

# ЗАДАЧА СИНТЕЗА ОПТИМАЛЬНОЙ МОДУЛЬНОЙ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ РЕАЛЬНОГО ВРЕМЕНИ С ДВУХЗВЕННОЙ АРХИТЕКТУРОЙ «КЛИЕНТ — СЕРВЕР»

С.А. Косяченко, С.С. Ковалевский, С.К. Сомов

Рассмотрена задача синтеза модульной системы обработки данных реального времени с архитектурой «Клиент — Сервер», реализуемой на основе распределенной вычислительной сети, особенности решения которой состоят в следующем: при определении времени реакции системы на заявку необходимо учитывать время доставки заявки от клиентской машины («Клиента») к «Серверу» и время доставки результата обработки заявки от «Сервера» к «Клиенту»; время обмена сообщениями между «Клиентом» и «Сервером» зависит от пропускной способности каналов передачи данных, объемов и трафика сообщений в сети.

**Ключевые слова:** задача, данные, система, сеть.

## ВВЕДЕНИЕ

В работах [1—3] сформулированы постановки и предложены методы решения задач синтеза оптимальных модульных систем обработки данных (СОД) реального времени, использующих при обработке заявок приоритетные дисциплины обслуживания и функционирующих на базе одного компьютера.

При постановке и решении аналогичной задачи синтеза СОД реального времени с двухзвенной архитектурой «Клиент — Сервер», реализуемой не на одном компьютере, а в среде распределенной вычислительной сети, необходимо принимать во внимание следующие особенности функционирования такой системы:

— при определении времени реакции системы на запрос следует учитывать время доставки запроса от клиентской машины («Клиента») к «Серверу» и ответа на запрос от «Сервера» к клиентской машине, так как они в общем случае располагаются в различных узлах сети;

— время обмена сообщениями между клиентскими машинами и «Сервером» зависит от пропускной способности каналов передачи данных, от

объемов передаваемых сообщений (запросов и ответов на запросы) и от имеющегося в сети трафика сообщений, не связанных с работой системы с архитектурой «Клиент — Сервер» и которыми обмениваются другие узлы сети. На базе данной вычислительной сети, помимо системы с архитектурой «Клиент — Сервер», могут функционировать и другие информационно-справочные системы, базы данных, системы обработки данных, порождающие свой трафик сообщений по каналам сети;

— вероятность успешной доставки запроса и ответа на него от одного узла сети к другому может отличаться от единицы в силу конечной надежности каналов связи и процессов обработки передаваемых сообщений в узлах коммутации сообщений;

— для размещения «Сервера» необходимо выбрать один из компьютеров сети, который обеспечивает оптимальное (или наиболее близкое к оптимальному) значение критерия оптимальности, используемого при решении задачи;

— при выборе компьютера сети для размещения сервера следует учитывать его физические характеристики, такие как, быстродействие, тип используемой внешней памяти, объем оперативной памяти и др.

С учетом данных особенностей и результатов, полученных в работах [2, 3], сформулируем задачу синтеза программного и информационного обеспечения СОД реального времени, имеющую двухзвенную архитектуру «Клиент — Сервер» и функционирующую в среде распределенной вычислительной системы.

### 1. ФОРМАЛИЗАЦИЯ ИСХОДНЫХ ДАННЫХ ЗАДАЧИ СИНТЕЗА

Предположим, что в СОД реального времени с архитектурой «Клиент — Сервер» поступает  $J$  пуассоновских потоков разнотипных заявок (запросов) с различными приоритетами  $j = \overline{1, J}$ , интенсивностями  $(\lambda_1, \dots, \lambda_j, \dots, \lambda_J)$  и функциями длительностей обслуживания с двумя конечными моментами  $\bar{a}_j$  и  $\bar{a}_j^{(2)}$ ,  $j = \overline{1, J}$ .

В соответствии с некоторым критерием оптимизации необходимо:

— выбрать узел сети, на базе которого должен функционировать «Сервер»;

— синтезировать систему прикладных программных модулей и информационных массивов СОД реального времени, необходимых для обработки множества поступающих запросов, размещаемых в данном узле сети («Сервере»).

Предполагается, что топология сети задана, т. е. в соответствии с некоторой процедурой определены маршруты, по которым передаются запросы от клиентских машин к «Серверу», расположенному в одном из узлов сети, и ответы на запросы от «Сервера» к клиентским машинам.

При решении рассматриваемой задачи необходимо соблюдать технологические ограничения на систему программных модулей и информационных массивов.

Для формализации постановки задачи введем следующие обозначения и переменные:

$E = \{\varepsilon_1, \dots, \varepsilon_j, \dots, \varepsilon_J\}$  — множество типов заявок, обрабатываемых в СОД реального времени ( $\varepsilon_j$  — заявка  $j$ -го типа);

$P = \{p_1, \dots, p_r, \dots, p_R\}$  — множество процедур обработки данных, используемых при обработке заявок всех типов;

$D = \{d_1, \dots, d_p, \dots, d_L\}$  — множество информационных элементов (входных, промежуточных и выходных), используемых при обработке заявок всеми процедурами множества  $P$ ;

$D' = \{d'_1, \dots, d'_j, \dots, d'_L\}$  — вектор размерностей (объемов) информационных элементов множества  $D$ ;

