

УДК 339.138

А.А. Прихожий

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ ОБЪЕКТОВ С УЧЕТОМ ИХ РЕПЛИКАЦИЙ НА УЗЛАХ РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Предлагается модель размещения объектов на узлах распределенной системы, учитывающая введение репликаций, параметры объектов и потоков запросов к объектам, характеристики каналов передачи данных и узлов обработки запросов. Формулируется оптимизационная задача распределения объектов на узлах, предлагается алгоритм наискорейшего спуска решения задачи. Проводятся эксперименты, выявляющие закономерности и тренды в изменении параметров функционирования оптимизированной системы.

Введение

Задача оптимизации размещения информации на узлах распределенной системы [1–4] актуальна для многих типов современных информационно-вычислительных систем, в которых важнейшую роль играет не только фактор совместного параллельного использования большого числа единиц вычислительного и сетевого оборудования, но и географический фактор. Целью решения задачи оптимизации может быть:

- эффективное разбиение системы на части;
- локализация взаимосвязанных операций и данных на одном и том же узле;
- минимизация потоков в каналах передачи данных;
- сокращение времени обработки запросов и др.

Задача размещения информации в распределенной системе тесно связана с задачей планирования информационно-вычислительных процессов, результатом решения которой являются назначение вычислительных операций на узлы и синхронизация их выполнения при условии достижения высокой загрузки оборудования [5] и сокращения времени выполнения сетевого алгоритма.

В работах [1–3] предложены эвристические алгоритмы назначения объектов на узлы распределенной системы. Наиболее эффективными являются так называемые «жадные» алгоритмы, использующие эвристики, которые базируются на критериях минимальной стоимости перераспределения объектов, максимального балансирования загрузки узлов и др. Описанные алгоритмы оценивают вес каждого объекта, вычисляют число копий объектов с учетом размера дискового пространства и параметра плотности, определяемого как нагрузка на единицу памяти, отводимой под объект, а также интенсивности трафика, возникающего при перераспределении объектов на узлах. Целью работы [4], посвященной оптимизации глобальных сетей, 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 – граф топологии.

В настоящей статье оптимизация назначения объектов и их репликаций на узлы выполняется с учетом двух основных критериев: минимума среднего времени обработки одного запроса и минимума времени обработки всех запросов в сети. Предлагаемая модель оптимизации учитывает характеристики объектов и потоков запросов к объектам, производительность узлов, пропускную способность каналов передачи данных. Для поиска оптимального (квазиоптимального) назначения объектов разрабатывается алгоритм наискорейшего спуска, программная реализация которого послужила базой для проведения серии вычислительных экспериментов, выявивших закономерности и тренды в улучшении параметров функционирования оптимизированной распределенной системы.

1. Модель размещения объектов и их репликаций на узлах распределенной системы

Построим модель распределенной системы, позволяющую, с одной стороны, с достаточной степенью точности учесть детали функционирования системы и, с другой стороны, с достаточной

степенью абстрактности сформулировать задачу оптимизации размещения объектов и их репликаций на узлах. 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 $Q = \{q_{ij} \mid i \in N, j \in M\}$ – множество запросов к объектам, принимаемых или генерируемых на узлах. Частоты появления запросов опишем множеством $F = \{f_{ij} \mid i \in N, j \in M\}$, 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 $R = \{r_{ij} \mid i \in N, j \in M\}$. 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 , на котором размещается сам объект или его репликация, данные об объекте длиной $send(l_j)$ составляют часть потока в канале (i, k) передачи данных. От того, на каких узлах размещаются объекты и их репликации, а также от того, какой узел обрабатывает запросы, поступающие с других узлов, зависят размеры потоков в каналах передачи данных. Временную задержку передачи единицы данных в канале (i, k) обозначим через c_{ik} . Множество $C = \{c_{ik} \mid i, k \in N\}$ 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 $1/c_{ik}$.

Пусть $B = \{b_{ij} \mid b_{ij} \in \{0,1\}, i \in N, j \in M\}$ – матрица возможностей обработки запросов на узлах. Если $b_{ij} = 1$, запрос к объекту j может обрабатываться узлом i , в противном случае не может. Способность узла обрабатывать запросы обусловлена рядом причин: наличием соответствующего программного обеспечения, наличием необходимых данных, достаточностью ресурсов и т. д. Обозначим через $X = \{x_{ij} \mid x_{ij} \in \{0,1\}, i \in N, j \in M\}$ матрицу размещения объектов на узлах. Если $x_{ij} = 1$, копия объекта j размещается на узле i , в противном случае не размещается. Значение x_{ij} связано со значением b_{ij} так, что логическая импликация $x_{ij} \rightarrow b_{ij}$ является тождественно истинным высказыванием. Число копий объекта j (сам объект и все его репликации) в распределенной системе определяется выражением

$$n_j^{all} = \sum_{i=1}^n x_{ij}, \quad (1)$$

а емкость памяти, занимаемая всеми копиями объекта j , которые размещаются на узлах, – выражением

$$S_j^{all} = \sum_{i=1}^n l_j * x_{ij}. \quad (2)$$

Фактическое число всех объектов и их репликаций, размещенных на узле i , определяется выражением

$$n_i^a = \sum_{j=1}^m x_{ij}, \quad (3)$$

а фактическая емкость памяти, занимаемая всеми объектами или их репликациями на узле i , – выражением

