Унификация модели данных, основанной на многомерных массивах, при интеграции неоднородных информационных ресурсов

Сергей Ступников

Институт проблем информатики, Российская академия наук

План

- Модели данных, основанные на многомерных массивах (ММ-модели)
- Унификация моделей данных для интеграции неоднородных ресурсов
- Отображение ММ-модели в объектную модель данных
  - Отображение языка определения данных
  - Отображение языка манипулирования данными
- Сохранение информации и семантики операций ЯМД при отображении
- Направления дальнейшей работы

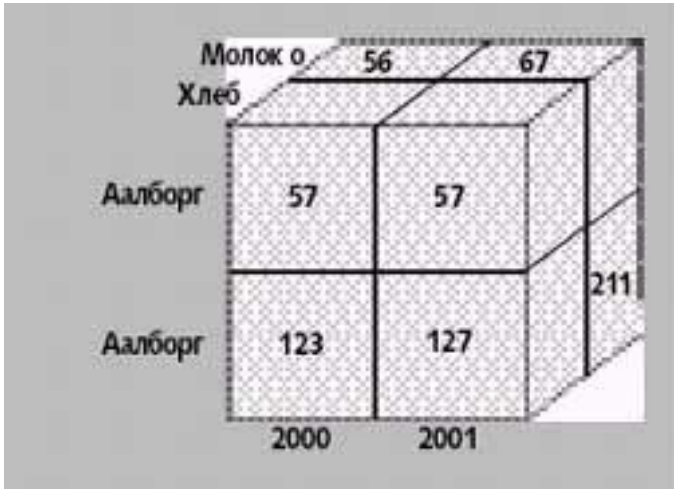
Большие данные – Big Data

- Происхождение данных: наблюдательные, экспериментальные, полученные в ходе компьютерных симуляций
- Большие данные (PB) плохо поддаются обработке и анализу в рамках хорошо известных технологий баз данных, опирающихся в основном на *реляционную модель данных*

- Развиваются различные модели данных, нацеленные на параллельную обработку и анализ данных в распределенных средах – гридах и облаках
- Модели данных, основанные на многомерных массивах (Array-based Data Models, Array Data Models)

MOLAP Data Cubes

- Факт – набор значений измерений (dimensions), которому сопоставлен набор параметров (покупка — факт, объем покупки и стоимость — параметры, тип продукта, время и место покупки — измерения)
- Запросы агрегируют значения параметров по всему диапазону измерения (общий месячный объем продаж данного продукта)
- Измерения организуются в иерархию, состоящую из нескольких уровней, каждый из которых представляет уровень детализации, требуемый для соответствующего анализа



- Реализации: Cognos Powerplay, Oracle Database OLAP Option, Microsoft Analysis Services, etc.

SciDB

- Симпозиум XLDB 2007
  - « … существующие СУБД не в состоянии манипулировать объемами данных, которые появятся в ближайшем будущем … » (LSST - Large Synoptic Survey Telescope)
  - Требования для СУБД нового поколения
    - модель данных основывается на многомерных массивах, а не на кортежах
    - модель хранения основывается на версионности, а не на обновлении значений
    - масштабируемость до сотен петабайт и высокая отказоустойчивость
    - свободно распространяемое ПО
- 2008 – запущен проект СУБД SciDB под руководством Майкла Стоунбрейкера
  - Свободно распространяется версия для ОС Ubuntu и RedHat
- Модель SciDB
  - AQL (Array Query Language)
  - AFL (Array Functional Language)