$D'' = \{d''_1, \dots, d''_j, \dots, d''_L\}$  — количество различных значений (мощность множества различных

значений) информационных элементов множества  $D$ ;

$H = \{\eta_1, \dots, \eta_n, \dots, \eta_N\}$  — множество задач, решаемых при обработке заявок;

$\bar{C}_{nl}^{r(c/3)}$  — среднее количество значений  $l$ -го информационного элемента, которое считывается/записывается  $r$ -й процедурой при решении  $n$ -й задачи;

$\bar{C}_{nl}^{r(c/3)}$  — максимальное количество значений  $l$ -го информационного элемента, которое считывается/записывается  $r$ -й процедурой при решении  $n$ -й задачи;

$\Omega$  и  $F$  — число программных модулей (ПМ) и число информационных массивов (ИМ) в СОД реального времени;

$z_{fj} =$

$$= \begin{cases} 1, & \text{если информационный элемент } d_l \text{ входит} \\ & \text{в состав информационного массива } f, \\ 0 & \text{в противном случае;} \end{cases}$$

$$w_{nl}^{r(c/3)} = \begin{cases} 1, & \text{если информационный элемент } d_l \\ & \text{считается/записывается процедурой } p_r \\ & \text{задачи } \eta_n, \\ 0 & \text{в противном случае;} \end{cases}$$

$$X_{r\omega} = \begin{cases} 1, & \text{если процедура } r \text{ входит} \\ & \text{в состав модуля } \omega, \\ 0 & \text{в противном случае;} \end{cases}$$

$\gamma_{l\omega}^{n(c/3)}$  — переменная, характеризующая взаимосвязь  $n$ -й задачи с множеством процедур  $\{p_r\}$  и программным модулем  $\omega$  по использованию  $j$ -го информационного элемента для считывания и записи:

$$\gamma_{l\omega}^{n(c/3)} = \begin{cases} 1, & \text{если } \sum_{r=1}^R w_{nl}^{r(c/3)} b_{nr} x_{r\omega} \geq 1, \\ 0, & \text{если } \sum_{r=1}^R w_{nl}^{r(c/3)} b_{nr} x_{r\omega} = 0 \end{cases}$$

(она принимает значение 1, если  $l$ -й информационный элемент считывается/записывается при работе  $\omega$ -го модуля во время решения  $n$ -й задачи);

$e_m^{(c/3)}$  — скорость считывания/записи информации с/на устройство внешней памяти  $m$ -го узла вычислительной сети;



$t_m^*$  — среднее время поиска локальной СУБД информации (ИМ или ПМ) во внешней памяти  $m$ -го узла вычислительной сети;

$\beta = (\beta_1, \dots, \beta_P, \dots, \beta_M)$  — вектор переменных задачи, определяющий размещение «Сервера» в узлах распределенной вычислительной сети;  $\beta_i = 1$ , если «Сервер» размещен и функционирует на базе  $i$ -го узла сети, и  $\beta_i = 0$  в противном случае; очевидно, что должно выполняться технологическое ограничение:

$$\sum_{k=1}^M \beta_k = 1,$$

означающее, в соответствии с выбранной архитектурой СОД реального времени, наличие в сети только одного сервера, обрабатывающего заявки клиентов;

$l_j^q$  — средняя длина запроса  $j$ -го типа,  $l_j^r$  — средняя длина сообщения — ответа на запрос  $j$ -го типа;

$$\lambda_j = \sum_{i=1}^M \lambda_i^j \quad (1)$$

суммарный поток заявок  $j$ -го типа,  $j = \overline{1, J}$ , поступающих на сервер от всех узлов — клиентов системы;

$\pi_{ik}$  — путь (маршрут), по которому передаются сообщения из узла  $i$  в узел  $k$ . Канал связи  $\phi$  включен в путь  $\pi_{ik}$ , если сообщения, идущие по этому пути, проходят через этот канал связи. При использовании фиксированной процедуры маршрутизации для передачи сообщений в сети можно считать, что задано множество матриц  $X^\phi = \|x_{ik}^\phi\|$ ,  $\phi = \overline{1, \Phi}$ ,  $i, k = \overline{1, M}$ , в которых элемент  $x_{ik}^\phi = 1$ , если канал  $\phi$  входит в состав пути  $\pi_{ik}$ .

Определим величину  $T^*$  — среднюю задержку передачи сообщения в сети (заявки «Клиента» и результата обработки заявки «Сервером»), когда в сети применяется метод коммутации сообщений (в случае метода коммутации пакетов оценка  $T^*$  осуществляется аналогично).

Показано, что в среднем в единицу времени из  $i$ -го узла сети в  $k$ -й узел сети отправляется трафик  $\lambda_{ik}$  сообщений с интенсивностью  $\lambda_{ik} = \lambda_{ik}^q + \lambda_{ik}^r$  [4].

