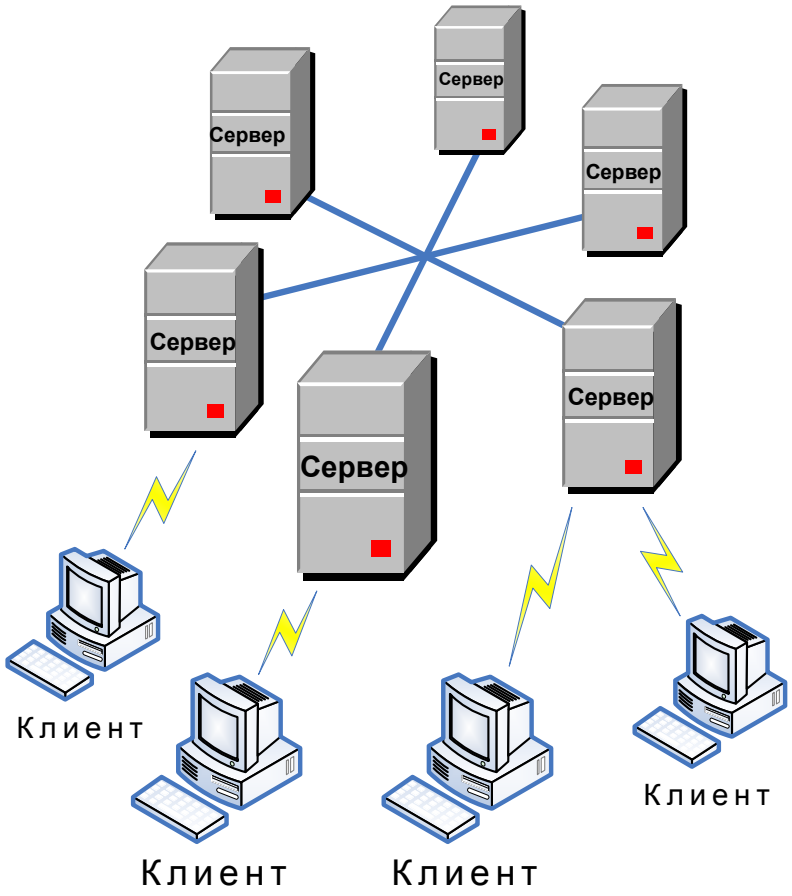
Об организации кэша распределённой графовой базы данных

