

УДК 004.4

ПРИНЦИПЫ ПОСТРОЕНИЯ ОБЛАЧНОЙ ПЛАТФОРМЫ¹

Еловиков Андрей Евгеньевич
Компания NAUMEN, Екатеринбург, Россия

В статье рассмотрены причины, обуславливающие необходимость разработки интеграционной облачной платформы, задачи, которые должны быть достигнуты при построении платформы, различные пути ее построения и обоснован выбор способа построения интеграционной облачной платформы

Ключевые слова: ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ, IPAAS, ИНТЕГРАЦИЯ В ОБЛАКЕ, МИГРАЦИЯ ДАННЫХ, ФАСАД ДАННЫХ

UDC 004.4

PRINCIPLES OF CLOUD PLATFORM BUILDING

Elovikov Andrey Evgenevich
Company NAUMEN, Ekaterinburg, Russia

This article presents the reasons causing the need for the integration cloud platform, the tasks to be achieved in the construction of the platform, the different ways to build and chosen are the method of construction of the integration cloud platform

Keywords: CLOUD COMPUTING, IPAAS, CLOUD INTEGRATION, DATA MIGRATION, DATA FACADE

Облачные вычисления в настоящее время являются одной из самых быстро развивающихся IT-технологией. Ввиду очевидных преимуществ, таких как ускорение цикла разработки, сокращение издержек, связанных с поддержанием парка собственных компьютеров, содержания IT-сотрудников и собственной инфраструктуры, для многих пользователей развертывание приложений на мощностях, предоставляемых сторонней организацией, является значительным подспорьем в контексте экономии и быстрого масштабирования требуемых ресурсов.

Эволюция IT-систем приводит к тому, что компания начинает пользоваться облачными решениями, но при этом у него остается программное обеспечение, которое расположено на внутренней площадке компании, и которое играет важную роль в бизнес-процессе. Наличие множества систем, отвечающих за различные аспекты внутренних процессов компании, уже само по себе приводит к необходимости структурировать взаимодействия этих систем между собой [1]. Этой проблемой занимается область интеграции. Значимость решений по

¹ Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 31.07.2012 г. № 14.514.11.4001 в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

интеграции облачных вычислений растет по мере увеличения степени их распространенности. По мере увеличения числа SaaS-приложений предприятия встречаются с новой проблемой разделения корпоративных данных на несвязанные области. Проблема усугубляется увеличением числа предложений на рынке SaaS и простотой доступа к их продуктам.

Применение таких облачных технологий как PaaS и IaaS приводит к возрастающему переносу данных в облачные хранилища, что вызывает необходимость решения задачи обеспечения взаимодействия приложений между собой и разработки эффективных платформ для интеграции облачных приложений и/или облачных приложений с корпоративными.

Как отмечается в исследовании [2], более половины опрошенных участников IT бизнеса назвали проблемы в области интеграции как основную причину отказа от использования SaaS-приложений.

Необходимость решения проблем интеграции облачных приложений привело к созданию технологии интеграционных облачных платформ Integration Platform as Service (iPaaS). Основные ключевые особенности iPaaS следующие [3]:

- iPaaS объединяет множественные облачные сервисы с целью интеграции и управления любой комбинацией внутренних и внешних (т.е. облачных) приложений, SOA и облачных сервисов, процессов и данных внутри и вне организации;
- iPaaS является эволюцией интеграционных сервисов, широко применявшихся в электронной коммерции как дополнение к традиционной инфраструктуре приложений.

Ключевыми функциями платформы iPaaS являются:

- средства поддержки потоков интеграции;
- средств разработки и сопровождения жизненного цикла интеграции;
- управление и мониторинг потоков приложений;

- обеспечение множественности владельцев приложений.

Обзор современного мирового состояния предложений по интеграции приведен в [4]. Там же показано, что основные современные решения реализуют не все ключевые функции платформы iPaaS.

