Система управления данными для больших научных проектов

Павел Велихов
НИИСИ РАН

http://www.scidb.org

Участники проекта

- Основатели: Michael Stonebraker, David DeWitt, Jacek Becla, KT Lim
- Российская команда разработчиков: Павел Велихов, Роман Симаков, Константин Книжник, Артем Смирнов

Michael Stonebraker

- Ingres
- Postgres
- Illustra
- Cohera
- Vertica
- VoltDB

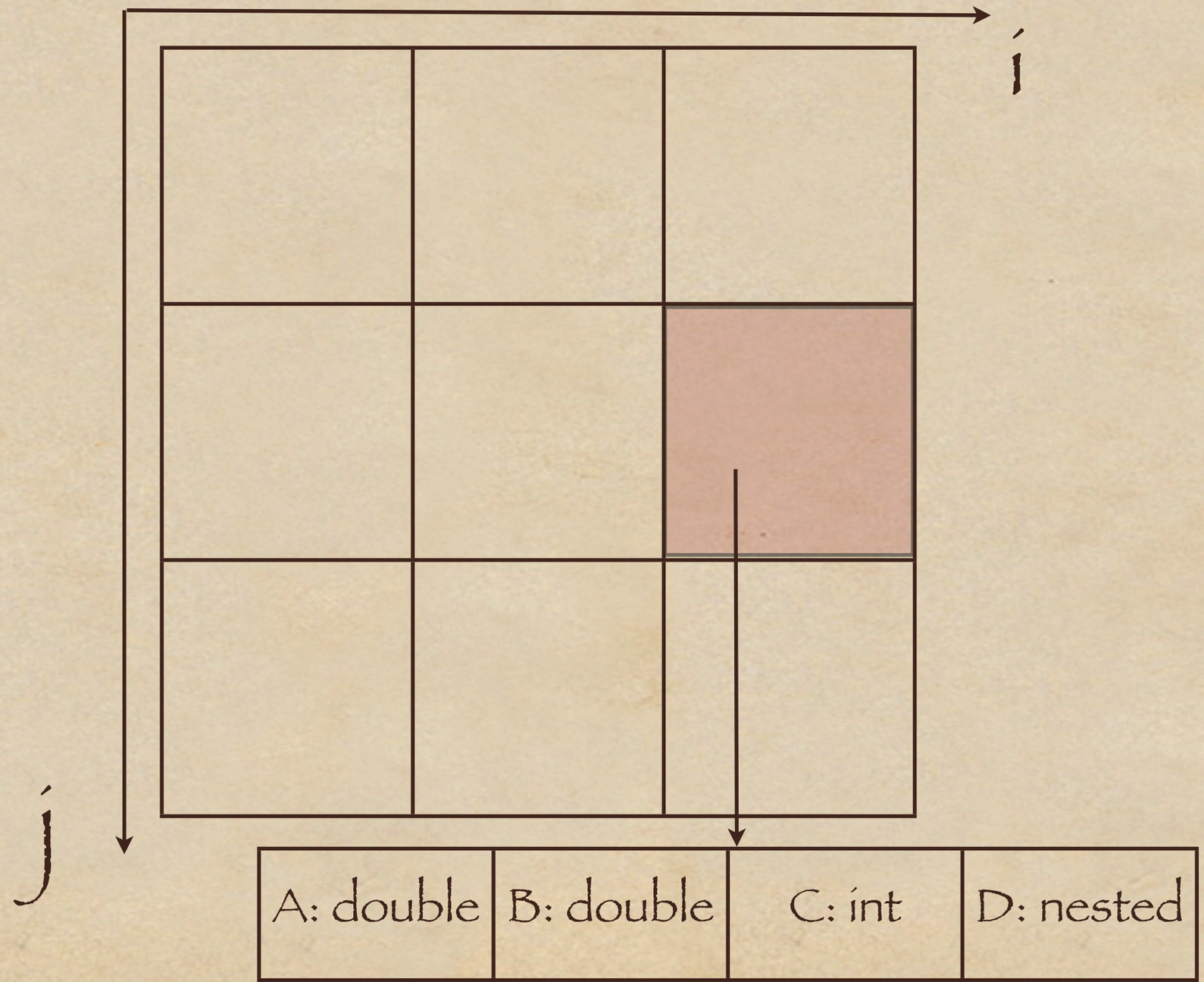
Зачем нужна еще одна СУБД

- Большинство научных проектов, где объемы данных внушительные (петабайты) почти не используют СУБД
  - Эффективность хранения
  - Производительность
  - Язык запросов
  - Открытый код для свободного обмена данными

