Санкт-Петербургский государственный университет

Программная инженерия

Гумин Егор Дмитриевич

Разработка системы для мониторинга состояния серверов посредством интерактивных витрин

Бакалаврская работа

Научный руководитель:
ст. преп. Д. В. Луцив

Рецензент:
инженер-консультант ООО "САП Лабс" Н. А. Ражев

Санкт-Петербург
2017

SAINT PETERSBURG STATE UNIVERSITY

Software engineering

Egor Gumin

Development of a system for server monitoring via dashboards

Graduation Thesis

Scientific supervisor:
assistant prof. Dmitry V. Luciv

Reviewer:
Engineer at SAP Labs Nikita Razhev

Saint Petersburg
2017

Оглавление

Введение

Мониторинг – это отслеживание ключевых показателей эффективности сервера: загрузки дисков, процессора и других параметров. Он является одним из ключевых инструментов обеспечения бесперебойного функционирования серверов. Мониторинг обычно производится удаленно, причем отслеживается состояние десятков или даже тысяч серверов одновременно. При несоответствии наблюдаемых показателей ожидаемым производится сигнализация о неполадках (например, отправляется письмо по электронной почте системному администратору). Такой способ мониторинга далее будет обозначаться как *превентивный*. Но в некоторых ситуациях уже известно, что на конкретном сервере имеются неполадки, однако нет информации о том, в чем именно они заключаются. Архивные данные, собранные с помощью превентивного мониторинга, не всегда обладают достаточной полнотой, что приводит к необходимости производить дополнительный мониторинг данного сервера в режиме реального времени. Чаще всего в таких случаях требуется отслеживать расширенный набор показателей для того, чтобы локализовать и устранить проблему.

Существует множество программных решений для осуществления мониторинга серверов, например, Zabbix [21], Nagios [18], Monit [17]. Однако, они чаще всего не отличаются гибкостью и либо являются слишком сложными для использования в целях мониторинга в режиме реального времени, либо не могут быть применены в компаниях с жесткой политикой безопасности из-за нестандартных способов подключения к серверам. Примером такого случая является ситуация, когда подключение к серверу возможно только через определенную внешнюю программу (SSH-клиент для операционной системы Windows с закрытым исходным кодом). В связи с этим при необходимости такого рода мониторинга инженеры компании SAP, которые занимаются поддержкой пользователей по вопросам функционирования серверов, вынуждены применять архаичные методы мониторинга: запуск мониторинговых утилит на сервере через SSH-подключение, с последующим анализиро-

ванием объемных текстовых журналов (логов), порожденных утилитами, либо экспортирование этих журналов в специальные программы для визуализации. Следует отметить, что такой подход требует больших временных затрат и исполнения большого количества действий для получения понятной и наглядной информации. Поэтому возникает необходимость в разработке системы, которая позволила бы облегчить процесс мониторинга серверов в режиме реального времени и обладала гибкостью, необходимой для функционирования в условиях, когда прямое подключение к серверу невозможно.

1. Постановка задачи

Целью данной работы является разработка системы для мониторинга серверов в режиме реального времени. Для достижения данной цели были поставлены следующие задачи.

- Провести анализ требований к мониторинговой системе.
- Проанализировать существующие программные продукты в сфере мониторинга.
- Разработать архитектуру системы.
- Реализовать прототип системы.
- Провести апробацию прототипа.

2. Требования к системе

В этом разделе описываются требования к системе с обсуждением причин, по которым они были выдвинуты. Требования были разделены на две категории – функциональные и нефункциональные.

2.1. Функциональные требования

2.1.1. Графический веб-интерфейс

Основная задача разрабатываемой системы – предоставить инженерам компании SAP возможность производить мониторинг серверов в режиме реального времени, отслеживая интересующие их показатели при помощи интерактивных витрин. В связи с этим система должна иметь графический веб-интерфейс, в котором информация о мониторинге будет представлена в виде графиков и таблиц.

2.1.1.1. Функциональность графиков

Графики являются важнейшей частью интерфейса мониторинговой системы. Был сформирован ряд требований, соблюдение которых необходимо для того, чтобы для инженеров работать с графиками было удобно.

a. Необходимо обеспечить обновление информации на графиках в режиме реального времени со скоростью не реже одного раза в 15 секунд.

b. Система должна поддерживать группировку мониторинговых данных по нескольким графикам по условиям пользователя и их отображение на одной странице.

c. Необходимо наличие функции изменения детализации графических данных (zoom).

d. Необходимо наличие функции временного анализа, позволяющего получать доступ к историческим данным (с помощью элемента

прокрутки).

e. Необходимо наличие функции анализа корреляционных зависимостей, выражающееся в возможности комбинирования различных графических данных, соотнесенных по временной шкале.

f. Показатели на графике должны быть представлены различными контрастными цветами.

2.1.1.2. Функциональность таблиц

Предполагается, что вариант табличного отображения данных будет использоваться не так часто, как вариант с графиками, тем не менее, были выделены базовые требования, соответствие которым сделает табличный интерфейс более дружелюбным для пользователя.

a. Необходимо обеспечить обновление информации в таблицах в режиме реального времени со скоростью не реже одного раза в 15 секунд.

b. Необходимо обеспечить возможность сортировки таблиц по любому из показателей (колонок) в лексикографическом порядке (как от меньшего к большему, так и наоборот).