С учетом уже имеющегося в сети трафика  $\delta_{ik}$  сообщений, порождаемых другими информационны-

ми системами со средней длиной  $l^*$ , трафик всех сообщений из узла  $i$  в узел  $k$  в единицу времени

$$\gamma_{ik} = \delta_{ik} + \lambda_{ik}^q + \lambda_{ik}^r = \delta_{ik} + \beta_k \sum_{j=1}^J \lambda_i^j + \beta_i \sum_{j=1}^J \lambda_k^j.$$

Трафик всех сообщений, которыми обмениваются между собой узлы сети в среднем в единицу времени

$$\begin{aligned} \gamma &= \sum_{i=1}^M \sum_{k=1}^M \gamma_{ik} = \\ &= \sum_{i=1}^M \sum_{k=1}^M \left( \delta_{ik} + \beta_k \sum_{j=1}^J \lambda_i^j + \beta_i \sum_{j=1}^J \lambda_k^j \right). \end{aligned} \quad (2)$$

Средняя длина сообщений, передающихся в сети,

$$\begin{aligned} 1/\mu &= \\ &= \sum_{i=1}^M \sum_{k=1}^M \frac{1}{\gamma} \left( \delta_{ik} l^* + \beta_k \sum_{j=1}^J \lambda_i^j l_j^q + \beta_i \sum_{j=1}^J \lambda_k^j l_j^r \right). \end{aligned} \quad (3)$$

В единицу времени через канал  $\phi$  проходит поток сообщений со средней интенсивностью

$$\lambda_\phi^* = \sum_{i=1}^M \sum_{k=1}^M \gamma_{ik} x_{ik}^\phi. \quad (4)$$

Для определения средней задержки  $T^*$  передачи сообщения в сети воспользуемся результатами, полученными ранее для сети с коммутацией сообщений в предположении, что трафик сообщений, возникающих в узлах сети, образует пуассоновский процесс со средним значением  $\gamma_{ik}$ , ( $i, k = \overline{1, M}$ ) сообщений в секунду. Трафик всех сообщений определяется по формуле (2). Каждое сообщение имеет независимую длину, значение которой распределено по показательному закону со средним значением  $\mu^{-1}$  бит, которое определяется по формуле (3). Узлы сети имеют память неограниченной емкости для размещения этих сообщений (т. е. сообщения не теряются при прохождении по каналам сети). Сообщения направляются от узла к узлу в соответствии с фиксированной процедурой выбора маршрутов.

Обозначим через  $T_\phi^*$  среднее время, затраченное сообщением на ожидание и процесс передачи по  $\phi$ -му каналу связи (обслуживания в канале). Если этот канал рассматривать как систему обслуживания, то время  $T_\phi^*$  — это время, проведенное сообщением в данной системе (время в очереди на обслуживание и время самого обслуживания).

С учетом сделанных ранее предположений, включая предположение об абсолютной надежности каналов связи сети, величина  $T_\varphi^*$  будет определяться по формуле [1]  $T_\varphi^* = 1/(\mu C_\varphi - \lambda_\varphi^*)$ , а

$$T^* = \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\gamma} T_\varphi^* = \frac{1}{\gamma} \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\mu C_\varphi - \lambda_\varphi^*}. \quad (5)$$

Если известно среднее время  $K$  обработки сообщения в узле коммутации при его передаче по каналу связи и время  $P_\varphi$  распространения сигнала в канале связи, то формулу (5) можно записать как

$$T^* = \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\gamma} T_\varphi^* = \frac{1}{\gamma} \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\mu C_\varphi - \lambda_\varphi^*}.$$

Если предположить, что каналы связи ненадежны, то с вероятностью  $\gamma_\varphi^* = 1 - \rho_\varphi^*$  в  $\varphi$ -м канале может возникнуть ошибка при передаче единицы данных. Возникновение ошибки приводит к тому, что через время  $t_{ACK}$  узел-отправитель не получает квитанцию (подтверждение успешной передачи сообщения) и повторно передает сообщение по каналу. Это приводит к увеличению общего времени обслуживания сообщения в канале связи.

Вероятность возникновения ошибки при передаче по  $\varphi$ -му каналу одного сообщения (запроса или ответа на запрос), которое имеет среднюю длину  $l = \mu^{-1}$  бит, будет определяться выражением

$$\gamma_\varphi^* = 1 - \rho_\varphi = 1 - (1 - \Upsilon_\varphi^*)^l = 1 - (\rho_\varphi^*)^l.$$

В итоге среднее время передачи сообщения по ненадежному  $\varphi$ -му каналу связи

$$T_\varphi = \rho_\varphi T_\varphi^* + (t_{ACK} + \rho_\varphi T_\varphi^*) \Upsilon_\varphi (1 - \gamma_\varphi)^{-1}.$$

Среднее время передачи сообщения из узла-отправителя в узел-получатель по сети с ненадежными каналами

$$\begin{aligned} T^* &= \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\gamma} T_\varphi = \\ &= \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\gamma} \{ \rho_\varphi T_\varphi^* + (t_{ACK} + \rho_\varphi T_\varphi^*) \Upsilon_\varphi (1 - \gamma_\varphi)^{-1} \}. \end{aligned} \quad (6)$$

С учетом среднего времени  $K$  обработки сообщения в узле коммутации и времени  $P_\varphi$  распространения сигнала в канале связи, среднее время

передачи сообщения по сети с ненадежными каналами

$$\begin{aligned} T^* &= \sum_{\varphi=1}^{\Phi} \frac{\lambda_\varphi^*}{\gamma} \{ \rho_\varphi (T_\varphi^* + K + P_\varphi) + \\ &+ (t_{ACK} + \rho_\varphi [T_\varphi^* + K + P_\varphi]) \Upsilon_\varphi (1 - \gamma_\varphi)^{-1} \}. \end{aligned}$$

