пространстве с множеством разнообразных вычислительных устройств и использующих технологии Интернета физических устройств (от англ. Internet of Things, далее — IoT) для организации сетевого взаимодействия друг с другом и с внешними системами (напр., веб-сервисами из сети Интернет). Интеллектуальное пространство (ИП), разворачиваемое в такой IoT-среде, обеспечивает программное приложение средствами [1-4] для

а) построения сервисов разнородными динамическими участниками на основе их взаимодействия через разделение накапливаемой информации и извлечение из нее знаний и б) доставки сервисов конечному пользователю через доступные интерфейсные устройства.

Платформа Smart-M3 [5] является исследовательским прототипом с открытым кодом для создания ИП в IoT-средах. Программное приложение (далее — Smart-M3 приложение) строится как система взаимодействующих друг с другом программных агентов, называемых процессорами знаний (от англ. knowledge processor, далее — агент *KP*) и работающих на вычислительных устройствах IoT-среды. Агенты *KP* взаимодействуют друг с другом через обмен информацией, используя семантического информационного брокера (от англ. semantic information broker, далее — брокер *SIB*). Он управляет доступом к общему информационному содержанию ИП, реализуя косвенное взаимодействие агентов [6, 7]. Особая роль отводится операции подписки, позволяющей агенту *KP* отслеживать происходящие изменения информационного содержимого [8].

В данной работе исследуется задача обеспечения устойчивости к сбоям Smart-M3 приложений. Работоспособность таких приложений подвержена частым сбоям как в IoT-среде (включая ее сетевую компоненту), так и в работе инфраструктурных элементов ИП и самого приложения. Как правило, устойчивость к сбоям заключается в возможности приложения предоставлять сервисы в условиях возникновения сбоев [9]. В случае Smart-M3 приложений наиболее чувствительной к сбоям является операция подписки. Такие проблемы, как разрыв сетевого соединения или потеря оповещения подписки приводят к сбоям, нарушающим работоспособность приложения.

Повысить работоспособность приложения можно на уровне его инфраструктуры [10]. В данной статье выделено понятие программной инфраструктуры Smart-M3 приложения и предлагаются два решения для обеспечения устойчивости к сбоям. Первое решение дополняет инфраструктуру приложения сервисом для хранения объемных данных, предотвращая их повреждение при сбоях. Второе предлагаемое решение определяет механизмы, реализуемые каждым инфраструктурным агентом *KP*, для восстановления после сбоев подписки.

2. Обзор работ. Вопросы построения инфраструктуры для приложений ИП рассматриваются, например, в [10-13]. Примером платформы для создания приложений на основе многоагентных систем может служить платформа SISS [11]. Она, в частности, определяет многоуровневую программную инфраструктуру: уровень коммуникации обеспечивает качество сервиса и бесперебойность работы при сетевом взаимодействии

агентов, уровень координации поддерживает совместную работу агентов, а уровень сервисов содержит общие разделяемые сервисы.

В работе Wang и др. [12] предложен класс инфраструктур для ИП, основанных на технологиях Семантического Веб. Такие инфраструктуры ориентированы на явное представление данных и их семантики, на возможности поисковых запросов и выполнения на их основе рассуждений в различных контекстах.

В работе Sathish и di Flora [13] представлен программный каркас для разработки инфраструктур в ИП. Он рассчитан на инфраструктуры с динамическим построением сервисов, а также учитывает вопросы безопасности и конфиденциальности. Инфраструктура состоит из нескольких модулей: представление и обеспечение доступа к данным, хранилище и менеджер данных, репозиторий внешних данных и поддержка безопасности и конфиденциальности.

В работе da Costa и др. [10] рассмотрены инфраструктуры для сред повсеместных вычислений и сформулированы ключевые особенности их разработки. Предложена архитектурная модель, в которой работоспособность приложения позиционируется как одно из ключевых свойств, наравне с неоднородностью, расширяемостью и интероперабельностью. Следует отметить важное заключение авторов, что в системах повсеместных вычислений должны применяться такие стратегии обнаружения сбоев и восстановления, как контрольные точки, компенсация, изоляция и реконфигурация.

Вышеперечисленные работы определяют важные свойства, модели и примеры инфраструктур приложений ИП. В то же время, эти результаты неприменимы напрямую к Smart-M3 приложениям — либо из-за высокой абстрактности инфраструктур, либо, наоборот, из-за специализации инфраструктуры для частного случая.

Распространенным подходом для повышения устойчивости к сбоям в многоагентных системах является реплицирование [14]. В систему добавляются копии агентов, которые используются при возникновении сбоев в оригинальных агентах. В работе [15] предлагается использование дополнительных агентов, которые контролируют остальные агентов и обнаруживают сбои. Для платформы Smart-M3 в работе Васильева и др. [16] предложен механизм для замены агента, в котором произошел сбой, на другого. Очевидно, что такие механизмы требуют увеличения инфраструктуры приложения, пропорционального числу агентов. В данной статье нами предлагаются решения, не требующие такого значительного расширения инфраструктуры.

В работе [17] рассматриваются различные подходы для повышения устойчивости к сбоям в грид-системах. В частности, помимо репликации, выделяется стратегия реконфигурации, которая заключа-

ется в обнаружении сбоя и восстановлении состояния системы после сбоя. Этот подход используется в предлагаемом нами далее решении восстановления подписки после сбоя.

3. Инфраструктура Smart-M3 приложения. В общем случае, под инфраструктурой понимаются средства, которые делают возможным функционирование приложения [18]. Она включает в себя программно-аппаратное обеспечение и вычислительную среду. В данной работе мы развиваем понятие инфраструктуры Smart-M3 приложения, начатое ранее в [19-21].

