

УДК 681.3.012.519.715

## АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СИСТЕМ

В.Г.Хорошевский

ЭВМ является широко распространенным достаточно производительным средством индустрии обработки информации. Характерным стало и то, что машины в расширенных и минимальных конфигурациях (процессор с памятью) выступают в качестве функциональных элементов больших ансамблей (коллективов), таких как параллельные вычислительные системы (ВС) и сети. Именно поэтому анализ производительности ЭВМ представляет не только самостоятельный интерес, но и является необходимым этапом изучения эффективности вычислительных систем. Под производительностью ЭВМ и ВС понимается способность каждой из них обработать определенный объем информации. Как правило, когда говорят о производительности, то понимают под этим потенциальную возможность ЭВМ или ВС по обработке информации (а не реальную, учитывающую аномальности в работе ЭВМ или ВС, например, учитывающую потери из-за отказов, из-за профилактического обслуживания и т.п.).

### 1. Показатели производительности ЭВМ

В процессе обработки информации в ЭВМ реализуются те или иные операции из ее набора операций. Состав набора операций безусловно определяет производительность ЭВМ.

Для оценки способности ЭВМ производить обработку информации используют количественные характеристики или показатели производительности, которые обычно связаны с временем выполнения операций. Время выполнения операции в общем случае складывается из времени выборки (из памяти ЭВМ в процессор) команды (кода операции и адресов операндов), времени чтения операндов, времени реализации собственно операции и времени занесения результатов в па-

мять. При выполнении последовательности операций (точнее последовательности команд, как это имеет место во всех программах решения задач) возможны совмещения во времени выполнения нескольких операций. При этом всегда имеет место зависимость времени выполнения операции от времени обращения к памяти.

В качестве простейших показателей производительности ЭВМ можно использовать количества однотипных операций, выполняемых в единицу времени. Для оценки производительности ЭВМ получили распространение числа операций, например, сложения с фиксированной запятой (регистр-регистр), сложения с плавающей запятой, умножения и деления, выполняемых в секунду. Так как в общем случае длительности операций даже при одних и тех же операндах зависят от их типов, то для характеристики производительности ЭВМ целесообразно использовать средние числа операций, выполняемых в единицу времени.

Введем следующие показатели: номинальное, среднее и среднее эффективное быстродействие ЭВМ. Номинальным быстродействием ЭВМ назовем среднее число операций, выполняемых машиной (при равновероятном их выборе) в единицу времени (секунду) при работе только с оперативным запоминающим устройством [I-3]. (Заметим, что в состав оперативного запоминающего устройства включается собственно оперативная память и регистры процессора.)

Пусть  $\{n_1, n_2, \dots, n_j, \dots, n_k\}$  - часть набора операций ЭВМ, не требующих обращений процессора к внешним устройствам и внешней памяти;  $\tau_j$  - время (в секундах) выполнения операции  $n_j$ , тогда номинальное быстродействие выражается в виде

$$\omega' = k \left[ \sum_{j=1}^k \tau_j \right]^{-1}. \quad (1)$$

Очевидно, что при реализации на ЭВМ реальных программ решения задач имеет место не равновероятное использование тех или иных операций. Следовательно, даже при работе с оперативной памятью ЭВМ будет выполняться в единицу времени различное число операций при решении задач различных типов. Кроме того, при решении задачи требуются, в общем случае, затраты машинного времени на ввод программы и данных, на обращение к внешней памяти, на работу операционной системы, на вывод результатов и т.п. Из сказанного видно, что число операций, связанных собственно с решением задач и выпол-

няемых ЭВМ в единицу времени при работе с оперативной памятью, будет меньше номинального быстродействия. Поэтому для характеристики производительности ЭВМ при решении задач целесообразно ввести дополнительные показатели.

