

УДК 339.138

МОДЕЛЬ И АЛГОРИТМ ОПТИМИЗАЦИИ НАЗНАЧЕНИЯ ОБЪЕКТОВ НА УЗЛЫ РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

А. А. ПРИХОЖИЙ

*Высший государственный колледж связи
Ф. Скорины, 8/2, Минск, 220114, Беларусь*

Поступила в редакцию 4 марта 2010

Предложена модель назначения объектов на узлы распределенной информационно-вычислительной системы, учитывающая характеристики объектов, потоков запросов к объектам, узлов обработки запросов, каналов передачи данных. Сформулирована оптимизационная задача назначения объектов на узлы по критерию минимума среднего времени обработки запроса с учетом ограничения на время обработки всех запросов. Предложены алгоритм решения задачи и его программная реализация. Проведены эксперименты, выявившие закономерности и тренды в изменении параметров функционирования оптимизированной системы в зависимости от параметров модели.

Ключевые слова: распределенная информационно-вычислительная система, модель назначения объектов на узлы, задача оптимизации, алгоритм оптимизации.

Введение

Эффективное функционирование распределенных параллельных информационно-вычислительных систем обеспечивается двумя главными составляющими: 1) назначением операций на узлы системы и определением механизмов синхронизации их выполнения с целью обеспечения высокой загрузки оборудования [1]; 2) распределением информации на узлах системы с возможным дублированием объектов с целью сокращения трафика в каналах передачи данных [2–4]. Задача оптимизации размещения информации на узлах актуальна для многих типов информационно-вычислительных систем. Критериями оптимизации являются: эффективное разбиение системы на части; локализация взаимосвязанных операций и данных на одном и том же узле; минимизация потоков в каналах передачи данных.

В работе [5], посвященной оптимизации глобальных сетей, целью является минимизация стоимости сети и поиск оптимальных топологии и маршрутов передачи данных при условии, что потоки между узлами известны. Для описания потоков используется граф требований, для описания сети и маршрутов используется граф топологии. В работах [2–4], посвященных задаче размещения объектов на узлах распределенной системы, предложены эвристические алгоритмы решения этой задачи. Среди множества других алгоритмов отметим так называемые "жадные" алгоритмы, использующие эвристики, базирующиеся на критериях минимальной стоимости перераспределения объектов, максимального балансирования загрузки узлов и других. Описанные алгоритмы оценивают вес каждого объекта, вычисляют число копий объектов с учетом размера дискового пространства, параметра плотности, определяемого как нагрузка на единицу памяти, отводимой под объект, интенсивности трафика, возникающего при перераспределении объектов на узлах.

В настоящей статье целью является поиск назначения объектов на узлы с учетом двух основных критериев: 1) нахождение минимума среднего времени обработки одного запроса; 2) сокращение времени обработки всех запросов в сети. Для этого предлагается модель задачи оптимизации, одновременно учитывающая характеристики объектов и потоков запросов к объ-

ектам, производительность узлов, пропускную способность каналов передачи данных. На базе модели разрабатывается точный алгоритм поиска оптимального назначения. Алгоритм реализуется программно и служит базой для проведения серии вычислительных экспериментов, выявляющих закономерности в функционировании оптимизированных распределенных информационно-вычислительных систем.

Модель назначения объектов на узлы

Построим модель распределенной системы, позволяющую, с одной стороны, с достаточной степенью точности и, с другой стороны, с достаточной степенью абстрактности сформулировать задачу оптимизации назначения объектов на узлы.