2.1.2. Поддержка мониторинговых утилит

Под мониторинговой утилитой в этой работе понимается любая программа, которая способна выводить в консоль информацию о показателях работы сервера. Весьма распространенными среди пользователей Unix-подобных операционных систем являются утилиты sar [7] и pidstat [6], которые позволяют получить информацию о загрузке ядер процессора, использовании дисков и многих других показателях.

2.1.2.1. Добавление поддержки мониторинговых утилит

Невозможно предусмотреть все проблемы, которые придется решать инженерам с помощью системы мониторинга и выяснить заранее, под-

держку каких мониторинговых утилит потребуется обеспечить в период эксплуатации системы. Поэтому необходимо предоставить инструмент, который поможет инженерам самостоятельно реализовывать поддержку необходимых мониторинговых утилит (утилитных расширений). Разработка утилитного расширения не должна требовать модификации основного исходного кода системы для мониторинга.

2.1.2.2. Сохранение утилитных расширений

Необходимо обеспечить функцию сохранения разработанных утилитных расширений. Если расширение было разработано во время одного из сеансов работы с мониторинговой системой, оно должно быть доступно и при следующих запусках системы, если не предпринимались действия по его отключению или удалению.

2.1.2.3. Распространение утилитных расширений

Поскольку при мониторинге нередко возникают однотипные задачи, нескольким инженерам могут понадобиться одни и те же утилитные расширения. Поэтому необходимо использовать для расширения формат, который обеспечит их легкую передачу между инженерами, либо реализовать подсистему для передачи расширений.

2.1.2.4. Базовый набор утилитных расширений

Сложно предсказать весь набор мониторинговых утилит, которые будут использоваться в период эксплуатации системы, но возможно ограничить их самое популярное множество. В компании SAP для мониторинга чаще всего используются уже упомянутые утилиты sar и pidstat. Необходимо реализовать утилитные расширения для них и включить их в стандартный комплект поставки системы.

2.1.3. Способы подключения к серверам

SSH-клиент – это программа, используемая для подключения к удаленному серверу через протокол SSH [1], который является сетевым протоколом прикладного уровня и позволяет производить удаленное

управление операционной системой. По функциональности этот протокол схож с протоколом Telnet [2], но SSH подразумевает шифрование всей передаваемой информации.

2.1.3.1. Подключение к серверу по протоколу SSH

Система должна предоставлять возможность подключения к серверу через протокол SSH для осуществления мониторинга. Для подключения пользователю необходимо указать адрес сервера, имя пользователя и пароль для входа в систему. Отличительной особенностью сервера SSH является консольно-текстовый режим ввода-вывода и его наличие в любом коммерческом дистрибутиве Linux, поддерживаемом компанией SAP.

2.1.3.2. Подключение к серверу через внешний SSH-клиент

Система должна предоставлять возможность подключения к внешнему Windows-приложению (SSH-клиенту SAPuTTY) для использования его в качестве канала связи при мониторинге сервера. SAPuTTY является модифицированной службой информационной безопасности компании SAP версией популярного SSH-клиента с открытым исходным кодом PuTTY [20]. В некоторых случаях использование SAPuTTY является единственным способом подключения к серверам.

2.2. Нефункциональные требования

2.2.1. Портативность системы

Во многих крупных компаниях, в том числе и в компании SAP, существует корпоративный образ операционной системы, который содержит все необходимые программы и обновления и поддерживается ИТ-службой компании. Для того чтобы установить дополнительное программное обеспечение в таком окружении, необходимо либо включение его в корпоративный образ, либо его самостоятельная установка по договоренности со службой информационной безопасности компании. Включение программного продукта в корпоративный образ - сложная

задача, поскольку необходимо доказать ИТ-службе необходимость этого продукта на всех компьютерах с этим образом, при этом большинству пользователей этого образа новый продукт не нужен. Кроме того, даже после этого согласования, ИТ-служба должна будет провести тестирование продукта на безопасность и совместимость и может отказать во внедрении при неудовлетворительных результатах. Вариант с самостоятельной установкой тоже не всегда возможен – некоторые продукты могут уже находиться в списке запрещенного к установке на компьютеры компании программного обеспечения, а некоторые могут быть запрещены отделом информационной безопасности при попытке согласования. Поскольку многие существующие мониторинговые системы и системы интерактивных витрин требуют установки большого количества дополнительного программного обеспечения, следует обратить внимание на то, разрешено ли использовать его в компании ИТ-службой. Кроме того, значительная часть такого программного обеспечения может распространяться бесплатно только на условиях использования в личных целях, а при работе в сферах, связанных с получением прибыли, требуется покупка лицензии, что также препятствует широкому использованию его в коммерческих компаниях. Поэтому по возможности следует использовать решения, которые не требуют установки, и работают портативно в своей изолированной среде.

2.2.2. Ограничения по расходу ресурсов

Системы, подвергаемые мониторингу, обычно имеют большое количество процессорных ядер и жестких дисков, что приводит к генерированию большого количества мониторинговых данных. Однако мониторинг предполагается проводить на персональных компьютерах инженеров с размером оперативной памяти не менее 8 Гб и размером жесткого диска до 1 Тб, при этом сохраняя возможность одновременного использования компьютера для других задач. В связи с этим, следует учесть нижеследующие ограничения.

2.2.3. Ограничения по расходу оперативной памяти

Система должна расходовать не более 4 Гб оперативной памяти при длительности мониторинга системы с 96 ядрами и 48 жесткими дисками по параметрам команд sar и pidstat до 10 часов.

