УДК 519.163

Васянин В.А., к.т.н, с.н.с.,
Институт телекоммуникаций и глобального информационного пространства НАН Украины, г. Киев

СТРУКТУРЫ ДАННЫХ В ЗАДАЧАХ ОПТИМИЗАЦИИ УПАКОВОК МЕЛКОПАРТИОННЫХ ПОТОКОВ В КОММУНИКАЦИОННЫХ СЕТЯХ

Введение. Задачи оптимизации упаковок находят широкое применение при проектировании и анализе функционирования многопродуктовых сетей с мелкопартионными дискретными потоками, передаваемыми по сети в некоторых транспортных блоках заданного объема [1, 2]. При этом объемы отдельно взятых потоков, выраженные количеством единиц потока из узлов источников в узлы назначения, значительно меньше объема транспортного блока. Задача оптимизации упаковок заключается в слиянии (объединении) нескольких исходящих из каждого узла потоков с разными адресами назначения в общие транспортные блоки. При решении задачи некоторые потоки могут несколько раз сливаться в разных узлах с другими потоками, поэтому необходимо иметь такую структуру данных, которая позволяла бы эффективно запоминать и определять узлы слияния и слитые потоки для всех корреспондирующихся пар.

Постановка задачи. Пусть задана неориентированная сеть $G(N,P)$ с множеством узлов N, $n=|N|$ и множеством ребер P, $p=|P|$. На сети задана целочисленная матрица потоков $A=\|a_{ij}\|_{n\times n}$, $a_{ii}=0$, $i=\overline{1,n}$, которые подлежат единовременной передаче из источников i в стоки j, $i,j=\overline{1,n}$. Потоки должны передаваться по сети в транспортных блоках объема $\omega >> a_{ij}$. При этом предполагается, что каждый поток может быть упакован в транспортный блок только целиком. При решении оптимизационной задачи над потоками a_{ij} итеративно выполняются следующие преобразования:

$$a_{ik}^{'} \leftarrow a_{ik}+a_{ij};\ a_{kj}^{'} \leftarrow a_{kj}+a_{ij};\ c_{ij} \leftarrow k,\ \ a_{ij} \leftarrow 0 \quad (1)$$

где c_{ij} – элементы справочной матрицы слияния потоков $C=\|c_{ij}\|_{n\times n}$, которые определяются так:

$$c_{ij}=\begin{cases} k, \text{если поток } a_{ij} \text{ сливается с потоком } a_{ik}, \\ i, \text{если поток } a_{ij} \text{ непосредственно направляется в узел } j, \\ 0, \text{если } i=j. \end{cases} \quad (2)$$

На различных шагах работы алгоритма оптимизации необходимо определять последовательность $\Omega_{ij}=\{(i,k_1),(k_1,k_2),...,(k_m,j)\}$ с промежуточными узлами $\{k_1,k_2,...,k_m\}$, в которых выполняется дополнительная обработка каждого из потоков a_{ij} и их общее число $\nu_{ij}=|\{k_1,k_2,...,k_m\}|$.

Основные результаты. Доказаны следующие утверждения [3].

Теорема 1. Для любого $a_{ij}\in\{a_{ij}\,|\,c_{ij}\neq i\wedge c_{ij}\neq 0\}$ построение матрицы C в соответствии с (2) позволяет найти последовательность Ω_{ij} и установить число дополнительных обработок потока a_{ij}. Показано, что определение Ω_{ij} сводится к построению и последующему прохождению в глубину бинарного дерева с корнем в вершине (i,j).

Лемма. Листья бинарного дерева, построенного прохождением в глубину с использованием элементов матрицы C, порождают последовательность Ω_{ij} при их просмотре слева направо. Доказательство леммы следует из правил построения бинарного дерева и того факта, что для каждого листа дерева соответствующий $c_{ij}=i$. Последовательность листьев c_{ik_1}, $c_{k_1k_2}$,…, $c_{k_{m-1}k_m}$, c_{k_mj} дает искомую последовательность $\Omega_{ij}=\{(i,k_1),(k_1,k_2),...,(k_m,j)\}$.

На основании теоремы 1 и леммы построен эффективный алгоритм построения и прохождения бинарного дерева, в котором поддеревья генерируются в порядке их просмотра.