Пусть $N=\{1, \dots, n\}$ — множество узлов, $M=\{1, \dots, m\}$ — множество объектов. Длины объектов опишем множеством $L=\{l_1, \dots, l_m\}$. Эти длины определяют размеры памяти, необходимые для размещения объектов на узлах. Далее пусть $Q=\{q_{ij}/i \in N, j \in M\}$ — множество запросов к объектам, принимаемых или генерируемых на узлах. Частоты появления запросов опишем множеством $F=\{f_{ij}/i \in N, j \in M\}$, где f_{ij} — число запросов q_{ij} к объекту j с узла i в течение времени наблюдения $Time_Period$. Общее число запросов, поступивших за $Time_Period$, обозначим через $\Sigma(F)$. В ответ на запрос пересылается либо весь объект длиной l_j , либо его часть длиной $send(l_j)$.

Узлы системы характеризуются параметрами. Пусть $S=\{s_1, \dots, s_n\}$ — множество объемов памяти на узлах. Для того, чтобы все объекты могли быть размещены, суммарный объем памяти $\Sigma(S)$ должен превышать суммарную длину объектов $\Sigma(L)$. Собственные времена обработки запросов на узлах опишем множеством $R=\{r_{ij}/i \in N, j \in M\}$. Величина $1/r_{ij}$ есть число запросов к объекту j , обрабатываемых узлом i за единицу времени. Узлы различной производительности обрабатывают запросы к одному объекту за различное время.

Если запрос к объекту j принят или сгенерирован на узле k , но обрабатывается узлом i , данные об объекте длиной $send(l_j)$ составляют часть потока в канале (i, k) передачи данных. От того, на каких узлах размещаются объекты, зависят размеры потоков в каналах передачи данных. Временную задержку передачи единицы данных в канале (i, k) обозначим через c_{ik} . Множество $C=\{c_{ik}/i, k \in N\}$ представляет задержки во всех каналах передачи данных. Если узлы i, k не связаны каналом передачи данных, $c_{ik}=\infty$. Производительность канала (i, k) определяется величиной $1/c_{ik}$.

Распределенная информационно-вычислительная система характеризуется двумя основными временными параметрами: средним временем $T_{request}$ обработки одного запроса к объекту и временем T_{total} обработки всех запросов. Каждый из этих параметров складывается из двух составляющих $T_{processing}+T_{waiting}$, где $T_{processing}$ — время непосредственной обработки (пересылки) на узле (канале связи); $T_{waiting}$ — общее время ожидания, обусловленное случайным характером процесса поступления и обработки запросов. В данной статье внимание сфокусируем на исследовании первой составляющей.

Обозначим через $X=\{x_{ij}/x_{ij} \in \{0, 1\}, i \in N, j \in M\}$ матрицу назначения объектов на узлы. Если $x_{ij}=1$, объект j размещается на узле i , в противном случае не размещается. Число объектов, размещенных на узле i , определяется выражением

$$n^a = \sum_{j=1}^m x_{ij}, \quad (1)$$

а емкость памяти, занимаемая объектами на узле i , определяется выражением

$$S^a_i = \sum_{j=1}^m l_j \cdot x_{ij}. \quad (2)$$

Время обработки запроса к объекту j , поступившего с узла i и обрабатываемого узлом k , можно описать с достаточной степенью точности выражением $f_{ij} \cdot (r_{kj} + c_{ik} \cdot l_j) \cdot x_{kj}$. Поскольку при размещении объектов без репликаций все запросы к объекту j обрабатываются на одном узле, процедура оценки временных показателей упрощается. Среднее время обработки запроса

по всем узлам, всем объектам и всем запросам зависит от матрицы назначения X и оценивается выражением:

$$T_{request}(X) = \frac{1}{\Sigma(F)} \sum_{k=1}^n \sum_{j=1}^m \sum_{i=1}^n f_{ij} \cdot r_{kj} + c_{ik} \cdot send(l_j) \cdot x_{kj} . \quad (3)$$

Время T_{total} обработки всех запросов оценим через максимальную временную загрузку всех узлов и каналов. Загрузка узла $k \in N$ определяется выражением

$$LoadNode_k(X) = \sum_{j=1}^m \sum_{i=1}^n f_{ij} \cdot r_{kj} \cdot x_{kj} , \quad (4)$$

