Анализ системного риска совместного кредитования над неоднородными коллекциями данных*

С. А. Ступников[1], Д. О. Брюхов[2], Н. А. Скворцов[3]

Аннотация: В статье рассматривается подход к решению задачи анализа системного риска совместного кредитования в области финансового макромоделирования – одной из областей с интенсивным использованием данных – над неоднородными коллекциями данных в виртуально-материализованной среде интеграции. Виртуальная интеграция в среде осуществляется с использованием технологии предметных посредников. Материализованная интеграция реализуется с использованием свободно распространяемой платформы распределенного хранения и обработки данных Hadoop, а также системы Hive, предназначенной для организации реляционных хранилищ данных над Hadoop.

Ключевые слова: системный риск совместного кредитования; решение задач; интеграция данных; неоднородные коллекции данных

1 Введение

Рост объема и разнообразия данных в науке и бизнесе в последние годы ведет к необходимости следования *Четвертой научной парадигме*, подчеркивающей роль данных в исследованиях с интенсивным использованием данных. Эта роль заключается в том, что новые знания, как и принятие решений, являются результатом анализа данных [1]. В различных областях, называемых *областями с интенсивным использованием данных*, происходит накопление массивных коллекций разнородных данных, представленных в различных моделях данных.

Спектр используемых моделей данных необычайно широк: он включает традиционные реляционные модели, объектные модели, основанные на много-

* Работа выполнена при поддержке РФФИ (проекты 13-07-00579, 14-07-00548).

[1] Институт проблем информатики ФИЦ ИУ РАН, sstupnikov@ipiran.ru

[2] Институт проблем информатики ФИЦ ИУ РАН, brd@ipi.ac.ru

[3] Институт проблем информатики ФИЦ ИУ РАН, nskv@ipi.ac.ru

мерных массивах модели, графовые модели, модели ключ-значение, документные модели, семантические модели (RDF, OWL) и другие. Такое разнообразие моделей данных, увеличивающееся со временем, приводит к необходимости создания подходов к интеграции моделей и коллекций данных, представленных в этих моделях, разработки подходов к решению задач над неоднородными коллекциями. В работе [2] была предложена архитектура комбинированной виртуально-материализованной среды интеграции неоднородных коллекций структурированных, слабоструктурированных и неструктурированных данных (рис. 1). Среда поддерживает как виртуальную, так и материализованную интеграцию коллекций данных, представленных как в традиционных (реляционных), так и нетрадиционных моделях данных.

Виртуальная интеграция в среде осуществляется с использованием технологии предметных посредников [3]. Посредники образуют промежуточный слой между пользователем (приложением) и неоднородными информационными ресурсами; данные из ресурсов в посреднике не материализуются.

При материализованной интеграции предполагается создание хранилища данных. В хранилище загружаются подлежащие интеграции коллекции данных, при этом данные преобразуются из схемы коллекции в общую схему хранилища. Материализованная интеграция реализуется с использованием свободно распространяемой платформы распределенного хранения и обработки данных Hadoop [4]; а также системы Hive [5], предназначенной для организации реляционных хранилищ данных над Hadoop.

В настоящей статье рассматривается подход к решению задач над неоднородными коллекциями данных в виртуально-материализованной среде интеграции. Подход иллюстрируется на примере задачи из области финансового макромоделирования – одной из областей с интенсивным использованием данных.

