

С. В. КУЛЕШОВ, И. О. ШАЛЬНЕВ, А. А. ЗАЙЦЕВА, А. Ю. АКСЕНОВ
Санкт-Петербургский Федеральный исследовательский центр Российской академии наук,
Санкт-Петербург

КОММУНИКАЦИОННАЯ ИНФРАСТРУКТУРА НА ОСНОВЕ АКТИВНЫХ ДАННЫХ В КИБЕРФИЗИЧЕСКИХ СИСТЕМАХ

В современных распределенных киберфизических системах актуальной является задача эффективной организации межкомпонентного взаимодействия.

Предложенная в работе методология организации взаимодействия между компонентами распределенной системы без явного обращения к коммуникационной инфраструктуре позволяет решать целевые задачи, без непосредственного обращения к низкоуровневым коммуникационным функциям, что обеспечивает динамическое реконфигурирование программного процесса, функционирующего на каждом узле распределенной системы, без необходимости обновления его программного кода и без остановки его работы.

Введение. В современных распределенных киберфизических системах, таких как робототехнические группы, группы БЛА, распределенные системы сбора данных и др. крайне актуальной является задача эффективной организации межкомпонентного взаимодействия. Для решения таких задач до настоящего времени нет универсального способа разработки коммуникационного программного обеспечения. Традиционно взаимодействие между компонентами, находящимися в узлах распределенной киберфизической системы, реализуется программистом путем использования стандартизованных или вновь разработанных средств коммуникационной подсистемы. Модели и методологии взаимодействия для обмена данными между узлами выбираются в зависимости от потребностей конкретной задачи (модель передачи команд, подписка на обновления, «доска объявлений»), но при этом взаимодействие с коммуникационной инфраструктурой, как правило, остается задачей разработчика [1].

Методология разработки распределенных систем на основе распределенной виртуальной машины. В рамках настоящего исследования предлагаются новая методология организации взаимодействия между компонентами распределенной системы без явного обращения к коммуникационной инфраструктуре, а также инструментарий разработки распределенных систем, которые основаны на построении приложений для распределенной виртуальной машины, функционирующей одновременно на всех узлах распределенной киберфизической системы. При этом коммуникационная среда для межкомпонентного взаимодействия реализуется средствами самой виртуальной машины. Технически взаимодействие между вычислительными узлами представлено последовательностью вызовов функций распределенных объектов (фрагментов программного кода на вычислительных узлах) за счет передачи байт-кода, определяющего удаленный вызов этих функций. Сами передаваемые данные (полезная нагрузка) при этом содержатся в параметрах удаленно вызываемых методов. [2]

Передача данных между распределенными объектами происходит на транспортном уровне стандартной модели OSI/ISO, реализованном на базе стандартных сетевых протоколов, применимых для используемой среды передачи данных, названном в предлагаемой архитектуре транспортным слоем. При этом виртуальная машина (слой VM), обеспечивающая взаимодействие данных объектов, находится на уровне выше (рисунок). Такая организация слоёв архитектуры обеспечивает изолированность уровней, позволяя осуществлять взаимодействие объектов на вычислительных устройствах (ВУ) уже существующими сетевыми протоколами без непосредственного обращения к низкоуровневым функциям. Это позволяет разработчикам сосредоточиться на решении целевой задачи, уменьшив долю рутинных участков программного кода, не связанных с предметной областью.

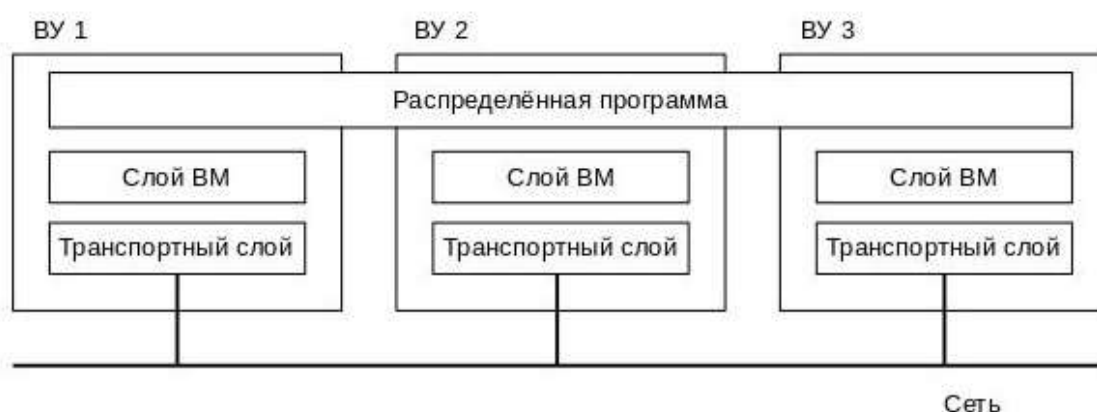


Рисунок. Архитектура слоев распределённой системы, функционирующей на основе распределенной виртуальной машины

Для реализации предлагаемой методологии взаимодействия необходимо определить структуру передаваемых данных, что позволит извлекать получаемые данные на принимающей стороне с минимальными ошибками. Для этого в процедуру отправки в качестве аргумента подаются данные, сформированные в виде последовательности байт (байт-код). Такая технология является усовершенствованным вариантом реализации концепции активных данных, когда передаваемые данные представляются в форме исполнимого кода, который, в свою очередь, исполняясь на стороне приемника, восстанавливается в исходную форму [2, 3, 5]. Ее преимуществом является то, что контент, составляющий полезную нагрузку, не обязательно должен подвергаться трансформации в форму команд исполнимого кода перед передачей, а может сохраняться в исходном виде, что позволяет избежать основной сложности практического применения активных данных – необходимости разработки кодека под любой тип передаваемых данных.

