Обработка и исполнение запросов в СУБД (Лекция 4)

Классические системы: введение в распределенные СУБД

Георгий Чернышев

Академический Университет

chernishev@gmail.com

25 сентября 2017 г.

Определение по [Kian-Lee Tan 2, 2009]

Определение

Распределенная база данных (distributed database) — коллекция баз данных, физически распределенных между узлами компьютерной сети.

Определение

Распределенная СУБД (РСУБД) — программная система, осуществляющая управление распределенной базой данных таким образом, что аспекты распределенности невидимы для пользователя.

Причины появления и популярности РСУБД

- Струтура РСУБД естественным образом ложится на структуру подразделений компании, каждое из которых имеет собственные данные.
- Уменьшаются издержки, связанные с передачей данных по сети.
- Появляется существенно больше возможностей использования параллелизма, как *внутризапросного (intrquery)*, так и *межзапросного (interquery)*.
- Увеличивается надежность — в случае выхода из строя одного из узлов сети, другие могут взять его работу на себя.
- Масштабирование существенно проще и дешевле.

Краткая историческая справка

Появились в конце 70х, вместе с shared-nothing системами.

Первые системы:

- Исследовательские:
  - SDD-1 (Computer Corporation of America)
  - Distributed INGRES (University of California at Berkeley)
  - R*STAR (IBM Research)
- Коммерческие:
  - INGRES/Star (1987)
  - Oracle (1987)