Пусть  $\{I_1, I_2, \dots, I_1, \dots, I_L\}$  - набор типовых задач;  $v_i^*$  - число операций, непосредственно входящих в программу решения задачи  $I_i$ ;  $t_i$  - время (в секундах) решения задачи  $I_i$  (в  $t_i$  входит время, расходуемое ЭВМ собственно на счет, и дополнительные затраты машинного времени при решении  $I_i$ ). Быстродействием ЭВМ при решении типовой задачи  $I_i$ ,  $i \in \{1, 2, \dots, L\}$ , назовем величину

$$\omega_i^* = v_i^*/t_i. \quad (2)$$

Величина  $1/\omega_i^*$  является средним временем выполнения одной операции при решении задачи типа  $i$ ,  $i \in \{1, 2, \dots, L\}$ . Пусть далее  $\{\pi_1, \pi_2, \dots, \pi_1, \dots, \pi_L\}$  - распределение вероятностей спроса на типовые задачи  $I_1, I_2, \dots, I_1, \dots, I_L$ ,  $\sum_{i=1}^L \pi_i = 1$ , тогда  $\sum_{i=1}^L \pi_i/\omega_i^*$  будет средним временем выполнения одной операции при решении набора типовых задач. Средним быстродействием ЭВМ назовем величину

$$\omega^* = \left[ \sum_{i=1}^L \pi_i/\omega_i^* \right]^{-1}. \quad (3)$$

При сравнении производительности нескольких ЭВМ необходима "стандартизация" операций и форматов операндов. В качестве стандартной может быть взята операция, через которую возможно выразить все остальные операции в каждой из сравниваемых машин. В качестве стандартного формата операндов может служить одна из структурных единиц информации, например, байт. Тогда в этих условиях в качестве показателей производительности ЭВМ целесообразно использовать эффективное и среднее эффективное быстродействия.

Эффективным быстродействием ЭВМ при решении типовой задачи  $I_i$ ,  $i \in \{1, 2, \dots, L\}$ , назовем величину

$$\omega_i = v_i/t_i, \quad (4)$$

где  $v_i$  - число стандартных операций, с помощью которого можно интерпретировать программу решения задачи  $I_i$ ;  $t_i$  - время решения задачи  $I_i$  на ЭВМ (включающее время реализации собственно программы и накладные расходы времени).

Среднее эффективное быстродействие ЭВМ определяется формулой

$$\omega = \left[ \sum_{i=1}^L \pi_i / \omega_i \right]^{-1}, \quad (5)$$

где  $\pi_i$  - вероятность спроса на типовую задачу  $I_i$  (или вес типовой задачи  $I_i$ ), а  $\omega_i$  - эффективное быстродействие ЭВМ при решении  $I_i$ .

Каждый из введенных выше показателей производительности (1)-(5) определяется при условии, что ЭВМ исправна; в противном случае эти показатели равны нулю.

Таким образом, введенные показатели производительности выражают значение потенциально возможной производительности ЭВМ, причем номинальное быстродействие (1) информирует о потенциально возможной производительности при условии, что ЭВМ исправна, а быстродействие (2), среднее быстродействие (3), эффективное (4) и среднее эффективное (5) быстродействия - при условии, что машина исправна и занята решением задач ("полезной работой").

Очевидно, что число, выражающее то или иное быстродействие ЭВМ, можно приравнять к некоторой условной единице. Тогда производительность  $\omega(\tau)$  ЭВМ в произвольный момент времени  $\tau$  будет равна либо этой условной единице, либо нулю в зависимости от того, исправна или нет ЭВМ в этот момент времени. Ясно, что функция  $\omega(\tau)$  является случайной. Через эту функцию удобно определить вероятностные показатели качества функционирования ЭВМ, в частности, количественные характеристики надежности. Например, функция готовности ЭВМ [3]:  $a(i, t) = P\{i; \omega(t)=1\}$  есть вероятность того, что (в условиях потока отказов и восстановлений) машина, начавшая функционировать в состоянии  $i \in \{0, 1\}$ , будет иметь в момент времени  $t > 0$  производительность, равную единице (т.е. равную потенциально возможной).

## 2. Анализ производительности вычислительных систем

В вычислительных системах ядро, обеспечивающее выполнение основного объема работ по переработке информации, представляет собой композицию вычислительных модулей и сети связи, через которую осуществляется взаимодействие между модулями. Вычислительные модули высокопроизводительных систем, как правило, однотипны [4-8].

