

УДК 681.518

## ИСПОЛЬЗОВАНИЕ МОДЕЛИРОВАНИЯ ПРИ БАЛАНСИРОВКЕ НАГРУЗКИ В КОМПЬЮТЕРНОЙ СЕТИ УПРАВЛЕНИЯ

М.Е. Щербакова

Северодонецкий технологический институт

*Предлагается метод балансировки нагрузки в распределенной компьютерной системе управления с использованием нескольких моделей программного комплекса в виде графов взаимосвязей между программными модулями.*

В любой распределенной параллельной вычислительной системе очень важно правильно распределить задачи между узлами сети, для того чтобы улучшить загрузку всех компьютеров и, следовательно, повысить производительность распределенной параллельной вычислительной системы. Решение этой задачи особенно важно для распределенных компьютерных систем управления.

Задача формулируется следующим образом. Имеется сеть компьютерной системы управления, состоящая из рабочих станций и контроллеров. Для сети известен состав модулей программного комплекса, их взаимосвязи друг с другом, а также затраты времени на работу каждого модуля. Требуется так распределить программные модули между рабочими станциями и контроллерами, чтобы затраты времени на работу программного комплекса были минимальными.

В случае, если для обмена данными между рабочими станциями и удаленными на большие расстояния контроллерами используются медленные линии связи, например, некоммутируемые или, что еще актуальнее, коммутируемые линии связи, возникает задача уменьшения времени взаимодействия и объема информации, передаваемой по таким линиям связи. Формально это можно записать так:

$$\sum_{i=1}^n \sum_{j=1}^m F_{ij} \Rightarrow \min \quad (1)$$

где  $F_{ij}$  – время взаимодействия программы  $j$  с программой  $i$ ;  $m$  – число программ, выполняемых рабочей станцией,  $n$  – число программ, выполняемых контроллером.

Задачу минимизации времени взаимодействия программ из разных узлов сети предлагается решать следующим образом.

1. Строится модель комплекса программ в виде неориентированного связного графа  $G = (U, V)$  с множеством узлов  $V$

$= \{v_1, v_2, \dots, v_n\}$ , каждый из которых соответствует отдельной программе, и множеством рёбер  $U = \{u_1, u_2, \dots, u_m\}$ , соответствующих связям между программами. Каждому ребру  $[v_i, v_j]$  припишем вес  $c(i, j)$ , равный затратам времени контроллера на взаимодействия между соответствующими программными модулями при условии, что одна из них выполняется рабочей станцией, другая – контроллером.

2. К графу добавляются ещё два узла: К и РС, представляющие контроллер и рабочую станцию соответственно. С узлом К (контроллер) соединяются узлы графа, представляющие программы, использующие специальное оборудование контроллера или же выполняющие специфичные функции, которые должны всегда выполняться контроллером. Вес таких рёбер равен бесконечности:  $c(K, j) = \infty$ . По аналогии, с узлом РС (рабочая станция) соединяются только те узлы графа, которые должны обязательно выполняться рабочей станцией, если таковые имеются. Вес этих рёбер тоже устанавливается равным бесконечности:  $c(i, РС) = \infty$ .

3. Полученный граф взаимосвязей между программами можно рассматривать как транспортную сеть с источником К и стоком РС. Исходная задача сводится к задаче нахождения минимального разреза построенной таким образом сети (рис. 1).

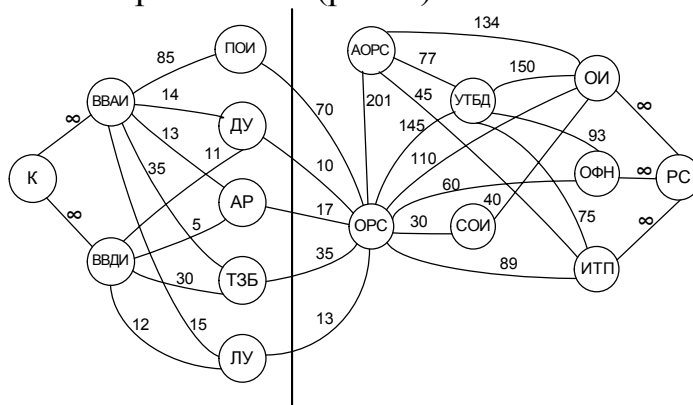


Рис. 1 Пример графа взаимосвязей для минимизации времени взаимодействия программных модулей

Решение задачи было проверено на модели программного комплекса, состоящего из следующих программных модулей: ввода – вывода аналоговой информации (ВВАИ), ввода – вывода дискретной информации (ВВДИ), первичной обработки информации (ПОИ), дистанционного управления (ДУ), автоматического регулирования (АР), технологических защит и блокировок (ТЗБ), логического управления (ЛУ), сервера ОРС (ОРС), адаптера ОРС (АОРС), управления технологической базой данных (УТБД), статистической обработки информации (СОИ), отображения информации (ОИ),

обработки и фиксации нарушений (ОФН), истории технологического процесса (ИТП). В графе на рис. 1 минимальный разрез делит множество программ на 2 части: те, что будут выполняться контроллером, и те, что будут выполняться рабочей станцией. Для нахождения минимального разреза на графе существует множество алгоритмов. В программной реализации метода был использован алгоритм Форда – Фалкерсона.

