

В. В. Панов, В. А. Балыбердин
(3-й ЦНИИ общетехнических и системных исследований, Москва)

ОПТИМИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ СИСТЕМ С РАСПРЕДЕЛЕННОЙ ОБРАБОТКОЙ ДАННЫХ

The work deals with an approach to the distributed data systems optimization. It is based on the principle of computational locality. The optimization problem is formulated by means of a sequence discrete pseudoboolean programming tasks.

В последние годы с развитием технических и программных средств, обеспечивающих работу в локальных и территориальных вычислительных сетях, появилась реальная возможность практической реализации концепции распределенной обработки данных для широкого круга информационных систем. Системы с распределенной обработкой данных (СРОД), позволяя достаточно гибко обеспечивать требования по оперативности, надежности, живучести и т.п. системы управления, являются идеальным средством для реализации различного рода специализированных АСУ, в которых выполнимость подобных требований жестко определена. Применительно к системам подобного рода и ведется дальнейшее изложение.

Вместе с тем, для успешной реализации СРОД необходимо решение ряда сложных научно-технических проблем. Важное место среди них занимает проблема обеспечения эффективного функционирования СРОД. Последнее обстоятельство объясняется тем, что характеристики функционирования СРОД в целом определяются не только характеристиками составляющих их элементов, но и во многом организацией информационного взаимодействия элементов в процессе работы СРОД. Имеющийся опыт свидетельствует о том, что во многих ситуациях за счет оптимизации функционирования СРОД ее производительность может быть повышена на порядок и более.

Однако для того, чтобы это стало возможным, необходимо иметь формальный аппарат, позволяющий получать рекомендации по оптимизации СРОД в практических ситуациях. В настоящей работе на базе обобщения и развития имеющихся результатов предлагается методологический подход, обеспечивающий возможности оптимизации СРОД для достаточно широкого круга практических применений.

Формализация описания СРОД. Функционирование СРОД можно представить в виде совокупности взаимодействий должностных лиц с системой. Любое такое взаимодействие может быть отображено в виде последовательности этапов передачи и обработки информации.

Определение. Будем называть последовательность этапов передачи и обработки информации на средствах СРОД, инициируемую при реализации заявки должностного лица на выполнение запрашиваемых от сети информационно-вычислительных работ (ИВР), информационным процессом.

Количественные характеристики ИП могут быть получены на основе характеристик составляющих его этапов. Пусть $M_i = \{i\}$ — множество ИП; $M_j = \{j\}$ — множество ресурсов системы. Тогда отдельный этап будет идентифицироваться как A_{ij} , $i = \overline{1, J}$; $j = \overline{1, J}$. При этом полагается, что каждый этап связан лишь с одним ресурсом — критическим, т. е. таким, учет которого

является существенным при совместной реализации ИП. Влияние некритических ресурсов учитывается косвенно, например, путем задания предельного числа одновременно реализуемых на критическом ресурсе ИП.

Каждый этап A_{ij} определяется временем использования j -го ресурса B_j и набором вероятностей обращения к другим ресурсам системы (выхода на последующие этапы) после завершения A_{ij}

$$\theta_{ij} = \{\theta_{ij}\}, k = \overline{1, J}. \quad \text{При этом } \sum_k \theta_{ijk} = 1.$$

Можно видеть, что показатели качества СРОД определяются характеристиками ИП и их совокупностей, а последние зависят от множества принятых решений по построению и организации функционирования СРОД [1]. Отсюда следует, что метод ИП позволяет проводить оценку указанных решений и поиск наиболее рациональных из них.

Общий подход к оптимизации СРОД. В основу предлагаемого подхода к оптимизации СРОД положен принцип локализации вычислений, который может быть сформулирован в виде двух постулатов:

1) система работает эффективно, если она выполняет требуемые функции в заданных условиях без излишних затрат ресурсов;

2) общие затраты ресурсов на выполнение требуемых функций можно уменьшить, если часто выполняемые действия будут потреблять мало ресурсов.

