

УДК 004.75

И.В. Бойченко, С.В. Корытников

Управление ресурсами в сервис-ориентированных системах типа «приложение как сервис»

Показано, что сервис-ориентированная архитектура (СОА) как основа для развития концепции «облачных вычислений», позволяет применить новые подходы к управлению ресурсами вычислительных систем. Предлагается рассматривать сервис как вычислительную задачу, предъявляющую конкретные требования к совокупности из 4 типов ресурсов вычислительных систем (ВС): процессору, оперативной памяти, подсистеме ввода-вывода, коммуникационным каналам. Диспетчеризация, основанная на априорно известных характеристиках задач (сервисов), позволит более эффективно управлять ресурсами ВС и точнее удовлетворять требованиям к качеству сервиса, с точки зрения известных критериев.

Ключевые слова: сервис-ориентированная архитектура, облачные вычисления, сервис, балансировка нагрузки, диспетчеризация, виртуализация.

Введение

Приведем два определения понятия «облачные вычисления».

1. Облачные вычисления – это окружение вычислительной среды, включающее множество заинтересованных сторон и предоставляющее измеряемый с различной степенью гранулярности информационный сервис на заданном уровне качества [1].

2. Облачная обработка данных (облачные вычисления) – это парадигма, в рамках которой информация постоянно хранится на серверах сети Интернет и временно кэшируется на клиентской стороне, например, на персональных компьютерах, игровых приставках, ноутбуках, смартфонах и т.д. [2].

Еще несколько лет назад понятие облачных вычислений было слишком размыто, этот термин часто подменялся другими: GRID, сервис-ориентированная архитектура (СОА) и др. В настоящее время многие понятия, связанные с распределенными вычислениями, обретают четкие контуры.

В первую очередь облачные вычисления (ОВ) – это концепция, охватывающая как организационные, так и технологические аспекты распределенных вычислений. Цель – предоставлять ресурсы для проведения вычислений (на любом уровне) в виде публичного сервиса, оплачиваемого потребителями в зависимости от затраченного времени и ресурсов.

ОВ – это не технология, это парадигма, призванная освободить пользователя или предприятие от необходимости поддерживать полномасштабную ИТ-инфраструктуру, позволяя решать любые информационно-вычислительные задачи посредством сервисов, доступных в глобальной сети. Такая возможность позволяет снизить не прямые затраты на продукцию и услуги. Стоимость аренды сервисов в долгосрочной перспективе существенно меньше, чем стоимость владения собственной ИТ-инфраструктурой.

С другой стороны, за счет консолидации вычислительных мощностей, предоставляемых множеству потребителей, поставщик услуг ОВ имеет существенные ресурсные резервы и возможность динамического перераспределения этих ресурсов в зависимости от текущих потребностей каждого отдельного пользователя.

Поставщики услуг ОВ должны гарантировать определенный уровень сервиса, например: обеспечивать минимальное время отклика, предоставлять определенный объем вычислительных мощностей за фиксированное время, предоставлять определенную функциональность и/или инфраструктуру, обеспечивать возможность надежного хранения данных, гарантировать заданную пропускную способность при обработке транзакций и т.д. При этом в любой момент потребителю могут потребоваться дополнительные ресурсы, которые со временем будут возвращены в общий пул. Такая бизнес-модель реализуется, в частности, компанией Amazon в виде сервиса EC2 (Elastic Computing). «Эластичность» этого сервиса как раз и заключается в том, что пользователь может «растягивать» свою задачу на любое количество вычислительных ресурсов, освобождая их по мере уменьшения нагрузки. За последние 10 лет подобные виды сервисов интенсивно развиваются множеством компаний, начиная от малоизвестных и заканчивая гигантами отрасли информационных технологий: Microsoft (сервис Azure), IBM (линейка сервисных продуктов IBM Cloud на базе мэйнфреймов), Yahoo (технология и сервис Hadoop), Hewlett Packard

(сервисы Cloud Discovery Workshop, Cloud Roadmap Service) и др. Также следует отметить, что некоторые крупные компании, кому это по силам, ведут разработки в рамках парадигмы ОБ для своих внутренних нужд. Такова, например, система хранения и поиска изображений издательского холдинга New York Times.