2.2.4. Ограничения по расходу постоянной памяти

Система должна расходовать не более 10 Гб постоянной памяти при длительности мониторинга системы с 96 ядрами и 48 жесткими дисками по параметрам команд sar и pidstat до 10 часов. После завершения сеанса мониторинга собранные данные не должны храниться на компьютере инженера и подлежат удалению.

2.2.5. Условия сопровождения системы

Поскольку в служебные обязанности инженеров компании SAP не входит настройка и сопровождение систем мониторинга, а выделение для этой цели системного администратора не планируется (при самом благоприятном сценарии один из инженеров сможет уделять задачам администрирования не более 10–15 минут в неделю на добровольной основе), система должна быть максимально устойчива к обновлениям корпоративного образа операционной системы. Такие обновления могут приводить к изменению версий программного обеспечения, используемых мониторинговой системой в своей работе или полному удалению некоторых программ.

2.2.6. Изменение графического интерфейса

В компании SAP для разработки веб-приложений используется фреймворк SAPUI5 [22]. Основное назначение SAPUI5 – разработка приложений, которые одинаково хорошо будут выглядеть как на мобильных устройствах, так и на компьютерах. С использованием SAPUI5 было написано большое количество корпоративных ресурсов и приложений, такой стиль интерфейса знаком и удобен инженерам. В связи с этим

необходимо применение фреймворка SAPUI5 или его аналога с открытым исходным кодом OpenUI5 [19] для графического интерфейса системы мониторинга. Хотя использование SAPUI5 непосредственно в прототипе системы не является обязательным, необходимо предусмотреть его интеграцию на дальнейших этапах разработки.

2.2.7. Загрузка данных в облачное хранилище

В компании SAP имеется система, которая анализирует данные, собранные в процессе мониторинга из различных источников. Обеспечение интеграции с этой системой позволит производить анализ данных, результаты которого могут использоваться для построения трендов и предиктивного анализа. По этой причине необходимо обеспечить возможность периодической загрузки данных о мониторинге в облачное хранилище компании в формате JSON по REST API, где они будут подвергаться дальнейшей обработке.

2.2.8. Запрет на модификацию или расширение серверной функциональности

По правилам компании SAP, ее сотрудники и представители не имеют права самостоятельно вносить какие-либо модификации в клиентские промышленные системы, из чего вытекает ключевое требование для утилиты мониторинга, а именно прозрачное использование встроенных средств операционной системы и программного обеспечения SAP. По этой причине мониторинг должен осуществляться без установке дополнительного программного обеспечения (так называемых агентских программ) на сервер.

2.2.9. Конфигурация операционной системы

Система должна работать на компьютерах с операционными системами Windows Server 2008 и Windows 10. Гарантируется наличие на целевом компьютере установленной платформы Java версии 8 и новее а также браузера Internet Explorer 11 или новее.

2.2.10. Документация

Для внедрения системы требуется провести необходимое обучение инженеров. В качестве способа обучения была выбрана документация, так как это требует наименьших затрат человеческих ресурсов (подробнее об ограничениях в пункте 2.2). Кроме того, при успешном опыте внедрения системы в подразделении компании в Санкт-Петербурге, планируется обсудить перспективы ее внедрения в подразделениях, расположенных в Германии. В связи с этим необходимо разработать документацию на русском и английском языках, которая будет содержать инструкции по первоначальной настройке системы и добавлению утилитных расширений.

3. Анализ существующих решений

В ходе анализа требований были определены основные требования к существующим решениям из области мониторинга и интерактивных витрин, на основе которых можно было бы реализовать систему.

- Бесплатно для коммерческого использования.
- Запуск с компьютера инженера под управлением ОС Windows Server 2008 или Windows 10.
- Имеется возможность впоследствии использовать веб-интерфейс на основе фрэймворка SAPUI5.
- Имеется возможность осуществлять мониторинг при помощи SSH-клиента SAPuTTY.
- Имеется возможность настроить периодическую отправку журналов мониторинга во внутреннюю Систему хранения и анализа журналов компании SAP.
- Нет необходимости устанавливать большое количество стороннего программного обеспечения. Работа самого решения в портативном режиме также является преимуществом.
- Возможность для инженеров добавлять утилитные расширения в короткие сроки (порядка 5–15 минут) без перезапуска системы мониторинга.
- Минимальные усилия по поддержке.

На предмет соответствия выдвинутым требованиям были проанализированы популярные мониторинговые системы и системы интерактивных витрин.

3.1. kSar

kSar [5] – это портативное Java-приложение с открытым исходным кодом, которое позволяет визуализировать журналы мониторинга команды sar. Хотя kSar обладает уже реализованной поддержкой утилиты sar, программа не предполагает возможности добавления новых утилитных расширений и способов подключения к серверам и не реализует поддержки веб-интерфейса, что делает ее неприменимой в рамках решения поставленной задачи.

3.2. Системы интерактивных витрин

Были рассмотрены системы интерактивных витрин данных, такие как Grafana [15], Kibana [16], Cacti [14]. Эти системы обладают широкими возможностями построения графиков по мониторинговым данным, однако, все эти преимущества будут утеряны при замене веб-интерфейса на вариант, использующий SAPUI5. При этом комбинирование интерфейсов систем интерактивных витрин с интерфейсом на основе фреймворка SAPUI5 является технически сложной задачей, а результат не будет соответствовать стандартам визуального оформления веб-приложений компании SAP. По этой причине при разработке этого проекта было принято решение отказаться от использования систем интерактивных витрин, за исключением той, которая поставляется с SAPUI5.