В качестве вычислительных модулей, например, могут выступать либо устройства для реализации одной из арифметических операций, либо элементарный процессор (с простейшими средствами управления вычислительным процессом), либо элементарная машина (ЭМ, дополненная средствами системного управления). Сеть связи представляется в виде графа, вершинам которого сопоставлены вычислительные модули, а ребрам — каналы связи между ними. Сеть связи мощных промышленных ВС, как правило, представляется однородным графом.

Среди семейства ВС системы с программируемой структурой (или однородные ВС) по своим возможностям являются наиболее общим классом. В таких системах достаточно полно воплощены перспективные принципы алгоритмического параллелизма при обработке информации, программируемости (адаптации) структуры под классы решаемых задач и сферы применения, гомогенности состава и регулярности (однородности) связей между вычислительными модулями. Под вычислительной системой с программируемой структурой понимается совокупность элементарных машин (ЭМ), функциональное взаимодействие между которыми осуществляется через программно настраиваемую сеть связи [3,4].

Рассмотрим ряд аспектов анализа производительности ВС. При этом не оставим без внимания собственно процесс решения задач на неабсолютно надежных системах. Для чего введем, в частности, показатели осуществимости решения задач, которые вполне удовлетворяют требованиям инженерного анализа функционирования ВС.

#### I. Способы обработки данных.

Технология решения произвольной задачи на ВС включает следующие моменты [5]:

- выбор способа обработки данных,
- разработку алгоритма (в общем случае параллельного), эффективно реализуемого на ВС (или на ее части — подсистеме) при заданной структуре системы и выбранном способе обработки данных;
- запись (параллельного) алгоритма решения задачи на языке (высокого уровня);
- получение объектной программы решения задачи на системе.

Выделяются распределенный, матричный и магистральный (поточный) способы обработки информации. Последние два способа обработки информации получили промышленное воплощение в виде высокопроизводительных (порядка  $10^7 - 10^8$  оп/с) систем. Построены матричная система ILLIAC-IV [3,4] и образцы (например, из 32x32 эле-

ментарных процессоров) системы DAP [4], освоено выпуск отечественной системы ПС-2000 [6] и др. Широкое распространение получили высокопроизводительные системы ASC, STAR-100, "Cray-1", "Cyber-205", BSP [3,4,7,8] и векторные процессоры IBM 3838, AP-120B, AP-190L, FPS-100, FPS-164 [7,8], основанные на магистральном способе обработки информации. Принципы, положенные в основу ВС с программируемой структурой, позволяют реализовывать в них любой из названных выше способов обработки данных.

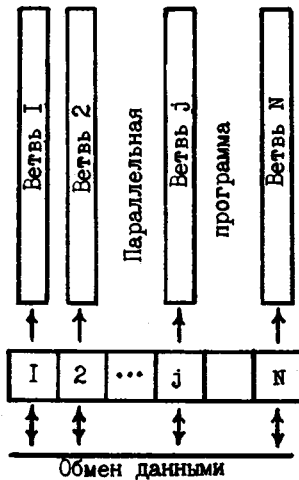


Рис. 1

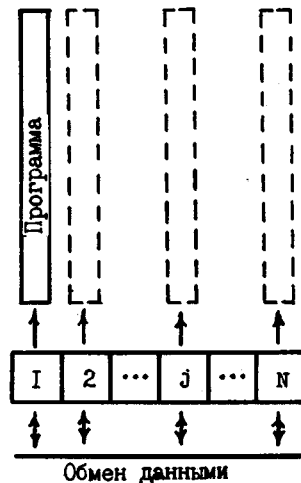


Рис. 2

При распределенной обработке информации (рис.1) программы и данные расчленяются и рассредоточиваются по элементарным машинам ВС. Допустимо построение адаптирующихся параллельных программ, число ветвей в которых в процессе их реализации соответствует числу (исправных) ЭМ в системе. Способ распределенной обработки данных был теоретически и экспериментально исследован в широком диапазоне классов сложных задач.

