Новые вехи в развитии систем массового обслуживания

Душкин Д.Н.
Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
ddushkin@asmon.ru

Каргин А.Н.
Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
akargin@asmon.ru

Фархадов М.П.
Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
mais@ipu.ru

1. Введение

Сфера систем массового обслуживания (СМО) за более чем 100-летний срок своего развития прошла длинный путь. Первые проблемы, которые необходимо было решать в сфере СМО, были расчет и определение оптимальной стратегии работы телефонных станций в начале XX века. В настоящее время центральным вопросом в разработке СМО является создание оптимальной архитектуры и алгоритмов обработки запросов в компьютерных сетевых системах.

Благодаря многолетним исследованиям и разработкам в настоящее время такие крупные сайты, как youtube.com, могут обслуживать более 3 миллиардов посетителей в день [1].

Каким образом это удается инженерам компании Google и других крупных ИТ компаний? Какой математический аппарат лежит в основе используемых технологий? Какую архитектуру должны иметь такие системы? Ответы на эти вопросы есть в области создания современных систем массового обслуживания.

2. Высоконагруженные системы

Современные крупномасштабные системы генерируют большие объёмы данных, что повышает общую загруженность систем. Однако эти данные могут быть полезны для улучшения и развития системы. Как отмечено в Клермонтском отчёте об исследованиях в области баз данных: «данные — это ключ к достижению эффективности и прибыльности бизнеса» [2]. Соответственно, накопленные данные необходимо не только хранить, и оперативно реагировать на запросы пользователя, сообщая ему требуемую выдачу, а также собранные данные необходимо различным образом обрабатывать: анализировать, производить над ними операции и на их основании строить различные модели. Указанные особенности диктуют новые задачи для методов анализа данных и параллельной обработки. В отличие от централизованных баз данных, при распределенной архитектуре отсутствует однозначность в месте нахождения необходимых данных, поэтому невозможно выполнить единый запрос к целевому серверу хранения данных. Вместо этого пользователю необходимо вызвать отдельный запрос на каждом хранилище данных, собрать результаты и осуществить многоуровневую агрегацию для всех его результатов.

Для ускорения процессов обработки, используется распараллеливание запросов для выполнения на каждом хранилище данных.

Как можно предположить, круг решаемых проблем достаточно обширный. Однако мы считаем разумным выделить два ключевых проблемных

полюса: обеспечение высокой интенсивности вычислений и хранение больших объёмов данных. Для каждой из этих проблем разрабатываются свои методы решения.

Для решения первой проблемы часто используется модель распределённых вычислений, которую принято называть *грид-вычисления*. Грид — это сервис-ориентированная архитектура, задача которой заключается в динамическом объединении (виртуализации) распределённых вычислительных интернет-ресурсов в единый ансамбль для выполнения масштабных информационно-вычислительных задач. Для сервис-ориентированных грид-систем определена спецификация архитектур открытых грид-сервисов (OGSA - Open Grid Services Architecture) [4].

Как правило, топология грид состоит из трёх уровней:

- планировщик,
- менеджер выполнения,
- вычислительный агент — исполнитель.

Существует и другая упрощенная грид модель – мастер/исполнитель. Такая модель не требует планировщика, а балансировка загрузки осуществляется с помощью механизма отчётов.

Решение второй проблемы также находится в сфере распределённых вычислений. Одним из наиболее успешных решений является программная модель от Google, *MapReduce* [3], в основе которого лежит функциональный подход к программированию.

В языке Lisp, откуда берёт своё начало концепция MapReduce, функция “map” принимает на вход функцию и последовательность значений. Затем “map” применяет функцию к каждому значению последовательности. “Reduce” объединяет элементы последовательности, используя бинарную операцию. Эта

логика была также реализована в модели MapReduce. В архитектуре MapReduce используется паттерн, или шаблон проектирования, мастер/исполнитель.

В настоящее время эти подходы являются наилучшим решением для своего класса задач. Тем не менее, с ростом систем растёт число записей в распределённых файловых системах, а с ростом бизнеса и технологий растёт также область задач по обработке данных, размывая границы между отмеченными полюсами. В такой ситуации логично подумать об оптимизации систем и об унификации архитектурных компонент. Шагом в сторону такой унификации могут стать *сетецентрические технологии* как единая среда для реализации концепции сервис-ориентированных систем.