Определим программную инфраструктуру Smart-M3 приложения как состоящую из брокера SIB и тех агентов *KP*, которые непосредственно отвечают за построение и доставку сервисов (рисунок 1). Таких агентов *KP* называют инфраструктурными [19]. Программная инфраструктура зависит от вычислительных устройств IoT-среды, сетевой аппаратуры и системного ПО, обеспечивающих выполнение брокера SIB, инфраструктурных агентов *KP* и их сетевого взаимодействия в рамках заданного приложения.



Рис. 1. Инфраструктура Smart-M3 приложения

Процесс установки, настройки и запуска основных элементов инфраструктуры на устройствах вычислительной среды называется развертыванием. Выделим три варианта развертывания инфраструктурных агентов *KP* (таблица 1).

Таблица 1. Развертывание инфраструктурного агента *KP*

Вариант развертывания	Особенности
1. Рядом с брокером SIB	Запуск основных сервисов в режиме 24/7. Простота установки и управления.
2. Рядом с аппаратурой	Взаимодействие с определенным типом устройств.
3. На отдельном сервере	Требуется трудоемкая обработка данных. Посредник между внешними источниками данных и сервисами.

Вариант 1 подразумевает запуск агента *KP* на той же ЭВМ, где запущен брокер SIB. Если брокер SIB запущен на серверной ЭВМ, то агент *KP* может функционировать постоянно, выполнять ресурсоемкие операции и получать доступ к информационному содержимому ИП с небольшой сетевой задержкой.

В варианте 2 агент *KP* запускается на отдельном компьютере, который непосредственно взаимодействует со специализированной аппаратурой для предоставления сервиса (проектор, интерактивная доска, сенсор, камера, микрофон и т.п.). Такие агенты *KP* могут служить посредниками для маломощных устройств, на которых нельзя запустить полноценных агентов *KP* (напр., сенсоры).

Вариант 3 заключается в запуске агента *KP* на серверной ЭВМ, отличной от ЭВМ с запущенным брокером SIB. Вариант предназначен для сервисов, построение которых требует ресурсоемких вычислений. Частный случай — это агенты-посредники, которые устанавливают связь между внешними источниками данных и ИП [22]. Здесь преобразование и синхронизация данных часто являются вычислительно ресурсоемкими. Развертывание на серверной ЭВМ подразумевает широкий диапазон используемых компьютеров: от локальных и корпоративных серверов до облачных систем в Интернет.

На рисунке 2 приведен вариант развертывания инфраструктуры на примере системы интеллектуального зала [20, 21]. Эта система предназначена для автоматизированного проведения таких мероприятий, как конференции, собрания, лекции. Для режима конференции выделены инфраструктурные агенты *KP*, реализующие опорные сервисы управления конференцией (Conference-service), программой мероприятия (Agenda-service), презентациями докладчиков (Presentation-service) и содержимым (Content-service).

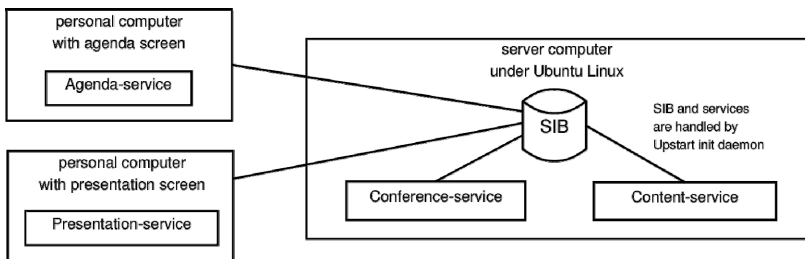


Рис. 2. Программная инфраструктура для опорных сервисов системы интеллектуального зала

Вычислительная среда системы локализована в выделенном помещении (напр., конференц-зал), оснащенном вычислительными и

медиа устройствами. Связь между устройствами обеспечивается проводными и беспроводными локальными сетями. Установлены два экрана: один отображает слайды выступающего, а другой — программу мероприятия. Два проектора для проецирования изображений на эти экраны подсоединены к двум персональным компьютерам. Брокер SIB запускается на серверной ЭВМ. Агенты для сервисов Conference-service и Content-service запускаются на ЭВМ вместе с SIB, чтобы уменьшить время сетевой передачи между агентами и ИП (вариант развертывания 1 в таблице 1). Агенты Agenda-service и Presentation-service запускаются на компьютерах, соединенных с проекторами (вариант развертывания 2).

В общем случае, программная инфраструктура предоставляет средства для функционирования приложения и обеспечивает корректность его работы. Важной характеристикой выступает работоспособность (от англ. термина *dependability*) приложения. Работоспособное приложение предоставляет сервис, которому можно обоснованно доверять [9]. Инфраструктура должна обеспечивать следующие общие свойства работоспособности программного приложения:

- доступность (*availability*): готовность к использованию;
- безотказность (*reliability*): продолжительное предоставление сервиса;
- безопасность (*safety*): отсутствие катастрофических последствий на окружающую среду;
- конфиденциальность (*confidentiality*): предотвращение неавторизованного доступа;
- целостность (*integrity*) используемых и обрабатываемых данных;
- восстановление работоспособности (*maintainability*).

Сбои во время выполнения приложения снижают его работоспособность. Обеспечение устойчивости к сбоям включает методы, позволяющие приложению предоставлять свои сервисы в условиях возникновения сбоев.

В настоящее время, для Smart-M3 приложений характерны проблемы с безотказностью работы и сохранением целостности данных, что приводит к снижению доступности сервисов. Например, сбои в операции подписки приводят к некорректной работе приложения [8] — от перерывов до полного прекращения предоставления сервисов.

Рассмотрим три причины возникновения сбоев, влияющих на работу Smart-M3 приложения: 1) сбои брокера SIB, 2) сбои инфраструктурного агента *KP*, 3) сбои сети передачи данных.