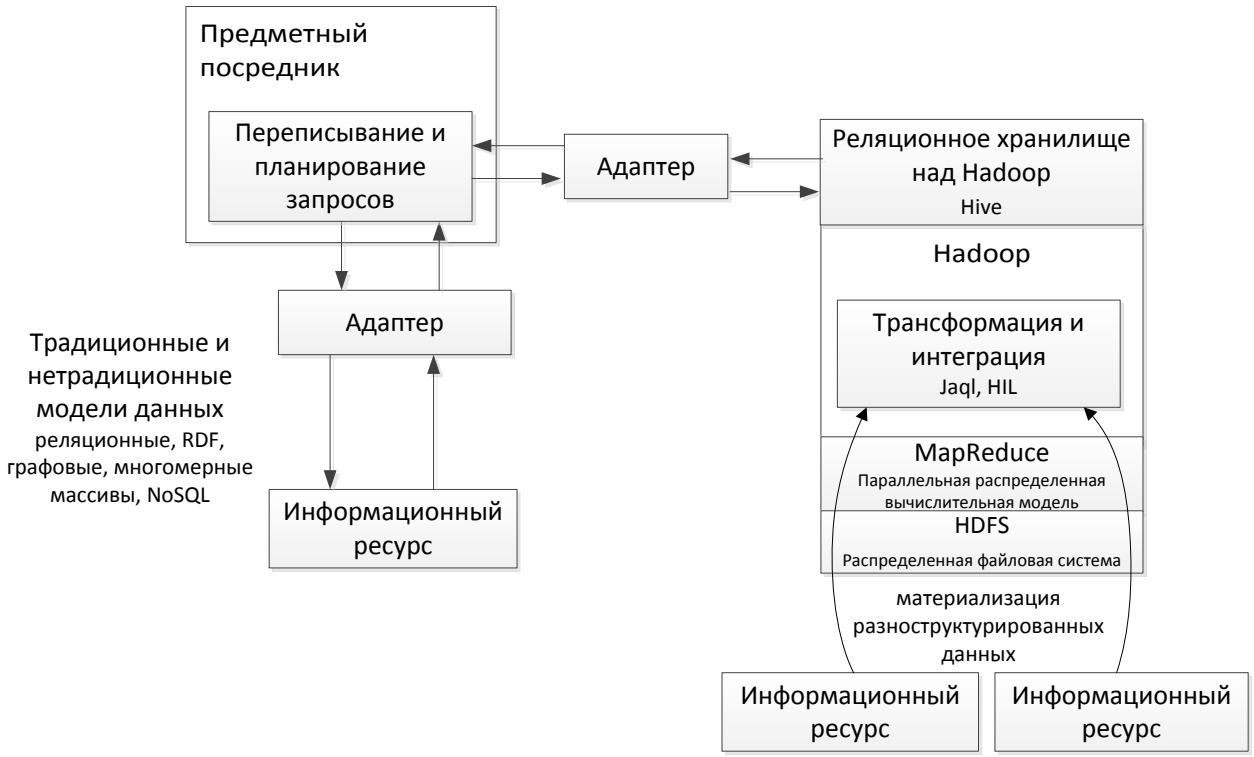

Рис. 1 Архитектура среды виртуально-материализованной интеграции

Статья структурирована следующим образом. В разделе 2 рассмотрена постановка задачи анализа системного риска совместного кредитования. В разделе 3 рассмотрены основные решения задачи в среде интеграции: определение концептуальной схемы предметной области задачи и декларативной спецификации задачи (подразделы 3.1, 3.2), выбор релевантных предметной области ресурсов и способов их интеграции (подраздел 3.3), определение схем ресурсов, подлежащих виртуальной интеграции, и взглядов, связывающих их со схемой посредника (подраздел 3.5), определение схемы хранилища для материализации коллекций данных и взглядов, связывающих ее со схемой посредника (подраздел 3.6), а также архитектура среды решения задачи (подраздел 3.7).

2 Интенсивное использование данных в задаче анализа системного риска совместного кредитования

В данной статье в качестве примера рассматривается конкретная задача анализа системного риска совместного кредитования. Она состоит в выявлении ведущих игроков на кредитном рынке, банкротство которых может вызвать си-

стемный финансовый кризис, оказать влияние на финансовое положение множества других игроков [6, 7].

Исходными данными для задачи являются документы (записи) о совместных (синдицированных) кредитах. Записи содержат даты подписания и погашения кредита, объем кредита, состав участников синдиката, предоставляющего кредит, и другую информацию. На основании исходных данных формируется граф совместного кредитования. Вершинами графа являются организации. Две вершины соединяются ненаправленным ребром, если соответствующие организации предоставляют некоторый совместный кредит (возможно, вместе с другими организациями). Так, если кредит предоставлен совместно пятью банками, ребрами соединяются всевозможные пары, образуемые этими пятью банками. Ребру приписывается вес в зависимости от того, сколько совместных кредитов предоставлено парой организаций. Банкротство некоторого банка, очевидно, окажет прямое влияние на банки, связанные с ним ребрами в графе совместного кредитования, а также окажет опосредованное влияние на банки, связанные с ним в графе путями [6].

Определение того, банкротство каких банков имеет наибольший вклад в риск системного финансового кризиса, базируется на вычислении *центральности* вершин в графе совместного кредитования.

Существуют различные варианты центральности вершин в графе. *Степенная центральность* вершины вычисляется как сумма весов ребер, связанных с вершиной. *Центральность по посредничеству (betweenness centrality)* [8] вычисляется как мера отношения числа кратчайших путей, проходящих через вершину, к общему числу кратчайших путей в графе. *Центральность по собственному значению (eigenvalue centrality)* [9] вычисляется как мера связи вершины с другими вершинами с высокой центральностью. Вершины с наивысшими показателями центральности с высокой вероятностью являются *критическими хабами* для сети совместного кредитования.

После того как в графе совместного кредитования определены вершины с наибольшей центральностью, представитель финансового регулятора может

быть заинтересован в дополнительной информации о соответствующих компаниях. К такой информации относятся связи с другими компаниями (владение, дочерняя компания), ключевые лица компании (директора), агрегированные финансовые данные и т. д.

3 Решение задач в виртуально-материализованной среде интеграции

Решение задачи (класса задач) в виртуально-материализованной среде интеграции включает следующие этапы:

- определение *концептуальной схемы* предметной области задачи; такая схема становится спецификацией предметного посредника для виртуальной интеграции релевантных ресурсов;
- *описание задачи* в виде декларативной программы над концептуальной схемой;
- определение *ресурсов, релевантных предметной области* (содержащих данные, необходимые для решения задачи), определение способа их интеграции (виртуальная или материализованная);
- создание *отображений моделей данных информационных ресурсов*, подлежащих виртуальной интеграции, в каноническую информационную модель предметных посредников;
- создание *адаптеров* информационных ресурсов, подлежащих виртуальной интеграции;
- определение *схем информационных ресурсов*, подлежащих виртуальной интеграции, и *отображение* этих *схем* в каноническую модель (создание *локальных* схем ресурсов);
- определение *взглядов* (представлений), связывающих элементы схемы посредника и схем ресурсов для обеспечения возможности переписывания запросов при виртуальной интеграции ресурсов в посреднике [3];
- определение *схемы хранилища* для материализации коллекций данных и ее отображение в каноническую модель;

- определение *взглядов* (представлений), связывающих элементы схемы посредника и схемы хранилища для обеспечения возможности переписывания запросов при виртуальной интеграции хранилища в посреднике;
- создание *преобразований коллекций данных*, подлежащих материализованной интеграции, в реляционную модель хранилища;
- создание приложения, связывающего исполнительную среду предметных посредников [3], адаптеры информационных ресурсов и хранилище материализованных данных.

Все эти этапы более подробно будут рассмотрены ниже на примере задачи анализа системного риска совместного кредитования.

3.1 Концептуальная схема предметной области задачи

Определение концептуальной схемы производится с использованием языка СИНТЕЗ [10], используемого в качестве канонической информационной модели предметных посредников.

