

УПРАВЛЕНИЕ

УДК 621.317.75:519.2

В.П. Корячко, Д.А. Перепелкин, А.И. Перепелкин**АЛГОРИТМ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ
НА БАЗЕ ПРОТОКОЛА IGRP
В КОРПОРАТИВНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ***Предложен алгоритм динамической маршрутизации на базе протокола IGRP, позволяющий уменьшить трудоемкость построения таблиц маршрутизации в корпоративных вычислительных сетях.***Ключевые слова:** динамическая маршрутизация, протокол IGRP.

Введение. Цель работы – разработать новый, более эффективный алгоритм поиска кратчайших путей на базе протокола IGRP, позволяющий повысить быстродействие корпоративных вычислительных сетей.

Необходимость обеспечения качественного обслуживания современного трафика, передаваемого через IP-сети, обуславливает высокие требования к эффективности передачи пакетов данных от отправителя к получателю. Загрузка и пропускная способность линий связи корпоративной сети динамически меняются, что, в свою очередь, может приводить к частой рассылке служебной информации об изменении маршрутов. Традиционно применяемый в корпоративных вычислительных сетях метод статической маршрутизации оказывается неэффективным [1]. Изменения характеристик каналов связи, модификация структуры сети, включение в нее новых узлов и линий связи приводят к полному пересчету таблиц маршрутизации.

Теоретические исследования. В настоящее время широкое применение получили алгоритмы динамической маршрутизации. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Используя протоколы адаптивных алгоритмов, маршрутизаторы могут собирать информацию о топологии связей в сети и оперативно реагировать на все изменения конфигурации связей.

Современные адаптивные протоколы обмена информацией о маршрутах, в свою очередь, делятся на две группы, каждая из которых связана с одним из следующих типов алгоритмов:

- дистанционно-векторные алгоритмы (Distance Vector Algorithm, DVA);
- алгоритмы состояния каналов (Link State Algorithm, LSA).

Протоколом, основанным на дистанционно-векторном алгоритме, является протокол IGRP (Interior Gateway Routing Protocol – протокол внутреннего шлюза) [2].

Этот протокол используется для передачи маршрутной информации между маршрутизаторами.

Пакеты IGRP передаются с использованием дейтаграмм IP с полем протокола IGP. Пакеты начинаются с заголовка IGRP, за которым сразу же следует заголовок IP. Структура заголовка IGRP показана на рисунке 1.

0		31
	Версия	
	Код операции	
	Редактирование	
	ASystem	
	NInterior	
	NSystem	
	NExterior	
	Контрольная сумма	

Рисунок 1 – Структура заголовка IGRP**Версия**

Номер версии протокола
(текущее значение – 1)

Код операции

Код операции, связанной с сообщением:

1. Update (обновление).
2. Request (запрос).

Редактирование

Порядковый номер, значение которого уменьшается при каждом внесении изменения в таблицу маршрутизации. Номер редактирования позволяет шлюзам избежать обработки обновлений таблиц маршрутизации, которые уже были учтены.

ASystem

Номер автономной системы. Шлюз может входить в несколько автономных систем, в каждой из которых используется свой протокол IGRP. Для каждой автономной системы используются свои таблицы маршрутизации. Это поле позволяет шлюзу выбрать набор используемых таблиц маршрутизации.

NInterior, NSystem, NExterior

Эти поля показывают номера записей в каждой из трех секций сообщений об обновлении таблиц. Первый элемент (NInterior) является внутренним, следующий (NSystem) – системным и последний (NExterior) – внешним.

Контрольная сумма

Контрольная сумма IP, рассчитанная по тому же алгоритму, который используется для дейтаграмм UDP. При вычислении контрольной суммы принимаются во внимание заголовки IGRP и маршрутная информация, которая следует после заголовка. При расчете поле контрольной суммы предполагается нулевым (не учитывается). Контрольная сумма не включает заголовки IP и не использует виртуальных заголовков, как в UDP и TCP.

Запрос IGRP требует от получателя передать таблицу маршрутизации. Для запросов используется только поле версии, кода операции и ASystem, остальные поля имеют нулевые значения.

Сообщения об обновлении таблиц содержат заголовки, сразу за которым располагается таблица маршрутизации. Количество записей в таблице ограничено размером дейтаграмм (1500 байтов с учетом заголовка IP). При используемой в настоящее время структуре записей таблица может содержать до 104 элементов. Если таблица маршрутизации содержит большее число записей, нужно использовать несколько сообщений.

Основные достоинства протокола IGRP:

- 1) стабильность маршрутов даже в очень больших и сложных сетях;
- 2) быстрый отклик на изменения топологии сети;
- 3) минимальная избыточность. Протокол IGRP не требует дополнительной пропускной способности каналов для своей работы;
- 4) разделение потока данных между несколькими параллельными маршрутами пример-

но равного достоинства;

5) учет частоты ошибок и уровня загрузки каналов;

6) возможность реализовать различные виды сервиса для одного и того же набора информации.

Сегодняшняя реализация протокола ориентирована на TCP/IP. Однако базовая конструкция системы позволяет использовать IGRP и с другими протоколами. IGRP имеет некоторое сходство со старыми протоколами, например с RIP и Hello. Здесь маршрутизатор обменивается маршрутной информацией только с непосредственными соседями. Поэтому задача маршрутизации решается всей совокупностью маршрутизаторов, а не каждым отдельно. Для того чтобы исключить осцилляции маршрутов, протокол IGRP должен игнорировать новую информацию в течение нескольких минут после ее возникновения. IGRP используется в маршрутизаторах, которые имеют связи с несколькими сетями и выполняют функции переключателей пакетов. Когда какой-то объект в одной сети хочет послать пакет в другую сеть, он должен послать его соответствующему маршрутизатору. Если адресат находится в одной из сетей, непосредственно связанной с маршрутизатором, он отправляет этот пакет по месту назначения. Если же адресат находится в более отдаленной сети, маршрутизатор перешлет пакет другому маршрутизатору, расположенному ближе к адресату. Здесь так же, как и в других протоколах, для хранения маршрутных данных используются специализированные базы данных. Протокол IGRP формирует эту базу данных на основе информации, которую он получит от соседних маршрутизаторов. В простейшем случае находится один путь для каждой из сетей. Сегменты пути характеризуются используемым сетевым интерфейсом, метрикой и маршрутизатором, куда следует сначала послать пакет. Метрика – то число, которое говорит о том, насколько хорош данный маршрут. Это число позволяет сравнить его с другими маршрутами, ведущими к тому же месту назначения и обеспечивающими тот же уровень QOS.

В протоколе IGRP предусматривается возможность разделять информационный поток между несколькими доступными эквивалентными маршрутами. Пользователь может сам разделить поток данных, если два или более пути оказались почти равными по метрике, при этом большая часть трафика будет послана по пути с лучшей метрикой. Метрика, используемая в IGRP, учитывает:

- 1) время задержки;

2) пропускную способность самого слабого сегмента пути (в битах в секунду);

3) загруженность канала (относительную);

4) надежность канала (определяется долей пакетов, достигших места назначения неповрежденными).

Время задержки предполагается равным времени, необходимому для достижения места назначения при нулевой загрузке сети. Дополнительные задержки, связанные с загрузкой, учитываются отдельно. Среди параметров, которые контролируются, но не учитываются метрикой, находятся число шагов до цели и MTU (maximum transfer unit - размер пакета, пересылаемого без фрагментации). Расчет метрики производится для каждого сегмента пути. Время от времени каждый маршрутизатор широковещательно рассылает свою маршрутную информацию всем соседним маршрутизаторам. Получатель сравнивает эти данные с уже имеющимися и вносит, если требуется, необходимые коррекции. На основании вновь полученной информации могут быть приняты решения об изменении маршрутов. Одним из преимуществ протокола IGRP является простота реконфигурации.

В настоящее время в протоколе маршрутизации IGRP используется алгоритм Беллмана – Форда. Трудоемкость данного алгоритма составляет порядка $O(N^3)$, где N – число маршрутизаторов в сети [3].

В условиях динамически изменяющейся структуры корпоративной вычислительной сети использование данного алгоритма оказывается неэффективным, так как он не учитывает возможные изменения топологии вычислительной сети. В работе [4] были предложены алгоритмы адаптивной маршрутизации, которые за счет сбора дополнительной информации позволяют учесть возможные изменения конфигурации вычислительной сети и не производить полный пересчет маршрутных таблиц. Это позволило в существующей вычислительной сети снизить трудоемкость пересчета таблиц маршрутизации до величины порядка $O(N)$.

Разработка алгоритма. Для повышения эффективности использования протокола IGRP разработан алгоритм, который учитывает предложенные в работе [4] алгоритмы адаптивной маршрутизации.