Сбои брокера SIB являются наиболее серьезными, т.к. влияют на работу всех агентов *KP*, взаимодействующих с брокером SIB. Такие

сбои связаны как с внутренними ошибками текущей реализации (статус исследовательского прототипа), так и влиянием нагрузки на брокера SIB при его работе с множеством параллельных агентов, частыми запросами к информационному содержимому ИП и с большими объемами обрабатываемой информации. Сбои инфраструктурного агента возможны как из-за наличия внутренних ошибок, так и из-за динамического участия агента в ИП. Сбои сети передачи данных связаны с ее загруженностью, приводящей к длительным задержкам или потерям данных, что особенно характерно для беспроводных коммуникаций.

Узким местом является операция подписки. Сбои приводят к потерям оповещений от брокера SIB к агентам по подписке (оповещения об изменении данных, на которые подписан агент). В частности, разрыв сетевого соединения для подписки требует от агента *KP* выполнения повторной подписки (перезапуска) [23].

Нами предлагаются два решения для применения в программной инфраструктуре Smart-M3 приложения, ориентированные на обеспечение его устойчивости к сбоям. Во-первых, в качестве одного из элементов инфраструктуры Smart-M3 приложения вводится сервис управления содержимым, который поддерживает целостность объемных данных. Во-вторых, предлагаются механизмы для восстановления подписки после сбоя. Они предназначены для обнаружения проблем с подпиской и последующего восстановления.

4. Сервис управления содержимым. Smart-M3 приложения часто требуют работы с объемным фактическими данными [4, 22] (проблемно-ориентированные массивы), которые могут быть получены из внешних источников (веб-сервисы, базы данных, облачные хранилища) или накапливаться во время работы приложения. Так, в рассматриваемом примере системы интеллектуального зала такими данными выступают презентации, аудио/видео материалы, истории произошедших событий.

Концептуально, ИП не подменяет собой хранилище информации, а выступает семантическим центром, позволяющим связать друг с другом разнородные источники данных и фрагменты знаний [3, 4, 24]. Фактические данные, в отличие от динамически меняющихся семантических связей между ними, рекомендуется хранить в специализированных хранилищах, обеспечивающих эффективное хранение и доступ с учетом формата этих данных.

Для хранения объемных данных предлагается в инфраструктуру приложения добавить сервис управления содержимым (Content-service). Какие данные пользователей или других сервисов требуется хранить — определяется логикой приложения на уровне каждого от-

дельного агента *КР*. Нами предлагается технически простое решение на основе известных веб-технологий для организации доступа к файлам. Каждый агент *КР* может загрузить в сервис управления содержимым некоторые данные в виде файла (текст, изображения, аудио, видео) и опубликовать в ИП ссылку на этот файл для использования другими агентами *КР*. Такие данные могут быть загружены и из внешних источников (напр., видео), что позволяет уменьшить время доступа агентов к внешним данным, так как сервис располагается ближе к пользователям. В этом случае, сервис управления содержимым служит кэшем. Другим преимуществом сервиса является удобство его использования в портативных (переносных) системах. Он позволяет накапливать и использовать данные в локальных вычислительных средах, в которых нет доступа к сети Интернет.

Предлагаемая архитектура сервиса управления содержимым представлена на рисунке 3. Сервис реализуется как веб-приложение на языке Python. Для его запуска используется веб-сервер Apache. Взаимодействие веб-сервера с веб-приложением осуществляется через специальный интерфейс WSGI (Web Server Gateway Interface). Сам сервис состоит из трех модулей: логика сервиса, HTML-шаблоны и инфраструктурный агент *КР*. Логика выполняет обработку HTTP-запросов, формирует ответ на основе HTML-шаблонов (шаблоны страниц сервиса), сохраняет полученные файлы (материалы пользователей). Агент *КР* обеспечивает взаимодействие сервиса с ИП: публикация в ИП ссылок на полученные и сохраненные файлы, доступ к необходимой информации из ИП (проверка аутентификации пользователя, уже опубликованные ссылки). Веб-сервер позволяет (агентам *КР*) напрямую обращаться к хранимым файлам по опубликованной в ИП ссылке.

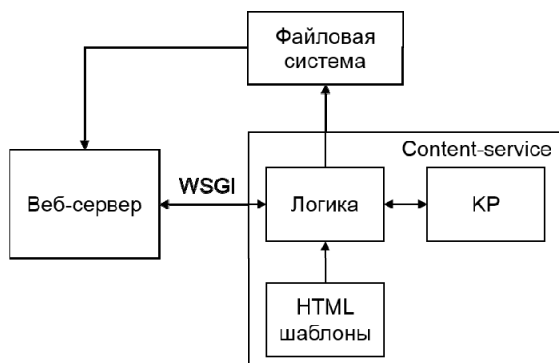


Рис. 3. Архитектура сервиса управления содержимым

Процесс загрузки и выполнения доступа к файлу изображен на рисунке 4. Пользователь 1 проходит аутентификацию и через веб-форму загружает файлы на хранение. Передача файла на веб-сервер происходит по протоколу HTTP. Файл передается на обработку в сервис Content-service, который сохраняет файл в специальную директорию в файловой системе. Доступ к файлам этой директории разрешен по протоколу HTTP через веб-сервер. Сервис публикует в ИП ссылку на этот файл. Если пользователь 2 получает из ИП соответствующую ссылку, то он может запросить файл у веб-сервера. Веб-сервер находит файл и отправляет его пользователю.