Что требуется от SciDB

- Эффективное хранение сырых данных
- Полный цикл анализа данных: data cleaning, feature extraction, data mining, data sharing
- Версионность и provenance (происхождение) для обеспечения повторяемости результатов
- Специфические требования для научных данных (погрешность измерений, расширяемость, т.д.)

Модель данных SciDB



Хранение

- Самая распространенная модель данных в науке - многомерный массив
- Массив может быть разреженным или/и с неровными краями
- Каждый элемент массивов может содержать несколько значений
  - пример: измеренная величина и ее погрешность
- Массивы могут быть вложенными

Что тут принципиально нового?

- Массив можно моделировать реляционной таблицей:
  - create table (i,j,A,B,C,D_key)
- Также, в современных базах есть тип данных “массив”
- Но: оба подхода имеют ряд серьезных недостатков

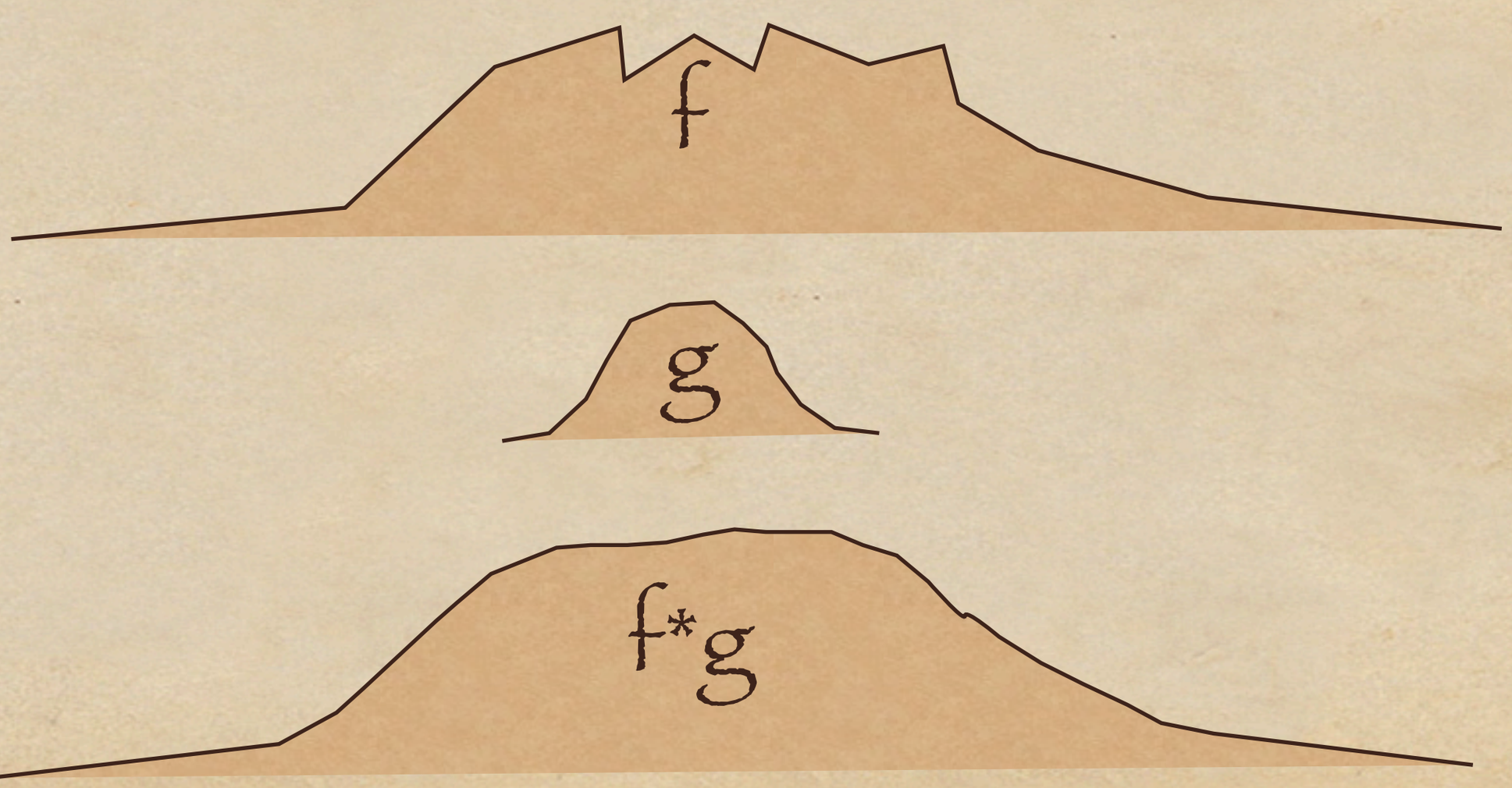
SQL: Массив-таблица

- Хранение индексов массива.
- Доступ к подмножествам массива?
- Вычисления на окрестностях точки?

  Используя SQL?