При работе со справочной матрицей слияния потоков в основном алгоритме оптимизации при выборе очередного потока a_{ij} для преобразования (1) может потребоваться нахождение множества других потоков $\{a_{ij}^{*}\}$,которые уже были слиты с потоком a_{ij} на предыдущих итерациях, так как преобразование a_{ij} приводит кувеличению времени доставки потока адресату не только для a_{ij}, но и для множества $\{a_{ij}^{*}\}$. Оказывается, что для определения множества $\{a_{ij}^{*}\}$ для любого a_{ij}, достаточно воспользоваться справочной матрицей C,

формируемой согласно (2).

Теорема 2. Для любого потока a_{ij}, над которым выполняется преобразование (1), множество $\{a_{ij}^*\}$ может быть определено из справочной матрицы C. Показано, что задача определения $\{a_{ij}^*\}$ эквивалентна построению некоторого дерева поиска ипрохождению его снизу вверх. Любой поток a_{ij} может быть представлен в дереве поиска самым нижним подкорнем (i, j) с двумя листьями: левым - (i,k) и правым - (k, j) ,где k - узел, через который выполняется преобразование потока a_{ij}. Из построения C ясно, что для первых потоков, охватывающих a_{ij}, будутвыполняться условия:

$$c_{\xi j} = i, \quad \xi = \overline{1,n}, \qquad (3)$$

$$c_{i\xi} = j, \quad \xi = \overline{1,n}, \qquad (4)$$

причем (3) означает охват слева, а (4) - охват справа. Для дерева поиска выполнение (3) означает, что для a_{ij} найден левый подкорень, находящийся на уровень выше, а выполнение (4), - что найден правый верхний подкорень. Поскольку в (3) и (4) $\xi = \overline{1,n}$, то может найтись несколько левых и правых подкорней, находящихся на одном уровне дерева. Это говорит о том, что дерево, которое строится, не является бинарным. Для каждого вновь полученного подкорня процесс построения новых подкорней продолжается до тех пор, пока будут выполняться условия (3) или (4). Число дополнительных обработок, определяющее время t_{ij} доставки любого потока, представленного подкорнем в дереве поиска, может быть получено с использованием алгоритма построения и прохождения бинарного дерева в глубину. Таким образом, как только найден очередной подкорень (k,l) ,для него необходимо построить бинарное дерево, определяющее последовательность Ω_{kl} ,исходя из которой для соответствующего потока a_{kl} может быть рассчитано время доставки. На основании доказанных утверждений построен рекурсивный алгоритм, осуществляющий построение дерева исчерпывающего поиска для определения множества $\{a_{ij}^*\}$. В алгоритме на всех уровнях рекурсии и для каждого вновь найденного подкорня выполняется исчерпывающий поиск новых подкорней, для которых справедливо (3)или (4). Это означает, что для любого потока a_{ij} алгоритм корректно строит множество потоков $\{a_{ij}^*\}$, которыебыли слиты с потоком a_{ij} на предыдущих итерациях основного алгоритма оптимизации.

Выводы. На основании доказанных утверждений предложены способ построения справочной матрицы слияния потоков и эффективные алгоритмы, позволяющие определять узлы слияния и слитые потоки для всех корреспондирующихся пар в многопродуктовой сети с дискретными мелкопартионными потоками. Разработанные алгоритмы могут быть использованы при решении различных задач оптимизации упаковок на транспортных сетях с мелкопартионными потоками грузов, а также при проектировании и анализе перспективных опорных сетей передачи данных в виртуальных контейнерах (таких, как Европейская опорная сеть E – bone).

Литература

1. Васянин В.А., Трофимчук А.Н. Задача выбора иерархической структуры многопродуктовой коммуникационной сети с мелкопартионными дискретными потоками // Екологічна безпека та природокористування: Зб. наук. праць. – Київ, 2012. – Вип. 10. – С. 182-204.

2. Васянин В.А. Обобщенная задача упаковки и распределения мелкопартионных потоков в многопродуктовых иерархических сетях и ее последовательная декомпозиция / В.А. Васянин // Екологічна безпека та природокористування: Зб. наук. праць. – Київ, 2012. – Вип. 11. – С. 136-154.

3. Васянин В.А. Справочная матрица слияния потоков в задачах оптимизации упаковок на многопродуктовых сетях / В.А. Васянин // Системні дослідження та інформаційні технології. – 2014. – № 2. – С.