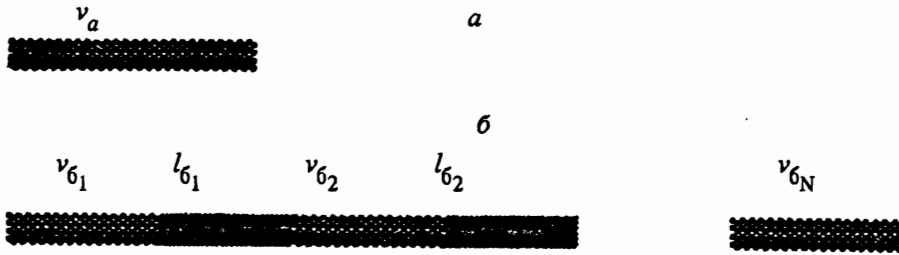
В терминах метода ИП принцип локализации означает, что его реализация позволяет построить такую совокупность ИП, в которой часто реализуемые ИП приобретают простую структуру с малым числом этапов. В результате затраты ресурсов на выполнение всей совокупности ИП уменьшаются.

На базе принципа локализации вычислений строится некоторая совокупность оптимизационных задач в соответствии со спецификой конкретной СРОД. Подобные задачи формулируются в терминах математического программирования. Общая их формулировка может быть записана следующим образом:

$$\begin{aligned} W(x_1, x_2, \dots, x_N) &\rightarrow \min \\ &\{x_i\} \\ a_i(x_1, x_2, \dots, x_N) &\leq A_i, \quad i = \overline{1, R}, \\ b_j(x_1, x_2, \dots, x_N) &\geq B_j, \quad j = \overline{R+1, M}. \end{aligned} \quad (1)$$

Здесь $W(x_1, x_2, \dots, x_N)$ определяет затраты ресурсов на реализацию совокупности ИП, выражаемые объемом "накладных расходов" на передачу информации в процессе выполнения заданного объема ИВР; $a_i(x_1, x_2, \dots, x_N)$ — выражения требований к СРОД; $b_j(x_1, x_2, \dots, x_N)$ — выражения, обеспечивающие учет реальных условий функционирования СРОД (ограничения по памяти, быстродействию и т. п.). Переменные x_1, x_2, \dots, x_N являются булевыми и определяют распределение информации в СРОД.

Обоснуем структуру целевой функции в задаче (1). Пусть мы решаем в системе некоторую задачу. Допустим, что все исходные данные, промежуточные и конечные результаты размещаются в памяти одной ЭВМ (в одном узле), из которого и генерируется обращение к задаче (пользователь находится в том же узле). Задача характеризуется некоторым объемом вычислений v_a . Соответствующий решению такой задачи ИП будет одноэтапным (рисунок, а).



Варианты реализации ИП (объяснение в тексте).

Допустим теперь, что часть данных, необходимых для решения задачи, хранится в других узлах СРОД. Для реализации требуемых передач информации необходимо выполнить ряд процедур, определяемых соответствующими протоколами. В результате количество этапов и общая продолжительность ИП значительно увеличиваются. При этом продолжительность ИП возрастает как за счет увеличения объема вычислений ($\sum V_{b_i} > V_a$), что связано с дополни-

тельными затратами на организацию передачи информации, так и за счет собственно периодов передачи информации l_{b_i} . Это хорошо видно на рисунке, б. Следовательно, в данном случае возникают дополнительные затраты ресурсов. Величина этих затрат зависит от объема передаваемой информации. Последний, в свою очередь, определяется распределением информации (программ и данных) в СРОД. Различные варианты распределения информации в системе требуют разных затрат ресурсов. Мерой же таких затрат может служить объем взаимообмена информацией, возможно, с учетом типа взаимообмена, поскольку эта величина характеризует дополнительные затраты в сравнении с неким идеальным вариантом, осуществляемым в отсутствие взаимообмена. Отсюда становится ясной структура целевой функции в (1).

Оптимизация СРОД на базе ЛВС. Конкретная формулировка оптимизационных задач определяется тем, к какому классу систем относится исследуемая.