в то время как максимальная загрузка узла выражением

$$LoadNode_{max}(X) = \max_{k \in N} LoadNode_k . \quad (5)$$

Загрузка канала (i, k) оценивается в виде

$$LoadChannel_{ik}(X) = \sum_{j=1}^m f_{ij} \cdot c_{ik} \cdot send(l_j) \cdot x_{kj} , \quad (6)$$

в то время как максимальная загрузка канала в виде

$$LoadChannel_{max}(X) = \max_{i, k \in N} LoadChannel_{ik} . \quad (7)$$

Тогда время обработки всех запросов определяется выражением

$$T_{total}(X) = \max LoadNode_{max} , LoadChannel_{max} . \quad (8)$$

Назначение объектов на узлы является корректным, если время обработки всех запросов не превышает ширину диапазона времени, на котором измерены частоты появления запросов, другими словами, если выполняется отношение

$$T_{total}(X) \leq Time_Period . \quad (9)$$

Относительная загрузка узлов оценивается как

$$LoadNdRel_k = (LoadNode_k(X) / T_{total}(X)) * 100\% , \text{ для } k \in N \quad (10)$$

а относительная загрузка каналов оценивается как

$$LoadNdRel_{ik} = (LoadChannel_{ik}(X) / T_{total}(X)) * 100\% , \text{ для } i, k \in N \quad (11)$$

Иллюстративный пример модели назначения объектов на узлы распределенной системы дан на рис. 1. Все компоненты модели представлены векторами и матрицами. Матрица X описывает одно из возможных назначений объектов на узлы и позволяет оценить объем используемой памяти на каждом узле, загрузку всех узлов и каналов, среднее время обработки одного запроса и время обработки всех запросов.

Алгоритм оптимизации назначения объектов на узлы

В предположении того, что назначение объектов на узлы корректно в том смысле, что любая часть распределенной информационно-вычислительной системы работает при обработке всех запросов не дольше, чем $Time_Period$, сформулируем задачу минимизации среднего времени обработки одного запроса. Целевая функция:

$$\min_{X \in \Omega} T_{request}(X) . \quad (12)$$

$$\begin{array}{l}
L = [742, 186, 381, 632, 704, 378, 37, 556, 535, 388] \quad S = \begin{bmatrix} 195 & 4252 & 2630 \end{bmatrix} \\
R = \begin{bmatrix} 9,4 & 3,9 & 7,7 & 1,8 & 9,3 & 7,9 & 8,8 & 2,7 & 7,1 & 4,5 \\ 6,0 & 7,0 & 4,5 & 8,2 & 2,8 & 5,2 & 2,9 & 7,4 & 9,0 & 9,6 \\ 8,1 & 3,0 & 1,5 & 8,7 & 8,3 & 1,7 & 0,8 & 2,2 & 5,3 & 3,2 \end{bmatrix} \quad C = \begin{bmatrix} 0,0000 & 0,0131 & 0,0113 \\ 0,0075 & 0,0000 & 0,0168 \\ 0,0186 & 0,0156 & 0,0000 \end{bmatrix} \\
F = \begin{bmatrix} 63 & 98 & 86 & 72 & 69 & 48 & 67 & 65 & 11 & 23 \\ 44 & 41 & 45 & 58 & 47 & 24 & 14 & 88 & 55 & 28 \\ 85 & 53 & 36 & 0 & 95 & 67 & 80 & 53 & 69 & 6 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \\
n^a = \begin{bmatrix} 2 & 4 \end{bmatrix} \quad S^a = \begin{bmatrix} 762 & 1446 & 1331 \end{bmatrix} \\
Lnd = \begin{bmatrix} 1,79 & 1,75 & 1,33 \end{bmatrix} \quad Lnd^{\%} = \begin{bmatrix} 82,1\% & 80,1\% & 60,8\% \end{bmatrix} \\
Lch = \begin{bmatrix} 0,0 & 0,71 & 1,10 \\ 1,37 & 0,0 & 0,88 \\ 0,47 & 2,19 & 0,0 \end{bmatrix} \quad Lch^{\%} = \begin{bmatrix} 0,0 & 32,7\% & 50,4\% \\ 62,5\% & 0,0 & 40,1\% \\ 21,6\% & 100,0\% & 0,0 \end{bmatrix}
\end{array}$$