Цель работы

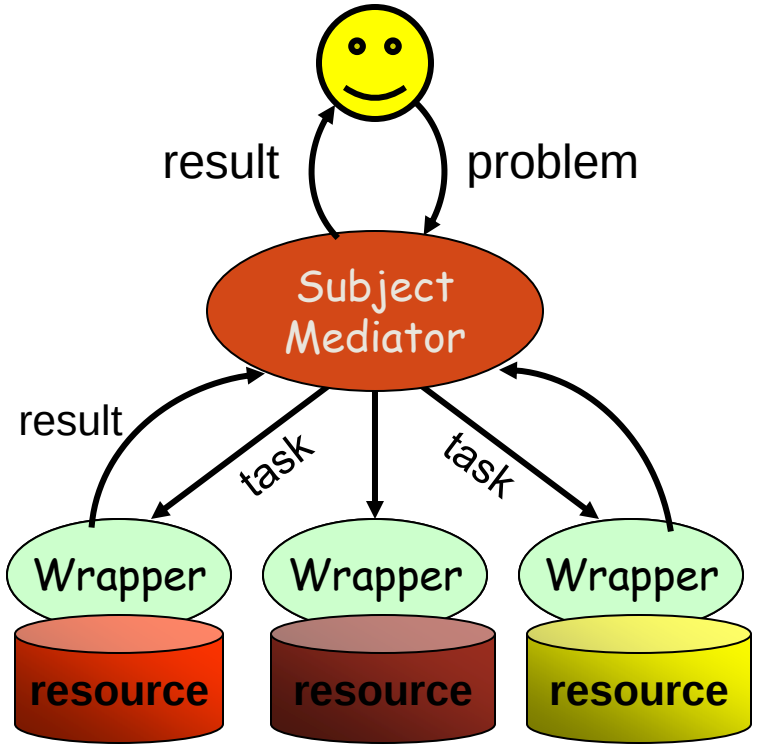
- Унификация модели данных системы SciDB (ADM – Array Data Model) для *виртуальной* или *материализованной* интеграции ресурсов при создании федеративных баз данных или хранилищ данных

- СУБД, основанные на многомерных массивах - новый вид ресурсов для интеграции вместе с привычными ресурсами – реляционными и объектными СУБД, веб-сервисами, …
- SciDB используется в проектах
  - научных - астрономия, физика высоких энергий
  - коммерческих - генетика, страхование, финансы
- Сравнительное тестирование SciDB с СУБД Postgres и статистическим ПО R показало преимущества SciDB по производительности и масштабируемости

- Критика SciDB: ADM - смесь SQL и деревьев алгебраических операций. Мнение: язык для СУБД, основанных на многомерных массивах, должен быть интегрирован с синтаксисом и семантикой SQL:2003.

Виртуальная интеграция в предметных посредниках

- Задача формулируется в терминах схемы посредника, затем
- трансформируется в набор подзадач (запросов) к ресурсам, зарегистрированным в посреднике;
- подзадачи исполняются на ресурсах, результаты возвращаются в посредник;
- результаты объединяются и представляются пользователю.



Унификация информационных моделей

- *Каноническая информационная модель* - общий язык, унифицирующий разнообразные модели ресурсов
- *Унификация* исходной модели данных - ее отображение в каноническую модель, сохраняющее информацию и семантику операций языка манипулирования данными (ЯМД)
  - унификация должна быть доказуемо правильной
  - унификация моделей ресурсов является необходимым предусловием для регистрации ресурсов в посреднике
- В качестве канонической модели в данной работе рассматривается язык СИНТЕЗ - комбинированная слабоструктурированная и объектная модель данных, нацеленная на разработку предметных посредников для решения задач в средах неоднородных ресурсов
  - Разработан прототип программных средств для поддержки среды предметных посредников

Отображение ЯОД (I)

```
CREATE ARRAY source
< ampExposureId: int64 NULL,
  filterId: int8,
  apMag: double >
[ ra(double),
 de(double),
 objectId=0:*];

CREATE ARRAY objectSummary
< uMag: float NULL,
   gMag: float NULL >
[ objectId=0:* ];
```

```
{ source; in: class;
  instance_type:{
  double ra;
  ra2long: {in: function; params: {-ret/long}; };
  double de;
  de2long: {in: function; params: {-ret/long}; };
  long objectId; metaslot lower: 0; higher: inf; end
  objectIdBounds: {in: invariant;
    {{ all s (source(s) -> s.objectId >= 0) }}
  };
  long ampExposureId;
  short filterId;
  double apMag;
  key: { unique; { ra, de, objectId } };
  definiteness: {obligatory;
    { ra, de, objectId, filterId, apMag } };
  };
```

Отображение ЯОД (II)

- Массив отображается в коллекцию типизированных объектов (класс) объектной модели
- Измерения и атрибуты, составляющие ячейку, представляются атрибутами типа экземпляров класса
- Между встроенными типами ADM (int8, int64, double и т.д.) и встроенными типами объектной модели (short, long, double) устанавливается взаимно-однозначное соответствие
- Совокупность атрибутов, соответствующих измерениям, объявляется уникальной
- Объявляется также, что атрибуты, соответствующие измерениям и не-NULL атрибутам ADM, должны быть определены у всех экземпляров класса
- Для нецелочисленных измерений кроме атрибутов определяются функции, преобразующие нецелочисленные значения в целочисленные
- Ограничения, связанные с нижними и верхними границами целочисленных измерений, отображаются при помощи инвариантов или встроенных утверждений о кардинальности соответствующих атрибутов

Отображение ЯОД (III)