Основные модели интеграции

Интеграция уровня приложения подразумевает почти мгновенное отображение изменений среди интегрируемых систем. Почти всегда такие системы событийно-ориентированы. То есть интеграция заключается в осуществлении небольших транзакций, происходящих примерно в то же время, что и действие в основной системе.

Интеграция данных. Эта модель интеграции по своей сути пакетно-ориентирована. Типичный пример: синхронизация данных между системами по задаче планировщика. Обычно данные перемещаются большими объемами, при этом нет жестких требований к времени выполнения. Такая интеграция выполняется в фоновом режиме (относительно бизнес-процесса).

Федеративная интеграция данных. Можно считать эту модель интеграции подтипом интеграции данных. Отличие заключается в том, что данные берутся из федеративного источника. То есть, по сути, из множества источников. Такой тип интеграции характерен для компаний, у которых в парке находится несколько бизнес-систем, данные из которых необходимо консолидировать для получения общего представления о бизнес-процессе.

Интеграция передачей файлов. Как следует из названия, в этом случае интеграция осуществляется отправкой / получением файлов.

Основными являются первые две модели интеграции, остальные остаются достаточно специфичными.

Задачи построения платформы iPaas

Основные задачи, которые необходимо решить при построении платформы iPaas следующие:

- целостность данных;
- простота администрирования;
- простота настройки интеграции;
- отказоустойчивость интеграции;
- возможность и простота получения статистических данных.

Целостность является необходимой и критической характеристикой. Потеря данных на этапе перемещения данных между системами недопустима. Сохранение целостности данных также должно осуществляться и при возможном отказе какой-либо из составных частей платформы.

Достаточными условиями целостности в распределенной среде будут являться следующие правила, которые должны соблюдаться при передаче интеграционного сообщения от одного узла к другому:

все входные данные, поступившие в исполнительный компонент, должны быть обработаны в соответствии с логикой интеграционного процесса, не должно быть потерянных данных;

результатирующие интеграционные данные должны идти в том порядке, в котором поступали входные данные.

Простота администрирования означает возникновение минимума накладных расходов на использование интеграционного программного обеспечения.

С точки зрения пользователя, наиболее удобным является работа с решением по модели SaaS. В этом случае пользователю нет необходимости

выделять мощности для интеграционного программного обеспечения, и он избавлен от последующего обслуживания этого ПО.

Для обеспечения простоты настройки в процессе настройки должен использоваться распространенный общепринятый подход. В этом случае пользователь избавлен от необходимости изучать новую систему, и он может использовать существующие наработанные навыки. Поскольку, тем не менее, любая система имеет свою специфику, необходимо, чтобы процесс интерактивного обмена информацией с пользователем позволял предоставлять ему информацию в максимально удобном и наглядном виде. Это требует повышенного внимания при разработке средств визуализации для средств интеграции.

Отказоустойчивость определяет время простоя интеграционных процессов, от которых может зависеть основной бизнес пользователя. Таким образом, недостаточное внимание к этому свойству будет косвенно влиять на убытки пользователя интеграционных сервисов.

Отказоустойчивость должна учитываться в архитектуре решения. Прежде всего, необходимо минимизировать количество частей, где возможны отказы. Это означает организацию архитектуры системы максимально простым образом.

Следующим подходом, увеличивающим отказоустойчивость решения, является уменьшение связности его частей. Для реализации этого подхода следует в рамках решения осуществлять выделение отдельных приложений, а также формировать сервис-ориентированный подход к решению. Основным методом взаимодействия между отдельными частями приложения считается RPC или удаленный вызов процедур. Его популярность заключается больше в мимикрии удаленных функций под локальные, что порой является причиной ошибок разработки. Альтернативным подходом к удаленному вызову процедур является

отправка сообщений, которая является низкоуровневой абстракцией, но привносит естественную асинхронность и масштабируемость (см. erlang).

Поскольку статистические данные определяют понимание происходящего процесса, рекомендуется собирать как можно больше параметров [5].