Мировое научное сообщество давно осознало, что облачные вычисления становятся новой парадигмой обработки данных. Университетскими центрами в сотрудничестве с IT-компаниями организованы исследовательские программы по развитию технологической платформы облачных вычислений. Так, в программе OpenCirrus (<http://opencirrus.org/>), организованной изначально компаниями Yahoo, Intel и HP, принимают участие Университет Иллинойса в Урбана-Шампейн (США), Технологический институт Карлсруэ (ФРГ), Институт системного программирования РАН (г. Москва) и др. В Евросоюзе (ЕС) принимаются объединенные усилия на межгосударственном уровне в рамках шестой и седьмой рамочных программ Еврокомиссии по информационно-коммуникационным технологиям (FP6-ICT, FP7-ICT). Под эгидой ЕС создана экспертная группа по облачным вычислениям, опубликовавшая аналитический отчет о проблемах и перспективах развития технологий ОБ [1].

В нашей стране с 2008 г. действует программа «Университетский кластер», целью которой является развитие и распространение технологий ОБ (<http://www.ispras.ru/unicluster/>). Программа «Университетский кластер» появилась как совместная инициатива компании Hewlett-Packard, национального оператора связи ЗАО «Синтерра» и учреждений Российской академии наук (ИСП РАН, МСЦ РАН). Участниками программы в настоящее время являются порядка 50 научных и научно-образовательных учреждений РФ, в том числе и Томский университет систем управления и радиоэлектроники.

1. Облачные вычисления и сервис-ориентированная архитектура

В целом сервис-ориентированную архитектуру (СОА) можно охарактеризовать как подход к разработке программного обеспечения, основанный на использовании служб (сервисов), со стандартизированными интерфейсами [3]. Интерфейс сервиса описывается с помощью формального платформонезависимого языка, например WSDL (Web Service Definition Language). Такой подход позволяет достаточно просто интегрировать логику и ресурсы ранее созданных информационных систем для решения новых задач. Отправной точкой в развитии СОА стали веб-сервисы, как расширения, позволяющие строить распределенные приложения на базе стека технологий Интернет. Веб-сервисы предоставили возможность считывать информацию с сайтов не только людям с помощью браузеров, но и специализированным приложениям. Практически сразу после появления первых стабильных реализаций (SOAP, XML-RPC) стало понятно, что этот инструмент подходит для интеграции произвольных приложений, а не только сайтов.

По сравнению с СОА облачные вычисления ставят более глобальную задачу – предоставить сервис вычислений любого уровня не только на уровне приложений, но и на уровне операционных систем, специализированных вычислительных ресурсов и т.д. Кроме того ОБ вводят дополнительные (не функциональные) требования к сервису. Так, в [1] отмечается, что системы ОБ должны обладать эластичностью, т.е. возможностью задействовать новые физические ресурсы при увеличении и освобождать ресурсы при уменьшении количества пользовательских запросов. Концепция ОБ охватывает больший круг вопросов, связанных с надежностью, адаптивностью, уровнем качества сервиса и многих других, имеющих отношение к построению и эксплуатации реально действующих систем, по сравнению с СОА, описывающей только схему построения приложений из компонентов.

В настоящее время сложилось три модели использования сервисов облачных вычислений:

1. Infrastructure as a Service (IaaS) – инфраструктура как сервис. IaaS предлагает доступ к низкоуровневым ресурсам: хранилищам данных, вычислительным устройствам и памяти. Естественно, доступ к таким ресурсам не может осуществляться напрямую. Здесь развиваются технологии виртуализации, использующие специальный управляющий процесс – гипервизор, который обеспечивает выполнение приложений (и даже операционных систем) пользователя в прозрачном режиме. Примеры: (хранилища) Amazon S3, SQL Azure, (вычисления) Amazon EC2, Zimory, Elastichosts.