В случае матричной обработки данных (рис.2) программа вычислений содержится в одной (управляющей) ЭМ, а данные однородно распределяются по всем машинам ВС (или подсистемы). Процесс решения задачи состоит из чередующихся процедур: из рассылки команд из управляющей ЭМ остальным машинам и из исполнения этих команд всеми машинами, но каждой над своими операндами.

Матричный способ в сравнении с распределенным дает экономию в использовании (распределенной по ЭМ) памяти ВС. Однако данному способу присущ недостаток, заключающийся в неоднородном использовании машин и, в частности, в неоднородной нагрузке на их память. Этому недостатка лишен обобщенный матричный способ обработки информации (рис.3). При последнем способе программа не целиком помещается в одной ЭМ, а предварительно сегментируется (не распараллеливается, а сегментируется!) и затем посегментно размещается в памяти машин. Последовательность сегментов, составляющих программу, может быть размещена в памяти машин, например, так, чтобы номер распределенного в машину сегмента совпадал с ее номером. И для распределенного, и для матричного способов обработки информации характерно то, что в процессе решения задачи имеют место обмены данными между ЭМ системы.

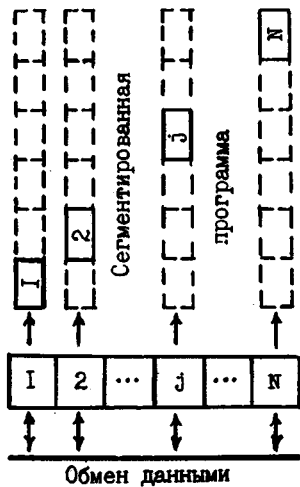


Рис. 3

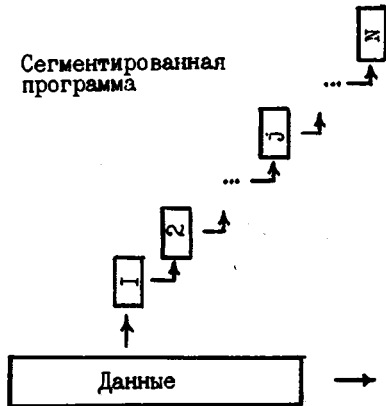


Рис. 4

При магистральном способе обработки данных (рис.4) структура ВС предварительно настраивается так, что машины образуют магистраль (или pipeline - "трубопровод", или конвейер, или "линейку", или "кольцо"). Затем осуществляется сегментирование программы и размещение в машинах ВС последовательности полученных сегментов в соответствии со структурой магистрали. Размещение данных может быть сосредоточенным (например, на внешней па-

мью одной ЭМ) или распределенным (по памяти всех машин магистрали). В процессе решения задачи данные "пропускаются" через последовательность машин, составляющих магистраль.

Заметим, что верхние части рис.1-4 соответствуют программам, а нижние - данным. Стрелками показаны направления потоков данных. Номер  $j \in \{1, 2, \dots, N\}$ , приписанный к блокам данных и программ (к ветви или сегменту), соответствует номеру ЭМ, в памяти которой они размещены. Пунктиром изображены те недостающие части программы, которые будут получены машиной в процессе решения задачи от ЭМ, хранящих программу или ее сегменты.

Итак, при распределенном способе обработки информации полностью используются возможности вычислительной системы с архитектурой MIMD. Матричный и магистральный способы обработки информации обеспечивают частичное использование возможностей архитектуры BC. Архитектура MIMD при первом способе трансформируется в архитектуру SIMD, а при втором - в архитектуру MISD[3].

2. Понятие о сложности задач. Эффективность функционирования BC при решении задачи зависит, как следует из п.1, и от способа обработки данных, и от степени адекватности вложения структурной схемы алгоритма решения в структуру BC, и от числа машин, на которых решается задача, и определяется в конечном счете качеством объектной программы. Среди показателей качества программы можно использовать

$$\delta(V, n) = \tau(V, n) / \theta(V, n), \quad (6)$$

где  $V$  - количество операций, которые необходимо выполнить при решении задачи на BC,  $n$  - число параллельных ветвей или число машин, на которых решается задача,  $n \geq 2$ ,  $\tau(V, n)$  - время, затрачиваемое на настройку (программирование структуры) системы, на реализацию обменов информацией между ветвями (машинами),  $\theta(V, n)$  - время, расходуемое системой собственно на счет.