Существует множество решений для получения статистических данных мониторинга, но их сложно администрировать или управлять извне. Хорошим тоном является внедрение датчиков статистических данных непосредственно в приложение. Таким образом, может собираться обширный набор статистических данных, включая те, которые специфичны для приложения, и определены лишь разработчиками. Для реализации подходов, направленных на повышение отказоустойчивости, инструмент сбора данных от датчиков рекомендуется реализовать как отдельное приложение.

Основные варианты реализации платформы iPaaS

Различные варианты решения проблем интеграции данных ИТ-систем, которые предлагаются в настоящее время, можно разделить на три категории:

- решения по миграции данных между системами;
- решения по фасаду данных или оркестровке данных;
- решения, предоставляющие управление облаками.

В категорию решений, связанных с миграцией данных, попадают решения, предназначенные для перемещения данных из одной системы в другую. Частными вариантами обозначения таких решений являются решения по репликации или синхронизации данных.

Основной характеристикой систем этого типа является пассивное выполнение перемещения данных. Это означает, что системы не встраиваются в бизнес-процесс заказчика, а надстраиваются над ним и

выполняют процесс миграции в фоновом режиме, по расписанию или требованию заказчика. Таким образом, решение данной категории представляет собой посредника между двумя или несколькими горизонтальными системами.

Основная задача, которая ставится пользователем перед системами миграции данных — это задача обеспечения целостности данных между системами наиболее простым и надежным способом.

Основные результаты, которые могут быть достигнуты при использовании решений по миграции данных:

- простота администрирования;
- простота настройки интеграции;
- надежность и отказоустойчивость интеграции.

Решения фасада данных (оркестровки данных) полноценно принимают участие в процессе работы пользовательских систем. Примерный сценарий работы может выглядеть следующим образом: пользовательские системы выстроены в некоторую иерархию, данные (или запрос на получение данных) поступают в мастер-систему, мастер-система обновляет данные у зависимых систем. При этом решение может выступать в роли и мастер-системы (диспетчера данных), и в роли посредника между системами, реализуя интеграционный бизнес-процесс.

Решение подобного рода предоставляет пользователям некоторый фасад (интерфейс), через который осуществляется вызов сервисов. При этом такой вызов осуществляется неявно, пользователь только запрашивает требуемые данные, а вся остальная работа по определению того, где они должны быть получены и как должна производиться интеграция, возлагается на мастер-систему.

Основные результаты, которые могут быть достигнуты при использовании решений по фасаду данных:

- надежность и отказоустойчивость фасада;
- высокая производительность фасада (сервисной шины);
- удобство пользователей сервисов;
- простота администрирования;
- простота и скорость настройки фасада;
- наличие статистических данных о работе фасада.

Решения, предоставляющие управление облаками, предназначены, в первую очередь, для решения задачи выбора провайдера (или нескольких провайдеров), подходящего под требуемые критерии (низкая стоимость, высокая надежность, обширный набор дополнительных услуг и т.д.). В связи с постоянным увеличением популярности облачных технологий и постоянным расширением различных предложений на рынке появляется множество IaaS, PaaS сервисов. Для потребителя возникает проблема выбора используемых облачных сервисов или необходимость для оптимизации затрат в их совместном использовании. Это приводит к необходимости использования интеграционного сервиса облачных провайдеров. Такой сервис должен решать задачу распределение сервисов между различными облачными провайдерами. Основные результаты, которые могут быть достигнуты при использовании решений по управлению облаками:

- обеспечение наилучших показателей по соотношению цена/качество при обеспечении требуемого качества услуг;
- достижение простоты администрирования.

Выбор варианта реализации платформы iPaaS

На основании сформулированных задач, которые необходимо решить в рамках построения платформы iPaaS, следует осуществить выбор направленности решения, которое должно быть реализовано.

Фактически, выбор следует производить между решениями по миграции данных и фасадам данных. Решения по управлению облаками с помощью интеграционного сервиса облачных провайдеров позволяют решать некоторую специализированную задачу и не являются общим решением, покрывающим все нужды пользователей по интеграции данных. Естественным представляется разработка одной архитектуры, предоставляющей сервисы всем пользователям, которым требуется интеграция данных между их системами. Кроме того, решения по миграции данных и решения по фасадам данных имеют много общих свойств и, следовательно, могут вытекать одна из другой.