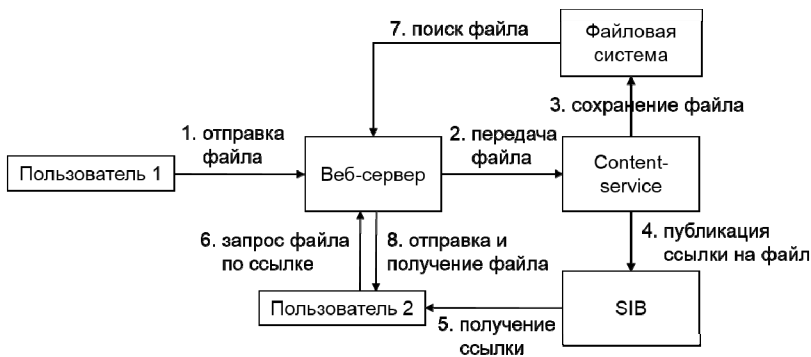


Рис. 4. Загрузка и осуществление доступа к файлу на сервисе управления содержимым

Сервис управления содержимым публикует ссылки на разделяемые данные в ИП только по запросу пользователя и не отслеживает изменения информации в ИП. Таким образом, нет необходимости использовать операцию подписки с поддержкой постоянного сетевого соединения с брокером SIB. Сетевое соединение устанавливаются только во время обработки запроса от пользователей.

В целом, за счет делегирования функций хранения объемных данных специализированному файловому хранилищу достигается повышение работоспособности работы Smart-M3 приложения. Целостность хранимых данных обеспечивается за счет использования проверенных и широко используемых технологий.

5. Обеспечение устойчивости подписки к сбоям. Контроль подписки и ее восстановление после сбоя требуют реализации на уровне каждого инфраструктурного агента *KP*. Совместно с сервисом управления содержимым такое решение позволяет реализовать инфраструктуру приложения, как показано на рисунке 5.

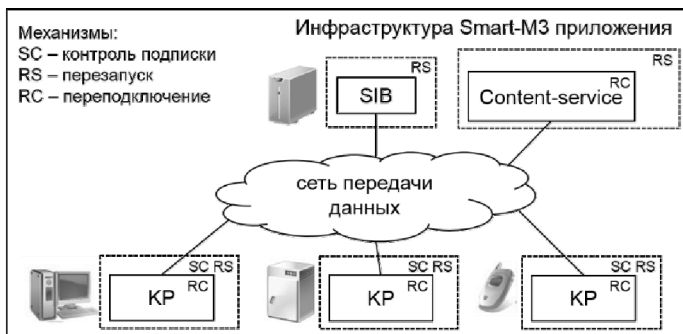


Рис. 5. Программная инфраструктура, обеспечивающая устойчивость Smart-M3 приложения к сбоям

Контроль подписки предназначен для обнаружения сбоев в подписке в двух ситуациях: 1) при потере оповещения подписки и 2) при разрыве сетевого соединения с брокером SIB. В первой ситуации инфраструктурные агенты *KP* производят проверку данных подписки (напр., по таймеру) на наличие изменений. Сбоем произошел, если агент обнаружил изменение данных. Таким образом, добавляется активная проверка подписки со стороны агента. Во второй ситуации агент обнаруживает разрыв соединения и должен перейти к процедуре его восстановления.

Для проверки пропущенных по подписке изменений может использоваться модель уведомлений, предложенная в [8]. Все сервисы приложения подписываются на собственные уведомления (специальные RDF-тройки). Уведомление представляет собой дополнительные данные в ИП, которые описывают факт изменения других данных. Если уведомление было опубликовано в ИП, но подписка не сработала, то агент *KP* может самостоятельно (автоматически или по команде пользователя) проверить наличие необработанных троек уведомления и продолжить работу. При обнаружении сбоя подписки для ее восстановления нами далее предлагаются механизмы перезапуска и переподключения.

Инфраструктурный агент *KP* переустанавливает сетевое соединение с брокером SIB и восстанавливает состояние сервиса. При переустановке соединения также происходит и переустановка подписки. При возникновении сбоя, сервис должен быть перезапущен и/или соединение агента *KP* с брокером SIB должно быть переустановлено. Ключевым моментом является восстановление предыдущего состояния сервиса (до возникновения сбоя). Перезапуск/переподключение сервиса происходит автоматически при обнаружении сбоя. При этом сохраняется возможность ручного вмешательства со стороны администратора приложения.

В механизме переподключения агент *KP* пытается возобновить сетевое соединение с брокером SIB, без прекращения работы самого сервиса. Переустановка соединения производится средствами агентов, реализующих сервис. В механизме перезапуска работа сервиса завершается, реализующие его агенты *KP* запускаются заново с установкой сетевых соединений с брокером SIB. Перезапуск осуществляется средствами внешней программной среды, где запущен сервис, например, средствами ОС.

Перезапуск рекомендуется к использованию в консольно реализованных агентах *KP*, работающих на удаленном сервере. Переподключение важно для агентов *KP* с графическим интерфейсом пользователя, т.к. их закрытие/открытие приводит к нежелательным визуальным эффектам. Перезапуск может использоваться и в консольно реализованных агентах *KP*, но для этого требуется использование обработчиков сигналов в программном коде агента.

Рассмотрим применение предложенных механизмов на примере системы интеллектуального зала (таблица 2). Брокер SIB и сервис Conference-service являются консольными приложениями, а сервис Content-service реализован как веб-приложение. Они запущены на одном сервере, чтобы работать постоянно и иметь эффективный доступ к информационному содержанию ИП. В интеллектуальном зале используется сервер под управлением операционной системы Ubuntu Linux, предоставляющей инструмент для запуска консольных приложений как сервисов (демонов): Upstart event-based init daemon. Этот инструмент управляет запуском и остановкой сервисов, поддерживает автоматический перезапуск. Если у сервиса произойдет сбой, и он завершится с ошибкой, то соответствующие инфраструктурные агенты *KP* будут перезапущены. Также Upstart позволяет организовать запуск и остановку сервисов по цепочке (при запуске одного сервиса запускаются все зависящие от него). В системе интеллектуального зала с помощью инструмента Upstart запущены консольные приложения брокера SIB и агента Conference-service.