$$S_i^a = \sum_{j=1}^m l_j * x_{ij}. \quad (4)$$

Пусть $N_j = \{i \mid i \in N, x_{ij} = 1\}$ – множество узлов, на которые назначен объект j и его репликации. Обозначим через $D = \{d_{ij} \mid d_{ij} \in N_j, i \in N, j \in M\}$ матрицу назначения запросам обрабатывающих их узлов. Запрос к объекту j , поступивший на узел i , обрабатывается узлом d_{ij} . Если объекты не имеют репликаций, то $d_{ij} = \dots = d_{nj}$, а это значит, что все запросы к одному объекту обрабатываются на одном узле. На рис. 1, а число узлов в распределенной системе $n = 3$, число объектов $m = 10$, время в матрицах R и C измерено в миллисекундах, частоты появления запросов к каждому объекту даны для каждого из узлов. На рис. 1, б матрица X описывает назначение объектов на узлы, матрица D – назначение запросам обрабатывающих узлов. Поскольку репликации объектов не использованы, элементы каждого из столбцов матрицы D идентичны. Векторы n^a и S^a показывают, сколько объектов размещено на узлах и какую память они занимают.

$$\begin{aligned}
 L &= [742, 186, 381, 632, 704, 378, 37, 556, 535, 388] & S &= [3071 \quad 2748 \quad 2804] \\
 R &= \begin{bmatrix} 6,6 & 2,8 & 5,4 & 1,4 & 6,5 & 5,6 & 6,2 & 2,0 & 5,0 & 3,2 \\ 4,3 & 5,0 & 3,2 & 5,8 & 2,1 & 3,7 & 2,1 & 5,2 & 6,3 & 6,8 \\ 5,7 & 2,2 & 1,2 & 6,1 & 5,9 & 1,3 & 0,7 & 1,7 & 3,8 & 2,4 \end{bmatrix} & C &= \begin{bmatrix} 0,0000 & 0,0131 & 0,0113 \\ 0,0075 & 0,0000 & 0,0168 \\ 0,0186 & 0,0156 & 0,0000 \end{bmatrix} \\
 F &= \begin{bmatrix} 63 & 98 & 86 & 72 & 69 & 48 & 67 & 65 & 11 & 23 \\ 44 & 41 & 45 & 58 & 47 & 24 & 14 & 88 & 55 & 28 \\ 85 & 53 & 36 & 0 & 95 & 67 & 80 & 53 & 69 & 6 \end{bmatrix} \\
 & a) \\
 X &= \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} & D &= \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 3 & 1 \\ 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 3 & 1 \\ 2 & 1 & 2 & 1 & 2 & 3 & 3 & 1 & 3 & 1 \end{bmatrix} \\
 n^a &= [4 \quad 3 \quad 3] & S^a &= [1762 \quad 1827 \quad 950] \\
 Lnd &= [1,32 \quad 1,80 \quad 0,80] & Lnd^a &= [54,6 \quad 74,5 \quad 33,2] \\
 Lch &= \begin{bmatrix} 0,0 & 0,96 & 0,49 \\ 1,37 & 0,0 & 0,61 \\ 0,47 & 2,42 & 0,0 \end{bmatrix} & Lch^a &= \begin{bmatrix} 0,0 & 39,8 & 20,4 \\ 56,6 & 0,0 & 25,2 \\ 19,5 & 100,0 & 0,0 \end{bmatrix} \\
 & b)
 \end{aligned}$$

Рис. 1. Пример модели назначения объектов на узлы: а) исходные компоненты модели; б) размещение объектов на узлах без использования репликаций и параметры функционирования распределенной системы

2. Оценка среднего времени обработки одного запроса

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

Время обработки запроса к объекту j , поступившего с узла i и обрабатываемого узлом k , можно оценить с достаточной степенью точности выражением $f_{ij} * (r_{kj} + c_{ik} * send(l_j)) * x_{kj}$. Среднее время обработки запроса по всем узлам, всем объектам и всем запросам зависит в первую очередь от матрицы назначения X и может быть оценено выражением

$$T_{request}(X) = \frac{1}{\Sigma(F)} \sum_{i=1}^n \sum_{j=1}^m f_{ij} * \min_{k \in N} ((r_{kj} + c_{ik} * send(l_j)) * x_{kj}). \quad (5)$$

В выражении (5) узел обработки запроса q_{ij} выбирается неявно из множества N_j узлов таким образом, чтобы минимизировать время обработки. Если узел $d(i,j) \in N_j$ обработки запроса q_{ij} задан явно посредством матрицы D , среднее время обработки оценивается выражением

$$T_{request}(D) = \frac{1}{\Sigma(F)} \sum_{i=1}^n \sum_{j=1}^m f_{ij} * (r_{d(i,j),j} + c_{i,d(i,j)} * send(l_j)). \quad (6)$$

Соотношение (6) для оценки среднего времени является более общим, чем соотношение (5), так как матрица D может быть построена не только из соображений минимизации времени обработки каждого из запросов, но также с учетом других более общих соображений.

С учетом изложенного применим соотношение (6) к оценке среднего времени обработки одного запроса для модели без репликаций и матриц X и D (см. рис. 1, б). Предположим, что значением функции $send(l_j)$ является l_j , а все элементы матрицы B единичные, т. е. все узлы способны обрабатывать все типы запросов. Тогда сумма времен обработки всех запросов равна 10235,4 мс при общем количестве запросов $\Sigma(F) = 1590$. Среднее время обработки одного запроса $T_{request} = 6,4$ мс.