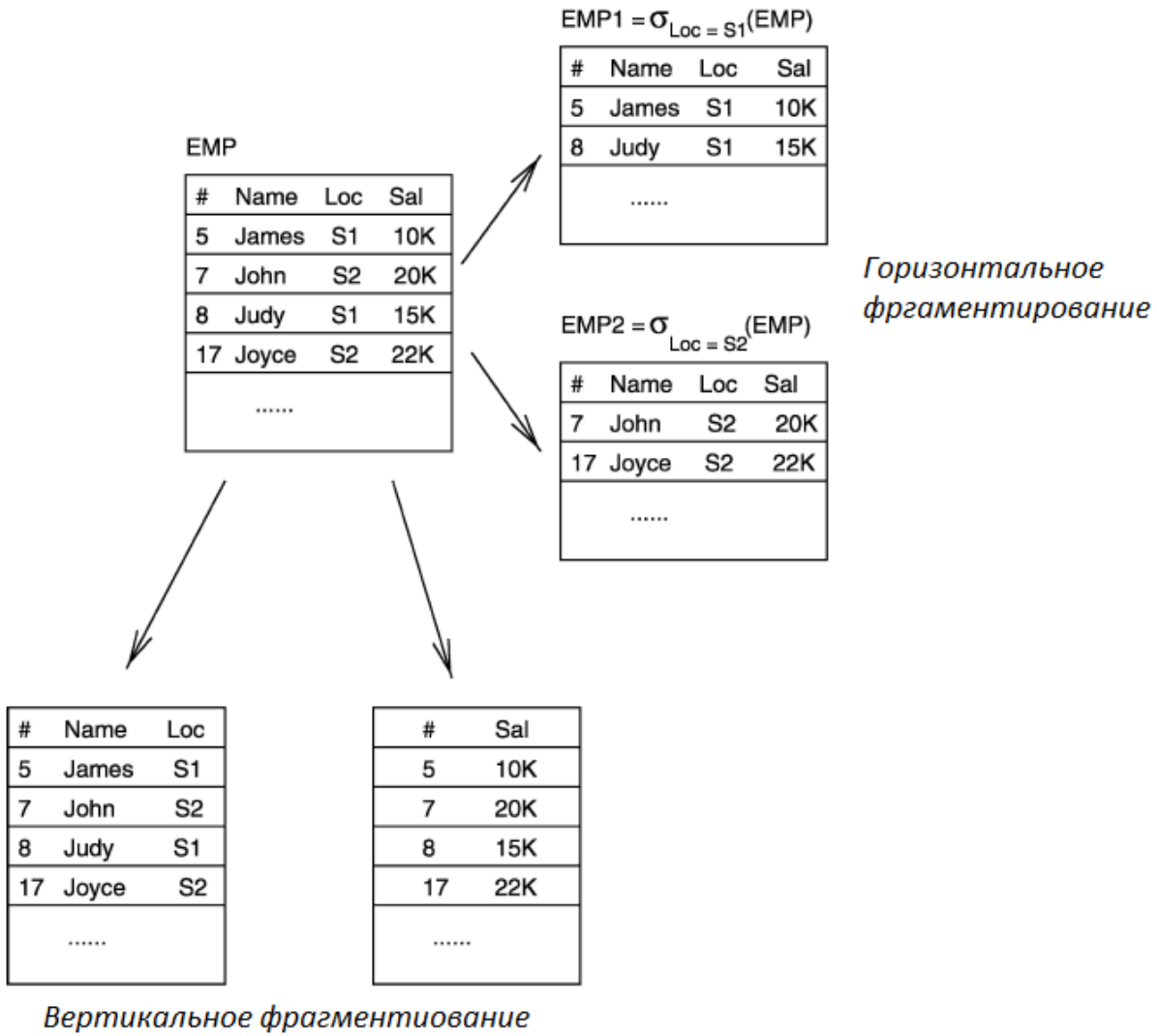
Распределение данных

Проблема распределения данных состоит из двух:

- Фрагменирование данных (data fragmentation)
  - каким образом следует разделить таблицы на фрагменты?
  - Два типа:
    - Горизонтальное, “подклеиваем” снизу с помощью UNION
    - Вертикальное, “подклеиваем” сбоку с помощью JOIN
- Аллокация данных (data allocation)
  - на каких узлах сети следует расположить фрагменты?
  - *Репликация (replication)* — распределение физических копий (реплик) логических фрагментов по узлам сети и поддержка их в согласованном состоянии

Доказано, что задача поиска оптимальной схемы распределения NP-трудная.

Пример фрагментирования данных[1]



#	Name	Loc	Sal
5	James	S1	10K
7	John	S2	20K
8	Judy	S1	15K
17	Joyce	S2	22K
.....			

#	Name	Loc
5	James	S1
7	John	S2
8	Judy	S1
17	Joyce	S2
.....		

#	Sal
5	10K
7	20K
8	15K
17	22K
.....	

[1] Изображение взято из [Kian-Lee Tan 1, 2009]

Использование ресурсов клиента

	Replication	*Caching*
target	server	client or middle-tier
granularity	coarse	fine
storage device	typically disk	typically main memory
impact on catalog	yes	no
update protocol	propagation	invalidation
remove copy	explicit	implicit
mechanism	separate fetch	fault in and keep copy after use

Fig. 16. Differences between replication and caching.

Прозрачность

Определение

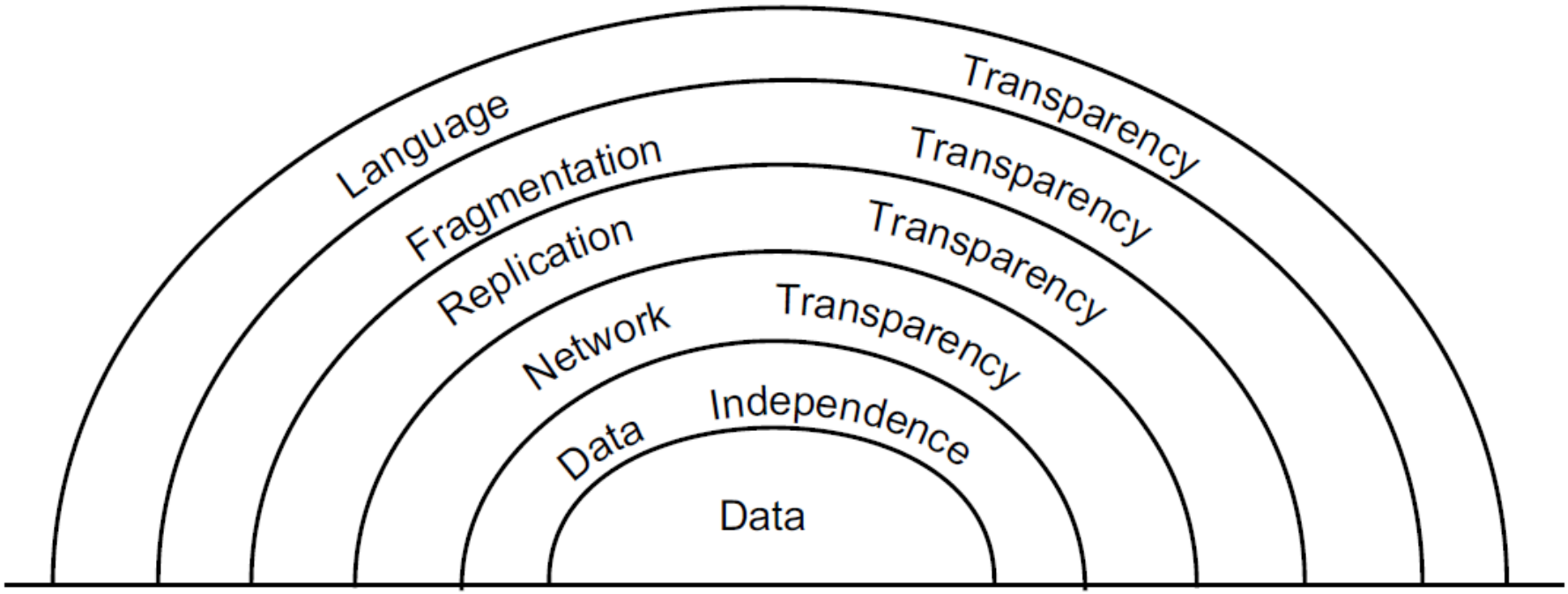
Прозрачность (transparency) – сокрытие деталей реализации от пользователя.