Концептуальная схема предметной области задачи системного риска совместного кредитования представляется в виде модуля (модуль является основной единицей спецификации канонической модели) *ColendingSystemicRisk*:

```
{ ColendingSystemicRisk; in: module;
class_specification: ...
function: ...
}
```

Модуль содержит секцию классов (моделирующих множества объектов предметной области) и секцию функций. Секция классов включает, в частности, классы *companies, persons, colendings*:

```
{ companies; in: class;
  instance_section: {
    id: string;
    names: {set; type_of_element: string;};
    ownedBy: {set; type_of_element: Company;};
    ownerOf: {set; type_of_element: Company;};
    keyPersons: {set; type_of_element: Person;};
}},
{ persons; in: class;
  instance_section: {
    id: string;
    names: {set; type_of_element: string;};
    keyPersonOf: {set; type_of_element: Company;};
```

```
}},
{ colendings; in: class;
  instance_section: {
    id: string;
    colender1: string;
    colender2: string;
    numberOfColendings: integer;
}};
```

Класс *companies* отвечает компаниям, предоставляющим кредиты (например, банкам). В секции описания экземпляров класса (*instance_section*) определены атрибуты *id* (уникальный идентификатор компании), *names* (различные варианты названия компании), *ownedBy* (множество компаний, владеющих долей данной компании), *ownerOf* (множество компаний, долями которых владеет данная компания), *keyPersons* (множество ключевых лиц компании – директоров, управляющих).

Класс *persons* отвечает лицам, принимающим участие в управлении компаниями. Для экземпляров класса определены атрибуты *id* (уникальный идентификатор персоны), *names* (различные варианты имени персоны), *keyPersonOf* (множество компаний, в которых персона занимает управляющую должность).

Класс *colendings* отвечает отношению совместного кредитования между компаниями. Для экземпляров класса определены атрибуты *id* (уникальный идентификатор отношения кредитования), *colender1* и *colender2* (идентификаторы пары компаний, участвующих в выдаче совместных кредитов), *numberOfColendings* (количество совместно выданных кредитов). Таким образом, совокупность экземпляров класса *colendings* задает граф совместного кредитования, вершинами которого являются компании. Атрибут *numberOfColendings* задает вес ребра в графе.

Секция функций включает, в частности, функцию *isValidColending*:

```
{ isValidColending; in: function;
  params: {+rel/integer, +clnd1/integer, +clnd2/integer, -returns/boolean};
  predicative: {
    ex c/colendings.inst, cmp1/companies.inst, cmp2/companies.inst(
      is_in(c, colending) & is_in(cmp1, companies) & is_in(cmp2, companies) &
      c.id = rel & cmp1.id = clnd1 & cmp2.id = clnd2 &
      (clnd1 = rel.colender1 & clnd2 = rel.colender2 -> returns = true) &
      (clnd1 <> rel.colender1 | clnd2 <> rel.colender2 -> returns = false)  )
};};
```

Функция принимает на вход идентификатор экземпляра класса *colendings* (*rel*), а также два идентификатора экземпляров класса *companies* (*clnd1* и *clnd2*). Предикативная спецификация функции задается формулой типизированной логики предикатов первого порядка [10]. Функция возвращает значение *true* в том случае, если компании с идентификаторами *clnd1* и *clnd2* являются сокредиторами в отношении совместного кредитования с идентификатором *rel*.

3.2 Описание задачи в виде декларативной программы

Первая часть задачи заключается в вычислении центральности компаний в графе совместного кредитования. Например, для вершинной центральности такое вычисление может быть описано в виде одного Даталог-подобного правила канонической модели [10]:

```
degreeCentrality(x/[companyId, sumColendings]) :-
companies(comp/[companyId: id]) & companies(neighbour/[neigbourId:id]) &
colendings(clnd/[colendId:id, numberOfColendings]) &
isValidColending(colendId, companyId, neigbourId) &
group_by(comp) &
sumColendings = sum(colend.numberOfColendings).
```

Предикат *degreeCentrality* в голове правила содержит атрибуты *companyName* и *sumColendings*, т. е. для каждой компании (вершины) возвращается суммарное число кредитов, выданных ей совместно с другими компаниями (сумма весов ребер). Переменная *comp* объявляется пробегающей по экземплярам класса *companies* с использованием одноименного предиката-класса. Атрибут *id* переименовывается в *companyId* (конструкция *id1: id*) для предотвращения смешивания с одноименными атрибутами других переменных. Аналогично объявляется переменная *neighbour*. Переменная *clnd* пробегает по экземплярам класса *colendings*. Предикат-функция *isValidColending* устанавливает связь между значениями *comp*, *neighbour*, *colend* через идентификаторы: компания *comp1* должна быть связана ребром *colend* с компанией *comp2*. Операция *group_by* производит группировку по имени компании, а функция *sum* суммирует веса ребер в соответствующей группе.

Вторая часть задачи заключается в извлечении дополнительной информации о компаниях с наибольшей центральностью. Пусть, например, наивысшей центральностью обладает компания *ING Group*. Запрос, выдающий идентификаторы всех компаний (*owned*), в которых у *ING* есть доля, представляется следующим декларативным правилом:

```
ownedByING([owned]) :-
  companies(x/[names]) & companies(y/[owned: id]) &
  is_in('ING', x.names) & is_in(y, x.ownerOf).
```

Здесь встроенный предикат *is_in* означает принадлежность элемента (*'ING'*) множеству (*x.names*).

Запрос, выдающий всех лиц (*pers*), принимающих участие в управлении компанией *ING* одновременно с управлением некоторой другой компанией (*cmpn*), представляется следующим декларативным правилом:

```
overlappedPersonsOfING([pers, cmpn]) :-
  companies(x) & companies(y/[cmpn:iri]) & persons(z/[pers: iri])
  is_in('ING', x.names) & is_in(z, x.keyPersons) & is_in(z, y.keyPersons).
```

3.3 Релевантные ресурсы и способы их интеграции

Для решения задачи могут быть использованы, например, следующие ресурсы:

- база данных на основе графовой СУБД Neo4j [11], содержащая информацию о совместном кредитовании компаний, представленную в виде графа;
- триплетная (RDF [12]) база данных DBpedia, содержащая структурированную информацию, извлеченную из Википедии (в частности, информацию о компаниях и персонах).