Рис. 1. Иллюстративный пример модели назначения объектов на узлы; число узлов $n=3$, число объектов $m=10$; время в R и C измерено в миллисекундах; время в векторе загрузки узлов Lnd и матрице загрузки каналов Lch выражено в секундах; значением функции $send(l_j)$ является l_j ; среднее время обработки запроса $T_{request}=7,3$ мс; время обработки всех запросов $T_{total}=2,19$ с; средняя загрузка узла 74,3%; средняя загрузка канала 51,2%

Ограничения:

1) на общий объем памяти в узле:

$$\sum_{j=1}^m x_{ij} * l_j \leq S_i \text{ для всех } i \in N; \quad (13)$$

2) на размещение объекта ровно на одном узле:

$$\sum_{i=1}^n x_{ij} = 1 \text{ для всех } j \in M. \quad (14)$$

С учетом (3) преобразуем целевую функцию (12) к виду

$$\min_{X \in \Omega} \frac{1}{\Sigma(F)} \sum_{k=1}^n \sum_{j=1}^m g_{kj} * x_{kj}, \quad (15)$$

где величина

$$g_{kj} = \sum_{i=1}^n f_{ij} * r_{kj} + c_{ik} * send(l_j). \quad (16)$$

Элементы g_{kj} при $k \in N, j \in M$ образуют множество G . Значением g_{kj} является суммарное время обработки узлом k всех запросов к объекту j , поступивших со всех узлов $i \in N$. Легко видеть, задача (12)–(15) относится к задаче целочисленного линейного программирования.

Для решения задачи предлагается точный алгоритм *AssignON* (который при ограничении на процессорное время поиска решения становится приближенным), построенный на базе метода ветвей и границ. Рекурсивное описание алгоритма на псевдокоде приведено на рис. 2. Для оценки нижней границы стоимости решения используются величины

$$G_{\min}(b) = \sum_{j=b}^m \min_{k \in N} G(j, k) \quad \text{для } b \in M. \quad (17)$$

```

function AssignON(j, T) {
    if(j >= M) {
        if(Topt < T) {
            Построить в качестве оптимального решения новое
            назначение объектов на узлы;
            Topt = T;
        }
    } else {
        object = orderO-1(j);
        for node = orderN-1(1, object), ..., orderN-1(n, object) {
            NewS = Sa(node) + L(object);
            if(S(node) > NewS) {
                NewT = T + G(object, node);
                if(NewT + Estimation(object) > Topt) {
                    Перейти к новому узлу node в цикле for
                } else {
                    Назначить объект object на узел node
                    Volume = Sa(node);
                    Sa(node) = NewS;
                    AssignON(j+1, newT);
                    Отменить назначение объекта object на узел node
                    Sa(node) = Volume;
                }
            }
        }
    }
}

function Estimation(object) {
    if(object < M-1)
        return Gmin(object + 1);
    else
        return 0;
}

```

Рис. 2. Алгоритм *AssignON* оптимизации назначения объектов на узлы, представленный на псевдокоде

В начале выполнения алгоритма *AssignON* все элементы множества S^a , занимаемых на узлах емкостей памяти, зануляются. Отображение $orderO(object) \rightarrow j$ для каждого объекта *object* устанавливает порядковый номер *j* такой, что выполняется сортировка объектов по убыванию разброса значений величин $G(object, i)$ для $i=1, \dots, n$. Отображение $orderO^{-1}(j) \rightarrow object$ является обратным для отображения *orderO*. Для всех объектов $object \in M$ отображение $orderN(node, object) \rightarrow k$ устанавливает для каждого узла *node* порядковый номер *k* такой, что выполняется сортировка узлов по неубыванию величин $G(object, node)$ для $node=1, \dots, n$. Отображение $orderN^{-1}(j, object) \rightarrow node$ является обратным для отображения *orderN*.