Типичные операции с массивами

- Свертка: $f*g(x)$



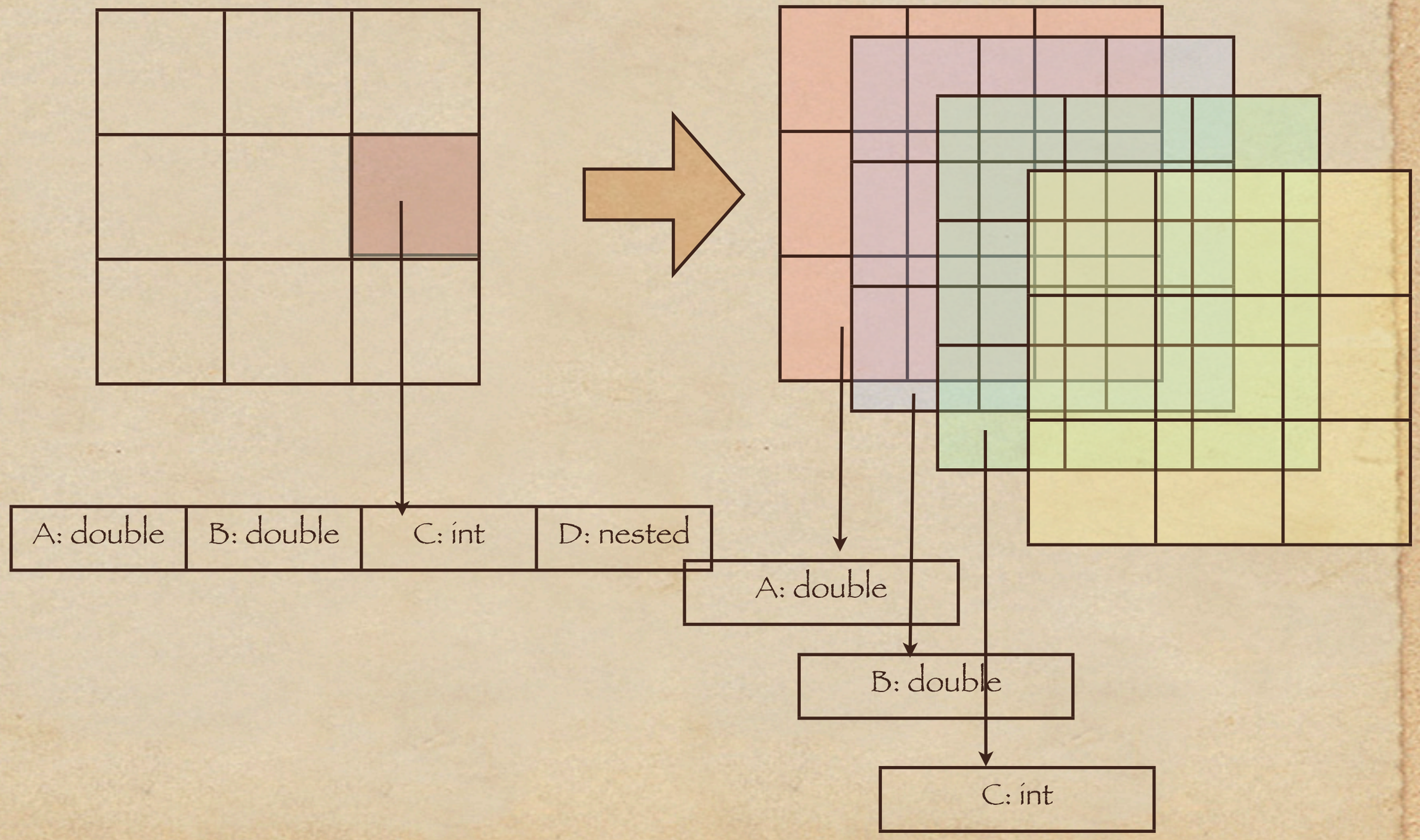
SQL: Массив - тип данных

- Пропали атрибуты!
- Язык запросов (SQL) с массивами не работает, работать можно только с композицией функций поверх массива!

Хранение данных в SciDB

- Модель данных ~ массив
- Вертикальное хранение с компрессией
- Горизонтальное разделение данных с перекрытием (для распараллеливания и отказоустойчивости)

Вертикальное хранение



Вертикальное хранение

- Легко сжимать данные (все атрибуты одного типа)
- Можно читать с диска только то, что нужно для запроса
- Отлично подходит для аналитики: Vertica, Big Table
- Но не для OLTP!

Транзакционные свойства SciDB

- Данные не меняются, только добавляются!
- Таким образом мы избегаем распределенных транзакций
- Все равно в распределенной системе добиться ACID свойств практически невозможно (что делать, если система развалилась на 2 части и каждая живет своею жизнью?)

Эффективность хранилища

- Основные накладные расходы современных СУБД
  - logging
  - locking
  - latching
  - page-based buffer management