Для определенности рассмотрим СРОД, построенную на базе ЛВС с моноканалом с клиент-серверной моделью функционирования. Рассмотрим вначале случай, когда серверы в ЛВС используются, главным образом, в качестве файл-серверов, т. е. для хранения информации. Будем называть информационным объектом (ИО) некоторую совокупность элементов информации, рассматриваемую в процессе осуществления информационных взаимодействий в сети как единое целое: массив данных программа.

Пусть λ_{ij} — интенсивность запросов на выполнение i -го ИО, возникающих в узле j , где $j = \overline{1, J}$ и j соответствует номеру рабочей станции, а i — номеру ИО. Очевидно, λ_{ij} можно полагать равным 0 для массивов данных и программ, обращение к которым возможно лишь из других программ. Так что $\lambda_{ij} \neq 0$, если к i -му ИО, являющемуся программой, поступают непосредственные обращения с рабочих мест пользователей. Такой ИО является головной программой. В процессе выполнения ИО, являющегося программой, возможны обращения к другим ИО - программам, массивам данных. Такие обращения будем задавать с помощью матрицы $A = \{a_{ik}\}$, где

$$a_{ik} = \begin{cases} 1, & \text{если при выполнении } i\text{-го ИО происходит обращение к } k\text{-му ИО;} \\ 0, & \text{в противном случае.} \end{cases}$$

Для удобства дальнейших построений целесообразно положить $a_{ij} = 1$.

Предположим, что корректировка ИО осуществляется средствами общего программного обеспечения, а интенсивность корректировок i -го ИО из узла j обозначается как λ'_{ij} . Для размещения ИО в системе можно использовать дисковую память файл-серверов объемом V_ϕ для каждого файл-сервера, а также дисковую память РС, для чего в узле j выделяется участок памяти емкостью V_j .

Распределение ИО для хранения в ЛВС можно осуществлять различными способами. При этом эффективность функционирования ЛВС может существенно меняться. Основываясь на принципе локализации вычислений, в качестве характеристики затрат ресурсов для различных вариантов распределения ИО будем использовать суммарный удельный объем взаимодействия информацией в сети. Введем переменную x_{kj} , определяемую следующим образом:

$$x_{kj} = \begin{cases} 1, & \text{если } k\text{-й ИО размещается в } j\text{-м узле;} \\ 0, & \text{в противном случае.} \end{cases}$$

При этом полагаем, что все ИО обязательно дублируются в файл-сервере (файл-серверах), а сообщение для корректировки ИО посылается столько раз, сколько копий данного ИО хранится в сети. В качестве узлов будем учитывать лишь РС. Тогда суммарный удельный обмен в сети выразится величиной

$$F_1 = \sum_i \sum_j \lambda_{ij} \sum_k a_{ik} v_k (1 - x_{kj}) + \sum_i \sum_j \lambda'_{ij} v'_1 \left(\sum_{l \neq j} x_{il} \right), \quad (2)$$

где v_k и v'_k — средний объем передаваемой информации на одно взаимодействие, соответственно, при обработке запросов и донесений.

В этом случае математическую модель оптимизационной задачи можно сформулировать в следующем виде. Найдем такой набор значений $X^* = \{x_{kj}^*\}$, при котором

$$F_1 \rightarrow \min_{\{x_{kj}\}} \quad (3)$$

при ограничениях

$$\sum_k w_k x_{kj} \leq V_j, \quad j = \overline{1, J}, \quad (4)$$

$$\sum_j x_{kj} > n_k, \quad k = \overline{1, I}. \quad (5)$$

Здесь величины w_k и V_j — объем памяти, занимаемый k -м ИО, и емкость j -го узла соответственно. Полагаем, что $\lambda_{ij} \neq 0$ лишь для ИО-головных программ. Последние выполняются обязательно в j -ом узле, при этом суммарная загрузка узла не должна превышать допустимой. Величина этой загрузки может быть рассчитана заранее (поскольку λ_{ij} известны) и учтена при выборе состава технических средств узла. n_k определяет минимальную степень дублирования ИО.

