

УДК 004.77

Е.В. Малахов, канд. техн. наук,
Б.Ф. Трофимов, магистр,
Одес. нац. политехн. ун-т

ИНСТРУМЕНТ РЕАЛИЗАЦИИ СВЯЗЕЙ МЕЖДУ УЗЛАМИ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

Е.В. Малахов, Б.Ф. Трофимов. **Инструмент реалізації зв'язків між вузлами розподіленої системи.** Розроблено інструмент реалізації зв'язків між вузлами розподіленої системи, який задовольнить таким вимогам, як швидкість обробки клієнтського запиту сервером, можливість роботи в гетерогенних середовищах, підтримка безпечного обміну даних.

E.V. Malakhov, B.F. Trofimov. **The communications generating tool for data exchange between the units of the distributed system.** The communications that generating tool for data exchange between the units of the distributed system is developed satisfies such requirements as the speed of client request processing with a server, an opportunity to work in heterogeneous environments, support of safe data exchange.

С развитием интернет-технологий количество передаваемой информации между пользователями глобальной сети резко увеличилось. Приложения, выполняющиеся ранее только на одном компьютере, перерастают в распределенные информационные системы (РИС) — программные комплексы, распределенные между несколькими информационными узлами, и реализующие связи между ними [1]. РИС становятся все более популярными и востребованными, и вопрос их качества разработки и функционирования приобретает актуальность.

Самым узким местом при разработке РИС является реализация качественных связей между ее информационными узлами (ИУ). ИУ — это базовая единица обработки информации РИС, например, пространственно удаленный сервер, либо приложение, запущенное на многопроцессорном компьютере. Интерфейс взаимодействия двух ИУ (ИВУ) — это формальное описание всех МС между ИУ. Как правило, совместная работа разработанных и отлаженных по отдельности ИУ РИС подвергается риску из-за реализации ИВУ, не всегда учитывающей индивидуальные особенности используемых аппаратных архитектур, типов используемых протоколов [2]. Инструмент реализации связей (ИРС) — набор программных средств, которые позволяют формально описать ИВУ некоторой РИС и по их формальному описанию сгенерировать стабы (автоматически сгенерированные программные модули с использованием некоторого языка программирования), обеспечивающие реализацию ИВУ. Существующие ИРС страдают чрезмерным абстрагированием от сетевого протокола transmission control protocol (TCP) [3], что приводит к потерям по времени при интенсивном обмене данными между ИУ РИС.

Для обеспечения качественной реализации ИВУ РИС предлагается ИРС, использующий единый протокол обмена данными, исключающий ошибки, связанные с особенностью аппаратной и программной архитектур, а также обеспечивающий минимально достаточное абстрагирование от протокола TCP.

При решении практической задачи, где задействован разработанный ИРС, необходимо учитывать ряд технических требований к нему.

1. Поддержка языков программирования C++/C. Эти языки наряду с выразительными средствами описания программы в достаточной мере обладают мощностью низкоуровневых средств, помогающих оптимизировать работу с памятью, а также автоматическое генерирование C++ кода дает возможность получить хорошо инкапсулированный и безопасный код.

2. Надежность и устойчивость работы стабов, сгенерированных ИРС. Работа стабов в составе РИС с должна быть устойчива не только к ошибкам пользователей, но также к сбоям в коммуникациях и системе, обеспечивать работу в течение длительного непрерывного времени.

3. Масштабируемость. Стаб должен легко внедряться в любой узел РИС, в тоже время, изменение интерфейса между клиентом и сервером должно приводить только к регенерации стаба.

4. Безопасная передача данных между ИУ, поддержка семейства защищенных протоколов передачи данных transport layer security (TLS) [4].

5. Поддержка механизма сбора и передачи информации об ошибках, происходящих как на коммуникационном уровне, так и уровне бизнес-логики РИС.

6. Эффективный коммуникационный уровень, позволяющий отождествить логическую сессию с TCP-сессией.