Среднее  $\bar{a}_m^j$  и максимальное  $\bar{\bar{a}}_m^j$  времена обработки заявки  $\varepsilon_j$  типа «Сервером», размещенным в  $m$ -м узле сети, включает в себя: процессорное время выполнения задач, необходимость в решении которых определяется типом заявки; время поиска и обмена с внешней памятью при считывании и записи информации, необходимой для решения задач; время поиска во внешней памяти и считывания в оперативную память программных модулей, содержащих процедуры обработки данных, используемые при решении задач. Значения среднего и максимального времен обработки заявки  $j$ -го типа оцениваются в соответствии с формулами:

$$\bar{a}_m^j = \sum_{v=1}^V q_v^j \bar{\sigma}_{vm}^j, \quad \bar{\bar{a}}_m^j = \max_{v=1, \bar{V}} \bar{\bar{\sigma}}_{vm}^j.$$

Среднее  $\bar{\sigma}_{vm}^j$  и максимальное время  $\bar{\bar{\sigma}}_{vm}^j$  реализации пути  $\pi_v^j$  при обработке заявки  $j$ -го типа на сервере  $m$  (время решения задач, входящих в состав данного пути) равно сумме соответственно средних и максимальных времен решения всех задач, входящих в состав этого пути, и определяются по формулам:

$$\begin{aligned} \bar{\sigma}_{vm}^j &= \sum_{n=1/x_n^j \in X_v^j}^N \bar{t}_m^n, & \bar{\bar{\sigma}}_{vm}^j &= \sum_{n=1/x_n^j \in X_v^j}^N \bar{\bar{t}}_m^n, \\ & & v &= \bar{1}, \bar{V}. \end{aligned}$$

Среднее время  $\bar{t}_m^n$  и максимальное время  $\bar{\bar{t}}_m^n$  решения задачи  $\eta_n$  в  $m$ -м узле вычислительной сети включают в себя процессорное время выполнения процедур обработки данных, используемых при решении задачи; время поиска и обмена с внешней памятью при считывании и записи информации, необходимой при выполнении процедур; время поиска и считывания в оперативную память программных модулей, необходимых для выполнения процедур. Значения среднего и максимального времен решения задачи оцениваются по формулам [1]:

$$\bar{t}_m^n = \sum_{v=1}^V q_v^n \bar{\tau}_{vm}^n, \quad \bar{\bar{t}}_m^n = \max_{v=1, \bar{V}} \bar{\bar{\tau}}_{vm}^n,$$



где  $V$  — мощность множества путей  $\Pi^n = \{\pi_v^n\}$  графа  $\widehat{G}^n$  решения задачи  $\eta_n$ .

Среднее время  $\bar{H}_j$  отклика «Сервера» системы (среднее время обработки заявки сервером и среднее время ожидания заявки в очереди на обработку) для заявки  $j$ -го типа определим в виде

$$\bar{H}_j = \sum_{m=1}^M \beta_m (Q_j^m + \bar{a}_j^m), \quad (7)$$

где  $\bar{a}_j^m$  — среднее время обработки заявки  $\varepsilon_j$  сервером, который размещен в  $m$ -м узле сети. Слагаемое  $Q_j^m$  представляет собой среднее время ожидания заявки  $j$ -го типа в очереди на обслуживание «Сервером», размещенным в  $m$ -м узле сети, и определяется как [3]:

$$Q_j^m = \left( \sum_{j=1}^J \lambda_j \bar{a}_j^{m(2)} \right) [2(1 - \upsilon_{j-1}^m)(1 - \upsilon_j^m)]^{-1}, \quad (8)$$

$$\upsilon_j^m = \sum_{i=1}^j \lambda_i \bar{a}_i^m.$$

Здесь  $\lambda_j$  — суммарный поток заявок  $\varepsilon_j$ , ( $j = \overline{1, J}$ ), поступающих на сервер от всех клиентов системы, который рассчитывается по формуле (1),  $\bar{a}_j^{m(2)}$  — второй начальный момент времени обработки заявки  $j$ -го типа.

При показательном распределении длительностей обслуживания заявок  $\bar{a}_j^{m(2)} = 2\bar{a}_j^m$ , а если время обслуживания постоянно, то  $\bar{a}_j^{m(2)} = \bar{a}_j^{m2}$ .

## 2. ПОСТАНОВКА ЗАДАЧИ СИНТЕЗА И МЕТОД ЕЕ РЕШЕНИЯ

Будем рассматривать стационарный режим функционирования системы, когда все поступающие на интервале времени  $[0, T]$  заявки обслуживаются системой (т. е. загрузка системы

$$\sum_{j=1}^J \lambda_j \bar{a}_j^m < 1).$$

Суммарная интенсивность поступления в систему заявок всех типов

$$\lambda = \sum_{j=1}^J \lambda_j,$$

где  $\lambda_j$  — интенсивность поступления заявок  $j$ -го типа. За время  $T$  в систему поступит  $\lambda T$  заявок различного типа. За время  $T$  заявка  $j$ -го типа поступит в систему с вероятностью

$$P_j = \lambda_j T \left( \sum_{j=1}^J \lambda_j T \right)^{-1} = \lambda_j \lambda^{-1}. \quad (9)$$

С учетом выражений (7) и (9) получим, что среднее время отклика СОД реального времени на заявку произвольного типа

$$\begin{aligned} \bar{H} &= \sum_{m=1}^M \beta_m \sum_{j=1}^J \lambda_j \lambda^{-1} \bar{H}_j = \\ &= \sum_{m=1}^M \beta_m \sum_{j=1}^J \lambda_j \lambda^{-1} (Q_j^m + \bar{a}_j^m). \end{aligned} \quad (10)$$

Производительность СОД реального времени, т. е. среднее число заявок, которое система обрабатывает в единицу времени, определяется формулой

$$\pi = \bar{H}^{-1}. \quad (11)$$

Очевидно, что максимальное значение производительности системы будет достигнуто при минимуме значения  $\bar{H}$  — среднего отклика системы на заявку произвольного типа.