Решение задачи (3) — (5) является исходной информацией для определения числа файл-серверов в системе. В самом деле, полагая, что заявки на использование и корректировку k -го ИО характеризуются затратами времени, соответственно, τ_k и τ'_k файл-сервера, можно вычислить суммарную загрузку всех файл-серверов — ρ

$$\rho = \sum_i \sum_j \lambda_{ij} \sum_k a_{ik} (1 - x_{kj}) \tau_k + \sum_i \sum_j \lambda'_{ij} \tau'_i. \quad (6)$$

При этом, как отмечалось выше, полагается, что все ИО дублируются в файл-сервере (в противном случае второй член в (6) будет иметь вид $\sum_i \sum_j (1 - x_{ij}) \lambda'_{ij} \tau'_i$).

Рассчитаем следующие величины:

$$N_\rho = \lceil \rho / \rho_{kp} \rceil, \quad (7)$$

$$N_v = \lceil V / V_\Phi \rceil, \quad (8)$$

где $V = \sum_i w_i$, а V_Φ — емкость памяти одного файл-сервера, используемого для хранения ИО.

Отсюда потребное количество файл-серверов определяется таким образом:

$$N = \max\{N_\rho; N_v\}.$$

Рассмотрим методический пример. Пусть имеем ЛВС, состоящую из 4 РС, в которой должно храниться 10 ИО. Емкость памяти, выделяемой для хранения ИО в РС такова, что возможно хранение одновременно лишь не более 3-х ИО (например, $W_j = 1500, j = 1, 4$, а $w_k = 500, k = 1, 10$). Величины $n_i = 0, i = \overline{1, 10}$. Значения элементов матрицы A приведены в табл. 1.

Таблица 1

Номер ИО головной программы	Номера ИО, используемых при выполнении головной программы									
	1	2	3	4	5	6	7	8	9	10
1	1	1	-	1	-	-	-	-	-	1
3	-	-	1	1	-	1	-	-	1	-
5	-	1	-	1	1	-	-	1	1	1
7	-	-	-	1	-	1	1	1	-	-

В табл. 2 представлены значения величины λ_{ij} (числитель) и λ'_{ij} (знаменатель). По структуре информации в строках табл. 2 видно, что ИО 1, 3, 5 и 7 являются головными программами, ИО 4, 8, 9, 10 — массивами, а ИО 2 и 6 — вызываемыми программами. Причем задано, что $v_i = 500$ при запросе массива; $v'_i = 50$ при корректировке массива и $v_i = 300$ при запросе программы.

Таблица 2

Номер ИО	Номер рабочей станции			
	1	2	3	4
1	500/-	5/-	20/-	10/-
2	-	-	-	-
3	1/-	1/-	500/-	5/-
4	-	-	-/100	-/1
5	1/-	-	1/-	100/-
6	-	-	-	-
7	-	500/-	-	1/-
8	-/1	-/100	-	-/1
9	-	-	-/100	-
10	-/100	-	-/1	-

Полученное в соответствии с (2) — (5) решение оптимизационной задачи представлено в табл.3.

Таблица 3

Номер ИО	Номер рабочей станции			
	1	2	3	4
1	1	-	-	-
2	-	-	-	-
3	-	-	1	-
4	1	1	1	1
5	-	-	-	-
6	-	-	-	-
7	-	1	-	-
8	-	1	-	-
9	1	-	1	1
10	-	-	-	1

Данное решение обеспечивает минимальный удельный объем взаимодействия информации в сети, равный 644950 (604300 по реализации запросов и 40550 — по корректировке). Для сравнения укажем, что в случае, когда все ИО хранятся только лишь в файл-сервере, объем взаимодействия информацией в сети составляет 2949400 (2934200+15200), т. е. более, чем в 4,5 раза больше. Таким образом, за счет оптимизации размещения ИО между РС и файл-серверами может быть значительно уменьшен объем обмена информацией даже в относительно небольшой по размерам сети.

В соответствии с (7) и (8) получаем, что $N_p = 1$, $N_v = 1$, т. е. в системе достаточно иметь один сервер. В то же время в случае размещения всех ИО только в файл-сервере $N_p = 2$, $N_v = 1$ и $N = \max\{2, 1\} = 2$. То есть необходимо два файл-сервера. Следовательно, оптимизация размещения ИО в ЛВС приводит не только к уменьшению взаимодействия информацией в сети, что само по себе дает экономический эффект, но и к уменьшению числа файл-серверов. Это также дает экономию средств.