7. Возможность передавать т.н. обратные вызовы (от сервера к клиенту) в контексте одной и той же сессии. Это достаточно актуально из-за использования, как клиентами, так и серверами, программ, защищающих и блокирующих выход в Интернет. На практике количество открытых соединений является критическим и часто ограниченным.

8. Развитой язык описания данных, таких как структуры, массивы, элементарные типы, которые подлежат передаче между ИУ.

9. Возможность работы сгенерированных стабов в гетерогенных средах (Linux, Windows X).

В соответствии с приведенными требованиями проведен анализ предлагаемого и известных ИРС: VisiBroker-CORBA [2], Sun-RPC [5], SOAP [6], DCOM, NET.Remoting [2]. Проведен их анализ в соответствии с приведенными требованиями (табл. 1).

Таблица 1

Результаты анализа существующих ИРС

Требования	ИРС				
	VisiBroker	Sun-RPC	DCOM, Net.Remoting	SOAP	Предлагаемый
1	+	+	+	+	+
2	+	+	–	+	+
3	+	+	+	+	+
4	+	+	+	+	+
5	+	+	+	+	+
6	–	+	–	–	+
7	–	–	–	–	+
8	+	+	+	+	+
9	+	+	–	+	+

Надежные и управляемые ИРС VisiBroker и SUN-RPC являются коммерческими, и окупаются только при реализации крупных проектов. DCOM и Net.Remoting ограничены работой только с ОС Windows. SOAP-реализации не обладают высокой скоростью обработки данных между ИУ.

Предлагается ИРС, удовлетворяющий всем требованиям.

Представлен общий алгоритм работы с ИРС (рис 1.). Пусть рассматривается некоторая РИС, состоящая из двух ИУ, и описание ИВУ. ИВУ подается на вход транслятору стабов в составе ИРС, который генерирует для обоих ИУ соответствующие стабы по заданному ИВУ. Стаб включает в себя реализацию упаковки-распаковки данных, передачи данных по каналам связи. Затем сгенерированные стабы подключаются к проекту РИС, после чего он компилируется и готов к использованию. Если в процессе работы РИС изменился ИВУ, необходимо весь процесс генерации стабов повторить заново.

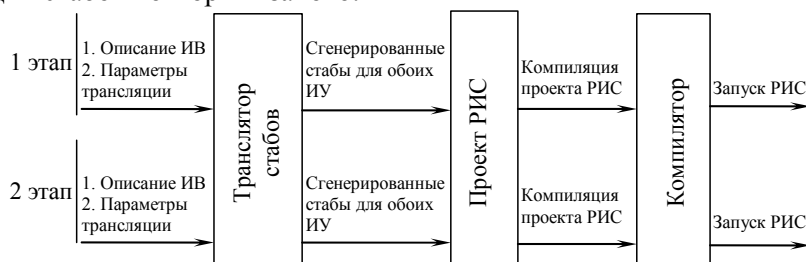


Рис. 1. Алгоритм работы с ИРС

ИВУ описывается при помощи разработанного для этих целей языка описания данных (IDL) и использует функциональный подход. Его суть заключается в том, что любое взаимодействие между двумя ИУ можно представить как взаимный вызов некоторой функции между узлами. Все вызовы описываются в виде списка удаленных функций (УФ), вызываемых одним узлом, а реализованных и выполняемых другим. Каждая УФ может иметь входные и выходные аргументы. Список описаний всех УФ формирует ИВУ.

Предлагается алгоритм взаимодействия двух ИУ на примере вызова УФ во время выполнения РИС (рис. 2). ИУ, инициирующий вызов, называется клиентом, а принимающий вызов ИУ — сервером. Клиент вызывает УФ, передав ей необходимые параметры, после чего управление передается соответствующему объекту-обертке из клиентского стаба. В этом объекте инкапсулирована упаковка параметров в некоторую транспортную форму, которая передается транспортному ядру в составе ИРС для отправки на сервер. Эта форма содержит информацию об УФ, ее входные параметры, флаг готовности получения выходных параметров и некоторые другие. После получения данных транспортным ядром сервера происходит их преобразование из транспортного представления в собственное, после чего серверный стаб вызывает закрепленную событийную процедуру, передав ей разобранные параметры. Если такая УФ не обнаружена, или формат входных параметров не корректен, то информация в виде кода ошибки немедленно возвращается клиенту.