3. Оценка времени обработки всех запросов

Время T_{total} обработки всех запросов оценим как максимальное время работы по всем компонентам распределенной системы, т. е. через максимальную временную загрузку всех узлов и каналов. Загрузка узла $k \in N$ зависит от матрицы D и определяется выражением

$$Lnd_k(D) = \sum_{\substack{i \in N, j \in M, \\ d(i,j)=k}} (f_{ij} * r_{kj}), \quad (7)$$

в то время как максимальная загрузка узла – выражением

$$LoadNode_{max}(D) = \max_{k \in N} \{Lnd_k(D)\}. \quad (8)$$

Загрузка канала (i, k) оценивается в виде

$$Lch_{ik}(D) = \sum_{\substack{j \in M, \\ d(i,j)=k}} (f_{ij} * c_{ik} * send(l_j)), \quad (9)$$

в то время как максимальная загрузка канала – в виде

$$LoadChannel_{max}(D) = \max_{i,k \in N} \{Lch_{ik}(D)\}. \quad (10)$$

Тогда время обработки всех запросов определяется выражением

$$T_{total}(D) = \max\{LoadNode_{max}, LoadChannel_{max}\}. \quad (11)$$

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

$$T_{total}(D) \leq Time_Period. \quad (12)$$

Относительная загрузка узлов может быть оценена как

$$Lnd_k^{\%} = (Lnd_k(D) / T_{total}(D)) * 100 \% \quad \text{для } k \in N, \quad (13)$$

а относительная загрузка каналов – как

$$Lch_{ik}^{\%} = (Lch_{ik}(D) / T_{total}(D)) * 100 \% \quad \text{для } i, k \in N. \quad (14)$$

Применим соотношения (7)–(14) к модели, показанной на рис. 1. Временная загрузка узлов и каналов описывается вектором Lnd и матрицей Lch . Канал между узлами 3 и 2 (пересечение третьей строки и второго столбца матрицы Lch) имеет максимальное время работы, равное 2,42 с. Относительная загрузка канала равна 100 %. Узел 2 имеет максимальную загрузку, равную 74,5 %.

4. Задача оптимизации размещения объектов на узлах

Предположим, что размещение объектов на узлах корректно в том смысле, что любая часть распределенной информационно-вычислительной системы расходует на обработку всех запросов время, не превышающее $Time_Period$. Сформулируем задачу оптимизации размещения объектов с учетом следующих допущений: один объект j может иметь несколько копий p_j и размещаться на подмножестве узлов N_j ; запрос на объект j , поступивший с узла i , обрабатывается узлом $d(i,j) \in N_j$, дающим минимальное время отклика. Последнее условие можно записать как

$$i \in N, j \in M \text{ и } d(i,j) \text{ – узел обработки, дающий время } \min_{k \in N} (r_{kj} + c_{ik} * send(l_j)) * x_{kj}. \quad (15)$$

Целевая функция в задаче оптимизации минимизирует среднее время предоставления объекта в ответ на один запрос пользователя по всем узлам, всем объектам и всем запросам:

$$\min_{D \in \Lambda} \{T_{request}(D)\}, \quad (16)$$

где Λ – множество всех возможных корректных матриц D , а $T_{request}(D)$ определяется формулой (6) с учетом соотношения (15) для узлов обработки запросов.

Ограничения в задаче оптимизации:

– на общий объем памяти в узле:

$$\sum_{j=1}^m x_{ij} * l_j \leq S_i \quad \text{для всех } i \in N; \quad (17)$$

– на размещение объекта не менее чем на одном узле:

$$\sum_{i=1}^n x_{ij} \geq 1 \quad \text{для всех } j \in M. \quad (18)$$

Очевидно, что целевая функция (16) является нелинейной, а задача оптимизации не относится к классу задач линейного целочисленного программирования. Возможны два подхода к решению задачи (16)–(18): эквивалентное преобразование к задаче целочисленного линейного программирования; разработка специального алгоритма, учитывающего особенности задачи (16)–(18). Переход к задаче линейного программирования требует введения $n \times n \times m$ дополнительных целочисленных переменных, что приводит к существенному увеличению размерности решаемой задачи для реальных распределенных систем [3]. Вследствие этого обоснованной является разработка алгоритма, учитывающего специфику задачи (16)–(18).

5. Алгоритм наискорейшего спуска решения задачи оптимизации

Декомпозируем задачу (16)–(18) размещения объектов и их репликаций на узлах распределенной системы на три подзадачи:

1) оптимальное по критерию (16) назначение объектов на узлы без использования репликаций;

2) размещение репликаций с целью максимального уменьшения среднего времени обработки одного запроса;

3) назначение обрабатывающих узлов, минимизирующих время обработки запросов и учитывающих ограничения на объемы памяти.

Для решения первой подзадачи реализован точный алгоритм, базирующийся на методе ветвей и границ (см. рис. 1, б). Результатом решения является матрица X , каждая строка которой содержит ровно одну единицу, векторы n^a , S^a и матрица D , в которой всеми элементами одного столбца j является число $i \in N$, такое, что $x_{ij} = 1$. По матрице D посредством соотношений (7), (9) вычисляется временная загрузка узлов и каналов, посредством соотношений (13), (14) – относительная загрузка узлов и каналов.