А.А. Демидов, alex@dem.botik.ru

Институт программных систем им. А.К. Айламазяна РАН

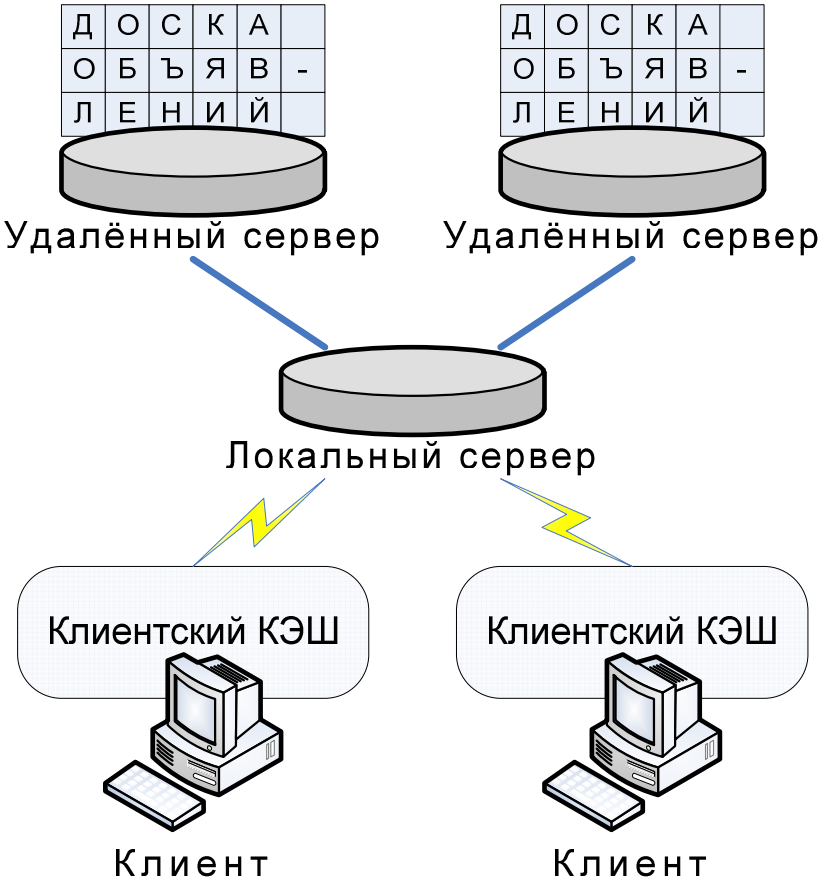
RCDL 2011

Наша главная задача – снизить синхронизационные издержки



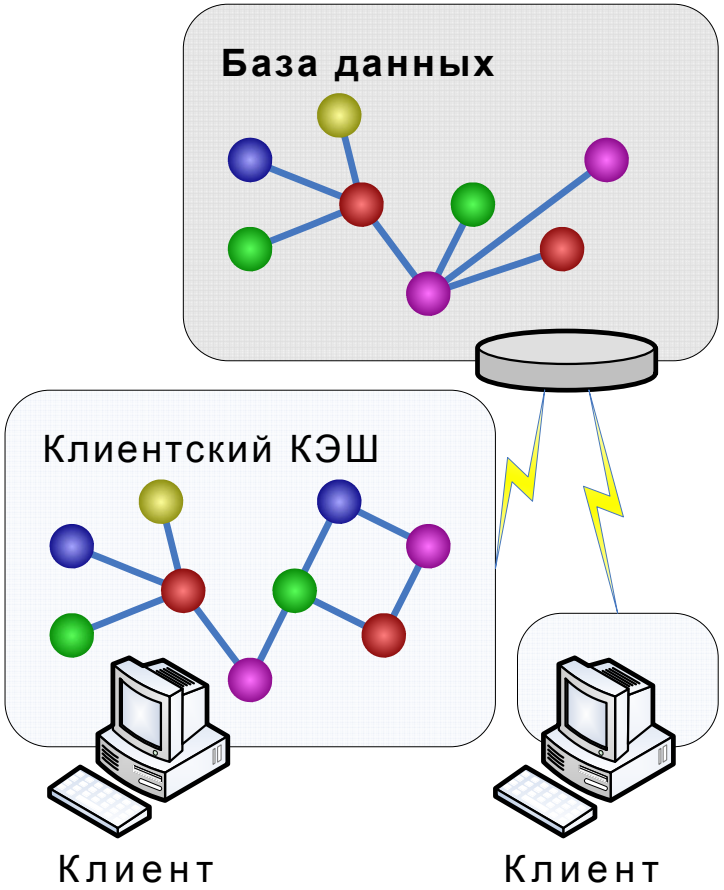
- Распределённая база данных – это сеть одноранговых серверов
- Данные разделяются на части и распределяются по серверам системы
- Клиент подключается к ближайшему серверу и через него имеет доступ ко всем данным в целом
- Сервер подключения направляет запросы удалённым серверам системы

Организация взаимодействия в распределённой среде



- Клиент обращается к ближайшему серверу через свой кэш
- Сервер разделяет запрос по узлам, а затем объединяет результаты
- Любой объект данных имеет постоянный узел приписки с доской объявлений, которая используется для синхронизации изменений
- Возможно создание копий объекта на промежуточных узлах с целью кэширования