3.3. Zabbix

Zabbix – это бесплатная система мониторинга серверов и компьютерного оборудования с открытым исходным кодом. Обычно с помощью Zabbix производится мониторинг состояния сотен или тысяч серверов одновременно. Эта система настолько комплексна, многофункциональна и сложна, что компания Zabbix предоставляет услуги по проведению обучающих курсов для инженеров, которым затем предлагается сдать сертификационный экзамен на уровень сертифицированного специалиста или сертифицированного профессионала по системе Zabbix. В связи

с тем, что с помощью этого решения обычно производится мониторинг большого количества устройств, для этой цели выделяется отдельный сервер под управлением ОС Linux, на который устанавливается программное обеспечение Zabbix-сервер. Для компьютеров под управлением ОС Windows версии Zabbix-сервера на данный момент не существует. Альтернативный вариант – установка образа Linux с Zabbix-сервером на виртуальную машину слишком сложен в первоначальной настройке, повседневном использовании и поддержке, а также требует согласования со службой информационной безопасности компании. При затраты времени на запуск такой системы мониторинга и потребление ресурсов будут значительными. По вышеперечисленным причинам принято решение не использовать Zabbix как основу разрабатываемой системы.

Таким образом, из-за специфических требований к интерфейсу и каналам сбора данных, а также других ограничений, ни одно из рассмотренных решений в области мониторинга или интерактивных витрин не было взято за основу реализации системы мониторинга в режиме реального времени.

4. Архитектура системы и особенности реализации

С учетом требований и анализа существующих решений, была разработана базовая архитектура системы для мониторинга серверов в режиме реального времени. Проект такой архитектуры представлен на рисунке 1.

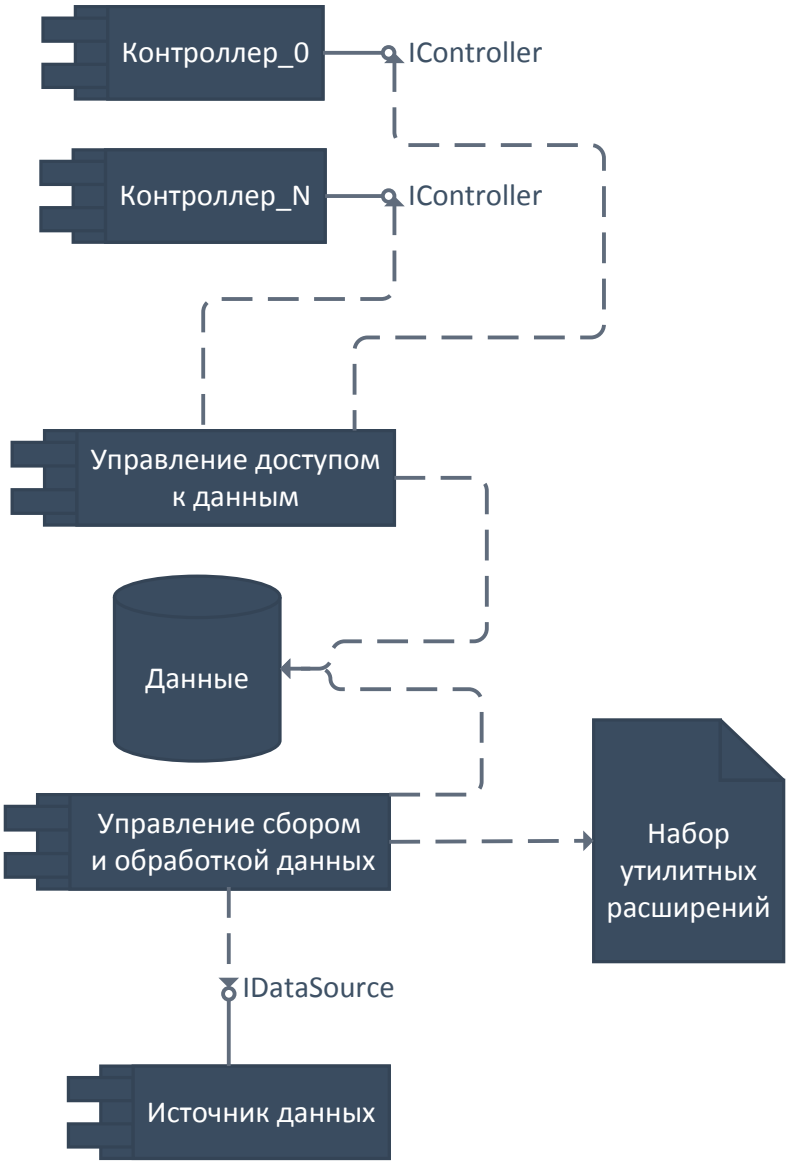

Рис. 1: Диаграмма компонентов базовой архитектуры системы мониторинга

Внутренняя логика системы реализована в виде трех основных блоков: сбор данных, обработка и хранение данных и манипуляция данными. Сбор данных производится через *источник данных* – класс, который является адаптером для любой сущности, способной генерировать мониторинговые данные по запросу. Затем собранные данные обрабатываются с учетом параметров, по которым ведется мониторинг и целей, в которых далее будут использованы данные. Такие параметры необходимо учитывать, потому что различные утилиты генерируют вывод в отличающихся форматах. Цели использования учитываются для выбора оптимального формата хранения данных, что позволяет сократить расходы памяти и повысить удобство их использования в различных системах. После первичной обработки данные складываются в хранилище. Данные из хранилища потребляются *контроллерами* – модулями, которые различными способами манипулируют данными, например, делают их доступными для веб-интерфейса или отправляют фрагменты мониторинговых журналов в облачное хранилище. Как показано на рисунке 1, к системе могут быть подключены несколько контроллеров одновременно.