2. Platform as a Service (PaaS) – платформа как сервис. Платформа – это прикладной программный интерфейс, обеспечивающий приложению возможность работы в условиях «облаков». Приложение фактически работает под управлением специализированной операционной системы, предоставляемой поставщиком ОБ. Между PaaS и IaaS отличие в масштабе ответственности программного слоя. IaaS может только гарантировать определенное количество процессоров или объем памяти, а все остальное должно делать разме-

щаемое пользователем приложение. PaaS – ситуация, когда поставщик фактически заявляет, что приложение пользователя будет работать эффективно, так как операционная система (платформа) оптимизирована для работы в «облаке». Примеры: Force.com, Google App Engine, Windows Azure (Platform).

3. Software as a Service (SaaS) – приложение как сервис. Поставщик реализует всю бизнес-логику в рамках определенного приложения. Примеры: Google Docs, Salesforce CRM, SAP Business by Design.

Нельзя утверждать, что ОБ являются надстройкой для СОА, или наоборот, СОА для ОБ. Интуитивно ощущается, что СОА предлагает удобный способ решения задач, стоящих перед облачными вычислениями. СОА для ОБ – это инструмент построения, важный, но не единственный.

2. Проблема управления ресурсами в сервис-ориентированных системах

Парадигма облачных вычислений подразумевает объединение в единое информационно-вычислительное пространство произвольного множества гетерогенных и пространственно-распределенных вычислительных узлов, находящихся в различных административных доменах, со своей локальной политикой безопасности. Управление ресурсами такого объединения совершенно не тривиальная задача.

Выделим особенности управления ресурсами, в зависимости от выделенных моделей использования (п. 1). Модели IaaS и PaaS предполагают, что динамические характеристики выполняемых приложений заведомо случайны, так как пользователь может решать произвольные задачи примерно так, как это происходит в операционных системах общего назначения. Модель SaaS отличается от моделей IaaS и PaaS тем, что здесь поставщик контролирует весь жизненный цикл приложения и может знать заранее статические и динамические требования к ресурсам.

Задача управления ресурсами осложняется еще и требованиями к качеству сервиса: время гарантированного отклика, уровень пропускной способности, доступность, отказоустойчивость, время восстановления после сбоя и т.д. Поэтому универсального решения задачи оптимального управления ресурсами в моделях IaaS и PaaS нет даже теоретически. В основном здесь ведутся исследования в области виртуализации элементов вычислительных систем. Современные достижения в этом направлении отражены в работах [4, 5].

Модель SaaS предоставляет больше возможностей по управлению ресурсами. Основная идея – зная характеристики задачи, более точно определять, когда и какие ресурсы должна получить задача с учетом текущего состояния ресурсов распределенной вычислительной системы.

В работе [6] предложено рассматривать 4 типа ресурсов (процессор, оперативная память, устройства ввода/вывода и устройства передачи данных) и принимать решение о балансировке исходя из учета потребностей выполняемой задачи именно в рамках этих четырех типов ресурсов. В этой же работе была предложена модель промежуточного программного обеспечения для балансировки нагрузки устройств ввода/вывода и сети в виде единой шины.

3. Иерархическое представление ресурсов и сервисов

Известно, что состав аппаратного обеспечения современных вычислительных систем становится все более разнообразным. Появляются специальные аппаратные ускорители счета, процессоры на базе программируемых интегральных логических схем, высокопроизводительные подсистемы хранения.

Любой новый ресурс в системе может быть отнесен к одному из классов по соответствующим признакам. Так, например, ресурс «графический процессор» будет отнесен к классу «вычислительный процессор».