3. Сетецентрические технологии

Сетецентризм (network-centric) – соотносится с системами и шаблонами поведения, значительно подверженным влиянию сети или включенным в сеть и в сетевые технологии. Часто в центр помещается ip-сеть, но иногда этот термин употребляется для обозначения использования сети любого типа.

Интероперабельность – одна из ключевых характеристик СМО, ориентированных на предоставление услуг другим системам, и является центральной проблемой сферы сетецентрических технологий.

Сервис-ориентированная архитектура (СОА) (Service-oriented architecture - SOA) – это архитектурный подход к определению, связыванию и интеграции повторно используемых бизнес-сервисов, имеющих четкие границы и самодостаточных по своей функциональности [5]. В рамках такой архитектуры можно организовывать бизнес-сервисы в бизнес-процессы. Внедряя концепцию сервисов (более высокого уровня абстракции, не зависящего от приложений и платформы информационной инфраструктуры, а также от контекста или других сервисов), СОА переносит информационные

технологии на следующий уровень, более подходящий для обеспечения функциональной совместимости и реализации в гетерогенных средах.

Архитектура сервисных компонент (ACK) (Service Component Architecture - SCA) – набор спецификаций, разработанный для описания модели создания приложений и систем на базе сервис-ориентированной архитектуры [6]. ACK предоставляет модель построения приложений, состоящих из совокупности сервисов и бизнес-функций. ACK – совместная инициатива ряда крупнейших ИТ компаний, таких как: BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase и др.

В настоящее время не существует методик оценки QoS (англ. Quality of Service — качество обслуживания) и формирования SLA (англ. Service Level Agreement — Соглашение об уровне предоставления услуги) для систем, построенных на базе ACK. В рамках исследовательского проекта Sensoria [7] были разработаны качественные и количественные оценки COA, предложена альтернатива ACK. Однако к моменту завершения Sensoria спецификация ACK стала стандартом, поэтому собственный программный фреймворк Sensoria JOLIE стал неактуальным.

В рамках исследовательской работы в лаборатории №17 ИПУ РАН была разработана концептуальная схема системы «Сурдо.Облако» (рис. 1) на базе архитектуры сервисных компонент. «Сурдо.Облако» является открытой системой, позволяющей разработчикам использовать её отдельные элементы и ресурсы для разработки новых социальных приложений из области русского жестового языка, дактильного языка и других смежных тематик. В состав системы планируется включить существующий сервис, разработанный в лаборатории, «Сурдосервер» (surdoserver.ru) [].

