***34. Транзакционная память***

Реляционная база данных — табличное представление конкретной модели. Каждая таблица описывает отношение между полями.

Есть проблема синхронизации.

Для БД была предложена концепция транзакции.

Транзакция: Атомарная, Согласованная, Изолированная, Устойчивая.

(По-английски аббревиатура ACID)

* Атомарность — «все или ничего», если часть транзакции не выполняется, не выполняется и вся транзакция и состояние памяти остается неизменным.
* Согласованность — любая транзакция переводит память из одного допустимого состояния в другое.
* Изолированность — параллельное исполнение транзакций выполняется в системе как будто транзакции исполняются последовательно(одна за другой~~(логично)~~)
* Устойчивость — нарушение инфраструктуры не должно мешать исполнению программы

Любой набор одновременно выполняемых транзакций упорядочен(параллельное исполнение эквивалентно последовательному).

Есть Halting Problem(Проблема остановки) — не существует такой программы, которая за конечное время проверяет, остановится ли данная программа или нет.

Транзакционная память дает другую абстрактную машину(паралл. ~ послед.), нет экспоненциального роста времени. Сложность эквивалентна числу путей в графе управления. Последовательную программу можно проверить, написав исчерпывающий набор тестов(пройти по всем путям в графе управления).

Способы исполнения транзакций:

Пессимистичный — сначала вычисляется предикат, потом выполняется транзакция(транзакция гарантировано завершится)

Оптимистичный — выполняется транзакция, потом в случае чего откатывается ~~(всегда так делаю)~~

*Транзакции VS Блокировки:*

Накладные расходы для выполнения транзакций выше чем на блокировки.

Могут возникать leavelocks. Leavelock — система не в тупике, но и ничего полезного не делает.

Но по сравнению с deadlock и одновременным доступом к памяти leavelock случается гораздо реже.

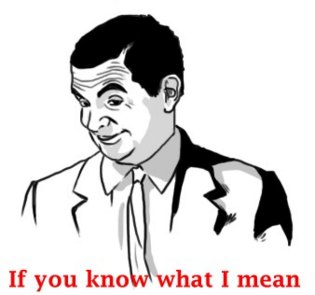
*Модель актеров:*

Транзакции — некоторый атомарный сервис, предоставляющий «актеров».

Модель актеров — некоторые параллельно существующие объекты обмениваются с помощью элементарных сообщений.

Модель актеров плохо масштабируема.

Вложенные транзакции — внутрь актёров суем других актёров(if you know what i mean). хех



***Реализации:***

*На софте:*

Read/Write барьеры

Отмена транзакции контролируется программно

Программисты ожидают последовательную согласованность

Способы реализации:

* Автоматические транзакции для всех доступов к памяти(Много накладных расходов)
* Явная отметка транзакционных переменных (дополнительная работа, склонность к ошибкам)

*На железе:*

Расширение к протоколу когерентности кэшей MESI

Все доступы к памяти транзакционные

Иногда случается отмена в свете отсутствия ресурсов

Поддерживается в продуктах:

* Azul Systems Vega 2
* Intel Haswell
* IBM Power8
* IBM zEC12
* IBM BlueGene/Q