Наиболее общий смысл при работе с РСУБД — пользователь работает с РСУБД так же, как если бы она была централизованной.

При написании СУБД важно соблюдать баланс:

- С одной стороны, чем больше прозрачности, тем проще работать с системой.
- С другой стороны, необходимо сохранить возможности более тонкой настройки, которая может быть сделана, исходя из бизнес-задач системы.

Уровни прозрачности[2]



В [Kian-Lee Tan 2, 2009] выделяется еще и транспарентность выполнения транзакций.

[2] Изображение взято из [Özsu and Valduriez, 2011]

Клиент-серверные системы по [Kossmann, 2000]

Клиент — узел сети, отправляющий запросы на исполнение на сервер. Сервер — узел сети, выполняющий запросы полученные от клиента и отсылающий клиенту результаты.

Типы клиент-серверных СУБД:

- Peer-to-peer
  - Все узлы сети могут выступать как в роли клиента, так и в роли сервера
  - Наиболее общая схема
- Строгие клиент-серверные
  - Роль каждого узла сети строго фиксирована
  - Клиенты не могут взаимодействовать между собой
- Многоуровневые
  - Узлы сети организованы иерархически
  - Каждый узел сети выступает в роли клиента для нижних уровней и в роли сервера для верхних

Автономность

Автономность показывает, насколько могут отдельные СУБД работать по отдельности.
В случае ее наличия:

- Распределенная система не влияет на локальные операции отдельных СУБД
- Способы обработки и оптимизации локальных запросов изолированы при выполнении глобальных запросов
- В случае добавления или выбывания отдельных СУБД консистентность работы РСУБД не должна страдать

Можно выделить:

- Автономность дизайна БД
- Автономность коммуникации
- Автономность выполнения

Гомогенность vs гетерогенность

Гомогенная распределенная база данных — такая БД, в которой все узлы находятся под управлением одной и той же распределенной СУБД, автономность отсутствует.

Гетерогенная распределенная база данных — БД, данные которой распределены между несколькими автономными СУБД, часто принципиально отличающимися.