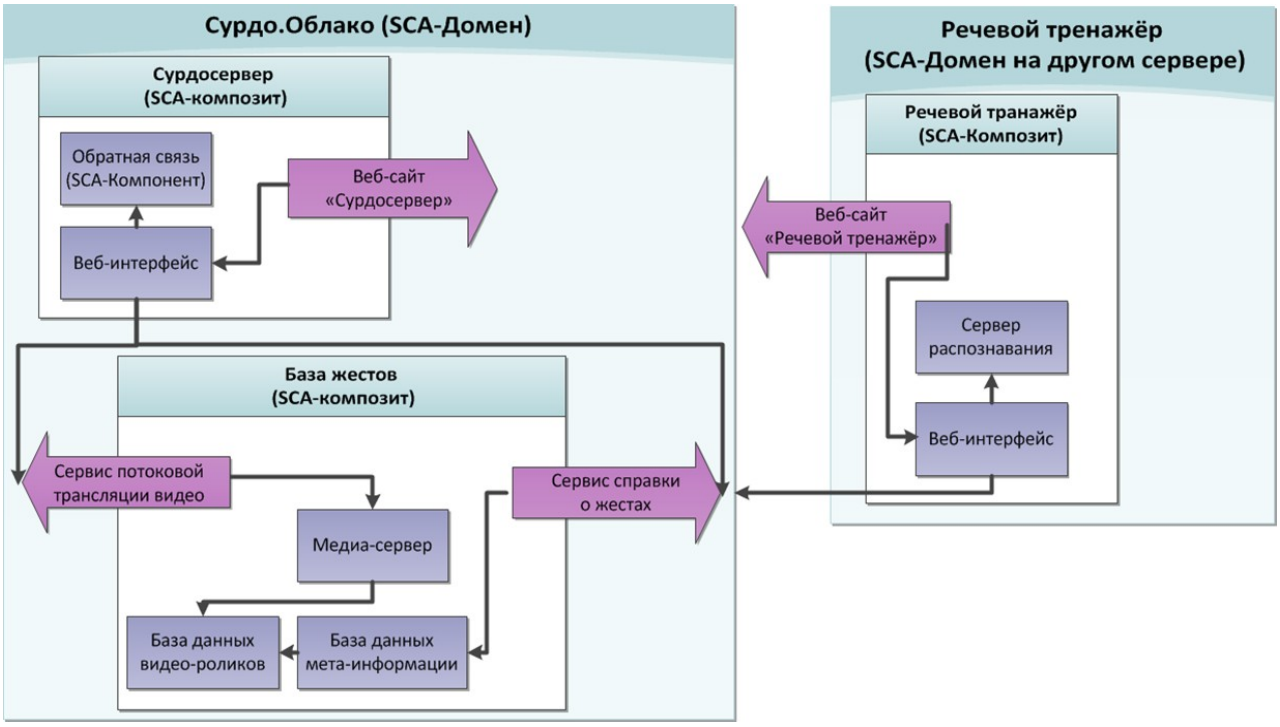

Рисунок 1. Система Сурдо.Облако на базе АСК

4. Заключение

Перспективным направлением развития СМО является применение техник создания систем с высокой нагрузкой в архитектуре сервисных компонент с адаптацией и развитием математического аппарата качественного и количественной оценки QoS и SLA таких систем.

Литература

1. *Robin Wauters*, YouTube Turns 6 Years Old, Daily Views Shoot Up To 3 Billion,Tech Crunch, URL: http://tcrn.ch/kHfYXH, 2011
2. *Ракеш Агравал*, *Анастасия Айламаки* и др. Клермонтский отчет об исследованиях в области баз данных, Калифорнийский университет в Беркли, URL: http://citforum.ru/database/articles/claremont_report/, 2008
3. Introduction to Parallel Programming and MapReduce, Google Code University, URL: http://code.google.com/intl/ru-RU/edu/parallel/mapreduce-tutorial.html

4. *I. Foster, H. Kishimoto,* The Open Grid Services Architecture, URL: http://www.gridforum.org/documents/GFD.80.pdf
5. *Мабрук М.* Краткие основы SOA [Электронный ресурс] // IBM developer works. 2010. URL: http://www.ibm.com/developerworks/ru/edu/ws-soa-ibmcertified/index.html
6. *Beisiegel M., Blohm H.* Service Component Architecture Building Systems using a Service Oriented Architecture. BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase, 2005
7. *Wirsing M.* SENSORIA Software Engineering for Service-Oriented Overlay Computers Result brochure. SENSORIA Consortium, 2010.
8. *Абраменков А.Н., Душкин Д.Н., Мясоедова З.П., Мясоедова М.А., Паршакова А.А., Петухова Н.В., Фархадов* М.П. Интернет сервис для людей с нарушениями слуха «СУРДОСЕРВЕР» // Proceedings of International Workshop "Distributed Computer and Communication Networks DCCN'2010". Moscow, Russia. M.: R&D Company "Information and Networking Technologies", 2010. P. 331-338.
9. *Абраменков А.Н., Душкин Д.Н., Мясоедова З.П., Мясоедова М.А., Паршакова А.А., Петухова Н.В., Фархадов* М.П. Интернет портал «Сурдосервер» // Материалы Первой Международной конференции "Интеллектуальные технологии и средства реабилитации людей с ограниченными возможностями" (ИТСР-2010). М.: Московский государственный социально – гуманитарный институт, Федеральный учебно-методический центр по обучению лиц с нарушением опорно-двигательной системы. 2010. 5 с. http://www.mgsgi.ru/up/1/programma.pdf