Если принять показатель (11) в качестве критерия оптимизации при решении рассматриваемой задачи, то с учетом выражений (7), (8) и (10) получим, что максимальное значение производительности системы достигается при таких значениях переменных  $\beta_m, x_{r\omega}, z_{jf}$ , которые обеспечат минимум функционала задачи

$$\min_{\{\beta_m, x_{r\omega}, z_{jf}\}} \sum_{m=1}^M \beta_m \sum_{j=1}^J \lambda_j \lambda^{-1} \left\{ \bar{a}_j^m + \frac{\sum_{j=1}^J \lambda_j \bar{a}_j^{m(2)}}{2(1 - \upsilon_{j-1}^m)(1 - \upsilon_j^m)} \right\},$$

где  $\upsilon_{j-1}^m = \sum_{i=1}^{j-1} \lambda_i \bar{a}_i^m$ ,  $\upsilon_j^m = \sum_{i=1}^j \lambda_i \bar{a}_i^m$ ; при ограничениях:

— на число узлов сети, используемых для размещения сервера:

$$\sum_{m=1}^M \beta_m = 1; \quad (12)$$

— на среднее время обслуживания заявок определенного типа (приоритета) с учетом среднего времени  $T^*$  доставки запроса на сервер для обра-

ботки и доставки результата обработки запроса обратно на клиентскую машину, которое определяется по формуле (5) или (6):

$$T^* + \bar{H}_j + T^* = 2T^* + \sum_{m=1}^M \beta_m (Q_j^m + \bar{a}_j^m) \leq T_j, \quad j = \overline{1, J}; \quad (13)$$

— на стационарность режима функционирования системы:

$$\sum_{m=1}^M \beta_m \sum_{j=1}^J \lambda_j \bar{a}_j^m < 1; \quad (14)$$

— на число  $M_{\text{пр}}$  дублируемых процедур обработки данных в синтезируемых модулях системы:

$$\left( \sum_{n=1}^N \sum_{\omega=1}^{\Omega} \sum_{r=1}^R b_{nr} x_{r\omega} \right) - R \leq M_{\text{пр}}; \quad (15)$$

— на максимальное число процедур в модуле:

$$1 < \sum_{r=1}^R x_{r\omega} \leq \bar{M}_{\omega}, \quad \omega = \overline{1, \Omega}; \quad (16)$$

— на степень дублирования информационных элементов в массивах:

$$\sum_{f=1}^F z_{if} \leq k, \quad i = \overline{1, L}; \quad (17)$$

— на размер записи каждого массива:

$$\sum_{l=1}^L z_{if} d_l^i \leq M_f, \quad f = \overline{1, F}; \quad (18)$$

— на интерфейс между отдельным модулем  $\omega'$  и другими модулями при решении задачи  $n$ :

$$\sum_{l=1}^L \sum_{\omega=1}^{\Omega} (\gamma_{l\omega}^{nc} \vee \gamma_{l\omega}^{n3}) (\gamma_{l\omega'}^{nc} \vee \gamma_{l\omega'}^{n3}) < S_{\omega}^n, \quad \omega \neq \omega'; \quad (19)$$

— на число информационных элементов, используемых программными модулями для чтения и/или записи при решении задачи  $n$ :

$$\sum_{l=1}^L (\gamma_{l\omega}^{nc} \vee \gamma_{l\omega}^{n3}), \quad \omega = \overline{1, \Omega}. \quad (20)$$

Отметим, что ряд ограничений данной задачи (например, (15) и (17)) могут отсутствовать по объективным причинам.

Таким же образом можно сформулировать рассматриваемую задачу, когда в качестве целевой

функции принят максимум коэффициента полезной работы системы.

Под коэффициентом полезной работы СОД реального времени будем понимать долю времени «Сервера», потраченную на обработку данных, требуемых при обслуживании заявки, от общего времени пребывания заявки в системе с учетом затрат времени на передачу данных по каналам связи сети. Для заявки  $j$ -го типа этот коэффициент

$$K_{nj} = \left( \sum_{m=1}^M \beta_m \bar{a}_j^m \right) / \left( \sum_{m=1}^M \beta_m (Q_j^m + \bar{a}_j^m + 2T^*) \right).$$

При обслуживании заявки произвольного типа коэффициент полезной работы равен

$$K_n = \lambda^{-1} \sum_{j=1}^J \lambda_j K_{nj}.$$

Критерий оптимальности синтеза СОД реального времени при заданных условиях имеет вид:

$$\max_{\{\beta_m, x_{r\omega}, z_{if}\}} K_n = \max \sum_{j=1}^J \frac{\lambda_j}{\lambda} \left( \sum_{m=1}^M \beta_m \bar{a}_j^m \right) / \left( \sum_{m=1}^M \beta_m \Phi_j \right),$$