Вторая подзадача решается процедурой *OrderReplications*. Для решения третьей подзадачи используется процедура *AssignReplications*. Совместно процедуры описывают предлагаемый алгоритм наискорейшего спуска, минимизирующий среднее время обработки одного запроса (рис. 2):

```

procedure OrderReplications {
  Inputs:  $N, M, X, L, R, C$ 
  Outputs:  $T, \Delta T, Y$ 
  Для  $j \in M$  {
    Для  $i \in N$  присвоить  $y_{ij} := x_{ij}$ .
    Выбрать  $w$ , для которого  $x_{wj} = 1$ , и присвоить  $N_j(I) := \{w\}$  и  $\Delta T_{wj} := 0$ ;  $T_{wj} := 0$ .
    Для  $i \in N$  присвоить  $\tau_{ij}(I) := f_{ij} * (r_{wj} + c_{iw} * send(I_j))$  и  $T_{wj} := T_{wj} + \tau_{ij}(I)$ .
    Для  $e := 2, \dots, n$  {
      Присвоить  $\Delta T_j^{max}(e) := 0$ .
      Для  $k \in N \setminus N_j(e-1)$  {
        Присвоить  $\Delta T_j^k(e) := 0$ .
        Для  $i \in N$  {
          Присвоить  $\Delta \tau_{ij}(e) := \tau_{ij}(e-1) - f_{ij} * (r_{kj} + c_{ik} * send(I_j))$ .
          Если  $\Delta \tau_{ij}(e) > 0$ , то присвоить  $\Delta T_j^k(e) := \Delta T_j^k(e) + \Delta \tau_{ij}(e)$ .
        }
        Присвоить  $\Delta T_j^k(e) := \Delta T_j^k(e) / l_j$ .
        Если  $\Delta T_j^{max}(e) < \Delta T_j^k(e)$ , то
          присвоить  $\Delta T_j^{max}(e) := \Delta T_j^k(e)$  и присвоить  $maxN := k$ .
      }
      Если  $\Delta T_j^{max}(e) > 0$ , то {
        Присвоить  $\Delta T_{maxNj} := \Delta T_j^{max}(e)$  и  $N_j(e) = N_j(e-1) \cup \{maxN\}$  и  $y_{maxNj} := e$  и  $T_{maxNj} := 0$ .
        Для  $i \in N$  {
          Присвоить  $\tau_{ij}(e) := f_{ij} * (r_{maxNj} + c_{i,maxN} * send(I_j))$  и  $\Delta \tau_{ij}(e) := \tau_{ij}(e-1) - \tau_{ij}(e)$ ;
          Если  $\Delta \tau_{ij}(e) > 0$ , то присвоить  $T_{maxNj} := T_{maxNj} + \tau_{ij}(e)$ ,
          иначе присвоить  $T_{maxNj} := T_{maxNj} + \tau_{ij}(e-1)$ ;
        }
      }
    }
  }
}
procedure AssignReplications {
  Inputs:  $Y, \Delta T, L, S$ 
  Outputs:  $X, S^a, n^a, D$ 
  Для  $i \in N, j \in M$ , если  $y_{ij} = 1$ , присвоить  $x_{ij} := 1$ , иначе  $x_{ij} := 0$ .
  Для  $i \in N$  присвоить  $S_i^a := 0$  и  $n_i^a := 0$ . Для  $j \in M$  присвоить  $best(j) := 2$ .
  Пока (возможно введение репликаций объектов) {
    По всем объектам выбрать  $j \in M$  и  $i \in N$ , такие, что  $y_{ij} = best(j)$  и  $\Delta T_{ij}$  является максимальным.
    Если  $S_i^a + l_j \leq S_i$  {
      Ввести репликацию объекта  $j$  на узле  $i$ . Присвоить  $x_{ij} := 1$ ,  $S_i^a := S_i^a + l_j$  и  $n_i^a := n_i^a + 1$ .
    }
    Присвоить  $best(j) := best(j) + 1$ .
  }
  По матрице  $X$  построить матрицу  $D$ , выбрав для  $i \in N, j \in M$  узел  $d_{ij}$ 
  с минимальным временем обработки запроса  $q_{ij}$ .
}

```

На вход процедуры *OrderReplications* подаются базовые компоненты модели и матрица X , значение которой получено при решении первой подзадачи. На выходе процедуры вычисляются матрицы T , ΔT и Y . В матрице T элементами столбца j в убывающем порядке являются времена обработки всех запросов к объекту $j \in M$ при условии пошагового раз-

мещения на узлах одной копии объекта, двух копий, трех копий и т. д. В матрице ΔT элемент j -го столбца есть сокращение времени обработки всех запросов к объекту $j \in M$, полученное за счет введения очередной копии, по сравнению с временем, обусловленным использованием ранее введенных копий объекта. В матрице Y элемент y_{ij} показывает уровень вклада репликации объекта j на узле i в уменьшение времени обработки всех запросов к объекту j . Процедура *OrderReplications* использует следующие промежуточные величины: e – шаг размещения репликаций; $N_j(e)$ – множество узлов, на которых размещены копии объекта j на шаге e ; $\tau_{ij}(e)$ – минимальное время обработки всех запросов к объекту j , поступивших на узел i , вычисленное на шаге e ; $\Delta T_j^{max}(e)$ – максимальное сокращение времени обработки всех запросов к объекту j , вычисленное на шаге e ; $maxN \in NN_j(e-1)$ – узел, размещение на котором очередной репликации дает максимальное сокращение времени обработки запросов к объекту j на шаге e .

