|  |  |
| --- | --- |
| **#** | **Речь** |
| 1 | Здравствуйте уважаемые члены Аттестационной Комиисии, сейчас вашему вниманию будет представлен дипломный проект, посвященный проетированию и реализации распределенной системы мониторинга. В проекте принимало участие два человека. В моем докладе будет освещена модель системы, ее архитектура а также рассмотрена реализация основополагающего компонента – службы мониторинга. Во втором докладе будет рассмотрены структура системы визуализации и управления, реализация ее подсистем, а также интерфейс програмирования модулей.  На текущем этапе информационного развития общества все более актуальными становятся проблемы обпеспечения должного уровня качества протекания информационных процессов. Для решения подобных задач существует специализированный класс программных систем – именуемый системами мониторинга, который появился сравнительно недавно и уже успел эволюионировать из примтивных инструментов администратрирования до универсальных коробочных решений уровня предприятий.  Не смотря на насыщенность класса, новые инструменты прододжают появляться, определяя его новые вектора развития. Данный дипломный проект представляет собой развитие нового подкласса систем мониторинга – распеделенных систем с динамически-расширяемым функционалом. |
| 2 | Рассмотрим выдвинутую нами модель требований к соврмеменным системам мониторинга. При этом, мало того, что система должна удовлетворять перечисленным требованиям, она должна еще и выдерживать динамику их изменений. Итак, отказоустойчивость системы подразумевает ее возможноть нормально функционировать после выхода из строя одной или более значимых компонент. Под масштабируемостью понимается легкость подключения к системе новых узлов. Эффективность и применимость системы мониторинга определяется возможностью ее внедрения в произвольную сферу эксплуатации, а также легкость сопровождения и администрирования. Наконец, расширяемость системы определяется наличием механизмов и примитивов, позволяющих динамически наращивать функционал системы через разработку и интреграцию дополнительных компоненов. |
| 3 | В процессе построения модели и архитеттуры системы, нами были изучены наиболее популярные системы мониторинга и произведена их классификация. Можно выделить клиент-серверны и распределенные системы, взависимоти от методологии построения транспорного уровня. А также расширяемые и не расширяемые системы в замисимости от наличия механизмов динамической загрузки и исполнения модулей или плагинов мониторинга. Рассмотррение аналогов, позволило сделать вывод о необходимости появления целого класса новых систем – решений на основе распределенных систем с динамически расширяемым функционалом. |
| 4 | Для рассмотрения особенностей предлагаемой модели определимся с базовыми понятиями. Предметная область описывается с помошью терминов Вычислительный узел, служба, хранилище данных и задача мониторинга. Вычислительный узел – это устройство способное выполнять код слажбы. В свою очередь, служба – это активная сущность, запущенная на узле и наблюдающая за его состоянием. Хранилище – пассиваня сущность, предоставлющая службе ресуры для приема сообщений. И наконец задача мониторинга – определяется как шаблонная проблема получения и анализа некоторой информации о удаленном узле.  Рассмотренные понятия позволяют описать систему с точки зрения естественного языка. Обратимся к формальному описанию распределенной системы мониторинга, разработанному нами в рамках дипломного проекта. |
| 5 | Ключевым моментом во всем процессе мониторинга является его цель. Цель описывается множеством задач мониторинга. В предлагаемой модели системы модуль является результатом отображения задачи мониторинга из предметной области в программную среду и представляет собой сущность, характеризующуюся: возможностью исполнения в ОС, входными данными, выходными данными, интерфейсом, реализацией. Понятие модуля было введено нами для обеспечения целевой системы механизмами динамического расширения функционала, которые наделяют ее дополнительным поведением и позволяют удовлетворять требованиям к применимости и эффективности. |
| 6 | Из теории распределенных систем, известно понятие глобального состояния системы, которое определяется графом связности узлов, расположением запущенных экземпляров модулей и нагрузкой на узлы. В предлагаемой модели роль распределенного модуля играет служба мониторинга. Нагрузки на узел — индекс производительности узла, определящий по некоторой шкале текущее количество свободных ресурсов системы. Статус элемента распределенной системы для службы мониторинга а придает ей следующие особенности: масштабируемость (возможность запуска дополнительного экземпляра), сериализуемость (возможность сохранения его внутреннего состояния), переносимость (возможность переноса службы в распределенной среде с сохранением ее внутреннего состояния).  Служба может находится в активном или пассивном состоянии. Активное состояние наделяет службу дополнительными обязанностями по отношению к соседним узлам: планирование запусков модулей мониторинга; мониторинг и диспетчеризация процессов исполнения модулей мониторинга; предоставление промежуточного хранилища для пересылаемых сообщений. |
