

**АРХИТЕКТУРА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ  
ТЕХНОЛОГИИ АКТИВНЫХ АГЕНТОВ**

**Э.А.Бабкин, М.Л.Зубов, О.Р.Козырев**

*Нижегородский государственный технический университет,  
Н.Новгород, Россия*

Рост производительности вычислительных сетей и другие факторы выводят проблемы создания и поддержки распределенных приложений в число основных на современном этапе развития вычислительных систем. На сегодняшний день типовая распределенная система рассматривается как набор процессов, выполняемых на однородных технических и программных платформах с использованием одного из протоколов рассылки сообщений. В большинстве случаев при этом подразумевается, что вычислительных узлы объединены в одну локальную вычислительную сеть для повышения производительности системы в целом. Системы такого класса можно рассматривать как системы малого или промежуточного уровня распределенности. Развитие глобальных сетевых коммуникаций изменяет ситуацию и позволяет переходить к созданию сильно распределенных систем, предназначенных для практического использования. В таких системах различные части приложений не только могут функционировать на различных программных и технических платформах, но и быть разделенными географически.

Эффективное использование сильнораспределенных вычислительных систем предполагает решение не только проблемы доступа удаленных пользователей к готовым алгоритмам, но и организацию открытого доступа к общим вычислительным ресурсам всей сети. В этом случае процессы распределенного приложения могут быть размещены на любом доступном узле, независимо от их программной или аппаратной платформы, географического положения и других факторов. Повышение производительности в таких системах по сравнению с локальными сетевыми комплексами возможно, конечно, только в приложениях со слабо связанной структурой и высокой степенью параллелизма.

Основными преимуществами использования сильно распределенных приложений перед локально распределенными приложениями выражаются в возможности кооперативной работы и в компьютерной специализации. На пути практической реализации подобных вычислительных систем, однако, появляется множество дополнительных проблем, связанных с недостатками используемых технологий создания таких приложений.

Децентрализация контроля, отсутствие предерминированной стратегии конфигурирования компонент приложения сети делают решение таких проблем невозможным при использовании традиционной технологии создания распределенных систем, основанной на применении RPC, CORBA, MPI, и т.п. Сложные алгоритмы поиска доступных ресурсов по всей глобальной сети требует различных реакций, которые не могут быть определены полностью на этапе разработки программ и включены заранее в состав вычислительных серверов. Попытки централизации контроля всей сети со стороны одного приложения приведут к катастрофическому усложнению программной структуры.

Ниже описывается одно из возможных решений: сетевая среда, использующая технологии активных программных агентов и позволяющая снизить сложность создания и настройки

программных комплексов. Эта среда служит для выполнения задач, требующихся для размещения модулей приложения по узлам глобальной сети. Приложение рассматривается как набор распределенных модулей, выполняемых на узле общей сети и использующий произвольный механизм рассылки сообщений для взаимодействия. Основное предназначение описываемой среды заключается в определении подходящей конфигурации модулей (с учетом доступности ресурсов), транспортирования модулей на выделенный узел и платформо-зависимой компиляции.

Основой среды являются специализированные сервера, так называемые place-узлы сети. Каждый такой узел реализует некоторые из общего пакета сервисных функций системы. Эти функции используются для доступа к данным узла и запуска программы на узле.

Динамическую часть среды составляют активные агенты, которые могут перемещаться в каждый узел и получать доступ к необходимым сервисным функциям. Пользователь системы, использующий среду для создания распределенного приложения, может начать работу из любого доступного узла сети, используя специализированное интерфейсное сопровождение. На первом этапе он определяет основные необходимые свойства узлов, где могут размещаться модули распределенного приложения: память, требования к процессорам и т.п. информация будет использована специальным агентом, который перемещается по каждому узлу сети.

При этом путь перехода на момент старта агента не известен заранее, а определяется динамически при обходе узлов. Для этого каждый узел предоставляет информацию о других доступных узлах. Агент может получить ее с помощью специального сервисного обеспечения. С помощью такого же механизма сервисов агент может получать информацию и о вычислительных характеристиках текущего узла сети. Узел, в свою очередь, получает доступ к некоторым характеристикам приложения, не известным агенту и запускает “ответного” агента в начальную точку для получения информации о пользователе. Это позволяет решать вопрос о санкционированности доступа и позволяет проводить политику безопасного использования среды. При обходе узлов сети стартовый агент получает всю необходимую информацию и возвращает в начальный узел список узлов-кандидатов для размещения распределенного приложения. Пользователь при этом может определить подходящие узлы сети и инициировать вторую фазу - размещение компонент приложения и компиляцию. Для этих целей в каждый выбранный узел посылаются специальные агенты, несущий текст модуля, размещаемого в данном узле. При этом агент, используя механизм сервисов, создает платформно-зависимые установки в исходном коде и проводит компиляцию.

Для реализации тестовой версии подобной среды в качестве базовой подсистемы агентов была избрана система Agent-Tcl. Сервисные функции в узлах реализованы как специализированные стационарные агенты. Для создания стационарных и мобильных агентов использован язык—Tcl/Tk. Узел реализуется как сервер со специальными стационарными агентами. На первой стадии реализации поддерживаются следующие функции: репозиторий свойств, поддержка компиляции и модули доступа.

Кроме самого сервера, где реализуется узел сети, описываемая среда состоит из среды программирования (СП) и среды администрирования (СА). С помощью СП осуществляется спецификация требований к узлам, где будут размещаться модули распределенного приложения, а также выбираются и сами узлы. СА используется для управления работой сервера и размещения сервисных функций.