

СИСТЕМА ПРОГРАММНЫХ МОБИЛЬНЫХ АГЕНТОВ НА ОСНОВЕ J2EE КОНТЕЙНЕРА ВЕБ-ПРИЛОЖЕНИЙ

Ю.И. Воротицкий, Н.А. Зенькевич
Беларусь, г. Минск

Одним из наиболее интересных направлений агентных технологий являются программные мобильные агенты. Данная работа представляет реализацию платформы мобильных агентов на основе J2EE контейнера веб приложений.

Мобильный код подразумевает перенос кода из одной среды выполнения в другую. Под средой выполнения может пониматься компьютер, операционная система, программный процесс и пр. Системы мобильного кода можно классифицировать несколькими способами. Первый - по источнику инициализации запроса на перемещение кода: источником кода, получателем кода или самим кодом. В третьем случае код называют программным мобильным агентом, а узлы, между которыми перемещается агент, называют узлами каркаса системы мобильных агентов [1]. Согласно теории агентных технологий, под программным агентом понимается объект, который обладает собственным потоком управления, соответственно самостоятельно принимает решения о собственном перемещении. В первом случае, мобильный код также может обладать собственным поведением и, по сути, являться агентом управления.

Второе направление классификации – уровень мобильности [2]. Слабая мобильность подразумевает перемещение только самого кода и его выполнение на новом месте. Сильная мобильность кроме перемещения подразумевает сохранение внутреннего состояния перемещаемого кода, а именно сегмента выполнения и внутренние переменные. Естественно реализация сильной мобильности требует более сложного программного обеспечения, чем слабой.

Для обработки данных либо данные должны быть перемещены к обрабатываемому коду, либо код должен быть перемещен к данным. Если необходимо обработать большой (относительно канала связи) объем данных, то второй вариант более предпочтителен. Кроме того, передача данных по каналу связи может быть небезопасной. Если обрабатывающий код статичен или легко параметризуем, а также известен заранее, то он может быть помещен рядом с источником данных заранее (при развертывании информационной системы). Например, хранимые процедуры баз данных. В том случае, когда обрабатывающий код неизвестен заранее, то рядом с источником данных можно разместить узел

каркаса мобильного кода, который будет принимать обрабатывающий код.

Системы программных мобильных агентов должны поддерживать сильную мобильность, в противном случае при перемещении агент будет терять состояние и соответственно, реализовать бизнес логику, выполняемую распределенно, не представляется возможным. При разработке подобных систем приходится решать несколько сложных проблем.

1. **Сериализация и десериализация объектов, загрузка классов.** При перемещении агента необходимо все объекты, составляющие агента, представить в виде массива байт (сохранить все значения атрибутов по иерархии) для передачи на другой узел с последующим восстановлением объектов. Данные процессы называются сериализацией и десериализацией соответственно.
2. **Обеспечение атомарности выполнения и перемещения агентов.** При перемещении агента необходимо обеспечивать атомарность выполнения его потока. А именно, перемещать все объекты, составляющие агента, а также обеспечивать уничтожение потока агента только после успешного перемещения, начиная выполнения на новом месте, в противном случае необходимо продолжить выполнения агента на старом месте. Когда агент делает вызов перемещения поток агента выполняется, и если выполнить сериализацию объектов агента в этот момент, то сериализованные внутренние переменные могут принять некорректные значения, т.к. агент все еще выполняется и может изменять свое состояние. Кроме того, необходимо остановить поток после успешного перемещения.
3. **Обеспечение безопасности.** Обеспечение безопасности узла информационной сети от разрушительных действий принимаемых агентов, а также защиты агентов друг от друга. Первое, как правило, заключается в ограничении доступа агента к средствам узла и операционной системе, а также проверке достоверности происхождения агента. Реализуется с использованием системы безопасности операционной системы (ограничение прав процесса узла каркаса) и реализации узла на основе виртуальной машины со внутренней системой безопасности.

Под платформой мобильных агентов понимают программное обеспечение, в рамках которого происходит выполнение и перемещение различных пользовательских агентов [3]. Платформа состоит из узлов, между которыми агенты могут перемещаться. Естественно, что все существующие платформы программных агентов обладают собственной архи-

тектурой агентов. Под архитектурой агентов в данном контексте будем понимать структуру класса(ов) представляющих агента и алгоритмов его взаимодействия с узлом каркаса. При использовании платформы Java узлы каркаса системы, как правило, реализуются на основе J2SE.

В данной работе предлагается реализация узла каркаса системы мобильных агентов в виде стандартного веб приложения J2EE, выполняемого в среде любого J2EE сервера. Что позволяет:

1. Упростить конфигурацию и развертывание узла каркаса (узел каркаса поставляется в виде war файла).
2. Стандартным образом конфигурировать предоставляемые агентам сервисы (JNDI контекст выбранного сервера J2EE).
3. Сделать возможным выбор узла платформы (Tomcat, Weblogic, JBoss, WebSphere).
4. Использовать стандартную и надежную реализацию транспортной системы для агентов. Использование протокола HTTP позволяет сделать транспортную систему более простой для конфигурации системы безопасности.
5. Сохранить все преимущества платформы Java (систему безопасности, многопоточность и пр.)

