

DOI:

## ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ФРАГМЕНТОВ ТАБЛИЦ ДАННЫХ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

<sup>1</sup>Микрин Е.А., <sup>2</sup>Чернов И.В., <sup>2</sup>Сомов С.К.

<sup>1</sup> ПАО «Ракетно-космическая корпорация «Энергия» им. С.П. Королева»,  
г. Королев, ул. Ленина, 4а  
Eugeny.Mikrin@rsce.ru

<sup>2</sup>Институт проблем управления им. В.А. Трапезникова РАН, Россия, г. Москва  
ул. Профсоюзная д.65, стр.1.  
ssomov2016@ipu.ru

*Аннотация:* В работе рассматривается задача повышения производительности распределенных систем обработки данных путем оптимального размещения фрагментов таблиц баз данных в узлах систем. Предложена модель динамического перераспределения фрагментов. Представлен эвристический алгоритм размещения фрагментов в зависимости от текущих параметров трафика запросов к данным, генерируемых в узлах сети. В модели используется стоимостной критерий оптимальности размещения фрагментов в узлах сети.

Ключевые слова: фрагменты таблиц распределенных баз данных, распределенные системы обработки данных, размещение фрагментов в узлах сети.

### Введение

Повышение производительности распределенных систем обработки данных (РСОД) может быть достигнуто различными методами. Одним из таких методов является разбиение таблиц баз данных, используемых в системе, на фрагменты. При этом эффективность работы РСОД в значительной степени будет зависеть от того, насколько эффективен выбранный метод фрагментации таблиц данных и от того, насколько оптимально размещены полученные фрагменты по различным узлам распределенной системы.

В научных сообществах интерес к фрагментации таблиц возник почти сразу после появления практических реализаций компьютерных сетей и построенных на их основе распределенных баз данных (РБД) [1]. В качестве наглядного примера одного из самых первых исследований, посвященных оптимизации распределения файлов данных, можно назвать работу [2]. В этой работе была предложена модель оптимизации распределения файлов в многомашинных комплексах для минимизации эксплуатационных затрат при ограничениях на время обработки запросов и количество копий для каждого файла. В последующие годы рассматривались различные методы для оптимизации работы уже распределенных баз данных. Широко используемым на практике методом оптимизации работы РБД является фрагментация таблиц данных и размещение полученных фрагментов в узлах компьютерной сети. Например, в работе [3] метод фрагментации и распределения фрагментов используется для увеличения производительности РБД. А в работах [4,5] в динамической модели распределения фрагментов таблиц используется критерий минимизации использования каналов связи для передачи данных.

Основная цель использования фрагментации таблиц заключается в уменьшении объема считываемых данных при поиске информации, необходимой для обработки запросов к исходной таблице данных. Использование подмножества атрибутов в записях фрагментов таблиц вместо полного перечня атрибутов записей оригинальной таблицы приводит к сокращению объема данных, которые считываются с устройств хранения данных, сокращению времени на передачу информации по каналам связи, и в итоге к уменьшению времени отклика системы на запросы. В результате оптимальное размещение фрагментов таблиц по узлам системы позволяет сократить затраты на функционирование РСОД.

На практике используется два основных метода фрагментации: горизонтальная и вертикальная фрагментация таблиц. Вертикальная фрагментация – это разделение атрибутов записей исходной таблицы на группы, горизонтальная фрагментация – такое разделение таблицы на подмножества записей этой таблицы, что каждое подмножество содержит записи с полным набором атрибутов записей исходной таблицы.

В данной работе представлен метод динамического перераспределения фрагментов таблиц по узлам распределенной системы, учитывающий текущие значения параметров системы, которые с течением времени могут изменяться.

В частности, принимаются во внимание такие параметры распределенной системы как:

- топология компьютерной сети, на базе которой функционирует система,
- трафик информационных запросов к данным,
- трафик запросов на модификацию данных,
- затраты на передачу информации между узлами сети,
- затраты на хранение фрагментов данных в узлах сети,
- затраты на обработку запросов в узлах сети с необходимыми данными.

Предлагаемая модель динамического перераспределения фрагментов таблиц исходит из того, что фрагментирование таблиц распределенной базы данных (РБД), используемой системой, выполнено. В модели используется неизбыточное размещение фрагментов, при котором каждый фрагмент таблицы размещается только в одном узле сети, т.е. копии (реплики) фрагментов таблиц не используются. При этом известно количество фрагментов, характеристики фрагментов и трафик информационных запросов к каждому из фрагментов и запросов на модификацию фрагментов.