Отказоустойчивость и производительность играют важную роль, поскольку являются важными факторами удовлетворенности пользователей сервисов, развернутых за фасадом.

В случае фасада данных многократно увеличивается стоимость ошибки, поскольку сервис предоставляется пользователям пользователей фасада данных. Поэтому при возникновении отказа или невозможности пользоваться сервисом убытки сразу же распространяются на конечных пользователей решения.

Решение по миграции данных служит внутренним целям синхронизации данных, и ошибка интеграционного решения принесет убытки только непосредственным процессам-пользователям решения. Конечные пользователи решения могут испытывать некоторые неудобства из-за возможного устаревания части данных, однако они не попадают в ситуацию отказа в обслуживании и потери доступа к данным.

Кроме того, для решения интеграции с помощью фасада данных повышаются требования к производительности, поскольку обращения (запросы) к фасаду данных будут происходить от рабочих мест конечных пользователей, то есть число запросов к сервису выше, чем в решении

миграции данных (в последнем случае запросы формируются в рамках самого решения).

Следует отметить, что оба вида решения разделяют общую интеграционную составляющую, а именно: в обоих видах решения требуется преобразование данных и маршрутизация этих данных до конечного сервиса. Это требует выполнения настройки для задания преобразований данных и их маршрутизации. Исходя из требований к простоте администрирования, такая настройка должна производиться не программно, а в рамках некоторого специально разработанного для этих целей графического интерфейса.

Исходя из вышеперечисленного, кажется логичным, что решение по предоставлению пользователям фасадов данных может являться надстройкой над решением миграции данных. Соответственно, эти два типа решения представляют собой некоторую иерархию — базовым является решение по миграции данных, и, в первую очередь, должно быть спроектировано и разработано именно это решение. При необходимости, в ходе дальнейшего эволюционирования и развития разрабатываемой системы, может быть предложено и решение по предоставлению пользователям фасада данных.

Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 31.07.2012 г. № 14.514.11.4001 в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

Список литературы:

1. How to integrate with the cloud. David Linthicum. InfoWorld. [Электронный ресурс]. — <http://www.infoworld.com/d/cloud-computing/how-integrate-the-cloud-714>.
2. Интеграция: большой вызов облакам: [Электронный ресурс]. <http://www.mulesoft.com/integration-clouds-big-challenge>.
3. Integration Platform as a Service: Moving Intergation to the Cloud. Gartner RAS Core Research Note G00210747, Massimo Pazzini, Benoit J. Lheureux, 7 march 2011.
4. Новые тенденции интеграции в облаке. КубГАУ, №83 (09), 2012.

5. Measure Anything, Measure Everything. Ian Malpass. [Code as Craft](http://codeascraft.etsy.com/2011/02/15/measure-anything-measure-everything/). [Электронный ресурс], — <http://codeascraft.etsy.com/2011/02/15/measure-anything-measure-everything/>.

References

1. How to integrate with the cloud. David Linthicum. InfoWorld. [Electronic resource]. - [Http://www.infoworld.com/d/cloud-computing/how-integrate-the-cloud-714](http://www.infoworld.com/d/cloud-computing/how-integrate-the-cloud-714).

2. Integration: a great challenge clouds [electronic resource].
<http://www.mulesoft.com/integration-clouds-big-challenge>.

3. Integration Platform as a Service: Moving Intergration to the Cloud. Gartner RAS Core Research Note G00210747, Massimo Pazzini, Benoit J. Lheureux, 7 march 2011.

4. New trends of integration in the cloud. KubGAU, № 83 (09), 2012.

5. Measure Anything, Measure Everything. Ian Malpass. Code as Craft. [Electronic resource], - <http://codeascraft.etsy.com/2011/02/15/measure-anything-measure-everything/>.