ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ

Кафедра информационных систем

УТВЕРЖДАЮ

Заведующий кафедрой

_____

«___» _____20__ г.

ЛЕКЦИЯ №5

по дисциплине «Облачные технологии»

Тема №1 Основы облачных вычислений

Занятие №4 Облачные хранилища

для студентов направления 38.03.05 «Бизнес-информатика»

ШИФР наименование

Рассмотрено УМК

" " _____ 20___ года

протокол N _____

Ставрополь, 2023

Цели лекции:

1. Сформировать информационно-наглядное представление о специфике организации облачного хранилища.
2. Дать характеристику понятию архитектура облачных приложений.
3. Показать особенности транзакционных вычислений.
4. Показать актуальность и значимость современных облачных технологий в ведении бизнеса.

Время: _____ 90 мин.

Учебно-материальное обеспечение:

1. Опорная лекция.
2. ГОС ВО по направлению подготовки.
3. Рабочая программа дисциплины.
4. Основная и дополнительная литература.

Распределение времени:

I. Вступительная часть	5 мин.
II. Основная часть:	
1. Понятие облачного хранилища	25 мин.
2. Архитектуры облачных приложений	25 мин.
3. Транзакционные вычисления	30 мин.
Ш. Заключительная часть	5 мин.

Вводная часть

Облачное хранилище данных (англ. cloud storage) — модель онлайн-хранилища, в котором данные хранятся на многочисленных распределённых в сети серверах, предоставляемых в пользование клиентам, в основном, третьей стороной. В отличие от модели хранения данных на собственных выделенных серверах, приобретаемых или арендуемых специально для подобных целей, количество или какая-либо внутренняя структура серверов клиенту, в общем случае, не видна. Данные хранятся и обрабатываются в так называемом облаке, которое представляет собой, с точки зрения клиента, один большой виртуальный сервер. Физически же такие серверы могут располагаться удалённо друг от друга географически, вплоть до расположения на разных континентах.

Первый учебный вопрос – Понятие облачного хранилища

Абстрагирование от аппаратных средств в облаке осуществляется не только благодаря замене физических серверов виртуальными. Виртуализации подлежат и системы физического хранения данных.

Облачное хранилище позволяет перебрасывать данные в облако, и при этом не беспокоиться о том, как именно они хранятся, и не задумываться об их резервном копировании. Когда данные, перемещенные в облако, понадобятся снова, достаточно будет просто обратиться в облако и получить данные. При этом пользователь может и не знать, как хранятся эти данные, где они хранятся, или того, что происходит с тем или иным оборудованием, когда он периодически перемещает данные в облако и извлекает их оттуда.

Как и в случае с другими элементами облачных вычислений, на рынке предлагается несколько подходов к облачному хранилищу. Говоря общими словами, они связаны с разбиением данных на небольшие цепочки и хранение их на множестве серверов. Цепочки данных снабжаются индивидуально вычисленными контрольными суммами, так, чтобы данные можно было быстро восстановить, вне зависимости от того, что могло бы произойти в течение времени хранения с накопителями, физически хранящими данные, и скомпрометировать облако.

Как правило, пользователи начинающие работать с облаком, пытаются обращаться с облачным хранилищем так, как если бы оно представляло собой сетевой накопитель. С точки зрения принципа работы облачное хранилище принципиально отличается от традиционных сетевых накопителей, и служит принципиально другим целям. Облачное хранилище имеет тенденцию работать гораздо медленнее и имеет гораздо бóльшую степень структурированности, вследствие чего его использование в качестве оперативного хранилища данных непрактично, вне зависимости от того, работает ли приложение, использующее эти данные, в облаке или где-то еще.

Облачное хранилище не подходит для оперативного использования

облачными приложениями на базе транзакций. Об облачном хранилище можно думать примерно так же, как об аналоге резервной копии на ленточном носителе, но, в отличие от системы резервного копирования с ленточным приводом, при работе в облаке вам не нужны ни привод, ни ленты.

Второй учебный вопрос - Архитектуры облачных приложений

Архитектуры облачных приложений реализуются базовыми концепциями, к которым можно отнести **концепцию Grid Computing**

Концепция Grid Computing представляет собой архитектуру приложений, предоставляющую самый простой метод перехода к облачной архитектуре. Приложения, использующие грид-технологии, представляют собой ПО, интенсивно потребляющее ресурсы процессора. Грид-приложения разбивают операции по обработке данных на небольшие последовательности элементарных операций, которые могут выполняться изолированно.