На вход процедуры *AssignReplications* подаются исходные компоненты L и S модели, а также матрицы Y и ΔT , полученные при выполнении процедуры *OrderReplications*. На выходе процедуры вычисляются матрицы X и D , а также векторы S^a , n^a . Процедура *AssignReplications* использует в качестве промежуточной величину $best(j)$ – наилучший узел для размещения объекта j на очередной итерации цикла размещения репликаций. Истинность высказывания «возможно введение репликаций объектов» зависит от двух факторов: наличия на узлах свободного пространства памяти для размещения новых репликаций и наличия потенциальных репликаций, способных сократить суммарное время обработки запросов. Работа процедуры *AssignReplications* завершается построением матрицы D , которая вычисляется по матрице X , определяющей размещение репликаций всех объектов, при этом для группы запросов q_{ij} выбирается по возможности та репликация $d(i,j)$, которая минимизирует время обработки запросов из данной группы.

Поясним смысл матриц T , ΔT и Y на примере (рис. 2, а), который является продолжением примера из рис. 1. Так, объект $j = 1$ размещается согласно решению первой подзадачи на узле $i = 2$, при этом время обработки всех запросов к объекту согласно матрице Y составляет 2,232 с. За счет введения первой репликации объекта на узле 3 время может быть сокращено до 1,295 с. Размещение второй репликации на узле 1 способно сократить время до 1,093 с. На пересечении столбца 1 и строки 2 матрицы ΔT стоит символ « \rightarrow » по той причине, что на узле $i = 2$ размещается сам объект $j = 1$. Величина 1,263 мс/байт в третьем элементе первого столбца есть частное от деления сокращения времени обработки запросов к объекту $j = 1$, полученного за счет введения первой репликации объекта на третьем узле, на длину этого объекта. Аналогично вычисляются остальные элементы матрицы ΔT . Первый столбец матрицы Y показывает, что с целью нахождения оптимума сам объект $j = 1$ должен размещаться на узле $i = 2$, его первая репликация – на узле $i = 3$, вторая репликация – на узле $i = 1$.

На рис. 2, б матрица X определяет размещение не только самих объектов, но и всех их репликаций. В большинстве случаев репликации введены в соответствии с градиентной матрицей ΔT и матрицей предпочтений Y . Столбец 3 матрицы X иллюстрирует ситуацию, когда на размещение репликаций влияет также ограниченный объем памяти на узлах. Репликация объекта 3 должна быть размещена согласно максимальному сокращению времени обработки на узле 3, однако из-за отсутствия свободной памяти она размещена на узле 1. Порядок введения репликаций описывается последовательностью пар объект \rightarrow узел: 1 \rightarrow 3, 5 \rightarrow 3, 2 \rightarrow 3, 9 \rightarrow 2, 6 \rightarrow 2, 1 \rightarrow 1, 3 \rightarrow 1. Матрица D показывает для всех запросов обрабатывающие их узлы. Вектор n^a определяет количество объектов на каждом из узлов при общем количестве копий объектов 18. Вектор S^a показывает, насколько использованы ресурсы памяти, предоставляемые узлами. Абсолютная и относительная загрузки узлов и каналов рассчитаны по формулам (7)–(14), при этом следует отметить, что использование репликаций позволило уменьшить загрузку каналов и увеличить загрузку узлов. В итоге получены следующие параметры функционирования оптимизированной распределенной системы: среднее время обработки запроса $T_{request} = 4,8$ мс; время обработки всех запросов $T_{total} = 1,94$ с; средняя загрузка узла 86,3 %; средняя загрузка канала 22,1 %.

$$T = \begin{bmatrix} 1,093 & 0,754 & 0,653 & 0,661 & 1,104 & 0,444 & 0,165 & 1,390 & 0,660 & 0,353 \\ 2,232 & 0,597 & 1,015 & 0,435 & 1,928 & 0,491 & 0,165 & 1,031 & 0,718 & 0,309 \\ 1,295 & 0,602 & 0,712 & 0,000 & 1,162 & 0,663 & 0,165 & 0,681 & 1,075 & 0,278 \end{bmatrix}$$

$$\Delta T = \begin{bmatrix} 0,272 & - & 0,153 & - & 0,082 & 0,122 & 0,000 & - & 0,108 & - \\ - & 0,030 & - & 0,357 & - & 0,455 & 0,000 & 0,645 & 0,666 & 0,113 \\ 1,263 & 0,815 & 0,795 & 0,000 & 1,089 & - & - & 0,630 & - & 0,081 \end{bmatrix}$$

$$Y = \begin{bmatrix} 3 & 1 & 3 & 1 & 3 & 3 & 0 & 1 & 3 & 1 \\ 1 & 3 & 1 & 2 & 1 & 2 & 0 & 2 & 2 & 2 \\ 2 & 2 & 2 & 0 & 2 & 1 & 1 & 3 & 1 & 3 \end{bmatrix}$$

а)