Neo4j представляет собой популярную графовую СУБД, поддерживающую декларативный язык запросов Cypher [11]. Естественным представляется использование такой СУБД для решения задач на графах и ее виртуальная интеграция в посреднике. База данных формируется на основе информации о сов-

местных кредитах, публикуемой на сайтах финансовых новостных агентств. Пример того, как может выглядеть информация о кредите, приведен в табл. 1.

Таблица 1 Пример данных о совместном кредите

Кредит	ВТБ, 2.2005
Страна	Россия
Объем	450 000 000 USD
Ставка по кредиту	LIBOR + 120.BP
Дата подписания	Февраль 2005
Период	36 месяцев
Дата погашения	Февраль 2008
Уполномоченные ведущие организаторы (MLAs)	ABN Amro, Citigroup и ING

В этом случае в графе совместного кредитования появятся вершины, соответствующие компаниям *ABN Amro*, *Citigroup*, *ING*, и ребра, попарно соединяющие эти вершины.

Дополнительную информацию о компаниях и персонах можно извлечь из DBpedia. Эта информация представлена в структурированном виде, удобном для материализации в Hadoop и преобразования к реляционному виду.

Доступ к DBPedia осуществляется посредством запросов на языке SPARQL через точку доступа http://dbpedia.org/sparql. Например, уникальные идентификаторы (URI) кредитных организаций могут быть извлечены при помощи следующего запроса:

```
SELECT DISTINCT ?bank
WHERE { {?bank rdf:type dbo:Bank} UNION {?bank dbp:industry dbr:Bank} }
```

Уникальные идентификаторы (URI) персон могут быть извлечены при помощи следующего запроса:

```
SELECT DISTINCT ?person
WHERE { {?person rdf:type dbo:Person} UNION {?person rdf:type foaf:Person} }
```

Далее, конкретный RDF-документ, описывающий банк или персону, можно извлечь из DBPedia в необходимом формате с использованием найденного URI. Так, для компании *BNP Paribas* с идентификатором *http://dbpedia.org/resource/BNP_Paribas*, RDF-документ с описанием в формате

JSON [13] доступен по ссылке http://dbpedia.org/data/BNP_Paribas.json (приведена лишь часть документа):

```
{ "http://dbpedia.org/resource/ING_Group" : {
  "http://dbpedia.org/ontology/industry" : [
    {"type": "uri", "value": "http://dbpedia.org/resource/Financial_services"},
    {"type" : "uri", "value" : "http://dbpedia.org/resource/Bank" } ],
  "http://dbpedia.org/property/assets" : [ { "type" : "literal",
    "value" : "1.169E12" , "datatype" : "http://dbpedia.org/datatype/euro" } ] ,
  "http://dbpedia.org/property/keyPeople" : [ { "type" : "literal",
    "value": "Ralph Hamers Patrick Flynn Jeroen van der Veer", "lang": "en" }],
  "http://dbpedia.org/resource/Bank_Mendes_Gans" : {
    "http://dbpedia.org/ontology/parentCompany" :
     [{ "type" : "uri", "value" : "http://dbpedia.org/resource/ING_Group" } ] },
  "http://dbpedia.org/resource/Voya_Financial" : {
    "http://dbpedia.org/ontology/parentCompany" :
    [ { "type" : "uri", "value" : "http://dbpedia.org/resource/ING_Group" } ] }
}
```

В документе содержится, в частности, информация о ключевых лицах компании (*keyPeople*), объеме имущества (*assets*), дочерних компаниях (связь, обратная *parentCompany*) и т. д.

Таким образом, все необходимые дополнительные данные о компаниях извлекаются из DPPedia и загружаются в Hadoop в виде RDF-документов.

3.4 Отображение моделей данных информационных ресурсов в каноническую модель и создание адаптеров

Необходимым предусловием виртуальной интеграции модельно однородного класса информационных ресурсов (представленных с использованием одной модели данных) в предметных посредниках является *унификация модели данных ресурсов* – ее отображение в каноническую информационную модель (служащую общим языком в среде разнообразных моделей ресурсов), сохраняющее информацию и семантику операций языка манипулирования данными, а также разработка адаптера для сопряжения ресурсов данного класса со средой исполнения предметных посредников.

Основные принципы отображения модели данных атрибутированных графов и вопросы доказательства сохранения информации и семантики операций при этом отображении рассмотрены в работе [14]. Общие принципы построения адаптеров изложены в работе [15].

Для реализации конкретного адаптера СУБД Neo4j необходима разработка трансформации запросов (программ) канонической модели в запросы на языке Cypher, основывающейся на упомянутом отображении моделей. Вопросы построения такой трансформации будут вынесены в отдельную статью.

3.5 Схема ресурса, подлежащего виртуальной интеграции, и взгляды, связывающие ее со схемой посредника

Схемой ресурса, подлежащего виртуальной интеграции – базы данных СУБД Neo4j, содержащей информацию о совместном кредитовании компаний, можно считать шаблоны операций языка Cypher, порождающие экземпляры базы данных. Например, операция вида

```
merge (c:Organization{id: URI});
```

создает в базе данных вершину с меткой *Organization* и атрибутом *id*.

Следующая операция:

```
match (c1: Organization{id: URI1}), (c2: Organization{id: URI2})
merge (n1)-[e:colends]-(n2)
on create set e.numberOfColendings = 1
on match set e.numberOfColendings = e.numberOfColendings + 1;
```

создает в базе данных ребро с меткой *colends* между двумя вершинами, помеченными как *Organization*, и устанавливает значение атрибута *numberOfColendings* равным *1* (либо увеличивает значение на *1*, если ребро уже существует в базе).