В русскоязычных ресурсах встречается и такое написание, как "грид" (англ. grid — решетка, сеть) — согласованная, открытая и стандартизованная компьютерная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение вычислительных ресурсов и ресурсов хранения информации, которые являются частью этой среды, в рамках одной виртуальной организации.

Примером грид-вычислений является проект по поиску внеземных цивилизаций SETI (Search for Extra-Terrestrial Intelligence), который использует радиотелескопы, ведущие постоянное наблюдение за активностью в космосе. Эти радиотелескопы собирают гигантские объемы данных, которые затем нуждаются в обработке с целью поиска сигналов, отличающихся от естественных — обнаружение сигнала, который может иметь искусственное происхождение, может служить признаком деятельности внеземной цивилизации. Обработка такого объема информации на одном компьютере потребует такого длительного времени, что вполне может статься, человечество к завершению этих вычислений действительно уже освоит межзвездные перелеты. Но вот множество компьютеров, каждый из которых использует для решения этой проблемы свои холостые циклы CPU, могут справиться с задачей намного быстрее.

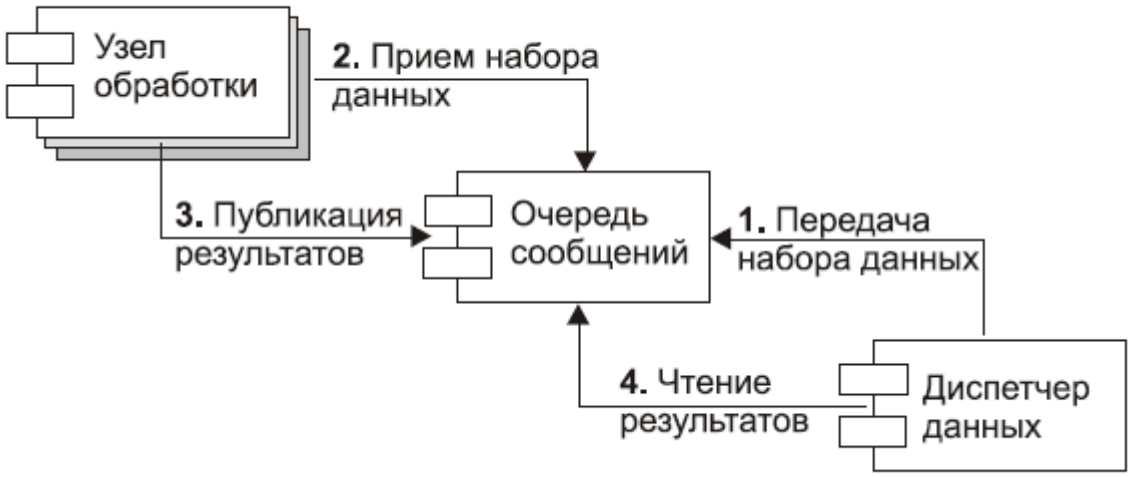
Эти компьютеры, на которых работает приложение SETI@home

(возможно, включая и ваш собственный компьютер), формируют сеть или "решетку" (grid). Каждый раз, когда у них есть свободные такты процессора, компьютеры направляют запросы серверам проекта SETI на получение наборов данных для обработки. Они обрабатывают наборы данных и передают результаты обратно серверам проекта SETI. Результаты, полученные одним из участников проекта, перепроверяются повторной обработкой на компьютерах других участников, а затем производится дальнейшая проверка наиболее интересных результатов. В 1999 г. участники программы SETI предложили использовать холостые циклы настольных компьютеров обычных пользователей для обработки данных проекта. Коммерческие и правительственные организации сформировали сеть из суперкомпьютеров для выполнения той же задачи. Впоследствии были созданы пулы серверов (так называемые "серверные фермы", server farms) для формирования сети, выполняющей такие задачи, как рендеринг видео (визуализация изображений и обсчет последовательностей кадров). Как суперкомпьютеры, так и серверные фермы оказались очень дорогими подходами к грид-вычислениям и требовали крупных капиталовложений.

Облачная инфраструктура серьезно упрощает и удешевляет создание грид-приложений. Если у вас есть данные, нуждающиеся в обработке, вы просто запрашиваете сервер для обработки этих данных. После завершения обработки этот сервер может быть остановлен или может запросить для обработки новую порцию данных.