Таблица 2. Применение механизма перезапуска/переподключения в системе интеллектуального зала

Механизм	SIB	Conference-service	Agenda-service	Presentation-service	Content-service
Перезапуск	+ (авто)	+ (авто)	+ (ручной)	+ (ручной)	+ (ручной)
Переподключение	-	-	+ (ручной)	+ (ручной)	-

Запуск брокера SIB (брокер состоит из двух процессов: redsibd и sib-tcp) и сервиса Conference-service на серверной машине должен происходить в определенной последовательности:

redsibd → sib-tcp → conference-service.

На рисунке 6 приведен пример конфигурационного файла Upstart для Conference-service. Аналогичным образом организован запуск процессов redsibd и sib-tcp. На рисунке 7 представлены зависимости, реализованные в конфигурационных файлах, между процессами SIB и Conference-service, отражающие их запуск и остановку по цепочке. При запуске процесса redsibd (при старте системы) запускаются все остальные процессы. При перезапуске/остановке одного из этих процессов (перезапуск может быть вызван автоматически, благодаря директиве respawn) все зависящие от него процессы также будут перезапущены/остановлены.

```
start on started sib-tcp          # запуск сервиса после запуска sib-tcp
stop on stopping sib-tcp        # остановка сервиса перед остановкой sib-tcp
respawn                          # директива перезапуска процесса при сбое
exec /usr/bin/conference-service # запуск сервиса
```

Рис. 6. Пример конфигурационного Upstart-файла

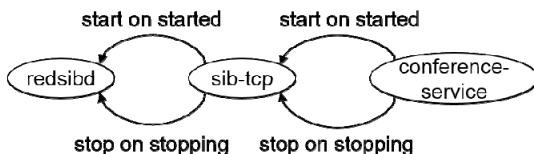


Рис. 7. Зависимости запуска брокера SIB и сервиса Conference-service

Запуском сервиса Content-service управляет веб-сервер Apache. Сетевое соединение с брокером SIB устанавливается только в момент обработки запроса от пользователя. В случае сбоя в работе для перезапуска сервиса требуется перезапустить веб-сервер.

В сервисах с графическим интерфейсом пользователя — Agenda-service и Presentation-service — перезапуск осуществляется вручную (завершение и повторный запуск). Перезапуск нежелателен для этих сервисов, т.к. они используются для отображения важной информации в интеллектуальном зале: программы мероприятия и презентации выступающего. Перезапуск испортит визуальное впечатление пользователей. Предпочтительным вариантом является использование переподключения, выполняемое администратором вручную (реализовано в программном коде, происходит при нажатии на определенную клавишу). В процессе переподключения происходит

переустановка соединения с брокером SIB и возобновление подписки без завершения самого приложения.

6. Анализ решений. Рассмотренные ранее решения в виде сервиса Content-service и механизмов восстановления после сбоев подписки предназначены для повышения работоспособности Smart-M3 приложения. Они улучшают такие характеристики, как безотказность и целостность данных. Рассмотрим далее применимость этих решений для Smart-M3 приложений, сходных с системой интеллектуального зала, и получаемое снижение негативных последствий от возможных сбоев.

Сервис Content-service. Допустим, что объемные файлы будут храниться в ИП. В этом случае, если произойдет сбой (с брокером SIB, сетью или агентом КР) во время сохранения или обработки такого файла, то он может быть поврежден или потерян. Содержимое ИП представлено набором RDF-троек. Преобразование объемных данных в тройки является сложной и затратной по времени операцией. Если сбой прервет эту операцию, то целостность данных будет нарушена. Предложенное решение в виде сервиса Content-service использует проверенные веб-технологии для сетевого обмена файлами и файловую систему для их хранения.

Сбой может произойти, когда приложение обрабатывает файл, хранящийся на удаленном компьютере. Если возникают сбои в работе сети или на удаленном компьютере, то файл становится недоступным. Сервис Content-service заранее дублирует файл локально и, таким образом, обеспечивается доступность файла.

Сервис Content-service также в определенной мере автоматизирует процесс сбора файлов. В таких системах, как система интеллектуального зала, участники должны предоставлять организаторам свои материалы для презентации. В случае ручного сбора файлов сбои часто возникают из-за влияния человеческого фактора. Организаторы могут ошибиться во время получения или сохранения файлов. Сервис Content-service автоматизирует и определяет общий подход к сбору файлов, тем самым снижая влияние человеческого фактора.

Предложенный сервис улучшает отдельные свойства работоспособности (безотказность, целостность), но он представляет централизованное решение. Отметим, что наличие распределенного хранилища может существенно усложнить систему и ухудшить ее работоспособность из-за увеличения числа элементов системы. Использование данного сервиса в системе интеллектуального зала показывает, что централизованное решение является приемлемым для рассмотренного класса Smart-M3 приложений. Более того, сбои в сервисе Content-service могут контролироваться и исправляться другими механизмами.

Механизмы восстановления подписки. Для определения сбоя осуществляется проверка измененных данных подписки. Тем не менее, не всегда возможно корректное определение изменения данных (например, между двумя проверками данные могут быть изменены, а затем возвращены в исходное состояние). Кроме этого, не всегда возможно быстро обнаружить разрыв сетевого соединения для подписки. Указанные проблемы требуют разработки математических методов для оценки частоты таких проверок. Для платформы Smart-M3 в работе [23] предлагается модель для определения интервалов активной проверки подписки.

Механизмы перезапуска и переподключения восстанавливают приложение после обнаруженного сбоя. Они могут применяться в инфраструктурных элементах при таких сбоях, как зависания, падения или некорректная работа. Помимо контроля подписки, требуются дополнительные механизмы для обнаружения других видов сбоев, после которых необходимы перезапуск или переподключение.