В модели предполагается, что известно первоначальное распределение фрагментов таблиц баз данных, используемых в РСОД, и необходимо выполнить перераспределение этих фрагментов для улучшения эффективности работы системы. При поиске нового распределения фрагментов учитываются изменения значений параметров системы, произошедшие после того, как был выполнен предыдущий процесс распределения фрагментов.

В качестве критерия оптимальности распределения фрагментов используется минимум затрат на функционирование распределенной системы.

## 1. Описание задачи и информационные требования

Предполагается, что РСОД использует распределенную базу данных и функционирует на основе компьютерной сети из  $K$  узлов:  $N = \{N_1, \dots, N_k, \dots, N_K\}$ . Топология сети представлена взвешенным графом  $G = (K, \Gamma)$ . Заданы длины дуг графа.

Предположим, что компьютерная сеть неоднородная и для нее задана матрица  $CoC = \{CoC_{ij}\}$  стоимости  $CoC_{ij}$  передачи единицы данных (стоимость связи) между всеми парами узлов сети  $N_i, N_j$  ( $i, j = \overline{1, K}$ ).

На основе матрицы  $CoC$ , с использованием любого известного алгоритма поиска кратчайших путей в графе, строится матрица  $DTC = \{DTC_{ij}\}$ ,  $i, j = \overline{1, K}$  - матрица затрат на передачу данных по кратчайшим путям между узлами сети (*Data Transfer Cost*).

Каждый узел  $N_k$  сети характеризуется его максимальной емкостью  $C_k$ , т.е. максимально возможным объемом размещенных в узле фрагментов, и ограничением  $FM_k$  на максимальное количество фрагментов, которые могут быть размещены в узле  $k$ .

В результате фрагментирования таблиц РБД получено множество из  $M$  фрагментов  $F = \{F_1, \dots, F_m, \dots, F_M\}$ . Для каждого фрагмента известна его мощность  $Crd(F_m)$ , т.е. количество кортежей во фрагменте, и длина  $Len(F_m)$  его кортежей, равная сумме длин атрибутов кортежа.

Размеры (объемы данных) фрагментов вычисляются по формуле:

$$FV(F_m) = Crd(F_m) * Len(F_m).$$

Длина кортежа  $Len(F_m)$  фрагмента  $m$  вычисляется как произведение:

$$Len(F_m) = \sum_{i=1}^{NA_m} LA_{im},$$

где  $NA_m$  - количество атрибутов в кортеже фрагмента  $F_m$ , а  $LA_{im}$  - длина  $i$ -го атрибута в кортеже этого фрагмента.

Таким образом:

$$FV(F_m) = Crd(F_m) * \sum_{i=1}^{NA_m} LA_{im}.$$

## 2. Параметры и ограничения размещения фрагментов

Размещение фрагментов таблиц по узлам сети описывается матрицей  $Y = \{y_{mk}\}$ , в которой:

$$y_{mk} = \begin{cases} 1, & \text{если фрагмент } F_m \text{ размещен в узле } k, \\ 0, & \text{если фрагмент } F_m \text{ не размещен в узле } k. \end{cases}$$

Матрица  $Y$  описывает как первоначальное размещение фрагментов по узлам сети, так и размещение этих фрагментов после их перераспределения.

При этом в отдельных узлах сети может быть размещен только один экземпляр конкретного фрагмента (используется неизбыточное размещение фрагментов).

В процессе перераспределения фрагментов должны выполняться следующие ограничения:

$$(1) \quad \sum_{m=1}^M y_{mk} * Vol(F_m) \leq C_k, \quad 1 \leq k \leq K.$$

Ограничение (1) означает, что суммарный объем всех фрагментов, размещенных в узле  $k$ , не может превышать емкость  $C_k$  этого узла.

$$(2) \quad \sum_{m=1}^M y_{mk} \leq FM_k, \quad 1 \leq k \leq K.$$

Ограничение (2) гарантирует, что в каждом узле  $k$  хранится фрагментов не более, чем заданное максимальное количество фрагментов  $FM_k$ .

$$(3) \quad \sum_{m=1}^M y_{mk} = 1, \quad 1 \leq k \leq K.$$

Ограничение (3) означает, что каждый фрагмент может быть распределен только в одном из узлов сети.

$$(4) \quad AvRPT \leq T_{max}.$$

Ограничение (4) требует, чтобы найденное распределение фрагментов по узлам обеспечивало среднее время обработки запроса не выше заданного максимального времени.

Предположим, что стоимость хранения фрагментов в узлах сети вычисляется с использованием э кортежа  $DSC(k)$  (data storage cost), элементы которого определяют стоимость хранения единицы данных в узлах сети.