Из этих операций можно заключить, что в базе данных имеются вершины типа *Organization* с атрибутом *id* и их соединяют ребра типа *colends* с атрибутом *numberOfColendings*.

Согласно принципам отображения модели данных атрибутированных графов в каноническую модель [14] схема базы данных, порождаемой приведенными выше операциями, представляется в канонической модели следующим образом (приведено упрощенное подмножество соответствующей спецификации):

```
{ PropertyGraph; in: module;
class_specification:
{ vertices; in: class;
```

```
  instance_section: {
    id: string;
}},
{ edges; in: class;
  instance_section: {
    id: string;
    startVertex: string;
    endVertex: string;
}};
function:
isValidEdge: { in: predicate;
  params: {+edg/string, +stVtx/string, +endVtx/string, returns/boolean};
};
}
{ Colending; in: module; import: PropertyGraph;
{ Organization; in: class; superclass: vertices;
  instance_section: {
   id: string;
};},
{ colends; in: class; superclass: edges;
  instance_section: {
    numberOfColendings: integer;
};};
}
```

Здесь модуль *PropertyGraph*, содержащий классы *vertices* (вершины), *edges* (ребра), задает граф общего вида [14], а модуль *Colending*, содержащий классы *Organization* и *colends*, задает атрибуты вершин и ребер конкретных типов. Вышеприведенная схема, представленная в канонической модели, называется *локальной схемой* ресурса (базы данных о совместном кредитовании компаний).

Для обеспечения возможности переписывания запросов при виртуальной интеграции ресурсов в посреднике необходимо определение взглядов, связывающих элементы схемы посредника и схем ресурсов [16, 17]. Взгляды представляются Даталог-подобными декларативными правилами канонической модели. Для сопоставления вышеприведенных фрагментов схем достаточно трех следующих взглядов вида LAV (Local As View), определяющих, как элемент (класс или функция) локальной схемы выражается через элементы схемы посредника:

```
Colending.organization(x/[id]):- companies(x/[id]).

Colending.colends(x/[id, colender1, colender2, numberOfColendings]):-
colendings(x/[id, colender1: startVertex, colender2: endVertex,
             numberOfColendings]).

Colending.isValidEdge(colend, comp1, comp2):-
isValidColending(colend, comp1, comp2).
```

Первый взгляд задает выражение класса *organization* через класс *companies*, второй – класса *colends* через класс *colendings*, третий – функции *isValidEdge* через функцию *isValidColending*.

Применение алгоритма переписывания запросов [16] с использованием вышеприведенных взглядов к правилу вычисления вершинной центральности (раздел 3.2) позволяет получить запрос в терминах локальной схемы:

```
degreeCentrality(x/[companyId, sumColendings]) :-
organization(comp/[companyId: id]) & organization(neighbour/[neigbourId:id]) &
colends(clnd/[colendId:id, numberOfColendings]) &
isValidEdge(colendId, companyId, neigbourId) &
group_by(comp) &
sumColendings = sum(colend.numberOfColendings).
```

Согласно принципам отображения языка правил канонической модели в язык Cypher [14] адаптер Neo4j должен преобразовывать такое правило в следующий запрос на языке Cypher:

```
match(comp: Organization)-[clnd: colends]-(neighbour: Organization)
return comp.id as companyId, sum(clnd.numberOfColendings) as sumColendings
```

3.6 Схема хранилища для материализации коллекций данных и взгляды, связывающие ее со схемой посредника

Фрагмент реляционной схемы для представления информации об именах банков (отношение *banks*) и персон (отношение *persons*), отношении владения между компаниями (*bankOwners*), ключевых лицах в компаниях (отношение *keyPersons*) выглядит следующим образом:

```
banks(iri STRING, name STRING)
persons(iri STRING, name STRING)
bankOwners(owner STRING, owned STRING)
bankKeyPersons(person_iri STRING, bank_iri STRING)
```

Локальная схема в канонической модели, соответствующая данной реляционной схеме, выглядит следующим образом:

```
{ BanksPersons; in: module;
class_specification:
{ banks; in: class;
  instance_section:{
    iri: string;
    name: string;
}},
{ persons; in: class;
  instance_section:{
    iri: string;
    name: string;
```

```
}},
{ bankOwners; in: class;
  instance_section:{
    owner: string;
    owned: string;
}},
{ keyPersons; in: class;
  instance_section:{
    person_iri: string;
    bank_iri: string;
}};
}
```

Схема представляется модулем *BanksPersons*, каждому отношению реляционной схемы соответствует одноименный класс модуля, каждому атрибуту отношения – одноименный атрибут типа экземпляров соответствующего класса.

Вопросы построения преобразования RDF-коллекций компаний и персон в реляционное представление при помощи языка Jaql [18] в Hadoop, необходимого для загрузки данных в реляционное хранилище Hive, будут рассмотрены в отдельной статье.

После того как данные загружены в Hive (осуществлена материализованная интеграция), необходимо осуществить виртуальную интеграцию хранилища в посреднике в соответствии с подходом, изложенным в [2]. Для этого, как и в случае с интеграцией СУБД Neo4j, требуется установить соответствие между элементами схемы хранилища и схемы посредника при помощи взглядов. LAV-взгляды для приведенных фрагментов схем выглядят следующим образом:

```
BanksPersons.banks(b/[iri, name]) :-
companies(c/[iri, names]) & is_in(name, names).

BanksPersons.people(p/[iri, name]) :-
persons(p/[iri, names]) & is_in(name, names).

BanksPersons.bankOwners(bo/[owner, owned]) :-
companies(x/[owner: iri]) & companies(y/[owned: iri]) & is_in(y, x.ownerOf).

BanksPersons.bankKeyPersons(kp/[person_iri, bank_iri]) :-
persons(p/[person_iri: iri]) & companies(b/[bank_iri: iri]) &
is_in(p, b.keyPersons).
```

Здесь встроенный предикат *is_in(x, y)* обозначает принадлежность элемента *x* множеству *y*. Первый взгляд выражает класс *banks* схемы хранилища через класс *companies* схемы посредника, второй – класс *people* через класс *per-*

sons, третий – класс *bankOwners* через класс *companies*, четвертый – класс *bankKeyPersons* через классы *persons* и *companies*.