Производительность. Проведены эксперименты по измерению времени, за которое механизмы перезапуска и переподключения восстанавливают работоспособность отдельного сервиса. Для запуска брокера SIB, сервисов Conference-service и Content-service использовалась серверная ЭВМ (Ubuntu Linux, Intel Xeon 2.30GHz, 4GB RAM). Для запуска сервисов Agenda-service и Presentation-service использовалась персональная ЭВМ (Windows, Intel Core 2 Quad 2.40GHz, 8GB RAM). При перезапуске измерялось время между завершением сервиса и его полным запуском (готовность к обработке запросов). При переподключении измерялось время с момента разрыва сетевого соединения с брокером SIB до полного восстановления соединения. В каждом эксперименте выполнено по 10 измерений для каждого сервиса.

Результаты сведены в таблицу 3. Можно сделать вывод, что механизм перезапуска позволяет восстановить работоспособность сервиса за приемлемое время (в условиях интеллектуального зала — не более 2 с). Переподключение происходит примерно в 10 раз быстрее, поскольку процесс агента *KP* продолжает работать.

Таблица 2. Время восстановления работоспособности сервисов

Механизм	SIB (redsibd + sib-tcp)	Conference- service	Agenda- service	Presentation- service	Content- service
Перезапуск	1,045 с	2,023 с	1,350 с	0,230 с	2,193 с
Переподключение	–	–	0,134 с	0,075 с	–

Для оценки эффективности сервиса управления содержимым измерялось время загрузки данных в виде файлов размера 10Мб. Указанный объем определяет максимально разрешенный размер одной презентации в системе интеллектуального зала и отражает наиболее часто загружаемый вид объемных фактических данных. Время загрузки измерялось для двух популярных веб-серверов: Apache и nginx. Они представляют «тяжеловесный» и «легковесный» варианты веб-серверов, соответственно. Проведенное измерение позволяет оценить зависимость сервиса от используемого промежуточного ПО. Среднее время загрузки презентации размером 10Мб составляет 1,814 с (стандартное отклонение 0,014) для Apache и 1,842 с (стандартное отклонение 0,030) для nginx. Таким образом, сервис Content-service имеет сходную эффективность вне зависимости от выбора веб-сервера. Загрузка презентации требует менее 2 с, что приемлемо для условий интеллектуального зала.

7. Заключение. В статье развивается понятие программной инфраструктуры Smart-M3 приложения. Для обеспечения устойчивости к сбоям предложены два решения, реализуемых на уровне программной инфраструктуры. Первое заключается в добавлении сервиса управления объемным содержимым, который поддерживает целостность объемных данных приложения. Второе решение состоит из механизмов восстановления после сбоя подписки. На примере системы интеллектуального зала показано, что применение сервиса управления содержимым позволяет обеспечить целостность объемных данных при их активном использовании. Экспериментально показано, что восстановление сетевого соединения после возникновения сбоя подписки происходит за приемлемое на практике время.

Литература

1. *Cook D.J., Das S.K.* How smart are our environments? An updated look at the state of the art // *Pervasive and mobile computing*. 2007. vol. 3. no. 2. pp. 53–73.
2. *Юсупов Р.М., Ронжин А.Л.* От умных приборов к интеллектуальному пространству // *Вестник РАН: научный и общественно-политический журнал*. 2010. Т. 80. № 1. С. 45–51.
3. *Balandin S., Waris H.* Key properties in the development of smart spaces // *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer Berlin Heidelberg, 2009. pp. 3–12.
4. *Korzun D.G., Balandin S.I., Gurtov A.V.* Deployment of Smart Spaces in Internet of Things: Overview of the design challenges // *Internet of Things, Smart Spaces, and Next Generation Networking*. Springer Berlin Heidelberg, 2013. pp. 48–59.
5. *Honkola J. et al.* Smart-M3 information sharing platform // *Proceeding of IEEE Symposium on Computers and Communications (ISCC)*. 2010. pp. 1041–1046.
6. *Smirnov A. et al.* Anonymous agent coordination in smart spaces: State-of-the-art // *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer Berlin Heidelberg, 2009. pp. 42–51.

7. Корзун Д.Ж., Ломов А.А., Ванаг П.И. Автоматизированная модельно-ориентированная разработка программных агентов для интеллектуальных пространств на платформе Smart-M3 // Программная инженерия. 2012. № 5. С. 6–14.
8. Галов И.В., Корзун Д.Ж. Модель уведомлений для разработки программных приложений интеллектуальных пространств // Труды СПИИРАН. 2014. Вып. 35. С. 189–211.
9. Lyu M.R. et al. Handbook of software reliability engineering // CA: IEEE computer society press. 1996. vol. 222.
10. da Costa C.A., Yamin A.C., Geyer C.F.R. Toward a general software infrastructure for ubiquitous computing // IEEE Pervasive Computing. 2008. vol. 7. no. 1. pp. 64–73.
11. Xie W. et al. Smart platform-a software infrastructure for smart space (siss) // Proceedings of Fourth IEEE International Conference on Multimodal Interfaces. IEEE. 2002. pp. 429–434.
12. Wang X. et al. Semantic space: An infrastructure for smart spaces // Computing. 2002. vol 1. no. 2. pp. 67–74.
13. Sathish S., di Flora C. Supporting smart space infrastructures: a dynamic context-model composition framework // Proceedings of the 3rd international conference on Mobile multimedia communications. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2007. pp. 67.
14. Guessoum Z. et al. Towards Reliable Multi-Agent Systems: An Adaptive Replication Mechanism // Multiagent and Grid Systems. 2010. vol. 6. no. 1. pp. 1–24.
15. Mitrovic D. et al. Agent-based approaches to managing fault-tolerant networks of distributed multi-agent systems // Multiagent and Grid Systems. 2011. vol. 7. no. 6 pp. 203–218.
16. Vasilev A. et al. Mechanism for context-aware substitution of Smart-M3 agents based on dataflow network model // Proceedings of International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE. 2013. pp. 113–117.
17. Latchoumy P., Sheik Abdul Khader P. Survey on Fault Tolerance in Grid Computing // International Journal of Computer Science & Engineering Survey. 2011. vol. 2. no. 4. pp. 97–110.
18. Трутнев Д.Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие // СПб.: НИУ ИТМО. 2012. 66 с.
19. Galov I., Korzun D. The smartroom infrastructure: Service runtime reliability // Proceedings of the 14th Conference of Open Innovations Association FRUCT. SUAI. 2013. pp. 188–189.
20. Korzun D., Galov I., Balandin S. Development of Smart Room Services on Top of Smart-M3 // Proceedings of the 14th Conf. of Open Innovations Association FRUCT. SPb.: SUAI. 2013. pp. 37–44.
21. Korzun D., Galov I., Kashevnik A., Balandin S. Virtual shared workspace for smart spaces and M3-based case study // Proceedings of the 15th Conference of Open Innovations Association FRUCT. SPb.: ITMO University. 2014. pp. 60–68.
22. Korzun D.G. et al. Integration of Smart-M3 applications: Blogging in smart conference // Smart Spaces and Next Generation Wired/Wireless Networking. Springer Berlin Heidelberg. 2011. pp. 51–62.
23. Vdovenko A., Korzun D. Active control by a mobile client of subscription notifications in smart space // Proceedings of the 16th Conference of Open Innovations Association FRUCT. SPb.: ITMO University. 2014. pp. 123–128.
24. Korzun D., Balandin S. A Peer-to-Peer Model for Virtualization and Knowledge Sharing in Smart Spaces // Proceedings of the 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014). 2014. pp. 87–92.