### 3 Параметры запросов и процессов их обработки в узлах сети

В отдельных узлах РСОД выполняется различное (один или несколько) количество прикладных процессов (пользовательских задач), каждый из которых принадлежит одному из  $P$  типов множества всех процессов  $P = \{P_p, p = 1, \dots, P\}$ . Каждый прикладной процесс в ходе работы генерирует в единицу времени некоторое количество информационных запросов (*SELECT*) и запросов на модификацию (*UPDATE*) данных в одном или нескольких фрагментах данных.

В предлагаемой модели сложные запросы, требующие данные из нескольких фрагментов, рассматриваются как набор из нескольких простых запросов, каждый из которых использует данные только одного конкретного фрагмента.

В модели используется несколько матриц, описывающих характеристики запросов, генерируемых в системе прикладными процессами.

Матрица  $PF = \|PF_{kp}\|$  ( $k = \overline{1, K}; p = \overline{1, P}$ ) задает частоту решения процессов в узлах сети. Здесь  $pf_{kp}$  это частота решения  $p$ -го процесса в  $k$ -м узле сети.

Матрица  $PQE = \|PQE_{pm}\|$ , задает значения  $PQE_{pm}$  – частот генерации  $p$ -м процессом информационных запросов к фрагменту  $m$ .

Матрица  $PQU = \|PQU_{pm}\|$  определяет значения частот  $PQU_{pm}$  генерации  $p$ -м процессом запросов на модификацию данных фрагмента  $m$ .

Общее количество  $NQE(k, m)$  информационных запросов к фрагменту  $m$  данных, возникающим в узле  $k$ , равно сумме произведений частоты выполнения всех прикладных процессов, выполняемых в узле  $k$ , на количество генерируемых этими процессами информационных запросов к фрагменту  $m$ .

Аналогично количество  $NQU(k, m)$  запросов на модификацию данных фрагмента  $m$ , возникающих в узле  $k$ , равно сумме произведений частоты выполнения всех прикладных процессов в этом узле на количество генерируемых этими процессами всех запросов на модификацию данных фрагмента  $m$ .

Значения  $NQE(k, m)$  и  $NQU(k, m)$  вычисляются по формулам 5 и 6.

$$(5) \quad NQE(k, m) = \sum_{p=1}^P PF(k, p) * PQE(p, m), \quad k = 1, \dots, K.$$

$$(6) \quad NQU(k, m) = \sum_{p=1}^P PF(k, p) * PQU(p, m), \quad k = 1, \dots, K.$$

Кортеж  $NQ$  состоит из элементов  $NQ(k)$  с общим количеством запросов (как информационных, так и запросов на модификацию данных) к фрагментам данных, генерируемых в отдельных узлах сети. Значения этих элементов подсчитываются суммированием соответствующих элементов матриц  $NQE(k, m)$  и  $NQU(k, m)$ .

Полагаем, что в узлах сети бесконечные очереди на обслуживание запросов, а система работает в установившемся режиме, без сбоев. При этих параметрах сети все запросы обрабатываются успешно.

Предполагаем так же, что обработка всех запросов к данным конкретного фрагмента производится в том же узле сети, в котором размещен сам фрагмент таблицы данных.

В общем случае, стоимостные и временные характеристики процессов обработки запросов в узлах сети значительно отличаются друг от друга. Это обусловлено тем, что время обработки различных запросов в различных узлах сети существенно зависит от вычислительной сложности алгоритмов обработки этих запросов и от параметров сервера конкретного узла сети, обрабатывающего запросы (от вычислительной мощности сервера и от скорости обмена данными с внешними запоминающими устройствами).

Для упрощения модели предположим, что стоимостные и временные характеристики обработки запросов к любому фрагменту данных, генерируемых любым прикладным процессом, зависят только от вычислительной мощности сервера узла сети, в котором производится обработка запросов.

Будем считать, что нам задан вектор (кортеж)  $T^E = (T_1^E, \dots, T_n^E, \dots, T_N^E)$  средних времен обработки информационных запросов в узлах сети, и вектор  $S^E = (S_1^E, \dots, S_n^E, \dots, S_N^E)$  средней стоимости единицы времени обработки информационных запросов в узлах сети.

Аналогично предположим, что задан (кортеж)  $T^U = (T_1^U, \dots, T_n^U, \dots, T_N^U)$  средних времен обработки в узлах сети запросов на модификацию данных фрагментов, и вектор  $S^U = (S_1^U, \dots, S_n^U, \dots, S_N^U)$  средней стоимости единицы времени обработки в узлах сети запросов на модификацию данных.