- Сохранение отличительных свойств многомерных массивов («кубов данных»), различающих измерения и атрибуты, составляющие ячейку
  - по набору значений измерений однозначно определяется набор значений атрибутов ячейки (уникальность измерений);
  - ячейка массива всегда определяется полным набором значений измерений (определенность измерений)

Виртуальная и материализованная интеграция

- При *виртуальной* интеграции отображение ЯМД обеспечивает возможность трансляции программ на языке посредника в запросы на языке ресурсов
- При *материализованной* интеграции данные извлекаются из ресурса и представляются в хранилище в канонической модели. При этом программы на языке канонической модели исполняются непосредственно на данных.
  - Отображение ЯМД нужно лишь для того, чтобы убедиться, что отображение моделей сохраняет информацию и семантику операций.
  - Семантически правильное отображение служит базой для процесса Извлечения-Преобразования-Загрузки (ETL), формирующего из данных ресурса данные хранилища: ETL-процесс может быть выражен только в терминах канонической модели.

Язык запросов (программ) объектной модели

- Datalog-подобный язык в объектной среде
- Программа - набор конъюнктивных запросов
- $q(x/T) :- C_1(x_1/T_1), \ldots, C_n(x_n/T_n), F_1(X_1, Y_1), \ldots F_m(X_m, Y_m), B.$

Предикаты-классы, условия, подзапросы

```
q([ra, de]) :- r([ra, de]), filterId= #filterId.
r([ra, de]) :- source([ra, de, apMag]), apMag >= #apMag.
```

```
SELECT apMag FROM
  ( SELECT apMag FROM source
    WHERE apMag >= #apMag )
WHERE filterId = #filterId;
```

Соединение классов

```
q2([ra, de, filterId, uMag]) :-
        source([ra, de, objectId, fliterId]),
        objectSummary([objectId, uMag])
```

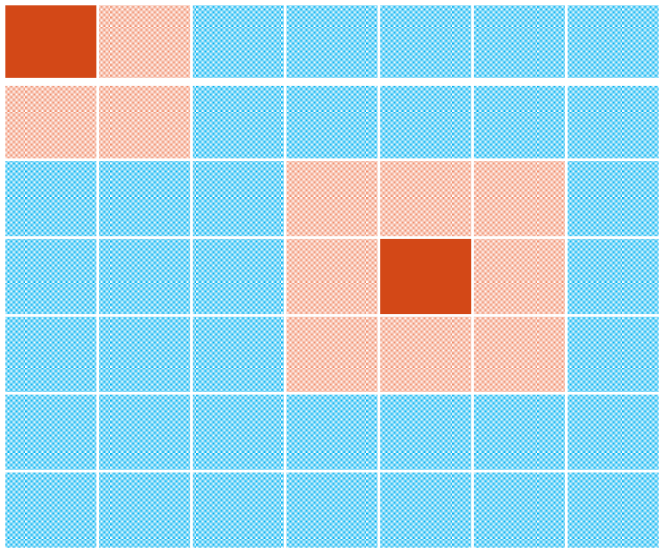
```
SELECT filterId, uMag
FROM source
JOIN objectSummary
ON Source.objectId = ObjectSummary.objectId;
```

Агрегация в бегущем окне
Window Aggregation

```
q([i, j, avgVal]) :- source(x/[i, j]), windowAggr(i, j, avgVal).

{ windowAggr; in: function;
   params: {+i0/long, +j0/long, -res/double};
   {{  source(x/[i, j, val]) & i >= i0 - 1 & i <= i0 + 1 &
         j >= j0 - 1 & j <= j0 + 1 & res = average(x.val)  }}
}
```

```
SELECT avg(val) AS avgVal INTO q
FROM source WINDOW 3, 3;
```



Обновление

```
source(x/[i, j, val]) :-
        source(x/[i, j, val1/val]), abs(val) > 5, val = -val1.
```

```
UPDATE source SET val = -val WHERE abs(val) > 5;
```

Сохранение информации и семантики операций ЯМД при отображении

- AMN - теоретико-модельная нотация, основанная на теории множеств и типизированном языке первого порядка
  - Спецификации AMN - абстрактные машины
  - Интегрированно рассматриваются спецификация пространства состояний и поведения машины
  - Формализуется отношение *уточнения* (спецификация B уточняет спецификацию A, если пользователь может использовать B вместо A, не замечая факта замены A на B)