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 3 & 1 & 2 & 1 \\ 2 & 3 & 2 & 1 & 2 & 2 & 3 & 1 & 2 & 1 \\ 3 & 3 & 2 & 1 & 3 & 3 & 3 & 1 & 3 & 1 \end{bmatrix}$$

$$n^a = [6 \quad 5 \quad 6]$$

$$S^a = [2885 \quad 2740 \quad 2582]$$

$$Lnd = [1,94 \quad 1,37 \quad 1,71]$$

$$Lnd^{\%} = [100,0 \quad 70,8 \quad 88,1]$$

$$Lch = \begin{bmatrix} 0,0 & 0,54 & 0,05 \\ 1,27 & 0,0 & 0,13 \\ 0,36 & 0,23 & 0,0 \end{bmatrix}$$

$$Lch^{\%} = \begin{bmatrix} 0,0 & 28,1 & 2,4 \\ 65,4 & 0,0 & 6,6 \\ 18,6 & 11,9 & 0,0 \end{bmatrix}$$

б)

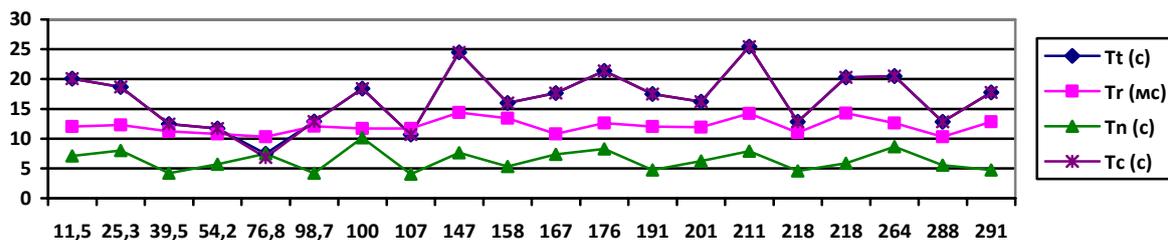
Рис. 2. Результаты работы процедур: а) *OrderReplications*;
 б) *AssignReplications* совместно с оценками загрузки узлов и каналов

6. Результаты вычислительных экспериментов

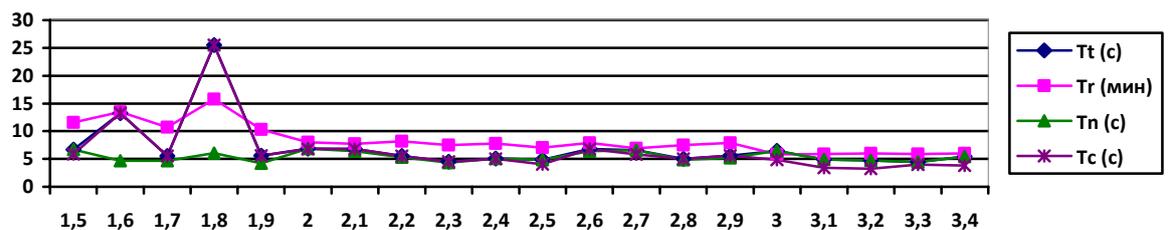
Алгоритм оптимизации размещения объектов и их репликаций на узлах распределенной системы реализован в виде программы, написанной на языке C/C++. Программа положена в основу вычислительных экспериментов, выполненных посредством генерации всех компонентов модели с использованием датчика случайных чисел и варьирования параметров модели в широких пределах.

Основные закономерности и тренды, выявленные в поведении и параметрах распределенной системы, которая построена посредством выполнения алгоритма оптимального распределения объектов и их репликаций на узлах, представлены на рис. 3. Время Tt обработки всех запросов, являющееся максимумом узлового времени Tn и канального времени Tc , и среднее время Tr 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 (рис. 3, а). Увеличение суммарной емкости памяти на узлах, превышающей суммарную длину объектов с коэффициентом от 1,5 до 3,4 раз, уменьшает среднее время Tr обработки одного запроса в 1,38 раза и общее время Tt обработки всех запросов в 1,88 раза (рис. 3, б). Средняя относительная загрузка узлов Tn составила 62,03 %, в то время как средняя относительная загрузка каналов Tc – 11,10 %. Очевидно то, что многие каналы загружены в среднем слабо. Заметим, что аналогичное соотношение не выполняется относительно максимальных загрузок узлов и каналов. С увеличением с 48 до 356 средней частоты запросов к одному объекту с одного узла (т. е. в 7,42 раза) среднее время Tr обработки одного запроса увеличилось лишь в 1,21 раза, в то время как общее время Tt , совпадающее большей частью с временной загрузкой узлов Tn , возросло существенно – в 6,07 раза (рис. 3, в). Временная загрузка каналов Tc , так же, как и временная загрузка узлов Tn , увеличилась значительно, однако не явилась доминирующей в Tt . Средняя относительная загрузка узлов составила 68,25 %, каналов – 30,22 %. Увеличение задержек в каналах в 3,76 раза вызвало увеличение среднего времени Tr в 2,47 раза и общего времени Tt в 2,66 раза (рис. 3, г). Временная загрузка узлов Tn и каналов Tc также увеличилась, при этом Tc стало доминирующим в Tt . Средняя относительная загрузка узлов составила 61,45 %, каналов – 39,13 %.