Укрупненная схема алгоритма на базе протокола IGRP имеет следующий вид

Шаг 1. Первоначальная инициализация соседей. Используя пакет HELLO протокола IGRP, опре-

делить веса линий связи W_{ij} до ближайших соседей.

Шаг 2. Используя сообщение query (запрос), получить маршрутную информацию от соседей для построения таблиц маршрутизации с использованием разработанных алгоритмов адаптивной маршрутизации.

Шаг 3. Таблицы маршрутизации построены:

а) если да – перейти к шагу 4;

б) иначе – к шагу 2.

Шаг 4. Определить, есть ли пакеты на передачу:

а) если есть – перейти к шагу 5;

б) иначе – к шагу 12.

Шаг 5. Используя поля «Время жизни» и «Контрольная сумма заголовка» протокола IP, определить, требуется ли уничтожить (отбросить) данный пакет:

а) если да – перейти к шагу 11;

б) иначе – к шагу 6.

Шаг 6. Используя таблицы маршрутизации, определить, требуется ли разделять информационный поток между несколькими доступными эквивалентными маршрутами:

а) если да – перейти к шагу 7;

б) иначе – к шагу 9.

Шаг 7. Разделить информационный поток между несколькими доступными эквивалентными маршрутами.

Шаг 8. а) передать пакеты по доступным эквивалентным маршрутам;

б) установить флаг передачи.

Шаг 9. Проверка флага передачи:

а) если флаг установлен – перейти к шагу 11;

б) иначе – к шагу 10.

Шаг 10. а) передать пакет, используя таблицу маршрутизации;

б) перейти к шагу 11.

Шаг 11. Проверка окончания работы маршрутизатора:

а) если да – перейти к шагу 19;

б) иначе – сбросить флаг передачи и перейти к шагу 4.

Шаг 12. Послать сообщение query (запрос) соседним маршрутизаторам.

Шаг 13. Ожидать ответа от соседей.

Шаг 14. Сообщение reply (отклик) от соседних маршрутизаторов получено:

а) если да – перейти к шагу 15;

б) иначе – к шагу 13.

Шаг 15. Требуется вносить изменения в таблицы маршрутизации:

а) если да – перейти к шагу 16;

б) иначе – к шагу 11.

Шаг 16. Перестроить таблицы маршрутизации, используя разработанные алгоритмы адаптивной маршрутизации.

Шаг 17. Для исключения осцилляции маршрутов сделать временную задержку.

Шаг 18. а) передать пакет со служебной информацией соседним маршрутизаторам;

б) перейти к шагу 11.

Шаг 19. Конец работы маршрутизатора.

Для подтверждения правильности предложенного алгоритма адаптивной маршрутизации на базе протокола IGRP разработана программа имитационного моделирования процессов маршрутизации в корпоративных вычислительных сетях. При разработке основное внимание уделялось корректности предлагаемого алгоритма и размерности решаемой задачи.

Целью исследования было оценить максимальное, минимальное и среднее значения размерности решаемой задачи. Исходный граф, ребро для изменения, удаления, добавления и приращения веса выбирались случайным образом. Для каждого испытания на множестве обработанных изменений выбирались минимальное, максимальное и среднее значения размерности задачи, выраженные через количество вершин, для которых необходим поиск кратчайшего пути. По этим значениям были построены графики. Для каждого эксперимента были найдены математическое ожидание и среднее квадратичное отклонение числа изменений. Для алгоритма адаптивной маршрутизации на базе протокола IGRP определялось число фактически выполненных парных переходов. На рисунках 2 – 4 представлены результаты моделирования разработанного алгоритма поиска оптимальных маршрутов.

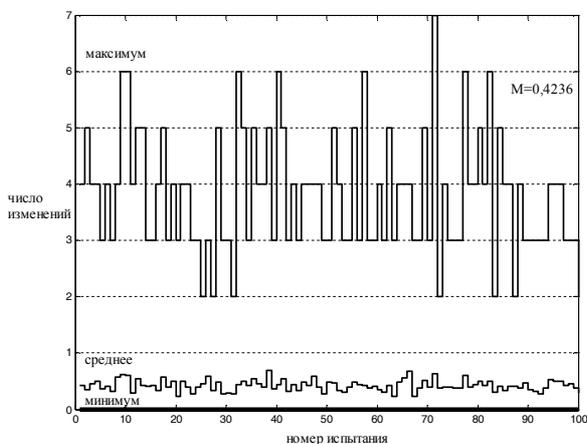


Рисунок 2 – Число изменений дерева в графе из 10 вершин

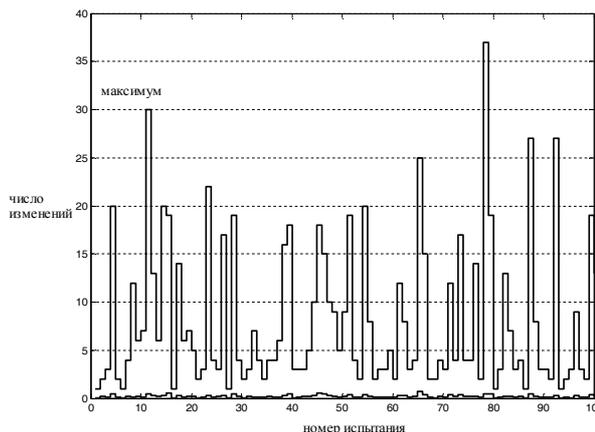


Рисунок 3 – Число изменений дерева в графе из 100 вершин

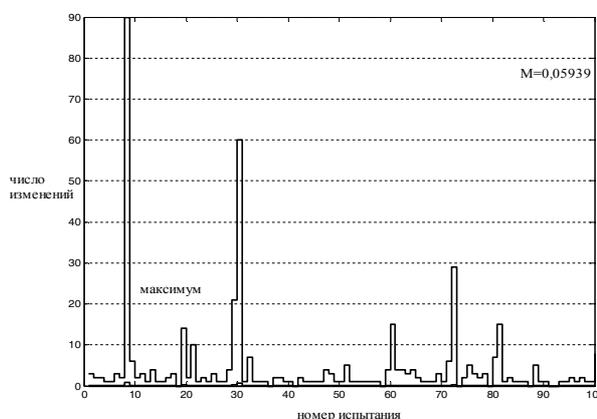


Рисунок 4 – Число изменений дерева в графе из 500 вершин

В таблице приведены обобщенные статистические характеристики для средней размерности задачи. В представленной таблице через СКО обозначено среднее квадратичное отклонение.

Число вершин графа	Min значение	Max значение	МО	СКО
10	0,30	0,8182	0,4236	0,0943
100	0,06	0,7401	0,2185	0,1368
500	0	0,9192	0,0594	0,1173

Были проведены исследования графов, состоящих из 10, 100 и 500 вершин. Исследование разработанного алгоритма адаптивной маршрутизации на базе протокола IGRP показало, что математическое ожидание числа изменений не превышает величины $N/2$, а его максимальное значение не превышает N . На основе этого можно сделать вывод, что предложенный адаптивный алгоритм маршрутизации на базе протокола IGRP является эффективным при поиске оптимальных маршрутов с учетом частичных динамических изменений структуры

сети на основе информации о возможных парных переходах.

Заключение. Использование данного подхода позволяет повысить эффективность алгоритмов динамической маршрутизации на базе протокола IGRP за счет уменьшения трудоемкости построения таблиц маршрутизации до величины порядка $O(N)$ в условиях динамически изменяющейся структуры корпоративной сети и характеристик линий связи.

Библиографический список

1. Куракин Д.В. Маршрутизаторы для глобаль-

ных телекоммуникационных сетей и реализуемые в них алгоритмы // Информационные технологии – 1996. № 2.

2. Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. – СПб.: БХВ-Санкт-Петербург, 2001. – 512 с.: ил.

3. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – 672 с.: ил.

4. Перепелкин Д.А., Перепелкин А.И. Разработка алгоритмов адаптивной маршрутизации в корпоративных вычислительных сетях // Вестник Рязанской государственной радиотехнической академии. – 2006.- № 19.-С. 114 – 116.

УДК 621.396

Н.В. Гоев

СИНТЕЗ АВТОМАТИЗИРОВАННЫХ СИСТЕМ КОНТРОЛЯ НА ОСНОВЕ АППАРАТНО-ПРОГРАММНЫХ МОДУЛЕЙ

Приведено определение модуля автоматизированных систем контроля (АСК) как аппаратно-программных систем. Сформулированы задачи синтеза базового аппаратного обеспечения и на основе него автоматизированных систем контроля. Предложенные решения позволяют в 2-3 раза сократить затраты на создание и производство АСК.