где  $\Phi_j = (Q_j^m + \bar{a}_j^m + 2T^*)$ ;  $Q_j^m$  — среднее время ожидания заявки  $j$ -го типа в очереди на обслуживание «Сервером», размещенным в  $m$ -м узле сети, которое определяется по формуле (8);  $\bar{a}_j^m$  — среднее время обработки заявки  $j$ -го типа «Сервером» СОД реального времени, который размещен в  $m$ -м узле сети;  $\beta_m$  — переменная задачи, равная 1, если «Сервер» системы размещен и функционирует на базе  $m$ -го узла сети;  $T^*$  — средняя задержка передачи сообщения в сети (заявки «Клиента» и результата обработки заявки «Сервером»).

В качестве ограничений в данной задаче служат ограничения (12)–(20) и ограничение на коэффициент полезной работы при обслуживании заявок высших приоритетов:  $K_{nj} > \bar{K}_{nj}, j \in J^* \subset J$ , где  $J^*$  — подмножество высокоприоритетных заявок.

Сформулированные задачи представляют собой задачи нелинейного целочисленного программирования большой размерности. Вместо одной задачи большой размерности, как правило, следует рассматривать вариант последовательного решения  $M$  задач (по числу узлов вычислительной сети). В каждой задаче предполагается, что сервер размещается в очередном рассматриваемом  $m$ -м узле сети ( $m = \overline{1, M}$ ), при этом вектор  $\beta$  переменных принимает фиксированное значение (одна из переменных  $\beta_m = 1$ ). Тогда данная частная задача



может решаться с помощью предложенных в работах [3, 5] эвристических алгоритмов. Затем из полученных решений  $M$  частных задач выбирается решение, которому соответствует наилучшее значение принятого в исходной задаче критерия оптимальности решения.

Рассмотрим применение предложенного в данной работе метода решения задачи синтеза СОД реального времени с двухзвенной архитектурой «Клиент — Сервер», реализуемой на основе распределенной вычислительной сети, на примере задачи синтеза СОД реального времени, приведенном в работе [3, с. 393–420]. В данном примере приведена постановка и результат решения задачи синтеза оптимальной по критерию максимума производительности СОД реального времени, функционирующей на базе одного компьютера и обслуживающей заявки пользователей с относительными приоритетами. Система обслуживает 8 типов заявок пользователей ( $E = \{\varepsilon_j\}, j = \overline{1, J}$ ), для обработки заявок используется 29 процедур обработки данных ( $P = \{\rho_r\}, r = \overline{1, R}$ ) и 47 информационных элементов ( $D = \{d_l\}, l = \overline{1, L}$ ).

Проиллюстрируем, какое влияние на решение задачи синтеза СОД реального времени оказывают особенности таких систем, функционирующих в среде распределенных вычислительных сетей. Рассмотрим модификацию описанной в настоящей