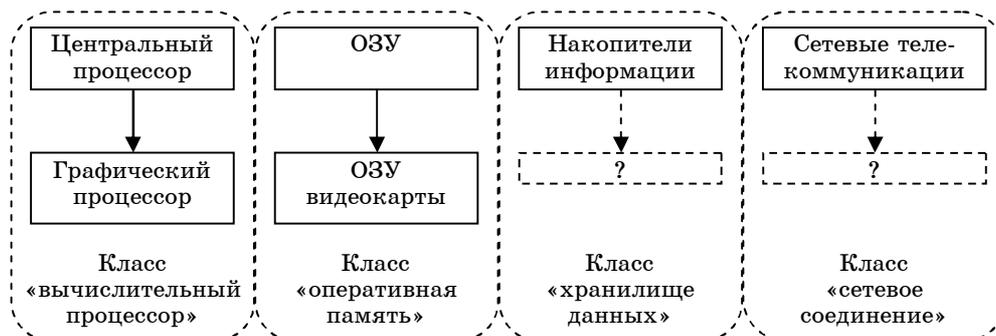


Рис. 1. Классификация вычислительных ресурсов

Такая организация позволяет применять один и тот же алгоритм балансировки, используемый в системе, ко всем ресурсам одного класса.

Также появилась необходимость выделить группы однотипных сервисов и представить их в виде иерархии. По нашему мнению, удачным является сравнение сервиса в СОА с понятием объекта (класса) в концепции объектно-ориентированного программирования, где существуют механизмы наследования. Если обратиться к основам построения информационно-вычислительных систем, то, например, операционная система предоставляет сервисы (набор API) приложениям для работы с файловой системой, сетевым окружением и т.д. Далее прикладные программы и программы ОС так же могут предоставлять более высокоуровневые сервисы либо пользователю напрямую, либо другим приложениям. Такой взгляд позволяет неограниченно расширять наборы сервисов, базирующихся на других низкоуровневых сервисах и предоставляющих новую функциональность, не меняя при этом алгоритмы диспетчеризации.

Общая схема СОА выделяет три роли (рис. 2): реестр сервисов, поставщик сервиса, потребитель сервиса.

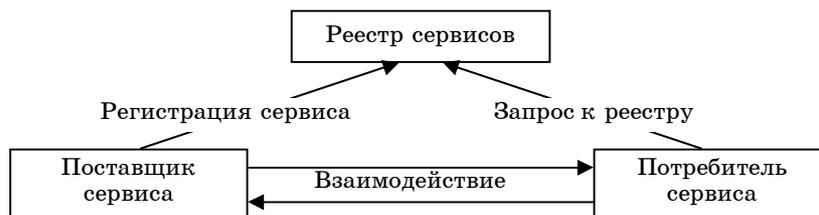


Рис. 2. Общая схема СОА

Мы предлагаем расширить функциональность реестра сервисов таким образом, чтобы внутри четко описывалась вертикальная иерархия сервисов (рис. 3). Например, сервис 5 наследует функциональность сервисов 1 и 2, а сервис 6 – функциональность сервиса 3.

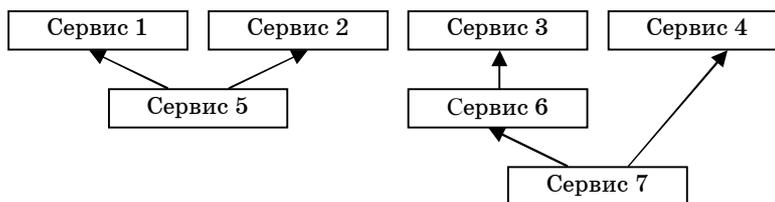


Рис. 3. Пример иерархии вычислительных сервисов

На реестр сервисов ложится дополнительная функциональность о своевременном предоставлении потребителю сервиса всех необходимых базовых сервисов, от которых унаследован дочерний сервис.

При управлении ресурсами менеджер ресурсов решает, какие устройства задействовать для наиболее эффективного выполнения вычислений, заложенных в функциональности сервиса.