После выполнения событийной процедуры выходные параметры, если они есть, упаковываются и передаются назад клиенту, который получает и представляет их как выходные значения изначально вызванной УФ.

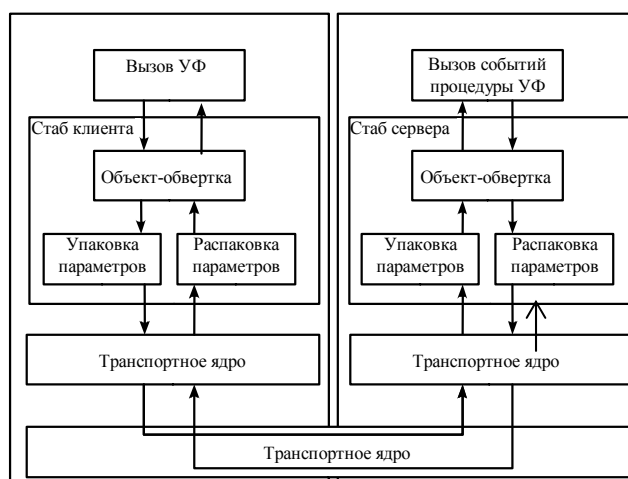


Рис. 2. Вызов УФ

Проведен тест замеров времени выполнения некоторой УФ с использованием предлагаемого ИРС, VisiBroker 6.0, DCOM. В примере РИС состоит из двух ИУ. Соединение между ними происходит через Internet. ИВУ состоит из одной функции с одним входным параметром — массивом из 50 чисел. УФ выполняет сортировку переданного массива. Измерение времени происходило при последовательном 100-кратном вызове УФ (табл. 2):

Таблица 2

Замеры времени выполнения УФ

ИРС, задействованные в тесте	Время выполнения УФ, мс			
	№ вызова			
	1	10	50	100
Предлагаемый ИРС	100	101	115	109

DCOM	153	170	220	190
VisiBroker	212	224	252	300

Результаті теста свідчать про значительное накопление отправляемых и получаемых данных в стеке при многократном вызове УФ func для ИРС DCOM и VisiBroker 6.0 по сравнению с предложенным. Задержки DCOM и VisiBroker можно объяснить неоптимальным управлением TCP-соединения, а также дополнительным временем, затраченным на упаковку/распаковку параметров.

Предложенный ИРС — завершённый продукт, удовлетворяющий всем поставленным требованиям, а также обладающий следующими достоинствами:

- поддержка автоматического сжатия больших объемов передаваемых данных;
- реализация транслятора стабов в виде генератора COM*-объектов для ОС Windows;
- поддержка асинхронного вызова УФ, под которым подразумевается вызов УФ, не блокирующий поток выполнения УФ до получения ответа на запрос.

Последняя модель вызовов наиболее часто используется в WindowsX-среде, где большинство действий выполняется в контексте главного потока приложения.

Литература

1. Олифер.В. Компьютерные сети: Принципы, технологии, протоколы / Олифер.В, Олифер Н — СПб.: Питер, 2007. — 960 с.
2. Цимбал А.В. Технология CORBA для профессионалов — СПб.: Питер, 2001— 624 с.
3. RFC 793 — Transmission Control Protocol <http://www.faqs.org/rfcs/rfc793.html> 20/04/07
4. RFC 2246 — The TLS Protocol <http://rfc.sunsite.dk/rfc/rfc2246.html> 20/04/07
5. RFC 1831 RPC: Remote Procedure Call Protocol Specification Version 2 <http://tools.ietf.org/html/rfc1831> — 20.04.07.
6. <http://www.w3.org/TR/soap/> — 20.04.07.

Поступила в редакцию 12 января 2007 г.

* COM — технология межпроцессного взаимодействия. Разработана корпорацией Microsoft (www.microsoft.com).