Нарушение структуры связей при конкурирующем доступе

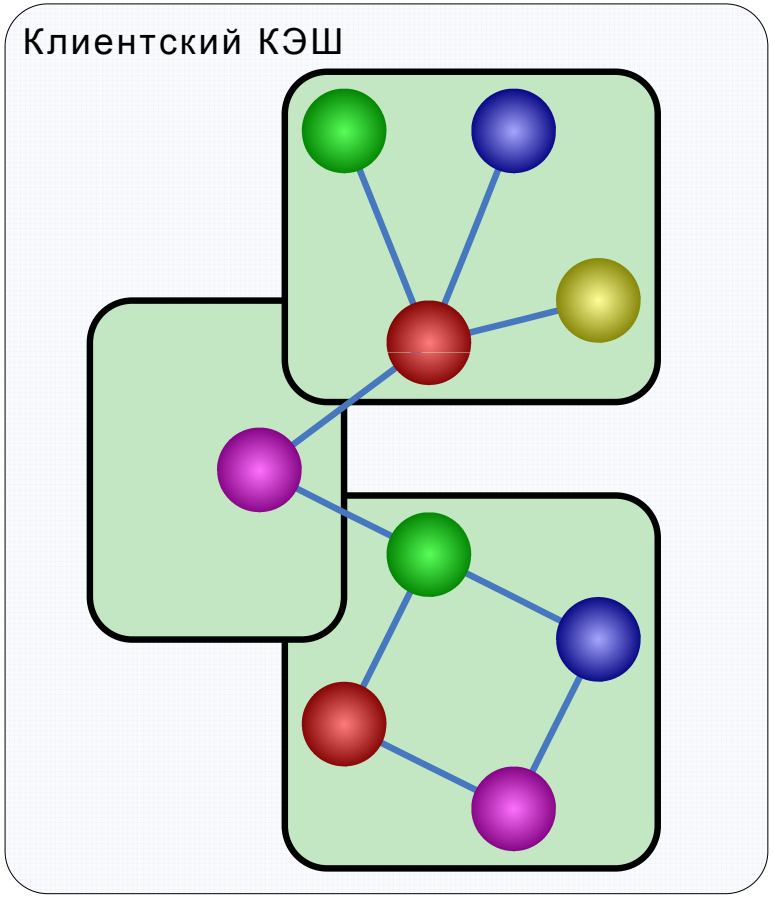


- Во время работы с кэшем те же объекты могут изменяться в БД конкурирующими транзакциями
- Доски объявлений разрешают конфликты запись/запись, для чтение/запись - неэффективны

Варианты решения:

1. Использование среза в длинных транзакциях уровня Snapshot
2. Контроль по формальным правилам (Amazon Dynamo)
3. Контроль неизменности при подтверждении (STM - Хаскель)
4. Контроль по непрерывности - необходимо задание *топологии*

Недостатки методов обеспечения непротиворечивости данных



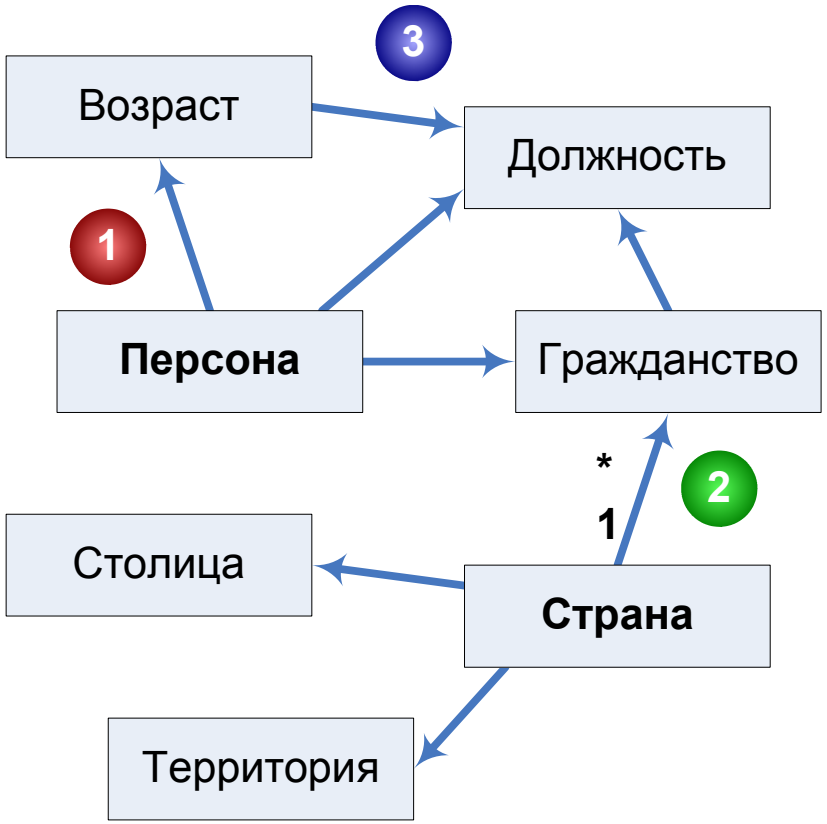
Недостатки:

1. Длинные транзакции Snapshot – потеря актуальности кэша по завершении транзакции
2. Набор формальных признаков – узкая специализация, сложность изменения системы
3. Транзакционная память – невозможно учесть влияние добавляемых записей
4. Непрерывность – нужен явный граф структуры данных

Важно!

- Подходы 2-4 позволяют кэшу использовать данные различной степени актуальности

Отношение обусловленности – обобщение функциональной зависимости



Если при изменении одного элемента данных ***a*** значение другого элемента ***b*** теряет смысл в рамках принятой модели данных, то имеется направленная зависимость:

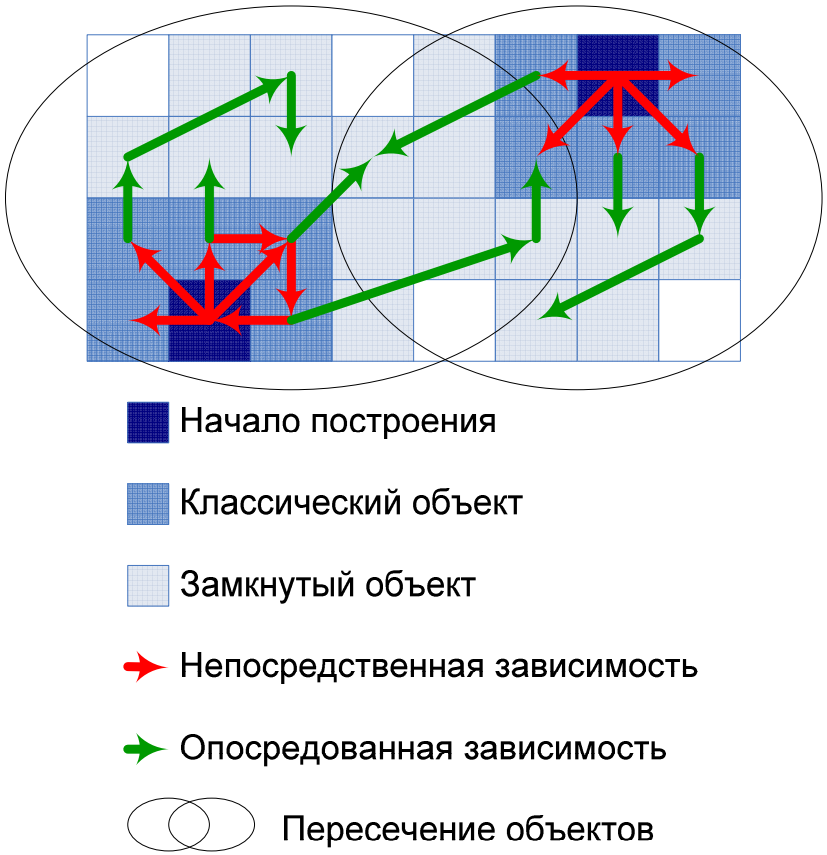
$$a \rightarrow b$$

Множество всех пар зависимых элементов базы данных назовём *непосредственным отношением обусловленности элементов*.

Классификация связей:

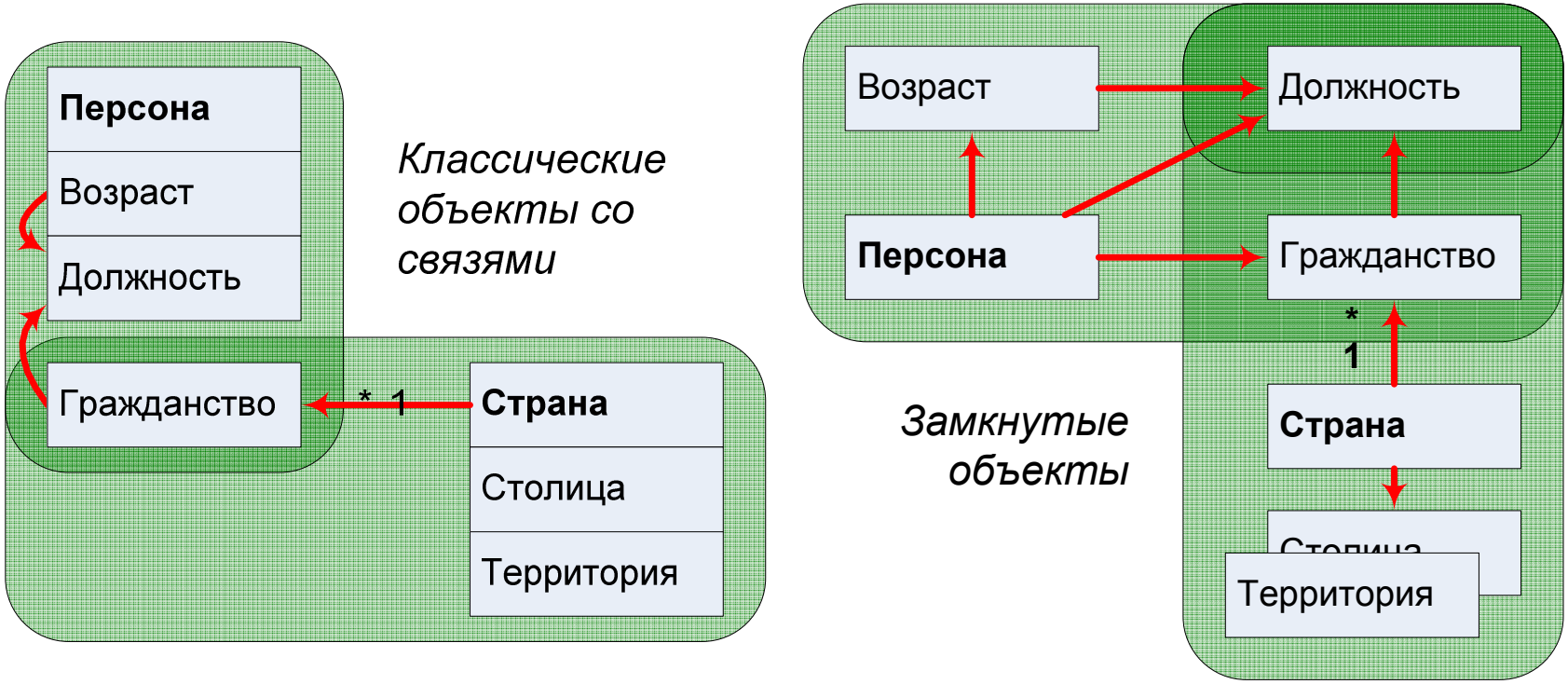
1. Зависимость от ключа
2. Внешняя зависимость
3. Внутренняя зависимость