Вторая модификация разработанного метода минимизирует время выполнения программ контроллерами. Этот вид оптимизации имеет смысл проводить в случаях, когда нужно уменьшить число контроллеров с целью снижения стоимости технических средств и повышения надежности функционирования системы в целом.

Граф в этом случае строится так же, как и в предыдущем, но каждый узел соединяется еще и с узлом РС. Этим ребрам присваивается вес, равный времени выполнения соответствующих программных модулей контроллером (рис. 2).

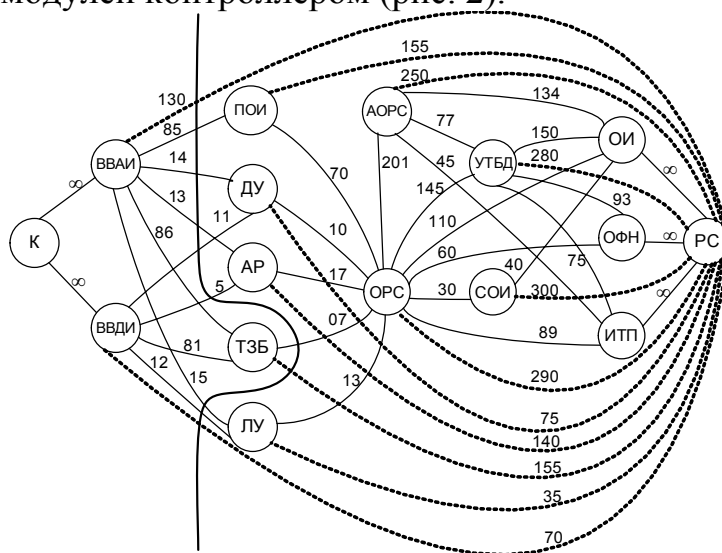


Рис. 2. Граф взаимосвязей для минимизации времени выполнения программ контроллером

Третья модификация метода минимизирует время выполнения программ рабочей станцией и контроллером и является дальнейшим развитием предыдущих двух методов. На этот раз минимизируется сумма времен выполнения программ контроллером, рабочей станцией и времени взаимодействия программ по сети. Использование этого метода минимизирует гарантированное время ответа всей системы в целом, уменьшает время реакции системы на внешние события, повышает оперативность реагирования на запросы технологического персонала. Метод аналогичен предыдущим двум методам во всем,

кроме способа построения графа для распределения программных модулей (рис. 3).

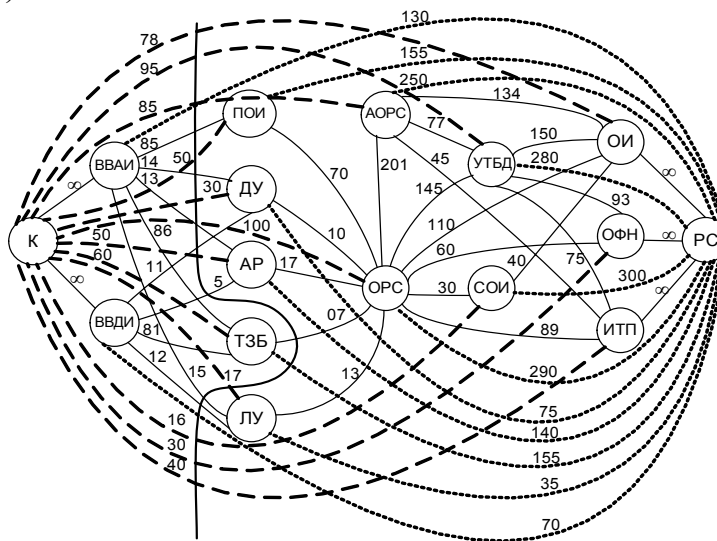


Рис. 3. Граф взаимосвязей для минимизации времени выполнения программ рабочей станцией и контроллером

Все вершины - программные модули соединяются ребрами с узлами РС и К. Этим ребрам присваивается вес, равный времени выполнения программного модуля контроллером и рабочей станцией соответственно. В минимальный разрез входят ребра, веса которых равны временам выполнения программ, размещаемых в рабочей станции, ребра, веса которых равны временам выполнения программ, размещаемых в контроллере, и ребра с весами, равными времени взаимодействия программ по сети. Таким образом, минимизируется суммарное время, которое контроллер и рабочая станция затратят на выполнение всех программ, если эти программы будут выполняться чисто последовательно, т. е. самое большое время одного цикла обработки системой всех входных и выходных данных.

Три вышеописанных метода предназначены для использования в программах установки и балансирования нагрузки пакетов систем управления. В частности, в пакете КВАРЦ [1] метод распределения программных модулей, минимизирующий время выполнения программ рабочей станцией и контроллером, дал возможность улучшить одну из важнейших характеристик систем реального масштаба времени – гарантированное время ответа на 15 – 19%.

### ***Литература***

1. Е.В. Щербаков, М.Е. Щербакова, В.К. Охрамович Автоматизированное проектирование ППО КСУ на базе пакета программ «Кварц». Монография /под ред. д.т.н., проф. А.Г. Руденко – Луганск: Издательство Восточноукраинского национального университета имени В.Даля, 2003.-200 с.

Получено 27.06.2007 г.