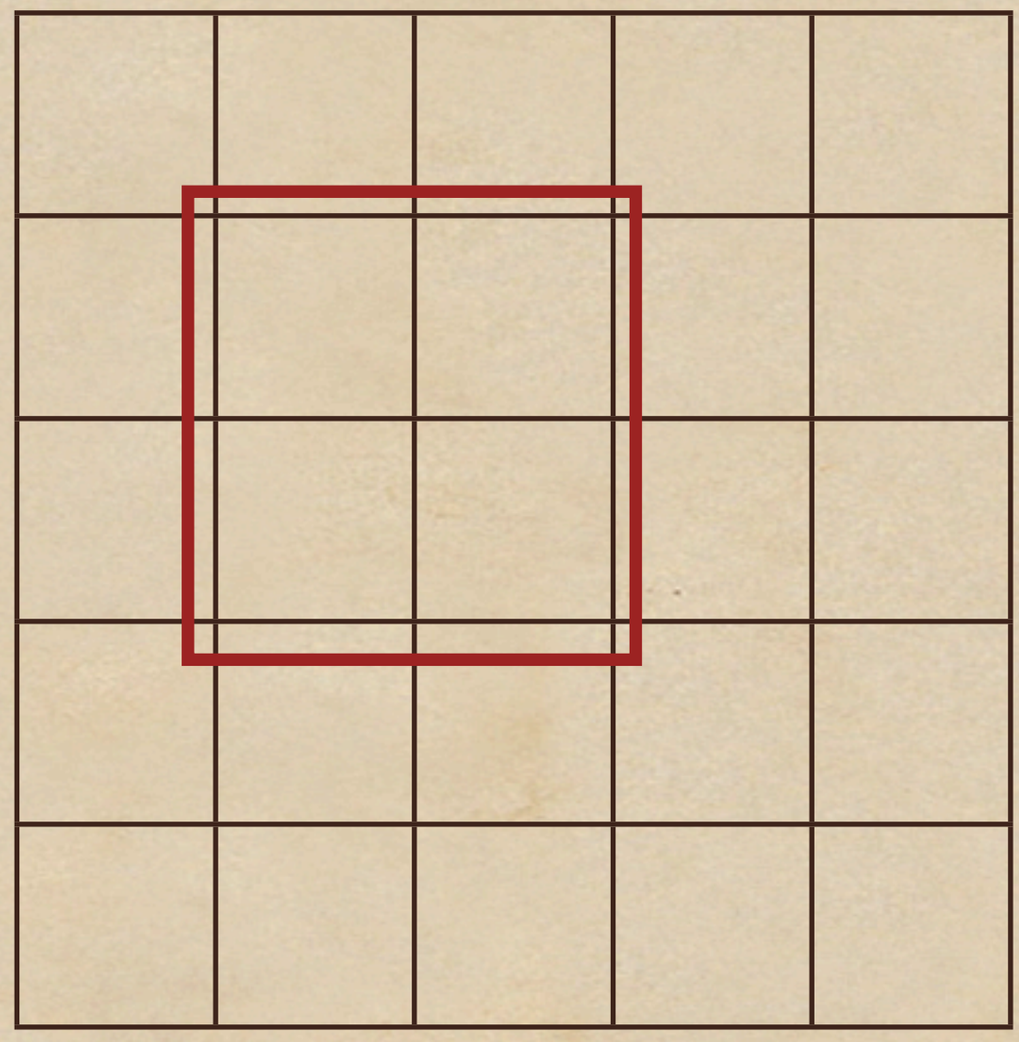
Что требуется от SciDB

- Эффективное хранение сырых данных
- Полный цикл анализа данных: data cleaning, feature extraction, data mining, data sharing
- Версионность и provenance для обеспечения повторяемости результатов
- Специфические требования для научных данных (uncertainty, т.д.)