References

1. Cook D.J., Das S.K. How smart are our environments? An updated look at the state of the art. Pervasive and mobile computing. 2007. vol. 3. no. 2. pp. 53–73.
2. Yusupov R.M., Ronzhin A.L. [From smart devices to smart space]. *Vestnik RAN: nauchnyy i obshchestvenno-politicheskiy zhurnal – RAN Bulletin: scientific and socio-political journal*. 2010. vol. 80. no. 1. pp. 45–51. (In Russ.).

3. Balandin S., Waris H. Key properties in the development of smart spaces. *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer Berlin Heidelberg, 2009. pp. 3–12.
4. Korzun D.G., Balandin S.I., Gurtov A.V. Deployment of Smart Spaces in Internet of Things: Overview of the design challenges. *Internet of Things, Smart Spaces, and Next Generation Networking*. Springer Berlin Heidelberg, 2013. pp. 48–59.
5. Honkola J. et al. Smart-M3 information sharing platform. *Proceeding of IEEE Symposium on Computers and Communications (ISCC)*. 2010. pp. 1041–1046.
6. Smirnov A. et al. Anonymous agent coordination in smart spaces: State-of-the-art. *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer Berlin Heidelberg, 2009. pp. 42–51.
7. Korzun D.G., Lomov A.A., Vanag P.I. [Automated model-oriented development of software agents for smart spaces on Smart-M3 platform]. *Programmnaya Inzheneriya – Software engineering*. 2012. no. 5. pp. 6–14. (In Russ.).
8. Galov I.V., Korzun D.G. [Notification model for smart space applications development]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2014. no. 35. pp. 189–211. (In Russ.).
9. Lyu M.R. et al. *Handbook of software reliability engineering*. CA: IEEE computer society press, 1996. vol. 222.
10. da Costa C.A., Yamin A.C., Geyer C.F.R. Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*. 2008. vol. 7. no. 1. pp. 64–73.
11. Xie W. et al. Smart platform-a software infrastructure for smart space (siss). *Proceedings of Fourth IEEE International Conference on Multimodal Interfaces*. IEEE. 2002. pp. 429–434.
12. Wang X. et al. Semantic space: An infrastructure for smart spaces. *Computing*. 2002. vol. 1. no. 2. pp. 67–74.
13. Sathish S., di Flora C. Supporting smart space infrastructures: a dynamic context-model composition framework. *Proceedings of the 3rd international conference on Mobile multimedia communications. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*. 2007. pp. 67.
14. Guessoum Z. et al. Towards Reliable Multi-Agent Systems: An Adaptive Replication Mechanism. *Multiagent and Grid Systems*. 2010. vol. 6. no. 1. pp. 1–24.
15. Mitrovic D. et al. Agent-based approaches to managing fault-tolerant networks of distributed multi-agent systems. *Multiagent and Grid Systems*. 2011. vol. 7. no. 6 pp. 203–218.
16. Vasilev A. et al. Mechanism for context-aware substitution of Smart-M3 agents based on dataflow network model. *Proceedings of International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE. 2013. pp. 113–117.
17. Latchoumy P., Sheik Abdul Khader P. Survey On Fault Tolerance In Grid Computing. *International Journal of Computer Science & Engineering Survey*. 2011. vol. 2. no. 4. pp. 97–110.
18. Trutnev D.R. *Arkhitektury informatsionnykh sistem. Osnovy proektirovaniya: Uchebnoe posobie* [Architectures of information systems. Basics of design: Textbook]. SPb.: NIU ITMO. 2012. 66 p. (In Russ.).
19. Galov I., Korzun D. The smartroom infrastructure: Service runtime reliability. *Proceedings of the 14th Conference of Open Innovations Association FRUCT. SUAI*. 2013. pp. 188–189.
20. Korzun D., Galov I., Balandin S. Development of Smart Room Services on Top of Smart-M3. *Proceedings of the 14th Conf. of Open Innovations Association FRUCT. SUAI*. 2013. pp. 37–44.
21. Korzun D., Galov I., Kashevnik A., Balandin S. Virtual shared workspace for smart spaces and M3-based case study. *Proceedings of the 15th Conference of Open Innovations Association FRUCT. ITMO*. 2014. pp. 60–68.