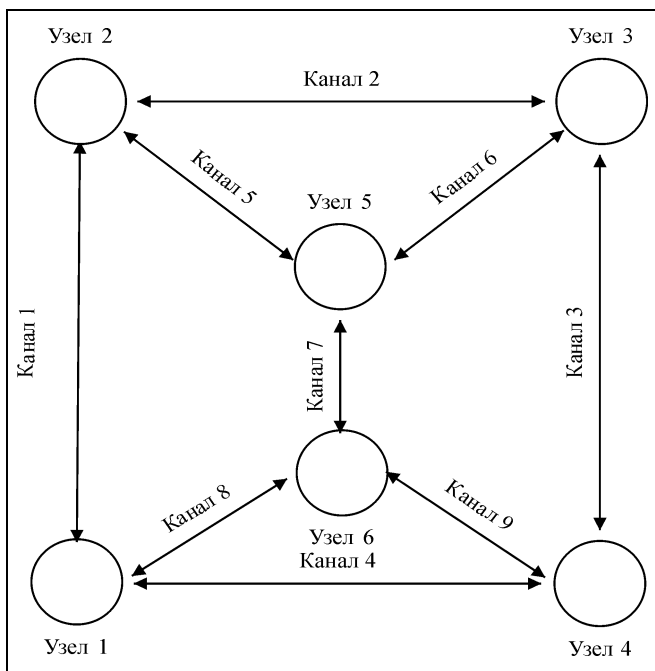


Рис. 1. Вычислительная сеть

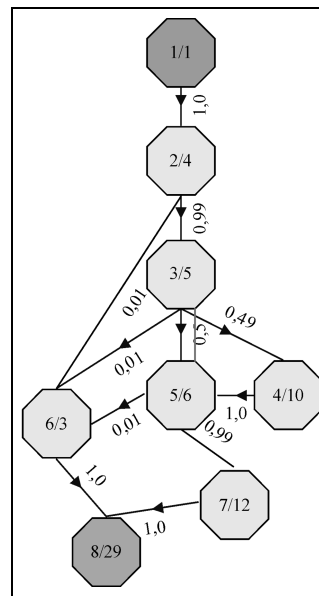


Рис. 2. Граф обработки задачи 1

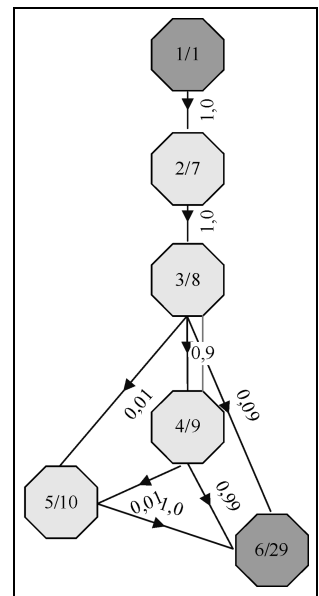


Рис. 3. Граф обработки задачи 2

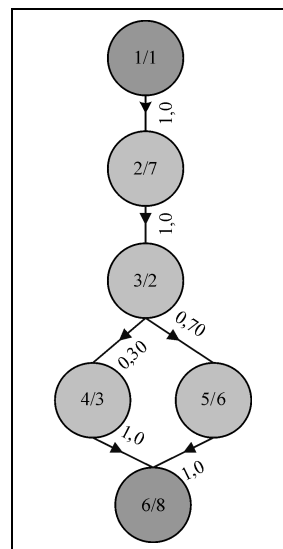


Рис. 4. Граф обработки запроса 1

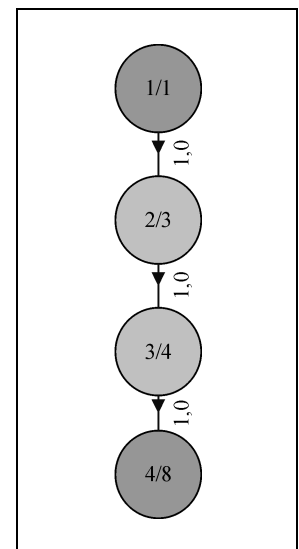


Рис. 5. Граф обработки запроса 2

работе задачи для ситуации синтеза СОД реального времени, построенной с двухзвенной архитектурой «Клиент — Сервер» и функционирующей в среде вычислительной сети со структурой, представленной взвешенным графом (рис. 1). Для получения характеристик процессов решения задач и обработки заявок используются соответствующие граф-модели. На рис. 2, 3 и 4, 5 показаны соответственно примеры графов обработки задач и графов обработки заявок.

Таблица 1

**Характеристики каналов связи сети**

Канал	1	2	3	4	5	6	7	8	9
Вес	2	3	2	2	3	3	3	2	2
$\gamma_{\phi}^* \cdot 10^6$	1	1	1,4	1	1,5	1,2	1	1,1	1,2
$C_{\phi}$ , МБ/с	6,1	4,0	6,3	3,4	4,5	3,3	5,7	6,4	8,9

Таблица 2

**Интенсивности заявок пользователей в узлах сети, с<sup>-1</sup>**

Узел	Тип заявки							
	1	2	3	4	5	6	7	8
1	0,02	0,04	0,05	0,02	0,04	0,06	0,05	0,03
2	0,01	0,03	0,03	0,03	0,05	0,05	0,02	0,05
3	0,03	0,04	0,05	0,05	0,04	0,04	0,04	0,06
4	0,05	0,01	0,04	0,04	0,05	0,06	0,03	0,02
5	0,03	0,03	0,02	0,06	0,04	0,06	0,01	0,02
6	0,04	0,02	0,05	0,03	0,05	0,04	0,03	0,01

Таблица 3

**Характеристики узлов сети**

Узел	1	2	3	4	5	6
$e_m^c$ , МБ/с	2	2	2	3	4	3
$e_m^3$ , МБ/с	1	1	2	2	2	2
$t_m^* \cdot 10^6$ , МБ/с	14	13	12	15	11	10
$K_m^{от}$	0,13	0,14	0,13	0,12	0,13	0,12

Таблица 4

**Характеристики заявок пользователей**

Тип заявки	1	3	4	5	6	7	8
$l_j^q$ , КБ	0,4	0,5	0,4	0,55	0,6	0,3	0,4
$l_j^r$ , КБ	2	22	5	15	40	7	10

**Значение времени исполнения процедур на эталонном компьютере**

$p_r$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\bar{t}^r \cdot 10^7$ , с	0	10	20	10	10	9	11	7	11	12	12	8	13	12	14
$p_r$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
$\bar{t}^r \cdot 10^7$ , с	12	10	11	19	18	14	16	15	4	6	8	3	5	0	

Таблица 5

Кратчайшие пути на рассматриваемых графах определяются с помощью алгоритма Беллмана — Форда поиска кратчайших путей на взвешенном графе. Под длиной пути понимается сумма весов ребер, входящих в этот путь. Веса ребер (каналов связи) вычислительной сети, вероятности  $\gamma_{\phi}^*$  возникновения ошибок в каналах при передаче данных и пропускные способности  $C_{\phi}$  каналов представлены в табл. 1. Среднее время  $K$  обработки одного сообщения в узле сети при его передаче по каналам связи и максимальное время  $t_{АСК}$  получения подтверждения о передаче сообщения равны, соответственно, 0,0032 и 0,0065 с. В узлах сети возникают заявки пользователей с интенсивностями  $\lambda_i^j$ ,  $i = \overline{1, M}$ ,  $j = \overline{1, J}$ , которые направляются по кратчайшим путям в узел сети, в котором размещен «Сервер» системы. Значения  $\lambda_i^j$  интенсивностей заявок представлены в табл. 2.

