

УДК 004.414.22

О подходах и методах интеграции распределенных компьютерных систем

Д.В. Росолько

Изложены наиболее часто применяемые подходы и методы интеграции распределенных компьютерных систем. Выделены требования, предъявляемые к интеграционному решению, а также вызовы, которые появляются во время проектирования и реализации интеграции систем. Рассматривается способ анализа применяемых подходов и методов интеграции распределенных компьютерных систем.

Ключевые слова: интеграция, распределенная компьютерная система.

The most frequently used approaches and methods of integration of distributed computer systems are described. The requirements for the integration solution as well as the challenges that arise during the design and implementation of system integration are highlighted. The way of analyzing the approaches and methods used to integrate distributed computer systems is considered.

Keywords: integration, distributed computer system.

Введение. Основная задача интеграции заключается в налаживании функциональных связей между компьютерными системами, выполняющимися на различных программных платформах, распределенными между несколькими географическими точками и т. п.

Выбор методов интеграции распределенных компьютерных систем – одна из наиболее актуальных и сложных задач в разработке архитектуры распределенных компьютерных систем.

Проблема интеграции распределенных компьютерных систем чрезвычайно многоаспектна и многообразна. Сложность и характер используемых методов ее решения существенно зависят от уровня интеграции, который необходимо обеспечить, свойств отдельных источников данных и всего множества источников в целом, требуемых способов интеграции.

Подходы и методы. Как правило, возникает ряд вызовов, с которыми сталкиваются при интеграции систем [1]:

– Ненадежность сети передачи данных. Все подходы интеграции предполагают передачу информации между распределенными компьютерными системами. В отличие от процессов, выполняющихся на одном компьютере, распределенной компьютерной среде свойственен ряд недостатков. Взаимодействующие системы разделены географически и передача данных может происходить по телефонным линиям, сегментам локальных сетей, общедоступные сети и спутниковые каналы связи, поэтому доставка информации происходит с задержкой, а также увеличивается риск потери.

– Низкая скорость передачи данных. Время вызова локального метода в системе намного меньше, чем доставка данных через компьютерную сеть. Таким образом, при создании архитектуры распределенных компьютерных систем применяются иные принципы, чем при создании системы, работающей в рамках одного компьютера.

– Различия между системами. Метод интеграции должен учитывать все различия (язык программирования, платформа, формат данных) между объединяемыми системами.

– Неизбежность изменений. Метод интеграции должен предоставлять возможность адаптации под изменения в объединяемых им системах. Метод интеграции должен уменьшать взаимосвязь между системами за счет так называемого слабого связывания [2], которое поможет избежать непредсказуемых последствий для всей системы вследствие изменений в одной компоненте системы.

– Несовместимость лицензий. В основе архитектурных решений лежат не только технологии, но и бизнес-правила, схемы лицензирования, а также множества других факторов [3].

Для преодоления указанных выше трудностей можно воспользоваться одним из подходов к интеграции распределенных компьютерных систем:

– Интеграция на уровне брокеров сообщений. Брокер сообщений принимает входящие сообщения, тестирует их на предмет наличия ошибок, преобразовывает и помещает в требуемый исходящий канал. Подобная архитектура позволяет максимально изолировать приложения друг от друга, что чрезвычайно важно для их интеграции. Брокер сообщений реализует всю логику, требующуюся для маршрутизации сообщений между приложениями, и, по сути, является «интеграционным» аналогом посредника (Mediator) [4]. К преимуществам можно отнести универсальность – практически всегда можно создать дополнительный программный модуль, который будет взаимодействовать с обеими системами, еще и разными способами (например, в одну через базу данных, а в другую через RPC). К недостаткам такого метода интеграции следует отнести сложность и трудоемкость, а, следовательно, высокую стоимость разработки, внедрения и владения.

– Обмен файлами является распространенным подходом к организации взаимодействия. Это связано с относительной простотой реализации, а также существованием стандартных форматов обмена. Но у этого подхода есть и недостатки: если необходимо оперировать сложными структурами, то простые форматы обмена уже не пригодны. Возникающие в таких случаях специализированные форматы файлов должны «понимать» взаимодействующие системы, что ведет к жесткой зависимости систем друг от друга.

