Асратян Р.Э.

Безопасная обработка информационных запросов в мульти-серверной среде

Аннотация: Рассмотрены принципы организации сетевой службы, предназначенной для защищенной обработки информационных запросов в распределенных информационных системах, ориентированных на работу в сложных сетевых структурах со многими обрабатывающими серверами. Отличительными особенностями службы являются тесная интеграция функций информационной защиты данных с функциями информационного взаимодействия и конвейерной обработки запросов в мульти-серверной среде.

Ключевые слова: распределенные системы, Интернет-технологии, информационное взаимодействие, информационная безопасность, конвейерная обработка

Средства информационного взаимодействия в сети являются основой функционирования распределенных информационных систем, в значительной степени определяющей их архитектуру и характеристики. Сегодня в распоряжении разработчиков таких систем имеется целый ряд сетевых информационных технологий высокой степени универсальности и гибкости [1-3]. Например, Web-технологии, имеющие широкий спектр применений от электронной прессы до распределенных вычислений. Однако универсальность имеет и оборотную сторону: она не позволяет продвинуться в сторону подготовки готовых решений целого ряда важных для разработчика задач, в том числе в области информационной безопасности [4]. Это пробуждает интерес к созданию более специализированных сетевых технологий, ориентированных на поддержку распределенных информационных систем.

Цель создания новой сетевой службы PMS (Protected Message Service) заключается в тесной интеграции функций сетевого информационного обмена с функциями защиты и аутентификации данных [5]. Внешне эта интеграция проявляется в том, что отмеченные функции входят в набор методов главного класса службы – класса «Защищенное сообщение» (PmsMessage),

отображающего электронный документ (информационный запрос или ответ), снабженный одной или несколькими удостоверяющими ЭЦП. В отличие от технологии Web-сервисов описываемая служба опирается не на модель вызова методов удаленных объектов, а на модель обмена сообщениями. В данном случае это означает, что все сервисные обрабатывающие функции (методы) имеют одинаковую, жесткую спецификацию: они получают объект класса «Защищенное сообщение» в качестве параметра и возвращают объект того же класса.

Важным преимуществом сетевых служб, построенных на модели обмена сообщениями, является принципиальная возможность организации «программного конвейера»: обработки информационного запроса не одной сервисной функцией, но цепочкой функций таким образом, что защищенное сообщение, возвращенное каждой сервисной функцией, или передается непосредственно на вход следующей функции в цепочке (если она имеется) или возвращается клиенту. Разумеется, здесь имеется прямая аналогия с известным еще с первых версий операционной системы UNIX механизмом «трубопровода» (pipeline), основанном на последовательном соединении стандартных выводов и вводов у нескольких процессов в компьютере. Однако, поскольку в данном случае речь идет о сетевой службе, сетевых сообщениях и о распределенных системах, наибольший интерес представляет мульти-серверная организация конвейера, в которой сервисные функции, задействованные в обработке информационного запроса, могут выполняться на разных серверах (в том числе, на серверах, размещенных в разных сетях).

На рис. 1 проиллюстрирована одно-серверная конвейерная обработка защищенного сообщения, а на рис. 2 – мульти-серверная обработка, в которой защищенное сообщение последовательно обрабатывается несколькими сервисными функциями (отображены маленькими светлыми прямоугольниками) в четырех серверах.

В обоих случаях обработка инициируется с помощью метода Process класса PmsMessage, в котором основным параметром является символьная строка «конвейерный список», содержащий последовательность имен сервисных функций, которые должны принять участие в обработке защищенного сообщения. Разница между двумя примерами заключается только в том, что в первом

случае конвейерный список включает лишь имена сервисных функций в форме «имя_библиотеки.имя_функции» (на рисунках оба имени обозначены латинскими буквами), а во втором случае в списке также задаются имена серверов и простые скобки. Например, обработке, проиллюстрированной на рис. 2, соответствует следующий конвейерный список:

L_1.A, L_1.B, beta/L_2.E, L_1.C, delta/(L_3.F, gamma/L_4.H, L_3.G), L_1.D

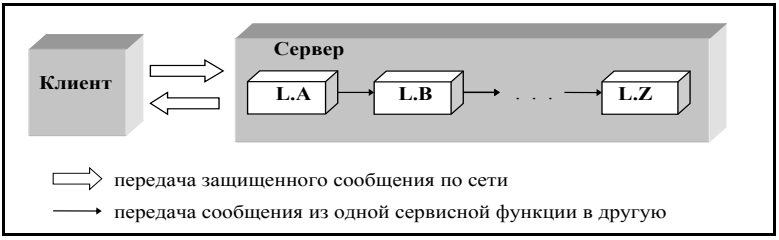

Рис. 1 – Пример одно-серверной конвейерной обработки запроса

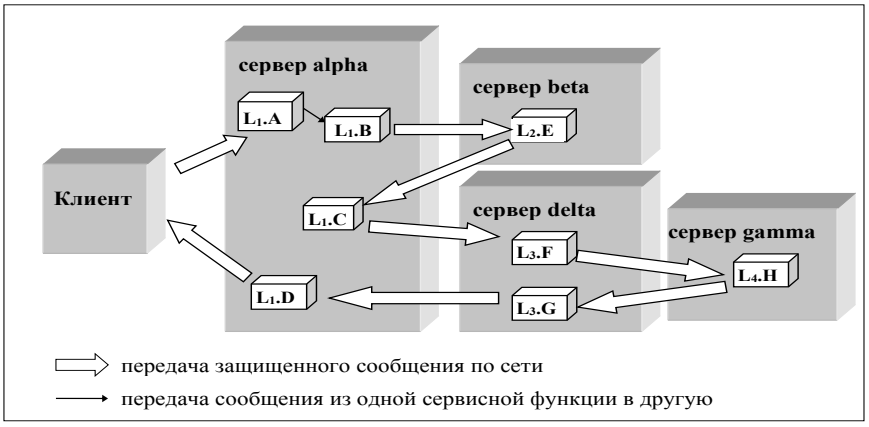

Рис. 2 – Пример мульти-серверной конвейерной обработки запроса

Логика «конвейера» всегда предполагала строго последовательную обработку данных. Однако, возможности сложных сетевых сред с десятками и сотнями сетевых узлов, в которых работают многие современные распределенные системы, пробуждают интерес к средствам параллельной обработки информационных запросов, как к источнику повышения

производительности. В новой версии PMS переход от строго последовательной обработки к параллельно-последовательной был выполнен по следующим принципам.

- В структуру конвейерного списка вводится разметка групп параллельно и последовательно выполняющихся элементов с помощью квадратных и фигурных скобок соответственно.
- Сервисные функции получают и возвращают не один объект класса PmsMessage, а произвольный массив таких объектов (разумеется этот массив может по-прежнему включать только один элемент, но может и несколько).

На рис. 3 проиллюстрирована параллельная обработка защищенного сообщения в трех серверах (beta, gamma и delta), в которой результат формируется в форме массива из трех сообщений. Обработке, проиллюстрированной на рис. 3, соответствует следующий конвейерный список:

$L_1.A$, [beta/$L_2.C$, gamma/$L_3.D$, delta/($L_4.F$, $L_4.F$)], $L_1.B$

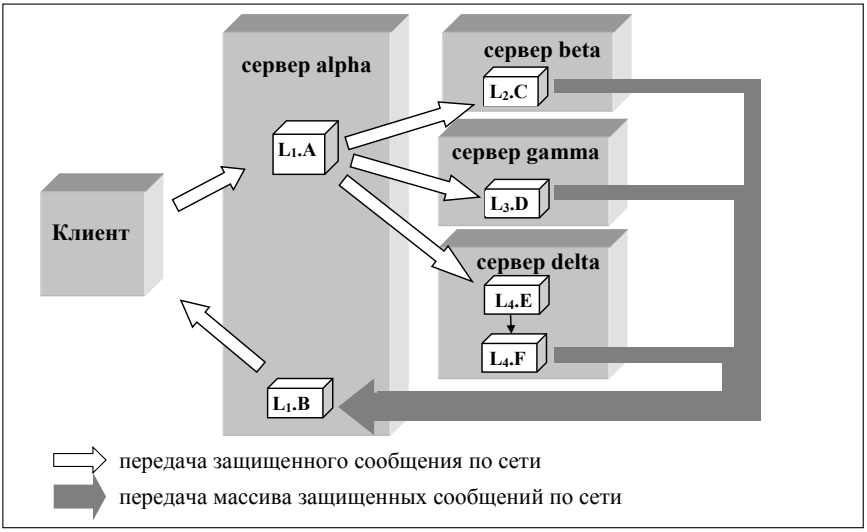

Рис. 3 – Пример параллельной мульти-серверной обработки запроса

Новая версия сетевой службы была реализована в форме двух дополняющих друг друга программных продуктов: сервера PMS (в форме постоянно активной Windows-службы) и библиотеки функций PmsBase.dll для клиентских приложений и библиотечных

сервисных функций. Оба продукта реализованы на языке C# в среде Microsoft Visual Studio для среды MS Framework 4.0. Первые опыты ее использования и лабораторные эксперименты показали высокое быстродействие и способность существенно сократить трудозатраты разработчиков на организацию распределенной обработки и защиты данных в мульти-серверной среде.

Литература:

1. *Мак-Дональд М., Шпушта М.* Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. – М.: Вильямс, 2009. – 1408 с.
2. *Хант К.* TCP/IP. Сетевое администрирование. – СПб.: Питер, 2007. – 816 с.
3. *Снейдер Й.* Эффективное программирование TCP/IP. Библиотека программиста. – СПб.: Символ-Плюс, 2002. – 320 с.
4. *Згоба А.И., Маркелов Д.В., Смирнов П.И.* Кибербезопасность: угроз ы, вызовы, решения // Вопросы кибербезопасности. – 2014. – № 5. – С. 30-38.
5. *Асратян Р.*Э. Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. – 2016. – № 11. – С. 490-497.