Задание топологии через отношение обусловленности

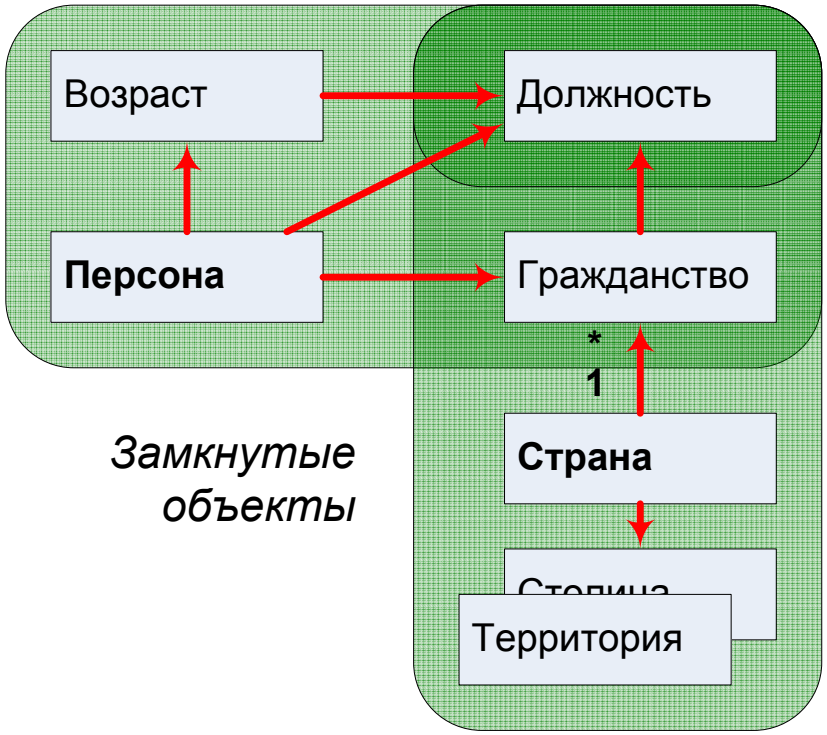


- Отношение обусловленности задаёт топологию зависимостей в пространстве данных всех серверов
- Непосредственные зависимости формируют классические объекты со связями
- Учёт опосредованных зависимостей приводит к независимым друг от друга объектам-замыканиям

Два вида объектов - классические и замкнутые



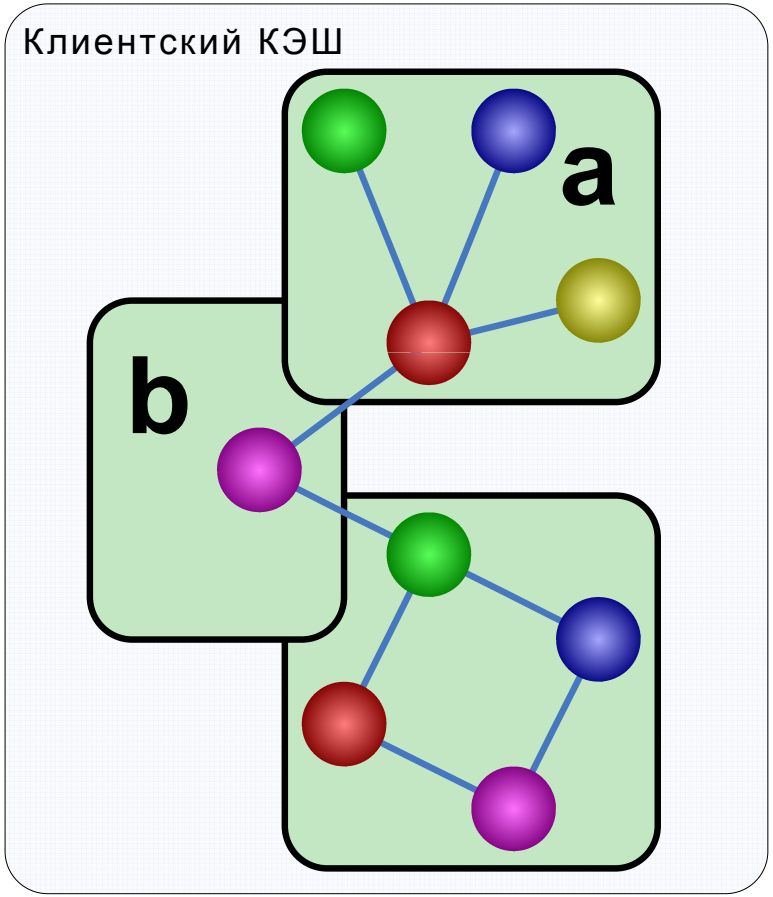
Замкнутые объекты – основа ослабления синхронизации