Ключевые слова: автоматизированный контроль, системы контроля, аппаратно-программные модули.

Введение. АСК, являющиеся одним из видов информационно-измерительных систем, предназначены для оценки технического состояния объектов техники и принятия соответствующего решения. От характеристик АСК, принципов их построения и проектирования существенным образом зависит качество и сроки производства объектов техники, а также эффективность их эксплуатации. Современные АСК являются аппаратно-программными системами, т.е. функции АСК выполняются как аппаратными, так и программными средствами. Обычно синтез АСК осуществляется на основе метода выбора состава аппаратных модулей, обеспечивающих выполнение заданной номенклатуры задач контроля, т.е. взаимосвязь аппаратных и программных средств в системе не учитывается [1, 2]. Такой подход приводит к большим затратам на создание АСК, высокой их стоимости, не обеспеченности современных требуемых технических характеристик АСК.

Целью работы является определение модуля для построения АСК как аппаратно-программной системы и на этой основе форму-

лировка задач синтеза базового аппаратного обеспечения (БАО) и АСК.

Определение модуля АСК. Учитывая, что АСК является аппаратно-программной системой, модуль как часть системы должен быть аппаратно-программным средством, содержащим как аппаратные, так и программные средства [3]. Для построения и проектирования АСК аппаратно-программные модули должны реализовывать функции АСК на уровне системы. Такие функциональные аппаратно-программные модули являются не физическими модулями - они состоят не из одного физического устройства (самостоятельное изделие, имеющее автономную документацию на изготовление, полностью собранное, прошедшее функциональную проверку и готовое к монтажу), а из нескольких физических устройств и программ. Такие совокупные модули целесообразно назвать виртуальными функциональными модулями или виртуальными устройствами (ВиУ). Виртуальное устройство - устройство, функции которого реализуются посредством программы и аппаратных средств на основе

одного или нескольких устройств. Виртуальные устройства реализуют методики выполнения функций, в частности методики измерений контролируемых и выдачи стимулирующих сигналов.

Подход выделения аппаратно-программных модулей - виртуальных устройств, реализующих функции АСК в целом, позволяет представлять АСК как совокупность виртуальных устройств. Т.е. определять функциональную структуру

АСК на основе ВиУ, отражающей проявление ее свойств во внешних по отношению к ней сечениях. При таком подходе функциональные возможности АСК описываются на уровне виртуальных устройств. В соответствии с определением обобщенная структура ВиУ, его взаимодействие с другими частями АСК на уровне исполнительной системы представлена на рисунке 1.

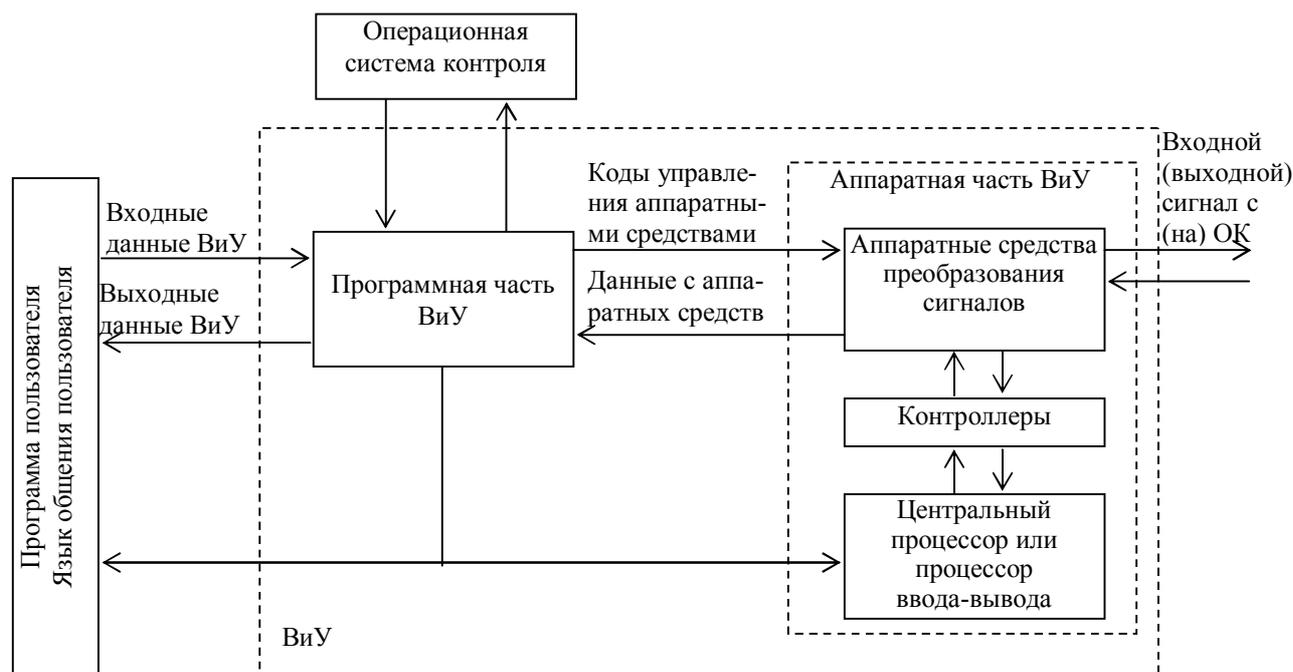


Рисунок 1 – Обобщенная структура ВиУ

В процессе выполнения функций контроля ВиУ взаимодействуют по входным и выходным сигналам с объектом контроля (ОК), по входным и выходным данным управления – с программой пользователя или с действиями пользователя на языке общения с АСК, по управлению процессом выполнения функций – с операционной системой контроля (ОСК).

Программная часть ВиУ обеспечивает преобразование входных данных из программы пользователя или входных данных пользователя в коды управления аппаратными средствами, чтение кодов с аппаратных средств, их обработку, вычисление результата и передачу его в качестве выходных данных в программу пользователя или непосредственно пользователю на языке общения.

Представление АСК совокупностью виртуальных устройств обуславливают необходимость разработки соответствующих методов синтеза как БАО, так и АСК на основе БАО, позволяющих устранить недостатки существующих методов и существенно снизить затраты на

обеспечение контроля сложных объектов техники.

Синтез базового аппаратного обеспечения. Задача синтеза БАО предполагает определение набора ВиУ, который обеспечивает выполнение требуемых задач в соответствии с предъявленными требованиями при использовании в качестве критерия синтеза минимальных суммарных затрат на разработку и производство БАО, а также удовлетворять условию полноты комплекса.

Постановка задачи синтеза БАО представляется следующим образом.

1. Пусть задано множество задач (функций) $A\{A_i\}, i=\overline{1, T}$, которые должны выполняться БАО. Под задачей понимается выполнение определенной функции – измерение характеристики сигнала, выдача стимулирующего сигнала и т.п. Заданное множество задач должно удовлетворять требованию функциональной и структурной полноты БАО.

2. Для каждой задачи A_i задано множество (совокупность) требований $t_{ij} \in T_i$. Требования

$t_{ij} \in T_i$ задаются в виде набора совокупностей $T_n\{T_{ij}\}$, $i=\overline{1, J_n}$ требований в терминах унифицированных описаний сигналов. Перечень требований в совокупности T_i для задачи A_i задается в виде $R_i\{R_m\}$, $m=\overline{1, M_i}$.

3. Известна функция потребности θ в БАО, которая может быть задана как общее число комплектов средств БАО, необходимых за весь период использования. Известна функция потребности φ_i в каждой задаче на весь период использования БАО, которая может быть выражена как доля общей потребности θ в БАО. Например, если какая-то функция БАО необходима в каждой комплектации, поставляемой потребителю, то $\varphi_i = 1$. Известна функция потребности φ_n в каждой совокупности требований $\varphi_i(T_i)$ для каждой задачи на весь период использования БАО, которая может быть выражена как доля потребности в задаче φ_i .

4. Определено множество виртуальных устройств, пригодных для решения задач A_i , $i=\overline{1, I}$, $V_i\{V_{ij}\}$, $j=\overline{1, J_i}$. Каждое ВиУ V_{ij} реализуется совокупностью программ и аппаратных средств. Будем считать, что каждому ВиУ соответствует одна программа. Одно аппаратное средство может реализовать одно или несколько ВиУ.

5. Определено исходное множество аппаратных средств $Y\{Y_k\}$, $k=\overline{1, K}$ для множества виртуальных устройств V_{ij} , $j=\overline{1, J_i}$.

6. Известна функция затрат ВиУ $\psi(\xi_{ij})$, которая определяется стоимостью программной части S_m и стоимостью аппаратных средств $C_k(Z_k)$, входящих в ВиУ.