На основе описанной базовой архитектуры системы была разработана более детальная архитектура системы для мониторинга серверов в режиме реального времени инженерами компании SAP во время сеансов поддержки пользователей. Ее упрощенная модель представлена на рисунке 2. Более светлым цветом обозначены части системы, которые не разрабатывались в рамках данной работы – система хранения и анализа журналов и SSH-клиент SAPuTTY.

В процессе выяснения требований выяснилось, что наличие виртуальной машины Java гарантируется на всех компьютерах с корпоративным образом операционной системы. Java – язык высокого уровня с широкими возможностями выбора библиотек, что позволяет быстро вести разработку. При этом Java позволяет получать доступ к Windows API, что необходимо для взаимодействия с SAPuTTY, а также создавать портативные приложения. По этим причинам для реализации системы была выбрана платформа Java. Далее приводится покомпонентное опи-

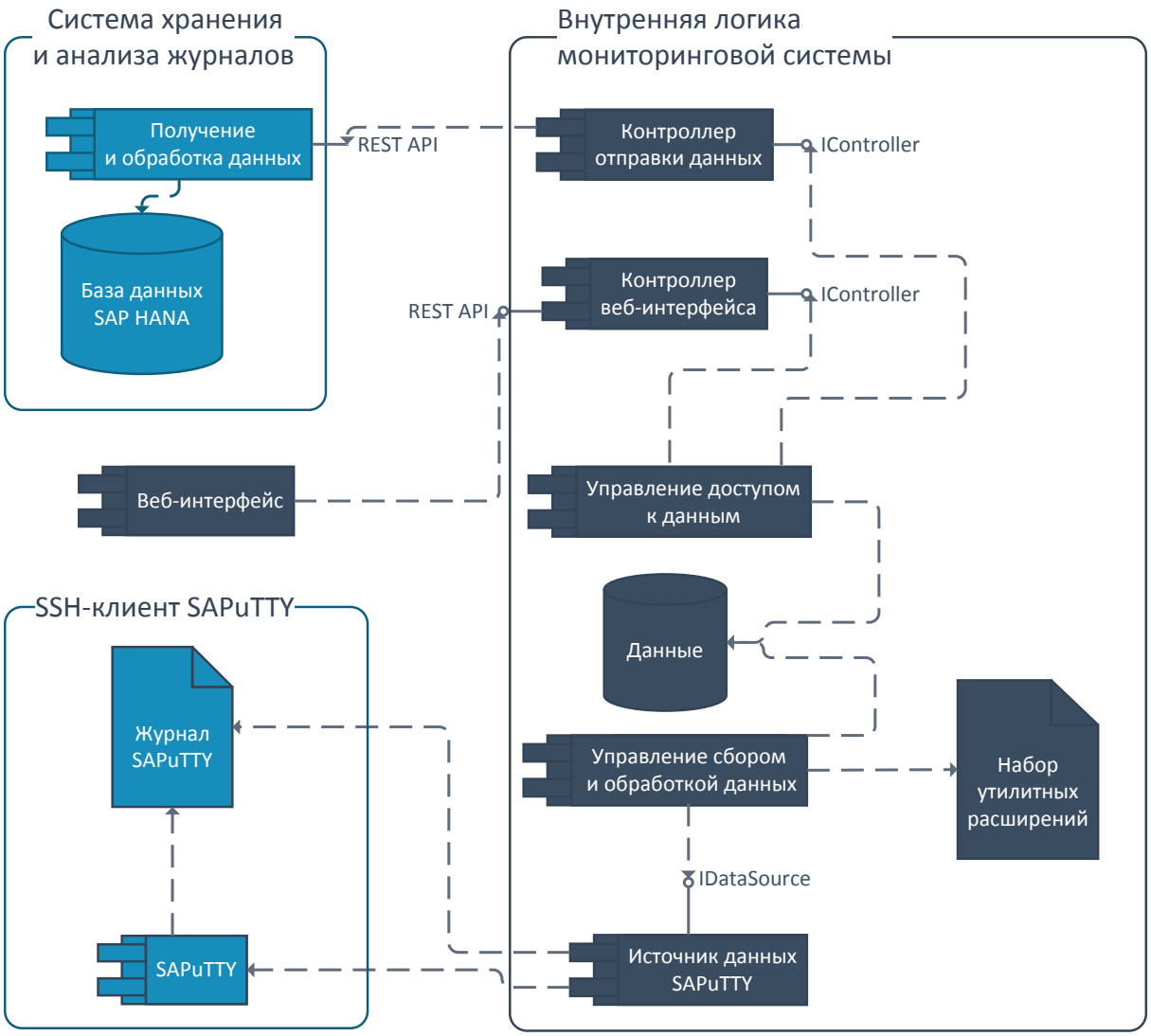

Рис. 2: Диаграмма компонентов архитектуры системы мониторинга, разработанной для компании SAP

сание архитектуры системы с некоторыми особенностями реализации.