На Слайде №____ проиллюстрировано, как работает грид-приложение. Сначала сервер или кластер серверов получает набор данных, которые нуждаются в обработке. На первом шаге эти данные передаются в очередь сообщений (1). Другие узлы, которые часто называются серверам обработки (или, как в случае SETI@home — другими настольными компьютерами), наблюдают за очередью сообщений (2) и ждут появления новых наборов данных. Когда набор данных появляется в очереди сообщений, он обрабатывается первым же компьютером, который его обнаружил, и

результаты отсылаются обратно в очередь сообщений (3), откуда они считываются сервером или кластером серверов (4). Оба компонента могут работать независимо друг от друга, и каждый из них может работать даже в том случае, если второй компонент не работает ни на одном компьютере.



Слайд №___ Архитектура грид-приложений отделяет основное приложение от узлов, осуществляющих обработку данных

Здесь приходят на помощь облачные вычисления, потому что при их использовании пользователи не должны иметь никаких серверов в личной собственности. Таким образом, возможно масштабировать выделяемые пользователю вычислительные мощности, запрашивая такие количества серверов, которые необходимы для переработки наборов данных, передаваемых приложению пользователя. Иными словами, вместо того, чтобы заставлять компьютеры, работающие вхолостую, заниматься обработкой данных по мере их поступления, вы можете сделать так, чтобы серверы включались, когда поток данных становится более интенсивным, и выключались по мере того, как интенсивность потока данных ослабевает.

Грид-приложения имеют очень ограниченную область применения (переработка больших объемов научных и финансовых данных).

Третий учебный вопрос - Транзакционные вычисления

Большая часть бизнес-приложений использует транзакционные вычисления. Транзакционная система — это система, в которой один или большее количество поступающих наборов данных обрабатываются совместно в рамках единой транзакции и устанавливают взаимосвязи с другими данными, уже введенными в систему. Основой транзакционной системы обычно является реляционная база данных, которая управляет взаимоотношениями между всеми данными, образующими систему.

На Слайде №____ показана логическая структура транзакционной системы высокой отказоустойчивости.

Слайд №___ Логическая структура транзакционной системы высокой отказоустойчивости

При использовании архитектуры этого типа сервер приложений обычно обрабатывает данные, хранящиеся в базе данных, и представляет их через Web-интерфейс. За счет этого пользователь может работать с данными. Большинство Web-сайтов и Web-приложений, которые использует

пользователь в своей повседневной работе, представляют собой ту или иную форму транзакционной системы. Для обеспечения высокой отказоустойчивости все эти компоненты могут формировать кластеры, а бизнес-логика и логика представления данных могут прятаться за балансировщиком нагрузки. Балансировщик нагрузки (load balancer) — это программный или аппаратный компонент сети, который распределяет процесс выполнения заданий между несколькими серверами сети с тем, чтобы оптимизировать использование ресурсов и сократить время вычисления.

Развертывание транзакционной системы в облаке несколько сложнее и не настолько самоочевидно, как развертывание грид-системы. В то время как в грид-системе узлы должны быть короткоживущими, в транзакционной системе они должны быть долгоживущими.

Ключевой проблемой для любой системы, нуждающейся в долгоживущих узлах в облачной инфраструктуре, является тот основополагающий факт, что среднее время наработки на отказ (MTBF) для виртуального сервера обязательно меньше. В предельно упрощенной формулировке эта проблема выражается так: если у вас есть два физических сервера, среднее время наработки на отказ для которых составляет три года, то вероятность сбоя во всей системе у вас будет меньше, чем если бы вы имели только один физический сервер, на котором работают два виртуальных узла. Фактически количество физических узлов управляет показателем MTBF, а так как физических узлов меньше, в вашей облачной транзакционной системе для физических узлов MTBF будет выше, чем для виртуальных.

В транзакционных приложениях функции подразделяются на представление данных, реализацию бизнес-логики и хранилище данных.

Однако облачные вычисления предлагают ряд подходов, которые помогают не только сгладить эту проблему, но и потенциально увеличить общий показатель MTBF для транзакционной системы в целом.

Заключение

В транзакционных приложениях функции подразделяются на представление данных, реализацию бизнес-логики и хранилище данных.

Однако облачные вычисления предлагают ряд подходов, которые помогают не только сгладить эту проблему, но и потенциально увеличить общий показатель MTBF для транзакционной системы в целом.

Доцент кафедры «Информационных систем»
к.т.н., доцент В.Е. Рачков

«___»_____20___ г.