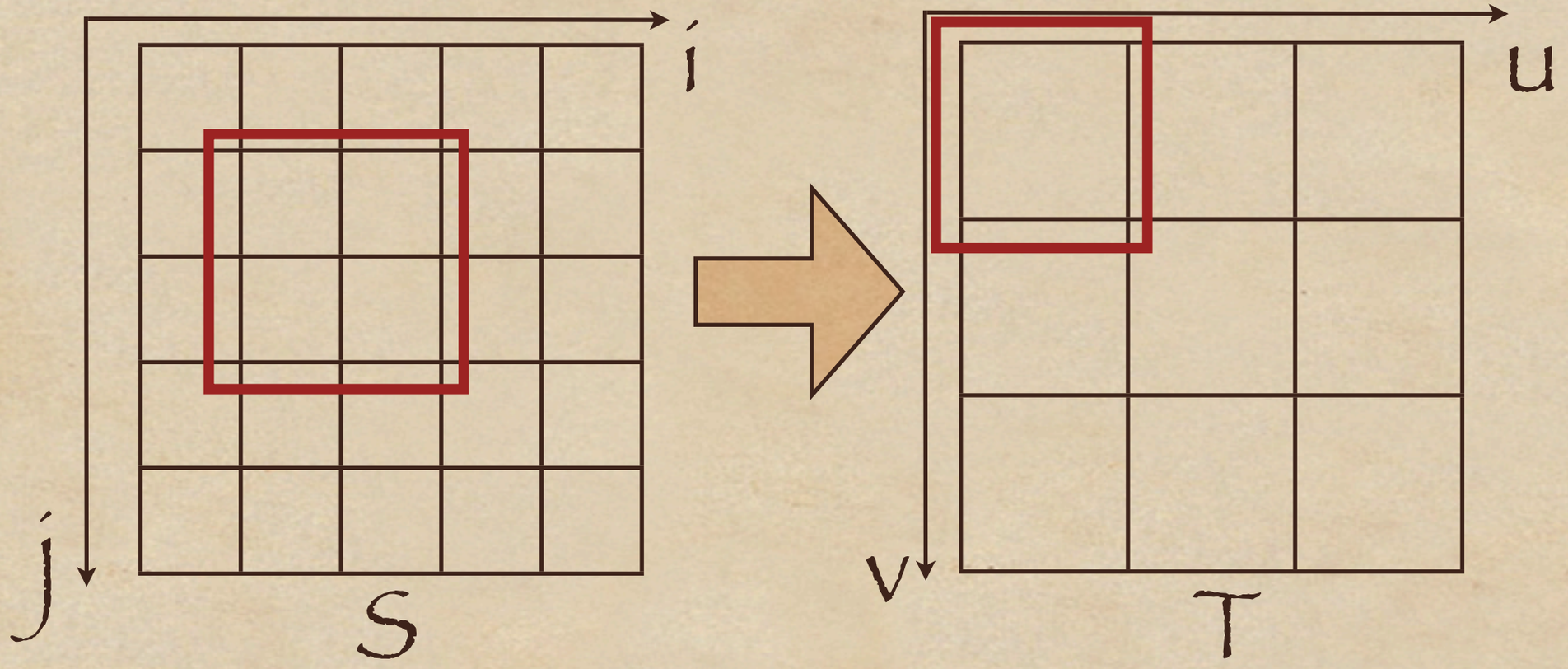
Операции над массивами

- Алгебра массивов (вместо реляционной)
- Многие операторы похожи на реляционные - filter, join, group-by
- Специфические операторы для научных вычислений: subsample, regrid

SciDB Algebra: Subsample



SciDB Algebra: Regrid



assign: (u,v) -> { (i,j) }

aggregate: f(assign(u,v)) -> value

Пример запроса в SciDB

S: <x: float>[i,j]

T: <z: float> [k,l]

```
regrid(
  subsample(S, [0,0, 1000, 1000]) as S',
  T,
  assign: subsample(S', [k-10,j-10, k+10, j+10])
  agg: sum ( x * ae^-(i-b)^2/2c^2 )
)
```

Вложенные массивы