Замкнутые объекты

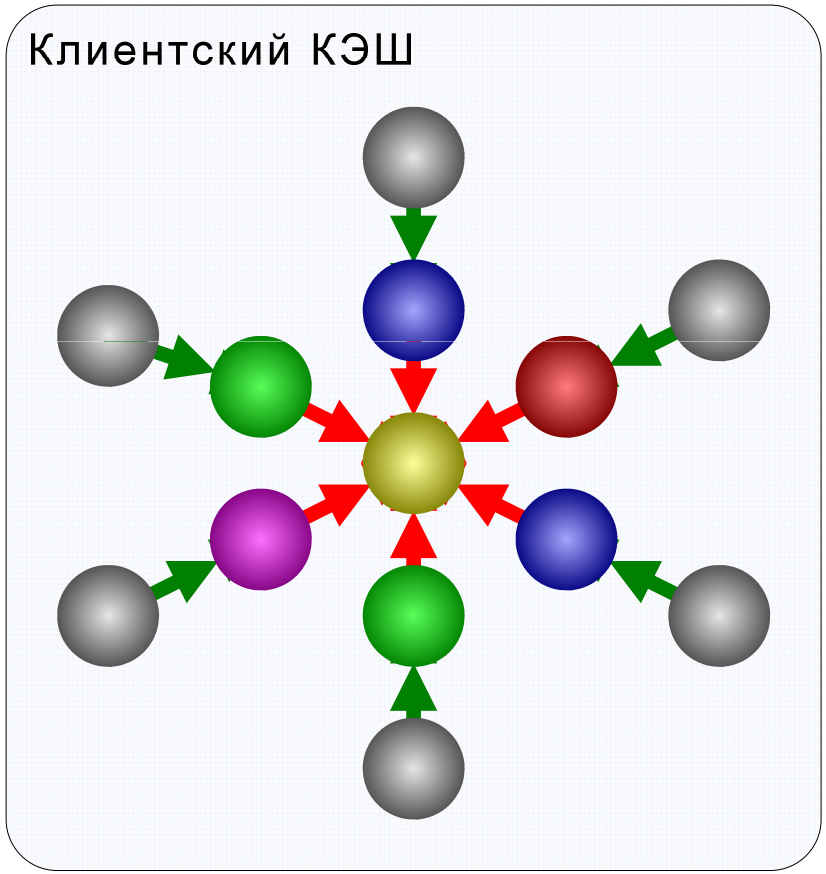
- Атомарность операций с замкнутым объектом **гарантирует отсутствие противоречий**
- Концепция замкнутых объектов **бесполезна в системах реального времени**: не должно быть разницы, какой из узлов обновил БД первым – задачи моделирования
- Замкнутые **объекты имеют свойство разрастаться** при усложнении связей, необходимо применять классические + вводить синхронизацию

Работа кэша клиента с данными различной степени актуальности



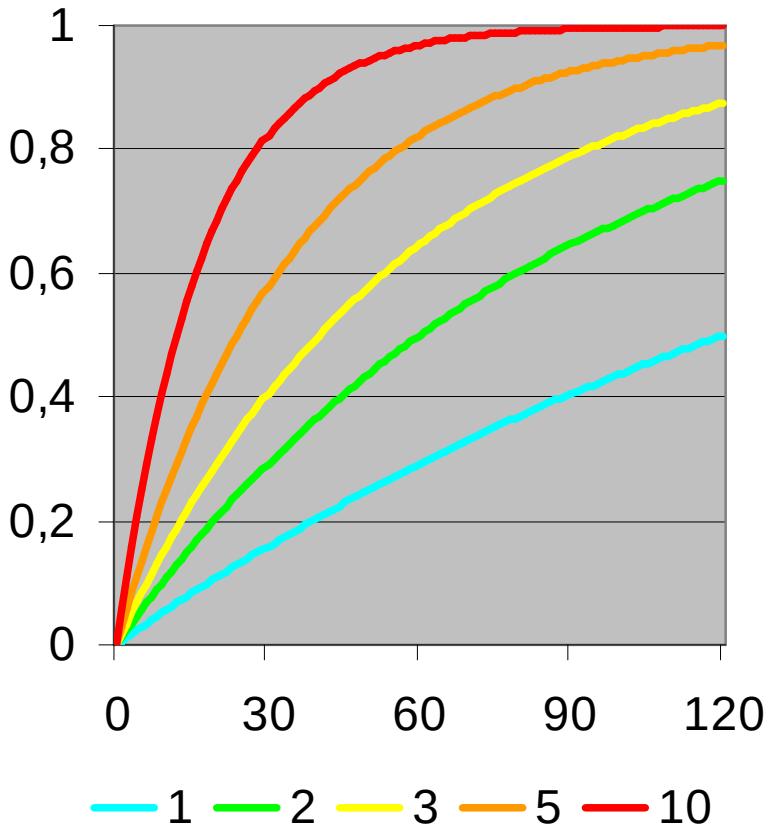
- Кэш не очищается – можно использовать данные, загруженные предыдущими транзакциями
- Каждый объект имеет две метки: время загрузки в кэш t_{TR} и время модификации на сервере t_{CH}
- Если при загрузке нового объекта ***a*** выясняется, что $a \rightarrow b$ и $t_{CH}(a) \leq t_{TR}(b)$, то старый объект ***b*** перегружается, либо происходит откат