– Интеграция на уровне базы данных – то есть несколько приложений могут обращаться в одну базу данных или в несколько баз данных, связанных репликацией. Общая база данных обеспечивает согласованность хранящейся в ней информации. Изменения одной и той же части данных из нескольких различных систем будут корректно обрабатываться базой данных в силу поддержки транзакций реляционными базами данных. Наличие общей базы данных устраняет проблему семантического диссонанса. К преимуществам можно отнести низкую стоимость интеграции, особенно при использовании одной СУБД. Недостатки: изменения структуры базы приведет к необходимости изменения всех использующих ее систем.

– Интеграция на уровне сервисов – это интеграция, основанная на фиксации интерфейсов и форматов данных с двух сторон и позволяющая наладить быструю отработку межсистемной бизнес-логики. Однако есть и недостатки: в этом методе присутствует фиксация, а если происходят изменения процессов или структур, то возникают проблемы и узко специализированные, частные решения.

– Интеграция на уровне пользователя – крайний случай не автоматизированной интеграции, т. к. пользователи перемещают данные между системами в ручном режиме при помощи файлов, почты и т. п.

– Динамическая интерпретация метаинформации.

Требования к интеграционному решению. В отличие от цельных систем, к распределенным системам предъявляются ужесточенные требования, наиболее важными из которых являются следующие:

1. Масштабируемость систем, т. е. способности эффективно обслуживать как малое, так и очень большое количество клиентов одновременно. Масштабируемость также относится и к другим параметрам системы, таким как количество дополнительного трафика, с которым она может справиться, насколько легко нарастить ёмкость запоминающего устройства или насколько больше других транзакций может быть обработано.

2. Надежность создаваемых приложений. Программный комплекс должен быть устойчив не только к ошибкам пользователей (это определяется главным образом качеством всех частей распределенной компьютерной системы), но и к сбоям в системе коммуникаций. Надежность подразумевает использование транзакций, т. е. гарантированного перехода системы в процессе функционирования из одного устойчивого и достоверного состояния в другое.

3. Доступность. Возможность непрерывной работы в течение длительного времени. Высокая доступность в распределенных системах требует внимательного рассмотрения избыточности для ключевых компонентов, быстрого восстановления после частичных системных отказов и сглаженного сокращения возможностей при возникновении проблем.

4. Высокий уровень безопасности системы, под которой понимается не только контроль доступности тех или иных ресурсов системы и защищенность информации на всех этапах функционирования, но и отслеживание выполняемых действий с высокой степенью достоверности.

5. Управляемость. Проектирование системы, которая проста в эксплуатации еще один важный фактор. Управляемость системы приравнивается к масштабируемости операций «обслуживание» и «обновления». Для обеспечения управляемости необходимо рассмотреть вопросы простоты диагностики и понимания возникающих проблем, легкости проведения обновлений или модификации, прихотливости системы в эксплуатации.

6. Стоимость. Является важным фактором, так как она, очевидно, может включать в себя расходы на аппаратное и программное обеспечение, однако важно также рассматривать другие аспекты, необходимые для развертывания и поддержания системы. Количество времени разработчиков, требуемое для построения системы, объем оперативных усилий, необходимые для запуска системы, и даже достаточный уровень обучения – все должно быть предусмотрено. Стоимость представляет собой общую стоимость владения.

Каждое из этих требований является основой для принятия решений в проектировании распределенной компьютерной системы. Тем не менее, они также могут находиться в противоречии друг с другом, потому что достижение целей одного происходит за счет пренебрежения другими.

Способ анализа подходов и методов интеграции. В целях сокращения сопутствующих расходов по проектированию архитектуры распределенной системы и выбору средств интеграции между ее компонентами, одной из сложнейших задач является разработка программного обеспечения, способного на основе компьютерной модели разрабатываемой системы, а также предъявленных требований к системе, представить наиболее подходящий способ интеграции и его качественные и количественные характеристики. Компьютерные модели в этом случае используются для получения новых знаний об объекте или для приближенной оценки поведения систем слишком сложных для аналитического исследования.

Для построения компьютерной модели системы необходимо учитывать требования к всей системе в целом, чтобы можно было оценить выбранные способы взаимодействия между различными компонентами системы.

Программное обеспечение, которое будет учитывать требования и может быть применено на самых ранних этапах разработки архитектуры новых распределенных компьютерных систем или интеграции уже существующих, несомненно, приведет к снижению временных и человеческих ресурсов при проектировании и разработке системы. Также снизятся риски выбора не подходящего решения интеграции в силу человеческого фактора.