Стоимость аппаратного средства определяется выражением

$$C_k(Z_k) = \begin{cases} a_k + b_k Z_k, & \text{если } Z_k > 0, \\ 0, & \text{если } Z_k = 0, \end{cases}$$

где a_k – стоимость разработки устройства Q_k , не зависящая от объема производства;

b_k – стоимость изготовления устройства Q_k ;

Z_k – объем производства устройства Q_k .

7. Необходимо определить такой набор ВиУ из множества V_{ij} , $j=\overline{1, J_i}$ и аппаратных средств их реализации из множества, которые обеспечивают решение всех задач A_i для каждой совокупности требований t_{ij} при минимальных суммарных затратах C_Σ на разработку и производство БАО и удовлетворяющие условию полноты комплекса.

8. Связи между различными множествами описываются матрицами соответствия.

$$\beta_{ijm} = \begin{cases} 1, & \text{если программный модуль } V_m \\ & \text{используется в ВиУ } V_{ij} \\ 0, & \text{если программный модуль } V_m \\ & \text{не используется в ВиУ } V_{ij} \end{cases}$$

$$\alpha_{ijk} = \begin{cases} 1, & \text{если устройство } Y_k \\ & \text{используется в ВиУ } V_{ij} \\ 0, & \text{если устройство } Y_k \\ & \text{не используется в ВиУ } V_{ij} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если для задачи } A_i \\ & \text{используется ВиУ } V_{ij} \\ 0, & \text{если для задачи } A_i \\ & \text{не используется ВиУ } V_{ij} \end{cases}$$

9. Математическая формулировка задачи синтеза БАО представляется следующим образом:

$$\begin{aligned} \min_{x_{ij}} C_\Sigma &= \sum_{k=1}^K a_k \operatorname{sgn} \sum_{i=1}^I \sum_{j=1}^{J_i} \alpha_{ijk} x_{ij} + \\ &+ \sum_{k=1}^K b_k \theta \varphi_n \varphi_i \operatorname{sgn} \sum_{i=1}^I \sum_{j=1}^{J_i} \alpha_{ijk} x_{ij} + \\ &+ \sum_{m=1}^M S_m \operatorname{sgn} \sum_{i=1}^I \sum_{j=1}^{J_i} \beta_{ijm} x_{ij} \quad , \end{aligned}$$

где

$$\operatorname{sgn}(u) = \begin{cases} 1, & \text{если } u > 0, \\ 0, & \text{если } u \leq 0 \quad , \end{cases}$$

при ограничениях

$$\sum_{j=1}^{J_i} x_{ij} = 1 \quad i = \overline{1, I} \quad , \quad (1)$$

Ограничение (1) означает, что для решения задачи A_i выбирается ровно один вариант V_{ij} .

Решение поставленной задачи на основе виртуальных устройств позволяет создавать оптимальные комплексы БАО с функциональной и структурной полнотой не менее 0,9. Опыт создания нескольких поколений БАО на ОАО «РКБ «Глобус» показал, что построение нескольких десятков АСК практически выполнено на базе средств БАО с объемом заимствования более 90 %.

Синтез АСК. Задачи синтеза АСК на основе БАО предполагают определение состава и характеристик ВиУ для заданных задач контроля. Рассматривается задача синтеза АСК для множества ОК. При этом необходимо определить набор ВиУ для обеспечения всех задач контроля заданных ОК при минимальных затратах на разработку и изготовление, а также выполнения требований по достоверности, времени контроля и объему памяти.

Постановка задачи оптимального синтеза

АСК для множества ОК на основе ВиУ представляется следующим образом.

1. Пусть задана совокупность задач контроля $A\{A_n\}$ $n=1, N$. Для каждой задачи контроля заданы: перечень контролируемых параметров $X_n\{X_i\}$, $i=1, I_n$, перечень стимулирующих и контролируемых сигналов $R_n\{R_m\}$, $m=1, M_n$, где m от 1 до h – стимулирующие сигналы и от $(h+1)$ до M_n – контролируемые сигналы; укрупненные алгоритмы контроля параметров $W_n\{W_i\}$. Известно требуемое число АСК φ_n для каждой задачи. Для контролируемых параметров известны законы распределения вероятностей значений параметров $f(x_i)$ и допустимые пределы их изменения. Известны описания стимулирующих и контролируемых сигналов - для всех задач контроля и описание общего исходного множества ВиУ B_0 , на основании которых определено исходное множество ВиУ по типам сигналов и ОК $B\{B_u\}$, $u=1, U$.

2. Совокупность аппаратных средств (устройств) всего исходного множества ВиУ $B\{B_u\}$ образует исходное множество устройств $Q\{Q_k\}$, $k=1, K$. Причем каждое ВиУ характеризуется:

временем исполнения t_u ;

погрешностью (выдачи или преобразования сигнала) δ_u ;

множеством используемых устройств $Q_u\{Q_k\}$, $k \in K_u$, где K_u - множество индексов устройств, используемых ВиУ B_u ;

объемом памяти для хранения программы ВиУ P_u ;

стоимостью разработки программы ВиУ C_{up} ;

стоимостью настройки БПО на ВиУ C_{un} .

Для каждого устройства Q_k задана стоимость $C_k(\varphi_k)$, как функция от объема производства φ_k

$$C_k(\varphi_k) = \begin{cases} a_k + b_k \varphi_k, & \text{если } \varphi_k > 0, \\ 0, & \text{если } \varphi_k = 0 \end{cases}$$

где a_k - стоимость разработки устройства Q_k , не зависящая от объема его производства, b_k - стоимость изготовления устройства.

3. Необходимо определить набор ВиУ, который обеспечивает выполнение заданных требований по достоверности D_{n3} , времени t_{n3} , объему памяти P_{n3} для каждой задачи контроля A_n при минимальной стоимости C решения всех задач контроля.

Связи между различными множествами описываются матрицами соответствия:

$$x_{mu} = \begin{cases} 1, & \text{если для выдачи стимулирующего или преобразования контролируемого сигнала } R_m \text{ используется ВиУ } B_u, \\ 0, & \text{в противном случае,} \end{cases}$$

$$\eta_{uk} = \begin{cases} 1, & \text{если для ВиУ } B_u \text{ требуется устройство } Q_k, \\ 0, & \text{в противном случае,} \end{cases}$$

$$\gamma_{im} = \begin{cases} 1, & \text{если сигнал } R_m \text{ используется при контроле параметра } X_i, \\ 0, & \text{в противном случае.} \end{cases}$$

Кроме того, вводятся коэффициенты:

ε_{mi} – коэффициент влияния погрешности формирования или измерения сигнала R_m на погрешность контроля параметра X_i ;

χ_{mi} – коэффициент влияния времени преобразования или выдачи сигнала R_m на время контроля параметра X_i .

4. Математическая формулировка задачи синтеза АСК для множества ОК представляется следующим образом:

$$\begin{aligned} \min C = & \sum_{k=1}^K \left\{ \left[a_k \operatorname{sgn} \left(\sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{u=1}^U x_{mu} \eta_{uk} \right) \right] + \right. \\ & \left. + b_k \sum_{n=1}^N \left[\varphi_n \operatorname{sgn} \left(\sum_{m=1}^{M_n} \sum_{u=1}^U x_{mu} \eta_{uk} \right) \right] \right\} + \\ & + \sum_{u=1}^U C_{up} \operatorname{sgn} \left(\sum_{n=1}^N \sum_{m=1}^{M_n} x_{mu} \right) + \\ & + \sum_{u=1}^U C_{un} \operatorname{sgn} \left(\sum_{n=1}^N \sum_{m=1}^{M_n} x_{mu} \right), \end{aligned}$$

при следующих ограничениях:

вероятности ложного отказа $P_{ло}$ и необнаруженного отказа $P_{но}$ при контроле каждого ОК должны быть меньше заданных значений:

$$P_{ло}^n(x_{mu}) = \prod_{i=1}^{I_n} \left[f_1 \left(\sqrt{\sum_{m=1}^{M_n} \varepsilon_{mi}^2 \sum_{u=1}^U x_{mu} \sigma_u^2} \right) - f_3 \left(\sqrt{\sum_{m=1}^{M_n} \varepsilon_{mi}^2 \sum_{u=1}^U x_{mu} \sigma_u^2} \right) \right] \leq P_{лоB}^n,$$

$$n = \overline{1, N},$$

$$P_{но}^n(x_{mu}) = \prod_{i=1}^{I_n} \left[f_2 \left(\sqrt{\sum_{m=1}^{M_n} \varepsilon_{mi}^2 \sum_{u=1}^U x_{mu} \sigma_u^2} \right) - f_3 \left(\sqrt{\sum_{m=1}^{M_n} \varepsilon_{mi}^2 \sum_{u=1}^U x_{mu} \sigma_u^2} \right) \right] \leq P_{ноB}^n,$$

$$n = \overline{1, N},$$

время контроля каждого ОК должно быть меньше заданного

$$t_{нс}(x_{mu}) = \sum_{i=1}^{I_n} \sum_{m=1}^{M_n} \chi_{mi} \sum_{u=1}^U x_{mu} t_u \leq t_{нзад} - t_{nm} - t_{ny},$$

$$n = \overline{1, N},$$

объем памяти для размещения программных средств не должен превышать заданный

$$V_u(x_{mu}) = \sum_{u=1}^U V_u \operatorname{sgn} \left(\sum_{m=1}^{M_n} x_{mu} \right) \leq V_{нзад} - V_{ny},$$

$$n = \overline{1, N},$$

а также для каждого требуемого при контроле сигнала должно быть назначено одно ВиУ

$$\sum_{u=1}^U x_{mu} \operatorname{sgn} \left(\sum_{i=1}^{I_n} \gamma_{im} \right) = 1 \quad m = \overline{1, M_n} \quad n = \overline{1, N}. \quad (2)$$

Выражение (2) определяет выбор единственного ВиУ для сигнала и, следовательно, единственное решение задачи оптимизации.

Выводы. Предложено определение модуля АСК, учитывающее ее особенности. Модуль является аппаратно-программным средством и реализует функции на уровне системы. Такие функциональные аппаратно-программные модули состоят из аппаратных средств и программ и называются виртуальными устройствами. Виртуальные устройства реализуют функции АСК в целом и позволяют представлять ее структуру как совокупность ВиУ и, соответственно, описывать функциональные возможности АСК на уровне виртуальных устройств. На основе ВиУ сформулированы задача синтеза БАО как определение набора унифицированных ВиУ и

задача синтеза АСК для множества ОК как определение состава ВиУ, обеспечивающего выполнение требований по достоверности, времени контроля, объему памяти при минимальной стоимости.

Применение изложенного подхода позволяет сократить затраты на разработку АСК в 2-3 раза.

Библиографический список

1. Автоматическая аппаратура контроля радиоэлектронного оборудования (вопросы проектирования) / Н.Н. Пономарев, И.С. Фрумкин, И.С. Гусинский и др.; под ред. Н.Н. Пономарева. – М.: Сов. Радио, 1975. – 328 с.
2. Проектирование внешних средств автоматизированного контроля радиоэлектронного оборудования / Н.Н. Пономарев, И.С. Фрумкин, И.С. Гусинский и др.; под ред. Н.Н. Пономарева, М.: Радио и связь, 1984. – 296 с.
3. Пономарев Н.Н., Максимов Ю.Н., Фрумкин И.С., Гоев Н.В. Современное состояние и направления развития внешних средств контроля // Радиопромышленность. 1991- № 4.- С.3-5.

УДК 658.562

В.А. Фаткин

СОВЕРШЕНСТВОВАНИЕ ОРГАНИЗАЦИОННОЙ СРЕДЫ ПРЕДПРИЯТИЯ НА ОСНОВЕ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ И ПРИНЦИПОВ TQM

Предложен подход к совершенствованию организационной среды предприятия на основе построения ячеек, использования информационных технологий, принципов TQM для сокращения потерь и совершенствования управления.

Ключевые слова: информационные технологии, качество, процессный подход, управление, ячейка, принципы TQM.

Введение. Цель работы – рассмотрение вопросов организационного развития на основе процессного подхода, современных информационных технологий, принципов TQM (всеобщего управления качеством) для совершенствования организационной среды предприятия и управления производством.

Совершенствование организационной и управленческой деятельности предприятий основано на всё большей ориентации предприятий на потребителей и формирующиеся социально-экономические отношения в обществе. Возникает необходимость использования и наращивания имеющегося потенциала предприятий с помощью организации производства как актив-

ного и малозатратного ресурса.

Формы организации производства представляют собой определенное сочетание во времени и в пространстве элементов производственного процесса при соответствующем уровне его интеграции, выраженное системой устойчивых связей. Несоответствия в производственном процессе, сбои в работе организации – это матрица устойчивого аномального состояния, чтобы устранить возникающие несоответствия, необходимы кардинальные действия. Совершенствование существующих структурных построений, которые во времени и пространстве образуют совокупность основных форм организации производства, можно получить, используя новые

принципы и методы, в том числе процессного подхода, информационного обеспечения и TQM (Total Quality Management) [1, 2, 5].

Опыт зарубежных компаний, разработчиков и производителей наукоёмкой продукции: Boeing, Aircraft, Airbus, Bell Helicopter Textron, General Motors, Rockwell International, Land Rover, Northrop Grumman и др. в создании и использовании современных систем информационного обеспечения производства показывает, что главным условием эффективности их функционирования является наличие интегрированной системы сбора и анализа информации о продукции на всех этапах её жизненного цикла (ЖЦ) [3, 4, 7, 8].

Актуальность и достоверность информации достигаются за счет того, что любое изменение становится доступным всем участникам ЖЦ на основе использования модели данных, не допускающей применения некорректных данных, и непротиворечивости информации. Для интеграции информационных процессов на различных этапах ЖЦ продукции используются CALS - технологии (Continuous Acquisition and LifeCycle Support – непрерывная поддержка поставок и ЖЦ продукции).

Между элементами организационной структуры предприятия существует информационный обмен, что позволяет обеспечивать функционирование организации как единого целого на базе CALS технологий. Но для этого целесообразно создание в организации первичных производственных ячеек. При организации информационного обмена и наделении ячеек определённой самостоятельностью это позволяет повысить эффективность и результативность действия организации [6, 9, 10, 11, 12].

Ячейка при этом будет содержать жёсткие и мягкие звенья. Жёсткие звенья: законодательные акты, нормативно – технические документы. Мягкие звенья – это создаваемые процессы, информационные связи. Объединение жёстких и мягких звеньев на основе процессного подхода даёт гибкость производству, а динамичность организации обеспечивается внедрением принципов TQM. В отличие от существующей практики раздельного проектирования производственных процессов требуется увязать эти частные процессы в единый. Это достигается путем объединения всех рабочих мест в ячейку с помощью средств вычислительной техники в единый комплекс, который представляет собой совокупность взаимосвязанных элементов структуры, средств вычислительной техники, предназначенных для организации процесса производства.

При использовании ячеек мы не только улучшаем структуру организации, но и совершенствуем процессы в организации. Основанием для использования ячеек служит возможность в определении требований к качеству, безотносительно к тому, производится физический продукт или оказывается услуга, делается ли это для внешнего или внутреннего потребителя. Организовывая рабочие ячейки и анализируя их взаимодействие, можно оценивать их качество. Рабочая ячейка при этом определяется как структурная единица, выполняющая ряд постоянных заданий и преобразующая определенные входы и выходы. Ячейкой может быть как один человек, так и целый цех (отдел) или его часть.

Более рациональный путь функционирования рабочих ячеек — дать свободу выбора сотрудникам, вовлеченным в отдельные сегменты бизнес-процесса, создать рабочую ячейку. Такая ячейка всегда будет иметь поставщиков и потребителей, доставку и получение, вход и выход.

Когда ячейка создана, то её работа, с учетом требований потребителей и возможностей поставщиков, должна удовлетворять следующим условиям:

- ячейка сама определяет вход и требования к продукции поставщиков и показатели для мониторинга соответствия их нормативно – технической документации (НТД);
- на основе требований потребителей и НТД определяют выходы ячейки и показатели для мониторинга соответствия им.

Применение ячейки преследует следующие цели: во-первых, это позволяет определить потребности и пожелания различных заинтересованных сторон. Таким образом, устанавливается взаимопонимание между сформированной ячейкой и её поставщиками и потребителями; во-вторых, это поможет ввести показатели, которые используются для мониторинга уровня качества на входе и на выходе ячейки и процесса. Всё это способствует тому, что на переделку будет направлено минимальное число изделий с дефектами и такая система препятствует попаданию бракованных изделий к потребителю.

Для эффективного функционирования ячейки в рамках бизнес - процессов можно предложить следующие этапы её построения и анализа функционирования.

Этапы построения ячейки.

1. Первый шаг – создание ячейки. Если созданная ячейка не может четко определить вход, что она получает, а также — выход, что

она поставляет, ячейку надо перестроить или ликвидировать.

