|  |  |
| --- | --- |
| **Номер слайда** | **Речь** |
| 1 | Я надесь, что все присутсвуюещие здесь, что такое системы мониторинга и даже пользовались ими. Если ктото думает что не пользовался – я раскрою ему секрет – мы пользуемся системами мониторинга каждый день, каждый час, каждую секунду. Мы делаем телефонные звонки, проверяем почту, стоим на светофоре по дороге наработу или набираем что-то в поисковой строке Google. Каждый раз при этом, мы неявно пользуемся той или иной системой мониторинга.  Системы мониторинга, как сектор ПО, были придуманы сравнительно давно. Пажалуй первая широко известная миру система – команда ping (определяющая доспуность узла). Сейас иструменты стали сложнее, умнее и дороже. Разработка подобных систем – удовольствие очень ресурсоемкое и лишь крупные софтверные компании могу себе это позволить. Например, Cisco Systems с их CiscoWorks. Тем не менее системы мониторинга уверено выходят на рынок ПО и польльзуются все большей поплуярностью среди IT-специалистов. В продобных системах зинтересованы промышленный, научный и даже бизнес сектора. В данном докладе рассматривается прелашаемая архитектура распределенной системы мониторинга общего назначения, ее основные компоненты и принципы экплуатации. |
| 2 | Прежде чем рассмотреть требования обратимся к самому понятию систем мониторига. Очевидно, что предназначением сисем мониторинга явлется систематический сбор и анализ информации о протекающийх в информациинной среде процессов. Полученный в результате мониторинга данные могу быть использованы как для улучшения процесса приниятия решений, так и для выявления узких мест исследуемой среды.  Итак, рассмотрим основные требования, которые предъявляются к современным системам мониторинга. Причем, мало того, что система должна удовлетворять этим требованиям, она должна еще и выдерживать динамику их изменений. Например, требования к функционалу системы изначально определяются предметной областью эксплуатации, однако предсказать, какие задачи будут возложены на систему, в процессе ее использования, практически невозможно. Отказоустойчивость. Совершенно очевидно, что система мониторинга, за состоянием которой, также требуется пристальный присмотр практически бесполезна. Поэтому надежность решения — это определяющий фактор, его использования. Кроме того, это свойство также подвержено изменению в процессе использования системы. В одной ситуации требуется обеспечить систему необходимым уровнем отказоустойчивости, заплатив за это определенно количество вычислительных ресурсов. В другой — этих ресурсов иожет попросту не быть. Масштабируемость. Под этим понятием подразумевается легкость расширения системы — подключения к ней новых узлов.  Очевидно, что для комплексного выполнения этих требований, а также возможности адаптации под их изменения, необходимо появление нового класса инструментов, лишенных недостатков классических клиент-серверных систем. |
| 3 | Нами предлагается архритектура распределенной системы мониторинга, которая способна удовлетворить всем перечисленным требованиям. Сущность предлагаемого подхода заключается в использовании механизма разработки и исполнения модулей в процессе решения задач мониторинга, а также свойств распределенных систем в процессе эксплуатации. При таком подходе, механизм разработки модулей является инструментом динамического расширения функционала системы. Свойства распределенных систем, позволят удовлетворить требованиям к отказоустойчивости и масштабируемости. |
| 4 | Для рассмотрения особенностей архитектуры определимся с базовыми понятиями. Предметная област описывается с помошью терминов Агент, сервер и задача мониторинга. Агент – это активная сущность, запущенная на узле и наюлюдающая за его состоянием. Сервер – пассиваня сущность, предоставлющая агенту ресуры для приема сообщений. И наконец задача мониторинга – определяется как шаблонная проблема получения и анализа некоторой информации о удаленном узле. |
| 5 | Ключевым моментом во всем процессе мониторинга является его цель. Цель описывается множеством задач мониторига о которых говорилось на предыдущем слайде. В предлагаемой архитектуре 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. |