Теперь рассмотрим случай, когда сервер обеспечивает вычисления, так что ряд запросов реализуется в сервере, а именно, те запросы, головные программы которых отсутствуют в соответствующих РС. Показатель эффективности для данного случая запишется следующим образом:

$$F_2 = \sum_{ij} \sum_{k=j} \lambda_{ij} [x_{ij} \sum_{k=j} a_{ik} v_k (1 - x_{kj}) + (1 - x_{kj}) v_{i0}] + \sum_{ij} \sum_{l=j} \lambda'_{ij} v'_i \sum_{l=j} x_{il}.$$

Здесь величина v_{i0} характеризует объем обмена информации с РС при реализации запроса в сервере. При этом полагается, что все ИО также дублируются в сервере. Математическая модель данной задачи запишется следующим образом:

$$F_2 \rightarrow \min_{\{x_{ij}\}} \quad (9)$$

$$\sum_k w_k x_{kj} \leq V_j, \quad j = \overline{1, J}, \quad (10)$$

$$\sum_j x_{ij} \geq n_i, \quad i = \overline{1, I}, \quad (11)$$

$$\sum_i \lambda_{ij} \tau_{ij} x_{ij} \leq \rho_j, \quad j = \overline{1, J}, \quad (12)$$

$$\sum_i \sum_j \lambda_{ij} \tau_{FS} (1 - x_{ij}) \leq \rho_{FS}. \quad (13)$$

Здесь соотношения (12) и (13) связаны с необходимостью обеспечения ограничений на предельную загрузку, соответственно, рабочих станций и серверов. Причем (13), по сути, неявно задает предельное число серверов.

Рассмотрим расширенный предыдущий пример. Пусть значения заданы ниже:

Номер головной программы	1	3	5	7
v_{i0}	10	1000	10	100

Задано также $\rho_j = 0,65$ для $j = 1,4$; $\rho_{FS} = 0,53$; $\tau_{ij} = 0,001$ для $i = 1,10$; $j = 1,4$; $\tau_{FS} = 0,002$. Решение задачи (9) — (13) дает следующий результат. Вычисления при инициализации ИО—3 выполняются на РС—3; ИО—7 на РС—2. Для ИО—1 и ИО—5 вычисления выполняются сервером. При этом суммарный взаимообмен информацией в сети будет равен 39770 (14370 по запросам и 25400 по корректировке). Если же все вычисления реализуются только в сервере, то объем взаимообмена информацией в сети составляет 1041570 (1021370 + 20200). Таким образом, при последнем варианте организации функционирования сети взаимообмен информацией увеличивается примерно в 26 раз.

Для реализации оптимального варианта требуется 2 сервера, в альтернативном варианте — 3. Налицо значительный выигрыш.

После решения рассмотренных задач в достаточно масштабных сетях может возникнуть необходимость оптимального распределения нагрузки между серверами либо организации функционирования сети, состоящей из некоторого количества подсетей. В некоторых случаях подобные задачи связывают с задачей оптимального выбора технических средств в узлах. Эти и другие задачи могут быть сформулированы и решены в рамках предлагаемого подхода.

Предлагаемый методологический подход к оптимизации СРОД основывается на простых физических соображениях и позволяет получить достаточно компактные формулировки оптимизационных задач в виде, как правило, задач математического псевдобулевого программирования. Для решения таких задач могут быть использованы известные алгоритмы [2]. Имеющийся опыт показывает, что во многих ситуациях за счет оптимизации функционирования СРОД ее производительность может быть повышена на порядок и более.

ЛИТЕРАТУРА

1. *Балыбердин В. А.* Оценка и оптимизация характеристик систем обработки данных. М., 1987. 176 с.
2. *Hammer P. L., Rudeanu S.* Pseudo-Boolean Methods for Bivalent Programming. Berlin, 1966.

Рукопись поступила 25.10.93