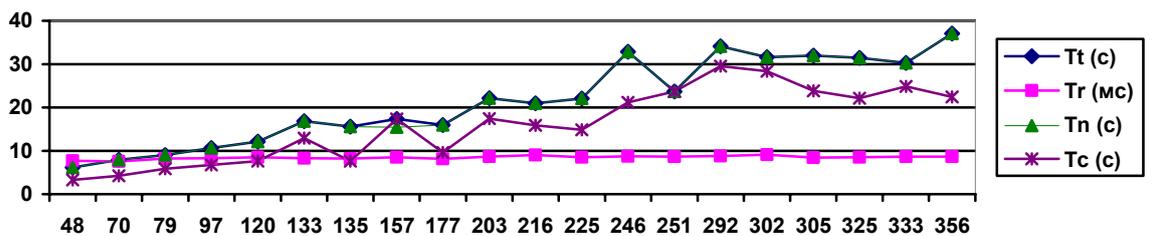
Проведенные эксперименты позволяют сделать следующий вывод: и узлы и каналы загружаются неравномерно без использования репликаций объектов при назначении их на узлы. С введением репликаций нагрузка на узлы возрастает, в то же время нагрузка на каналы уменьшается.



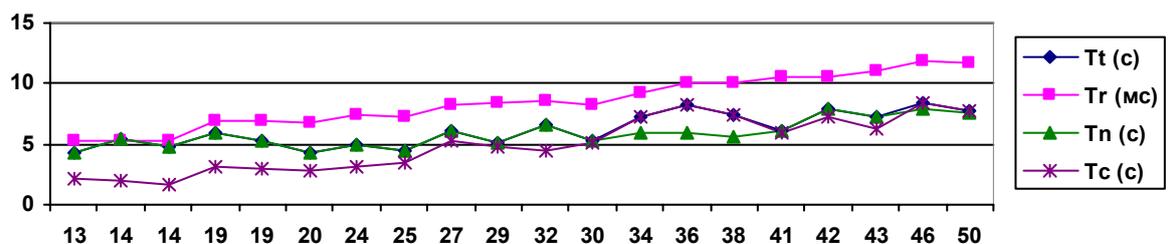
а)



б)



в)



з)

Рис. 3. Зависимости параметров функционирования распределенной системы, оптимизированной с использованием репликаций, от следующих величин: а) разброса длин объектов; б) отношения «объем памяти узлов / суммарная длина объектов»; в) частоты запросов в узлах; з) задержек в каналах передачи данных

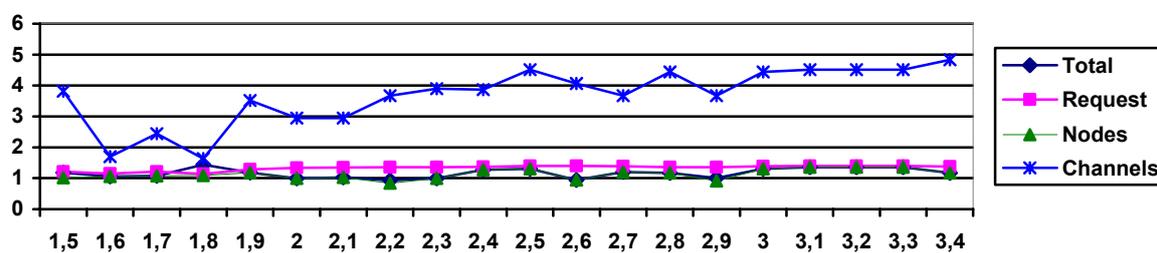
7. Эксперименты по оценке эффективности введения репликаций

Сопоставление параметров функционирования распределенной системы, построенной с использованием технологии репликаций, с параметрами системы, построенной без использования технологии репликаций, является базой для обоснования применения технологии репликаций. Сравним параметры двух реализаций системы в зависимости от изменения трех параметров модели: емкости памяти, частоты запросов и производительности каналов.

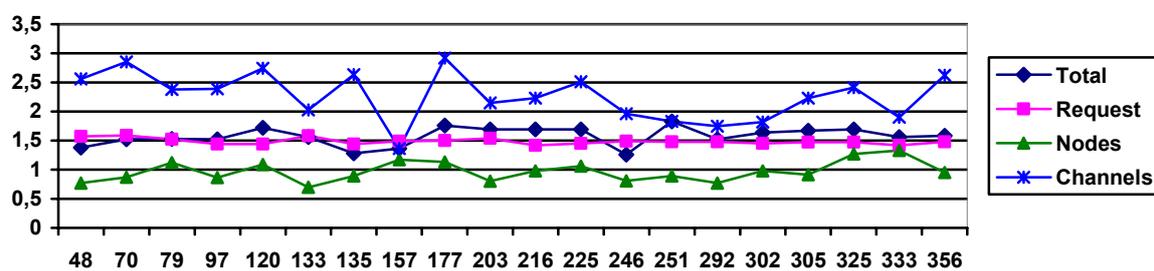
При увеличении коэффициента «объем памяти узлов / суммарная длина объектов» до 2,27 общее время Tt обработки запросов уменьшилось за счет введения репликаций в среднем в 1,16 раза, а среднее время Tr обработки одного запроса сократилось в 1,33 раза (рис. 4, а). При этом средняя относительная загрузка узлов уменьшилась в 1,13 раза, каналов – в 3,68 раза. Очевидно, что увеличение объема памяти и числа репликаций разгружает каналы.