На основе анализа задач и опыта их решения на вычислительных системах установлено, что при  $n = \text{const}$  показатель (6) асимптотически стремится к нулю с ростом объема операций в задаче, т.е. имеет место:  $\delta(V, n) \rightarrow 0$  при  $V \rightarrow \infty$ . Значения  $\delta(V, n)$  будут практически удовлетворительными при выполнении неравенства

$$V \geq n \cdot 10^k, \quad (7)$$

где  $k \in \{3, 4, 5, \dots\}$ . Следовательно, при удовлетворении (7) достигается адекватное размещение объектной параллельной программы на подсистеме из  $n$  машин при произвольной структуре сети связи и обе-



специвается эффективное использование  $n$  машин ВС при допустимых способах обработки информации (см. п. I).

Таким образом, если объем операций  $V$ , связанных с решением задачи, на несколько порядков превышает число машин  $n$ , на котором должна выполняться программа, то достигается эффективное функционирование системы.

Задачу, для которой выполняется (7), будем называть сложной, или трудоемкой, или с большим объемом вычислений. Сложность задачи будем характеризовать количеством операций, которые необходимо выполнить при ее решении. Задача тем сложнее, чем больше  $V$ . Задачу, которая имеет небольшой объем вычислений, и, следовательно, не допускает эффективное распараллеливание, будем называть простой. Простая задача требует для своего решения одну машину.

3. Режим функционирования ВС. В зависимости от сложности задач и характера их поступления можно выделить следующие основные режимы работы ВС: 1) решение одной сложной задачи, 2) обработку набора задач, 3) обслуживание потока задач.

Первый режим - монопрограммный, т.е. для решения задачи используются все ресурсы ВС. Задача представляется в виде параллельной программы, число ветвей в которой либо фиксировано, либо допускает варьирование в заданном диапазоне. В качестве единицы ресурса выступает элементарная машина ВС. Все машины используются для решения задачи. Если максимальное число ветвей в параллельной программе меньше общего числа ЭМ в системе, то "избыточные" машины используются для повышения надежности функционирования ВС.

Второй и третий режимы функционирования ВС относятся к мультипрограммным. При работе ВС в этих режимах одновременно решается несколько задач, следовательно, ресурсы системы делятся между несколькими задачами.

При организации функционирования ВС в случае набора задач учитывается не только количество задач, но и их параметры: число ветвей в программе (точнее число машин, на которых она будет выполняться), время решения или вероятностный закон распределения времени решения и др. Алгоритмы организации функционирования ВС [3] задают распределение задач по машинам и последовательность выполнения задач на каждой машине. В результате становится известным, в каком промежутке времени и на каких машинах (или на какой подсистеме) будет решаться любая задача набора. Этот режим безус-

ловно является обобщением режима решения сложной задачи на ВС. Вместе с тем, с позиций теории осуществимости решения задач, этот режим, без нарушения общности, сводится к первому. В самом деле, осуществимость решения любой задачи набора зависит от параметров только заданной для нее подсистемы, а анализ эффективности работы последней ни чем не отличается от анализа всей системы при решении одной сложной задачи.

Третий режим - обслуживание потока задач на ВС принципиально отличается от обработки задач набора: задачи поступают в случайные моменты времени, их параметры случайны, следовательно, детерминированный выбор подсистем для решения тех или иных задач исключен. Вместе с тем анализ производительности подсистем остается прежним. Поэтому в дальнейшем мы ограничимся рассмотрением только режима решения одной сложной задачи на ВС.

4. Показатели производительности ВС. Возможность программирования структуры ВС позволяет организовать решение одной сложной задачи на "схемах", названных системами со структурной избыточностью и живучими ВС [4]. В первом случае число параллельных ветвей в программе фиксировано и равно  $n$ , а во втором случае это число варьируемо в пределах от  $n$  до  $N$ , где  $N$  - общее число ЭМ в ВС,  $n \leq N$ . Следовательно, ВС со структурной избыточностью можно рассматривать как частный случай живучих ВС. Дальнейшее изложение будет вестись применительно к живучим системам.