2. Анализ работы ячейки следует начинать со стороны потребителя, так как на основании его желаний и требований НТД определяются требования к продукции ячейки и поставщика. Выявляют постоянных потребителей ячейки. Для этого необходимо сосредоточиться на продуктах, поставляемых ячейкой. Ячейки, занятые материальным производством, обычно имеют одного-двух потребителей, а ячейки, занятые администрированием, как правило, имеют много потребителей. В последнем случае следует сконцентрировать анализ вокруг наиболее важных клиентов.

3. Выявление требований, определяющих набор выходов ячейки, и возможных нечётких требований. Особое внимание следует уделять базовым требованиям, так как они, как правило, оговорены в техническом задании.

4. Анализ входа. После завершения фиксации потребителей выхода ячейки надо зафиксировать поставщиков и вход.

5. Определение поставщиков элементов входа и соответствующих требований проводится с учётом запросов поставщиков.

6. Определение процессов, которые выполняются в ячейке для преобразования входов в выходы. Для анализа этой работы используется блок-схема процесса.

7. Для бизнес-процессов ячейки определяют показатели, назначение которых в том, чтобы дать возможность ячейке вести мониторинг совершенствования удовлетворенности потребителя и своей внутренней эффективности и производительности. Число показателей должно быть минимально необходимым, чтобы эффективно организовать работу.

8. Выработка предложений по улучшению на основе анализа требований к входам и выходам на основе определенных ранее требований и показателей при мониторинге процессов.

Применение CALS -технологий позволяет дополнительно обеспечить повышение эффективности производства за счёт:

- моделирования материальных, информационных и финансовых потоков в ячейках, характеризующих процессы производства продукции, с целью выбора оптимального комплекса технологических процессов, обеспечивающих достижение заданных технико-экономических параметров продукции, выпуск которой планируется;

- интегрированной информационной системы сопровождения продукции на всех этапах её

ЖЦ, обеспечивающей сведение к минимуму производственных издержек в ячейке;

- системы информационного взаимодействия с субподрядчиками, обеспечивающей высокую эффективность процессов материально-технического снабжения ячеек;

- интегрированной информационной системы управления качеством продукции на всех этапах её ЖЦ;

- интегрированной информационной системы взаимодействия с потребителями продукции, обеспечивающей сведение к минимуму затрат на техническое обслуживание и ремонт.

Реализация такого подхода приводит к изменению взгляда на производство и позволяет устранить те области в потоке стоимости, которые не добавляют стоимости продукту, а повышают затраты. Предлагаемый подход к организации с использованием ячеек и применением CALS - технологий поможет осуществить реализуемые в корпорации «Тойота» методы в устранении напрасных трат на производство или на японском языке «Muda».

Это следующие формы напрасных трат:

1. Напрасная трата на перепроизводство.

2. Напрасная трата на запасы.

3. Напрасно растроченное движение.

4. Ожидание.

5. Напрасная трата на корректировку необходимости исправления, сортировки и ремонта продуктов, которые плохо сделаны.

6. Напрасная трата перемещения, вызванная ненужной транспортировкой, например между цехами и т.д.

7. Напрасная трата вследствие совершения большего числа процессов, чем необходимо.

8. Напрасно потраченный капитал вследствие излишне закупленных станков и оборудования производство.

Предлагаемые методы, обеспечивающие создание единой информационной среды для процессов проектирования, производства, испытаний, поставки и эксплуатации продукции, способствуют оптимизации организационной среды предприятия через построение системы управления на основе принципов TQM.

Анализ сведений таблицы указывает на то, что принципы TQM уже заложены в идеологии CALS - систем. Следовательно, внедрение на предприятиях интегрированных информационных технологий означает если не автоматическую, то, по крайней мере, более быструю реализацию принципов TQM. Таким образом, составляющие CALS в значительной степени соответствуют идеологии TQM, что показано в таблице и влияют на компоненты органи-

зационно - производственной среды предприятия. При этом становится возможным решение задачи преобразования ЖЦ изделия в высокоавтоматизированный процесс информационного обеспечения.

Заключение. Совместное использование методологии процессного подхода к построению и функционированию производства на основе принципов TQM и CALS технологий в сочетании с построением производственных ячеек предоставит возможность к более эффективному функционированию организации. Преобладание в организации процессного подхода существенно меняет логику функционирования и механизм её управления. Процессы «разрезают» организацию поперек, пересекая границы функциональных подразделений, что обеспечивает разрушение барьеров между подразделениями и снятие ограничений для совершенствования. Использование структуры на основе ячеек освобождает высшее руководство от оперативного управления, позволяя ему сосредоточиться на стратегических вопросах, передавая часть функций управления ячейкам. Это способствует повышению синергетики процессов и приводит к тому, что результаты одного процесса способствуют улучшению результатов другого.

Библиографический список

1. *Андерсен Бьёрн.* Бизнес-процессы. Инструменты совершенствования / пер. с англ. С.В. Ариничева / науч. ред. Ю.П. Адлер. - М.: РИА «Стандарты и качество», 2003.- 272 с., ил. - (Серия «Практический менеджмент»). ISBN 5-94938-012.
2. *Казакова Н.В., Фаткин В.А.* Реинжиниринг бизнес – процессов: учебно-практическое пособие (учебно-методическая разработка)/ Рязанская государственная радиотехническая академия, – Рязань, 2005.-120 с.
3. *Белов А.В.* Особенности разработки интегрированной информационной среды в CALS-техно-

логиях / А.В. Белов, С.Я. Гродзенский, В.В. Сидорин. Проблемы качества, безопасности и диагностики в условиях информ. о-ва. КБД - Инфо - 2005: матер. науч.- практ. конф., Сочи, 1-10 окт. 2005. - М.: МГИЭМ, 2005. - С. 163-165.

4. *Бриндигов А.Н.* Состояние и перспективы развития ИПИ - технологий на предприятиях ОПК / А.Н. Бриндигов, В.В. Барабанов.- Качество и ИПИ (CALS) - технологии. - 2005. - N 4(8). - С. 31-43.

5. *Джордж С.* Всеобщее управление качеством / С.Джордж, А. Ваймерскирх. - Издательство Виктория плюс, 2002. - 256 с.

6. *Кеворков С.* Поддержка жизненного цикла изделия // Открытые системы. - 2005. - N 12(116). - С. 54-58.

7. *Кирпичев И.Г.* Разработка методологии построения и функционального развития информационно - аналитической системы мониторинга жизненного цикла компонентов воздушных судов: автореф. дис. д-ра техн. наук / Гос. НИИ гражд. авиации. - М., 2006. - 38 с.

8. *Костюков В.Д.* CALS-технологии в технологической подготовке производства авиакосмической техники: учеб. пособие / В.Д.Костюков, Э.М.Годин, В.П. Соколов [и др.] - М.: МАИ, 2005. - 552 с.

9. *Надеев А.И.* Принципы построения автоматизированных систем поддержки жизненного цикла инновационных продуктов / А.И. Надеев, А.И. Попова, А.В. Сурина, Ю.К. Свечников, Датчики и системы. - 2006. - С. 59-63.

10. *Сабина А.Л.* Особенности организации жизненного цикла машиностроительной продукции специального назначения / А.Л. Сабина, И.В. Сычева Изв. Тульск. гос. ун-та. Сер. Технол. машиностр. - 2004. - С. 117-128.

11. *Судов Е.В.* Интегрированная информационная поддержка жизненного цикла машиностроительной продукции. Принципы. Технологии. Методы. Модели/ Е.В. Судов - М.: ИД "МВМ", 2003. - 264 с. - Библиогр.: С. 257-261.

12. *Тарасов А.П.* Система информационной поддержки жизненного цикла вооружения и военной техники. Вопросы стандартизации / А.П. Тарасов ИТПП. - 2006. - С. 26-29

УДК 004.75

И.В. Логинов

ПЛАНИРОВАНИЕ ПРОЦЕССОВ ОБРАБОТКИ ЗАПРОСОВ В ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ АСУП НА ОСНОВЕ ФУНКЦИЙ ПОЛЕЗНОСТИ

В работе предложен способ планирования процессов обработки вычислительных запросов в однородных вычислительных сетях на основе функций полезности, заключающийся в распределении вычислительных

ресурсов между запросами из очереди в соответствии с прогнозируемой полезностью результата обработки.

Ключевые слова: планирование, обработка запросов, вычислительная сеть, функция полезности, прогнозирование.

Введение. Широкое внедрение технологий распределенных вычислений в информационно-вычислительных системах и АСУП инновационных промышленных предприятий в целях обеспечения непрерывности управления как оперативного, так и стратегического, моделирования рынков и жизненных циклов продукции, прогнозирования потребительского поведения и объемов продаж, определяют особую актуальность обеспечения качества обслуживания операторов АСУП при решении ими ресурсоемких задач. В работе предлагается повысить качество обслуживания операторов, генерирующих ресурсоемкие запросы (моделирование, прогнозирование, обработка многомерных статистических данных) к вычислительной системе (ВС) промышленного предприятия путем повышения своевременности и полезности результатов их обработки. Целью работы является разработка способа планирования процессов обработки ресурсоемких запросов в однородных ВС промышленных предприятий, позволяющих повысить своевременность представления результатов операторам.

