ЦЕНТРАЛЬНЫЙ ПУЛЬТ



Описание автоматизируемых функций

Содержание

1. Исходные данные 3
1.1. Материалы, используемые при разработке функциональной части АС 3
1.2. Особенности объекта управления 3
1.3. Системы управления, взаимосвязанные с разрабатываемой АС 3
1.4. Описание информационной модели объекта 4
2. Цели АС и автоматизируемые функции 4
3. Характеристика функциональной структуры 5
3.1. Перечень подсистем АС 5
3.1.1. Подсистема "Saymon-server" 6
3.1.2. Подсистема "СУБД" 6
3.1.3. Подсистема "Saymon-agent" 7
3.1.4. Подсистема "Клиент" 7
3.2. Требования к временному регламенту и характеристикам 7
4. Типовые решения 8

1. Исходные данные

Центральный Пульт — это платформа для визуализации и мониторинга работы сети, оборудования, приложений и служб.

1.1. Материалы, используемые при разработке функциональной части АС

При разработке функциональной части проекта были использованы следующие материалы:

- концепция АС "Центральный Пульт",
- техническое задание по разработке пилотного проекта АС "Центральный Пульт",
- бизнес-требования от потенциальных пользователей,
- данные, полученные в результате опроса лиц, ответственных за разработку системы,
- схема функциональной структуры.

1.2. Особенности объекта управления

При разработке АС "Центральный Пульт" учитывался ряд особенностей, которые повлияли на проектные решения. К этим особенностям относится увеличение числа партнёров из различных сфер деятельности, что влечёт за собой регулярное усовершенствование функционала, интеграцию с локальными и вспомогательными АС партнёров, введение в эксплуатацию новых способов сбора данных.

В связи с этим закладываемые в АС "Центральный Пульт" проектные решения должны обладать достаточной степенью гибкости и масштабируемости.

1.3. Системы управления, взаимосвязанные с разрабатываемой АС

Система состоит из следующих логических подсистем:

- **Server** — централизованный сервер, на котором хранится и анализируется информация, полученная от агентов, а затем отдаётся клиенту. Также здесь содержится информация об учётных записях пользователей.
- **СУБД** (MongoDB, OpenTSDB) — совокупность программных средств, предназначенных для создания, использования и управления базами данных.
- **Agent** — множество агентов системы, установленных на узлах инфраструктуры и собирающих информацию по ним. Полученные данные периодически отправляются в кэши затем анализируются сервером.
- **Клиент** — тонкий web-клиент системы и клиенты для мобильных операционных систем Android и iOS.

1.4. Описание информационной модели объекта

Информационное обеспечение АС "Центральный Пульт" включает в себя внутримашинное и внемашинное информационное обеспечение.

В состав внемашинного информационного обеспечения входит система документации.

В состав внутримашинного информационного обеспечения входят:

- центральное хранилище данных,
- центр обработки запросов клиента,
- набор NodeJS-приложений,
- сетевое журналируемое хранилище данных Redis,
- система управления базами данных MongoDB,
- база данных временных рядов OpenTSDB.

2. Цели АС и автоматизируемые функции

Цели создания платформы:

- упростить сбор и анализ информации;
- ускорить процесс обработки данных и сделать его автоматизированным;
- визуализировать полученные и обработанные данные;
- обеспечить беспрерывное хранение информации;
- автоматически уведомлять пользователей о состояниях объектов удобным для них способом;
- автоматически исправлять аварийные ситуации;
- управлять объектами мониторинга.

Платформа "Центральный Пульт" обеспечивает реализацию следующих функций:

- представление практически любого объекта окружающего мира в качестве объекта мониторинга;
- смена состояния объекта в соответствии с заданными условиями;
- автоматическое выполнение предопределённых действий;
- хранение оригинальных значений показателей за промежутки времени;
- обеспечение анализа в табличной и графической формах;
- представление объектов, согласно их географическому месторасположению;
- использование гибкого механизма оповещений;

- группировка объектов по заданным общим критериям;
- преобразование данных в компактный вид и их экспорт;
- прикрепление документации и отображение свойств объектов;
- управление административными настройками из web-интерфейса.

3. Характеристика функциональной структуры

Программное обеспечение платформы "Центральный Пульт" имеет открытые API-интерфейсы, которые обеспечивают информационную совместимость системы и возможность интеграции с другими автоматизированными системами.

3.1. Перечень подсистем АС

Система состоит из следующих логических подсистем:

1. Saymon-server.

   Подсистема обеспечивает выполнение следующих функций:

   - получение уведомлений об ошибках передачи данных;
   - настройка и контроль доступа к объектам системы;
   - контроль целостности данных;
   - управление работой других подсистем;
   - реагирование на возникновение аварийных ситуаций;
   - преобразование данных в требуемый системой формат.

2. СУБД (MongoDB, OpenTSDB).

   Подсистема обеспечивает выполнение следующих функций:

   - хранение данных;
   - загрузка полученных данных в систему;
   - резервное копирование;
   - восстановление базы данных после сбоев;
   - журналирование переданной и полученной информации от сервера к клиенту и наоборот.