#### 4 Вес фрагмента

В предлагаемой модели используется параметр  $FW = FW(N_k, F_m)$  - «вес» фрагмента  $F_m$  для узла  $N_k$ . Значение данного параметра зависит от трех компонент:

- сумма  $ACSR(N_k, F_m)$  средних затрат на передачу по каналам сети запросов (average cost of sending requests), сгенерированных за единицу времени в узле  $N_k$  к фрагменту  $F_m$ , размещенному в одном из узлов сети;

- сумма  $ACPR(N_k, F_m)$  средних затрат на обработку этих запросов (average cost of processing requests) в узле сети, в котором размещен фрагмент;

- стоимость  $FSC(N_k, F_m)$  хранения фрагмента  $F_m$  в узле  $N_k$  (fragment storage costs), которая зависит от объема фрагмента, при условии размещения фрагмента в узле  $k$ .

Вес  $FW$  рассчитывается для каждого узла  $N_k$  сети для всех запросов к фрагменту  $F_m$ , которые генерируются в данном узле за единицу времени, в соответствии с формулой:

$$(7) \quad FW(k, m) = ACSR(k, m) + ACPR(k, m) - FSC(k, m), \quad 1 \leq k \leq K, 1 \leq m \leq M.$$

Значения компонент этой формулы рассчитываются следующим образом:

- Сумма  $ACSR(k, m)$  затрат на передачу запросов из узла  $N_k$  в узел  $N_l$  с размещенным в нем фрагментом  $F_m$ . Размещение фрагментов по узлам сети определено в матрице  $Y(m, k)$ .

$$ACSR(k, m) = NQM(k, m) \times \sum_{l=1, l \neq k}^K y_{ml} \times DTC_{kj},$$

здесь  $NQM(k, m)$  сумма всех запросов к фрагменту  $F_m$ , генерируемых в узле  $N_k$  за единицу времени. Значения в матрице  $NQM$  рассчитываются по формуле:

$$NQM(k, m) = \sum_{p=1}^P PF(k, p) \times [(PQE(p, m) + PQU(p, m))]$$

- Сумма  $ACPR(k, m)$  затрат на обработку запросов, сгенерированных в узле  $N_k$ , в узле сети, в котором размещен фрагмент:

$$ACPR(k, m) = \sum_{l=1, l \neq k}^K y_{ml} \times \{(T_l^E \times S_l^E) \times NQE(k, m) + (T_l^U \times S_l^U) \times NQU(k, m)\},$$

здесь:

$NQE(k, m)$  – количество информационных запросов к фрагменту  $m$ , генерируемых в узле  $k$ ,

$NQU(k, m)$  – количество запросов на модификацию данных фрагмента, генерируемых в узле  $k$ .

Фактически вес узла  $FW(N_k, F_m)$  соответствует сумме, на которую изменятся (уменьшатся или увеличатся) затраты на эксплуатацию системы в случае, если фрагмент  $F_m$  будет размещен в узле  $N_k$  вместо его размещения в любом произвольном узле сети.

В процессе размещения фрагментов по узлам сети каждый фрагмент размещается в том узле сети, который для этого фрагмента имеет наибольшее значение веса узла.

Значение веса  $FW$  фрагмента используется также и в том случае, когда из нескольких узлов поступают запросы к данным одного и того же фрагмента  $F_m$ . Чтобы избежать дублирования информации, фрагмент  $F_m$  будет размещен только в одном из этих узлов, в том, который имеет наибольшее значение «веса» фрагмента  $FW(N_k, F_m)$ .

## 5 Затраты на функционирование распределенной системы

Величина затрат на функционирование распределенной системы с размещенными в узлах системы фрагментами таблиц данных складывается из следующих составляющих:

- затраты на хранение фрагментов в узлах сети,
- затраты на передачу запросов из узлов – источников в узлы с фрагментами,
- затраты на обработку запросов в узлах сети с фрагментами таблиц.

### А. 5.1. Затраты на хранение фрагментов

Затраты на хранение фрагментов таблиц в узлах сети определяются на основе данных матрицы стоимости хранения фрагментов в узлах сети  $FSC$  (fragment storage cost) и текущего распределения фрагментов по узлам сети, заданного матрицей  $Y = \{y_{mk}\}$ . Элемент матрицы  $y_{mn} = 1$ , если фрагмент  $F_m$  размещен в узле  $N_k$ , и равен 0 в противном случае:

$$y_{mk} = \begin{cases} 1, & \text{если фрагмент } F_m \text{ размещен в узле } N_k \\ 0, & \text{если фрагмент } F_m \text{ не размещен в узле } k \end{cases}$$

Тогда затраты  $TSTF$  (total fragments storing cost) на хранение всех фрагментов таблиц, распределенных по узлам системы в соответствии с матрицей  $Y$ , с учетом таблицы  $FSC$  стоимости хранения фрагментов в отдельных узлах сети, рассчитываются по следующей формуле:

$$(8) \quad TSTF = \sum_{m=1}^M \sum_{k=1}^K y_{mk} \times FSC(k, m)$$

Здесь  $FSC(m, k)$  – (fragment storage cost) затраты на хранение одного фрагмента  $F_m$  при условии, что он размещен в узле  $N_k$ .