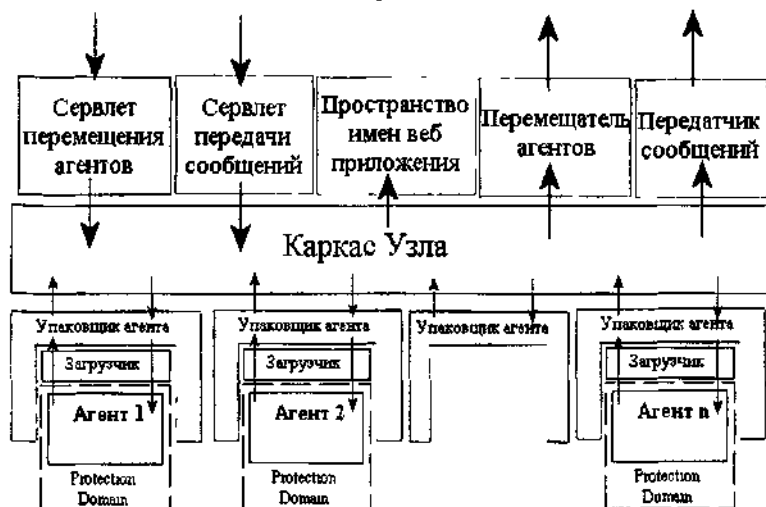


Рис. 1. Схема взаимодействий узла каркаса и агентов.

Узел каркаса разработанной системы может быть размещен не только внутри J2EE веб контейнера, но также просто в среде J2SE. Именно

ется парадигма того, что программный агент обладает собственным фокусом управления.

В других системах мобильных агентов на основе Java (Tryllian ADK, DIET, Kaariboga) атомарность передвижения и безопасная сериализация достигается разными способами. В разработанной системе реализуется механизм, похожий на DIET, только управление метода агента не прерывается исключительной ситуацией (которая может быть перехвачена кодом пользовательского агента), а упаковщик агента помещает требование перемещение в стек событий данного агента и ждет окончания выполнения метода агента, чтобы продолжить обработку сообщений. Когда упаковщик агента получает из стека событие с требованием перемещения, он начинает сериализацию и перемещение. После перемещения обработка сообщения из стека продолжается: события сердцебиения и управления жизнью агента опускаются, сообщения направляются на новое место жизни агента. Диаграмма последовательности действий по перемещению агента приведена на рисунке 2.

В начальной реализации системы агент представляет собой один класс, во время перемещения сериализуемый вместе с классом в тело HTTP POST запроса. В заголовках запроса передаются размеры сериализованного объекта, класса и имя класса. С помощью выделения загрузчика классов для каждого агента решается проблема нахождения двух агентов разных, но одноименных классов, а также проблема безопасности путем создания Protection Domain, что естественно требует использования менеджера защиты на уровне J2EE сервера. Загруженному агенту разрешается только обращения через суперкласс к объекту своего упаковщика. Также лимитируется загрузка других классов.

На основе разработанного каркаса системы мобильных агентов была реализована система поиска на основе мобильных агентов.

Исходная ставилась следующим образом: группа веб серверов хранит информацию, необходимо обеспечить возможность поиска по этой информации минимизировав обмен информацией с клиентами. Механизмы поиска заранее не известны и не стандартизированы.

Задача решалась с помощью развертывания на каждом из серверов узла каркаса системы мобильных агентов. Далее пользователи системы для осуществления поиска должны отправить мобильного агента, который, перемещаясь между серверами, осуществляет поиск необходимой ему информации. Узлы предоставляют агентам следующие сервисы: поисковый индекс, доступ к полному списку и текстам документов, отправка информации по электронной почте, адрес следующего узла в группе для перемещения.

Таким образом, в отличие от обычного, в таких случаях, варианта решения нет необходимости передавать все данные с серверов для обработки, что может быть весьма эффективным в том случае если объем найденных данных значительно меньше хранимых серверами

Литература

- 1 Weiss G Wooldridge M , etc , Multiagent System A Modern Approach to Distributed Artificial Intelligence – London The MIT Press, 1999 – 618 p
- 2 Ван Стеен М , Таненбаум Э Распределенные системы Принципы и парадигмы - СПб Питер 2003 - 880 с (Классика computer science)
- 3 Тарасов В Б От многоагентных систем к интеллектуальным организациям: философия, психология, информатика – М Эдиториал УРСС 2002 – 352 с (Науки об искусственном)