Применение алгоритма переписывания запросов [16] с использованием вышеприведенных взглядов к правилу вычисления лиц с конфликтом интересов в компании *ING* (раздел 3.2) позволяет получить запрос в терминах локальной схемы:

```
ownedByING([owned]) :-
 banks(x/[iri, name]) & name = 'ING' &
  bankOwners(y/[iri, owned]).
```

Адаптер Hive должен преобразовывать такое правило в следующий запрос на языке HiveQL:

```
SELECT owned
FROM banks x JOIN bankOwners y ON x.iri = y.iri
WHERE x.name like "ING";
```

3.7 Архитектура среды решения задачи анализа системного риска совместного кредитования

Архитектура среды решения задачи представлена на рис. 2.

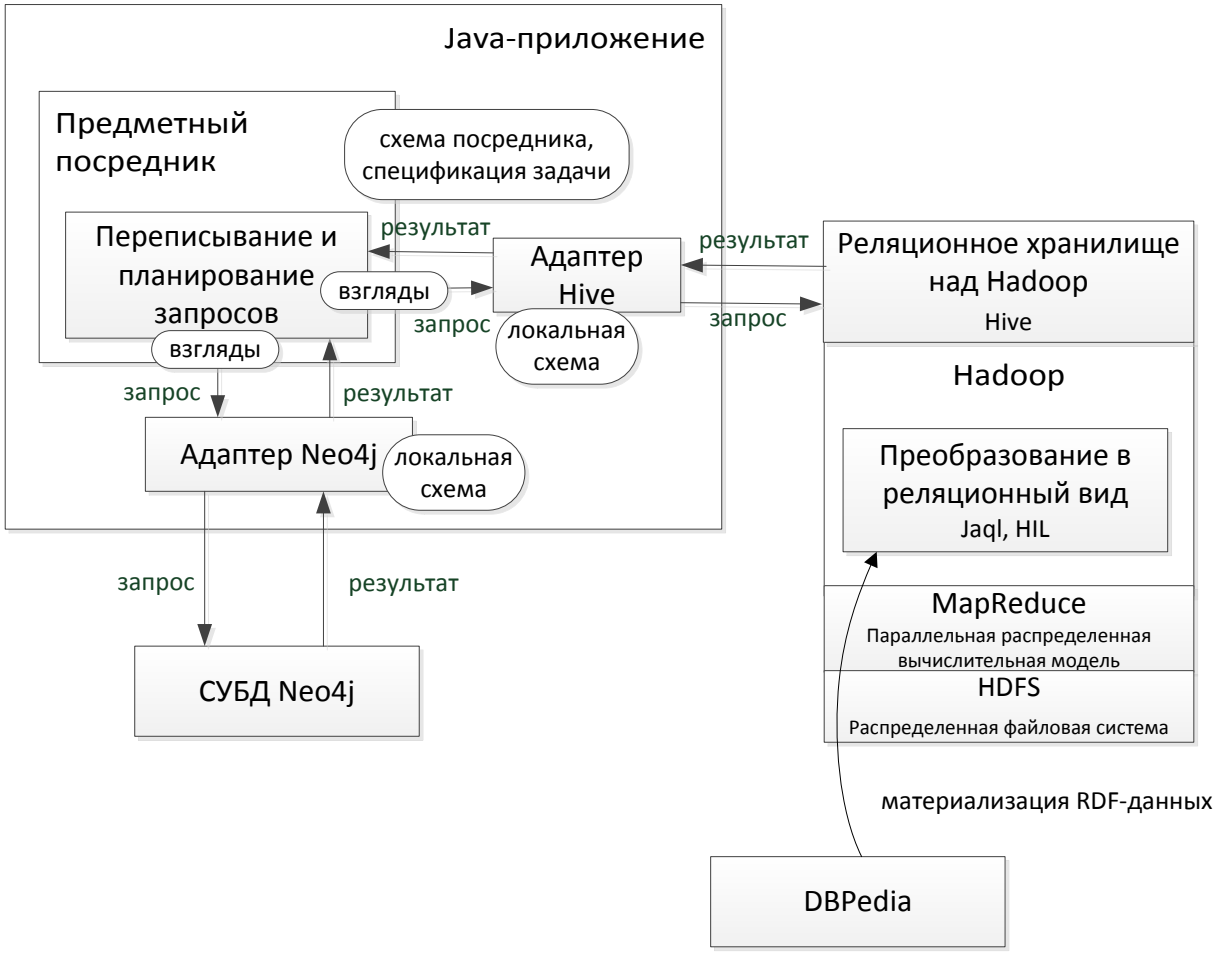

Рис. 2 Архитектура среды решения задачи анализа системного риска совместного кредитования

Среда включает СУБД Neo4j; RDF-базу данных DBPedia; хранилище на основе Hive над Hadoop; а также приложение, связывающее среду исполнения предметных посредников и адаптеры ресурсов.

Для решения задачи анализа системного риска совместного кредитования в СУБД Neo4j были загружены данные о 1500 совместных кредитах, выданных компаниям России и стран ближнего зарубежья. Данные были получены с веб-страниц сайта одного из финансовых новостных агентств. Полученный граф совместного кредитования включает около 400 вершин (компаний) и 4500 ребер.

Также из базы данных DBPedia были извлечены RDF-документы о финансовых организациях и персонах, принимающих участие в управлении такими организациями. RDF-документы были преобразованы к реляционному виду и помещены в хранилище Hive.