### 5.2. Затраты на передачу запросов по каналам связи

Величина  $CTR$  (Cost of Transferring Requests) затрат на передачу всех запросов, генерируемых во всех узлах сети за единицу времени, для текущего размещения фрагментов  $Y$  рассчитывается на основе данных матрицы  $DTC$  (Data transfer costs) затрат на передачу единицы данных по кратчайшим путям между узлом-источником запросов и узлом-получателем, в котором обрабатываются запросы к фрагменту данных. Для расчета рассматриваемых затрат также необходимо использовать матрицу  $NQ$  частот генерации прикладными процессами всех запросов (информационных и запросов на модификацию данных) в узлах сети.

Для упрощения модели предположим, что размер всех запросов одинаков и равен одной единице информации.

Величина затрат на передачу одного запроса, возникшего в некотором узле  $N_k$  к фрагменту  $F_m$ , размещенному в узле  $N_j$ , равна значению элемента  $DTC(k,j)$  матрицы  $DTC$  затрат на передачу информации по кратчайшему пути между этими узлами.

Таким образом затраты на передачу всех запросов, генерируемых во всех узлах системы за единицу времени будут равны:

$$(9) \quad CTR = \sum_{k=1}^K \sum_{p=1}^P PF(k,p) \sum_{m=1}^M (PQE(p,m) + PQU(p,m)) \times \sum_{j=1}^K y_{mj} DTC(k,j)$$

### 5.3. Затраты на обработку запросов в узлах с фрагментами данных

Величина  $RPC$  (*Request Processing Cost*) затрат на обработку всех запросов, генерируемых в узлах сети за единицу времени, зависит от следующих данных:

- трафика информационных запросов к фрагментам таблиц;
- трафика запросов на модификацию данных фрагментов;
- текущего размещения фрагментов по узлам сети;
- стоимости обработки запросов узлах сети с фрагментами.

Так как в рассматриваемой модели используется неизбыточное размещение фрагментов, то только один экземпляр фрагмента размещается в одном из узлов сети. Поэтому все запросы к фрагменту  $F_m$  из узла  $N_j$  пересылаются для обработки по кратчайшему пути в узел  $N_k$ , в котором размещен фрагмент данных  $F_m$ , то есть запросы адресуются в узел, для которого справедливо  $y_{mk} = 1$ .

При этом надо учесть, что в соответствии с рассматриваемой моделью, информационные запросы и запросы на модификацию данных имеют разные значения стоимости их обработки в разных узлах. Величины стоимостных затрат на обработку запросов в узлах сети представлены в кортежах  $S^E$  для информационных запросов и в  $S^U$  для запросов на модификацию данных.

Величина затрат на обработку запросов, генерируемых во всех узлах системы за единицу времени, рассчитывается по формуле:

$$(10) \quad RPC = \sum_{j=1}^K \sum_{p=1}^P PF(j,p) \left( \sum_{m=1}^M PQE(p,m) \sum_{k=1}^K y_{mk} S^E(k,m) + \sum_{m=1}^M PQU(p,m) \sum_{k=1}^K y_{mk} S^U(k,m) \right)$$

Подсчет величины  $RPC$  в формуле (10) ведется по следующему алгоритму. Для каждого узла  $N_j$  определяется из матрицы  $PF(j,p)$  частота выполняемых в этом узле прикладных процессов, для каждого такого процесса  $P_p$  определяется количество генерируемых этим процессом информационных запросов (из матрицы  $PQE$ ) и запросов на модификацию данных (из матрицы  $PQU$ ) ко всем фрагментам данных системы. И для каждого запроса определяется стоимость его обработки на основе данных из кортежа  $S^E$  или  $S^U$ , соответственно для информационных запросов и запросов на модификацию данных.

### 5.4. Общие затраты на функционирование системы

Таким образом, величина  $SOC$  (*system operation costs*) затрат на функционирование системы будет равна:

$$(11) \quad SOC = TSTF + CTR + RPC$$

В формуле (11) используются следующие слагаемые:

- $TSTF$  – затраты на хранение всех фрагментов таблиц, распределенных по узлам системы в соответствии с матрицей  $Y$ .
- $CTR$  - затраты на передачу всех запросов, генерируемых во всех узлах сети за единицу времени, для текущего размещения фрагментов.
- $RPC$  – затраты на обработку всех запросов, генерируемых в узлах сети за единицу времени.

