|  |  |
| --- | --- |
| **#** | **Речь** |
| 1 | Здравствуйте уважаемые члены Аттестационной Комиисии, сейчас вашему вниманию будет представлен дипломный проект, посвященный проетированию и реализации каркаса 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 докладе моего коллеги будет рассмотрены структура системы визуализации и управления, реализация ее подсистем, а также интерфейс програмирования модулей.  На текущем этапе информатизации общества все более актуальными становятся проблемы обпеспечения должного уровня качества протекания информационных процессов. Для решения подобных задач существуют так называемые системы мониторинга. Этот класс систем появился сравнительно недавно, но уже успел эволюионировать из примтивных инструментов администратрирования до универсальных коробочных решений уровня предприятий.  Не смотря на насыщенность класса, новые инструменты прододжают появляться, определяя новые вектора развития. Данный дипломный проект представляет собой развитие нового подкласса инструментов мониторинга – распеделенных систем с динамически-расширяемым функционалом. |
| 2 | На этапе формирования требований к проекту, стало понятным, что будущая истема должна не только удовлетворять перечисленным требованиям, но и выдерживать динамику их изменений. Поэтому нами были сформулированы следующие требования. Соверменная система мониторинга должна быть отказоустойчивой, масштабируемой, эффективной, применимой и наконец расширяемой.  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 определяются возможностью ее внедрения в произвольную сферу эксплуатации, а также легкостью сопровождения и администрирования. Наконец, расширяемость системы определяется наличием механизмов и примитивов, позволяющих динамически наращивать функционал системы через разработку и интреграцию дополнительных компонент. |
| 3 | В процессе построения модели и архитеттуры системы, нами были изучены наиболее популярные решения в области 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 загрузки и исполнения модулей). Рассмотррение аналогов, позволило сделать вывод о необходимости появления целого класса инструментов мониторинга – решений на основе распределенных систем с расширяемым функционалом. |
| 4 | Для рассмотрения особенностей предлагаемой модели определимся с базовыми понятиями предметной области. Вычислительный узел – это програмно-аппаратное устройство способное загружать в память, а затем выполнять код службы. В свою очередь, служба – это активная сущность, запущенная на узле, непререывно наблюдающая за его состояние и сохраняющая в хранилище сообщения об изменении этого состояния. 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. |
| 5 | Рассмотрим формальное описание модели системы, представленое с помощью абстракции модуля и понятия глобального состояния.  Ключевым моментом во всем процессе мониторинга является его цель. Цель описывается множеством задач. В предлагаемой модели модуль мониторинга является результатом отображения задачи из предметной области в программную среду и представляет собой сущность, характеризующуюся: возможностью исполнения в ОС, входными данными, выходными данными (представляющими собой вектора произвольных объектов), интерфейсом и реализацией. Понятие модуля было введено нами для обеспечения целевой системы механизмами динамического расширения функционала, а следовательно для удовлетворения требованиям к эффективности и расширяемости. |
| 6 | Из теории распределенных систем, известно понятие глобального состояния, 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; мониторинг и диспетчеризация процессов исполнения; а также предоставление промежуточного хранилища для пересылаемых сообщений.  Испольлозование свойств распределенных сиситем в процессе эксплуатации, позволяет удволтеворить ранее выдвинутым требованиям к отказоустойчивости и масшиабируемости. |
| 7 | На основе формальной модели, нами была спроектирована архитектура распределенной системы мониторинга, состоящая из четырех компонентов – службы, менеджера модулей, интерфейса программирования и панели управления. Взаимодействие компонентов системы осуществляется посредством вызовов RPC-методов платформы среднего слоя.  Для понимая общей структуры системы можно рассмотреть типичный вариант ее использования. Пользователь, в терминах предметной области, согласно интерфейсу программирвоания, разрабатывает требуемый модуль мониторинга на поддерживаемом платформой языке программирования и через панель управления разворачивает его на наобходимое количество узлов. В свою очередь, служба, запущенная на узле, инициализирует расписание запуска , набор входных параметров и оповещает менеджера модулей о необходимости запуска исполняемой сущности. Менеджер модулей инициализирует откружени, передает и возвращает параметры и осуществляет запуск. Полученный результат сохраняется в хранилище и отображается в панели управления.  Моим вкладом в проект явлются проектирование и реализации службы мониторинга и менеджера модулей. |
| 8 | Основополагающей компонентой распределенной системы мониторинга является – служба. Служба мониторинга представляет собой программный комплекс написанный на языке Java, обеспечивающий использование ресурсов вычислительной среды, адресацию и поддержание поведения системы в целом.  Служба состоит из двух основных подсистем – исполнительной и транспортной, функционирующих в рамках неделимой программной платформы или ядра. Исполнительная подсистема обеспечивает планирование и запуск модулей мониторинга. В свою очередь, транспортная реализует распределенную модель сетевого взаимодействия.  Служба мониторинга представляет собой в полном смысле распределенное приложение. |
| 9 | Базовой компонентой 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, 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, инкапсулирующую некоторую часть общего поведения системы.  Взаимодействие драйверов не осуществляется напрямую. Вместо этого используется генерация, обработка и передача внутрисистемных событий. Событие 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 одновременном использовании одного драйвера несколькими службами. Для реализации данного поведения мною были разработаны так называемые адаптеры ядра.  Для удаленного взаимодействия между узлами применяются сессии режима пользователя и сессии режима ядра. Первые устанавливаются между ядром и панелью управления. Вторые – между двумя удаленными службами. |
| 10 | Мною было решено описывать поведение ядра в терминах конечных автоматов. Для этого было введено пять – активное, пассивное, неопределенное, сетевое и автономное.  Сетевое и автономное состояния характеризуются наличием или отсутсвием сетевой подсистемы узла. Активное и пассивное состояния соотсвествуют формальным состояниям службы, описанным в разработанной модели. Наконец, неопределенное состояние характеризуется отсутсвием информации о подсистеме окружения.  Переходы между сосояниями осуществляют с помощью механизмов обработки и генерации внутрисистемных событий. Источниками этих событий могу выступать как дайрверы ядра так и внутренние ее подсистемы. |
| 11 | Распределенная модель сетевого взаимодействия инкапсулирована в транспортной 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 базируется на кольцевой топологии сети с односторонней передачей данных. На его основе 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 кеширования контекстных пакетов.  Процесс выбора лидера представляет собой циклическую последовательность логических действий, заканчивающуюся при переходе узла в новое состояние. Это - выбор узла (и его удаление) с максимальным рейтингом из кеша, проверка доступности выбранного узла и подключение к лидеру, а также 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 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 инструмента 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 разделяемым ресурсом 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 так называемый менеджер модулей, представляющий собой обособленное приложение написанное на языке Python, взаимодействующее с основной 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 программных пакетов, каждый из которых соовтествует той или иной подсистеме или части общей программной платформы. Среди наиболее удачных архитектурных решений можно выделить реализацию модели поведения системы в на основе 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 модели спроектирована архитектура высоконагруженной распределенной системы, с примненением паттернов. На основе проекта, реализован каркас распределенной системы мониторинга, с применением современных технологий программирования распределенных ситем на основе библиотек среднего слоя. В часности, мною были реализованы служба мониторинга включающая подсистемы транспорта, исполнения и непосредтсвенно ядр, 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, и т.п. Во-вторых, совершенствование компонентов и оптмизация алгоритмов базовой платформы системы позволит повысить эффектиность ее использования и сократить накладные расходы в работе 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, у меня все. Спасибо за Ваше внимание. Я готов ответить на возникшие вопросы. |