Для оценки эффективности компьютерной модели в разрабатываемом программном обеспечении разработана тестовая компьютерная система, в которой реализована возможность применения наиболее часто используемых методов интеграции:

- интеграция на уровне брокеров сообщений;
- интеграция на уровне данных;
- интеграция на уровне сервисов.

Тестовая компьютерная система выполнена в виде набора независимых компонент таким образом, чтобы можно было применить различные методы интеграции между ними. Каждая компонента системы является независимой самодостаточной системой, которая может быть интегрирована с другой компонентой любым из возможных способов интеграции. Компонента имеет гексагональную архитектуру (рисунок 1), которая способствует высокой поддерживаемости тестовой системы и минимизации технического долга при анализе различных подходов к интеграции систем. Возможности интеграции компоненты выполнены с помощью адаптеров, реализованных на уровне фреймворка. Это дает возможность гибкой и быстрой смены метода интеграции без необходимости изменения внутреннего кода самой компоненты. Особенно полезно это будет при калибровке разрабатываемого программного

обеспечения для представления наиболее подходящего способа интеграции на основе предъявляемых требований.

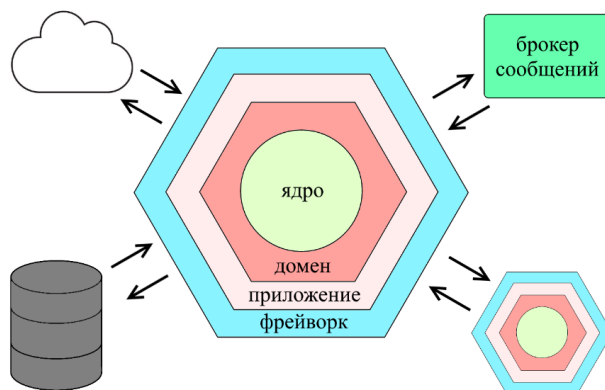


Рисунок 1 – Архитектура тестовой компоненты системы

Примеры систем с использованием компонент для анализа метода интеграции между частями системы представлены на рисунках 2 и 3.

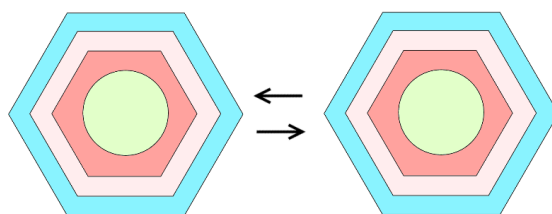


Рисунок 2 – Интеграция компонент через сервисы

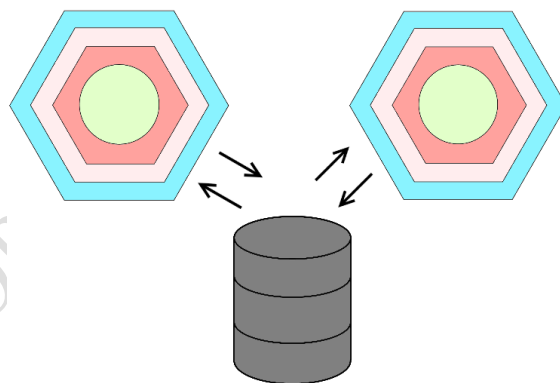


Рисунок 3 – Интеграция компонент через базу данных

Языком программирования тестовой системы, как и программного обеспечения для моделирования возможных вариантов интеграции систем, выбрана Java. Одинаковый выбор языка программирования обуславливается минимизацией оказываемого влияния среды выполнения на получаемые результаты.

Разработанная тестовая система позволяет выполнить анализ примененных интеграционных решений и получить их качественные и количественные характеристики. В дальнейшем тестовая система может быть использована для сопоставления, калибровки и анализа результатов работы программного обеспечения для моделирования возможных вариантов интеграции систем.

Пример анализа применяемых интеграционных решений. На примере систем, представленных на рисунках 2 и 3, проведем анализ выбора наиболее подходящего метода интеграции между частями системы для набора требований (в порядке приоритетов):

1. система должна продолжить функционирование при отказе одной из ее частей;

2. система должна иметь наилучшую пропускную способность при размере передаваемых данных в 10 килобайт, ожидаемое количество запросов в секунду – 800;

3. система должна гарантировать минимальное количество ошибок при взаимодействии частей системы, ожидаемый максимальный процент ошибок – 5;

4. система должна иметь минимальное среднее время выполнения одного запроса, ожидаемое среднее время – 140 миллисекунд;

5. система должна иметь минимально возможную стоимость владения.