А. В случае живучих ВС все исправные ресурсы - машины - могут быть включены в решение сложной задачи. Будем говорить, что система находится в состоянии  $k \in E = \{0, 1, \dots, N\}$ , если в ней имеется  $k$  исправных ЭМ. Теоретически (на основе анализа параллельных алгоритмов) и экспериментально (путем реализации параллельных программ сложных задач на действующих ВС) показано [2-5], что производительность системы, находящейся в состоянии  $k \in E$ , равна

$$\Omega(k) = A_k \cdot k \cdot \omega, \quad (8)$$

где  $\omega$  - показатель производительности элементарной машины,  $A_k$  - коэффициент. В качестве  $\omega$  может выступать номинальное (2), среднее (3) и среднее эффективное (5) быстроедействие машины.

Линейная зависимость производительности ВС от числа исправных ЭМ - следствие применения методики структурного глобального или крупноблочного распараллеливания сложных задач. В самом деле, суть методики - требование расчленять задачу на крупные блоки-

подзадачи, между которыми существует слабая связность. Говоря иначе, методика предписывает организацию таких параллельных ветвей алгоритма решения сложной задачи, общее число операций в которых много больше числа операций настройки структуры ВС, обмена информацией между ветвями, т.е. методика обеспечивает выполнение неравенства (7). Ясно, что это приводит к значению коэффициента  $A_k$ , близкого к единице. Практически (на основе обработки результатов решения сложных задач различных классов на одномерных и двумерных сосредоточенных и распределенных ВС) установлено, что коэффициент  $A_k$  не менее единицы ( $A_k \geq 1$ ). Это неравенство - следствие "эффекта памяти" (в ВС сокращается число обращений к внешним запоминающим устройствам, быстродействие которых меньше скорости передачи информации по каналам связи между ЭМ) и применения новых методов решения сложных задач, не реализуемых в ЭВМ.

Формула (8) остается верной и при работе ВС в мультипрограммных режимах. В самом деле, в системах с программируемой структурой, в отличие от матричных ВС, каждая элементарная машина имеет свое устройство управления. Следовательно, в таких ВС имеется возможность одновременной реализации нескольких программ, причем каждой - на своей части системы, т.е. на своей подсистеме. В пределах каждой ЭМ можно заставить решать свою задачу. Накладные расходы, связанные с мультипрограммной работой ВС, зависят от методов и алгоритмов организации функционирования. Созданные методы и алгоритмы (см., например, [3]) эффективно реализуются на ЭВМ и системах (не требуют большой памяти и значительного машинного времени) и обеспечивают высокую эффективность использования технических ресурсов. Можно ожидать, что при мультипрограммной работе системы с программируемой структурой будет иметь место:  $A_k \geq 0,8$ .

Таким образом, реальная производительность ВС будет не менее 80% от суммарного быстродействия (причем как для режима решения одной сложной задачи, так и для мультипрограммных режимов). Заметим, что такой уровень достигнут в ILLIAC-IV только в режиме реализации одной параллельной программы на всех 64-х элементарных процессорах. При реализации программ, имеющих число ветвей  $l < 64$ , будут простаивать  $(64-l)$  элементарных процессоров.

Б. Математическое ожидание  $\mathcal{N}(i, t)$  числа исправных машин [3] (при условии, что в начальный момент времени было исправно  $i \in E$  ЭМ) достаточно точно говорит об уровне потенциальной производительности ВС в момент времени  $t \geq 0$ . Этот уровень оценивается величиной

$$\alpha(t) = \mathcal{K}(i, t) \omega, \quad (9)$$

где  $\omega$  – показатель производительности ЭМ. Формула (9) тем точнее оценивает производительность ВС, чем больше общее число машин в системе, на практике достаточно иметь  $N \geq 10$ .