Пример данных, полученных в результате решения задачи в комбинированной среде интеграции, приведен в табл. 2. Были обнаружены четыре компании, нормализованная вершинная центральность которых превышает 0,5 (кандидаты в критические хабы). На основании данных, извлеченных из DBPedia, были найдены компании, владельцами или совладельцами которых являются критические хабы (приведена лишь часть данных).

Таблица 2 Пример результата решения анализа задачи риска совместного кредитования

Название	**Нормализованная вершинная центральность**	**Владение другими компаниями**
ING Group	1	Bank Mendes Gans Voya Financial
UniCredit	0,608	Bank Austria Pioneer Investments
HSBC Bank	0,581	HSBC Bank Canada HSBC Bank Australia
BNP Paribas	0,504	SBI Life Insurance Company Bank Insinger de Beaufort

4 Заключение

В статье рассмотрен подход к решению задачи анализа системного риска совместного кредитования в области финансового макромоделирования над неоднородными коллекциями данных в виртуально-материализованной среде интеграции. Основные этапы решения иллюстрируются на примере задачи анализа системного риска совместного кредитования. В статье остались нераскрытыми следующие важные вопросы: создание трансформации графовой модели данных Neo4j в каноническую информационную модель и построение адаптера Neo4j, а также создание преобразования коллекции RDF-документов из DBPedia в реляционные данные, пригодные для загрузки в хранилище Hive над Hadoop. Эти вопросы станут предметом отдельной статьи.

Литература

1. *Hey T., Tansley S., Tolle K.* The Fourth Paradigm – Data Intensive Scientific Discovery. 2009. http://goo.gl/edvr6W.
2. *Ступников С. А., Вовченко А. Е.* Комбинированная виртуально-материализованная среда интеграции больших неоднородных коллекций данных // Электронные библиотеки: перспективные методы и технологии, электронные коллекции (RCDL 2014): Труды 16-й Всероссийской научной конференции. – Дубна: ОИЯИ, 2014. С. 339–348.
3. *Брюхов Д. О., Вовченко А. Е., Захаров В. Н., Желенкова О. П., Калиниченко Л. А., Мартынов Д. О., Скворцов Н. А., Ступников С. А.* Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий // Информатика и ее применения, 2008. Т. 2. Вып. 1. С. 2–34.
4. Apache Hadoop Project. 2014. http://hadoop.apache.org.
5. *Capriolo E., Wampler D., Rutherglen J.* Programming Hive Data Warehouse and Query Language for Hadoop. – O'Reilly Media, 2012.
6. *Burdick D., Hernández M. A., Ho H., Koutrika G., Krishnamurthy R., Popa L., Stanoi I., Vaithyanathan S., Das S. R.* Extracting, Linking and Integrating Data from Public Sources: A Financial Case Study // IEEE Data Eng. Bull., 2011. Vol. 34. No. 3. P. 60–67.
7. *Burdick D., Evfimievski A., Krishnamurthy R., Lewis N., Popa L., Rickards S., Williams P.* Financial Analytics from Public Data // SIGMOD/PODS'2014: Proceedings of the International Workshop on Data Science for Macro-Modeling Conference. – New York: ACM, 2014. P. 1–6.
8. *Freeman L. C.* A Set of Measures of Centrality Based on Betweenness // Sociometry, 1977. Vol. 40. No. 1. P. 35–41.
9. *Bonacich P.* Power and Centrality: A Family of Measures // American Journal of Sociology, 1987. Vol. 92. No. 5. P. 1170–1182.
10. *Kalinichenko L. A., Stupnikov S. A., Martynov D. O.* SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. – M.: IPI RAN, 2007. 171 p.
11. The Neo4j Manual. 2014. http://goo.gl/cHiOGF.
12. *Cyganiak R., Wood D., Lanthaler M. (eds.)* RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 25 February 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225.
13. Introducing JSON. 2014. http://www.json.org.

14. *Ступников С. А.* Отображение графовых моделей данных в каноническую модель в системах с интенсивным использованием данных // Системы высокой доступности, 2014. Вып. 2. С. 13–31.
15. *Вовченко А. Е.* Рассредоточенная реализация приложений в среде предметных посредников: Дис. … канд. техн. наук. – М.: ИПИ РАН, 2012. 216 с.
16. *Kalinichenko L. A., Martynov D. O., Stupnikov S. A.* Query rewriting using views in a typed mediator environment // Advances in Databases and Information Systems: Proceedings of the 8th East European Conference. LNCS 3255. – Berlin, Heidelberg: Springer-Verlag, 2004. P. 37–53.
17. *Briukhov D. O., Kalinichenko L. A., Martynov D. O.* Source Registration and Query Rewriting Applying LAV/GLAV Techniques in a Typed Subject Mediator // RCDL'2007: Proceedings of the Ninth Russian Conference on Digital Libraries. – Pereslavl-Zalesskij: Pereslavl University, 2007. P. 253–262.
18. *Beyer K. S., Ercegovac V., Gemulla R., Balmin A., Eltabakh M., Kanne C.-Ch., Ozcan F., Shekita E. J.* Jaql: A Scripting Language for Large Scale Semistructured Data Analysis // Proceedings of the VLDB Endowment, 2011. Vol. 4. No. 12. P. 1272–1283.