## 6 Среднее время обработки запроса в системе

Так как в рассматриваемой модели предполагается, что система работает в установившемся режиме, очереди на обслуживание бесконечны и все запросы обрабатываются, то среднее время обработки запроса в системе  $AvRPT$  - (average request processing time) рассчитывается следующим образом.

Сначала подсчитывается общее количество всех запросов, генерируемых во всех узлах системы за единицу времени. Затем подсчитывается суммарное время обработки в системе данного количества запросов. Затем вычисляется среднее время обработки одного запроса.

Общее количество  $ToNR$  (*total number of requests*) всех запросов, генерируемых в системе за единицу времени подсчитывается по формуле (12) и ведется по следующему алгоритму. Для каждого узла  $N_k$  определяется из матрицы  $PF(k, p)$  частота выполняемых в этом узле прикладных процессов, Для каждого такого процесса  $P_p$  определяется сумма количества генерируемых этим процессом информационных запросов (из матрицы  $PQE$ ) и запросов на модификацию данных (из матрицы  $PQU$ ) ко всем фрагментам данных системы.

$$(12) ToNR = \sum_{k=1}^K \sum_{p=1}^P PF(k, p) \times \sum_{m=1}^M [PQE(p, m) + PQU(p, m)].$$

Суммарное время обработки в системе всех запросов  $ToRPT$  (*total request processing time*) подсчитывается по формуле (13) и ведется примерно по такому же алгоритму, который используется для подсчета общего количества запросов. Отличие в алгоритмах заключается в том, что для каждого запроса определяется среднее время его обработки на основе данных из кортежа  $T^E$  или  $T^U$ , соответственно для информационных запросов и запросов на модификацию данных.

$$(13) ToRPT = \sum_{k=1}^K \sum_{p=1}^P PF(k, p) \left( \sum_{m=1}^M PQE(p, m) \sum_{j=1}^K y_{mj} T^E(j, m) + \sum_{m=1}^M PQU(p, m) \sum_{j=1}^K y_{mj} T^U(j, m) \right)$$

Таким образом, среднее время обработки одного запроса в системе с учетом формул (12) и (13) равно:

$$(14) AvRPT = ToRPT / ToNR.$$

## 7. Формулировка и алгоритм решения задачи оптимального размещения фрагментов

### 7.1. Формулировка задачи

Согласно рассматриваемой модели и принятыми ограничениями задача оптимального размещения фрагментов таблиц по узлам распределенной системы имеет следующую формулировку.

Необходимо найти такое распределение фрагментов  $Y$  по узлам распределенной системы, которое обеспечивает минимум затрат  $SOC$  на функционирование системы:

$$\min_Y SOC = \min_Y (TSTF + CTR + RPC)$$

при выполнении ограничений (1)-(4):

- на суммарный объем всех фрагментов, размещенных в одном узле (1),
- на количество фрагментов в одном узле (2),
- каждый фрагмент может быть распределен только в одном узле (3),
- на среднее время обработки запроса (4).

При этом известны следующие исходные данные задачи:

- топология компьютерной сети,
- количество и характеристики фрагментов таблиц данных,
- стоимость передачи единицы данных по каналам связи сети,
- стоимость хранения единицы данных в узлах сети,
- частоты выполнения прикладных процессов в узлах сети,
- частоты генерации прикладными процессами информационных запросов и запросов на модификацию данных фрагментов таблиц,
- стоимость обработки запросов в узлах сети с фрагментами таблиц.

Так как в соответствии с данной моделью допускается, что параметры распределенной системы динамичны, и со временем они могут меняться (например, изменяется трафик запросов к фрагментам,

генерируемый в отдельных узлах сети), то данная задача должна решаться периодически. Задача также может решаться и в произвольный момент, если произошло событие, очевидно приводящее к изменениям в параметрах работы системы. Например, если в состав сети включено или наоборот исключено несколько узлов сети, или в рамках РБД создана новая таблица данных и она разбита на несколько новых фрагментов, или появился новый канал связи между узлами и т.д.

## 7.2. Алгоритм решения задачи

В случае распределения большого количества фрагментов таблиц РБД в крупномасштабной РСОД с большим количеством узлов задача поиска оптимального распределения фрагментов будет иметь большую вычислительную сложность и потребует слишком больших затрат времени.

По причине большой вычислительной сложности задачи для ее решения предлагается использовать эвристический алгоритм. Данный алгоритм запускается с некоторой заданной периодичностью либо в любой момент, когда параметры системы меняются таким образом, что эти изменения очевидно требуют перераспределения используемых в системе фрагментов таблиц (например, появляется новый канал связи либо добавляется новый узел в компьютерной сети).