В процессе работы алгоритма используется текущее оптимальное назначение объектов на узлы со средним временем T_{opt} обработки запроса. Если глубина рекурсивного вызова функции *AssignON* равна *m* и новое найденное назначение лучше текущего оптимального, оптимальное назначение обновляется. Если глубина рекурсивного вызова меньше *m*, выполняется перебор назначений текущего объекта на различные узлы, при этом оценивается минимальное значение целевой функции и принимается решение об отсечении ветви поиска либо продолжении рекурсивных вызовов.

Результаты вычислительных экспериментов

Алгоритм оптимизации назначения объектов на узлы реализован в виде программы, написанной на языке C/C++. Программа положена в основу вычислительных экспериментов, выполненных посредством генерации всех компонентов модели с использованием датчика слу-

чайных чисел и варьированием параметров модели в широких пределах. Основные закономерности и тренды, выявленные в поведении распределенной системы, построенной посредством выполнения оптимального алгоритма назначения объектов на узлы, представлены на рис. 3, 4.

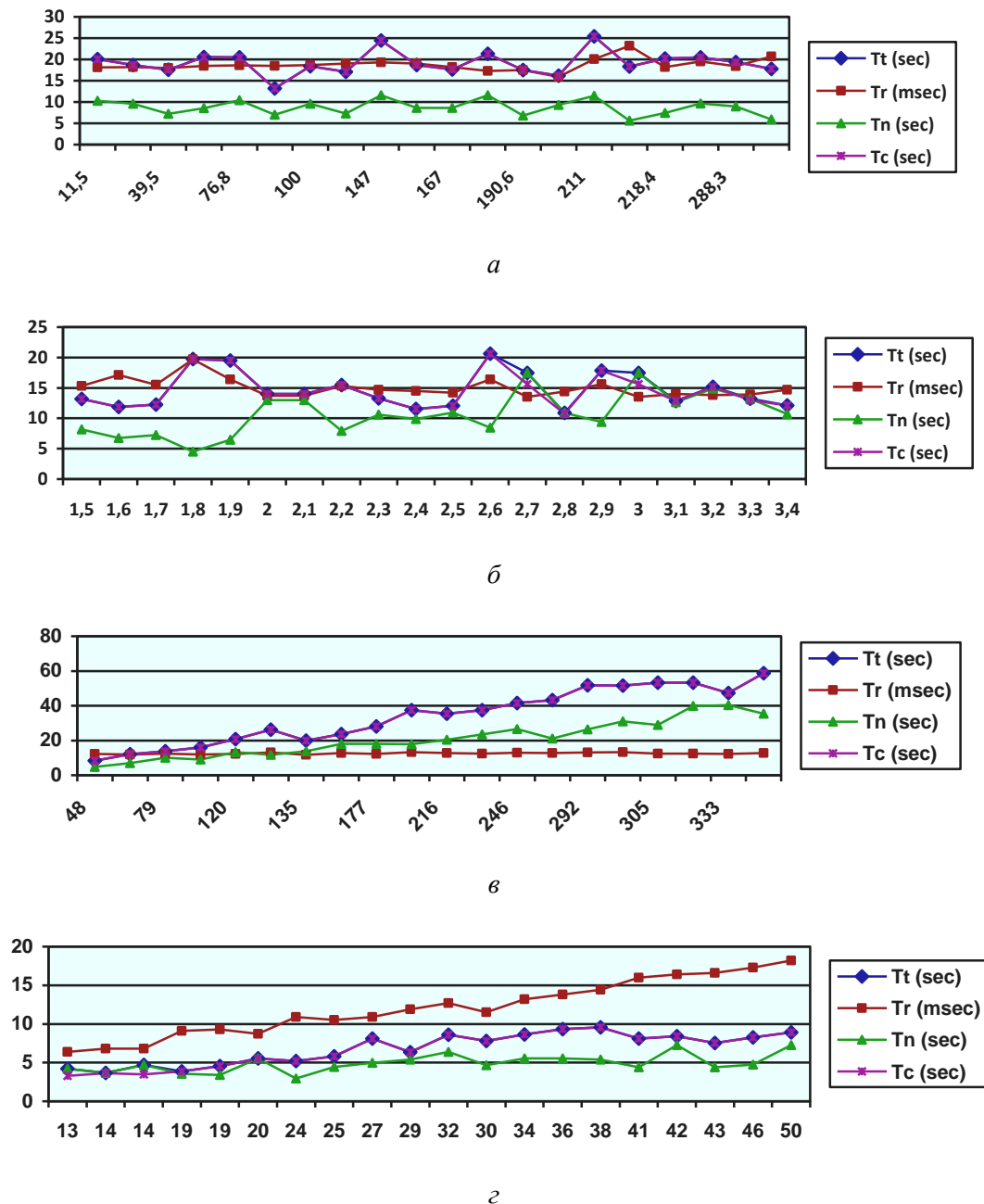


Рис. 3. Зависимость параметров распределенной системы: *а* — от разброса длин объектов; *б* — от отношения "объем памяти узлов / суммарная длина объектов"; *в* — частоты запросов в узлах; *г* — производительности каналов передачи данных