При увеличении средней частоты запросов в 7,42 раза общее время Tt уменьшилось в среднем в 1,57 раза благодаря использованию репликаций, а среднее время Tr сократилось в 1,49 раза. При этом средняя относительная загрузка узлов увеличилась в 1,03 раза, загрузка каналов уменьшилась в 2,26 раза (рис. 4, б).

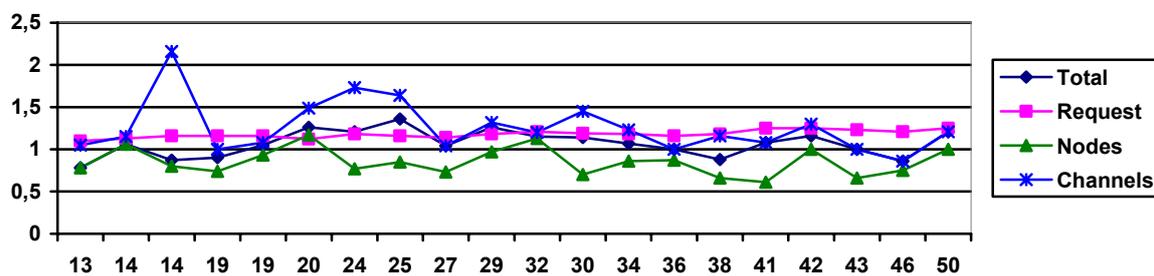
При снижении средней пропускной способности каналов передачи данных в 3,76 раза общее время Tt обработки запросов с использованием репликаций уменьшилось в 1,07 раза по сравнению с временем, соответствующим отсутствию репликаций, а среднее время Tr обработки одного запроса сократилось в 1,18 раза. Благодаря использованию репликаций средняя загрузка узлов возросла в 1,18 раза, загрузка каналов снизилась в 1,26 раза (рис. 4, в).



а)



б)



в)

Рис. 4. Коэффициенты изменения параметров распределенной системы при переходе от размещения объектов без репликаций к размещению объектов с репликациями в зависимости от следующих величин: а) отношения «объем памяти узлов / суммарная длина объектов»; б) частоты запросов в узлах; в) производительности каналов передачи данных

Общий вывод состоит в том, что технология репликаций существенно сокращает среднее время обработки одного запроса, разгружая каналы передачи данных и увеличивая загрузку узлов.

Заключение

Предложена модель размещения объектов и их репликаций на узлах распределенной информационно-вычислительной системы, ориентированная на оценку и оптимизацию базовой составляющей временных параметров функционирования системы без учета времен ожидания, обусловленных случайным характером процесса поступления и совместной обработки всех запросов. Разработан алгоритм наискорейшего спуска для минимизации среднего времени обработки одного запроса к объекту, выполнена его программная реализация. Проведена серия вычислительных экспериментов, выявивших основные закономерности и тренды в изменениях параметров функционирования всей системы, вызванных введением репликаций объектов на узлах при вариациях таких параметров модели, 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. Дальнейшее развитие модели и алгоритмов будет направлено на учет других критериев оптимизации, а также на выявление влияния случайных процессов, имеющих место при поступлении и обработке запросов, на параметры функционирования распределенной системы.

Список литературы

1. Zhuo, L. Load Balancing in Distributed Web Server Systems with Partial Document Replication / L. Zhuo, C.-L. Wang, F.C. M. Lau // Proc. of the 2002 International Conference on Parallel Processing. – IEEE Computer Society, Washington, 2002. – P. 305–312.
2. Khan, S.U. Comparison and analysis of ten static heuristics-based Internet data replication techniques / S.U. Khan, I. Ahmad // Journal of Parallel and Distributed Computing. – 2008. – Vol. 68, № 2. – P. 113–136.
3. Прихожий, А.А. Эффективное программирование распределенной системы web-серверов на базе оптимизации размещения и репликации web-объектов / А.А. Прихожий, Т.Л. Труханович // Web-программирование и интернет-технологии (WebConf-09) : материалы Междунар. конф, Минск, БГУ, 8–10 июня 2009 г. – Минск : Ин-т математики НАН Беларуси, 2009. – С. 12–15.
4. Листопад, Н.И. Моделирование и оптимизация глобальных сетей / Н.И. Листопад. – Минск : Изд-во БГУ, 2000. – 156 с.
5. Прихожий, А.А. Распараллеливание и планирование вычислительных и информационных процессов / А.А. Прихожий, М.В. Соломенник // Доклады БГУИР. – 2003. – № 4(4). – С. 104–114.

Поступила 02.04.10

*Высший государственный колледж связи,
Минск, ул. Ф. Скарныны, 8/2
e-mail: prihozhy@yahoo.com*

A.A. Prihozhy

OPTIMIZATION OF OBJECT ALLOCATION FOR DISTRIBUTED SYSTEM NODES IN THE MODEL WITH REPLICATION

A model of allocation of the objects and their replications for the distributed system nodes which takes into account the object attributes, request flows, node and channel parameters is proposed. The object allocation optimization task minimizing the average processing time per request with a constraint on the total execution time of all requests is formulated. A gradient algorithm for solving the task is developed. The optimization program is created which presents a basis for conducting experiments, comparative study of the distributed systems optimized with and without replications, and finding out the rules and trends in system parameter variations.