| 7 | На основе предолженной модели, нами была спроектирована архитектура распределенной системы мониторинга, состоящая из четырех основных компонентов – службы мониторинга, менеджера модулей, интерфейса программирования и панели управления.  Взаимодействие компонентов системы осуществляется посредством вызовов RPC-методов платформы среднего слоя.  Для понимая структуры системы можно рассмотреть типичный вариант ее использования. Пользователь, в терминах предметной области, с использованием интерфейса программирвоания модулей, разрабатывает необходимый модуль мониторинга на поддерживаемом платформой языке программирования и через панель управления разворачивает его на наобходимое количество узлов в сети. В свою очередь, служба, запущенная на узле, инициализирует расписание запуска модуля, набор входных параметеров и запускает планировщик ассоциированный с развернутым модулем. Планировщик, инкапсулирует запуск модуля во внутрисистенмное собитие и передает его менеджеру модулей, который уже и осуществляет запуск, передачу параметров и возращение результата. Полученный результат сохраняется в хранилище и отображается в панели управления.  Моим вкладом в проект явлются проктирование и реализации службы мониторинга и менеджера модулей. |
| 8 | Основополагающей компонентой распределенной системы мониторинга является – служба. Служба мониторинга представляет собой программный комплекс, обеспечивающий использование ресурсов вычислительной среды, адресацию, поддержание поведения распределенной системы мониторинга (модулей мониторинга, распределенной коммуникации, программной системы в целом).  Служба состоит из двух основных подсистем – исполнительной и транспортной, запускаемых и функционируемых в рамках неделимой программной платформы или ядра. Исполнительная подсистема обеспечивает планирование и запуск модулей мониторинга. В свою очередь, транспортная реализует распределенную модель сетевого взаимодействия.  Наконец, служба мониторинга представляет собой в полном смысле распределенное приложение. |
| 9 | Базовым компонентом службы мониторинга является ядро, реализующее динамически расширяемую программную платформу, в рамках которой запускаются и функционируют основные подсистемы службы. Кроме того, ядро обеспечивает работу загружаемых компонентов службы, и содержит базовые механизмы и примитивы для их взаимодействия и синхронизации.  В качестве механизмов динамического расширения функционала мною было введено понятие драйвера ядра, который представляет собой сущность, реализующую некоторую часть общего поведения системы.  Взаимодействие драйверов не осуществляется напрямую. Вместо этого используется генерация, обработка и передача специальных событий ядру. Событие ядра инкапсулирует тип случившейся внутрисистемной ситуации и содержит необходимые параметры и структуры для ее корректной обработки.  Основная идея предлагаемого подхода при разработе ядра службы заключается в общем использовании одного драйвера несколькими службами одновременно. Для реализации данного поведения мною были разработаны так называемые адаптеры ядра.  Для удаленного взаимодействия между узлами применяются сессии, которые бывают двух видов – режима пользователя и режима ядра. Сессии режима пользователя устанавливаются между ядром и панелью управления. Режима ядра – между двумя удаленными службами. |
| 10 | Мною было решено описывать поведение ядра в терминах конечных автоматов. Для этого было введено пять состояний ядра службы мониторинга – активное, пассивное, неопределенное, сетевое и автономное.  Сетевое и автономное состояния характеризуются наличием или отсутсвием сетевой подсистемы узла. Активное и пассивное состояния соотсвествуют формальным состояниям слжубы, описанным в разработанной модели.  Переходы между сосояниями осуществляют с помощью механизмов обработки и генерации внутрисистемных событий. Источниками этих событий могу быть как дайрверы ядра так и внутренние ее подсистемы. |
| 11 | Распределенная модель сетевого взаимодействия инкапсулирована в тренспортной подсистеме службы мониторинга, которая представляет собой совокупность подсистем ядра, драйверов транспортного уровня, менеджера сессий, а также мнопоточных распределенных алгоритмов.  Подсистема транспортного уровня службы реализует управление удаленными сессиями, мониторинг сетевой активности и именование распределенных объектов, адресацию и балансировку нагрузки в распределенной среде а также механизмы и алгоритмы выбора лидеров. |