References

1. Hey T., S. Tansley, and K. Tolle. 2009. The Fourth Paradigm – Data Intensive Scientific Discovery. Available at: http://goo.gl/edvr6W (accessed November 17, 2015).
2. Stupnikov, S. A., A. E. Vovchenko. 2014. Kombinirovannaya virtual'no-materializovannaya sreda integratsii bol'shikh neodnorodnykh kollektsiy dannykh [Combined Virtual and Materialized Environment for Integration of Large Heterogeneous Data Collections] // Trudy 16-y Vserossiyskoy konferentsii «Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii» [Proceedings of the 16th Russian Conference on Digital Libraries]. CEUR Workshop Proceedings. 1297:339–348.
3. Briukhov, D. O., A. E. Vovchenko, V. N. Zakharov, O. P. Zhelenkova, L. A. Kalinichenko, D. O. Martynov, N. A. Skvortsov, S. A. Stupnikov. 2008. Arhitektura promezhutochnogo sloja predmetnyh posrednikov dlja reshenija zadach nad mnozhestvom integriruemyh neodnorodnyh raspredelennyh informacionnyh resursov v gibridnoj grid-infrastrukture virtual'nyh observatorij [The Middleware Architecture of the Subject Mediators for Problem Solving over a Set of Integrated Heterogeneous Distributed Information Resources in the Hybrid Grid-Infrastucture of Virtual Observatories]. Informatika i ee primenenija [Informatics and Applications] 2(1):2–34.
4. Apache Hadoop Project. 2015. Available at: http://hadoop.apache.org (accessed November 17, 2015).
5. Capriolo, E., D. Wampler, and J. Rutherglen. 2012. Programming Hive Data Warehouse and Query Language for Hadoop. O'Reilly Media.
6. Burdick, D., M. A. Hernández, H. Ho, G. Koutrika, R. Krishnamurthy, L. Popa, I. Stanoi, S. Vaithyanathan, and S. R. Das. 2011. Extracting, Linking and Integrating Data from Public Sources: A Financial Case Study. IEEE Data Eng. Bull.34(3):60–67.
7. Burdick, D., A. Evfimievski, R. Krishnamurthy, N. Lewis, L. Popa, S. Rickards, and P. Williams. 2014. Financial Analytics from Public Data. Proceedings of the International Workshop on Data Science for Macro-Modeling, SIGMOD/PODS'2014 Conference. P. 1–6.
8. Freeman, L. C. 1977. A Set of Measures of Centrality Based on Betweenness. Sociometry. 40(1): 35–41.
9. Bonacich, P. 1987. Power and Centrality: A Family of Measures. American Journal of Sociology. 92(5):1170–1182.

10. Kalinichenko, L. A., S. A. Stupnikov, and D. O. Martynov. 2007. SYNTHESIS: a Language for Canoni-cal Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Infor-mation Resource Environments. Moscow: IPI RAN, 171 p.
11. The Neo4j Manual. 2015. Available at: http://goo.gl/cHiOGF (accessed November 17, 2015).
12. Cyganiak, R., D. Wood, and M. Lanthaler. (eds.). 2014. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 25 February 2014. Available at: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225 (accessed November 17, 2015).
13. Introducing JSON. 2015. Available at: http://www.json.org/ (accessed November 17, 2015).
14. Stupnikov, S. A. 2014. Otobrazhenie grafovykh modeley dannykh v kanonicheskuyu model' v sistemakh s intensivnym ispol'zovaniem dannykh [Mapping of Graph Data Models into a Canonical Model for the Development of Data Intensive Systems]. Sistemy vysokoy dostupnosti [Systems of High Availability]. Moscow: Radiotechnika. 2:13–31.
15. Vovchenko, A. E. 2012. Rassredotochennaya realizatsiya prilozheniy v srede predmetnykh posrednikov [Distributed Implementation of the Applications in the Subject Mediation Environbment]. PhD Thesis. Moscow: IPI RAN. 216 p.
16. Kalinichenko, L. A., D. O. Martynov, and S. A. Stupnikov. 2004. Query rewriting using views in a typed mediator environment. In: Proc. of the 8th East European Conference on Advances in Databases and Information Systems. LNCS 3255. Berlin-Heidelberg: Springer-Verlag. P. 37–53.
17. Briukhov, D. O., L. A. Kalinichenko, and D. O. Martynov. Source Registration and Query Rewriting Applying LAV/GLAV Techniques in a Typed Subject Mediator. 2007. In: Proc. of the Ninth Russian Conference on Digital Libraries RCDL'2007. Pereslavl-Zalesskij: Pereslavl University. P. 253–262.
18. Beyer, K. S., V. Ercegovac, R. Gemulla, A. Balmin, M. Eltabakh, C.-Ch. Kanne, F. Ozcan, and E. J. Shekita. 2011. Jaql: A Scripting Language for Large Scale Semistructured Data Analysis. Proceedings of the VLDB Endowment. 4(12):1272–1283.

Ступников Сергей Александрович – к.т.н., с.н.с. лаб. Композиционных методов и средств построения информационных систем Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sstupnikov@ipi.ac.ru

Sergey Stupnikov – Candidate of Science (Ph.D.) in theoretical informatics, senior research scientist, Institute of Informatics Problems, Federal Research Center

“Computer Science and Control” of the Russian Academy of Sciences, sstupnikov@ipi.ac.ru

Брюхов Дмитрий Олегович – к.т.н., с.н.с. лаб. Композиционных методов и средств построения информационных систем Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, brd@ipi.ac.ru

Dmitry Briukhov – Candidate of Science (Ph.D.) in technology, senior research scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, brd@ipi.ac.ru

Скворцов Николай Алексеевич – н. с. лаб. Композиционных методов и средств построения информационных систем Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, nskv@ipi.ac.ru

Nikolay Skvortsov – research scientist, Institute of Informatics Problems, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, nskv@ipi.ac.ru

Title: Co-lending Systemic Risk Analysis over Heterogeneous Data Collections

Abstract: The paper considers an approach for co-lending systemic risk analysis problem solving over heterogeneous data collections in a combined virtual and materialized integration environment. The problem belongs to the data intensive domain of financial macromodeling. Virtual integration is implemented using subject mediation technology. Materialized integration is implemented using open source Hadoop software framework for distributed storage and processing of large datasets accompanied by the Hive system intended for relational warehousing over Hadoop.

Keywords: co-lending systemic risk, problem solving, data integration, heterogeneous data collections