Рис. 3,а показывает, что время T_t обработки всех запросов, являющееся максимумом узлового времени T_n и канального времени T_c , и среднее время T_r обработки одного запроса не имеют устойчивого тренда в зависимости от изменения разброса длин объектов в диапазоне от 11,5 до 291,0 при сохранении средней длины объекта.

Из рис. 3,б следует, что увеличение суммарной емкости памяти на узлах, превышающей суммарную длину объектов с коэффициентом от 1,5 до 3,4 раз, не дает существенного уменьшения среднего времени T_r обработки одного запроса. Время загрузки каналов T_c вносит до-

минирующий вклад в общее время Tt по сравнению с временем загрузки узлов Tn , хотя загрузка узлов и ее влияние на общее время увеличиваются с увеличением емкости памяти узлов.

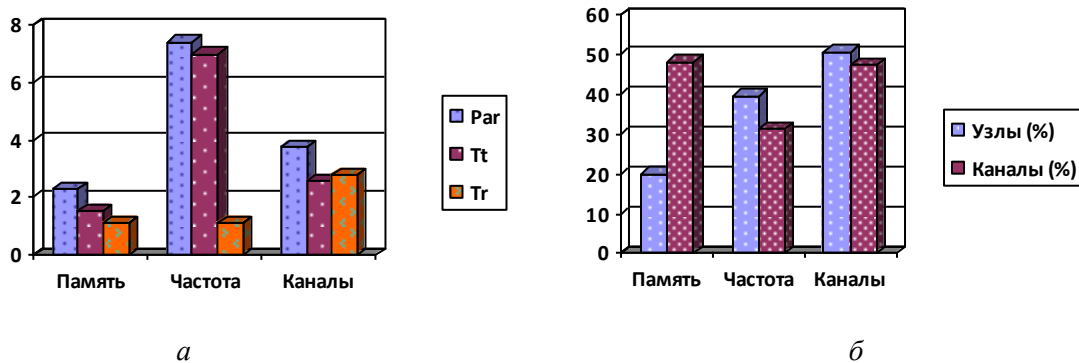


Рис. 4. Диапазоны: *a* — изменения параметров и средняя загруженность; *б* — узлов и каналов при изменениях в емкости памяти узлов, частоте запросов, производительности каналов передачи данных

Рис. 3,в показывает, что с увеличением с 48 до 356 средней частоты запросов к одному объекту с одного узла, среднее время Tr обработки одного запроса почти не увеличивается, в то время как общее время Tt , совпадающее в эксперименте с временной загрузкой каналов Tc , возрастает существенно. Временная нагрузка узлов Tn увеличивается, однако не является доминирующей в Tt .