Пример гетерогенной БД — мультибаза

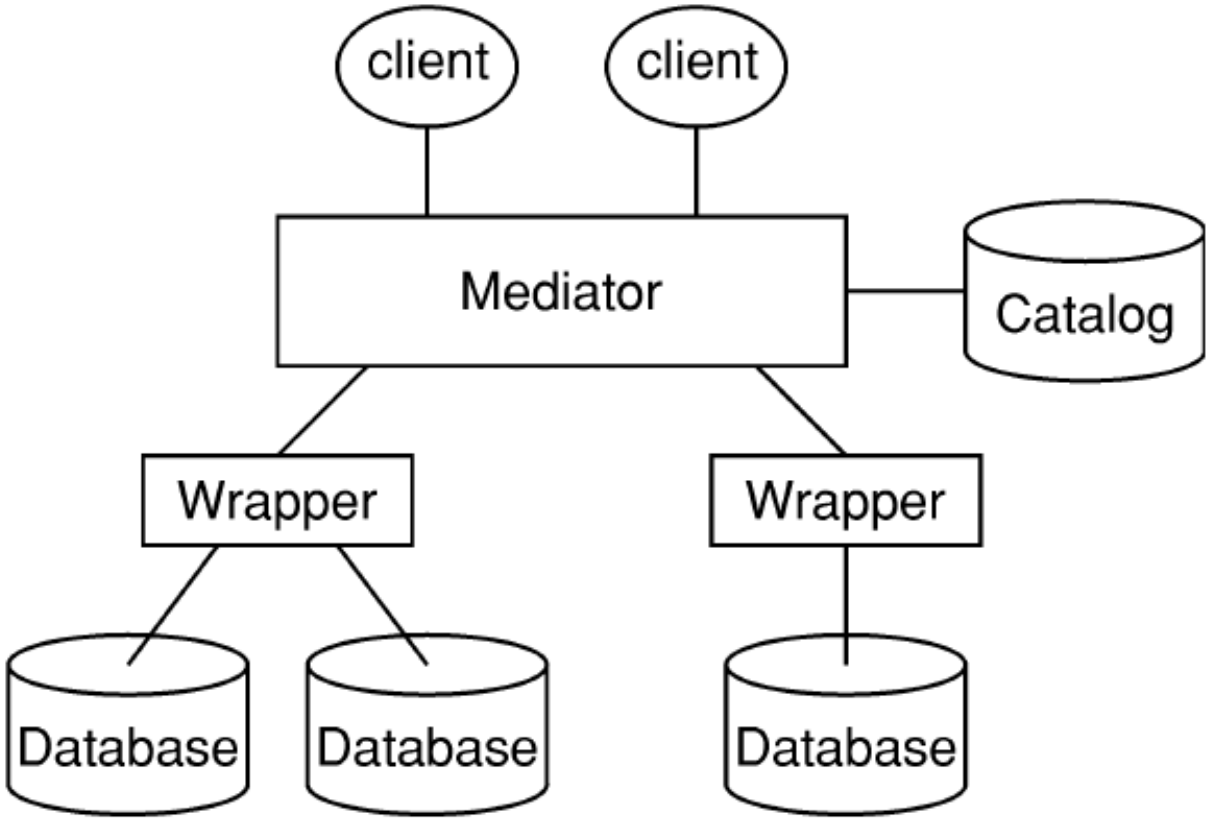


Fig. 12. Wrapper architecture of heterogeneous databases.

3

[3] Изображение взято из [Kossmann, 2000]

Функции компонентов мультибазы

Функции медиатора:

- синтаксический разбор запроса
- оптимизация запроса
- поддержка глобального каталога

Функции обертки:

- переписывание запроса, полученного от медиатора, к API соответствующей БД
- переписывание результатов запроса к форме, понятной медиатору
- может обеспечивать кэширование и поблочную обработку, участвует в оптимизации

Основные исследовательские задачи

- Как хранить данные?
- Как при обновлениях синхронизировать копии?
- Как осуществлять координацию и коммуникации между узлами при исполнении запроса?

РСУБД, сложности и частичные ответы I

Основные исследовательские задачи связанные с РСУБД:

- Как хранить данные?
  - Принцип локальности данных $\longrightarrow$ увеличение количества реплик $\longrightarrow$ страдают апдейты;
  - Ручной труд администратора по фрагментированию;
  - Аллокация: статическая и динамическая, экономические модели;
- Как при обновлениях синхронизировать копии?
  - Протоколы кворума;
  - Коммерческие используют ROWAA;
  - Семантика одной копии;
  - Распространения обновлений: lazy и eager методы;
- Как осуществлять координацию и коммуникации между узлами при исполнении запроса?
  - Разрезать план запроса;
  - Стоимость коммуникации высока;

РСУБД, сложности и частичные ответы II

- Нужна глобальная изоляция $\longrightarrow$ нужна глобальная сериализуемость, локальной не хватит $\longrightarrow$ d2PL, мультиверсионные на временных метках;
- Замки: primary site vs primary узел;
- Координация очень дорога;

- Как обеспечивать надежность?
  - Для коммитов d2PC, 3PC;

Ссылки I


Distributed Database Design. Kian-Lee Tan. Encyclopedia of Database Systems. Ling Liu and M. Tamer Özsu (eds), p. 890–894. Springer US, 2009. https://doi.org/10.1007/978-0-387-39940-9_703


Distributed Database Systems. Kian-Lee Tan. Encyclopedia of Database Systems. Ling Liu and M. Tamer Özsu (eds), p. 894–896. Springer US, 2009. http://dx.doi.org/10.1007/978-0-387-39940-9_701


Distributed DBMS. Sameh Elnikety. Encyclopedia of Database Systems. Ling Liu and M. Tamer Özsu (eds), p. 896–899. Springer US, 2009. http://dx.doi.org/10.1007/978-0-387-39940-9_654


Özsu M.T. and Valduriez P. Principles of Distributed Database Systems, 3rd ed. Prentice-Hall, 2011.


Donald Kossmann. 2000. The State of the Art in Distributed Query Processing. ACM Comput. Surv. 32, 4 (December 2000), 422–469. http://dx.doi.org/10.1145/371578.371598