Описание эвристического алгоритма:

Шаг 0. Присваиваем переменной  $SOC$  максимально возможное целое число.

Шаг 1. Вычисляем элементы кортежа  $FQ(m)$  с частотами запросов к фрагментам. Элемент  $m$  кортежа это сумма частот всех запросов к фрагменту  $F_m$ , генерируемых во всех отдельных узлах сети.

Присваиваем значение 0 всем элементам кортежа  $FQ$ .

В цикле для каждого фрагмента  $F_m$  с номерами  $m$  от 1 до  $M$  подсчитываем:

для каждого узла  $N_k$  сети (для  $k$  от 1 до  $K$ ),

и для всех прикладных процессов  $P_p$ , выполняемых в этом узле (данные матрицы  $PF$ ),

- вычисляем сумму  $SQ(p,k,m)$  частот генерации процессом  $p$  в узле  $k$  информационных запросов к фрагменту  $F_m$  (данные из матрицы  $PQE$ ) и частот генерации запросов на модификацию этого фрагмента (данные из матрицы  $PQU$ ),
- полученную сумму  $SQ(p,k,m)$  умножаем на частоту  $PF(k,p)$  выполнения процесса  $p$  в узле  $k$  (данные матрицы  $PF$ ),
- результат прибавляем к  $FQ(m)$ .

Конец цикла.

В итоге мы имеем кортеж  $FQ(m)$  элемент  $m$  которого равен сумме частот всех запросов к фрагменту  $F_m$ , генерируемых в системе всеми прикладными процессами во всех узлах сети.

Шаг 2. Заполняем элементы кортежа  $FDS$  (fragment distribution sequence) номерами фрагментов, отсортированных в порядке убывания количества запросов к фрагменту. Кортеж  $FDS$ , который будет определять очередность распределения фрагментов по узлам сети, формируем следующим образом.

Присваиваем значение 0 всем элементам кортежа  $FDS$ .

В цикле по  $fn$  от 1 до  $M$  заполняем кортеж  $FDS$ :

- В цикле по всем элементам кортежа  $FQ(m)$  ищем элемент с максимальным значением суммы частот запросов.
- Запоминаем его номер в элементе  $FDS(fn)$  кортежа.
- Удаляем этот элемент из кортежа  $FQ(m)$ , тем самым уменьшая количество его элементов.

Конец цикла по номерам фрагментов  $fn$ .

Получили кортеж  $FDS$ , заполненный номерами фрагментов, упорядоченных в порядке убывания сумм частот запросов к данному фрагменту.

Далее на шаге 4 в этом порядке будет производиться распределение фрагментов по узлам сети.

Шаг 3. Вычисляем для всех фрагментов и для всех узлов сети значения элементов матрицы  $FW$  с весами фрагментов по формуле (7).

В цикле для каждого узла  $N_k$  сети (для  $k$  от 1 до  $K$ )

и для каждого фрагмента  $F_m$  с номерами  $m$  от 1 до  $M$

рассчитываем значение элемента матрицы  $FW(k,m)$  веса фрагмента  $m$  для узла  $k$ .

Конец цикла по  $k$  и по  $m$ .

По завершению цикла имеем матрицу  $FW$  с весами фрагментов для каждого узла сети.



Шаг 4. Распределяем фрагменты  $F_m$  по узлам сети последовательно, в цикле по отсортированным номерам фрагментов, сохраненным в кортеже  $FDS$ .

В цикле N1 по номерам фрагментов  $m$  от 1 до  $M$  выбираем номер очередного фрагмента из кортежа  $FDS(m)$ , и присваиваем этот номер переменной  $nm$ . Получили в переменной  $nm$  номер очередного фрагмента для которого будет произведен поиск поиска узла сети – кандидата для размещения в нем фрагмента с номером в  $nm$ .

Шаг 4.1. В цикле N2 по узлам сети (для  $k$  от 1 до  $K$ ) ищем в элементах  $nm$ -го столбца матрицы  $FW(k, nm)$  элемент столбца с наибольшим значением веса фрагмента с номером из  $nm$ . Запоминаем в переменной  $nk$  номер строки столбца найденного элемента столбца (равный номеру узла сети) с наибольшим значением соответствующего элемента  $FW(k, nm)$ .

Конец цикла N2 поиска узла с наибольшим весом текущего фрагмента.

В итоге выбран узел с номером  $nk$  – как кандидат для размещения в нем фрагмента с номером  $nm$ . Присваиваем  $Y(nm, nk)=1$ .