В табл. 3 представлены характеристики узлов вычислительной сети: скорость  $e_m^{(c/3)}$  считывания/записи информации с/на устройство внешней памяти узла  $m$ ; среднее время  $t_m^*$  поиска информации (ИМ или ПМ) во внешней памяти  $m$ -го узла, коэффициенты  $K_m^{от}$  относительной производительности сервера  $m$ -го узла по сравнению с эталонным компьютером. Средние значения длины заявок  $l_j^q$  и ответов на заявки  $l_j^r$  представлены в табл. 4. Значения среднего процессорного времени  $\bar{t}^r$  исполнения процедур  $p_r$  на эталонном компьютере представлены в табл. 5.

Предложенный метод решения данной задачи позволил получить следующее оптимальное решение по критерию минимума времени  $\bar{H}$  отклика системы на произвольную заявку пользователя со значением 0,088584 с при размещении сервера в узле 5. В табл. 6 приведены значения времени отклика системы, рассчитанные для всех узлов сети.

При использовании критерия максимума коэффициента  $K_{п}$  полезной работы СОД реального времени при обслуживании заявки произвольного типа оптимальным является размещение сервера в





Таблица 6

Значения времени отклика системы, рассчитанные для отдельных узлов сети

Узел	1	2	3	4	5	6
$\bar{H}$ , с	0,233498	0,233425	0,150042	0,111496	0,088584	0,149904

Таблица 7

Значения коэффициента, рассчитанные для отдельных узлов сети

Узел	1	2	3	4	5	6
$K_{\Pi}$	0,267970	0,269768	0,253056	0,238458	0,220114	0,263854

узле 2, что обеспечивает значение данного коэффициента, равное 0,269768. В табл. 7 приведены значения коэффициента, рассчитанные для всех узлов сети.

### ЗАКЛЮЧЕНИЕ

Системы обработки данных, работающие на базе распределительных вычислительных сетей в режиме реального времени, обладают рядом существенных особенностей. Комплекс этих особенностей определил необходимость дальнейшего развития теоретических положений, формализованных моделей и методов оптимального синтеза программного и информационного обеспечения модульных систем обработки данных, разработанных к настоящему моменту в Институте проблем управления им. В.А. Трапезникова РАН.

В данной статье изложены новые результаты работ, выполненных в Институте проблем управления, по созданию теоретических положений и разработке формализованных моделей и методов оптимального синтеза программного и информационного обеспечения распределенных систем обработки данных, работающих в режиме реального времени. Проведена формализация исходных данных для задачи оптимального синтеза. Получены оценки времени передачи сообщений СОД реального времени в вычислительной сети, оценки длительности обработки заявок пользователей сервером системы. Предложены постановка и метод решения задач синтеза оптимальных модульных структур программного и информационного обеспечения распределенной СОД реального времени с двухзвенной архитектурой «Клиент — Сервер». Дан пример решения этой задачи для СОД, обрабатывающей множество запросов пользователей и функционирующей на базе вычислительной сети определенной топологии.

Полученные в данной работе аналитические выражения для определения среднего и максимального времени выполнения процедур, решения отдельных задач и времени обработки заявок раз-

личных типов можно использовать в задачах анализа и синтеза структуры распределенных модульных СОД реального времени.

Приведенные постановка и метод решения задачи синтеза оптимальной модульной СОД реального времени, работающей на базе вычислительной сети с двухзвенной архитектурой «Клиент — Сервер», могут быть полезны при проектировании структуры программного и информационного обеспечения систем данного типа.

### ЛИТЕРАТУРА

1. Микрин Е.А., Кульба В.В., Сомов С.К. Синтез оптимальных распределенных модульных систем обработки данных реального времени / Науч. издание. — М.: ИПУ РАН, 2012. — 160 с.
2. Теоретические основы проектирования информационно-управляющих систем космических аппаратов / В.В. Кульба, Е.А. Микрин, Б.В. Павлов, В.Н. Платонов; под ред. Е.А. Микрина. — М.: Наука, 2006. — 579 с.
3. Кузнецов Н.А., Кульба В.В., Ковалевский С.С., Косяченко С.А. Методы анализа и синтеза модульных информационно-управляющих систем. — М.: Физматлит, 2002. — 800 с.
4. Микрин Е.А., Сомов С.К. Задача синтеза оптимальной СОД реального времени с резервированием информации и структурной избыточностью / Тр. XVI междунар. конф. «Проблемы управления безопасностью сложных систем». Москва, декабрь 2008 г. / Под ред. Н.И. Архиповой, В.В. Кульбы. — М., 2008. — С. 137—141.
5. Оптимальные модульные системы реального времени (анализ и синтез) / Н.А. Кузнецов, В.В. Кульба, С.А. Косяченко и др. — М.: ИППИ РАН, 1994.

Статья представлена к публикации членом редколлегии А.Д. Цвиркуном.

**Станислав Анатольевич Косяченко** — д-р техн. наук, гл. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ☎ (495) 334-89-59, ✉ kulba@ipu.ru,

**Сергей Станиславович Ковалевский** — д-р техн. наук, профессор, НИЯУ МИФИ, г. Москва, ✉ sergey.kss@gmail.com,

**Сергей Константинович Сомов** — канд. техн. наук, ст. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ☎ (495) 334-89-59, ✉ ssomov2009@gmail.com.