Научная постановка задачи. Результат обработки запроса обладает полезностью, которую можно выразить в условно-денежных единицах, отражающих величину повышения эффективности решения задачи (принятия управленческого решения). Использование экономического подхода к планированию в ВС рассматривалось в работах [1,2,3,4], которые показали возможность его применения. В настоящей работе рассматривается ВС, каждый запрос к которой (в отличие от рассмотренного в [1] случая с функциями полезности вида Кобба-Дугласа) обладает непрерывной убывающей функцией полезности, представляющей собой зависимость ценности результата обработки запроса для оператора АСУП от времени $\xi = \xi(t)$. Цена вычислительного ресурса в отличие от способа, предложенного в [3,5], устанавливается равной максимальной величине спроса конкурирующих запросов из очереди. Бюджеты пользователей определяются администратором ВС в соответствии с правилом, предложенным в [3]. Управление потоком запросов осуществляется в отличие от способа, представленного в работе [6], на основе модели управляемого ресурса. Поставщик вычислительного ресурса максимизирует суммарную полез-

ность результатов обработки потока запросов за промежутки времени:

$$Z = \sum_{i=1}^N p_i = \sum_{i=1}^N \xi_i(t_i) \rightarrow \max, \quad (1)$$

где $p_i = \xi_i(t_i)$ - полезность результата обработки i -го запроса, T - промежуток времени планирования, t_i - момент времени завершения обработки i -го запроса, $N = N(A, T)$ - количество обработанных запросов за время T с использованием алгоритма управления A .

Модель вычислительной системы. ВС состоит из M однородных вычислительных узлов. Каждый узел имеет производительность V . На вход ВС поступает поток запросов с интенсивностью λ , распределенный по известному закону. Параметры закона распределения определяются экспериментально. Оператор АСУП, сгенерировавший запрос, формирует для него функцию полезности $\xi_i(t)$ с использованием принятых в вычислительной системе методик. При наличии свободного вычислительного узла поступивший запрос назначается на него, в противном случае поступает в неупорядоченную очередь. При наличии очереди и освобождении вычислительного узла для обработки выбирается один из запросов в очереди (размером K) по заданному правилу A . При этом в системе управления составляется план распределения ресурсов ВС Π .

Обозначим g_j - последовательность обработки запросов на j -м вычислительном узле. $g_j = q_{j1} \rightarrow \dots \rightarrow q_{jk} \rightarrow \dots \rightarrow q_{jN_j}$, где N_j - количество запросов в g_j последовательности. План распределения ресурсов Π представляет собой последовательность обработки запросов на всех вычислительных узлах вычислительной системы и равен множеству всех последовательностей обработки запросов: $\Pi = \{g_j, j = \overline{1, M}\}$.

После обработки на вычислительном узле результаты предоставляются оператору.

Способ составления плана распределения ресурсов ВС между запросами в очереди. Для максимизации суммарной полезности запросов согласно выражению (1) составляется такой план распределения ресурсов, который обеспечивает максимальную прогнозируемую полезность результатов обработки запросов:

$$Z_{\Pi} = \sum_i^K \hat{p}_i \rightarrow \max, \quad (2)$$

где \hat{p}_i - прогноз полезности результата обработки i -го запроса, K - количество запросов в очереди. Расчет прогноза полезности проводится на средневероятный случай:

$$\hat{p}_i = \int_0^{\infty} \xi_i(t + \Delta t_i) \varphi_i(t) dt. \quad (3)$$

где Δt_i - промежуток времени прошедший с момента постановки запроса в очередь и началом работы алгоритма планирования, $\varphi_i(t)$ - функция плотности вероятности завершения обработки запроса.

Функция плотности вероятности завершения обработки запроса рассчитывается на основе регрессионных методов и является различной для каждого класса запросов. Для каждого класса запросов рассчитываются нормированные функции плотности вероятности $\bar{\varphi}(t)$, на основе которых с учетом ресурсоемкости запросов рассчитываются $\varphi_i(t)$. Варианты широко используемых в процессе планирования функций плотности вероятности времени обработки представлены на рисунке 1, где MO - математическое ожидание, а σ - среднеквадратичное отклонение случайной величины.

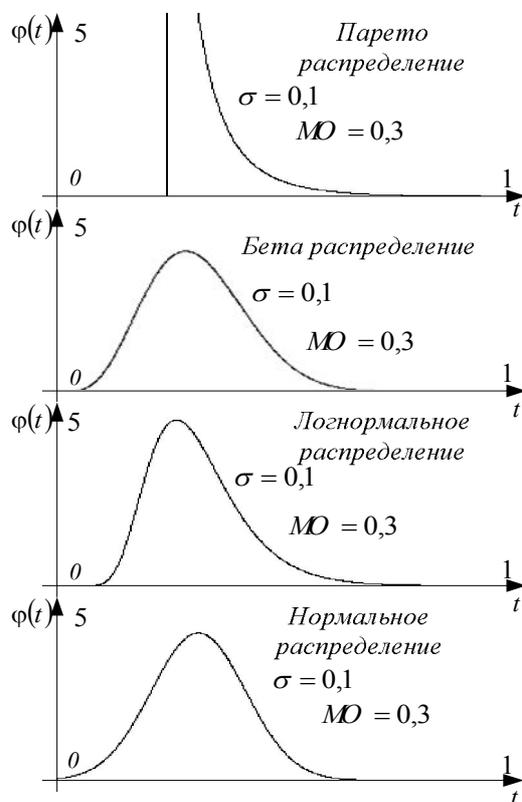


Рисунок 1 – Варианты нормированных функций плотности вероятности времени обработки запроса

Исходными данными для планирования выступают: целевая функция Z_{Π} согласно (2), множество функций полезности для всех запросов в очереди $\xi_i(t), i = \overline{1, K}$, множество функций плотности вероятности времени обработки запросов $\varphi_i(t), i = \overline{1, K}$, максимальная длина последовательности g обработки запросов - $l = \max_j N_j$.

Способ предполагает назначение на обработку наиболее своевременного запроса (FMT – first most timeliest). Предлагается следующая последовательность действий по формированию плана распределения ресурсов.

1. Определение l – максимальной длины последовательности запросов g , учитываемых в плане. Максимальная длина последовательности равна количеству запросов находящихся на обработке в ВС (сумма находящихся в очереди и обрабатываемых на вычислительных узлах).

2. Генерация начального плана Π_i и расчет прогноза его полезности $\hat{p}(\Pi)$.

3. Циклическая генерация семейства альтернатив планов Π_i на основе алгоритмов оптимизации, расчет их полезности и выбор лучших по критерию максимума полезности. Цикл продолжается до срабатывания критерия останова, заданного для используемого алгоритма оптимизации.

4. Формирование нового плана и его вывод для реализации.

На освободившийся вычислительный узел назначается первый запрос из плана Π .

Расчет прогноза полезности i -го плана распределения ресурсов проводится по формулам (4-6):

$$\hat{p}(\Pi_i) = \sum_{j=1}^M \sum_{k=1}^{N_j} \hat{p}_{ijk}, \quad (4)$$

где \hat{p}_{ijk} в соответствии с (3) - прогноз полезности результата обработки k -го запроса на j -м вычислительном узле (в g_j последовательности обработки запросов) в i -м плане распределения ресурсов при условии, что сначала будет обработан $k-1$ запрос:

$$\hat{p}_{ijk} = \int_0^{\infty} \xi_{ijk}(t + \Delta t_{ijk}) \varphi'_{ijk}(t) dt, \quad (5)$$

где $\varphi'_{ijk}(t)$ - условная функция плотности вероятности времени обработки i -го запроса при условии, что был обработан $k-1$ запрос:

$$\varphi'_{ijk}(t) = \int_0^{\infty} \varphi'_{ij(k-1)}(t') \varphi_{ijk}(t+t') dt' \quad (6)$$

Пример расчета плана распределения ресурсов: вычислительная система состоит из одного узла, в очереди на обработку находятся два запроса: q_1 и q_2 . При максимальной длине последовательности обработки запросов, равной одному $l = 1$:

$$\int_{t=0}^{\infty} \xi_1(t + \Delta t_1) \varphi_1(t) dt \geq \int_{t=0}^{\infty} \xi_2(t + \Delta t_2) \varphi_2(t) dt \Rightarrow \Pi = g = (q_1) \quad (7)$$