- Как работать с вложенными массивами?
- Вложенные планы (аля OQL, XQuery)
- Оператор Apply применяет вложенный план к вложенному массиву
- Полученный результат добавляет как атрибут к массиву

Пример с вложенными массивами

S: <x: float>[i,j]
T: <z: float> [k,l]

```
apply(
  T,
  sum ( apply( subsample(S, [k-10,j-10, k+10, j+10]),
              x * ae^-(i-b)^2/2c^2 as y))
)
```

Выполнение запросов в SciDB

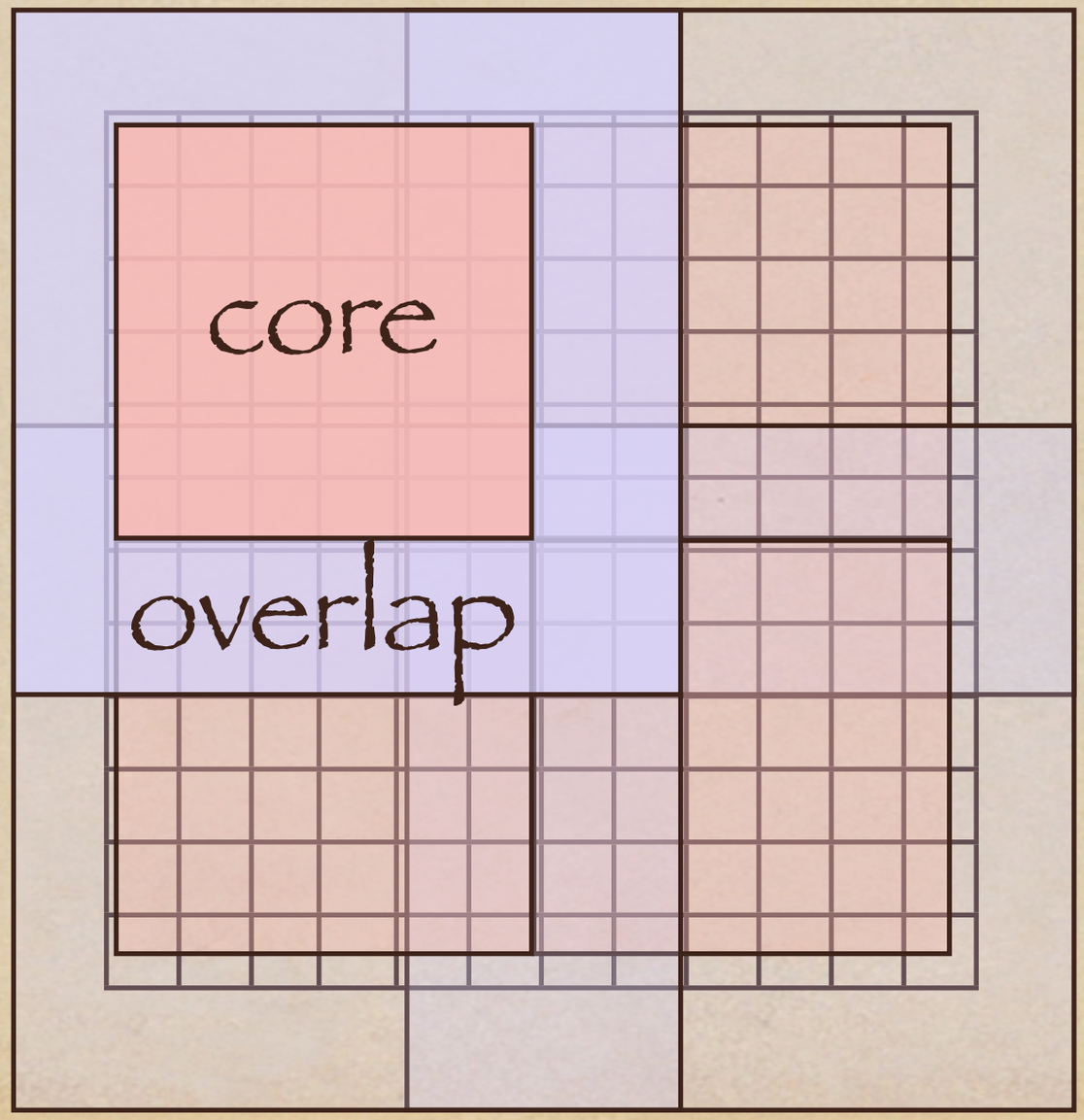
- Вертикальное хранение с компрессированными чанками
- Каждый оператор SciDB работает с чанками (аля Vertica), а не с кортежами (Oracle, PostgreSQL,....)