4.1. Источники данных

Существует несколько способов получения мониторинговых данных. Одними из самых распространенных являются внешние базы данных, которые наполняются данными с помощью других систем, и подключение к серверу, например, по протоколу SSH с последующим запуском мониторинговых утилит. В задаче, для решения которой разрабаты-

вается эта система – мониторинг в режиме реального времени, компанией SAP используется способ с SSH-подключением. Стоит отметить, что как указано в Требовании 2.1, кроме прямого SSH-подключения необходимо обеспечить возможность сбора и отправки данных через SSH-клиент для ОС Windows SAPuTTY, который часто является единственно возможным способом подключения.

Чтобы обеспечить поддержку нескольких вариантов подключения, Источник данных спроектирован как адаптер между внешней системой, у которой можно получить мониторинговые данные по запросу, и мониторинговой системой. Такая система также позволяет не нарушать Требования 2.2, которое запрещает устанавливать на сервер, подвергаемый мониторингу стороннее программное обеспечение, которое могло бы самостоятельно, без запросов, отправлять данные в мониторинговую систему.

При прямом SSH-подключении можно использовать Источник данных, который получает мониторинговую информацию из SSH-библиотеки для платформы Java – JSch[9] или sshj[4] стандартными средствами. При необходимости же мониторинга с использованием внешнего SSH-клиента SAPuTTY (такой вариант изображен на рисунке 2), необходимо переключиться на Источник данных SAPuTTY, который способен отправлять команды в SAPuTTY посредством Windows API и читать данные из журнала SAPuTTY.

4.2. Обработка данных

После сбора данных производится их обработка, в ходе которой данные в формате простого текста преобразуются в формат, принимаемый потребителями данных – веб-интерфейсом и Системой хранения и анализа журналов. В обоих случаях это JSON-форматы, в которых мониторинговые данные сгруппированы необходимым для использования потребителем данных образом и добавлена необходимая служебная информация.

Для того, чтобы обеспечить максимальную расширяемость системы

относительно поддержки всевозможных мониторинговых утилит и реализовать возможность добавлять поддержку новых утилит без необходимости модифицировать и перекомпилировать исходный код системы для мониторинга, было выбрано представление утилитных расширений в виде исполняемых скриптов. Языком написания скриптов был определен JavaScript. Важной причиной использования этого языка для разработки утилитных расширений послужило то, что один из движков JavaScript – Nashorn[8], поставляется вместе с виртуальной машиной Java, что позволяет использовать его без необходимости установки интерпретатора и легко запускать исполнение скриптов на этом диалекте JavaScript из Java. Не менее важно, что JavaScript является распространенным языком программирования как вообще (7 место в индексе популярности языков программирования TIOBE[3] по данным на май 2017 года), так и внутри компании SAP.

Дополнительной особенностью диалекта JavaScript, используемого в Nashorn, является возможность использования классов и методов из языка Java. Это позволяет нейтрализовать некоторые недостатки языка JavaScript, например, сложности в работе с файловой системой, путем использования для таких целей методов языка Java.

Основная задача утилитного расширения (скрипта) состоит в преобразовании поступающей на вход мониторинговой информации в один из специфицированных форматов, чаще всего – в JSON определенной структуры. Основные составляющие скрипта – служебная информация и главная функция *parse()*, которая и должна осуществить преобразование, будучи вызванной из Java. В случае с написанием утилитного расширения для построения графиков по данным одной из мониторинговых утилит, эта функция будет содержать в себе фильтрацию данных и настройку их последующего отображения.

Для облегчения задачи написания утилитных расширений для инженеров была разработана JavaScript-библиотека, в которую вошли функции разбиения данных на блоки, удаления колонок с данными по имени или индексу и другие функции, необходимые для обработки мониторинговой информации.

4.3. Контроллеры

Контроллеры – это модули, задачей которых является доставка данных до их потребителей. В зависимости от системы, в которую доставляются данные, это может быть реализовано различными способами. Далее приводятся два описания контроллеров, реализованных в рамках разработки системы.

4.3.1. Контроллер веб-интерфейса

Задача этого контроллера – поставка данных в веб-интерфейс системы, где они будут отображены в виде графиков и таблиц, а также обеспечение возможности управлять процессом мониторинга. Контроллер реализован в формате веб-сервера, который предоставляет REST API для доступа к данным и настройки. Поддерживаются следующие запросы:

- запуск команды мониторинга;
- запрос последних результатов мониторинга указанной команды;
- запрос последних результатов мониторинга запущенной команды;
- запрос результатов мониторинга указанной команды за указанный период.

4.3.2. Контроллер отправки данных

Контроллер отправки данных отвечает за доставку снимков журналов мониторинга в Систему анализа и хранения журналов. Для этого он один раз за фиксированный промежуток времени (по умолчанию – 15 минут) с помощью POST-запроса отправляет массив новых мониторинговых JSON-данных в эту систему.