- Метод доказательства сохранения информации и семантики операций
  - θ - отображение модели исходной модели S в целевую модель T
  - семантика моделей представляется в виде абстрактных машин AMN M_S и M_T
    - структуры данных моделей – классы, массивы представляются переменными машин
    - структур данных представляются инвариантами машин
    - операции моделей данных представляются операциями машин
  - отображение θ сохраняет информацию и семантику операций, если машина M_S уточняет машину M_T

Семантика объектной модели в AMN

```
REFINEMENT  ObjectDM
ABSTRACT_VARIABLES
   classNames, instanceType, typeAttributes, attributeType, …
INVARIANT
  classNames: POW(STRING_Type) &
  !(oo, aa).(oo: dom(objectType) & aa: typeAttributes(objectType(oo)) &
     aa: obligatory(objectType(oo)) =>  (attributeType(aa) = Integer => oo: dom(integerAttributeValue(aa)) ) &
   …
OPERATIONS
    update(cls, attr, exp, cond) =
    PRE cls: classNames & attr: typeAttributes(instanceType(cls)) &
      attributeType(attr) = Integer & exp: INT --> INT & cond: NAT --> BOOL
    THEN
      integerAttributeValue :=   integerAttributeValue <+
      { xx | xx: (NAT*(NAT<->INT)) &  #(oo, val).( oo: objectsOfClass(cls) & val: INT &
         xx = attr |-> ({oo |-> val}) &
         (cond(integerAttributeValue(attr)(oo)) = TRUE => val = exp(integerAttributeValue(attr)(oo))) &
         (cond(integerAttributeValue(attr)(oo)) = FALSE => val = integerAttributeValue(attr)(oo))  ) }
    END
END
```

Семантика ММ-модели в AMN

```
REFINEMENT ArrayDM
REFINES ObjectDM
ABSTRACT_VARIABLES
    arrayNames, arrayCellAttributes, cellAtrributeType, ...
INVARIANT
!(arr, cell1, cell2).(arr: arrayNames & cell1: cells(arr) &  cell2: cells(arr) &
        !(dim).(dim: arrayDimensions(arr) => dimensionValue(cell1, dim) = dimensionValue(cell2, dim))
       => cell1 = cell2 ) & ...
OPERATIONS
update(cls, attr, exp, cond) =
PRE cls: arrayNames & attr: arrayCellAttributes(cls) &  cellAtrributeType(attr) = Integer &
       exp: INT --> INT & cond: NAT --> BOOL
THEN
  integerCellAttributeValue := integerCellAttributeValue <+
  { yy | yy: (NAT*NAT)*INT &  #(cell, val).(cell: cells(cls) & val: INT &  yy = ((cell |-> attr)|-> val) &
  (cond(integerCellAttributeValue(cell, attr)) = TRUE =>  val = exp(integerCellAttributeValue(cell, attr))) &
  (cond(integerCellAttributeValue(cell, attr))= FALSE=> val = integerCellAttributeValue(cell, attr)) ) }
END
END
```

Инвариант уточнения

- Связывает переменные уточняемой и уточняющей машин, добавляется в инвариант уточняющей машины

- множество имен массив совпадает с множеством имен классов
  classNames = arrayNames
- непустые ячейки массивов соответствуют объектам классов
  cells = objectsOfClass
- для любой ячейки значения ее измерений совпадают со значениями соответствующих атрибутов объекта, соответствующего ячейке

```
!(cell, cattr).(cell: NAT & cattr: NAT &
    (cell |-> cattr): dom(integerCellAttributeValue) =>
    cell: dom(integerAttributeValue(cattr)) &
    integerCellAttributeValue(cell, cattr) = integerAttributeValue(cattr)(cell) )
```

Доказательство уточнения

- Машины ObjectDM и ArrayDM загружены в инструментальное средство доказательства уточнения Atelier И
- Автоматически сгенерированы теоремы уточнения
  - для операции update – 10 теорем
- Теоремы доказываются автоматически или интерактивно
  - для операции update – 3 теоремы доказаны автоматически
- Доказательство проводится для всех операций ЯМД

Дальнейшая работа

- построение трансформации, реализующей отображение
- расширение инструментальных средств поддержки предметных посредников для виртуальной интеграции SciDB-ресурсов:
  - расширение средств регистрации ресурсов в посреднике трансформацией ЯОД ADM в каноническую модель
  - создание SciDB-адаптера - специального ПО, связывающего исполнительную среду посредников с SciDB-ресурсами (составной частью адаптера является трансформация ЯМД)
- применение технологии предметных посредников для решения научных задач в некоторой предметной области над множеством неоднородных ресурсов, включающем SciDB-ресурсы