3. Saymon-agent.

   Подсистема обеспечивает выполнение следующих функций:

   - сбор информации на выбранном объекте мониторинга;

- анализ полученной информации;
- выполнение пользовательских скриптов;
- осуществление как пассивного, так и активного мониторинга;
- отправка обработанных данных серверу.

4. Клиент (Web, Android, IOS).

   Подсистема обеспечивает выполнение следующих функций:

   - настройка условий мониторинга;
   - создание и выбор объектов;
   - настройка объектов мониторинга;
   - управление пользователями;
   - кастомизация интерфейса;
   - просмотр и изменение текущих состояний;
   - построение графиков.

3.1.1. Подсистема "Saymon-server"

Saymon-server — набор NodeJS-приложений, которые взаимодействуют между собой и со всеми остальными компонентами.

Выделяются следующие составляющие подсистемы:

- **Web-сервер** — это HTTP-сервер, обслуживающий запросы клиента. Он отдаёт статические элементы web-интерфейса (HTML, JavaScript, CSS), а также проксирует запросы от клиента к REST-серверу. В качестве web-сервера используется NGINX.
- **REST-сервер** — серверный компонент, обрабатывающий REST-запросы от JavaScript-компонентов на тонком клиенте. Через REST API клиент получает всю информацию об инфраструктуре, а также производит манипуляции с инфраструктурой. REST-сервер реализован в виде отдельного NodeJS-приложения.
- **Сервер данных** — производит анализ данных, поступивших от агентов. В частности управляет логикой смены состояний у объектов и связей. Сервер данных реализован в виде отдельного NodeJS-приложения.
- **Кэш в памяти (In-Memory кэш)** — NoSQL-хранилище данных типа "ключ-значение". Хранит базу данных в оперативной памяти, благодаря чему возможен быстрый доступ к данным и их быстрая обработка. В качестве In-Memory кэша используется Redis.

3.1.2. Подсистема "СУБД"

СУБД — подсистема, которая отвечает за хранение и передачу данных между участниками

обмена.

СУБД-платформы включает в себя два компонента:

- **MongoDB** — система управления базами данных, классифицированная как NoSQL;
- **OpenTSDB** — база данных временных рядов (Time series).

СУБД имеет API-интерфейс, который позволяет загружать данные напрямую, не дожидаясь ответа от сервера.

3.1.3. Подсистема "Saymon-agent"

Saymon-agent — компонент системы, собирающий данные о наблюдаемом узле и расположенных на нём объектах, а также связях данного узла.

Данные, собранные агентом, периодически отправляются в In-Memory кэш и затем анализируются сервером. Агент реализован в виде Java-приложения.

Подсистема позволяет осуществлять как активный, так и пассивный мониторинг.

3.1.4. Подсистема "Клиент"

Клиент — основной инструмент конечного пользователя.

Подсистема "Клиент" может быть представлена двумя способами:

- через браузер Google Chrome версии не ниже 58.0;

WARNING	*Стабильность работы клиента системы в браузерах Яндекс, Safari, Opera, FireFox не гарантируется.*

- через мобильное приложение для Android или iOS.

3.2. Требования к временному регламенту и характеристикам

Требования к временному регламенту и характеристикам процесса реализации автоматизированных функций соответствует общим требованиям к автоматизированной системе, изложенным ниже.

При проектировании и разработке подсистем учитывались следующие общие требования:

- работа с программным обеспечением должна осуществляться пользователями и администраторами системы;
- доступ к функционалу должен осуществляться в соответствии с выделенными правами и уровнем доступа пользователей;
- происходящие события должны фиксироваться в системном журнале с указанием

типа событий, времени его выполнения и имени учётной записи пользователя, инициировавшего его.

Теоретическая нагрузка, учитывавшаяся при проектировании программного обеспечения:

Тестируемый процесс	Показатель	Средняя величина
SNMP-трапы	Максимальная пропускная способность агента	12 300 сообщения/с
	Максимальная пропускная способность сервера	3 800 сообщения/с
	Задержка	12 мс
Запись пришедших данных	Максимальная пропускная способность	45 000 события/с
	Задержка	1 мс
Уведомления о данных датчиков на удалённый агент	Максимальная пропускная способность	2 150 события/с
	Задержка	12 мс
Исторические данные	Пропускная способность	50 000 метрики/с
	Скорость отображения данных	83 000 метрики/с
	Среднее время отклика при записи данных	201 мс
	Среднее время отклика при отображении данных	2 мс
	Размер записи	6 байт
	Объём исторических данных за один год по одной метрике	3 Мбайт
	Объём использованной памяти агента	55 Мбайт
	Скорость отображения данных	700 запросы/с
	Среднее время отклика	2 мс

4. Типовые решения

В процессе разработки системы были использованы следующие типовые решения:

- использование архитектуры "Клиент-Сервер-Агент" для построения системы мониторинга;
- использование архитектуры MVC (Model-View-Controller) для построения Web-приложения;
- использование JSON-формата для передачи данных между клиентом, сервером и агентом;
- использование шаблона проектирования Message Bus для обмена сообщениями между модулями сервера.