| 12 | За основу алгоритма выбора лидера мною был взят классический алгоритм Чанди-Робертса, который основывается на кольцевой топологии сети с односторонней передачей данных. На его базе был разработан более современный алгоритм с применением широковещательных запросов и негарантированной доставки сообщений.  Согласно алгоритму, выбор лидера осуществляется после истечения определенного периода времени, на протяжении которого система находится в режиме исследования или в режиме кеширования приходящих от узлов контекстных пакетов.  Процесс выбора лидера представляет собой циклическую последовательность логических действий, заканчивающуюся после перехода узла в новое состояние. Это - выбор узла (и его удаление) с максимальным рейтингом из кеша, проверка доступности выбранного узла и подключение к лидеру и переход в активное или пассивное состояние если подключение прошло успешно. |
| 13 | В процессе реализации инструментов динамического расширения функционала мною была разработана исполнительная подсистема службы мониторинга, которая реализует основные механизмы исполнения модулей мониторинга и состоит из драйверов подсистемы исполнения и менеджера модулей. Функционирование подсистемы исполнения обеспечивается ядром службы.  Исполнительная подсистема реализует планирование и запуск модулей мониторинга, обработку их результатов, а также высокоуровневые операции развертывания модулей на удаленных узлах. |
| 14 | В качестве механизма планирования и запуска модулей, выступает планировщик подсистемы исполнения, предствляющий собой запускаемый драйвер ядра. Планироващик характеризуется самостоятельным программным потоком, который запускается при переходе ядра в активное состояние и останавливается при выходе из него.  Мною было решено использовать модель делегирования обязанностей по планированию запусков модулей от дочерних узлов – родительским. При таком подходе, планировщик, запущенный на родительском узле, является общим для дочерних. В основе реализации модели делигирования лежат основные свойства распределенных объектов, такие как сериализуемость и переносимость.  Кроме того, планировщик поддерживает два вида запусков модулей – по расписанию и принудительно. |
| 15 | Для управления модулями в рамках запускающей службы, мною был спроектирован и реализован менеджер модулей, представляющий собой обособленное приложение на языке Python, взаимодействующее с овновной службой через RPC-протокол. Менеджер модулей реализует большинство низкоуровневых операций при работе с модулями, таких как: а) генерация кода каркаса модуля, включающая инициализацию окружения, передачу и возврат значений; б) исполнение модуля в операционной среде а также в) платформенно зависимые файловые операции, выполняемые при развертывании модулей. |
| 16 | Все расмотренные ранее подсистемы и компоненты были реализованы мною на основе объектно-ориетированного подхода с применением шаблонов проектирования и поддержкой парадигм «низкая связность» и «выское зацепление». Реализованные в рамках проекта классы, представляют собой объединение пакетов языка Java, каждый из которых соовтествует подсистеме или части общей программной платформы. Среди наиболее удачных архитектурных решений можно выделить реализацию модели поведения системы в терминах конечного автомата; обмен информацией между подсистемами и пакетами на основе генерации и оработки событий, а также реализацию механизмов расшериня функционала системы на основе динамически загружаемых драйверов с унифицированным интерфейсом. |
| 17 | В результате работы над дипломным проектом, нами была разработана и формализована модель распределенной системы мониторинга с динамически расширяемым функционалом. На базе это модели была спроектирована архитектура высоконагруженной распределенной системы мониторинга, с примненением паттернов. На основе проекта системы, нами был реализован каркас распределенной системы мониторинга, с применением современных технологий программирования распределенных ситем на основе платформ среднего слоя, отвечающая всем предъявляемымм требованиям. В часности, мною были реализованы служба мониторинга включающая подсистемы транспорта и исполнения и запускаемые в рамках неделимой платформы или ядра, а также менеджер модулей, реализующий кодогенерацию, низукоуровневыне операции с файловой системой и исполнение модулей мониторинга. |
| 18 | Можно выделить следующие пути развития проекта. Во-первых, для наделения системы коробочным функционалом, требуется – разработка набора шаблонных модулей мониторинга, для решения круга повседневных задач, таких как анализ сететвого трафика, загрузка и температура процессора, количество свободной памяти и т.п. Во-вторых, совершенствование компонентов и оптмизация алгоритмов базовой платформы системы позволит повысить эффектиность ее использования и сократить накладные расходы полезной части. Наконец, полномасштабное внедрение и нагрузочное тестирование на базе существующей инфраструктуры лаборатории МикроЭВМ АлтГТУ позволит оценить применимость и полезность системы в условиях реальной эксплуатации на примере, автоматизации процессов мониторига учебного оборудования. |
| 19 | На этом, мой доклад заканчивается. Спасибо за внимание. Я готов ответить на ваши вопросы. |