Для анализа интеграционных решений необходимо собрать их качественные и количественные характеристики.

Таблица 1 – Качественные и количественные характеристики применяемых в примере интеграционных решений

Метод интеграции	Конфигурация		Пропускная способность Запросы в секунду	Количество ошибок %	Среднее время выполнения одного запроса Миллисекунды
	№	Количество каждой из компонент с одинаковым функционалом / объем оперативной памяти каждой, Количество/Мегабайты			
На уровне сервисов	1	2/128 2/128	681	10.67	253
	2	1/256 2/256	737	12.99	236
	3	2/256 2/256	890	6.12	139
	4	1/512 2/512	902	7.24	148
	5	2/512 2/512	1145	5.98	117
	6	3/128 3/128	773	7.35	170
	7	2/256 3/256	920	6.88	198
	8	3/256 3/256	1207	5.32	189
На уровне базы данных	9	2/256 2/256	526	3.06	223
	10	2/512 2/512	784	5.57	169
	11	3/256 3/256	803	4.78	194
	12	3/512 3/512	1029	4.01	207

Для обеспечения первого требования каждая из частей системы представлена как минимум в двух экземплярах. Конфигурации 3, 5, 8 представлены с целью сравнения характеристик систем в случае отказа одной из ее компонент. В конфигурациях с методом интеграции на уровне базы данных база данных имеет постоянную конфигурацию – 2 сервера по 512 мегабайт каждый.

На основании полученных данных, представленных в таблице 1, можно сделать следующие выводы:

1. В требованиях не задано обязательное хранение данных, поэтому следует рассматривать метод интеграции на уровне сервисов, так как стоимость владения таких систем будет намного ниже. Следует отметить, что конфигурация 12 достаточно хорошо себя показала к предъявляемым требованиям 1–3, однако она достаточно сильно уступает другим конфигурациям по требованиям 4 и 5.

2. Из конфигураций систем, основанных на методе интеграции на уровне сервисов, наиболее подходящими являются 5 и 8, опережающими остальные по требованиям пропускной способности минимум на 24,5 %, количества ошибок минимум на 2,3 %.

3. Конфигурация 8 выигрывает у 5 незначительно по пропускной способности на 5,4 % и количеству ошибок на 12,4 %, однако проигрывает по среднему времени выполнения одного запроса на 61,5 %.

4. Преимуществом конфигурации 5 перед 8 будет значительно меньшая стоимость владения, однако недостатком будет то, что конфигурации 5 на 1 % не соответствует требованию 3 в части ожидаемого максимального процента ошибок.

5. При отказе одной из частей систем в конфигурациях 5 и 8 (характеристики конфигураций отказа одной из частей системы – 4 и 7 соответственно), разница характеристик систем становится меньше: пропускная способность – 1,9 %, количество ошибок – 5,2 %, среднее время выполнения одного запроса – 33,7 %.

Результатом анализа данного конкретного примера будет решение, что наиболее подходящим методом интеграции между частями системы для заданного набора требований будет метод интеграции на уровне сервисов с конфигурацией, состоящей из четырех компонент (двух компонент с одним функционалом, двух компонент с другим) и количеством выделяемой памяти каждой из них по 512 мегабайт (всего 2048 мегабайта).

Заключение. В современном мире компьютерные системы не могут существовать обособлено друг от друга, так как не способны покрыть все потребности. В большинстве случаев встает вопрос выбора подхода интеграции систем на основании требований и характеристик имеющихся систем.

Использованный способ анализа подходов и методов интеграции может использоваться для сопоставления, калибровки и анализа результатов работы программного обеспечения для моделирования возможных вариантов интеграции систем на основе предъявленных требований к системе, а также ее характеристик.

Литература

1. Хоп, Г. Шаблоны интеграции корпоративных приложений / Г. Хоп, Б. Вульф ; Пер. с англ. – М. : ООО «И.Д. Вильямс», 2007. – 672 с.
2. Kaye, D. Loosely Coupled: The missing pieces of web services / D. Kaye. – RDS Press, 2003. – 334 p.
3. Hohmann, L. Beyond software architecture: creating and sustaining winning solutions / L. Hohmann. – Addison-Wesley, 2003. – 352 p.
4. Гамма, Э. Приемы объектно-ориентированного программирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес ; Пер. с англ. – СПб. : Питер, 2001. – 368 с.

Гродненский государственный
университет им. Я. Купалы

Поступила в редакцию 13.04.2017