4.3.3. Система управления устаревшими данными

Обзор

Как уже было отмечено, в систему мониторинга, подключенную к серверу с большим количеством процессорных ядер и жестких дисков, поступает огромное количество данных. Для удовлетворения Требований 2.2 и 2.2, регулирующих расход оперативной и постоянной памяти системой, было принято решение создать систему отслеживания и удаления устаревших данных. Задача обнаружения устаревших данных состоит в том, чтобы найти мониторинговую информацию, которая уже не понадобится ни одному из подключенных к системе контроллеров. В конкретном случае, рассмотренном на рисунке 2 необходимо определить, какие данные уже были как запрошены пользовательским веб-интерфейсом, так и отправлены в Систему анализа и хранения журналов, после чего удалить их. В связи со спецификой предметной области мониторинга серверов в режиме реального времени отпадает необходимость сохранять данные, которые уже были отображены в пользовательском интерфейсе, на компьютере инженера.

Реализация

Когда контроллер считывает данные из хранилища, отмечается последний прочитанный им фрагмент данных. Все фрагменты данных, собранные ранее, помечаются для этого контроллера как устаревшие. Если фрагмент оказывается помечен для всех контроллеров, он удаляется.

4.4. Веб-интерфейс

С учетом Требования 2.2 был разработан прототип графического веб-интерфейса системы для мониторинга. Он получает веб-страницы с сервера и генерирует графики и таблицы на основе данных, которые получает из хранилища. Структурно интерфейс состоит из страниц выбора команд, отображения таблицы, отображения простого графика и отображения комплексного графика. При разработке использовалась популярная JavaScript-библиотека для манипулирования объектной моделью документа jQuery [13] и библиотека C3 [10] для отображе-

ния графиков, основанная на распространенной библиотеке D3 [12]. C3 выбрана как бесплатное решение с минималистичным современным дизайном, возможностью строить графики с временной осью, включать и отключать отображение параметров на графике, поддержкой подсказок при наведении и встроенной реализацией функций изменения масштаба и прокрутки.

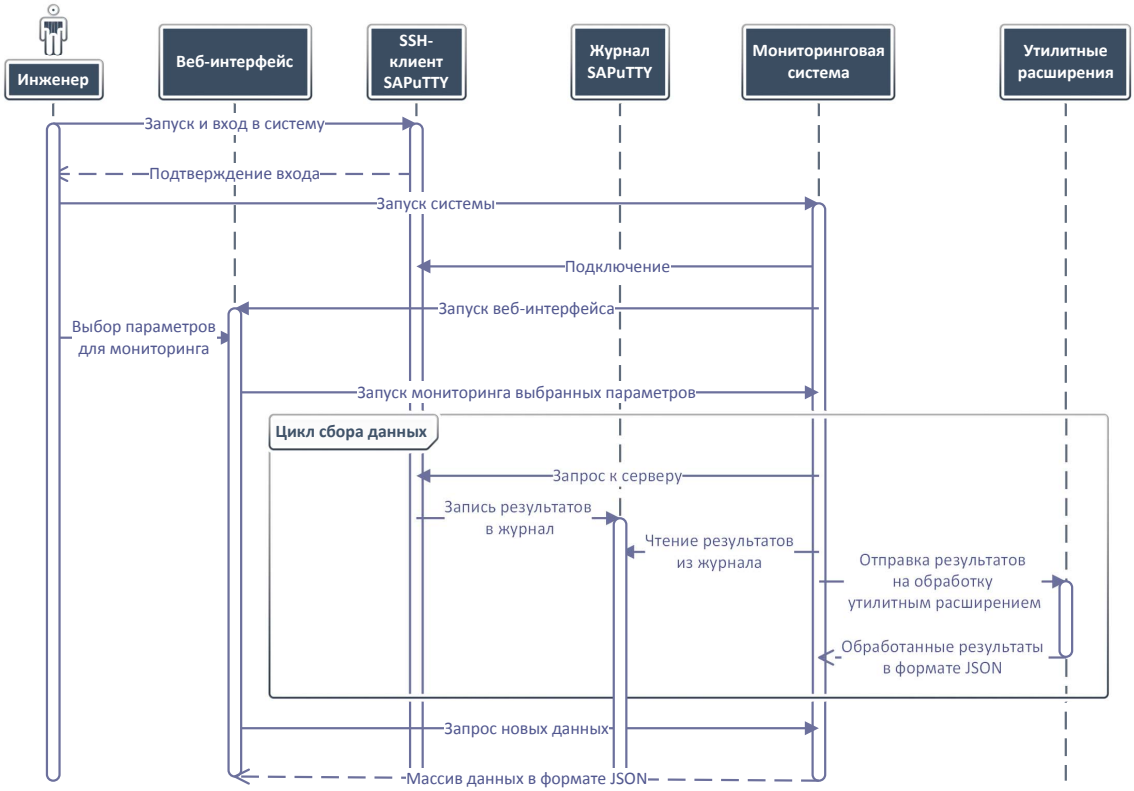

Рис. 3: Диаграмма последовательности для процесса мониторинга параметра, для которого существует утилитное расширение

Заключение

В заключение предлагается рассмотреть один из вариантов использования описанной системы мониторинга. На рисунке 3 изображена диаграмма последовательности для задачи запуска системы и мониторинга параметра, для которого уже реализовано утилитное расширение.