Влияние средней временной задержки передачи данных в каналах на временные параметры всей системы показано на рис. 3,г. Увеличение задержек в каналах вызывает существенное увеличение среднего времени Tr и общего времени Tt . Временная нагрузка узлов Tn также увеличивается.

Рис. 4,а показывает зависимость обобщенных параметров функционирования информационно-вычислительной системы, а рис. 4,б — зависимость загрузки узлов и каналов от изменения трех параметров модели: емкости памяти, частоты запросов, производительности каналов. При увеличении коэффициента "объем памяти узлов / суммарная длина объектов" до 2,27 общее время Tt обработки запросов изменялось до 1,51 раза, а среднее время Tr обработки одного запроса — до 1,12 раза. Средняя относительная нагрузка узлов составила 48,05%, каналов — 19,96%. Очевидно то, что многие каналы загружены слабо.

При увеличении средней частоты запросов в 7,42 раза общее время Tt возросло в 6,99 раза, среднее время Tr — в 1,12 раза. При этом средняя относительная нагрузка узлов составила 31,20%, каналов — 39,62%.

При снижении средней пропускной способности каналов передачи данных в 3,76 раза общее время Tt возросло в 2,57 раза, среднее время Tr — в 2,80 раза. При этом средняя нагрузка узлов составила 47,33%, каналов — 50,47%. Общий вывод состоит в том, что и узлы, и каналы загружаются неравномерно без использования репликаций объектов при назначении их на узлы.

Заключение

Предложена модель информационно-вычислительной распределенной системы, ориентированная на оптимизацию размещения объектов на узлах без введения репликаций. Разработан точный алгоритм минимизации среднего времени обработки одного запроса к объекту, выполнена его программная реализация. Проведена серия вычислительных экспериментов, выявивших основные закономерности и тренды в изменении параметров функционирования всей системы в зависимости от параметров модели, таких как емкость памяти узлов, частота запросов, производительность каналов передачи данных. Найденная в процессе оптимизации загрузка узлов и каналов может быть использована при выборе сетевого оборудования, определении топологии сети. Необходимо дальнейшее развитие модели и алгоритмов, учитывающих репликации объектов, минимизирующих общее время обработки всех запросов.

MODEL AND ALGORITHM FOR OPTIMIZATION OF OBJECT ASSIGNMENT TO DISTRIBUTED INFORMATION AND COMPUTING SYSTEM NODES

A.A. PRIHOZHY

Abstract

A model of mapping the objects onto the distributed system nodes which takes into account the object attributes, request flows, node and channel parameters is proposed. The object assignment optimization task that minimizes the average time of processing a request with a constraint on the total execution time of all requests is formulated. The recursive algorithm for solving the task and its software implementation are developed. The conducted experiments have found out the rules and trends in optimized system parameter variations depending on the model parameters.

Литература

1. Прихожий А.А., Соломенник М.В. // Докл. БГУИР. 2003. № 4. С. 104–114.
2. Zhuo L., Wang C.-L., Lau F.C.M. // Proc. of the 2002 International Conference on Parallel Processing. IEEE Computer Society, Washington. 2002. P. 305.
3. Khan S.U., Ahmad I. // J. of Parallel and Distributed Computing. 2008. Vol. 68, No. 2. P. 113–136.
4. Прихожий А.А., Труханович Т.Л. // Web-программирование и Интернет технологии. (WebConf-09). Минск, БГУ, 8–10 июня 2009 г. Минск, 2009. С. 12–15.
5. Листопад Н.И. Моделирование и оптимизация глобальных сетей. Минск, 2000.