22. Korzun D.G. et al. Integration of Smart-M3 applications: Blogging in smart conference. *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer Berlin Heidelberg. 2011. pp. 51–62.
23. Vdovenko A., Korzun D. Active control by a mobile client of subscription notifications in smart space. *Proceedings of the 16th Conference of Open Innovations Association FRUCT*. 2014. pp. 123–128.
24. Korzun D., Balandin S. A Peer-to-Peer Model for Virtualization and Knowledge Sharing in Smart Spaces. *Proceedings of the 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014)*. 2014. pp. 87–92.

Галов Иван Викторович — аспирант, младший научный сотрудник, кафедры информатики и математического обеспечения, математический факультет, Петрозаводский государственный университет (ПетрГУ). Область научных интересов: интеллектуальные пространства, семантический веб, онтологическое моделирование, программная инженерия. Число научных публикаций — 25. galov@cs.karelia.ru; пр. Ленина, д. 33, г. Петрозаводск, 185910, РФ; p.t. +7(8142)711015

Galov Ivan Viktorovich — Ph.D. student, junior researcher, Department of Computer Science, Faculty of Mathematics, Petrozavodsk State University (PetrSU). Research interests: smart spaces, semantic web, ontological modeling, software engineering. The number of publications — 25. galov@cs.karelia.ru; Lenin St. 33, Petrozavodsk, 185910, Russia; office phone +7(8142)711015

Корзун Дмитрий Жоржевич — к-т физ.-мат. наук, доцент, ведущий научный сотрудник, кафедра информатики и математического обеспечения, математический факультет, Петрозаводский государственный университет (ПетрГУ). Область научных интересов: научных интересов: анализ распределенных систем, дискретное моделирование, повсеместные вычисления и интеллектуальные пространства, Интернет физических устройств, технологии разработки ПО, проектирование алгоритмов и вычислительная сложность, линейный диофантов анализ и его приложения, теория формальных языков и методы трансляции. Число научных публикаций — 135. dkorzun@cs.karelia.ru; 185910, Республика Карелия, г. Петрозаводск, пр. Ленина, 33, Каб.: 217; p.t. +79095689571, факс +7(8142)711000

Korzun Dmitry Georzhevich — Ph.D., associate professor, leading researcher, Department of Computer Science, Faculty of Mathematics, Petrozavodsk State University (PetrSU). Research interests: analysis and evaluation of distributed systems, discrete modeling, ubiquitous computing in smart spaces, Internet of Things, software engineering, algorithm design and complexity, linear Diophantine analysis and its applications, theory of formal languages and parsing. The number of publications — 135. dkorzun@cs.karelia.ru; Lenin St. 33, Petrozavodsk, 185910, Russia; office phone +79095689571, fax +7(8142)711000

Поддержка исследований. Исследование выполнено при финансовой поддержке Минобрнауки России по заданию № 2014/154 на выполнение государственных работ в сфере научной деятельности в рамках базовой части государственного задания, НИР № 1481. Работа поддержана федеральной целевой программой «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы», соглашение № 14.574.21.0060 (RFMEFI57414X0060).

Acknowledgements. Project # 1481 from the basic part of state research assignment # 2014/154 of the Ministry of Education and Science of Russia. Project # 14.574.21.0060 (RFMEFI57414X0060) of Federal Target Program.

РЕФЕРАТ

Галов И.В., Корзун Д.Ж. **Обеспечение устойчивости к сбоям Smart-M3 приложения на уровне программной инфраструктуры.**

Платформа Smart-M3 позволяет создавать программные приложения интеллектуальных пространств на основе набора взаимодействующих друг с другом агентов. Такое взаимодействие выполняется через общее разделяемое информационное содержимое с использованием механизма публикации/подписки. Работоспособность Smart-M3 приложений подвержена частым сбоям в агентах, где наиболее чувствительной к сбоям является операция подписки. Одним из способов повышения работоспособности приложения является обеспечение его устойчивости к сбоям.

В работе рассматривается обеспечение устойчивости к сбоям на уровне программной инфраструктуры Smart-M3 приложения. Авторами предлагаются два решения для обеспечения устойчивости к сбоям. Первое решение заключается во введении в программную инфраструктуру агента, который обеспечивает сохранность объемных данных и их целостность. Это достигается за счет делегирования функций хранения отдельному элементу инфраструктуры приложения. Второе решение состоит из механизмов восстановления сетевых соединений после сбоев в элементах инфраструктуры приложения, в частности при проблемах с операцией подписки. Данные механизмы позволяют обнаружить сбой и восстановить работу агента после него. Для экспериментального исследования используется существующее Smart-M3 приложение — система интеллектуального зала SmartRoom. На ее примере показана эффективность применения предлагаемых решений.

SUMMARY

Galov I.V., Korzun D.G. Fault Tolerance Support for a Smart-M3 Application on the Software Infrastructure Level.

The Smart-M3 platform allows creating software applications of smart spaces based on a set of interacting agents. Interaction is organized via shared informational content using publish/subscribe mechanism. Dependability of Smart-M3 applications suffers from frequent failures in software agents where subscription operation is the most failure-sensitive operation. A possible way of improving the application dependability is implementation of fault tolerance support within a given application.

This work considers fault tolerance support on the software infrastructure level of a Smart-M3 application. The authors suggest two solutions for fault tolerance support. The first solution is extending the software infrastructure with an agent that provides safety and integrity of volumetric data due to delegation of storage functions to a separate element of the application infrastructure. The second solution is mechanisms for network connections recovery after failures in infrastructural elements of an application especially after failures in subscription operation. These mechanisms allow to detect a failure and to recover agent runtime after the failure. For the reference case study, an existing Smart-M3 application—the SmartRoom system—is used. Based on the experimental analysis for the SmartRoom system, the authors show the effectiveness of proposed solutions.