Программное обеспечение распределённой системы при использовании такой методологии должно разрабатываться как единая программная среда, в которой взаимодействие изолированных программных компонентов синтаксически схоже с организацией взаимодействия в классическом (не распределенном) приложении. При этом узлы распределённой системы должны рассматриваться не по отдельности, а как единый вычислительный ресурс [2, 3], соответственно отдельные функции для межкомпонентной синхронизации не требуются. Конкретное распределение функциональных модулей по узлам системы (то есть её конфигурация), может быть различной. Например, графический терминал распределённой системы сам по себе не определяет пользовательский интерфейс, он имеет функционал реализации графических элементов, но структура их расположения в пользовательском интерфейсе и бизнес-логика реализована во внешнем компоненте [4]. Компоненты на узле распределённой системы предоставляют библиотеку предустановленных функций, обеспечивающих необходимый набор моделей и алгоритмов, к которой можно обратиться удалённо. В этом случае данные, передаваемые на удалённый узел, являются последовательностью вызовов удалённых (то есть находящихся на внешнем узле) предустановленных функций.

Приведем пример построения распределенной системы с использованием описанной методологии. Узлами данной системы являются группа БЛА и наземная станция управления. Рассмотрим два варианта реализации алгоритмов движения отдельных БЛА по маршруту. В первом случае в каждом БЛА реализовано множество алгоритмов анализа окружающей обстановки, необходимых для реализации алгоритмов полёта и разработчику программного обеспечения для организации полета БЛА по маршруту фактически достаточно вызвать одну функцию – запуск полета функции полета с параметрами маршрута. Во втором случае на БЛА реализованы низкоуровневые команды управления: движение с заданным курсом на заданное расстояние, простые полетные эволюции, изменение высоты и т.п. При этом алгоритмы, реализующие высокоуровневые функции полета, будут локализованы на вычислителе наземной станции управления, а не на вычислительном устройстве БЛА. В обоих рассмотренных случаях алгоритмы полета будут достаточно эффективно реализованы, но локализация их компонентов зависит от текущих задач и возможностей конкретной распределенной системы. Это является

актуальным, когда мощность вычислительного устройства (в данном случае, находящегося на БЛА) ограничена.

Предложенный подход при разработке распределённых киберфизических систем для организации межкомпонентного взаимодействия позволяет выбирать устройства, на которых будут исполняться высокоуровневые алгоритмы, реализующие целевую задачу. Это повышает масштабируемость всей системы в целом, так как устраняет «привязку» исполнимого кода к вычислительному устройству, позволяя изменять архитектурные решения не в процессе разработки, а в процессе эксплуатации системы.

Заключение. Предложенный подход при разработке распределённых киберфизических систем для организации межкомпонентного взаимодействия позволяет разработчикам сосредоточиться на решении целевой задачи, устранив непосредственное обращение к низкоуровневым коммуникационным функциям.

Кроме того, становится возможна динамическая модификация поведения изолированного программного процесса (динамическое реконфигурирование), функционирующего на каждом узле распределенной системы, без необходимости обновления его программного кода (фактически кода виртуальной машины и программного кода системного уровня) и без остановки его работы.

Работа проводилась в рамках исполнения Государственного задания, бюджетная тема № FFZF-2022-0005

ЛИТЕРАТУРА

1. **Бытачевская Т.Н., Плужникова В.Г.** Дизайн создания и коммуникации киберфизических систем и интернета вещей. *Интеллектуальные технологии и средства реабилитации и абилитации людей с ограниченными возможностями* (ИТСР-2017), 2018.
2. **Шальнев И.О.** Подход к построению распределенной виртуальной машины на основе объектно-ориентированного программирования. *Известия Тульского государственного университета*. 2020. №9. С. 40–47.
3. **Кулешов С.В., Зайцева А.А., Шальнев И.О.** Подход к реализации распределенной системы виртуальных машин для самоорганизующихся сетей. *Информационно-управляющие системы*, №5, 2019, с. 30-37. doi:10.31799/1684-8853-2019-5-30-37
4. **Шальнев И.О., Аксенов А.Ю.** Генерация пользовательского интерфейса на основе технологии распределенной виртуальной среды. *Информационно-измерительные и управляющие системы*, 2019. Т. 17. №5 С. 44-50.
5. **Кулешов С.В. Цветков О.В.** Активные данные в цифровых программно-определяемых системах. *Информационно-измерительные и управляющие системы*, 2014, №6. С. 12-19.

S.V.Kuleshov, I.O.Shalnev, A.A.Zaytseva, A.Y.Aksenov (SPC RAS, St.-Petersburg)
Communication infrastructure based on active data in cyber-physical systems

The problem of effective organization of component interaction is actual in modern distributed cyber-physical systems. The methodology proposed in the paper for organizing interaction between the components of a distributed system without explicitly referring to the communication infrastructure allows solving target problems without directly accessing low-level communication functions. This provides dynamic reconfiguration of the software process that operates on each node of the distributed system, without the need to update its program code and without interrupting his work.

Авторы готовы представить текст на английском языке для сборника материалов мультиконференции, который будет подан для индексирования в Scopus.