Инженер запускает SSH-клиент SAPuTTY, вводит имя пользователя и пароль, получает подтверждение входа. Затем ему необходимо запустить мониторинговую систему, настроенную на работу с SAPuTTY. Система подключается к SAPuTTY и запускает веб-интерфейс. В веб-интерфейсе инженер выбирает параметр, который необходимо отслеживать в режиме реального времени. В систему отправляется команда о необходимости начать мониторинг этого параметра с использованием соответствующего утилитного расширения. Система начинает цикл сбора данных. В его процессе через SAPuTTY отправляются мониторинговые команды к серверу, результат исполнения которых записывается в журнал программы. Данные считываются из журнала, подвергаются обработке утилитным расширением и складываются в хранилище. Веб-интерфейс периодически опрашивает систему о наличии новых данных, получает их из хранилища и отображает на экране.

5. Апробация прототипа

Система была протестирована в компании SAP (обособленное подразделение ООО «САП Лабс» в Санкт-Петербурге) и получила положительные отзывы инженеров. Была отмечена информативность графического интерфейса и высокая скорость разработки утилитных расширений. Ниже приведен один из отзывов инженеров компании SAP.

Пример отзыва

Разработанная система была опробована в опытно-промышленной эксплуатации при разработке параметров оптимизации ввода-вывода In-Memory базы данных SAP HANA для одного из крупнейших клиентов в регионе EMEA. Благодаря визуализации выявлены зависимости пропускной способности от косвенных параметров настройки ОС, неочевидные при анализе текстовых данных. Кроме того, модификация этих параметров «на лету» и изменение графиков ввода-вывода в режиме «онлайн» позволило подобрать их оптимальные сочетания.

Результаты апробации

Также в ходе апробации выяснилось, что первоначально выбранная для отображения графиков библиотека ChartJS [11] предоставляет недостаточно удобные способы детализации графиков и прокрутки. По этой причине для отображения графиков была применена библиотека C3, которая значительно лучше справилась с поставленными задачами, повысив дружелюбность системы к пользователю.

Заключение

В ходе выполнения данной выпускной квалификационной работы были достигнуты следующие результаты.

- Разработаны требования к системе для мониторинга серверов в режиме реального времени.
- Проведен анализ существующих решений в сфере мониторинга серверов.
- Разработана архитектура системы, основанная на принципе модульности.
- Реализован прототип системы, включающий следущие компоненты:
  - источники данных для внешнего и встроенного SSH-клиентов;
  - контроллеры для графического интерфейса и отправки данных в Систему хранения и анализа журналов;
  - утилитные расширения sar и pidstat;
  - графический веб-интерфейс.
- Прототип системы был опробован в подразделении компании SAP в Санкт-Петербурге и получил положительные отзывы инженеров.

Список литературы

[1] RFC 4251: The Secure Shell (SSH) Protocol Architecture. — URL: https://tools.ietf.org/html/rfc4251 (online; accessed: 30.04.2017).

[2] RFC 854: TELNET PROTOCOL SPECIFICATION. — URL: https://tools.ietf.org/html/rfc854 (online; accessed: 30.04.2017).

[3] Индекс популярности языков TIOBE. — URL: https://www.tiobe.com/tiobe-index/ (online; accessed: 22.05.2017).

[4] Исходный код SSH-библиотеки sshj. — URL: https://github.com/hierynomus/sshj (online; accessed: 22.05.2017).

[5] Исходный код программы kSar. — URL: https://sourceforge.net/projects/ksar/ (online; accessed: 26.04.2017).

[6] Описание утилиты pidstat на сайте Ubuntu manuals. — URL: http://manpages.ubuntu.com/manpages/zesty/en/man1/pidstat.1.html (online; accessed: 30.04.2017).

[7] Описание утилиты sar на сайте Ubuntu manuals. — URL: http://manpages.ubuntu.com/manpages/zesty/man1/sar.sysstat.1.html (online; accessed: 30.04.2017).

[8] Руководство пользователя Nashorn. — URL: https://docs.oracle.com/javase/8/docs/technotes/guides/scripting/nashorn/ (online; accessed: 22.05.2017).

[9] Сайт SSH-библиотеки Jsch. — URL: http://www.jcraft.com/jsch/ (online; accessed: 22.05.2017).

[10] Сайт библиотеки C3. — URL: http://c3js.org/gettingstarted.html (online; accessed: 22.05.2017).

[11] Сайт библиотеки ChartJS. — URL: http://www.chartjs.org/ (online; accessed: 22.05.2017).

[12] Сайт библиотеки D3. — URL: https://d3js.org/ (online; accessed: 22.05.2017).

[13] Сайт библиотеки jQuery. — URL: https://jquery.com/ (online; accessed: 22.05.2017).

[14] Сайт проекта Cacti. — URL: http://www.cacti.net/ (online; accessed: 26.04.2017).

[15] Сайт проекта Grafana. — URL: https://grafana.com/ (online; accessed: 26.04.2017).

[16] Сайт проекта Kibana. — URL: https://www.elastic.co/products/kibana (online; accessed: 26.04.2017).

[17] Сайт проекта Monit. — URL: https://mmonit.com/monit/ (online; accessed: 26.04.2017).

[18] Сайт проекта Nagios. — URL: https://www.nagios.org/ (online; accessed: 26.04.2017).

[19] Сайт проекта OpenUI5. — URL: http://openui5.org/index.html (online; accessed: 22.05.2017).

[20] Сайт проекта PuTTY. — URL: http://www.putty.org/ (online; accessed: 22.05.2017).

[21] Сайт проекта Zabbix. — URL: http://www.zabbix.com/ (online; accessed: 26.04.2017).

[22] Сайт фреймворка SAPUI5. — URL: https://sapui5.hana.ondemand.com/ (online; accessed: 22.05.2017).