Пример: Subsample

- Конвейерный (pipeline) интерфейс: getChunk (attr, pos)
- Пересылаем вниз getChunk(attr, pos2)
  - pos2 = pos + offset
- Если чанк внутри “окна” оператора, отдаем вверх не распаковывая
- Распаковываем, обрезаем, передаем вверх

Параллелизм в SciDB

- Горизонтальное разбиение с перекрытием

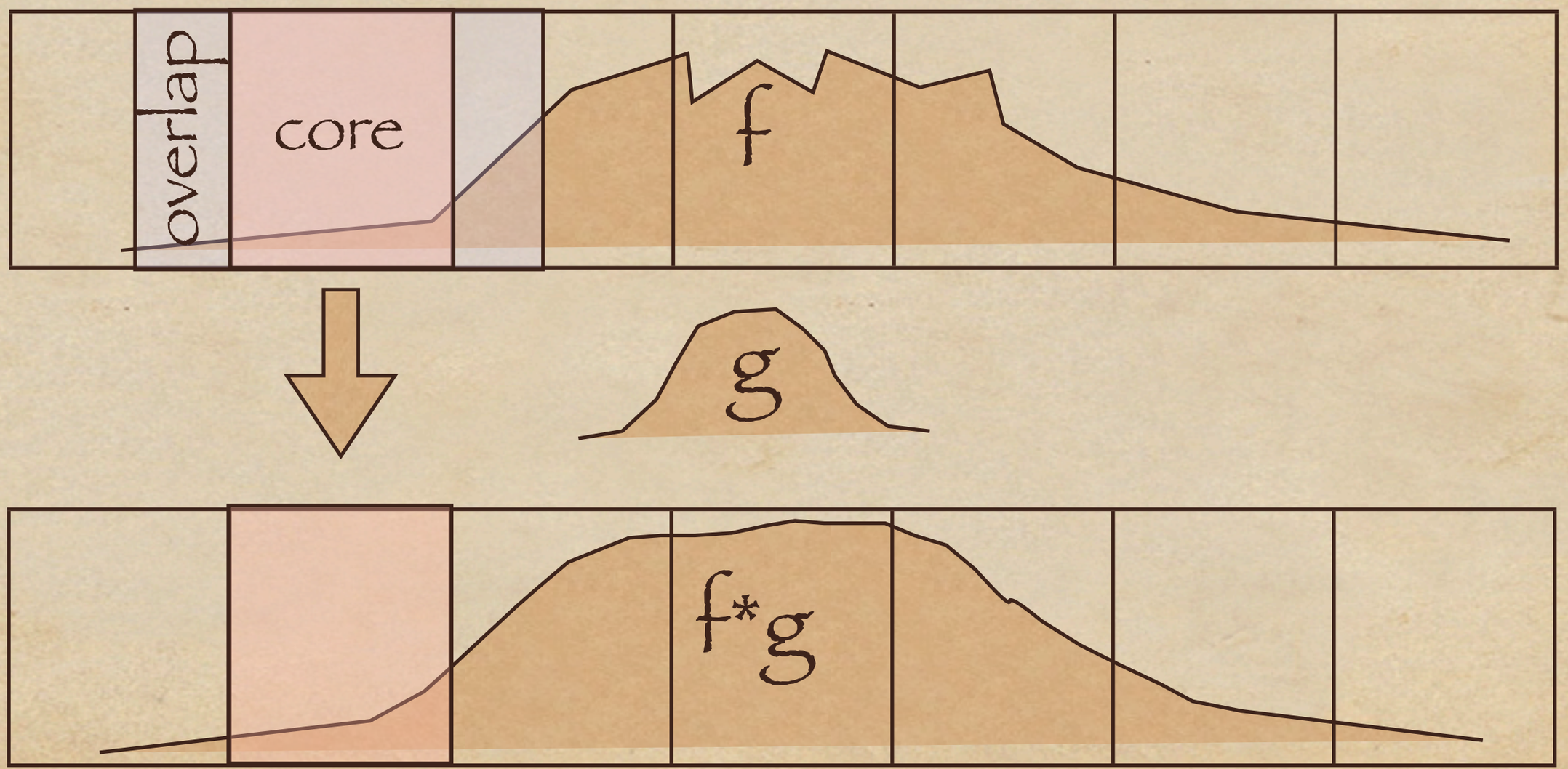


Параллелизм в SciDB

- Оптимизатор запроса вычисляет, достаточно ли первоначальное разбиение для выполнения запроса
- Если нет, то в план добавляется оператор Scatter, который “докидывает” дополнительные данные

Параллелизм в SciDB

- Свертка: f*g(x)



Что требуется от SciDB

- Эффективное хранение сырых данных
- Полный цикл анализа данных: data cleaning, feature extraction, data mining, data sharing
- Версионность и provenance для обеспечения повторяемости результатов
- Специфические требования для научных данных (uncertainty, т.д.)

Дополнительные возможности

- Версионность
  - изменения данных ≈ новая версия
  - релиз и последняя версия ≈ быстрый доступ

Дополнительные возможности

- Provenance (происхождение данных) ~ “откуда взялась эта звезда?”
  - playback
  - отладка анализа данных
  - повторяемость научных данных

Что требуется от SciDB

- Эффективное хранение сырых данных