При максимальной длине последовательности обработки запросов, равной двум $l = 2$:

$$\begin{aligned} & \int_{t=0}^{\infty} \xi_1(t + \Delta t_1) \varphi_1(t) dt + \\ & + \int_{t=0}^{\infty} \xi_2(t + \Delta t_2) \int_{t'=0}^{\infty} \varphi_1(t') \varphi_2(t+t') dt' dt \geq \\ & \int_{t=0}^{\infty} \xi_2(t + \Delta t_2) \varphi_2(t) dt + \\ & + \int_{t=0}^{\infty} \xi_1(t + \Delta t_1) \int_{t'=0}^{\infty} \varphi_2(t') \varphi_1(t+t') dt' dt \\ & \Rightarrow \Pi = g = (q_1 \rightarrow q_2). \end{aligned} \quad (8)$$

Задача нахождения плана распределения ресурсов является пр-полной и сложность ее решения экспоненциально зависит от количества запросов в очереди, что не позволяет использовать методы динамического программирования при большой длине очереди. Для получения приближенного решения на практике возможно ограничение максимальной длины последовательности l , а также использовании эвристических и приближенных методов оптимизации (нейронных, генетических, локального поиска). Пример расчета плана распределения ресурсов для случая трех запросов в очереди представлен на рисунке 2.

$$\begin{aligned} & \text{при } l=1: \Pi = g = (q_1) \\ & \text{при } l=2: \Pi = g = (q_3 \rightarrow q_1) \\ & \text{при } l=3: \Pi = g = (q_3 \rightarrow q_1 \rightarrow q_2) \end{aligned} \quad (9)$$

Экспериментальное исследование. Практическое исследование возможности планирования обработки запросов в ВС на основе FMT проведено путем сравнительного анализа с широко используемыми в вычислительных сетях АСУП способов планирования на основе FCFS и FCLS по показателю полезности с использованием имитационного моделирования [4,7].

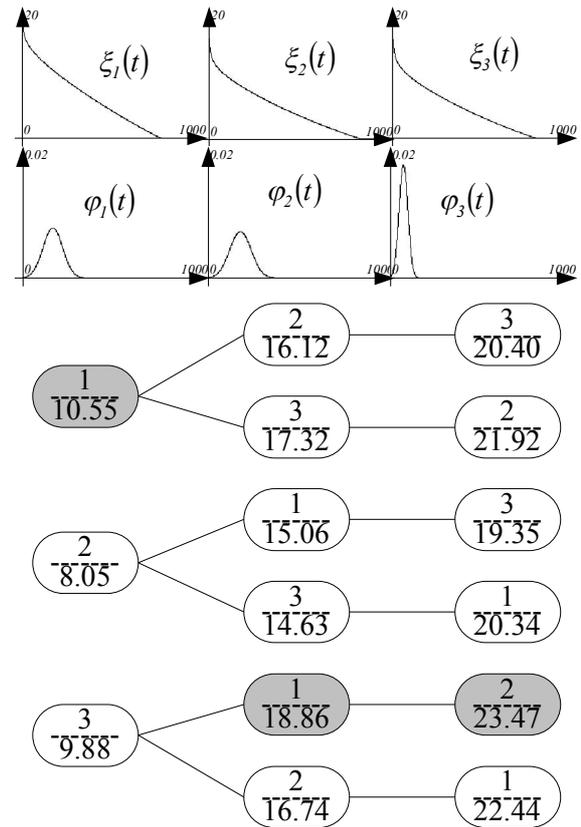


Рисунок 2 – Пример расчета плана распределения ресурсов вычислительной системы

Условия проведения эксперимента. Количество вычислительных узлов $n = 1..2$; производительность узлов одинаковая и постоянная $V = const$. Задержки на запуск обработки и выдачу результата оператору являются много меньшими времени обработки запроса. На вход ВС поступает простейший поток запросов с интенсивностью λ . Функции полезности являются монотонно убывающими неотрицательными функциями. Время планирования для всех дисциплин обслуживания является сопоставимым и много меньшим времени обработки. В процессе эксперимента получены зависимости (рисунок 3):

– полезности от отношения среднего времени обработки запроса $t_{обp}$ и длины интервала положительности функции полезности и при нагрузке $\rho = 0.9$:

$$k = \frac{\bar{t}_{обp}}{t_{\xi(t)>0}} \quad (10)$$

– полезности от нагрузки.

На рисунке 3 представлены зависимости нормированной полезности результатов обработки запросов для исследуемых альтернатив дисциплин обслуживания a , рассчитанные по формуле:

$$z = \frac{Z_a}{Z_{FCLS}} = \frac{\sum_{j=0}^J \xi_j(t_j^{30})}{\sum_{i=0}^I \xi_i(t_i^{30})}, \quad (11)$$

где J - количество запросов, обработанных с использованием альтернативы a , I - количество запросов, обработанных с использованием дисциплины обслуживания FCLS за время эксперимента.

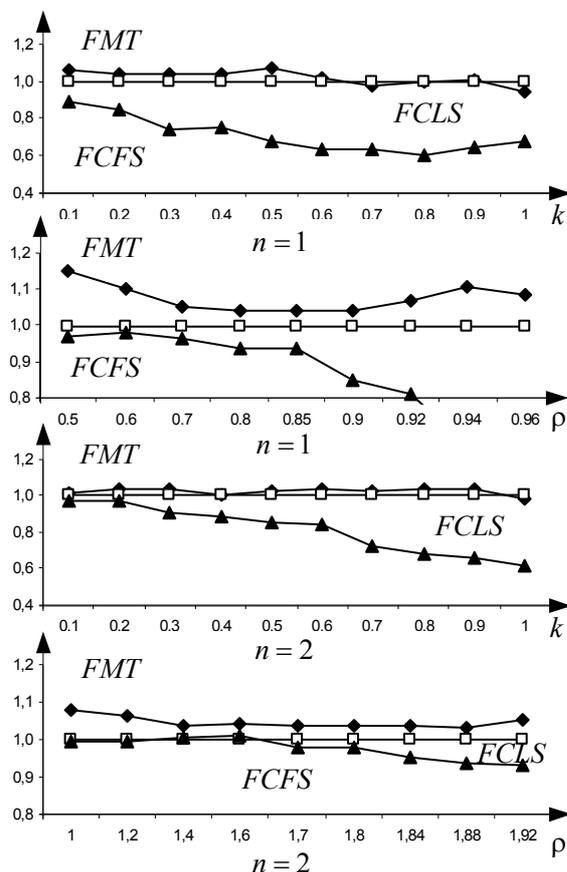


Рисунок 3 – Результаты эксперимента

Результаты эксперимента показывают, что планирование обработки запросов в ВС АСУП на основе функций полезности позволяет повысить качество обслуживания операторов при решении ими ресурсоемких задач по сравнению с дисциплинами обслуживания FCFS

и FCLS по показателю полезности результатов обработки запросов.

Заключение. Описанный выше способ планирования процессов обработки запросов FMT опирается на использование функций полезности результата обработки запросов для операторов АСУП, сгенерировавших запрос. Проведенные исследования показали возможность применения данного способа и его эффективность по сравнению с широко используемыми в однородных вычислительных системах. Направлением дальнейших исследований является обобщение способа на случай гетерогенных вычислительных систем.

Библиографический список

1. Бредихин С.В., Тиунова Е.М., Хуторецкий А.Б. Ценовое согласование спроса и предложения при распределении мощности многопроцессорной системы // Сиб. журн. индустр. матем., 2007, 10:3, 20–28.
2. Buyya R. An Economy Driven Resource Management Architecture For Global Computational Power Grids / R. Buyya, D. Abramson, J. Giddy. International Conference on Parallel and Distributed Processing. Techniques and Applications, 2000.
3. Киселев А.В., Голосов П.Е. Планирование заданий на основе экономической модели как подход к обеспечению качества обслуживания в распределенной вычислительной среде // Сборник трудов конференции «Научный сервис в сети интернет - 2008». 2008. С. 216-220.
4. Yeo C.S., Buyya R. Integrated Risk Analysis for a Commercial Computing Service in Utility Computing, Journal of Grid Computing, Volume 7, Number 1, Pages: 1-24, ISSN: 1570-7873, Springer, Germany, March 2009.
5. Бредихин С.В., Вялков И.А., Савченко И.Ю., Хуторецкий А.Б. Две модели ценового согласования при распределении вычислительных ресурсов // Сиб. журн. индустр. матем., 2006, 9:1, 28–46.
6. Altman E. Flow control using the theory of zero sum markov games. 1992. – 17 p.
7. Desic S. Different approaches to experiments for load balancing method comparison // 6th international conference on Telecommunications. 2001. P. 177-184.