Для сложных задач допустимо составление адаптирующихся параллельных программ, в которых нижняя грань  $n$  для числа параллельных ветвей не превышает ожидаемое число исправных машин в любой момент времени, т.е. в которых  $n \leq \mathcal{K}(i, t)$ . Тогда осуществимость решения сложной задачи на живучей ВС целесообразно оценивать функцией

$$\alpha(i, t) = 1 - \exp\left[-\beta \int_0^t \mathcal{K}(i, \tau) d\tau\right], \quad (10)$$

где  $\beta$  – интенсивность решения задач на одной ЭМ.

Из определения (10) видно, что функция  $\alpha(i, t)$  является вероятностью того, что сложная задача, представленная адаптирующейся параллельной программой, будет решена за время  $t$  на ВС, начавшей функционировать в состоянии  $i \in E$  и использующей для решения задачи все исправные ЭМ.

В условиях коммерческого ВЦ режим работы вычислительной системы стационарен, поэтому достаточно вспомнить [3], что

$$\mathcal{K} = \lim_{t \rightarrow \infty} \mathcal{K}(i, t) = \begin{cases} N\mu / (\lambda + \mu), & \text{если } N\lambda \leq m\mu, \\ m\mu / \lambda - \text{в противном случае,} \end{cases}$$

где  $\lambda$  и  $\mu$  – интенсивности соответственно отказов и восстановления одной ЭМ,  $m$  – число бригад обслуживания ЭМ, обеспечить необходимую производительность бригад обслуживания (т.е.  $N\lambda \leq m\mu$ ) и оценивать производительность по простейшей формуле:

$$\alpha = \omega N\mu / (\lambda + \mu). \quad (11)$$

При стационарном режиме работы ВС вероятность решения задачи  $\alpha(t) = 1 - \exp(-\beta \mathcal{K} \cdot t)$ . (12)

Функции (10) и (12) называются функциями осуществимости решения сложной задачи на живучей ВС или функциями потенциальной осуществимости. При этом  $\alpha(i, t)$  и  $\alpha(t)$  характеризуют осуществимость соответственно в переходном и стационарном режимах работы системы. Считают, что решение сложной задачи осуществимо на живучей ВС

в промежутке времени  $[0, t)$ , если выполняются неравенства  $\Phi(i, t) \geq \Phi^0$ ,  $t \leq t^0$  для переходного режима и неравенства  $\Phi(t) \geq \Phi^0$ ,  $t \leq t^0$  для стационарного режима функционирования системы. Величины  $\Phi^0$  и  $t^0$  называются порогами потенциальной осуществимости решения сложной задачи на ВС.

Итак, введены показатели, характеризующие производительность ЭВМ и вычислительных систем. Предложены нетрудеёмкие (пригодные для инженерной практики) методы расчета показателей производительности вычислительных средств.

#### Л и т е р а т у р а

1. ГЛУШКОВ В.М. Синтез цифровых автоматов. - М.: Физматгиз, 1962. - 476 с.
2. ЕВРЕЙНОВ Э.В., КОСАРЕВ Д.Г. Однородные универсальные вычислительные системы высокой производительности. - Новосибирск: Наука, 1966. - 308 с.
3. ЕВРЕЙНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. - Новосибирск: Наука, 1978. - 319 с.
4. ДИМИТРИЕВ Д.К., ХОРОШЕВСКИЙ В.Г. Вычислительные системы из мини-ЭВМ. - М.: Радио и связь, 1982. - 304 с.
5. ХОРОШЕВСКИЙ В.Г. Вычислительная система МИКРОС. - Новосибирск, 1983. (Препринт/ Ин-т математики СО АН СССР: ОВС-19).
6. ПРАНГИШВИЛИ И.В., ВИЛЕНКИН С.Я., МЕДВЕДЕВ И.Л. Параллельные вычислительные системы с общим управлением. - М.: Энергоатомиздат, 1983. - 313 с.
7. Parallel Computations./Ed. by G.Rodrigue.- New York:Academic Press,1982.- 403 p.
8. KAI Hwang, SHUN-PIAO Su, LIONEL M.Ni. Vector Computer Architecture and Processing Techniques. - In: Advances in Computers. Ed. by M.C.Yovits. V.20.- New York: Academic Press,1981, p.116-191.

Поступила в ред.-изд.отд.  
2 октября 1984 года