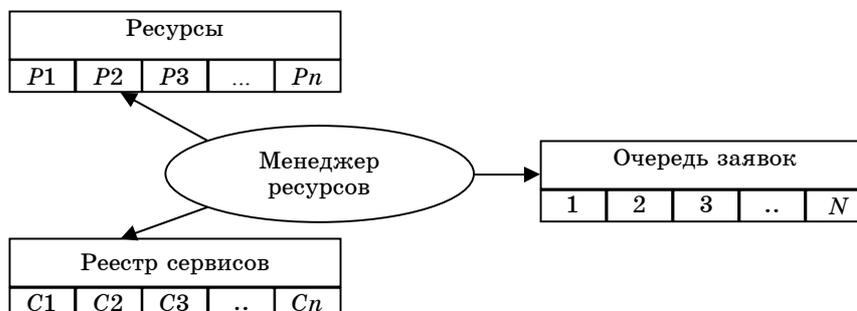


Рис. 4. Менеджер ресурсов в системе с СОА

Обратим внимание, что достаточно сложно предположить, насколько интенсивно сервисом будет использоваться тот или иной вычислительный ресурс, для этого принято решение снабжать сервис метаданными, характеризующими статические и динамические потребности в ресурсах.

Заключение

В статье предложен подход к построению алгоритмов балансировки нагрузки (диспетчеризации) в сервис-ориентированных системах на основе учета априорной информации о потребности исполняемых задач (сервисов) в наборе ресурсов четырех типов (вычислитель, оперативная память, подсистема хранения, сетевой канал). Предполагается, что выбранный подход обеспечит построение более эффективных алгоритмов диспетчеризации в модели «приложение как сервис». В настоящее время нами предпринимаются усилия по разработке и исследованию алгоритмов такого типа.

Литература

1. Jefferey K. The future of cloud computing / K. Jefferey, B. Neidecker-Lutz // Cloud Computing Expert Group Report 2009 [Электронный ресурс]. – Режим доступа: <http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf>, свободный (дата обращения: 15.04.2010).
2. Hewitt C. ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing [Электронный ресурс]. Режим доступа: <http://www.computer.org/portal/web/csdl/doi/10.1109/MIC.2008.107>, для зарегистрированных пользователей (дата обращения: 15.04.2010).
3. Сервис-ориентированная архитектура. Википедия свободная энциклопедия [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура, свободный (дата обращения: 15.04.2010).
4. Harchol-Balter M. Exploiting Process Lifetime Distributions for Dynamic Load Balancing / M. Harchol-Balter, A.B. Downey // ACM Trans. Computer Systems. – 1997. – Vol. 15, №3. – P. 253–285.
5. Zhang X.-D. Improving Distributed Workload Performance by Sharing Both CPU and Memory Resources / X.-D. Zhang, L. Xiao, Y.-X. Qu // Proc. 20th Int'l Conf. Distributed Computing Systems (ICDCS '00). – 2000. – P. 233–241
6. Qin X. Communication-Aware Load Balancing for Parallel Applications on Clusters / X. Qin, H. Jiang, A. Manzanares et al. // IEEE Transactions on computers. – 2010. – Vol. 59, №1. – P. 42–52.

Бойченко Иван Валентинович

Докторант каф. автоматизированных систем управления ТУСУРа
Тел.: 8-906-958-24-83
Эл. почта: biv@asu.tusur.ru

Корытников Сергей Викторович

Аспирант каф. автоматизированных систем управления ТУСУРа
Тел.: 8-923-415-94-36
Эл. почта: biox@ms.tusur.ru

Boichenko I.V., Korytnikov S.V.

Resource management in the service-oriented systems of the «software as a service» type

It is shown, that service-oriented architecture enables to apply new approaches to resource management of computing systems as a basis for “Cloud Computing” concept development. It is offered to consider service as the computing task, which makes specific demands to set from 4 types of computing systems resources: processor, random-access memory, I/O system, and network. The dispatching based on a priori known characteristics of the tasks (services) allows to control computer resources more effectively and to satisfy the quality of service requirements more accurately from the point of view of predefined criteria.

Keywords: service-oriented architecture, cloud computing, service, load-balancing, dispatching, virtualization.