

вания реляционных СУБД их роль свелась к применению в специализированных хранилищах (например, иерархических службах каталогов). Новый рост интереса к отходу от реляционной модели (и появление нереляционных СУБД нового поколения) инициирован в первую очередь потребностью в параллельных распределённых системах для высокомасштабируемых интернет-приложений, таких как поисковые системы. Также интересу к отходу от реляционной модели способствовал ряд постепенно выкристаллизовавшихся проблем традиционных SQL-СУБД:

- Ограничения по масштабированию. РСУБД более доступно вертикальное масштабирование, то есть повышение общей производительности системы за счет применения более мощных компонентов. Горизонтальное масштабирование, то есть увеличение количества серверов (или компонентов, разнесенных на отдельные серверы), выполняющих параллельную обработку, требует в их случае сложных технических решений.

- Большие данные. Сложность хранения огромных объемов данных определяется проблемой горизонтального масштабирования.

- Для реляционных СУБД характерны достаточно низкая скорость операций записи (особенно с индексами), что может быть критично в ряде задач; отказ же от индексов ускоряет запись, но сильно замедляет операции чтения данных.

- Также реляционная модель хранения данных плохо приспособлена для ситуаций, требующих оперативных изменений в схеме (структуре хранимых данных).

- Наконец, при соблюдении требований нормализации, большие схемы оказываются достаточно сложными для понимания человеком.

Разумеется, преимущества реляционных СУБД, благодаря которым они стали доминирующими на рынке систем хранения данных, по-прежнему актуальны. В первую очередь это транзакционная модель и соответствующий ей набор требований, описываемый аббревиатурой ACID (Atomicity, Consistency, Isolation, Durability – Атомарность, Согласованность, Изолированность, Надежность).

Под транзакцией понимается группа последовательных операций, которая представляет собой логическую единицу работы с данными. В рамках транзакционной модели группа команд выполняется либо полностью, либо не выполняется ни одна команда в группе (атомарность), что исключает «застывание» данных в промежуточном состоянии. При этом данные должны отражать реальность в момент получения ответа на запрос (согласованность), а параллельные транзакции не должны влиять друг на друга (изолированность). Наконец, результат выполнения транзакций не должен зависеть от сбоев аппаратного обеспечения (надежность).

Это может показаться неочевидным, но на практике отказ от реляционной СУБД обычно означает расставание с транзакционной моделью. Разработчики вынуждены обходить проблему, реализуя поддержку транзакций на уровне приложения либо закладывая издержки на неатомарность в рабочем процессе, в котором используются данные. Однако реализация транзакционной модели в СУБД остается критическим фактором, обеспечивающим реляционным базам данных уверенные позиции на рынке.

Тем не менее накопившиеся изменения в задачах, требующих применения систем хранения и доступа к данным (в плане вычислительных мощностей, параллелизма либо специфики самих обрабатываемых данных), делают отход от реляционной модели вос-

требуемым и обоснованным в ряде приложений. Взамен ACID современные нереляционные СУБД ориентируются на следующий набор требований:

- базовая доступность (basic availability), то есть гарантированное завершение всех запросов (раньше или позже, успешно или нет);
- гибкое состояние (soft state) – возможность изменения состояния системы во времени независимо от наличия или отсутствия внешних воздействий (например, из-за процессов увеличения согласованности данных);
- финальная согласованность (eventual consistency), допускающая рассогласованность данных на какое-то время при условии, что они в конечном счете придут к согласованию.

СУБД, соответствующие перечисленным положениям, на данный момент можно отнести к следующим категориям:

- баз данных на основе модели «ключ-значение» (key-value; характерный пример – СУБД Redis);
- базы данных на основе столбцов (column; характерный пример – СУБД Cassandra);
- документарная модель представления данных (document; популярный представитель данного семейства СУБД – MongoDB);
- графовые базы данных (graph; характерный пример – СУБД Neo4j);
- базы данных хранения временных рядов (time-series database; например, СУБД Prometheus).

Особенность баз данных на основе пар «ключ-значение» заложена в их названии: ключ используется для доступа к значению. Эти СУБД отличаются тем, что поддерживают высокую разделяемость данных и обеспечивают хорошее горизонтальное масштабирование. Такая система хранения хорошо подходит для изображений и кэширования различной информации, а примерами использования являются игровые и рекламные приложения. В ряде случаев одновременно с этой моделью реализуют резидентное хранение данных (in-memory database или база данных, размещаемая в ОЗУ), для задач, требующих лучшего быстродействия, чем могут предоставить традиционные СУБД (некоторые из которых, впрочем, также поддерживают опционально резидентный режим работы).

Системы на основе столбцов хранят данные в виде разреженной матрицы, строки и столбцы которой используются как ключи. Типичными областями применения таких баз являются веб-индексирование и задачи, связанные с большими данными, вынужденно допускающие меньшую согласованность.

Преимуществом документарных баз данных является то, что они позволяют разработчикам хранить и запрашивать данные с помощью той же документной модели, которую те используют в коде приложения. Гибкий, полуструктурированный, иерархический характер документов и соответствующих баз данных позволяет схеме эволюционировать в соответствии с потребностями приложения. Документная модель хорошо работает в каталогах, пользовательских профилях, системах управления контентом (где каждый документ уникален, а его схема изменяется со временем). При этом в коде приложения данные часто представлены как объект или документ в JSON-подобном формате, который поддерживается и самой СУБД.

Графовые базы данных упрощают разработку и запуск приложений, работающих с наборами сложносвязанных данных. Типичные примеры их использования – социальные сети, рекомендательные сервисы, системы выявления мошенничества, графы знаний.

Базы данных временных рядов – это специализированные СУБД для хранения проиндексированных по времени данных. Причины их появления – необходимость сбора, хранения и обработки больших массивов разных метрик (системы мониторинга), а также то, что реляционная модель в системах со сложной логикой и высокими объемами поступающих разнородных метрик – не самый практичный выбор.

Таким образом, современные нереляционные базы данных оптимизированы для приложений, которые работают с чрезвычайно большими объемами данных, нуждаются в низкой задержке или гибких моделях представления данных, что достигается ценой смягчения жестких требований к непротиворечивости и согласованности данных, характерных для реляционных СУБД, и выражается в невозможности достижения сравнимой с ними универсальности.

Список цитированных источников

1. Knowledge Base of Relational and NoSQL Database Management Systems. [Электронный ресурс]. – Режим доступа: <https://db-engines.com/en/> – Дата доступа: 06.11.2018.

2. Introduction to Time Series Database. Режим доступа: <https://www.linkedin.com/pulse/introduction-time-series-database-pinglei-guo/> – Дата доступа: 15.09.2019.

УДК 004.67

ПРИМЕНЕНИЕ ФИЛЬТРА КАЛМАНА В ЗАДАЧЕ ВОССТАНОВЛЕНИЯ ГЕОКООРДИНАТ

Ковалюк П. А.

Брестский государственный технический университет, г. Брест, Беларусь

Научный руководитель: Кузьмицкий Н. Н., канд. тех. наук

Введение

Проблемы определения координат можно классифицировать следующим образом:

- выбросы координат;
- «блуждание» координат при нулевой или близкой к нулю скорости;
- отклонение координат вблизи высотных зданий;
- ошибка времени фиксации координат.

Под выбросами понимают ошибочные данные, не отражающие реальную ситуацию, которые были получены в результате технической ошибки аппаратуры, входящей в состав терминала или алгоритмической ошибки модуля определения координат 1.

Отклонение координат вблизи высотных зданий также наблюдается при отражении сигнала от зданий, что приводит к смещению объекта от реального местоположения.