| 6 | Система исполнения наряду с прикладным интерфейсом программирования модулей мониторинга реализуют механизм разработки и исполнения модулей. В свою очередь систем исполнения обеспечивает генерацию кода каркаса модулей, их исполнение в ОС. Является некоторым промежуточным слоем, между модулем мониторинга и агентом, в рамках которого он запускается, что обеспечивает независимость программного кода модуля от физического расположения агентов (адресации, топологии сети). |
| 7 | Код каркаса генерируется системой исполнения на основании текущего глобального состояния для каждого запускающегося в рамках агента модуля. Код каркаса является своего рода оберткой для модуля, которая абстрагирует его от реального физического расположения. Каркас реулизует инициализацию окружения, создание экземпляра модуля и его исполнение, а также передачу параметров и вовзрат результата модуля серверу. |
| 8 | Модули мониторинга разработываются в терминах предметной области с использованием прикладного интерфейса программирования — высокоуровнего объектно-ориентированного набора инструментов.API является промежуточным слоем между модулем мониторинга и операционной средой на которой он запущен. Он призван сосредоточить программиста на решаемой задачи, скрыв от него подробности реализации солжных моментов, таких как распределенная коммуникация, маршализация/демаршализация параметров и возвращаемого результата, системные вызовы ОС. |
| 9 | Из теории распределенных систем, известно понятие глобального состояния системы, которое определяется графом связности узлов, расположением запущенных экземпляров модулей и нагрузкой на узлы. В предлагаемой архитектуре роль распределенного модуля играет вторичный сервер. Нагрузки на узел — индекс производительности узла (некоторое число). Первичный же сервер является клиентом по отношению к распределенной системе. Статус элемента распределенной системы для вторичного сервер а придает ему следующие особенности: масштабируемость (возможность запуска дополнительного экземпляра), сериализуемость (возможность сохранения его внутреннего состояния), переносимость (возможность переноса вторичного сервера в распределенной среде с сохранением его внутреннего состояния). |
| 10 | Рассмотрим механизмы воздействия на состояние распределенной системы мониторинга. В первую очередь, это пороговое значение, которое задается глобально для системы в целом. Именно на основании этого значения, система принимает решения о переходе в новое, более эффективное состояние. Сервера запущенные на узлах, с индексом производительности меньше порогового значения подвергаются масштабированию (запуску дополнительных экземпляров, сопровождаемому балансировкой нагрузки). Пороговое значение также позволяет управлять таким свойством распределенной системы как отказоустойчивость. Теоретически, чем выше это значение, тем распределенная система будет устойчивее. Однако, на практике, возможные расходы на синхронизацию узлов, как правило, могут быть сопоставимы с работой полезной части. |
| 11 | На текущем этапе развитя проекта нами был разработан прототип предлагаемой архитектуры распределенной системы мониторинга, опытные результаты эксплуатации которого лишь подтвердили выдвинутое предположение о практической возможности создания подобной системы. Разработанный прототип обладает следующими особенностями: управляемая распределенная коммуникация — это означает, что переход из одного состояния в другое возможен лишь по вмешательству пользователя. Кросплатформенность. Ввиду того, что серверная часть системы реализована на Java, ее можно запускать на любой ОС, для которой существует сборка JVM. Кроме того, платформа среднего слоя, которая использовалась при разработки системы является кросплатформенной, что позволит в будущем без труда портировать агентскую часть на другие операционные системы. |
| 12 | Можно выделить несколько путей развития проекта. Во-первых, это конечно, полнофункциональная реализация предложенной архитектуры. Во-вторых, реализация агентов и API модулей для популярных ОС позволит добиться платформенной независимости решения. Более того, можно предусмотреть вариант портирования системы на мобильные платформы. Наконец, разработка шаблонных модулей мониторинга, обеспечит систему необходимым «коробочным» функционалом. |
| 13 | Спасибо за внимание. Вопросы? |