Software Transactional Memory++ запись незамкнутых объектов



- При изменении объекта ***a*** достаточно установить метку версии на сам ***a*** и все непосредственно влияющие на него объекты: ***x*→*a***
- При изменении метки сервер контролирует совпадение текущего значения и значения, видимого транзакции

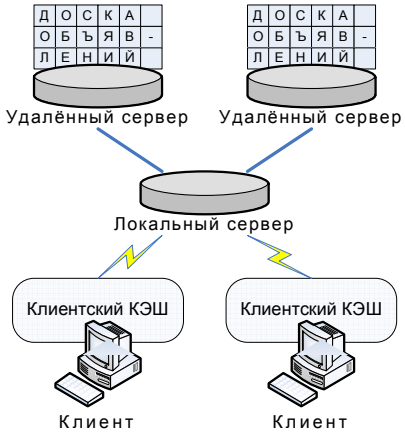
Оценка эффективности кэша ($T_{1/2}$ = 120, $K_{сложности}$ = 1..10)

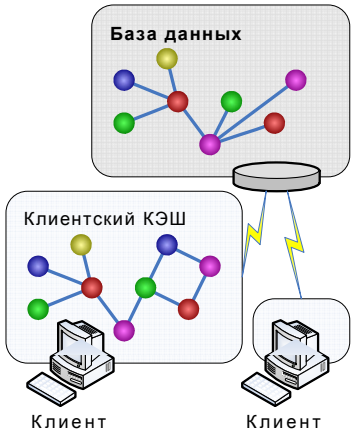


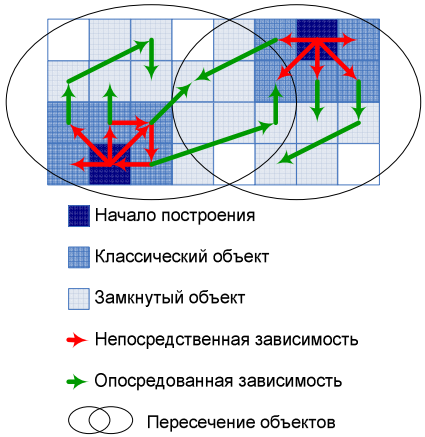
Пусть общее число объектов БД равно ***N***, в транзакции в среднем участвует ***K*** объектов, а в каждый период времени ***T*** происходит изменение ***N/2*** объектов. Вероятность встретить изменённую область:

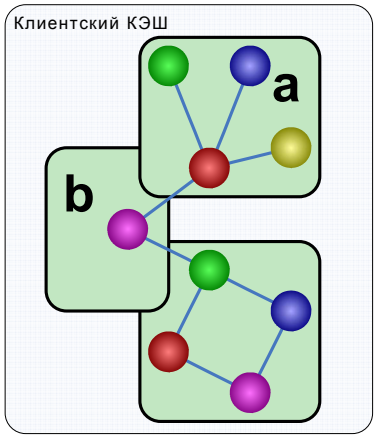
$$P(t) \sim 1 - 2^{-tK/T}$$

Вопросы?


Удалённый сервер
Удалённый сервер
Локальный сервер
Клиентский КЭШ
Клиентский КЭШ
Клиент
Клиент


База данных
Клиентский КЭШ
Клиент
Клиент


Начало построения
Классический объект
Замкнутый объект
Непосредственная зависимость
Опосредованная зависимость
Пересечение объектов


Клиентский КЭШ
a
b

Спасибо за внимание