- Полный цикл анализа данных: data cleaning, feature extraction, data mining, data sharing
- Версионность и provenance для обеспечения повторяемости результатов
- Специфические требования для научных данных (uncertainty, расширяемость, ...)

Дополнительные возможности

- Uncertainty - измерения приборов всегда неточные
- Частичная поддержка арифметики интервалов в алгебре SciDB (filter,join):
  - a - err < C < a + err
  - a ≈ C

Дополнительные требования

- Расширяемость:
  - User Defined Types: модель PostgreSQL
  - User Defined Functions: произвольные функции над массивами
  - Интерфейсы к научным пакетам ПО: R, Matlab.

Наша цель:

Сырые данные

SciDB

обработка сырых данных, анализ, верификация, получение научных результатов

Повторяемые научные результаты

Пример использования SciDB: Анализ генома

- Последовательности нуклеотидов AGCT
- много N - ошибки прибора
- контроль качества:
  - сколько ошибок, где они встречаются, есть ли корреляция с нуклеотидами, какая временная зависимость
- дополнительные данные о точности сканирования

Статус SciDB

- Проект с открытым кодом
- Разработка ведется с начала 2009
- Демо прототипа: сентябрь 2009

  VLDB 09
- Альфа версия: SciDB V1, 2010

- Спасибо за внимание
- Вопросы & Ответы