

# Задача распределения ресурсов в центрах обработки данных с единым планировщиком для всех типов ресурсов

## 1 Математическая постановка задачи распределения ресурсов ЦОД

Модель физических ресурсов ЦОД будем задавать графом  $H = (P \cup M \cup K, L)$ , где  $P$  — множество вычислительных узлов,  $M$  — множество хранилищ данных,  $K$  — множество коммутационных элементов сети обмена ЦОД,  $L$  — множество физических каналов передачи данных. На множествах  $P$ ,  $M$ ,  $K$  и  $L$  определены векторные функции скалярного аргумента, задающие соответственно характеристики вычислительных узлов, хранилищ данных, коммутационных элементов и каналов передачи данных:

$$\begin{aligned} ph(p) &= (ph_1(p), ph_2(p), \dots, ph_{n_1}(p)), \\ mh(m) &= (mh_1(m), mh_2(m), \dots, mh_{n_2}(m)), \\ kh(k) &= (kh_1(k), kh_2(k), \dots, kh_{n_3}(k)), \\ lh(l) &= (lh_1(l), lh_2(l), \dots, lh_{n_4}(l)). \end{aligned}$$

Здесь  $p \in P, m \in M, k \in K, l \in L$ . Компоненты векторной функции могут принимать целочисленные или вещественные значения. В данной работе предполагается, что для коммутационных элементов и каналов передачи данных определяются одинаковые характеристики.

Ресурсный запрос будем задавать графом  $G = (W \cup S, E)$ , где  $W$  — множество виртуальных машин, используемых приложениями,  $S$  — множество виртуальных хранилищ данных (storage-элементов),  $E$  — множество виртуальных каналов передачи данных между виртуальными машинами и storage-элементами запроса. На множествах  $W$ ,  $S$  и  $E$  определены векторные функции скалярного аргумента, задающие характеристики запрашиваемого виртуального элемента (требуемое качество сервиса, SLA):

$$\begin{aligned} wg(w) &= (wg_1(w), wg_2(w), \dots, wg_{n_1}(w)), \\ sg(s) &= (sg_1(s), sg_2(s), \dots, sg_{n_2}(s)), \\ eg(e) &= (eg_1(e), eg_2(e), \dots, eg_{n_4}(e)). \end{aligned}$$

Здесь  $w \in W, s \in S, e \in E$ . Компоненты векторной функции также принимают целочисленные или вещественные значения. Характеристики SLA элемента запроса совпадают с характеристиками соответствующего ему физического ресурса.

Назначением ресурсного запроса будем называть отображение:

$$A : G \rightarrow H = \{W \rightarrow P, S \rightarrow M, E \rightarrow \{K, L\}\}.$$

Выделим три типа отношений между характеристиками элементов запросов и соответствующих характеристик физических ресурсов. Обозначим через  $x_i$  характеристику  $i$ -го элемента запроса, через  $y_j$  — соответствующую ей характеристику  $j$ -го физического ресурса. Тогда эти отношения можно записать следующим образом:

1. Недопустимость перегрузки емкости физического ресурса:

$$\sum_{i \in R_j} x_i \leq y_j,$$

здесь  $R_j$  — множество элементов запросов, назначенных на выполнение на физическом ресурсе  $y_j$ .

2. Соответствие типа запрашиваемого ресурса типу физического ресурса:

$$x_i = y_i.$$

3. Наличие требуемых характеристик у физического ресурса:

$$x_i \leq y_i.$$

Отображение  $A$  будем называть *корректным*, если для всех физических ресурсов и всех их характеристик выполняются отношения 1-3.

*Остаточным графом* доступных ресурсов называется граф  $H_{res}$ , для которого переопределены значения функций по характеристикам, которые должны удовлетворять отношению 1:

$$\begin{aligned} ph_{res}(p) &= ph(p) - \sum_{w \in W_p} wg(w), \\ mh_{res}(m) &= mh(m) - \sum_{s \in S_m} sg(s), \\ kh_{res}(k) &= kh(k) - \sum_{e \in E_k} eg(e), \\ lh_{res}(l) &= lh(l) - \sum_{e \in E_l} eg(e). \end{aligned} \tag{1}$$

Здесь  $W_p$  — множество виртуальных машин, назначенных на выполнение на вычислительном узле  $p$ ,  $S_m$  — множество storage-элементов, размещенных в хранилище данных  $m$ ,  $E_k$  — множество виртуальных каналов, проходящих через коммутационный элемент  $k$ ,  $E_l$  — множество виртуальных каналов, отображенных на физический канал  $l$ .

В качестве *исходных данных* задачи назначения ресурсных запросов на физические ресурсы заданы:

1. множество запросов  $Z = \{G_i\}$ , поступивших планировщику;
2. остаточный граф доступных ресурсов  $H_{res} = (P \cup M \cup K, L)$ .

*Требуется:* из множества  $Z$  разместить на выполнение в ЦОД максимальное число запросов, таких, что отображение  $A$  является корректным.

В случае, когда невозможно назначить виртуальное хранилище данных, возможен вызов процедуры репликации, позволяющей дублировать данные на нескольких физических хранилищах данных. Данная проблема возникает, когда имеются виртуальные хранилища данных с высокой интенсивностью считывания и низкой интенсивностью записи. В этом случае для обеспечения консистентности данных не требуется канал с высокой пропускной способностью. При этом часть приложений может работать с виртуальным хранилищем данных, а другая — с его копией, которая располагается в другом физическом хранилище данных.

*Репликацией* называется отображение  $R : H \rightarrow H$ , дублирующее данные некоторого  $m \in M$  и создающее виртуальный канала поддержки консистентности данных  $(m', l_1, k_1, \dots, k_{n-1}, l_n, m); k_i \in K, j_i \in L, m' \in M, m' — реплика хранилища  $m$ .$

Если storage-элемент является репликацией некоторого виртуального хранилища данных  $s$ , то в граф запрашиваемых ресурсов  $G$  добавляется вершина  $s'$ . В граф  $G$  также добавляется виртуальный канал между вершинами  $s$  и  $s'$ , пропускная способность которого определяется исходя из требования обеспечения консистентности репликации и базы данных.

*Входом алгоритма* назначения запросов на физические ресурсы является остаточный граф доступных ресурсов  $H_{res}$  и множество ресурсных запросов  $\{G_i\}$ . Множество  $\{G_i\}$  формирует диспетчер задач. В него кроме вновь поступивших запросов могут входить и запросы, которые выполняются и для которых допустима миграция. Диспетчер задач также определяет время запуска планировщика.

*Выходом алгоритма* назначения запросов на физические ресурсы является множество назначений ресурсных запросов на физические ресурсы  $\{A_i : G_i \rightarrow H, i = \overline{1, n}\}$  и множество репликаций  $\{R_i\}, i = 0, 1, \dots$ .