Шаг 4.2. Проверка выполнения ограничений задачи при размещении фрагмента  $nm$  в узле  $nk$ .

- Условие (1): суммарный объем всех фрагментов, размещенных в узле  $nk$ , не превышает емкость  $C_{nk}$  этого узла.

- Условие (2): количество фрагментов, размещенных в узле  $nk$ , не превышает максимальное количество  $FM_{nk}$ .

- Условие (3): фрагмент  $nm$  размещен в одном экземпляре в узле сети  $nk$ .

Если хотя бы одно из ограничений не выполняется, то

Присваиваем  $Y(nm, nk)=0$ .

Обнуляем вес фрагмента  $nm$  для узла  $nk$  для того, чтобы этот узел больше не рассматривался в цикле N2 поиска узла – кандидата для размещения фрагмента  $nm$ :

Присваиваем  $FW(nk, nm) = 0$ .

Возврат на Шаг 4.1. к проверке возможности размещения фрагмента в другом узле.

Если ограничения выполняются, то проверяем, приводит ли данное размещение текущего фрагмента  $nm$  к улучшению (уменьшению) значения целевой функции.

Для этого:

Рассчитываем сумму  $L\_SOC$  затрат на функционирование системы при текущем распределении  $Y$  фрагментов в узлах сети по формуле (12):

$$L\_SOC = TSTF + CTR + RPC$$

Если  $L\_SOC < SOC$ , то

Значение целевой функции улучшается.

Возврат на Шаг 4.1.

иначе

Значение целевой функции не улучшается.

Присваиваем  $Y(nm, nk)=0$ .

Возврат на Шаг 4.1.

Конец цикла N2 (Шаг 4.1).

Возврат в Шаг 4. Переходим к размещению очередного фрагмента.

Конец цикла N 1 (Шаг 4.)

Процесс распределения фрагментов закончен.

Шаг 5. Проверяем то, что все  $M$  фрагментов размещены по узлам сети.

Так как в рассматриваемой модели каждый из  $M$  фрагментов может и должен быть размещен только в одном узле сети, то должно выполняться условие:

$$(15) \quad \sum_{k=1}^K \sum_{m=1}^M y_{mk} = M$$

Если условие (15) не выполнено, то при заданных параметрах распределение фрагментов не найдено и необходима корректировка параметров или ограничений задачи и ее повторное решение.

Шаг 6. Для полученного распределения рассчитываем по формуле (14) значение  $AvRPT$  среднего времени ответа на запрос и проверяем выполнение ограничения (4):

$$AvRPT \leq Tmax.$$

Если ограничение не выполняется, то необходимо повторное решение после корректировки параметров задачи или данного ограничения.

Данный алгоритм реализован на языке C++ в среде MS Visual Studio 2017.

## Заключение

Предложена формальная модель распределенной системы, использующей РБД с фрагментами таблиц данных. При этом подразумевается, что ряд параметров распределенной системы динамичны и со временем меняют свои значения. Сформулирована задача поиска оптимального перераспределения фрагментов с тем, что бы достигалось оптимальное значение суммарных затрат на функционирование системы. Учитываются затраты на хранение фрагментов таблиц в узлах системы, затраты на передачу запросов по каналам связи и затраты на обработку запросов в узлах сети. В силу большой вычислительной сложности сформулированной задачи для ее решения эвристический алгоритм решения задачи оптимального и избыточного размещения фрагментов таблиц, использующий в качестве критерия оптимизации минимум затрат на эксплуатацию распределенной системы. При решении задачи используются ограничения на объем данных, размещенных в узлах сети и на среднее время обработки запроса к фрагменту таблицы.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта 18-29-16151 «Разработка методов управления процессами трансформации права в условиях цифровой технологии».

## Литература

1. *Чернышев Г.А.* Обзор подходов к организации физического уровня в СУБД // Труды СПИИРАН. 2013. – Санкт-Петербург, 2013. Вып. 1(24). – с. 222 – 275.
2. *Chu W.W.* File Allocation in a Multiple Computer System // IEEE Transactions on Computers. – 1969. – Vol. C– 18, – № 10. – p. 885-889.
3. *Tamhankar A and Ram S.* Database Fragmentation and Allocation: An Integrated Methodology and Case Study. IEEE Trans. Systems, Man and Cybernetics—Part A, vol. 28, № 3, 1998.
4. *Tambulea L., Horvat. M.* Dynamic Distribution Model in Distributed Database//Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III, 2008, Suppl. issue: Proceedings of ICCCC 2008, pp. 512-515.
5. *Abdalla H.I.* A New Data Re-Allocation Model for Distributed Database Systems// International Journal of Database Theory and Application. 2012. V.5, №.2, pp.45-60.